

# Commande des systèmes par ordinateur

Yves Briere  
ENSICA

Notes de cours  
2007



# TABLE DES MATIERES

Table des matières .....	1
1. Introduction .....	1
2. Rappels et prérequis .....	3
3. Signaux et systèmes discrets .....	5
3.1 Signaux discrets .....	5
3.1.1 Transformée en z .....	5
3.1.2 Signaux types .....	7
3.2 Systèmes discrets .....	9
3.2.1 Fonction de transfert discrète, définition .....	9
3.2.2 Notation "opérateur de retard $q^{-1}$ " .....	10
3.2.3 Réponse impulsionnelle .....	11
3.2.4 Causalité .....	11
3.2.5 Réponses indicielle .....	11
3.2.6 Régime stationnaire (gain statique) .....	12
3.2.7 Pôles, zéros, stabilité .....	12
3.2.8 Systèmes du premier ordre .....	15
3.2.9 Systèmes du second ordre .....	16
3.2.10 Calcul itératif de la réponse d'un système .....	18
3.2.11 Représentation par équation d'état .....	18
3.3 Réponse fréquentielle des systèmes échantillonnés .....	19
3.3.1 Diagramme de Bode d'un système échantillonné .....	19
3.3.2 Où l'on voit apparaître le théorème de Shannon .....	20
4. Systèmes continus échantillonnés .....	23
4.1 Structure des systèmes commandés par ordinateur .....	23
4.2 Échantillonnage des systèmes continus .....	24
4.3 Effets de l'échantillonnage .....	25
4.4 Où l'on revoit le théorème de Shannon .....	26
4.5 Discrétisation de fonction de transfert continues .....	29
4.5.1 Approximation numérique d'équation différentielle .....	30
4.5.2 Superposition des réponses indicielles .....	32
4.5.3 Correspondances pôles - zéros .....	33
4.5.4 Bilan .....	33
5. Identification des systèmes échantillonnés .....	35
5.1 Identification d'un modèle ARX par la méthode des moindres carrés récurrents .....	35
5.2 L'identification paramétrique en général et en pratique .....	37

5.2.1	Choix du signal d'excitation, traitement des mesures .....	37
5.2.2	Choix d'une structure de modèle.....	38
5.2.3	Choix d'un critère à minimiser.....	38
5.2.4	Choix d'un algorithme.....	38
5.2.5	Validation.....	39
5.2.6	Utilisation de logiciels .....	39
5.3	Identification "simple" .....	39
5.4	Identification non paramétrique .....	41
5.4.1	Estimation de la réponse impulsionnelle .....	41
5.4.2	Estimation de la réponse fréquentielle.....	42
6.	Systèmes bouclés .....	45
6.1	Fonction de transfert en boucle fermée .....	45
6.2	Erreur statique .....	46
6.3	Rejet de perturbations.....	46
6.4	Stabilité des systèmes à simple gain de rétroaction.....	47
6.5	Analyse des systèmes bouclés dans le domaine fréquentiel : stabilité.....	49
6.5.1	Théorème de Nyquist dans le cas où le système est stable en boucle ouverte (critère du revers) .....	50
6.5.2	Théorème de Nyquist dans le cas où le système est instable en boucle ouverte .....	50
6.5.3	Interprétations du lieu de Nyquist.....	50
7.	Méthodes de régulation des systèmes échantillonnés .....	53
7.1	Introduction .....	53
7.2	Structure d'un système commandé par ordinateur, hypothèses.....	54
7.2.1	Discretisation, choix de la période d'échantillonnage.....	54
7.2.2	Discretisation d'un correcteur continu .....	55
7.3	Régulateur PID numérique .....	55
7.3.1	Cas 1 .....	55
7.3.2	Cas 2 .....	58
7.4	Placement de pôles .....	59
7.4.1	Choix des pôles en boucle fermée .....	60
7.4.2	Régulation : calcul de R et S.....	60
7.4.3	Poursuite : calcul de T .....	61
7.5	Poursuite et régulation à objectifs indépendants .....	62
8.	Technologie de la commande par ordinateur.....	65
8.1	Calculateurs et interfaces numériques / analogiques.....	65
8.2	Transport de l'information : les bus et les protocoles .....	67
8.3	Capteurs et actionneurs "intelligents", "smart sensors" .....	68
8.4	La philosophie "hardware in the loop" .....	68
8.5	Le prototypage rapide en automatique .....	68
9.	Annexe .....	69
9.1	Annexe 1 : Résolution de l'équation polynomiale de Bezout.....	69
10.	Bibliographie.....	73
11.	Index.....	1





# 1. INTRODUCTION



Figure 1 : commande d'un systèmes de télémédecine par calculateur, vu par Claude Serre





## 2. RAPPELS ET PREREQUIS

Ce polycopié ne comporte pas de rappels sur les systèmes continus. S'adressant aux étudiants en deuxième année de l'ENSICA il suppose connu les enseignements de première année et de seconde année en automatique et traitement du signal.

En première année le cours de théorie du signal [2] a rendu familier les notions de :

- transformée de Fourier, transformée de Laplace
- corrélation, convolution

Le cours de traitement numérique du signal [3] a abordé en plus :

- la discrétisation et la quantification des signaux.
- la transformée en z et la transformée de Fourier discrète.
- les filtres numériques RIF et RII.

En automatique [4] les bases de la représentation et de l'analyse des systèmes continus ont été posées. On ne reviendra pas dans ce cours sur les notions de :

- modèles mathématique des systèmes linéaires à temps continus à temps invariant (LTI) : fonction de transfert, équation d'état, pôles, zéros et dynamique.
- commandabilité et observabilité.
- stabilité des systèmes en boucle ouverte : pôles, critères de Routh-Hurwitz.
- stabilité des systèmes bouclés : critère de Nyquist, critère de Bode.
- lieu des racines.

En deuxième année les notions suivantes sont acquises [5] :

- synthèse des correcteurs par méthodes fréquentielles (avance et retard de phase, PID)
- commande par retour d'état (placement de pôles), commande Linéaire Quadratique (LQG)
- reconstruction de l'état par Observateur, observateur optimal de Kalman.



## 3. SIGNAUX ET SYSTEMES DISCRETS

### 3.1 Signaux discrets

#### 3.1.1 Transformée en z

Pour une approche mathématiquement rigoureuse de représentation des signaux discrets par la transformée en z consulter [3] et sa bibliographie. Dans ce cours on se cantonne à une approche utilitariste des outils et méthodes mathématiques tout en essayant de ne pas contredire fondamentalement la rigueur mathématique.

Un signal discret  $s(k)$  est une séquence de nombres réels appelés échantillons. L'indice  $k$  est généralement associé à un instant d'échantillonnage :  $t = k.T_e$ . La période d'échantillonnage  $T_e$  est supposée constante. Par définition la transformée en z du signal  $s(k)$  est la fonction de la variable  $z$ , notée abusivement  $s(z)$  :

$$s(z) = Z(s(k)) = \sum_{k=0}^{\infty} s(k) \cdot z^{-k} \quad (1)$$

La transformée définie ici est plus précisément appelée transformée unilatérale puisque les termes correspondant à  $k < 0$  ne sont pas pris en compte. Ceci n'a pas d'importance puisque les signaux que nous utilisons sont systématiquement causaux<sup>1</sup> :  $s(k) = 0$  pour  $k < 0$ .

Il y a bien évidemment un lien avec la transformée de Laplace et la transformée de Fourier (voir [3])

La fonction de la variable complexe  $s(z)$  est une série et ne converge pas forcément pour toute valeur de  $z$ . En toute rigueur il faudrait associer le rayon de convergence à chaque fonction en  $z$  :  $\{s(z) \text{ converge pour } |z| > \alpha\}$ .

Les automaticiens pour qui la transformée en z n'est bien souvent qu'une façon bien commode de regrouper des termes de récurrence utilisent souvent la pseudo-variable<sup>2</sup>  $q$ .

A partir d'une transformée en z  $s(z)$  il est possible de retrouver la séquence  $s(k)$  correspondante en appliquant la transformation inverse, dont on ne donnera pas la formule ici. A la place de faire appel à cette formule compliquée il est possible d'utiliser d'autres méthodes pour retrouver la séquence  $s(k)$  lorsque la fonction  $s(z)$  est donnée sous forme de fraction rationnelle :

<sup>1</sup> Pour une justification du terme « causal » voir page 11

<sup>2</sup> Voir page 10

- division polynomiale
- décomposition de la fraction rationnelle en éléments simples puis utilisation de transformations élémentaires telles que celles données dans le Tableau 2

La propriété fondamentale de la transformée en z est qu'elle est linéaire (la transformée en z d'une composition linéaire de signaux est la composition linéaire de la transformée en z de chaque signal). Les autres propriétés fondamentales sont données dans le Tableau 2.

Tableau 1 : propriétés de la transformée en z

1	$s(z) = Z(s(k)) = \sum_{k=0}^{\infty} s(k) \cdot z^{-k}$
2	$Z(\alpha \cdot s_1(k) + \beta \cdot s_2(k)) = \alpha \cdot Z(s_1(k)) + \beta \cdot Z(s_2(k))$
3	$Z(k \cdot s(k)) = -z \cdot \frac{ds(z)}{dz}$
4	$Z(c^k \cdot s(k)) = s\left(\frac{z}{c}\right)$
5	$Z(s(k-1)) = s(k=-1) + z^{-1} \cdot s(z)$
6	$Z(s(k+1)) = z \cdot s(z) - z \cdot s(k=0)$
7	$s(k=0) = \lim_{z \rightarrow \infty} (s(z))$
8	$s(k \rightarrow \infty) = \lim_{z \rightarrow 1} \left( \frac{z-1}{z} \cdot s(z) \right)$

Remarque 1 : la propriété 5 ne prend pas en compte le fait que le signal est causal : en pratique  $s(k=-1) = 0$ .

Remarque 2 : la propriété 5 justifie le fait que le terme  $z^{-1}$  soit appelé "opérateur de retard unitaire", cette propriété est aussi appelée "théorème du retard".

Remarque 3 : les propriétés 7 et 8 sont données sous réserve de convergence.

Transformées inverses

### Exemple 1 : transformée inverse par division polynomiale

On considère un signal  $s(k)$  dont la transformée en z est donnée par la fraction rationnelle :

$$S(z) = \frac{0.1 \cdot z^2 + 0.1 \cdot z}{z^3 - 2.6 \cdot z^2 + 2.2 \cdot z - 0.6} \quad (3.1)$$

L'objectif est d'obtenir  $S(z)$  sous forme polynomiale :

$$S(z) = x_0 + x_1 \cdot z^{-1} + x_2 \cdot z^{-2} + \dots \text{etc.} \tag{3.2}$$

Cette suite est généralement infinie, la réponse temporelle est donnée par la succession des termes  $\{x_i\}$ . En traitant l'équation (3.1) comme une simple division l'algorithme de division classique (celui appris en CM2) permet d'obtenir par récurrence tous les termes de la suite (3.2) :

$0.1 \cdot z^2$	$+ 0.1 \cdot z$				$z^3 - 2.6 \cdot z^2 + 2.2 \cdot z - 0.6$
$0.1 \cdot z^2$	$- 0.26 \cdot z$	$+ 0.22$	$- 0.06 \cdot z^{-1}$		
$0$	$0.36 \cdot z$	$- 0.22$	$0.06 \cdot z^{-1}$		
	$0.36 \cdot z$	$- 0.936$	$0.792 \cdot z^{-1}$	$- 0.216 \cdot z^{-2}$	
$0$	$0.716$	$- 0.732 \cdot z^{-1}$	$\dots$	$\dots$	

Les éléments successifs de la suite  $s(k)$  sont donc  $s(k) = \{0, 0.1, 0.36, 0.716, \dots\}$ . Pour obtenir les termes suivants il faudrait continuer la division.

**Exemple 2 : transformée inverse par décomposition en éléments simples**

La deuxième méthode présentée consiste à décomposer la fraction rationnelle (3.1) en éléments simples puis d'utiliser les tables de transformées en  $z$  (Tableau 2). Pour notre exemple cela donne :

$$S(z) = \frac{0.1 \cdot z^2 + 0.1 \cdot z}{z^3 - 2.6 \cdot z^2 + 2.2 \cdot z - 0.6} = \frac{-z}{z-1} + \frac{0.5 \cdot z}{(z-1)^2} + \frac{z}{z-0.6}$$

La réponse temporelle est donc composée des trois éléments :

$$s(k) = -(1 \ 1 \ 1 \ \dots) + 0.5 \cdot (1 \ 2 \ 3 \ \dots) + (1 \ 0.6 \ 0.6^2 \ \dots)$$

$$\text{soit : } s(k) = \begin{cases} s(0) = -1 + 0.5 + 1 \\ s(2) = -1 + 0.5 \cdot 2 + 0.6 \cdot 0.5 \\ s(3) = -1 + 0.5 \cdot 3 + 0.6^2 \\ s(4) = \dots \end{cases}$$

**3.1.2 Signaux types**

Le Tableau 2 donne les transformée en  $z$  de certains signaux fondamentaux. La Figure 2 représente l'allure de quelques signaux types.

Tableau 2 : transformée en z élémentaires

$s(k)$	$S(z)$
signal temporel	Transformée en z
$\delta(k)$ : impulsion unité	1
$u(k)$ : échelon unité	$\frac{z}{z-1}$
$k \cdot u(k)$	$\frac{z}{(z-1)^2}$
$c^k \cdot u(k)$	$\frac{z}{z-c}$
$k \cdot a^k \cdot u(k)$	$\frac{a \cdot z}{(z-a)^2}$
$\sin(\omega \cdot k) \cdot u(k)$	$\frac{z \cdot \sin(\omega)}{z^2 - 2 \cdot z \cdot \sin(\omega) + 1}$
$\cos(\omega \cdot k) \cdot u(k)$	$\frac{z \cdot (z - \cos(\omega))}{z^2 - 2 \cdot z \cdot \sin(\omega) + 1}$
$a^k \cdot \sin(\omega \cdot k) \cdot u(k)$	$\frac{a \cdot z \cdot \sin(\omega)}{z^2 - 2 \cdot a \cdot \cos(\omega) \cdot z + a^2}$
$a^k \cdot \cos(\omega \cdot k) \cdot u(k)$	$\frac{z \cdot (z - a \cdot \cos(\omega))}{z^2 - 2 \cdot a \cdot \cos(\omega) \cdot z + a^2}$

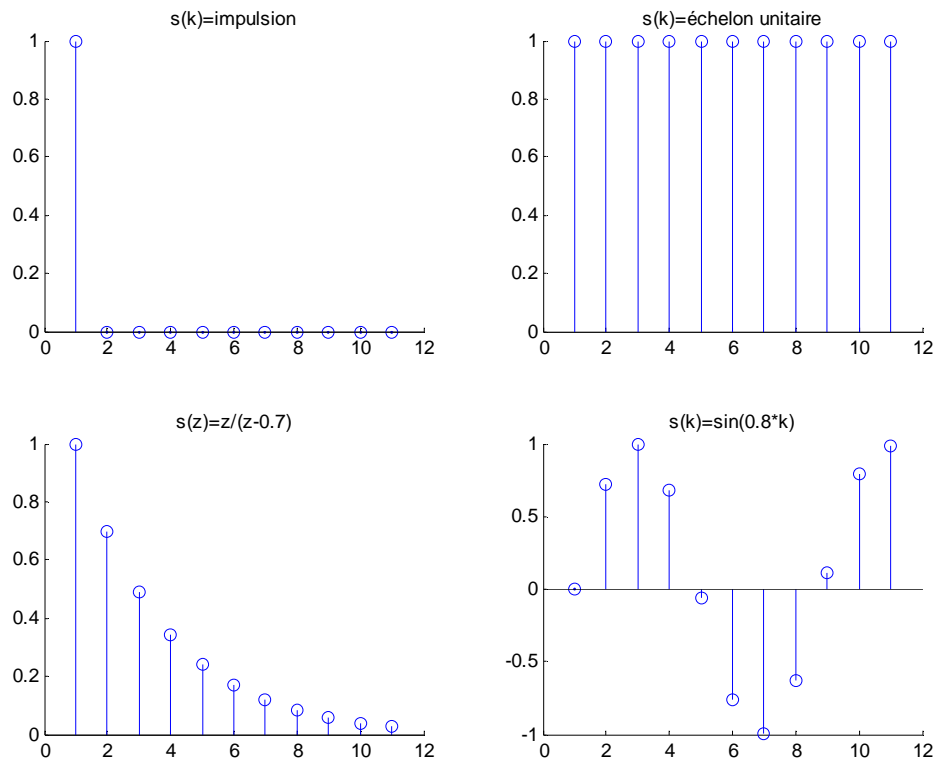


Figure 2 : réponses de quelques signaux types

### 3.2 Systèmes discrets

#### 3.2.1 Fonction de transfert discrète, définition

On considère maintenant un système discret linéaire et invariant dans le temps (LTI) représenté sur la Figure 3.

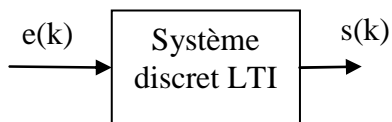


Figure 3 : système discret, notations

La relation entre la séquence d'entrée et la séquence de sortie est linéaire et définie par un ensemble de coefficients constants  $a_{1:n}$  et  $b_{1:m}$  :

$$\begin{aligned} s(k) + a_1 \cdot s(k-1) + a_2 \cdot s(k-2) + \dots + a_n \cdot s(k-n) = \\ b_0 \cdot e(k) + b_1 \cdot e(k-1) + b_2 \cdot e(k-2) + \dots + b_m \cdot e(k-m) \end{aligned} \quad (2)$$

On peut appliquer la transformée en  $z$  à chaque membre de l'équation puis en appliquant en récurrence la propriété 5 (théorème du retard) on obtient la fonction de transfert discrète du système :

$$\frac{s(z)}{e(z)} = F(z) = \frac{b_0 + b_1 \cdot z^{-1} + b_2 \cdot z^{-2} + \dots + b_m \cdot z^{-m}}{1 + a_1 \cdot z^{-1} + a_2 \cdot z^{-2} + \dots + a_n \cdot z^{-n}} \quad (3)$$

Remarque 1 : la fonction de transfert précédente est normalisée en prenant  $a_0 = 1$ .

Remarque 2 : on remarquera que la fonction de transfert est donnée en  $z^{-1}$  plutôt qu'en  $z$ . Cela est justifié par le fait que celle-ci traduit l'équation d'évolution du système physique causal (2).

Le script Matlab permettant de créer un objet Matlab correspondant à une fonction de transfert discrète est donné ci-dessous.

#### Matlab

```
% Création d'un système linéaire discret :
Te=0.1;
sysd = tf([b0 b1 b2],[1 a0 a2 a3],Te,'variable','z^-1');
```

On notera l'utilisation de l'argument 'variable', 'z^-1'.

### 3.2.2 Notation "opérateur de retard $q^{-1}$ "

La fonction de transfert est une fonction de la variable complexe  $z$ . Il faudrait donc en toute rigueur l'associer à son rayon de convergence. En pratique on ne s'embarrasse jamais de préciser cette zone de convergence. C'est pourquoi on trouve souvent la notation utilisant l'opérateur  $q^{-1}$  de décalage à la place de la variable complexe  $z^{-1}$  (4).  $F(q)$  n'est alors plus à proprement parler une fonction mais on lui appliquera les mêmes manipulations algébriques qu'à  $F(z)$ .

$$F(q) = \frac{b_0 + b_1 \cdot q^{-1} + b_2 \cdot q^{-2} + \dots + b_m \cdot q^{-m}}{1 + a_1 \cdot q^{-1} + a_2 \cdot q^{-2} + \dots + a_n \cdot q^{-n}} \quad (4)$$

Attention : selon les auteurs l'opérateur qui remplace  $z^{-1}$  est noté  $q$  (Matlab) ou  $q^{-1}$  (Landau [1]). Dans la suite du document c'est cette dernière convention qui sera appliquée.



L'utilisation de l'opérateur de retard  $q^{-1}$  à la place de la variable complexe  $z^{-1}$  permet de traiter du cas des fonctions de transfert à coefficients non constant, cas non prévu par la transformation en  $z$ .

### 3.2.3 Réponse impulsionnelle

Essayons d'appliquer au système un signal d'entrée particulier : une impulsion unitaire. La transformée en  $z$  du système est obtenue en multipliant la fonction de transfert par la transformée en  $z$  de l'impulsion unitaire : 1. On obtient :

$$s(z) = \frac{b_0 + b_1 \cdot z^{-1} + b_2 \cdot z^{-2} + \dots + b_m \cdot z^{-m}}{1 + a_1 \cdot z^{-1} + a_2 \cdot z^{-2} + \dots + a_n \cdot z^{-n}} \times 1 \quad (5)$$

Cela justifie le fait que la fonction de transfert du système peut aussi s'appeler "réponse impulsionnelle".

#### Matlab

```
% Réponse impulsionnelle d'un système :
[y,t] =impulse(sysd);
```

### 3.2.4 Causalité

Un système est dit causal si les sorties ne dépendent que du passé et non pas du futur. La conséquence est que la réponse impulsionnelle de  $F(z)$  ne peut pas comporter de terme non nuls pour  $k < 0$ . Cela justifie la confusion sémantique entre "système causal" et "signal causal", c'est-à-dire ne comportant pas de termes non nuls pour  $k < 0$ .

Attention : à la différence des systèmes continus un système physique (causal) discret peut très bien être modélisé avec un degré  $m$  supérieur au degré  $n$  : le système peut avoir plus de zéros que de pôles.

La transformée en  $z$  représentant des signaux échantillonnés elle peut tout aussi bien représenter les mesures prises à différents endroits d'une poutre. Le temps étant remplacé ici par une abscisse la notion de causalité ne s'applique pas ! Attention : beaucoup des propriétés de systèmes échantillonnés que l'on étudie sont liés à la causalité, prudence donc si l'on applique ces méthodes à d'autres domaines !

### 3.2.5 Réponses indicielle

Pour caractériser des systèmes on s'intéresse souvent à leur réponse impulsionnelle ou, plus souvent, à leur réponse indicielle. Pour calculer la réponse

à un échelon d'un système il faut donc multiplier la fonction de transfert par la transformée en z d'un échelon unitaire :

$$s(z) = \frac{b_0 + b_1 \cdot z^{-1} + b_2 \cdot z^{-2} + \dots + b_m \cdot z^{-m}}{1 + a_1 \cdot z^{-1} + a_2 \cdot z^{-2} + \dots + a_n \cdot z^{-n}} \times \frac{1}{1 - z^{-1}} \quad (6)$$

Il faudrait ensuite calculer la transformée en z inverse de s(z). En pratique, si l'on doit faire le calcul à la main on décompose en éléments simples puis on utilise des tables. Avec Matlab c'est plus simple. Voici par exemple le script pour obtenir les 20 premiers échantillons de la réponse indicielle d'un système :

**Matlab**

```
% Réponse indicielle d'un système :
[y,t] =step(sysd,20);
```

### 3.2.6 Régime stationnaire (gain statique)

Le gain statique est obtenu en appliquant une entrée indicielle et en calculant la valeur du régime permanent. En appliquant le théorème de la valeur finale il vient :

$$F_0 = \lim_{z \rightarrow 1} \left( \frac{z-1}{z} \cdot \frac{b_0 + b_1 \cdot z^{-1} + b_2 \cdot z^{-2} + \dots + b_m \cdot z^{-m}}{1 + a_1 \cdot z^{-1} + a_2 \cdot z^{-2} + \dots + a_n \cdot z^{-n}} \times \frac{1}{1 - z^{-1}} \right) \quad (7)$$

Soit :

$$F_0 = \frac{\sum_{i=0}^{i=m} b_i}{1 + \sum_{i=1}^{i=n} a_i} \quad (8)$$

Le gain statique est obtenu donc simplement en faisant le quotient de la somme des coefficients.

### 3.2.7 Pôles, zéros, stabilité

Comme pour les systèmes continus la dynamique et la stabilité des systèmes dépend des propriétés du dénominateur de la fonction de transfert.

$$F(z) = \frac{e_0 \cdot (1 + e_1 \cdot z^{-1}) \cdot (1 + e_2 \cdot z^{-1}) \cdots (1 + e_m \cdot z^{-1})}{(1 + c_1 \cdot z^{-1}) \cdot (1 + c_2 \cdot z^{-1}) \cdots (1 + c_n \cdot z^{-1})} \quad (9)$$

Les coefficients  $c_i$  sont les pôles de la fonction de transfert, les coefficients  $e_i$  sont les zéros. La fonction de transfert étant au départ composée de coefficients purement réels les coefficients  $c_i$  sont soit réels soit imaginaires par paires conjuguées.

Une fonction de transfert est donc aussi bien définie par la donnée de ses pôles et de ses zéros. Le script Matlab pour créer un système ainsi est donné ci-dessous :

#### Matlab

```
% Système discret défini par ses pôles et zéros :
Te=0.1;
sysd=zpk([c1 c2 c3],[e1 e2],e0,Te);
```

Si l'on décompose la fonction de transfert en éléments simples on obtient une somme d'éléments du type :

$$F(z) = \frac{d_1}{1 + c_1 \cdot z^{-1}} + \frac{d_2}{1 + c_2 \cdot z^{-1}} + \cdots + \frac{d_n}{1 + c_n \cdot z^{-1}} \quad (10)$$

Un système linéaire temps invariant est dit stable si sa réponse impulsionnelle tend asymptotiquement vers 0. Compte tenu de (10) cela nécessite que chaque pôle  $c_i$  soit de module inférieur à 1.

**Propriété** : un système discret LTI est stable si et seulement si tous ses pôles ont leur modules inférieur ou égal à 1.

#### Critère algébrique attestant de la stabilité : critère de Routh

On rappelle que le critère de Routh permet de prédire si les racines d'un polynôme sont toutes à partie réelle négative, donc stables. Énonçons le théorème de Routh dans le cas d'un polynôme  $P'(w)$  :

$$P'(w) = a_0 + a_1 \cdot w + a_2 \cdot w^2 + \cdots + a_n \cdot w^n \quad (11)$$

Dans un premier temps on construit le tableau :

$$\begin{array}{cccccc|c}
 \beta_{n,0} & \beta_{n,1} & \beta_{n,2} & \beta_{n,q-1} & \beta_{n,q} & 0 & \\
 \beta_{n-1,0} & \beta_{n-1,1} & \beta_{n-1,2} & \beta_{n-1,q-1} & \beta_{n-1,q-1} & 0 & \\
 \beta_{n-2,0} & \beta_{n-2,1} & \beta_{n-2,2} & \beta_{n-2,q-1} & \beta_{n-2,q-1} & & \\
 \\ 
 \beta_{2,0} & \beta_{2,1} & 0 & & & & \\
 \beta_{1,0} & 0 & & & & & \\
 \beta_{0,0} & 0 & & & & & 
 \end{array} \quad (12)$$

Puis on vérifie que tous les premiers éléments de la première colonne ont le même signe : cela garantit que tous les pôles de  $P'(w)$  ont leur partie réelle négative.

Les éléments du tableau sont construits par récurrence ligne par ligne en appliquant les formules suivantes :

Pour  $i = n$  et  $i = n - 1$  :

$$\begin{cases} \beta_{n,j} = a_{n-2j} \\ \beta_{n-1,j} = a_{n-1-2j} \end{cases} \quad (13)$$

Et pour  $i < n-1$  :

$$\beta_{i,j} = \frac{\beta_{i+1,0} \cdot \beta_{i+2,j+1} - \beta_{i+1,j+1} \cdot \beta_{i+2,0}}{\beta_{i+1,0}} \quad (14)$$

Dans le cas qui nous intéresse, un système échantillonné, on cherche plutôt à vérifier que les pôles du polynôme  $P(z)$  sont de module inférieur à un. L'idée est d'effectuer un changement de variable  $z \rightarrow w$  pour pouvoir appliquer le critère de Routh.

$$P'(w) = (1-w)^n \cdot P\left(\frac{1+w}{1-w}\right) \quad (15)$$

Si l'on effectue la transformation (15) alors le fait que  $P'(w)$  ait toutes ses racines à partie réelle négative garantit que  $P(z)$  a toutes ses racines dans le cercle unité. Pour vérifier la stabilité de  $P(z)$  il suffit donc d'effectuer le changement de variable (15) puis d'appliquer le critère de Routh conventionnel (12 13 14).

### Critère algébrique attestant de la stabilité : critère de Jury

Le critère de Jury s'apparente au critère de Routh. Nous le donnons pour un polynôme de degré 2, 3 et 4 du type :

$$P(z) = a_0 + a_1 \cdot z + a_2 \cdot z^2 + \dots + a_n \cdot z^n \quad (16)$$

Les conditions de stabilité sont données ci-dessous (ces conditions sont données dans le cas  $a_n > 0$ , il suffit de multiplier tous les termes par  $-1$  si cette condition n'est pas remplie par  $P(z)$ ) :

$$\begin{aligned}
 n = 2 & \begin{cases} a_0 + a_1 + a_2 > 0 \\ a_0 - a_1 + a_2 > 0 \\ -a_0 + a_2 > 0 \end{cases} \\
 n = 3 & \begin{cases} a_0 + a_1 + a_2 + a_3 > 0 \\ -a_0 + a_1 - a_2 + a_3 > 0 \\ -|a_0| + a_3 > 0 \\ a_0 a_2 - a_1 a_3 - a_0^2 + a_3^2 > 0 \end{cases} \\
 n = 4 & \begin{cases} a_0 + a_1 + a_2 + a_3 + a_4 > 0 \\ a_0 - a_1 + a_2 - a_3 + a_4 > 0 \\ a_4^2 - a_0^2 - |a_0 a_3 - a_1 a_4| > 0 \\ (a_0 - a_4)^2 (a_0 - a_2 + a_4) + (a_1 - a_3)(a_0 a_3 - a_1 a_4) > 0 \end{cases}
 \end{aligned} \quad (17)$$

### 3.2.8 Systèmes du premier ordre

On donne dans la Figure 4 les réponses indicielles de systèmes du premier ordre ainsi que la localisation des pôles dans le plan complexe. Le cercle de rayon unité représente la zone dans laquelle les pôles doivent se situer pour assurer la stabilité. On remarque sur les deux premières figures l'influence de la place du zéro par rapport au point  $(1,0)$  pour la rapidité.

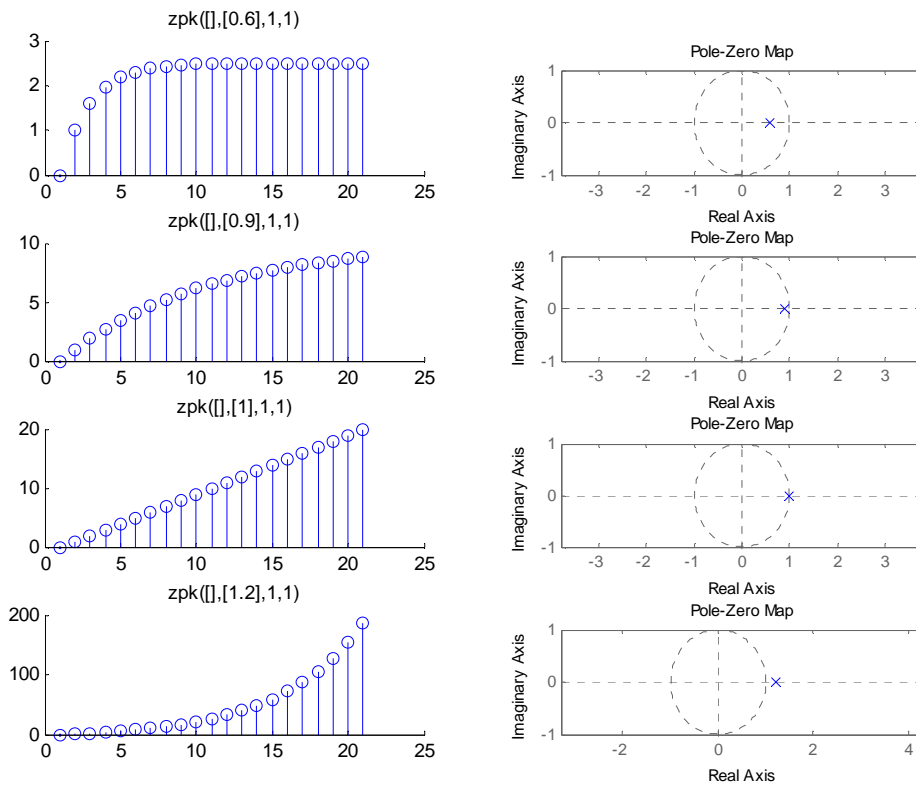


Figure 4 réponses indicielles de systèmes du premier ordre

### 3.2.9 Systèmes du second ordre

On donne dans la Figure 6 les réponses indicielles de systèmes du second ordre ainsi que la localisation des pôles dans le plan complexe. La rapidité et l'amortissement dépend de la position des pôles dans le cercle unité. La Figure 5 représente les courbes iso-amortissement et iso-pulsation des systèmes du second ordre.

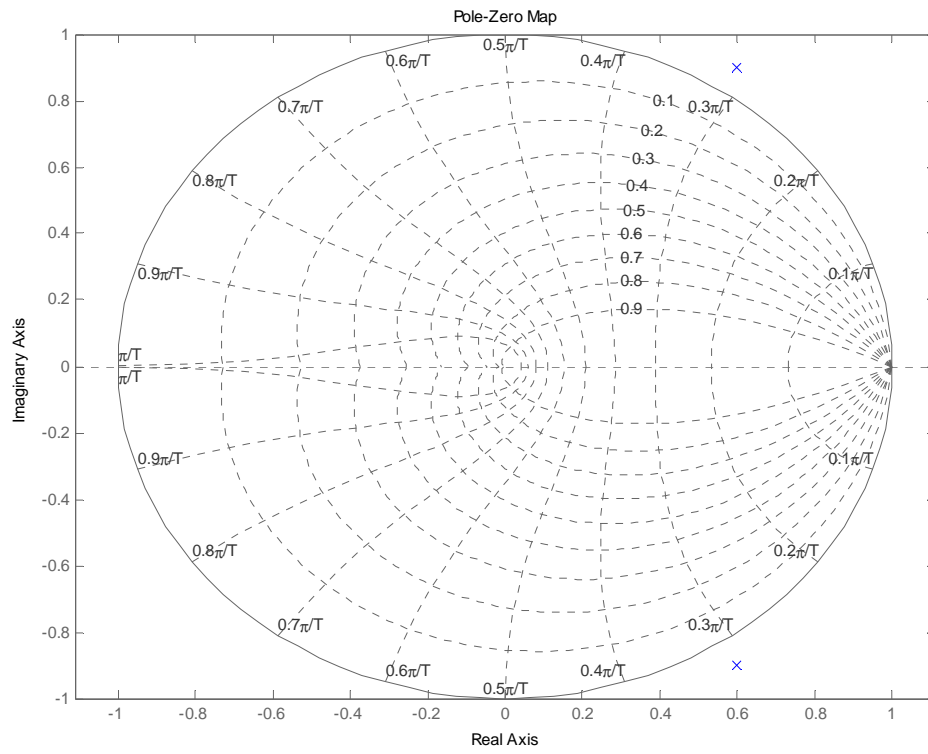


Figure 5 : courbes iso-amortissement et iso-pulsation des systèmes discrets du second ordre

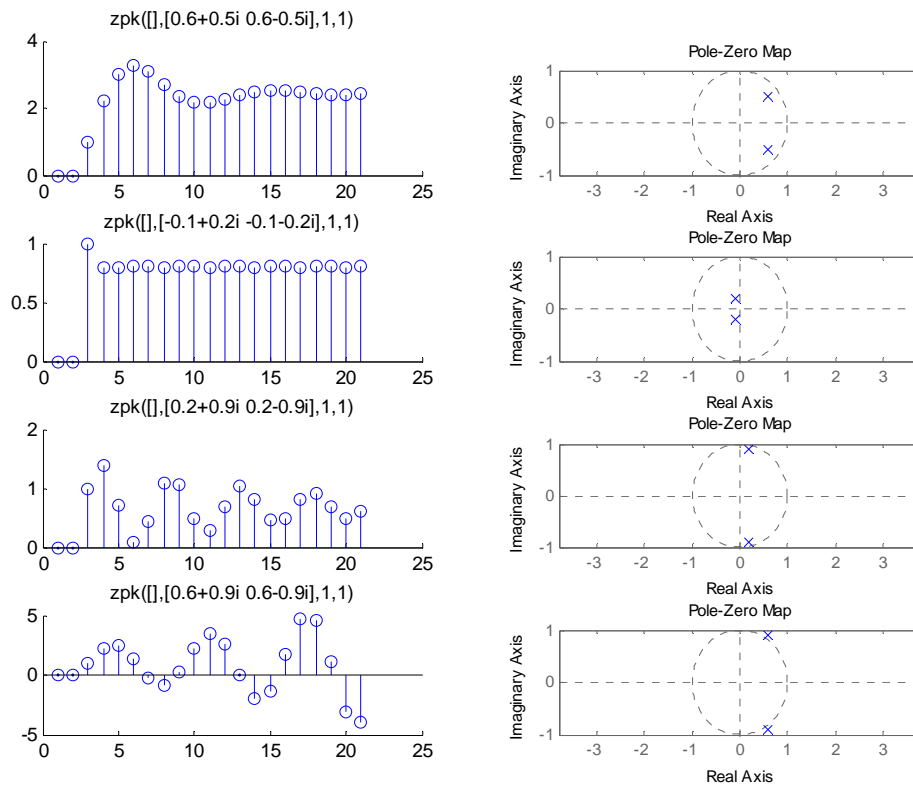


Figure 6 : réponses de systèmes du second ordre

### 3.2.10 Calcul itératif de la réponse d'un système

Si l'on dispose de la fonction de transfert d'un système et que l'on connaît les valeurs successives  $e(k)$  du signal d'entrée il est facile de calculer itérativement la sortie  $y(k)$  du système par l'équation de récurrence (2). Il faut initialiser par les  $n$  valeurs initiales de  $s(k)$ , prises généralement égales à 0.

### 3.2.11 Représentation par équation d'état

Un système discret peut aussi être représenté par un ensemble d'équations d'état comme dans le domaine continu (52). Cette représentation est naturellement adaptée aux systèmes multi-entrées multi-sorties. On définit alors un état interne  $x(k)$  ainsi qu'une fonction d'évolution  $f$  et une équation de mesure  $h$ .

$$\begin{cases} x(k+1) = f(x(k), u(k)) \\ s(k) = h(x(k), u(k)) \end{cases} \quad (18)$$



### 3.3 Réponse fréquentielle des systèmes échantillonnés

#### 3.3.1 Diagramme de Bode d'un système échantillonné

Lorsqu'on étudie la réponse fréquentielle des systèmes continus on considère des entrées particulières sinusoïdales :

$$\begin{cases} e(t) = \cos(2 \cdot \pi \cdot f \cdot t) + j \cdot \sin(2 \cdot \pi \cdot f \cdot t) = e^{j \cdot 2 \cdot \pi \cdot f \cdot t} \\ f \in [0 \quad \infty] \end{cases} \quad (19)$$

De même, pour des systèmes échantillonnés on considèrera des entrées sinusoïdales échantillonnées :

$$\begin{cases} e(k) = \cos(2 \cdot \pi \cdot f \cdot k) + j \cdot \sin(2 \cdot \pi \cdot f \cdot k) = e^{j \cdot 2 \cdot \pi \cdot f \cdot k} \\ f \in [0 \quad 1] \end{cases} \quad (20)$$

La sortie est aussi un signal échantillonné sinusoïdal de gain  $A(f)$  et de phase  $\Phi(f)$  variables et dépendant de  $f$ ; la fréquence réduite.

$$\begin{cases} s(k) = A(f) \cdot e^{j \cdot (2 \cdot \pi \cdot f \cdot k + \Phi(f))} \\ f \in [0 \quad 1] \end{cases} \quad (21)$$

Le nombre complexe, noté abusivement  $F(f)$ , d'amplitude  $A(f)$  et de phase  $\Phi(f)$  est obtenu à partir de  $F(z)$  en remplaçant  $z$  par  $e^{j \cdot 2 \cdot \pi \cdot f}$  :

$$\begin{cases} F(f) = F(z = e^{j \cdot 2 \cdot \pi \cdot f}) \\ f \in [0 \quad 1] \end{cases} \quad (22)$$

Le diagramme représentant le gain et la phase en fonction de la fréquence réduite s'appelle aussi un diagramme de Bode. L'échelle des fréquences est généralement linéaire de 0 à 0.5 (puisque le diagramme est nécessairement symétrique). La Figure 7 représente par exemple le diagramme de Bode d'un système du second ordre.

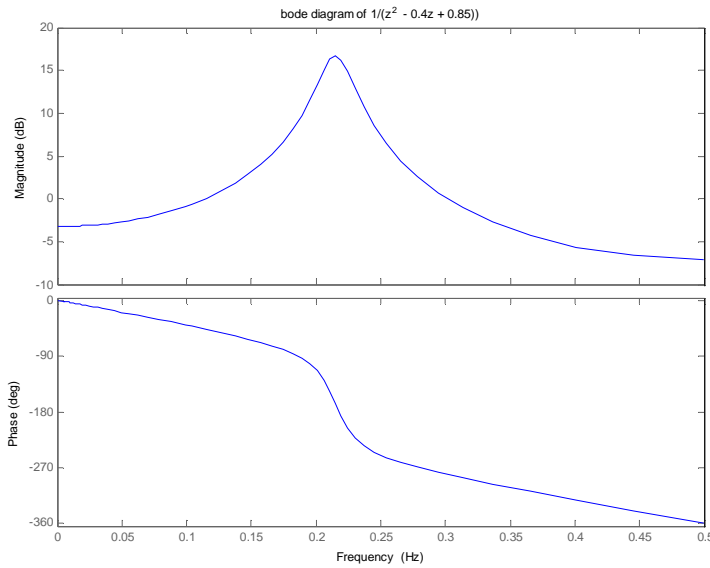


Figure 7 : exemple du diagramme de Bode d'un système discret du second ordre

### 3.3.2 Où l'on voit apparaître le théorème de Shannon

Le changement de variable  $z = e^{j2\pi f}$  n'est pas bijectif puisque  $e^{j2\pi f} = e^{j2\pi(f+k)}$  avec  $k$  entier. Si l'on fait le lien entre domaine continu et domaine fréquentiel on peut remarquer que le changement de variable  $z = e^{j2\pi f}$  fait correspondre une zone du plan complexe continu à une zone du plan complexe discret. La transformation est bijective si l'on se contente de la zone du plan complexe continu contenu entre les droites d'ordonnée 0 et 1.

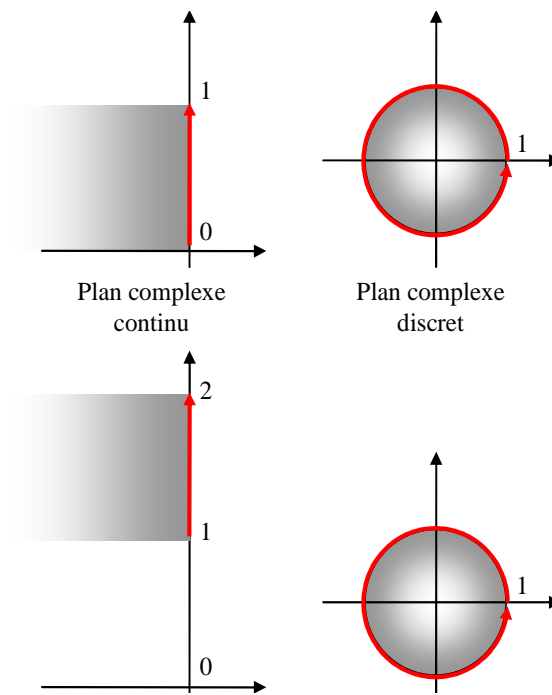


Figure 8 : correspondance entre plan complexe continu et plan complexe discret.



## 4. SYSTEMES CONTINUS ECHANTILLONNES

### 4.1 Structure des systèmes commandés par ordinateur

Les systèmes commandés par ordinateurs sont presque toujours des systèmes physiques continus par nature. Le ordinateur traite par contre les entrées et les sorties de manière discrète avec une période d'échantillonnage  $T_e$ . Les éléments importants, du point de vue de l'automaticien sont donc :

- le convertisseur analogique / numérique CAN ou du moins le système qui fournit séquentiellement les mesures au ordinateur à la période  $T_e$ .
- le ordinateur lui-même. Il doit effectuer l'acquisition des mesures, le traitement des informations, l'envoi des consignes dans un temps inférieur à  $T_e$ .
- le convertisseur numérique / analogique CNA.
- le système physique.

On suppose, pour faire simple et pour pouvoir utiliser la panoplie des outils mathématiques, que les systèmes sont échantillonnés à la même période  $T_e$  et au même moment.

La Figure 9 montre le système physique commandé par le ordinateur, dans sa version simpliste.

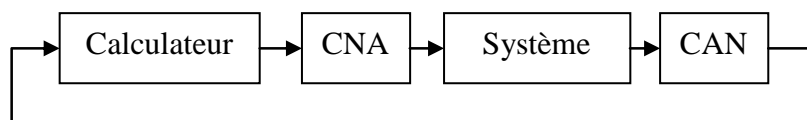


Figure 9 : système commandé par ordinateur

Au XXIème siècle les choses sont quand même parfois un petit peu plus compliquées :

- le ordinateur fournit bien souvent un signal de commande numérique (transporté par bus informatique USB, RS232, CAN, Ethernet, etc...) directement interprété par l'actionneur. On parle "d'actionneur intelligent".
- de même le "capteur intelligent" fournit aussi bien souvent la mesure sous forme d'un signal numérique.
- Les signaux sont transportés par des bus informatiques qui n'ont pas que ça à faire : il peut y avoir du retard.
- de même les capteurs et actionneurs sont rarement tous seuls, ce qui se traduit par des périodes d'échantillonnage différentes, et parfois non synchrones...

- enfin le ordinateur peut aussi avoir d'autres choses à faire...

Ceci étant dit nous allons pour le moment nous restreindre au cas représenté sur la Figure 9 avec des instants d'échantillonnage synchrones et régulièrement espacés d'un temps  $T_e$ .

## 4.2 Echantillonnage des systèmes continus

On dispose généralement de la fonction de transfert continue (notée en utilisant la variable complexe  $s$  ou  $p$ )  $F(s)$  du système continu. Le problème est d'obtenir la fonction de transfert discrète du système échantillonné de manière à pouvoir synthétiser un correcteur. Le système continu est associé au convertisseur numérique – analogique qu'il faut prendre en compte.

Les convertisseurs numérique analogiques peuvent généralement se représenter par des "échantillonneurs – bloqueurs d'ordre 0" ou BOZ. Cela signifie que la commande est maintenue à une valeur constante pendant toute la durée du temps d'échantillonnage.

Si l'on applique une commande discrète impulsionnelle à un BOZ il va maintenir la commande 1 pendant la durée  $T_e$  puis revenir à zéro (Figure 10).

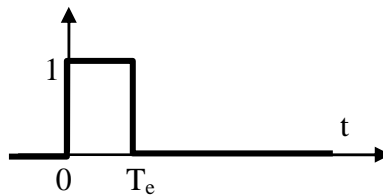


Figure 10 : réponse impulsionnelle d'un bloqueur d'ordre 0

La réponse impulsionnelle  $BOZ(t)$  est donc composée d'un échelon unitaire suivi d'un échelon unitaire de signe opposé et décalé d'un temps  $T_e$ . La transformée de Laplace de la réponse impulsionnelle du bloqueur est donnée par la formule (23) :

$$BOZ(s) = \frac{1}{s} - \frac{e^{-jT_e}}{s} \quad (23)$$

Du point de vue du ordinateur il faut donc prendre en compte le système continu de fonction de transfert  $F_{BOZ}(s)$  que l'on devra échantillonner :

$$F_{BOZ}(s) = \frac{1 - e^{-jT_e}}{s} \cdot F(s) \quad (24)$$

$$F_{\text{BOZ}}(z) = Z^{-1} \left( \frac{1 - e^{-jT_e}}{s} \cdot F(s) \right) \quad (25)$$

En pratique l'obtention de la transformée en z d'un système échantillonné par bloqueur d'ordre 0 se fait à l'aide de tables. On donne quelques exemples fondamentaux dans le Tableau 3 ci-dessous.

Tableau 3 : transformée en z de systèmes continus échantillonnés

F(s)	F <sub>BOZ</sub> (z)
1	1
$\frac{1}{s}$	$\frac{T_e \cdot z^{-1}}{1 - z^{-1}}$
$\frac{1}{1 + T \cdot s}$	$\begin{cases} \frac{b_1 \cdot z^{-1}}{1 + a_1 \cdot z^{-1}} \\ b_1 = 1 - e^{-T_e/T} \quad a_1 = -e^{-T_e/T} \end{cases}$
$\frac{e^{-Ls}}{1 + T \cdot s}$	$\begin{cases} \frac{b_1 \cdot z^{-1} + b_2 \cdot z^{-2}}{1 + a_1 \cdot z^{-1}} \quad a_1 = -e^{-T_e/T} \\ b_1 = 1 - e^{-(L-T_e)/T} \quad b_2 = e^{-T_e/T} (e^{L/T} - 1) \end{cases}$
$L < T_e$	

Si l'on dispose de Matlab c'est bien sûr encore plus simple. On utilise la fonction c2d avec l'argument 'zoh' pour "zero holder hold" (bloqueur d'ordre zéro).

#### Matlab

```
% Définition d'un système continu :
sysc=tf(1,[1 1]);
% Obtention de la transformée en z du système échantillonné à Te
Te=0.1;
sysd=c2d(sysc,Te,'zoh');
% Notation en z^-1
sysd.variable='z'
```

### 4.3 Effets de l'échantillonnage

Le convertisseur numérique a deux effets (néfastes) importants : l'effet de retard et l'effet de quantification.

L'effet de retard se voit dans (24) : on peut dire en première approximation que l'échantillonnage se traduit par un retard pur de  $T_e/2$ . Cela a donc, si ce temps

d'échantillonnage est non négligeable par rapport aux constantes de temps du système, un effet déstabilisant.

Pour illustrer l'effet de quantification supposons par exemple que l'on utilise un convertisseur analogique/numérique 8 bits 0-10V. Le plus petit incrément que le convertisseur peut distinguer est de  $10V/2^8 = 40mV$ . Si le signal utile varie physiquement entre 5V et 5,4V on n'aura que dix incréments utilisables... Ce phénomène n'entre pas dans la panoplie des phénomènes modélisables par la transformée en z. En pratique il est donc souvent nécessaire d'étudier le système et de synthétiser la commande numérique en négligeant ce phénomène puis de vérifier a posteriori qu'il n'a pas un effet trop néfaste.

#### 4.4 Où l'on revoit le théorème de Shannon

Le théorème de Shannon est bien connu en ce qui concerne l'échantillonnage de signaux discrets : on sait que le signal ne doit pas comporter de composante de fréquence supérieure à  $f_e/2$  si l'on ne veut pas observer de repliement. On doit aussi prendre des précautions si l'on veut travailler sur la fonction de transfert discrète correspondant à un système continu échantillonné.

On va traiter un exemple concret pour mettre en évidence le problème de repliement dans le cas de la commande des systèmes continus. On s'intéresse à deux systèmes continus du second ordre ayant des pôles de même partie réelle mais de parties imaginaires séparées de  $2\pi T_e$ . Le script Matlab ci-dessous montre la création des deux systèmes :

##### Matlab

```
% Définition de deux systèmes continus :  
sys1=zpk([],[-1-2i -1+2i],2^2+1);  
sys2=zpk([],[-1-(2+2*pi)*i -1+(2+2*pi)*i],1+(2+2*pi)^2);
```

On voit sur la Figure 11 que la réponse indicielle des deux systèmes est bien différente.



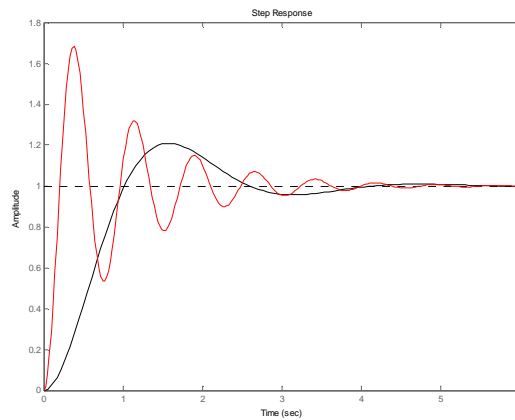


Figure 11 : réponse indicielle de deux systèmes du second ordre

Si l'on calcule la fonction de transfert échantillonnée les deux systèmes à la période de 1 seconde on obtient deux réponses indicielles très proches (représentés par des  $\circ$  sur la Figure 12). Conclusion : la période d'échantillonnage était mal choisie

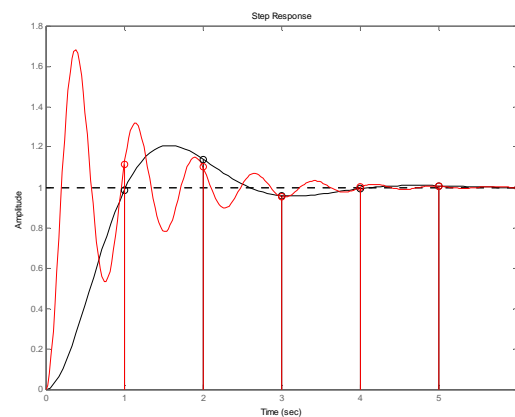


Figure 12 : réponse indicielle de deux systèmes du second ordre échantillonnés à 1s

En pratique la commande échantillonnée de systèmes continus doit être associée à certaines précautions :

- le système ne doit pas comporter de dynamiques propres situées dans les fréquences supérieures à  $f_c/2$  (exemple illustré dans ce chapitre).
- les signaux échantillonnés doivent être "nettoyés" des fréquences supérieures à  $f_c/2$  avec un filtre anti-repliement. (A moins que le signal soit suffisamment propre et ne comporte pas de composantes significatives au-delà de ces fréquences).

### Choix de la période d'échantillonnage

Le choix de la période d'échantillonnage est très important et dépend de la dynamique des signaux. Attention, les systèmes commandés en boucle fermée ont souvent une dynamique plus rapide que le système en boucle ouverte : il faut donc prendre en compte la dynamique en boucle fermée pour le choix de la fréquence d'échantillonnage.

En pratique on préconise de prendre une fréquence 5 à 25 fois supérieure à la bande passante souhaitée en boucle fermée.

On peut être tenté de prendre une fréquence d'échantillonnage très élevée si la technologie le permet (ordinateur et convertisseurs suffisamment rapide). Ceci peut amener rapidement à des problèmes numériques. Imaginons par exemple le cas d'un système ayant une dynamique très lente (quelques secondes) et un système de mesure codant les mesures sur 10 bits (cas classique). La Figure 13 illustre cet exemple. Le signal varie peu entre deux instants d'échantillonnage et les variations du signal sont faibles par rapport à la valeur de l'incrément numérique (figure du haut). Le calcul de la simple dérivée numérique se traduit donc par une succession de  $-1$   $0$  et  $+1$ .

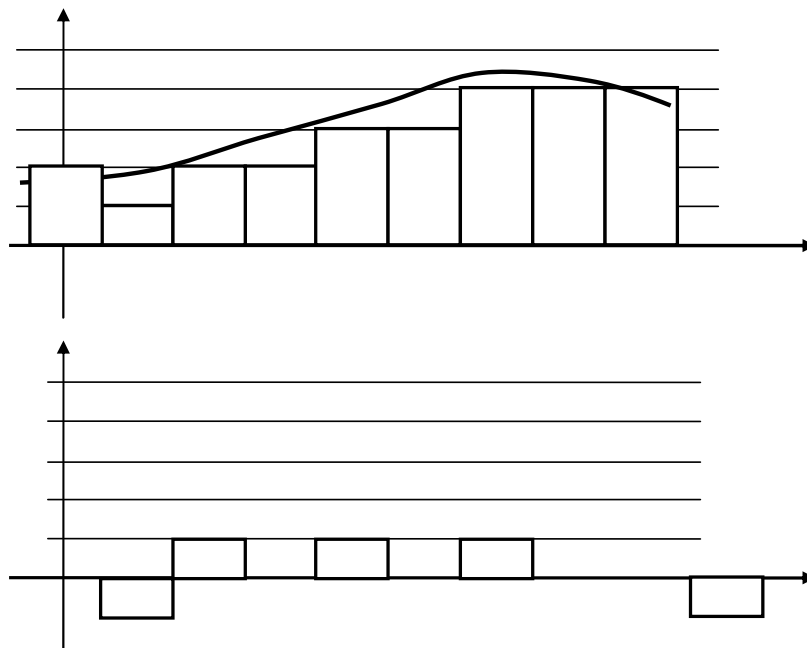


Figure 13 : période d'échantillonnage trop faible. En haut : signal échantillonné et discrétisé ; en bas : calcul de la dérivée numérique.

### Filtre anti-repliement

En toute rigueur chaque fois qu'un signal est échantillonné il faudrait faire précéder le convertisseur numérique d'un filtre analogique anti-repliement éliminant physiquement toutes les composantes du signal de fréquence supérieure à  $f_e/2$ .

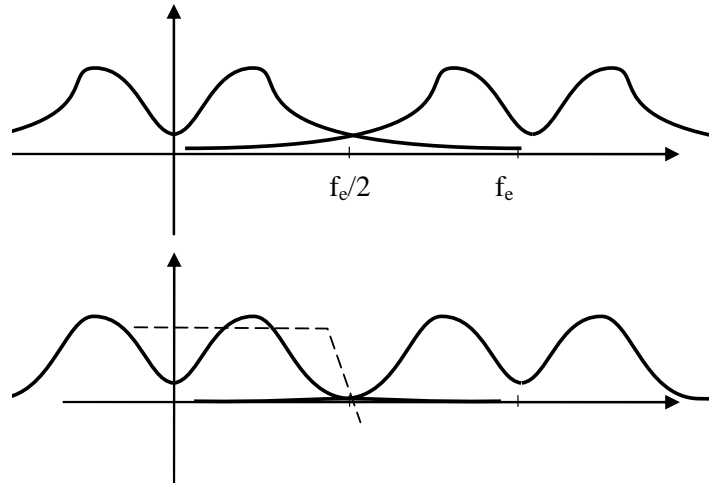


Figure 14 : Effet du filtre anti-repliement

Bien souvent la période d'échantillonnage de l'asservissement est relativement lente (quelques secondes) et il est alors impossible de réaliser un filtre de constante de temps aussi faible. On procède alors à un sur-échantillonnage : le signal est filtré, échantillonné à haute fréquence  $T'_e$  puis filtré numériquement et sous-échantillonné à la période  $T_e$  beaucoup plus grande que  $T'_e$  comme illustré sur la

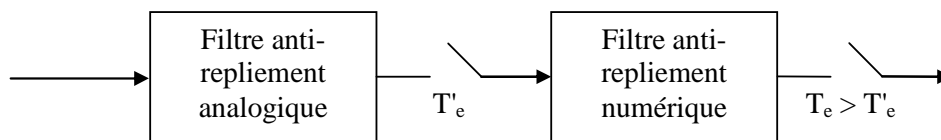


Figure 15 : Sur échantillonnage

Un simple filtre numérique de type moyenne mobile est souvent suffisant.

### 4.5 Discrétisation de fonction de transfert continues

Une manière classique pour réaliser un contrôle par ordinateur est de commencer par synthétiser un correcteur continu avec les outils de l'automatique classique puis de trouver un correcteur discret qui "approche" convenablement le correcteur continu.

#### 4.5.1 Approximation numérique d'équation différentielle

Supposons que le correcteur continu que l'on a synthétisé soit du type PD :

$$\text{PID}(s) = \frac{e(s)}{\varepsilon(s)} = k_p + k_D \cdot s \quad (26)$$

Il s'agit de calculer les commandes  $e(k)$  successives issues du correcteur en fonction des erreurs  $\varepsilon(k)$ . Une approche classique et intuitive est d'approximer la dérivée par une différence :

$$e(k) = k_p \cdot \varepsilon(k) + k_D \cdot \frac{\varepsilon(k) - \varepsilon(k-1)}{T_e} \quad (27)$$

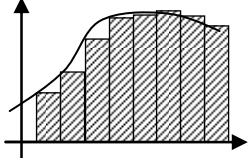
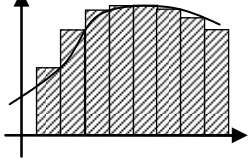
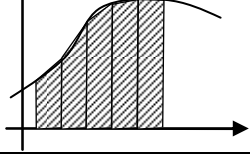
Si on calcule la transformée en  $z$  de  $e(k)$  on obtient :

$$e(z) = k_p \cdot \varepsilon(z) + k_D \cdot \frac{\varepsilon(z) - z^{-1}\varepsilon(z)}{T_e} = \left( k_p + k_D \cdot \frac{1 - z^{-1}}{T_e} \right) \cdot \varepsilon(z) \quad (28)$$

On retrouve la fonction de transfert PD(s) dans laquelle on a effectué le changement de variable  $s = \frac{1-z^{-1}}{T_e}$

De manière plus générale, l'approximation de fonction de transfert revient à effectuer l'approximation numérique d'intégration. Selon la méthode d'intégration choisie on effectue des changements de variable différents.

Tableau 4 : méthodes d'intégration

Méthode	Changement de variable	
Euler progressif (Euler's forward method)	$s = \frac{1 - z^{-1}}{z^{-1} \cdot T_e}$	
Euler rétrograde (Euler's backward method)	$s = \frac{1 - z^{-1}}{T_e}$	
Méthode des trapèzes, ou bilinéaire, ou de Tustin	$s = \frac{2}{T_e} \cdot \frac{1 - z^{-1}}{1 + z^{-1}}$	

En pratique le changement de variable  $s = \frac{1 - z^{-1}}{z^{-1} \cdot T_e}$  n'est pas sans risque. On donne sur la Figure 16 la correspondance entre la zone stable du plan complexe en  $s$  et la zone correspondante après la transformation (Euler progressif, Euler rétrograde, bilinéaire). On voit notamment qu'il est possible, partant d'un système continu stable, d'obtenir des pôles dans le domaine discret en dehors du cercle de stabilité si la transformation appliquée est celle d'Euler progressif.

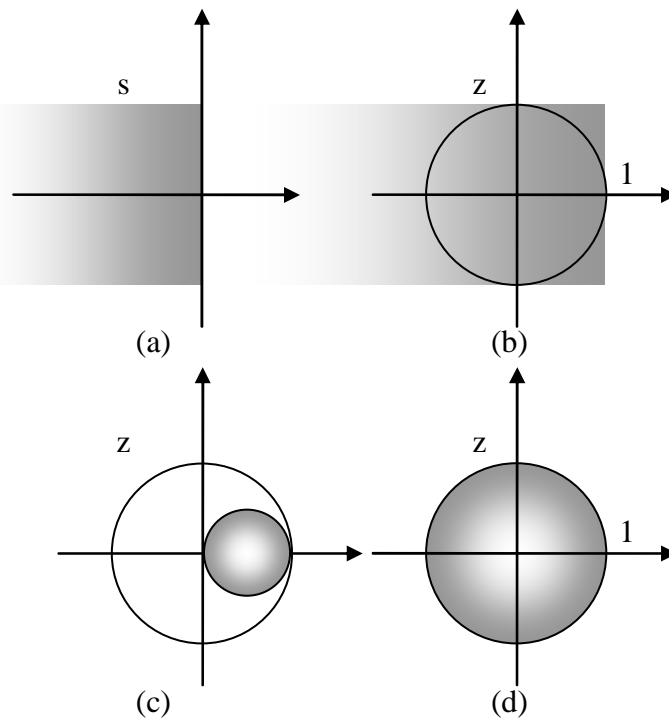


Figure 16 : correspondance des plans complexes. (a) : zone de stabilité dans le plan complexe continu en  $z$  ; (b) zone correspondante avec l'approximation d'Euler progressif ; (c) zone correspondante avec l'approximation d'Euler rétrograde ; (d) zone correspondante avec l'approximation bilinéaire.

La fonction `c2d` de Matlab permet d'obtenir l'équivalent discret d'une fonction de transfert continue avec la méthode d'intégration comme argument :

```

Matlab
sysc=tf([1 0],[1 2 5]);
Te=0.1;
% Méthode d'euler :
sysd1=c2d(sysc,Te,'zoh');
% Méthode des trapèzes :
sysd2=c2d(sysc,Te,'foh');
% Méthode bilinéaire :
sysd3=c2d(sysc,Te,'tustin');
    
```

#### 4.5.2 Superposition des réponses indicielles

Une autre méthode pour approximer une fonction de transfert discrète est de s'assurer que la réponse indicielle de sa version discrétisée correspond exactement lors des temps d'échantillonnage.

### 4.5.3 Correspondances pôles - zéros

La dernière méthode est de partir d'une décomposition de la fonction de transfert continue en pôles et zéros et d'effectuer la conversion pôles à pôles et zéros par zéros. La correspondance, selon que le pôle est réel ou complexe est donnée ci-dessous :

$$\begin{aligned} (s + a) &\rightarrow (z - e^{-a \cdot T_c}) \\ (s + a)^2 + b^2 &\rightarrow z^2 - 2 \cdot e^{-a \cdot T_c} \cdot \cos(b \cdot T_c) \cdot z + e^{-2 \cdot a \cdot T_c} \end{aligned} \quad (29)$$

### 4.5.4 Bilan

Aucune méthode n'est parfaite et on n'aura jamais en même temps :

- même réponse indicielle
- même réponse impulsionnelle
- approximation (Euler, Tustin, etc...) de l'équation différentielle
- correspondance des pôles et des zéros
- correspondance des réponses fréquentielles

Ceci étant dit le choix de la période d'échantillonnage est primordial. Si la fréquence d'échantillonnage est beaucoup plus faible que la dynamique de la fonction de transfert à approximer la différence entre chaque méthode sera minimale.





## 5. IDENTIFICATION DES SYSTEMES ECHANTILLONES

Dans le présent chapitre on donne les bases des méthodes d'identification. Pour une approche complète, voir le cours de troisième année "Identification avion" [6] et les ouvrages [1] et [7].

Même si l'on dispose d'un modèle théorique du processus que l'on souhaite commander il est presque toujours nécessaire de l'affiner en procédant à des essais sur le système réel. Bien souvent le seul modèle dont l'on dispose est ce modèle issu de l'expérimentation. Différentes méthodes s'appliquent selon l'objectif recherché et le niveau de précision exigé. On distinguera :

- les modèles paramétriques
- les modèles simples
- les modèles non paramétriques

### 5.1 Identification d'un modèle ARX par la méthode des moindres carrés récursifs

A partir de cet exemple, le plus simple qui soit, on pose les bases de toutes les méthodes d'identification.

On considère un système d'entrée  $u(t)$  et de sortie  $y(t)$  représenté par l'équation de récurrence suivante :

$$y(t) + a_1 \cdot y(t-1) + a_2 \cdot y(t-2) + \dots + a_n \cdot y(t-n) = b_0 \cdot u(t) + b_1 \cdot u(t-1) + b_2 \cdot u(t-2) + \dots + b_m \cdot u(t-m) \quad (30)$$

L'objectif est, à partir d'essais et de mesures, d'identifier les meilleures valeurs possibles pour les coefficients  $a_i$  et  $b_i$ .

L'équation (30) peut être réécrite pour mettre en évidence la *prédiction* de la sortie  $y(t)$  à partir des mesures passées :

$$y(t) = -a_1 \cdot y(t-1) - a_2 \cdot y(t-2) - \dots - a_n \cdot y(t-n) + b_0 \cdot u(t) + b_1 \cdot u(t-1) + b_2 \cdot u(t-2) + \dots + b_m \cdot u(t-m) \quad (31)$$

Soit :

$$y(t) = \varphi^T(t) \cdot \theta(t)$$

avec :

$$\begin{cases} \varphi^T(t) = [-y(t-1) \ \dots \ -y(t-n) \ , \ u(t-1) \ \dots \ u(t-m)] \\ \theta(t) = [a_1 \ \dots \ a_n \ , \ b_1 \ \dots \ b_m] \end{cases} \quad (32)$$

Le but est de trouver le meilleur ensemble de paramètres  $\theta(t)$  : celui qui, à partir d'un ensemble de mesures  $\varphi(t)$  permet de prédire avec la meilleure précision la sortie effectivement mesurée.

Supposons qu'à la suite d'une expérience on dispose d'un ensemble de mesures  $\{y(t), u(t)\}_{t=1:N}$  et que l'ensemble des paramètres  $\theta$  n'est pas bien connu. On construit la séquence des *sorties prédites* :

$$\hat{y}(t) = \varphi^T(t) \cdot \theta(t) \quad (33)$$

Le paramètre  $\theta$  est celui qui minimise une norme quadratique :

$$\theta = \arg \max \left( \frac{1}{N} \sum_{t=1}^N (y(t) - \hat{y}(t))^2 \right) = \arg \max \left( \frac{1}{N} \sum_{t=1}^N (y(t) - \varphi^T(t) \cdot \theta(t))^2 \right) \quad (34)$$

Compte tenu du fait que l'équation est linéaire au départ et que le terme à minimiser est du type quadratique la solution est simplement obtenue par une pseudo-inverse :

$$\theta = \left[ \sum_{t=1}^N (\varphi(t) \cdot \varphi^T(t)) \right]^{-1} \sum_{t=1}^N (\varphi(t) \cdot y(t)) \quad (35)$$

On obtient donc la solution classique des moindres carrés. La formule (35) est le plus souvent utilisée sous sa forme récursive :

$$\theta(t) = \theta(t-1) + F(t) \cdot \varphi(t) \cdot (y(t) - \theta^T(t-1) \cdot \varphi(t))$$

avec

$$F(t)^{-1} = F(t-1)^{-1} + \varphi(t) \cdot \varphi(t)^T \quad (36)$$

ou

$$F(t) = F(t-1) - \frac{F(t-1) \cdot \varphi(t) \cdot \varphi(t)^T \cdot F(t-1)}{1 + \varphi(t)^T \cdot F(t-1) \cdot \varphi(t)}$$

(On laisse au lecteur le plaisir d'effectuer lui-même les démonstrations, qu'il peut du reste trouver dans [1] et [7]).

## 5.2 L'identification paramétrique en général et en pratique

L'exemple précédent a permis de mettre en évidence tous les aspects de la méthodologie classique de l'identification :

- il a fallu choisir un ensemble de mesures  $\{y(t), u(t)\}_{t=1:N}$ . Le choix du signal d'excitation  $u(t)$  est en pratique primordial,
- il a fallu choisir une structure pour le modèle à identifier (ici les ordres  $n$  et  $m$ ),
- il a fallu choisir un critère (ici un critère quadratique sur l'erreur entre la sortie et la sortie prédite),
- il a fallu choisir un algorithme et le paramétrer (ici l'algorithme des moindres carrés récursifs qu'il a fallu initialiser)

Enfin un point non traité dans le chapitre précédent concerne la validation :

- il aurait fallu déterminer une méthode pour comparer différents modèles et choisir le meilleur

### 5.2.1 Choix du signal d'excitation, traitement des mesures

De manière générale le système que l'on doit identifier doit être excité dans tout le domaine fréquentiel qui nous intéresse. Le signal classiquement utilisé est une séquence pseudo binaire aléatoire (Pseudo Random Binary Signal) tel celui représenté sur la Figure 17.

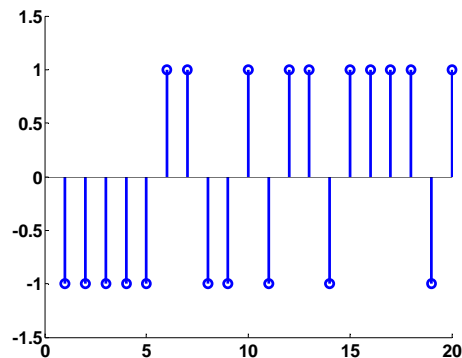


Figure 17 : exemple d'une séquence PRBS

Le signal d'excitation doit se superposer au signal de commande correspondant au point de fonctionnement étudié. Son amplitude doit être faible par rapport à l'amplitude des commandes du système (quelques %).

Une fois le système excité et les mesures effectuées il faut supprimer les valeurs moyennes des mesures.

### 5.2.2 Choix d'une structure de modèle

Le modèle étudié dans le chapitre précédent est de type ARX. On peut le noter sous forme polynomiale :

$$A \cdot y = B \cdot u \quad (37)$$

Ce modèle correspond en fait à un parti-pris sur la manière dont on traite les perturbations. En réalité on prend en compte un signal d'erreurs possibles  $e$  (de moyenne nulle) tels que :

$$A \cdot y = B \cdot u + e \quad (38)$$

Si l'on fait cette hypothèse de modélisation alors la méthode optimale est celle des moindres carrés. Malheureusement cette manière de prendre en compte les perturbations est bien restrictive. D'autres modélisations sont possibles, adaptées à d'autres algorithmes, par exemple le modèle OE :

$$y = \frac{B}{F} \cdot u + e \quad (39)$$

Il est aussi possible de "structurer" le modèle de bruit comme par exemple dans les modèles ARMAX ou BJ :

$$\begin{aligned} A \cdot y &= B \cdot u + C \cdot e \\ y &= \frac{B}{F} \cdot u + \frac{C}{D} \cdot e \end{aligned} \quad (40)$$

La structure du modèle étant choisie il faut choisir l'ordre de ses différents polynômes.

### 5.2.3 Choix d'un critère à minimiser

Dans la plupart des cas le critère à minimiser est l'erreur quadratique entre la sortie et la sortie prédite à un coup (one step ahead prediction). Ceci n'est bien évidemment pas restrictif et peut être généralisé à la prédiction à  $N$  coups.

### 5.2.4 Choix d'un algorithme

Même si chaque structure de modèle est généralement associée à un algorithme "optimal" celui-ci doit généralement se paramétrer.

Prenons par exemple l'algorithme des moindres carrés récursifs tel que présenté dans (36). Cette équation peut être modifiée en intégrant un "facteur d'oubli"  $\lambda$  :

$$\begin{aligned} \theta(t) &= \theta(t-1) + F(t) \cdot \varphi(t) \cdot (y(t) - \theta^T(t-1) \cdot \varphi(t)) \\ \text{avec} & \\ F(t)^{-1} &= \lambda \cdot F(t-1)^{-1} + \varphi(t) \cdot \varphi(t)^T \end{aligned} \quad (41)$$

Ce paramètre (typiquement entre 0.95 et 0.99) permet, comme son nom l'indique, "d'oublier" les anciennes mesures et de se concentrer sur les mesures les plus récentes.

### 5.2.5 Validation

Une fois l'algorithme d'identification appliqué et le modèle obtenu (ce qui prend généralement quelques secondes avec un ordinateur moderne) il reste à expertiser celui-ci. Selon la "qualité" du modèle obtenu on remet en cause l'une ou l'autre des étapes (signal d'excitation, structure du modèle, algorithme) et on réitère jusqu'à obtenir un modèle satisfaisant.

En pratique on essaie un certain nombre de modèles avant de choisir le "meilleur".

Le premier critère de qualité est la capacité du modèle à prédire effectivement la sortie (sortie prédite à un coup, sortie prédite à N coups). Il faut pour cela utiliser un jeu de mesures non utilisé pour l'identification. (En pratique le jeu de mesures initiales est coupé en deux : une partie pour l'identification et une partie pour la validation).

Le second critère de qualité est l'analyse du résidu : la composition spectrale de l'erreur de prédiction.

### 5.2.6 Utilisation de logiciels

En pratique il est commode d'utiliser un célèbre outil de calcul scientifique et sa toolboxe d'identification.

Comme les calculs proprement dit ne prennent que quelques secondes il est facile de réaliser un grand nombre d'identifications et de stocker les différents modèles pour les comparer ensuite. On est rapidement noyé sous le grand nombre de modèles possibles...

## 5.3 Identification "simple"

Il est souvent suffisant d'obtenir un modèle "simple" de type premier ordre et un simple essai indiciel convient souvent. Le système est excité avec un échelon unité et on cherche le meilleur modèle du premier ordre correspondant.

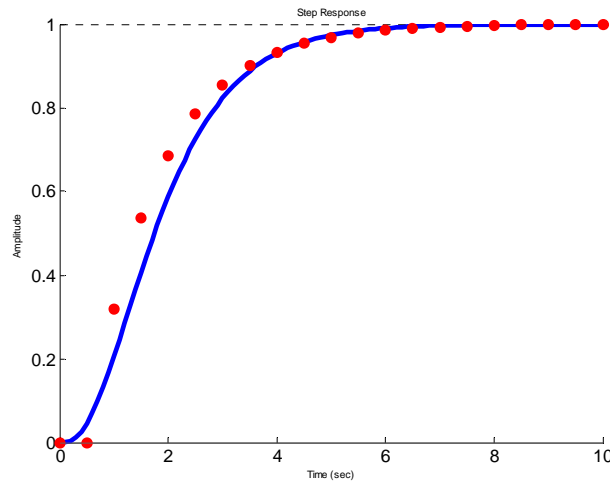


Figure 18 : identification sur un essai indiciel

Dans l'exemple de la Figure 18 on superpose le résultat d'une mesure (en bleu) avec la simulation d'un modèle discret avec une période de 0.5s. Le modèle cherché est du premier ordre avec un retard pur  $d$  :

$$y(z) = \frac{(1 + c_1) \cdot z^{-d}}{1 + c_1 \cdot z^{-1}} \cdot u(z) \quad (42)$$

Il s'agit de trouver l'entier  $d$  et le nombre réel  $c_1$  tels que la réponse indiciale du modèle corresponde pour le mieux aux mesures issues de l'essai. Ceci peut se formaliser par exemple de la manière suivante :

$$(c_1 \quad d_1) = \arg \min \sum_{t=1}^N (y_{\text{modèle}}(t) - y_{\text{mesure}}(t))^2 \quad (43)$$

Cette optimisation peut se faire à la main ou grâce à une procédure automatisée (avec le célèbre outil de calcul scientifique).

Dans le cas où le système comprend naturellement une intégration (par exemple la commande d'un moteur à courant continu par une tension et la mesure de l'angle de rotation) il est utile de prendre en compte explicitement cette intégration dans le modèle :

$$y(z) = \frac{1}{1 - z^{-1}} \cdot \frac{(1 + c_1) \cdot z^{-d}}{1 + c_1 \cdot z^{-1}} \cdot u(z) \quad (44)$$

La procédure est la même mais au lieu de prendre en compte la sortie  $y(t)$  on prend en compte la dérivée  $y'(t)$  :

$$y'(t) = y(t) - y(t-1) \quad (45)$$

## 5.4 Identification non paramétrique

Pour terminer ce panorama rapide sur les méthodes d'identification il faut parler des méthodes non paramétriques : celles qui permettent de tracer la réponse impulsionnelle ou la réponse fréquentielle du système.

### 5.4.1 Estimation de la réponse impulsionnelle

La réponse impulsionnelle d'un système peut être obtenue simplement en appliquant une entrée de type "impulsion" au système et d'en mesurer les conséquences.

La seconde méthode consiste à utiliser la méthode des moindres carrés (ou une méthode dérivée) comme vue au chapitre 5.1. La réponse impulsionnelle d'un système peut en effet être vue comme la fonction de transfert particulière où le dénominateur est de degré nul et le numérateur comporte un nombre infini (en pratique : grand) d'éléments ( $n = 0$  et  $m$  grand) :

$$y(t) = b_0 \cdot u(t) + b_1 \cdot u(t-1) + b_2 \cdot u(t-2) + \dots + b_m \cdot u(t-m) \quad (46)$$

La dernière méthode, qui sera développée ici est l'analyse d'intercorrélation.

Par définition l'intercorrélation  $R_{yu}$  entre deux signaux est :

$$R_{yu}(\tau) = E[(y(t) - E(y(t))) \cdot (u(t-\tau) - E(u(t-\tau)))] \quad (47)$$

Dans cette formule l'espérance mathématique peut être approchée grâce à la loi des grands nombres :

$$R_{yu}(\tau) \approx R_{yu}^N(\tau) = \lim_{N \rightarrow \infty} \frac{1}{N} \sum_{t=1}^N [(y(t) - m_y) \cdot (u(t-\tau) - m_u)] \quad (48)$$

On s'intéresse maintenant à un système tel que décrit par l'équation impulsionnelle (46) et on suppose que le signal d'entrée est un bruit blanc. Le signal d'entrée vérifie donc :

$$\begin{cases} \tau \neq 0 & R_{uu}(\tau) = 0 \\ \tau = 0 & R_{uu}(\tau) = R \end{cases} \quad (49)$$

Il vient alors naturellement :

$$b_k \approx \frac{1}{R} R_{yu}(k) \quad (50)$$

### 5.4.2 Estimation de la réponse fréquentielle

On utilise ici le spectre d'intercorrélation. Par définition le spectre d'intercorrélation entre deux signaux est :

$$\Phi_{yu}(\omega) = \sum_{\tau=-\infty}^{\infty} R_{yu}(\tau) \cdot e^{-i\omega\tau} \quad (51)$$

On suppose que le signal d'entrée  $u(t)$  et le signal de sortie  $y(t)$  sont reliés par une fonction de transfert linéaire du type :

$$y(t) = F \cdot u(t) + v(t) \quad (52)$$

où  $v(t)$  est une perturbation indépendante de  $u(t)$ . On a alors les relations :

$$\Phi_{yu}(\omega) = F(j\omega) \cdot \Phi_{uu}(\omega) \quad (53)$$

Il reste à estimer les spectres d'intercorrélation pour remonter à la réponse fréquentielle  $F(j\omega)$  recherchée. Une idée naturelle est d'utiliser la relation (48) mais cela marche très mal dans presque tous les cas, même en choisissant  $N$  assez grand. L'amélioration est d'utiliser une "fenêtre" pour favoriser les valeurs faibles de  $\tau$  par rapport aux valeurs élevées. La solution classique utilisant la *fenêtre de Hamming* donne :

$$\Phi_{yu}^N(\omega) = \sum_{\tau=-\gamma}^{\gamma} R_{yu}^N(\tau) \cdot w_{\gamma}(\tau) \cdot e^{-i\omega\tau} \quad (54)$$

avec

$$\begin{cases} w_{\gamma}(\tau) = \frac{1}{2} \left( 1 + \cos\left(\frac{\pi \cdot \tau}{\gamma}\right) \right) & |\tau| < \gamma \\ w_{\gamma}(\tau) = 0 & |\tau| > \gamma \end{cases}$$

On utilise ensuite la relation (51) pour estimer la réponse fréquentielle :



$$F(j \cdot \omega) = \frac{\Phi_u^N(\omega)}{\Phi_{yu}^N(\omega)} \quad (55)$$



## 6. SYSTEMES BOUCLES

### 6.1 Fonction de transfert en boucle fermée

Un système et sa commande par ordinateur peut être représenté par un ensemble de fonctions de transfert formant un schéma-bloc plus ou moins compliqué (Figure 19).

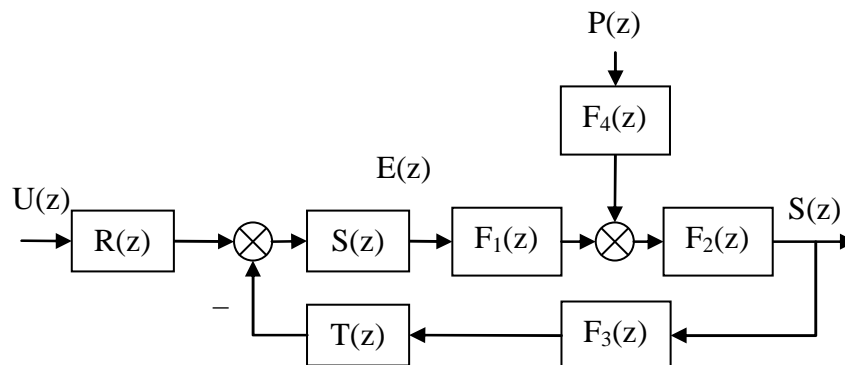


Figure 19 : système discret commandé par ordinateur. Le correcteur est représenté par les fonctions  $R, S, T$ , le système par  $F_1$  et  $F_2$ , le modèle de perturbations par  $F_4$  et le modèle des actionneurs par  $F_4$ .

Quelle que soit la structure d'un système asservi il est possible de se ramener à une forme canonique comprenant une fonction de transfert en boucle ouverte en combinant judicieusement les fonctions de transfert du procédé, du régulateur et des perturbations. La Figure 20 représente par exemple le système de la Figure 19 où  $F$  est une combinaison des fonctions de transfert du système, du régulateur, des perturbations et des capteurs.

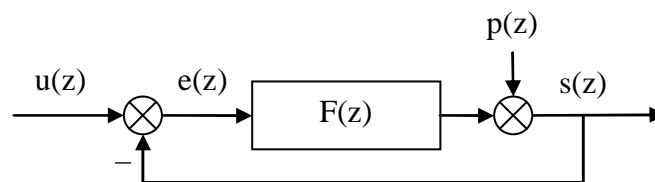


Figure 20 : système discret, notations.

Par défaut on considère une contre-réaction négative et une perturbation additive sur la sortie. La fonction de transfert  $F$  est appelée *fonction de transfert en boucle ouverte*. Comme pour les systèmes continus la *fonction de transfert en boucle fermée* est obtenue simplement à partir de  $F$  (58).

$$\begin{aligned}
 F(z) &= \frac{s(z)}{e(z)} = \frac{B(z^{-1})}{A(z^{-1})} \\
 G(z) &= \frac{s(z)}{u(z)} = \frac{B(z^{-1})}{A(z^{-1}) + B(z^{-1})}
 \end{aligned}
 \tag{56}$$

## 6.2 Erreur statique

L'erreur statique du système en boucle fermée, comme pour le système en boucle ouverte, est simplement le rapport des coefficients.

$$G_0 = \frac{\sum_{i=0}^{i=m} b_i}{1 + \sum_{i=0}^{i=m} b_i + \sum_{i=1}^{i=n} a_i}
 \tag{57}$$

Si l'on souhaite que l'asservissement confère un gain statique unitaire au système en boucle fermée (c'est-à-dire que l'on force la sortie  $s(k)$  à tendre vers la valeur de consigne  $u(k)$ ) il faut que ce rapport de coefficients soit égal à un. On montre facilement que cela est obtenu si la fonction de transfert  $F(z)$  comporte un intégrateur :

$$F(z) = \frac{s(z)}{e(z)} = \frac{1}{1 - z^{-1}} \cdot \frac{B(z)}{A'(z)}
 \tag{58}$$

## 6.3 Rejet de perturbations

La fonction de transfert entre la perturbation  $p(z)$  et la sortie  $y(z)$  est appelée fonction de sensibilité perturbation-sortie, notée  $S_{yp}$ .

$$S_{yp}(z) = \frac{s(z)}{p(z)} = \frac{A(z^{-1})}{A(z^{-1}) + B(z^{-1})}
 \tag{59}$$

Si l'on souhaite qu'une perturbation de type échelon soit parfaitement rejetée il faut que  $A(1) = 0$ , on retrouve donc la condition (58) à savoir que le système doit comporter une intégration dans la boucle directe. On peut être en outre être amené à spécifier que les perturbations ne soient pas amplifiées dans une certaine bande de fréquence. On rajoute donc une contrainte du type  $|S_{yp}(e^{j2\pi f})| < S_{max}$  pour  $f$  appartenant à un certain domaine de  $[0 \ 0.5]$ .

## 6.4 Stabilité des systèmes à simple gain de rétroaction

Les critères de Routh et de Jury (donnés dans le chapitre 3.2.7 p13) ou le calcul numérique des pôles permettent de déterminer la stabilité du système en boucle fermée si l'on connaît sa fonction de transfert.

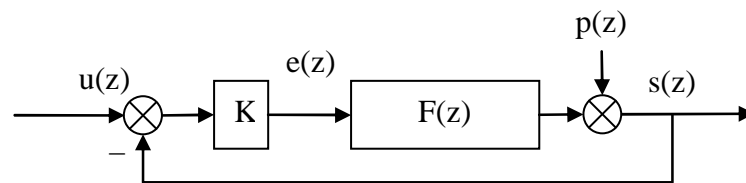


Figure 21 : système bouclé par un simple gain.

On s'intéresse ici au cas particulier où l'asservissement est réalisé par un simple gain de rétroaction  $K$  (Figure 21). On cherche la valeur de  $K$  qui amène le système en limite de stabilité. L'outil adapté ici est le lieu d'Evans ou lieu des racines (root locus) qui permet de visualiser l'évolution des pôles du système bouclé lorsque le gain  $K$  varie.

Prenons l'exemple d'un système comportant trois pôles et deux zéros. L'exemple est donné sous forme de script Matlab :

### Matlab

```
% Définition d'un système discret :
sys=zpk([-2.037 -0.1406],[0.6065 0.3679 0.3679],1,1)
%Tracé du lieu des racines
rlocus(sys);
```

Le script précédent produit le tracé de la Figure 22. On pourrait y lire que la limite de stabilité est obtenue pour  $K=0.2$  environ (détail de la Figure 23).

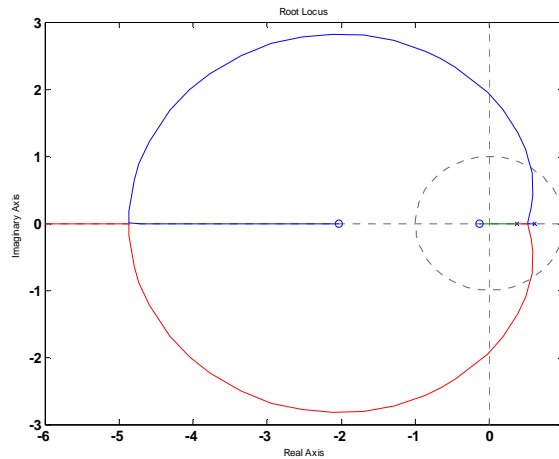


Figure 22 : lieu des racines

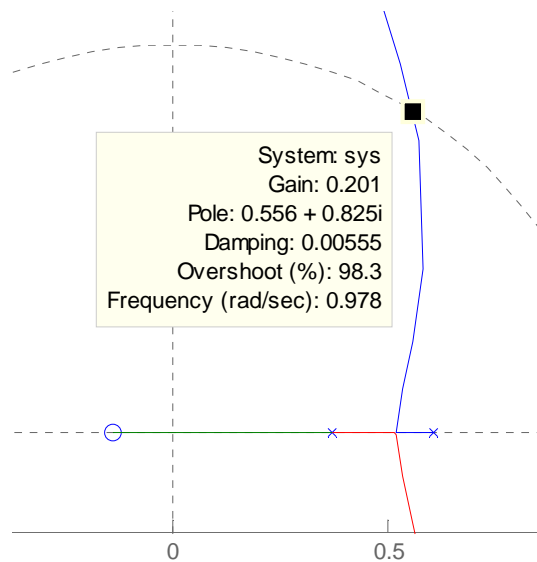


Figure 23 : lieu des racines : détail de la limite de stabilité

On voit sur la Figure 22 que le lieu des racines d'un système échantillonné ressemble à celui d'un système continu :

- on commence par tracer les pôles (x) et les zéros (o) du système en boucle ouverte.
- les pôles du système en boucle fermé décrivent un lieu qui part des pôles (pour  $K = 0$ ) et arrive aux zéros (pour  $K = \infty$ )
- lorsque il y a  $n-m$  plus de pôles que de zéros le système comporte  $n-m$  branches asymptotiques.

On ne décrira pas ici les diverses propriétés du lieu des racines qui permettent de le tracer à la main puisque il est facile à obtenir avec un ordinateur.

Il peut aussi être utilisé pour régler "à la main" le gain permettant d'obtenir un ou plusieurs modes dominants ayant certaines propriétés de fréquence et d'amortissement. On voit sur la Figure 24 que le gain 0.017 permet d'obtenir un pôle double d'amortissement 0.706 et de fréquence 0.65 rad/s et un pôle simple de fréquence 1.63 rad/s.

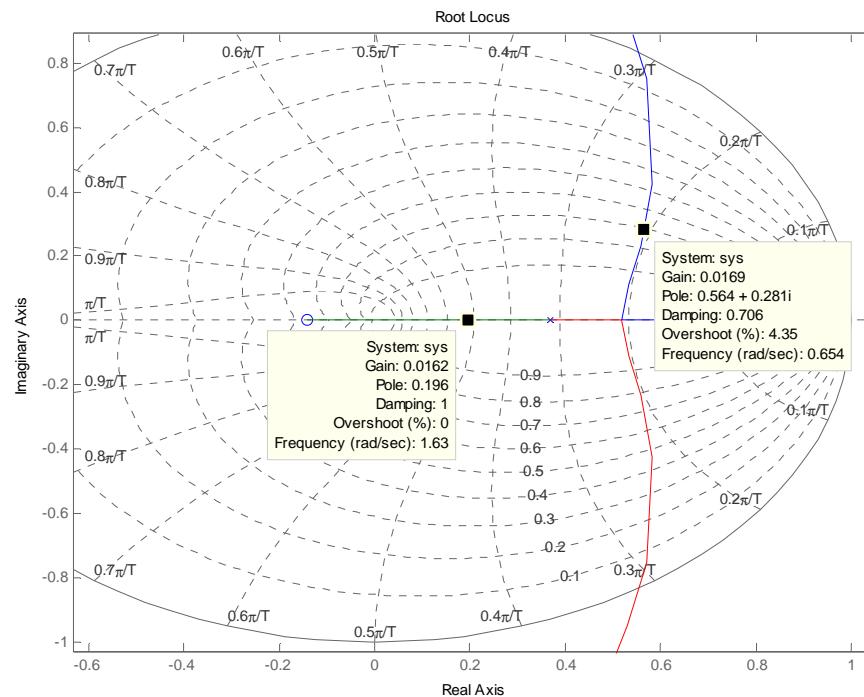


Figure 24 : utilisation du lieu des racines pour régler le gain de la boucle fermée

L'outil rltool de Matlab permet de nombreuses manipulations graphiques de ce type.

## 6.5 Analyse des systèmes bouclés dans le domaine fréquentiel : stabilité

Comme dans le cas continu il est possible de prévoir des propriétés, en particulier la stabilité, d'un système en boucle fermée à partir de l'examen de la fonction de transfert en boucle ouverte.

On ne redémontrera pas ici le théorème de Nyquist mais on se contente de rappeler l'énoncé.

On rappelle que le *lieu de Nyquist* est le lieu de  $F(j2\pi f)$  dans le plan complexe pour  $f$  variant de 0 à  $+\infty$ . A noter que le lieu pour la zone  $[0, 0.5]$  est symétrique de celui pour la zone  $[0.5, 1]$ . Pour cette raison on ne représente généralement que le lieu pour  $f$  appartenant à  $[0, 0.5]$ .

### 6.5.1 Théorème de Nyquist dans le cas où le système est stable en boucle ouverte (critère du revers)

Le système est stable en boucle fermée si le lieu de Nyquist de la fonction de transfert en boucle ouverte parcouru dans le sens des fréquences croissantes de 0 à 0.5 laisse le point critique  $(-1,0)$  à gauche.

La Figure 25 représente par exemple le lieu de Nyquist d'une fonction de transfert en boucle ouverte qui sera stable en boucle fermée.

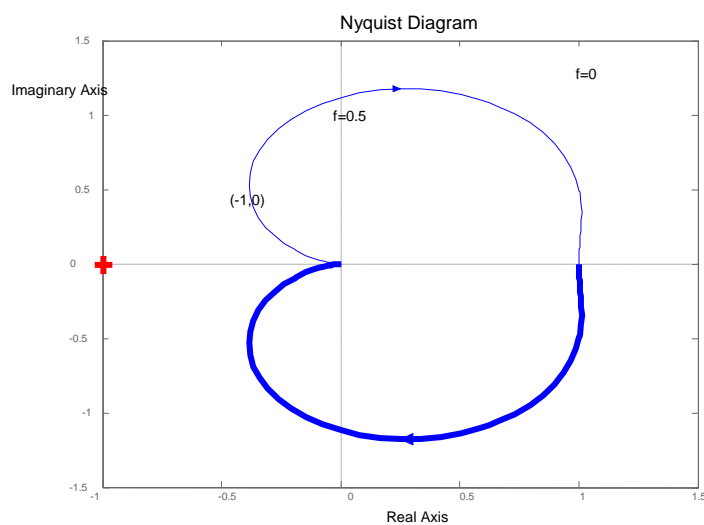


Figure 25 : lieu de Nyquist d'un système stable

### 6.5.2 Théorème de Nyquist dans le cas où le système est instable en boucle ouverte

Si le système en boucle ouverte comporte  $N$  pôles instables il sera stable en boucle fermée si le lieu de Nyquist de la fonction de transfert en boucle ouverte parcouru dans le sens des fréquences croissantes de  $-0.5$  à  $0.5$  entoure le point critique  $(-1,0)$   $N$  fois dans le sens des aiguilles d'une montre..

### 6.5.3 Interprétations du lieu de Nyquist

Comme dans le cas continu, la distance à laquelle on passe du point critique renseigne sur la robustesse du système bouclé. La distance sera exprimée de différente manière (Figure 26) :

- marge de module  $\Delta M$  : plus petite distance entre le lieu des racines et le point critique,



- marge de gain  $\Delta G$
- marge de phase  $\Delta P$

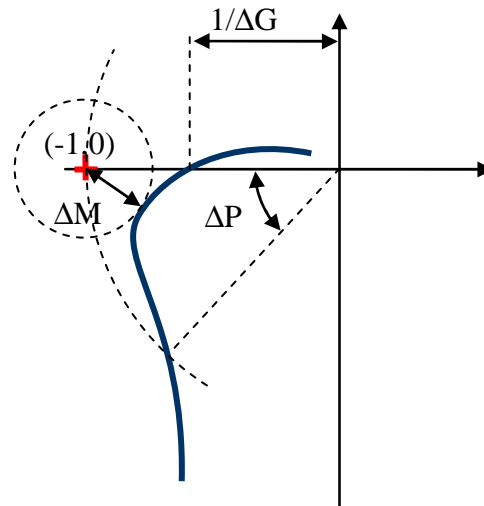


Figure 26 : Marges de stabilité

Les valeurs typiques de ces marges sont données ci-après :

- $\Delta G \geq 6\text{dB}$
- $\Delta M \geq -6\text{dB}$
- $30^\circ \leq \Delta P \leq 60^\circ$

La distance entre le point critique et le lieu des racines est la norme de  $1+F(j2\pi f)$ . On peut donc exprimer la marge de module en fonction de la fonction de sensibilité :

$$\begin{aligned} \Delta M &= \min |1 + F(j \cdot \omega)|_{\omega \in [0, \pi]} \\ \Delta M &= \min \left| 1 + \frac{B(j \cdot \omega)}{A(j \cdot \omega)} \right|_{\omega \in [0, \pi]} = \max \left| \frac{A(j \cdot \omega)}{A(j \cdot \omega) + B(j \cdot \omega)} \right|_{\omega \in [0, \pi]} \quad (60) \\ \Delta M &= \max |S_{yp}(j \cdot \omega)|_{\omega \in [0, \pi]}^{-1} \\ \Delta M_{\text{dB}} &= -\max |S_{yp}(j \cdot \omega)|_{\text{dB}} \end{aligned}$$

Si l'on trace l'évolution de  $S_{yp}$  en fonction de  $f$  on obtient donc la valeur de la marge de module.



# 7. METHODES DE REGULATION DES SYSTEMES ECHANTILLONES

## 7.1 Introduction

Les bases théoriques permettant de concevoir le correcteur pour un système échantillonné ont été posées dans les chapitres précédents. Ce chapitre fait le bilan des connaissances acquises et donne quelques solutions pour calculer des correcteurs à partir d'un cahier des charges précis.

Les systèmes que l'on cherche à commander sont généralement, par nature, continus. La réalisation d'un correcteur par un ordinateur implique que l'on génère une équation de commande discrète (généralement une équation récurrente). L'obtention de cette équation de commande peut se faire de deux manières :

- la première méthode consiste à d'abord synthétiser un correcteur continu (en utilisant les outils classiques de l'automatique et la simulation) et discrétiser ensuite ce correcteur. C'est l'objet du chapitre 7.2.2.
- la seconde méthode consiste à tenir la représentation discrète du système continu associé à son interface numérique (convertisseurs analogique-numérique et numérique-analogique) comme expliqué dans le chapitre 4.5 puis synthétiser directement le correcteur discret. Cela sera expliqué dans le chapitre 7.3 et suivants.

Ce choix peut être résumé par la figure suivante :

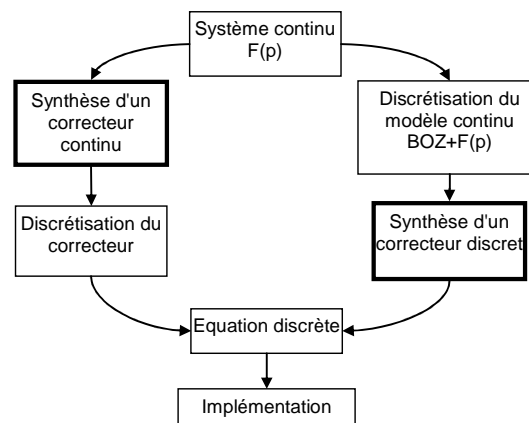


Figure 27 : Réalisation d'un correcteur discret.

Au-delà des affaires de goût, le choix de la méthode sera souvent dicté par l'application. En effet, la première méthode est d'autant meilleure que le correcteur discrétisé approche correctement le correcteur continu. Cela implique que le temps d'échantillonnage soit très faible par rapport aux constantes de temps du système

bouclé. Dans le cas où les contraintes temps réel sont fortes (capacités de calcul partagées par exemple) il sera préférable de synthétiser un correcteur discret.

## 7.2 Structure d'un système commandé par ordinateur, hypothèses

Par la suite on se ramènera toujours à un système tel que celui représenté sur la Figure 28 en cherchant un correcteur  $K(z)$  linéaire. On montre que quel que soit le correcteur linéaire on peut se ramener à la forme représenté sur la Figure 29 (forme RST).

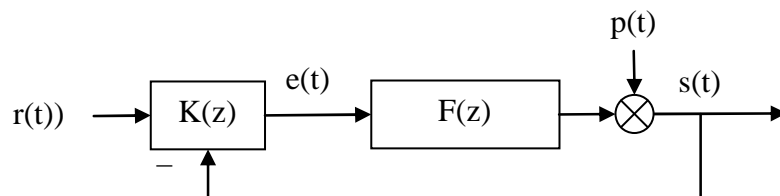


Figure 28 : système bouclé, forme générale.

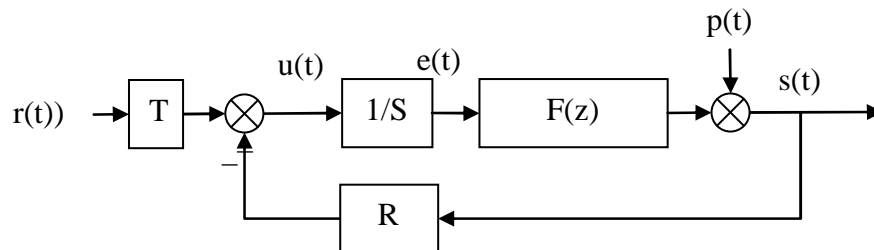


Figure 29 : système bouclé, forme canonique.

### 7.2.1 Discrétisation, choix de la période d'échantillonnage

Du point de vue du ordinateur, l'extérieur (le procédé) est représenté par la fonction de transfert  $F(z)$ . L'hypothèse importante est que la commande (sortie du ordinateur) et la mesure (entrée du ordinateur) sont échantillonnées de manière synchrone à la même période  $T_e$ .

Le choix judicieux de la fréquence d'échantillonnage est très important. En pratique on choisit cette fréquence en fonction de la bande passante du système en **boucle fermée** :

$$f_e = (6 \text{ à } 25) f^{BF} \quad (61)$$

### 7.2.2 Discrétisation d'un correcteur continu

La première méthode pour obtenir les équations d'un correcteur numérique est de faire sa synthèse avec les outils de l'automatique continue puis de discrétiser la fonction de transfert du correcteur continu obtenu (voir pour cela le chapitre 4.2 *Echantillonnage des systèmes continus*).

On va voir par la suite que la synthèse directe d'un correcteur discret offre plus de degré de liberté et permet en général de remplir un cahier des charges plus complet.

## 7.3 Régulateur PID numérique

Le régulateur le plus populaire est le régulateur PID. Nous allons voir dans un premier temps comment on génère un régulateur PID à partir des équations bien connues du régulateur PID continu et comment ces équations peuvent être généralisées pour obtenir plus de degrés de liberté pour le réglage.

### 7.3.1 Cas 1

Un régulateur PID continu a une fonction de transfert du type :

$$K_{\text{PID}}(s) = K \cdot \left( 1 + \frac{1}{T_i \cdot s} + \frac{T_d \cdot s}{1 + \frac{T_d}{N} \cdot s} \right) \quad (62)$$

La discrétisation de ce correcteur continu par la méthode d'Euler (Tableau 4 page 31) se fait en remplaçant la variable de Laplace  $s$  par le terme  $\frac{1-q^{-1}}{q^{-1} \cdot T_e}$ . On obtient :

$$K_{\text{PID}}(q) = K \cdot \left( 1 + \frac{1}{T_i \cdot \frac{1-q^{-1}}{T_e}} + \frac{T_d \cdot \frac{1-q^{-1}}{T_e}}{1 + \frac{T_d}{N} \cdot \frac{1-q^{-1}}{T_e}} \right) \quad (63)$$

En réarrangeant on obtient la forme générale RST (Figure 30):

$$K_{PID}(q) = K \cdot \frac{r_0 + r_1 \cdot q^{-1} + r_2 \cdot q^{-2}}{(1 - q^{-1}) \cdot (1 + s_1' \cdot q^{-1})}$$

$$\begin{cases} r_0 = K \cdot \left( 1 + \frac{T_e}{T_i} + \frac{N \cdot T_d}{T_d + N \cdot T_e} \right) \\ r_1 = -K \cdot \left( 1 + \frac{T_d}{T_d + N \cdot T_e} + \frac{T_e}{T_i} \cdot \frac{T_d}{T_d + N \cdot T_e} + \frac{N \cdot T_d}{T_d + N \cdot T_e} \right) \\ r_2 = K \cdot \left( \frac{T_d}{T_d + N \cdot T_e} \right) \\ s_1' = \frac{T_d}{T_d + N \cdot T_e} \end{cases} \quad (64)$$

Comme on le voit le correcteur obtenu comporte autant de degrés de liberté que le correcteur PID analogique. On reconnaît au dénominateur le terme  $(1 - q^{-1})$  en facteur qui assure l'effet d'intégration.

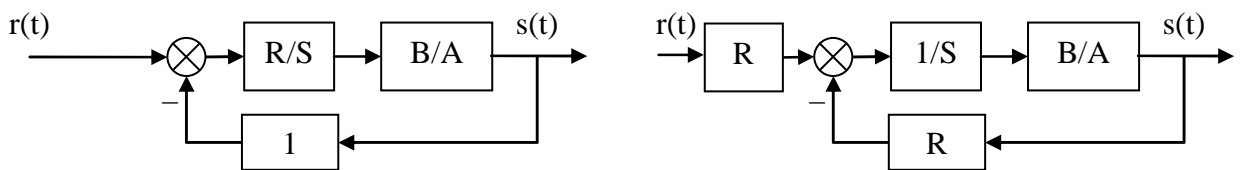


Figure 30 : PID sous forme RST

La fonction de transfert en boucle fermée obtenue est donnée par l'équation suivante :

$$H_{BF} = \frac{B \cdot R}{A \cdot S + B \cdot R} = \frac{B \cdot R}{P} \quad (65)$$

De manière générale le polynôme caractéristique  $P$  fixe les performances visées en boucle fermée. Le calcul de  $R$  et  $S$ ,  $P$  étant fixé et  $A$  et  $B$  étant connus, est réalisé en résolvant l'équation de Bézout  $P = A \cdot S + B \cdot R$ . (voir 9.1 Annexe 1 : Résolution de l'équation polynomiale de Bezout)

On voit que le numérateur  $B \cdot R$  est imposé une fois que  $R$  et  $S$  sont trouvés. Le régulateur introduit donc naturellement des zéros dans la boucle fermée.

Traisons l'exemple entièrement dans le cas où le système est modélisé comme un système du deuxième ordre avec faible retard. La fonction de transfert du procédé est donnée par :

$$H_{BO} = \frac{B}{A} = \frac{b_1 \cdot q^{-1} + b_2 \cdot q^{-2}}{1 + a_1 \cdot q^{-1} + a_2 \cdot q^{-2}} \quad (66)$$

Le correcteur PID rajoutant deux pôles, le polynôme caractéristique en boucle fermée spécifié doit posséder quatre pôles. Le polynôme caractéristique en boucle fermée doit donc être de la forme :

$$\begin{aligned} P &= 1 + p_1 \cdot q^{-1} + p_2 \cdot q^{-2} + p_2 \cdot q^{-3} + p_2 \cdot q^{-4} \\ P &= (1 + p_1' \cdot q^{-1} + p_2' \cdot q^{-2}) \cdot (1 + \alpha_1 \cdot q^{-1}) \cdot (1 + \alpha_2 \cdot q^{-1}) \end{aligned} \quad (67)$$

Et il doit être identifié au produit  $AS + BR$  avec  $R$  et  $S$  ayant les formes données par (64) :

$$\begin{aligned} &(1 + p_1' \cdot q^{-1} + p_2' \cdot q^{-2}) \cdot (1 + \alpha_1 \cdot q^{-1}) \cdot (1 + \alpha_2 \cdot q^{-1}) \\ &= (1 + a_1 \cdot q^{-1} + a_2 \cdot q^{-2}) \cdot (1 - q^{-1}) \cdot (1 + s_1' \cdot q^{-1}) \\ &+ (b_1 \cdot q^{-1} + b_2 \cdot q^{-2}) \cdot (r_0 + r_1 \cdot q^{-1} + r_2 \cdot q^{-2}) \end{aligned} \quad (68)$$

Les coefficients  $p_1'$  et  $p_2'$  définissent un modèle du second ordre aux caractéristiques (pulsation propre  $\omega_0$  et amortissement  $\sigma$  choisis et coefficients déduits conformément au Tableau 3 page 25). Les coefficients  $\alpha_1$  et  $\alpha_2$  correspondent à des pôles auxiliaires aperiodes. Les choix typiques pour ces coefficients sont :

$$-0.05 < \alpha_1, \alpha_2 < -0.5 \quad (69)$$

On montre que la résolution de (68) est réalisée par une inversion matricielle simple :

$$M \cdot x = p \quad (70)$$

avec

$$\begin{cases} x^T = [1 & s_1' & r_0 & r_1 & r_2] \\ p^T = [1 & p_1 & p_2 & p_3 & p_4] \end{cases}$$

et

$$M = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ a_1' & 1 & b_1 & 0 & 0 \\ a_2' & a_1' & b_2 & b_1 & 0 \\ a_3' & a_2' & 0 & b_2 & b_1 \\ 0 & a_3' & 0 & 0 & b_2 \end{pmatrix} \quad (71)$$

On montre que M est inversible à la condition que A et B soient premiers entre eux, c'est-à-dire que A et B n'aient pas de pôles et de zéros en commun.

(Le problème peut aussi être résolu en utilisant la résolution de l'équation de Bézout).

### 7.3.2 Cas 2

Comme on l'a vu le correcteur PI numérique directement transposé du correcteur PID analogique décrit dans le chapitre 7.3.1 a l'inconvénient de ramener des zéros au dénominateur de la fonction de transfert en boucle fermée (équation 65). Il est possible, grâce à la flexibilité du régulateur numérique d'imposer au correcteur de ne pas ajouter de zéros en boucle fermée. La fonction de transfert désirée en boucle fermée sera donc de la forme :

$$H_{BF} = \frac{P(1)}{B(1)} \cdot \frac{B}{P} \quad (72)$$

Le facteur  $P(1)/B(1)$  garantit le gain statique unitaire du système en boucle fermée (voir page 12). Si le régulateur a la forme générale RST alors les polynômes R S et T doivent vérifier :

$$H_{BF} = \frac{B \cdot T}{A \cdot S + B \cdot R} = \frac{P(1) B}{B(1) P} \quad (73)$$

S et R sont obtenus par la résolution de la même équation que précédemment (68). Le polynôme T est obtenu par simple identification :  $T = R(1)$ .

La structure, sous forme de schéma-bloc est donnée ci-dessous :

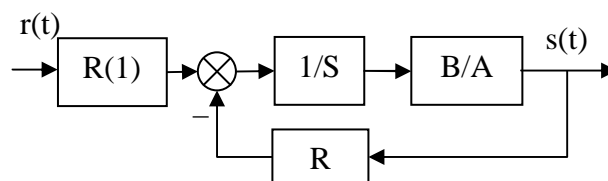


Figure 31 : PID2 sous forme RST

On montre que cette structure de PID, ramené en continu, correspond au PID canonique pour l'action intégrale. L'action proportionnelle et dérivée n'est appliquée que sur le signal  $y$ .

La structure du PID2 par rapport au PID1 a pour avantage de ne pas ramener de zéros à la boucle fermée. Cela se traduit par un dépassement moindre en transitoire lorsqu'on applique un échelon de commande. Par contre les deux systèmes réagissent de la même manière à un échelon de perturbation.



## 7.4 Placement de pôles

Les deux structures de PID décrites dans le chapitre précédent sont deux cas particulier d'une stratégie plus générale de placement de pôles.

La synthèse est plus générale que les PID présentés précédemment puisqu'il n'est pas nécessaire de faire d'hypothèses sur le degré de A et B, sur le retard, ni sur la présence ou non de zéros stables ou instables dans le procédé.

La structure du correcteur est la structure type RST telle que présentée dans la Figure 29. Pour la suite des calculs on posera pour la fonction de transfert les conventions suivantes :

$$H_{BO} = \frac{q^{-d} \cdot B(q^{-1})}{A(q^{-1})} \quad (74)$$

$$\begin{cases} A(q^{-1}) = 1 + a_1 \cdot q^{-1} + \dots + a_{n_A} \cdot q^{-n_A} \\ B(q^{-1}) = b_1 \cdot q^{-1} + \dots + b_{n_B} \cdot q^{-n_B} = q^{-1} \cdot B^*(q^{-1}) \end{cases}$$

Compte tenu de la forme RST du régulateur la fonction de transfert en boucle fermée vérifie :

$$H_{BF} = \frac{q^{-d} \cdot B(q^{-1}) \cdot T(q^{-1})}{P(q^{-1})} \quad (75)$$

$$P(q^{-1}) = A(q^{-1}) \cdot S(q^{-1}) + q^{-d} \cdot B(q^{-1}) \cdot R(q^{-1}) = 1 + p_1 \cdot q^{-1} + \dots + p_{n_P} \cdot q^{-n_P}$$

Le polynôme  $P(q^{-1})$  définit donc les pôles de la fonction de transfert en boucle fermée et traduit donc le comportement en *poursuite*.

De la même manière la fonction de transfert entre une perturbation et la sortie (fonction de sensibilité perturbation-sortie) est donnée par :

$$S_{yp} = \frac{A(q^{-1}) \cdot S(q^{-1})}{P(q^{-1})} \quad (76)$$

Ce même polynôme traduit donc aussi le comportement en *régulation*.

Les consignes en poursuite et en régulation peuvent être différentes comme cela est représenté par exemple sur la Figure 32.

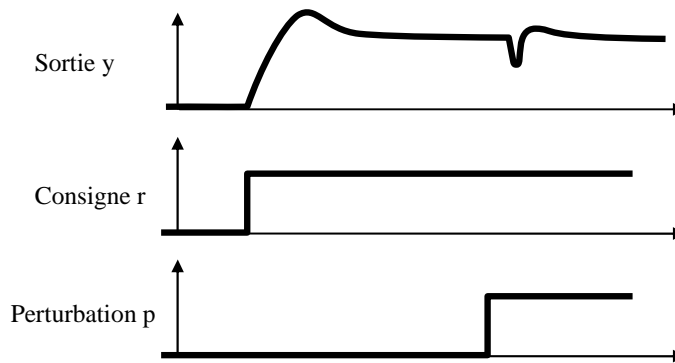


Figure 32 : Performances différentes en régulation et en poursuite

### 7.4.1 Choix des pôles en boucle fermée

De manière générale le comportement en boucle fermée est spécifié sous la forme d'une pulsation propre  $\omega_0$  et d'un amortissement  $\sigma$ , donc une fonction de transfert du second ordre. La spécification étant généralement donnée en continu il est nécessaire de traduire la valeur désirée de  $\omega_0$  et  $\sigma$  en un polynôme  $P_D(q^{-1})$  par discrétisation de la fonction de transfert continue (chapitre 4.5 page 29). Ce polynôme  $P_D$  traduit le comportement dominant du système bouclé.

Si l'on souhaite introduire un effet de filtrage supplémentaire dans certaines bandes de fréquence (réduction des bruits de mesure, adoucissement de la commande, robustesse) cela est décrit par un polynôme  $P_F$  comportant des pôles auxiliaires, plus rapides que les pôles de  $P_D$ .

Le polynôme caractéristique en boucle fermée est donc constitué de pôles dominants et de pôles auxiliaires :

$$P(q^{-1}) = P_D(q^{-1}) \cdot P_F(q^{-1}) \quad (77)$$

### 7.4.2 Régulation : calcul de R et S

Le polynôme  $P(q^{-1})$  étant fixé il s'agit de trouver R et S vérifiant :

$$P(q^{-1}) = A(q^{-1}) \cdot S(q^{-1}) + q^{-d} \cdot B(q^{-1}) \cdot R(q^{-1}) \quad (78)$$

Comme déjà vu au chapitre 7.3.1 on obtient une équation de Bézout qu'il s'agit de résoudre. (Voir aussi 9.1 Annexe 1 : Résolution de l'équation polynomiale de Bezout)

Il est parfois nécessaire de spécifier R et S avec des parties fixes  $H_R$  et  $H_S$ . Dans ce cas l'équation (78) est modifiée et on l'écrit sous la forme:

$$P(q^{-1}) = (A(q^{-1}) \cdot H_S(q^{-1})) \cdot S'(q^{-1}) + (q^{-d} \cdot B(q^{-1}) \cdot H_R(q^{-1})) \cdot R'(q^{-1}) \quad (79)$$

Les inconnues que l'on souhaite trouver sont  $S'$  et  $R'$ . Ceci est nécessaire dans les cas particuliers suivants :

- spécification d'erreur statique nulle,
- rejet d'une perturbation harmonique,
- blocage d'un signal,

Détaillons maintenant ces cas particuliers.

### Spécification d'une erreur statique nulle

On retrouve la nécessité d'avoir un terme intégrateur dans la boucle directe. La spécification se retrouve donc sur  $S$  :

$$S(q^{-1}) = (1 - q^{-1}) \cdot S'(q^{-1}) = H_S(q^{-1}) \cdot S'(q^{-1}) \quad (80)$$

### Rejet d'une perturbation harmonique

Si le système est perturbé par une perturbation harmonique à une fréquence fixe (par exemple 50Hz...) il est possible d'imposer à  $S$  d'avoir un module *nul* ou amorti à cette fréquence. Cela se traduit par la présence dans  $S$  d'une paire de zéros complexe à amortissement nul ou faible à cette fréquence :

$$S(q^{-1}) = (1 + \alpha \cdot q^{-1} + \beta \cdot q^{-2}) \cdot S'(q^{-1}) = H_S(q^{-1}) \cdot S'(q^{-1}) \quad (81)$$

### Blocage d'un signal

Pour certaines application on souhaite que le régulateur "laisse passer" certaines composantes du signal de sortie, c'est-à-dire qu'il ne réagisse pas à ces composantes. Cela se traduit par  $R=0$  à ces fréquences. On retrouve la même forme que précédemment :

$$R(q^{-1}) = (1 + \alpha \cdot q^{-1} + \beta \cdot q^{-2}) \cdot R'(q^{-1}) = H_R(q^{-1}) \cdot R'(q^{-1}) \quad (82)$$

### 7.4.3 Poursuite : calcul de T

Le schéma-bloc de l'asservissement est donné sur la Figure 33. Le terme  $T$  est décomposé en un terme  $T'$  et un terme  $B^*/A^*$ .

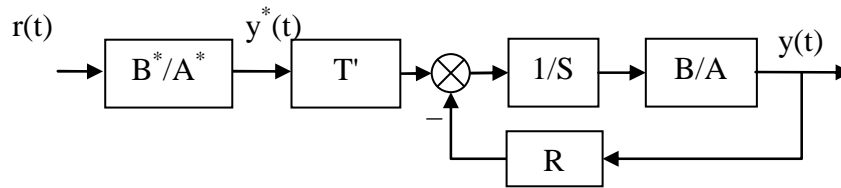


Figure 33 : placement de pôle

L'idée est de régler  $B^*$  et  $A^*$  de telle manière que le signal  $y^*$  ait la dynamique de poursuite désirée (par exemple un second ordre avec pulsation propre et amortissement donné) puis de régler  $T'$  de telle manière que le transfert entre  $y^*$  et  $y$  soit le plus "transparent" possible.

### Réglage de $T'$

Le transfert entre  $y^*$  et  $y$  est :

$$S_{yy^*}(q^{-1}) = T'(q^{-1}) \frac{q^{-d} \cdot B(q^{-1})}{P(q^{-1})} \quad (83)$$

Pour assurer le transfert le plus "transparent" possible il faut donc choisir

$$T'(q^{-1}) = \frac{P(q^{-1})}{B(1)} \quad (84)$$

Remarques :

- Si  $B(1)=0$  alors on remplace  $B(1)$  par 1 dans (84)
- De même dans (84)  $P(q^{-1})$  peut être remplacé par  $P_D$  correspondant à la dynamique dominante.

### Réglage de $B^*$ et $A^*$

$B^*$  et  $A^*$  sont choisis de telle manière que la fonction de transfert  $B^*/A^*$  corresponde à la dynamique de poursuite souhaitée.

## 7.5 Poursuite et régulation à objectifs indépendants

Cette méthode ressemble à la méthode de placement de pôles à ceci près que l'on va effectuer une simplification des zéros de  $B^*(q^{-1})$  : cela ne s'applique donc qu'aux systèmes avec des pôles stables. Plus précisément il faut qu'ils aient un amortissement suffisant ( $> 0,2$ ) de manière à ne pas être trop près du cercle unité.

La structure de l'asservissement est toujours celle donnée sur la Figure 33.

**Choix de P**

$P(q^{-1})$  est choisi de manière à simplifier  $B^*(q^{-1})$  (rappel : la forme de B est donnée dans l'équation 74). Comme indiqué dans le chapitre 7.4.2 page 60 il s'agit donc d'imposer les zéros de  $B(q^{-1})$  dans  $H_R(q^{-1})$  et  $H_S(q^{-1})$ .

**Choix de T**

Si l'on choisit  $T(q^{-1}) = P(q^{-1})$  alors le transfert entre  $y^*$  et  $y$  est :

$$S_{yy^*}(q^{-1}) = q^{-(d+1)} \quad (85)$$

**Réglage de  $B^*$  et  $A^*$** 

$B^*$  et  $A^*$  sont choisis de telle manière que la fonction de transfert  $B^*/A^*$  corresponde à la dynamique de poursuite souhaitée.



## 8. TECHNOLOGIE DE LA COMMANDE PAR ORDINATEUR

Du point de vue de l'automaticien le ordinateur est relié au système physique à piloter par un certain nombre d'intermédiaires plus ou moins gênants :

- les interfaces analogiques et numériques,
- le transport de l'information par bus et protocole,
- l'utilisation éventuelle de capteurs ou d'actionneurs « intelligents ».

Il ne s'agit pas ici de faire un exposé complet sur ces trois aspects mais de poser les problèmes qui se posent à l'automaticien lorsque le système qu'il doit commander est perturbé par ces éléments.

Enfin ce chapitre se termine sur les notions de « Hardware In the Loop » et de « prototypage rapide » qui sont partie intégrante des processus de réalisation de systèmes de contrôle.

### 8.1 Calculateurs et interfaces numériques / analogiques

Le premier problème à prendre en compte est la quantification de l'information. Du point de vue du ordinateur d'une part et du point de vue des convertisseurs d'autre part. Dans un premier temps, considérons les aspects performance de calcul lié à la nature du codage des nombres dans le ordinateur.

#### Les performances liées au types de variables

Prenons l'exemple où le concepteur de la loi de commande souhaite réaliser un correcteur de type proportionnel. Entre l'acquisition des mesures et l'écriture de la consigne il y aura la réalisation d'une de commande du type :

$$y = K \cdot (\text{consigne} - \text{mesure}) \quad (86)$$

En utilisant un langage de programmation évolué (par exemple le C) la ligne de code sera la transcription littérale de (86). Le programmeur devra décider de la définition et du type des variables, principalement du nombre d'octets utilisés pour stocker ses variables. Le Tableau 5 résume par exemple les types utilisés en langage C (norme ANSI)

Tableau 5 : Types ANSI C

Types entiers	char	8 bits
	short int	16 bits
	long int	32 bits
Types flottants (mantisse et exposant)	float	32 bits
	double	64 bits
	longdouble	128 bits

Une fois le code écrit en C l'exécution du programme sur un processeur cible ne peut se faire que par l'intermédiaire d'une compilation du programme. A chaque processeur est donc adapté une version de compilateur.

Les performances du programme (sa rapidité d'exécution) dépendront donc en partie de la qualité du compilateur. (Par exemple, une multiplication s'effectue t'elle bêtement par un décalage de tous les bits ou bien des solutions astucieuses sont-elles implémentées pour arrêter le décalage selon le contexte ?)

Le principal déterminant de la performance est cependant le processeur : son type (8 bits, 16 bits ou 64 bits) et son cadencement (en MHz ou GHz) sont les deux plus importants paramètres. Coder les variables sur 64 bits (double) dans le programme coûtera cher en temps de calcul si le processeur est un 8 bits. Le type de mémoire et leurs temps d'accès peuvent aussi être déterminants. Enfin, certains processeurs disposent d'un processeur spécifiquement dédié aux opérations mathématiques (coprocesseur mathématique).

Il n'est pas possible de dire exactement, pour tel type de compilateur associé à tel type de processeur, le temps que prendra par exemple une multiplication. En pratique on effectue une série de tests types (benchmark) et on affiche un temps moyen d'exécution.

En résumé, on peut être tenté, pour ne pas s'embêter, de coder toutes les variables en virgule flottante. Mais il faut bien savoir que cela a un coût en terme de temps de calcul, surtout si le processeur n'est pas adapté. Pour fixer les idées on peut dire qu'il y a un rapport 20 entre une addition réalisée en "short" et une addition réalisée en virgule flottante.

Enfin, l'utilisation de fonctions mathématiques (sinus, exponentielles, etc...) peut aussi avoir un coût très lourd si le travail n'est pas réalisé par un coprocesseur mathématique dédié. Le compilateur et sa librairie mathématique joue là encore un grand rôle.

### La quantification imposée par le processeur

Au-delà de l'aspect performance et temps de calcul, la quantification de l'information doit parfois aussi être prise en compte. Rappelons qu'un entier codé sur N bits peut prendre seulement  $2^N - 1$  valeurs différentes. Avant d'effectuer les calculs mathématiques dans le corps du programme il est donc judicieux d'appliquer un



facteur d'échelle de manière à utiliser l'amplitude maximale possible (entre 0 et  $2^N - 1$ ).

### **La quantification imposée par les conversions analogique et numérique**

Les convertisseurs analogique-numérique et numérique-analogique sont principalement caractérisés par leur amplitude de signal analogique d'entrée (généralement en volts, par exemple +/- 10V) et par leur résolution en bits (8 ou 10 bits pour les plus courants, jusqu'à 16 bits pour les convertisseurs haut de gamme).

Prenons l'exemple d'un capteur gyromètre transformant une vitesse de rotation en tension. Compte tenu de l'application le signal utile varie entre 2,4V et 2,6V. Si le signal mesuré est branché directement sur un convertisseur 0-10V 10bits alors un problème de conditionnement du signal se pose. En effet la quantification est de  $\frac{10V}{2^{10}-1} = 0,098V$ . Les 0,2V de signal utile seront donc discrétisés en  $\frac{0,2}{0,098} = 20$  intervalles seulement !

Si la discrétisation du signal utile est trop grossière cela peut se traduire par une diminution des performances de l'asservissement, voire une instabilité. Le système n'est en effet plus linéaire et il n'existe pas d'outils théoriques permettant de prévoir les performances du système bouclé.

## **8.2 Transport de l'information : les bus et les protocoles**

Pas question de faire ni même d'ébaucher un cours sur les bus et les protocoles ici. Mais on va quand même donner quelques indications. L'automaticien a l'habitude de simplifier le problème qu'il se pose en dessinant un schéma simplifié de son problème d'asservissement (du type de la Figure 20 page 45) pour pouvoir appliquer les théories de la commande. Cependant il arrive bien souvent que l'information transite à l'intérieur du système, soit entre différents ordinateurs effectuant différentes parties du travail soit entre ordinateur et périphériques, par l'intermédiaire de bus informatique. A chaque type de bus correspond un ou plusieurs protocoles qui peuvent poser des problèmes pour l'automaticien :

- les données sont-elles discrétisées ? (le protocole peut par exemple ne permettre de transmettre que des paquets de données de 16 bits),
- le bus induit-t'il un retard ? (le composant mettant en forme les données du ordinateur pour les publier sur le bus a certaines performances qui peuvent induire un délai)
- le bus est-il déterministe ? (Certains bus ne garantissent pas que les données arriveront en un temps donné),
- le bus est-il sûr ? (Certains bus ne garantissent pas non plus que les données arriveront bien)

### **8.3 Capteurs et actionneurs "intelligents", "smart sensors"**

De plus en plus souvent les capteurs et les actionneurs sont associés à leur électronique de puissance, électronique de commande et de conditionnement de signal. L'interface avec le capteur ou l'actionneur se fait donc via un protocole de communication (par exemple via un bus RS232).

L'intérêt est de véhiculer non pas des signaux analogiques, tensions ou courants, mais des signaux numériques beaucoup moins sensibles aux éventuelles perturbations.

On retrouve donc, entre le capteur ou l'actionneur et le ordinateur, la présence d'un bus et d'un protocole de communication.

### **8.4 La philosophie "hardware in the loop"**

Les systèmes embarqués sont de plus en plus complexes et touchent des domaines de plus en plus variés : automobile, satellites, aéronautique, UAV, systèmes d'arme, sous-marins, systèmes, etc... Les systèmes à piloter ainsi que les systèmes de commande sont de plus en plus complexes.

Les techniques de simulation « Hardware In the Loop » (HIL) sont de plus en plus utilisées dans le développement et dans le test de ces systèmes embarqués. L'idée est de développer entièrement le système de contrôle sur une plateforme de test en utilisant le maximum de composants réels et en ne remplaçant que le minimum par la simulation. Par exemple en aéronautique l'« Iron Bird » regroupe dans un hangar tous les systèmes réels de l'avion en cours de développement. Seule la mécanique du vol est simulée : les sorties des actionneurs sont mesurées, rentrées dans un modèle de simulation numérique, puis les mesures effectuées normalement par les capteurs sont calculées et restituées par le modèle.

L'intérêt est de diminuer de manière importante les coûts et les délais en économisant le plus grand nombre possible de tests sur le système réel en situation. Lors des premiers tests réels les techniques « Hardware In the Loop » permettent d'avoir effectué la plus grande part du développement des lois de commande.

### **8.5 Le prototypage rapide en automatique**

Un peu dans le même ordre d'idée que le « Hardware In the Loop » le prototypage rapide permet de réaliser la conception des lois de commande en simulation puis de transporter rapidement (en un simple click !) le code du contrôleur simulé vers la cible temps réel implémentée sur le système réel.

La société Mathworks [9] investit massivement dans ce domaine en ayant enrichi son logiciel de calcul scientifique (Matlab) avec un logiciel de simulation graphique très évolué (Simulink) puis en développant des outils de prototypage rapide associés (Real Time Workshop, Embedded Target For...).

## 9. ANNEXE

### 9.1 Annexe 1 : Résolution de l'équation polynomiale de Bezout

Il s'agit de trouver R et S vérifiant :

$$P(q^{-1}) = A(q^{-1}).S(q^{-1}) + B(q^{-1}).R(q^{-1})$$

Où P, A et B sont connus.

Dans le cas où P, A, B, R et S sont des nombres entiers la solution de l'algorithme est donnée par l'algorithme de division Euclidienne étendu [8].

Dans le cas où P, A, B, R et S sont des polynômes, l'algorithme est donné ci-dessous, programmé en matlab ([1]).

```
function [Rp,Sp,nrp,nsp]=bezoutd(A,B,Hs,Hr,P)
%function [Rp,Sp,nrp,nsp]=bezoutd(A,B,Hs,Hr,P)
%solves AHsSp+BHrRp=P by coefficient comparison. Delay and discretization delay need to be integrated in B
%(zeros at the beginning).
%
%inputs:
%A=[a0 a1 ... aNa] ... vector of model denominator coefficients A=a0 + a1z^(-1) + a2z^(-2) +...+ aNaz^(-Na)
%B=[b0 b1 ... bNb] ... vector of model numerator coefficients B=b0 + b1z^(-1) + b2z^(-2) +...+ bNbz^(-Nb)
%Hs=[hs0 hs1 ... hsNhs] ... vector of controller denominator fixed part Hs=hs0 + hs1z^(-1) +...+ hsNhsz^(-Nhs)
%Hr=[hr0 hr1 ... hrNhr] ... vector of controller denominator fixed part Hr=hr0 + hr1z^(-1) +...+ hrNhrz^(-Nhr)
%P=[p0 p1 ... pNp] ... vector of desired polynomial coefficients P=p0 + p1z^(-1) + p2z^(-2) +...+ pNpz^(-Np)
%outputs:
%Rp=[rp0 rp1 rp2 ...] ... vector of coefficients for resulted controller numerator
%Sp=[sp0 sp1 sp2 ...] ... vector of coefficients for resulted controller denominator
%nrp ... order of Rp
%nsp ... order of Sp
%
%written by: J. Langer, I.D. Landau, H. Prochazka
%7th june 2002

PRECISION=1e-16;

D=size(A);
if D(1)>1, A=A'; end;
```

```

D=size(B);
if D(1)>1, B=B'; end;
D=size(Hs);
if D(1)>1, Hs=Hs'; end;
if D(1)==0, Hs=1; end;
D=size(Hr);
if D(1)>1, Hr=Hr'; end;
if D(1)==0, Hr=1;end;
D=size(P);
if D(1)>1, P=P'; end;

na=length(A)-1;
nb=length(B)-1;
np=length(P)-1;
nhs=length(Hs)-1;
nhr=length(Hr)-1;

if (nhs>0), Ah=real(convz(A,Hs)); else Ah=A*Hs; end; % Ah = A * Hs
nah=length(Ah) -1;
if (nhr>0), Bh=real(convz(B,Hr)); else Bh=B*Hr; end;
nbh=length(Bh) -1;
if (np>nah+nbh-1), disp('Bezout error: too many poles');end;

% increase size of P using zeros if necessary
if (np<nah+nbh-1),
%
% set remaining poles onto a circle with radius rmin
P=[P zeros(1,nah+nbh-1-np)];
rootsPdes=roots(P);
nextra=nah+nbh-1-np;
rmin=1e-16;
angle=[0:nextra-1]'/nextra*2*pi;
j=sqrt(-1);
rootsPextra=rmin*exp(j*angle);
P=poly([rootsPdes;rootsPextra]);
np=nah+nbh-1;

end;
P,

nsp=nbh-1;
nrp=nah-1;

%matrix is smaller than vector PD

```

```

if (np>nah+nbh-1),
    disp('Order of model denominator is too low! Add a polynom of higher order to Hs or Hr. ');
end;

% ns=nsp+nhs
% nr=nrp+nhr

M=[];
for j=1:nsp+1,
    V=[];
    if (j>1), V=[V ; zeros(j-1,1)]; end;% zeros in front of Ah
    V=[V ; Ah'];% Ah
    if (j<=nsp), V=[V ; zeros(nsp+1-j,1)]; end;% zeros behind Ah
    if (length(V)~=nah+nbh), disp('bezoutb: error V'); end;
    M=[M V]; % add one column to M
end;

for j=1:nrp+1,
    V=[];
    if (j>1), V=[V ; zeros(j-1,1)]; end;
    V=[V ; Bh'];
    if (j<=nrp), V=[V ; zeros(nrp+1-j,1)]; end;
    if (length(V)~=nah+nbh), disp('bezoutb: error V'); end;
    M=[M V];
end;

D=size(M);
if (D(1)~=nah+nbh), disp('bezoutb: error size M row'); end;
if (D(2)~=nah+nbh), disp('bezoutb: error size M column'); end;

% make P column vector
P=P';

global M1;
M1=M;

X= M\P;
% coefficients are real values
X=real(X);

% make X row vector
X=X';

```

## Commande des systèmes par ordinateur

```
Sp=X(1:nsp+1);  
Rp=X(nsp+2:nsp+nrp+2);
```

## 10. BIBLIOGRAPHIE

- [1] I. D. Landau, "*Commande des systèmes*", Edition Lavoisier 2002.
- [2] C. Nouals, "Théorie du Signal", Notes de cours ENSICA
- [3] F. Vincent, "*Traitement Numérique du Signal*", Notes de cours ENSICA.
- [4] D. Arzelier, "*Représentation et analyse des systèmes, notes de cours*", Edition ENSICA. Voir aussi <http://www.laas.fr/~arzelier/>
- [5] J. Bordeneuve-Guibé, "*Commande des systèmes linéaires*", Notes de cours ENSICA.
- [6] Y. Briere, "*Identification*", Notes de cours ENSICA.
- [7] L. Ljung, "*System Identification, Theory for the User*", Prentice Hall 1987.
- [8] Wikipedia , équation de Bezout : [http://en.wikipedia.org/wiki/Bézout's\\_identity](http://en.wikipedia.org/wiki/Bézout's_identity)
- [9] <http://www.mathworks.com>





# 11. INDEX

<b>A</b>	
actionneur intelligent .....	21
amortissement .....	15
ANSI C .....	60
Approximation numérique d'équation différentielle .....	28
ARMAX .....	35
<b>B</b>	
BJ35 .....	
Blocage d'un signal .....	56
bloqueurs d'ordre 0 .....	22
BOZ .....	22
bus .....	21, 61
<b>C</b>	
CAN .....	21
capteur intelligent .....	21
causal .....	6, 9
Causalité .....	10
CNA .....	21
compilateur .....	60
convertisseur analogique / numérique .....	21
convertisseur numérique / analogique .....	21
convertisseurs analogique-numérique .....	61
coprocesseur mathématique .....	60
Correspondances pôles - zéros .....	31
<b>D</b>	
Diagramme de Bode .....	18
Discrétisation de fonction de transfert continues .....	27
division polynomiale .....	6
<b>E</b>	
échelon unitaire .....	11
équation d'état .....	17
Erreur statique .....	42
Estimation de la réponse fréquentielle .....	39
Estimation de la réponse impulsionnelle .....	38
Euler .....	29
<b>F</b>	
facteur d'échelle .....	61
facteur d'oubli .....	36
<i>fenêtre de Hamming</i> .....	39
filtre anti-repliement .....	25
fonction de sensibilité .....	42
fonction de transfert discrète .....	9
<b>G</b>	
gain statique .....	11
gain statique unitaire .....	42
<b>I</b>	
Identification .....	32
Identification non paramétrique .....	38
intercorrélation .....	38
<b>L</b>	
lieu des racines .....	43
lieu d'Evans .....	43
LTI .....	8
<b>M</b>	
marge de gain .....	47
marge de module .....	46
marge de phase .....	47
méthode des moindres carrés récursifs .....	32
Méthode des trapèzes .....	29
modèle ARX .....	32
<b>O</b>	
OE .....	35
one step ahead prediction .....	35
opérateur de retard .....	9
<b>P</b>	
période d'échantillonnage (choix) .....	26
Placement de pôles .....	54
Pôles .....	11
Poursuite .....	56
<i>prédiction</i> .....	32
protocoles .....	61
Pseudo Random Binary Signal .....	34
pulsation .....	15
<b>Q</b>	
quantification .....	23, 24, 60
<b>R</b>	
R,S,T .....	41
rapidité .....	15
rayon de convergence .....	5, 9
Régulation .....	55

Rejet de perturbations .....	42
Rejet d'une perturbation harmonique .....	56
réponse à un échelon.....	11
réponse fréquentielle.....	18
Réponse impulsionnelle.....	10
Réponses indicielle .....	10
retard.....	23
root locus .....	43

### S

signal discret .....	5
smart sensors.....	62
stabilité.....	11
Systèmes du premier ordre.....	14
Systèmes du second ordre.....	15

### T

Théorème de Nyquist.....	46
théorème de Shannon.....	19, 24
théorème du retard .....	6
Transformée en z .....	5
transformée unilatérale .....	5
Tustin.....	29

### V

virgule flottante.....	60
------------------------	----

### Z

zero holder hold .....	23
zéros.....	11
zoh .....	23

# Commande des systèmes par ordinateur

Yves Briere  
ENSICA

Notes de cours  
2007



# TABLE DES MATIERES

Table des matières .....	1
1. Introduction .....	1
2. Rappels et prérequis .....	3
3. Signaux et systèmes discrets .....	5
3.1 Signaux discrets .....	5
3.1.1 Transformée en z .....	5
3.1.2 Signaux types .....	7
3.2 Systèmes discrets .....	9
3.2.1 Fonction de transfert discrète, définition .....	9
3.2.2 Notation "opérateur de retard $q^{-1}$ " .....	10
3.2.3 Réponse impulsionnelle .....	11
3.2.4 Causalité .....	11
3.2.5 Réponses indicielle .....	11
3.2.6 Régime stationnaire (gain statique) .....	12
3.2.7 Pôles, zéros, stabilité .....	12
3.2.8 Systèmes du premier ordre .....	15
3.2.9 Systèmes du second ordre .....	16
3.2.10 Calcul itératif de la réponse d'un système .....	18
3.2.11 Représentation par équation d'état .....	18
3.3 Réponse fréquentielle des systèmes échantillonnés .....	19
3.3.1 Diagramme de Bode d'un système échantillonné .....	19
3.3.2 Où l'on voit apparaître le théorème de Shannon .....	20
4. Systèmes continus échantillonnés .....	23
4.1 Structure des systèmes commandés par ordinateur .....	23
4.2 Échantillonnage des systèmes continus .....	24
4.3 Effets de l'échantillonnage .....	25
4.4 Où l'on revoit le théorème de Shannon .....	26
4.5 Discrétisation de fonction de transfert continues .....	29
4.5.1 Approximation numérique d'équation différentielle .....	30
4.5.2 Superposition des réponses indicielles .....	32
4.5.3 Correspondances pôles - zéros .....	33
4.5.4 Bilan .....	33
5. Identification des systèmes échantillonnés .....	35
5.1 Identification d'un modèle ARX par la méthode des moindres carrés récurrents .....	35
5.2 L'identification paramétrique en général et en pratique .....	37

5.2.1	Choix du signal d'excitation, traitement des mesures .....	37
5.2.2	Choix d'une structure de modèle.....	38
5.2.3	Choix d'un critère à minimiser.....	38
5.2.4	Choix d'un algorithme.....	38
5.2.5	Validation.....	39
5.2.6	Utilisation de logiciels .....	39
5.3	Identification "simple" .....	39
5.4	Identification non paramétrique .....	41
5.4.1	Estimation de la réponse impulsionnelle .....	41
5.4.2	Estimation de la réponse fréquentielle.....	42
6.	Systèmes bouclés .....	45
6.1	Fonction de transfert en boucle fermée .....	45
6.2	Erreur statique .....	46
6.3	Rejet de perturbations.....	46
6.4	Stabilité des systèmes à simple gain de rétroaction.....	47
6.5	Analyse des systèmes bouclés dans le domaine fréquentiel : stabilité.....	49
6.5.1	Théorème de Nyquist dans le cas où le système est stable en boucle ouverte (critère du revers) .....	50
6.5.2	Théorème de Nyquist dans le cas où le système est instable en boucle ouverte .....	50
6.5.3	Interprétations du lieu de Nyquist.....	50
7.	Méthodes de régulation des systèmes échantillonnés .....	53
7.1	Introduction .....	53
7.2	Structure d'un système commandé par ordinateur, hypothèses.....	54
7.2.1	Discretisation, choix de la période d'échantillonnage.....	54
7.2.2	Discretisation d'un correcteur continu .....	55
7.3	Régulateur PID numérique .....	55
7.3.1	Cas 1 .....	55
7.3.2	Cas 2 .....	58
7.4	Placement de pôles .....	59
7.4.1	Choix des pôles en boucle fermée .....	60
7.4.2	Régulation : calcul de R et S.....	60
7.4.3	Poursuite : calcul de T .....	61
7.5	Poursuite et régulation à objectifs indépendants .....	62
8.	Technologie de la commande par ordinateur.....	65
8.1	Calculateurs et interfaces numériques / analogiques.....	65
8.2	Transport de l'information : les bus et les protocoles .....	67
8.3	Capteurs et actionneurs "intelligents", "smart sensors" .....	68
8.4	La philosophie "hardware in the loop" .....	68
8.5	Le prototypage rapide en automatique .....	68
9.	Annexe .....	69
9.1	Annexe 1 : Résolution de l'équation polynomiale de Bezout.....	69
10.	Bibliographie.....	73
11.	Index.....	1







# 1. INTRODUCTION



Figure 1 : commande d'un systèmes de télémédecine par ordinateur, vu par Claude Serre



## 2. RAPPELS ET PREREQUIS

Ce polycopié ne comporte pas de rappels sur les systèmes continus. S'adressant aux étudiants en deuxième année de l'ENSICA il suppose connu les enseignements de première année et de seconde année en automatique et traitement du signal.

En première année le cours de théorie du signal [2] a rendu familier les notions de :

- transformée de Fourier, transformée de Laplace
- corrélation, convolution

Le cours de traitement numérique du signal [3] a abordé en plus :

- la discrétisation et la quantification des signaux.
- la transformée en z et la transformée de Fourier discrète.
- les filtres numériques RIF et RII.

En automatique [4] les bases de la représentation et de l'analyse des systèmes continus ont été posées. On ne reviendra pas dans ce cours sur les notions de :

- modèles mathématique des systèmes linéaires à temps continus à temps invariant (LTI) : fonction de transfert, équation d'état, pôles, zéros et dynamique.
- commandabilité et observabilité.
- stabilité des systèmes en boucle ouverte : pôles, critères de Routh-Hurwitz.
- stabilité des systèmes bouclés : critère de Nyquist, critère de Bode.
- lieu des racines.

En deuxième année les notions suivantes sont acquises [5] :

- synthèse des correcteurs par méthodes fréquentielles (avance et retard de phase, PID)
- commande par retour d'état (placement de pôles), commande Linéaire Quadratique (LQG)
- reconstruction de l'état par Observateur, observateur optimal de Kalman.



## 3. SIGNAUX ET SYSTEMES DISCRETS

### 3.1 Signaux discrets

#### 3.1.1 Transformée en z

Pour une approche mathématiquement rigoureuse de représentation des signaux discrets par la transformée en z consulter [3] et sa bibliographie. Dans ce cours on se cantonne à une approche utilitariste des outils et méthodes mathématiques tout en essayant de ne pas contredire fondamentalement la rigueur mathématique.

Un signal discret  $s(k)$  est une séquence de nombres réels appelés échantillons. L'indice  $k$  est généralement associé à un instant d'échantillonnage :  $t = k.T_e$ . La période d'échantillonnage  $T_e$  est supposée constante. Par définition la transformée en z du signal  $s(k)$  est la fonction de la variable  $z$ , notée abusivement  $s(z)$  :

$$s(z) = Z(s(k)) = \sum_{k=0}^{\infty} s(k) \cdot z^{-k} \quad (1)$$

La transformée définie ici est plus précisément appelée transformée unilatérale puisque les termes correspondant à  $k < 0$  ne sont pas pris en compte. Ceci n'a pas d'importance puisque les signaux que nous utilisons sont systématiquement causaux<sup>1</sup> :  $s(k) = 0$  pour  $k < 0$ .

Il y a bien évidemment un lien avec la transformée de Laplace et la transformée de Fourier (voir [3])

La fonction de la variable complexe  $s(z)$  est une série et ne converge pas forcément pour toute valeur de  $z$ . En toute rigueur il faudrait associer le rayon de convergence à chaque fonction en  $z$  :  $\{s(z) \text{ converge pour } |z| > \alpha\}$ .

Les automaticiens pour qui la transformée en z n'est bien souvent qu'une façon bien commode de regrouper des termes de récurrence utilisent souvent la pseudo-variable<sup>2</sup>  $q$ .

A partir d'une transformée en z  $s(z)$  il est possible de retrouver la séquence  $s(k)$  correspondante en appliquant la transformation inverse, dont on ne donnera pas la formule ici. A la place de faire appel à cette formule compliquée il est possible d'utiliser d'autres méthodes pour retrouver la séquence  $s(k)$  lorsque la fonction  $s(z)$  est donnée sous forme de fraction rationnelle :

<sup>1</sup> Pour une justification du terme « causal » voir page 11

<sup>2</sup> Voir page 10

- division polynomiale
- décomposition de la fraction rationnelle en éléments simples puis utilisation de transformations élémentaires telles que celles données dans le Tableau 2

La propriété fondamentale de la transformée en z est qu'elle est linéaire (la transformée en z d'une composition linéaire de signaux est la composition linéaire de la transformée en z de chaque signal). Les autres propriétés fondamentales sont données dans le Tableau 2.

Tableau 1 : propriétés de la transformée en z

1	$s(z) = Z(s(k)) = \sum_{k=0}^{\infty} s(k) \cdot z^{-k}$
2	$Z(\alpha \cdot s_1(k) + \beta \cdot s_2(k)) = \alpha \cdot Z(s_1(k)) + \beta \cdot Z(s_2(k))$
3	$Z(k \cdot s(k)) = -z \cdot \frac{ds(z)}{dz}$
4	$Z(c^k \cdot s(k)) = s\left(\frac{z}{c}\right)$
5	$Z(s(k-1)) = s(k=-1) + z^{-1} \cdot s(z)$
6	$Z(s(k+1)) = z \cdot s(z) - z \cdot s(k=0)$
7	$s(k=0) = \lim_{z \rightarrow \infty} (s(z))$
8	$s(k \rightarrow \infty) = \lim_{z \rightarrow 1} \left( \frac{z-1}{z} \cdot s(z) \right)$

Remarque 1 : la propriété 5 ne prend pas en compte le fait que le signal est causal : en pratique  $s(k=-1) = 0$ .

Remarque 2 : la propriété 5 justifie le fait que le terme  $z^{-1}$  soit appelé "opérateur de retard unitaire", cette propriété est aussi appelée "théorème du retard".

Remarque 3 : les propriétés 7 et 8 sont données sous réserve de convergence.

Transformées inverses

### Exemple 1 : transformée inverse par division polynomiale

On considère un signal  $s(k)$  dont la transformée en z est donnée par la fraction rationnelle :

$$S(z) = \frac{0.1 \cdot z^2 + 0.1 \cdot z}{z^3 - 2.6 \cdot z^2 + 2.2 \cdot z - 0.6} \quad (3.1)$$

L'objectif est d'obtenir  $S(z)$  sous forme polynomiale :

$$S(z) = x_0 + x_1 \cdot z^{-1} + x_2 \cdot z^{-2} + \dots \text{etc.} \tag{3.2}$$

Cette suite est généralement infinie, la réponse temporelle est donnée par la succession des termes  $\{x_i\}$ . En traitant l'équation (3.1) comme une simple division l'algorithme de division classique (celui appris en CM2) permet d'obtenir par récurrence tous les termes de la suite (3.2) :

$0.1 \cdot z^2$	$+ 0.1 \cdot z$				$z^3 - 2.6 \cdot z^2 + 2.2 \cdot z - 0.6$
$0.1 \cdot z^2$	$- 0.26 \cdot z$	$+ 0.22$	$- 0.06 \cdot z^{-1}$		
$0$	$0.36 \cdot z$	$- 0.22$	$0.06 \cdot z^{-1}$		
	$0.36 \cdot z$	$- 0.936$	$0.792 \cdot z^{-1}$	$- 0.216 \cdot z^{-2}$	
$0$	$0.716$	$- 0.732 \cdot z^{-1}$	$\dots$	$\dots$	

Les éléments successifs de la suite  $s(k)$  sont donc  $s(k) = \{0, 0.1, 0.36, 0.716, \dots\}$ . Pour obtenir les termes suivants il faudrait continuer la division.

**Exemple 2 : transformée inverse par décomposition en éléments simples**

La deuxième méthode présentée consiste à décomposer la fraction rationnelle (3.1) en éléments simples puis d'utiliser les tables de transformées en  $z$  (Tableau 2). Pour notre exemple cela donne :

$$S(z) = \frac{0.1 \cdot z^2 + 0.1 \cdot z}{z^3 - 2.6 \cdot z^2 + 2.2 \cdot z - 0.6} = \frac{-z}{z-1} + \frac{0.5 \cdot z}{(z-1)^2} + \frac{z}{z-0.6}$$

La réponse temporelle est donc composée des trois éléments :

$$s(k) = -(1 \ 1 \ 1 \ \dots) + 0.5 \cdot (1 \ 2 \ 3 \ \dots) + (1 \ 0.6 \ 0.6^2 \ \dots)$$

$$\text{soit : } s(k) = \begin{cases} s(0) = -1 + 0.5 + 1 \\ s(2) = -1 + 0.5 \cdot 2 + 0.6 \cdot 0.5 \\ s(3) = -1 + 0.5 \cdot 3 + 0.6^2 \\ s(4) = \dots \end{cases}$$

**3.1.2 Signaux types**

Le Tableau 2 donne les transformée en  $z$  de certains signaux fondamentaux. La Figure 2 représente l'allure de quelques signaux types.

Tableau 2 : transformée en z élémentaires

$s(k)$	$S(z)$
signal temporel	Transformée en z
$\delta(k)$ : impulsion unité	1
$u(k)$ : échelon unité	$\frac{z}{z-1}$
$k \cdot u(k)$	$\frac{z}{(z-1)^2}$
$c^k \cdot u(k)$	$\frac{z}{z-c}$
$k \cdot a^k \cdot u(k)$	$\frac{a \cdot z}{(z-a)^2}$
$\sin(\omega \cdot k) \cdot u(k)$	$\frac{z \cdot \sin(\omega)}{z^2 - 2 \cdot z \cdot \sin(\omega) + 1}$
$\cos(\omega \cdot k) \cdot u(k)$	$\frac{z \cdot (z - \cos(\omega))}{z^2 - 2 \cdot z \cdot \sin(\omega) + 1}$
$a^k \cdot \sin(\omega \cdot k) \cdot u(k)$	$\frac{a \cdot z \cdot \sin(\omega)}{z^2 - 2 \cdot a \cdot \cos(\omega) \cdot z + a^2}$
$a^k \cdot \cos(\omega \cdot k) \cdot u(k)$	$\frac{z \cdot (z - a \cdot \cos(\omega))}{z^2 - 2 \cdot a \cdot \cos(\omega) \cdot z + a^2}$



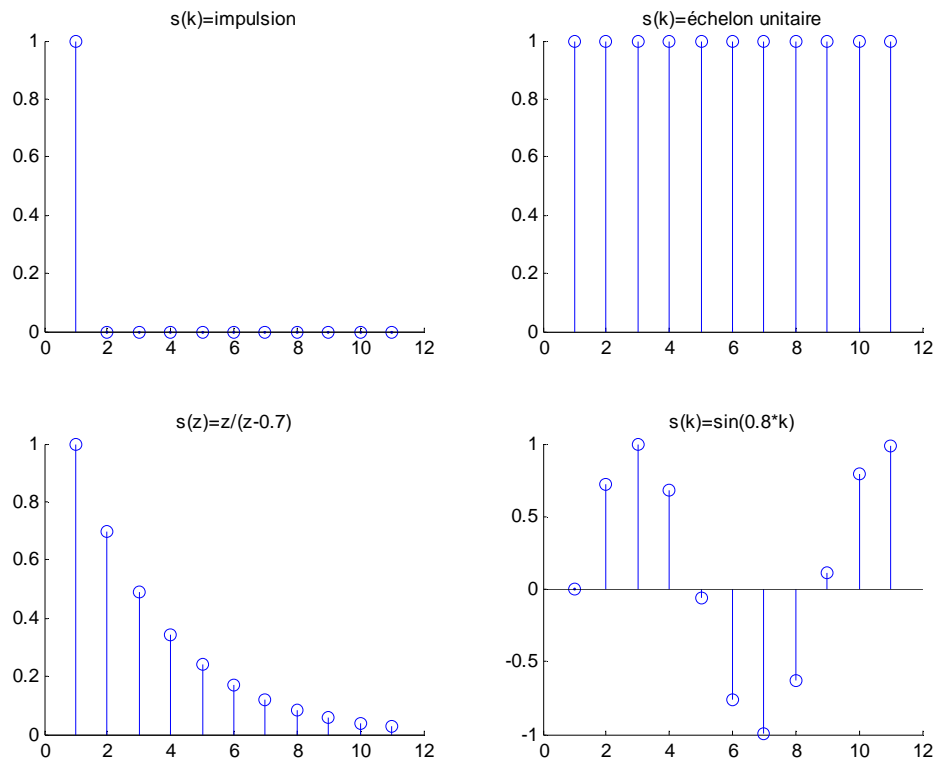


Figure 2 : réponses de quelques signaux types

## 3.2 Systèmes discrets

### 3.2.1 Fonction de transfert discrète, définition

On considère maintenant un système discret linéaire et invariant dans le temps (LTI) représenté sur la Figure 3.

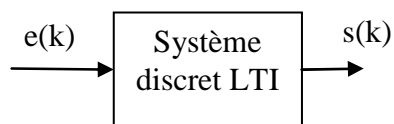


Figure 3 : système discret, notations

La relation entre la séquence d'entrée et la séquence de sortie est linéaire et définie par un ensemble de coefficients constants  $a_{1:n}$  et  $b_{1:m}$  :

$$\begin{aligned} s(k) + a_1 \cdot s(k-1) + a_2 \cdot s(k-2) + \dots + a_n \cdot s(k-n) = \\ b_0 \cdot e(k) + b_1 \cdot e(k-1) + b_2 \cdot e(k-2) + \dots + b_m \cdot e(k-m) \end{aligned} \quad (2)$$

On peut appliquer la transformée en  $z$  à chaque membre de l'équation puis en appliquant en récurrence la propriété 5 (théorème du retard) on obtient la fonction de transfert discrète du système :

$$\frac{s(z)}{e(z)} = F(z) = \frac{b_0 + b_1 \cdot z^{-1} + b_2 \cdot z^{-2} + \dots + b_m \cdot z^{-m}}{1 + a_1 \cdot z^{-1} + a_2 \cdot z^{-2} + \dots + a_n \cdot z^{-n}} \quad (3)$$

Remarque 1 : la fonction de transfert précédente est normalisée en prenant  $a_0 = 1$ .

Remarque 2 : on remarquera que la fonction de transfert est donnée en  $z^{-1}$  plutôt qu'en  $z$ . Cela est justifié par le fait que celle-ci traduit l'équation d'évolution du système physique causal (2).

Le script Matlab permettant de créer un objet Matlab correspondant à une fonction de transfert discrète est donné ci-dessous.

#### Matlab

```
% Création d'un système linéaire discret :
Te=0.1;
sysd = tf([b0 b1 b2],[1 a0 a2 a3],Te,'variable','z^-1');
```

On notera l'utilisation de l'argument 'variable', 'z^-1'.

### 3.2.2 Notation "opérateur de retard $q^{-1}$ "

La fonction de transfert est une fonction de la variable complexe  $z$ . Il faudrait donc en toute rigueur l'associer à son rayon de convergence. En pratique on ne s'embarrasse jamais de préciser cette zone de convergence. C'est pourquoi on trouve souvent la notation utilisant l'opérateur  $q^{-1}$  de décalage à la place de la variable complexe  $z^{-1}$  (4).  $F(q)$  n'est alors plus à proprement parler une fonction mais on lui appliquera les mêmes manipulations algébriques qu'à  $F(z)$ .

$$F(q) = \frac{b_0 + b_1 \cdot q^{-1} + b_2 \cdot q^{-2} + \dots + b_m \cdot q^{-m}}{1 + a_1 \cdot q^{-1} + a_2 \cdot q^{-2} + \dots + a_n \cdot q^{-n}} \quad (4)$$

Attention : selon les auteurs l'opérateur qui remplace  $z^{-1}$  est noté  $q$  (Matlab) ou  $q^{-1}$  (Landau [1]). Dans la suite du document c'est cette dernière convention qui sera appliquée.

L'utilisation de l'opérateur de retard  $q^{-1}$  à la place de la variable complexe  $z^{-1}$  permet de traiter du cas des fonctions de transfert à coefficients non constant, cas non prévu par la transformation en  $z$ .

### 3.2.3 Réponse impulsionnelle

Essayons d'appliquer au système un signal d'entrée particulier : une impulsion unitaire. La transformée en  $z$  du système est obtenue en multipliant la fonction de transfert par la transformée en  $z$  de l'impulsion unitaire : 1. On obtient :

$$s(z) = \frac{b_0 + b_1 \cdot z^{-1} + b_2 \cdot z^{-2} + \dots + b_m \cdot z^{-m}}{1 + a_1 \cdot z^{-1} + a_2 \cdot z^{-2} + \dots + a_n \cdot z^{-n}} \times 1 \quad (5)$$

Cela justifie le fait que la fonction de transfert du système peut aussi s'appeler "réponse impulsionnelle".

#### Matlab

```
% Réponse impulsionnelle d'un système :
[y,t] =impulse(sysd);
```

### 3.2.4 Causalité

Un système est dit causal si les sorties ne dépendent que du passé et non pas du futur. La conséquence est que la réponse impulsionnelle de  $F(z)$  ne peut pas comporter de terme non nuls pour  $k < 0$ . Cela justifie la confusion sémantique entre "système causal" et "signal causal", c'est-à-dire ne comportant pas de termes non nuls pour  $k < 0$ .

Attention : à la différence des systèmes continus un système physique (causal) discret peut très bien être modélisé avec un degré  $m$  supérieur au degré  $n$  : le système peut avoir plus de zéros que de pôles.

La transformée en  $z$  représentant des signaux échantillonnés elle peut tout aussi bien représenter les mesures prises à différents endroits d'une poutre. Le temps étant remplacé ici par une abscisse la notion de causalité ne s'applique pas ! Attention : beaucoup des propriétés de systèmes échantillonnés que l'on étudie sont liés à la causalité, prudence donc si l'on applique ces méthodes à d'autres domaines !

### 3.2.5 Réponses indicielle

Pour caractériser des systèmes on s'intéresse souvent à leur réponse impulsionnelle ou, plus souvent, à leur réponse indicielle. Pour calculer la réponse

à un échelon d'un système il faut donc multiplier la fonction de transfert par la transformée en z d'un échelon unitaire :

$$s(z) = \frac{b_0 + b_1 \cdot z^{-1} + b_2 \cdot z^{-2} + \dots + b_m \cdot z^{-m}}{1 + a_1 \cdot z^{-1} + a_2 \cdot z^{-2} + \dots + a_n \cdot z^{-n}} \times \frac{1}{1 - z^{-1}} \quad (6)$$

Il faudrait ensuite calculer la transformée en z inverse de s(z). En pratique, si l'on doit faire le calcul à la main on décompose en éléments simples puis on utilise des tables. Avec Matlab c'est plus simple. Voici par exemple le script pour obtenir les 20 premiers échantillons de la réponse indicielle d'un système :

**Matlab**

```
% Réponse indicielle d'un système :
[y,t] =step(sysd,20);
```

### 3.2.6 Régime stationnaire (gain statique)

Le gain statique est obtenu en appliquant une entrée indicielle et en calculant la valeur du régime permanent. En appliquant le théorème de la valeur finale il vient :

$$F_0 = \lim_{z \rightarrow 1} \left( \frac{z-1}{z} \cdot \frac{b_0 + b_1 \cdot z^{-1} + b_2 \cdot z^{-2} + \dots + b_m \cdot z^{-m}}{1 + a_1 \cdot z^{-1} + a_2 \cdot z^{-2} + \dots + a_n \cdot z^{-n}} \times \frac{1}{1 - z^{-1}} \right) \quad (7)$$

Soit :

$$F_0 = \frac{\sum_{i=0}^{i=m} b_i}{1 + \sum_{i=1}^{i=n} a_i} \quad (8)$$

Le gain statique est obtenu donc simplement en faisant le quotient de la somme des coefficients.

### 3.2.7 Pôles, zéros, stabilité

Comme pour les systèmes continus la dynamique et la stabilité des systèmes dépend des propriétés du dénominateur de la fonction de transfert.

$$F(z) = \frac{e_0 \cdot (1 + e_1 \cdot z^{-1}) \cdot (1 + e_2 \cdot z^{-1}) \cdots (1 + e_m \cdot z^{-1})}{(1 + c_1 \cdot z^{-1}) \cdot (1 + c_2 \cdot z^{-1}) \cdots (1 + c_n \cdot z^{-1})} \quad (9)$$

Les coefficients  $c_i$  sont les pôles de la fonction de transfert, les coefficients  $e_i$  sont les zéros. La fonction de transfert étant au départ composée de coefficients purement réels les coefficients  $c_i$  sont soit réels soit imaginaires par paires conjuguées.

Une fonction de transfert est donc aussi bien définie par la donnée de ses pôles et de ses zéros. Le script Matlab pour créer un système ainsi est donné ci-dessous :

#### Matlab

```
% Système discret défini par ses pôles et zéros :
Te=0.1;
sysd=zpk([c1 c2 c3],[e1 e2],e0,Te);
```

Si l'on décompose la fonction de transfert en éléments simples on obtient une somme d'éléments du type :

$$F(z) = \frac{d_1}{1 + c_1 \cdot z^{-1}} + \frac{d_2}{1 + c_2 \cdot z^{-1}} + \cdots + \frac{d_n}{1 + c_n \cdot z^{-1}} \quad (10)$$

Un système linéaire temps invariant est dit stable si sa réponse impulsionnelle tend asymptotiquement vers 0. Compte tenu de (10) cela nécessite que chaque pôle  $c_i$  soit de module inférieur à 1.

**Propriété** : un système discret LTI est stable si et seulement si tous ses pôles ont leur modules inférieur ou égal à 1.

#### Critère algébrique attestant de la stabilité : critère de Routh

On rappelle que le critère de Routh permet de prédire si les racines d'un polynôme sont toutes à partie réelle négative, donc stables. Énonçons le théorème de Routh dans le cas d'un polynôme  $P'(w)$  :

$$P'(w) = a_0 + a_1 \cdot w + a_2 \cdot w^2 + \cdots + a_n \cdot w^n \quad (11)$$

Dans un premier temps on construit le tableau :

$$\begin{array}{cccccc|c}
 \beta_{n,0} & \beta_{n,1} & \beta_{n,2} & \beta_{n,q-1} & \beta_{n,q} & 0 & \\
 \beta_{n-1,0} & \beta_{n-1,1} & \beta_{n-1,2} & \beta_{n-1,q-1} & \beta_{n-1,q-1} & 0 & \\
 \beta_{n-2,0} & \beta_{n-2,1} & \beta_{n-2,2} & \beta_{n-2,q-1} & \beta_{n-2,q-1} & & \\
 & & & & & & \\
 \beta_{2,0} & \beta_{2,1} & 0 & & & & \\
 \beta_{1,0} & 0 & & & & & \\
 \beta_{0,0} & 0 & & & & & 
 \end{array} \quad (12)$$

Puis on vérifie que tous les premiers éléments de la première colonne ont le même signe : cela garantit que tous les pôles de  $P'(w)$  ont leur partie réelle négative.

Les éléments du tableau sont construits par récurrence ligne par ligne en appliquant les formules suivantes :

Pour  $i = n$  et  $i = n - 1$  :

$$\begin{cases} \beta_{n,j} = a_{n-2j} \\ \beta_{n-1,j} = a_{n-1-2j} \end{cases} \quad (13)$$

Et pour  $i < n-1$  :

$$\beta_{i,j} = \frac{\beta_{i+1,0} \cdot \beta_{i+2,j+1} - \beta_{i+1,j+1} \cdot \beta_{i+2,0}}{\beta_{i+1,0}} \quad (14)$$

Dans le cas qui nous intéresse, un système échantillonné, on cherche plutôt à vérifier que les pôles du polynôme  $P(z)$  sont de module inférieur à un. L'idée est d'effectuer un changement de variable  $z \rightarrow w$  pour pouvoir appliquer le critère de Routh.

$$P'(w) = (1-w)^n \cdot P\left(\frac{1+w}{1-w}\right) \quad (15)$$

Si l'on effectue la transformation (15) alors le fait que  $P'(w)$  ait toutes ses racines à partie réelle négative garantit que  $P(z)$  a toutes ses racines dans le cercle unité. Pour vérifier la stabilité de  $P(z)$  il suffit donc d'effectuer le changement de variable (15) puis d'appliquer le critère de Routh conventionnel (12 13 14).

### Critère algébrique attestant de la stabilité : critère de Jury

Le critère de Jury s'apparente au critère de Routh. Nous le donnons pour un polynôme de degré 2, 3 et 4 du type :

$$P(z) = a_0 + a_1 \cdot z + a_2 \cdot z^2 + \dots + a_n \cdot z^n \quad (16)$$

Les conditions de stabilité sont données ci-dessous (ces conditions sont données dans le cas  $a_n > 0$ , il suffit de multiplier tous les termes par  $-1$  si cette condition n'est pas remplie par  $P(z)$ ) :

$$\begin{aligned}
 n = 2 & \begin{cases} a_0 + a_1 + a_2 > 0 \\ a_0 - a_1 + a_2 > 0 \\ -a_0 + a_2 > 0 \end{cases} \\
 n = 3 & \begin{cases} a_0 + a_1 + a_2 + a_3 > 0 \\ -a_0 + a_1 - a_2 + a_3 > 0 \\ -|a_0| + a_3 > 0 \\ a_0 a_2 - a_1 a_3 - a_0^2 + a_3^2 > 0 \end{cases} \\
 n = 4 & \begin{cases} a_0 + a_1 + a_2 + a_3 + a_4 > 0 \\ a_0 - a_1 + a_2 - a_3 + a_4 > 0 \\ a_4^2 - a_0^2 - |a_0 a_3 - a_1 a_4| > 0 \\ (a_0 - a_4)^2 (a_0 - a_2 + a_4) + (a_1 - a_3)(a_0 a_3 - a_1 a_4) > 0 \end{cases}
 \end{aligned} \quad (17)$$

### 3.2.8 Systèmes du premier ordre

On donne dans la Figure 4 les réponses indicielles de systèmes du premier ordre ainsi que la localisation des pôles dans le plan complexe. Le cercle de rayon unité représente la zone dans laquelle les pôles doivent se situer pour assurer la stabilité. On remarque sur les deux premières figures l'influence de la place du zéro par rapport au point  $(1,0)$  pour la rapidité.

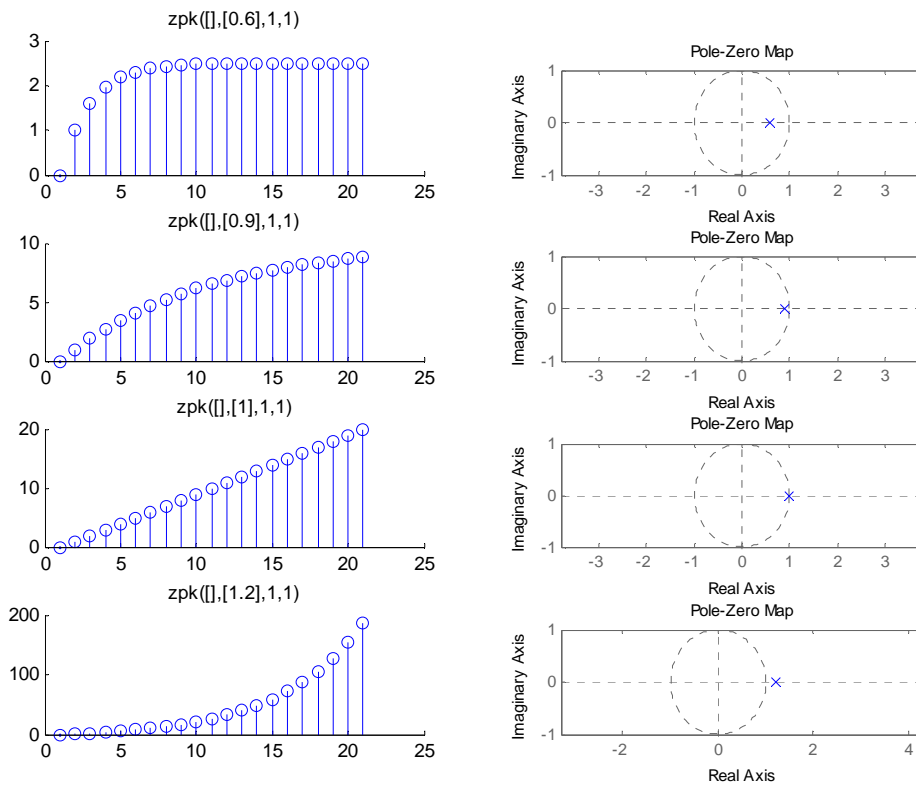


Figure 4 réponses indicielles de systèmes du premier ordre

### 3.2.9 Systèmes du second ordre

On donne dans la Figure 6 les réponses indicielles de systèmes du second ordre ainsi que la localisation des pôles dans le plan complexe. La rapidité et l'amortissement dépend de la position des pôles dans le cercle unité. La Figure 5 représente les courbes iso-amortissement et iso-pulsation des systèmes du second ordre.



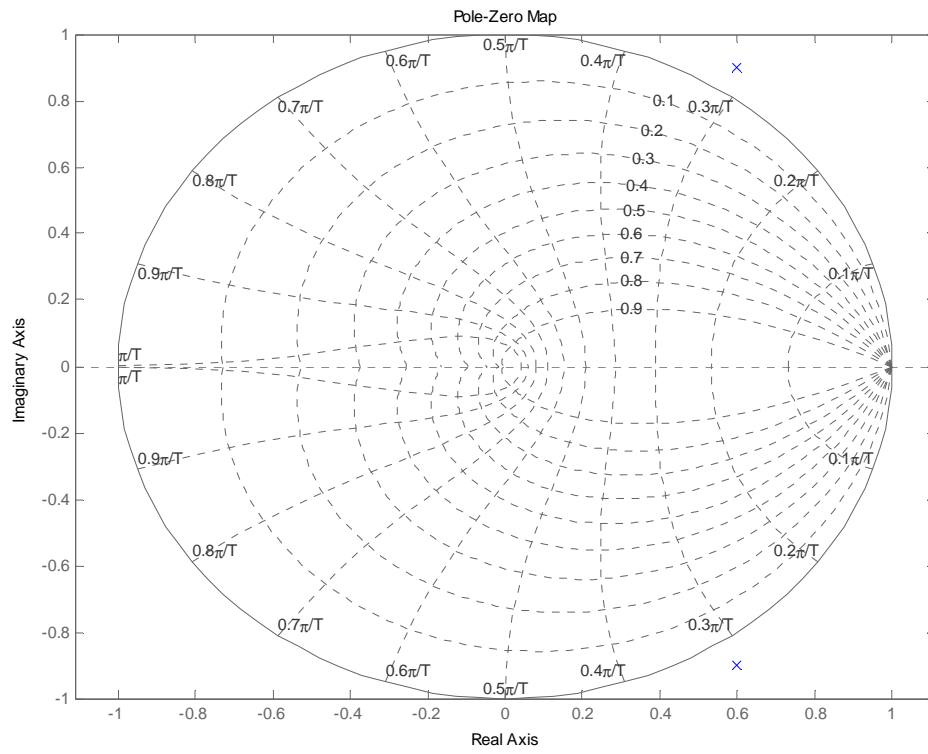


Figure 5 : courbes iso-amortissement et iso-pulsation des systèmes discrets du second ordre

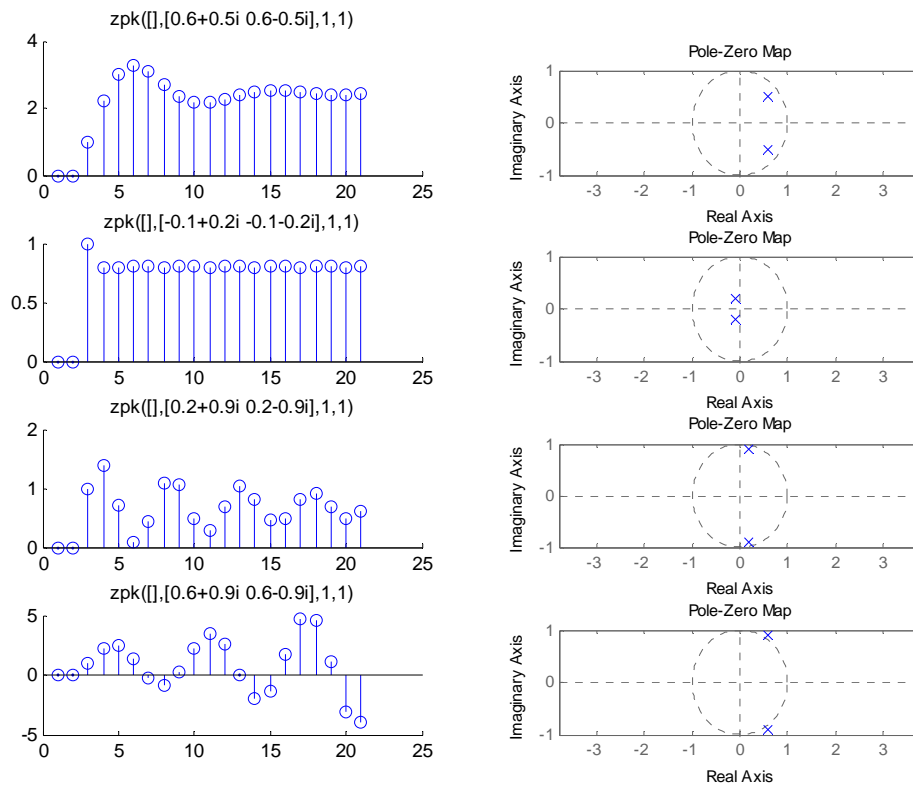


Figure 6 : réponses de systèmes du second ordre

### 3.2.10 Calcul itératif de la réponse d'un système

Si l'on dispose de la fonction de transfert d'un système et que l'on connaît les valeurs successives  $e(k)$  du signal d'entrée il est facile de calculer itérativement la sortie  $y(k)$  du système par l'équation de récurrence (2). Il faut initialiser par les  $n$  valeurs initiales de  $s(k)$ , prises généralement égales à 0.

### 3.2.11 Représentation par équation d'état

Un système discret peut aussi être représenté par un ensemble d'équations d'état comme dans le domaine continu (52). Cette représentation est naturellement adaptée aux systèmes multi-entrées multi-sorties. On définit alors un état interne  $x(k)$  ainsi qu'une fonction d'évolution  $f$  et une équation de mesure  $h$ .

$$\begin{cases} x(k+1) = f(x(k), u(k)) \\ s(k) = h(x(k), u(k)) \end{cases} \quad (18)$$

### 3.3 Réponse fréquentielle des systèmes échantillonnés

#### 3.3.1 Diagramme de Bode d'un système échantillonné

Lorsqu'on étudie la réponse fréquentielle des systèmes continus on considère des entrées particulières sinusoïdales :

$$\begin{cases} e(t) = \cos(2 \cdot \pi \cdot f \cdot t) + j \cdot \sin(2 \cdot \pi \cdot f \cdot t) = e^{j \cdot 2 \cdot \pi \cdot f \cdot t} \\ f \in [0 \quad \infty] \end{cases} \quad (19)$$

De même, pour des systèmes échantillonnés on considèrera des entrées sinusoïdales échantillonnées :

$$\begin{cases} e(k) = \cos(2 \cdot \pi \cdot f \cdot k) + j \cdot \sin(2 \cdot \pi \cdot f \cdot k) = e^{j \cdot 2 \cdot \pi \cdot f \cdot k} \\ f \in [0 \quad 1] \end{cases} \quad (20)$$

La sortie est aussi un signal échantillonné sinusoïdal de gain  $A(f)$  et de phase  $\Phi(f)$  variables et dépendant de  $f$ ; la fréquence réduite.

$$\begin{cases} s(k) = A(f) \cdot e^{j \cdot (2 \cdot \pi \cdot f \cdot k + \Phi(f))} \\ f \in [0 \quad 1] \end{cases} \quad (21)$$

Le nombre complexe, noté abusivement  $F(f)$ , d'amplitude  $A(f)$  et de phase  $\Phi(f)$  est obtenu à partir de  $F(z)$  en remplaçant  $z$  par  $e^{j \cdot 2 \cdot \pi \cdot f}$  :

$$\begin{cases} F(f) = F(z = e^{j \cdot 2 \cdot \pi \cdot f}) \\ f \in [0 \quad 1] \end{cases} \quad (22)$$

Le diagramme représentant le gain et la phase en fonction de la fréquence réduite s'appelle aussi un diagramme de Bode. L'échelle des fréquences est généralement linéaire de 0 à 0.5 (puisque le diagramme est nécessairement symétrique). La Figure 7 représente par exemple le diagramme de Bode d'un système du second ordre.

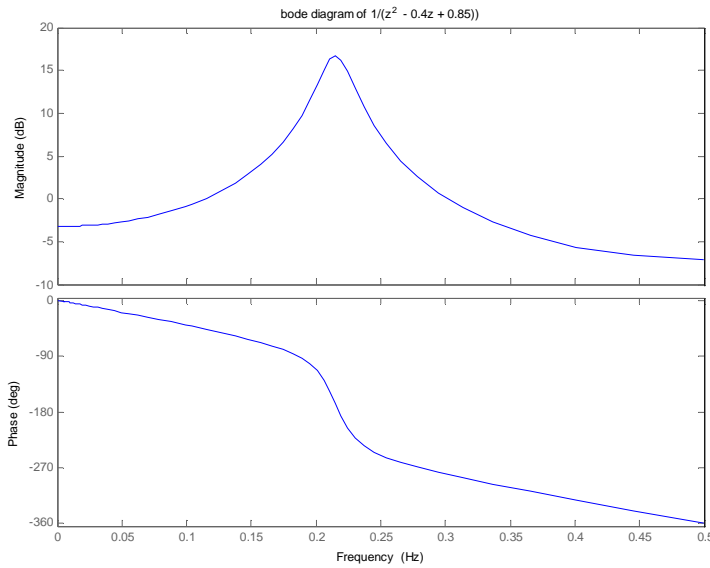


Figure 7 : exemple du diagramme de Bode d'un système discret du second ordre

### 3.3.2 Où l'on voit apparaître le théorème de Shannon

Le changement de variable  $z = e^{j2\pi f}$  n'est pas bijectif puisque  $e^{j2\pi f} = e^{j2\pi(f+k)}$  avec  $k$  entier. Si l'on fait le lien entre domaine continu et domaine fréquentiel on peut remarquer que le changement de variable  $z = e^{j2\pi f}$  fait correspondre une zone du plan complexe continu à une zone du plan complexe discret. La transformation est bijective si l'on se contente de la zone du plan complexe continu contenu entre les droites d'ordonnée 0 et 1.

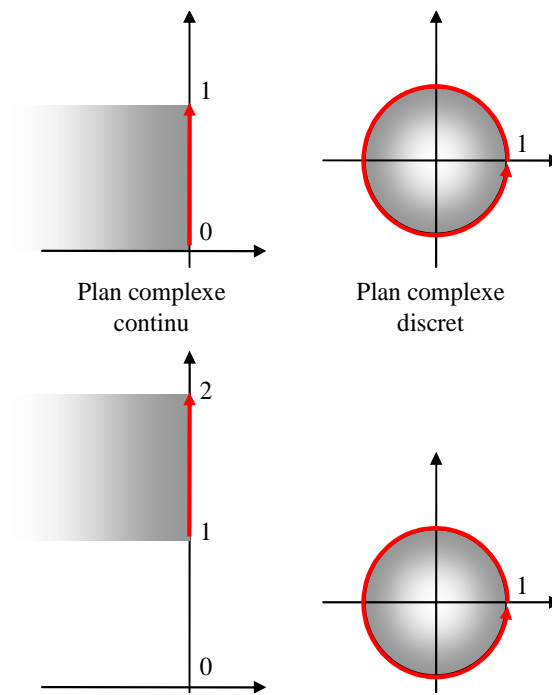


Figure 8 : correspondance entre plan complexe continu et plan complexe discret.



## 4. SYSTEMES CONTINUS ECHANTILLONNES

### 4.1 Structure des systèmes commandés par ordinateur

Les systèmes commandés par ordinateurs sont presque toujours des systèmes physiques continus par nature. Le ordinateur traite par contre les entrées et les sorties de manière discrète avec une période d'échantillonnage  $T_e$ . Les éléments importants, du point de vue de l'automaticien sont donc :

- le convertisseur analogique / numérique CAN ou du moins le système qui fournit séquentiellement les mesures au ordinateur à la période  $T_e$ .
- le ordinateur lui-même. Il doit effectuer l'acquisition des mesures, le traitement des informations, l'envoi des consignes dans un temps inférieur à  $T_e$ .
- le convertisseur numérique / analogique CNA.
- le système physique.

On suppose, pour faire simple et pour pouvoir utiliser la panoplie des outils mathématiques, que les systèmes sont échantillonnés à la même période  $T_e$  et au même moment.

La Figure 9 montre le système physique commandé par le ordinateur, dans sa version simpliste.

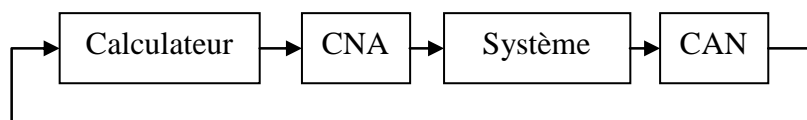


Figure 9 : système commandé par ordinateur

Au XXIème siècle les choses sont quand même parfois un petit peu plus compliquées :

- le ordinateur fournit bien souvent un signal de commande numérique (transporté par bus informatique USB, RS232, CAN, Ethernet, etc...) directement interprété par l'actionneur. On parle "d'actionneur intelligent".
- de même le "capteur intelligent" fournit aussi bien souvent la mesure sous forme d'un signal numérique.
- Les signaux sont transportés par des bus informatiques qui n'ont pas que ça à faire : il peut y avoir du retard.
- de même les capteurs et actionneurs sont rarement tous seuls, ce qui se traduit par des périodes d'échantillonnage différentes, et parfois non synchrones...

- enfin le ordinateur peut aussi avoir d'autres choses à faire...

Ceci étant dit nous allons pour le moment nous restreindre au cas représenté sur la Figure 9 avec des instants d'échantillonnage synchrones et régulièrement espacés d'un temps  $T_e$ .

## 4.2 Echantillonnage des systèmes continus

On dispose généralement de la fonction de transfert continue (notée en utilisant la variable complexe  $s$  ou  $p$ )  $F(s)$  du système continu. Le problème est d'obtenir la fonction de transfert discrète du système échantillonné de manière à pouvoir synthétiser un correcteur. Le système continu est associé au convertisseur numérique – analogique qu'il faut prendre en compte.

Les convertisseurs numérique analogiques peuvent généralement se représenter par des "échantillonneurs – bloqueurs d'ordre 0" ou BOZ. Cela signifie que la commande est maintenue à une valeur constante pendant toute la durée du temps d'échantillonnage.

Si l'on applique une commande discrète impulsionnelle à un BOZ il va maintenir la commande 1 pendant la durée  $T_e$  puis revenir à zéro (Figure 10).

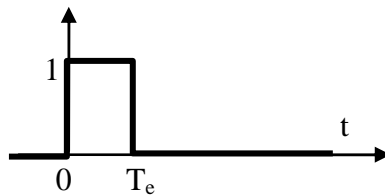


Figure 10 : réponse impulsionnelle d'un bloqueur d'ordre 0

La réponse impulsionnelle  $BOZ(t)$  est donc composée d'un échelon unitaire suivi d'un échelon unitaire de signe opposé et décalé d'un temps  $T_e$ . La transformée de Laplace de la réponse impulsionnelle du bloqueur est donnée par la formule (23) :

$$BOZ(s) = \frac{1}{s} - \frac{e^{-jT_e}}{s} \quad (23)$$

Du point de vue du ordinateur il faut donc prendre en compte le système continu de fonction de transfert  $F_{BOZ}(s)$  que l'on devra échantillonner :

$$F_{BOZ}(s) = \frac{1 - e^{-jT_e}}{s} \cdot F(s) \quad (24)$$



$$F_{\text{BOZ}}(z) = Z^{-1} \left( \frac{1 - e^{-jT_e}}{s} \cdot F(s) \right) \quad (25)$$

En pratique l'obtention de la transformée en z d'un système échantillonné par bloqueur d'ordre 0 se fait à l'aide de tables. On donne quelques exemples fondamentaux dans le Tableau 3 ci-dessous.

Tableau 3 : transformée en z de systèmes continus échantillonnés

F(s)	F <sub>BOZ</sub> (z)
1	1
$\frac{1}{s}$	$\frac{T_e \cdot z^{-1}}{1 - z^{-1}}$
$\frac{1}{1 + T \cdot s}$	$\begin{cases} \frac{b_1 \cdot z^{-1}}{1 + a_1 \cdot z^{-1}} \\ b_1 = 1 - e^{-T_e/T} \quad a_1 = -e^{-T_e/T} \end{cases}$
$\frac{e^{-Ls}}{1 + T \cdot s}$	$\begin{cases} \frac{b_1 \cdot z^{-1} + b_2 \cdot z^{-2}}{1 + a_1 \cdot z^{-1}} \quad a_1 = -e^{-T_e/T} \\ b_1 = 1 - e^{-(L-T_e)/T} \quad b_2 = e^{-T_e/T} (e^{L/T} - 1) \end{cases}$
$L < T_e$	

Si l'on dispose de Matlab c'est bien sûr encore plus simple. On utilise la fonction c2d avec l'argument 'zoh' pour "zero holder hold" (bloqueur d'ordre zéro).

#### Matlab

```
% Définition d'un système continu :
sysc=tf(1,[1 1]);
% Obtention de la transformée en z du système échantillonné à Te
Te=0.1;
sysd=c2d(sysc,Te,'zoh');
% Notation en z^-1
sysd.variable='z'
```

### 4.3 Effets de l'échantillonnage

Le convertisseur numérique a deux effets (néfastes) importants : l'effet de retard et l'effet de quantification.

L'effet de retard se voit dans (24) : on peut dire en première approximation que l'échantillonnage se traduit par un retard pur de  $T_e/2$ . Cela a donc, si ce temps

d'échantillonnage est non négligeable par rapport aux constantes de temps du système, un effet déstabilisant.

Pour illustrer l'effet de quantification supposons par exemple que l'on utilise un convertisseur analogique/numérique 8 bits 0-10V. Le plus petit incrément que le convertisseur peut distinguer est de  $10V/2^8 = 40mV$ . Si le signal utile varie physiquement entre 5V et 5,4V on n'aura que dix incréments utilisables... Ce phénomène n'entre pas dans la panoplie des phénomènes modélisables par la transformée en z. En pratique il est donc souvent nécessaire d'étudier le système et de synthétiser la commande numérique en négligeant ce phénomène puis de vérifier a posteriori qu'il n'a pas un effet trop néfaste.

#### 4.4 Où l'on revoit le théorème de Shannon

Le théorème de Shannon est bien connu en ce qui concerne l'échantillonnage de signaux discrets : on sait que le signal ne doit pas comporter de composante de fréquence supérieure à  $f_e/2$  si l'on ne veut pas observer de repliement. On doit aussi prendre des précautions si l'on veut travailler sur la fonction de transfert discrète correspondant à un système continu échantillonné.

On va traiter un exemple concret pour mettre en évidence le problème de repliement dans le cas de la commande des systèmes continus. On s'intéresse à deux systèmes continus du second ordre ayant des pôles de même partie réelle mais de parties imaginaires séparées de  $2\pi T_e$ . Le script Matlab ci-dessous montre la création des deux systèmes :

##### Matlab

```
% Définition de deux systèmes continus :  
sys1=zpk([],[-1-2i -1+2i],2^2+1);  
sys2=zpk([],[-1-(2+2*pi)*i -1+(2+2*pi)*i],1+(2+2*pi)^2);
```

On voit sur la Figure 11 que la réponse indicielle des deux systèmes est bien différente.

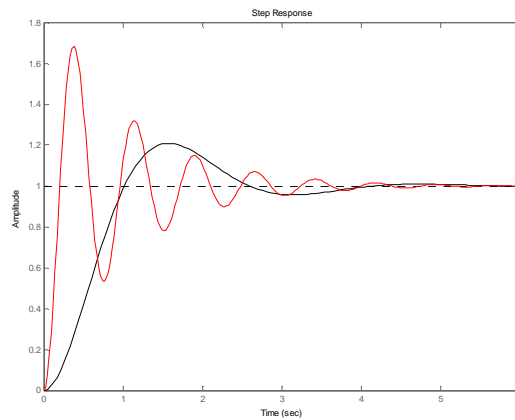


Figure 11 : réponse indicielle de deux systèmes du second ordre

Si l'on calcule la fonction de transfert échantillonnée des deux systèmes à la période de 1 seconde on obtient deux réponses indicielles très proches (représentés par des  $\circ$  sur la Figure 12). Conclusion : la période d'échantillonnage était mal choisie

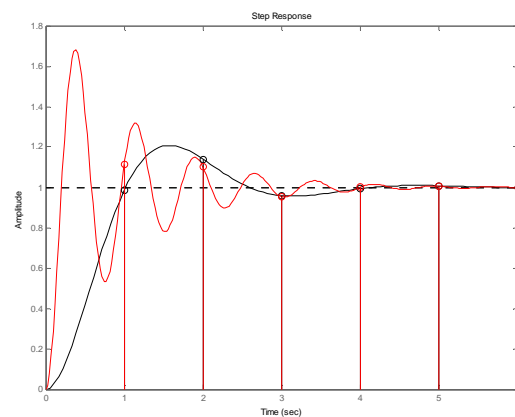


Figure 12 : réponse indicielle de deux systèmes du second ordre échantillonnés à 1s

En pratique la commande échantillonnée de systèmes continus doit être associée à certaines précautions :

- le système ne doit pas comporter de dynamiques propres situées dans les fréquences supérieures à  $f_c/2$  (exemple illustré dans ce chapitre).
- les signaux échantillonnés doivent être "nettoyés" des fréquences supérieures à  $f_c/2$  avec un filtre anti-repliement. (A moins que le signal soit suffisamment propre et ne comporte pas de composantes significatives au-delà de ces fréquences).

### Choix de la période d'échantillonnage

Le choix de la période d'échantillonnage est très important et dépend de la dynamique des signaux. Attention, les systèmes commandés en boucle fermée ont souvent une dynamique plus rapide que le système en boucle ouverte : il faut donc prendre en compte la dynamique en boucle fermée pour le choix de la fréquence d'échantillonnage.

En pratique on préconise de prendre une fréquence 5 à 25 fois supérieure à la bande passante souhaitée en boucle fermée.

On peut être tenté de prendre une fréquence d'échantillonnage très élevée si la technologie le permet (ordinateur et convertisseurs suffisamment rapide). Ceci peut amener rapidement à des problèmes numériques. Imaginons par exemple le cas d'un système ayant une dynamique très lente (quelques secondes) et un système de mesure codant les mesures sur 10 bits (cas classique). La Figure 13 illustre cet exemple. Le signal varie peu entre deux instants d'échantillonnage et les variations du signal sont faibles par rapport à la valeur de l'incrément numérique (figure du haut). Le calcul de la simple dérivée numérique se traduit donc par une succession de  $-1$   $0$  et  $+1$ .

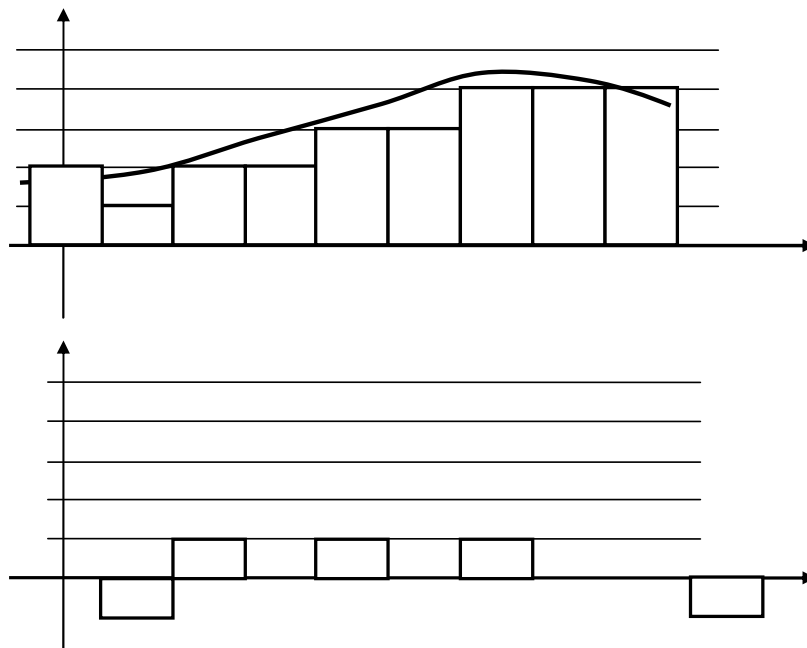


Figure 13 : période d'échantillonnage trop faible. En haut : signal échantillonné et discrétisé ; en bas : calcul de la dérivée numérique.

### Filtre anti-repliement

En toute rigueur chaque fois qu'un signal est échantillonné il faudrait faire précéder le convertisseur numérique d'un filtre analogique anti-repliement éliminant physiquement toutes les composantes du signal de fréquence supérieure à  $f_e/2$ .

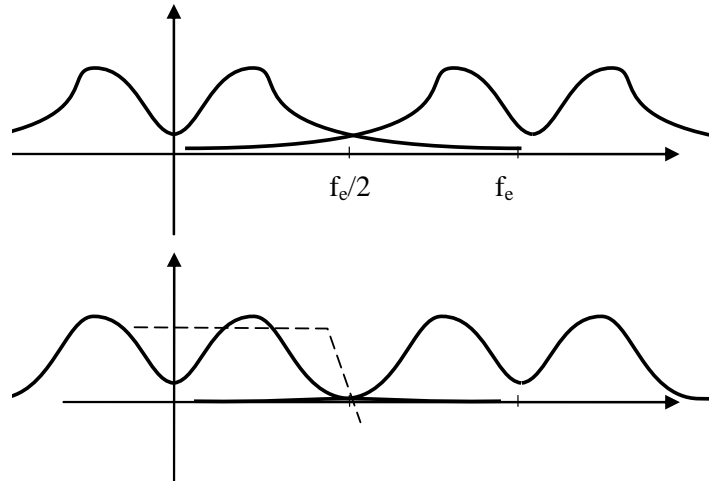


Figure 14 : Effet du filtre anti-repliement

Bien souvent la période d'échantillonnage de l'asservissement est relativement lente (quelques secondes) et il est alors impossible de réaliser un filtre de constante de temps aussi faible. On procède alors à un sur-échantillonnage : le signal est filtré, échantillonné à haute fréquence  $T'_e$  puis filtré numériquement et sous-échantillonné à la période  $T_e$  beaucoup plus grande que  $T'_e$  comme illustré sur la

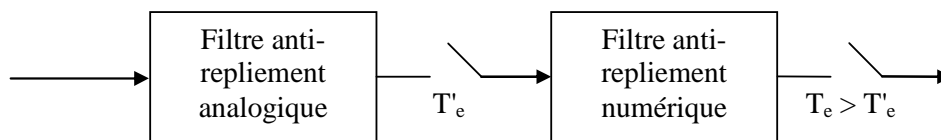


Figure 15 : Sur échantillonnage

Un simple filtre numérique de type moyenne mobile est souvent suffisant.

### 4.5 Discrétisation de fonction de transfert continues

Une manière classique pour réaliser un contrôle par ordinateur est de commencer par synthétiser un correcteur continu avec les outils de l'automatique classique puis de trouver un correcteur discret qui "approche" convenablement le correcteur continu.

#### 4.5.1 Approximation numérique d'équation différentielle

Supposons que le correcteur continu que l'on a synthétisé soit du type PD :

$$\text{PID}(s) = \frac{e(s)}{\varepsilon(s)} = k_p + k_D \cdot s \quad (26)$$

Il s'agit de calculer les commandes  $e(k)$  successives issues du correcteur en fonction des erreurs  $\varepsilon(k)$ . Une approche classique et intuitive est d'approximer la dérivée par une différence :

$$e(k) = k_p \cdot \varepsilon(k) + k_D \cdot \frac{\varepsilon(k) - \varepsilon(k-1)}{T_e} \quad (27)$$

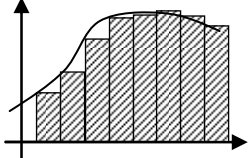
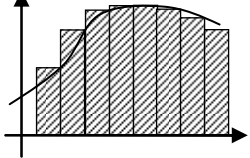
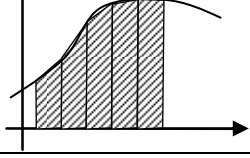
Si on calcule la transformée en  $z$  de  $e(k)$  on obtient :

$$e(z) = k_p \cdot \varepsilon(z) + k_D \cdot \frac{\varepsilon(z) - z^{-1}\varepsilon(z)}{T_e} = \left( k_p + k_D \cdot \frac{1 - z^{-1}}{T_e} \right) \cdot \varepsilon(z) \quad (28)$$

On retrouve la fonction de transfert PD(s) dans laquelle on a effectué le changement de variable  $s = \frac{1-z^{-1}}{T_e}$

De manière plus générale, l'approximation de fonction de transfert revient à effectuer l'approximation numérique d'intégration. Selon la méthode d'intégration choisie on effectue des changements de variable différents.

Tableau 4 : méthodes d'intégration

Méthode	Changement de variable	
Euler progressif (Euler's forward method)	$s = \frac{1 - z^{-1}}{z^{-1} \cdot T_e}$	
Euler rétrograde (Euler's backward method)	$s = \frac{1 - z^{-1}}{T_e}$	
Méthode des trapèzes, ou bilinéaire, ou de Tustin	$s = \frac{2}{T_e} \cdot \frac{1 - z^{-1}}{1 + z^{-1}}$	

En pratique le changement de variable  $s = \frac{1 - z^{-1}}{z^{-1} \cdot T_e}$  n'est pas sans risque. On donne sur la Figure 16 la correspondance entre la zone stable du plan complexe en  $s$  et la zone correspondante après la transformation (Euler progressif, Euler rétrograde, bilinéaire). On voit notamment qu'il est possible, partant d'un système continu stable, d'obtenir des pôles dans le domaine discret en dehors du cercle de stabilité si la transformation appliquée est celle d'Euler progressif.

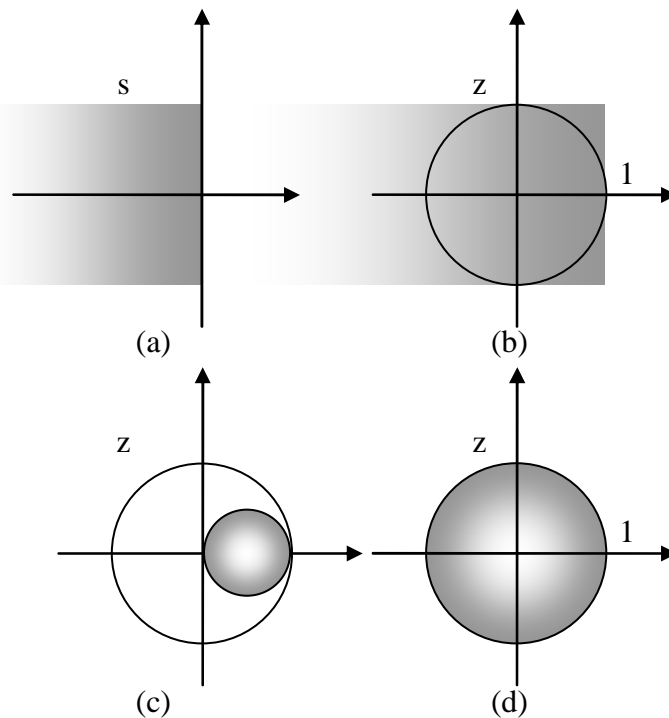


Figure 16 : correspondance des plans complexes. (a) : zone de stabilité dans le plan complexe continu en  $z$  ; (b) zone correspondante avec l'approximation d'Euler progressif ; (c) zone correspondante avec l'approximation d'Euler rétrograde ; (d) zone correspondante avec l'approximation bilinéaire.

La fonction `c2d` de Matlab permet d'obtenir l'équivalent discret d'une fonction de transfert continue avec la méthode d'intégration comme argument :

```

Matlab
sysc=tf([1 0],[1 2 5]);
Te=0.1;
% Méthode d'euler :
sysd1=c2d(sysc,Te,'zoh');
% Méthode des trapèzes :
sysd2=c2d(sysc,Te,'foh');
% Méthode bilinéaire :
sysd3=c2d(sysc,Te,'tustin');
    
```

#### 4.5.2 Superposition des réponses indicielles

Une autre méthode pour approximer une fonction de transfert discrète est de s'assurer que la réponse indicielle de sa version discrétisée correspond exactement lors des temps d'échantillonnage.



### 4.5.3 Correspondances pôles - zéros

La dernière méthode est de partir d'une décomposition de la fonction de transfert continue en pôles et zéros et d'effectuer la conversion pôles à pôles et zéros par zéros. La correspondance, selon que le pôle est réel ou complexe est donnée ci-dessous :

$$\begin{aligned} (s + a) &\rightarrow (z - e^{-a \cdot T_c}) \\ (s + a)^2 + b^2 &\rightarrow z^2 - 2 \cdot e^{-a \cdot T_c} \cdot \cos(b \cdot T_c) \cdot z + e^{-2 \cdot a \cdot T_c} \end{aligned} \quad (29)$$

### 4.5.4 Bilan

Aucune méthode n'est parfaite et on n'aura jamais en même temps :

- même réponse indicielle
- même réponse impulsionnelle
- approximation (Euler, Tustin, etc...) de l'équation différentielle
- correspondance des pôles et des zéros
- correspondance des réponses fréquentielles

Ceci étant dit le choix de la période d'échantillonnage est primordial. Si la fréquence d'échantillonnage est beaucoup plus faible que la dynamique de la fonction de transfert à approximer la différence entre chaque méthode sera minimale.



## 5. IDENTIFICATION DES SYSTEMES ECHANTILLONES

Dans le présent chapitre on donne les bases des méthodes d'identification. Pour une approche complète, voir le cours de troisième année "Identification avion" [6] et les ouvrages [1] et [7].

Même si l'on dispose d'un modèle théorique du processus que l'on souhaite commander il est presque toujours nécessaire de l'affiner en procédant à des essais sur le système réel. Bien souvent le seul modèle dont l'on dispose est ce modèle issu de l'expérimentation. Différentes méthodes s'appliquent selon l'objectif recherché et le niveau de précision exigé. On distinguera :

- les modèles paramétriques
- les modèles simples
- les modèles non paramétriques

### 5.1 Identification d'un modèle ARX par la méthode des moindres carrés récursifs

A partir de cet exemple, le plus simple qui soit, on pose les bases de toutes les méthodes d'identification.

On considère un système d'entrée  $u(t)$  et de sortie  $y(t)$  représenté par l'équation de récurrence suivante :

$$y(t) + a_1 \cdot y(t-1) + a_2 \cdot y(t-2) + \dots + a_n \cdot y(t-n) = b_0 \cdot u(t) + b_1 \cdot u(t-1) + b_2 \cdot u(t-2) + \dots + b_m \cdot u(t-m) \quad (30)$$

L'objectif est, à partir d'essais et de mesures, d'identifier les meilleures valeurs possibles pour les coefficients  $a_i$  et  $b_i$ .

L'équation (30) peut être réécrite pour mettre en évidence la *prédiction* de la sortie  $y(t)$  à partir des mesures passées :

$$y(t) = -a_1 \cdot y(t-1) - a_2 \cdot y(t-2) - \dots - a_n \cdot y(t-n) + b_0 \cdot u(t) + b_1 \cdot u(t-1) + b_2 \cdot u(t-2) + \dots + b_m \cdot u(t-m) \quad (31)$$

Soit :

$$y(t) = \varphi^T(t) \cdot \theta(t)$$

avec :

$$\begin{cases} \varphi^T(t) = [-y(t-1) \ \dots \ -y(t-n) \ , \ u(t-1) \ \dots \ u(t-m)] \\ \theta(t) = [a_1 \ \dots \ a_n \ , \ b_1 \ \dots \ b_m] \end{cases} \quad (32)$$

Le but est de trouver le meilleur ensemble de paramètres  $\theta(t)$  : celui qui, à partir d'un ensemble de mesures  $\varphi(t)$  permet de prédire avec la meilleure précision la sortie effectivement mesurée.

Supposons qu'à la suite d'une expérience on dispose d'un ensemble de mesures  $\{y(t), u(t)\}_{t=1:N}$  et que l'ensemble des paramètres  $\theta$  n'est pas bien connu. On construit la séquence des *sorties prédites* :

$$\hat{y}(t) = \varphi^T(t) \cdot \theta(t) \quad (33)$$

Le paramètre  $\theta$  est celui qui minimise une norme quadratique :

$$\theta = \arg \max \left( \frac{1}{N} \sum_{t=1}^N (y(t) - \hat{y}(t))^2 \right) = \arg \max \left( \frac{1}{N} \sum_{t=1}^N (y(t) - \varphi^T(t) \cdot \theta(t))^2 \right) \quad (34)$$

Compte tenu du fait que l'équation est linéaire au départ et que le terme à minimiser est du type quadratique la solution est simplement obtenue par une pseudo-inverse :

$$\theta = \left[ \sum_{t=1}^N (\varphi(t) \cdot \varphi^T(t)) \right]^{-1} \sum_{t=1}^N (\varphi(t) \cdot y(t)) \quad (35)$$

On obtient donc la solution classique des moindres carrés. La formule (35) est le plus souvent utilisée sous sa forme récursive :

$$\theta(t) = \theta(t-1) + F(t) \cdot \varphi(t) \cdot (y(t) - \theta^T(t-1) \cdot \varphi(t))$$

avec

$$F(t)^{-1} = F(t-1)^{-1} + \varphi(t) \cdot \varphi(t)^T \quad (36)$$

ou

$$F(t) = F(t-1) - \frac{F(t-1) \cdot \varphi(t) \cdot \varphi(t)^T \cdot F(t-1)}{1 + \varphi(t)^T \cdot F(t-1) \cdot \varphi(t)}$$

(On laisse au lecteur le plaisir d'effectuer lui-même les démonstrations, qu'il peut du reste trouver dans [1] et [7]).

## 5.2 L'identification paramétrique en général et en pratique

L'exemple précédent a permis de mettre en évidence tous les aspects de la méthodologie classique de l'identification :

- il a fallu choisir un ensemble de mesures  $\{y(t), u(t)\}_{t=1:N}$ . Le choix du signal d'excitation  $u(t)$  est en pratique primordial,
- il a fallu choisir une structure pour le modèle à identifier (ici les ordres  $n$  et  $m$ ),
- il a fallu choisir un critère (ici un critère quadratique sur l'erreur entre la sortie et la sortie prédite),
- il a fallu choisir un algorithme et le paramétrer (ici l'algorithme des moindres carrés récursifs qu'il a fallu initialiser)

Enfin un point non traité dans le chapitre précédent concerne la validation :

- il aurait fallu déterminer une méthode pour comparer différents modèles et choisir le meilleur

### 5.2.1 Choix du signal d'excitation, traitement des mesures

De manière générale le système que l'on doit identifier doit être excité dans tout le domaine fréquentiel qui nous intéresse. Le signal classiquement utilisé est une séquence pseudo binaire aléatoire (Pseudo Random Binary Signal) tel celui représenté sur la Figure 17.

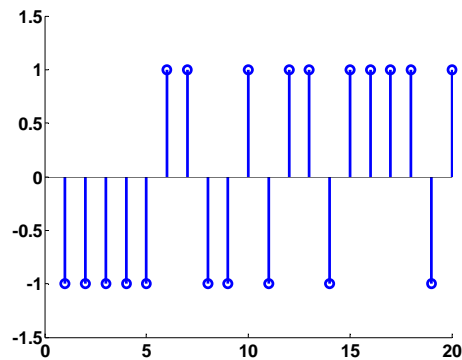


Figure 17 : exemple d'une séquence PRBS

Le signal d'excitation doit se superposer au signal de commande correspondant au point de fonctionnement étudié. Son amplitude doit être faible par rapport à l'amplitude des commandes du système (quelques %).

Une fois le système excité et les mesures effectuées il faut supprimer les valeurs moyennes des mesures.

### 5.2.2 Choix d'une structure de modèle

Le modèle étudié dans le chapitre précédent est de type ARX. On peut le noter sous forme polynomiale :

$$A \cdot y = B \cdot u \quad (37)$$

Ce modèle correspond en fait à un parti-pris sur la manière dont on traite les perturbations. En réalité on prend en compte un signal d'erreurs possibles  $e$  (de moyenne nulle) tels que :

$$A \cdot y = B \cdot u + e \quad (38)$$

Si l'on fait cette hypothèse de modélisation alors la méthode optimale est celle des moindres carrés. Malheureusement cette manière de prendre en compte les perturbations est bien restrictive. D'autres modélisations sont possibles, adaptées à d'autres algorithmes, par exemple le modèle OE :

$$y = \frac{B}{F} \cdot u + e \quad (39)$$

Il est aussi possible de "structurer" le modèle de bruit comme par exemple dans les modèles ARMAX ou BJ :

$$\begin{aligned} A \cdot y &= B \cdot u + C \cdot e \\ y &= \frac{B}{F} \cdot u + \frac{C}{D} \cdot e \end{aligned} \quad (40)$$

La structure du modèle étant choisie il faut choisir l'ordre de ses différents polynômes.

### 5.2.3 Choix d'un critère à minimiser

Dans la plupart des cas le critère à minimiser est l'erreur quadratique entre la sortie et la sortie prédite à un coup (one step ahead prediction). Ceci n'est bien évidemment pas restrictif et peut être généralisé à la prédiction à  $N$  coups.

### 5.2.4 Choix d'un algorithme

Même si chaque structure de modèle est généralement associée à un algorithme "optimal" celui-ci doit généralement se paramétrer.

Prenons par exemple l'algorithme des moindres carrés récursifs tel que présenté dans (36). Cette équation peut être modifiée en intégrant un "facteur d'oubli"  $\lambda$  :

$$\begin{aligned} \theta(t) &= \theta(t-1) + F(t) \cdot \varphi(t) \cdot (y(t) - \theta^T(t-1) \cdot \varphi(t)) \\ \text{avec} & \\ F(t)^{-1} &= \lambda \cdot F(t-1)^{-1} + \varphi(t) \cdot \varphi(t)^T \end{aligned} \quad (41)$$

Ce paramètre (typiquement entre 0.95 et 0.99) permet, comme son nom l'indique, "d'oublier" les anciennes mesures et de se concentrer sur les mesures les plus récentes.

### 5.2.5 Validation

Une fois l'algorithme d'identification appliqué et le modèle obtenu (ce qui prend généralement quelques secondes avec un ordinateur moderne) il reste à expertiser celui-ci. Selon la "qualité" du modèle obtenu on remet en cause l'une ou l'autre des étapes (signal d'excitation, structure du modèle, algorithme) et on réitère jusqu'à obtenir un modèle satisfaisant.

En pratique on essaie un certain nombre de modèles avant de choisir le "meilleur".

Le premier critère de qualité est la capacité du modèle à prédire effectivement la sortie (sortie prédite à un coup, sortie prédite à N coups). Il faut pour cela utiliser un jeu de mesures non utilisé pour l'identification. (En pratique le jeu de mesures initiales est coupé en deux : une partie pour l'identification et une partie pour la validation).

Le second critère de qualité est l'analyse du résidu : la composition spectrale de l'erreur de prédiction.

### 5.2.6 Utilisation de logiciels

En pratique il est commode d'utiliser un célèbre outil de calcul scientifique et sa toolboxe d'identification.

Comme les calculs proprement dit ne prennent que quelques secondes il est facile de réaliser un grand nombre d'identifications et de stocker les différents modèles pour les comparer ensuite. On est rapidement noyé sous le grand nombre de modèles possibles...

## 5.3 Identification "simple"

Il est souvent suffisant d'obtenir un modèle "simple" de type premier ordre et un simple essai indiciel convient souvent. Le système est excité avec un échelon unité et on cherche le meilleur modèle du premier ordre correspondant.

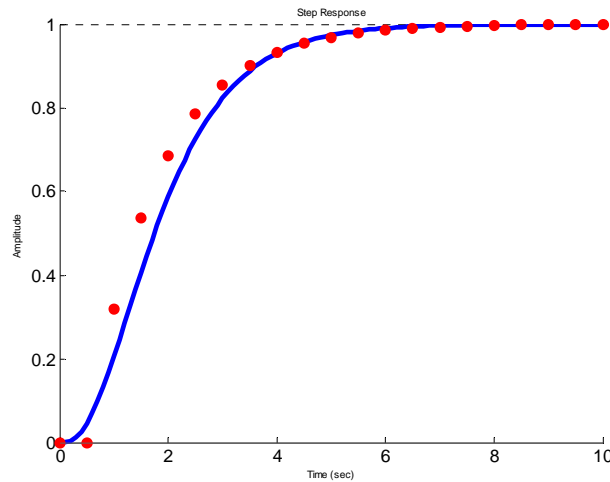


Figure 18 : identification sur un essai indiciel

Dans l'exemple de la Figure 18 on superpose le résultat d'une mesure (en bleu) avec la simulation d'un modèle discret avec une période de 0.5s. Le modèle cherché est du premier ordre avec un retard pur  $d$  :

$$y(z) = \frac{(1 + c_1) \cdot z^{-d}}{1 + c_1 \cdot z^{-1}} \cdot u(z) \quad (42)$$

Il s'agit de trouver l'entier  $d$  et le nombre réel  $c_1$  tels que la réponse indiciale du modèle corresponde pour le mieux aux mesures issues de l'essai. Ceci peut se formaliser par exemple de la manière suivante :

$$(c_1 \quad d_1) = \arg \min \sum_{t=1}^N (y_{\text{modèle}}(t) - y_{\text{mesure}}(t))^2 \quad (43)$$

Cette optimisation peut se faire à la main ou grâce à une procédure automatisée (avec le célèbre outil de calcul scientifique).

Dans le cas où le système comprend naturellement une intégration (par exemple la commande d'un moteur à courant continu par une tension et la mesure de l'angle de rotation) il est utile de prendre en compte explicitement cette intégration dans le modèle :

$$y(z) = \frac{1}{1 - z^{-1}} \cdot \frac{(1 + c_1) \cdot z^{-d}}{1 + c_1 \cdot z^{-1}} \cdot u(z) \quad (44)$$



La procédure est la même mais au lieu de prendre en compte la sortie  $y(t)$  on prend en compte la dérivée  $y'(t)$  :

$$y'(t) = y(t) - y(t-1) \quad (45)$$

## 5.4 Identification non paramétrique

Pour terminer ce panorama rapide sur les méthodes d'identification il faut parler des méthodes non paramétriques : celles qui permettent de tracer la réponse impulsionnelle ou la réponse fréquentielle du système.

### 5.4.1 Estimation de la réponse impulsionnelle

La réponse impulsionnelle d'un système peut être obtenue simplement en appliquant une entrée de type "impulsion" au système et d'en mesurer les conséquences.

La seconde méthode consiste à utiliser la méthode des moindres carrés (ou une méthode dérivée) comme vue au chapitre 5.1. La réponse impulsionnelle d'un système peut en effet être vue comme la fonction de transfert particulière où le dénominateur est de degré nul et le numérateur comporte un nombre infini (en pratique : grand) d'éléments ( $n = 0$  et  $m$  grand) :

$$y(t) = b_0 \cdot u(t) + b_1 \cdot u(t-1) + b_2 \cdot u(t-2) + \dots + b_m \cdot u(t-m) \quad (46)$$

La dernière méthode, qui sera développée ici est l'analyse d'intercorrélation.

Par définition l'intercorrélation  $R_{yu}$  entre deux signaux est :

$$R_{yu}(\tau) = E[(y(t) - E(y(t))) \cdot (u(t-\tau) - E(u(t-\tau)))] \quad (47)$$

Dans cette formule l'espérance mathématique peut être approchée grâce à la loi des grands nombres :

$$R_{yu}(\tau) \approx R_{yu}^N(\tau) = \lim_{N \rightarrow \infty} \frac{1}{N} \sum_{t=1}^N [(y(t) - m_y) \cdot (u(t-\tau) - m_u)] \quad (48)$$

On s'intéresse maintenant à un système tel que décrit par l'équation impulsionnelle (46) et on suppose que le signal d'entrée est un bruit blanc. Le signal d'entrée vérifie donc :

$$\begin{cases} \tau \neq 0 & R_{uu}(\tau) = 0 \\ \tau = 0 & R_{uu}(\tau) = R \end{cases} \quad (49)$$

Il vient alors naturellement :

$$b_k \approx \frac{1}{R} R_{yu}(k) \quad (50)$$

#### 5.4.2 Estimation de la réponse fréquentielle

On utilise ici le spectre d'intercorrélation. Par définition le spectre d'intercorrélation entre deux signaux est :

$$\Phi_{yu}(\omega) = \sum_{\tau=-\infty}^{\infty} R_{yu}(\tau) \cdot e^{-i\omega\tau} \quad (51)$$

On suppose que le signal d'entrée  $u(t)$  et le signal de sortie  $y(t)$  sont reliés par une fonction de transfert linéaire du type :

$$y(t) = F \cdot u(t) + v(t) \quad (52)$$

où  $v(t)$  est une perturbation indépendante de  $u(t)$ . On a alors les relations :

$$\Phi_{yu}(\omega) = F(j\omega) \cdot \Phi_{uu}(\omega) \quad (53)$$

Il reste à estimer les spectres d'intercorrélation pour remonter à la réponse fréquentielle  $F(j\omega)$  recherchée. Une idée naturelle est d'utiliser la relation (48) mais cela marche très mal dans presque tous les cas, même en choisissant  $N$  assez grand. L'amélioration est d'utiliser une "fenêtre" pour favoriser les valeurs faibles de  $\tau$  par rapport aux valeurs élevées. La solution classique utilisant la *fenêtre de Hamming* donne :

$$\Phi_{yu}^N(\omega) = \sum_{\tau=-\gamma}^{\gamma} R_{yu}^N(\tau) \cdot w_{\gamma}(\tau) \cdot e^{-i\omega\tau} \quad (54)$$

avec

$$\begin{cases} w_{\gamma}(\tau) = \frac{1}{2} \left( 1 + \cos\left(\frac{\pi \cdot \tau}{\gamma}\right) \right) & |\tau| < \gamma \\ w_{\gamma}(\tau) = 0 & |\tau| > \gamma \end{cases}$$

On utilise ensuite la relation (51) pour estimer la réponse fréquentielle :

$$F(j \cdot \omega) = \frac{\Phi_u^N(\omega)}{\Phi_{yu}^N(\omega)} \quad (55)$$



## 6. SYSTEMES BOUCLES

### 6.1 Fonction de transfert en boucle fermée

Un système et sa commande par ordinateur peut être représenté par un ensemble de fonctions de transfert formant un schéma-bloc plus ou moins compliqué (Figure 19).

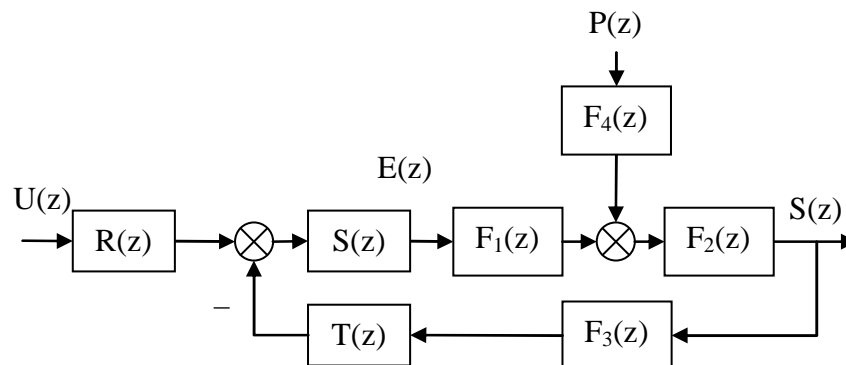


Figure 19 : système discret commandé par ordinateur. Le correcteur est représenté par les fonctions R,S,T, le système par  $F_1$  et  $F_2$ , le modèle de perturbations par  $F_4$  et le modèle des actionneurs par  $F_4$ .

Quelle que soit la structure d'un système asservi il est possible de se ramener à une forme canonique comprenant une fonction de transfert en boucle ouverte en combinant judicieusement les fonctions de transfert du procédé, du régulateur et des perturbations. La Figure 20 représente par exemple le système de la Figure 19 où  $F$  est une combinaison des fonctions de transfert du système, du régulateur, des perturbations et des capteurs.

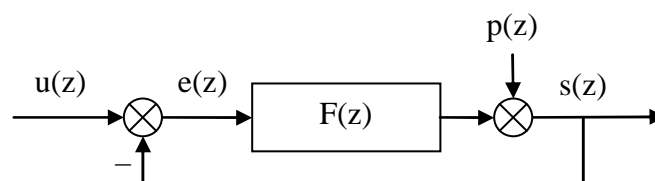


Figure 20 : système discret, notations.

Par défaut on considère une contre-réaction négative et une perturbation additive sur la sortie. La fonction de transfert  $F$  est appelée *fonction de transfert en boucle ouverte*. Comme pour les systèmes continus la *fonction de transfert en boucle fermée* est obtenue simplement à partir de  $F$  (58).

$$F(z) = \frac{s(z)}{e(z)} = \frac{B(z^{-1})}{A(z^{-1})} \quad (56)$$

$$G(z) = \frac{s(z)}{u(z)} = \frac{B(z^{-1})}{A(z^{-1}) + B(z^{-1})}$$

## 6.2 Erreur statique

L'erreur statique du système en boucle fermée, comme pour le système en boucle ouverte, est simplement le rapport des coefficients.

$$G_0 = \frac{\sum_{i=0}^{i=m} b_i}{1 + \sum_{i=0}^{i=m} b_i + \sum_{i=1}^{i=n} a_i} \quad (57)$$

Si l'on souhaite que l'asservissement confère un gain statique unitaire au système en boucle fermée (c'est-à-dire que l'on force la sortie  $s(k)$  à tendre vers la valeur de consigne  $u(k)$ ) il faut que ce rapport de coefficients soit égal à un. On montre facilement que cela est obtenu si la fonction de transfert  $F(z)$  comporte un intégrateur :

$$F(z) = \frac{s(z)}{e(z)} = \frac{1}{1 - z^{-1}} \cdot \frac{B(z)}{A'(z)} \quad (58)$$

## 6.3 Rejet de perturbations

La fonction de transfert entre la perturbation  $p(z)$  et la sortie  $y(z)$  est appelée fonction de sensibilité perturbation-sortie, notée  $S_{yp}$ .

$$S_{yp}(z) = \frac{s(z)}{p(z)} = \frac{A(z^{-1})}{A(z^{-1}) + B(z^{-1})} \quad (59)$$

Si l'on souhaite qu'une perturbation de type échelon soit parfaitement rejetée il faut que  $A(1) = 0$ , on retrouve donc la condition (58) à savoir que le système doit comporter une intégration dans la boucle directe. On peut être en outre être amené à spécifier que les perturbations ne soient pas amplifiées dans une certaine bande de fréquence. On rajoute donc une contrainte du type  $|S_{yp}(e^{j2\pi f})| < S_{\max}$  pour  $f$  appartenant à un certain domaine de  $[0 \ 0.5]$ .

## 6.4 Stabilité des systèmes à simple gain de rétroaction

Les critères de Routh et de Jury (donnés dans le chapitre 3.2.7 p13) ou le calcul numérique des pôles permettent de déterminer la stabilité du système en boucle fermée si l'on connaît sa fonction de transfert.

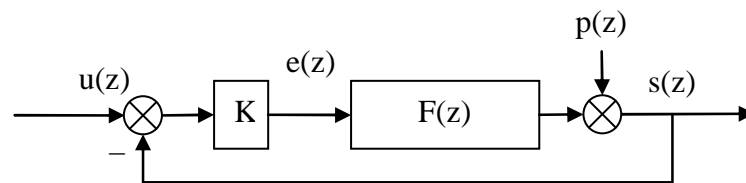


Figure 21 : système bouclé par un simple gain.

On s'intéresse ici au cas particulier où l'asservissement est réalisé par un simple gain de rétroaction  $K$  (Figure 21). On cherche la valeur de  $K$  qui amène le système en limite de stabilité. L'outil adapté ici est le lieu d'Evans ou lieu des racines (root locus) qui permet de visualiser l'évolution des pôles du système bouclé lorsque le gain  $K$  varie.

Prenons l'exemple d'un système comportant trois pôles et deux zéros. L'exemple est donné sous forme de script Matlab :

### Matlab

```
% Définition d'un système discret :
sys=zpk([-2.037 -0.1406],[0.6065 0.3679 0.3679],1,1)
%Tracé du lieu des racines
rlocus(sys);
```

Le script précédent produit le tracé de la Figure 22. On pourrait y lire que la limite de stabilité est obtenue pour  $K=0.2$  environ (détail de la Figure 23).

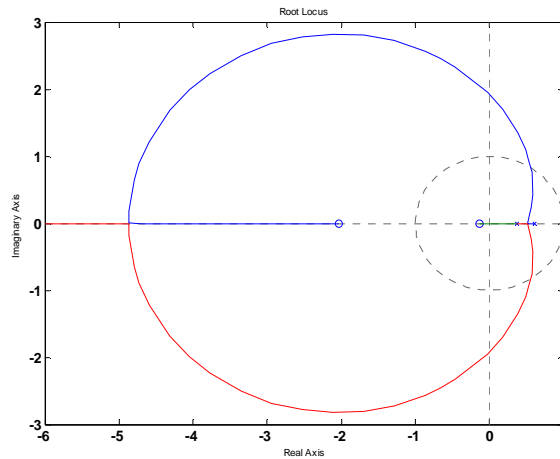


Figure 22 : lieu des racines

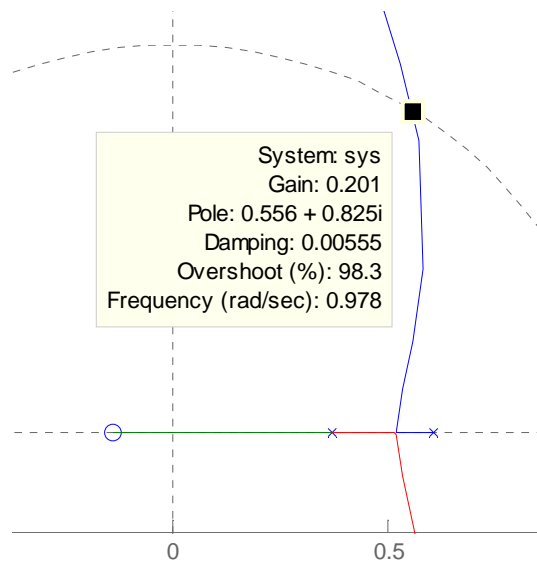


Figure 23 : lieu des racines : détail de la limite de stabilité

On voit sur la Figure 22 que le lieu des racines d'un système échantillonné ressemble à celui d'un système continu :

- on commence par tracer les pôles ( $\times$ ) et les zéros ( $o$ ) du système en boucle ouverte.
- les pôles du système en boucle fermée décrivent un lieu qui part des pôles (pour  $K = 0$ ) et arrive aux zéros (pour  $K = \infty$ )
- lorsque il y a  $n-m$  plus de pôles que de zéros le système comporte  $n-m$  branches asymptotiques.

On ne décrira pas ici les diverses propriétés du lieu des racines qui permettent de le tracer à la main puisque il est facile à obtenir avec un ordinateur.



Il peut aussi être utilisé pour régler "à la main" le gain permettant d'obtenir un ou plusieurs modes dominants ayant certaines propriétés de fréquence et d'amortissement. On voit sur la Figure 24 que le gain 0.017 permet d'obtenir un pôle double d'amortissement 0.706 et de fréquence 0.65 rad/s et un pôle simple de fréquence 1.63 rad/s.

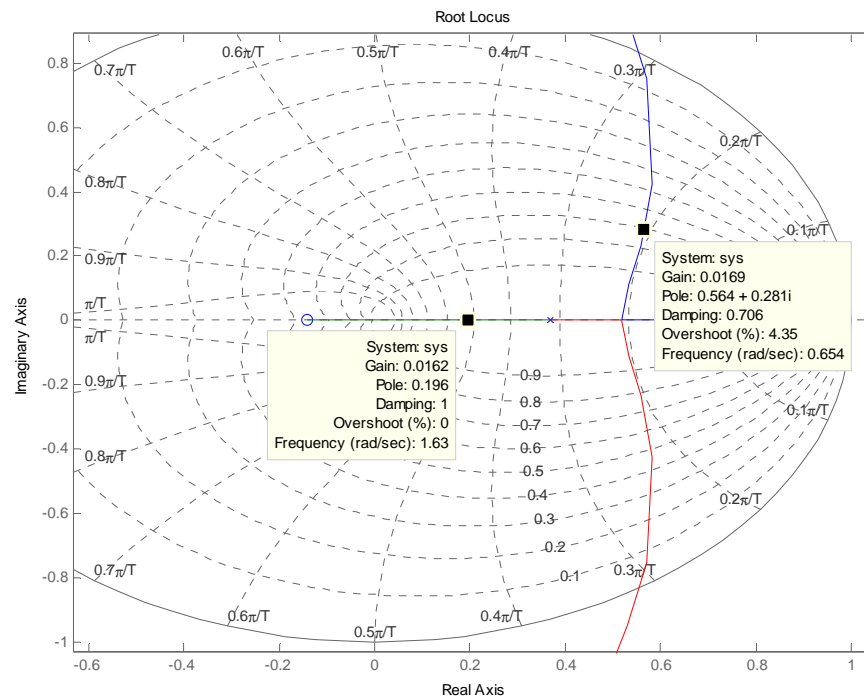


Figure 24 : utilisation du lieu des racines pour régler le gain de la boucle fermée

L'outil rltool de Matlab permet de nombreuses manipulations graphiques de ce type.

## 6.5 Analyse des systèmes bouclés dans le domaine fréquentiel : stabilité

Comme dans le cas continu il est possible de prévoir des propriétés, en particulier la stabilité, d'un système en boucle fermée à partir de l'examen de la fonction de transfert en boucle ouverte.

On ne redémontrera pas ici le théorème de Nyquist mais on se contente de rappeler l'énoncé.

On rappelle que le *lieu de Nyquist* est le lieu de  $F(j2\pi f)$  dans le plan complexe pour  $f$  variant de 0 à  $+\infty$ . A noter que le lieu pour la zone  $[0, 0.5]$  est symétrique de celui pour la zone  $[0.5, 1]$ . Pour cette raison on ne représente généralement que le lieu pour  $f$  appartenant à  $[0, 0.5]$ .

### 6.5.1 Théorème de Nyquist dans le cas où le système est stable en boucle ouverte (critère du revers)

Le système est stable en boucle fermée si le lieu de Nyquist de la fonction de transfert en boucle ouverte parcouru dans le sens des fréquences croissantes de 0 à 0.5 laisse le point critique  $(-1,0)$  à gauche.

La Figure 25 représente par exemple le lieu de Nyquist d'une fonction de transfert en boucle ouverte qui sera stable en boucle fermée.

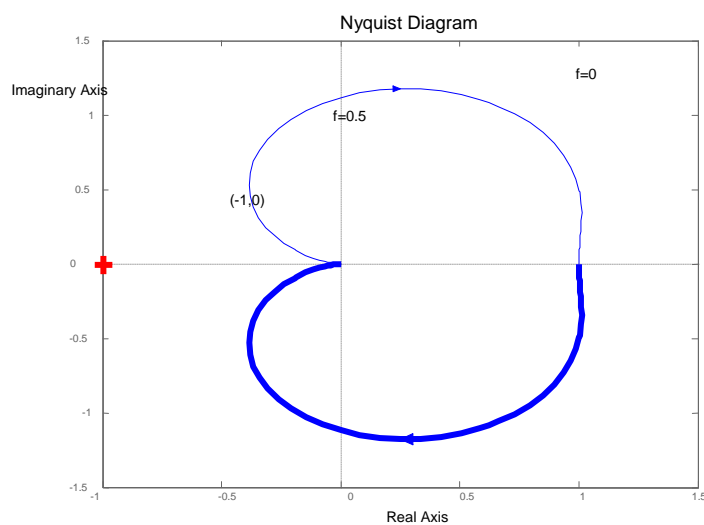


Figure 25 : lieu de Nyquist d'un système stable

### 6.5.2 Théorème de Nyquist dans le cas où le système est instable en boucle ouverte

Si le système en boucle ouverte comporte  $N$  pôles instables il sera stable en boucle fermée si le lieu de Nyquist de la fonction de transfert en boucle ouverte parcouru dans le sens des fréquences croissantes de  $-0.5$  à  $0.5$  entoure le point critique  $(-1,0)$   $N$  fois dans le sens des aiguilles d'une montre..

### 6.5.3 Interprétations du lieu de Nyquist

Comme dans le cas continu, la distance à laquelle on passe du point critique renseigne sur la robustesse du système bouclé. La distance sera exprimée de différente manière (Figure 26) :

- marge de module  $\Delta M$  : plus petite distance entre le lieu des racines et le point critique,

- marge de gain  $\Delta G$
- marge de phase  $\Delta P$

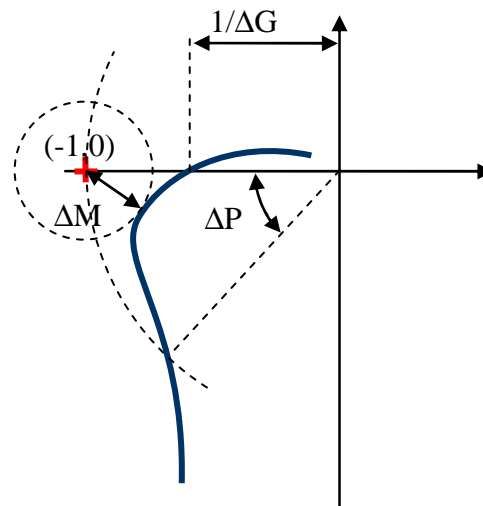


Figure 26 : Marges de stabilité

Les valeurs typiques de ces marges sont données ci-après :

- $\Delta G \geq 6\text{dB}$
- $\Delta M \geq -6\text{dB}$
- $30^\circ \leq \Delta P \leq 60^\circ$

La distance entre le point critique et le lieu des racines est la norme de  $1+F(j2\pi f)$ . On peut donc exprimer la marge de module en fonction de la fonction de sensibilité :

$$\begin{aligned} \Delta M &= \min |1 + F(j \cdot \omega)|_{\omega \in [0, \pi]} \\ \Delta M &= \min \left| 1 + \frac{B(j \cdot \omega)}{A(j \cdot \omega)} \right|_{\omega \in [0, \pi]} = \max \left| \frac{A(j \cdot \omega)}{A(j \cdot \omega) + B(j \cdot \omega)} \right|_{\omega \in [0, \pi]} \quad (60) \\ \Delta M &= \max |S_{yp}(j \cdot \omega)|_{\omega \in [0, \pi]}^{-1} \\ \Delta M_{\text{dB}} &= -\max |S_{yp}(j \cdot \omega)|_{\text{dB}} \end{aligned}$$

Si l'on trace l'évolution de  $S_{yp}$  en fonction de  $f$  on obtient donc la valeur de la marge de module.



# 7. METHODES DE REGULATION DES SYSTEMES ECHANTILLONES

## 7.1 Introduction

Les bases théoriques permettant de concevoir le correcteur pour un système échantillonné ont été posées dans les chapitres précédents. Ce chapitre fait le bilan des connaissances acquises et donne quelques solutions pour calculer des correcteurs à partir d'un cahier des charges précis.

Les systèmes que l'on cherche à commander sont généralement, par nature, continus. La réalisation d'un correcteur par un ordinateur implique que l'on génère une équation de commande discrète (généralement une équation récurrente). L'obtention de cette équation de commande peut se faire de deux manières :

- la première méthode consiste à d'abord synthétiser un correcteur continu (en utilisant les outils classiques de l'automatique et la simulation) et discrétiser ensuite ce correcteur. C'est l'objet du chapitre 7.2.2.
- la seconde méthode consiste à tenir la représentation discrète du système continu associé à son interface numérique (convertisseurs analogique-numérique et numérique-analogique) comme expliqué dans le chapitre 4.5 puis synthétiser directement le correcteur discret. Cela sera expliqué dans le chapitre 7.3 et suivants.

Ce choix peut être résumé par la figure suivante :

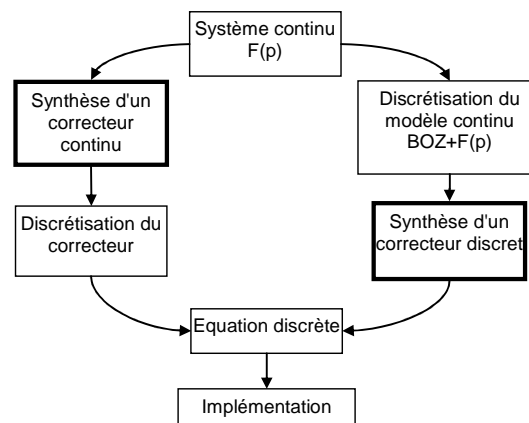


Figure 27 : Réalisation d'un correcteur discret.

Au-delà des affaires de goût, le choix de la méthode sera souvent dicté par l'application. En effet, la première méthode est d'autant meilleure que le correcteur discrétisé approche correctement le correcteur continu. Cela implique que le temps d'échantillonnage soit très faible par rapport aux constantes de temps du système

bouclé. Dans le cas où les contraintes temps réel sont fortes (capacités de calcul partagées par exemple) il sera préférable de synthétiser un correcteur discret.

## 7.2 Structure d'un système commandé par ordinateur, hypothèses

Par la suite on se ramènera toujours à un système tel que celui représenté sur la Figure 28 en cherchant un correcteur  $K(z)$  linéaire. On montre que quel que soit le correcteur linéaire on peut se ramener à la forme représenté sur la Figure 29 (forme RST).

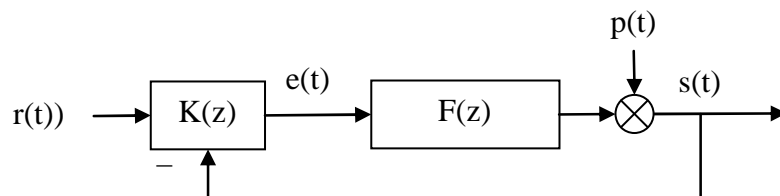


Figure 28 : système bouclé, forme générale.

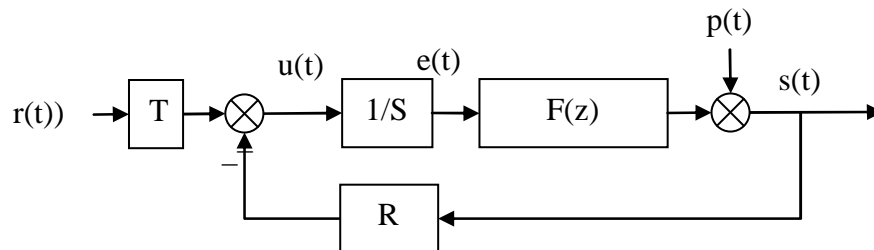


Figure 29 : système bouclé, forme canonique.

### 7.2.1 Discrétisation, choix de la période d'échantillonnage

Du point de vue du ordinateur, l'extérieur (le procédé) est représenté par la fonction de transfert  $F(z)$ . L'hypothèse importante est que la commande (sortie du ordinateur) et la mesure (entrée du ordinateur) sont échantillonnées de manière synchrone à la même période  $T_e$ .

Le choix judicieux de la fréquence d'échantillonnage est très important. En pratique on choisit cette fréquence en fonction de la bande passante du système en **boucle fermée** :

$$f_e = (6 \text{ à } 25) f^{BF} \quad (61)$$

### 7.2.2 Discrétisation d'un correcteur continu

La première méthode pour obtenir les équations d'un correcteur numérique est de faire sa synthèse avec les outils de l'automatique continue puis de discrétiser la fonction de transfert du correcteur continu obtenu (voir pour cela le chapitre 4.2 *Echantillonnage des systèmes continus*).

On va voir par la suite que la synthèse directe d'un correcteur discret offre plus de degré de liberté et permet en général de remplir un cahier des charges plus complet.

## 7.3 Régulateur PID numérique

Le régulateur le plus populaire est le régulateur PID. Nous allons voir dans un premier temps comment on génère un régulateur PID à partir des équations bien connues du régulateur PID continu et comment ces équations peuvent être généralisées pour obtenir plus de degrés de liberté pour le réglage.

### 7.3.1 Cas 1

Un régulateur PID continu a une fonction de transfert du type :

$$K_{\text{PID}}(s) = K \cdot \left( 1 + \frac{1}{T_i \cdot s} + \frac{T_d \cdot s}{1 + \frac{T_d}{N} \cdot s} \right) \quad (62)$$

La discrétisation de ce correcteur continu par la méthode d'Euler (Tableau 4 page 31) se fait en remplaçant la variable de Laplace  $s$  par le terme  $\frac{1-q^{-1}}{q^{-1} \cdot T_e}$ . On obtient :

$$K_{\text{PID}}(q) = K \cdot \left( 1 + \frac{1}{T_i \cdot \frac{1-q^{-1}}{T_e}} + \frac{T_d \cdot \frac{1-q^{-1}}{T_e}}{1 + \frac{T_d}{N} \cdot \frac{1-q^{-1}}{T_e}} \right) \quad (63)$$

En réarrangeant on obtient la forme générale RST (Figure 30):

$$K_{PID}(q) = K \cdot \frac{r_0 + r_1 \cdot q^{-1} + r_2 \cdot q^{-2}}{(1 - q^{-1}) \cdot (1 + s_1' \cdot q^{-1})}$$

$$\begin{cases} r_0 = K \cdot \left( 1 + \frac{T_e}{T_i} + \frac{N \cdot T_d}{T_d + N \cdot T_e} \right) \\ r_1 = -K \cdot \left( 1 + \frac{T_d}{T_d + N \cdot T_e} + \frac{T_e}{T_i} \cdot \frac{T_d}{T_d + N \cdot T_e} + \frac{N \cdot T_d}{T_d + N \cdot T_e} \right) \\ r_2 = K \cdot \left( \frac{T_d}{T_d + N \cdot T_e} \right) \\ s_1' = \frac{T_d}{T_d + N \cdot T_e} \end{cases} \quad (64)$$

Comme on le voit le correcteur obtenu comporte autant de degrés de liberté que le correcteur PID analogique. On reconnaît au dénominateur le terme  $(1 - q^{-1})$  en facteur qui assure l'effet d'intégration.

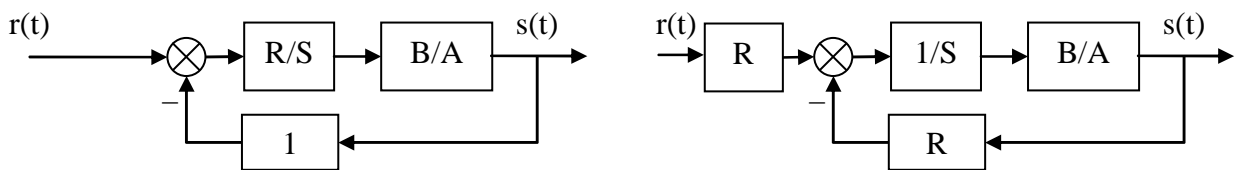


Figure 30 : PID sous forme RST

La fonction de transfert en boucle fermée obtenue est donnée par l'équation suivante :

$$H_{BF} = \frac{B \cdot R}{A \cdot S + B \cdot R} = \frac{B \cdot R}{P} \quad (65)$$

De manière générale le polynôme caractéristique  $P$  fixe les performances visées en boucle fermée. Le calcul de  $R$  et  $S$ ,  $P$  étant fixé et  $A$  et  $B$  étant connus, est réalisé en résolvant l'équation de Bézout  $P = A \cdot S + B \cdot R$ . (voir 9.1 Annexe 1 : Résolution de l'équation polynomiale de Bezout)

On voit que le numérateur  $B \cdot R$  est imposé une fois que  $R$  et  $S$  sont trouvés. Le régulateur introduit donc naturellement des zéros dans la boucle fermée.

Traisons l'exemple entièrement dans le cas où le système est modélisé comme un système du deuxième ordre avec faible retard. La fonction de transfert du procédé est donnée par :



$$H_{BO} = \frac{B}{A} = \frac{b_1 \cdot q^{-1} + b_2 \cdot q^{-2}}{1 + a_1 \cdot q^{-1} + a_2 \cdot q^{-2}} \quad (66)$$

Le correcteur PID rajoutant deux pôles, le polynôme caractéristique en boucle fermée spécifié doit posséder quatre pôles. Le polynôme caractéristique en boucle fermée doit donc être de la forme :

$$\begin{aligned} P &= 1 + p_1 \cdot q^{-1} + p_2 \cdot q^{-2} + p_2 \cdot q^{-3} + p_2 \cdot q^{-4} \\ P &= (1 + p_1' \cdot q^{-1} + p_2' \cdot q^{-2}) \cdot (1 + \alpha_1 \cdot q^{-1}) \cdot (1 + \alpha_2 \cdot q^{-1}) \end{aligned} \quad (67)$$

Et il doit être identifié au produit  $AS + BR$  avec  $R$  et  $S$  ayant les formes données par (64) :

$$\begin{aligned} &(1 + p_1' \cdot q^{-1} + p_2' \cdot q^{-2}) \cdot (1 + \alpha_1 \cdot q^{-1}) \cdot (1 + \alpha_2 \cdot q^{-1}) \\ &= (1 + a_1 \cdot q^{-1} + a_2 \cdot q^{-2}) \cdot (1 - q^{-1}) \cdot (1 + s_1' \cdot q^{-1}) \\ &+ (b_1 \cdot q^{-1} + b_2 \cdot q^{-2}) \cdot (r_0 + r_1 \cdot q^{-1} + r_2 \cdot q^{-2}) \end{aligned} \quad (68)$$

Les coefficients  $p_1'$  et  $p_2'$  définissent un modèle du second ordre aux caractéristiques (pulsation propre  $\omega_0$  et amortissement  $\sigma$  choisis et coefficients déduits conformément au Tableau 3 page 25). Les coefficients  $\alpha_1$  et  $\alpha_2$  correspondent à des pôles auxiliaires aperiodiques. Les choix typiques pour ces coefficients sont :

$$-0.05 < \alpha_1, \alpha_2 < -0.5 \quad (69)$$

On montre que la résolution de (68) est réalisée par une inversion matricielle simple :

$$M \cdot x = p \quad (70)$$

avec

$$\begin{cases} x^T = [1 & s_1' & r_0 & r_1 & r_2] \\ p^T = [1 & p_1 & p_2 & p_3 & p_4] \end{cases}$$

et

$$M = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ a_1' & 1 & b_1 & 0 & 0 \\ a_2' & a_1' & b_2 & b_1 & 0 \\ a_3' & a_2' & 0 & b_2 & b_1 \\ 0 & a_3' & 0 & 0 & b_2 \end{pmatrix} \quad (71)$$

On montre que M est inversible à la condition que A et B soient premiers entre eux, c'est-à-dire que A et B n'aient pas de pôles et de zéros en commun.

(Le problème peut aussi être résolu en utilisant la résolution de l'équation de Bézout).

### 7.3.2 Cas 2

Comme on l'a vu le correcteur PI numérique directement transposé du correcteur PID analogique décrit dans le chapitre 7.3.1 a l'inconvénient de ramener des zéros au dénominateur de la fonction de transfert en boucle fermée (équation 65). Il est possible, grâce à la flexibilité du régulateur numérique d'imposer au correcteur de ne pas ajouter de zéros en boucle fermée. La fonction de transfert désirée en boucle fermée sera donc de la forme :

$$H_{BF} = \frac{P(1)}{B(1)} \cdot \frac{B}{P} \quad (72)$$

Le facteur  $P(1)/B(1)$  garantit le gain statique unitaire du système en boucle fermée (voir page 12). Si le régulateur a la forme générale RST alors les polynômes R S et T doivent vérifier :

$$H_{BF} = \frac{B \cdot T}{A \cdot S + B \cdot R} = \frac{P(1) B}{B(1) P} \quad (73)$$

S et R sont obtenus par la résolution de la même équation que précédemment (68). Le polynôme T est obtenu par simple identification :  $T = R(1)$ .

La structure, sous forme de schéma-bloc est donnée ci-dessous :

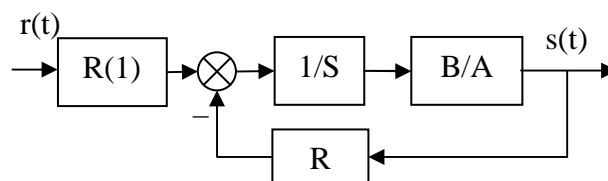


Figure 31 : PID2 sous forme RST

On montre que cette structure de PID, ramené en continu, correspond au PID canonique pour l'action intégrale. L'action proportionnelle et dérivée n'est appliquée que sur le signal  $y$ .

La structure du PID2 par rapport au PID1 a pour avantage de ne pas ramener de zéros à la boucle fermée. Cela se traduit par un dépassement moindre en transitoire lorsqu'on applique un échelon de commande. Par contre les deux systèmes réagissent de la même manière à un échelon de perturbation.

## 7.4 Placement de pôles

Les deux structures de PID décrites dans le chapitre précédent sont deux cas particulier d'une stratégie plus générale de placement de pôles.

La synthèse est plus générale que les PID présentés précédemment puisqu'il n'est pas nécessaire de faire d'hypothèses sur le degré de A et B, sur le retard, ni sur la présence ou non de zéros stables ou instables dans le procédé.

La structure du correcteur est la structure type RST telle que présentée dans la Figure 29. Pour la suite des calculs on posera pour la fonction de transfert les conventions suivantes :

$$H_{BO} = \frac{q^{-d} \cdot B(q^{-1})}{A(q^{-1})} \quad (74)$$

$$\begin{cases} A(q^{-1}) = 1 + a_1 \cdot q^{-1} + \dots + a_{n_A} \cdot q^{-n_A} \\ B(q^{-1}) = b_1 \cdot q^{-1} + \dots + b_{n_B} \cdot q^{-n_B} = q^{-1} \cdot B^*(q^{-1}) \end{cases}$$

Compte tenu de la forme RST du régulateur la fonction de transfert en boucle fermée vérifie :

$$H_{BF} = \frac{q^{-d} \cdot B(q^{-1}) \cdot T(q^{-1})}{P(q^{-1})} \quad (75)$$

$$P(q^{-1}) = A(q^{-1}) \cdot S(q^{-1}) + q^{-d} \cdot B(q^{-1}) \cdot R(q^{-1}) = 1 + p_1 \cdot q^{-1} + \dots + p_{n_p} \cdot q^{-n_p}$$

Le polynôme  $P(q^{-1})$  définit donc les pôles de la fonction de transfert en boucle fermée et traduit donc le comportement en *poursuite*.

De la même manière la fonction de transfert entre une perturbation et la sortie (fonction de sensibilité perturbation-sortie) est donnée par :

$$S_{yp} = \frac{A(q^{-1}) \cdot S(q^{-1})}{P(q^{-1})} \quad (76)$$

Ce même polynôme traduit donc aussi le comportement en *régulation*.

Les consignes en poursuite et en régulation peuvent être différentes comme cela est représenté par exemple sur la Figure 32.

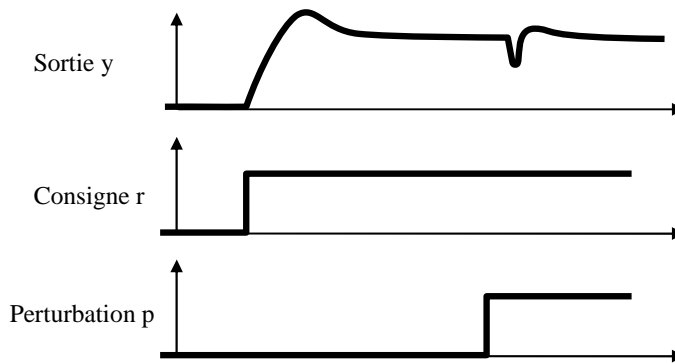


Figure 32 : Performances différentes en régulation et en poursuite

### 7.4.1 Choix des pôles en boucle fermée

De manière générale le comportement en boucle fermée est spécifié sous la forme d'une pulsation propre  $\omega_0$  et d'un amortissement  $\sigma$ , donc une fonction de transfert du second ordre. La spécification étant généralement donnée en continu il est nécessaire de traduire la valeur désirée de  $\omega_0$  et  $\sigma$  en un polynôme  $P_D(q^{-1})$  par discrétisation de la fonction de transfert continue (chapitre 4.5 page 29). Ce polynôme  $P_D$  traduit le comportement dominant du système bouclé.

Si l'on souhaite introduire un effet de filtrage supplémentaire dans certaines bandes de fréquence (réduction des bruits de mesure, adoucissement de la commande, robustesse) cela est décrit par un polynôme  $P_F$  comportant des pôles auxiliaires, plus rapides que les pôles de  $P_D$ .

Le polynôme caractéristique en boucle fermée est donc constitué de pôles dominants et de pôles auxiliaires :

$$P(q^{-1}) = P_D(q^{-1}) \cdot P_F(q^{-1}) \quad (77)$$

### 7.4.2 Régulation : calcul de R et S

Le polynôme  $P(q^{-1})$  étant fixé il s'agit de trouver R et S vérifiant :

$$P(q^{-1}) = A(q^{-1}) \cdot S(q^{-1}) + q^{-d} \cdot B(q^{-1}) \cdot R(q^{-1}) \quad (78)$$

Comme déjà vu au chapitre 7.3.1 on obtient une équation de Bézout qu'il s'agit de résoudre. (Voir aussi 9.1 Annexe 1 : Résolution de l'équation polynomiale de Bezout)

Il est parfois nécessaire de spécifier R et S avec des parties fixes  $H_R$  et  $H_S$ . Dans ce cas l'équation (78) est modifiée et on l'écrit sous la forme:

$$P(q^{-1}) = (A(q^{-1}) \cdot H_S(q^{-1})) \cdot S'(q^{-1}) + (q^{-d} \cdot B(q^{-1}) \cdot H_R(q^{-1})) \cdot R'(q^{-1}) \quad (79)$$

Les inconnues que l'on souhaite trouver sont  $S'$  et  $R'$ . Ceci est nécessaire dans les cas particuliers suivants :

- spécification d'erreur statique nulle,
- rejet d'une perturbation harmonique,
- blocage d'un signal,

Détaillons maintenant ces cas particuliers.

### Spécification d'une erreur statique nulle

On retrouve la nécessité d'avoir un terme intégrateur dans la boucle directe. La spécification se retrouve donc sur  $S$  :

$$S(q^{-1}) = (1 - q^{-1}) \cdot S'(q^{-1}) = H_S(q^{-1}) \cdot S'(q^{-1}) \quad (80)$$

### Rejet d'une perturbation harmonique

Si le système est perturbé par une perturbation harmonique à une fréquence fixe (par exemple 50Hz...) il est possible d'imposer à  $S$  d'avoir un module *nul* ou amorti à cette fréquence. Cela se traduit par la présence dans  $S$  d'une paire de zéros complexe à amortissement nul ou faible à cette fréquence :

$$S(q^{-1}) = (1 + \alpha \cdot q^{-1} + \beta \cdot q^{-2}) \cdot S'(q^{-1}) = H_S(q^{-1}) \cdot S'(q^{-1}) \quad (81)$$

### Blocage d'un signal

Pour certaines application on souhaite que le régulateur "laisse passer" certaines composantes du signal de sortie, c'est-à-dire qu'il ne réagisse pas à ces composantes. Cela se traduit par  $R=0$  à ces fréquences. On retrouve la même forme que précédemment :

$$R(q^{-1}) = (1 + \alpha \cdot q^{-1} + \beta \cdot q^{-2}) \cdot R'(q^{-1}) = H_R(q^{-1}) \cdot R'(q^{-1}) \quad (82)$$

### 7.4.3 Poursuite : calcul de T

Le schéma-bloc de l'asservissement est donné sur la Figure 33. Le terme  $T$  est décomposé en un terme  $T'$  et un terme  $B^*/A^*$ .

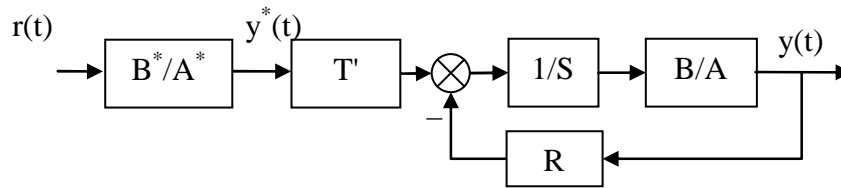


Figure 33 : placement de pôle

L'idée est de régler  $B^*$  et  $A^*$  de telle manière que le signal  $y^*$  ait la dynamique de poursuite désirée (par exemple un second ordre avec pulsation propre et amortissement donné) puis de régler  $T'$  de telle manière que le transfert entre  $y^*$  et  $y$  soit le plus "transparent" possible.

### Réglage de $T'$

Le transfert entre  $y^*$  et  $y$  est :

$$S_{yy^*}(q^{-1}) = T'(q^{-1}) \frac{q^{-d} \cdot B(q^{-1})}{P(q^{-1})} \quad (83)$$

Pour assurer le transfert le plus "transparent" possible il faut donc choisir

$$T'(q^{-1}) = \frac{P(q^{-1})}{B(1)} \quad (84)$$

Remarques :

- Si  $B(1)=0$  alors on remplace  $B(1)$  par 1 dans (84)
- De même dans (84)  $P(q^{-1})$  peut être remplacé par  $P_D$  correspondant à la dynamique dominante.

### Réglage de $B^*$ et $A^*$

$B^*$  et  $A^*$  sont choisis de telle manière que la fonction de transfert  $B^*/A^*$  corresponde à la dynamique de poursuite souhaitée.

## 7.5 Poursuite et régulation à objectifs indépendants

Cette méthode ressemble à la méthode de placement de pôles à ceci près que l'on va effectuer une simplification des zéros de  $B^*(q^{-1})$  : cela ne s'applique donc qu'aux systèmes avec des pôles stables. Plus précisément il faut qu'ils aient un amortissement suffisant ( $> 0,2$ ) de manière à ne pas être trop près du cercle unité.

La structure de l'asservissement est toujours celle donnée sur la Figure 33.

**Choix de P**

$P(q^{-1})$  est choisi de manière à simplifier  $B^*(q^{-1})$  (rappel : la forme de B est donnée dans l'équation 74). Comme indiqué dans le chapitre 7.4.2 page 60 il s'agit donc d'imposer les zéros de  $B(q^{-1})$  dans  $H_R(q^{-1})$  et  $H_S(q^{-1})$ .

**Choix de T**

Si l'on choisit  $T(q^{-1}) = P(q^{-1})$  alors le transfert entre  $y^*$  et  $y$  est :

$$S_{yy^*}(q^{-1}) = q^{-(d+1)} \quad (85)$$

**Réglage de  $B^*$  et  $A^*$** 

$B^*$  et  $A^*$  sont choisis de telle manière que la fonction de transfert  $B^*/A^*$  corresponde à la dynamique de poursuite souhaitée.





## 8. TECHNOLOGIE DE LA COMMANDE PAR ORDINATEUR

Du point de vue de l'automaticien le ordinateur est relié au système physique à piloter par un certain nombre d'intermédiaires plus ou moins gênants :

- les interfaces analogiques et numériques,
- le transport de l'information par bus et protocole,
- l'utilisation éventuelle de capteurs ou d'actionneurs « intelligents ».

Il ne s'agit pas ici de faire un exposé complet sur ces trois aspects mais de poser les problèmes qui se posent à l'automaticien lorsque le système qu'il doit commander est perturbé par ces éléments.

Enfin ce chapitre se termine sur les notions de « Hardware In the Loop » et de « prototypage rapide » qui sont partie intégrante des processus de réalisation de systèmes de contrôle.

### 8.1 Calculateurs et interfaces numériques / analogiques

Le premier problème à prendre en compte est la quantification de l'information. Du point de vue du ordinateur d'une part et du point de vue des convertisseurs d'autre part. Dans un premier temps, considérons les aspects performance de calcul lié à la nature du codage des nombres dans le ordinateur.

#### Les performances liées au types de variables

Prenons l'exemple où le concepteur de la loi de commande souhaite réaliser un correcteur de type proportionnel. Entre l'acquisition des mesures et l'écriture de la consigne il y aura la réalisation d'une de commande du type :

$$y = K \cdot (\text{consigne} - \text{mesure}) \quad (86)$$

En utilisant un langage de programmation évolué (par exemple le C) la ligne de code sera la transcription littérale de (86). Le programmeur devra décider de la définition et du type des variables, principalement du nombre d'octets utilisés pour stocker ses variables. Le Tableau 5 résume par exemple les types utilisés en langage C (norme ANSI)

Tableau 5 : Types ANSI C

Types entiers	char	8 bits
	short int	16 bits
	long int	32 bits
Types flottants (mantisse et exposant)	float	32 bits
	double	64 bits
	longdouble	128 bits

Une fois le code écrit en C l'exécution du programme sur un processeur cible ne peut se faire que par l'intermédiaire d'une compilation du programme. A chaque processeur est donc adapté une version de compilateur.

Les performances du programme (sa rapidité d'exécution) dépendront donc en partie de la qualité du compilateur. (Par exemple, une multiplication s'effectue t'elle bêtement par un décalage de tous les bits ou bien des solutions astucieuses sont-elles implémentées pour arrêter le décalage selon le contexte ?)

Le principal déterminant de la performance est cependant le processeur : son type (8 bits, 16 bits ou 64 bits) et son cadencement (en MHz ou GHz) sont les deux plus importants paramètres. Coder les variables sur 64 bits (double) dans le programme coûtera cher en temps de calcul si le processeur est un 8 bits. Le type de mémoire et leurs temps d'accès peuvent aussi être déterminants. Enfin, certains processeurs disposent d'un processeur spécifiquement dédié aux opérations mathématiques (coprocesseur mathématique).

Il n'est pas possible de dire exactement, pour tel type de compilateur associé à tel type de processeur, le temps que prendra par exemple une multiplication. En pratique on effectue une série de tests types (benchmark) et on affiche un temps moyen d'exécution.

En résumé, on peut être tenté, pour ne pas s'embêter, de coder toutes les variables en virgule flottante. Mais il faut bien savoir que cela a un coût en terme de temps de calcul, surtout si le processeur n'est pas adapté. Pour fixer les idées on peut dire qu'il y a un rapport 20 entre une addition réalisée en "short" et une addition réalisée en virgule flottante.

Enfin, l'utilisation de fonctions mathématiques (sinus, exponentielles, etc...) peut aussi avoir un coût très lourd si le travail n'est pas réalisé par un coprocesseur mathématique dédié. Le compilateur et sa librairie mathématique joue là encore un grand rôle.

### La quantification imposée par le processeur

Au-delà de l'aspect performance et temps de calcul, la quantification de l'information doit parfois aussi être prise en compte. Rappelons qu'un entier codé sur N bits peut prendre seulement  $2^N - 1$  valeurs différentes. Avant d'effectuer les calculs mathématiques dans le corps du programme il est donc judicieux d'appliquer un

facteur d'échelle de manière à utiliser l'amplitude maximale possible (entre 0 et  $2^N - 1$ ).

### **La quantification imposée par les conversions analogique et numérique**

Les convertisseurs analogique-numérique et numérique-analogique sont principalement caractérisés par leur amplitude de signal analogique d'entrée (généralement en volts, par exemple +/- 10V) et par leur résolution en bits (8 ou 10 bits pour les plus courants, jusqu'à 16 bits pour les convertisseurs haut de gamme).

Prenons l'exemple d'un capteur gyromètre transformant une vitesse de rotation en tension. Compte tenu de l'application le signal utile varie entre 2,4V et 2,6V. Si le signal mesuré est branché directement sur un convertisseur 0-10V 10bits alors un problème de conditionnement du signal se pose. En effet la quantification est de  $\frac{10V}{2^{10}-1} = 0,098V$ . Les 0,2V de signal utile seront donc discrétisés en  $\frac{0,2}{0,098} = 20$  intervalles seulement !

Si la discrétisation du signal utile est trop grossière cela peut se traduire par une diminution des performances de l'asservissement, voire une instabilité. Le système n'est en effet plus linéaire et il n'existe pas d'outils théoriques permettant de prévoir les performances du système bouclé.

## **8.2 Transport de l'information : les bus et les protocoles**

Pas question de faire ni même d'ébaucher un cours sur les bus et les protocoles ici. Mais on va quand même donner quelques indications. L'automaticien a l'habitude de simplifier le problème qu'il se pose en dessinant un schéma simplifié de son problème d'asservissement (du type de la Figure 20 page 45) pour pouvoir appliquer les théories de la commande. Cependant il arrive bien souvent que l'information transite à l'intérieur du système, soit entre différents calculateurs effectuant différentes parties du travail soit entre ordinateur et périphériques, par l'intermédiaire de bus informatique. A chaque type de bus correspond un ou plusieurs protocoles qui peuvent poser des problèmes pour l'automaticien :

- les données sont-elles discrétisées ? (le protocole peut par exemple ne permettre de transmettre que des paquets de données de 16 bits),
- le bus induit-t'il un retard ? (le composant mettant en forme les données du ordinateur pour les publier sur le bus a certaines performances qui peuvent induire un délai)
- le bus est-il déterministe ? (Certains bus ne garantissent pas que les données arriveront en un temps donné),
- le bus est-il sûr ? (Certains bus ne garantissent pas non plus que les données arriveront bien)

### **8.3 Capteurs et actionneurs "intelligents", "smart sensors"**

De plus en plus souvent les capteurs et les actionneurs sont associés à leur électronique de puissance, électronique de commande et de conditionnement de signal. L'interface avec le capteur ou l'actionneur se fait donc via un protocole de communication (par exemple via un bus RS232).

L'intérêt est de véhiculer non pas des signaux analogiques, tensions ou courants, mais des signaux numériques beaucoup moins sensibles aux éventuelles perturbations.

On retrouve donc, entre le capteur ou l'actionneur et le ordinateur, la présence d'un bus et d'un protocole de communication.

### **8.4 La philosophie "hardware in the loop"**

Les systèmes embarqués sont de plus en plus complexes et touchent des domaines de plus en plus variés : automobile, satellites, aéronautique, UAV, systèmes d'arme, sous-marins, systèmes, etc... Les systèmes à piloter ainsi que les systèmes de commande sont de plus en plus complexes.

Les techniques de simulation « Hardware In the Loop » (HIL) sont de plus en plus utilisées dans le développement et dans le test de ces systèmes embarqués. L'idée est de développer entièrement le système de contrôle sur une plateforme de test en utilisant le maximum de composants réels et en ne remplaçant que le minimum par la simulation. Par exemple en aéronautique l'« Iron Bird » regroupe dans un hangar tous les systèmes réels de l'avion en cours de développement. Seule la mécanique du vol est simulée : les sorties des actionneurs sont mesurées, rentrées dans un modèle de simulation numérique, puis les mesures effectuées normalement par les capteurs sont calculées et restituées par le modèle.

L'intérêt est de diminuer de manière importante les coûts et les délais en économisant le plus grand nombre possible de tests sur le système réel en situation. Lors des premiers tests réels les techniques « Hardware In the Loop » permettent d'avoir effectué la plus grande part du développement des lois de commande.

### **8.5 Le prototypage rapide en automatique**

Un peu dans le même ordre d'idée que le « Hardware In the Loop » le prototypage rapide permet de réaliser la conception des lois de commande en simulation puis de transporter rapidement (en un simple click !) le code du contrôleur simulé vers la cible temps réel implémentée sur le système réel.

La société Mathworks [9] investit massivement dans ce domaine en ayant enrichi son logiciel de calcul scientifique (Matlab) avec un logiciel de simulation graphique très évolué (Simulink) puis en développant des outils de prototypage rapide associés (Real Time Workshop, Embedded Target For...).

## 9. ANNEXE

### 9.1 Annexe 1 : Résolution de l'équation polynomiale de Bezout

Il s'agit de trouver R et S vérifiant :

$$P(q^{-1}) = A(q^{-1}).S(q^{-1}) + B(q^{-1}).R(q^{-1})$$

Où P, A et B sont connus.

Dans le cas où P, A, B, R et S sont des nombres entiers la solution de l'algorithme est donnée par l'algorithme de division Euclidienne étendu [8].

Dans le cas où P, A, B, R et S sont des polynômes, l'algorithme est donné ci-dessous, programmé en matlab ([1]).

```
function [Rp,Sp,nrp,nsp]=bezoutd(A,B,Hs,Hr,P)
%function [Rp,Sp,nrp,nsp]=bezoutd(A,B,Hs,Hr,P)
%solves AHsSp+BHRp=P by coefficient comparison. Delay and discretization delay need to be integrated in B
%(zeros at the beginning).
%
%inputs:
%A=[a0 a1 ... aNa] ... vector of model denominator coefficients A=a0 + a1z^(-1) + a2z^(-2) +...+ aNaz^(-Na)
%B=[b0 b1 ... bNb] ... vector of model numerator coefficients B=b0 + b1z^(-1) + b2z^(-2) +...+ bNbz^(-Nb)
%Hs=[hs0 hs1 ... hsNhs] ... vector of controller denominator fixed part Hs=hs0 + hs1z^(-1) +...+ hsNhsz^(-Nhs)
%Hr=[hr0 hr1 ... hrNhr] ... vector of controller denominator fixed part Hr=hr0 + hr1z^(-1) +...+ hrNhrz^(-Nhr)
%P=[p0 p1 ... pNp] ... vector of desired polynomial coefficients P=p0 + p1z^(-1) + p2z^(-2) +...+ pNpz^(-Np)
%outputs:
%Rp=[rp0 rp1 rp2 ...] ... vector of coefficients for resulted controller numerator
%Sp=[sp0 sp1 sp2 ...] ... vector of coefficients for resulted controller denominator
%nrp ... order of Rp
%nsp ... order of Sp
%
%written by: J. Langer, I.D. Landau, H. Prochazka
%7th june 2002

PRECISION=1e-16;

D=size(A);
if D(1)>1, A=A'; end;
```

```

D=size(B);
if D(1)>1, B=B'; end;
D=size(Hs);
if D(1)>1, Hs=Hs'; end;
if D(1)==0, Hs=1; end;
D=size(Hr);
if D(1)>1, Hr=Hr'; end;
if D(1)==0, Hr=1;end;
D=size(P);
if D(1)>1, P=P'; end;

na=length(A)-1;
nb=length(B)-1;
np=length(P)-1;
nhs=length(Hs)-1;
nhr=length(Hr)-1;

if (nhs>0), Ah=real(convz(A,Hs)); else Ah=A*Hs; end; % Ah = A * Hs
nah=length(Ah) -1;
if (nhr>0), Bh=real(convz(B,Hr)); else Bh=B*Hr; end;
nbh=length(Bh) -1;
if (np>nah+nbh-1), disp('Bezout error: too many poles');end;

% increase size of P using zeros if necessary
if (np<nah+nbh-1),
%
% set remaining poles onto a circle with radius rmin
P=[P zeros(1,nah+nbh-1-np)];
rootsPdes=roots(P);
nextra=nah+nbh-1-np;
rmin=1e-16;
angle=[0:nextra-1]'/nextra*2*pi;
j=sqrt(-1);
rootsPextra=rmin*exp(j*angle);
P=poly([rootsPdes;rootsPextra]);
np=nah+nbh-1;

end;
P,

nsp=nbh-1;
nrp=nah-1;

%matrix is smaller than vector PD

```

```

if (np>nah+nbh-1),
    disp('Order of model denominator is too low! Add a polynom of higher order to Hs or Hr. ');
end;

% ns=nsp+nhs
% nr=nrp+nhr

M=[];
for j=1:nsp+1,
    V=[];
    if (j>1), V=[V ; zeros(j-1,1)]; end;% zeros in front of Ah
    V=[V ; Ah'];% Ah
    if (j<=nsp), V=[V ; zeros(nsp+1-j,1)]; end;% zeros behind Ah
    if (length(V)~=nah+nbh), disp('bezoutb: error V'); end;
    M=[M V]; % add one column to M
end;

for j=1:nrp+1,
    V=[];
    if (j>1), V=[V ; zeros(j-1,1)]; end;
    V=[V ; Bh'];
    if (j<=nrp), V=[V ; zeros(nrp+1-j,1)]; end;
    if (length(V)~=nah+nbh), disp('bezoutb: error V'); end;
    M=[M V];
end;

D=size(M);
if (D(1)~=nah+nbh), disp('bezoutb: error size M row'); end;
if (D(2)~=nah+nbh), disp('bezoutb: error size M column'); end;

% make P column vector
P=P';

global M1;
M1=M;

X= M\P;
% coefficients are real values
X=real(X);

% make X row vector
X=X';

```

## Commande des systèmes par ordinateur

```
Sp=X(1:nsp+1);  
Rp=X(nsp+2:nsp+nrp+2);
```



## 10. BIBLIOGRAPHIE

- [1] I. D. Landau, "*Commande des systèmes*", Edition Lavoisier 2002.
- [2] C. Nouals, "Théorie du Signal", Notes de cours ENSICA
- [3] F. Vincent, "*Traitement Numérique du Signal*", Notes de cours ENSICA.
- [4] D. Arzelier, "*Représentation et analyse des systèmes, notes de cours*", Edition ENSICA. Voir aussi <http://www.laas.fr/~arzelier/>
- [5] J. Bordeneuve-Guibé, "*Commande des systèmes linéaires*", Notes de cours ENSICA.
- [6] Y. Briere, "*Identification*", Notes de cours ENSICA.
- [7] L. Ljung, "*System Identification, Theory for the User*", Prentice Hall 1987.
- [8] Wikipedia , équation de Bezout : [http://en.wikipedia.org/wiki/Bézout's\\_identity](http://en.wikipedia.org/wiki/Bézout's_identity)
- [9] <http://www.mathworks.com>



# 11. INDEX

<b>A</b>	
actionneur intelligent .....	21
amortissement .....	15
ANSI C .....	60
Approximation numérique d'équation différentielle .....	28
ARMAX .....	35
<b>B</b>	
BJ35 .....	
Blocage d'un signal .....	56
bloqueurs d'ordre 0 .....	22
BOZ .....	22
bus.....	21, 61
<b>C</b>	
CAN.....	21
capteur intelligent .....	21
causal .....	6, 9
Causalité .....	10
CNA.....	21
compilateur .....	60
convertisseur analogique / numérique.....	21
convertisseur numérique / analogique.....	21
convertisseurs analogique-numérique .....	61
coprocesseur mathématique .....	60
Correspondances pôles - zéros.....	31
<b>D</b>	
Diagramme de Bode .....	18
Discrétisation de fonction de transfert continues .....	27
division polynomiale.....	6
<b>E</b>	
échelon unitaire.....	11
équation d'état .....	17
Erreur statique.....	42
Estimation de la réponse fréquentielle .....	39
Estimation de la réponse impulsionnelle.....	38
Euler .....	29
<b>F</b>	
facteur d'échelle .....	61
facteur d'oubli .....	36
<i>fenêtre de Hamming</i> .....	39
filtre anti-repliement .....	25
fonction de sensibilité .....	42
fonction de transfert discrète.....	9
<b>G</b>	
gain statique.....	11
gain statique unitaire.....	42
<b>I</b>	
Identification.....	32
Identification non paramétrique .....	38
intercorrélation.....	38
<b>L</b>	
lieu des racines.....	43
lieu d'Evans.....	43
LTI.....	8
<b>M</b>	
marge de gain.....	47
marge de module.....	46
marge de phase .....	47
méthode des moindres carrés récursifs .....	32
Méthode des trapèzes.....	29
modèle ARX .....	32
<b>O</b>	
OE.....	35
one step ahead prediction.....	35
opérateur de retard .....	9
<b>P</b>	
période d'échantillonnage (choix) .....	26
Placement de pôles .....	54
Pôles .....	11
Poursuite .....	56
<i>prédiction</i> .....	32
protocoles.....	61
Pseudo Random Binary Signal .....	34
pulsation .....	15
<b>Q</b>	
quantification .....	23, 24, 60
<b>R</b>	
R,S,T.....	41
rapidité.....	15
rayon de convergence .....	5, 9
Régulation.....	55

Rejet de perturbations .....	42
Rejet d'une perturbation harmonique .....	56
réponse à un échelon.....	11
réponse fréquentielle.....	18
Réponse impulsionnelle.....	10
Réponses indicielle .....	10
retard.....	23
root locus .....	43

### S

signal discret .....	5
smart sensors.....	62
stabilité.....	11
Systèmes du premier ordre.....	14
Systèmes du second ordre.....	15

### T

Théorème de Nyquist.....	46
théorème de Shannon.....	19, 24
théorème du retard .....	6
Transformée en z .....	5
transformée unilatérale .....	5
Tustin.....	29

### V

virgule flottante.....	60
------------------------	----

### Z

zero holder hold .....	23
zéros.....	11
zoh .....	23

# Commande des systèmes par ordinateur

Yves Briere  
ENSICA

Notes de cours  
2007



# TABLE DES MATIERES

Table des matières .....	1
1. Introduction .....	1
2. Rappels et prérequis .....	3
3. Signaux et systèmes discrets .....	5
3.1 Signaux discrets .....	5
3.1.1 Transformée en z .....	5
3.1.2 Signaux types .....	7
3.2 Systèmes discrets .....	9
3.2.1 Fonction de transfert discrète, définition .....	9
3.2.2 Notation "opérateur de retard $q^{-1}$ " .....	10
3.2.3 Réponse impulsionnelle .....	11
3.2.4 Causalité .....	11
3.2.5 Réponses indicielle .....	11
3.2.6 Régime stationnaire (gain statique) .....	12
3.2.7 Pôles, zéros, stabilité .....	12
3.2.8 Systèmes du premier ordre .....	15
3.2.9 Systèmes du second ordre .....	16
3.2.10 Calcul itératif de la réponse d'un système .....	18
3.2.11 Représentation par équation d'état .....	18
3.3 Réponse fréquentielle des systèmes échantillonnés .....	19
3.3.1 Diagramme de Bode d'un système échantillonné .....	19
3.3.2 Où l'on voit apparaître le théorème de Shannon .....	20
4. Systèmes continus échantillonnés .....	23
4.1 Structure des systèmes commandés par ordinateur .....	23
4.2 Échantillonnage des systèmes continus .....	24
4.3 Effets de l'échantillonnage .....	25
4.4 Où l'on revoit le théorème de Shannon .....	26
4.5 Discrétisation de fonction de transfert continues .....	29
4.5.1 Approximation numérique d'équation différentielle .....	30
4.5.2 Superposition des réponses indicielles .....	32
4.5.3 Correspondances pôles - zéros .....	33
4.5.4 Bilan .....	33
5. Identification des systèmes échantillonnés .....	35
5.1 Identification d'un modèle ARX par la méthode des moindres carrés récurrents .....	35
5.2 L'identification paramétrique en général et en pratique .....	37

5.2.1	Choix du signal d'excitation, traitement des mesures .....	37
5.2.2	Choix d'une structure de modèle.....	38
5.2.3	Choix d'un critère à minimiser.....	38
5.2.4	Choix d'un algorithme.....	38
5.2.5	Validation.....	39
5.2.6	Utilisation de logiciels .....	39
5.3	Identification "simple" .....	39
5.4	Identification non paramétrique .....	41
5.4.1	Estimation de la réponse impulsionnelle .....	41
5.4.2	Estimation de la réponse fréquentielle.....	42
6.	Systèmes bouclés .....	45
6.1	Fonction de transfert en boucle fermée .....	45
6.2	Erreur statique .....	46
6.3	Rejet de perturbations.....	46
6.4	Stabilité des systèmes à simple gain de rétroaction.....	47
6.5	Analyse des systèmes bouclés dans le domaine fréquentiel : stabilité.....	49
6.5.1	Théorème de Nyquist dans le cas où le système est stable en boucle ouverte (critère du revers) .....	50
6.5.2	Théorème de Nyquist dans le cas où le système est instable en boucle ouverte .....	50
6.5.3	Interprétations du lieu de Nyquist.....	50
7.	Méthodes de régulation des systèmes échantillonnés .....	53
7.1	Introduction .....	53
7.2	Structure d'un système commandé par ordinateur, hypothèses.....	54
7.2.1	Discretisation, choix de la période d'échantillonnage.....	54
7.2.2	Discretisation d'un correcteur continu .....	55
7.3	Régulateur PID numérique .....	55
7.3.1	Cas 1 .....	55
7.3.2	Cas 2 .....	58
7.4	Placement de pôles .....	59
7.4.1	Choix des pôles en boucle fermée .....	60
7.4.2	Régulation : calcul de R et S.....	60
7.4.3	Poursuite : calcul de T .....	61
7.5	Poursuite et régulation à objectifs indépendants .....	62
8.	Technologie de la commande par ordinateur.....	65
8.1	Calculateurs et interfaces numériques / analogiques.....	65
8.2	Transport de l'information : les bus et les protocoles .....	67
8.3	Capteurs et actionneurs "intelligents", "smart sensors" .....	68
8.4	La philosophie "hardware in the loop" .....	68
8.5	Le prototypage rapide en automatique .....	68
9.	Annexe .....	69
9.1	Annexe 1 : Résolution de l'équation polynomiale de Bezout.....	69
10.	Bibliographie.....	73
11.	Index.....	1







# 1. INTRODUCTION



Figure 1 : commande d'un systèmes de télémédecine par calculateur, vu par Claude Serre



## 2. RAPPELS ET PREREQUIS

Ce polycopié ne comporte pas de rappels sur les systèmes continus. S'adressant aux étudiants en deuxième année de l'ENSICA il suppose connu les enseignements de première année et de seconde année en automatique et traitement du signal.

En première année le cours de théorie du signal [2] a rendu familier les notions de :

- transformée de Fourier, transformée de Laplace
- corrélation, convolution

Le cours de traitement numérique du signal [3] a abordé en plus :

- la discrétisation et la quantification des signaux.
- la transformée en z et la transformée de Fourier discrète.
- les filtres numériques RIF et RII.

En automatique [4] les bases de la représentation et de l'analyse des systèmes continus ont été posées. On ne reviendra pas dans ce cours sur les notions de :

- modèles mathématique des systèmes linéaires à temps continus à temps invariant (LTI) : fonction de transfert, équation d'état, pôles, zéros et dynamique.
- commandabilité et observabilité.
- stabilité des systèmes en boucle ouverte : pôles, critères de Routh-Hurwitz.
- stabilité des systèmes bouclés : critère de Nyquist, critère de Bode.
- lieu des racines.

En deuxième année les notions suivantes sont acquises [5] :

- synthèse des correcteurs par méthodes fréquentielles (avance et retard de phase, PID)
- commande par retour d'état (placement de pôles), commande Linéaire Quadratique (LQG)
- reconstruction de l'état par Observateur, observateur optimal de Kalman.



## 3. SIGNAUX ET SYSTEMES DISCRETS

### 3.1 Signaux discrets

#### 3.1.1 Transformée en z

Pour une approche mathématiquement rigoureuse de représentation des signaux discrets par la transformée en z consulter [3] et sa bibliographie. Dans ce cours on se cantonne à une approche utilitariste des outils et méthodes mathématiques tout en essayant de ne pas contredire fondamentalement la rigueur mathématique.

Un signal discret  $s(k)$  est une séquence de nombres réels appelés échantillons. L'indice  $k$  est généralement associé à un instant d'échantillonnage :  $t = k.T_e$ . La période d'échantillonnage  $T_e$  est supposée constante. Par définition la transformée en z du signal  $s(k)$  est la fonction de la variable  $z$ , notée abusivement  $s(z)$  :

$$s(z) = Z(s(k)) = \sum_{k=0}^{\infty} s(k) \cdot z^{-k} \quad (1)$$

La transformée définie ici est plus précisément appelée transformée unilatérale puisque les termes correspondant à  $k < 0$  ne sont pas pris en compte. Ceci n'a pas d'importance puisque les signaux que nous utilisons sont systématiquement causaux<sup>1</sup> :  $s(k) = 0$  pour  $k < 0$ .

Il y a bien évidemment un lien avec la transformée de Laplace et la transformée de Fourier (voir [3])

La fonction de la variable complexe  $s(z)$  est une série et ne converge pas forcément pour toute valeur de  $z$ . En toute rigueur il faudrait associer le rayon de convergence à chaque fonction en  $z$  :  $\{s(z) \text{ converge pour } |z| > \alpha\}$ .

Les automaticiens pour qui la transformée en z n'est bien souvent qu'une façon bien commode de regrouper des termes de récurrence utilisent souvent la pseudo-variable<sup>2</sup>  $q$ .

A partir d'une transformée en z  $s(z)$  il est possible de retrouver la séquence  $s(k)$  correspondante en appliquant la transformation inverse, dont on ne donnera pas la formule ici. A la place de faire appel à cette formule compliquée il est possible d'utiliser d'autres méthodes pour retrouver la séquence  $s(k)$  lorsque la fonction  $s(z)$  est donnée sous forme de fraction rationnelle :

<sup>1</sup> Pour une justification du terme « causal » voir page 11

<sup>2</sup> Voir page 10

- division polynomiale
- décomposition de la fraction rationnelle en éléments simples puis utilisation de transformations élémentaires telles que celles données dans le Tableau 2

La propriété fondamentale de la transformée en z est qu'elle est linéaire (la transformée en z d'une composition linéaire de signaux est la composition linéaire de la transformée en z de chaque signal). Les autres propriétés fondamentales sont données dans le Tableau 2.

Tableau 1 : propriétés de la transformée en z

1	$s(z) = Z(s(k)) = \sum_{k=0}^{\infty} s(k) \cdot z^{-k}$
2	$Z(\alpha \cdot s_1(k) + \beta \cdot s_2(k)) = \alpha \cdot Z(s_1(k)) + \beta \cdot Z(s_2(k))$
3	$Z(k \cdot s(k)) = -z \cdot \frac{ds(z)}{dz}$
4	$Z(c^k \cdot s(k)) = s\left(\frac{z}{c}\right)$
5	$Z(s(k-1)) = s(k=-1) + z^{-1} \cdot s(z)$
6	$Z(s(k+1)) = z \cdot s(z) - z \cdot s(k=0)$
7	$s(k=0) = \lim_{z \rightarrow \infty} (s(z))$
8	$s(k \rightarrow \infty) = \lim_{z \rightarrow 1} \left( \frac{z-1}{z} \cdot s(z) \right)$

Remarque 1 : la propriété 5 ne prend pas en compte le fait que le signal est causal : en pratique  $s(k=-1) = 0$ .

Remarque 2 : la propriété 5 justifie le fait que le terme  $z^{-1}$  soit appelé "opérateur de retard unitaire", cette propriété est aussi appelée "théorème du retard".

Remarque 3 : les propriétés 7 et 8 sont données sous réserve de convergence.

Transformées inverses

### Exemple 1 : transformée inverse par division polynomiale

On considère un signal  $s(k)$  dont la transformée en z est donnée par la fraction rationnelle :

$$S(z) = \frac{0.1 \cdot z^2 + 0.1 \cdot z}{z^3 - 2.6 \cdot z^2 + 2.2 \cdot z - 0.6} \quad (3.1)$$

L'objectif est d'obtenir  $S(z)$  sous forme polynomiale :



$$S(z) = x_0 + x_1 \cdot z^{-1} + x_2 \cdot z^{-2} + \dots \text{etc.} \tag{3.2}$$

Cette suite est généralement infinie, la réponse temporelle est donnée par la succession des termes  $\{x_i\}$ . En traitant l'équation (3.1) comme une simple division l'algorithme de division classique (celui appris en CM2) permet d'obtenir par récurrence tous les termes de la suite (3.2) :

$0.1 \cdot z^2$	$+ 0.1 \cdot z$				$z^3 - 2.6 \cdot z^2 + 2.2 \cdot z - 0.6$ $0.1 \cdot z^{-1} + 0.36 \cdot z^{-2} + 0.716 \cdot z^{-3}$	
$0.1 \cdot z^2$	$- 0.26 \cdot z$	$+ 0.22$	$- 0.06 \cdot z^{-1}$			
$0$	$0.36 \cdot z$	$- 0.22$	$0.06 \cdot z^{-1}$			
	$0.36 \cdot z$	$- 0.936$	$0.792 \cdot z^{-1}$	$- 0.216 \cdot z^{-2}$		
$0$	$0.716$	$- 0.732 \cdot z^{-1}$	$\dots$	$\dots$		

Les éléments successifs de la suite  $s(k)$  sont donc  $s(k) = \{0, 0.1, 0.36, 0.716, \dots\}$ . Pour obtenir les termes suivants il faudrait continuer la division.

**Exemple 2 : transformée inverse par décomposition en éléments simples**

La deuxième méthode présentée consiste à décomposer la fraction rationnelle (3.1) en éléments simples puis d'utiliser les tables de transformées en  $z$  (Tableau 2). Pour notre exemple cela donne :

$$S(z) = \frac{0.1 \cdot z^2 + 0.1 \cdot z}{z^3 - 2.6 \cdot z^2 + 2.2 \cdot z - 0.6} = \frac{-z}{z-1} + \frac{0.5 \cdot z}{(z-1)^2} + \frac{z}{z-0.6}$$

La réponse temporelle est donc composée des trois éléments :

$$s(k) = -(1 \ 1 \ 1 \ \dots) + 0.5 \cdot (1 \ 2 \ 3 \ \dots) + (1 \ 0.6 \ 0.6^2 \ \dots)$$

$$\text{soit : } s(k) = \begin{cases} s(0) = -1 + 0.5 + 1 \\ s(2) = -1 + 0.5 \cdot 2 + 0.6 \cdot 0.5 \\ s(3) = -1 + 0.5 \cdot 3 + 0.6^2 \\ s(4) = \dots \end{cases}$$

**3.1.2 Signaux types**

Le Tableau 2 donne les transformée en  $z$  de certains signaux fondamentaux. La Figure 2 représente l'allure de quelques signaux types.

Tableau 2 : transformée en z élémentaires

$s(k)$	$S(z)$
signal temporel	Transformée en z
$\delta(k)$ : impulsion unité	1
$u(k)$ : échelon unité	$\frac{z}{z-1}$
$k \cdot u(k)$	$\frac{z}{(z-1)^2}$
$c^k \cdot u(k)$	$\frac{z}{z-c}$
$k \cdot a^k \cdot u(k)$	$\frac{a \cdot z}{(z-a)^2}$
$\sin(\omega \cdot k) \cdot u(k)$	$\frac{z \cdot \sin(\omega)}{z^2 - 2 \cdot z \cdot \sin(\omega) + 1}$
$\cos(\omega \cdot k) \cdot u(k)$	$\frac{z \cdot (z - \cos(\omega))}{z^2 - 2 \cdot z \cdot \sin(\omega) + 1}$
$a^k \cdot \sin(\omega \cdot k) \cdot u(k)$	$\frac{a \cdot z \cdot \sin(\omega)}{z^2 - 2 \cdot a \cdot \cos(\omega) \cdot z + a^2}$
$a^k \cdot \cos(\omega \cdot k) \cdot u(k)$	$\frac{z \cdot (z - a \cdot \cos(\omega))}{z^2 - 2 \cdot a \cdot \cos(\omega) \cdot z + a^2}$

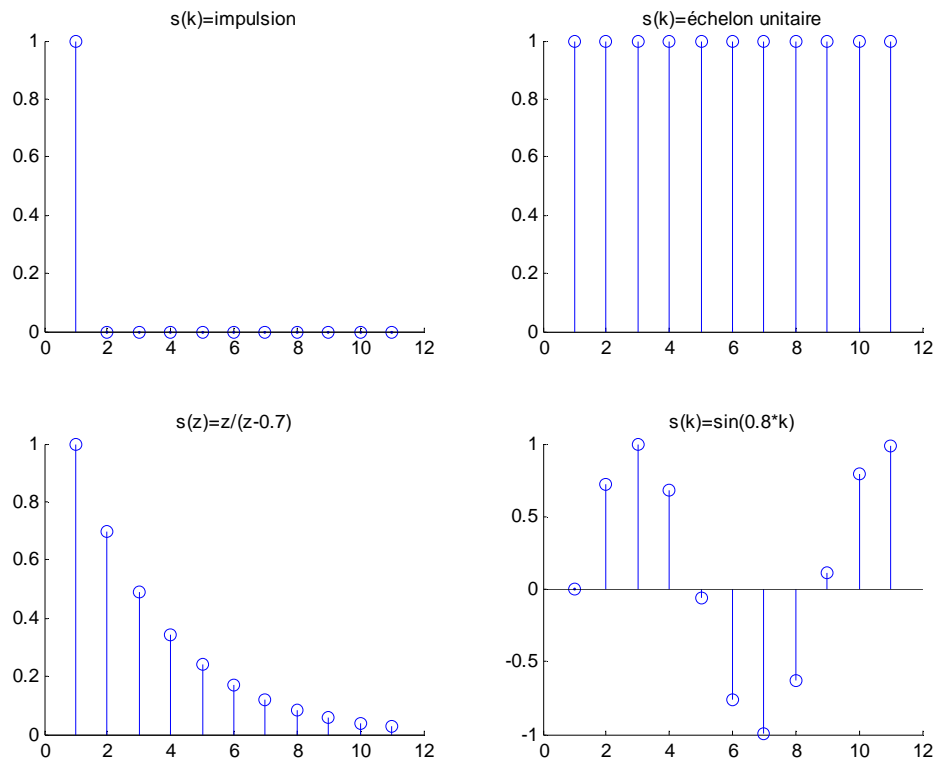


Figure 2 : réponses de quelques signaux types

### 3.2 Systèmes discrets

#### 3.2.1 Fonction de transfert discrète, définition

On considère maintenant un système discret linéaire et invariant dans le temps (LTI) représenté sur la Figure 3.

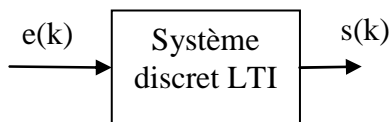


Figure 3 : système discret, notations

La relation entre la séquence d'entrée et la séquence de sortie est linéaire et définie par un ensemble de coefficients constants  $a_{1:n}$  et  $b_{1:m}$  :

$$\begin{aligned} s(k) + a_1 \cdot s(k-1) + a_2 \cdot s(k-2) + \dots + a_n \cdot s(k-n) = \\ b_0 \cdot e(k) + b_1 \cdot e(k-1) + b_2 \cdot e(k-2) + \dots + b_m \cdot e(k-m) \end{aligned} \quad (2)$$

On peut appliquer la transformée en  $z$  à chaque membre de l'équation puis en appliquant en récurrence la propriété 5 (théorème du retard) on obtient la fonction de transfert discrète du système :

$$\frac{s(z)}{e(z)} = F(z) = \frac{b_0 + b_1 \cdot z^{-1} + b_2 \cdot z^{-2} + \dots + b_m \cdot z^{-m}}{1 + a_1 \cdot z^{-1} + a_2 \cdot z^{-2} + \dots + a_n \cdot z^{-n}} \quad (3)$$

Remarque 1 : la fonction de transfert précédente est normalisée en prenant  $a_0 = 1$ .

Remarque 2 : on remarquera que la fonction de transfert est donnée en  $z^{-1}$  plutôt qu'en  $z$ . Cela est justifié par le fait que celle-ci traduit l'équation d'évolution du système physique causal (2).

Le script Matlab permettant de créer un objet Matlab correspondant à une fonction de transfert discrète est donné ci-dessous.

#### Matlab

```
% Création d'un système linéaire discret :
Te=0.1;
sysd = tf([b0 b1 b2],[1 a0 a2 a3],Te,'variable','z^-1');
```

On notera l'utilisation de l'argument 'variable', 'z^-1'.

### 3.2.2 Notation "opérateur de retard $q^{-1}$ "

La fonction de transfert est une fonction de la variable complexe  $z$ . Il faudrait donc en toute rigueur l'associer à son rayon de convergence. En pratique on ne s'embarrasse jamais de préciser cette zone de convergence. C'est pourquoi on trouve souvent la notation utilisant l'opérateur  $q^{-1}$  de décalage à la place de la variable complexe  $z^{-1}$  (4).  $F(q)$  n'est alors plus à proprement parler une fonction mais on lui appliquera les mêmes manipulations algébriques qu'à  $F(z)$ .

$$F(q) = \frac{b_0 + b_1 \cdot q^{-1} + b_2 \cdot q^{-2} + \dots + b_m \cdot q^{-m}}{1 + a_1 \cdot q^{-1} + a_2 \cdot q^{-2} + \dots + a_n \cdot q^{-n}} \quad (4)$$

Attention : selon les auteurs l'opérateur qui remplace  $z^{-1}$  est noté  $q$  (Matlab) ou  $q^{-1}$  (Landau [1]). Dans la suite du document c'est cette dernière convention qui sera appliquée.

L'utilisation de l'opérateur de retard  $q^{-1}$  à la place de la variable complexe  $z^{-1}$  permet de traiter du cas des fonctions de transfert à coefficients non constant, cas non prévu par la transformation en  $z$ .

### 3.2.3 Réponse impulsionnelle

Essayons d'appliquer au système un signal d'entrée particulier : une impulsion unitaire. La transformée en  $z$  du système est obtenue en multipliant la fonction de transfert par la transformée en  $z$  de l'impulsion unitaire : 1. On obtient :

$$s(z) = \frac{b_0 + b_1 \cdot z^{-1} + b_2 \cdot z^{-2} + \dots + b_m \cdot z^{-m}}{1 + a_1 \cdot z^{-1} + a_2 \cdot z^{-2} + \dots + a_n \cdot z^{-n}} \times 1 \quad (5)$$

Cela justifie le fait que la fonction de transfert du système peut aussi s'appeler "réponse impulsionnelle".

#### Matlab

```
% Réponse impulsionnelle d'un système :
[y,t] =impulse(sysd);
```

### 3.2.4 Causalité

Un système est dit causal si les sorties ne dépendent que du passé et non pas du futur. La conséquence est que la réponse impulsionnelle de  $F(z)$  ne peut pas comporter de terme non nuls pour  $k < 0$ . Cela justifie la confusion sémantique entre "système causal" et "signal causal", c'est-à-dire ne comportant pas de termes non nuls pour  $k < 0$ .

Attention : à la différence des systèmes continus un système physique (causal) discret peut très bien être modélisé avec un degré  $m$  supérieur au degré  $n$  : le système peut avoir plus de zéros que de pôles.

La transformée en  $z$  représentant des signaux échantillonnés elle peut tout aussi bien représenter les mesures prises à différents endroits d'une poutre. Le temps étant remplacé ici par une abscisse la notion de causalité ne s'applique pas ! Attention : beaucoup des propriétés de systèmes échantillonnés que l'on étudie sont liés à la causalité, prudence donc si l'on applique ces méthodes à d'autres domaines !

### 3.2.5 Réponses indicielle

Pour caractériser des systèmes on s'intéresse souvent à leur réponse impulsionnelle ou, plus souvent, à leur réponse indicielle. Pour calculer la réponse

à un échelon d'un système il faut donc multiplier la fonction de transfert par la transformée en z d'un échelon unitaire :

$$s(z) = \frac{b_0 + b_1 \cdot z^{-1} + b_2 \cdot z^{-2} + \dots + b_m \cdot z^{-m}}{1 + a_1 \cdot z^{-1} + a_2 \cdot z^{-2} + \dots + a_n \cdot z^{-n}} \times \frac{1}{1 - z^{-1}} \quad (6)$$

Il faudrait ensuite calculer la transformée en z inverse de s(z). En pratique, si l'on doit faire le calcul à la main on décompose en éléments simples puis on utilise des tables. Avec Matlab c'est plus simple. Voici par exemple le script pour obtenir les 20 premiers échantillons de la réponse indicielle d'un système :

**Matlab**

```
% Réponse indicielle d'un système :
[y,t] =step(sysd,20);
```

### 3.2.6 Régime stationnaire (gain statique)

Le gain statique est obtenu en appliquant une entrée indicielle et en calculant la valeur du régime permanent. En appliquant le théorème de la valeur finale il vient :

$$F_0 = \lim_{z \rightarrow 1} \left( \frac{z-1}{z} \cdot \frac{b_0 + b_1 \cdot z^{-1} + b_2 \cdot z^{-2} + \dots + b_m \cdot z^{-m}}{1 + a_1 \cdot z^{-1} + a_2 \cdot z^{-2} + \dots + a_n \cdot z^{-n}} \times \frac{1}{1 - z^{-1}} \right) \quad (7)$$

Soit :

$$F_0 = \frac{\sum_{i=0}^{i=m} b_i}{1 + \sum_{i=1}^{i=n} a_i} \quad (8)$$

Le gain statique est obtenu donc simplement en faisant le quotient de la somme des coefficients.

### 3.2.7 Pôles, zéros, stabilité

Comme pour les systèmes continus la dynamique et la stabilité des systèmes dépend des propriétés du dénominateur de la fonction de transfert.

$$F(z) = \frac{e_0 \cdot (1 + e_1 \cdot z^{-1}) \cdot (1 + e_2 \cdot z^{-1}) \cdots (1 + e_m \cdot z^{-1})}{(1 + c_1 \cdot z^{-1}) \cdot (1 + c_2 \cdot z^{-1}) \cdots (1 + c_n \cdot z^{-1})} \quad (9)$$

Les coefficients  $c_i$  sont les pôles de la fonction de transfert, les coefficients  $e_i$  sont les zéros. La fonction de transfert étant au départ composée de coefficients purement réels les coefficients  $c_i$  sont soit réels soit imaginaires par paires conjuguées.

Une fonction de transfert est donc aussi bien définie par la donnée de ses pôles et de ses zéros. Le script Matlab pour créer un système ainsi est donné ci-dessous :

#### Matlab

```
% Système discret défini par ses pôles et zéros :
Te=0.1;
sysd=zpk([c1 c2 c3],[e1 e2],e0,Te);
```

Si l'on décompose la fonction de transfert en éléments simples on obtient une somme d'éléments du type :

$$F(z) = \frac{d_1}{1 + c_1 \cdot z^{-1}} + \frac{d_2}{1 + c_2 \cdot z^{-1}} + \cdots + \frac{d_n}{1 + c_n \cdot z^{-1}} \quad (10)$$

Un système linéaire temps invariant est dit stable si sa réponse impulsionnelle tend asymptotiquement vers 0. Compte tenu de (10) cela nécessite que chaque pôle  $c_i$  soit de module inférieur à 1.

**Propriété** : un système discret LTI est stable si et seulement si tous ses pôles ont leur modules inférieur ou égal à 1.

#### Critère algébrique attestant de la stabilité : critère de Routh

On rappelle que le critère de Routh permet de prédire si les racines d'un polynôme sont toutes à partie réelle négative, donc stables. Énonçons le théorème de Routh dans le cas d'un polynôme  $P'(w)$  :

$$P'(w) = a_0 + a_1 \cdot w + a_2 \cdot w^2 + \cdots + a_n \cdot w^n \quad (11)$$

Dans un premier temps on construit le tableau :

$$\begin{array}{cccccc|c}
 \beta_{n,0} & \beta_{n,1} & \beta_{n,2} & \beta_{n,q-1} & \beta_{n,q} & 0 & \\
 \beta_{n-1,0} & \beta_{n-1,1} & \beta_{n-1,2} & \beta_{n-1,q-1} & \beta_{n-1,q-1} & 0 & \\
 \beta_{n-2,0} & \beta_{n-2,1} & \beta_{n-2,2} & \beta_{n-2,q-1} & \beta_{n-2,q-1} & & \\
 \\ 
 \beta_{2,0} & \beta_{2,1} & 0 & & & & \\
 \beta_{1,0} & 0 & & & & & \\
 \beta_{0,0} & 0 & & & & & 
 \end{array} \quad (12)$$

Puis on vérifie que tous les premiers éléments de la première colonne ont le même signe : cela garantit que tous les pôles de  $P'(w)$  ont leur partie réelle négative.

Les éléments du tableau sont construits par récurrence ligne par ligne en appliquant les formules suivantes :

Pour  $i = n$  et  $i = n - 1$  :

$$\begin{cases} \beta_{n,j} = a_{n-2j} \\ \beta_{n-1,j} = a_{n-1-2j} \end{cases} \quad (13)$$

Et pour  $i < n-1$  :

$$\beta_{i,j} = \frac{\beta_{i+1,0} \cdot \beta_{i+2,j+1} - \beta_{i+1,j+1} \cdot \beta_{i+2,0}}{\beta_{i+1,0}} \quad (14)$$

Dans le cas qui nous intéresse, un système échantillonné, on cherche plutôt à vérifier que les pôles du polynôme  $P(z)$  sont de module inférieur à un. L'idée est d'effectuer un changement de variable  $z \rightarrow w$  pour pouvoir appliquer le critère de Routh.

$$P'(w) = (1-w)^n \cdot P\left(\frac{1+w}{1-w}\right) \quad (15)$$

Si l'on effectue la transformation (15) alors le fait que  $P'(w)$  ait toutes ses racines à partie réelle négative garantit que  $P(z)$  a toutes ses racines dans le cercle unité. Pour vérifier la stabilité de  $P(z)$  il suffit donc d'effectuer le changement de variable (15) puis d'appliquer le critère de Routh conventionnel (12 13 14).

### Critère algébrique attestant de la stabilité : critère de Jury

Le critère de Jury s'apparente au critère de Routh. Nous le donnons pour un polynôme de degré 2, 3 et 4 du type :



$$P(z) = a_0 + a_1 \cdot z + a_2 \cdot z^2 + \dots + a_n \cdot z^n \quad (16)$$

Les conditions de stabilité sont données ci-dessous (ces conditions sont données dans le cas  $a_n > 0$ , il suffit de multiplier tous les termes par  $-1$  si cette condition n'est pas remplie par  $P(z)$ ) :

$$\begin{aligned}
 n = 2 & \begin{cases} a_0 + a_1 + a_2 > 0 \\ a_0 - a_1 + a_2 > 0 \\ -a_0 + a_2 > 0 \end{cases} \\
 n = 3 & \begin{cases} a_0 + a_1 + a_2 + a_3 > 0 \\ -a_0 + a_1 - a_2 + a_3 > 0 \\ -|a_0| + a_3 > 0 \\ a_0 a_2 - a_1 a_3 - a_0^2 + a_3^2 > 0 \end{cases} \\
 n = 4 & \begin{cases} a_0 + a_1 + a_2 + a_3 + a_4 > 0 \\ a_0 - a_1 + a_2 - a_3 + a_4 > 0 \\ a_4^2 - a_0^2 - |a_0 a_3 - a_1 a_4| > 0 \\ (a_0 - a_4)^2 (a_0 - a_2 + a_4) + (a_1 - a_3)(a_0 a_3 - a_1 a_4) > 0 \end{cases}
 \end{aligned} \quad (17)$$

### 3.2.8 Systèmes du premier ordre

On donne dans la Figure 4 les réponses indicielles de systèmes du premier ordre ainsi que la localisation des pôles dans le plan complexe. Le cercle de rayon unité représente la zone dans laquelle les pôles doivent se situer pour assurer la stabilité. On remarque sur les deux premières figures l'influence de la place du zéro par rapport au point  $(1,0)$  pour la rapidité.

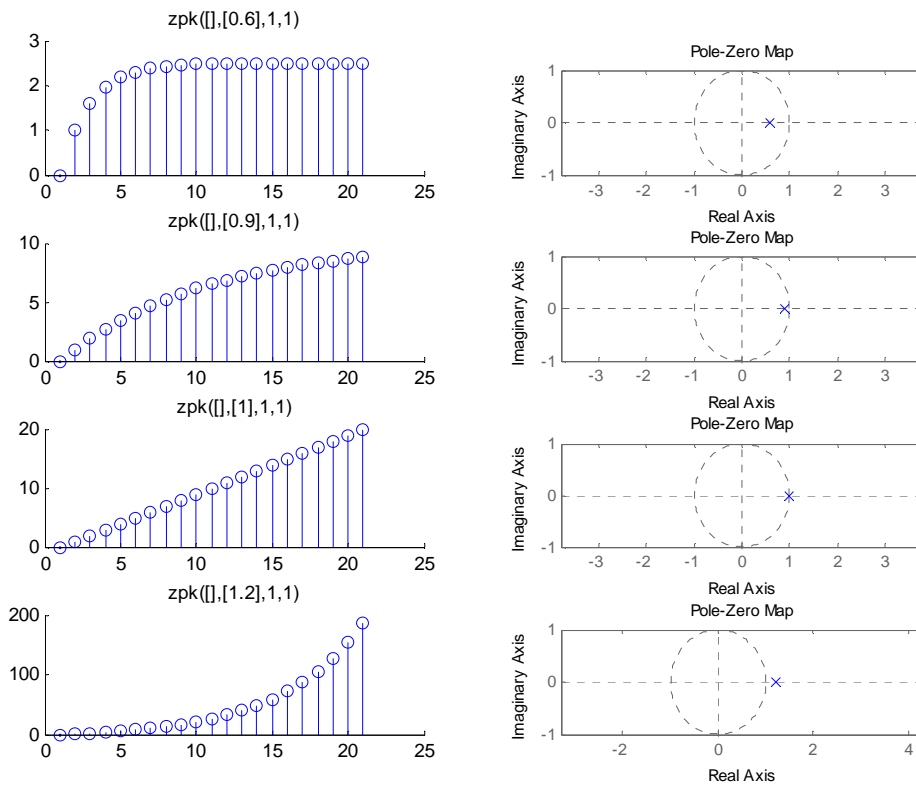


Figure 4 réponses indicielles de systèmes du premier ordre

### 3.2.9 Systèmes du second ordre

On donne dans la Figure 6 les réponses indicielles de systèmes du second ordre ainsi que la localisation des pôles dans le plan complexe. La rapidité et l'amortissement dépend de la position des pôles dans le cercle unité. La Figure 5 représente les courbes iso-amortissement et iso-pulsation des systèmes du second ordre.

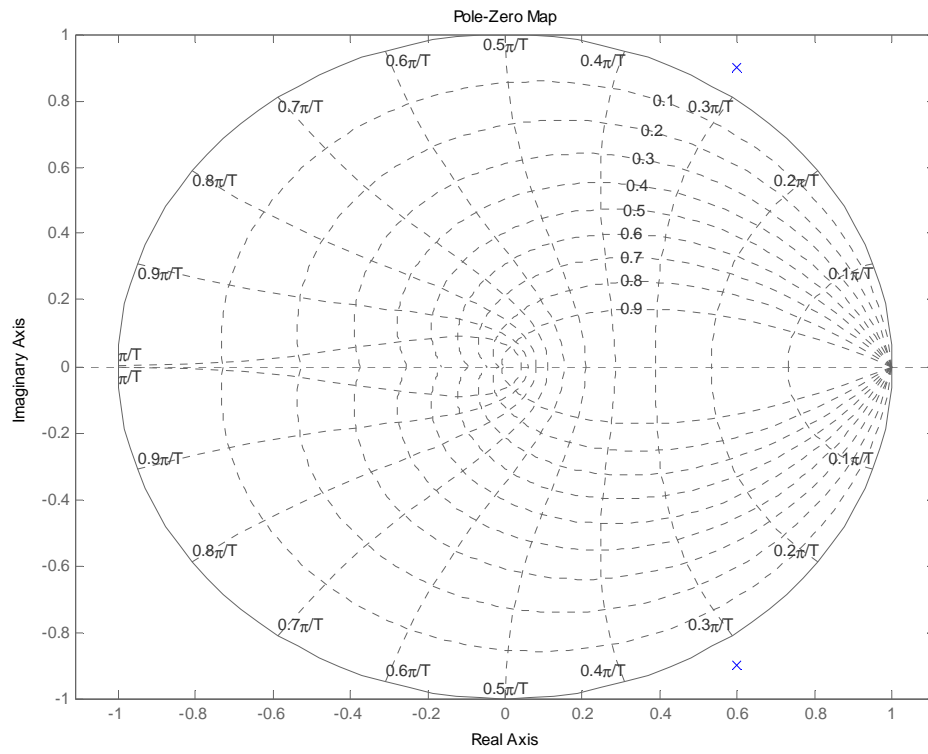


Figure 5 : courbes iso-amortissement et iso-pulsation des systèmes discrets du second ordre

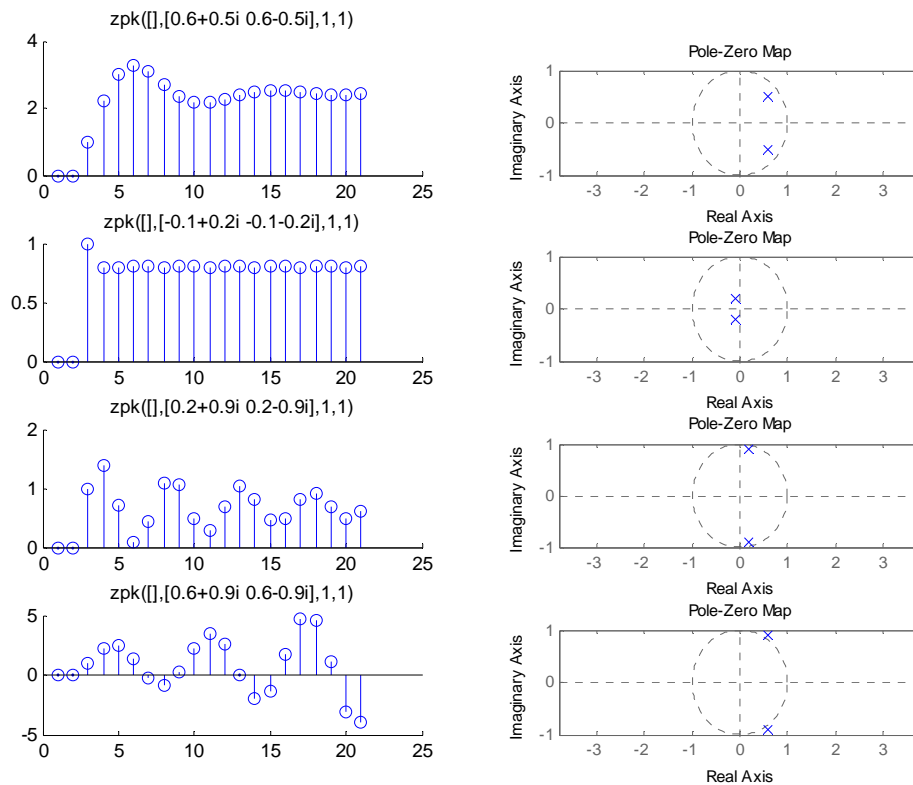


Figure 6 : réponses de systèmes du second ordre

### 3.2.10 Calcul itératif de la réponse d'un système

Si l'on dispose de la fonction de transfert d'un système et que l'on connaît les valeurs successives  $e(k)$  du signal d'entrée il est facile de calculer itérativement la sortie  $y(k)$  du système par l'équation de récurrence (2). Il faut initialiser par les  $n$  valeurs initiales de  $s(k)$ , prises généralement égales à 0.

### 3.2.11 Représentation par équation d'état

Un système discret peut aussi être représenté par un ensemble d'équations d'état comme dans le domaine continu (52). Cette représentation est naturellement adaptée aux systèmes multi-entrées multi-sorties. On définit alors un état interne  $x(k)$  ainsi qu'une fonction d'évolution  $f$  et une équation de mesure  $h$ .

$$\begin{cases} x(k+1) = f(x(k), u(k)) \\ s(k) = h(x(k), u(k)) \end{cases} \quad (18)$$

### 3.3 Réponse fréquentielle des systèmes échantillonnés

#### 3.3.1 Diagramme de Bode d'un système échantillonné

Lorsqu'on étudie la réponse fréquentielle des systèmes continus on considère des entrées particulières sinusoïdales :

$$\begin{cases} e(t) = \cos(2 \cdot \pi \cdot f \cdot t) + j \cdot \sin(2 \cdot \pi \cdot f \cdot t) = e^{j \cdot 2 \cdot \pi \cdot f \cdot t} \\ f \in [0 \quad \infty] \end{cases} \quad (19)$$

De même, pour des systèmes échantillonnés on considèrera des entrées sinusoïdales échantillonnées :

$$\begin{cases} e(k) = \cos(2 \cdot \pi \cdot f \cdot k) + j \cdot \sin(2 \cdot \pi \cdot f \cdot k) = e^{j \cdot 2 \cdot \pi \cdot f \cdot k} \\ f \in [0 \quad 1] \end{cases} \quad (20)$$

La sortie est aussi un signal échantillonné sinusoïdal de gain  $A(f)$  et de phase  $\Phi(f)$  variables et dépendant de  $f$ ; la fréquence réduite.

$$\begin{cases} s(k) = A(f) \cdot e^{j \cdot (2 \cdot \pi \cdot f \cdot k + \Phi(f))} \\ f \in [0 \quad 1] \end{cases} \quad (21)$$

Le nombre complexe, noté abusivement  $F(f)$ , d'amplitude  $A(f)$  et de phase  $\Phi(f)$  est obtenu à partir de  $F(z)$  en remplaçant  $z$  par  $e^{j \cdot 2 \cdot \pi \cdot f}$  :

$$\begin{cases} F(f) = F(z = e^{j \cdot 2 \cdot \pi \cdot f}) \\ f \in [0 \quad 1] \end{cases} \quad (22)$$

Le diagramme représentant le gain et la phase en fonction de la fréquence réduite s'appelle aussi un diagramme de Bode. L'échelle des fréquences est généralement linéaire de 0 à 0.5 (puisque le diagramme est nécessairement symétrique). La Figure 7 représente par exemple le diagramme de Bode d'un système du second ordre.

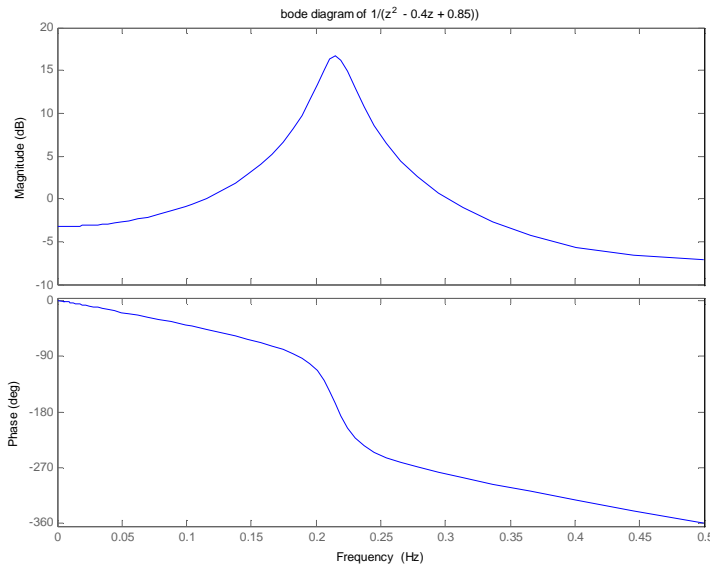


Figure 7 : exemple du diagramme de Bode d'un système discret du second ordre

### 3.3.2 Où l'on voit apparaître le théorème de Shannon

Le changement de variable  $z = e^{j2\pi f}$  n'est pas bijectif puisque  $e^{j2\pi f} = e^{j2\pi(f+k)}$  avec  $k$  entier. Si l'on fait le lien entre domaine continu et domaine fréquentiel on peut remarquer que le changement de variable  $z = e^{j2\pi f}$  fait correspondre une zone du plan complexe continu à une zone du plan complexe discret. La transformation est bijective si l'on se contente de la zone du plan complexe continu contenu entre les droites d'ordonnée 0 et 1.

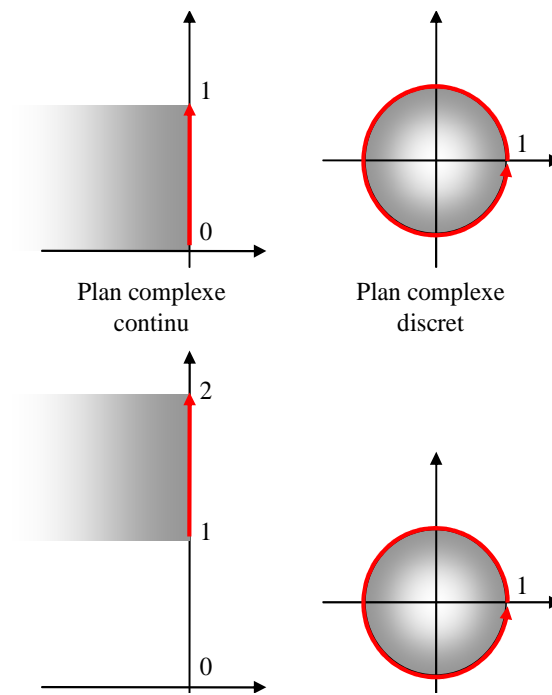


Figure 8 : correspondance entre plan complexe continu et plan complexe discret.





## 4. SYSTEMES CONTINUS ECHANTILLONNES

### 4.1 Structure des systèmes commandés par ordinateur

Les systèmes commandés par ordinateurs sont presque toujours des systèmes physiques continus par nature. Le ordinateur traite par contre les entrées et les sorties de manière discrète avec une période d'échantillonnage  $T_e$ . Les éléments importants, du point de vue de l'automaticien sont donc :

- le convertisseur analogique / numérique CAN ou du moins le système qui fournit séquentiellement les mesures au ordinateur à la période  $T_e$ .
- le ordinateur lui-même. Il doit effectuer l'acquisition des mesures, le traitement des informations, l'envoi des consignes dans un temps inférieur à  $T_e$ .
- le convertisseur numérique / analogique CNA.
- le système physique.

On suppose, pour faire simple et pour pouvoir utiliser la panoplie des outils mathématiques, que les systèmes sont échantillonnés à la même période  $T_e$  et au même moment.

La Figure 9 montre le système physique commandé par le ordinateur, dans sa version simpliste.

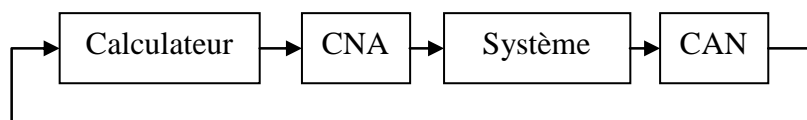


Figure 9 : système commandé par ordinateur

Au XXIème siècle les choses sont quand même parfois un petit peu plus compliquées :

- le ordinateur fournit bien souvent un signal de commande numérique (transporté par bus informatique USB, RS232, CAN, Ethernet, etc...) directement interprété par l'actionneur. On parle "d'actionneur intelligent".
- de même le "capteur intelligent" fournit aussi bien souvent la mesure sous forme d'un signal numérique.
- Les signaux sont transportés par des bus informatiques qui n'ont pas que ça à faire : il peut y avoir du retard.
- de même les capteurs et actionneurs sont rarement tous seuls, ce qui se traduit par des périodes d'échantillonnage différentes, et parfois non synchrones...

- enfin le ordinateur peut aussi avoir d'autres choses à faire...

Ceci étant dit nous allons pour le moment nous restreindre au cas représenté sur la Figure 9 avec des instants d'échantillonnage synchrones et régulièrement espacés d'un temps  $T_e$ .

## 4.2 Echantillonnage des systèmes continus

On dispose généralement de la fonction de transfert continue (notée en utilisant la variable complexe  $s$  ou  $p$ )  $F(s)$  du système continu. Le problème est d'obtenir la fonction de transfert discrète du système échantillonné de manière à pouvoir synthétiser un correcteur. Le système continu est associé au convertisseur numérique – analogique qu'il faut prendre en compte.

Les convertisseurs numérique analogiques peuvent généralement se représenter par des "échantillonneurs – bloqueurs d'ordre 0" ou BOZ. Cela signifie que la commande est maintenue à une valeur constante pendant toute la durée du temps d'échantillonnage.

Si l'on applique une commande discrète impulsionnelle à un BOZ il va maintenir la commande 1 pendant la durée  $T_e$  puis revenir à zéro (Figure 10).

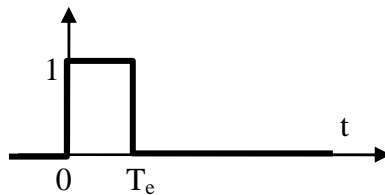


Figure 10 : réponse impulsionnelle d'un bloqueur d'ordre 0

La réponse impulsionnelle  $BOZ(t)$  est donc composée d'un échelon unitaire suivi d'un échelon unitaire de signe opposé et décalé d'un temps  $T_e$ . La transformée de Laplace de la réponse impulsionnelle du bloqueur est donnée par la formule (23) :

$$BOZ(s) = \frac{1}{s} - \frac{e^{-jT_e}}{s} \quad (23)$$

Du point de vue du ordinateur il faut donc prendre en compte le système continu de fonction de transfert  $F_{BOZ}(s)$  que l'on devra échantillonner :

$$F_{BOZ}(s) = \frac{1 - e^{-jT_e}}{s} \cdot F(s) \quad (24)$$

$$F_{\text{BOZ}}(z) = Z^{-1} \left( \frac{1 - e^{-jT_e}}{s} \cdot F(s) \right) \quad (25)$$

En pratique l'obtention de la transformée en z d'un système échantillonné par bloqueur d'ordre 0 se fait à l'aide de tables. On donne quelques exemples fondamentaux dans le Tableau 3 ci-dessous.

Tableau 3 : transformée en z de systèmes continus échantillonnés

F(s)	F <sub>BOZ</sub> (z)
1	1
$\frac{1}{s}$	$\frac{T_e \cdot z^{-1}}{1 - z^{-1}}$
$\frac{1}{1 + T \cdot s}$	$\begin{cases} \frac{b_1 \cdot z^{-1}}{1 + a_1 \cdot z^{-1}} \\ b_1 = 1 - e^{-T_e/T} \quad a_1 = -e^{-T_e/T} \end{cases}$
$\frac{e^{-Ls}}{1 + T \cdot s}$	$\begin{cases} \frac{b_1 \cdot z^{-1} + b_2 \cdot z^{-2}}{1 + a_1 \cdot z^{-1}} \quad a_1 = -e^{-T_e/T} \\ b_1 = 1 - e^{-(L-T_e)/T} \quad b_2 = e^{-T_e/T} (e^{L/T} - 1) \end{cases}$
$L < T_e$	

Si l'on dispose de Matlab c'est bien sûr encore plus simple. On utilise la fonction c2d avec l'argument 'zoh' pour "zero holder hold" (bloqueur d'ordre zéro).

#### Matlab

```
% Définition d'un système continu :
sysc=tf(1,[1 1]);
% Obtention de la transformée en z du système échantillonné à Te
Te=0.1;
sysd=c2d(sysc,Te,'zoh');
% Notation en z^-1
sysd.variable='z'
```

### 4.3 Effets de l'échantillonnage

Le convertisseur numérique a deux effets (néfastes) importants : l'effet de retard et l'effet de quantification.

L'effet de retard se voit dans (24) : on peut dire en première approximation que l'échantillonnage se traduit par un retard pur de  $T_e/2$ . Cela a donc, si ce temps

d'échantillonnage est non négligeable par rapport aux constantes de temps du système, un effet déstabilisant.

Pour illustrer l'effet de quantification supposons par exemple que l'on utilise un convertisseur analogique/numérique 8 bits 0-10V. Le plus petit incrément que le convertisseur peut distinguer est de  $10V/2^8 = 40mV$ . Si le signal utile varie physiquement entre 5V et 5,4V on n'aura que dix incréments utilisables... Ce phénomène n'entre pas dans la panoplie des phénomènes modélisables par la transformée en z. En pratique il est donc souvent nécessaire d'étudier le système et de synthétiser la commande numérique en négligeant ce phénomène puis de vérifier a posteriori qu'il n'a pas un effet trop néfaste.

#### 4.4 Où l'on revoit le théorème de Shannon

Le théorème de Shannon est bien connu en ce qui concerne l'échantillonnage de signaux discrets : on sait que le signal ne doit pas comporter de composante de fréquence supérieure à  $f_e/2$  si l'on ne veut pas observer de repliement. On doit aussi prendre des précautions si l'on veut travailler sur la fonction de transfert discrète correspondant à un système continu échantillonné.

On va traiter un exemple concret pour mettre en évidence le problème de repliement dans le cas de la commande des systèmes continus. On s'intéresse à deux systèmes continus du second ordre ayant des pôles de même partie réelle mais de parties imaginaires séparées de  $2\pi T_e$ . Le script Matlab ci-dessous montre la création des deux systèmes :

##### Matlab

```
% Définition de deux systèmes continus :  
sys1=zpk([],[-1-2i -1+2i],2^2+1);  
sys2=zpk([],[-1-(2+2*pi)*i -1+(2+2*pi)*i],1+(2+2*pi)^2);
```

On voit sur la Figure 11 que la réponse indicielle des deux systèmes est bien différente.

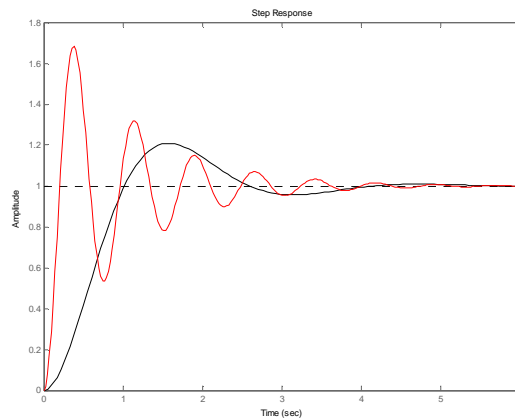


Figure 11 : réponse indicielle de deux systèmes du second ordre

Si l'on calcule la fonction de transfert échantillonnée des deux systèmes à la période de 1 seconde on obtient deux réponses indicielles très proches (représentés par des  $\circ$  sur la Figure 12). Conclusion : la période d'échantillonnage était mal choisie

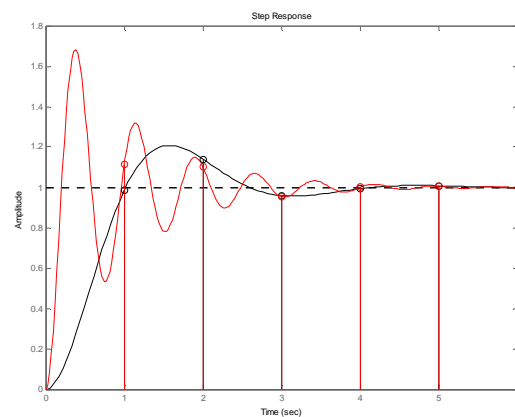


Figure 12 : réponse indicielle de deux systèmes du second ordre échantillonnés à 1s

En pratique la commande échantillonnée de systèmes continus doit être associée à certaines précautions :

- le système ne doit pas comporter de dynamiques propres situées dans les fréquences supérieures à  $f_c/2$  (exemple illustré dans ce chapitre).
- les signaux échantillonnés doivent être "nettoyés" des fréquences supérieures à  $f_c/2$  avec un filtre anti-repliement. (A moins que le signal soit suffisamment propre et ne comporte pas de composantes significatives au-delà de ces fréquences).

### Choix de la période d'échantillonnage

Le choix de la période d'échantillonnage est très important et dépend de la dynamique des signaux. Attention, les systèmes commandés en boucle fermée ont souvent une dynamique plus rapide que le système en boucle ouverte : il faut donc prendre en compte la dynamique en boucle fermée pour le choix de la fréquence d'échantillonnage.

En pratique on préconise de prendre une fréquence 5 à 25 fois supérieure à la bande passante souhaitée en boucle fermée.

On peut être tenté de prendre une fréquence d'échantillonnage très élevée si la technologie le permet (ordinateur et convertisseurs suffisamment rapide). Ceci peut amener rapidement à des problèmes numériques. Imaginons par exemple le cas d'un système ayant une dynamique très lente (quelques secondes) et un système de mesure codant les mesures sur 10 bits (cas classique). La Figure 13 illustre cet exemple. Le signal varie peu entre deux instants d'échantillonnage et les variations du signal sont faibles par rapport à la valeur de l'incrément numérique (figure du haut). Le calcul de la simple dérivée numérique se traduit donc par une succession de  $-1$   $0$  et  $+1$ .

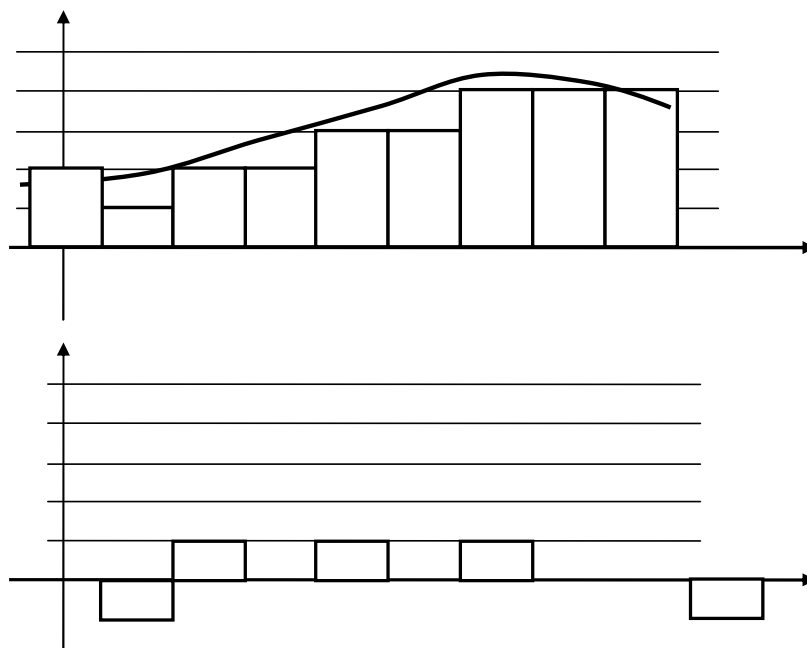


Figure 13 : période d'échantillonnage trop faible. En haut : signal échantillonné et discrétisé ; en bas : calcul de la dérivée numérique.

### Filtre anti-repliement

En toute rigueur chaque fois qu'un signal est échantillonné il faudrait faire précéder le convertisseur numérique d'un filtre analogique anti-repliement éliminant physiquement toutes les composantes du signal de fréquence supérieure à  $f_e/2$ .

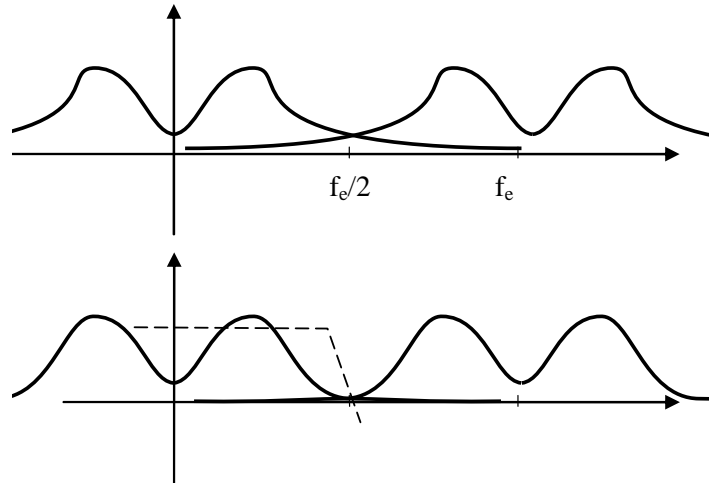


Figure 14 : Effet du filtre anti-repliement

Bien souvent la période d'échantillonnage de l'asservissement est relativement lente (quelques secondes) et il est alors impossible de réaliser un filtre de constante de temps aussi faible. On procède alors à un sur-échantillonnage : le signal est filtré, échantillonné à haute fréquence  $T'_e$  puis filtré numériquement et sous-échantillonné à la période  $T_e$  beaucoup plus grande que  $T'_e$  comme illustré sur la

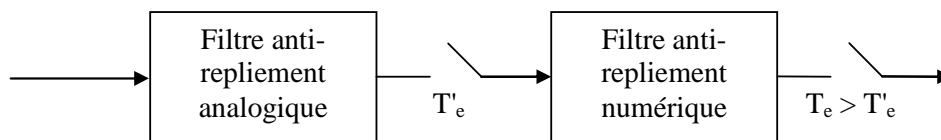


Figure 15 : Sur échantillonnage

Un simple filtre numérique de type moyenne mobile est souvent suffisant.

### 4.5 Discrétisation de fonction de transfert continues

Une manière classique pour réaliser un contrôle par ordinateur est de commencer par synthétiser un correcteur continu avec les outils de l'automatique classique puis de trouver un correcteur discret qui "approche" convenablement le correcteur continu.

#### 4.5.1 Approximation numérique d'équation différentielle

Supposons que le correcteur continu que l'on a synthétisé soit du type PD :

$$\text{PID}(s) = \frac{e(s)}{\varepsilon(s)} = k_p + k_D \cdot s \quad (26)$$

Il s'agit de calculer les commandes  $e(k)$  successives issues du correcteur en fonction des erreurs  $\varepsilon(k)$ . Une approche classique et intuitive est d'approximer la dérivée par une différence :

$$e(k) = k_p \cdot \varepsilon(k) + k_D \cdot \frac{\varepsilon(k) - \varepsilon(k-1)}{T_e} \quad (27)$$

Si on calcule la transformée en  $z$  de  $e(k)$  on obtient :

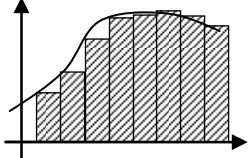
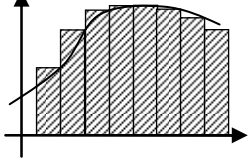
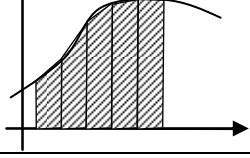
$$e(z) = k_p \cdot \varepsilon(z) + k_D \cdot \frac{\varepsilon(z) - z^{-1}\varepsilon(z)}{T_e} = \left( k_p + k_D \cdot \frac{1 - z^{-1}}{T_e} \right) \cdot \varepsilon(z) \quad (28)$$

On retrouve la fonction de transfert PD(s) dans laquelle on a effectué le changement de variable  $s = \frac{1-z^{-1}}{T_e}$

De manière plus générale, l'approximation de fonction de transfert revient à effectuer l'approximation numérique d'intégration. Selon la méthode d'intégration choisie on effectue des changements de variable différents.



Tableau 4 : méthodes d'intégration

Méthode	Changement de variable	
Euler progressif (Euler's forward method)	$s = \frac{1 - z^{-1}}{z^{-1} \cdot T_e}$	
Euler rétrograde (Euler's backward method)	$s = \frac{1 - z^{-1}}{T_e}$	
Méthode des trapèzes, ou bilinéaire, ou de Tustin	$s = \frac{2}{T_e} \cdot \frac{1 - z^{-1}}{1 + z^{-1}}$	

En pratique le changement de variable  $s = \frac{1 - z^{-1}}{z^{-1} \cdot T_e}$  n'est pas sans risque. On donne sur la Figure 16 la correspondance entre la zone stable du plan complexe en  $s$  et la zone correspondante après la transformation (Euler progressif, Euler rétrograde, bilinéaire). On voit notamment qu'il est possible, partant d'un système continu stable, d'obtenir des pôles dans le domaine discret en dehors du cercle de stabilité si la transformation appliquée est celle d'Euler progressif.

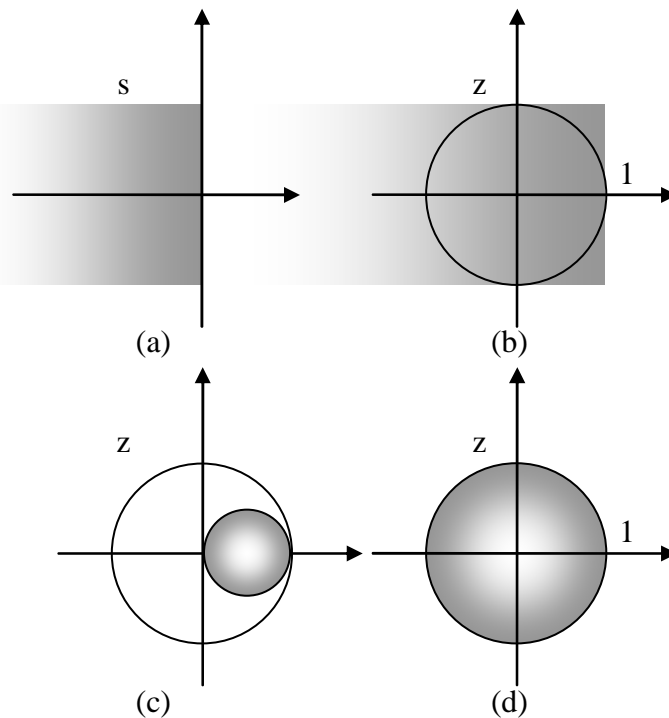


Figure 16 : correspondance des plans complexes. (a) : zone de stabilité dans le plan complexe continu en  $z$  ; (b) zone correspondante avec l'approximation d'Euler progressif ; (c) zone correspondante avec l'approximation d'Euler rétrograde ; (d) zone correspondante avec l'approximation bilinéaire.

La fonction `c2d` de Matlab permet d'obtenir l'équivalent discret d'une fonction de transfert continue avec la méthode d'intégration comme argument :

```

Matlab
sysc=tf([1 0],[1 2 5]);
Te=0.1;
% Méthode d'euler :
sysd1=c2d(sysc,Te,'zoh');
% Méthode des trapèzes :
sysd2=c2d(sysc,Te,'foh');
% Méthode bilinéaire :
sysd3=c2d(sysc,Te,'tustin');
    
```

#### 4.5.2 Superposition des réponses indicielles

Une autre méthode pour approximer une fonction de transfert discrète est de s'assurer que la réponse indicielle de sa version discrétisée correspond exactement lors des temps d'échantillonnage.

### 4.5.3 Correspondances pôles - zéros

La dernière méthode est de partir d'une décomposition de la fonction de transfert continue en pôles et zéros et d'effectuer la conversion pôles à pôles et zéros par zéros. La correspondance, selon que le pôle est réel ou complexe est donnée ci-dessous :

$$\begin{aligned} (s + a) &\rightarrow (z - e^{-a \cdot T_e}) \\ (s + a)^2 + b^2 &\rightarrow z^2 - 2 \cdot e^{-a \cdot T_e} \cdot \cos(b \cdot T_e) \cdot z + e^{-2 \cdot a \cdot T_e} \end{aligned} \quad (29)$$

### 4.5.4 Bilan

Aucune méthode n'est parfaite et on n'aura jamais en même temps :

- même réponse indicielle
- même réponse impulsionnelle
- approximation (Euler, Tustin, etc...) de l'équation différentielle
- correspondance des pôles et des zéros
- correspondance des réponses fréquentielles

Ceci étant dit le choix de la période d'échantillonnage est primordial. Si la fréquence d'échantillonnage est beaucoup plus faible que la dynamique de la fonction de transfert à approximer la différence entre chaque méthode sera minimale.



## 5. IDENTIFICATION DES SYSTEMES ECHANTILLONES

Dans le présent chapitre on donne les bases des méthodes d'identification. Pour une approche complète, voir le cours de troisième année "Identification avion" [6] et les ouvrages [1] et [7].

Même si l'on dispose d'un modèle théorique du processus que l'on souhaite commander il est presque toujours nécessaire de l'affiner en procédant à des essais sur le système réel. Bien souvent le seul modèle dont l'on dispose est ce modèle issu de l'expérimentation. Différentes méthodes s'appliquent selon l'objectif recherché et le niveau de précision exigé. On distinguera :

- les modèles paramétriques
- les modèles simples
- les modèles non paramétriques

### 5.1 Identification d'un modèle ARX par la méthode des moindres carrés récursifs

A partir de cet exemple, le plus simple qui soit, on pose les bases de toutes les méthodes d'identification.

On considère un système d'entrée  $u(t)$  et de sortie  $y(t)$  représenté par l'équation de récurrence suivante :

$$y(t) + a_1 \cdot y(t-1) + a_2 \cdot y(t-2) + \dots + a_n \cdot y(t-n) = b_0 \cdot u(t) + b_1 \cdot u(t-1) + b_2 \cdot u(t-2) + \dots + b_m \cdot u(t-m) \quad (30)$$

L'objectif est, à partir d'essais et de mesures, d'identifier les meilleures valeurs possibles pour les coefficients  $a_i$  et  $b_i$ .

L'équation (30) peut être réécrite pour mettre en évidence la *prédiction* de la sortie  $y(t)$  à partir des mesures passées :

$$y(t) = -a_1 \cdot y(t-1) - a_2 \cdot y(t-2) - \dots - a_n \cdot y(t-n) + b_0 \cdot u(t) + b_1 \cdot u(t-1) + b_2 \cdot u(t-2) + \dots + b_m \cdot u(t-m) \quad (31)$$

Soit :

$$y(t) = \varphi^T(t) \cdot \theta(t)$$

avec :

$$\begin{cases} \varphi^T(t) = [-y(t-1) \ \dots \ -y(t-n) \ , \ u(t-1) \ \dots \ u(t-m)] \\ \theta(t) = [a_1 \ \dots \ a_n \ , \ b_1 \ \dots \ b_m] \end{cases} \quad (32)$$

Le but est de trouver le meilleur ensemble de paramètres  $\theta(t)$  : celui qui, à partir d'un ensemble de mesures  $\varphi(t)$  permet de prédire avec la meilleure précision la sortie effectivement mesurée.

Supposons qu'à la suite d'une expérience on dispose d'un ensemble de mesures  $\{y(t), u(t)\}_{t=1:N}$  et que l'ensemble des paramètres  $\theta$  n'est pas bien connu. On construit la séquence des *sorties prédites* :

$$\hat{y}(t) = \varphi^T(t) \cdot \theta(t) \quad (33)$$

Le paramètre  $\theta$  est celui qui minimise une norme quadratique :

$$\theta = \arg \max \left( \frac{1}{N} \sum_{t=1}^N (y(t) - \hat{y}(t))^2 \right) = \arg \max \left( \frac{1}{N} \sum_{t=1}^N (y(t) - \varphi^T(t) \cdot \theta(t))^2 \right) \quad (34)$$

Compte tenu du fait que l'équation est linéaire au départ et que le terme à minimiser est du type quadratique la solution est simplement obtenue par une pseudo-inverse :

$$\theta = \left[ \sum_{t=1}^N (\varphi(t) \cdot \varphi^T(t)) \right]^{-1} \sum_{t=1}^N (\varphi(t) \cdot y(t)) \quad (35)$$

On obtient donc la solution classique des moindres carrés. La formule (35) est le plus souvent utilisée sous sa forme récursive :

$$\theta(t) = \theta(t-1) + F(t) \cdot \varphi(t) \cdot (y(t) - \theta^T(t-1) \cdot \varphi(t))$$

avec

$$F(t)^{-1} = F(t-1)^{-1} + \varphi(t) \cdot \varphi(t)^T \quad (36)$$

ou

$$F(t) = F(t-1) - \frac{F(t-1) \cdot \varphi(t) \cdot \varphi(t)^T \cdot F(t-1)}{1 + \varphi(t)^T \cdot F(t-1) \cdot \varphi(t)}$$

(On laisse au lecteur le plaisir d'effectuer lui-même les démonstrations, qu'il peut du reste trouver dans [1] et [7]).

## 5.2 L'identification paramétrique en général et en pratique

L'exemple précédent a permis de mettre en évidence tous les aspects de la méthodologie classique de l'identification :

- il a fallu choisir un ensemble de mesures  $\{y(t), u(t)\}_{t=1:N}$ . Le choix du signal d'excitation  $u(t)$  est en pratique primordial,
- il a fallu choisir une structure pour le modèle à identifier (ici les ordres  $n$  et  $m$ ),
- il a fallu choisir un critère (ici un critère quadratique sur l'erreur entre la sortie et la sortie prédite),
- il a fallu choisir un algorithme et le paramétrer (ici l'algorithme des moindres carrés récursifs qu'il a fallu initialiser)

Enfin un point non traité dans le chapitre précédent concerne la validation :

- il aurait fallu déterminer une méthode pour comparer différents modèles et choisir le meilleur

### 5.2.1 Choix du signal d'excitation, traitement des mesures

De manière générale le système que l'on doit identifier doit être excité dans tout le domaine fréquentiel qui nous intéresse. Le signal classiquement utilisé est une séquence pseudo binaire aléatoire (Pseudo Random Binary Signal) tel celui représenté sur la Figure 17.

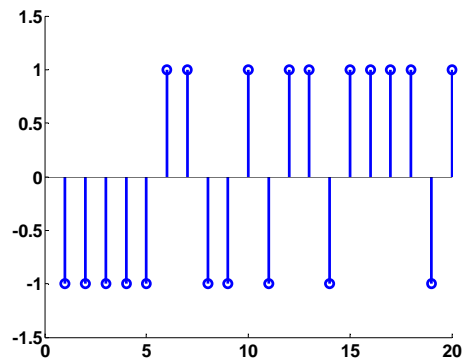


Figure 17 : exemple d'une séquence PRBS

Le signal d'excitation doit se superposer au signal de commande correspondant au point de fonctionnement étudié. Son amplitude doit être faible par rapport à l'amplitude des commandes du système (quelques %).

Une fois le système excité et les mesures effectuées il faut supprimer les valeurs moyennes des mesures.

### 5.2.2 Choix d'une structure de modèle

Le modèle étudié dans le chapitre précédent est de type ARX. On peut le noter sous forme polynomiale :

$$A \cdot y = B \cdot u \quad (37)$$

Ce modèle correspond en fait à un parti-pris sur la manière dont on traite les perturbations. En réalité on prend en compte un signal d'erreurs possibles  $e$  (de moyenne nulle) tels que :

$$A \cdot y = B \cdot u + e \quad (38)$$

Si l'on fait cette hypothèse de modélisation alors la méthode optimale est celle des moindres carrés. Malheureusement cette manière de prendre en compte les perturbations est bien restrictive. D'autres modélisations sont possibles, adaptées à d'autres algorithmes, par exemple le modèle OE :

$$y = \frac{B}{F} \cdot u + e \quad (39)$$

Il est aussi possible de "structurer" le modèle de bruit comme par exemple dans les modèles ARMAX ou BJ :

$$\begin{aligned} A \cdot y &= B \cdot u + C \cdot e \\ y &= \frac{B}{F} \cdot u + \frac{C}{D} \cdot e \end{aligned} \quad (40)$$

La structure du modèle étant choisie il faut choisir l'ordre de ses différents polynômes.

### 5.2.3 Choix d'un critère à minimiser

Dans la plupart des cas le critère à minimiser est l'erreur quadratique entre la sortie et la sortie prédite à un coup (one step ahead prediction). Ceci n'est bien évidemment pas restrictif et peut être généralisé à la prédiction à  $N$  coups.

### 5.2.4 Choix d'un algorithme

Même si chaque structure de modèle est généralement associée à un algorithme "optimal" celui-ci doit généralement se paramétrer.



Prenons par exemple l'algorithme des moindres carrés récursifs tel que présenté dans (36). Cette équation peut être modifiée en intégrant un "facteur d'oubli"  $\lambda$  :

$$\begin{aligned} \theta(t) &= \theta(t-1) + F(t) \cdot \varphi(t) \cdot (y(t) - \theta^T(t-1) \cdot \varphi(t)) \\ \text{avec} & \\ F(t)^{-1} &= \lambda \cdot F(t-1)^{-1} + \varphi(t) \cdot \varphi(t)^T \end{aligned} \quad (41)$$

Ce paramètre (typiquement entre 0.95 et 0.99) permet, comme son nom l'indique, "d'oublier" les anciennes mesures et de se concentrer sur les mesures les plus récentes.

### 5.2.5 Validation

Une fois l'algorithme d'identification appliqué et le modèle obtenu (ce qui prend généralement quelques secondes avec un ordinateur moderne) il reste à expertiser celui-ci. Selon la "qualité" du modèle obtenu on remet en cause l'une ou l'autre des étapes (signal d'excitation, structure du modèle, algorithme) et on réitère jusqu'à obtenir un modèle satisfaisant.

En pratique on essaie un certain nombre de modèles avant de choisir le "meilleur".

Le premier critère de qualité est la capacité du modèle à prédire effectivement la sortie (sortie prédite à un coup, sortie prédite à N coups). Il faut pour cela utiliser un jeu de mesures non utilisé pour l'identification. (En pratique le jeu de mesures initiales est coupé en deux : une partie pour l'identification et une partie pour la validation).

Le second critère de qualité est l'analyse du résidu : la composition spectrale de l'erreur de prédiction.

### 5.2.6 Utilisation de logiciels

En pratique il est commode d'utiliser un célèbre outil de calcul scientifique et sa toolboxe d'identification.

Comme les calculs proprement dit ne prennent que quelques secondes il est facile de réaliser un grand nombre d'identifications et de stocker les différents modèles pour les comparer ensuite. On est rapidement noyé sous le grand nombre de modèles possibles...

## 5.3 Identification "simple"

Il est souvent suffisant d'obtenir un modèle "simple" de type premier ordre et un simple essai indiciel convient souvent. Le système est excité avec un échelon unité et on cherche le meilleur modèle du premier ordre correspondant.

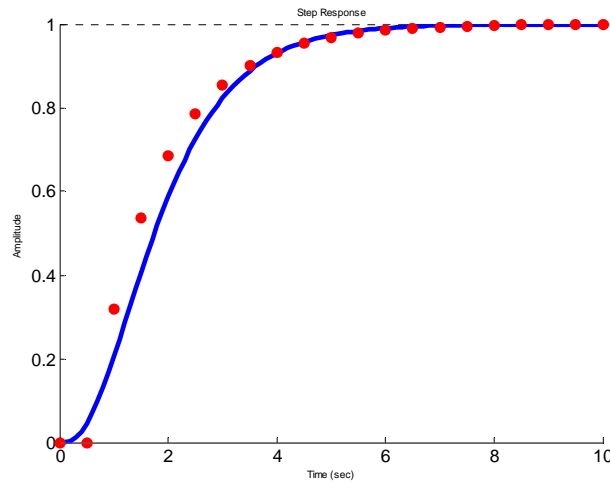


Figure 18 : identification sur un essai indiciel

Dans l'exemple de la Figure 18 on superpose le résultat d'une mesure (en bleu) avec la simulation d'un modèle discret avec une période de 0.5s. Le modèle cherché est du premier ordre avec un retard pur  $d$  :

$$y(z) = \frac{(1 + c_1) \cdot z^{-d}}{1 + c_1 \cdot z^{-1}} \cdot u(z) \quad (42)$$

Il s'agit de trouver l'entier  $d$  et le nombre réel  $c_1$  tels que la réponse indicelle du modèle corresponde pour le mieux aux mesures issues de l'essai. Ceci peut se formaliser par exemple de la manière suivante :

$$(c_1 \quad d_1) = \arg \min \sum_{t=1}^N (y_{\text{modèle}}(t) - y_{\text{mesure}}(t))^2 \quad (43)$$

Cette optimisation peut se faire à la main ou grâce à une procédure automatisée (avec le célèbre outil de calcul scientifique).

Dans le cas où le système comprend naturellement une intégration (par exemple la commande d'un moteur à courant continu par une tension et la mesure de l'angle de rotation) il est utile de prendre en compte explicitement cette intégration dans le modèle :

$$y(z) = \frac{1}{1 - z^{-1}} \cdot \frac{(1 + c_1) \cdot z^{-d}}{1 + c_1 \cdot z^{-1}} \cdot u(z) \quad (44)$$

La procédure est la même mais au lieu de prendre en compte la sortie  $y(t)$  on prend en compte la dérivée  $y'(t)$  :

$$y'(t) = y(t) - y(t-1) \quad (45)$$

## 5.4 Identification non paramétrique

Pour terminer ce panorama rapide sur les méthodes d'identification il faut parler des méthodes non paramétriques : celles qui permettent de tracer la réponse impulsionnelle ou la réponse fréquentielle du système.

### 5.4.1 Estimation de la réponse impulsionnelle

La réponse impulsionnelle d'un système peut être obtenue simplement en appliquant une entrée de type "impulsion" au système et d'en mesurer les conséquences.

La seconde méthode consiste à utiliser la méthode des moindres carrés (ou une méthode dérivée) comme vue au chapitre 5.1. La réponse impulsionnelle d'un système peut en effet être vue comme la fonction de transfert particulière où le dénominateur est de degré nul et le numérateur comporte un nombre infini (en pratique : grand) d'éléments ( $n = 0$  et  $m$  grand) :

$$y(t) = b_0 \cdot u(t) + b_1 \cdot u(t-1) + b_2 \cdot u(t-2) + \dots + b_m \cdot u(t-m) \quad (46)$$

La dernière méthode, qui sera développée ici est l'analyse d'intercorrélation.

Par définition l'intercorrélation  $R_{yu}$  entre deux signaux est :

$$R_{yu}(\tau) = E[(y(t) - E(y(t))) \cdot (u(t-\tau) - E(u(t-\tau)))] \quad (47)$$

Dans cette formule l'espérance mathématique peut être approchée grâce à la loi des grands nombres :

$$R_{yu}(\tau) \approx R_{yu}^N(\tau) = \lim_{N \rightarrow \infty} \frac{1}{N} \sum_{t=1}^N [(y(t) - m_y) \cdot (u(t-\tau) - m_u)] \quad (48)$$

On s'intéresse maintenant à un système tel que décrit par l'équation impulsionnelle (46) et on suppose que le signal d'entrée est un bruit blanc. Le signal d'entrée vérifie donc :

$$\begin{cases} \tau \neq 0 & R_{uu}(\tau) = 0 \\ \tau = 0 & R_{uu}(\tau) = R \end{cases} \quad (49)$$

Il vient alors naturellement :

$$b_k \approx \frac{1}{R} R_{yu}(k) \quad (50)$$

### 5.4.2 Estimation de la réponse fréquentielle

On utilise ici le spectre d'intercorrélation. Par définition le spectre d'intercorrélation entre deux signaux est :

$$\Phi_{yu}(\omega) = \sum_{\tau=-\infty}^{\infty} R_{yu}(\tau) \cdot e^{-i\omega\tau} \quad (51)$$

On suppose que le signal d'entrée  $u(t)$  et le signal de sortie  $y(t)$  sont reliés par une fonction de transfert linéaire du type :

$$y(t) = F \cdot u(t) + v(t) \quad (52)$$

où  $v(t)$  est une perturbation indépendante de  $u(t)$ . On a alors les relations :

$$\Phi_{yu}(\omega) = F(j\omega) \cdot \Phi_{uu}(\omega) \quad (53)$$

Il reste à estimer les spectres d'intercorrélation pour remonter à la réponse fréquentielle  $F(j\omega)$  recherchée. Une idée naturelle est d'utiliser la relation (48) mais cela marche très mal dans presque tous les cas, même en choisissant  $N$  assez grand. L'amélioration est d'utiliser une "fenêtre" pour favoriser les valeurs faibles de  $\tau$  par rapport aux valeurs élevées. La solution classique utilisant la *fenêtre de Hamming* donne :

$$\Phi_{yu}^N(\omega) = \sum_{\tau=-\gamma}^{\gamma} R_{yu}^N(\tau) \cdot w_{\gamma}(\tau) \cdot e^{-i\omega\tau} \quad (54)$$

avec

$$\begin{cases} w_{\gamma}(\tau) = \frac{1}{2} \left( 1 + \cos\left(\frac{\pi \cdot \tau}{\gamma}\right) \right) & |\tau| < \gamma \\ w_{\gamma}(\tau) = 0 & |\tau| > \gamma \end{cases}$$

On utilise ensuite la relation (51) pour estimer la réponse fréquentielle :

$$F(j \cdot \omega) = \frac{\Phi_u^N(\omega)}{\Phi_{yu}^N(\omega)} \quad (55)$$



## 6. SYSTEMES BOUCLES

### 6.1 Fonction de transfert en boucle fermée

Un système et sa commande par ordinateur peut être représenté par un ensemble de fonctions de transfert formant un schéma-bloc plus ou moins compliqué (Figure 19).

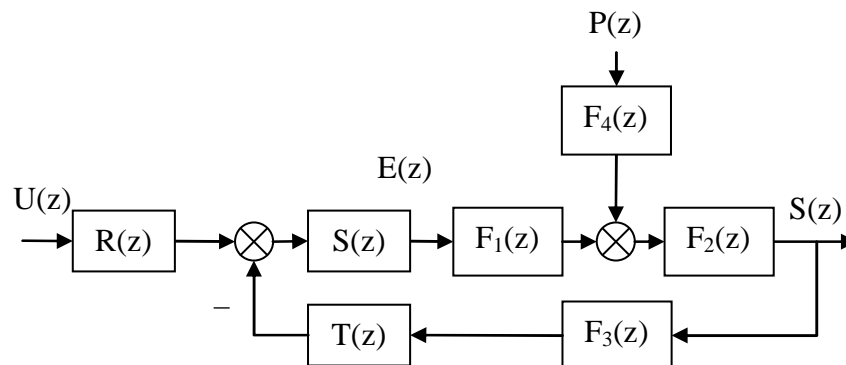


Figure 19 : système discret commandé par ordinateur. Le correcteur est représenté par les fonctions R,S,T, le système par  $F_1$  et  $F_2$ , le modèle de perturbations par  $F_4$  et le modèle des actionneurs par  $F_4$ .

Quelle que soit la structure d'un système asservi il est possible de se ramener à une forme canonique comprenant une fonction de transfert en boucle ouverte en combinant judicieusement les fonctions de transfert du procédé, du régulateur et des perturbations. La Figure 20 représente par exemple le système de la Figure 19 où  $F$  est une combinaison des fonctions de transfert du système, du régulateur, des perturbations et des capteurs.

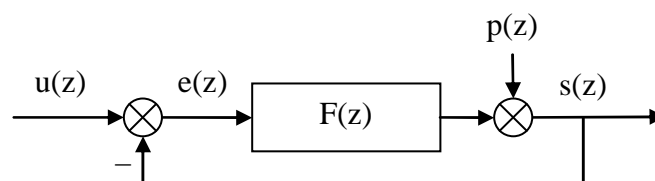


Figure 20 : système discret, notations.

Par défaut on considère une contre-réaction négative et une perturbation additive sur la sortie. La fonction de transfert  $F$  est appelée *fonction de transfert en boucle ouverte*. Comme pour les systèmes continus la *fonction de transfert en boucle fermée* est obtenue simplement à partir de  $F$  (58).

$$F(z) = \frac{s(z)}{e(z)} = \frac{B(z^{-1})}{A(z^{-1})} \quad (56)$$

$$G(z) = \frac{s(z)}{u(z)} = \frac{B(z^{-1})}{A(z^{-1}) + B(z^{-1})}$$

## 6.2 Erreur statique

L'erreur statique du système en boucle fermée, comme pour le système en boucle ouverte, est simplement le rapport des coefficients.

$$G_0 = \frac{\sum_{i=0}^{i=m} b_i}{1 + \sum_{i=0}^{i=m} b_i + \sum_{i=1}^{i=n} a_i} \quad (57)$$

Si l'on souhaite que l'asservissement confère un gain statique unitaire au système en boucle fermée (c'est-à-dire que l'on force la sortie  $s(k)$  à tendre vers la valeur de consigne  $u(k)$ ) il faut que ce rapport de coefficients soit égal à un. On montre facilement que cela est obtenu si la fonction de transfert  $F(z)$  comporte un intégrateur :

$$F(z) = \frac{s(z)}{e(z)} = \frac{1}{1 - z^{-1}} \cdot \frac{B(z)}{A'(z)} \quad (58)$$

## 6.3 Rejet de perturbations

La fonction de transfert entre la perturbation  $p(z)$  et la sortie  $y(z)$  est appelée fonction de sensibilité perturbation-sortie, notée  $S_{yp}$ .

$$S_{yp}(z) = \frac{s(z)}{p(z)} = \frac{A(z^{-1})}{A(z^{-1}) + B(z^{-1})} \quad (59)$$

Si l'on souhaite qu'une perturbation de type échelon soit parfaitement rejetée il faut que  $A(1) = 0$ , on retrouve donc la condition (58) à savoir que le système doit comporter une intégration dans la boucle directe. On peut être en outre être amené à spécifier que les perturbations ne soient pas amplifiées dans une certaine bande de fréquence. On rajoute donc une contrainte du type  $|S_{yp}(e^{j2\pi f})| < S_{\max}$  pour  $f$  appartenant à un certain domaine de  $[0, 0.5]$ .



## 6.4 Stabilité des systèmes à simple gain de rétroaction

Les critères de Routh et de Jury (donnés dans le chapitre 3.2.7 p13) ou le calcul numérique des pôles permettent de déterminer la stabilité du système en boucle fermée si l'on connaît sa fonction de transfert.

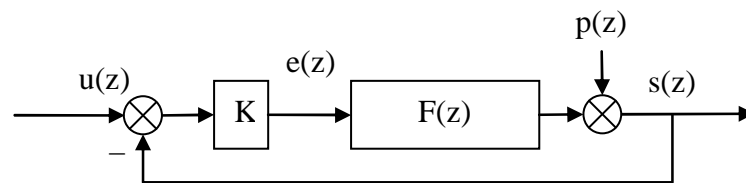


Figure 21 : système bouclé par un simple gain.

On s'intéresse ici au cas particulier où l'asservissement est réalisé par un simple gain de rétroaction  $K$  (Figure 21). On cherche la valeur de  $K$  qui amène le système en limite de stabilité. L'outil adapté ici est le lieu d'Evans ou lieu des racines (root locus) qui permet de visualiser l'évolution des pôles du système bouclé lorsque le gain  $K$  varie.

Prenons l'exemple d'un système comportant trois pôles et deux zéros. L'exemple est donné sous forme de script Matlab :

### Matlab

```
% Définition d'un système discret :
sys=zpk([-2.037 -0.1406],[0.6065 0.3679 0.3679],1,1)
%Tracé du lieu des racines
rlocus(sys);
```

Le script précédent produit le tracé de la Figure 22. On pourrait y lire que la limite de stabilité est obtenue pour  $K=0.2$  environ (détail de la Figure 23).

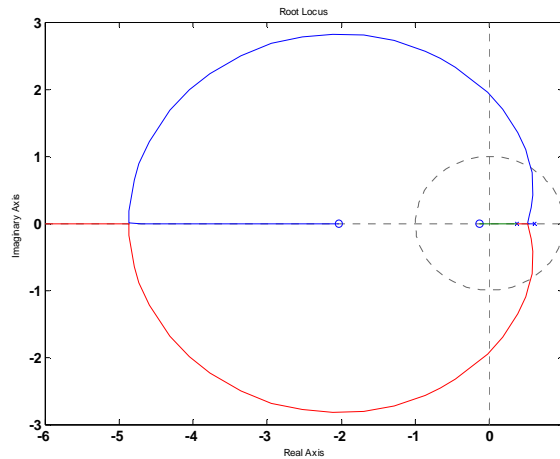


Figure 22 : lieu des racines

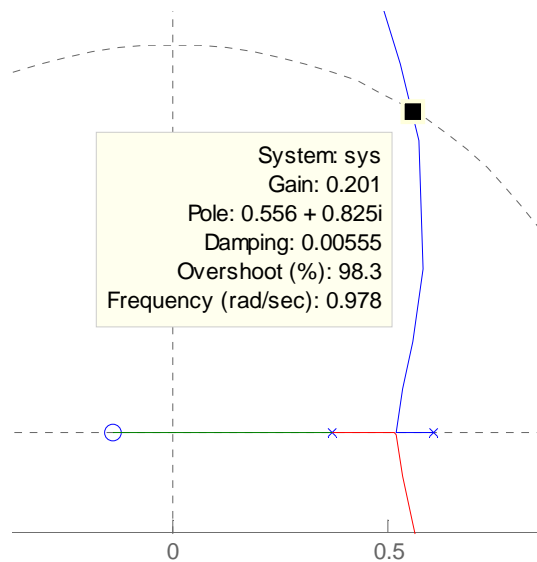


Figure 23 : lieu des racines : détail de la limite de stabilité

On voit sur la Figure 22 que le lieu des racines d'un système échantillonné ressemble à celui d'un système continu :

- on commence par tracer les pôles ( $\times$ ) et les zéros ( $o$ ) du système en boucle ouverte.
- les pôles du système en boucle fermée décrivent un lieu qui part des pôles (pour  $K = 0$ ) et arrive aux zéros (pour  $K = \infty$ )
- lorsque il y a  $n-m$  plus de pôles que de zéros le système comporte  $n-m$  branches asymptotiques.

On ne décrira pas ici les diverses propriétés du lieu des racines qui permettent de le tracer à la main puisque il est facile à obtenir avec un ordinateur.

Il peut aussi être utilisé pour régler "à la main" le gain permettant d'obtenir un ou plusieurs modes dominants ayant certaines propriétés de fréquence et d'amortissement. On voit sur la Figure 24 que le gain 0.017 permet d'obtenir un pôle double d'amortissement 0.706 et de fréquence 0.65 rad/s et un pôle simple de fréquence 1.63 rad/s.

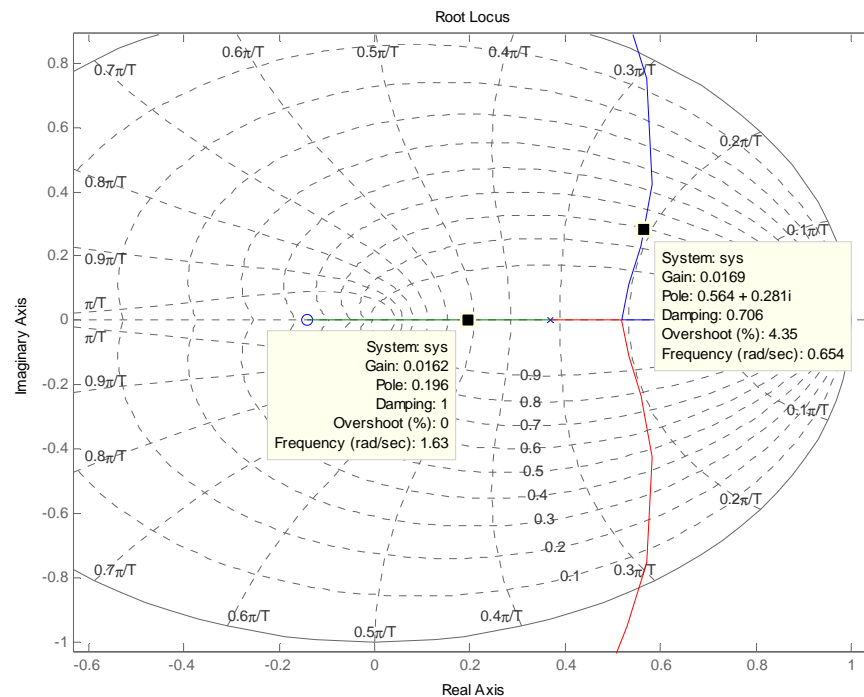


Figure 24 : utilisation du lieu des racines pour régler le gain de la boucle fermée

L'outil rltool de Matlab permet de nombreuses manipulations graphiques de ce type.

## 6.5 Analyse des systèmes bouclés dans le domaine fréquentiel : stabilité

Comme dans le cas continu il est possible de prévoir des propriétés, en particulier la stabilité, d'un système en boucle fermée à partir de l'examen de la fonction de transfert en boucle ouverte.

On ne redémontrera pas ici le théorème de Nyquist mais on se contente de rappeler l'énoncé.

On rappelle que le *lieu de Nyquist* est le lieu de  $F(j2\pi f)$  dans le plan complexe pour  $f$  variant de 0 à  $+\infty$ . A noter que le lieu pour la zone  $[0, 0.5]$  est symétrique de celui pour la zone  $[0.5, 1]$ . Pour cette raison on ne représente généralement que le lieu pour  $f$  appartenant à  $[0, 0.5]$ .

### 6.5.1 Théorème de Nyquist dans le cas où le système est stable en boucle ouverte (critère du revers)

Le système est stable en boucle fermée si le lieu de Nyquist de la fonction de transfert en boucle ouverte parcouru dans le sens des fréquences croissantes de 0 à 0.5 laisse le point critique (-1,0) à gauche.

La Figure 25 représente par exemple le lieu de Nyquist d'une fonction de transfert en boucle ouverte qui sera stable en boucle fermée.

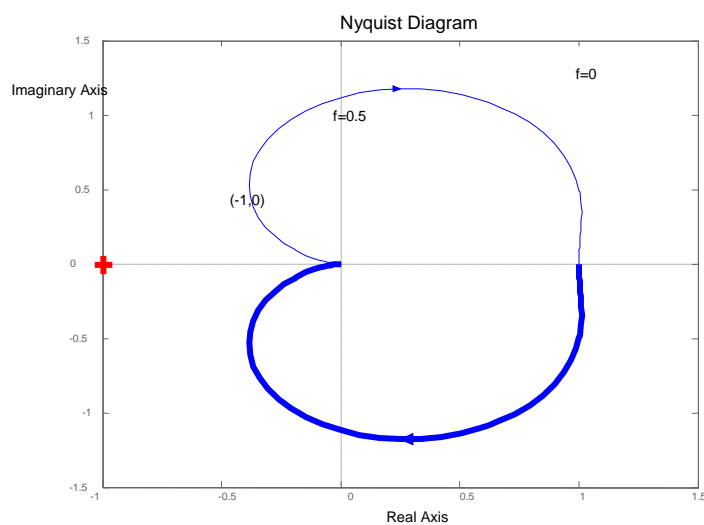


Figure 25 : lieu de Nyquist d'un système stable

### 6.5.2 Théorème de Nyquist dans le cas où le système est instable en boucle ouverte

Si le système en boucle ouverte comporte  $N$  pôles instables il sera stable en boucle fermée si le lieu de Nyquist de la fonction de transfert en boucle ouverte parcouru dans le sens des fréquences croissantes de  $-0.5$  à  $0.5$  entoure le point critique  $(-1,0)$   $N$  fois dans le sens des aiguilles d'une montre..

### 6.5.3 Interprétations du lieu de Nyquist

Comme dans le cas continu, la distance à laquelle on passe du point critique renseigne sur la robustesse du système bouclé. La distance sera exprimée de différente manière (Figure 26) :

- marge de module  $\Delta M$  : plus petite distance entre le lieu des racines et le point critique,

- marge de gain  $\Delta G$
- marge de phase  $\Delta P$

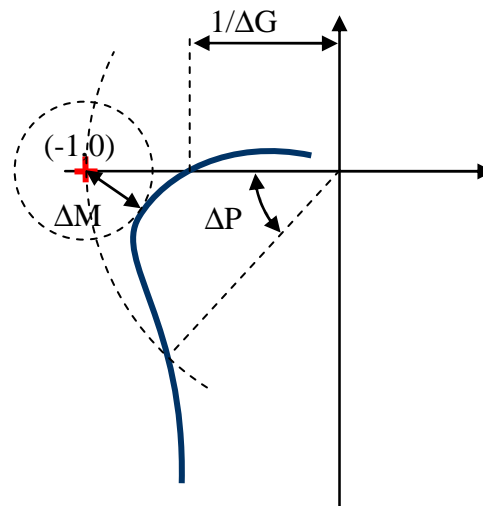


Figure 26 : Marges de stabilité

Les valeurs typiques de ces marges sont données ci-après :

- $\Delta G \geq 6\text{dB}$
- $\Delta M \geq -6\text{dB}$
- $30^\circ \leq \Delta P \leq 60^\circ$

La distance entre le point critique et le lieu des racines est la norme de  $1+F(j2\pi f)$ . On peut donc exprimer la marge de module en fonction de la fonction de sensibilité :

$$\begin{aligned} \Delta M &= \min |1 + F(j \cdot \omega)|_{\omega \in [0, \pi]} \\ \Delta M &= \min \left| 1 + \frac{B(j \cdot \omega)}{A(j \cdot \omega)} \right|_{\omega \in [0, \pi]} = \max \left| \frac{A(j \cdot \omega)}{A(j \cdot \omega) + B(j \cdot \omega)} \right|_{\omega \in [0, \pi]} \quad (60) \\ \Delta M &= \max |S_{yp}(j \cdot \omega)|_{\omega \in [0, \pi]}^{-1} \\ \Delta M_{\text{dB}} &= -\max |S_{yp}(j \cdot \omega)|_{\text{dB}} \end{aligned}$$

Si l'on trace l'évolution de  $S_{yp}$  en fonction de  $f$  on obtient donc la valeur de la marge de module.



# 7. METHODES DE REGULATION DES SYSTEMES ECHANTILLONES

## 7.1 Introduction

Les bases théoriques permettant de concevoir le correcteur pour un système échantillonné ont été posées dans les chapitres précédents. Ce chapitre fait le bilan des connaissances acquises et donne quelques solutions pour calculer des correcteurs à partir d'un cahier des charges précis.

Les systèmes que l'on cherche à commander sont généralement, par nature, continus. La réalisation d'un correcteur par un ordinateur implique que l'on génère une équation de commande discrète (généralement une équation récurrente). L'obtention de cette équation de commande peut se faire de deux manières :

- la première méthode consiste à d'abord synthétiser un correcteur continu (en utilisant les outils classiques de l'automatique et la simulation) et discrétiser ensuite ce correcteur. C'est l'objet du chapitre 7.2.2.
- la seconde méthode consiste à tenir la représentation discrète du système continu associé à son interface numérique (convertisseurs analogique-numérique et numérique-analogique) comme expliqué dans le chapitre 4.5 puis synthétiser directement le correcteur discret. Cela sera expliqué dans le chapitre 7.3 et suivants.

Ce choix peut être résumé par la figure suivante :

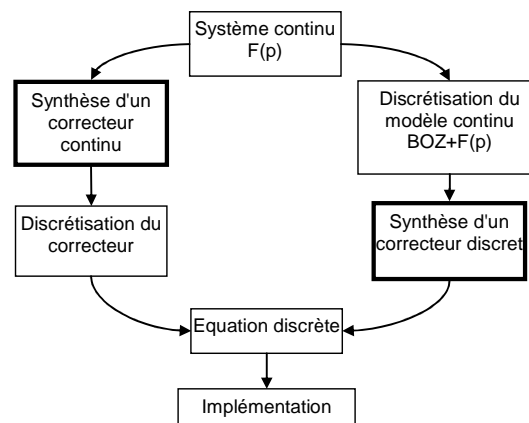


Figure 27 : Réalisation d'un correcteur discret.

Au-delà des affaires de goût, le choix de la méthode sera souvent dicté par l'application. En effet, la première méthode est d'autant meilleure que le correcteur discrétisé approche correctement le correcteur continu. Cela implique que le temps d'échantillonnage soit très faible par rapport aux constantes de temps du système

bouclé. Dans le cas où les contraintes temps réel sont fortes (capacités de calcul partagées par exemple) il sera préférable de synthétiser un correcteur discret.

## 7.2 Structure d'un système commandé par ordinateur, hypothèses

Par la suite on se ramènera toujours à un système tel que celui représenté sur la Figure 28 en cherchant un correcteur  $K(z)$  linéaire. On montre que quel que soit le correcteur linéaire on peut se ramener à la forme représenté sur la Figure 29 (forme RST).

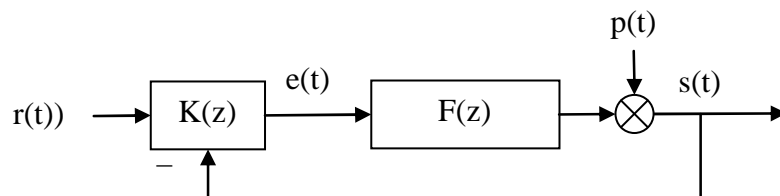


Figure 28 : système bouclé, forme générale.

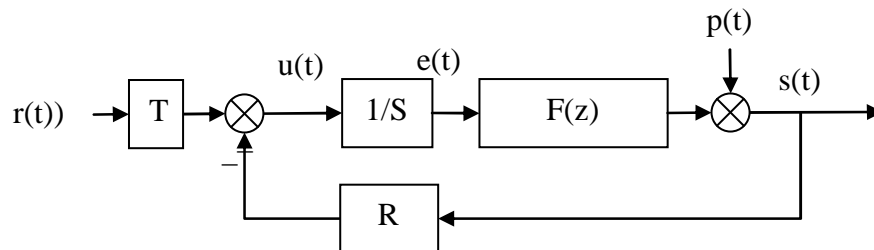


Figure 29 : système bouclé, forme canonique.

### 7.2.1 Discrétisation, choix de la période d'échantillonnage

Du point de vue du ordinateur, l'extérieur (le procédé) est représenté par la fonction de transfert  $F(z)$ . L'hypothèse importante est que la commande (sortie du ordinateur) et la mesure (entrée du ordinateur) sont échantillonnées de manière synchrone à la même période  $T_e$ .

Le choix judicieux de la fréquence d'échantillonnage est très important. En pratique on choisit cette fréquence en fonction de la bande passante du système en **boucle fermée** :

$$f_e = (6 \text{ à } 25) f^{BF} \quad (61)$$



### 7.2.2 Discrétisation d'un correcteur continu

La première méthode pour obtenir les équations d'un correcteur numérique est de faire sa synthèse avec les outils de l'automatique continue puis de discrétiser la fonction de transfert du correcteur continu obtenu (voir pour cela le chapitre 4.2 *Echantillonnage des systèmes continus*).

On va voir par la suite que la synthèse directe d'un correcteur discret offre plus de degré de liberté et permet en général de remplir un cahier des charges plus complet.

## 7.3 Régulateur PID numérique

Le régulateur le plus populaire est le régulateur PID. Nous allons voir dans un premier temps comment on génère un régulateur PID à partir des équations bien connues du régulateur PID continu et comment ces équations peuvent être généralisées pour obtenir plus de degrés de liberté pour le réglage.

### 7.3.1 Cas 1

Un régulateur PID continu a une fonction de transfert du type :

$$K_{\text{PID}}(s) = K \cdot \left( 1 + \frac{1}{T_i \cdot s} + \frac{T_d \cdot s}{1 + \frac{T_d}{N} \cdot s} \right) \quad (62)$$

La discrétisation de ce correcteur continu par la méthode d'Euler (Tableau 4 page 31) se fait en remplaçant la variable de Laplace  $s$  par le terme  $\frac{1-q^{-1}}{q^{-1} \cdot T_e}$ . On obtient :

$$K_{\text{PID}}(q) = K \cdot \left( 1 + \frac{1}{T_i \cdot \frac{1-q^{-1}}{T_e}} + \frac{T_d \cdot \frac{1-q^{-1}}{T_e}}{1 + \frac{T_d}{N} \cdot \frac{1-q^{-1}}{T_e}} \right) \quad (63)$$

En réarrangeant on obtient la forme générale RST (Figure 30):

$$K_{PID}(q) = K \cdot \frac{r_0 + r_1 \cdot q^{-1} + r_2 \cdot q^{-2}}{(1 - q^{-1}) \cdot (1 + s_1' \cdot q^{-1})}$$

$$\begin{cases} r_0 = K \cdot \left( 1 + \frac{T_e}{T_i} + \frac{N \cdot T_d}{T_d + N \cdot T_e} \right) \\ r_1 = -K \cdot \left( 1 + \frac{T_d}{T_d + N \cdot T_e} + \frac{T_e}{T_i} \cdot \frac{T_d}{T_d + N \cdot T_e} + \frac{N \cdot T_d}{T_d + N \cdot T_e} \right) \\ r_2 = K \cdot \left( \frac{T_d}{T_d + N \cdot T_e} \right) \\ s_1' = \frac{T_d}{T_d + N \cdot T_e} \end{cases} \quad (64)$$

Comme on le voit le correcteur obtenu comporte autant de degrés de liberté que le correcteur PID analogique. On reconnaît au dénominateur le terme  $(1 - q^{-1})$  en facteur qui assure l'effet d'intégration.

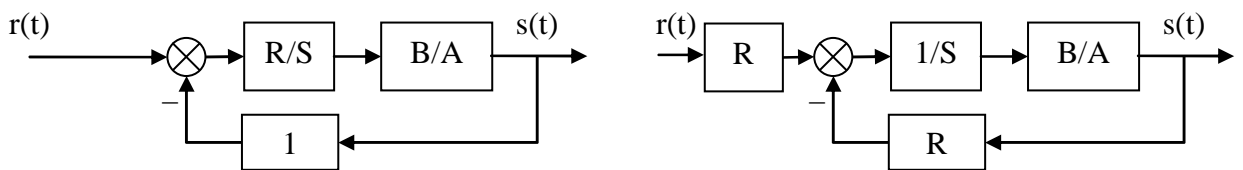


Figure 30 : PID sous forme RST

La fonction de transfert en boucle fermée obtenue est donnée par l'équation suivante :

$$H_{BF} = \frac{B \cdot R}{A \cdot S + B \cdot R} = \frac{B \cdot R}{P} \quad (65)$$

De manière générale le polynôme caractéristique  $P$  fixe les performances visées en boucle fermée. Le calcul de  $R$  et  $S$ ,  $P$  étant fixé et  $A$  et  $B$  étant connus, est réalisé en résolvant l'équation de Bézout  $P = A \cdot S + B \cdot R$ . (voir 9.1 Annexe 1 : Résolution de l'équation polynomiale de Bezout)

On voit que le numérateur  $B \cdot R$  est imposé une fois que  $R$  et  $S$  sont trouvés. Le régulateur introduit donc naturellement des zéros dans la boucle fermée.

Traisons l'exemple entièrement dans le cas où le système est modélisé comme un système du deuxième ordre avec faible retard. La fonction de transfert du procédé est donnée par :

$$H_{BO} = \frac{B}{A} = \frac{b_1 \cdot q^{-1} + b_2 \cdot q^{-2}}{1 + a_1 \cdot q^{-1} + a_2 \cdot q^{-2}} \quad (66)$$

Le correcteur PID rajoutant deux pôles, le polynôme caractéristique en boucle fermée spécifié doit posséder quatre pôles. Le polynôme caractéristique en boucle fermée doit donc être de la forme :

$$\begin{aligned} P &= 1 + p_1 \cdot q^{-1} + p_2 \cdot q^{-2} + p_2 \cdot q^{-3} + p_2 \cdot q^{-4} \\ P &= (1 + p_1' \cdot q^{-1} + p_2' \cdot q^{-2}) \cdot (1 + \alpha_1 \cdot q^{-1}) \cdot (1 + \alpha_2 \cdot q^{-1}) \end{aligned} \quad (67)$$

Et il doit être identifié au produit  $AS + BR$  avec  $R$  et  $S$  ayant les formes données par (64) :

$$\begin{aligned} &(1 + p_1' \cdot q^{-1} + p_2' \cdot q^{-2}) \cdot (1 + \alpha_1 \cdot q^{-1}) \cdot (1 + \alpha_2 \cdot q^{-1}) \\ &= (1 + a_1 \cdot q^{-1} + a_2 \cdot q^{-2}) \cdot (1 - q^{-1}) \cdot (1 + s_1' \cdot q^{-1}) \\ &+ (b_1 \cdot q^{-1} + b_2 \cdot q^{-2}) \cdot (r_0 + r_1 \cdot q^{-1} + r_2 \cdot q^{-2}) \end{aligned} \quad (68)$$

Les coefficients  $p_1'$  et  $p_2'$  définissent un modèle du second ordre aux caractéristiques (pulsation propre  $\omega_0$  et amortissement  $\sigma$  choisis et coefficients déduits conformément au Tableau 3 page 25). Les coefficients  $\alpha_1$  et  $\alpha_2$  correspondent à des pôles auxiliaires aperiodiques. Les choix typiques pour ces coefficients sont :

$$-0.05 < \alpha_1, \alpha_2 < -0.5 \quad (69)$$

On montre que la résolution de (68) est réalisée par une inversion matricielle simple :

$$M \cdot x = p \quad (70)$$

avec

$$\begin{cases} x^T = [1 & s_1' & r_0 & r_1 & r_2] \\ p^T = [1 & p_1 & p_2 & p_3 & p_4] \end{cases}$$

et

$$M = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ a_1' & 1 & b_1 & 0 & 0 \\ a_2' & a_1' & b_2 & b_1 & 0 \\ a_3' & a_2' & 0 & b_2 & b_1 \\ 0 & a_3' & 0 & 0 & b_2 \end{pmatrix} \quad (71)$$

On montre que M est inversible à la condition que A et B soient premiers entre eux, c'est-à-dire que A et B n'aient pas de pôles et de zéros en commun.

(Le problème peut aussi être résolu en utilisant la résolution de l'équation de Bézout).

### 7.3.2 Cas 2

Comme on l'a vu le correcteur PI numérique directement transposé du correcteur PID analogique décrit dans le chapitre 7.3.1 a l'inconvénient de ramener des zéros au dénominateur de la fonction de transfert en boucle fermée (équation 65). Il est possible, grâce à la flexibilité du régulateur numérique d'imposer au correcteur de ne pas ajouter de zéros en boucle fermée. La fonction de transfert désirée en boucle fermée sera donc de la forme :

$$H_{BF} = \frac{P(1)}{B(1)} \cdot \frac{B}{P} \quad (72)$$

Le facteur  $P(1)/B(1)$  garantit le gain statique unitaire du système en boucle fermée (voir page 12). Si le régulateur a la forme générale RST alors les polynômes R S et T doivent vérifier :

$$H_{BF} = \frac{B \cdot T}{A \cdot S + B \cdot R} = \frac{P(1) B}{B(1) P} \quad (73)$$

S et R sont obtenus par la résolution de la même équation que précédemment (68). Le polynôme T est obtenu par simple identification :  $T = R(1)$ .

La structure, sous forme de schéma-bloc est donnée ci-dessous :

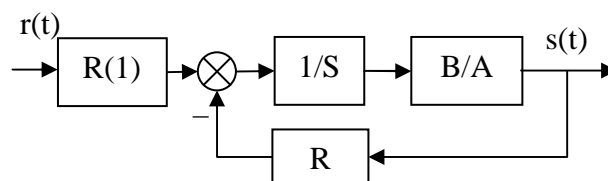


Figure 31 : PID2 sous forme RST

On montre que cette structure de PID, ramené en continu, correspond au PID canonique pour l'action intégrale. L'action proportionnelle et dérivée n'est appliquée que sur le signal y.

La structure du PID2 par rapport au PID1 a pour avantage de ne pas ramener de zéros à la boucle fermée. Cela se traduit par un dépassement moindre en transitoire lorsqu'on applique un échelon de commande. Par contre les deux systèmes réagissent de la même manière à un échelon de perturbation.

## 7.4 Placement de pôles

Les deux structures de PID décrites dans le chapitre précédent sont deux cas particulier d'une stratégie plus générale de placement de pôles.

La synthèse est plus générale que les PID présentés précédemment puisqu'il n'est pas nécessaire de faire d'hypothèses sur le degré de A et B, sur le retard, ni sur la présence ou non de zéros stables ou instables dans le procédé.

La structure du correcteur est la structure type RST telle que présentée dans la Figure 29. Pour la suite des calculs on posera pour la fonction de transfert les conventions suivantes :

$$H_{BO} = \frac{q^{-d} \cdot B(q^{-1})}{A(q^{-1})} \quad (74)$$

$$\begin{cases} A(q^{-1}) = 1 + a_1 \cdot q^{-1} + \dots + a_{n_A} \cdot q^{-n_A} \\ B(q^{-1}) = b_1 \cdot q^{-1} + \dots + b_{n_B} \cdot q^{-n_B} = q^{-1} \cdot B^*(q^{-1}) \end{cases}$$

Compte tenu de la forme RST du régulateur la fonction de transfert en boucle fermée vérifie :

$$H_{BF} = \frac{q^{-d} \cdot B(q^{-1}) \cdot T(q^{-1})}{P(q^{-1})} \quad (75)$$

$$P(q^{-1}) = A(q^{-1}) \cdot S(q^{-1}) + q^{-d} \cdot B(q^{-1}) \cdot R(q^{-1}) = 1 + p_1 \cdot q^{-1} + \dots + p_{n_p} \cdot q^{-n_p}$$

Le polynôme  $P(q^{-1})$  définit donc les pôles de la fonction de transfert en boucle fermée et traduit donc le comportement en *poursuite*.

De la même manière la fonction de transfert entre une perturbation et la sortie (fonction de sensibilité perturbation-sortie) est donnée par :

$$S_{yp} = \frac{A(q^{-1}) \cdot S(q^{-1})}{P(q^{-1})} \quad (76)$$

Ce même polynôme traduit donc aussi le comportement en *régulation*.

Les consignes en poursuite et en régulation peuvent être différentes comme cela est représenté par exemple sur la Figure 32.

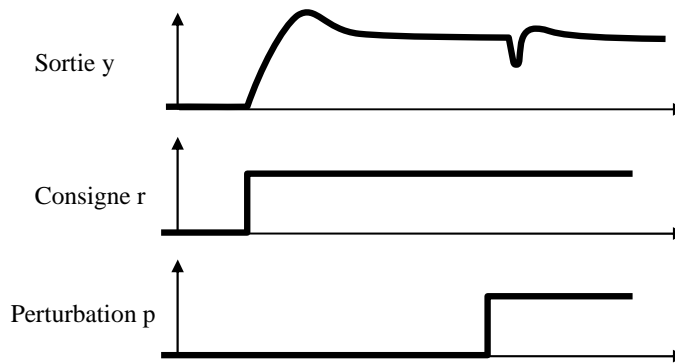


Figure 32 : Performances différentes en régulation et en poursuite

### 7.4.1 Choix des pôles en boucle fermée

De manière générale le comportement en boucle fermée est spécifié sous la forme d'une pulsation propre  $\omega_0$  et d'un amortissement  $\sigma$ , donc une fonction de transfert du second ordre. La spécification étant généralement donnée en continu il est nécessaire de traduire la valeur désirée de  $\omega_0$  et  $\sigma$  en un polynôme  $P_D(q^{-1})$  par discrétisation de la fonction de transfert continue (chapitre 4.5 page 29). Ce polynôme  $P_D$  traduit le comportement dominant du système bouclé.

Si l'on souhaite introduire un effet de filtrage supplémentaire dans certaines bandes de fréquence (réduction des bruits de mesure, adoucissement de la commande, robustesse) cela est décrit par un polynôme  $P_F$  comportant des pôles auxiliaires, plus rapides que les pôles de  $P_D$ .

Le polynôme caractéristique en boucle fermée est donc constitué de pôles dominants et de pôles auxiliaires :

$$P(q^{-1}) = P_D(q^{-1}) \cdot P_F(q^{-1}) \quad (77)$$

### 7.4.2 Régulation : calcul de R et S

Le polynôme  $P(q^{-1})$  étant fixé il s'agit de trouver R et S vérifiant :

$$P(q^{-1}) = A(q^{-1}) \cdot S(q^{-1}) + q^{-d} \cdot B(q^{-1}) \cdot R(q^{-1}) \quad (78)$$

Comme déjà vu au chapitre 7.3.1 on obtient une équation de Bézout qu'il s'agit de résoudre. (Voir aussi 9.1 Annexe 1 : Résolution de l'équation polynomiale de Bezout)

Il est parfois nécessaire de spécifier R et S avec des parties fixes  $H_R$  et  $H_S$ . Dans ce cas l'équation (78) est modifiée et on l'écrit sous la forme:

$$P(q^{-1}) = (A(q^{-1}) \cdot H_S(q^{-1})) \cdot S'(q^{-1}) + (q^{-d} \cdot B(q^{-1}) \cdot H_R(q^{-1})) \cdot R'(q^{-1}) \quad (79)$$

Les inconnues que l'on souhaite trouver sont  $S'$  et  $R'$ . Ceci est nécessaire dans les cas particuliers suivants :

- spécification d'erreur statique nulle,
- rejet d'une perturbation harmonique,
- blocage d'un signal,

Détaillons maintenant ces cas particuliers.

### Spécification d'une erreur statique nulle

On retrouve la nécessité d'avoir un terme intégrateur dans la boucle directe. La spécification se retrouve donc sur  $S$  :

$$S(q^{-1}) = (1 - q^{-1}) \cdot S'(q^{-1}) = H_S(q^{-1}) \cdot S'(q^{-1}) \quad (80)$$

### Rejet d'une perturbation harmonique

Si le système est perturbé par une perturbation harmonique à une fréquence fixe (par exemple 50Hz...) il est possible d'imposer à  $S$  d'avoir un module *nul* ou amorti à cette fréquence. Cela se traduit par la présence dans  $S$  d'une paire de zéros complexe à amortissement nul ou faible à cette fréquence :

$$S(q^{-1}) = (1 + \alpha \cdot q^{-1} + \beta \cdot q^{-2}) \cdot S'(q^{-1}) = H_S(q^{-1}) \cdot S'(q^{-1}) \quad (81)$$

### Blocage d'un signal

Pour certaines application on souhaite que le régulateur "laisse passer" certaines composantes du signal de sortie, c'est-à-dire qu'il ne réagisse pas à ces composantes. Cela se traduit par  $R=0$  à ces fréquences. On retrouve la même forme que précédemment :

$$R(q^{-1}) = (1 + \alpha \cdot q^{-1} + \beta \cdot q^{-2}) \cdot R'(q^{-1}) = H_R(q^{-1}) \cdot R'(q^{-1}) \quad (82)$$

### 7.4.3 Poursuite : calcul de T

Le schéma-bloc de l'asservissement est donné sur la Figure 33. Le terme  $T$  est décomposé en un terme  $T'$  et un terme  $B^*/A^*$ .

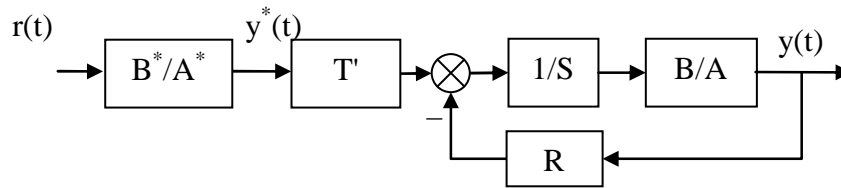


Figure 33 : placement de pôle

L'idée est de régler  $B^*$  et  $A^*$  de telle manière que le signal  $y^*$  ait la dynamique de poursuite désirée (par exemple un second ordre avec pulsation propre et amortissement donné) puis de régler  $T'$  de telle manière que le transfert entre  $y^*$  et  $y$  soit le plus "transparent" possible.

### Réglage de $T'$

Le transfert entre  $y^*$  et  $y$  est :

$$S_{yy^*}(q^{-1}) = T'(q^{-1}) \frac{q^{-d} \cdot B(q^{-1})}{P(q^{-1})} \quad (83)$$

Pour assurer le transfert le plus "transparent" possible il faut donc choisir

$$T'(q^{-1}) = \frac{P(q^{-1})}{B(1)} \quad (84)$$

Remarques :

- Si  $B(1)=0$  alors on remplace  $B(1)$  par 1 dans (84)
- De même dans (84)  $P(q^{-1})$  peut être remplacé par  $P_D$  correspondant à la dynamique dominante.

### Réglage de $B^*$ et $A^*$

$B^*$  et  $A^*$  sont choisis de telle manière que la fonction de transfert  $B^*/A^*$  corresponde à la dynamique de poursuite souhaitée.

## 7.5 Poursuite et régulation à objectifs indépendants

Cette méthode ressemble à la méthode de placement de pôles à ceci près que l'on va effectuer une simplification des zéros de  $B^*(q^{-1})$  : cela ne s'applique donc qu'aux systèmes avec des pôles stables. Plus précisément il faut qu'ils aient un amortissement suffisant ( $> 0,2$ ) de manière à ne pas être trop près du cercle unité.

La structure de l'asservissement est toujours celle donnée sur la Figure 33.



**Choix de P**

$P(q^{-1})$  est choisi de manière à simplifier  $B^*(q^{-1})$  (rappel : la forme de B est donnée dans l'équation 74). Comme indiqué dans le chapitre 7.4.2 page 60 il s'agit donc d'imposer les zéros de  $B(q^{-1})$  dans  $H_R(q^{-1})$  et  $H_S(q^{-1})$ .

**Choix de T**

Si l'on choisit  $T(q^{-1}) = P(q^{-1})$  alors le transfert entre  $y^*$  et  $y$  est :

$$S_{yy^*}(q^{-1}) = q^{-(d+1)} \quad (85)$$

**Réglage de  $B^*$  et  $A^*$** 

$B^*$  et  $A^*$  sont choisis de telle manière que la fonction de transfert  $B^*/A^*$  corresponde à la dynamique de poursuite souhaitée.



## 8. TECHNOLOGIE DE LA COMMANDE PAR ORDINATEUR

Du point de vue de l'automaticien le ordinateur est relié au système physique à piloter par un certain nombre d'intermédiaires plus ou moins gênants :

- les interfaces analogiques et numériques,
- le transport de l'information par bus et protocole,
- l'utilisation éventuelle de capteurs ou d'actionneurs « intelligents ».

Il ne s'agit pas ici de faire un exposé complet sur ces trois aspects mais de poser les problèmes qui se posent à l'automaticien lorsque le système qu'il doit commander est perturbé par ces éléments.

Enfin ce chapitre se termine sur les notions de « Hardware In the Loop » et de « prototypage rapide » qui sont partie intégrante des processus de réalisation de systèmes de contrôle.

### 8.1 Calculateurs et interfaces numériques / analogiques

Le premier problème à prendre en compte est la quantification de l'information. Du point de vue du ordinateur d'une part et du point de vue des convertisseurs d'autre part. Dans un premier temps, considérons les aspects performance de calcul lié à la nature du codage des nombres dans le ordinateur.

#### Les performances liées au types de variables

Prenons l'exemple où le concepteur de la loi de commande souhaite réaliser un correcteur de type proportionnel. Entre l'acquisition des mesures et l'écriture de la consigne il y aura la réalisation d'une de commande du type :

$$y = K \cdot (\text{consigne} - \text{mesure}) \quad (86)$$

En utilisant un langage de programmation évolué (par exemple le C) la ligne de code sera la transcription littérale de (86). Le programmeur devra décider de la définition et du type des variables, principalement du nombre d'octets utilisés pour stocker ses variables. Le Tableau 5 résume par exemple les types utilisés en langage C (norme ANSI)

Tableau 5 : Types ANSI C

Types entiers	char	8 bits
	short int	16 bits
	long int	32 bits
Types flottants (mantisse et exposant)	float	32 bits
	double	64 bits
	longdouble	128 bits

Une fois le code écrit en C l'exécution du programme sur un processeur cible ne peut se faire que par l'intermédiaire d'une compilation du programme. A chaque processeur est donc adapté une version de compilateur.

Les performances du programme (sa rapidité d'exécution) dépendront donc en partie de la qualité du compilateur. (Par exemple, une multiplication s'effectue t'elle bêtement par un décalage de tous les bits ou bien des solutions astucieuses sont-elles implémentées pour arrêter le décalage selon le contexte ?)

Le principal déterminant de la performance est cependant le processeur : son type (8 bits, 16 bits ou 64 bits) et son cadencement (en MHz ou GHz) sont les deux plus importants paramètres. Coder les variables sur 64 bits (double) dans le programme coûtera cher en temps de calcul si le processeur est un 8 bits. Le type de mémoire et leurs temps d'accès peuvent aussi être déterminants. Enfin, certains processeurs disposent d'un processeur spécifiquement dédié aux opérations mathématiques (coprocesseur mathématique).

Il n'est pas possible de dire exactement, pour tel type de compilateur associé à tel type de processeur, le temps que prendra par exemple une multiplication. En pratique on effectue une série de tests types (benchmark) et on affiche un temps moyen d'exécution.

En résumé, on peut être tenté, pour ne pas s'embêter, de coder toutes les variables en virgule flottante. Mais il faut bien savoir que cela a un coût en terme de temps de calcul, surtout si le processeur n'est pas adapté. Pour fixer les idées on peut dire qu'il y a un rapport 20 entre une addition réalisée en "short" et une addition réalisée en virgule flottante.

Enfin, l'utilisation de fonctions mathématiques (sinus, exponentielles, etc...) peut aussi avoir un coût très lourd si le travail n'est pas réalisé par un coprocesseur mathématique dédié. Le compilateur et sa librairie mathématique joue là encore un grand rôle.

### La quantification imposée par le processeur

Au-delà de l'aspect performance et temps de calcul, la quantification de l'information doit parfois aussi être prise en compte. Rappelons qu'un entier codé sur N bits peut prendre seulement  $2^N - 1$  valeurs différentes. Avant d'effectuer les calculs mathématiques dans le corps du programme il est donc judicieux d'appliquer un

facteur d'échelle de manière à utiliser l'amplitude maximale possible (entre 0 et  $2^N - 1$ ).

### **La quantification imposée par les conversions analogique et numérique**

Les convertisseurs analogique-numérique et numérique-analogique sont principalement caractérisés par leur amplitude de signal analogique d'entrée (généralement en volts, par exemple +/- 10V) et par leur résolution en bits (8 ou 10 bits pour les plus courants, jusqu'à 16 bits pour les convertisseurs haut de gamme).

Prenons l'exemple d'un capteur gyromètre transformant une vitesse de rotation en tension. Compte tenu de l'application le signal utile varie entre 2,4V et 2,6V. Si le signal mesuré est branché directement sur un convertisseur 0-10V 10bits alors un problème de conditionnement du signal se pose. En effet la quantification est de  $\frac{10V}{2^{10}-1} = 0,098V$ . Les 0,2V de signal utile seront donc discrétisés en  $\frac{0,2}{0,098} = 20$  intervalles seulement !

Si la discrétisation du signal utile est trop grossière cela peut se traduire par une diminution des performances de l'asservissement, voire une instabilité. Le système n'est en effet plus linéaire et il n'existe pas d'outils théoriques permettant de prévoir les performances du système bouclé.

## **8.2 Transport de l'information : les bus et les protocoles**

Pas question de faire ni même d'ébaucher un cours sur les bus et les protocoles ici. Mais on va quand même donner quelques indications. L'automaticien a l'habitude de simplifier le problème qu'il se pose en dessinant un schéma simplifié de son problème d'asservissement (du type de la Figure 20 page 45) pour pouvoir appliquer les théories de la commande. Cependant il arrive bien souvent que l'information transite à l'intérieur du système, soit entre différents ordinateurs effectuant différentes parties du travail soit entre ordinateur et périphériques, par l'intermédiaire de bus informatique. A chaque type de bus correspond un ou plusieurs protocoles qui peuvent poser des problèmes pour l'automaticien :

- les données sont-elles discrétisées ? (le protocole peut par exemple ne permettre de transmettre que des paquets de données de 16 bits),
- le bus induit-t'il un retard ? (le composant mettant en forme les données du ordinateur pour les publier sur le bus a certaines performances qui peuvent induire un délai)
- le bus est-il déterministe ? (Certains bus ne garantissent pas que les données arriveront en un temps donné),
- le bus est-il sûr ? (Certains bus ne garantissent pas non plus que les données arriveront bien)

### **8.3 Capteurs et actionneurs "intelligents", "smart sensors"**

De plus en plus souvent les capteurs et les actionneurs sont associés à leur électronique de puissance, électronique de commande et de conditionnement de signal. L'interface avec le capteur ou l'actionneur se fait donc via un protocole de communication (par exemple via un bus RS232).

L'intérêt est de véhiculer non pas des signaux analogiques, tensions ou courants, mais des signaux numériques beaucoup moins sensibles aux éventuelles perturbations.

On retrouve donc, entre le capteur ou l'actionneur et le ordinateur, la présence d'un bus et d'un protocole de communication.

### **8.4 La philosophie "hardware in the loop"**

Les systèmes embarqués sont de plus en plus complexes et touchent des domaines de plus en plus variés : automobile, satellites, aéronautique, UAV, systèmes d'arme, sous-marins, systèmes, etc... Les systèmes à piloter ainsi que les systèmes de commande sont de plus en plus complexes.

Les techniques de simulation « Hardware In the Loop » (HIL) sont de plus en plus utilisées dans le développement et dans le test de ces systèmes embarqués. L'idée est de développer entièrement le système de contrôle sur une plateforme de test en utilisant le maximum de composants réels et en ne remplaçant que le minimum par la simulation. Par exemple en aéronautique l'« Iron Bird » regroupe dans un hangar tous les systèmes réels de l'avion en cours de développement. Seule la mécanique du vol est simulée : les sorties des actionneurs sont mesurées, rentrées dans un modèle de simulation numérique, puis les mesures effectuées normalement par les capteurs sont calculées et restituées par le modèle.

L'intérêt est de diminuer de manière importante les coûts et les délais en économisant le plus grand nombre possible de tests sur le système réel en situation. Lors des premiers tests réels les techniques « Hardware In the Loop » permettent d'avoir effectué la plus grande part du développement des lois de commande.

### **8.5 Le prototypage rapide en automatique**

Un peu dans le même ordre d'idée que le « Hardware In the Loop » le prototypage rapide permet de réaliser la conception des lois de commande en simulation puis de transporter rapidement (en un simple click !) le code du contrôleur simulé vers la cible temps réel implémentée sur le système réel.

La société Mathworks [9] investit massivement dans ce domaine en ayant enrichi son logiciel de calcul scientifique (Matlab) avec un logiciel de simulation graphique très évolué (Simulink) puis en développant des outils de prototypage rapide associés (Real Time Workshop, Embedded Target For...).

## 9. ANNEXE

### 9.1 Annexe 1 : Résolution de l'équation polynomiale de Bezout

Il s'agit de trouver R et S vérifiant :

$$P(q^{-1}) = A(q^{-1}).S(q^{-1}) + B(q^{-1}).R(q^{-1})$$

Où P, A et B sont connus.

Dans le cas où P, A, B, R et S sont des nombres entiers la solution de l'algorithme est donnée par l'algorithme de division Euclidienne étendu [8].

Dans le cas où P, A, B, R et S sont des polynômes, l'algorithme est donné ci-dessous, programmé en matlab ([1]).

```
function [Rp,Sp,nrp,nsp]=bezoutd(A,B,Hs,Hr,P)
%function [Rp,Sp,nrp,nsp]=bezoutd(A,B,Hs,Hr,P)
%solves AHsSp+BHRp=P by coefficient comparison. Delay and discretization delay need to be integrated in B
%(zeros at the beginning).
%
%inputs:
%A=[a0 a1 ... aNa] ... vector of model denominator coefficients A=a0 + a1z^(-1) + a2z^(-2) +...+ aNaz^(-Na)
%B=[b0 b1 ... bNb] ... vector of model numerator coefficients B=b0 + b1z^(-1) + b2z^(-2) +...+ bNbz^(-Nb)
%Hs=[hs0 hs1 ... hsNhs] ... vector of controller denominator fixed part Hs=hs0 + hs1z^(-1) +...+ hsNhsz^(-Nhs)
%Hr=[hr0 hr1 ... hrNhr] ... vector of controller denominator fixed part Hr=hr0 + hr1z^(-1) +...+ hrNhrz^(-Nhr)
%P=[p0 p1 ... pNp] ... vector of desired polynomial coefficients P=p0 + p1z^(-1) + p2z^(-2) +...+ pNpz^(-Np)
%outputs:
%Rp=[rp0 rp1 rp2 ...] ... vector of coefficients for resulted controller numerator
%Sp=[sp0 sp1 sp2 ...] ... vector of coefficients for resulted controller denominator
%nrp ... order of Rp
%nsp ... order of Sp
%
%written by: J. Langer, I.D. Landau, H. Prochazka
%7th june 2002

PRECISION=1e-16;

D=size(A);
if D(1)>1, A=A'; end;
```

```

D=size(B);
if D(1)>1, B=B'; end;
D=size(Hs);
if D(1)>1, Hs=Hs'; end;
if D(1)==0, Hs=1; end;
D=size(Hr);
if D(1)>1, Hr=Hr'; end;
if D(1)==0, Hr=1;end;
D=size(P);
if D(1)>1, P=P'; end;

na=length(A)-1;
nb=length(B)-1;
np=length(P)-1;
nhs=length(Hs)-1;
nhr=length(Hr)-1;

if (nhs>0), Ah=real(convz(A,Hs)); else Ah=A*Hs; end; % Ah = A * Hs
nah=length(Ah) -1;
if (nhr>0), Bh=real(convz(B,Hr)); else Bh=B*Hr; end;
nbh=length(Bh) -1;
if (np>nah+nbh-1), disp('Bezout error: too many poles');end;

% increase size of P using zeros if necessary
if (np<nah+nbh-1),
%
% set remaining poles onto a circle with radius rmin
P=[P zeros(1,nah+nbh-1-np)];
rootsPdes=roots(P);
nextra=nah+nbh-1-np;
rmin=1e-16;
angle=[0:nextra-1]'/nextra*2*pi;
j=sqrt(-1);
rootsPextra=rmin*exp(j*angle);
P=poly([rootsPdes;rootsPextra]);
np=nah+nbh-1;

end;
P,

nsp=nbh-1;
nrp=nah-1;

%matrix is smaller than vector PD

```



```

if (np>nah+nbh-1),
    disp('Order of model denominator is too low! Add a polynom of higher order to Hs or Hr. ');
end;

% ns=nsp+nhs
% nr=nrp+nhr

M=[];
for j=1:nsp+1,
    V=[];
    if (j>1), V=[V ; zeros(j-1,1)]; end;% zeros in front of Ah
    V=[V ; Ah'];% Ah
    if (j<=nsp), V=[V ; zeros(nsp+1-j,1)]; end;% zeros behind Ah
    if (length(V)~=nah+nbh), disp('bezoutb: error V'); end;
    M=[M V]; % add one column to M
end;

for j=1:nrp+1,
    V=[];
    if (j>1), V=[V ; zeros(j-1,1)]; end;
    V=[V ; Bh'];
    if (j<=nrp), V=[V ; zeros(nrp+1-j,1)]; end;
    if (length(V)~=nah+nbh), disp('bezoutb: error V'); end;
    M=[M V];
end;

D=size(M);
if (D(1)~=nah+nbh), disp('bezoutb: error size M row'); end;
if (D(2)~=nah+nbh), disp('bezoutb: error size M column'); end;

% make P column vector
P=P';

global M1;
M1=M;

X= M\P;
% coefficients are real values
X=real(X);

% make X row vector
X=X';

```

## Commande des systèmes par ordinateur

```
Sp=X(1:nsp+1);  
Rp=X(nsp+2:nsp+nrp+2);
```

## 10. BIBLIOGRAPHIE

- [1] I. D. Landau, "*Commande des systèmes*", Edition Lavoisier 2002.
- [2] C. Nouals, "Théorie du Signal", Notes de cours ENSICA
- [3] F. Vincent, "*Traitement Numérique du Signal*", Notes de cours ENSICA.
- [4] D. Arzelier, "*Représentation et analyse des systèmes, notes de cours*", Edition ENSICA. Voir aussi <http://www.laas.fr/~arzelier/>
- [5] J. Bordeneuve-Guibé, "*Commande des systèmes linéaires*", Notes de cours ENSICA.
- [6] Y. Briere, "*Identification*", Notes de cours ENSICA.
- [7] L. Ljung, "*System Identification, Theory for the User*", Prentice Hall 1987.
- [8] Wikipedia , équation de Bezout : [http://en.wikipedia.org/wiki/Bézout's\\_identity](http://en.wikipedia.org/wiki/Bézout's_identity)
- [9] <http://www.mathworks.com>



# 11. INDEX

<b>A</b>	
actionneur intelligent .....	21
amortissement .....	15
ANSI C .....	60
Approximation numérique d'équation différentielle .....	28
ARMAX .....	35
<b>B</b>	
BJ35 .....	
Blocage d'un signal .....	56
bloqueurs d'ordre 0 .....	22
BOZ .....	22
bus.....	21, 61
<b>C</b>	
CAN.....	21
capteur intelligent .....	21
causal .....	6, 9
Causalité .....	10
CNA.....	21
compilateur .....	60
convertisseur analogique / numérique.....	21
convertisseur numérique / analogique.....	21
convertisseurs analogique-numérique .....	61
coprocesseur mathématique .....	60
Correspondances pôles - zéros.....	31
<b>D</b>	
Diagramme de Bode .....	18
Discrétisation de fonction de transfert continues .....	27
division polynomiale.....	6
<b>E</b>	
échelon unitaire.....	11
équation d'état .....	17
Erreur statique.....	42
Estimation de la réponse fréquentielle .....	39
Estimation de la réponse impulsionnelle.....	38
Euler .....	29
<b>F</b>	
facteur d'échelle .....	61
facteur d'oubli .....	36
<i>fenêtre de Hamming</i> .....	39
filtre anti-repliement .....	25
fonction de sensibilité .....	42
fonction de transfert discrète.....	9
<b>G</b>	
gain statique.....	11
gain statique unitaire.....	42
<b>I</b>	
Identification.....	32
Identification non paramétrique .....	38
intercorrélation.....	38
<b>L</b>	
lieu des racines.....	43
lieu d'Evans.....	43
LTI.....	8
<b>M</b>	
marge de gain.....	47
marge de module.....	46
marge de phase .....	47
méthode des moindres carrés récursifs .....	32
Méthode des trapèzes.....	29
modèle ARX .....	32
<b>O</b>	
OE.....	35
one step ahead prediction.....	35
opérateur de retard .....	9
<b>P</b>	
période d'échantillonnage (choix) .....	26
Placement de pôles .....	54
Pôles .....	11
Poursuite .....	56
<i>prédiction</i> .....	32
protocoles.....	61
Pseudo Random Binary Signal .....	34
pulsation .....	15
<b>Q</b>	
quantification .....	23, 24, 60
<b>R</b>	
R,S,T.....	41
rapidité.....	15
rayon de convergence .....	5, 9
Régulation.....	55

Rejet de perturbations .....	42
Rejet d'une perturbation harmonique .....	56
réponse à un échelon.....	11
réponse fréquentielle.....	18
Réponse impulsionnelle.....	10
Réponses indicielle .....	10
retard.....	23
root locus .....	43

### **S**

signal discret .....	5
smart sensors.....	62
stabilité.....	11
Systèmes du premier ordre.....	14
Systèmes du second ordre.....	15

### **T**

Théorème de Nyquist.....	46
théorème de Shannon.....	19, 24
théorème du retard .....	6
Transformée en z .....	5
transformée unilatérale .....	5
Tustin.....	29

### **V**

virgule flottante.....	60
------------------------	----

### **Z**

zero holder hold .....	23
zéros.....	11
zoh .....	23

# Commande des systèmes par ordinateur

Yves Briere  
ENSICA

Notes de cours  
2007





# TABLE DES MATIERES

Table des matières .....	1
1. Introduction .....	1
2. Rappels et prérequis .....	3
3. Signaux et systèmes discrets .....	5
3.1 Signaux discrets .....	5
3.1.1 Transformée en z .....	5
3.1.2 Signaux types .....	7
3.2 Systèmes discrets .....	9
3.2.1 Fonction de transfert discrète, définition .....	9
3.2.2 Notation "opérateur de retard $q^{-1}$ " .....	10
3.2.3 Réponse impulsionnelle .....	11
3.2.4 Causalité .....	11
3.2.5 Réponses indicielle .....	11
3.2.6 Régime stationnaire (gain statique) .....	12
3.2.7 Pôles, zéros, stabilité .....	12
3.2.8 Systèmes du premier ordre .....	15
3.2.9 Systèmes du second ordre .....	16
3.2.10 Calcul itératif de la réponse d'un système .....	18
3.2.11 Représentation par équation d'état .....	18
3.3 Réponse fréquentielle des systèmes échantillonnés .....	19
3.3.1 Diagramme de Bode d'un système échantillonné .....	19
3.3.2 Où l'on voit apparaître le théorème de Shannon .....	20
4. Systèmes continus échantillonnés .....	23
4.1 Structure des systèmes commandés par ordinateur .....	23
4.2 Échantillonnage des systèmes continus .....	24
4.3 Effets de l'échantillonnage .....	25
4.4 Où l'on revoit le théorème de Shannon .....	26
4.5 Discrétisation de fonction de transfert continues .....	29
4.5.1 Approximation numérique d'équation différentielle .....	30
4.5.2 Superposition des réponses indicielles .....	32
4.5.3 Correspondances pôles - zéros .....	33
4.5.4 Bilan .....	33
5. Identification des systèmes échantillonnés .....	35
5.1 Identification d'un modèle ARX par la méthode des moindres carrés récurrents .....	35
5.2 L'identification paramétrique en général et en pratique .....	37

5.2.1	Choix du signal d'excitation, traitement des mesures .....	37
5.2.2	Choix d'une structure de modèle.....	38
5.2.3	Choix d'un critère à minimiser.....	38
5.2.4	Choix d'un algorithme.....	38
5.2.5	Validation.....	39
5.2.6	Utilisation de logiciels .....	39
5.3	Identification "simple" .....	39
5.4	Identification non paramétrique .....	41
5.4.1	Estimation de la réponse impulsionnelle .....	41
5.4.2	Estimation de la réponse fréquentielle.....	42
6.	Systèmes bouclés .....	45
6.1	Fonction de transfert en boucle fermée .....	45
6.2	Erreur statique .....	46
6.3	Rejet de perturbations.....	46
6.4	Stabilité des systèmes à simple gain de rétroaction.....	47
6.5	Analyse des systèmes bouclés dans le domaine fréquentiel : stabilité.....	49
6.5.1	Théorème de Nyquist dans le cas où le système est stable en boucle ouverte (critère du revers) .....	50
6.5.2	Théorème de Nyquist dans le cas où le système est instable en boucle ouverte .....	50
6.5.3	Interprétations du lieu de Nyquist.....	50
7.	Méthodes de régulation des systèmes échantillonnés .....	53
7.1	Introduction .....	53
7.2	Structure d'un système commandé par ordinateur, hypothèses.....	54
7.2.1	Discretisation, choix de la période d'échantillonnage.....	54
7.2.2	Discretisation d'un correcteur continu .....	55
7.3	Régulateur PID numérique .....	55
7.3.1	Cas 1 .....	55
7.3.2	Cas 2 .....	58
7.4	Placement de pôles .....	59
7.4.1	Choix des pôles en boucle fermée .....	60
7.4.2	Régulation : calcul de R et S.....	60
7.4.3	Poursuite : calcul de T .....	61
7.5	Poursuite et régulation à objectifs indépendants .....	62
8.	Technologie de la commande par ordinateur.....	65
8.1	Calculateurs et interfaces numériques / analogiques.....	65
8.2	Transport de l'information : les bus et les protocoles .....	67
8.3	Capteurs et actionneurs "intelligents", "smart sensors" .....	68
8.4	La philosophie "hardware in the loop" .....	68
8.5	Le prototypage rapide en automatique .....	68
9.	Annexe .....	69
9.1	Annexe 1 : Résolution de l'équation polynomiale de Bezout.....	69
10.	Bibliographie.....	73
11.	Index.....	1





# 1. INTRODUCTION



Figure 1 : commande d'un systèmes de télémédecine par ordinateur, vu par Claude Serre



## 2. RAPPELS ET PREREQUIS

Ce polycopié ne comporte pas de rappels sur les systèmes continus. S'adressant aux étudiants en deuxième année de l'ENSICA il suppose connu les enseignements de première année et de seconde année en automatique et traitement du signal.

En première année le cours de théorie du signal [2] a rendu familier les notions de :

- transformée de Fourier, transformée de Laplace
- corrélation, convolution

Le cours de traitement numérique du signal [3] a abordé en plus :

- la discrétisation et la quantification des signaux.
- la transformée en z et la transformée de Fourier discrète.
- les filtres numériques RIF et RII.

En automatique [4] les bases de la représentation et de l'analyse des systèmes continus ont été posées. On ne reviendra pas dans ce cours sur les notions de :

- modèles mathématique des systèmes linéaires à temps continus à temps invariant (LTI) : fonction de transfert, équation d'état, pôles, zéros et dynamique.
- commandabilité et observabilité.
- stabilité des systèmes en boucle ouverte : pôles, critères de Routh-Hurwitz.
- stabilité des systèmes bouclés : critère de Nyquist, critère de Bode.
- lieu des racines.

En deuxième année les notions suivantes sont acquises [5] :

- synthèse des correcteurs par méthodes fréquentielles (avance et retard de phase, PID)
- commande par retour d'état (placement de pôles), commande Linéaire Quadratique (LQG)
- reconstruction de l'état par Observateur, observateur optimal de Kalman.





## 3. SIGNAUX ET SYSTEMES DISCRETS

### 3.1 Signaux discrets

#### 3.1.1 Transformée en z

Pour une approche mathématiquement rigoureuse de représentation des signaux discrets par la transformée en z consulter [3] et sa bibliographie. Dans ce cours on se cantonne à une approche utilitariste des outils et méthodes mathématiques tout en essayant de ne pas contredire fondamentalement la rigueur mathématique.

Un signal discret  $s(k)$  est une séquence de nombres réels appelés échantillons. L'indice  $k$  est généralement associé à un instant d'échantillonnage :  $t = k.T_e$ . La période d'échantillonnage  $T_e$  est supposée constante. Par définition la transformée en z du signal  $s(k)$  est la fonction de la variable  $z$ , notée abusivement  $s(z)$  :

$$s(z) = Z(s(k)) = \sum_{k=0}^{\infty} s(k) \cdot z^{-k} \quad (1)$$

La transformée définie ici est plus précisément appelée transformée unilatérale puisque les termes correspondant à  $k < 0$  ne sont pas pris en compte. Ceci n'a pas d'importance puisque les signaux que nous utilisons sont systématiquement causaux<sup>1</sup> :  $s(k) = 0$  pour  $k < 0$ .

Il y a bien évidemment un lien avec la transformée de Laplace et la transformée de Fourier (voir [3])

La fonction de la variable complexe  $s(z)$  est une série et ne converge pas forcément pour toute valeur de  $z$ . En toute rigueur il faudrait associer le rayon de convergence à chaque fonction en  $z$  :  $\{s(z) \text{ converge pour } |z| > \alpha\}$ .

Les automaticiens pour qui la transformée en z n'est bien souvent qu'une façon bien commode de regrouper des termes de récurrence utilisent souvent la pseudo-variable<sup>2</sup>  $q$ .

A partir d'une transformée en z  $s(z)$  il est possible de retrouver la séquence  $s(k)$  correspondante en appliquant la transformation inverse, dont on ne donnera pas la formule ici. A la place de faire appel à cette formule compliquée il est possible d'utiliser d'autres méthodes pour retrouver la séquence  $s(k)$  lorsque la fonction  $s(z)$  est donnée sous forme de fraction rationnelle :

<sup>1</sup> Pour une justification du terme « causal » voir page 11

<sup>2</sup> Voir page 10

- division polynomiale
- décomposition de la fraction rationnelle en éléments simples puis utilisation de transformations élémentaires telles que celles données dans le Tableau 2

La propriété fondamentale de la transformée en z est qu'elle est linéaire (la transformée en z d'une composition linéaire de signaux est la composition linéaire de la transformée en z de chaque signal). Les autres propriétés fondamentales sont données dans le Tableau 2.

Tableau 1 : propriétés de la transformée en z

1	$s(z) = Z(s(k)) = \sum_{k=0}^{\infty} s(k) \cdot z^{-k}$
2	$Z(\alpha \cdot s_1(k) + \beta \cdot s_2(k)) = \alpha \cdot Z(s_1(k)) + \beta \cdot Z(s_2(k))$
3	$Z(k \cdot s(k)) = -z \cdot \frac{ds(z)}{dz}$
4	$Z(c^k \cdot s(k)) = s\left(\frac{z}{c}\right)$
5	$Z(s(k-1)) = s(k=-1) + z^{-1} \cdot s(z)$
6	$Z(s(k+1)) = z \cdot s(z) - z \cdot s(k=0)$
7	$s(k=0) = \lim_{z \rightarrow \infty} (s(z))$
8	$s(k \rightarrow \infty) = \lim_{z \rightarrow 1} \left( \frac{z-1}{z} \cdot s(z) \right)$

Remarque 1 : la propriété 5 ne prend pas en compte le fait que le signal est causal : en pratique  $s(k=-1) = 0$ .

Remarque 2 : la propriété 5 justifie le fait que le terme  $z^{-1}$  soit appelé "opérateur de retard unitaire", cette propriété est aussi appelée "théorème du retard".

Remarque 3 : les propriétés 7 et 8 sont données sous réserve de convergence.

Transformées inverses

### Exemple 1 : transformée inverse par division polynomiale

On considère un signal  $s(k)$  dont la transformée en z est donnée par la fraction rationnelle :

$$S(z) = \frac{0.1 \cdot z^2 + 0.1 \cdot z}{z^3 - 2.6 \cdot z^2 + 2.2 \cdot z - 0.6} \quad (3.1)$$

L'objectif est d'obtenir  $S(z)$  sous forme polynomiale :

$$S(z) = x_0 + x_1 \cdot z^{-1} + x_2 \cdot z^{-2} + \dots \text{etc.} \tag{3.2}$$

Cette suite est généralement infinie, la réponse temporelle est donnée par la succession des termes  $\{x_i\}$ . En traitant l'équation (3.1) comme une simple division l'algorithme de division classique (celui appris en CM2) permet d'obtenir par récurrence tous les termes de la suite (3.2) :

$0.1 \cdot z^2$	$+ 0.1 \cdot z$				$z^3 - 2.6 \cdot z^2 + 2.2 \cdot z - 0.6$
$0.1 \cdot z^2$	$- 0.26 \cdot z$	$+ 0.22$	$- 0.06 \cdot z^{-1}$		
0	$0.36 \cdot z$	$- 0.22$	$0.06 \cdot z^{-1}$		
	$0.36 \cdot z$	$- 0.936$	$0.792 \cdot z^{-1}$	$- 0.216 \cdot z^{-2}$	
0	$0.716$	$- 0.732 \cdot z^{-1}$	$\dots$	$\dots$	

Les éléments successifs de la suite  $s(k)$  sont donc  $s(k) = \{0, 0.1, 0.36, 0.716, \dots\}$ . Pour obtenir les termes suivants il faudrait continuer la division.

**Exemple 2 : transformée inverse par décomposition en éléments simples**

La deuxième méthode présentée consiste à décomposer la fraction rationnelle (3.1) en éléments simples puis d'utiliser les tables de transformées en  $z$  (Tableau 2). Pour notre exemple cela donne :

$$S(z) = \frac{0.1 \cdot z^2 + 0.1 \cdot z}{z^3 - 2.6 \cdot z^2 + 2.2 \cdot z - 0.6} = \frac{-z}{z-1} + \frac{0.5 \cdot z}{(z-1)^2} + \frac{z}{z-0.6}$$

La réponse temporelle est donc composée des trois éléments :

$$s(k) = -(1 \ 1 \ 1 \ \dots) + 0.5 \cdot (1 \ 2 \ 3 \ \dots) + (1 \ 0.6 \ 0.6^2 \ \dots)$$

$$\text{soit : } s(k) = \begin{cases} s(0) = -1 + 0.5 + 1 \\ s(2) = -1 + 0.5 \cdot 2 + 0.6 \cdot 0.5 \\ s(3) = -1 + 0.5 \cdot 3 + 0.6^2 \\ s(4) = \dots \end{cases}$$

**3.1.2 Signaux types**

Le Tableau 2 donne les transformée en  $z$  de certains signaux fondamentaux. La Figure 2 représente l'allure de quelques signaux types.

Tableau 2 : transformée en z élémentaires

$s(k)$	$S(z)$
signal temporel	Transformée en z
$\delta(k)$ : impulsion unité	1
$u(k)$ : échelon unité	$\frac{z}{z-1}$
$k \cdot u(k)$	$\frac{z}{(z-1)^2}$
$c^k \cdot u(k)$	$\frac{z}{z-c}$
$k \cdot a^k \cdot u(k)$	$\frac{a \cdot z}{(z-a)^2}$
$\sin(\omega \cdot k) \cdot u(k)$	$\frac{z \cdot \sin(\omega)}{z^2 - 2 \cdot z \cdot \sin(\omega) + 1}$
$\cos(\omega \cdot k) \cdot u(k)$	$\frac{z \cdot (z - \cos(\omega))}{z^2 - 2 \cdot z \cdot \sin(\omega) + 1}$
$a^k \cdot \sin(\omega \cdot k) \cdot u(k)$	$\frac{a \cdot z \cdot \sin(\omega)}{z^2 - 2 \cdot a \cdot \cos(\omega) \cdot z + a^2}$
$a^k \cdot \cos(\omega \cdot k) \cdot u(k)$	$\frac{z \cdot (z - a \cdot \cos(\omega))}{z^2 - 2 \cdot a \cdot \cos(\omega) \cdot z + a^2}$

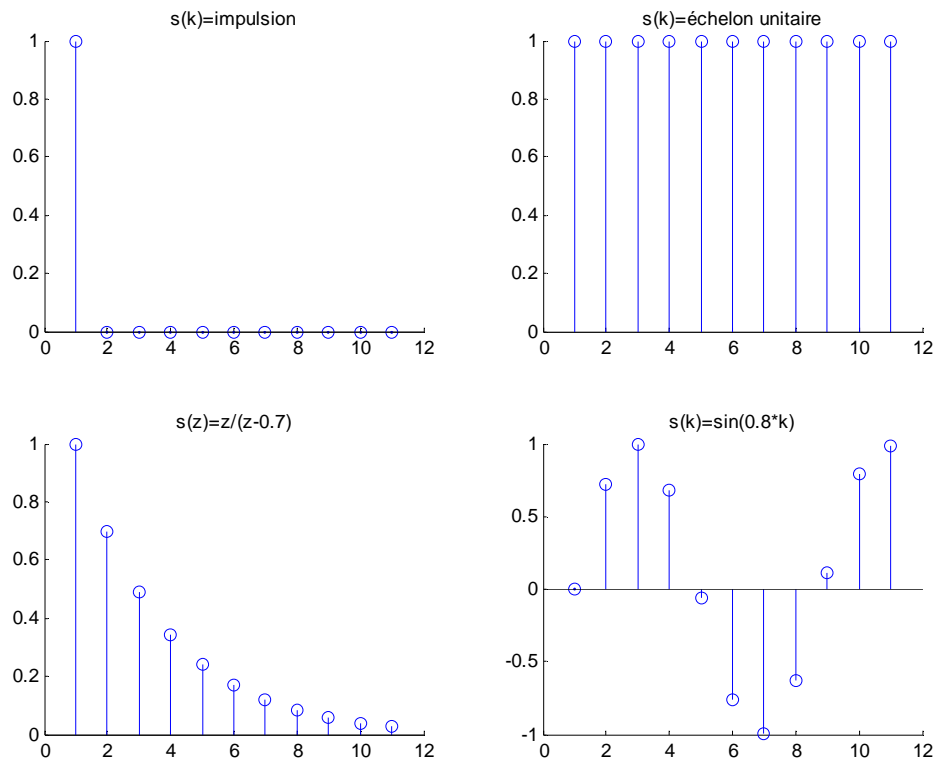


Figure 2 : réponses de quelques signaux types

### 3.2 Systèmes discrets

#### 3.2.1 Fonction de transfert discrète, définition

On considère maintenant un système discret linéaire et invariant dans le temps (LTI) représenté sur la Figure 3.

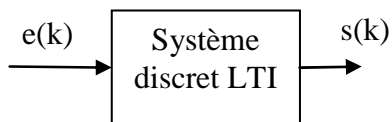


Figure 3 : système discret, notations

La relation entre la séquence d'entrée et la séquence de sortie est linéaire et définie par un ensemble de coefficients constants  $a_{1:n}$  et  $b_{1:m}$  :

$$\begin{aligned} s(k) + a_1 \cdot s(k-1) + a_2 \cdot s(k-2) + \dots + a_n \cdot s(k-n) = \\ b_0 \cdot e(k) + b_1 \cdot e(k-1) + b_2 \cdot e(k-2) + \dots + b_m \cdot e(k-m) \end{aligned} \quad (2)$$

On peut appliquer la transformée en  $z$  à chaque membre de l'équation puis en appliquant en récurrence la propriété 5 (théorème du retard) on obtient la fonction de transfert discrète du système :

$$\frac{s(z)}{e(z)} = F(z) = \frac{b_0 + b_1 \cdot z^{-1} + b_2 \cdot z^{-2} + \dots + b_m \cdot z^{-m}}{1 + a_1 \cdot z^{-1} + a_2 \cdot z^{-2} + \dots + a_n \cdot z^{-n}} \quad (3)$$

Remarque 1 : la fonction de transfert précédente est normalisée en prenant  $a_0 = 1$ .

Remarque 2 : on remarquera que la fonction de transfert est donnée en  $z^{-1}$  plutôt qu'en  $z$ . Cela est justifié par le fait que celle-ci traduit l'équation d'évolution du système physique causal (2).

Le script Matlab permettant de créer un objet Matlab correspondant à une fonction de transfert discrète est donné ci-dessous.

#### Matlab

```
% Création d'un système linéaire discret :
Te=0.1;
sysd = tf([b0 b1 b2],[1 a0 a2 a3],Te,'variable','z^-1');
```

On notera l'utilisation de l'argument 'variable', 'z^-1'.

### 3.2.2 Notation "opérateur de retard $q^{-1}$ "

La fonction de transfert est une fonction de la variable complexe  $z$ . Il faudrait donc en toute rigueur l'associer à son rayon de convergence. En pratique on ne s'embarrasse jamais de préciser cette zone de convergence. C'est pourquoi on trouve souvent la notation utilisant l'opérateur  $q^{-1}$  de décalage à la place de la variable complexe  $z^{-1}$  (4).  $F(q)$  n'est alors plus à proprement parler une fonction mais on lui appliquera les mêmes manipulations algébriques qu'à  $F(z)$ .

$$F(q) = \frac{b_0 + b_1 \cdot q^{-1} + b_2 \cdot q^{-2} + \dots + b_m \cdot q^{-m}}{1 + a_1 \cdot q^{-1} + a_2 \cdot q^{-2} + \dots + a_n \cdot q^{-n}} \quad (4)$$

Attention : selon les auteurs l'opérateur qui remplace  $z^{-1}$  est noté  $q$  (Matlab) ou  $q^{-1}$  (Landau [1]). Dans la suite du document c'est cette dernière convention qui sera appliquée.

L'utilisation de l'opérateur de retard  $q^{-1}$  à la place de la variable complexe  $z^{-1}$  permet de traiter du cas des fonctions de transfert à coefficients non constant, cas non prévu par la transformation en  $z$ .

### 3.2.3 Réponse impulsionnelle

Essayons d'appliquer au système un signal d'entrée particulier : une impulsion unitaire. La transformée en  $z$  du système est obtenue en multipliant la fonction de transfert par la transformée en  $z$  de l'impulsion unitaire : 1. On obtient :

$$s(z) = \frac{b_0 + b_1 \cdot z^{-1} + b_2 \cdot z^{-2} + \dots + b_m \cdot z^{-m}}{1 + a_1 \cdot z^{-1} + a_2 \cdot z^{-2} + \dots + a_n \cdot z^{-n}} \times 1 \quad (5)$$

Cela justifie le fait que la fonction de transfert du système peut aussi s'appeler "réponse impulsionnelle".

#### Matlab

```
% Réponse impulsionnelle d'un système :
[y,t] =impulse(sysd);
```

### 3.2.4 Causalité

Un système est dit causal si les sorties ne dépendent que du passé et non pas du futur. La conséquence est que la réponse impulsionnelle de  $F(z)$  ne peut pas comporter de terme non nuls pour  $k < 0$ . Cela justifie la confusion sémantique entre "système causal" et "signal causal", c'est-à-dire ne comportant pas de termes non nuls pour  $k < 0$ .

Attention : à la différence des systèmes continus un système physique (causal) discret peut très bien être modélisé avec un degré  $m$  supérieur au degré  $n$  : le système peut avoir plus de zéros que de pôles.

La transformée en  $z$  représentant des signaux échantillonnés elle peut tout aussi bien représenter les mesures prises à différents endroits d'une poutre. Le temps étant remplacé ici par une abscisse la notion de causalité ne s'applique pas ! Attention : beaucoup des propriétés de systèmes échantillonnés que l'on étudie sont liés à la causalité, prudence donc si l'on applique ces méthodes à d'autres domaines !

### 3.2.5 Réponses indicielle

Pour caractériser des systèmes on s'intéresse souvent à leur réponse impulsionnelle ou, plus souvent, à leur réponse indicielle. Pour calculer la réponse

à un échelon d'un système il faut donc multiplier la fonction de transfert par la transformée en z d'un échelon unitaire :

$$s(z) = \frac{b_0 + b_1 \cdot z^{-1} + b_2 \cdot z^{-2} + \dots + b_m \cdot z^{-m}}{1 + a_1 \cdot z^{-1} + a_2 \cdot z^{-2} + \dots + a_n \cdot z^{-n}} \times \frac{1}{1 - z^{-1}} \quad (6)$$

Il faudrait ensuite calculer la transformée en z inverse de s(z). En pratique, si l'on doit faire le calcul à la main on décompose en éléments simples puis on utilise des tables. Avec Matlab c'est plus simple. Voici par exemple le script pour obtenir les 20 premiers échantillons de la réponse indicielle d'un système :

**Matlab**

```
% Réponse indicielle d'un système :
[y,t] =step(sysd,20);
```

**3.2.6 Régime stationnaire (gain statique)**

Le gain statique est obtenu en appliquant une entrée indicielle et en calculant la valeur du régime permanent. En appliquant le théorème de la valeur finale il vient :

$$F_0 = \lim_{z \rightarrow 1} \left( \frac{z-1}{z} \cdot \frac{b_0 + b_1 \cdot z^{-1} + b_2 \cdot z^{-2} + \dots + b_m \cdot z^{-m}}{1 + a_1 \cdot z^{-1} + a_2 \cdot z^{-2} + \dots + a_n \cdot z^{-n}} \times \frac{1}{1 - z^{-1}} \right) \quad (7)$$

Soit :

$$F_0 = \frac{\sum_{i=0}^{i=m} b_i}{1 + \sum_{i=1}^{i=n} a_i} \quad (8)$$

Le gain statique est obtenu donc simplement en faisant le quotient de la somme des coefficients.

**3.2.7 Pôles, zéros, stabilité**

Comme pour les systèmes continus la dynamique et la stabilité des systèmes dépend des propriétés du dénominateur de la fonction de transfert.



$$F(z) = \frac{e_0 \cdot (1 + e_1 \cdot z^{-1}) \cdot (1 + e_2 \cdot z^{-1}) \cdots (1 + e_m \cdot z^{-1})}{(1 + c_1 \cdot z^{-1}) \cdot (1 + c_2 \cdot z^{-1}) \cdots (1 + c_n \cdot z^{-1})} \quad (9)$$

Les coefficients  $c_i$  sont les pôles de la fonction de transfert, les coefficients  $e_i$  sont les zéros. La fonction de transfert étant au départ composée de coefficients purement réels les coefficients  $c_i$  sont soit réels soit imaginaires par paires conjuguées.

Une fonction de transfert est donc aussi bien définie par la donnée de ses pôles et de ses zéros. Le script Matlab pour créer un système ainsi est donné ci-dessous :

#### Matlab

```
% Système discret défini par ses pôles et zéros :
Te=0.1;
sysd=zpk([c1 c2 c3],[e1 e2],e0,Te);
```

Si l'on décompose la fonction de transfert en éléments simples on obtient une somme d'éléments du type :

$$F(z) = \frac{d_1}{1 + c_1 \cdot z^{-1}} + \frac{d_2}{1 + c_2 \cdot z^{-1}} + \cdots + \frac{d_n}{1 + c_n \cdot z^{-1}} \quad (10)$$

Un système linéaire temps invariant est dit stable si sa réponse impulsionnelle tend asymptotiquement vers 0. Compte tenu de (10) cela nécessite que chaque pôle  $c_i$  soit de module inférieur à 1.

**Propriété** : un système discret LTI est stable si et seulement si tous ses pôles ont leur modules inférieur ou égal à 1.

#### Critère algébrique attestant de la stabilité : critère de Routh

On rappelle que le critère de Routh permet de prédire si les racines d'un polynôme sont toutes à partie réelle négative, donc stables. Énonçons le théorème de Routh dans le cas d'un polynôme  $P'(w)$  :

$$P'(w) = a_0 + a_1 \cdot w + a_2 \cdot w^2 + \cdots + a_n \cdot w^n \quad (11)$$

Dans un premier temps on construit le tableau :

$$\begin{array}{cccccc|c}
 \beta_{n,0} & \beta_{n,1} & \beta_{n,2} & \beta_{n,q-1} & \beta_{n,q} & 0 & \\
 \beta_{n-1,0} & \beta_{n-1,1} & \beta_{n-1,2} & \beta_{n-1,q-1} & \beta_{n-1,q-1} & 0 & \\
 \beta_{n-2,0} & \beta_{n-2,1} & \beta_{n-2,2} & \beta_{n-2,q-1} & \beta_{n-2,q-1} & & \\
 & & & & & & \\
 \beta_{2,0} & \beta_{2,1} & 0 & & & & \\
 \beta_{1,0} & 0 & & & & & \\
 \beta_{0,0} & 0 & & & & & 
 \end{array} \quad (12)$$

Puis on vérifie que tous les premiers éléments de la première colonne ont le même signe : cela garantit que tous les pôles de  $P'(w)$  ont leur partie réelle négative.

Les éléments du tableau sont construits par récurrence ligne par ligne en appliquant les formules suivantes :

Pour  $i = n$  et  $i = n - 1$  :

$$\begin{cases} \beta_{n,j} = a_{n-2j} \\ \beta_{n-1,j} = a_{n-1-2j} \end{cases} \quad (13)$$

Et pour  $i < n-1$  :

$$\beta_{i,j} = \frac{\beta_{i+1,0} \cdot \beta_{i+2,j+1} - \beta_{i+1,j+1} \cdot \beta_{i+2,0}}{\beta_{i+1,0}} \quad (14)$$

Dans le cas qui nous intéresse, un système échantillonné, on cherche plutôt à vérifier que les pôles du polynôme  $P(z)$  sont de module inférieur à un. L'idée est d'effectuer un changement de variable  $z \rightarrow w$  pour pouvoir appliquer le critère de Routh.

$$P'(w) = (1-w)^n \cdot P\left(\frac{1+w}{1-w}\right) \quad (15)$$

Si l'on effectue la transformation (15) alors le fait que  $P'(w)$  ait toutes ses racines à partie réelle négative garantit que  $P(z)$  a toutes ses racines dans le cercle unité. Pour vérifier la stabilité de  $P(z)$  il suffit donc d'effectuer le changement de variable (15) puis d'appliquer le critère de Routh conventionnel (12 13 14).

### Critère algébrique attestant de la stabilité : critère de Jury

Le critère de Jury s'apparente au critère de Routh. Nous le donnons pour un polynôme de degré 2, 3 et 4 du type :

$$P(z) = a_0 + a_1 \cdot z + a_2 \cdot z^2 + \dots + a_n \cdot z^n \quad (16)$$

Les conditions de stabilité sont données ci-dessous (ces conditions sont données dans le cas  $a_n > 0$ , il suffit de multiplier tous les termes par  $-1$  si cette condition n'est pas remplie par  $P(z)$ ) :

$$\begin{aligned}
 n = 2 & \begin{cases} a_0 + a_1 + a_2 > 0 \\ a_0 - a_1 + a_2 > 0 \\ -a_0 + a_2 > 0 \end{cases} \\
 n = 3 & \begin{cases} a_0 + a_1 + a_2 + a_3 > 0 \\ -a_0 + a_1 - a_2 + a_3 > 0 \\ -|a_0| + a_3 > 0 \\ a_0 a_2 - a_1 a_3 - a_0^2 + a_3^2 > 0 \end{cases} \\
 n = 4 & \begin{cases} a_0 + a_1 + a_2 + a_3 + a_4 > 0 \\ a_0 - a_1 + a_2 - a_3 + a_4 > 0 \\ a_4^2 - a_0^2 - |a_0 a_3 - a_1 a_4| > 0 \\ (a_0 - a_4)^2 (a_0 - a_2 + a_4) + (a_1 - a_3)(a_0 a_3 - a_1 a_4) > 0 \end{cases}
 \end{aligned} \quad (17)$$

### 3.2.8 Systèmes du premier ordre

On donne dans la Figure 4 les réponses indicielles de systèmes du premier ordre ainsi que la localisation des pôles dans le plan complexe. Le cercle de rayon unité représente la zone dans laquelle les pôles doivent se situer pour assurer la stabilité. On remarque sur les deux premières figures l'influence de la place du zéro par rapport au point  $(1,0)$  pour la rapidité.

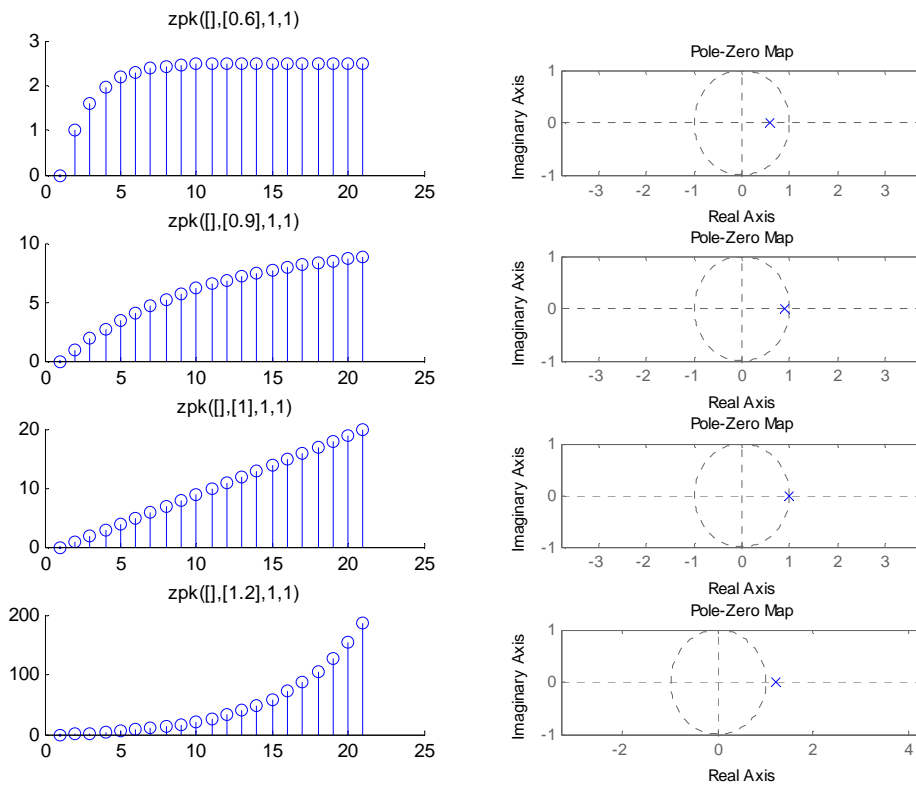


Figure 4 réponses indicielles de systèmes du premier ordre

### 3.2.9 Systèmes du second ordre

On donne dans la Figure 6 les réponses indicielles de systèmes du second ordre ainsi que la localisation des pôles dans le plan complexe. La rapidité et l'amortissement dépend de la position des pôles dans le cercle unité. La Figure 5 représente les courbes iso-amortissement et iso-pulsation des systèmes du second ordre.

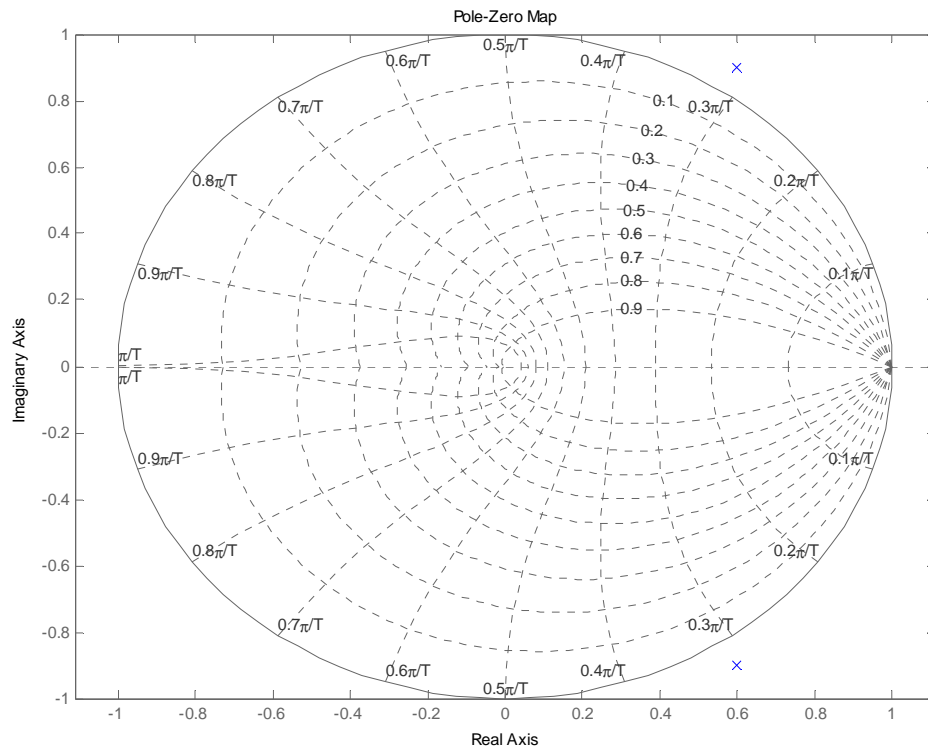


Figure 5 : courbes iso-amortissement et iso-pulsation des systèmes discrets du second ordre

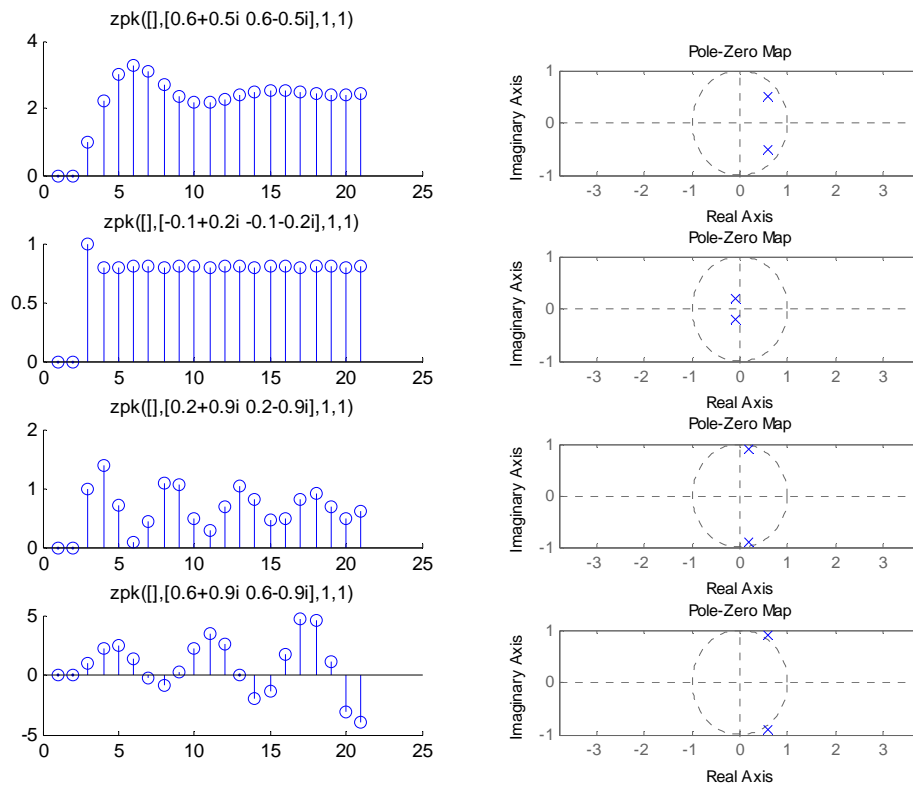


Figure 6 : réponses de systèmes du second ordre

### 3.2.10 Calcul itératif de la réponse d'un système

Si l'on dispose de la fonction de transfert d'un système et que l'on connaît les valeurs successives  $e(k)$  du signal d'entrée il est facile de calculer itérativement la sortie  $y(k)$  du système par l'équation de récurrence (2). Il faut initialiser par les  $n$  valeurs initiales de  $s(k)$ , prises généralement égales à 0.

### 3.2.11 Représentation par équation d'état

Un système discret peut aussi être représenté par un ensemble d'équations d'état comme dans le domaine continu (52). Cette représentation est naturellement adaptée aux systèmes multi-entrées multi-sorties. On définit alors un état interne  $x(k)$  ainsi qu'une fonction d'évolution  $f$  et une équation de mesure  $h$ .

$$\begin{cases} x(k+1) = f(x(k), u(k)) \\ s(k) = h(x(k), u(k)) \end{cases} \quad (18)$$

### 3.3 Réponse fréquentielle des systèmes échantillonnés

#### 3.3.1 Diagramme de Bode d'un système échantillonné

Lorsqu'on étudie la réponse fréquentielle des systèmes continus on considère des entrées particulières sinusoïdales :

$$\begin{cases} e(t) = \cos(2 \cdot \pi \cdot f \cdot t) + j \cdot \sin(2 \cdot \pi \cdot f \cdot t) = e^{j \cdot 2 \cdot \pi \cdot f \cdot t} \\ f \in [0 \quad \infty] \end{cases} \quad (19)$$

De même, pour des systèmes échantillonnés on considèrera des entrées sinusoïdales échantillonnées :

$$\begin{cases} e(k) = \cos(2 \cdot \pi \cdot f \cdot k) + j \cdot \sin(2 \cdot \pi \cdot f \cdot k) = e^{j \cdot 2 \cdot \pi \cdot f \cdot k} \\ f \in [0 \quad 1] \end{cases} \quad (20)$$

La sortie est aussi un signal échantillonné sinusoïdal de gain  $A(f)$  et de phase  $\Phi(f)$  variables et dépendant de  $f$ ; la fréquence réduite.

$$\begin{cases} s(k) = A(f) \cdot e^{j \cdot (2 \cdot \pi \cdot f \cdot k + \Phi(f))} \\ f \in [0 \quad 1] \end{cases} \quad (21)$$

Le nombre complexe, noté abusivement  $F(f)$ , d'amplitude  $A(f)$  et de phase  $\Phi(f)$  est obtenu à partir de  $F(z)$  en remplaçant  $z$  par  $e^{j \cdot 2 \cdot \pi \cdot f}$  :

$$\begin{cases} F(f) = F(z = e^{j \cdot 2 \cdot \pi \cdot f}) \\ f \in [0 \quad 1] \end{cases} \quad (22)$$

Le diagramme représentant le gain et la phase en fonction de la fréquence réduite s'appelle aussi un diagramme de Bode. L'échelle des fréquences est généralement linéaire de 0 à 0.5 (puisque le diagramme est nécessairement symétrique). La Figure 7 représente par exemple le diagramme de Bode d'un système du second ordre.

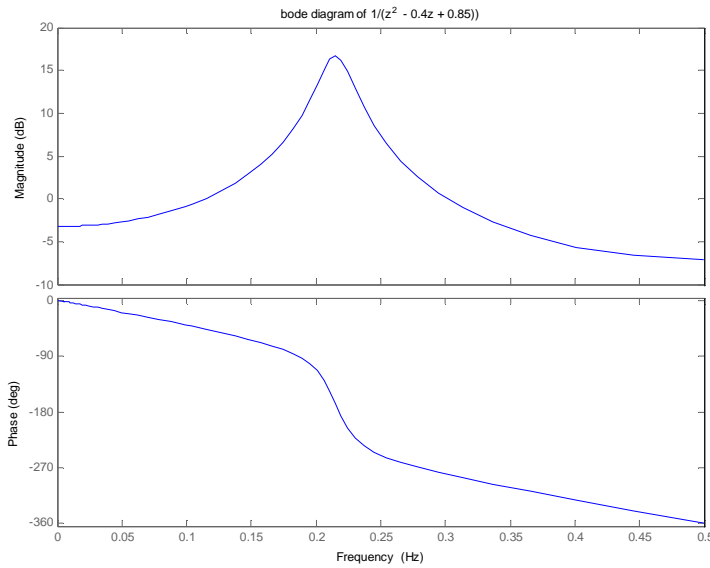


Figure 7 : exemple du diagramme de Bode d'un système discret du second ordre

### 3.3.2 Où l'on voit apparaître le théorème de Shannon

Le changement de variable  $z = e^{j2\pi f}$  n'est pas bijectif puisque  $e^{j2\pi f} = e^{j2\pi(f+k)}$  avec  $k$  entier. Si l'on fait le lien entre domaine continu et domaine fréquentiel on peut remarquer que le changement de variable  $z = e^{j2\pi f}$  fait correspondre une zone du plan complexe continu à une zone du plan complexe discret. La transformation est bijective si l'on se contente de la zone du plan complexe continu contenu entre les droites d'ordonnée 0 et 1.



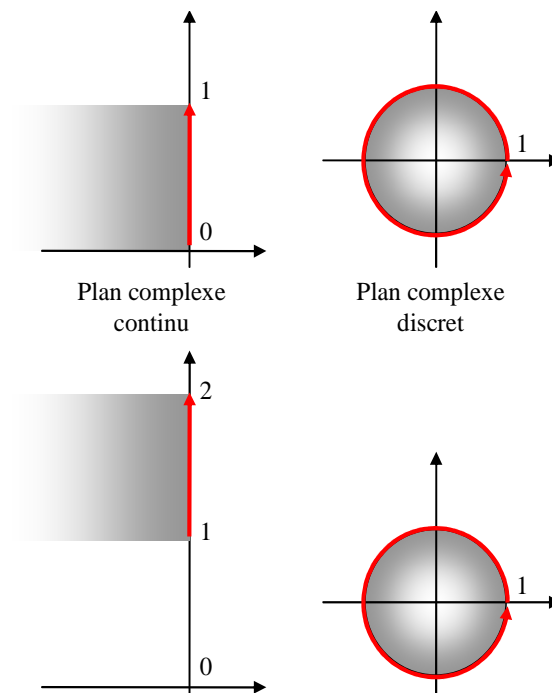


Figure 8 : correspondance entre plan complexe continu et plan complexe discret.



## 4. SYSTEMES CONTINUS ECHANTILLONNES

### 4.1 Structure des systèmes commandés par ordinateur

Les systèmes commandés par ordinateurs sont presque toujours des systèmes physiques continus par nature. Le ordinateur traite par contre les entrées et les sorties de manière discrète avec une période d'échantillonnage  $T_e$ . Les éléments importants, du point de vue de l'automaticien sont donc :

- le convertisseur analogique / numérique CAN ou du moins le système qui fournit séquentiellement les mesures au ordinateur à la période  $T_e$ .
- le ordinateur lui-même. Il doit effectuer l'acquisition des mesures, le traitement des informations, l'envoi des consignes dans un temps inférieur à  $T_e$ .
- le convertisseur numérique / analogique CNA.
- le système physique.

On suppose, pour faire simple et pour pouvoir utiliser la panoplie des outils mathématiques, que les systèmes sont échantillonnés à la même période  $T_e$  et au même moment.

La Figure 9 montre le système physique commandé par le ordinateur, dans sa version simpliste.

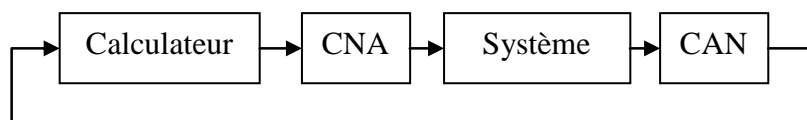


Figure 9 : système commandé par ordinateur

Au XXIème siècle les choses sont quand même parfois un petit peu plus compliquées :

- le ordinateur fournit bien souvent un signal de commande numérique (transporté par bus informatique USB, RS232, CAN, Ethernet, etc...) directement interprété par l'actionneur. On parle "d'actionneur intelligent".
- de même le "capteur intelligent" fournit aussi bien souvent la mesure sous forme d'un signal numérique.
- Les signaux sont transportés par des bus informatiques qui n'ont pas que ça à faire : il peut y avoir du retard.
- de même les capteurs et actionneurs sont rarement tous seuls, ce qui se traduit par des périodes d'échantillonnage différentes, et parfois non synchrones...

- enfin le ordinateur peut aussi avoir d'autres choses à faire...

Ceci étant dit nous allons pour le moment nous restreindre au cas représenté sur la Figure 9 avec des instants d'échantillonnage synchrones et régulièrement espacés d'un temps  $T_e$ .

## 4.2 Echantillonnage des systèmes continus

On dispose généralement de la fonction de transfert continue (notée en utilisant la variable complexe  $s$  ou  $p$ )  $F(s)$  du système continu. Le problème est d'obtenir la fonction de transfert discrète du système échantillonné de manière à pouvoir synthétiser un correcteur. Le système continu est associé au convertisseur numérique – analogique qu'il faut prendre en compte.

Les convertisseurs numérique analogiques peuvent généralement se représenter par des "échantillonneurs – bloqueurs d'ordre 0" ou BOZ. Cela signifie que la commande est maintenue à une valeur constante pendant toute la durée du temps d'échantillonnage.

Si l'on applique une commande discrète impulsionnelle à un BOZ il va maintenir la commande 1 pendant la durée  $T_e$  puis revenir à zéro (Figure 10).

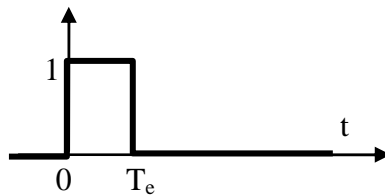


Figure 10 : réponse impulsionnelle d'un bloqueur d'ordre 0

La réponse impulsionnelle  $BOZ(t)$  est donc composée d'un échelon unitaire suivi d'un échelon unitaire de signe opposé et décalé d'un temps  $T_e$ . La transformée de Laplace de la réponse impulsionnelle du bloqueur est donnée par la formule (23) :

$$BOZ(s) = \frac{1}{s} - \frac{e^{-jT_e}}{s} \quad (23)$$

Du point de vue du ordinateur il faut donc prendre en compte le système continu de fonction de transfert  $F_{BOZ}(s)$  que l'on devra échantillonner :

$$F_{BOZ}(s) = \frac{1 - e^{-jT_e}}{s} \cdot F(s) \quad (24)$$

$$F_{\text{BOZ}}(z) = Z^{-1} \left( \frac{1 - e^{-jT_e}}{s} \cdot F(s) \right) \quad (25)$$

En pratique l'obtention de la transformée en z d'un système échantillonné par bloqueur d'ordre 0 se fait à l'aide de tables. On donne quelques exemples fondamentaux dans le Tableau 3 ci-dessous.

Tableau 3 : transformée en z de systèmes continus échantillonnés

F(s)	F <sub>BOZ</sub> (z)
1	1
$\frac{1}{s}$	$\frac{T_e \cdot z^{-1}}{1 - z^{-1}}$
$\frac{1}{1 + T \cdot s}$	$\begin{cases} \frac{b_1 \cdot z^{-1}}{1 + a_1 \cdot z^{-1}} \\ b_1 = 1 - e^{-T_e/T} \quad a_1 = -e^{-T_e/T} \end{cases}$
$\frac{e^{-Ls}}{1 + T \cdot s}$	$\begin{cases} \frac{b_1 \cdot z^{-1} + b_2 \cdot z^{-2}}{1 + a_1 \cdot z^{-1}} \quad a_1 = -e^{-T_e/T} \\ b_1 = 1 - e^{-(L-T_e)/T} \quad b_2 = e^{-T_e/T} (e^{L/T} - 1) \end{cases}$
$L < T_e$	

Si l'on dispose de Matlab c'est bien sûr encore plus simple. On utilise la fonction c2d avec l'argument 'zoh' pour "zero holder hold" (bloqueur d'ordre zéro).

#### Matlab

```
% Définition d'un système continu :
sysc=tf(1,[1 1]);
% Obtention de la transformée en z du système échantillonné à Te
Te=0.1;
sysd=c2d(sysc,Te,'zoh');
% Notation en z^-1
sysd.variable='z'
```

### 4.3 Effets de l'échantillonnage

Le convertisseur numérique a deux effets (néfastes) importants : l'effet de retard et l'effet de quantification.

L'effet de retard se voit dans (24) : on peut dire en première approximation que l'échantillonnage se traduit par un retard pur de  $T_e/2$ . Cela a donc, si ce temps

d'échantillonnage est non négligeable par rapport aux constantes de temps du système, un effet déstabilisant.

Pour illustrer l'effet de quantification supposons par exemple que l'on utilise un convertisseur analogique/numérique 8 bits 0-10V. Le plus petit incrément que le convertisseur peut distinguer est de  $10V/2^8 = 40mV$ . Si le signal utile varie physiquement entre 5V et 5,4V on n'aura que dix incréments utilisables... Ce phénomène n'entre pas dans la panoplie des phénomènes modélisables par la transformée en z. En pratique il est donc souvent nécessaire d'étudier le système et de synthétiser la commande numérique en négligeant ce phénomène puis de vérifier a posteriori qu'il n'a pas un effet trop néfaste.

#### 4.4 Où l'on revoit le théorème de Shannon

Le théorème de Shannon est bien connu en ce qui concerne l'échantillonnage de signaux discrets : on sait que le signal ne doit pas comporter de composante de fréquence supérieure à  $f_e/2$  si l'on ne veut pas observer de repliement. On doit aussi prendre des précautions si l'on veut travailler sur la fonction de transfert discrète correspondant à un système continu échantillonné.

On va traiter un exemple concret pour mettre en évidence le problème de repliement dans le cas de la commande des systèmes continus. On s'intéresse à deux systèmes continus du second ordre ayant des pôles de même partie réelle mais de parties imaginaires séparées de  $2\pi T_e$ . Le script Matlab ci-dessous montre la création des deux systèmes :

##### Matlab

```
% Définition de deux systèmes continus :  
sys1=zpk([],[-1-2i -1+2i],2^2+1);  
sys2=zpk([],[-1-(2+2*pi)*i -1+(2+2*pi)*i],1+(2+2*pi)^2);
```

On voit sur la Figure 11 que la réponse indicielle des deux systèmes est bien différente.

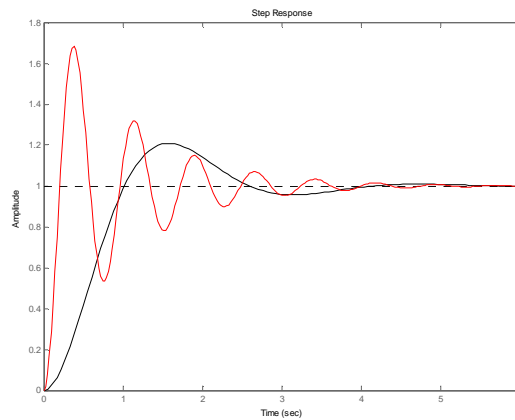


Figure 11 : réponse indicielle de deux systèmes du second ordre

Si l'on calcule la fonction de transfert échantillonnée les deux systèmes à la période de 1 seconde on obtient deux réponses indicielles très proches (représentés par des  $\circ$  sur la Figure 12). Conclusion : la période d'échantillonnage était mal choisie

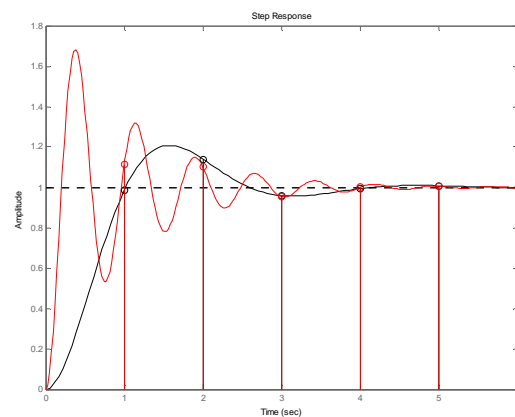


Figure 12 : réponse indicielle de deux systèmes du second ordre échantillonnés à 1s

En pratique la commande échantillonnée de systèmes continus doit être associée à certaines précautions :

- le système ne doit pas comporter de dynamiques propres situées dans les fréquences supérieures à  $f_c/2$  (exemple illustré dans ce chapitre).
- les signaux échantillonnés doivent être "nettoyés" des fréquences supérieures à  $f_c/2$  avec un filtre anti-repliement. (A moins que le signal soit suffisamment propre et ne comporte pas de composantes significatives au-delà de ces fréquences).

### Choix de la période d'échantillonnage

Le choix de la période d'échantillonnage est très important et dépend de la dynamique des signaux. Attention, les systèmes commandés en boucle fermée ont souvent une dynamique plus rapide que le système en boucle ouverte : il faut donc prendre en compte la dynamique en boucle fermée pour le choix de la fréquence d'échantillonnage.

En pratique on préconise de prendre une fréquence 5 à 25 fois supérieure à la bande passante souhaitée en boucle fermée.

On peut être tenté de prendre une fréquence d'échantillonnage très élevée si la technologie le permet (ordinateur et convertisseurs suffisamment rapide). Ceci peut amener rapidement à des problèmes numériques. Imaginons par exemple le cas d'un système ayant une dynamique très lente (quelques secondes) et un système de mesure codant les mesures sur 10 bits (cas classique). La Figure 13 illustre cet exemple. Le signal varie peu entre deux instants d'échantillonnage et les variations du signal sont faibles par rapport à la valeur de l'incrément numérique (figure du haut). Le calcul de la simple dérivée numérique se traduit donc par une succession de  $-1$   $0$  et  $+1$ .

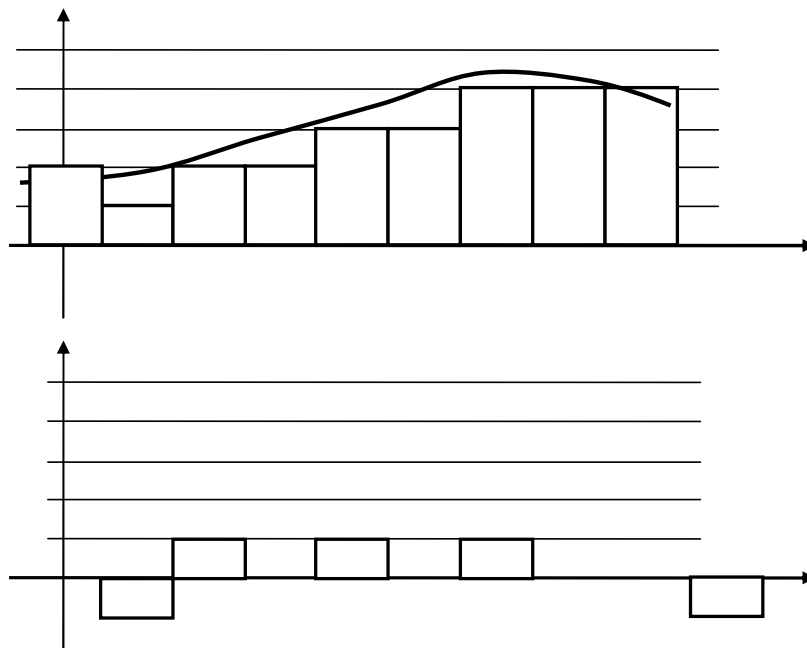


Figure 13 : période d'échantillonnage trop faible. En haut : signal échantillonné et discrétisé ; en bas : calcul de la dérivée numérique.



### Filtre anti-repliement

En toute rigueur chaque fois qu'un signal est échantillonné il faudrait faire précéder le convertisseur numérique d'un filtre analogique anti-repliement éliminant physiquement toutes les composantes du signal de fréquence supérieure à  $f_e/2$ .

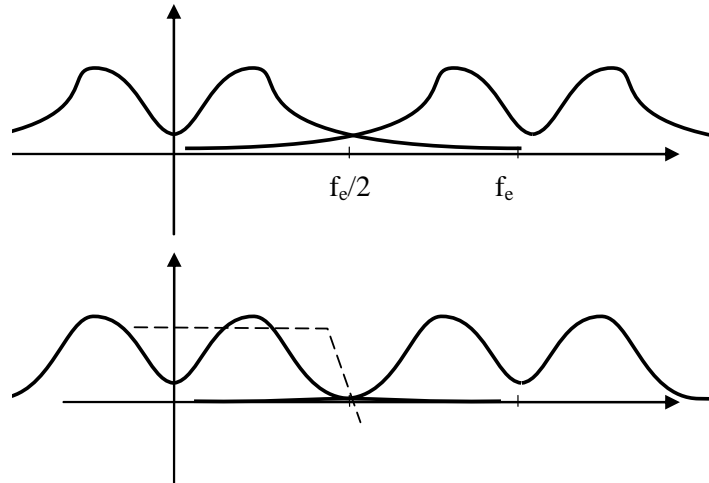


Figure 14 : Effet du filtre anti-repliement

Bien souvent la période d'échantillonnage de l'asservissement est relativement lente (quelques secondes) et il est alors impossible de réaliser un filtre de constante de temps aussi faible. On procède alors à un sur-échantillonnage : le signal est filtré, échantillonné à haute fréquence  $T'_e$  puis filtré numériquement et sous-échantillonné à la période  $T_e$  beaucoup plus grande que  $T'_e$  comme illustré sur la

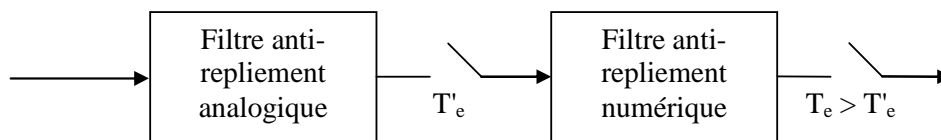


Figure 15 : Sur échantillonnage

Un simple filtre numérique de type moyenne mobile est souvent suffisant.

### 4.5 Discrétisation de fonction de transfert continues

Une manière classique pour réaliser un contrôle par ordinateur est de commencer par synthétiser un correcteur continu avec les outils de l'automatique classique puis de trouver un correcteur discret qui "approche" convenablement le correcteur continu.

#### 4.5.1 Approximation numérique d'équation différentielle

Supposons que le correcteur continu que l'on a synthétisé soit du type PD :

$$\text{PID}(s) = \frac{e(s)}{\varepsilon(s)} = k_p + k_D \cdot s \quad (26)$$

Il s'agit de calculer les commandes  $e(k)$  successives issues du correcteur en fonction des erreurs  $\varepsilon(k)$ . Une approche classique et intuitive est d'approximer la dérivée par une différence :

$$e(k) = k_p \cdot \varepsilon(k) + k_D \cdot \frac{\varepsilon(k) - \varepsilon(k-1)}{T_e} \quad (27)$$

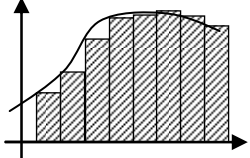
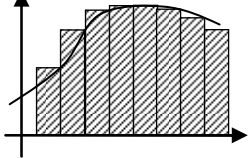
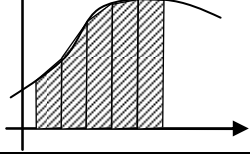
Si on calcule la transformée en  $z$  de  $e(k)$  on obtient :

$$e(z) = k_p \cdot \varepsilon(z) + k_D \cdot \frac{\varepsilon(z) - z^{-1}\varepsilon(z)}{T_e} = \left( k_p + k_D \cdot \frac{1 - z^{-1}}{T_e} \right) \cdot \varepsilon(z) \quad (28)$$

On retrouve la fonction de transfert PD(s) dans laquelle on a effectué le changement de variable  $s = \frac{1-z^{-1}}{T_e}$

De manière plus générale, l'approximation de fonction de transfert revient à effectuer l'approximation numérique d'intégration. Selon la méthode d'intégration choisie on effectue des changements de variable différents.

Tableau 4 : méthodes d'intégration

Méthode	Changement de variable	
Euler progressif (Euler's forward method)	$s = \frac{1 - z^{-1}}{z^{-1} \cdot T_e}$	
Euler rétrograde (Euler's backward method)	$s = \frac{1 - z^{-1}}{T_e}$	
Méthode des trapèzes, ou bilinéaire, ou de Tustin	$s = \frac{2}{T_e} \cdot \frac{1 - z^{-1}}{1 + z^{-1}}$	

En pratique le changement de variable  $s = \frac{1 - z^{-1}}{z^{-1} \cdot T_e}$  n'est pas sans risque. On donne sur la Figure 16 la correspondance entre la zone stable du plan complexe en  $s$  et la zone correspondante après la transformation (Euler progressif, Euler rétrograde, bilinéaire). On voit notamment qu'il est possible, partant d'un système continu stable, d'obtenir des pôles dans le domaine discret en dehors du cercle de stabilité si la transformation appliquée est celle d'Euler progressif.

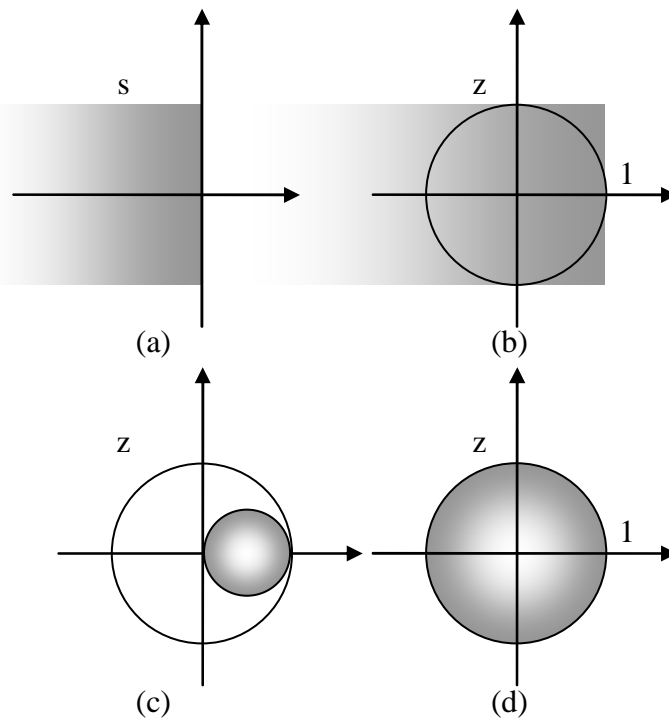


Figure 16 : correspondance des plans complexes. (a) : zone de stabilité dans le plan complexe continu en  $z$  ; (b) zone correspondante avec l'approximation d'Euler progressif ; (c) zone correspondante avec l'approximation d'Euler rétrograde ; (d) zone correspondante avec l'approximation bilinéaire.

La fonction `c2d` de Matlab permet d'obtenir l'équivalent discret d'une fonction de transfert continue avec la méthode d'intégration comme argument :

```

Matlab
sysc=tf([1 0],[1 2 5]);
Te=0.1;
% Méthode d'euler :
sysd1=c2d(sysc,Te,'zoh');
% Méthode des trapèzes :
sysd2=c2d(sysc,Te,'foh');
% Méthode bilinéaire :
sysd3=c2d(sysc,Te,'tustin');
    
```

#### 4.5.2 Superposition des réponses indicelles

Une autre méthode pour approximer une fonction de transfert discrète est de s'assurer que la réponse indicelle de sa version discrétisée correspond exactement lors des temps d'échantillonnage.

### 4.5.3 Correspondances pôles - zéros

La dernière méthode est de partir d'une décomposition de la fonction de transfert continue en pôles et zéros et d'effectuer la conversion pôles à pôles et zéros par zéros. La correspondance, selon que le pôle est réel ou complexe est donnée ci-dessous :

$$\begin{aligned} (s + a) &\rightarrow (z - e^{-a \cdot T_c}) \\ (s + a)^2 + b^2 &\rightarrow z^2 - 2 \cdot e^{-a \cdot T_c} \cdot \cos(b \cdot T_c) \cdot z + e^{-2 \cdot a \cdot T_c} \end{aligned} \quad (29)$$

### 4.5.4 Bilan

Aucune méthode n'est parfaite et on n'aura jamais en même temps :

- même réponse indicielle
- même réponse impulsionnelle
- approximation (Euler, Tustin, etc...) de l'équation différentielle
- correspondance des pôles et des zéros
- correspondance des réponses fréquentielles

Ceci étant dit le choix de la période d'échantillonnage est primordial. Si la fréquence d'échantillonnage est beaucoup plus faible que la dynamique de la fonction de transfert à approximer la différence entre chaque méthode sera minimale.



## 5. IDENTIFICATION DES SYSTEMES ECHANTILLONES

Dans le présent chapitre on donne les bases des méthodes d'identification. Pour une approche complète, voir le cours de troisième année "Identification avion" [6] et les ouvrages [1] et [7].

Même si l'on dispose d'un modèle théorique du processus que l'on souhaite commander il est presque toujours nécessaire de l'affiner en procédant à des essais sur le système réel. Bien souvent le seul modèle dont l'on dispose est ce modèle issu de l'expérimentation. Différentes méthodes s'appliquent selon l'objectif recherché et le niveau de précision exigé. On distinguera :

- les modèles paramétriques
- les modèles simples
- les modèles non paramétriques

### 5.1 Identification d'un modèle ARX par la méthode des moindres carrés récursifs

A partir de cet exemple, le plus simple qui soit, on pose les bases de toutes les méthodes d'identification.

On considère un système d'entrée  $u(t)$  et de sortie  $y(t)$  représenté par l'équation de récurrence suivante :

$$y(t) + a_1 \cdot y(t-1) + a_2 \cdot y(t-2) + \dots + a_n \cdot y(t-n) = b_0 \cdot u(t) + b_1 \cdot u(t-1) + b_2 \cdot u(t-2) + \dots + b_m \cdot u(t-m) \quad (30)$$

L'objectif est, à partir d'essais et de mesures, d'identifier les meilleures valeurs possibles pour les coefficients  $a_i$  et  $b_i$ .

L'équation (30) peut être réécrite pour mettre en évidence la *prédiction* de la sortie  $y(t)$  à partir des mesures passées :

$$y(t) = -a_1 \cdot y(t-1) - a_2 \cdot y(t-2) - \dots - a_n \cdot y(t-n) + b_0 \cdot u(t) + b_1 \cdot u(t-1) + b_2 \cdot u(t-2) + \dots + b_m \cdot u(t-m) \quad (31)$$

Soit :

$$y(t) = \varphi^T(t) \cdot \theta(t)$$

avec :

$$\begin{cases} \varphi^T(t) = [-y(t-1) \ \dots \ -y(t-n) \ , \ u(t-1) \ \dots \ u(t-m)] \\ \theta(t) = [a_1 \ \dots \ a_n \ , \ b_1 \ \dots \ b_m] \end{cases} \quad (32)$$

Le but est de trouver le meilleur ensemble de paramètres  $\theta(t)$  : celui qui, à partir d'un ensemble de mesures  $\varphi(t)$  permet de prédire avec la meilleure précision la sortie effectivement mesurée.

Supposons qu'à la suite d'une expérience on dispose d'un ensemble de mesures  $\{y(t), u(t)\}_{t=1:N}$  et que l'ensemble des paramètres  $\theta$  n'est pas bien connu. On construit la séquence des *sorties prédites* :

$$\hat{y}(t) = \varphi^T(t) \cdot \theta(t) \quad (33)$$

Le paramètre  $\theta$  est celui qui minimise une norme quadratique :

$$\theta = \arg \max \left( \frac{1}{N} \sum_{t=1}^N (y(t) - \hat{y}(t))^2 \right) = \arg \max \left( \frac{1}{N} \sum_{t=1}^N (y(t) - \varphi^T(t) \cdot \theta(t))^2 \right) \quad (34)$$

Compte tenu du fait que l'équation est linéaire au départ et que le terme à minimiser est du type quadratique la solution est simplement obtenue par une pseudo-inverse :

$$\theta = \left[ \sum_{t=1}^N (\varphi(t) \cdot \varphi^T(t)) \right]^{-1} \sum_{t=1}^N (\varphi(t) \cdot y(t)) \quad (35)$$

On obtient donc la solution classique des moindres carrés. La formule (35) est le plus souvent utilisée sous sa forme récursive :

$$\theta(t) = \theta(t-1) + F(t) \cdot \varphi(t) \cdot (y(t) - \theta^T(t-1) \cdot \varphi(t))$$

avec

$$F(t)^{-1} = F(t-1)^{-1} + \varphi(t) \cdot \varphi(t)^T \quad (36)$$

ou

$$F(t) = F(t-1) - \frac{F(t-1) \cdot \varphi(t) \cdot \varphi(t)^T \cdot F(t-1)}{1 + \varphi(t)^T \cdot F(t-1) \cdot \varphi(t)}$$

(On laisse au lecteur le plaisir d'effectuer lui-même les démonstrations, qu'il peut du reste trouver dans [1] et [7]).



## 5.2 L'identification paramétrique en général et en pratique

L'exemple précédent a permis de mettre en évidence tous les aspects de la méthodologie classique de l'identification :

- il a fallu choisir un ensemble de mesures  $\{y(t), u(t)\}_{t=1:N}$ . Le choix du signal d'excitation  $u(t)$  est en pratique primordial,
- il a fallu choisir une structure pour le modèle à identifier (ici les ordres  $n$  et  $m$ ),
- il a fallu choisir un critère (ici un critère quadratique sur l'erreur entre la sortie et la sortie prédite),
- il a fallu choisir un algorithme et le paramétrer (ici l'algorithme des moindres carrés récursifs qu'il a fallu initialiser)

Enfin un point non traité dans le chapitre précédent concerne la validation :

- il aurait fallu déterminer une méthode pour comparer différents modèles et choisir le meilleur

### 5.2.1 Choix du signal d'excitation, traitement des mesures

De manière générale le système que l'on doit identifier doit être excité dans tout le domaine fréquentiel qui nous intéresse. Le signal classiquement utilisé est une séquence pseudo binaire aléatoire (Pseudo Random Binary Signal) tel celui représenté sur la Figure 17.

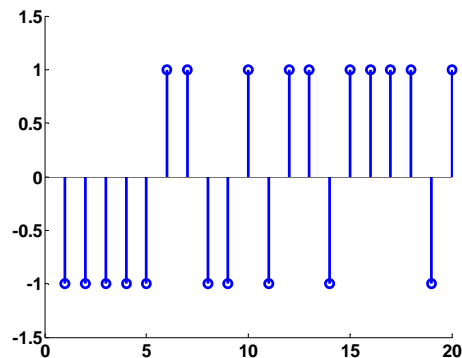


Figure 17 : exemple d'une séquence PRBS

Le signal d'excitation doit se superposer au signal de commande correspondant au point de fonctionnement étudié. Son amplitude doit être faible par rapport à l'amplitude des commandes du système (quelques %).

Une fois le système excité et les mesures effectuées il faut supprimer les valeurs moyennes des mesures.

### 5.2.2 Choix d'une structure de modèle

Le modèle étudié dans le chapitre précédent est de type ARX. On peut le noter sous forme polynomiale :

$$A \cdot y = B \cdot u \quad (37)$$

Ce modèle correspond en fait à un parti-pris sur la manière dont on traite les perturbations. En réalité on prend en compte un signal d'erreurs possibles  $e$  (de moyenne nulle) tels que :

$$A \cdot y = B \cdot u + e \quad (38)$$

Si l'on fait cette hypothèse de modélisation alors la méthode optimale est celle des moindres carrés. Malheureusement cette manière de prendre en compte les perturbations est bien restrictive. D'autres modélisations sont possibles, adaptées à d'autres algorithmes, par exemple le modèle OE :

$$y = \frac{B}{F} \cdot u + e \quad (39)$$

Il est aussi possible de "structurer" le modèle de bruit comme par exemple dans les modèles ARMAX ou BJ :

$$\begin{aligned} A \cdot y &= B \cdot u + C \cdot e \\ y &= \frac{B}{F} \cdot u + \frac{C}{D} \cdot e \end{aligned} \quad (40)$$

La structure du modèle étant choisie il faut choisir l'ordre de ses différents polynômes.

### 5.2.3 Choix d'un critère à minimiser

Dans la plupart des cas le critère à minimiser est l'erreur quadratique entre la sortie et la sortie prédite à un coup (one step ahead prediction). Ceci n'est bien évidemment pas restrictif et peut être généralisé à la prédiction à  $N$  coups.

### 5.2.4 Choix d'un algorithme

Même si chaque structure de modèle est généralement associée à un algorithme "optimal" celui-ci doit généralement se paramétrer.

Prenons par exemple l'algorithme des moindres carrés récursifs tel que présenté dans (36). Cette équation peut être modifiée en intégrant un "facteur d'oubli"  $\lambda$  :

$$\begin{aligned} \theta(t) &= \theta(t-1) + F(t) \cdot \varphi(t) \cdot (y(t) - \theta^T(t-1) \cdot \varphi(t)) \\ \text{avec} & \\ F(t)^{-1} &= \lambda \cdot F(t-1)^{-1} + \varphi(t) \cdot \varphi(t)^T \end{aligned} \quad (41)$$

Ce paramètre (typiquement entre 0.95 et 0.99) permet, comme son nom l'indique, "d'oublier" les anciennes mesures et de se concentrer sur les mesures les plus récentes.

### 5.2.5 Validation

Une fois l'algorithme d'identification appliqué et le modèle obtenu (ce qui prend généralement quelques secondes avec un ordinateur moderne) il reste à expertiser celui-ci. Selon la "qualité" du modèle obtenu on remet en cause l'une ou l'autre des étapes (signal d'excitation, structure du modèle, algorithme) et on réitère jusqu'à obtenir un modèle satisfaisant.

En pratique on essaie un certain nombre de modèles avant de choisir le "meilleur".

Le premier critère de qualité est la capacité du modèle à prédire effectivement la sortie (sortie prédite à un coup, sortie prédite à N coups). Il faut pour cela utiliser un jeu de mesures non utilisé pour l'identification. (En pratique le jeu de mesures initiales est coupé en deux : une partie pour l'identification et une partie pour la validation).

Le second critère de qualité est l'analyse du résidu : la composition spectrale de l'erreur de prédiction.

### 5.2.6 Utilisation de logiciels

En pratique il est commode d'utiliser un célèbre outil de calcul scientifique et sa toolboxe d'identification.

Comme les calculs proprement dit ne prennent que quelques secondes il est facile de réaliser un grand nombre d'identifications et de stocker les différents modèles pour les comparer ensuite. On est rapidement noyé sous le grand nombre de modèles possibles...

## 5.3 Identification "simple"

Il est souvent suffisant d'obtenir un modèle "simple" de type premier ordre et un simple essai indiciel convient souvent. Le système est excité avec un échelon unité et on cherche le meilleur modèle du premier ordre correspondant.

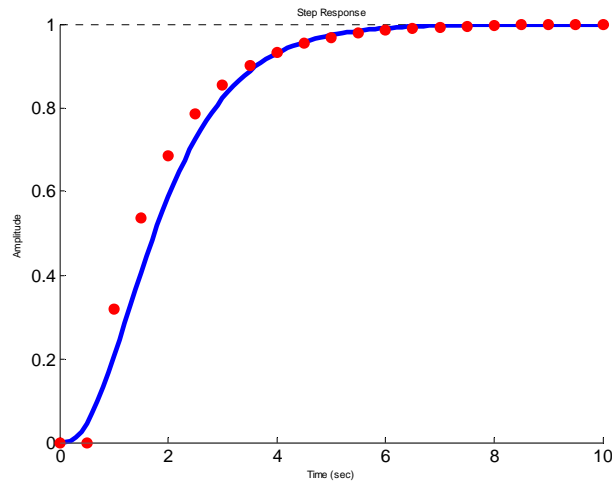


Figure 18 : identification sur un essai indiciel

Dans l'exemple de la Figure 18 on superpose le résultat d'une mesure (en bleu) avec la simulation d'un modèle discret avec une période de 0.5s. Le modèle cherché est du premier ordre avec un retard pur  $d$  :

$$y(z) = \frac{(1 + c_1) \cdot z^{-d}}{1 + c_1 \cdot z^{-1}} \cdot u(z) \quad (42)$$

Il s'agit de trouver l'entier  $d$  et le nombre réel  $c_1$  tels que la réponse indicelle du modèle corresponde pour le mieux aux mesures issues de l'essai. Ceci peut se formaliser par exemple de la manière suivante :

$$(c_1 \quad d_1) = \arg \min \sum_{t=1}^N (y_{\text{modèle}}(t) - y_{\text{mesure}}(t))^2 \quad (43)$$

Cette optimisation peut se faire à la main ou grâce à une procédure automatisée (avec le célèbre outil de calcul scientifique).

Dans le cas où le système comprend naturellement une intégration (par exemple la commande d'un moteur à courant continu par une tension et la mesure de l'angle de rotation) il est utile de prendre en compte explicitement cette intégration dans le modèle :

$$y(z) = \frac{1}{1 - z^{-1}} \cdot \frac{(1 + c_1) \cdot z^{-d}}{1 + c_1 \cdot z^{-1}} \cdot u(z) \quad (44)$$

La procédure est la même mais au lieu de prendre en compte la sortie  $y(t)$  on prend en compte la dérivée  $y'(t)$  :

$$y'(t) = y(t) - y(t-1) \quad (45)$$

## 5.4 Identification non paramétrique

Pour terminer ce panorama rapide sur les méthodes d'identification il faut parler des méthodes non paramétriques : celles qui permettent de tracer la réponse impulsionnelle ou la réponse fréquentielle du système.

### 5.4.1 Estimation de la réponse impulsionnelle

La réponse impulsionnelle d'un système peut être obtenue simplement en appliquant une entrée de type "impulsion" au système et d'en mesurer les conséquences.

La seconde méthode consiste à utiliser la méthode des moindres carrés (ou une méthode dérivée) comme vue au chapitre 5.1. La réponse impulsionnelle d'un système peut en effet être vue comme la fonction de transfert particulière où le dénominateur est de degré nul et le numérateur comporte un nombre infini (en pratique : grand) d'éléments ( $n = 0$  et  $m$  grand) :

$$y(t) = b_0 \cdot u(t) + b_1 \cdot u(t-1) + b_2 \cdot u(t-2) + \dots + b_m \cdot u(t-m) \quad (46)$$

La dernière méthode, qui sera développée ici est l'analyse d'intercorrélation.

Par définition l'intercorrélation  $R_{yu}$  entre deux signaux est :

$$R_{yu}(\tau) = E[(y(t) - E(y(t))) \cdot (u(t-\tau) - E(u(t-\tau)))] \quad (47)$$

Dans cette formule l'espérance mathématique peut être approchée grâce à la loi des grands nombres :

$$R_{yu}(\tau) \approx R_{yu}^N(\tau) = \lim_{N \rightarrow \infty} \frac{1}{N} \sum_{t=1}^N [(y(t) - m_y) \cdot (u(t-\tau) - m_u)] \quad (48)$$

On s'intéresse maintenant à un système tel que décrit par l'équation impulsionnelle (46) et on suppose que le signal d'entrée est un bruit blanc. Le signal d'entrée vérifie donc :

$$\begin{cases} \tau \neq 0 & R_{uu}(\tau) = 0 \\ \tau = 0 & R_{uu}(\tau) = R \end{cases} \quad (49)$$

Il vient alors naturellement :

$$b_k \approx \frac{1}{R} R_{yu}(k) \quad (50)$$

### 5.4.2 Estimation de la réponse fréquentielle

On utilise ici le spectre d'intercorrélation. Par définition le spectre d'intercorrélation entre deux signaux est :

$$\Phi_{yu}(\omega) = \sum_{\tau=-\infty}^{\infty} R_{yu}(\tau) \cdot e^{-i\omega\tau} \quad (51)$$

On suppose que le signal d'entrée  $u(t)$  et le signal de sortie  $y(t)$  sont reliés par une fonction de transfert linéaire du type :

$$y(t) = F \cdot u(t) + v(t) \quad (52)$$

où  $v(t)$  est une perturbation indépendante de  $u(t)$ . On a alors les relations :

$$\Phi_{yu}(\omega) = F(j\omega) \cdot \Phi_{uu}(\omega) \quad (53)$$

Il reste à estimer les spectres d'intercorrélation pour remonter à la réponse fréquentielle  $F(j\omega)$  recherchée. Une idée naturelle est d'utiliser la relation (48) mais cela marche très mal dans presque tous les cas, même en choisissant  $N$  assez grand. L'amélioration est d'utiliser une "fenêtre" pour favoriser les valeurs faibles de  $\tau$  par rapport aux valeurs élevées. La solution classique utilisant la *fenêtre de Hamming* donne :

$$\Phi_{yu}^N(\omega) = \sum_{\tau=-\gamma}^{\gamma} R_{yu}^N(\tau) \cdot w_{\gamma}(\tau) \cdot e^{-i\omega\tau} \quad (54)$$

avec

$$\begin{cases} w_{\gamma}(\tau) = \frac{1}{2} \left( 1 + \cos\left(\frac{\pi \cdot \tau}{\gamma}\right) \right) & |\tau| < \gamma \\ w_{\gamma}(\tau) = 0 & |\tau| > \gamma \end{cases}$$

On utilise ensuite la relation (51) pour estimer la réponse fréquentielle :

$$F(j \cdot \omega) = \frac{\Phi_u^N(\omega)}{\Phi_{yu}^N(\omega)} \quad (55)$$





## 6. SYSTEMES BOUCLES

### 6.1 Fonction de transfert en boucle fermée

Un système et sa commande par ordinateur peut être représenté par un ensemble de fonctions de transfert formant un schéma-bloc plus ou moins compliqué (Figure 19).

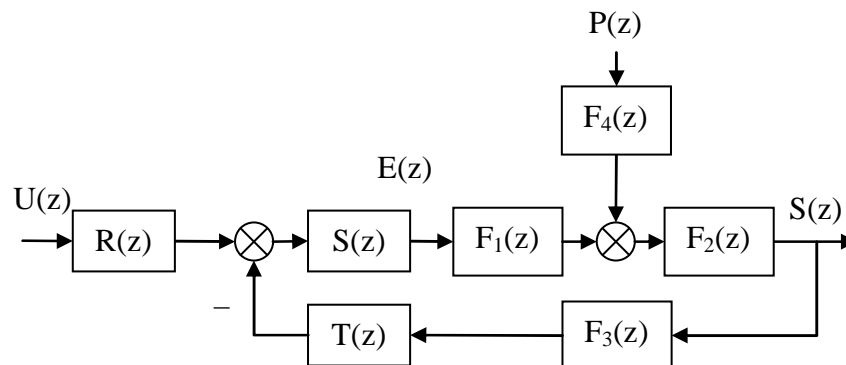


Figure 19 : système discret commandé par ordinateur. Le correcteur est représenté par les fonctions R,S,T, le système par  $F_1$  et  $F_2$ , le modèle de perturbations par  $F_4$  et le modèle des actionneurs par  $F_4$ .

Quelle que soit la structure d'un système asservi il est possible de se ramener à une forme canonique comprenant une fonction de transfert en boucle ouverte en combinant judicieusement les fonctions de transfert du procédé, du régulateur et des perturbations. La Figure 20 représente par exemple le système de la Figure 19 où  $F$  est une combinaison des fonctions de transfert du système, du régulateur, des perturbations et des capteurs.

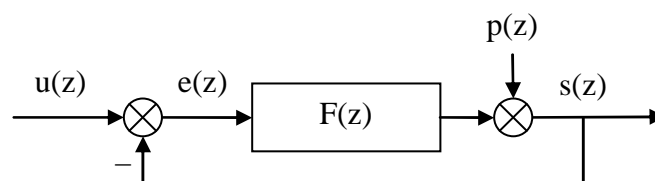


Figure 20 : système discret, notations.

Par défaut on considère une contre-réaction négative et une perturbation additive sur la sortie. La fonction de transfert  $F$  est appelée *fonction de transfert en boucle ouverte*. Comme pour les systèmes continus la *fonction de transfert en boucle fermée* est obtenue simplement à partir de  $F$  (58).

$$F(z) = \frac{s(z)}{e(z)} = \frac{B(z^{-1})}{A(z^{-1})} \quad (56)$$

$$G(z) = \frac{s(z)}{u(z)} = \frac{B(z^{-1})}{A(z^{-1}) + B(z^{-1})}$$

## 6.2 Erreur statique

L'erreur statique du système en boucle fermée, comme pour le système en boucle ouverte, est simplement le rapport des coefficients.

$$G_0 = \frac{\sum_{i=0}^{i=m} b_i}{1 + \sum_{i=0}^{i=m} b_i + \sum_{i=1}^{i=n} a_i} \quad (57)$$

Si l'on souhaite que l'asservissement confère un gain statique unitaire au système en boucle fermée (c'est-à-dire que l'on force la sortie  $s(k)$  à tendre vers la valeur de consigne  $u(k)$ ) il faut que ce rapport de coefficients soit égal à un. On montre facilement que cela est obtenu si la fonction de transfert  $F(z)$  comporte un intégrateur :

$$F(z) = \frac{s(z)}{e(z)} = \frac{1}{1 - z^{-1}} \cdot \frac{B(z)}{A'(z)} \quad (58)$$

## 6.3 Rejet de perturbations

La fonction de transfert entre la perturbation  $p(z)$  et la sortie  $y(z)$  est appelée fonction de sensibilité perturbation-sortie, notée  $S_{yp}$ .

$$S_{yp}(z) = \frac{s(z)}{p(z)} = \frac{A(z^{-1})}{A(z^{-1}) + B(z^{-1})} \quad (59)$$

Si l'on souhaite qu'une perturbation de type échelon soit parfaitement rejetée il faut que  $A(1) = 0$ , on retrouve donc la condition (58) à savoir que le système doit comporter une intégration dans la boucle directe. On peut être en outre être amené à spécifier que les perturbations ne soient pas amplifiées dans une certaine bande de fréquence. On rajoute donc une contrainte du type  $|S_{yp}(e^{j2\pi f})| < S_{\max}$  pour  $f$  appartenant à un certain domaine de  $[0 \ 0.5]$ .

## 6.4 Stabilité des systèmes à simple gain de rétroaction

Les critères de Routh et de Jury (donnés dans le chapitre 3.2.7 p13) ou le calcul numérique des pôles permettent de déterminer la stabilité du système en boucle fermée si l'on connaît sa fonction de transfert.

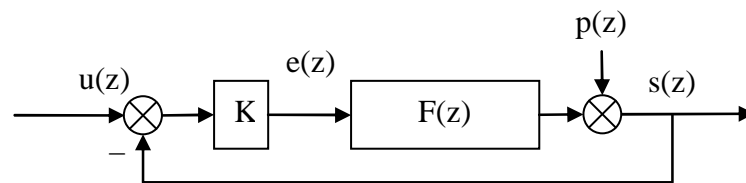


Figure 21 : système bouclé par un simple gain.

On s'intéresse ici au cas particulier où l'asservissement est réalisé par un simple gain de rétroaction  $K$  (Figure 21). On cherche la valeur de  $K$  qui amène le système en limite de stabilité. L'outil adapté ici est le lieu d'Evans ou lieu des racines (root locus) qui permet de visualiser l'évolution des pôles du système bouclé lorsque le gain  $K$  varie.

Prenons l'exemple d'un système comportant trois pôles et deux zéros. L'exemple est donné sous forme de script Matlab :

### Matlab

```
% Définition d'un système discret :
sys=zpk([-2.037 -0.1406],[0.6065 0.3679 0.3679],1,1)
%Tracé du lieu des racines
rlocus(sys);
```

Le script précédent produit le tracé de la Figure 22. On pourrait y lire que la limite de stabilité est obtenue pour  $K=0.2$  environ (détail de la Figure 23).

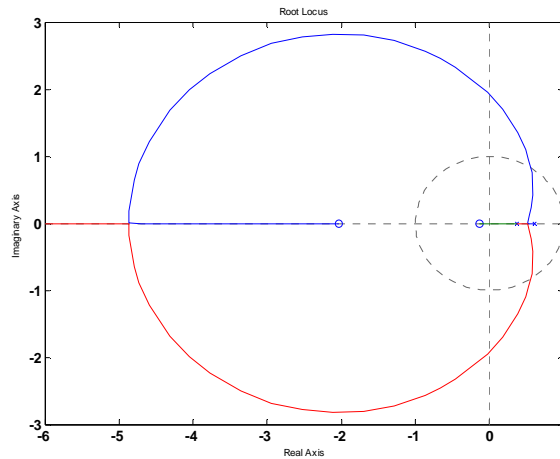


Figure 22 : lieu des racines

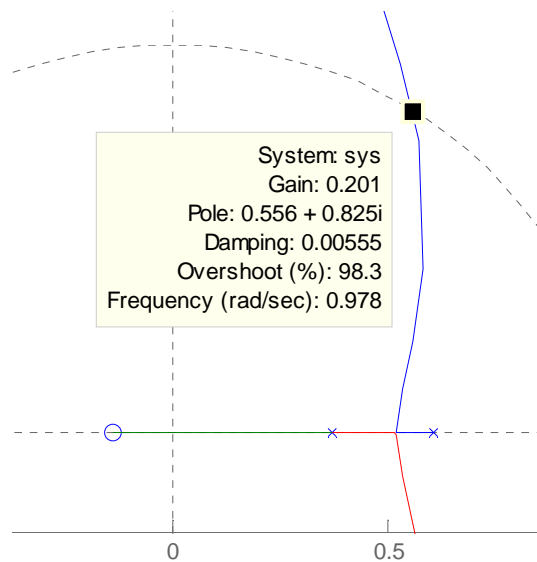


Figure 23 : lieu des racines : détail de la limite de stabilité

On voit sur la Figure 22 que le lieu des racines d'un système échantillonné ressemble à celui d'un système continu :

- on commence par tracer les pôles (×) et les zéros (o) du système en boucle ouverte.
- les pôles du système en boucle fermée décrivent un lieu qui part des pôles (pour  $K = 0$ ) et arrive aux zéros (pour  $K = \infty$ )
- lorsque il y a  $n-m$  plus de pôles que de zéros le système comporte  $n-m$  branches asymptotiques.

On ne décrira pas ici les diverses propriétés du lieu des racines qui permettent de le tracer à la main puisque il est facile à obtenir avec un ordinateur.

Il peut aussi être utilisé pour régler "à la main" le gain permettant d'obtenir un ou plusieurs modes dominants ayant certaines propriétés de fréquence et d'amortissement. On voit sur la Figure 24 que le gain 0.017 permet d'obtenir un pôle double d'amortissement 0.706 et de fréquence 0.65 rad/s et un pôle simple de fréquence 1.63 rad/s.

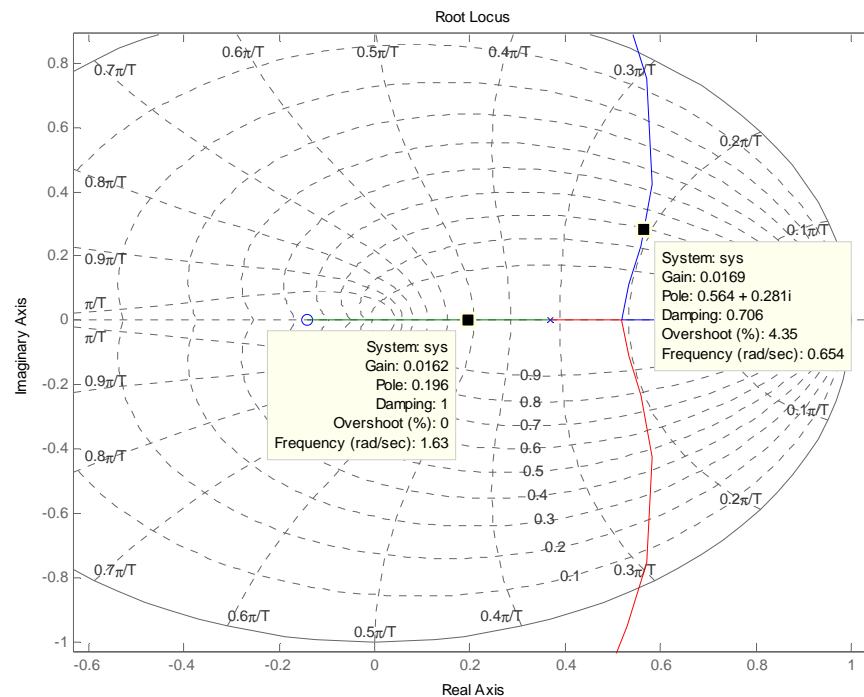


Figure 24 : utilisation du lieu des racines pour régler le gain de la boucle fermée

L'outil rltool de Matlab permet de nombreuses manipulations graphiques de ce type.

## 6.5 Analyse des systèmes bouclés dans le domaine fréquentiel : stabilité

Comme dans le cas continu il est possible de prévoir des propriétés, en particulier la stabilité, d'un système en boucle fermée à partir de l'examen de la fonction de transfert en boucle ouverte.

On ne redémontrera pas ici le théorème de Nyquist mais on se contente de rappeler l'énoncé.

On rappelle que le *lieu de Nyquist* est le lieu de  $F(j2\pi f)$  dans le plan complexe pour  $f$  variant de 0 à  $+\infty$ . A noter que le lieu pour la zone  $[0, 0.5]$  est symétrique de celui pour la zone  $[0.5, 1]$ . Pour cette raison on ne représente généralement que le lieu pour  $f$  appartenant à  $[0, 0.5]$ .

### 6.5.1 Théorème de Nyquist dans le cas où le système est stable en boucle ouverte (critère du revers)

Le système est stable en boucle fermée si le lieu de Nyquist de la fonction de transfert en boucle ouverte parcouru dans le sens des fréquences croissantes de 0 à 0.5 laisse le point critique  $(-1,0)$  à gauche.

La Figure 25 représente par exemple le lieu de Nyquist d'une fonction de transfert en boucle ouverte qui sera stable en boucle fermée.

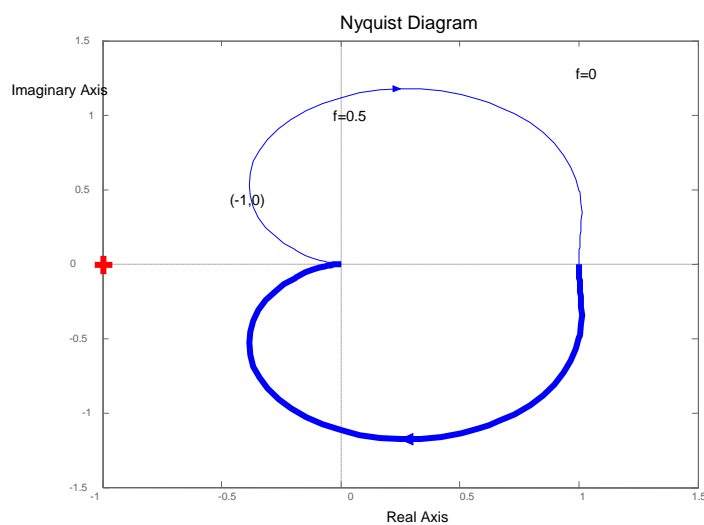


Figure 25 : lieu de Nyquist d'un système stable

### 6.5.2 Théorème de Nyquist dans le cas où le système est instable en boucle ouverte

Si le système en boucle ouverte comporte  $N$  pôles instables il sera stable en boucle fermée si le lieu de Nyquist de la fonction de transfert en boucle ouverte parcouru dans le sens des fréquences croissantes de  $-0.5$  à  $0.5$  entoure le point critique  $(-1,0)$   $N$  fois dans le sens des aiguilles d'une montre..

### 6.5.3 Interprétations du lieu de Nyquist

Comme dans le cas continu, la distance à laquelle on passe du point critique renseigne sur la robustesse du système bouclé. La distance sera exprimée de différente manière (Figure 26) :

- marge de module  $\Delta M$  : plus petite distance entre le lieu des racines et le point critique,

- marge de gain  $\Delta G$
- marge de phase  $\Delta P$

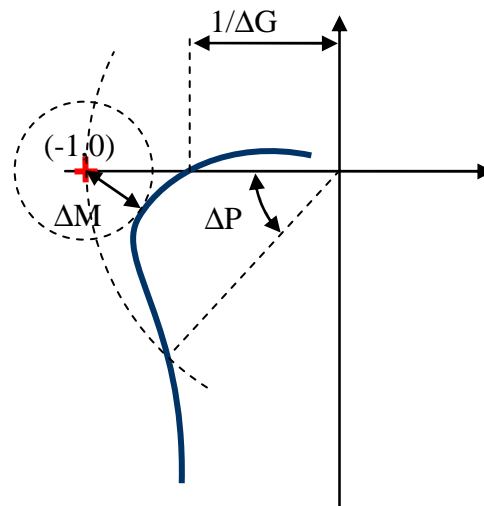


Figure 26 : Marges de stabilité

Les valeurs typiques de ces marges sont données ci-après :

- $\Delta G \geq 6\text{dB}$
- $\Delta M \geq -6\text{dB}$
- $30^\circ \leq \Delta P \leq 60^\circ$

La distance entre le point critique et le lieu des racines est la norme de  $1+F(j2\pi f)$ . On peut donc exprimer la marge de module en fonction de la fonction de sensibilité :

$$\begin{aligned} \Delta M &= \min |1 + F(j \cdot \omega)|_{\omega \in [0, \pi]} \\ \Delta M &= \min \left| 1 + \frac{B(j \cdot \omega)}{A(j \cdot \omega)} \right|_{\omega \in [0, \pi]} = \max \left| \frac{A(j \cdot \omega)}{A(j \cdot \omega) + B(j \cdot \omega)} \right|_{\omega \in [0, \pi]} \quad (60) \\ \Delta M &= \max |S_{yp}(j \cdot \omega)|_{\omega \in [0, \pi]}^{-1} \\ \Delta M_{\text{dB}} &= -\max |S_{yp}(j \cdot \omega)|_{\text{dB}} \end{aligned}$$

Si l'on trace l'évolution de  $S_{yp}$  en fonction de  $f$  on obtient donc la valeur de la marge de module.





# 7. METHODES DE REGULATION DES SYSTEMES ECHANTILLONES

## 7.1 Introduction

Les bases théoriques permettant de concevoir le correcteur pour un système échantillonné ont été posées dans les chapitres précédents. Ce chapitre fait le bilan des connaissances acquises et donne quelques solutions pour calculer des correcteurs à partir d'un cahier des charges précis.

Les systèmes que l'on cherche à commander sont généralement, par nature, continus. La réalisation d'un correcteur par un ordinateur implique que l'on génère une équation de commande discrète (généralement une équation récurrente). L'obtention de cette équation de commande peut se faire de deux manières :

- la première méthode consiste à d'abord synthétiser un correcteur continu (en utilisant les outils classiques de l'automatique et la simulation) et discrétiser ensuite ce correcteur. C'est l'objet du chapitre 7.2.2.
- la seconde méthode consiste à tenir la représentation discrète du système continu associé à son interface numérique (convertisseurs analogique-numérique et numérique-analogique) comme expliqué dans le chapitre 4.5 puis synthétiser directement le correcteur discret. Cela sera expliqué dans le chapitre 7.3 et suivants.

Ce choix peut être résumé par la figure suivante :

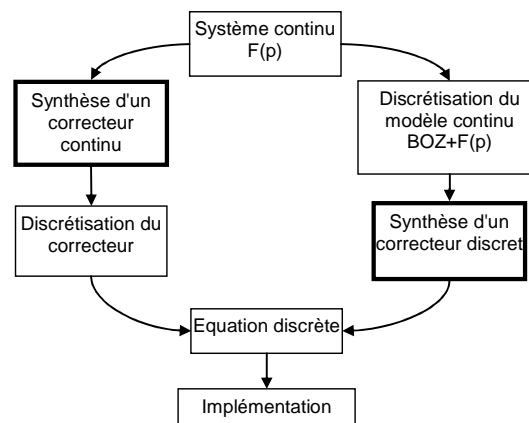


Figure 27 : Réalisation d'un correcteur discret.

Au-delà des affaires de goût, le choix de la méthode sera souvent dicté par l'application. En effet, la première méthode est d'autant meilleure que le correcteur discrétisé approche correctement le correcteur continu. Cela implique que le temps d'échantillonnage soit très faible par rapport aux constantes de temps du système

bouclé. Dans le cas où les contraintes temps réel sont fortes (capacités de calcul partagées par exemple) il sera préférable de synthétiser un correcteur discret.

## 7.2 Structure d'un système commandé par ordinateur, hypothèses

Par la suite on se ramènera toujours à un système tel que celui représenté sur la Figure 28 en cherchant un correcteur  $K(z)$  linéaire. On montre que quel que soit le correcteur linéaire on peut se ramener à la forme représenté sur la Figure 29 (forme RST).

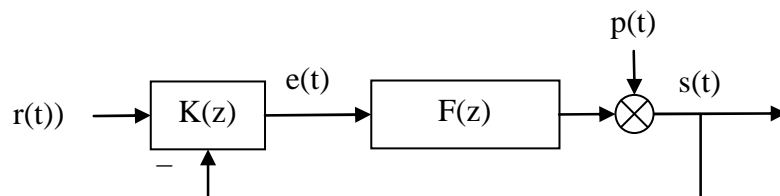


Figure 28 : système bouclé, forme générale.

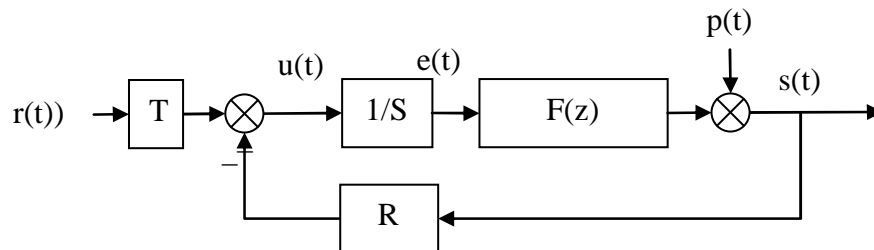


Figure 29 : système bouclé, forme canonique.

### 7.2.1 Discrétisation, choix de la période d'échantillonnage

Du point de vue du ordinateur, l'extérieur (le procédé) est représenté par la fonction de transfert  $F(z)$ . L'hypothèse importante est que la commande (sortie du ordinateur) et la mesure (entrée du ordinateur) sont échantillonnées de manière synchrone à la même période  $T_e$ .

Le choix judicieux de la fréquence d'échantillonnage est très important. En pratique on choisit cette fréquence en fonction de la bande passante du système en **boucle fermée** :

$$f_e = (6 \text{ à } 25) f^{BF} \quad (61)$$

### 7.2.2 Discrétisation d'un correcteur continu

La première méthode pour obtenir les équations d'un correcteur numérique est de faire sa synthèse avec les outils de l'automatique continue puis de discrétiser la fonction de transfert du correcteur continu obtenu (voir pour cela le chapitre 4.2 *Echantillonnage des systèmes continus*).

On va voir par la suite que la synthèse directe d'un correcteur discret offre plus de degré de liberté et permet en général de remplir un cahier des charges plus complet.

## 7.3 Régulateur PID numérique

Le régulateur le plus populaire est le régulateur PID. Nous allons voir dans un premier temps comment on génère un régulateur PID à partir des équations bien connues du régulateur PID continu et comment ces équations peuvent être généralisées pour obtenir plus de degrés de liberté pour le réglage.

### 7.3.1 Cas 1

Un régulateur PID continu a une fonction de transfert du type :

$$K_{\text{PID}}(s) = K \cdot \left( 1 + \frac{1}{T_i \cdot s} + \frac{T_d \cdot s}{1 + \frac{T_d}{N} \cdot s} \right) \quad (62)$$

La discrétisation de ce correcteur continu par la méthode d'Euler (Tableau 4 page 31) se fait en remplaçant la variable de Laplace  $s$  par le terme  $\frac{1-q^{-1}}{q^{-1} \cdot T_e}$ . On obtient :

$$K_{\text{PID}}(q) = K \cdot \left( 1 + \frac{1}{T_i \cdot \frac{1-q^{-1}}{T_e}} + \frac{T_d \cdot \frac{1-q^{-1}}{T_e}}{1 + \frac{T_d}{N} \cdot \frac{1-q^{-1}}{T_e}} \right) \quad (63)$$

En réarrangeant on obtient la forme générale RST (Figure 30):

$$K_{PID}(q) = K \cdot \frac{r_0 + r_1 \cdot q^{-1} + r_2 \cdot q^{-2}}{(1 - q^{-1}) \cdot (1 + s_1' \cdot q^{-1})}$$

$$\begin{cases} r_0 = K \cdot \left( 1 + \frac{T_e}{T_i} + \frac{N \cdot T_d}{T_d + N \cdot T_e} \right) \\ r_1 = -K \cdot \left( 1 + \frac{T_d}{T_d + N \cdot T_e} + \frac{T_e}{T_i} \cdot \frac{T_d}{T_d + N \cdot T_e} + \frac{N \cdot T_d}{T_d + N \cdot T_e} \right) \\ r_2 = K \cdot \left( \frac{T_d}{T_d + N \cdot T_e} \right) \\ s_1' = \frac{T_d}{T_d + N \cdot T_e} \end{cases} \quad (64)$$

Comme on le voit le correcteur obtenu comporte autant de degrés de liberté que le correcteur PID analogique. On reconnaît au dénominateur le terme  $(1 - q^{-1})$  en facteur qui assure l'effet d'intégration.

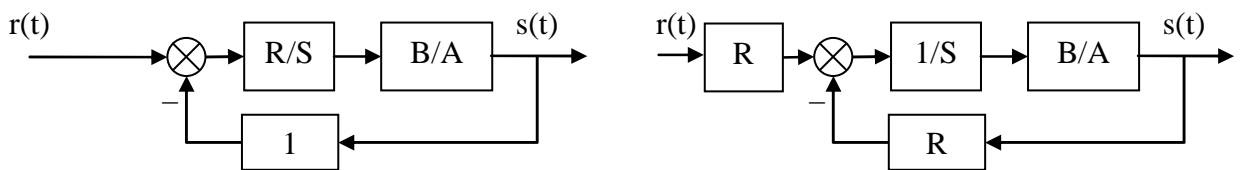


Figure 30 : PID sous forme RST

La fonction de transfert en boucle fermée obtenue est donnée par l'équation suivante :

$$H_{BF} = \frac{B \cdot R}{A \cdot S + B \cdot R} = \frac{B \cdot R}{P} \quad (65)$$

De manière générale le polynôme caractéristique  $P$  fixe les performances visées en boucle fermée. Le calcul de  $R$  et  $S$ ,  $P$  étant fixé et  $A$  et  $B$  étant connus, est réalisé en résolvant l'équation de Bézout  $P = A \cdot S + B \cdot R$ . (voir 9.1 Annexe 1 : Résolution de l'équation polynomiale de Bezout)

On voit que le numérateur  $B \cdot R$  est imposé une fois que  $R$  et  $S$  sont trouvés. Le régulateur introduit donc naturellement des zéros dans la boucle fermée.

Traisons l'exemple entièrement dans le cas où le système est modélisé comme un système du deuxième ordre avec faible retard. La fonction de transfert du procédé est donnée par :

$$H_{BO} = \frac{B}{A} = \frac{b_1 \cdot q^{-1} + b_2 \cdot q^{-2}}{1 + a_1 \cdot q^{-1} + a_2 \cdot q^{-2}} \quad (66)$$

Le correcteur PID rajoutant deux pôles, le polynôme caractéristique en boucle fermée spécifié doit posséder quatre pôles. Le polynôme caractéristique en boucle fermée doit donc être de la forme :

$$\begin{aligned} P &= 1 + p_1 \cdot q^{-1} + p_2 \cdot q^{-2} + p_2 \cdot q^{-3} + p_2 \cdot q^{-4} \\ P &= (1 + p_1' \cdot q^{-1} + p_2' \cdot q^{-2}) \cdot (1 + \alpha_1 \cdot q^{-1}) \cdot (1 + \alpha_2 \cdot q^{-1}) \end{aligned} \quad (67)$$

Et il doit être identifié au produit  $AS + BR$  avec  $R$  et  $S$  ayant les formes données par (64) :

$$\begin{aligned} &(1 + p_1' \cdot q^{-1} + p_2' \cdot q^{-2}) \cdot (1 + \alpha_1 \cdot q^{-1}) \cdot (1 + \alpha_2 \cdot q^{-1}) \\ &= (1 + a_1 \cdot q^{-1} + a_2 \cdot q^{-2}) \cdot (1 - q^{-1}) \cdot (1 + s_1' \cdot q^{-1}) \\ &+ (b_1 \cdot q^{-1} + b_2 \cdot q^{-2}) \cdot (r_0 + r_1 \cdot q^{-1} + r_2 \cdot q^{-2}) \end{aligned} \quad (68)$$

Les coefficients  $p_1'$  et  $p_2'$  définissent un modèle du second ordre aux caractéristiques (pulsation propre  $\omega_0$  et amortissement  $\sigma$  choisis et coefficients déduits conformément au Tableau 3 page 25). Les coefficients  $\alpha_1$  et  $\alpha_2$  correspondent à des pôles auxiliaires aperiodiques. Les choix typiques pour ces coefficients sont :

$$-0.05 < \alpha_1, \alpha_2 < -0.5 \quad (69)$$

On montre que la résolution de (68) est réalisée par une inversion matricielle simple :

$$M \cdot x = p \quad (70)$$

avec

$$\begin{cases} x^T = [1 & s_1' & r_0 & r_1 & r_2] \\ p^T = [1 & p_1 & p_2 & p_3 & p_4] \end{cases}$$

et

$$M = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ a_1' & 1 & b_1 & 0 & 0 \\ a_2' & a_1' & b_2 & b_1 & 0 \\ a_3' & a_2' & 0 & b_2 & b_1 \\ 0 & a_3' & 0 & 0 & b_2 \end{pmatrix} \quad (71)$$

On montre que M est inversible à la condition que A et B soient premiers entre eux, c'est-à-dire que A et B n'aient pas de pôles et de zéros en commun.

(Le problème peut aussi être résolu en utilisant la résolution de l'équation de Bézout).

### 7.3.2 Cas 2

Comme on l'a vu le correcteur PI numérique directement transposé du correcteur PID analogique décrit dans le chapitre 7.3.1 a l'inconvénient de ramener des zéros au dénominateur de la fonction de transfert en boucle fermée (équation 65). Il est possible, grâce à la flexibilité du régulateur numérique d'imposer au correcteur de ne pas ajouter de zéros en boucle fermée. La fonction de transfert désirée en boucle fermée sera donc de la forme :

$$H_{BF} = \frac{P(1)}{B(1)} \cdot \frac{B}{P} \quad (72)$$

Le facteur  $P(1)/B(1)$  garantit le gain statique unitaire du système en boucle fermée (voir page 12). Si le régulateur a la forme générale RST alors les polynômes R S et T doivent vérifier :

$$H_{BF} = \frac{B \cdot T}{A \cdot S + B \cdot R} = \frac{P(1) B}{B(1) P} \quad (73)$$

S et R sont obtenus par la résolution de la même équation que précédemment (68). Le polynôme T est obtenu par simple identification :  $T = R(1)$ .

La structure, sous forme de schéma-bloc est donnée ci-dessous :

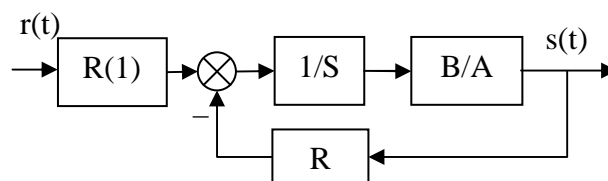


Figure 31 : PID2 sous forme RST

On montre que cette structure de PID, ramené en continu, correspond au PID canonique pour l'action intégrale. L'action proportionnelle et dérivée n'est appliquée que sur le signal y.

La structure du PID2 par rapport au PID1 a pour avantage de ne pas ramener de zéros à la boucle fermée. Cela se traduit par un dépassement moindre en transitoire lorsqu'on applique un échelon de commande. Par contre les deux systèmes réagissent de la même manière à un échelon de perturbation.

## 7.4 Placement de pôles

Les deux structures de PID décrites dans le chapitre précédent sont deux cas particulier d'une stratégie plus générale de placement de pôles.

La synthèse est plus générale que les PID présentés précédemment puisqu'il n'est pas nécessaire de faire d'hypothèses sur le degré de A et B, sur le retard, ni sur la présence ou non de zéros stables ou instables dans le procédé.

La structure du correcteur est la structure type RST telle que présentée dans la Figure 29. Pour la suite des calculs on posera pour la fonction de transfert les conventions suivantes :

$$H_{BO} = \frac{q^{-d} \cdot B(q^{-1})}{A(q^{-1})} \quad (74)$$

$$\begin{cases} A(q^{-1}) = 1 + a_1 \cdot q^{-1} + \dots + a_{n_A} \cdot q^{-n_A} \\ B(q^{-1}) = b_1 \cdot q^{-1} + \dots + b_{n_B} \cdot q^{-n_B} = q^{-1} \cdot B^*(q^{-1}) \end{cases}$$

Compte tenu de la forme RST du régulateur la fonction de transfert en boucle fermée vérifie :

$$H_{BF} = \frac{q^{-d} \cdot B(q^{-1}) \cdot T(q^{-1})}{P(q^{-1})} \quad (75)$$

$$P(q^{-1}) = A(q^{-1}) \cdot S(q^{-1}) + q^{-d} \cdot B(q^{-1}) \cdot R(q^{-1}) = 1 + p_1 \cdot q^{-1} + \dots + p_{n_P} \cdot q^{-n_P}$$

Le polynôme  $P(q^{-1})$  définit donc les pôles de la fonction de transfert en boucle fermée et traduit donc le comportement en *poursuite*.

De la même manière la fonction de transfert entre une perturbation et la sortie (fonction de sensibilité perturbation-sortie) est donnée par :

$$S_{yp} = \frac{A(q^{-1}) \cdot S(q^{-1})}{P(q^{-1})} \quad (76)$$

Ce même polynôme traduit donc aussi le comportement en *régulation*.

Les consignes en poursuite et en régulation peuvent être différentes comme cela est représenté par exemple sur la Figure 32.

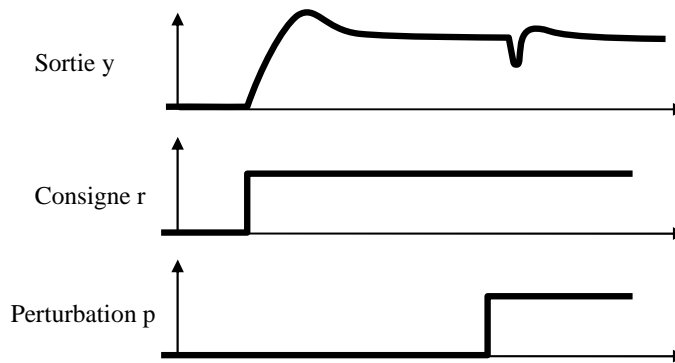


Figure 32 : Performances différentes en régulation et en poursuite

### 7.4.1 Choix des pôles en boucle fermée

De manière générale le comportement en boucle fermée est spécifié sous la forme d'une pulsation propre  $\omega_0$  et d'un amortissement  $\sigma$ , donc une fonction de transfert du second ordre. La spécification étant généralement donnée en continu il est nécessaire de traduire la valeur désirée de  $\omega_0$  et  $\sigma$  en un polynôme  $P_D(q^{-1})$  par discrétisation de la fonction de transfert continue (chapitre 4.5 page 29). Ce polynôme  $P_D$  traduit le comportement dominant du système bouclé.

Si l'on souhaite introduire un effet de filtrage supplémentaire dans certaines bandes de fréquence (réduction des bruits de mesure, adoucissement de la commande, robustesse) cela est décrit par un polynôme  $P_F$  comportant des pôles auxiliaires, plus rapides que les pôles de  $P_D$ .

Le polynôme caractéristique en boucle fermée est donc constitué de pôles dominants et de pôles auxiliaires :

$$P(q^{-1}) = P_D(q^{-1}) \cdot P_F(q^{-1}) \quad (77)$$

### 7.4.2 Régulation : calcul de R et S

Le polynôme  $P(q^{-1})$  étant fixé il s'agit de trouver R et S vérifiant :

$$P(q^{-1}) = A(q^{-1}) \cdot S(q^{-1}) + q^{-d} \cdot B(q^{-1}) \cdot R(q^{-1}) \quad (78)$$

Comme déjà vu au chapitre 7.3.1 on obtient une équation de Bézout qu'il s'agit de résoudre. (Voir aussi 9.1 Annexe 1 : Résolution de l'équation polynomiale de Bezout)

Il est parfois nécessaire de spécifier R et S avec des parties fixes  $H_R$  et  $H_S$ . Dans ce cas l'équation (78) est modifiée et on l'écrit sous la forme:



$$P(q^{-1}) = (A(q^{-1}) \cdot H_S(q^{-1})) \cdot S'(q^{-1}) + (q^{-d} \cdot B(q^{-1}) \cdot H_R(q^{-1})) \cdot R'(q^{-1}) \quad (79)$$

Les inconnues que l'on souhaite trouver sont  $S'$  et  $R'$ . Ceci est nécessaire dans les cas particuliers suivants :

- spécification d'erreur statique nulle,
- rejet d'une perturbation harmonique,
- blocage d'un signal,

Détaillons maintenant ces cas particuliers.

### Spécification d'une erreur statique nulle

On retrouve la nécessité d'avoir un terme intégrateur dans la boucle directe. La spécification se retrouve donc sur  $S$  :

$$S(q^{-1}) = (1 - q^{-1}) \cdot S'(q^{-1}) = H_S(q^{-1}) \cdot S'(q^{-1}) \quad (80)$$

### Rejet d'une perturbation harmonique

Si le système est perturbé par une perturbation harmonique à une fréquence fixe (par exemple 50Hz...) il est possible d'imposer à  $S$  d'avoir un module *nul* ou amorti à cette fréquence. Cela se traduit par la présence dans  $S$  d'une paire de zéros complexe à amortissement nul ou faible à cette fréquence :

$$S(q^{-1}) = (1 + \alpha \cdot q^{-1} + \beta \cdot q^{-2}) \cdot S'(q^{-1}) = H_S(q^{-1}) \cdot S'(q^{-1}) \quad (81)$$

### Blocage d'un signal

Pour certaines application on souhaite que le régulateur "laisse passer" certaines composantes du signal de sortie, c'est-à-dire qu'il ne réagisse pas à ces composantes. Cela se traduit par  $R=0$  à ces fréquences. On retrouve la même forme que précédemment :

$$R(q^{-1}) = (1 + \alpha \cdot q^{-1} + \beta \cdot q^{-2}) \cdot R'(q^{-1}) = H_R(q^{-1}) \cdot R'(q^{-1}) \quad (82)$$

### 7.4.3 Poursuite : calcul de T

Le schéma-bloc de l'asservissement est donné sur la Figure 33. Le terme  $T$  est décomposé en un terme  $T'$  et un terme  $B^*/A^*$ .

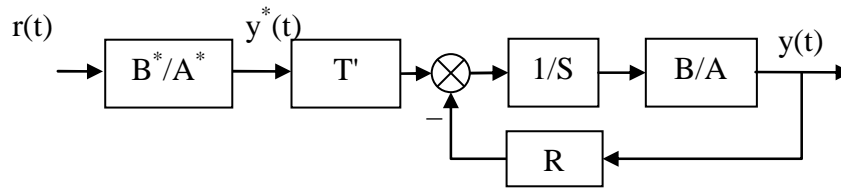


Figure 33 : placement de pôle

L'idée est de régler  $B^*$  et  $A^*$  de telle manière que le signal  $y^*$  ait la dynamique de poursuite désirée (par exemple un second ordre avec pulsation propre et amortissement donné) puis de régler  $T'$  de telle manière que le transfert entre  $y^*$  et  $y$  soit le plus "transparent" possible.

### Réglage de $T'$

Le transfert entre  $y^*$  et  $y$  est :

$$S_{yy^*}(q^{-1}) = T'(q^{-1}) \frac{q^{-d} \cdot B(q^{-1})}{P(q^{-1})} \quad (83)$$

Pour assurer le transfert le plus "transparent" possible il faut donc choisir

$$T'(q^{-1}) = \frac{P(q^{-1})}{B(1)} \quad (84)$$

Remarques :

- Si  $B(1)=0$  alors on remplace  $B(1)$  par 1 dans (84)
- De même dans (84)  $P(q^{-1})$  peut être remplacé par  $P_D$  correspondant à la dynamique dominante.

### Réglage de $B^*$ et $A^*$

$B^*$  et  $A^*$  sont choisis de telle manière que la fonction de transfert  $B^*/A^*$  corresponde à la dynamique de poursuite souhaitée.

## 7.5 Poursuite et régulation à objectifs indépendants

Cette méthode ressemble à la méthode de placement de pôles à ceci près que l'on va effectuer une simplification des zéros de  $B^*(q^{-1})$  : cela ne s'applique donc qu'aux systèmes avec des pôles stables. Plus précisément il faut qu'ils aient un amortissement suffisant ( $> 0,2$ ) de manière à ne pas être trop près du cercle unité.

La structure de l'asservissement est toujours celle donnée sur la Figure 33.

**Choix de P**

$P(q^{-1})$  est choisi de manière à simplifier  $B^*(q^{-1})$  (rappel : la forme de B est donnée dans l'équation 74). Comme indiqué dans le chapitre 7.4.2 page 60 il s'agit donc d'imposer les zéros de  $B(q^{-1})$  dans  $H_R(q^{-1})$  et  $H_S(q^{-1})$ .

**Choix de T**

Si l'on choisit  $T(q^{-1}) = P(q^{-1})$  alors le transfert entre  $y^*$  et  $y$  est :

$$S_{yy^*}(q^{-1}) = q^{-(d+1)} \quad (85)$$

**Réglage de  $B^*$  et  $A^*$** 

$B^*$  et  $A^*$  sont choisis de telle manière que la fonction de transfert  $B^*/A^*$  corresponde à la dynamique de poursuite souhaitée.



## 8. TECHNOLOGIE DE LA COMMANDE PAR ORDINATEUR

Du point de vue de l'automaticien le ordinateur est relié au système physique à piloter par un certain nombre d'intermédiaires plus ou moins gênants :

- les interfaces analogiques et numériques,
- le transport de l'information par bus et protocole,
- l'utilisation éventuelle de capteurs ou d'actionneurs « intelligents ».

Il ne s'agit pas ici de faire un exposé complet sur ces trois aspects mais de poser les problèmes qui se posent à l'automaticien lorsque le système qu'il doit commander est perturbé par ces éléments.

Enfin ce chapitre se termine sur les notions de « Hardware In the Loop » et de « prototypage rapide » qui sont partie intégrante des processus de réalisation de systèmes de contrôle.

### 8.1 Calculateurs et interfaces numériques / analogiques

Le premier problème à prendre en compte est la quantification de l'information. Du point de vue du ordinateur d'une part et du point de vue des convertisseurs d'autre part. Dans un premier temps, considérons les aspects performance de calcul lié à la nature du codage des nombres dans le ordinateur.

#### Les performances liées au types de variables

Prenons l'exemple où le concepteur de la loi de commande souhaite réaliser un correcteur de type proportionnel. Entre l'acquisition des mesures et l'écriture de la consigne il y aura la réalisation d'une de commande du type :

$$y = K \cdot (\text{consigne} - \text{mesure}) \quad (86)$$

En utilisant un langage de programmation évolué (par exemple le C) la ligne de code sera la transcription littérale de (86). Le programmeur devra décider de la définition et du type des variables, principalement du nombre d'octets utilisés pour stocker ses variables. Le Tableau 5 résume par exemple les types utilisés en langage C (norme ANSI)

Tableau 5 : Types ANSI C

Types entiers	char	8 bits
	short int	16 bits
	long int	32 bits
Types flottants (mantisse et exposant)	float	32 bits
	double	64 bits
	longdouble	128 bits

Une fois le code écrit en C l'exécution du programme sur un processeur cible ne peut se faire que par l'intermédiaire d'une compilation du programme. A chaque processeur est donc adapté une version de compilateur.

Les performances du programme (sa rapidité d'exécution) dépendront donc en partie de la qualité du compilateur. (Par exemple, une multiplication s'effectue t'elle bêtement par un décalage de tous les bits ou bien des solutions astucieuses sont-elles implémentées pour arrêter le décalage selon le contexte ?)

Le principal déterminant de la performance est cependant le processeur : son type (8 bits, 16 bits ou 64 bits) et son cadencement (en MHz ou GHz) sont les deux plus importants paramètres. Coder les variables sur 64 bits (double) dans le programme coûtera cher en temps de calcul si le processeur est un 8 bits. Le type de mémoire et leurs temps d'accès peuvent aussi être déterminants. Enfin, certains processeurs disposent d'un processeur spécifiquement dédié aux opérations mathématiques (coprocesseur mathématique).

Il n'est pas possible de dire exactement, pour tel type de compilateur associé à tel type de processeur, le temps que prendra par exemple une multiplication. En pratique on effectue une série de tests types (benchmark) et on affiche un temps moyen d'exécution.

En résumé, on peut être tenté, pour ne pas s'embêter, de coder toutes les variables en virgule flottante. Mais il faut bien savoir que cela a un coût en terme de temps de calcul, surtout si le processeur n'est pas adapté. Pour fixer les idées on peut dire qu'il y a un rapport 20 entre une addition réalisée en "short" et une addition réalisée en virgule flottante.

Enfin, l'utilisation de fonctions mathématiques (sinus, exponentielles, etc...) peut aussi avoir un coût très lourd si le travail n'est pas réalisé par un coprocesseur mathématique dédié. Le compilateur et sa librairie mathématique joue là encore un grand rôle.

### La quantification imposée par le processeur

Au-delà de l'aspect performance et temps de calcul, la quantification de l'information doit parfois aussi être prise en compte. Rappelons qu'un entier codé sur N bits peut prendre seulement  $2^N - 1$  valeurs différentes. Avant d'effectuer les calculs mathématiques dans le corps du programme il est donc judicieux d'appliquer un

facteur d'échelle de manière à utiliser l'amplitude maximale possible (entre 0 et  $2^N - 1$ ).

### **La quantification imposée par les conversions analogique et numérique**

Les convertisseurs analogique-numérique et numérique-analogique sont principalement caractérisés par leur amplitude de signal analogique d'entrée (généralement en volts, par exemple +/- 10V) et par leur résolution en bits (8 ou 10 bits pour les plus courants, jusqu'à 16 bits pour les convertisseurs haut de gamme).

Prenons l'exemple d'un capteur gyromètre transformant une vitesse de rotation en tension. Compte tenu de l'application le signal utile varie entre 2,4V et 2,6V. Si le signal mesuré est branché directement sur un convertisseur 0-10V 10bits alors un problème de conditionnement du signal se pose. En effet la quantification est de  $\frac{10V}{2^{10}-1} = 0,098V$ . Les 0,2V de signal utile seront donc discrétisés en  $\frac{0,2}{0,098} = 20$  intervalles seulement !

Si la discrétisation du signal utile est trop grossière cela peut se traduire par une diminution des performances de l'asservissement, voire une instabilité. Le système n'est en effet plus linéaire et il n'existe pas d'outils théoriques permettant de prévoir les performances du système bouclé.

## **8.2 Transport de l'information : les bus et les protocoles**

Pas question de faire ni même d'ébaucher un cours sur les bus et les protocoles ici. Mais on va quand même donner quelques indications. L'automaticien a l'habitude de simplifier le problème qu'il se pose en dessinant un schéma simplifié de son problème d'asservissement (du type de la Figure 20 page 45) pour pouvoir appliquer les théories de la commande. Cependant il arrive bien souvent que l'information transite à l'intérieur du système, soit entre différents calculateurs effectuant différentes parties du travail soit entre ordinateur et périphériques, par l'intermédiaire de bus informatique. A chaque type de bus correspond un ou plusieurs protocoles qui peuvent poser des problèmes pour l'automaticien :

- les données sont-elles discrétisées ? (le protocole peut par exemple ne permettre de transmettre que des paquets de données de 16 bits),
- le bus induit-t'il un retard ? (le composant mettant en forme les données du ordinateur pour les publier sur le bus a certaines performances qui peuvent induire un délai)
- le bus est-il déterministe ? (Certains bus ne garantissent pas que les données arriveront en un temps donné),
- le bus est-il sûr ? (Certains bus ne garantissent pas non plus que les données arriveront bien)

### **8.3 Capteurs et actionneurs "intelligents", "smart sensors"**

De plus en plus souvent les capteurs et les actionneurs sont associés à leur électronique de puissance, électronique de commande et de conditionnement de signal. L'interface avec le capteur ou l'actionneur se fait donc via un protocole de communication (par exemple via un bus RS232).

L'intérêt est de véhiculer non pas des signaux analogiques, tensions ou courants, mais des signaux numériques beaucoup moins sensibles aux éventuelles perturbations.

On retrouve donc, entre le capteur ou l'actionneur et le ordinateur, la présence d'un bus et d'un protocole de communication.

### **8.4 La philosophie "hardware in the loop"**

Les systèmes embarqués sont de plus en plus complexes et touchent des domaines de plus en plus variés : automobile, satellites, aéronautique, UAV, systèmes d'arme, sous-marins, systèmes, etc... Les systèmes à piloter ainsi que les systèmes de commande sont de plus en plus complexes.

Les techniques de simulation « Hardware In the Loop » (HIL) sont de plus en plus utilisées dans le développement et dans le test de ces systèmes embarqués. L'idée est de développer entièrement le système de contrôle sur une plateforme de test en utilisant le maximum de composants réels et en ne remplaçant que le minimum par la simulation. Par exemple en aéronautique l'« Iron Bird » regroupe dans un hangar tous les systèmes réels de l'avion en cours de développement. Seule la mécanique du vol est simulée : les sorties des actionneurs sont mesurées, rentrées dans un modèle de simulation numérique, puis les mesures effectuées normalement par les capteurs sont calculées et restituées par le modèle.

L'intérêt est de diminuer de manière importante les coûts et les délais en économisant le plus grand nombre possible de tests sur le système réel en situation. Lors des premiers tests réels les techniques « Hardware In the Loop » permettent d'avoir effectué la plus grande part du développement des lois de commande.

### **8.5 Le prototypage rapide en automatique**

Un peu dans le même ordre d'idée que le « Hardware In the Loop » le prototypage rapide permet de réaliser la conception des lois de commande en simulation puis de transporter rapidement (en un simple click !) le code du contrôleur simulé vers la cible temps réel implémentée sur le système réel.

La société Mathworks [9] investit massivement dans ce domaine en ayant enrichi son logiciel de calcul scientifique (Matlab) avec un logiciel de simulation graphique très évolué (Simulink) puis en développant des outils de prototypage rapide associés (Real Time Workshop, Embedded Target For...).



## 9. ANNEXE

### 9.1 Annexe 1 : Résolution de l'équation polynomiale de Bezout

Il s'agit de trouver R et S vérifiant :

$$P(q^{-1}) = A(q^{-1}).S(q^{-1}) + B(q^{-1}).R(q^{-1})$$

Où P, A et B sont connus.

Dans le cas où P, A, B, R et S sont des nombres entiers la solution de l'algorithme est donnée par l'algorithme de division Euclidienne étendu [8].

Dans le cas où P, A, B, R et S sont des polynômes, l'algorithme est donné ci-dessous, programmé en matlab ([1]).

```
function [Rp,Sp,nrp,nsp]=bezoutd(A,B,Hs,Hr,P)
%function [Rp,Sp,nrp,nsp]=bezoutd(A,B,Hs,Hr,P)
%solves AHsSp+BHrRp=P by coefficient comparison. Delay and discretization delay need to be integrated in B
%(zeros at the beginning).
%
%inputs:
%A=[a0 a1 ... aNa] ... vector of model denominator coefficients A=a0 + a1z^(-1) + a2z^(-2) +...+ aNaz^(-Na)
%B=[b0 b1 ... bNb] ... vector of model numerator coefficients B=b0 + b1z^(-1) + b2z^(-2) +...+ bNbz^(-Nb)
%Hs=[hs0 hs1 ... hsNhs] ... vector of controller denominator fixed part Hs=hs0 + hs1z^(-1) +...+ hsNhsz^(-Nhs)
%Hr=[hr0 hr1 ... hrNhr] ... vector of controller denominator fixed part Hr=hr0 + hr1z^(-1) +...+ hrNhrz^(-Nhr)
%P=[p0 p1 ... pNp] ... vector of desired polynomial coefficients P=p0 + p1z^(-1) + p2z^(-2) +...+ pNpz^(-Np)
%outputs:
%Rp=[rp0 rp1 rp2 ...] ... vector of coefficients for resulted controller numerator
%Sp=[sp0 sp1 sp2 ...] ... vector of coefficients for resulted controller denominator
%nrp ... order of Rp
%nsp ... order of Sp
%
%written by: J. Langer, I.D. Landau, H. Prochazka
%7th june 2002

PRECISION=1e-16;

D=size(A);
if D(1)>1, A=A'; end;
```

```

D=size(B);
if D(1)>1, B=B'; end;
D=size(Hs);
if D(1)>1, Hs=Hs'; end;
if D(1)==0, Hs=1; end;
D=size(Hr);
if D(1)>1, Hr=Hr'; end;
if D(1)==0, Hr=1;end;
D=size(P);
if D(1)>1, P=P'; end;

na=length(A)-1;
nb=length(B)-1;
np=length(P)-1;
nhs=length(Hs)-1;
nhr=length(Hr)-1;

if (nhs>0), Ah=real(convz(A,Hs)); else Ah=A*Hs; end; % Ah = A * Hs
nah=length(Ah) -1;
if (nhr>0), Bh=real(convz(B,Hr)); else Bh=B*Hr; end;
nbh=length(Bh) -1;
if (np>nah+nbh-1), disp('Bezout error: too many poles');end;

% increase size of P using zeros if necessary
if (np<nah+nbh-1),
%
% set remaining poles onto a circle with radius rmin
P=[P zeros(1,nah+nbh-1-np)];
rootsPdes=roots(P);
nextra=nah+nbh-1-np;
rmin=1e-16;
angle=[0:nextra-1]'/nextra*2*pi;
j=sqrt(-1);
rootsPextra=rmin*exp(j*angle);
P=poly([rootsPdes;rootsPextra]);
np=nah+nbh-1;

end;
P,

nsp=nbh-1;
nrp=nah-1;

%matrix is smaller than vector PD

```

```

if (np>nah+nbh-1),
    disp('Order of model denominator is too low! Add a polynom of higher order to Hs or Hr. ');
end;

% ns=nsp+nhs
% nr=nrp+nhr

M=[];
for j=1:nsp+1,
    V=[];
    if (j>1), V=[V ; zeros(j-1,1)]; end;% zeros in front of Ah
    V=[V ; Ah'];% Ah
    if (j<=nsp), V=[V ; zeros(nsp+1-j,1)]; end;% zeros behind Ah
    if (length(V)~=nah+nbh), disp('bezoutb: error V'); end;
    M=[M V]; % add one column to M
end;

for j=1:nrp+1,
    V=[];
    if (j>1), V=[V ; zeros(j-1,1)]; end;
    V=[V ; Bh'];
    if (j<=nrp), V=[V ; zeros(nrp+1-j,1)]; end;
    if (length(V)~=nah+nbh), disp('bezoutb: error V'); end;
    M=[M V];
end;

D=size(M);
if (D(1)~=nah+nbh), disp('bezoutb: error size M row'); end;
if (D(2)~=nah+nbh), disp('bezoutb: error size M column'); end;

% make P column vector
P=P';

global M1;
M1=M;

X= M\P;
% coefficients are real values
X=real(X);

% make X row vector
X=X';

```

## Commande des systèmes par ordinateur

```
Sp=X(1:nsp+1);  
Rp=X(nsp+2:nsp+nrp+2);
```

## 10. BIBLIOGRAPHIE

- [1] I. D. Landau, "*Commande des systèmes*", Edition Lavoisier 2002.
- [2] C. Nouals, "Théorie du Signal", Notes de cours ENSICA
- [3] F. Vincent, "*Traitement Numérique du Signal*", Notes de cours ENSICA.
- [4] D. Arzelier, "*Représentation et analyse des systèmes, notes de cours*", Edition ENSICA. Voir aussi <http://www.laas.fr/~arzelier/>
- [5] J. Bordeneuve-Guibé, "*Commande des systèmes linéaires*", Notes de cours ENSICA.
- [6] Y. Briere, "*Identification*", Notes de cours ENSICA.
- [7] L. Ljung, "*System Identification, Theory for the User*", Prentice Hall 1987.
- [8] Wikipedia , équation de Bezout : [http://en.wikipedia.org/wiki/Bézout's\\_identity](http://en.wikipedia.org/wiki/Bézout's_identity)
- [9] <http://www.mathworks.com>



# 11. INDEX

<b>A</b>	
actionneur intelligent .....	21
amortissement .....	15
ANSI C .....	60
Approximation numérique d'équation différentielle .....	28
ARMAX .....	35
<b>B</b>	
BJ35 .....	
Blocage d'un signal .....	56
bloqueurs d'ordre 0 .....	22
BOZ .....	22
bus.....	21, 61
<b>C</b>	
CAN.....	21
capteur intelligent .....	21
causal .....	6, 9
Causalité .....	10
CNA.....	21
compilateur .....	60
convertisseur analogique / numérique.....	21
convertisseur numérique / analogique.....	21
convertisseurs analogique-numérique .....	61
coprocesseur mathématique .....	60
Correspondances pôles - zéros.....	31
<b>D</b>	
Diagramme de Bode .....	18
Discrétisation de fonction de transfert continues .....	27
division polynomiale.....	6
<b>E</b>	
échelon unitaire.....	11
équation d'état .....	17
Erreur statique.....	42
Estimation de la réponse fréquentielle .....	39
Estimation de la réponse impulsionnelle.....	38
Euler .....	29
<b>F</b>	
facteur d'échelle .....	61
facteur d'oubli .....	36
<i>fenêtre de Hamming</i> .....	39
filtre anti-repliement .....	25
fonction de sensibilité .....	42
fonction de transfert discrète.....	9
<b>G</b>	
gain statique.....	11
gain statique unitaire.....	42
<b>I</b>	
Identification.....	32
Identification non paramétrique .....	38
intercorrélation.....	38
<b>L</b>	
lieu des racines.....	43
lieu d'Evans.....	43
LTI.....	8
<b>M</b>	
marge de gain.....	47
marge de module.....	46
marge de phase .....	47
méthode des moindres carrés récursifs .....	32
Méthode des trapèzes.....	29
modèle ARX .....	32
<b>O</b>	
OE.....	35
one step ahead prediction.....	35
opérateur de retard .....	9
<b>P</b>	
période d'échantillonnage (choix) .....	26
Placement de pôles .....	54
Pôles .....	11
Poursuite .....	56
<i>prédiction</i> .....	32
protocoles.....	61
Pseudo Random Binary Signal .....	34
pulsation .....	15
<b>Q</b>	
quantification .....	23, 24, 60
<b>R</b>	
R,S,T.....	41
rapidité.....	15
rayon de convergence .....	5, 9
Régulation.....	55
<b>Fonction de transfert en boucle fermée.....</b>	
41	

Rejet de perturbations .....	42
Rejet d'une perturbation harmonique .....	56
réponse à un échelon.....	11
réponse fréquentielle.....	18
Réponse impulsionnelle.....	10
Réponses indicielle .....	10
retard.....	23
root locus .....	43

**S**

signal discret .....	5
smart sensors.....	62
stabilité.....	11
Systèmes du premier ordre.....	14
Systèmes du second ordre.....	15

**T**

Théorème de Nyquist.....	46
théorème de Shannon.....	19, 24
théorème du retard .....	6
Transformée en z .....	5
transformée unilatérale .....	5
Tustin.....	29

**V**

virgule flottante.....	60
------------------------	----

**Z**

zero holder hold .....	23
zéros.....	11
zoh .....	23



# Commande des systèmes par ordinateur

Yves Briere  
ENSICA

Notes de cours  
2007



# TABLE DES MATIERES

Table des matières .....	1
1. Introduction .....	1
2. Rappels et prérequis .....	3
3. Signaux et systèmes discrets .....	5
3.1 Signaux discrets .....	5
3.1.1 Transformée en z .....	5
3.1.2 Signaux types .....	7
3.2 Systèmes discrets .....	9
3.2.1 Fonction de transfert discrète, définition .....	9
3.2.2 Notation "opérateur de retard $q^{-1}$ " .....	10
3.2.3 Réponse impulsionnelle .....	11
3.2.4 Causalité .....	11
3.2.5 Réponses indicielle .....	11
3.2.6 Régime stationnaire (gain statique) .....	12
3.2.7 Pôles, zéros, stabilité .....	12
3.2.8 Systèmes du premier ordre .....	15
3.2.9 Systèmes du second ordre .....	16
3.2.10 Calcul itératif de la réponse d'un système .....	18
3.2.11 Représentation par équation d'état .....	18
3.3 Réponse fréquentielle des systèmes échantillonnés .....	19
3.3.1 Diagramme de Bode d'un système échantillonné .....	19
3.3.2 Où l'on voit apparaître le théorème de Shannon .....	20
4. Systèmes continus échantillonnés .....	23
4.1 Structure des systèmes commandés par ordinateur .....	23
4.2 Échantillonnage des systèmes continus .....	24
4.3 Effets de l'échantillonnage .....	25
4.4 Où l'on revoit le théorème de Shannon .....	26
4.5 Discrétisation de fonction de transfert continues .....	29
4.5.1 Approximation numérique d'équation différentielle .....	30
4.5.2 Superposition des réponses indicielles .....	32
4.5.3 Correspondances pôles - zéros .....	33
4.5.4 Bilan .....	33
5. Identification des systèmes échantillonnés .....	35
5.1 Identification d'un modèle ARX par la méthode des moindres carrés récurrents .....	35
5.2 L'identification paramétrique en général et en pratique .....	37

5.2.1	Choix du signal d'excitation, traitement des mesures .....	37
5.2.2	Choix d'une structure de modèle.....	38
5.2.3	Choix d'un critère à minimiser.....	38
5.2.4	Choix d'un algorithme.....	38
5.2.5	Validation.....	39
5.2.6	Utilisation de logiciels .....	39
5.3	Identification "simple" .....	39
5.4	Identification non paramétrique .....	41
5.4.1	Estimation de la réponse impulsionnelle .....	41
5.4.2	Estimation de la réponse fréquentielle.....	42
6.	Systèmes bouclés .....	45
6.1	Fonction de transfert en boucle fermée .....	45
6.2	Erreur statique .....	46
6.3	Rejet de perturbations.....	46
6.4	Stabilité des systèmes à simple gain de rétroaction.....	47
6.5	Analyse des systèmes bouclés dans le domaine fréquentiel : stabilité.....	49
6.5.1	Théorème de Nyquist dans le cas où le système est stable en boucle ouverte (critère du revers) .....	50
6.5.2	Théorème de Nyquist dans le cas où le système est instable en boucle ouverte .....	50
6.5.3	Interprétations du lieu de Nyquist.....	50
7.	Méthodes de régulation des systèmes échantillonnés .....	53
7.1	Introduction .....	53
7.2	Structure d'un système commandé par ordinateur, hypothèses.....	54
7.2.1	Discretisation, choix de la période d'échantillonnage.....	54
7.2.2	Discretisation d'un correcteur continu .....	55
7.3	Régulateur PID numérique .....	55
7.3.1	Cas 1 .....	55
7.3.2	Cas 2 .....	58
7.4	Placement de pôles .....	59
7.4.1	Choix des pôles en boucle fermée .....	60
7.4.2	Régulation : calcul de R et S.....	60
7.4.3	Poursuite : calcul de T .....	61
7.5	Poursuite et régulation à objectifs indépendants .....	62
8.	Technologie de la commande par ordinateur.....	65
8.1	Calculateurs et interfaces numériques / analogiques.....	65
8.2	Transport de l'information : les bus et les protocoles .....	67
8.3	Capteurs et actionneurs "intelligents", "smart sensors" .....	68
8.4	La philosophie "hardware in the loop" .....	68
8.5	Le prototypage rapide en automatique .....	68
9.	Annexe .....	69
9.1	Annexe 1 : Résolution de l'équation polynomiale de Bezout.....	69
10.	Bibliographie.....	73
11.	Index.....	1





# 1. INTRODUCTION



Figure 1 : commande d'un systèmes de télémédecine par ordinateur, vu par Claude Serre





## 2. RAPPELS ET PREREQUIS

Ce polycopié ne comporte pas de rappels sur les systèmes continus. S'adressant aux étudiants en deuxième année de l'ENSICA il suppose connu les enseignements de première année et de seconde année en automatique et traitement du signal.

En première année le cours de théorie du signal [2] a rendu familier les notions de :

- transformée de Fourier, transformée de Laplace
- corrélation, convolution

Le cours de traitement numérique du signal [3] a abordé en plus :

- la discrétisation et la quantification des signaux.
- la transformée en z et la transformée de Fourier discrète.
- les filtres numériques RIF et RII.

En automatique [4] les bases de la représentation et de l'analyse des systèmes continus ont été posées. On ne reviendra pas dans ce cours sur les notions de :

- modèles mathématique des systèmes linéaires à temps continus à temps invariant (LTI) : fonction de transfert, équation d'état, pôles, zéros et dynamique.
- commandabilité et observabilité.
- stabilité des systèmes en boucle ouverte : pôles, critères de Routh-Hurwitz.
- stabilité des systèmes bouclés : critère de Nyquist, critère de Bode.
- lieu des racines.

En deuxième année les notions suivantes sont acquises [5] :

- synthèse des correcteurs par méthodes fréquentielles (avance et retard de phase, PID)
- commande par retour d'état (placement de pôles), commande Linéaire Quadratique (LQG)
- reconstruction de l'état par Observateur, observateur optimal de Kalman.



## 3. SIGNAUX ET SYSTEMES DISCRETS

### 3.1 Signaux discrets

#### 3.1.1 Transformée en z

Pour une approche mathématiquement rigoureuse de représentation des signaux discrets par la transformée en z consulter [3] et sa bibliographie. Dans ce cours on se cantonne à une approche utilitariste des outils et méthodes mathématiques tout en essayant de ne pas contredire fondamentalement la rigueur mathématique.

Un signal discret  $s(k)$  est une séquence de nombres réels appelés échantillons. L'indice  $k$  est généralement associé à un instant d'échantillonnage :  $t = k.T_e$ . La période d'échantillonnage  $T_e$  est supposée constante. Par définition la transformée en z du signal  $s(k)$  est la fonction de la variable  $z$ , notée abusivement  $s(z)$  :

$$s(z) = Z(s(k)) = \sum_{k=0}^{\infty} s(k) \cdot z^{-k} \quad (1)$$

La transformée définie ici est plus précisément appelée transformée unilatérale puisque les termes correspondant à  $k < 0$  ne sont pas pris en compte. Ceci n'a pas d'importance puisque les signaux que nous utilisons sont systématiquement causaux<sup>1</sup> :  $s(k) = 0$  pour  $k < 0$ .

Il y a bien évidemment un lien avec la transformée de Laplace et la transformée de Fourier (voir [3])

La fonction de la variable complexe  $s(z)$  est une série et ne converge pas forcément pour toute valeur de  $z$ . En toute rigueur il faudrait associer le rayon de convergence à chaque fonction en  $z$  :  $\{s(z) \text{ converge pour } |z| > \alpha\}$ .

Les automaticiens pour qui la transformée en z n'est bien souvent qu'une façon bien commode de regrouper des termes de récurrence utilisent souvent la pseudo-variable<sup>2</sup>  $q$ .

A partir d'une transformée en z  $s(z)$  il est possible de retrouver la séquence  $s(k)$  correspondante en appliquant la transformation inverse, dont on ne donnera pas la formule ici. A la place de faire appel à cette formule compliquée il est possible d'utiliser d'autres méthodes pour retrouver la séquence  $s(k)$  lorsque la fonction  $s(z)$  est donnée sous forme de fraction rationnelle :

<sup>1</sup> Pour une justification du terme « causal » voir page 11

<sup>2</sup> Voir page 10

- division polynomiale
- décomposition de la fraction rationnelle en éléments simples puis utilisation de transformations élémentaires telles que celles données dans le Tableau 2

La propriété fondamentale de la transformée en  $z$  est qu'elle est linéaire (la transformée en  $z$  d'une composition linéaire de signaux est la composition linéaire de la transformée en  $z$  de chaque signal). Les autres propriétés fondamentales sont données dans le Tableau 2.

Tableau 1 : propriétés de la transformée en  $z$

1	$s(z) = Z(s(k)) = \sum_{k=0}^{\infty} s(k) \cdot z^{-k}$
2	$Z(\alpha \cdot s_1(k) + \beta \cdot s_2(k)) = \alpha \cdot Z(s_1(k)) + \beta \cdot Z(s_2(k))$
3	$Z(k \cdot s(k)) = -z \cdot \frac{ds(z)}{dz}$
4	$Z(c^k \cdot s(k)) = s\left(\frac{z}{c}\right)$
5	$Z(s(k-1)) = s(k=-1) + z^{-1} \cdot s(z)$
6	$Z(s(k+1)) = z \cdot s(z) - z \cdot s(k=0)$
7	$s(k=0) = \lim_{z \rightarrow \infty} (s(z))$
8	$s(k \rightarrow \infty) = \lim_{z \rightarrow 1} \left( \frac{z-1}{z} \cdot s(z) \right)$

Remarque 1 : la propriété 5 ne prend pas en compte le fait que le signal est causal : en pratique  $s(k=-1) = 0$ .

Remarque 2 : la propriété 5 justifie le fait que le terme  $z^{-1}$  soit appelé "opérateur de retard unitaire", cette propriété est aussi appelée "théorème du retard".

Remarque 3 : les propriétés 7 et 8 sont données sous réserve de convergence.

Transformées inverses

### Exemple 1 : transformée inverse par division polynomiale

On considère un signal  $s(k)$  dont la transformée en  $z$  est donnée par la fraction rationnelle :

$$S(z) = \frac{0.1 \cdot z^2 + 0.1 \cdot z}{z^3 - 2.6 \cdot z^2 + 2.2 \cdot z - 0.6} \quad (3.1)$$

L'objectif est d'obtenir  $S(z)$  sous forme polynomiale :

$$S(z) = x_0 + x_1 \cdot z^{-1} + x_2 \cdot z^{-2} + \dots \text{etc.} \quad (3.2)$$

Cette suite est généralement infinie, la réponse temporelle est donnée par la succession des termes  $\{x_i\}$ . En traitant l'équation (3.1) comme une simple division l'algorithme de division classique (celui appris en CM2) permet d'obtenir par récurrence tous les termes de la suite (3.2) :

$0.1 \cdot z^2$	$+ 0.1 \cdot z$				$z^3 - 2.6 \cdot z^2 + 2.2 \cdot z - 0.6$
$0.1 \cdot z^2$	$- 0.26 \cdot z$	$+ 0.22$	$- 0.06 \cdot z^{-1}$		
$0$	$0.36 \cdot z$	$- 0.22$	$0.06 \cdot z^{-1}$		
	$0.36 \cdot z$	$- 0.936$	$0.792 \cdot z^{-1}$	$- 0.216 \cdot z^{-2}$	
$0$	$0.716$	$- 0.732 \cdot z^{-1}$	$\dots$	$\dots$	

Les éléments successifs de la suite  $s(k)$  sont donc  $s(k) = \{0, 0.1, 0.36, 0.716, \dots\}$ . Pour obtenir les termes suivants il faudrait continuer la division.

**Exemple 2 : transformée inverse par décomposition en éléments simples**

La deuxième méthode présentée consiste à décomposer la fraction rationnelle (3.1) en éléments simples puis d'utiliser les tables de transformées en  $z$  (Tableau 2). Pour notre exemple cela donne :

$$S(z) = \frac{0.1 \cdot z^2 + 0.1 \cdot z}{z^3 - 2.6 \cdot z^2 + 2.2 \cdot z - 0.6} = \frac{-z}{z-1} + \frac{0.5 \cdot z}{(z-1)^2} + \frac{z}{z-0.6}$$

La réponse temporelle est donc composée des trois éléments :

$$s(k) = -(1 \ 1 \ 1 \ \dots) + 0.5 \cdot (1 \ 2 \ 3 \ \dots) + (1 \ 0.6 \ 0.6^2 \ \dots)$$

$$\text{soit : } s(k) = \begin{cases} s(0) = -1 + 0.5 + 1 \\ s(2) = -1 + 0.5 \cdot 2 + 0.6 \cdot 0.5 \\ s(3) = -1 + 0.5 \cdot 3 + 0.6^2 \\ s(4) = \dots \end{cases}$$

**3.1.2 Signaux types**

Le Tableau 2 donne les transformée en  $z$  de certains signaux fondamentaux. La Figure 2 représente l'allure de quelques signaux types.

Tableau 2 : transformée en z élémentaires

$s(k)$	$S(z)$
signal temporel	Transformée en z
$\delta(k)$ : impulsion unité	1
$u(k)$ : échelon unité	$\frac{z}{z-1}$
$k \cdot u(k)$	$\frac{z}{(z-1)^2}$
$c^k \cdot u(k)$	$\frac{z}{z-c}$
$k \cdot a^k \cdot u(k)$	$\frac{a \cdot z}{(z-a)^2}$
$\sin(\omega \cdot k) \cdot u(k)$	$\frac{z \cdot \sin(\omega)}{z^2 - 2 \cdot z \cdot \sin(\omega) + 1}$
$\cos(\omega \cdot k) \cdot u(k)$	$\frac{z \cdot (z - \cos(\omega))}{z^2 - 2 \cdot z \cdot \sin(\omega) + 1}$
$a^k \cdot \sin(\omega \cdot k) \cdot u(k)$	$\frac{a \cdot z \cdot \sin(\omega)}{z^2 - 2 \cdot a \cdot \cos(\omega) \cdot z + a^2}$
$a^k \cdot \cos(\omega \cdot k) \cdot u(k)$	$\frac{z \cdot (z - a \cdot \cos(\omega))}{z^2 - 2 \cdot a \cdot \cos(\omega) \cdot z + a^2}$

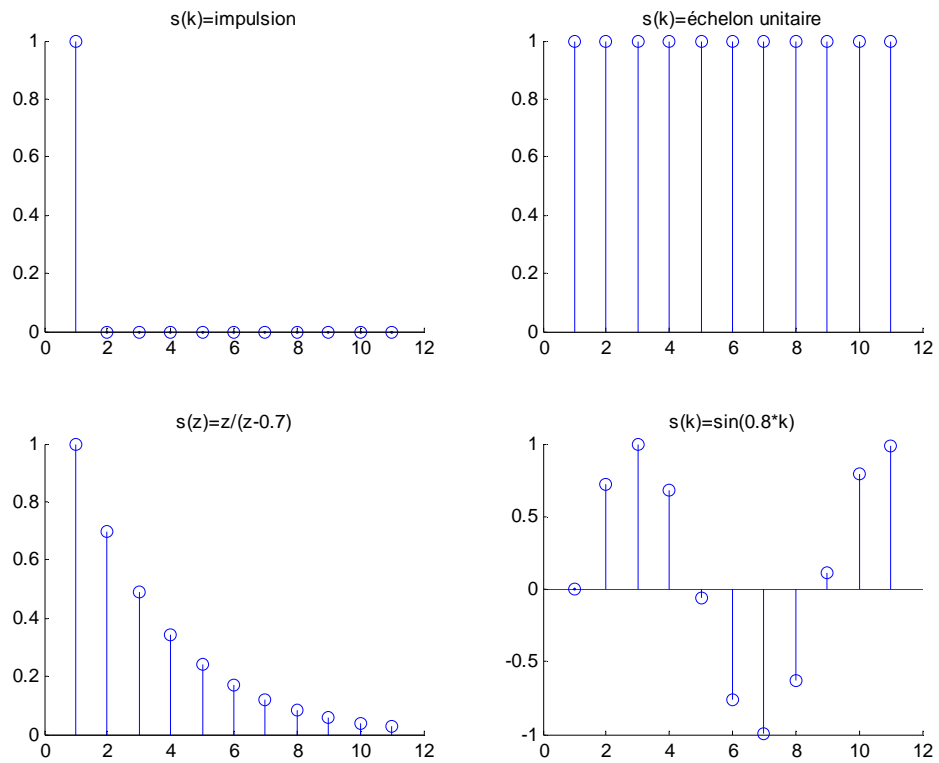


Figure 2 : réponses de quelques signaux types

## 3.2 Systèmes discrets

### 3.2.1 Fonction de transfert discrète, définition

On considère maintenant un système discret linéaire et invariant dans le temps (LTI) représenté sur la Figure 3.

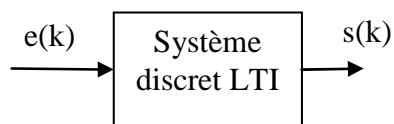


Figure 3 : système discret, notations

La relation entre la séquence d'entrée et la séquence de sortie est linéaire et définie par un ensemble de coefficients constants  $a_{1:n}$  et  $b_{1:m}$  :

$$\begin{aligned} s(k) + a_1 \cdot s(k-1) + a_2 \cdot s(k-2) + \dots + a_n \cdot s(k-n) = \\ b_0 \cdot e(k) + b_1 \cdot e(k-1) + b_2 \cdot e(k-2) + \dots + b_m \cdot e(k-m) \end{aligned} \quad (2)$$

On peut appliquer la transformée en  $z$  à chaque membre de l'équation puis en appliquant en récurrence la propriété 5 (théorème du retard) on obtient la fonction de transfert discrète du système :

$$\frac{s(z)}{e(z)} = F(z) = \frac{b_0 + b_1 \cdot z^{-1} + b_2 \cdot z^{-2} + \dots + b_m \cdot z^{-m}}{1 + a_1 \cdot z^{-1} + a_2 \cdot z^{-2} + \dots + a_n \cdot z^{-n}} \quad (3)$$

Remarque 1 : la fonction de transfert précédente est normalisée en prenant  $a_0 = 1$ .

Remarque 2 : on remarquera que la fonction de transfert est donnée en  $z^{-1}$  plutôt qu'en  $z$ . Cela est justifié par le fait que celle-ci traduit l'équation d'évolution du système physique causal (2).

Le script Matlab permettant de créer un objet Matlab correspondant à une fonction de transfert discrète est donné ci-dessous.

#### Matlab

```
% Création d'un système linéaire discret :
Te=0.1;
sysd = tf([b0 b1 b2],[1 a0 a2 a3],Te,'variable','z^-1');
```

On notera l'utilisation de l'argument 'variable', 'z^-1'.

### 3.2.2 Notation "opérateur de retard $q^{-1}$ "

La fonction de transfert est une fonction de la variable complexe  $z$ . Il faudrait donc en toute rigueur l'associer à son rayon de convergence. En pratique on ne s'embarrasse jamais de préciser cette zone de convergence. C'est pourquoi on trouve souvent la notation utilisant l'opérateur  $q^{-1}$  de décalage à la place de la variable complexe  $z^{-1}$  (4).  $F(q)$  n'est alors plus à proprement parler une fonction mais on lui appliquera les mêmes manipulations algébriques qu'à  $F(z)$ .

$$F(q) = \frac{b_0 + b_1 \cdot q^{-1} + b_2 \cdot q^{-2} + \dots + b_m \cdot q^{-m}}{1 + a_1 \cdot q^{-1} + a_2 \cdot q^{-2} + \dots + a_n \cdot q^{-n}} \quad (4)$$

Attention : selon les auteurs l'opérateur qui remplace  $z^{-1}$  est noté  $q$  (Matlab) ou  $q^{-1}$  (Landau [1]). Dans la suite du document c'est cette dernière convention qui sera appliquée.



L'utilisation de l'opérateur de retard  $q^{-1}$  à la place de la variable complexe  $z^{-1}$  permet de traiter du cas des fonctions de transfert à coefficients non constant, cas non prévu par la transformation en  $z$ .

### 3.2.3 Réponse impulsionnelle

Essayons d'appliquer au système un signal d'entrée particulier : une impulsion unitaire. La transformée en  $z$  du système est obtenue en multipliant la fonction de transfert par la transformée en  $z$  de l'impulsion unitaire : 1. On obtient :

$$s(z) = \frac{b_0 + b_1 \cdot z^{-1} + b_2 \cdot z^{-2} + \dots + b_m \cdot z^{-m}}{1 + a_1 \cdot z^{-1} + a_2 \cdot z^{-2} + \dots + a_n \cdot z^{-n}} \times 1 \quad (5)$$

Cela justifie le fait que la fonction de transfert du système peut aussi s'appeler "réponse impulsionnelle".

#### Matlab

```
% Réponse impulsionnelle d'un système :
[y,t] =impulse(sysd);
```

### 3.2.4 Causalité

Un système est dit causal si les sorties ne dépendent que du passé et non pas du futur. La conséquence est que la réponse impulsionnelle de  $F(z)$  ne peut pas comporter de terme non nuls pour  $k < 0$ . Cela justifie la confusion sémantique entre "système causal" et "signal causal", c'est-à-dire ne comportant pas de termes non nuls pour  $k < 0$ .

Attention : à la différence des systèmes continus un système physique (causal) discret peut très bien être modélisé avec un degré  $m$  supérieur au degré  $n$  : le système peut avoir plus de zéros que de pôles.

La transformée en  $z$  représentant des signaux échantillonnés elle peut tout aussi bien représenter les mesures prises à différents endroits d'une poutre. Le temps étant remplacé ici par une abscisse la notion de causalité ne s'applique pas ! Attention : beaucoup des propriétés de systèmes échantillonnés que l'on étudie sont liés à la causalité, prudence donc si l'on applique ces méthodes à d'autres domaines !

### 3.2.5 Réponses indicielle

Pour caractériser des systèmes on s'intéresse souvent à leur réponse impulsionnelle ou, plus souvent, à leur réponse indicielle. Pour calculer la réponse

à un échelon d'un système il faut donc multiplier la fonction de transfert par la transformée en z d'un échelon unitaire :

$$s(z) = \frac{b_0 + b_1 \cdot z^{-1} + b_2 \cdot z^{-2} + \dots + b_m \cdot z^{-m}}{1 + a_1 \cdot z^{-1} + a_2 \cdot z^{-2} + \dots + a_n \cdot z^{-n}} \times \frac{1}{1 - z^{-1}} \quad (6)$$

Il faudrait ensuite calculer la transformée en z inverse de s(z). En pratique, si l'on doit faire le calcul à la main on décompose en éléments simples puis on utilise des tables. Avec Matlab c'est plus simple. Voici par exemple le script pour obtenir les 20 premiers échantillons de la réponse indicielle d'un système :

**Matlab**

```
% Réponse indicielle d'un système :
[y,t] =step(sysd,20);
```

### 3.2.6 Régime stationnaire (gain statique)

Le gain statique est obtenu en appliquant une entrée indicielle et en calculant la valeur du régime permanent. En appliquant le théorème de la valeur finale il vient :

$$F_0 = \lim_{z \rightarrow 1} \left( \frac{z-1}{z} \cdot \frac{b_0 + b_1 \cdot z^{-1} + b_2 \cdot z^{-2} + \dots + b_m \cdot z^{-m}}{1 + a_1 \cdot z^{-1} + a_2 \cdot z^{-2} + \dots + a_n \cdot z^{-n}} \times \frac{1}{1 - z^{-1}} \right) \quad (7)$$

Soit :

$$F_0 = \frac{\sum_{i=0}^{i=m} b_i}{1 + \sum_{i=1}^{i=n} a_i} \quad (8)$$

Le gain statique est obtenu donc simplement en faisant le quotient de la somme des coefficients.

### 3.2.7 Pôles, zéros, stabilité

Comme pour les systèmes continus la dynamique et la stabilité des systèmes dépend des propriétés du dénominateur de la fonction de transfert.

$$F(z) = \frac{e_0 \cdot (1 + e_1 \cdot z^{-1}) \cdot (1 + e_2 \cdot z^{-1}) \cdots (1 + e_m \cdot z^{-1})}{(1 + c_1 \cdot z^{-1}) \cdot (1 + c_2 \cdot z^{-1}) \cdots (1 + c_n \cdot z^{-1})} \quad (9)$$

Les coefficients  $c_i$  sont les pôles de la fonction de transfert, les coefficients  $e_i$  sont les zéros. La fonction de transfert étant au départ composée de coefficients purement réels les coefficients  $c_i$  sont soit réels soit imaginaires par paires conjuguées.

Une fonction de transfert est donc aussi bien définie par la donnée de ses pôles et de ses zéros. Le script Matlab pour créer un système ainsi est donné ci-dessous :

#### Matlab

```
% Système discret défini par ses pôles et zéros :
Te=0.1;
sysd=zpk([c1 c2 c3],[e1 e2],e0,Te);
```

Si l'on décompose la fonction de transfert en éléments simples on obtient une somme d'éléments du type :

$$F(z) = \frac{d_1}{1 + c_1 \cdot z^{-1}} + \frac{d_2}{1 + c_2 \cdot z^{-1}} + \cdots + \frac{d_n}{1 + c_n \cdot z^{-1}} \quad (10)$$

Un système linéaire temps invariant est dit stable si sa réponse impulsionnelle tend asymptotiquement vers 0. Compte tenu de (10) cela nécessite que chaque pôle  $c_i$  soit de module inférieur à 1.

**Propriété** : un système discret LTI est stable si et seulement si tous ses pôles ont leur modules inférieur ou égal à 1.

#### Critère algébrique attestant de la stabilité : critère de Routh

On rappelle que le critère de Routh permet de prédire si les racines d'un polynôme sont toutes à partie réelle négative, donc stables. Énonçons le théorème de Routh dans le cas d'un polynôme  $P'(w)$  :

$$P'(w) = a_0 + a_1 \cdot w + a_2 \cdot w^2 + \cdots + a_n \cdot w^n \quad (11)$$

Dans un premier temps on construit le tableau :

$$\begin{array}{cccccc|c}
 \beta_{n,0} & \beta_{n,1} & \beta_{n,2} & \beta_{n,q-1} & \beta_{n,q} & 0 & \\
 \beta_{n-1,0} & \beta_{n-1,1} & \beta_{n-1,2} & \beta_{n-1,q-1} & \beta_{n-1,q-1} & 0 & \\
 \beta_{n-2,0} & \beta_{n-2,1} & \beta_{n-2,2} & \beta_{n-2,q-1} & \beta_{n-2,q-1} & & \\
 & & & & & & \\
 \beta_{2,0} & \beta_{2,1} & 0 & & & & \\
 \beta_{1,0} & 0 & & & & & \\
 \beta_{0,0} & 0 & & & & & 
 \end{array} \quad (12)$$

Puis on vérifie que tous les premiers éléments de la première colonne ont le même signe : cela garantit que tous les pôles de  $P'(w)$  ont leur partie réelle négative.

Les éléments du tableau sont construits par récurrence ligne par ligne en appliquant les formules suivantes :

Pour  $i = n$  et  $i = n - 1$  :

$$\begin{cases} \beta_{n,j} = a_{n-2j} \\ \beta_{n-1,j} = a_{n-1-2j} \end{cases} \quad (13)$$

Et pour  $i < n-1$  :

$$\beta_{i,j} = \frac{\beta_{i+1,0} \cdot \beta_{i+2,j+1} - \beta_{i+1,j+1} \cdot \beta_{i+2,0}}{\beta_{i+1,0}} \quad (14)$$

Dans le cas qui nous intéresse, un système échantillonné, on cherche plutôt à vérifier que les pôles du polynôme  $P(z)$  sont de module inférieur à un. L'idée est d'effectuer un changement de variable  $z \rightarrow w$  pour pouvoir appliquer le critère de Routh.

$$P'(w) = (1-w)^n \cdot P\left(\frac{1+w}{1-w}\right) \quad (15)$$

Si l'on effectue la transformation (15) alors le fait que  $P'(w)$  ait toutes ses racines à partie réelle négative garantit que  $P(z)$  a toutes ses racines dans le cercle unité. Pour vérifier la stabilité de  $P(z)$  il suffit donc d'effectuer le changement de variable (15) puis d'appliquer le critère de Routh conventionnel (12 13 14).

### Critère algébrique attestant de la stabilité : critère de Jury

Le critère de Jury s'apparente au critère de Routh. Nous le donnons pour un polynôme de degré 2, 3 et 4 du type :

$$P(z) = a_0 + a_1 \cdot z + a_2 \cdot z^2 + \dots + a_n \cdot z^n \quad (16)$$

Les conditions de stabilité sont données ci-dessous (ces conditions sont données dans le cas  $a_n > 0$ , il suffit de multiplier tous les termes par  $-1$  si cette condition n'est pas remplie par  $P(z)$ ) :

$$\begin{aligned}
 n = 2 & \begin{cases} a_0 + a_1 + a_2 > 0 \\ a_0 - a_1 + a_2 > 0 \\ -a_0 + a_2 > 0 \end{cases} \\
 n = 3 & \begin{cases} a_0 + a_1 + a_2 + a_3 > 0 \\ -a_0 + a_1 - a_2 + a_3 > 0 \\ -|a_0| + a_3 > 0 \\ a_0 a_2 - a_1 a_3 - a_0^2 + a_3^2 > 0 \end{cases} \\
 n = 4 & \begin{cases} a_0 + a_1 + a_2 + a_3 + a_4 > 0 \\ a_0 - a_1 + a_2 - a_3 + a_4 > 0 \\ a_4^2 - a_0^2 - |a_0 a_3 - a_1 a_4| > 0 \\ (a_0 - a_4)^2 (a_0 - a_2 + a_4) + (a_1 - a_3)(a_0 a_3 - a_1 a_4) > 0 \end{cases}
 \end{aligned} \quad (17)$$

### 3.2.8 Systèmes du premier ordre

On donne dans la Figure 4 les réponses indicielles de systèmes du premier ordre ainsi que la localisation des pôles dans le plan complexe. Le cercle de rayon unité représente la zone dans laquelle les pôles doivent se situer pour assurer la stabilité. On remarque sur les deux premières figures l'influence de la place du zéro par rapport au point  $(1,0)$  pour la rapidité.

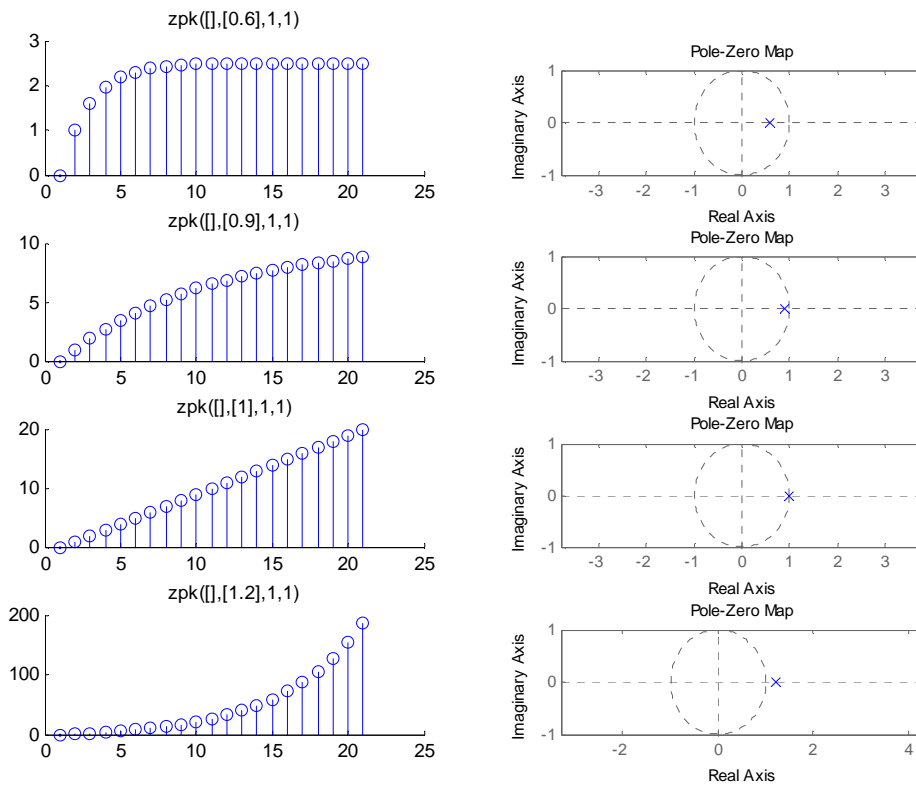


Figure 4 réponses indicielles de systèmes du premier ordre

### 3.2.9 Systèmes du second ordre

On donne dans la Figure 6 les réponses indicielles de systèmes du second ordre ainsi que la localisation des pôles dans le plan complexe. La rapidité et l'amortissement dépend de la position des pôles dans le cercle unité. La Figure 5 représente les courbes iso-amortissement et iso-pulsation des systèmes du second ordre.

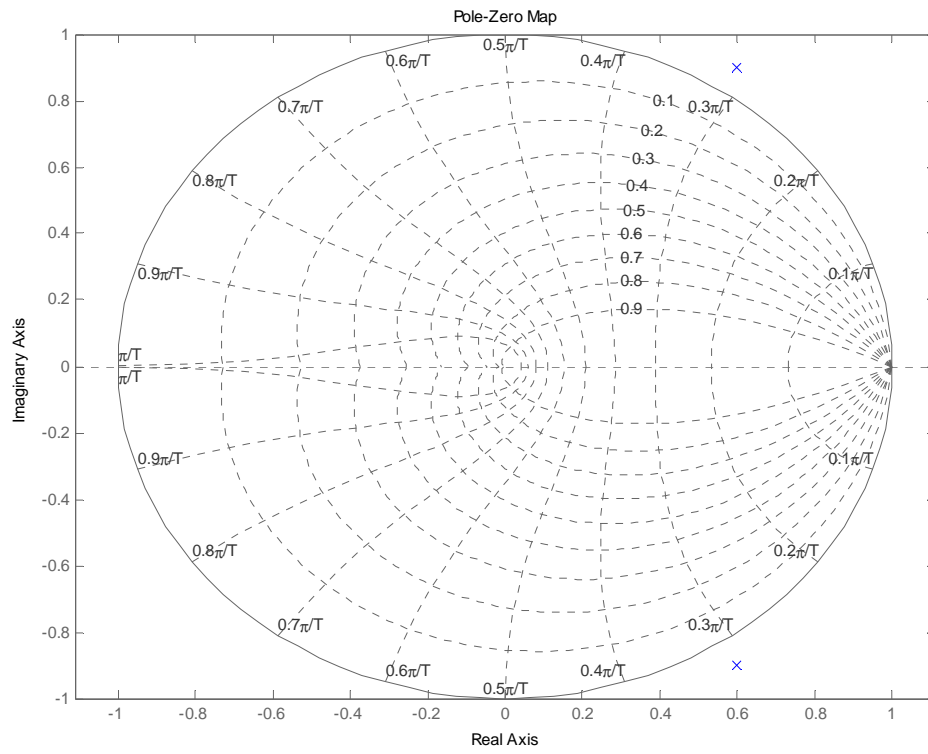


Figure 5 : courbes iso-amortissement et iso-pulsation des systèmes discrets du second ordre

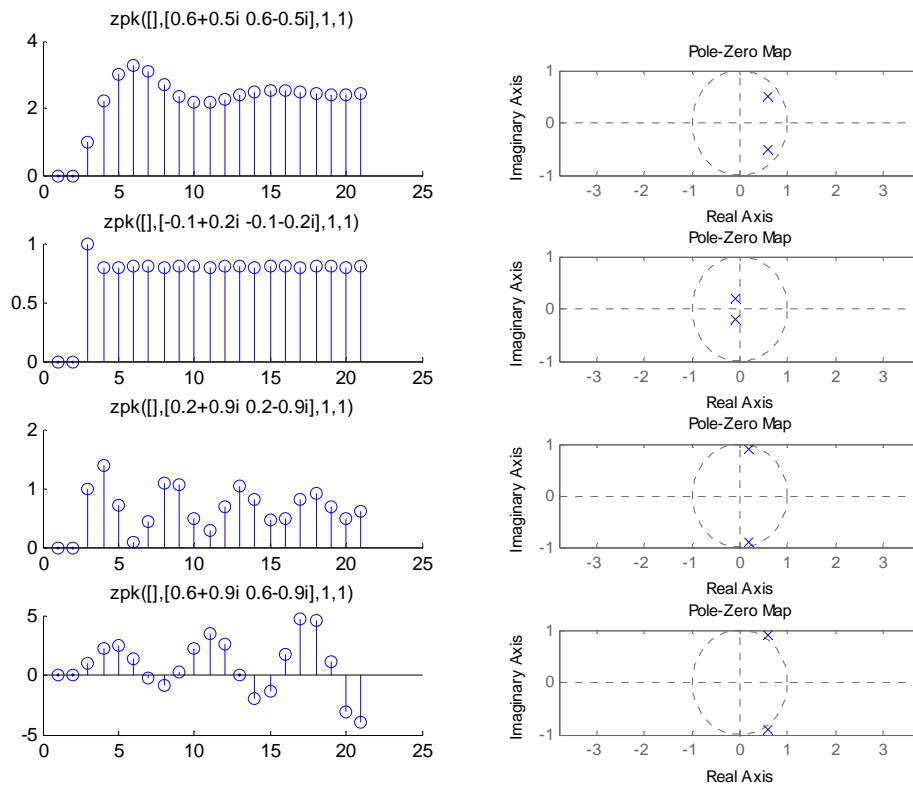


Figure 6 : réponses de systèmes du second ordre

### 3.2.10 Calcul itératif de la réponse d'un système

Si l'on dispose de la fonction de transfert d'un système et que l'on connaît les valeurs successives  $e(k)$  du signal d'entrée il est facile de calculer itérativement la sortie  $y(k)$  du système par l'équation de récurrence (2). Il faut initialiser par les  $n$  valeurs initiales de  $s(k)$ , prises généralement égales à 0.

### 3.2.11 Représentation par équation d'état

Un système discret peut aussi être représenté par un ensemble d'équations d'état comme dans le domaine continu (52). Cette représentation est naturellement adaptée aux systèmes multi-entrées multi-sorties. On définit alors un état interne  $x(k)$  ainsi qu'une fonction d'évolution  $f$  et une équation de mesure  $h$ .

$$\begin{cases} x(k+1) = f(x(k), u(k)) \\ s(k) = h(x(k), u(k)) \end{cases} \quad (18)$$



### 3.3 Réponse fréquentielle des systèmes échantillonnés

#### 3.3.1 Diagramme de Bode d'un système échantillonné

Lorsqu'on étudie la réponse fréquentielle des systèmes continus on considère des entrées particulières sinusoïdales :

$$\begin{cases} e(t) = \cos(2 \cdot \pi \cdot f \cdot t) + j \cdot \sin(2 \cdot \pi \cdot f \cdot t) = e^{j \cdot 2 \cdot \pi \cdot f \cdot t} \\ f \in [0 \quad \infty] \end{cases} \quad (19)$$

De même, pour des systèmes échantillonnés on considèrera des entrées sinusoïdales échantillonnées :

$$\begin{cases} e(k) = \cos(2 \cdot \pi \cdot f \cdot k) + j \cdot \sin(2 \cdot \pi \cdot f \cdot k) = e^{j \cdot 2 \cdot \pi \cdot f \cdot k} \\ f \in [0 \quad 1] \end{cases} \quad (20)$$

La sortie est aussi un signal échantillonné sinusoïdal de gain  $A(f)$  et de phase  $\Phi(f)$  variables et dépendant de  $f$ ; la fréquence réduite.

$$\begin{cases} s(k) = A(f) \cdot e^{j \cdot (2 \cdot \pi \cdot f \cdot k + \Phi(f))} \\ f \in [0 \quad 1] \end{cases} \quad (21)$$

Le nombre complexe, noté abusivement  $F(f)$ , d'amplitude  $A(f)$  et de phase  $\Phi(f)$  est obtenu à partir de  $F(z)$  en remplaçant  $z$  par  $e^{j \cdot 2 \cdot \pi \cdot f}$  :

$$\begin{cases} F(f) = F(z = e^{j \cdot 2 \cdot \pi \cdot f}) \\ f \in [0 \quad 1] \end{cases} \quad (22)$$

Le diagramme représentant le gain et la phase en fonction de la fréquence réduite s'appelle aussi un diagramme de Bode. L'échelle des fréquences est généralement linéaire de 0 à 0.5 (puisque le diagramme est nécessairement symétrique). La Figure 7 représente par exemple le diagramme de Bode d'un système du second ordre.

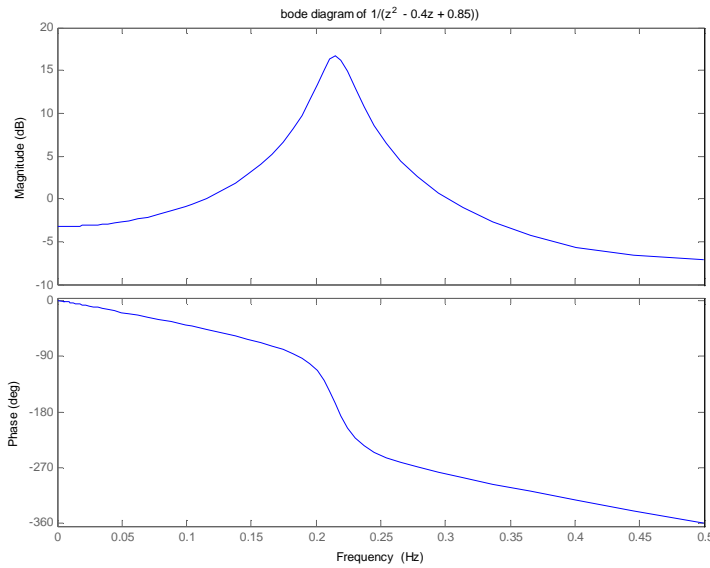


Figure 7 : exemple du diagramme de Bode d'un système discret du second ordre

### 3.3.2 Où l'on voit apparaître le théorème de Shannon

Le changement de variable  $z = e^{j2\pi f}$  n'est pas bijectif puisque  $e^{j2\pi f} = e^{j2\pi(f+k)}$  avec  $k$  entier. Si l'on fait le lien entre domaine continu et domaine fréquentiel on peut remarquer que le changement de variable  $z = e^{j2\pi f}$  fait correspondre une zone du plan complexe continu à une zone du plan complexe discret. La transformation est bijective si l'on se contente de la zone du plan complexe continu contenu entre les droites d'ordonnée 0 et 1.

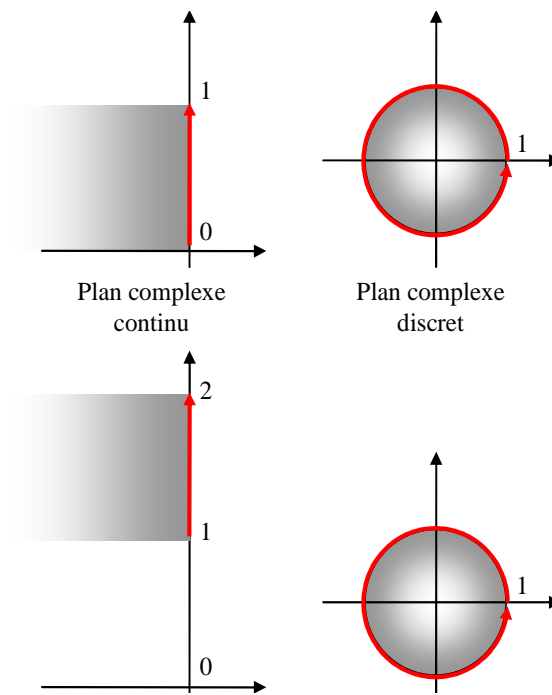


Figure 8 : correspondance entre plan complexe continu et plan complexe discret.



## 4. SYSTEMES CONTINUS ECHANTILLONNES

### 4.1 Structure des systèmes commandés par ordinateur

Les systèmes commandés par ordinateurs sont presque toujours des systèmes physiques continus par nature. Le ordinateur traite par contre les entrées et les sorties de manière discrète avec une période d'échantillonnage  $T_e$ . Les éléments importants, du point de vue de l'automaticien sont donc :

- le convertisseur analogique / numérique CAN ou du moins le système qui fournit séquentiellement les mesures au ordinateur à la période  $T_e$ .
- le ordinateur lui-même. Il doit effectuer l'acquisition des mesures, le traitement des informations, l'envoi des consignes dans un temps inférieur à  $T_e$ .
- le convertisseur numérique / analogique CNA.
- le système physique.

On suppose, pour faire simple et pour pouvoir utiliser la panoplie des outils mathématiques, que les systèmes sont échantillonnés à la même période  $T_e$  et au même moment.

La Figure 9 montre le système physique commandé par le ordinateur, dans sa version simplifiée.

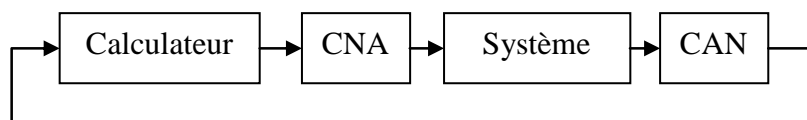


Figure 9 : système commandé par ordinateur

Au XXI<sup>ème</sup> siècle les choses sont quand même parfois un petit peu plus compliquées :

- le ordinateur fournit bien souvent un signal de commande numérique (transporté par bus informatique USB, RS232, CAN, Ethernet, etc...) directement interprété par l'actionneur. On parle "d'actionneur intelligent".
- de même le "capteur intelligent" fournit aussi bien souvent la mesure sous forme d'un signal numérique.
- Les signaux sont transportés par des bus informatiques qui n'ont pas que ça à faire : il peut y avoir du retard.
- de même les capteurs et actionneurs sont rarement tous seuls, ce qui se traduit par des périodes d'échantillonnage différentes, et parfois non synchrones...

- enfin le ordinateur peut aussi avoir d'autres choses à faire...

Ceci étant dit nous allons pour le moment nous restreindre au cas représenté sur la Figure 9 avec des instants d'échantillonnage synchrones et régulièrement espacés d'un temps  $T_e$ .

## 4.2 Echantillonnage des systèmes continus

On dispose généralement de la fonction de transfert continue (notée en utilisant la variable complexe  $s$  ou  $p$ )  $F(s)$  du système continu. Le problème est d'obtenir la fonction de transfert discrète du système échantillonné de manière à pouvoir synthétiser un correcteur. Le système continu est associé au convertisseur numérique – analogique qu'il faut prendre en compte.

Les convertisseurs numérique analogiques peuvent généralement se représenter par des "échantillonneurs – bloqueurs d'ordre 0" ou BOZ. Cela signifie que la commande est maintenue à une valeur constante pendant toute la durée du temps d'échantillonnage.

Si l'on applique une commande discrète impulsionnelle à un BOZ il va maintenir la commande 1 pendant la durée  $T_e$  puis revenir à zéro (Figure 10).

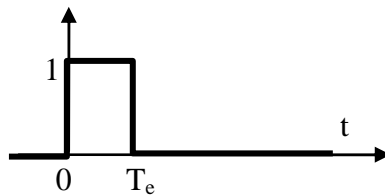


Figure 10 : réponse impulsionnelle d'un bloqueur d'ordre 0

La réponse impulsionnelle  $BOZ(t)$  est donc composée d'un échelon unitaire suivi d'un échelon unitaire de signe opposé et décalé d'un temps  $T_e$ . La transformée de Laplace de la réponse impulsionnelle du bloqueur est donnée par la formule (23) :

$$BOZ(s) = \frac{1}{s} - \frac{e^{-jT_e}}{s} \quad (23)$$

Du point de vue du ordinateur il faut donc prendre en compte le système continu de fonction de transfert  $F_{BOZ}(s)$  que l'on devra échantillonner :

$$F_{BOZ}(s) = \frac{1 - e^{-jT_e}}{s} \cdot F(s) \quad (24)$$

$$F_{\text{BOZ}}(z) = Z^{-1} \left( \frac{1 - e^{-jT_e}}{s} \cdot F(s) \right) \quad (25)$$

En pratique l'obtention de la transformée en z d'un système échantillonné par bloqueur d'ordre 0 se fait à l'aide de tables. On donne quelques exemples fondamentaux dans le Tableau 3 ci-dessous.

Tableau 3 : transformée en z de systèmes continus échantillonnés

F(s)	F <sub>BOZ</sub> (z)
1	1
$\frac{1}{s}$	$\frac{T_e \cdot z^{-1}}{1 - z^{-1}}$
$\frac{1}{1 + T \cdot s}$	$\begin{cases} \frac{b_1 \cdot z^{-1}}{1 + a_1 \cdot z^{-1}} \\ b_1 = 1 - e^{-T_e/T} \quad a_1 = -e^{-T_e/T} \end{cases}$
$\frac{e^{-Ls}}{1 + T \cdot s}$	$\begin{cases} \frac{b_1 \cdot z^{-1} + b_2 \cdot z^{-2}}{1 + a_1 \cdot z^{-1}} \quad a_1 = -e^{-T_e/T} \\ b_1 = 1 - e^{-(L-T_e)/T} \quad b_2 = e^{-T_e/T} (e^{L/T} - 1) \end{cases}$
$L < T_e$	

Si l'on dispose de Matlab c'est bien sûr encore plus simple. On utilise la fonction c2d avec l'argument 'zoh' pour "zero holder hold" (bloqueur d'ordre zéro).

#### Matlab

```
% Définition d'un système continu :
sysc=tf(1,[1 1]);
% Obtention de la transformée en z du système échantillonné à Te
Te=0.1;
sysd=c2d(sysc,Te,'zoh');
% Notation en z^-1
sysd.variable='z'
```

### 4.3 Effets de l'échantillonnage

Le convertisseur numérique a deux effets (néfastes) importants : l'effet de retard et l'effet de quantification.

L'effet de retard se voit dans (24) : on peut dire en première approximation que l'échantillonnage se traduit par un retard pur de  $T_e/2$ . Cela a donc, si ce temps

d'échantillonnage est non négligeable par rapport aux constantes de temps du système, un effet déstabilisant.

Pour illustrer l'effet de quantification supposons par exemple que l'on utilise un convertisseur analogique/numérique 8 bits 0-10V. Le plus petit incrément que le convertisseur peut distinguer est de  $10V/2^8 = 40mV$ . Si le signal utile varie physiquement entre 5V et 5,4V on n'aura que dix incréments utilisables... Ce phénomène n'entre pas dans la panoplie des phénomènes modélisables par la transformée en z. En pratique il est donc souvent nécessaire d'étudier le système et de synthétiser la commande numérique en négligeant ce phénomène puis de vérifier a posteriori qu'il n'a pas un effet trop néfaste.

#### 4.4 Où l'on revoit le théorème de Shannon

Le théorème de Shannon est bien connu en ce qui concerne l'échantillonnage de signaux discrets : on sait que le signal ne doit pas comporter de composante de fréquence supérieure à  $f_e/2$  si l'on ne veut pas observer de repliement. On doit aussi prendre des précautions si l'on veut travailler sur la fonction de transfert discrète correspondant à un système continu échantillonné.

On va traiter un exemple concret pour mettre en évidence le problème de repliement dans le cas de la commande des systèmes continus. On s'intéresse à deux systèmes continus du second ordre ayant des pôles de même partie réelle mais de parties imaginaires séparées de  $2.\pi.T_e$ . Le script Matlab ci-dessous montre la création des deux systèmes :

##### Matlab

```
% Définition de deux systèmes continus :
sys1=zpk([],[-1-2i -1+2i],2^2+1);
sys2=zpk([],[-1-(2+2*pi)*i -1+(2+2*pi)*i],1+(2+2*pi)^2);
```

On voit sur la Figure 11 que la réponse indicielle des deux systèmes est bien différente.



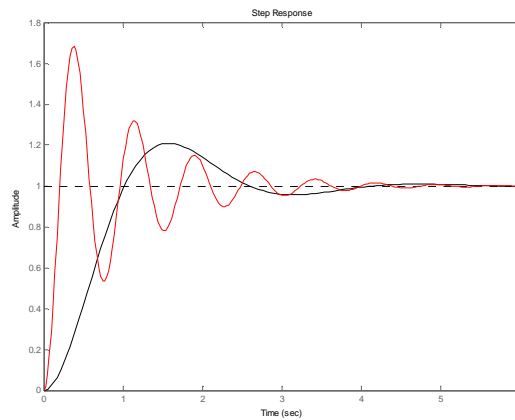


Figure 11 : réponse indicielle de deux systèmes du second ordre

Si l'on calcule la fonction de transfert échantillonnée les deux systèmes à la période de 1 seconde on obtient deux réponses indicielles très proches (représentés par des  $\circ$  sur la Figure 12). Conclusion : la période d'échantillonnage était mal choisie

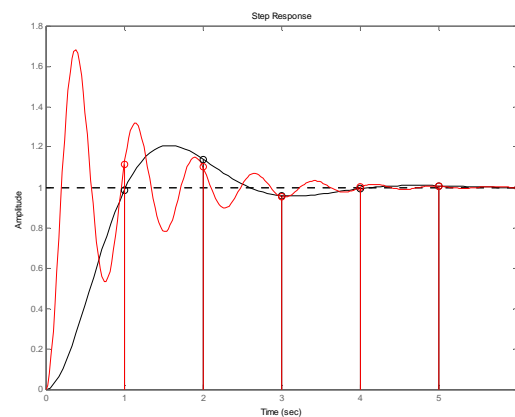


Figure 12 : réponse indicielle de deux systèmes du second ordre échantillonnés à 1s

En pratique la commande échantillonnée de systèmes continus doit être associée à certaines précautions :

- le système ne doit pas comporter de dynamiques propres situées dans les fréquences supérieures à  $f_c/2$  (exemple illustré dans ce chapitre).
- les signaux échantillonnés doivent être "nettoyés" des fréquences supérieures à  $f_c/2$  avec un filtre anti-repliement. (A moins que le signal soit suffisamment propre et ne comporte pas de composantes significatives au-delà de ces fréquences).

### Choix de la période d'échantillonnage

Le choix de la période d'échantillonnage est très important et dépend de la dynamique des signaux. Attention, les systèmes commandés en boucle fermée ont souvent une dynamique plus rapide que le système en boucle ouverte : il faut donc prendre en compte la dynamique en boucle fermée pour le choix de la fréquence d'échantillonnage.

En pratique on préconise de prendre une fréquence 5 à 25 fois supérieure à la bande passante souhaitée en boucle fermée.

On peut être tenté de prendre une fréquence d'échantillonnage très élevée si la technologie le permet (ordinateur et convertisseurs suffisamment rapide). Ceci peut amener rapidement à des problèmes numériques. Imaginons par exemple le cas d'un système ayant une dynamique très lente (quelques secondes) et un système de mesure codant les mesures sur 10 bits (cas classique). La Figure 13 illustre cet exemple. Le signal varie peu entre deux instants d'échantillonnage et les variations du signal sont faibles par rapport à la valeur de l'incrément numérique (figure du haut). Le calcul de la simple dérivée numérique se traduit donc par une succession de  $-1$   $0$  et  $+1$ .

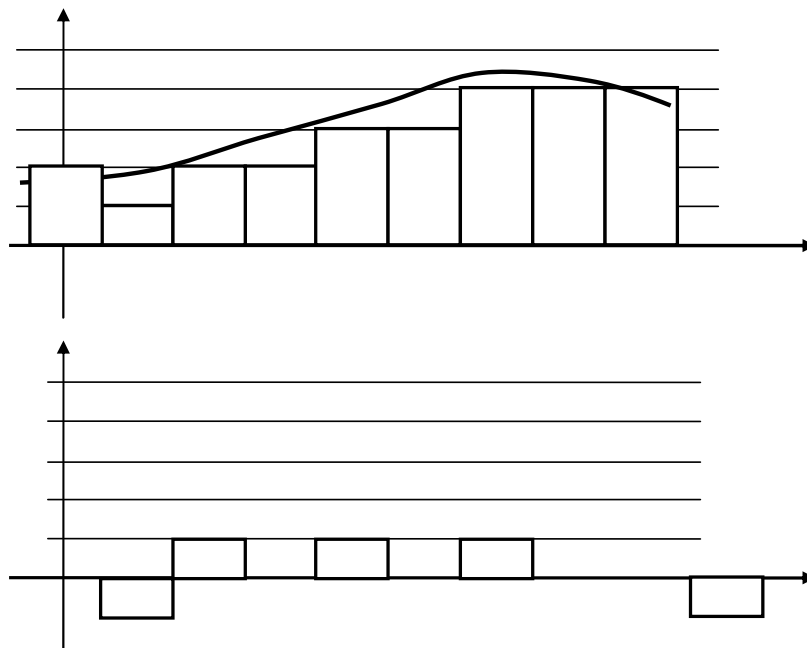


Figure 13 : période d'échantillonnage trop faible. En haut : signal échantillonné et discrétisé ; en bas : calcul de la dérivée numérique.

### Filtre anti-repliement

En toute rigueur chaque fois qu'un signal est échantillonné il faudrait faire précéder le convertisseur numérique d'un filtre analogique anti-repliement éliminant physiquement toutes les composantes du signal de fréquence supérieure à  $f_e/2$ .

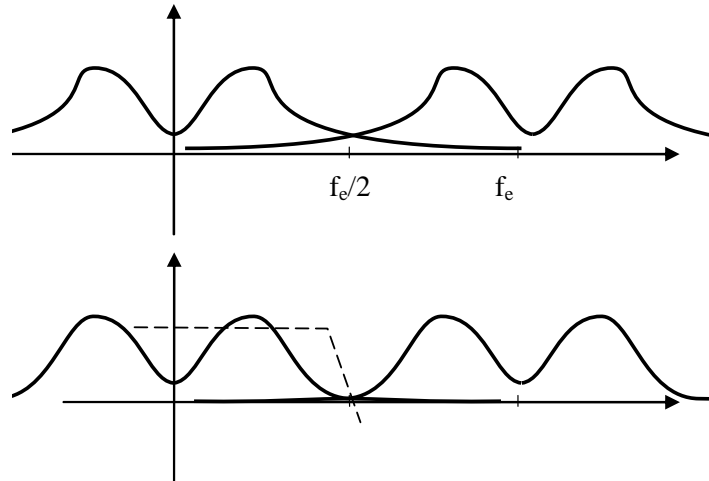


Figure 14 : Effet du filtre anti-repliement

Bien souvent la période d'échantillonnage de l'asservissement est relativement lente (quelques secondes) et il est alors impossible de réaliser un filtre de constante de temps aussi faible. On procède alors à un sur-échantillonnage : le signal est filtré, échantillonné à haute fréquence  $T'_e$  puis filtré numériquement et sous-échantillonné à la période  $T_e$  beaucoup plus grande que  $T'_e$  comme illustré sur la

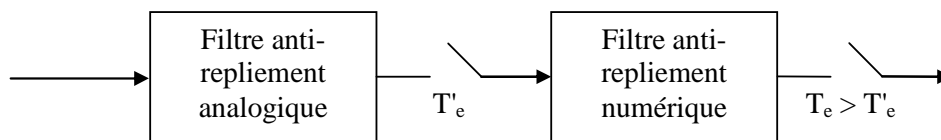


Figure 15 : Sur échantillonnage

Un simple filtre numérique de type moyenne mobile est souvent suffisant.

### 4.5 Discrétisation de fonction de transfert continues

Une manière classique pour réaliser un contrôle par ordinateur est de commencer par synthétiser un correcteur continu avec les outils de l'automatique classique puis de trouver un correcteur discret qui "approche" convenablement le correcteur continu.

#### 4.5.1 Approximation numérique d'équation différentielle

Supposons que le correcteur continu que l'on a synthétisé soit du type PD :

$$\text{PID}(s) = \frac{e(s)}{\varepsilon(s)} = k_p + k_D \cdot s \quad (26)$$

Il s'agit de calculer les commandes  $e(k)$  successives issues du correcteur en fonction des erreurs  $\varepsilon(k)$ . Une approche classique et intuitive est d'approximer la dérivée par une différence :

$$e(k) = k_p \cdot \varepsilon(k) + k_D \cdot \frac{\varepsilon(k) - \varepsilon(k-1)}{T_e} \quad (27)$$

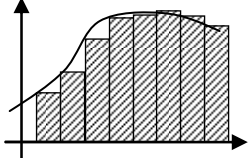
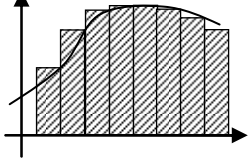
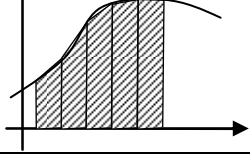
Si on calcule la transformée en  $z$  de  $e(k)$  on obtient :

$$e(z) = k_p \cdot \varepsilon(z) + k_D \cdot \frac{\varepsilon(z) - z^{-1}\varepsilon(z)}{T_e} = \left( k_p + k_D \cdot \frac{1 - z^{-1}}{T_e} \right) \cdot \varepsilon(z) \quad (28)$$

On retrouve la fonction de transfert PD(s) dans laquelle on a effectué le changement de variable  $s = \frac{1-z^{-1}}{T_e}$

De manière plus générale, l'approximation de fonction de transfert revient à effectuer l'approximation numérique d'intégration. Selon la méthode d'intégration choisie on effectue des changements de variable différents.

Tableau 4 : méthodes d'intégration

Méthode	Changement de variable	
Euler progressif (Euler's forward method)	$s = \frac{1 - z^{-1}}{z^{-1} \cdot T_e}$	
Euler rétrograde (Euler's backward method)	$s = \frac{1 - z^{-1}}{T_e}$	
Méthode des trapèzes, ou bilinéaire, ou de Tustin	$s = \frac{2}{T_e} \cdot \frac{1 - z^{-1}}{1 + z^{-1}}$	

En pratique le changement de variable  $s = \frac{1 - z^{-1}}{z^{-1} \cdot T_e}$  n'est pas sans risque. On donne sur la Figure 16 la correspondance entre la zone stable du plan complexe en  $s$  et la zone correspondante après la transformation (Euler progressif, Euler rétrograde, bilinéaire). On voit notamment qu'il est possible, partant d'un système continu stable, d'obtenir des pôles dans le domaine discret en dehors du cercle de stabilité si la transformation appliquée est celle d'Euler progressif.

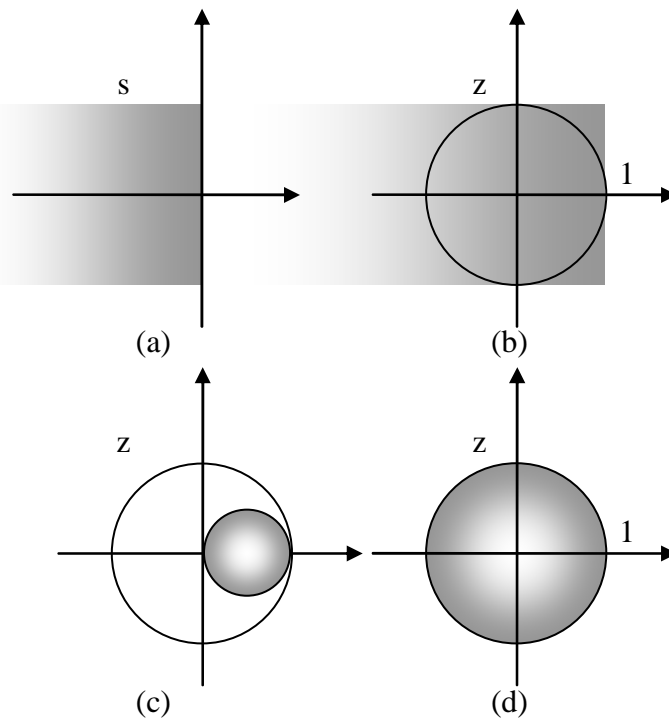


Figure 16 : correspondance des plans complexes. (a) : zone de stabilité dans le plan complexe continu en  $z$  ; (b) zone correspondante avec l'approximation d'Euler progressif ; (c) zone correspondante avec l'approximation d'Euler rétrograde ; (d) zone correspondante avec l'approximation bilinéaire.

La fonction `c2d` de Matlab permet d'obtenir l'équivalent discret d'une fonction de transfert continue avec la méthode d'intégration comme argument :

```

Matlab
sysc=tf([1 0],[1 2 5]);
Te=0.1;
% Méthode d'euler :
sysd1=c2d(sysc,Te,'zoh');
% Méthode des trapèzes :
sysd2=c2d(sysc,Te,'foh');
% Méthode bilinéaire :
sysd3=c2d(sysc,Te,'tustin');
    
```

#### 4.5.2 Superposition des réponses indicielles

Une autre méthode pour approximer une fonction de transfert discrète est de s'assurer que la réponse indicielle de sa version discrétisée correspond exactement lors des temps d'échantillonnage.

### 4.5.3 Correspondances pôles - zéros

La dernière méthode est de partir d'une décomposition de la fonction de transfert continue en pôles et zéros et d'effectuer la conversion pôles à pôles et zéros par zéros. La correspondance, selon que le pôle est réel ou complexe est donnée ci-dessous :

$$\begin{aligned} (s + a) &\rightarrow (z - e^{-a \cdot T_c}) \\ (s + a)^2 + b^2 &\rightarrow z^2 - 2 \cdot e^{-a \cdot T_c} \cdot \cos(b \cdot T_c) \cdot z + e^{-2 \cdot a \cdot T_c} \end{aligned} \quad (29)$$

### 4.5.4 Bilan

Aucune méthode n'est parfaite et on n'aura jamais en même temps :

- même réponse indicielle
- même réponse impulsionnelle
- approximation (Euler, Tustin, etc...) de l'équation différentielle
- correspondance des pôles et des zéros
- correspondance des réponses fréquentielles

Ceci étant dit le choix de la période d'échantillonnage est primordial. Si la fréquence d'échantillonnage est beaucoup plus faible que la dynamique de la fonction de transfert à approximer la différence entre chaque méthode sera minimale.





## 5. IDENTIFICATION DES SYSTEMES ECHANTILLONES

Dans le présent chapitre on donne les bases des méthodes d'identification. Pour une approche complète, voir le cours de troisième année "Identification avion" [6] et les ouvrages [1] et [7].

Même si l'on dispose d'un modèle théorique du processus que l'on souhaite commander il est presque toujours nécessaire de l'affiner en procédant à des essais sur le système réel. Bien souvent le seul modèle dont l'on dispose est ce modèle issu de l'expérimentation. Différentes méthodes s'appliquent selon l'objectif recherché et le niveau de précision exigé. On distinguera :

- les modèles paramétriques
- les modèles simples
- les modèles non paramétriques

### 5.1 Identification d'un modèle ARX par la méthode des moindres carrés récursifs

A partir de cet exemple, le plus simple qui soit, on pose les bases de toutes les méthodes d'identification.

On considère un système d'entrée  $u(t)$  et de sortie  $y(t)$  représenté par l'équation de récurrence suivante :

$$y(t) + a_1 \cdot y(t-1) + a_2 \cdot y(t-2) + \dots + a_n \cdot y(t-n) = b_0 \cdot u(t) + b_1 \cdot u(t-1) + b_2 \cdot u(t-2) + \dots + b_m \cdot u(t-m) \quad (30)$$

L'objectif est, à partir d'essais et de mesures, d'identifier les meilleures valeurs possibles pour les coefficients  $a_i$  et  $b_i$ .

L'équation (30) peut être réécrite pour mettre en évidence la *prédiction* de la sortie  $y(t)$  à partir des mesures passées :

$$y(t) = -a_1 \cdot y(t-1) - a_2 \cdot y(t-2) - \dots - a_n \cdot y(t-n) + b_0 \cdot u(t) + b_1 \cdot u(t-1) + b_2 \cdot u(t-2) + \dots + b_m \cdot u(t-m) \quad (31)$$

Soit :

$$y(t) = \varphi^T(t) \cdot \theta(t)$$

avec :

$$\begin{cases} \varphi^T(t) = [-y(t-1) \ \dots \ -y(t-n) \ , \ u(t-1) \ \dots \ u(t-m)] \\ \theta(t) = [a_1 \ \dots \ a_n \ , \ b_1 \ \dots \ b_m] \end{cases} \quad (32)$$

Le but est de trouver le meilleur ensemble de paramètres  $\theta(t)$  : celui qui, à partir d'un ensemble de mesures  $\varphi(t)$  permet de prédire avec la meilleure précision la sortie effectivement mesurée.

Supposons qu'à la suite d'une expérience on dispose d'un ensemble de mesures  $\{y(t), u(t)\}_{t=1:N}$  et que l'ensemble des paramètres  $\theta$  n'est pas bien connu. On construit la séquence des *sorties prédites* :

$$\hat{y}(t) = \varphi^T(t) \cdot \theta(t) \quad (33)$$

Le paramètre  $\theta$  est celui qui minimise une norme quadratique :

$$\theta = \arg \max \left( \frac{1}{N} \sum_{t=1}^N (y(t) - \hat{y}(t))^2 \right) = \arg \max \left( \frac{1}{N} \sum_{t=1}^N (y(t) - \varphi^T(t) \cdot \theta(t))^2 \right) \quad (34)$$

Compte tenu du fait que l'équation est linéaire au départ et que le terme à minimiser est du type quadratique la solution est simplement obtenue par une pseudo-inverse :

$$\theta = \left[ \sum_{t=1}^N (\varphi(t) \cdot \varphi^T(t)) \right]^{-1} \sum_{t=1}^N (\varphi(t) \cdot y(t)) \quad (35)$$

On obtient donc la solution classique des moindres carrés. La formule (35) est le plus souvent utilisée sous sa forme récursive :

$$\theta(t) = \theta(t-1) + F(t) \cdot \varphi(t) \cdot (y(t) - \theta^T(t-1) \cdot \varphi(t))$$

avec

$$F(t)^{-1} = F(t-1)^{-1} + \varphi(t) \cdot \varphi(t)^T \quad (36)$$

ou

$$F(t) = F(t-1) - \frac{F(t-1) \cdot \varphi(t) \cdot \varphi(t)^T \cdot F(t-1)}{1 + \varphi(t)^T \cdot F(t-1) \cdot \varphi(t)}$$

(On laisse au lecteur le plaisir d'effectuer lui-même les démonstrations, qu'il peut du reste trouver dans [1] et [7]).

## 5.2 L'identification paramétrique en général et en pratique

L'exemple précédent a permis de mettre en évidence tous les aspects de la méthodologie classique de l'identification :

- il a fallu choisir un ensemble de mesures  $\{y(t), u(t)\}_{t=1:N}$ . Le choix du signal d'excitation  $u(t)$  est en pratique primordial,
- il a fallu choisir une structure pour le modèle à identifier (ici les ordres  $n$  et  $m$ ),
- il a fallu choisir un critère (ici un critère quadratique sur l'erreur entre la sortie et la sortie prédite),
- il a fallu choisir un algorithme et le paramétrer (ici l'algorithme des moindres carrés récursifs qu'il a fallu initialiser)

Enfin un point non traité dans le chapitre précédent concerne la validation :

- il aurait fallu déterminer une méthode pour comparer différents modèles et choisir le meilleur

### 5.2.1 Choix du signal d'excitation, traitement des mesures

De manière générale le système que l'on doit identifier doit être excité dans tout le domaine fréquentiel qui nous intéresse. Le signal classiquement utilisé est une séquence pseudo binaire aléatoire (Pseudo Random Binary Signal) tel celui représenté sur la Figure 17.

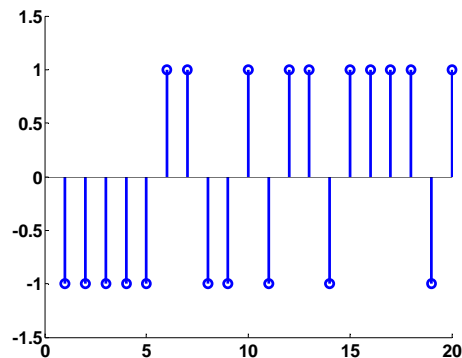


Figure 17 : exemple d'une séquence PRBS

Le signal d'excitation doit se superposer au signal de commande correspondant au point de fonctionnement étudié. Son amplitude doit être faible par rapport à l'amplitude des commandes du système (quelques %).

Une fois le système excité et les mesures effectuées il faut supprimer les valeurs moyennes des mesures.

### 5.2.2 Choix d'une structure de modèle

Le modèle étudié dans le chapitre précédent est de type ARX. On peut le noter sous forme polynomiale :

$$A \cdot y = B \cdot u \quad (37)$$

Ce modèle correspond en fait à un parti-pris sur la manière dont on traite les perturbations. En réalité on prend en compte un signal d'erreurs possibles  $e$  (de moyenne nulle) tels que :

$$A \cdot y = B \cdot u + e \quad (38)$$

Si l'on fait cette hypothèse de modélisation alors la méthode optimale est celle des moindres carrés. Malheureusement cette manière de prendre en compte les perturbations est bien restrictive. D'autres modélisations sont possibles, adaptées à d'autres algorithmes, par exemple le modèle OE :

$$y = \frac{B}{F} \cdot u + e \quad (39)$$

Il est aussi possible de "structurer" le modèle de bruit comme par exemple dans les modèles ARMAX ou BJ :

$$\begin{aligned} A \cdot y &= B \cdot u + C \cdot e \\ y &= \frac{B}{F} \cdot u + \frac{C}{D} \cdot e \end{aligned} \quad (40)$$

La structure du modèle étant choisie il faut choisir l'ordre de ses différents polynômes.

### 5.2.3 Choix d'un critère à minimiser

Dans la plupart des cas le critère à minimiser est l'erreur quadratique entre la sortie et la sortie prédite à un coup (one step ahead prediction). Ceci n'est bien évidemment pas restrictif et peut être généralisé à la prédiction à  $N$  coups.

### 5.2.4 Choix d'un algorithme

Même si chaque structure de modèle est généralement associée à un algorithme "optimal" celui-ci doit généralement se paramétrer.

Prenons par exemple l'algorithme des moindres carrés récursifs tel que présenté dans (36). Cette équation peut être modifiée en intégrant un "facteur d'oubli"  $\lambda$  :

$$\begin{aligned} \theta(t) &= \theta(t-1) + F(t) \cdot \varphi(t) \cdot (y(t) - \theta^T(t-1) \cdot \varphi(t)) \\ \text{avec} & \\ F(t)^{-1} &= \lambda \cdot F(t-1)^{-1} + \varphi(t) \cdot \varphi(t)^T \end{aligned} \quad (41)$$

Ce paramètre (typiquement entre 0.95 et 0.99) permet, comme son nom l'indique, "d'oublier" les anciennes mesures et de se concentrer sur les mesures les plus récentes.

### 5.2.5 Validation

Une fois l'algorithme d'identification appliqué et le modèle obtenu (ce qui prend généralement quelques secondes avec un ordinateur moderne) il reste à expertiser celui-ci. Selon la "qualité" du modèle obtenu on remet en cause l'une ou l'autre des étapes (signal d'excitation, structure du modèle, algorithme) et on réitère jusqu'à obtenir un modèle satisfaisant.

En pratique on essaie un certain nombre de modèles avant de choisir le "meilleur".

Le premier critère de qualité est la capacité du modèle à prédire effectivement la sortie (sortie prédite à un coup, sortie prédite à N coups). Il faut pour cela utiliser un jeu de mesures non utilisé pour l'identification. (En pratique le jeu de mesures initiales est coupé en deux : une partie pour l'identification et une partie pour la validation).

Le second critère de qualité est l'analyse du résidu : la composition spectrale de l'erreur de prédiction.

### 5.2.6 Utilisation de logiciels

En pratique il est commode d'utiliser un célèbre outil de calcul scientifique et sa toolboxe d'identification.

Comme les calculs proprement dit ne prennent que quelques secondes il est facile de réaliser un grand nombre d'identifications et de stocker les différents modèles pour les comparer ensuite. On est rapidement noyé sous le grand nombre de modèles possibles...

## 5.3 Identification "simple"

Il est souvent suffisant d'obtenir un modèle "simple" de type premier ordre et un simple essai indiciel convient souvent. Le système est excité avec un échelon unité et on cherche le meilleur modèle du premier ordre correspondant.

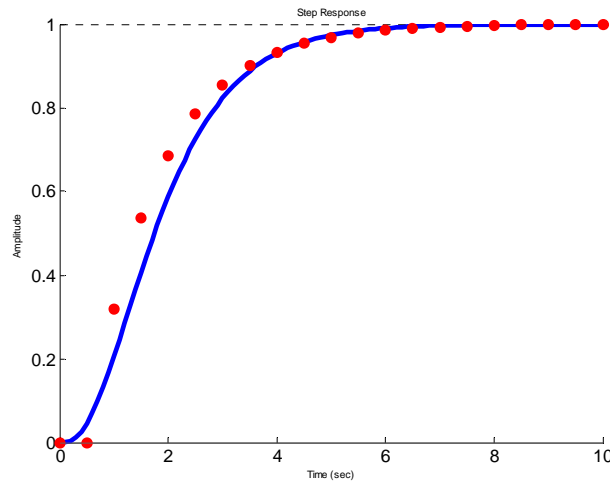


Figure 18 : identification sur un essai indiciel

Dans l'exemple de la Figure 18 on superpose le résultat d'une mesure (en bleu) avec la simulation d'un modèle discret avec une période de 0.5s. Le modèle cherché est du premier ordre avec un retard pur  $d$  :

$$y(z) = \frac{(1 + c_1) \cdot z^{-d}}{1 + c_1 \cdot z^{-1}} \cdot u(z) \quad (42)$$

Il s'agit de trouver l'entier  $d$  et le nombre réel  $c_1$  tels que la réponse indiciale du modèle corresponde pour le mieux aux mesures issues de l'essai. Ceci peut se formaliser par exemple de la manière suivante :

$$(c_1 \quad d_1) = \arg \min \sum_{t=1}^N (y_{\text{modèle}}(t) - y_{\text{mesure}}(t))^2 \quad (43)$$

Cette optimisation peut se faire à la main ou grâce à une procédure automatisée (avec le célèbre outil de calcul scientifique).

Dans le cas où le système comprend naturellement une intégration (par exemple la commande d'un moteur à courant continu par une tension et la mesure de l'angle de rotation) il est utile de prendre en compte explicitement cette intégration dans le modèle :

$$y(z) = \frac{1}{1 - z^{-1}} \cdot \frac{(1 + c_1) \cdot z^{-d}}{1 + c_1 \cdot z^{-1}} \cdot u(z) \quad (44)$$

La procédure est la même mais au lieu de prendre en compte la sortie  $y(t)$  on prend en compte la dérivée  $y'(t)$  :

$$y'(t) = y(t) - y(t-1) \quad (45)$$

## 5.4 Identification non paramétrique

Pour terminer ce panorama rapide sur les méthodes d'identification il faut parler des méthodes non paramétriques : celles qui permettent de tracer la réponse impulsionnelle ou la réponse fréquentielle du système.

### 5.4.1 Estimation de la réponse impulsionnelle

La réponse impulsionnelle d'un système peut être obtenue simplement en appliquant une entrée de type "impulsion" au système et d'en mesurer les conséquences.

La seconde méthode consiste à utiliser la méthode des moindres carrés (ou une méthode dérivée) comme vue au chapitre 5.1. La réponse impulsionnelle d'un système peut en effet être vue comme la fonction de transfert particulière où le dénominateur est de degré nul et le numérateur comporte un nombre infini (en pratique : grand) d'éléments ( $n = 0$  et  $m$  grand) :

$$y(t) = b_0 \cdot u(t) + b_1 \cdot u(t-1) + b_2 \cdot u(t-2) + \dots + b_m \cdot u(t-m) \quad (46)$$

La dernière méthode, qui sera développée ici est l'analyse d'intercorrélation.

Par définition l'intercorrélation  $R_{yu}$  entre deux signaux est :

$$R_{yu}(\tau) = E[(y(t) - E(y(t))) \cdot (u(t-\tau) - E(u(t-\tau)))] \quad (47)$$

Dans cette formule l'espérance mathématique peut être approchée grâce à la loi des grands nombres :

$$R_{yu}(\tau) \approx R_{yu}^N(\tau) = \lim_{N \rightarrow \infty} \frac{1}{N} \sum_{t=1}^N [(y(t) - m_y) \cdot (u(t-\tau) - m_u)] \quad (48)$$

On s'intéresse maintenant à un système tel que décrit par l'équation impulsionnelle (46) et on suppose que le signal d'entrée est un bruit blanc. Le signal d'entrée vérifie donc :

$$\begin{cases} \tau \neq 0 & R_{uu}(\tau) = 0 \\ \tau = 0 & R_{uu}(\tau) = R \end{cases} \quad (49)$$

Il vient alors naturellement :

$$b_k \approx \frac{1}{R} R_{yu}(k) \quad (50)$$

### 5.4.2 Estimation de la réponse fréquentielle

On utilise ici le spectre d'intercorrélation. Par définition le spectre d'intercorrélation entre deux signaux est :

$$\Phi_{yu}(\omega) = \sum_{\tau=-\infty}^{\infty} R_{yu}(\tau) \cdot e^{-i\omega\tau} \quad (51)$$

On suppose que le signal d'entrée  $u(t)$  et le signal de sortie  $y(t)$  sont reliés par une fonction de transfert linéaire du type :

$$y(t) = F \cdot u(t) + v(t) \quad (52)$$

où  $v(t)$  est une perturbation indépendante de  $u(t)$ . On a alors les relations :

$$\Phi_{yu}(\omega) = F(j\omega) \cdot \Phi_{uu}(\omega) \quad (53)$$

Il reste à estimer les spectres d'intercorrélation pour remonter à la réponse fréquentielle  $F(j\omega)$  recherchée. Une idée naturelle est d'utiliser la relation (48) mais cela marche très mal dans presque tous les cas, même en choisissant  $N$  assez grand. L'amélioration est d'utiliser une "fenêtre" pour favoriser les valeurs faibles de  $\tau$  par rapport aux valeurs élevées. La solution classique utilisant la *fenêtre de Hamming* donne :

$$\Phi_{yu}^N(\omega) = \sum_{\tau=-\gamma}^{\gamma} R_{yu}^N(\tau) \cdot w_{\gamma}(\tau) \cdot e^{-i\omega\tau} \quad (54)$$

avec

$$\begin{cases} w_{\gamma}(\tau) = \frac{1}{2} \left( 1 + \cos\left(\frac{\pi \cdot \tau}{\gamma}\right) \right) & |\tau| < \gamma \\ w_{\gamma}(\tau) = 0 & |\tau| > \gamma \end{cases}$$

On utilise ensuite la relation (51) pour estimer la réponse fréquentielle :



$$F(j \cdot \omega) = \frac{\Phi_u^N(\omega)}{\Phi_{yu}^N(\omega)} \quad (55)$$



## 6. SYSTEMES BOUCLES

### 6.1 Fonction de transfert en boucle fermée

Un système et sa commande par ordinateur peut être représenté par un ensemble de fonctions de transfert formant un schéma-bloc plus ou moins compliqué (Figure 19).

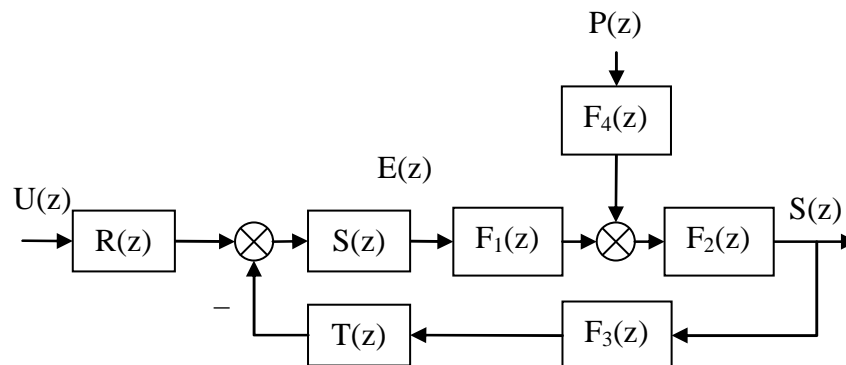


Figure 19 : système discret commandé par ordinateur. Le correcteur est représenté par les fonctions R,S,T, le système par  $F_1$  et  $F_2$ , le modèle de perturbations par  $F_4$  et le modèle des actionneurs par  $F_4$ .

Quelle que soit la structure d'un système asservi il est possible de se ramener à une forme canonique comprenant une fonction de transfert en boucle ouverte en combinant judicieusement les fonctions de transfert du procédé, du régulateur et des perturbations. La Figure 20 représente par exemple le système de la Figure 19 où  $F$  est une combinaison des fonctions de transfert du système, du régulateur, des perturbations et des capteurs.

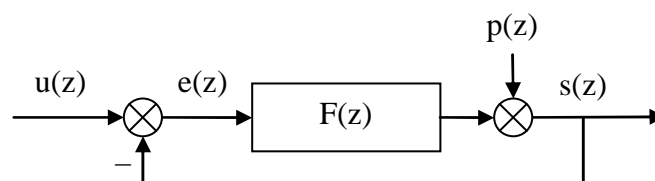


Figure 20 : système discret, notations.

Par défaut on considère une contre-réaction négative et une perturbation additive sur la sortie. La fonction de transfert  $F$  est appelée *fonction de transfert en boucle ouverte*. Comme pour les systèmes continus la *fonction de transfert en boucle fermée* est obtenue simplement à partir de  $F$  (58).

$$F(z) = \frac{s(z)}{e(z)} = \frac{B(z^{-1})}{A(z^{-1})} \quad (56)$$

$$G(z) = \frac{s(z)}{u(z)} = \frac{B(z^{-1})}{A(z^{-1}) + B(z^{-1})}$$

## 6.2 Erreur statique

L'erreur statique du système en boucle fermée, comme pour le système en boucle ouverte, est simplement le rapport des coefficients.

$$G_0 = \frac{\sum_{i=0}^{i=m} b_i}{1 + \sum_{i=0}^{i=m} b_i + \sum_{i=1}^{i=n} a_i} \quad (57)$$

Si l'on souhaite que l'asservissement confère un gain statique unitaire au système en boucle fermée (c'est-à-dire que l'on force la sortie  $s(k)$  à tendre vers la valeur de consigne  $u(k)$ ) il faut que ce rapport de coefficients soit égal à un. On montre facilement que cela est obtenu si la fonction de transfert  $F(z)$  comporte un intégrateur :

$$F(z) = \frac{s(z)}{e(z)} = \frac{1}{1 - z^{-1}} \cdot \frac{B(z)}{A'(z)} \quad (58)$$

## 6.3 Rejet de perturbations

La fonction de transfert entre la perturbation  $p(z)$  et la sortie  $y(z)$  est appelée fonction de sensibilité perturbation-sortie, notée  $S_{yp}$ .

$$S_{yp}(z) = \frac{s(z)}{p(z)} = \frac{A(z^{-1})}{A(z^{-1}) + B(z^{-1})} \quad (59)$$

Si l'on souhaite qu'une perturbation de type échelon soit parfaitement rejetée il faut que  $A(1) = 0$ , on retrouve donc la condition (58) à savoir que le système doit comporter une intégration dans la boucle directe. On peut être en outre être amené à spécifier que les perturbations ne soient pas amplifiées dans une certaine bande de fréquence. On rajoute donc une contrainte du type  $|S_{yp}(e^{j2\pi f})| < S_{\max}$  pour  $f$  appartenant à un certain domaine de  $[0 \ 0.5]$ .

## 6.4 Stabilité des systèmes à simple gain de rétroaction

Les critères de Routh et de Jury (donnés dans le chapitre 3.2.7 p13) ou le calcul numérique des pôles permettent de déterminer la stabilité du système en boucle fermée si l'on connaît sa fonction de transfert.

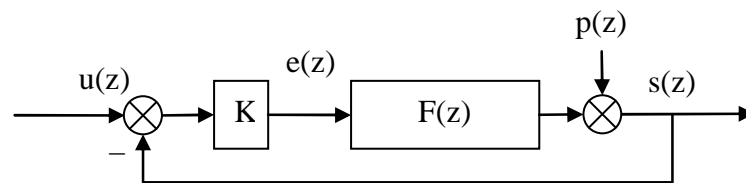


Figure 21 : système bouclé par un simple gain.

On s'intéresse ici au cas particulier où l'asservissement est réalisé par un simple gain de rétroaction  $K$  (Figure 21). On cherche la valeur de  $K$  qui amène le système en limite de stabilité. L'outil adapté ici est le lieu d'Evans ou lieu des racines (root locus) qui permet de visualiser l'évolution des pôles du système bouclé lorsque le gain  $K$  varie.

Prenons l'exemple d'un système comportant trois pôles et deux zéros. L'exemple est donné sous forme de script Matlab :

### Matlab

```
% Définition d'un système discret :
sys=zpk([-2.037 -0.1406],[0.6065 0.3679 0.3679],1,1)
%Tracé du lieu des racines
rlocus(sys);
```

Le script précédent produit le tracé de la Figure 22. On pourrait y lire que la limite de stabilité est obtenue pour  $K=0.2$  environ (détail de la Figure 23).

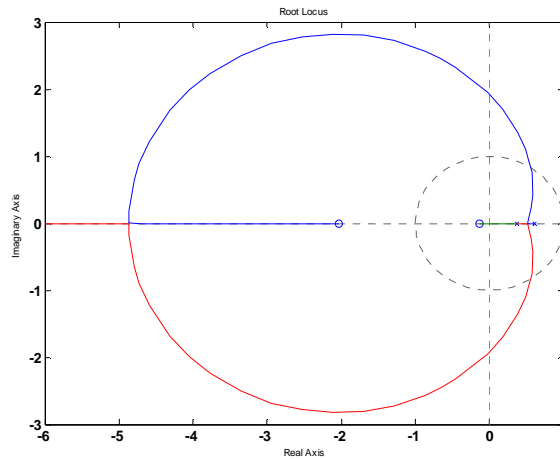


Figure 22 : lieu des racines

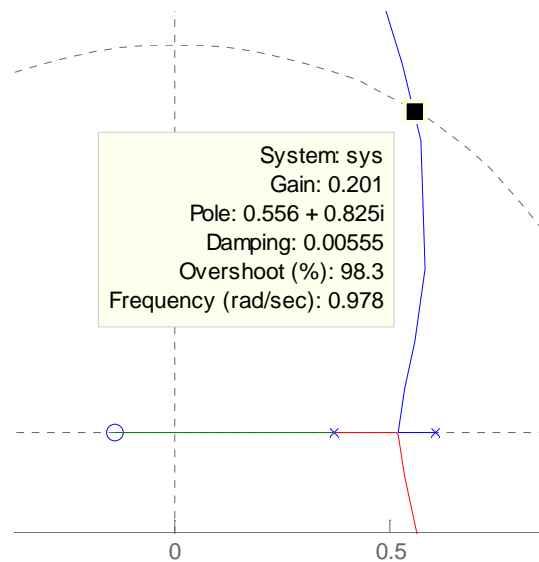


Figure 23 : lieu des racines : détail de la limite de stabilité

On voit sur la Figure 22 que le lieu des racines d'un système échantillonné ressemble à celui d'un système continu :

- on commence par tracer les pôles (×) et les zéros (o) du système en boucle ouverte.
- les pôles du système en boucle fermé décrivent un lieu qui part des pôles (pour  $K = 0$ ) et arrive aux zéros (pour  $K = \infty$ )
- lorsque il y a  $n-m$  plus de pôles que de zéros le système comporte  $n-m$  branches asymptotiques.

On ne décrira pas ici les diverses propriétés du lieu des racines qui permettent de le tracer à la main puisque il est facile à obtenir avec un ordinateur.

Il peut aussi être utilisé pour régler "à la main" le gain permettant d'obtenir un ou plusieurs modes dominants ayant certaines propriétés de fréquence et d'amortissement. On voit sur la Figure 24 que le gain 0.017 permet d'obtenir un pôle double d'amortissement 0.706 et de fréquence 0.65 rad/s et un pôle simple de fréquence 1.63 rad/s.

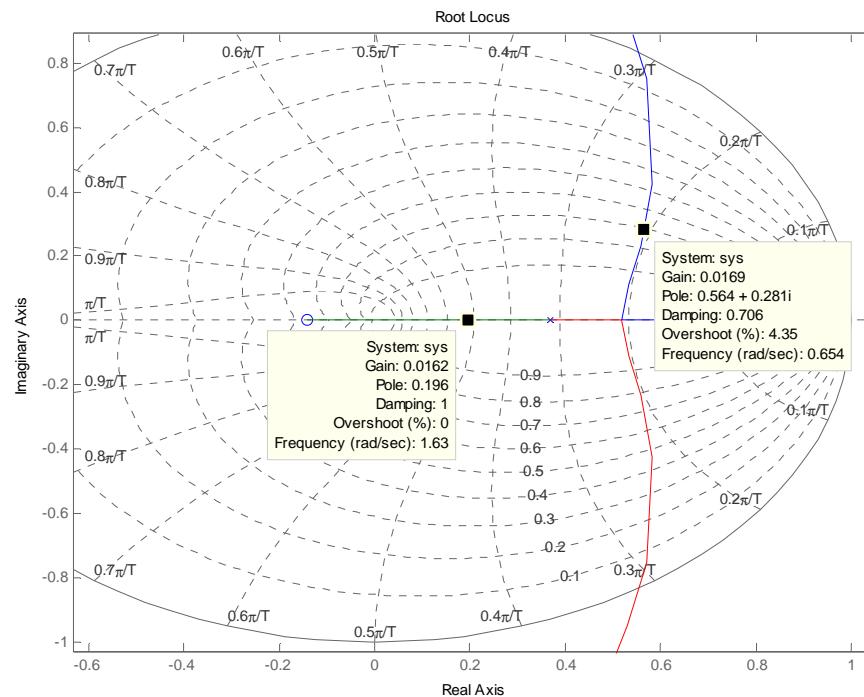


Figure 24 : utilisation du lieu des racines pour régler le gain de la boucle fermée

L'outil rltool de Matlab permet de nombreuses manipulations graphiques de ce type.

## 6.5 Analyse des systèmes bouclés dans le domaine fréquentiel : stabilité

Comme dans le cas continu il est possible de prévoir des propriétés, en particulier la stabilité, d'un système en boucle fermée à partir de l'examen de la fonction de transfert en boucle ouverte.

On ne redémontrera pas ici le théorème de Nyquist mais on se contente de rappeler l'énoncé.

On rappelle que le *lieu de Nyquist* est le lieu de  $F(j2\pi f)$  dans le plan complexe pour  $f$  variant de 0 à  $+\infty$ . A noter que le lieu pour la zone  $[0, 0.5]$  est symétrique de celui pour la zone  $[0.5, 1]$ . Pour cette raison on ne représente généralement que le lieu pour  $f$  appartenant à  $[0, 0.5]$ .

### 6.5.1 Théorème de Nyquist dans le cas où le système est stable en boucle ouverte (critère du revers)

Le système est stable en boucle fermée si le lieu de Nyquist de la fonction de transfert en boucle ouverte parcouru dans le sens des fréquences croissantes de 0 à 0.5 laisse le point critique  $(-1,0)$  à gauche.

La Figure 25 représente par exemple le lieu de Nyquist d'une fonction de transfert en boucle ouverte qui sera stable en boucle fermée.

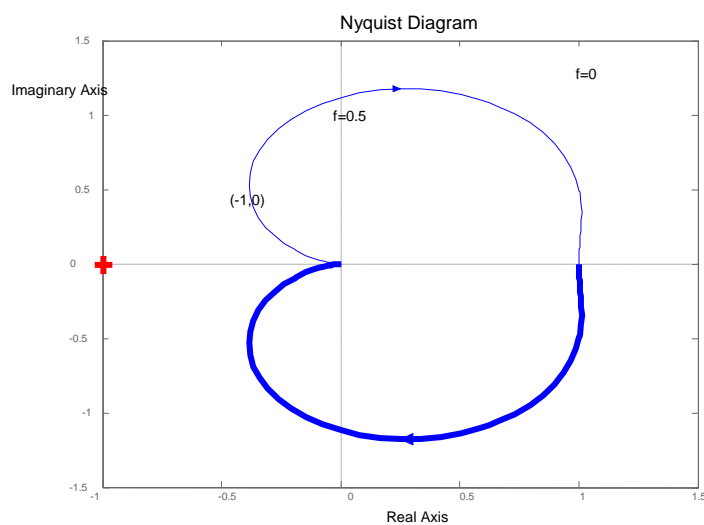


Figure 25 : lieu de Nyquist d'un système stable

### 6.5.2 Théorème de Nyquist dans le cas où le système est instable en boucle ouverte

Si le système en boucle ouverte comporte  $N$  pôles instables il sera stable en boucle fermée si le lieu de Nyquist de la fonction de transfert en boucle ouverte parcouru dans le sens des fréquences croissantes de  $-0.5$  à  $0.5$  entoure le point critique  $(-1,0)$   $N$  fois dans le sens des aiguilles d'une montre..

### 6.5.3 Interprétations du lieu de Nyquist

Comme dans le cas continu, la distance à laquelle on passe du point critique renseigne sur la robustesse du système bouclé. La distance sera exprimée de différente manière (Figure 26) :

- marge de module  $\Delta M$  : plus petite distance entre le lieu des racines et le point critique,



- marge de gain  $\Delta G$
- marge de phase  $\Delta P$

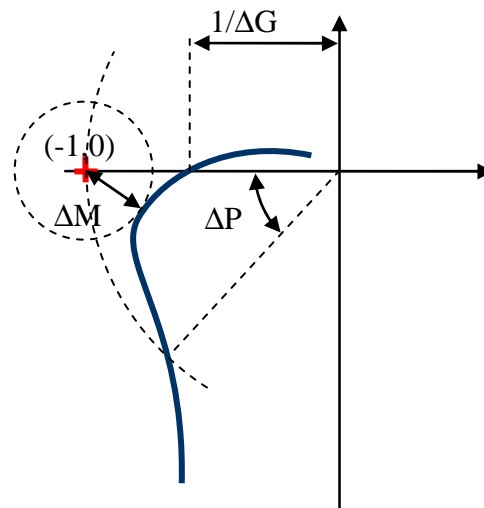


Figure 26 : Marges de stabilité

Les valeurs typiques de ces marges sont données ci-après :

- $\Delta G \geq 6\text{dB}$
- $\Delta M \geq -6\text{dB}$
- $30^\circ \leq \Delta P \leq 60^\circ$

La distance entre le point critique et le lieu des racines est la norme de  $1+F(j2\pi f)$ . On peut donc exprimer la marge de module en fonction de la fonction de sensibilité :

$$\begin{aligned} \Delta M &= \min |1 + F(j \cdot \omega)|_{\omega \in [0, \pi]} \\ \Delta M &= \min \left| 1 + \frac{B(j \cdot \omega)}{A(j \cdot \omega)} \right|_{\omega \in [0, \pi]} = \max \left| \frac{A(j \cdot \omega)}{A(j \cdot \omega) + B(j \cdot \omega)} \right|_{\omega \in [0, \pi]} \quad (60) \\ \Delta M &= \max |S_{yp}(j \cdot \omega)|_{\omega \in [0, \pi]}^{-1} \\ \Delta M_{\text{dB}} &= -\max |S_{yp}(j \cdot \omega)|_{\text{dB}} \end{aligned}$$

Si l'on trace l'évolution de  $S_{yp}$  en fonction de  $f$  on obtient donc la valeur de la marge de module.



# 7. METHODES DE REGULATION DES SYSTEMES ECHANTILLONES

## 7.1 Introduction

Les bases théoriques permettant de concevoir le correcteur pour un système échantillonné ont été posées dans les chapitres précédents. Ce chapitre fait le bilan des connaissances acquises et donne quelques solutions pour calculer des correcteurs à partir d'un cahier des charges précis.

Les systèmes que l'on cherche à commander sont généralement, par nature, continus. La réalisation d'un correcteur par un ordinateur implique que l'on génère une équation de commande discrète (généralement une équation récurrente). L'obtention de cette équation de commande peut se faire de deux manières :

- la première méthode consiste à d'abord synthétiser un correcteur continu (en utilisant les outils classiques de l'automatique et la simulation) et discrétiser ensuite ce correcteur. C'est l'objet du chapitre 7.2.2.
- la seconde méthode consiste à tenir la représentation discrète du système continu associé à son interface numérique (convertisseurs analogique-numérique et numérique-analogique) comme expliqué dans le chapitre 4.5 puis synthétiser directement le correcteur discret. Cela sera expliqué dans le chapitre 7.3 et suivants.

Ce choix peut être résumé par la figure suivante :

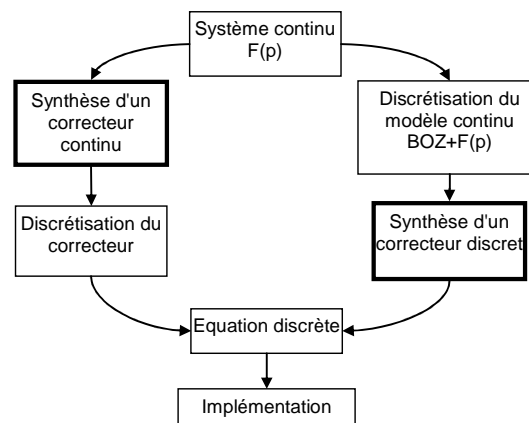


Figure 27 : Réalisation d'un correcteur discret.

Au-delà des affaires de goût, le choix de la méthode sera souvent dicté par l'application. En effet, la première méthode est d'autant meilleure que le correcteur discrétisé approche correctement le correcteur continu. Cela implique que le temps d'échantillonnage soit très faible par rapport aux constantes de temps du système

bouclé. Dans le cas où les contraintes temps réel sont fortes (capacités de calcul partagées par exemple) il sera préférable de synthétiser un correcteur discret.

## 7.2 Structure d'un système commandé par ordinateur, hypothèses

Par la suite on se ramènera toujours à un système tel que celui représenté sur la Figure 28 en cherchant un correcteur  $K(z)$  linéaire. On montre que quel que soit le correcteur linéaire on peut se ramener à la forme représenté sur la Figure 29 (forme RST).

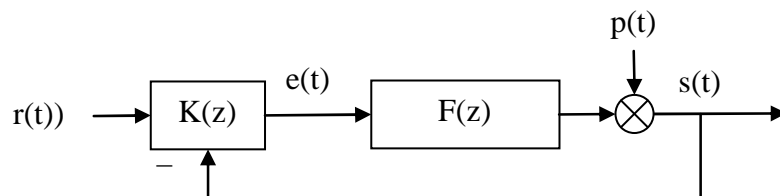


Figure 28 : système bouclé, forme générale.

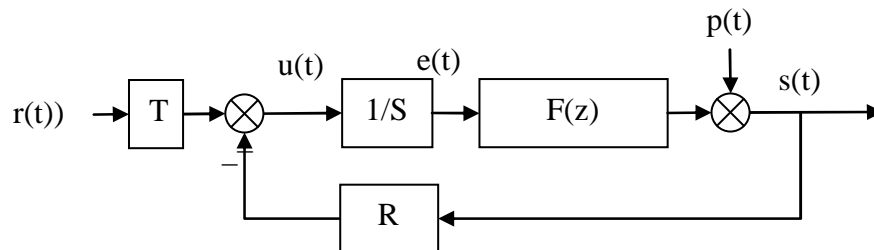


Figure 29 : système bouclé, forme canonique.

### 7.2.1 Discrétisation, choix de la période d'échantillonnage

Du point de vue du ordinateur, l'extérieur (le procédé) est représenté par la fonction de transfert  $F(z)$ . L'hypothèse importante est que la commande (sortie du ordinateur) et la mesure (entrée du ordinateur) sont échantillonnées de manière synchrone à la même période  $T_e$ .

Le choix judicieux de la fréquence d'échantillonnage est très important. En pratique on choisit cette fréquence en fonction de la bande passante du système en **boucle fermée** :

$$f_e = (6 \text{ à } 25) f^{BF} \quad (61)$$

### 7.2.2 Discrétisation d'un correcteur continu

La première méthode pour obtenir les équations d'un correcteur numérique est de faire sa synthèse avec les outils de l'automatique continue puis de discrétiser la fonction de transfert du correcteur continu obtenu (voir pour cela le chapitre 4.2 *Echantillonnage des systèmes continus*).

On va voir par la suite que la synthèse directe d'un correcteur discret offre plus de degré de liberté et permet en général de remplir un cahier des charges plus complet.

## 7.3 Régulateur PID numérique

Le régulateur le plus populaire est le régulateur PID. Nous allons voir dans un premier temps comment on génère un régulateur PID à partir des équations bien connues du régulateur PID continu et comment ces équations peuvent être généralisées pour obtenir plus de degrés de liberté pour le réglage.

### 7.3.1 Cas 1

Un régulateur PID continu a une fonction de transfert du type :

$$K_{\text{PID}}(s) = K \cdot \left( 1 + \frac{1}{T_i \cdot s} + \frac{T_d \cdot s}{1 + \frac{T_d}{N} \cdot s} \right) \quad (62)$$

La discrétisation de ce correcteur continu par la méthode d'Euler (Tableau 4 page 31) se fait en remplaçant la variable de Laplace  $s$  par le terme  $\frac{1-q^{-1}}{q^{-1} \cdot T_e}$ . On obtient :

$$K_{\text{PID}}(q) = K \cdot \left( 1 + \frac{1}{T_i \cdot \frac{1-q^{-1}}{T_e}} + \frac{T_d \cdot \frac{1-q^{-1}}{T_e}}{1 + \frac{T_d}{N} \cdot \frac{1-q^{-1}}{T_e}} \right) \quad (63)$$

En réarrangeant on obtient la forme générale RST (Figure 30):

$$K_{PID}(q) = K \cdot \frac{r_0 + r_1 \cdot q^{-1} + r_2 \cdot q^{-2}}{(1 - q^{-1}) \cdot (1 + s_1' \cdot q^{-1})}$$

$$\begin{cases} r_0 = K \cdot \left( 1 + \frac{T_e}{T_i} + \frac{N \cdot T_d}{T_d + N \cdot T_e} \right) \\ r_1 = -K \cdot \left( 1 + \frac{T_d}{T_d + N \cdot T_e} + \frac{T_e}{T_i} \cdot \frac{T_d}{T_d + N \cdot T_e} + \frac{N \cdot T_d}{T_d + N \cdot T_e} \right) \\ r_2 = K \cdot \left( \frac{T_d}{T_d + N \cdot T_e} \right) \\ s_1' = \frac{T_d}{T_d + N \cdot T_e} \end{cases} \quad (64)$$

Comme on le voit le correcteur obtenu comporte autant de degrés de liberté que le correcteur PID analogique. On reconnaît au dénominateur le terme  $(1 - q^{-1})$  en facteur qui assure l'effet d'intégration.

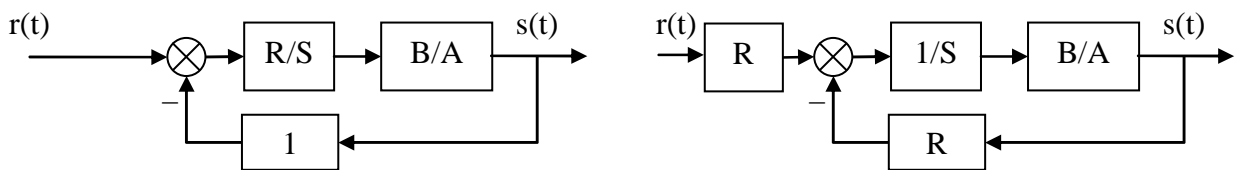


Figure 30 : PID sous forme RST

La fonction de transfert en boucle fermée obtenue est donnée par l'équation suivante :

$$H_{BF} = \frac{B \cdot R}{A \cdot S + B \cdot R} = \frac{B \cdot R}{P} \quad (65)$$

De manière générale le polynôme caractéristique  $P$  fixe les performances visées en boucle fermée. Le calcul de  $R$  et  $S$ ,  $P$  étant fixé et  $A$  et  $B$  étant connus, est réalisé en résolvant l'équation de Bézout  $P = A \cdot S + B \cdot R$ . (voir 9.1 Annexe 1 : Résolution de l'équation polynomiale de Bezout)

On voit que le numérateur  $B \cdot R$  est imposé une fois que  $R$  et  $S$  sont trouvés. Le régulateur introduit donc naturellement des zéros dans la boucle fermée.

Traisons l'exemple entièrement dans le cas où le système est modélisé comme un système du deuxième ordre avec faible retard. La fonction de transfert du procédé est donnée par :

$$H_{BO} = \frac{B}{A} = \frac{b_1 \cdot q^{-1} + b_2 \cdot q^{-2}}{1 + a_1 \cdot q^{-1} + a_2 \cdot q^{-2}} \quad (66)$$

Le correcteur PID rajoutant deux pôles, le polynôme caractéristique en boucle fermée spécifié doit posséder quatre pôles. Le polynôme caractéristique en boucle fermée doit donc être de la forme :

$$\begin{aligned} P &= 1 + p_1 \cdot q^{-1} + p_2 \cdot q^{-2} + p_2 \cdot q^{-3} + p_2 \cdot q^{-4} \\ P &= (1 + p_1' \cdot q^{-1} + p_2' \cdot q^{-2}) \cdot (1 + \alpha_1 \cdot q^{-1}) \cdot (1 + \alpha_2 \cdot q^{-1}) \end{aligned} \quad (67)$$

Et il doit être identifié au produit  $AS + BR$  avec  $R$  et  $S$  ayant les formes données par (64) :

$$\begin{aligned} &(1 + p_1' \cdot q^{-1} + p_2' \cdot q^{-2}) \cdot (1 + \alpha_1 \cdot q^{-1}) \cdot (1 + \alpha_2 \cdot q^{-1}) \\ &= (1 + a_1 \cdot q^{-1} + a_2 \cdot q^{-2}) \cdot (1 - q^{-1}) \cdot (1 + s_1' \cdot q^{-1}) \\ &+ (b_1 \cdot q^{-1} + b_2 \cdot q^{-2}) \cdot (r_0 + r_1 \cdot q^{-1} + r_2 \cdot q^{-2}) \end{aligned} \quad (68)$$

Les coefficients  $p_1'$  et  $p_2'$  définissent un modèle du second ordre aux caractéristiques (pulsation propre  $\omega_0$  et amortissement  $\sigma$  choisis et coefficients déduits conformément au Tableau 3 page 25). Les coefficients  $\alpha_1$  et  $\alpha_2$  correspondent à des pôles auxiliaires aperiodiques. Les choix typiques pour ces coefficients sont :

$$-0.05 < \alpha_1, \alpha_2 < -0.5 \quad (69)$$

On montre que la résolution de (68) est réalisée par une inversion matricielle simple :

$$M \cdot x = p \quad (70)$$

avec

$$\begin{cases} x^T = [1 & s_1' & r_0 & r_1 & r_2] \\ p^T = [1 & p_1 & p_2 & p_3 & p_4] \end{cases}$$

et

$$M = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ a_1' & 1 & b_1 & 0 & 0 \\ a_2' & a_1' & b_2 & b_1 & 0 \\ a_3' & a_2' & 0 & b_2 & b_1 \\ 0 & a_3' & 0 & 0 & b_2 \end{pmatrix} \quad (71)$$

On montre que M est inversible à la condition que A et B soient premiers entre eux, c'est-à-dire que A et B n'aient pas de pôles et de zéros en commun.

(Le problème peut aussi être résolu en utilisant la résolution de l'équation de Bézout).

### 7.3.2 Cas 2

Comme on l'a vu le correcteur PI numérique directement transposé du correcteur PID analogique décrit dans le chapitre 7.3.1 a l'inconvénient de ramener des zéros au dénominateur de la fonction de transfert en boucle fermée (équation 65). Il est possible, grâce à la flexibilité du régulateur numérique d'imposer au correcteur de ne pas ajouter de zéros en boucle fermée. La fonction de transfert désirée en boucle fermée sera donc de la forme :

$$H_{BF} = \frac{P(1)}{B(1)} \cdot \frac{B}{P} \quad (72)$$

Le facteur  $P(1)/B(1)$  garantit le gain statique unitaire du système en boucle fermée (voir page 12). Si le régulateur a la forme générale RST alors les polynômes R S et T doivent vérifier :

$$H_{BF} = \frac{B \cdot T}{A \cdot S + B \cdot R} = \frac{P(1) B}{B(1) P} \quad (73)$$

S et R sont obtenus par la résolution de la même équation que précédemment (68). Le polynôme T est obtenu par simple identification :  $T = R(1)$ .

La structure, sous forme de schéma-bloc est donnée ci-dessous :

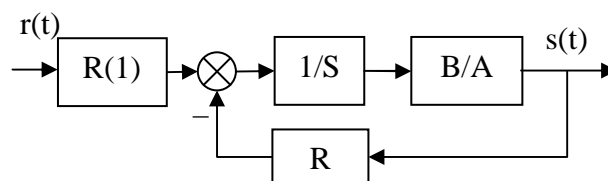


Figure 31 : PID2 sous forme RST

On montre que cette structure de PID, ramené en continu, correspond au PID canonique pour l'action intégrale. L'action proportionnelle et dérivée n'est appliquée que sur le signal y.

La structure du PID2 par rapport au PID1 a pour avantage de ne pas ramener de zéros à la boucle fermée. Cela se traduit par un dépassement moindre en transitoire lorsqu'on applique un échelon de commande. Par contre les deux systèmes réagissent de la même manière à un échelon de perturbation.



## 7.4 Placement de pôles

Les deux structures de PID décrites dans le chapitre précédent sont deux cas particulier d'une stratégie plus générale de placement de pôles.

La synthèse est plus générale que les PID présentés précédemment puisqu'il n'est pas nécessaire de faire d'hypothèses sur le degré de A et B, sur le retard, ni sur la présence ou non de zéros stables ou instables dans le procédé.

La structure du correcteur est la structure type RST telle que présentée dans la Figure 29. Pour la suite des calculs on posera pour la fonction de transfert les conventions suivantes :

$$H_{BO} = \frac{q^{-d} \cdot B(q^{-1})}{A(q^{-1})} \quad (74)$$

$$\begin{cases} A(q^{-1}) = 1 + a_1 \cdot q^{-1} + \dots + a_{n_A} \cdot q^{-n_A} \\ B(q^{-1}) = b_1 \cdot q^{-1} + \dots + b_{n_B} \cdot q^{-n_B} = q^{-1} \cdot B^*(q^{-1}) \end{cases}$$

Compte tenu de la forme RST du régulateur la fonction de transfert en boucle fermée vérifie :

$$H_{BF} = \frac{q^{-d} \cdot B(q^{-1}) \cdot T(q^{-1})}{P(q^{-1})} \quad (75)$$

$$P(q^{-1}) = A(q^{-1}) \cdot S(q^{-1}) + q^{-d} \cdot B(q^{-1}) \cdot R(q^{-1}) = 1 + p_1 \cdot q^{-1} + \dots + p_{n_P} \cdot q^{-n_P}$$

Le polynôme  $P(q^{-1})$  définit donc les pôles de la fonction de transfert en boucle fermée et traduit donc le comportement en *poursuite*.

De la même manière la fonction de transfert entre une perturbation et la sortie (fonction de sensibilité perturbation-sortie) est donnée par :

$$S_{yp} = \frac{A(q^{-1}) \cdot S(q^{-1})}{P(q^{-1})} \quad (76)$$

Ce même polynôme traduit donc aussi le comportement en *régulation*.

Les consignes en poursuite et en régulation peuvent être différentes comme cela est représenté par exemple sur la Figure 32.

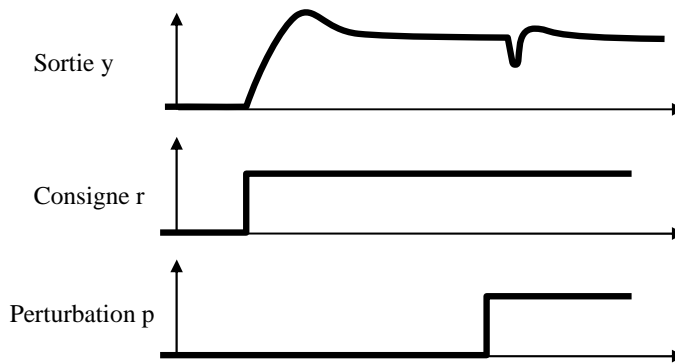


Figure 32 : Performances différentes en régulation et en poursuite

### 7.4.1 Choix des pôles en boucle fermée

De manière générale le comportement en boucle fermée est spécifié sous la forme d'une pulsation propre  $\omega_0$  et d'un amortissement  $\sigma$ , donc une fonction de transfert du second ordre. La spécification étant généralement donnée en continu il est nécessaire de traduire la valeur désirée de  $\omega_0$  et  $\sigma$  en un polynôme  $P_D(q^{-1})$  par discrétisation de la fonction de transfert continue (chapitre 4.5 page 29). Ce polynôme  $P_D$  traduit le comportement dominant du système bouclé.

Si l'on souhaite introduire un effet de filtrage supplémentaire dans certaines bandes de fréquence (réduction des bruits de mesure, adoucissement de la commande, robustesse) cela est décrit par un polynôme  $P_F$  comportant des pôles auxiliaires, plus rapides que les pôles de  $P_D$ .

Le polynôme caractéristique en boucle fermée est donc constitué de pôles dominants et de pôles auxiliaires :

$$P(q^{-1}) = P_D(q^{-1}) \cdot P_F(q^{-1}) \quad (77)$$

### 7.4.2 Régulation : calcul de R et S

Le polynôme  $P(q^{-1})$  étant fixé il s'agit de trouver R et S vérifiant :

$$P(q^{-1}) = A(q^{-1}) \cdot S(q^{-1}) + q^{-d} \cdot B(q^{-1}) \cdot R(q^{-1}) \quad (78)$$

Comme déjà vu au chapitre 7.3.1 on obtient une équation de Bézout qu'il s'agit de résoudre. (Voir aussi 9.1 Annexe 1 : Résolution de l'équation polynomiale de Bezout)

Il est parfois nécessaire de spécifier R et S avec des parties fixes  $H_R$  et  $H_S$ . Dans ce cas l'équation (78) est modifiée et on l'écrit sous la forme:

$$P(q^{-1}) = (A(q^{-1}) \cdot H_S(q^{-1})) \cdot S'(q^{-1}) + (q^{-d} \cdot B(q^{-1}) \cdot H_R(q^{-1})) \cdot R'(q^{-1}) \quad (79)$$

Les inconnues que l'on souhaite trouver sont  $S'$  et  $R'$ . Ceci est nécessaire dans les cas particuliers suivants :

- spécification d'erreur statique nulle,
- rejet d'une perturbation harmonique,
- blocage d'un signal,

Détaillons maintenant ces cas particuliers.

### Spécification d'une erreur statique nulle

On retrouve la nécessité d'avoir un terme intégrateur dans la boucle directe. La spécification se retrouve donc sur  $S$  :

$$S(q^{-1}) = (1 - q^{-1}) \cdot S'(q^{-1}) = H_S(q^{-1}) \cdot S'(q^{-1}) \quad (80)$$

### Rejet d'une perturbation harmonique

Si le système est perturbé par une perturbation harmonique à une fréquence fixe (par exemple 50Hz...) il est possible d'imposer à  $S$  d'avoir un module *nul* ou amorti à cette fréquence. Cela se traduit par la présence dans  $S$  d'une paire de zéros complexe à amortissement nul ou faible à cette fréquence :

$$S(q^{-1}) = (1 + \alpha \cdot q^{-1} + \beta \cdot q^{-2}) \cdot S'(q^{-1}) = H_S(q^{-1}) \cdot S'(q^{-1}) \quad (81)$$

### Blocage d'un signal

Pour certaines application on souhaite que le régulateur "laisse passer" certaines composantes du signal de sortie, c'est-à-dire qu'il ne réagisse pas à ces composantes. Cela se traduit par  $R=0$  à ces fréquences. On retrouve la même forme que précédemment :

$$R(q^{-1}) = (1 + \alpha \cdot q^{-1} + \beta \cdot q^{-2}) \cdot R'(q^{-1}) = H_R(q^{-1}) \cdot R'(q^{-1}) \quad (82)$$

### 7.4.3 Poursuite : calcul de T

Le schéma-bloc de l'asservissement est donné sur la Figure 33. Le terme  $T$  est décomposé en un terme  $T'$  et un terme  $B^*/A^*$ .

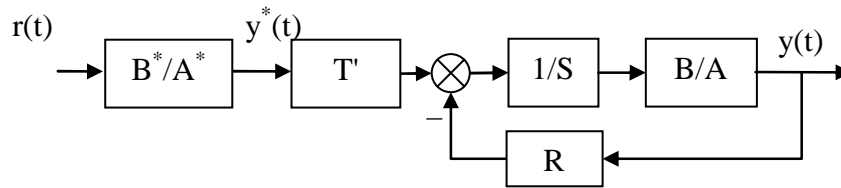


Figure 33 : placement de pôle

L'idée est de régler  $B^*$  et  $A^*$  de telle manière que le signal  $y^*$  ait la dynamique de poursuite désirée (par exemple un second ordre avec pulsation propre et amortissement donné) puis de régler  $T'$  de telle manière que le transfert entre  $y^*$  et  $y$  soit le plus "transparent" possible.

### Réglage de $T'$

Le transfert entre  $y^*$  et  $y$  est :

$$S_{y^*y}(q^{-1}) = T'(q^{-1}) \frac{q^{-d} \cdot B(q^{-1})}{P(q^{-1})} \quad (83)$$

Pour assurer le transfert le plus "transparent" possible il faut donc choisir

$$T'(q^{-1}) = \frac{P(q^{-1})}{B(1)} \quad (84)$$

Remarques :

- Si  $B(1)=0$  alors on remplace  $B(1)$  par 1 dans (84)
- De même dans (84)  $P(q^{-1})$  peut être remplacé par  $P_D$  correspondant à la dynamique dominante.

### Réglage de $B^*$ et $A^*$

$B^*$  et  $A^*$  sont choisis de telle manière que la fonction de transfert  $B^*/A^*$  corresponde à la dynamique de poursuite souhaitée.

## 7.5 Poursuite et régulation à objectifs indépendants

Cette méthode ressemble à la méthode de placement de pôles à ceci près que l'on va effectuer une simplification des zéros de  $B^*(q^{-1})$  : cela ne s'applique donc qu'aux systèmes avec des pôles stables. Plus précisément il faut qu'ils aient un amortissement suffisant ( $> 0,2$ ) de manière à ne pas être trop près du cercle unité.

La structure de l'asservissement est toujours celle donnée sur la Figure 33.

**Choix de P**

$P(q^{-1})$  est choisi de manière à simplifier  $B^*(q^{-1})$  (rappel : la forme de B est donnée dans l'équation 74). Comme indiqué dans le chapitre 7.4.2 page 60 il s'agit donc d'imposer les zéros de  $B(q^{-1})$  dans  $H_R(q^{-1})$  et  $H_S(q^{-1})$ .

**Choix de T**

Si l'on choisit  $T(q^{-1}) = P(q^{-1})$  alors le transfert entre  $y^*$  et  $y$  est :

$$S_{yy^*}(q^{-1}) = q^{-(d+1)} \quad (85)$$

**Réglage de  $B^*$  et  $A^*$** 

$B^*$  et  $A^*$  sont choisis de telle manière que la fonction de transfert  $B^*/A^*$  corresponde à la dynamique de poursuite souhaitée.



## 8. TECHNOLOGIE DE LA COMMANDE PAR ORDINATEUR

Du point de vue de l'automaticien le ordinateur est relié au système physique à piloter par un certain nombre d'intermédiaires plus ou moins gênants :

- les interfaces analogiques et numériques,
- le transport de l'information par bus et protocole,
- l'utilisation éventuelle de capteurs ou d'actionneurs « intelligents ».

Il ne s'agit pas ici de faire un exposé complet sur ces trois aspects mais de poser les problèmes qui se posent à l'automaticien lorsque le système qu'il doit commander est perturbé par ces éléments.

Enfin ce chapitre se termine sur les notions de « Hardware In the Loop » et de « prototypage rapide » qui sont partie intégrante des processus de réalisation de systèmes de contrôle.

### 8.1 Calculateurs et interfaces numériques / analogiques

Le premier problème à prendre en compte est la quantification de l'information. Du point de vue du ordinateur d'une part et du point de vue des convertisseurs d'autre part. Dans un premier temps, considérons les aspects performance de calcul lié à la nature du codage des nombres dans le ordinateur.

#### Les performances liées au types de variables

Prenons l'exemple où le concepteur de la loi de commande souhaite réaliser un correcteur de type proportionnel. Entre l'acquisition des mesures et l'écriture de la consigne il y aura la réalisation d'une de commande du type :

$$y = K \cdot (\text{consigne} - \text{mesure}) \quad (86)$$

En utilisant un langage de programmation évolué (par exemple le C) la ligne de code sera la transcription littérale de (86). Le programmeur devra décider de la définition et du type des variables, principalement du nombre d'octets utilisés pour stocker ses variables. Le Tableau 5 résume par exemple les types utilisés en langage C (norme ANSI)

Tableau 5 : Types ANSI C

Types entiers	char	8 bits
	short int	16 bits
	long int	32 bits
Types flottants (mantisse et exposant)	float	32 bits
	double	64 bits
	longdouble	128 bits

Une fois le code écrit en C l'exécution du programme sur un processeur cible ne peut se faire que par l'intermédiaire d'une compilation du programme. A chaque processeur est donc adapté une version de compilateur.

Les performances du programme (sa rapidité d'exécution) dépendront donc en partie de la qualité du compilateur. (Par exemple, une multiplication s'effectue t'elle bêtement par un décalage de tous les bits ou bien des solutions astucieuses sont-elles implémentées pour arrêter le décalage selon le contexte ?)

Le principal déterminant de la performance est cependant le processeur : son type (8 bits, 16 bits ou 64 bits) et son cadencement (en MHz ou GHz) sont les deux plus importants paramètres. Coder les variables sur 64 bits (double) dans le programme coûtera cher en temps de calcul si le processeur est un 8 bits. Le type de mémoire et leurs temps d'accès peuvent aussi être déterminants. Enfin, certains processeurs disposent d'un processeur spécifiquement dédié aux opérations mathématiques (coprocesseur mathématique).

Il n'est pas possible de dire exactement, pour tel type de compilateur associé à tel type de processeur, le temps que prendra par exemple une multiplication. En pratique on effectue une série de tests types (benchmark) et on affiche un temps moyen d'exécution.

En résumé, on peut être tenté, pour ne pas s'embêter, de coder toutes les variables en virgule flottante. Mais il faut bien savoir que cela a un coût en terme de temps de calcul, surtout si le processeur n'est pas adapté. Pour fixer les idées on peut dire qu'il y a un rapport 20 entre une addition réalisée en "short" et une addition réalisée en virgule flottante.

Enfin, l'utilisation de fonctions mathématiques (sinus, exponentielles, etc...) peut aussi avoir un coût très lourd si le travail n'est pas réalisé par un coprocesseur mathématique dédié. Le compilateur et sa librairie mathématique joue là encore un grand rôle.

### La quantification imposée par le processeur

Au-delà de l'aspect performance et temps de calcul, la quantification de l'information doit parfois aussi être prise en compte. Rappelons qu'un entier codé sur N bits peut prendre seulement  $2^N - 1$  valeurs différentes. Avant d'effectuer les calculs mathématiques dans le corps du programme il est donc judicieux d'appliquer un



facteur d'échelle de manière à utiliser l'amplitude maximale possible (entre 0 et  $2^N - 1$ ).

### **La quantification imposée par les conversions analogique et numérique**

Les convertisseurs analogique-numérique et numérique-analogique sont principalement caractérisés par leur amplitude de signal analogique d'entrée (généralement en volts, par exemple +/- 10V) et par leur résolution en bits (8 ou 10 bits pour les plus courants, jusqu'à 16 bits pour les convertisseurs haut de gamme).

Prenons l'exemple d'un capteur gyromètre transformant une vitesse de rotation en tension. Compte tenu de l'application le signal utile varie entre 2,4V et 2,6V. Si le signal mesuré est branché directement sur un convertisseur 0-10V 10bits alors un problème de conditionnement du signal se pose. En effet la quantification est de  $\frac{10V}{2^{10}-1} = 0,098V$ . Les 0,2V de signal utile seront donc discrétisés en  $\frac{0,2}{0,098} = 20$  intervalles seulement !

Si la discrétisation du signal utile est trop grossière cela peut se traduire par une diminution des performances de l'asservissement, voire une instabilité. Le système n'est en effet plus linéaire et il n'existe pas d'outils théoriques permettant de prévoir les performances du système bouclé.

## **8.2 Transport de l'information : les bus et les protocoles**

Pas question de faire ni même d'ébaucher un cours sur les bus et les protocoles ici. Mais on va quand même donner quelques indications. L'automaticien a l'habitude de simplifier le problème qu'il se pose en dessinant un schéma simplifié de son problème d'asservissement (du type de la Figure 20 page 45) pour pouvoir appliquer les théories de la commande. Cependant il arrive bien souvent que l'information transite à l'intérieur du système, soit entre différents ordinateurs effectuant différentes parties du travail soit entre ordinateur et périphériques, par l'intermédiaire de bus informatique. A chaque type de bus correspond un ou plusieurs protocoles qui peuvent poser des problèmes pour l'automaticien :

- les données sont-elles discrétisées ? (le protocole peut par exemple ne permettre de transmettre que des paquets de données de 16 bits),
- le bus induit-t-il un retard ? (le composant mettant en forme les données du ordinateur pour les publier sur le bus a certaines performances qui peuvent induire un délai)
- le bus est-il déterministe ? (Certains bus ne garantissent pas que les données arriveront en un temps donné),
- le bus est-il sûr ? (Certains bus ne garantissent pas non plus que les données arriveront bien)

### **8.3 Capteurs et actionneurs "intelligents", "smart sensors"**

De plus en plus souvent les capteurs et les actionneurs sont associés à leur électronique de puissance, électronique de commande et de conditionnement de signal. L'interface avec le capteur ou l'actionneur se fait donc via un protocole de communication (par exemple via un bus RS232).

L'intérêt est de véhiculer non pas des signaux analogiques, tensions ou courants, mais des signaux numériques beaucoup moins sensibles aux éventuelles perturbations.

On retrouve donc, entre le capteur ou l'actionneur et le ordinateur, la présence d'un bus et d'un protocole de communication.

### **8.4 La philosophie "hardware in the loop"**

Les systèmes embarqués sont de plus en plus complexes et touchent des domaines de plus en plus variés : automobile, satellites, aéronautique, UAV, systèmes d'arme, sous-marins, systèmes, etc... Les systèmes à piloter ainsi que les systèmes de commande sont de plus en plus complexes.

Les techniques de simulation « Hardware In the Loop » (HIL) sont de plus en plus utilisées dans le développement et dans le test de ces systèmes embarqués. L'idée est de développer entièrement le système de contrôle sur une plateforme de test en utilisant le maximum de composants réels et en ne remplaçant que le minimum par la simulation. Par exemple en aéronautique l'« Iron Bird » regroupe dans un hangar tous les systèmes réels de l'avion en cours de développement. Seule la mécanique du vol est simulée : les sorties des actionneurs sont mesurées, rentrées dans un modèle de simulation numérique, puis les mesures effectuées normalement par les capteurs sont calculées et restituées par le modèle.

L'intérêt est de diminuer de manière importante les coûts et les délais en économisant le plus grand nombre possible de tests sur le système réel en situation. Lors des premiers tests réels les techniques « Hardware In the Loop » permettent d'avoir effectué la plus grande part du développement des lois de commande.

### **8.5 Le prototypage rapide en automatique**

Un peu dans le même ordre d'idée que le « Hardware In the Loop » le prototypage rapide permet de réaliser la conception des lois de commande en simulation puis de transporter rapidement (en un simple click !) le code du contrôleur simulé vers la cible temps réel implémentée sur le système réel.

La société Mathworks [9] investit massivement dans ce domaine en ayant enrichi son logiciel de calcul scientifique (Matlab) avec un logiciel de simulation graphique très évolué (Simulink) puis en développant des outils de prototypage rapide associés (Real Time Workshop, Embedded Target For...).

## 9. ANNEXE

### 9.1 Annexe 1 : Résolution de l'équation polynomiale de Bezout

Il s'agit de trouver R et S vérifiant :

$$P(q^{-1}) = A(q^{-1}).S(q^{-1}) + B(q^{-1}).R(q^{-1})$$

Où P, A et B sont connus.

Dans le cas où P, A, B, R et S sont des nombres entiers la solution de l'algorithme est donnée par l'algorithme de division Euclidienne étendu [8].

Dans le cas où P, A, B, R et S sont des polynômes, l'algorithme est donné ci-dessous, programmé en matlab ([1]).

```
function [Rp,Sp,nrp,nsp]=bezoutd(A,B,Hs,Hr,P)
%function [Rp,Sp,nrp,nsp]=bezoutd(A,B,Hs,Hr,P)
%solves AHsSp+BHRp=P by coefficient comparison. Delay and discretization delay need to be integrated in B
%(zeros at the beginning).
%
%inputs:
%A=[a0 a1 ... aNa] ... vector of model denominator coefficients A=a0 + a1z^(-1) + a2z^(-2) +...+ aNaz^(-Na)
%B=[b0 b1 ... bNb] ... vector of model numerator coefficients B=b0 + b1z^(-1) + b2z^(-2) +...+ bNbz^(-Nb)
%Hs=[hs0 hs1 ... hsNhs] ... vector of controller denominator fixed part Hs=hs0 + hs1z^(-1) +...+ hsNhsz^(-Nhs)
%Hr=[hr0 hr1 ... hrNhr] ... vector of controller denominator fixed part Hr=hr0 + hr1z^(-1) +...+ hrNhrz^(-Nhr)
%P=[p0 p1 ... pNp] ... vector of desired polynomial coefficients P=p0 + p1z^(-1) + p2z^(-2) +...+ pNpz^(-Np)
%outputs:
%Rp=[rp0 rp1 rp2 ...] ... vector of coefficients for resulted controller numerator
%Sp=[sp0 sp1 sp2 ...] ... vector of coefficients for resulted controller denominator
%nrp ... order of Rp
%nsp ... order of Sp
%
%written by: J. Langer, I.D. Landau, H. Prochazka
%7th june 2002

PRECISION=1e-16;

D=size(A);
if D(1)>1, A=A'; end;
```

```

D=size(B);
if D(1)>1, B=B'; end;
D=size(Hs);
if D(1)>1, Hs=Hs'; end;
if D(1)==0, Hs=1; end;
D=size(Hr);
if D(1)>1, Hr=Hr'; end;
if D(1)==0, Hr=1;end;
D=size(P);
if D(1)>1, P=P'; end;

na=length(A)-1;
nb=length(B)-1;
np=length(P)-1;
nhs=length(Hs)-1;
nhr=length(Hr)-1;

if (nhs>0), Ah=real(convz(A,Hs)); else Ah=A*Hs; end; % Ah = A * Hs
nah=length(Ah) -1;
if (nhr>0), Bh=real(convz(B,Hr)); else Bh=B*Hr; end;
nbh=length(Bh) -1;
if (np>nah+nbh-1), disp('Bezout error: too many poles');end;

% increase size of P using zeros if necessary
if (np<nah+nbh-1),
%
% set remaining poles onto a circle with radius rmin
P=[P zeros(1,nah+nbh-1-np)];
rootsPdes=roots(P);
nextra=nah+nbh-1-np;
rmin=1e-16;
angle=[0:nextra-1]'/nextra*2*pi;
j=sqrt(-1);
rootsPextra=rmin*exp(j*angle);
P=poly([rootsPdes;rootsPextra]);
np=nah+nbh-1;

end;
P,

nsp=nbh-1;
nrp=nah-1;

%matrix is smaller than vector PD

```

```

if (np>nah+nbh-1),
    disp('Order of model denominator is too low! Add a polynom of higher order to Hs or Hr. ');
end;

% ns=nsp+nhs
% nr=nrp+nhr

M=[];
for j=1:nsp+1,
    V=[];
    if (j>1), V=[V ; zeros(j-1,1)]; end;% zeros in front of Ah
    V=[V ; Ah'];% Ah
    if (j<=nsp), V=[V ; zeros(nsp+1-j,1)]; end;% zeros behind Ah
    if (length(V)~=nah+nbh), disp('bezoutb: error V'); end;
    M=[M V]; % add one column to M
end;

for j=1:nrp+1,
    V=[];
    if (j>1), V=[V ; zeros(j-1,1)]; end;
    V=[V ; Bh'];
    if (j<=nrp), V=[V ; zeros(nrp+1-j,1)]; end;
    if (length(V)~=nah+nbh), disp('bezoutb: error V'); end;
    M=[M V];
end;

D=size(M);
if (D(1)~=nah+nbh), disp('bezoutb: error size M row'); end;
if (D(2)~=nah+nbh), disp('bezoutb: error size M column'); end;

% make P column vector
P=P';

global M1;
M1=M;

X= M\P;
% coefficients are real values
X=real(X);

% make X row vector
X=X';

```

## Commande des systèmes par ordinateur

```
Sp=X(1:nsp+1);  
Rp=X(nsp+2:nsp+nrp+2);
```

## 10. BIBLIOGRAPHIE

- [1] I. D. Landau, "*Commande des systèmes*", Edition Lavoisier 2002.
- [2] C. Nouals, "Théorie du Signal", Notes de cours ENSICA
- [3] F. Vincent, "*Traitement Numérique du Signal*", Notes de cours ENSICA.
- [4] D. Arzelier, "*Représentation et analyse des systèmes, notes de cours*", Edition ENSICA. Voir aussi <http://www.laas.fr/~arzelier/>
- [5] J. Bordeneuve-Guibé, "*Commande des systèmes linéaires*", Notes de cours ENSICA.
- [6] Y. Briere, "*Identification*", Notes de cours ENSICA.
- [7] L. Ljung, "*System Identification, Theory for the User*", Prentice Hall 1987.
- [8] Wikipedia , équation de Bezout : [http://en.wikipedia.org/wiki/Bézout's\\_identity](http://en.wikipedia.org/wiki/Bézout's_identity)
- [9] <http://www.mathworks.com>





# 11. INDEX

<b>A</b>	
actionneur intelligent .....	21
amortissement .....	15
ANSI C .....	60
Approximation numérique d'équation différentielle .....	28
ARMAX .....	35
<b>B</b>	
BJ35 .....	
Blocage d'un signal .....	56
bloqueurs d'ordre 0 .....	22
BOZ .....	22
bus .....	21, 61
<b>C</b>	
CAN .....	21
capteur intelligent .....	21
causal .....	6, 9
Causalité .....	10
CNA .....	21
compilateur .....	60
convertisseur analogique / numérique .....	21
convertisseur numérique / analogique .....	21
convertisseurs analogique-numérique .....	61
coprocesseur mathématique .....	60
Correspondances pôles - zéros .....	31
<b>D</b>	
Diagramme de Bode .....	18
Discrétisation de fonction de transfert continues .....	27
division polynomiale .....	6
<b>E</b>	
échelon unitaire .....	11
équation d'état .....	17
Erreur statique .....	42
Estimation de la réponse fréquentielle .....	39
Estimation de la réponse impulsionnelle .....	38
Euler .....	29
<b>F</b>	
facteur d'échelle .....	61
facteur d'oubli .....	36
<i>fenêtre de Hamming</i> .....	39
filtre anti-repliement .....	25
fonction de sensibilité .....	42
fonction de transfert discrète .....	9
<b>G</b>	
gain statique .....	11
gain statique unitaire .....	42
<b>I</b>	
Identification .....	32
Identification non paramétrique .....	38
intercorrélation .....	38
<b>L</b>	
lieu des racines .....	43
lieu d'Evans .....	43
LTI .....	8
<b>M</b>	
marge de gain .....	47
marge de module .....	46
marge de phase .....	47
méthode des moindres carrés récursifs .....	32
Méthode des trapèzes .....	29
modèle ARX .....	32
<b>O</b>	
OE .....	35
one step ahead prediction .....	35
opérateur de retard .....	9
<b>P</b>	
période d'échantillonnage (choix) .....	26
Placement de pôles .....	54
Pôles .....	11
Poursuite .....	56
<i>prédiction</i> .....	32
protocoles .....	61
Pseudo Random Binary Signal .....	34
pulsation .....	15
<b>Q</b>	
quantification .....	23, 24, 60
<b>R</b>	
R,S,T .....	41
rapidité .....	15
rayon de convergence .....	5, 9
Régulation .....	55

Rejet de perturbations .....	42
Rejet d'une perturbation harmonique .....	56
réponse à un échelon.....	11
réponse fréquentielle.....	18
Réponse impulsionnelle.....	10
Réponses indicielle .....	10
retard.....	23
root locus .....	43

### S

signal discret .....	5
smart sensors.....	62
stabilité.....	11
Systèmes du premier ordre.....	14
Systèmes du second ordre.....	15

### T

Théorème de Nyquist.....	46
théorème de Shannon.....	19, 24
théorème du retard .....	6
Transformée en z .....	5
transformée unilatérale .....	5
Tustin.....	29

### V

virgule flottante.....	60
------------------------	----

### Z

zero holder hold .....	23
zéros.....	11
zoh .....	23

# Commande des systèmes par ordinateur

Yves Briere  
ENSICA

Notes de cours  
2007



# TABLE DES MATIERES

Table des matières .....	1
1. Introduction .....	1
2. Rappels et prérequis .....	3
3. Signaux et systèmes discrets .....	5
3.1 Signaux discrets .....	5
3.1.1 Transformée en z .....	5
3.1.2 Signaux types .....	7
3.2 Systèmes discrets .....	9
3.2.1 Fonction de transfert discrète, définition .....	9
3.2.2 Notation "opérateur de retard $q^{-1}$ " .....	10
3.2.3 Réponse impulsionnelle .....	11
3.2.4 Causalité .....	11
3.2.5 Réponses indicielle .....	11
3.2.6 Régime stationnaire (gain statique) .....	12
3.2.7 Pôles, zéros, stabilité .....	12
3.2.8 Systèmes du premier ordre .....	15
3.2.9 Systèmes du second ordre .....	16
3.2.10 Calcul itératif de la réponse d'un système .....	18
3.2.11 Représentation par équation d'état .....	18
3.3 Réponse fréquentielle des systèmes échantillonnés .....	19
3.3.1 Diagramme de Bode d'un système échantillonné .....	19
3.3.2 Où l'on voit apparaître le théorème de Shannon .....	20
4. Systèmes continus échantillonnés .....	23
4.1 Structure des systèmes commandés par ordinateur .....	23
4.2 Échantillonnage des systèmes continus .....	24
4.3 Effets de l'échantillonnage .....	25
4.4 Où l'on revoit le théorème de Shannon .....	26
4.5 Discrétisation de fonction de transfert continues .....	29
4.5.1 Approximation numérique d'équation différentielle .....	30
4.5.2 Superposition des réponses indicielles .....	32
4.5.3 Correspondances pôles - zéros .....	33
4.5.4 Bilan .....	33
5. Identification des systèmes échantillonnés .....	35
5.1 Identification d'un modèle ARX par la méthode des moindres carrés récurrents .....	35
5.2 L'identification paramétrique en général et en pratique .....	37

5.2.1	Choix du signal d'excitation, traitement des mesures .....	37
5.2.2	Choix d'une structure de modèle.....	38
5.2.3	Choix d'un critère à minimiser.....	38
5.2.4	Choix d'un algorithme.....	38
5.2.5	Validation.....	39
5.2.6	Utilisation de logiciels .....	39
5.3	Identification "simple" .....	39
5.4	Identification non paramétrique .....	41
5.4.1	Estimation de la réponse impulsionnelle .....	41
5.4.2	Estimation de la réponse fréquentielle.....	42
6.	Systèmes bouclés .....	45
6.1	Fonction de transfert en boucle fermée .....	45
6.2	Erreur statique .....	46
6.3	Rejet de perturbations.....	46
6.4	Stabilité des systèmes à simple gain de rétroaction.....	47
6.5	Analyse des systèmes bouclés dans le domaine fréquentiel : stabilité.....	49
6.5.1	Théorème de Nyquist dans le cas où le système est stable en boucle ouverte (critère du revers) .....	50
6.5.2	Théorème de Nyquist dans le cas où le système est instable en boucle ouverte .....	50
6.5.3	Interprétations du lieu de Nyquist.....	50
7.	Méthodes de régulation des systèmes échantillonnés .....	53
7.1	Introduction .....	53
7.2	Structure d'un système commandé par ordinateur, hypothèses.....	54
7.2.1	Discretisation, choix de la période d'échantillonnage.....	54
7.2.2	Discretisation d'un correcteur continu .....	55
7.3	Régulateur PID numérique .....	55
7.3.1	Cas 1 .....	55
7.3.2	Cas 2 .....	58
7.4	Placement de pôles .....	59
7.4.1	Choix des pôles en boucle fermée .....	60
7.4.2	Régulation : calcul de R et S.....	60
7.4.3	Poursuite : calcul de T .....	61
7.5	Poursuite et régulation à objectifs indépendants .....	62
8.	Technologie de la commande par ordinateur.....	65
8.1	Calculateurs et interfaces numériques / analogiques.....	65
8.2	Transport de l'information : les bus et les protocoles .....	67
8.3	Capteurs et actionneurs "intelligents", "smart sensors" .....	68
8.4	La philosophie "hardware in the loop" .....	68
8.5	Le prototypage rapide en automatique .....	68
9.	Annexe .....	69
9.1	Annexe 1 : Résolution de l'équation polynomiale de Bezout.....	69
10.	Bibliographie.....	73
11.	Index.....	1







# 1. INTRODUCTION



Figure 1 : commande d'un systèmes de télémédecine par ordinateur, vu par Claude Serre



## 2. RAPPELS ET PREREQUIS

Ce polycopié ne comporte pas de rappels sur les systèmes continus. S'adressant aux étudiants en deuxième année de l'ENSICA il suppose connu les enseignements de première année et de seconde année en automatique et traitement du signal.

En première année le cours de théorie du signal [2] a rendu familier les notions de :

- transformée de Fourier, transformée de Laplace
- corrélation, convolution

Le cours de traitement numérique du signal [3] a abordé en plus :

- la discrétisation et la quantification des signaux.
- la transformée en z et la transformée de Fourier discrète.
- les filtres numériques RIF et RII.

En automatique [4] les bases de la représentation et de l'analyse des systèmes continus ont été posées. On ne reviendra pas dans ce cours sur les notions de :

- modèles mathématique des systèmes linéaires à temps continus à temps invariant (LTI) : fonction de transfert, équation d'état, pôles, zéros et dynamique.
- commandabilité et observabilité.
- stabilité des systèmes en boucle ouverte : pôles, critères de Routh-Hurwitz.
- stabilité des systèmes bouclés : critère de Nyquist, critère de Bode.
- lieu des racines.

En deuxième année les notions suivantes sont acquises [5] :

- synthèse des correcteurs par méthodes fréquentielles (avance et retard de phase, PID)
- commande par retour d'état (placement de pôles), commande Linéaire Quadratique (LQG)
- reconstruction de l'état par Observateur, observateur optimal de Kalman.



## 3. SIGNAUX ET SYSTEMES DISCRETS

### 3.1 Signaux discrets

#### 3.1.1 Transformée en z

Pour une approche mathématiquement rigoureuse de représentation des signaux discrets par la transformée en z consulter [3] et sa bibliographie. Dans ce cours on se cantonne à une approche utilitariste des outils et méthodes mathématiques tout en essayant de ne pas contredire fondamentalement la rigueur mathématique.

Un signal discret  $s(k)$  est une séquence de nombres réels appelés échantillons. L'indice  $k$  est généralement associé à un instant d'échantillonnage :  $t = k.T_e$ . La période d'échantillonnage  $T_e$  est supposée constante. Par définition la transformée en z du signal  $s(k)$  est la fonction de la variable  $z$ , notée abusivement  $s(z)$  :

$$s(z) = Z(s(k)) = \sum_{k=0}^{\infty} s(k) \cdot z^{-k} \quad (1)$$

La transformée définie ici est plus précisément appelée transformée unilatérale puisque les termes correspondant à  $k < 0$  ne sont pas pris en compte. Ceci n'a pas d'importance puisque les signaux que nous utilisons sont systématiquement causaux<sup>1</sup> :  $s(k) = 0$  pour  $k < 0$ .

Il y a bien évidemment un lien avec la transformée de Laplace et la transformée de Fourier (voir [3])

La fonction de la variable complexe  $s(z)$  est une série et ne converge pas forcément pour toute valeur de  $z$ . En toute rigueur il faudrait associer le rayon de convergence à chaque fonction en  $z$  :  $\{s(z) \text{ converge pour } |z| > \alpha\}$ .

Les automaticiens pour qui la transformée en z n'est bien souvent qu'une façon bien commode de regrouper des termes de récurrence utilisent souvent la pseudo-variable<sup>2</sup>  $q$ .

A partir d'une transformée en z  $s(z)$  il est possible de retrouver la séquence  $s(k)$  correspondante en appliquant la transformation inverse, dont on ne donnera pas la formule ici. A la place de faire appel à cette formule compliquée il est possible d'utiliser d'autres méthodes pour retrouver la séquence  $s(k)$  lorsque la fonction  $s(z)$  est donnée sous forme de fraction rationnelle :

<sup>1</sup> Pour une justification du terme « causal » voir page 11

<sup>2</sup> Voir page 10

- division polynomiale
- décomposition de la fraction rationnelle en éléments simples puis utilisation de transformations élémentaires telles que celles données dans le Tableau 2

La propriété fondamentale de la transformée en z est qu'elle est linéaire (la transformée en z d'une composition linéaire de signaux est la composition linéaire de la transformée en z de chaque signal). Les autres propriétés fondamentales sont données dans le Tableau 2.

Tableau 1 : propriétés de la transformée en z

1	$s(z) = Z(s(k)) = \sum_{k=0}^{\infty} s(k) \cdot z^{-k}$
2	$Z(\alpha \cdot s_1(k) + \beta \cdot s_2(k)) = \alpha \cdot Z(s_1(k)) + \beta \cdot Z(s_2(k))$
3	$Z(k \cdot s(k)) = -z \cdot \frac{ds(z)}{dz}$
4	$Z(c^k \cdot s(k)) = s\left(\frac{z}{c}\right)$
5	$Z(s(k-1)) = s(k=-1) + z^{-1} \cdot s(z)$
6	$Z(s(k+1)) = z \cdot s(z) - z \cdot s(k=0)$
7	$s(k=0) = \lim_{z \rightarrow \infty} (s(z))$
8	$s(k \rightarrow \infty) = \lim_{z \rightarrow 1} \left( \frac{z-1}{z} \cdot s(z) \right)$

Remarque 1 : la propriété 5 ne prend pas en compte le fait que le signal est causal : en pratique  $s(k=-1) = 0$ .

Remarque 2 : la propriété 5 justifie le fait que le terme  $z^{-1}$  soit appelé "opérateur de retard unitaire", cette propriété est aussi appelée "théorème du retard".

Remarque 3 : les propriétés 7 et 8 sont données sous réserve de convergence.

Transformées inverses

### Exemple 1 : transformée inverse par division polynomiale

On considère un signal  $s(k)$  dont la transformée en z est donnée par la fraction rationnelle :

$$S(z) = \frac{0.1 \cdot z^2 + 0.1 \cdot z}{z^3 - 2.6 \cdot z^2 + 2.2 \cdot z - 0.6} \quad (3.1)$$

L'objectif est d'obtenir  $S(z)$  sous forme polynomiale :

$$S(z) = x_0 + x_1 \cdot z^{-1} + x_2 \cdot z^{-2} + \dots \text{etc.} \tag{3.2}$$

Cette suite est généralement infinie, la réponse temporelle est donnée par la succession des termes  $\{x_i\}$ . En traitant l'équation (3.1) comme une simple division l'algorithme de division classique (celui appris en CM2) permet d'obtenir par récurrence tous les termes de la suite (3.2) :

$0.1 \cdot z^2$	$+ 0.1 \cdot z$				$z^3 - 2.6 \cdot z^2 + 2.2 \cdot z - 0.6$
$0.1 \cdot z^2$	$- 0.26 \cdot z$	$+ 0.22$	$- 0.06 \cdot z^{-1}$		
0	$0.36 \cdot z$	$- 0.22$	$0.06 \cdot z^{-1}$		
	$0.36 \cdot z$	$- 0.936$	$0.792 \cdot z^{-1}$	$- 0.216 \cdot z^{-2}$	
0	$0.716$	$- 0.732 \cdot z^{-1}$	$\dots$	$\dots$	

Les éléments successifs de la suite  $s(k)$  sont donc  $s(k) = \{0, 0.1, 0.36, 0.716, \dots\}$ . Pour obtenir les termes suivants il faudrait continuer la division.

**Exemple 2 : transformée inverse par décomposition en éléments simples**

La deuxième méthode présentée consiste à décomposer la fraction rationnelle (3.1) en éléments simples puis d'utiliser les tables de transformées en  $z$  (Tableau 2). Pour notre exemple cela donne :

$$S(z) = \frac{0.1 \cdot z^2 + 0.1 \cdot z}{z^3 - 2.6 \cdot z^2 + 2.2 \cdot z - 0.6} = \frac{-z}{z-1} + \frac{0.5 \cdot z}{(z-1)^2} + \frac{z}{z-0.6}$$

La réponse temporelle est donc composée des trois éléments :

$$s(k) = -(1 \ 1 \ 1 \ \dots) + 0.5 \cdot (1 \ 2 \ 3 \ \dots) + (1 \ 0.6 \ 0.6^2 \ \dots)$$

$$\text{soit : } s(k) = \begin{cases} s(0) = -1 + 0.5 + 1 \\ s(2) = -1 + 0.5 \cdot 2 + 0.6 \cdot 0.5 \\ s(3) = -1 + 0.5 \cdot 3 + 0.6^2 \\ s(4) = \dots \end{cases}$$

**3.1.2 Signaux types**

Le Tableau 2 donne les transformée en  $z$  de certains signaux fondamentaux. La Figure 2 représente l'allure de quelques signaux types.

Tableau 2 : transformée en z élémentaires

$s(k)$	$S(z)$
signal temporel	Transformée en z
$\delta(k)$ : impulsion unité	1
$u(k)$ : échelon unité	$\frac{z}{z-1}$
$k \cdot u(k)$	$\frac{z}{(z-1)^2}$
$c^k \cdot u(k)$	$\frac{z}{z-c}$
$k \cdot a^k \cdot u(k)$	$\frac{a \cdot z}{(z-a)^2}$
$\sin(\omega \cdot k) \cdot u(k)$	$\frac{z \cdot \sin(\omega)}{z^2 - 2 \cdot z \cdot \sin(\omega) + 1}$
$\cos(\omega \cdot k) \cdot u(k)$	$\frac{z \cdot (z - \cos(\omega))}{z^2 - 2 \cdot z \cdot \sin(\omega) + 1}$
$a^k \cdot \sin(\omega \cdot k) \cdot u(k)$	$\frac{a \cdot z \cdot \sin(\omega)}{z^2 - 2 \cdot a \cdot \cos(\omega) \cdot z + a^2}$
$a^k \cdot \cos(\omega \cdot k) \cdot u(k)$	$\frac{z \cdot (z - a \cdot \cos(\omega))}{z^2 - 2 \cdot a \cdot \cos(\omega) \cdot z + a^2}$



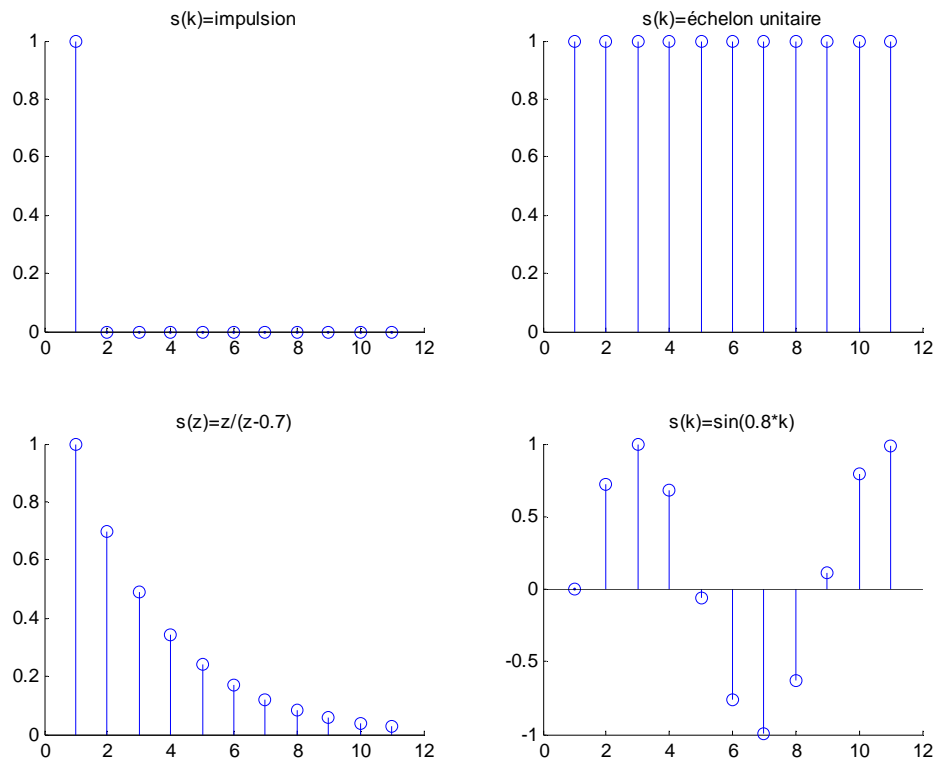


Figure 2 : réponses de quelques signaux types

## 3.2 Systèmes discrets

### 3.2.1 Fonction de transfert discrète, définition

On considère maintenant un système discret linéaire et invariant dans le temps (LTI) représenté sur la Figure 3.

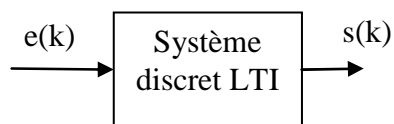


Figure 3 : système discret, notations

La relation entre la séquence d'entrée et la séquence de sortie est linéaire et définie par un ensemble de coefficients constants  $a_{1:n}$  et  $b_{1:m}$  :

$$\begin{aligned} s(k) + a_1 \cdot s(k-1) + a_2 \cdot s(k-2) + \dots + a_n \cdot s(k-n) = \\ b_0 \cdot e(k) + b_1 \cdot e(k-1) + b_2 \cdot e(k-2) + \dots + b_m \cdot e(k-m) \end{aligned} \quad (2)$$

On peut appliquer la transformée en  $z$  à chaque membre de l'équation puis en appliquant en récurrence la propriété 5 (théorème du retard) on obtient la fonction de transfert discrète du système :

$$\frac{s(z)}{e(z)} = F(z) = \frac{b_0 + b_1 \cdot z^{-1} + b_2 \cdot z^{-2} + \dots + b_m \cdot z^{-m}}{1 + a_1 \cdot z^{-1} + a_2 \cdot z^{-2} + \dots + a_n \cdot z^{-n}} \quad (3)$$

Remarque 1 : la fonction de transfert précédente est normalisée en prenant  $a_0 = 1$ .

Remarque 2 : on remarquera que la fonction de transfert est donnée en  $z^{-1}$  plutôt qu'en  $z$ . Cela est justifié par le fait que celle-ci traduit l'équation d'évolution du système physique causal (2).

Le script Matlab permettant de créer un objet Matlab correspondant à une fonction de transfert discrète est donné ci-dessous.

#### Matlab

```
% Création d'un système linéaire discret :
Te=0.1;
sysd = tf([b0 b1 b2],[1 a0 a2 a3],Te,'variable','z^-1');
```

On notera l'utilisation de l'argument 'variable', 'z^-1'.

### 3.2.2 Notation "opérateur de retard $q^{-1}$ "

La fonction de transfert est une fonction de la variable complexe  $z$ . Il faudrait donc en toute rigueur l'associer à son rayon de convergence. En pratique on ne s'embarrasse jamais de préciser cette zone de convergence. C'est pourquoi on trouve souvent la notation utilisant l'opérateur  $q^{-1}$  de décalage à la place de la variable complexe  $z^{-1}$  (4).  $F(q)$  n'est alors plus à proprement parler une fonction mais on lui appliquera les mêmes manipulations algébriques qu'à  $F(z)$ .

$$F(q) = \frac{b_0 + b_1 \cdot q^{-1} + b_2 \cdot q^{-2} + \dots + b_m \cdot q^{-m}}{1 + a_1 \cdot q^{-1} + a_2 \cdot q^{-2} + \dots + a_n \cdot q^{-n}} \quad (4)$$

Attention : selon les auteurs l'opérateur qui remplace  $z^{-1}$  est noté  $q$  (Matlab) ou  $q^{-1}$  (Landau [1]). Dans la suite du document c'est cette dernière convention qui sera appliquée.

L'utilisation de l'opérateur de retard  $q^{-1}$  à la place de la variable complexe  $z^{-1}$  permet de traiter du cas des fonctions de transfert à coefficients non constant, cas non prévu par la transformation en  $z$ .

### 3.2.3 Réponse impulsionnelle

Essayons d'appliquer au système un signal d'entrée particulier : une impulsion unitaire. La transformée en  $z$  du système est obtenue en multipliant la fonction de transfert par la transformée en  $z$  de l'impulsion unitaire : 1. On obtient :

$$s(z) = \frac{b_0 + b_1 \cdot z^{-1} + b_2 \cdot z^{-2} + \dots + b_m \cdot z^{-m}}{1 + a_1 \cdot z^{-1} + a_2 \cdot z^{-2} + \dots + a_n \cdot z^{-n}} \times 1 \quad (5)$$

Cela justifie le fait que la fonction de transfert du système peut aussi s'appeler "réponse impulsionnelle".

#### Matlab

```
% Réponse impulsionnelle d'un système :
[y,t] =impulse(sysd);
```

### 3.2.4 Causalité

Un système est dit causal si les sorties ne dépendent que du passé et non pas du futur. La conséquence est que la réponse impulsionnelle de  $F(z)$  ne peut pas comporter de terme non nuls pour  $k < 0$ . Cela justifie la confusion sémantique entre "système causal" et "signal causal", c'est-à-dire ne comportant pas de termes non nuls pour  $k < 0$ .

Attention : à la différence des systèmes continus un système physique (causal) discret peut très bien être modélisé avec un degré  $m$  supérieur au degré  $n$  : le système peut avoir plus de zéros que de pôles.

La transformée en  $z$  représentant des signaux échantillonnés elle peut tout aussi bien représenter les mesures prises à différents endroits d'une poutre. Le temps étant remplacé ici par une abscisse la notion de causalité ne s'applique pas ! Attention : beaucoup des propriétés de systèmes échantillonnés que l'on étudie sont liés à la causalité, prudence donc si l'on applique ces méthodes à d'autres domaines !

### 3.2.5 Réponses indicielle

Pour caractériser des systèmes on s'intéresse souvent à leur réponse impulsionnelle ou, plus souvent, à leur réponse indicielle. Pour calculer la réponse

à un échelon d'un système il faut donc multiplier la fonction de transfert par la transformée en z d'un échelon unitaire :

$$s(z) = \frac{b_0 + b_1 \cdot z^{-1} + b_2 \cdot z^{-2} + \dots + b_m \cdot z^{-m}}{1 + a_1 \cdot z^{-1} + a_2 \cdot z^{-2} + \dots + a_n \cdot z^{-n}} \times \frac{1}{1 - z^{-1}} \quad (6)$$

Il faudrait ensuite calculer la transformée en z inverse de s(z). En pratique, si l'on doit faire le calcul à la main on décompose en éléments simples puis on utilise des tables. Avec Matlab c'est plus simple. Voici par exemple le script pour obtenir les 20 premiers échantillons de la réponse indicielle d'un système :

**Matlab**

```
% Réponse indicielle d'un système :
[y,t] =step(sysd,20);
```

### 3.2.6 Régime stationnaire (gain statique)

Le gain statique est obtenu en appliquant une entrée indicielle et en calculant la valeur du régime permanent. En appliquant le théorème de la valeur finale il vient :

$$F_0 = \lim_{z \rightarrow 1} \left( \frac{z-1}{z} \cdot \frac{b_0 + b_1 \cdot z^{-1} + b_2 \cdot z^{-2} + \dots + b_m \cdot z^{-m}}{1 + a_1 \cdot z^{-1} + a_2 \cdot z^{-2} + \dots + a_n \cdot z^{-n}} \times \frac{1}{1 - z^{-1}} \right) \quad (7)$$

Soit :

$$F_0 = \frac{\sum_{i=0}^{i=m} b_i}{1 + \sum_{i=1}^{i=n} a_i} \quad (8)$$

Le gain statique est obtenu donc simplement en faisant le quotient de la somme des coefficients.

### 3.2.7 Pôles, zéros, stabilité

Comme pour les systèmes continus la dynamique et la stabilité des systèmes dépend des propriétés du dénominateur de la fonction de transfert.

$$F(z) = \frac{e_0 \cdot (1 + e_1 \cdot z^{-1}) \cdot (1 + e_2 \cdot z^{-1}) \cdots (1 + e_m \cdot z^{-1})}{(1 + c_1 \cdot z^{-1}) \cdot (1 + c_2 \cdot z^{-1}) \cdots (1 + c_n \cdot z^{-1})} \quad (9)$$

Les coefficients  $c_i$  sont les pôles de la fonction de transfert, les coefficients  $e_i$  sont les zéros. La fonction de transfert étant au départ composée de coefficients purement réels les coefficients  $c_i$  sont soit réels soit imaginaires par paires conjuguées.

Une fonction de transfert est donc aussi bien définie par la donnée de ses pôles et de ses zéros. Le script Matlab pour créer un système ainsi est donné ci-dessous :

#### Matlab

```
% Système discret défini par ses pôles et zéros :
Te=0.1;
sysd=zpk([c1 c2 c3],[e1 e2],e0,Te);
```

Si l'on décompose la fonction de transfert en éléments simples on obtient une somme d'éléments du type :

$$F(z) = \frac{d_1}{1 + c_1 \cdot z^{-1}} + \frac{d_2}{1 + c_2 \cdot z^{-1}} + \cdots + \frac{d_n}{1 + c_n \cdot z^{-1}} \quad (10)$$

Un système linéaire temps invariant est dit stable si sa réponse impulsionnelle tend asymptotiquement vers 0. Compte tenu de (10) cela nécessite que chaque pôle  $c_i$  soit de module inférieur à 1.

**Propriété** : un système discret LTI est stable si et seulement si tous ses pôles ont leur modules inférieur ou égal à 1.

#### Critère algébrique attestant de la stabilité : critère de Routh

On rappelle que le critère de Routh permet de prédire si les racines d'un polynôme sont toutes à partie réelle négative, donc stables. Énonçons le théorème de Routh dans le cas d'un polynôme  $P'(w)$  :

$$P'(w) = a_0 + a_1 \cdot w + a_2 \cdot w^2 + \cdots + a_n \cdot w^n \quad (11)$$

Dans un premier temps on construit le tableau :

$$\begin{array}{cccccc|c}
 \beta_{n,0} & \beta_{n,1} & \beta_{n,2} & \beta_{n,q-1} & \beta_{n,q} & 0 & \\
 \beta_{n-1,0} & \beta_{n-1,1} & \beta_{n-1,2} & \beta_{n-1,q-1} & \beta_{n-1,q-1} & 0 & \\
 \beta_{n-2,0} & \beta_{n-2,1} & \beta_{n-2,2} & \beta_{n-2,q-1} & \beta_{n-2,q-1} & & \\
 \\ 
 \beta_{2,0} & \beta_{2,1} & 0 & & & & \\
 \beta_{1,0} & 0 & & & & & \\
 \beta_{0,0} & 0 & & & & & 
 \end{array} \quad (12)$$

Puis on vérifie que tous les premiers éléments de la première colonne ont le même signe : cela garantit que tous les pôles de  $P'(w)$  ont leur partie réelle négative.

Les éléments du tableau sont construits par récurrence ligne par ligne en appliquant les formules suivantes :

Pour  $i = n$  et  $i = n - 1$  :

$$\begin{cases} \beta_{n,j} = a_{n-2j} \\ \beta_{n-1,j} = a_{n-1-2j} \end{cases} \quad (13)$$

Et pour  $i < n-1$  :

$$\beta_{i,j} = \frac{\beta_{i+1,0} \cdot \beta_{i+2,j+1} - \beta_{i+1,j+1} \cdot \beta_{i+2,0}}{\beta_{i+1,0}} \quad (14)$$

Dans le cas qui nous intéresse, un système échantillonné, on cherche plutôt à vérifier que les pôles du polynôme  $P(z)$  sont de module inférieur à un. L'idée est d'effectuer un changement de variable  $z \rightarrow w$  pour pouvoir appliquer le critère de Routh.

$$P'(w) = (1-w)^n \cdot P\left(\frac{1+w}{1-w}\right) \quad (15)$$

Si l'on effectue la transformation (15) alors le fait que  $P'(w)$  ait toutes ses racines à partie réelle négative garantit que  $P(z)$  a toutes ses racines dans le cercle unité. Pour vérifier la stabilité de  $P(z)$  il suffit donc d'effectuer le changement de variable (15) puis d'appliquer le critère de Routh conventionnel (12 13 14).

### Critère algébrique attestant de la stabilité : critère de Jury

Le critère de Jury s'apparente au critère de Routh. Nous le donnons pour un polynôme de degré 2, 3 et 4 du type :

$$P(z) = a_0 + a_1 \cdot z + a_2 \cdot z^2 + \dots + a_n \cdot z^n \quad (16)$$

Les conditions de stabilité sont données ci-dessous (ces conditions sont données dans le cas  $a_n > 0$ , il suffit de multiplier tous les termes par  $-1$  si cette condition n'est pas remplie par  $P(z)$ ) :

$$\begin{aligned}
 n = 2 & \begin{cases} a_0 + a_1 + a_2 > 0 \\ a_0 - a_1 + a_2 > 0 \\ -a_0 + a_2 > 0 \end{cases} \\
 n = 3 & \begin{cases} a_0 + a_1 + a_2 + a_3 > 0 \\ -a_0 + a_1 - a_2 + a_3 > 0 \\ -|a_0| + a_3 > 0 \\ a_0 a_2 - a_1 a_3 - a_0^2 + a_3^2 > 0 \end{cases} \\
 n = 4 & \begin{cases} a_0 + a_1 + a_2 + a_3 + a_4 > 0 \\ a_0 - a_1 + a_2 - a_3 + a_4 > 0 \\ a_4^2 - a_0^2 - |a_0 a_3 - a_1 a_4| > 0 \\ (a_0 - a_4)^2 (a_0 - a_2 + a_4) + (a_1 - a_3)(a_0 a_3 - a_1 a_4) > 0 \end{cases}
 \end{aligned} \quad (17)$$

### 3.2.8 Systèmes du premier ordre

On donne dans la Figure 4 les réponses indicielles de systèmes du premier ordre ainsi que la localisation des pôles dans le plan complexe. Le cercle de rayon unité représente la zone dans laquelle les pôles doivent se situer pour assurer la stabilité. On remarque sur les deux premières figures l'influence de la place du zéro par rapport au point  $(1,0)$  pour la rapidité.

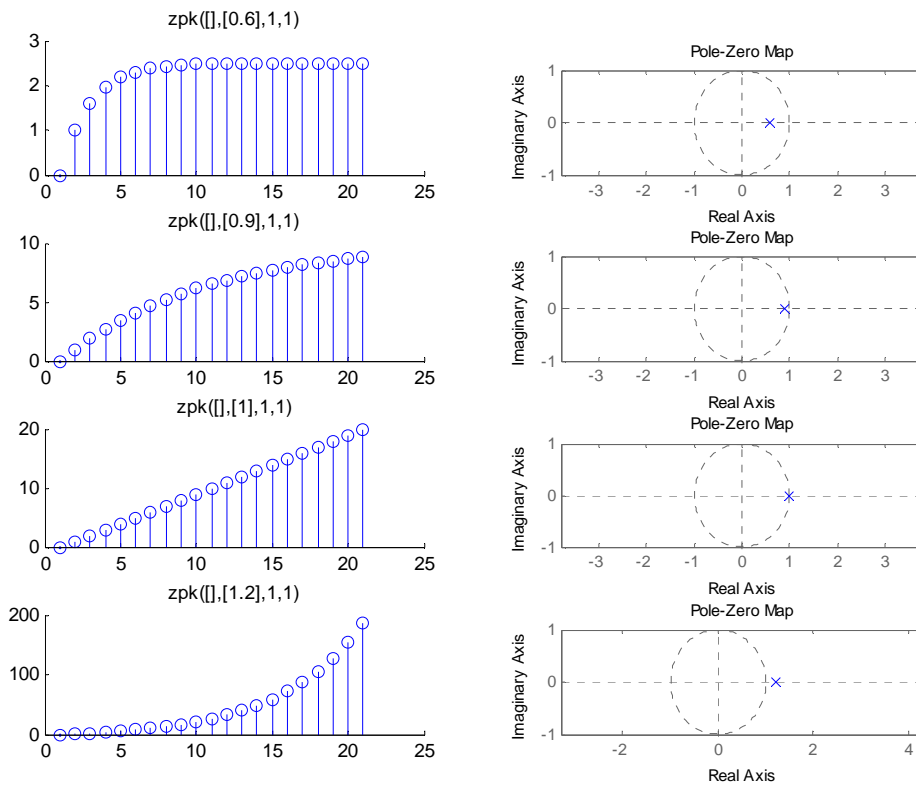


Figure 4 réponses indicielles de systèmes du premier ordre

### 3.2.9 Systèmes du second ordre

On donne dans la Figure 6 les réponses indicielles de systèmes du second ordre ainsi que la localisation des pôles dans le plan complexe. La rapidité et l'amortissement dépend de la position des pôles dans le cercle unité. La Figure 5 représente les courbes iso-amortissement et iso-pulsation des systèmes du second ordre.



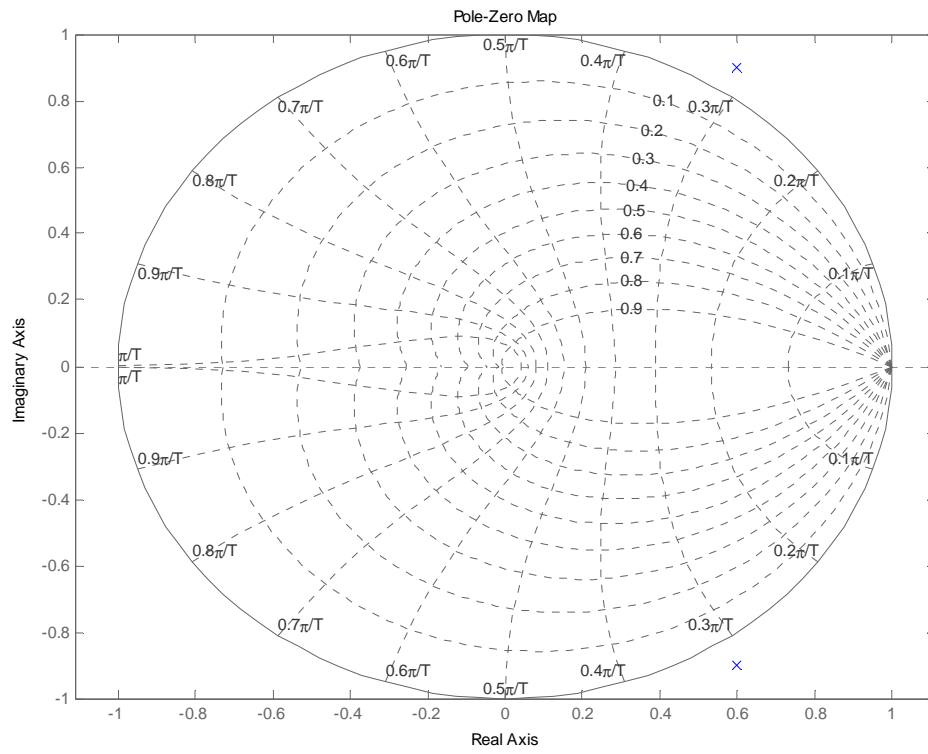


Figure 5 : courbes iso-amortissement et iso-pulsation des systèmes discrets du second ordre

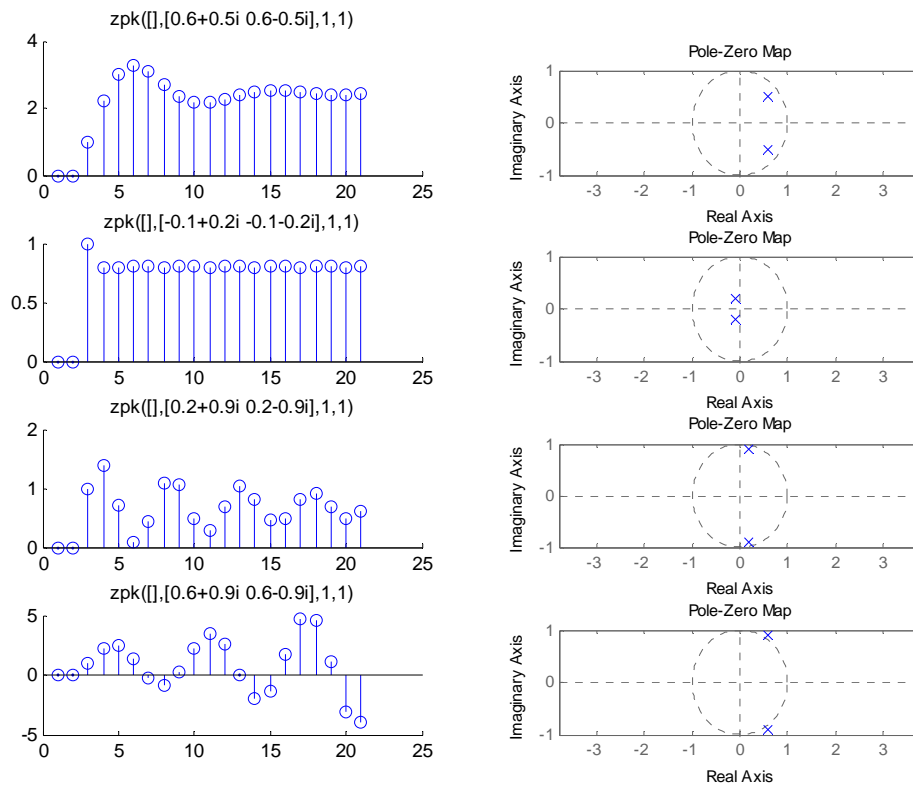


Figure 6 : réponses de systèmes du second ordre

### 3.2.10 Calcul itératif de la réponse d'un système

Si l'on dispose de la fonction de transfert d'un système et que l'on connaît les valeurs successives  $e(k)$  du signal d'entrée il est facile de calculer itérativement la sortie  $y(k)$  du système par l'équation de récurrence (2). Il faut initialiser par les  $n$  valeurs initiales de  $s(k)$ , prises généralement égales à 0.

### 3.2.11 Représentation par équation d'état

Un système discret peut aussi être représenté par un ensemble d'équations d'état comme dans le domaine continu (52). Cette représentation est naturellement adaptée aux systèmes multi-entrées multi-sorties. On définit alors un état interne  $x(k)$  ainsi qu'une fonction d'évolution  $f$  et une équation de mesure  $h$ .

$$\begin{cases} x(k+1) = f(x(k), u(k)) \\ s(k) = h(x(k), u(k)) \end{cases} \quad (18)$$

### 3.3 Réponse fréquentielle des systèmes échantillonnés

#### 3.3.1 Diagramme de Bode d'un système échantillonné

Lorsqu'on étudie la réponse fréquentielle des systèmes continus on considère des entrées particulières sinusoïdales :

$$\begin{cases} e(t) = \cos(2 \cdot \pi \cdot f \cdot t) + j \cdot \sin(2 \cdot \pi \cdot f \cdot t) = e^{j \cdot 2 \cdot \pi \cdot f \cdot t} \\ f \in [0 \quad \infty] \end{cases} \quad (19)$$

De même, pour des systèmes échantillonnés on considèrera des entrées sinusoïdales échantillonnées :

$$\begin{cases} e(k) = \cos(2 \cdot \pi \cdot f \cdot k) + j \cdot \sin(2 \cdot \pi \cdot f \cdot k) = e^{j \cdot 2 \cdot \pi \cdot f \cdot k} \\ f \in [0 \quad 1] \end{cases} \quad (20)$$

La sortie est aussi un signal échantillonné sinusoïdal de gain  $A(f)$  et de phase  $\Phi(f)$  variables et dépendant de  $f$ ; la fréquence réduite.

$$\begin{cases} s(k) = A(f) \cdot e^{j \cdot (2 \cdot \pi \cdot f \cdot k + \Phi(f))} \\ f \in [0 \quad 1] \end{cases} \quad (21)$$

Le nombre complexe, noté abusivement  $F(f)$ , d'amplitude  $A(f)$  et de phase  $\Phi(f)$  est obtenu à partir de  $F(z)$  en remplaçant  $z$  par  $e^{j \cdot 2 \cdot \pi \cdot f}$  :

$$\begin{cases} F(f) = F(z = e^{j \cdot 2 \cdot \pi \cdot f}) \\ f \in [0 \quad 1] \end{cases} \quad (22)$$

Le diagramme représentant le gain et la phase en fonction de la fréquence réduite s'appelle aussi un diagramme de Bode. L'échelle des fréquences est généralement linéaire de 0 à 0.5 (puisque le diagramme est nécessairement symétrique). La Figure 7 représente par exemple le diagramme de Bode d'un système du second ordre.

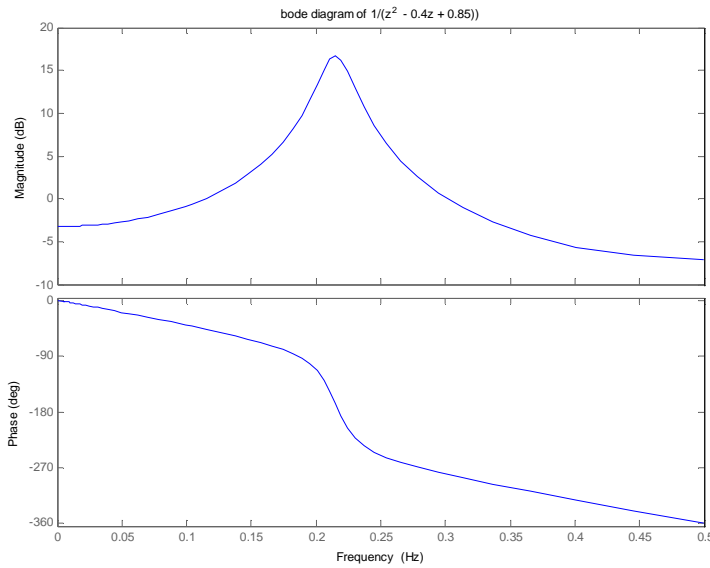


Figure 7 : exemple du diagramme de Bode d'un système discret du second ordre

### 3.3.2 Où l'on voit apparaître le théorème de Shannon

Le changement de variable  $z = e^{j2\pi f}$  n'est pas bijectif puisque  $e^{j2\pi f} = e^{j2\pi(f+k)}$  avec  $k$  entier. Si l'on fait le lien entre domaine continu et domaine fréquentiel on peut remarquer que le changement de variable  $z = e^{j2\pi f}$  fait correspondre une zone du plan complexe continu à une zone du plan complexe discret. La transformation est bijective si l'on se contente de la zone du plan complexe continu contenu entre les droites d'ordonnée 0 et 1.

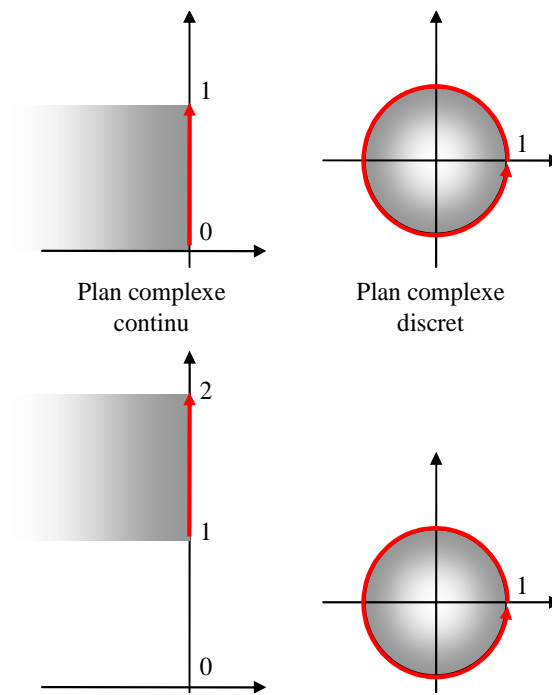


Figure 8 : correspondance entre plan complexe continu et plan complexe discret.



## 4. SYSTEMES CONTINUS ECHANTILLONNES

### 4.1 Structure des systèmes commandés par ordinateur

Les systèmes commandés par ordinateurs sont presque toujours des systèmes physiques continus par nature. Le ordinateur traite par contre les entrées et les sorties de manière discrète avec une période d'échantillonnage  $T_e$ . Les éléments importants, du point de vue de l'automaticien sont donc :

- le convertisseur analogique / numérique CAN ou du moins le système qui fournit séquentiellement les mesures au ordinateur à la période  $T_e$ .
- le ordinateur lui-même. Il doit effectuer l'acquisition des mesures, le traitement des informations, l'envoi des consignes dans un temps inférieur à  $T_e$ .
- le convertisseur numérique / analogique CNA.
- le système physique.

On suppose, pour faire simple et pour pouvoir utiliser la panoplie des outils mathématiques, que les systèmes sont échantillonnés à la même période  $T_e$  et au même moment.

La Figure 9 montre le système physique commandé par le ordinateur, dans sa version simpliste.

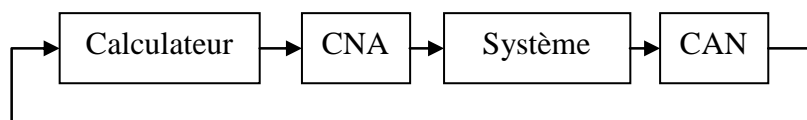


Figure 9 : système commandé par ordinateur

Au XXI<sup>ème</sup> siècle les choses sont quand même parfois un petit peu plus compliquées :

- le ordinateur fournit bien souvent un signal de commande numérique (transporté par bus informatique USB, RS232, CAN, Ethernet, etc...) directement interprété par l'actionneur. On parle "d'actionneur intelligent".
- de même le "capteur intelligent" fournit aussi bien souvent la mesure sous forme d'un signal numérique.
- Les signaux sont transportés par des bus informatiques qui n'ont pas que ça à faire : il peut y avoir du retard.
- de même les capteurs et actionneurs sont rarement tous seuls, ce qui se traduit par des périodes d'échantillonnage différentes, et parfois non synchrones...

- enfin le ordinateur peut aussi avoir d'autres choses à faire...

Ceci étant dit nous allons pour le moment nous restreindre au cas représenté sur la Figure 9 avec des instants d'échantillonnage synchrones et régulièrement espacés d'un temps  $T_e$ .

## 4.2 Echantillonnage des systèmes continus

On dispose généralement de la fonction de transfert continue (notée en utilisant la variable complexe  $s$  ou  $p$ )  $F(s)$  du système continu. Le problème est d'obtenir la fonction de transfert discrète du système échantillonné de manière à pouvoir synthétiser un correcteur. Le système continu est associé au convertisseur numérique – analogique qu'il faut prendre en compte.

Les convertisseurs numérique analogiques peuvent généralement se représenter par des "échantillonneurs – bloqueurs d'ordre 0" ou BOZ. Cela signifie que la commande est maintenue à une valeur constante pendant toute la durée du temps d'échantillonnage.

Si l'on applique une commande discrète impulsionnelle à un BOZ il va maintenir la commande 1 pendant la durée  $T_e$  puis revenir à zéro (Figure 10).

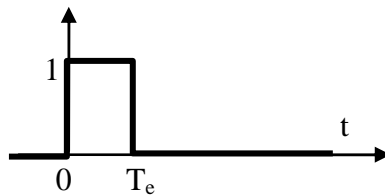


Figure 10 : réponse impulsionnelle d'un bloqueur d'ordre 0

La réponse impulsionnelle  $BOZ(t)$  est donc composée d'un échelon unitaire suivi d'un échelon unitaire de signe opposé et décalé d'un temps  $T_e$ . La transformée de Laplace de la réponse impulsionnelle du bloqueur est donnée par la formule (23) :

$$BOZ(s) = \frac{1}{s} - \frac{e^{-jT_e}}{s} \quad (23)$$

Du point de vue du ordinateur il faut donc prendre en compte le système continu de fonction de transfert  $F_{BOZ}(s)$  que l'on devra échantillonner :

$$F_{BOZ}(s) = \frac{1 - e^{-jT_e}}{s} \cdot F(s) \quad (24)$$



$$F_{\text{BOZ}}(z) = Z^{-1} \left( \frac{1 - e^{-jT_e}}{s} \cdot F(s) \right) \quad (25)$$

En pratique l'obtention de la transformée en z d'un système échantillonné par bloqueur d'ordre 0 se fait à l'aide de tables. On donne quelques exemples fondamentaux dans le Tableau 3 ci-dessous.

Tableau 3 : transformée en z de systèmes continus échantillonnés

F(s)	F <sub>BOZ</sub> (z)
1	1
$\frac{1}{s}$	$\frac{T_e \cdot z^{-1}}{1 - z^{-1}}$
$\frac{1}{1 + T \cdot s}$	$\begin{cases} \frac{b_1 \cdot z^{-1}}{1 + a_1 \cdot z^{-1}} \\ b_1 = 1 - e^{-T_e/T} \quad a_1 = -e^{-T_e/T} \end{cases}$
$\frac{e^{-Ls}}{1 + T \cdot s}$	$\begin{cases} \frac{b_1 \cdot z^{-1} + b_2 \cdot z^{-2}}{1 + a_1 \cdot z^{-1}} \quad a_1 = -e^{-T_e/T} \\ b_1 = 1 - e^{-(L-T_e)/T} \quad b_2 = e^{-T_e/T} (e^{L/T} - 1) \end{cases}$
$L < T_e$	

Si l'on dispose de Matlab c'est bien sûr encore plus simple. On utilise la fonction c2d avec l'argument 'zoh' pour "zero holder hold" (bloqueur d'ordre zéro).

#### Matlab

```
% Définition d'un système continu :
sysc=tf(1,[1 1]);
% Obtention de la transformée en z du système échantillonné à Te
Te=0.1;
sysd=c2d(sysc,Te,'zoh');
% Notation en z^-1
sysd.variable='z'
```

### 4.3 Effets de l'échantillonnage

Le convertisseur numérique a deux effets (néfastes) importants : l'effet de retard et l'effet de quantification.

L'effet de retard se voit dans (24) : on peut dire en première approximation que l'échantillonnage se traduit par un retard pur de  $T_e/2$ . Cela a donc, si ce temps

d'échantillonnage est non négligeable par rapport aux constantes de temps du système, un effet déstabilisant.

Pour illustrer l'effet de quantification supposons par exemple que l'on utilise un convertisseur analogique/numérique 8 bits 0-10V. Le plus petit incrément que le convertisseur peut distinguer est de  $10V/2^8 = 40mV$ . Si le signal utile varie physiquement entre 5V et 5,4V on n'aura que dix incréments utilisables... Ce phénomène n'entre pas dans la panoplie des phénomènes modélisables par la transformée en z. En pratique il est donc souvent nécessaire d'étudier le système et de synthétiser la commande numérique en négligeant ce phénomène puis de vérifier a posteriori qu'il n'a pas un effet trop néfaste.

#### 4.4 Où l'on revoit le théorème de Shannon

Le théorème de Shannon est bien connu en ce qui concerne l'échantillonnage de signaux discrets : on sait que le signal ne doit pas comporter de composante de fréquence supérieure à  $f_e/2$  si l'on ne veut pas observer de repliement. On doit aussi prendre des précautions si l'on veut travailler sur la fonction de transfert discrète correspondant à un système continu échantillonné.

On va traiter un exemple concret pour mettre en évidence le problème de repliement dans le cas de la commande des systèmes continus. On s'intéresse à deux systèmes continus du second ordre ayant des pôles de même partie réelle mais de parties imaginaires séparées de  $2\pi T_e$ . Le script Matlab ci-dessous montre la création des deux systèmes :

##### Matlab

```
% Définition de deux systèmes continus :
sys1=zpk([],[-1-2i -1+2i],2^2+1);
sys2=zpk([],[-1-(2+2*pi)*i -1+(2+2*pi)*i],1+(2+2*pi)^2);
```

On voit sur la Figure 11 que la réponse indicielle des deux systèmes est bien différente.

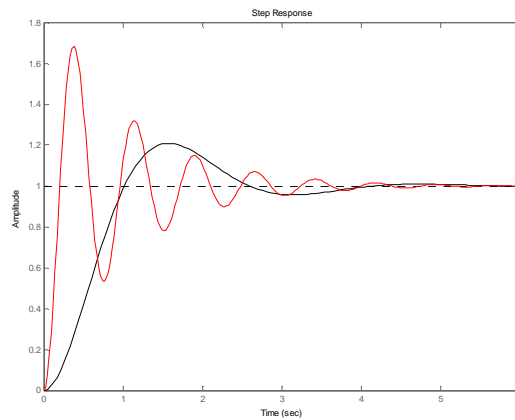


Figure 11 : réponse indicielle de deux systèmes du second ordre

Si l'on calcule la fonction de transfert échantillonnée les deux systèmes à la période de 1 seconde on obtient deux réponses indicielles très proches (représentés par des  $\circ$  sur la Figure 12). Conclusion : la période d'échantillonnage était mal choisie

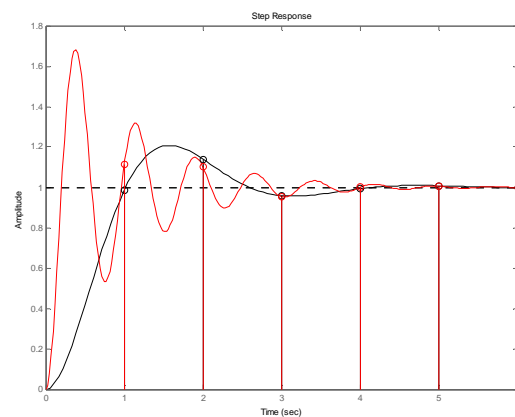


Figure 12 : réponse indicielle de deux systèmes du second ordre échantillonnés à 1s

En pratique la commande échantillonnée de systèmes continus doit être associée à certaines précautions :

- le système ne doit pas comporter de dynamiques propres situées dans les fréquences supérieures à  $f_c/2$  (exemple illustré dans ce chapitre).
- les signaux échantillonnés doivent être "nettoyés" des fréquences supérieures à  $f_c/2$  avec un filtre anti-repliement. (A moins que le signal soit suffisamment propre et ne comporte pas de composantes significatives au-delà de ces fréquences).

### Choix de la période d'échantillonnage

Le choix de la période d'échantillonnage est très important et dépend de la dynamique des signaux. Attention, les systèmes commandés en boucle fermée ont souvent une dynamique plus rapide que le système en boucle ouverte : il faut donc prendre en compte la dynamique en boucle fermée pour le choix de la fréquence d'échantillonnage.

En pratique on préconise de prendre une fréquence 5 à 25 fois supérieure à la bande passante souhaitée en boucle fermée.

On peut être tenté de prendre une fréquence d'échantillonnage très élevée si la technologie le permet (ordinateur et convertisseurs suffisamment rapide). Ceci peut amener rapidement à des problèmes numériques. Imaginons par exemple le cas d'un système ayant une dynamique très lente (quelques secondes) et un système de mesure codant les mesures sur 10 bits (cas classique). La Figure 13 illustre cet exemple. Le signal varie peu entre deux instants d'échantillonnage et les variations du signal sont faibles par rapport à la valeur de l'incrément numérique (figure du haut). Le calcul de la simple dérivée numérique se traduit donc par une succession de  $-1$   $0$  et  $+1$ .

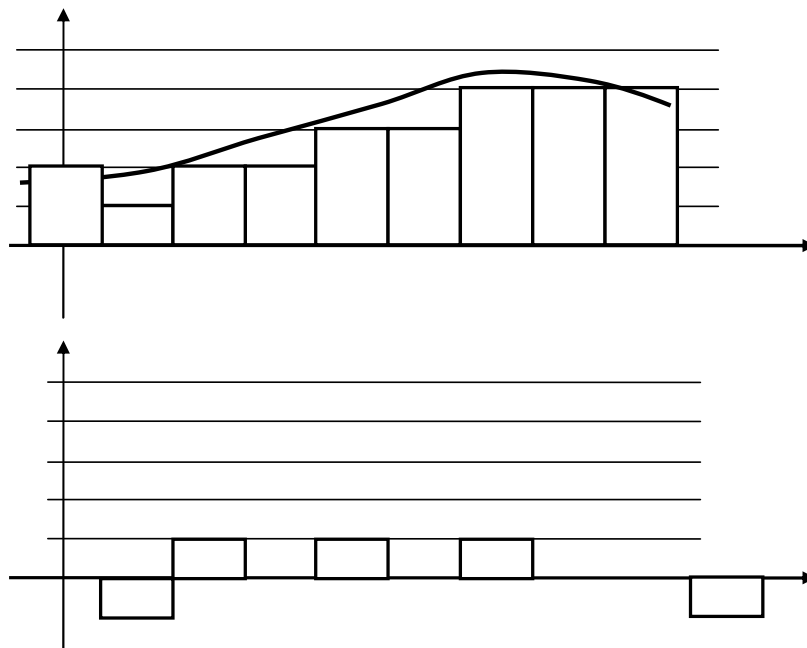


Figure 13 : période d'échantillonnage trop faible. En haut : signal échantillonné et discrétisé ; en bas : calcul de la dérivée numérique.

### Filtre anti-repliement

En toute rigueur chaque fois qu'un signal est échantillonné il faudrait faire précéder le convertisseur numérique d'un filtre analogique anti-repliement éliminant physiquement toutes les composantes du signal de fréquence supérieure à  $f_e/2$ .

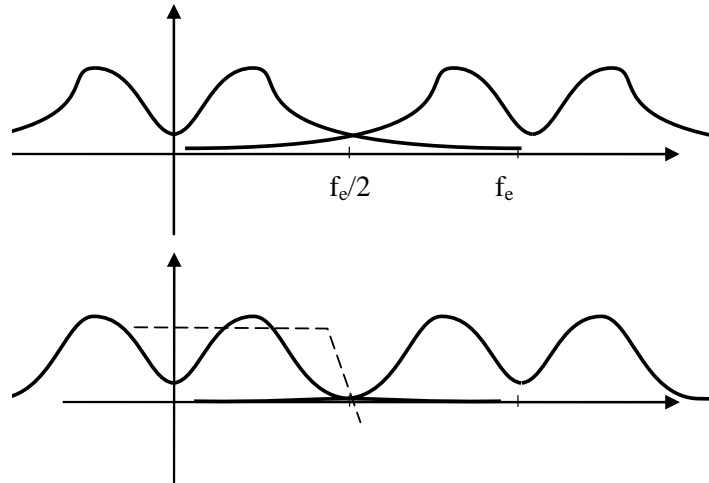


Figure 14 : Effet du filtre anti-repliement

Bien souvent la période d'échantillonnage de l'asservissement est relativement lente (quelques secondes) et il est alors impossible de réaliser un filtre de constante de temps aussi faible. On procède alors à un sur-échantillonnage : le signal est filtré, échantillonné à haute fréquence  $T'_e$  puis filtré numériquement et sous-échantillonné à la période  $T_e$  beaucoup plus grande que  $T'_e$  comme illustré sur la

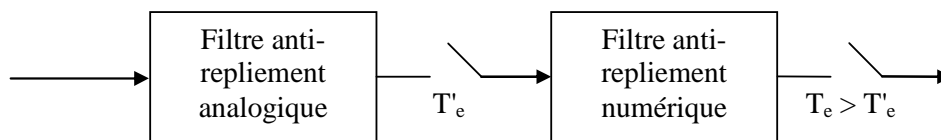


Figure 15 : Sur échantillonnage

Un simple filtre numérique de type moyenne mobile est souvent suffisant.

### 4.5 Discrétisation de fonction de transfert continues

Une manière classique pour réaliser un contrôle par ordinateur est de commencer par synthétiser un correcteur continu avec les outils de l'automatique classique puis de trouver un correcteur discret qui "approche" convenablement le correcteur continu.

#### 4.5.1 Approximation numérique d'équation différentielle

Supposons que le correcteur continu que l'on a synthétisé soit du type PD :

$$\text{PID}(s) = \frac{e(s)}{\varepsilon(s)} = k_p + k_D \cdot s \quad (26)$$

Il s'agit de calculer les commandes  $e(k)$  successives issues du correcteur en fonction des erreurs  $\varepsilon(k)$ . Une approche classique et intuitive est d'approximer la dérivée par une différence :

$$e(k) = k_p \cdot \varepsilon(k) + k_D \cdot \frac{\varepsilon(k) - \varepsilon(k-1)}{T_e} \quad (27)$$

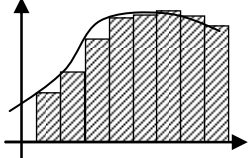
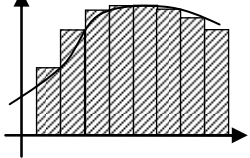
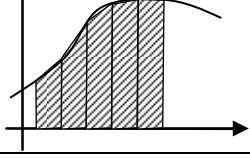
Si on calcule la transformée en  $z$  de  $e(k)$  on obtient :

$$e(z) = k_p \cdot \varepsilon(z) + k_D \cdot \frac{\varepsilon(z) - z^{-1}\varepsilon(z)}{T_e} = \left( k_p + k_D \cdot \frac{1 - z^{-1}}{T_e} \right) \cdot \varepsilon(z) \quad (28)$$

On retrouve la fonction de transfert PD(s) dans laquelle on a effectué le changement de variable  $s = \frac{1-z^{-1}}{T_e}$

De manière plus générale, l'approximation de fonction de transfert revient à effectuer l'approximation numérique d'intégration. Selon la méthode d'intégration choisie on effectue des changements de variable différents.

Tableau 4 : méthodes d'intégration

Méthode	Changement de variable	
Euler progressif (Euler's forward method)	$s = \frac{1 - z^{-1}}{z^{-1} \cdot T_e}$	
Euler rétrograde (Euler's backward method)	$s = \frac{1 - z^{-1}}{T_e}$	
Méthode des trapèzes, ou bilinéaire, ou de Tustin	$s = \frac{2}{T_e} \cdot \frac{1 - z^{-1}}{1 + z^{-1}}$	

En pratique le changement de variable  $s = \frac{1 - z^{-1}}{z^{-1} \cdot T_e}$  n'est pas sans risque. On donne sur la Figure 16 la correspondance entre la zone stable du plan complexe en  $s$  et la zone correspondante après la transformation (Euler progressif, Euler rétrograde, bilinéaire). On voit notamment qu'il est possible, partant d'un système continu stable, d'obtenir des pôles dans le domaine discret en dehors du cercle de stabilité si la transformation appliquée est celle d'Euler progressif.

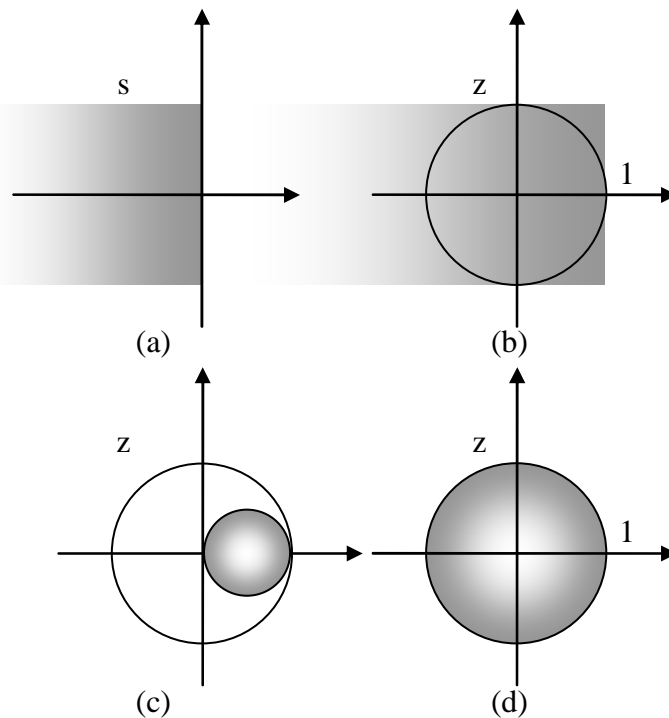


Figure 16 : correspondance des plans complexes. (a) : zone de stabilité dans le plan complexe continu en  $z$  ; (b) zone correspondante avec l'approximation d'Euler progressif ; (c) zone correspondante avec l'approximation d'Euler rétrograde ; (d) zone correspondante avec l'approximation bilinéaire.

La fonction `c2d` de Matlab permet d'obtenir l'équivalent discret d'une fonction de transfert continue avec la méthode d'intégration comme argument :

```

Matlab
sysc=tf([1 0],[1 2 5]);
Te=0.1;
% Méthode d'euler :
sysd1=c2d(sysc,Te,'zoh');
% Méthode des trapèzes :
sysd2=c2d(sysc,Te,'foh');
% Méthode bilinéaire :
sysd3=c2d(sysc,Te,'tustin');
    
```

#### 4.5.2 Superposition des réponses indicielles

Une autre méthode pour approximer une fonction de transfert discrète est de s'assurer que la réponse indicielle de sa version discrétisée correspond exactement lors des temps d'échantillonnage.



### 4.5.3 Correspondances pôles - zéros

La dernière méthode est de partir d'une décomposition de la fonction de transfert continue en pôles et zéros et d'effectuer la conversion pôles à pôles et zéros par zéros. La correspondance, selon que le pôle est réel ou complexe est donnée ci-dessous :

$$\begin{aligned} (s + a) &\rightarrow (z - e^{-a \cdot T_e}) \\ (s + a)^2 + b^2 &\rightarrow z^2 - 2 \cdot e^{-a \cdot T_e} \cdot \cos(b \cdot T_e) \cdot z + e^{-2 \cdot a \cdot T_e} \end{aligned} \quad (29)$$

### 4.5.4 Bilan

Aucune méthode n'est parfaite et on n'aura jamais en même temps :

- même réponse indicielle
- même réponse impulsionnelle
- approximation (Euler, Tustin, etc...) de l'équation différentielle
- correspondance des pôles et des zéros
- correspondance des réponses fréquentielles

Ceci étant dit le choix de la période d'échantillonnage est primordial. Si la fréquence d'échantillonnage est beaucoup plus faible que la dynamique de la fonction de transfert à approximer la différence entre chaque méthode sera minimale.



## 5. IDENTIFICATION DES SYSTEMES ECHANTILLONES

Dans le présent chapitre on donne les bases des méthodes d'identification. Pour une approche complète, voir le cours de troisième année "Identification avion" [6] et les ouvrages [1] et [7].

Même si l'on dispose d'un modèle théorique du processus que l'on souhaite commander il est presque toujours nécessaire de l'affiner en procédant à des essais sur le système réel. Bien souvent le seul modèle dont l'on dispose est ce modèle issu de l'expérimentation. Différentes méthodes s'appliquent selon l'objectif recherché et le niveau de précision exigé. On distinguera :

- les modèles paramétriques
- les modèles simples
- les modèles non paramétriques

### 5.1 Identification d'un modèle ARX par la méthode des moindres carrés récursifs

A partir de cet exemple, le plus simple qui soit, on pose les bases de toutes les méthodes d'identification.

On considère un système d'entrée  $u(t)$  et de sortie  $y(t)$  représenté par l'équation de récurrence suivante :

$$y(t) + a_1 \cdot y(t-1) + a_2 \cdot y(t-2) + \dots + a_n \cdot y(t-n) = b_0 \cdot u(t) + b_1 \cdot u(t-1) + b_2 \cdot u(t-2) + \dots + b_m \cdot u(t-m) \quad (30)$$

L'objectif est, à partir d'essais et de mesures, d'identifier les meilleures valeurs possibles pour les coefficients  $a_i$  et  $b_i$ .

L'équation (30) peut être réécrite pour mettre en évidence la *prédiction* de la sortie  $y(t)$  à partir des mesures passées :

$$y(t) = -a_1 \cdot y(t-1) - a_2 \cdot y(t-2) - \dots - a_n \cdot y(t-n) + b_0 \cdot u(t) + b_1 \cdot u(t-1) + b_2 \cdot u(t-2) + \dots + b_m \cdot u(t-m) \quad (31)$$

Soit :

$$y(t) = \varphi^T(t) \cdot \theta(t)$$

avec :

$$\begin{cases} \varphi^T(t) = [-y(t-1) \ \dots \ -y(t-n) \ , \ u(t-1) \ \dots \ u(t-m)] \\ \theta(t) = [a_1 \ \dots \ a_n \ , \ b_1 \ \dots \ b_m] \end{cases} \quad (32)$$

Le but est de trouver le meilleur ensemble de paramètres  $\theta(t)$  : celui qui, à partir d'un ensemble de mesures  $\varphi(t)$  permet de prédire avec la meilleure précision la sortie effectivement mesurée.

Supposons qu'à la suite d'une expérience on dispose d'un ensemble de mesures  $\{y(t), u(t)\}_{t=1:N}$  et que l'ensemble des paramètres  $\theta$  n'est pas bien connu. On construit la séquence des *sorties prédites* :

$$\hat{y}(t) = \varphi^T(t) \cdot \theta(t) \quad (33)$$

Le paramètre  $\theta$  est celui qui minimise une norme quadratique :

$$\theta = \arg \max \left( \frac{1}{N} \sum_{t=1}^N (y(t) - \hat{y}(t))^2 \right) = \arg \max \left( \frac{1}{N} \sum_{t=1}^N (y(t) - \varphi^T(t) \cdot \theta(t))^2 \right) \quad (34)$$

Compte tenu du fait que l'équation est linéaire au départ et que le terme à minimiser est du type quadratique la solution est simplement obtenue par une pseudo-inverse :

$$\theta = \left[ \sum_{t=1}^N (\varphi(t) \cdot \varphi^T(t)) \right]^{-1} \sum_{t=1}^N (\varphi(t) \cdot y(t)) \quad (35)$$

On obtient donc la solution classique des moindres carrés. La formule (35) est le plus souvent utilisée sous sa forme récursive :

$$\theta(t) = \theta(t-1) + F(t) \cdot \varphi(t) \cdot (y(t) - \theta^T(t-1) \cdot \varphi(t))$$

avec

$$F(t)^{-1} = F(t-1)^{-1} + \varphi(t) \cdot \varphi(t)^T \quad (36)$$

ou

$$F(t) = F(t-1) - \frac{F(t-1) \cdot \varphi(t) \cdot \varphi(t)^T \cdot F(t-1)}{1 + \varphi(t)^T \cdot F(t-1) \cdot \varphi(t)}$$

(On laisse au lecteur le plaisir d'effectuer lui-même les démonstrations, qu'il peut du reste trouver dans [1] et [7]).

## 5.2 L'identification paramétrique en général et en pratique

L'exemple précédent a permis de mettre en évidence tous les aspects de la méthodologie classique de l'identification :

- il a fallu choisir un ensemble de mesures  $\{y(t), u(t)\}_{t=1:N}$ . Le choix du signal d'excitation  $u(t)$  est en pratique primordial,
- il a fallu choisir une structure pour le modèle à identifier (ici les ordres  $n$  et  $m$ ),
- il a fallu choisir un critère (ici un critère quadratique sur l'erreur entre la sortie et la sortie prédite),
- il a fallu choisir un algorithme et le paramétrer (ici l'algorithme des moindres carrés récursifs qu'il a fallu initialiser)

Enfin un point non traité dans le chapitre précédent concerne la validation :

- il aurait fallu déterminer une méthode pour comparer différents modèles et choisir le meilleur

### 5.2.1 Choix du signal d'excitation, traitement des mesures

De manière générale le système que l'on doit identifier doit être excité dans tout le domaine fréquentiel qui nous intéresse. Le signal classiquement utilisé est une séquence pseudo binaire aléatoire (Pseudo Random Binary Signal) tel celui représenté sur la Figure 17.

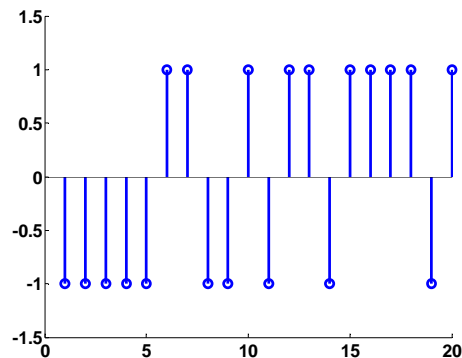


Figure 17 : exemple d'une séquence PRBS

Le signal d'excitation doit se superposer au signal de commande correspondant au point de fonctionnement étudié. Son amplitude doit être faible par rapport à l'amplitude des commandes du système (quelques %).

Une fois le système excité et les mesures effectuées il faut supprimer les valeurs moyennes des mesures.

### 5.2.2 Choix d'une structure de modèle

Le modèle étudié dans le chapitre précédent est de type ARX. On peut le noter sous forme polynomiale :

$$A \cdot y = B \cdot u \quad (37)$$

Ce modèle correspond en fait à un parti-pris sur la manière dont on traite les perturbations. En réalité on prend en compte un signal d'erreurs possibles  $e$  (de moyenne nulle) tels que :

$$A \cdot y = B \cdot u + e \quad (38)$$

Si l'on fait cette hypothèse de modélisation alors la méthode optimale est celle des moindres carrés. Malheureusement cette manière de prendre en compte les perturbations est bien restrictive. D'autres modélisations sont possibles, adaptées à d'autres algorithmes, par exemple le modèle OE :

$$y = \frac{B}{F} \cdot u + e \quad (39)$$

Il est aussi possible de "structurer" le modèle de bruit comme par exemple dans les modèles ARMAX ou BJ :

$$\begin{aligned} A \cdot y &= B \cdot u + C \cdot e \\ y &= \frac{B}{F} \cdot u + \frac{C}{D} \cdot e \end{aligned} \quad (40)$$

La structure du modèle étant choisie il faut choisir l'ordre de ses différents polynômes.

### 5.2.3 Choix d'un critère à minimiser

Dans la plupart des cas le critère à minimiser est l'erreur quadratique entre la sortie et la sortie prédite à un coup (one step ahead prediction). Ceci n'est bien évidemment pas restrictif et peut être généralisé à la prédiction à  $N$  coups.

### 5.2.4 Choix d'un algorithme

Même si chaque structure de modèle est généralement associée à un algorithme "optimal" celui-ci doit généralement se paramétrer.

Prenons par exemple l'algorithme des moindres carrés récursifs tel que présenté dans (36). Cette équation peut être modifiée en intégrant un "facteur d'oubli"  $\lambda$  :

$$\begin{aligned} \theta(t) &= \theta(t-1) + F(t) \cdot \varphi(t) \cdot (y(t) - \theta^T(t-1) \cdot \varphi(t)) \\ \text{avec} & \\ F(t)^{-1} &= \lambda \cdot F(t-1)^{-1} + \varphi(t) \cdot \varphi(t)^T \end{aligned} \quad (41)$$

Ce paramètre (typiquement entre 0.95 et 0.99) permet, comme son nom l'indique, "d'oublier" les anciennes mesures et de se concentrer sur les mesures les plus récentes.

### 5.2.5 Validation

Une fois l'algorithme d'identification appliqué et le modèle obtenu (ce qui prend généralement quelques secondes avec un ordinateur moderne) il reste à expertiser celui-ci. Selon la "qualité" du modèle obtenu on remet en cause l'une ou l'autre des étapes (signal d'excitation, structure du modèle, algorithme) et on réitère jusqu'à obtenir un modèle satisfaisant.

En pratique on essaie un certain nombre de modèles avant de choisir le "meilleur".

Le premier critère de qualité est la capacité du modèle à prédire effectivement la sortie (sortie prédite à un coup, sortie prédite à N coups). Il faut pour cela utiliser un jeu de mesures non utilisé pour l'identification. (En pratique le jeu de mesures initiales est coupé en deux : une partie pour l'identification et une partie pour la validation).

Le second critère de qualité est l'analyse du résidu : la composition spectrale de l'erreur de prédiction.

### 5.2.6 Utilisation de logiciels

En pratique il est commode d'utiliser un célèbre outil de calcul scientifique et sa toolboxe d'identification.

Comme les calculs proprement dit ne prennent que quelques secondes il est facile de réaliser un grand nombre d'identifications et de stocker les différents modèles pour les comparer ensuite. On est rapidement noyé sous le grand nombre de modèles possibles...

## 5.3 Identification "simple"

Il est souvent suffisant d'obtenir un modèle "simple" de type premier ordre et un simple essai indiciel convient souvent. Le système est excité avec un échelon unité et on cherche le meilleur modèle du premier ordre correspondant.

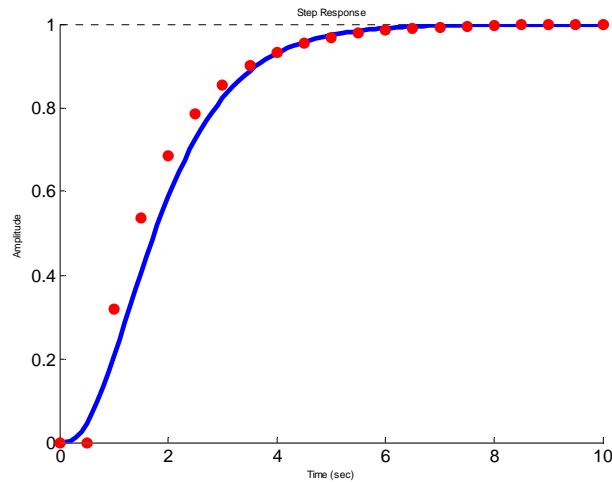


Figure 18 : identification sur un essai indiciel

Dans l'exemple de la Figure 18 on superpose le résultat d'une mesure (en bleu) avec la simulation d'un modèle discret avec une période de 0.5s. Le modèle cherché est du premier ordre avec un retard pur  $d$  :

$$y(z) = \frac{(1 + c_1) \cdot z^{-d}}{1 + c_1 \cdot z^{-1}} \cdot u(z) \quad (42)$$

Il s'agit de trouver l'entier  $d$  et le nombre réel  $c_1$  tels que la réponse indicielle du modèle corresponde pour le mieux aux mesures issues de l'essai. Ceci peut se formaliser par exemple de la manière suivante :

$$(c_1 \quad d_1) = \arg \min \sum_{t=1}^N (y_{\text{modèle}}(t) - y_{\text{mesure}}(t))^2 \quad (43)$$

Cette optimisation peut se faire à la main ou grâce à une procédure automatisée (avec le célèbre outil de calcul scientifique).

Dans le cas où le système comprend naturellement une intégration (par exemple la commande d'un moteur à courant continu par une tension et la mesure de l'angle de rotation) il est utile de prendre en compte explicitement cette intégration dans le modèle :

$$y(z) = \frac{1}{1 - z^{-1}} \cdot \frac{(1 + c_1) \cdot z^{-d}}{1 + c_1 \cdot z^{-1}} \cdot u(z) \quad (44)$$



La procédure est la même mais au lieu de prendre en compte la sortie  $y(t)$  on prend en compte la dérivée  $y'(t)$  :

$$y'(t) = y(t) - y(t-1) \quad (45)$$

## 5.4 Identification non paramétrique

Pour terminer ce panorama rapide sur les méthodes d'identification il faut parler des méthodes non paramétriques : celles qui permettent de tracer la réponse impulsionnelle ou la réponse fréquentielle du système.

### 5.4.1 Estimation de la réponse impulsionnelle

La réponse impulsionnelle d'un système peut être obtenue simplement en appliquant une entrée de type "impulsion" au système et d'en mesurer les conséquences.

La seconde méthode consiste à utiliser la méthode des moindres carrés (ou une méthode dérivée) comme vue au chapitre 5.1. La réponse impulsionnelle d'un système peut en effet être vue comme la fonction de transfert particulière où le dénominateur est de degré nul et le numérateur comporte un nombre infini (en pratique : grand) d'éléments ( $n = 0$  et  $m$  grand) :

$$y(t) = b_0 \cdot u(t) + b_1 \cdot u(t-1) + b_2 \cdot u(t-2) + \dots + b_m \cdot u(t-m) \quad (46)$$

La dernière méthode, qui sera développée ici est l'analyse d'intercorrélation.

Par définition l'intercorrélation  $R_{yu}$  entre deux signaux est :

$$R_{yu}(\tau) = E[(y(t) - E(y(t))) \cdot (u(t-\tau) - E(u(t-\tau)))] \quad (47)$$

Dans cette formule l'espérance mathématique peut être approchée grâce à la loi des grands nombres :

$$R_{yu}(\tau) \approx R_{yu}^N(\tau) = \lim_{N \rightarrow \infty} \frac{1}{N} \sum_{t=1}^N [(y(t) - m_y) \cdot (u(t-\tau) - m_u)] \quad (48)$$

On s'intéresse maintenant à un système tel que décrit par l'équation impulsionnelle (46) et on suppose que le signal d'entrée est un bruit blanc. Le signal d'entrée vérifie donc :

$$\begin{cases} \tau \neq 0 & R_{uu}(\tau) = 0 \\ \tau = 0 & R_{uu}(\tau) = R \end{cases} \quad (49)$$

Il vient alors naturellement :

$$b_k \approx \frac{1}{R} R_{yu}(k) \quad (50)$$

### 5.4.2 Estimation de la réponse fréquentielle

On utilise ici le spectre d'intercorrélation. Par définition le spectre d'intercorrélation entre deux signaux est :

$$\Phi_{yu}(\omega) = \sum_{\tau=-\infty}^{\infty} R_{yu}(\tau) \cdot e^{-i\omega\tau} \quad (51)$$

On suppose que le signal d'entrée  $u(t)$  et le signal de sortie  $y(t)$  sont reliés par une fonction de transfert linéaire du type :

$$y(t) = F \cdot u(t) + v(t) \quad (52)$$

où  $v(t)$  est une perturbation indépendante de  $u(t)$ . On a alors les relations :

$$\Phi_{yu}(\omega) = F(j\omega) \cdot \Phi_{uu}(\omega) \quad (53)$$

Il reste à estimer les spectres d'intercorrélation pour remonter à la réponse fréquentielle  $F(j\omega)$  recherchée. Une idée naturelle est d'utiliser la relation (48) mais cela marche très mal dans presque tous les cas, même en choisissant  $N$  assez grand. L'amélioration est d'utiliser une "fenêtre" pour favoriser les valeurs faibles de  $\tau$  par rapport aux valeurs élevées. La solution classique utilisant la *fenêtre de Hamming* donne :

$$\Phi_{yu}^N(\omega) = \sum_{\tau=-\gamma}^{\gamma} R_{yu}^N(\tau) \cdot w_{\gamma}(\tau) \cdot e^{-i\omega\tau} \quad (54)$$

avec

$$\begin{cases} w_{\gamma}(\tau) = \frac{1}{2} \left( 1 + \cos\left(\frac{\pi \cdot \tau}{\gamma}\right) \right) & |\tau| < \gamma \\ w_{\gamma}(\tau) = 0 & |\tau| > \gamma \end{cases}$$

On utilise ensuite la relation (51) pour estimer la réponse fréquentielle :

$$F(j \cdot \omega) = \frac{\Phi_u^N(\omega)}{\Phi_{yu}^N(\omega)} \quad (55)$$



## 6. SYSTEMES BOUCLES

### 6.1 Fonction de transfert en boucle fermée

Un système et sa commande par ordinateur peut être représenté par un ensemble de fonctions de transfert formant un schéma-bloc plus ou moins compliqué (Figure 19).

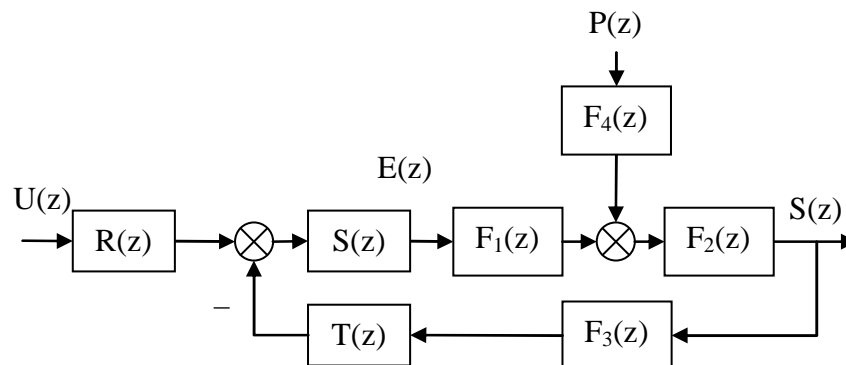


Figure 19 : système discret commandé par ordinateur. Le correcteur est représenté par les fonctions R,S,T, le système par  $F_1$  et  $F_2$ , le modèle de perturbations par  $F_4$  et le modèle des actionneurs par  $F_4$ .

Quelle que soit la structure d'un système asservi il est possible de se ramener à une forme canonique comprenant une fonction de transfert en boucle ouverte en combinant judicieusement les fonctions de transfert du procédé, du régulateur et des perturbations. La Figure 20 représente par exemple le système de la Figure 19 où  $F$  est une combinaison des fonctions de transfert du système, du régulateur, des perturbations et des capteurs.

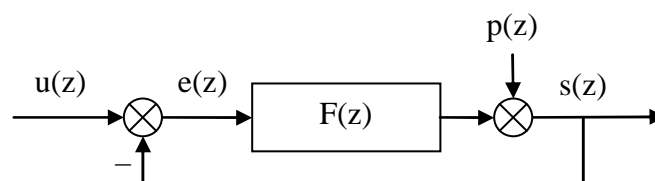


Figure 20 : système discret, notations.

Par défaut on considère une contre-réaction négative et une perturbation additive sur la sortie. La fonction de transfert  $F$  est appelée *fonction de transfert en boucle ouverte*. Comme pour les systèmes continus la *fonction de transfert en boucle fermée* est obtenue simplement à partir de  $F$  (58).

$$F(z) = \frac{s(z)}{e(z)} = \frac{B(z^{-1})}{A(z^{-1})} \quad (56)$$

$$G(z) = \frac{s(z)}{u(z)} = \frac{B(z^{-1})}{A(z^{-1}) + B(z^{-1})}$$

## 6.2 Erreur statique

L'erreur statique du système en boucle fermée, comme pour le système en boucle ouverte, est simplement le rapport des coefficients.

$$G_0 = \frac{\sum_{i=0}^{i=m} b_i}{1 + \sum_{i=0}^{i=m} b_i + \sum_{i=1}^{i=n} a_i} \quad (57)$$

Si l'on souhaite que l'asservissement confère un gain statique unitaire au système en boucle fermée (c'est-à-dire que l'on force la sortie  $s(k)$  à tendre vers la valeur de consigne  $u(k)$ ) il faut que ce rapport de coefficients soit égal à un. On montre facilement que cela est obtenu si la fonction de transfert  $F(z)$  comporte un intégrateur :

$$F(z) = \frac{s(z)}{e(z)} = \frac{1}{1 - z^{-1}} \cdot \frac{B(z)}{A'(z)} \quad (58)$$

## 6.3 Rejet de perturbations

La fonction de transfert entre la perturbation  $p(z)$  et la sortie  $y(z)$  est appelée fonction de sensibilité perturbation-sortie, notée  $S_{yp}$ .

$$S_{yp}(z) = \frac{s(z)}{p(z)} = \frac{A(z^{-1})}{A(z^{-1}) + B(z^{-1})} \quad (59)$$

Si l'on souhaite qu'une perturbation de type échelon soit parfaitement rejetée il faut que  $A(1) = 0$ , on retrouve donc la condition (58) à savoir que le système doit comporter une intégration dans la boucle directe. On peut être en outre être amené à spécifier que les perturbations ne soient pas amplifiées dans une certaine bande de fréquence. On rajoute donc une contrainte du type  $|S_{yp}(e^{j2\pi f})| < S_{max}$  pour  $f$  appartenant à un certain domaine de  $[0 \ 0.5]$ .

## 6.4 Stabilité des systèmes à simple gain de rétroaction

Les critères de Routh et de Jury (donnés dans le chapitre 3.2.7 p13) ou le calcul numérique des pôles permettent de déterminer la stabilité du système en boucle fermée si l'on connaît sa fonction de transfert.

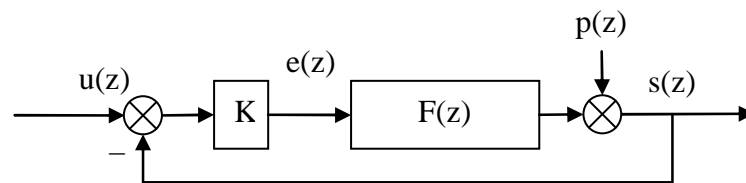


Figure 21 : système bouclé par un simple gain.

On s'intéresse ici au cas particulier où l'asservissement est réalisé par un simple gain de rétroaction  $K$  (Figure 21). On cherche la valeur de  $K$  qui amène le système en limite de stabilité. L'outil adapté ici est le lieu d'Evans ou lieu des racines (root locus) qui permet de visualiser l'évolution des pôles du système bouclé lorsque le gain  $K$  varie.

Prenons l'exemple d'un système comportant trois pôles et deux zéros. L'exemple est donné sous forme de script Matlab :

### Matlab

```
% Définition d'un système discret :
sys=zpk([-2.037 -0.1406],[0.6065 0.3679 0.3679],1,1)
%Tracé du lieu des racines
rlocus(sys);
```

Le script précédent produit le tracé de la Figure 22. On pourrait y lire que la limite de stabilité est obtenue pour  $K=0.2$  environ (détail de la Figure 23).

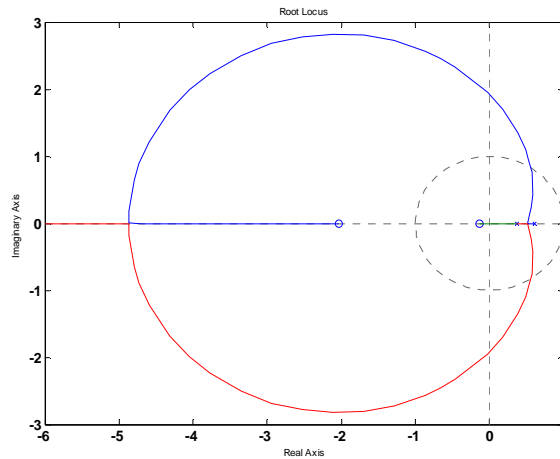


Figure 22 : lieu des racines

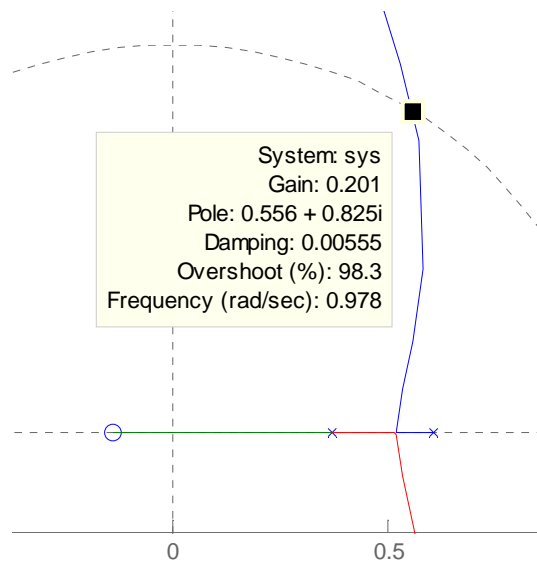


Figure 23 : lieu des racines : détail de la limite de stabilité

On voit sur la Figure 22 que le lieu des racines d'un système échantillonné ressemble à celui d'un système continu :

- on commence par tracer les pôles (×) et les zéros (o) du système en boucle ouverte.
- les pôles du système en boucle fermée décrivent un lieu qui part des pôles (pour  $K = 0$ ) et arrive aux zéros (pour  $K = \infty$ )
- lorsque il y a  $n-m$  plus de pôles que de zéros le système comporte  $n-m$  branches asymptotiques.

On ne décrira pas ici les diverses propriétés du lieu des racines qui permettent de le tracer à la main puisque il est facile à obtenir avec un ordinateur.



Il peut aussi être utilisé pour régler "à la main" le gain permettant d'obtenir un ou plusieurs modes dominants ayant certaines propriétés de fréquence et d'amortissement. On voit sur la Figure 24 que le gain 0.017 permet d'obtenir un pôle double d'amortissement 0.706 et de fréquence 0.65 rad/s et un pôle simple de fréquence 1.63 rad/s.

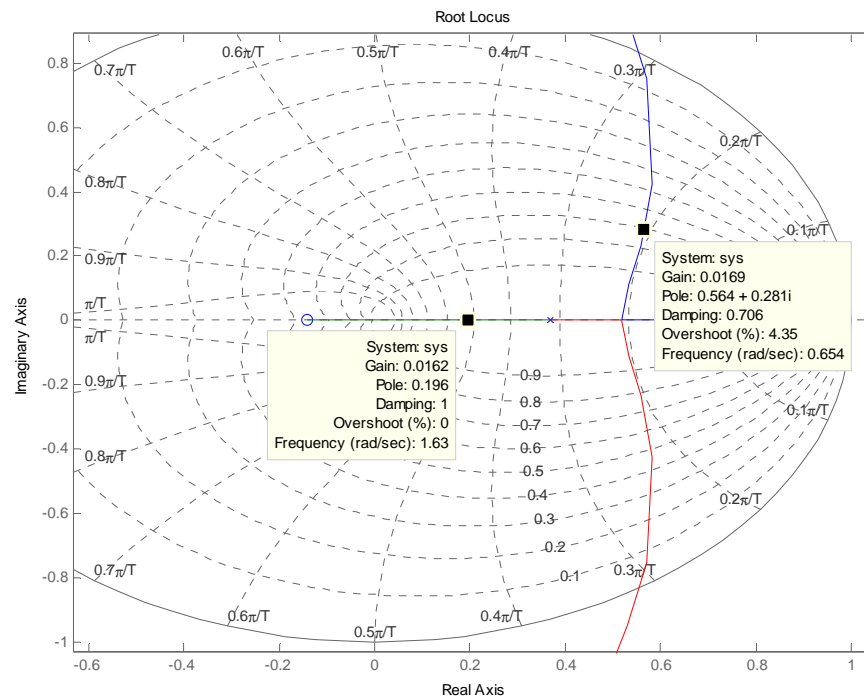


Figure 24 : utilisation du lieu des racines pour régler le gain de la boucle fermée

L'outil rltool de Matlab permet de nombreuses manipulations graphiques de ce type.

## 6.5 Analyse des systèmes bouclés dans le domaine fréquentiel : stabilité

Comme dans le cas continu il est possible de prévoir des propriétés, en particulier la stabilité, d'un système en boucle fermée à partir de l'examen de la fonction de transfert en boucle ouverte.

On ne redémontrera pas ici le théorème de Nyquist mais on se contente de rappeler l'énoncé.

On rappelle que le *lieu de Nyquist* est le lieu de  $F(j2\pi f)$  dans le plan complexe pour  $f$  variant de 0 à  $+\infty$ . A noter que le lieu pour la zone  $[0, 0.5]$  est symétrique de celui pour la zone  $[0.5, 1]$ . Pour cette raison on ne représente généralement que le lieu pour  $f$  appartenant à  $[0, 0.5]$ .

### 6.5.1 Théorème de Nyquist dans le cas où le système est stable en boucle ouverte (critère du revers)

Le système est stable en boucle fermée si le lieu de Nyquist de la fonction de transfert en boucle ouverte parcouru dans le sens des fréquences croissantes de 0 à 0.5 laisse le point critique  $(-1,0)$  à gauche.

La Figure 25 représente par exemple le lieu de Nyquist d'une fonction de transfert en boucle ouverte qui sera stable en boucle fermée.

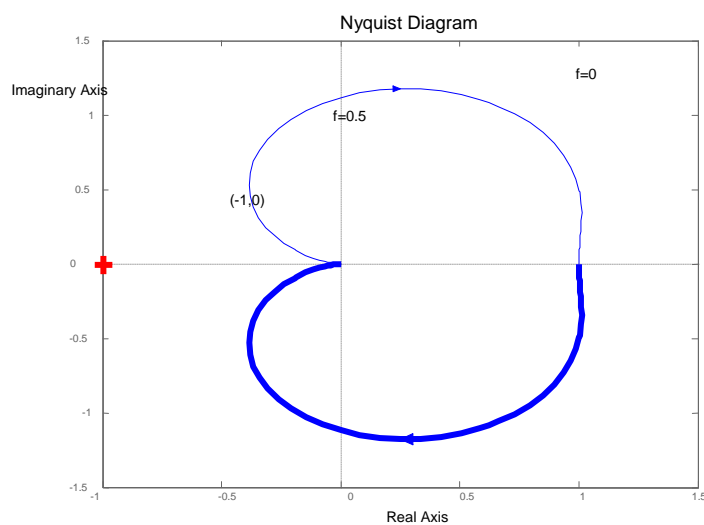


Figure 25 : lieu de Nyquist d'un système stable

### 6.5.2 Théorème de Nyquist dans le cas où le système est instable en boucle ouverte

Si le système en boucle ouverte comporte  $N$  pôles instables il sera stable en boucle fermée si le lieu de Nyquist de la fonction de transfert en boucle ouverte parcouru dans le sens des fréquences croissantes de  $-0.5$  à  $0.5$  entoure le point critique  $(-1,0)$   $N$  fois dans le sens des aiguilles d'une montre..

### 6.5.3 Interprétations du lieu de Nyquist

Comme dans le cas continu, la distance à laquelle on passe du point critique renseigne sur la robustesse du système bouclé. La distance sera exprimée de différente manière (Figure 26) :

- marge de module  $\Delta M$  : plus petite distance entre le lieu des racines et le point critique,

- marge de gain  $\Delta G$
- marge de phase  $\Delta P$

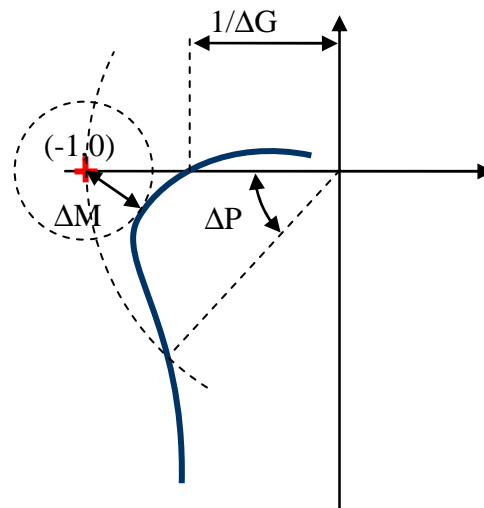


Figure 26 : Marges de stabilité

Les valeurs typiques de ces marges sont données ci-après :

- $\Delta G \geq 6\text{dB}$
- $\Delta M \geq -6\text{dB}$
- $30^\circ \leq \Delta P \leq 60^\circ$

La distance entre le point critique et le lieu des racines est la norme de  $1+F(j2\pi f)$ . On peut donc exprimer la marge de module en fonction de la fonction de sensibilité :

$$\begin{aligned} \Delta M &= \min |1 + F(j \cdot \omega)|_{\omega \in [0, \pi]} \\ \Delta M &= \min \left| 1 + \frac{B(j \cdot \omega)}{A(j \cdot \omega)} \right|_{\omega \in [0, \pi]} = \max \left| \frac{A(j \cdot \omega)}{A(j \cdot \omega) + B(j \cdot \omega)} \right|_{\omega \in [0, \pi]} \quad (60) \\ \Delta M &= \max |S_{yp}(j \cdot \omega)|_{\omega \in [0, \pi]}^{-1} \\ \Delta M_{\text{dB}} &= -\max |S_{yp}(j \cdot \omega)|_{\text{dB}} \end{aligned}$$

Si l'on trace l'évolution de  $S_{yp}$  en fonction de  $f$  on obtient donc la valeur de la marge de module.



# 7. METHODES DE REGULATION DES SYSTEMES ECHANTILLONES

## 7.1 Introduction

Les bases théoriques permettant de concevoir le correcteur pour un système échantillonné ont été posées dans les chapitres précédents. Ce chapitre fait le bilan des connaissances acquises et donne quelques solutions pour calculer des correcteurs à partir d'un cahier des charges précis.

Les systèmes que l'on cherche à commander sont généralement, par nature, continus. La réalisation d'un correcteur par un ordinateur implique que l'on génère une équation de commande discrète (généralement une équation récurrente). L'obtention de cette équation de commande peut se faire de deux manières :

- la première méthode consiste à d'abord synthétiser un correcteur continu (en utilisant les outils classiques de l'automatique et la simulation) et discrétiser ensuite ce correcteur. C'est l'objet du chapitre 7.2.2.
- la seconde méthode consiste à tenir la représentation discrète du système continu associé à son interface numérique (convertisseurs analogique-numérique et numérique-analogique) comme expliqué dans le chapitre 4.5 puis synthétiser directement le correcteur discret. Cela sera expliqué dans le chapitre 7.3 et suivants.

Ce choix peut être résumé par la figure suivante :

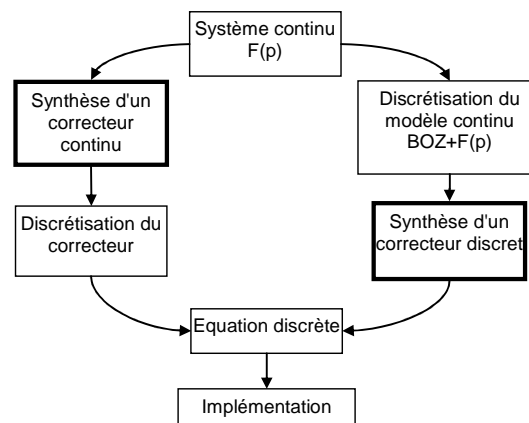


Figure 27 : Réalisation d'un correcteur discret.

Au-delà des affaires de goût, le choix de la méthode sera souvent dicté par l'application. En effet, la première méthode est d'autant meilleure que le correcteur discrétisé approche correctement le correcteur continu. Cela implique que le temps d'échantillonnage soit très faible par rapport aux constantes de temps du système

bouclé. Dans le cas où les contraintes temps réel sont fortes (capacités de calcul partagées par exemple) il sera préférable de synthétiser un correcteur discret.

## 7.2 Structure d'un système commandé par ordinateur, hypothèses

Par la suite on se ramènera toujours à un système tel que celui représenté sur la Figure 28 en cherchant un correcteur  $K(z)$  linéaire. On montre que quel que soit le correcteur linéaire on peut se ramener à la forme représenté sur la Figure 29 (forme RST).

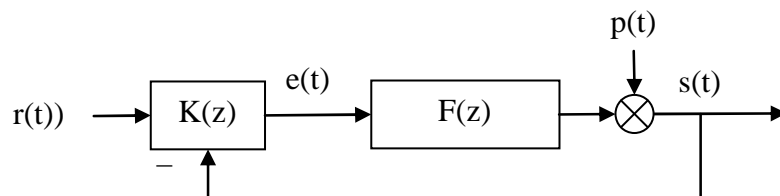


Figure 28 : système bouclé, forme générale.

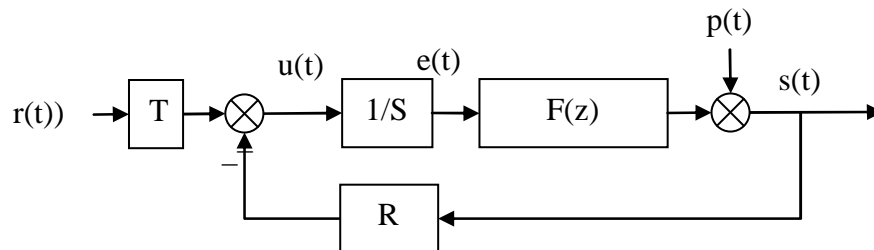


Figure 29 : système bouclé, forme canonique.

### 7.2.1 Discrétisation, choix de la période d'échantillonnage

Du point de vue du ordinateur, l'extérieur (le procédé) est représenté par la fonction de transfert  $F(z)$ . L'hypothèse importante est que la commande (sortie du ordinateur) et la mesure (entrée du ordinateur) sont échantillonnées de manière synchrone à la même période  $T_e$ .

Le choix judicieux de la fréquence d'échantillonnage est très important. En pratique on choisit cette fréquence en fonction de la bande passante du système en **boucle fermée** :

$$f_e = (6 \text{ à } 25) f^{BF} \quad (61)$$

### 7.2.2 Discrétisation d'un correcteur continu

La première méthode pour obtenir les équations d'un correcteur numérique est de faire sa synthèse avec les outils de l'automatique continue puis de discrétiser la fonction de transfert du correcteur continu obtenu (voir pour cela le chapitre 4.2 *Echantillonnage des systèmes continus*).

On va voir par la suite que la synthèse directe d'un correcteur discret offre plus de degré de liberté et permet en général de remplir un cahier des charges plus complet.

## 7.3 Régulateur PID numérique

Le régulateur le plus populaire est le régulateur PID. Nous allons voir dans un premier temps comment on génère un régulateur PID à partir des équations bien connues du régulateur PID continu et comment ces équations peuvent être généralisées pour obtenir plus de degrés de liberté pour le réglage.

### 7.3.1 Cas 1

Un régulateur PID continu a une fonction de transfert du type :

$$K_{\text{PID}}(s) = K \cdot \left( 1 + \frac{1}{T_i \cdot s} + \frac{T_d \cdot s}{1 + \frac{T_d}{N} \cdot s} \right) \quad (62)$$

La discrétisation de ce correcteur continu par la méthode d'Euler (Tableau 4 page 31) se fait en remplaçant la variable de Laplace  $s$  par le terme  $\frac{1-q^{-1}}{q^{-1} \cdot T_e}$ . On obtient :

$$K_{\text{PID}}(q) = K \cdot \left( 1 + \frac{1}{T_i \cdot \frac{1-q^{-1}}{T_e}} + \frac{T_d \cdot \frac{1-q^{-1}}{T_e}}{1 + \frac{T_d}{N} \cdot \frac{1-q^{-1}}{T_e}} \right) \quad (63)$$

En réarrangeant on obtient la forme générale RST (Figure 30):

$$K_{PID}(q) = K \cdot \frac{r_0 + r_1 \cdot q^{-1} + r_2 \cdot q^{-2}}{(1 - q^{-1}) \cdot (1 + s_1' \cdot q^{-1})}$$

$$\begin{cases} r_0 = K \cdot \left( 1 + \frac{T_e}{T_i} + \frac{N \cdot T_d}{T_d + N \cdot T_e} \right) \\ r_1 = -K \cdot \left( 1 + \frac{T_d}{T_d + N \cdot T_e} + \frac{T_e}{T_i} \cdot \frac{T_d}{T_d + N \cdot T_e} + \frac{N \cdot T_d}{T_d + N \cdot T_e} \right) \\ r_2 = K \cdot \left( \frac{T_d}{T_d + N \cdot T_e} \right) \\ s_1' = \frac{T_d}{T_d + N \cdot T_e} \end{cases} \quad (64)$$

Comme on le voit le correcteur obtenu comporte autant de degrés de liberté que le correcteur PID analogique. On reconnaît au dénominateur le terme  $(1 - q^{-1})$  en facteur qui assure l'effet d'intégration.

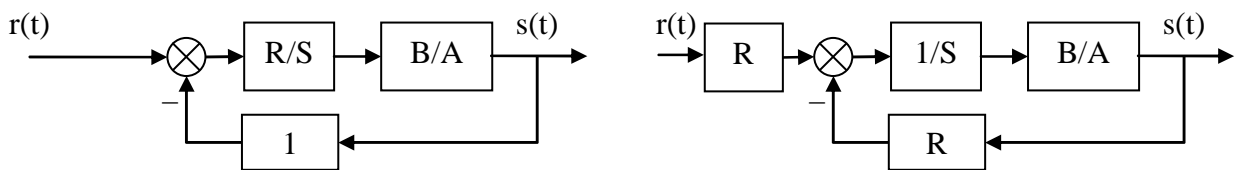


Figure 30 : PID sous forme RST

La fonction de transfert en boucle fermée obtenue est donnée par l'équation suivante :

$$H_{BF} = \frac{B \cdot R}{A \cdot S + B \cdot R} = \frac{B \cdot R}{P} \quad (65)$$

De manière générale le polynôme caractéristique  $P$  fixe les performances visées en boucle fermée. Le calcul de  $R$  et  $S$ ,  $P$  étant fixé et  $A$  et  $B$  étant connus, est réalisé en résolvant l'équation de Bézout  $P = A \cdot S + B \cdot R$ . (voir 9.1 Annexe 1 : Résolution de l'équation polynomiale de Bezout)

On voit que le numérateur  $B \cdot R$  est imposé une fois que  $R$  et  $S$  sont trouvés. Le régulateur introduit donc naturellement des zéros dans la boucle fermée.

Traisons l'exemple entièrement dans le cas où le système est modélisé comme un système du deuxième ordre avec faible retard. La fonction de transfert du procédé est donnée par :



$$H_{BO} = \frac{B}{A} = \frac{b_1 \cdot q^{-1} + b_2 \cdot q^{-2}}{1 + a_1 \cdot q^{-1} + a_2 \cdot q^{-2}} \quad (66)$$

Le correcteur PID rajoutant deux pôles, le polynôme caractéristique en boucle fermée spécifié doit posséder quatre pôles. Le polynôme caractéristique en boucle fermée doit donc être de la forme :

$$\begin{aligned} P &= 1 + p_1 \cdot q^{-1} + p_2 \cdot q^{-2} + p_2 \cdot q^{-3} + p_2 \cdot q^{-4} \\ P &= (1 + p_1' \cdot q^{-1} + p_2' \cdot q^{-2}) \cdot (1 + \alpha_1 \cdot q^{-1}) \cdot (1 + \alpha_2 \cdot q^{-1}) \end{aligned} \quad (67)$$

Et il doit être identifié au produit  $AS + BR$  avec  $R$  et  $S$  ayant les formes données par (64) :

$$\begin{aligned} &(1 + p_1' \cdot q^{-1} + p_2' \cdot q^{-2}) \cdot (1 + \alpha_1 \cdot q^{-1}) \cdot (1 + \alpha_2 \cdot q^{-1}) \\ &= (1 + a_1 \cdot q^{-1} + a_2 \cdot q^{-2}) \cdot (1 - q^{-1}) \cdot (1 + s_1' \cdot q^{-1}) \\ &+ (b_1 \cdot q^{-1} + b_2 \cdot q^{-2}) \cdot (r_0 + r_1 \cdot q^{-1} + r_2 \cdot q^{-2}) \end{aligned} \quad (68)$$

Les coefficients  $p_1'$  et  $p_2'$  définissent un modèle du second ordre aux caractéristiques (pulsation propre  $\omega_0$  et amortissement  $\sigma$  choisis et coefficients déduits conformément au Tableau 3 page 25). Les coefficients  $\alpha_1$  et  $\alpha_2$  correspondent à des pôles auxiliaires aperiodiques. Les choix typiques pour ces coefficients sont :

$$-0.05 < \alpha_1, \alpha_2 < -0.5 \quad (69)$$

On montre que la résolution de (68) est réalisée par une inversion matricielle simple :

$$M \cdot x = p \quad (70)$$

avec

$$\begin{cases} x^T = [1 & s_1' & r_0 & r_1 & r_2] \\ p^T = [1 & p_1 & p_2 & p_3 & p_4] \end{cases}$$

et

$$M = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ a_1' & 1 & b_1 & 0 & 0 \\ a_2' & a_1' & b_2 & b_1 & 0 \\ a_3' & a_2' & 0 & b_2 & b_1 \\ 0 & a_3' & 0 & 0 & b_2 \end{pmatrix} \quad (71)$$

On montre que M est inversible à la condition que A et B soient premiers entre eux, c'est-à-dire que A et B n'aient pas de pôles et de zéros en commun.

(Le problème peut aussi être résolu en utilisant la résolution de l'équation de Bézout).

### 7.3.2 Cas 2

Comme on l'a vu le correcteur PI numérique directement transposé du correcteur PID analogique décrit dans le chapitre 7.3.1 a l'inconvénient de ramener des zéros au dénominateur de la fonction de transfert en boucle fermée (équation 65). Il est possible, grâce à la flexibilité du régulateur numérique d'imposer au correcteur de ne pas ajouter de zéros en boucle fermée. La fonction de transfert désirée en boucle fermée sera donc de la forme :

$$H_{BF} = \frac{P(1)}{B(1)} \cdot \frac{B}{P} \quad (72)$$

Le facteur  $P(1)/B(1)$  garantit le gain statique unitaire du système en boucle fermée (voir page 12). Si le régulateur a la forme générale RST alors les polynômes R S et T doivent vérifier :

$$H_{BF} = \frac{B \cdot T}{A \cdot S + B \cdot R} = \frac{P(1)}{B(1)} \frac{B}{P} \quad (73)$$

S et R sont obtenus par la résolution de la même équation que précédemment (68). Le polynôme T est obtenu par simple identification :  $T = R(1)$ .

La structure, sous forme de schéma-bloc est donnée ci-dessous :

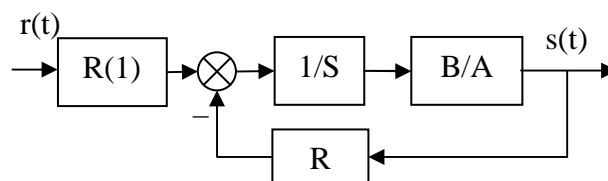


Figure 31 : PID2 sous forme RST

On montre que cette structure de PID, ramené en continu, correspond au PID canonique pour l'action intégrale. L'action proportionnelle et dérivée n'est appliquée que sur le signal y.

La structure du PID2 par rapport au PID1 a pour avantage de ne pas ramener de zéros à la boucle fermée. Cela se traduit par un dépassement moindre en transitoire lorsqu'on applique un échelon de commande. Par contre les deux systèmes réagissent de la même manière à un échelon de perturbation.

## 7.4 Placement de pôles

Les deux structures de PID décrites dans le chapitre précédent sont deux cas particulier d'une stratégie plus générale de placement de pôles.

La synthèse est plus générale que les PID présentés précédemment puisqu'il n'est pas nécessaire de faire d'hypothèses sur le degré de A et B, sur le retard, ni sur la présence ou non de zéros stables ou instables dans le procédé.

La structure du correcteur est la structure type RST telle que présentée dans la Figure 29. Pour la suite des calculs on posera pour la fonction de transfert les conventions suivantes :

$$H_{BO} = \frac{q^{-d} \cdot B(q^{-1})}{A(q^{-1})} \quad (74)$$

$$\begin{cases} A(q^{-1}) = 1 + a_1 \cdot q^{-1} + \dots + a_{n_A} \cdot q^{-n_A} \\ B(q^{-1}) = b_1 \cdot q^{-1} + \dots + b_{n_B} \cdot q^{-n_B} = q^{-1} \cdot B^*(q^{-1}) \end{cases}$$

Compte tenu de la forme RST du régulateur la fonction de transfert en boucle fermée vérifie :

$$H_{BF} = \frac{q^{-d} \cdot B(q^{-1}) \cdot T(q^{-1})}{P(q^{-1})} \quad (75)$$

$$P(q^{-1}) = A(q^{-1}) \cdot S(q^{-1}) + q^{-d} \cdot B(q^{-1}) \cdot R(q^{-1}) = 1 + p_1 \cdot q^{-1} + \dots + p_{n_P} \cdot q^{-n_P}$$

Le polynôme  $P(q^{-1})$  définit donc les pôles de la fonction de transfert en boucle fermée et traduit donc le comportement en *poursuite*.

De la même manière la fonction de transfert entre une perturbation et la sortie (fonction de sensibilité perturbation-sortie) est donnée par :

$$S_{yp} = \frac{A(q^{-1}) \cdot S(q^{-1})}{P(q^{-1})} \quad (76)$$

Ce même polynôme traduit donc aussi le comportement en *régulation*.

Les consignes en poursuite et en régulation peuvent être différentes comme cela est représenté par exemple sur la Figure 32.

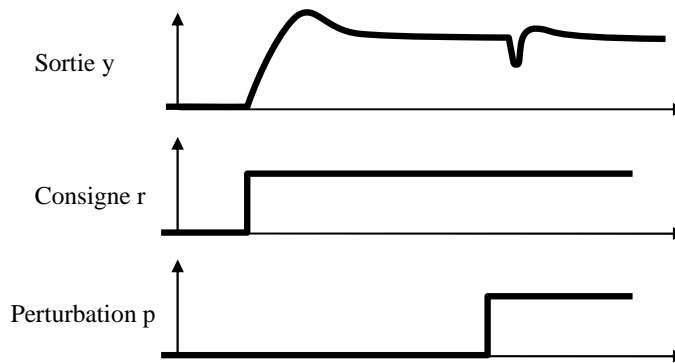


Figure 32 : Performances différentes en régulation et en poursuite

### 7.4.1 Choix des pôles en boucle fermée

De manière générale le comportement en boucle fermée est spécifié sous la forme d'une pulsation propre  $\omega_0$  et d'un amortissement  $\sigma$ , donc une fonction de transfert du second ordre. La spécification étant généralement donnée en continu il est nécessaire de traduire la valeur désirée de  $\omega_0$  et  $\sigma$  en un polynôme  $P_D(q^{-1})$  par discrétisation de la fonction de transfert continue (chapitre 4.5 page 29). Ce polynôme  $P_D$  traduit le comportement dominant du système bouclé.

Si l'on souhaite introduire un effet de filtrage supplémentaire dans certaines bandes de fréquence (réduction des bruits de mesure, adoucissement de la commande, robustesse) cela est décrit par un polynôme  $P_F$  comportant des pôles auxiliaires, plus rapides que les pôles de  $P_D$ .

Le polynôme caractéristique en boucle fermée est donc constitué de pôles dominants et de pôles auxiliaires :

$$P(q^{-1}) = P_D(q^{-1}) \cdot P_F(q^{-1}) \quad (77)$$

### 7.4.2 Régulation : calcul de R et S

Le polynôme  $P(q^{-1})$  étant fixé il s'agit de trouver R et S vérifiant :

$$P(q^{-1}) = A(q^{-1}) \cdot S(q^{-1}) + q^{-d} \cdot B(q^{-1}) \cdot R(q^{-1}) \quad (78)$$

Comme déjà vu au chapitre 7.3.1 on obtient une équation de Bézout qu'il s'agit de résoudre. (Voir aussi 9.1 Annexe 1 : Résolution de l'équation polynomiale de Bezout)

Il est parfois nécessaire de spécifier R et S avec des parties fixes  $H_R$  et  $H_S$ . Dans ce cas l'équation (78) est modifiée et on l'écrit sous la forme:

$$P(q^{-1}) = (A(q^{-1}) \cdot H_S(q^{-1})) \cdot S'(q^{-1}) + (q^{-d} \cdot B(q^{-1}) \cdot H_R(q^{-1})) \cdot R'(q^{-1}) \quad (79)$$

Les inconnues que l'on souhaite trouver sont  $S'$  et  $R'$ . Ceci est nécessaire dans les cas particuliers suivants :

- spécification d'erreur statique nulle,
- rejet d'une perturbation harmonique,
- blocage d'un signal,

Détaillons maintenant ces cas particuliers.

### Spécification d'une erreur statique nulle

On retrouve la nécessité d'avoir un terme intégrateur dans la boucle directe. La spécification se retrouve donc sur  $S$  :

$$S(q^{-1}) = (1 - q^{-1}) \cdot S'(q^{-1}) = H_S(q^{-1}) \cdot S'(q^{-1}) \quad (80)$$

### Rejet d'une perturbation harmonique

Si le système est perturbé par une perturbation harmonique à une fréquence fixe (par exemple 50Hz...) il est possible d'imposer à  $S$  d'avoir un module *nul* ou amorti à cette fréquence. Cela se traduit par la présence dans  $S$  d'une paire de zéros complexe à amortissement nul ou faible à cette fréquence :

$$S(q^{-1}) = (1 + \alpha \cdot q^{-1} + \beta \cdot q^{-2}) \cdot S'(q^{-1}) = H_S(q^{-1}) \cdot S'(q^{-1}) \quad (81)$$

### Blocage d'un signal

Pour certaines application on souhaite que le régulateur "laisse passer" certaines composantes du signal de sortie, c'est-à-dire qu'il ne réagisse pas à ces composantes. Cela se traduit par  $R=0$  à ces fréquences. On retrouve la même forme que précédemment :

$$R(q^{-1}) = (1 + \alpha \cdot q^{-1} + \beta \cdot q^{-2}) \cdot R'(q^{-1}) = H_R(q^{-1}) \cdot R'(q^{-1}) \quad (82)$$

### 7.4.3 Poursuite : calcul de T

Le schéma-bloc de l'asservissement est donné sur la Figure 33. Le terme  $T$  est décomposé en un terme  $T'$  et un terme  $B^*/A^*$ .

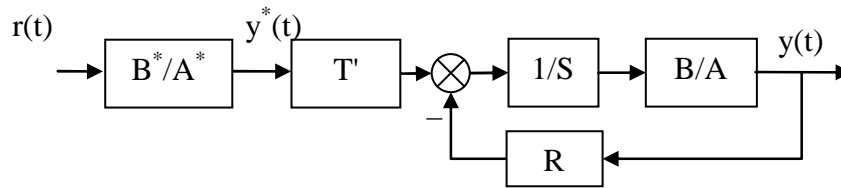


Figure 33 : placement de pôle

L'idée est de régler  $B^*$  et  $A^*$  de telle manière que le signal  $y^*$  ait la dynamique de poursuite désirée (par exemple un second ordre avec pulsation propre et amortissement donné) puis de régler  $T'$  de telle manière que le transfert entre  $y^*$  et  $y$  soit le plus "transparent" possible.

### Réglage de $T'$

Le transfert entre  $y^*$  et  $y$  est :

$$S_{yy^*}(q^{-1}) = T'(q^{-1}) \frac{q^{-d} \cdot B(q^{-1})}{P(q^{-1})} \quad (83)$$

Pour assurer le transfert le plus "transparent" possible il faut donc choisir

$$T'(q^{-1}) = \frac{P(q^{-1})}{B(1)} \quad (84)$$

Remarques :

- Si  $B(1)=0$  alors on remplace  $B(1)$  par 1 dans (84)
- De même dans (84)  $P(q^{-1})$  peut être remplacé par  $P_D$  correspondant à la dynamique dominante.

### Réglage de $B^*$ et $A^*$

$B^*$  et  $A^*$  sont choisis de telle manière que la fonction de transfert  $B^*/A^*$  corresponde à la dynamique de poursuite souhaitée.

## 7.5 Poursuite et régulation à objectifs indépendants

Cette méthode ressemble à la méthode de placement de pôles à ceci près que l'on va effectuer une simplification des zéros de  $B^*(q^{-1})$  : cela ne s'applique donc qu'aux systèmes avec des pôles stables. Plus précisément il faut qu'ils aient un amortissement suffisant ( $> 0,2$ ) de manière à ne pas être trop près du cercle unité.

La structure de l'asservissement est toujours celle donnée sur la Figure 33.

**Choix de P**

$P(q^{-1})$  est choisi de manière à simplifier  $B^*(q^{-1})$  (rappel : la forme de B est donnée dans l'équation 74). Comme indiqué dans le chapitre 7.4.2 page 60 il s'agit donc d'imposer les zéros de  $B(q^{-1})$  dans  $H_R(q^{-1})$  et  $H_S(q^{-1})$ .

**Choix de T**

Si l'on choisit  $T(q^{-1}) = P(q^{-1})$  alors le transfert entre  $y^*$  et  $y$  est :

$$S_{yy^*}(q^{-1}) = q^{-(d+1)} \quad (85)$$

**Réglage de  $B^*$  et  $A^*$** 

$B^*$  et  $A^*$  sont choisis de telle manière que la fonction de transfert  $B^*/A^*$  corresponde à la dynamique de poursuite souhaitée.





## 8. TECHNOLOGIE DE LA COMMANDE PAR ORDINATEUR

Du point de vue de l'automaticien le ordinateur est relié au système physique à piloter par un certain nombre d'intermédiaires plus ou moins gênants :

- les interfaces analogiques et numériques,
- le transport de l'information par bus et protocole,
- l'utilisation éventuelle de capteurs ou d'actionneurs « intelligents ».

Il ne s'agit pas ici de faire un exposé complet sur ces trois aspects mais de poser les problèmes qui se posent à l'automaticien lorsque le système qu'il doit commander est perturbé par ces éléments.

Enfin ce chapitre se termine sur les notions de « Hardware In the Loop » et de « prototypage rapide » qui sont partie intégrante des processus de réalisation de systèmes de contrôle.

### 8.1 Calculateurs et interfaces numériques / analogiques

Le premier problème à prendre en compte est la quantification de l'information. Du point de vue du ordinateur d'une part et du point de vue des convertisseurs d'autre part. Dans un premier temps, considérons les aspects performance de calcul lié à la nature du codage des nombres dans le ordinateur.

#### Les performances liées au types de variables

Prenons l'exemple où le concepteur de la loi de commande souhaite réaliser un correcteur de type proportionnel. Entre l'acquisition des mesures et l'écriture de la consigne il y aura la réalisation d'une de commande du type :

$$y = K \cdot (\text{consigne} - \text{mesure}) \quad (86)$$

En utilisant un langage de programmation évolué (par exemple le C) la ligne de code sera la transcription littérale de (86). Le programmeur devra décider de la définition et du type des variables, principalement du nombre d'octets utilisés pour stocker ses variables. Le Tableau 5 résume par exemple les types utilisés en langage C (norme ANSI)

Tableau 5 : Types ANSI C

Types entiers	char	8 bits
	short int	16 bits
	long int	32 bits
Types flottants (mantisse et exposant)	float	32 bits
	double	64 bits
	longdouble	128 bits

Une fois le code écrit en C l'exécution du programme sur un processeur cible ne peut se faire que par l'intermédiaire d'une compilation du programme. A chaque processeur est donc adapté une version de compilateur.

Les performances du programme (sa rapidité d'exécution) dépendront donc en partie de la qualité du compilateur. (Par exemple, une multiplication s'effectue t'elle bêtement par un décalage de tous les bits ou bien des solutions astucieuses sont-elles implémentées pour arrêter le décalage selon le contexte ?)

Le principal déterminant de la performance est cependant le processeur : son type (8 bits, 16 bits ou 64 bits) et son cadencement (en MHz ou GHz) sont les deux plus importants paramètres. Coder les variables sur 64 bits (double) dans le programme coûtera cher en temps de calcul si le processeur est un 8 bits. Le type de mémoire et leurs temps d'accès peuvent aussi être déterminants. Enfin, certains processeurs disposent d'un processeur spécifiquement dédié aux opérations mathématiques (coprocesseur mathématique).

Il n'est pas possible de dire exactement, pour tel type de compilateur associé à tel type de processeur, le temps que prendra par exemple une multiplication. En pratique on effectue une série de tests types (benchmark) et on affiche un temps moyen d'exécution.

En résumé, on peut être tenté, pour ne pas s'embêter, de coder toutes les variables en virgule flottante. Mais il faut bien savoir que cela a un coût en terme de temps de calcul, surtout si le processeur n'est pas adapté. Pour fixer les idées on peut dire qu'il y a un rapport 20 entre une addition réalisée en "short" et une addition réalisée en virgule flottante.

Enfin, l'utilisation de fonctions mathématiques (sinus, exponentielles, etc...) peut aussi avoir un coût très lourd si le travail n'est pas réalisé par un coprocesseur mathématique dédié. Le compilateur et sa librairie mathématique joue là encore un grand rôle.

### La quantification imposée par le processeur

Au-delà de l'aspect performance et temps de calcul, la quantification de l'information doit parfois aussi être prise en compte. Rappelons qu'un entier codé sur N bits peut prendre seulement  $2^N - 1$  valeurs différentes. Avant d'effectuer les calculs mathématiques dans le corps du programme il est donc judicieux d'appliquer un

facteur d'échelle de manière à utiliser l'amplitude maximale possible (entre 0 et  $2^N - 1$ ).

### **La quantification imposée par les conversions analogique et numérique**

Les convertisseurs analogique-numérique et numérique-analogique sont principalement caractérisés par leur amplitude de signal analogique d'entrée (généralement en volts, par exemple +/- 10V) et par leur résolution en bits (8 ou 10 bits pour les plus courants, jusqu'à 16 bits pour les convertisseurs haut de gamme).

Prenons l'exemple d'un capteur gyromètre transformant une vitesse de rotation en tension. Compte tenu de l'application le signal utile varie entre 2,4V et 2,6V. Si le signal mesuré est branché directement sur un convertisseur 0-10V 10bits alors un problème de conditionnement du signal se pose. En effet la quantification est de  $\frac{10V}{2^{10}-1} = 0,098V$ . Les 0,2V de signal utile seront donc discrétisés en  $\frac{0,2}{0,098} = 20$  intervalles seulement !

Si la discrétisation du signal utile est trop grossière cela peut se traduire par une diminution des performances de l'asservissement, voire une instabilité. Le système n'est en effet plus linéaire et il n'existe pas d'outils théoriques permettant de prévoir les performances du système bouclé.

## **8.2 Transport de l'information : les bus et les protocoles**

Pas question de faire ni même d'ébaucher un cours sur les bus et les protocoles ici. Mais on va quand même donner quelques indications. L'automaticien a l'habitude de simplifier le problème qu'il se pose en dessinant un schéma simplifié de son problème d'asservissement (du type de la Figure 20 page 45) pour pouvoir appliquer les théories de la commande. Cependant il arrive bien souvent que l'information transite à l'intérieur du système, soit entre différents ordinateurs effectuant différentes parties du travail soit entre ordinateur et périphériques, par l'intermédiaire de bus informatique. A chaque type de bus correspond un ou plusieurs protocoles qui peuvent poser des problèmes pour l'automaticien :

- les données sont-elles discrétisées ? (le protocole peut par exemple ne permettre de transmettre que des paquets de données de 16 bits),
- le bus induit-t'il un retard ? (le composant mettant en forme les données du ordinateur pour les publier sur le bus a certaines performances qui peuvent induire un délai)
- le bus est-il déterministe ? (Certains bus ne garantissent pas que les données arriveront en un temps donné),
- le bus est-il sûr ? (Certains bus ne garantissent pas non plus que les données arriveront bien)

### **8.3 Capteurs et actionneurs "intelligents", "smart sensors"**

De plus en plus souvent les capteurs et les actionneurs sont associés à leur électronique de puissance, électronique de commande et de conditionnement de signal. L'interface avec le capteur ou l'actionneur se fait donc via un protocole de communication (par exemple via un bus RS232).

L'intérêt est de véhiculer non pas des signaux analogiques, tensions ou courants, mais des signaux numériques beaucoup moins sensibles aux éventuelles perturbations.

On retrouve donc, entre le capteur ou l'actionneur et le ordinateur, la présence d'un bus et d'un protocole de communication.

### **8.4 La philosophie "hardware in the loop"**

Les systèmes embarqués sont de plus en plus complexes et touchent des domaines de plus en plus variés : automobile, satellites, aéronautique, UAV, systèmes d'arme, sous-marins, systèmes, etc... Les systèmes à piloter ainsi que les systèmes de commande sont de plus en plus complexes.

Les techniques de simulation « Hardware In the Loop » (HIL) sont de plus en plus utilisées dans le développement et dans le test de ces systèmes embarqués. L'idée est de développer entièrement le système de contrôle sur une plateforme de test en utilisant le maximum de composants réels et en ne remplaçant que le minimum par la simulation. Par exemple en aéronautique l'« Iron Bird » regroupe dans un hangar tous les systèmes réels de l'avion en cours de développement. Seule la mécanique du vol est simulée : les sorties des actionneurs sont mesurées, rentrées dans un modèle de simulation numérique, puis les mesures effectuées normalement par les capteurs sont calculées et restituées par le modèle.

L'intérêt est de diminuer de manière importante les coûts et les délais en économisant le plus grand nombre possible de tests sur le système réel en situation. Lors des premiers tests réels les techniques « Hardware In the Loop » permettent d'avoir effectué la plus grande part du développement des lois de commande.

### **8.5 Le prototypage rapide en automatique**

Un peu dans le même ordre d'idée que le « Hardware In the Loop » le prototypage rapide permet de réaliser la conception des lois de commande en simulation puis de transporter rapidement (en un simple click !) le code du contrôleur simulé vers la cible temps réel implémentée sur le système réel.

La société Mathworks [9] investit massivement dans ce domaine en ayant enrichi son logiciel de calcul scientifique (Matlab) avec un logiciel de simulation graphique très évolué (Simulink) puis en développant des outils de prototypage rapide associés (Real Time Workshop, Embedded Target For...).

## 9. ANNEXE

### 9.1 Annexe 1 : Résolution de l'équation polynomiale de Bezout

Il s'agit de trouver R et S vérifiant :

$$P(q^{-1}) = A(q^{-1}).S(q^{-1}) + B(q^{-1}).R(q^{-1})$$

Où P, A et B sont connus.

Dans le cas où P, A, B, R et S sont des nombres entiers la solution de l'algorithme est donnée par l'algorithme de division Euclidienne étendu [8].

Dans le cas où P, A, B, R et S sont des polynômes, l'algorithme est donné ci-dessous, programmé en matlab ([1]).

```
function [Rp,Sp,nrp,nsp]=bezoutd(A,B,Hs,Hr,P)
%function [Rp,Sp,nrp,nsp]=bezoutd(A,B,Hs,Hr,P)
%solves AHsSp+BHRp=P by coefficient comparison. Delay and discretization delay need to be integrated in B
%(zeros at the beginning).
%
%inputs:
%A=[a0 a1 ... aNa] ... vector of model denominator coefficients A=a0 + a1z^(-1) + a2z^(-2) +...+ aNaz^(-Na)
%B=[b0 b1 ... bNb] ... vector of model numerator coefficients B=b0 + b1z^(-1) + b2z^(-2) +...+ bNbz^(-Nb)
%Hs=[hs0 hs1 ... hsNhs] ... vector of controller denominator fixed part Hs=hs0 + hs1z^(-1) +...+ hsNhsz^(-Nhs)
%Hr=[hr0 hr1 ... hrNhr] ... vector of controller denominator fixed part Hr=hr0 + hr1z^(-1) +...+ hrNhrz^(-Nhr)
%P=[p0 p1 ... pNp] ... vector of desired polynomial coefficients P=p0 + p1z^(-1) + p2z^(-2) +...+ pNpz^(-Np)
%outputs:
%Rp=[rp0 rp1 rp2 ...] ... vector of coefficients for resulted controller numerator
%Sp=[sp0 sp1 sp2 ...] ... vector of coefficients for resulted controller denominator
%nrp ... order of Rp
%nsp ... order of Sp
%
%written by: J. Langer, I.D. Landau, H. Prochazka
%7th june 2002

PRECISION=1e-16;

D=size(A);
if D(1)>1, A=A'; end;
```

```

D=size(B);
if D(1)>1, B=B'; end;
D=size(Hs);
if D(1)>1, Hs=Hs'; end;
if D(1)==0, Hs=1; end;
D=size(Hr);
if D(1)>1, Hr=Hr'; end;
if D(1)==0, Hr=1;end;
D=size(P);
if D(1)>1, P=P'; end;

na=length(A)-1;
nb=length(B)-1;
np=length(P)-1;
nhs=length(Hs)-1;
nhr=length(Hr)-1;

if (nhs>0), Ah=real(convz(A,Hs)); else Ah=A*Hs; end; % Ah = A * Hs
nah=length(Ah) -1;
if (nhr>0), Bh=real(convz(B,Hr)); else Bh=B*Hr; end;
nbh=length(Bh) -1;
if (np>nah+nbh-1), disp('Bezout error: too many poles');end;

% increase size of P using zeros if necessary
if (np<nah+nbh-1),
%
% set remaining poles onto a circle with radius rmin
P=[P zeros(1,nah+nbh-1-np)];
rootsPdes=roots(P);
nextra=nah+nbh-1-np;
rmin=1e-16;
angle=[0:nextra-1]'/nextra*2*pi;
j=sqrt(-1);
rootsPextra=rmin*exp(j*angle);
P=poly([rootsPdes;rootsPextra]);
np=nah+nbh-1;

end;
P,

nsp=nbh-1;
nrp=nah-1;

%matrix is smaller than vector PD

```

```

if (np>nah+nbh-1),
    disp('Order of model denominator is too low! Add a polynom of higher order to Hs or Hr. ');
end;

% ns=nsp+nhs
% nr=nrp+nhr

M=[];
for j=1:nsp+1,
    V=[];
    if (j>1), V=[V ; zeros(j-1,1)]; end;% zeros in front of Ah
    V=[V ; Ah'];% Ah
    if (j<=nsp), V=[V ; zeros(nsp+1-j,1)]; end;% zeros behind Ah
    if (length(V)~=nah+nbh), disp('bezoutb: error V'); end;
    M=[M V]; % add one column to M
end;

for j=1:nrp+1,
    V=[];
    if (j>1), V=[V ; zeros(j-1,1)]; end;
    V=[V ; Bh'];
    if (j<=nrp), V=[V ; zeros(nrp+1-j,1)]; end;
    if (length(V)~=nah+nbh), disp('bezoutb: error V'); end;
    M=[M V];
end;

D=size(M);
if (D(1)~=nah+nbh), disp('bezoutb: error size M row'); end;
if (D(2)~=nah+nbh), disp('bezoutb: error size M column'); end;

% make P column vector
P=P';

global M1;
M1=M;

X= M\P;
% coefficients are real values
X=real(X);

% make X row vector
X=X';

```

## Commande des systèmes par ordinateur

```
Sp=X(1:nsp+1);  
Rp=X(nsp+2:nsp+nrp+2);
```



## 10. BIBLIOGRAPHIE

- [1] I. D. Landau, "*Commande des systèmes*", Edition Lavoisier 2002.
- [2] C. Nouals, "Théorie du Signal", Notes de cours ENSICA
- [3] F. Vincent, "*Traitement Numérique du Signal*", Notes de cours ENSICA.
- [4] D. Arzelier, "*Représentation et analyse des systèmes, notes de cours*", Edition ENSICA. Voir aussi <http://www.laas.fr/~arzelier/>
- [5] J. Bordeneuve-Guibé, "*Commande des systèmes linéaires*", Notes de cours ENSICA.
- [6] Y. Briere, "*Identification*", Notes de cours ENSICA.
- [7] L. Ljung, "*System Identification, Theory for the User*", Prentice Hall 1987.
- [8] Wikipedia , équation de Bezout : [http://en.wikipedia.org/wiki/Bézout's\\_identity](http://en.wikipedia.org/wiki/Bézout's_identity)
- [9] <http://www.mathworks.com>



# 11. INDEX

<b>A</b>	
actionneur intelligent .....	21
amortissement .....	15
ANSI C .....	60
Approximation numérique d'équation différentielle .....	28
ARMAX .....	35
<b>B</b>	
BJ35 .....	
Blocage d'un signal .....	56
bloqueurs d'ordre 0 .....	22
BOZ .....	22
bus.....	21, 61
<b>C</b>	
CAN.....	21
capteur intelligent .....	21
causal .....	6, 9
Causalité .....	10
CNA.....	21
compilateur .....	60
convertisseur analogique / numérique.....	21
convertisseur numérique / analogique.....	21
convertisseurs analogique-numérique .....	61
coprocesseur mathématique .....	60
Correspondances pôles - zéros.....	31
<b>D</b>	
Diagramme de Bode .....	18
Discrétisation de fonction de transfert continues .....	27
division polynomiale.....	6
<b>E</b>	
échelon unitaire.....	11
équation d'état .....	17
Erreur statique.....	42
Estimation de la réponse fréquentielle .....	39
Estimation de la réponse impulsionnelle.....	38
Euler .....	29
<b>F</b>	
facteur d'échelle .....	61
facteur d'oubli .....	36
<i>fenêtre de Hamming</i> .....	39
filtre anti-repliement .....	25
fonction de sensibilité .....	42
fonction de transfert discrète.....	9
<b>G</b>	
gain statique.....	11
gain statique unitaire.....	42
<b>I</b>	
Identification.....	32
Identification non paramétrique .....	38
intercorrélation.....	38
<b>L</b>	
lieu des racines.....	43
lieu d'Evans.....	43
LTI.....	8
<b>M</b>	
marge de gain.....	47
marge de module.....	46
marge de phase .....	47
méthode des moindres carrés récursifs .....	32
Méthode des trapèzes.....	29
modèle ARX .....	32
<b>O</b>	
OE.....	35
one step ahead prediction.....	35
opérateur de retard .....	9
<b>P</b>	
période d'échantillonnage (choix) .....	26
Placement de pôles .....	54
Pôles .....	11
Poursuite .....	56
<i>prédiction</i> .....	32
protocoles.....	61
Pseudo Random Binary Signal .....	34
pulsation .....	15
<b>Q</b>	
quantification .....	23, 24, 60
<b>R</b>	
R,S,T.....	41
rapidité.....	15
rayon de convergence .....	5, 9
Régulation.....	55

Rejet de perturbations .....	42
Rejet d'une perturbation harmonique .....	56
réponse à un échelon.....	11
réponse fréquentielle.....	18
Réponse impulsionnelle.....	10
Réponses indicielle .....	10
retard.....	23
root locus .....	43

### S

signal discret .....	5
smart sensors.....	62
stabilité.....	11
Systèmes du premier ordre.....	14
Systèmes du second ordre.....	15

### T

Théorème de Nyquist.....	46
théorème de Shannon.....	19, 24
théorème du retard .....	6
Transformée en z .....	5
transformée unilatérale .....	5
Tustin.....	29

### V

virgule flottante.....	60
------------------------	----

### Z

zero holder hold .....	23
zéros.....	11
zoh .....	23

# Commande des systèmes par ordinateur

Yves Briere  
ENSICA

Notes de cours  
2007



# TABLE DES MATIERES

Table des matières .....	1
1. Introduction .....	1
2. Rappels et prérequis .....	3
3. Signaux et systèmes discrets .....	5
3.1 Signaux discrets .....	5
3.1.1 Transformée en z .....	5
3.1.2 Signaux types .....	7
3.2 Systèmes discrets .....	9
3.2.1 Fonction de transfert discrète, définition .....	9
3.2.2 Notation "opérateur de retard $q^{-1}$ " .....	10
3.2.3 Réponse impulsionnelle .....	11
3.2.4 Causalité .....	11
3.2.5 Réponses indicielle .....	11
3.2.6 Régime stationnaire (gain statique) .....	12
3.2.7 Pôles, zéros, stabilité .....	12
3.2.8 Systèmes du premier ordre .....	15
3.2.9 Systèmes du second ordre .....	16
3.2.10 Calcul itératif de la réponse d'un système .....	18
3.2.11 Représentation par équation d'état .....	18
3.3 Réponse fréquentielle des systèmes échantillonnés .....	19
3.3.1 Diagramme de Bode d'un système échantillonné .....	19
3.3.2 Où l'on voit apparaître le théorème de Shannon .....	20
4. Systèmes continus échantillonnés .....	23
4.1 Structure des systèmes commandés par ordinateur .....	23
4.2 Échantillonnage des systèmes continus .....	24
4.3 Effets de l'échantillonnage .....	25
4.4 Où l'on revoit le théorème de Shannon .....	26
4.5 Discrétisation de fonction de transfert continues .....	29
4.5.1 Approximation numérique d'équation différentielle .....	30
4.5.2 Superposition des réponses indicielles .....	32
4.5.3 Correspondances pôles - zéros .....	33
4.5.4 Bilan .....	33
5. Identification des systèmes échantillonnés .....	35
5.1 Identification d'un modèle ARX par la méthode des moindres carrés récurrents .....	35
5.2 L'identification paramétrique en général et en pratique .....	37

5.2.1	Choix du signal d'excitation, traitement des mesures .....	37
5.2.2	Choix d'une structure de modèle.....	38
5.2.3	Choix d'un critère à minimiser.....	38
5.2.4	Choix d'un algorithme.....	38
5.2.5	Validation.....	39
5.2.6	Utilisation de logiciels .....	39
5.3	Identification "simple" .....	39
5.4	Identification non paramétrique .....	41
5.4.1	Estimation de la réponse impulsionnelle .....	41
5.4.2	Estimation de la réponse fréquentielle.....	42
6.	Systèmes bouclés .....	45
6.1	Fonction de transfert en boucle fermée .....	45
6.2	Erreur statique .....	46
6.3	Rejet de perturbations.....	46
6.4	Stabilité des systèmes à simple gain de rétroaction.....	47
6.5	Analyse des systèmes bouclés dans le domaine fréquentiel : stabilité.....	49
6.5.1	Théorème de Nyquist dans le cas où le système est stable en boucle ouverte (critère du revers) .....	50
6.5.2	Théorème de Nyquist dans le cas où le système est instable en boucle ouverte .....	50
6.5.3	Interprétations du lieu de Nyquist.....	50
7.	Méthodes de régulation des systèmes échantillonnés .....	53
7.1	Introduction .....	53
7.2	Structure d'un système commandé par ordinateur, hypothèses.....	54
7.2.1	Discretisation, choix de la période d'échantillonnage.....	54
7.2.2	Discretisation d'un correcteur continu .....	55
7.3	Régulateur PID numérique .....	55
7.3.1	Cas 1 .....	55
7.3.2	Cas 2 .....	58
7.4	Placement de pôles .....	59
7.4.1	Choix des pôles en boucle fermée .....	60
7.4.2	Régulation : calcul de R et S.....	60
7.4.3	Poursuite : calcul de T .....	61
7.5	Poursuite et régulation à objectifs indépendants .....	62
8.	Technologie de la commande par ordinateur.....	65
8.1	Calculateurs et interfaces numériques / analogiques.....	65
8.2	Transport de l'information : les bus et les protocoles .....	67
8.3	Capteurs et actionneurs "intelligents", "smart sensors" .....	68
8.4	La philosophie "hardware in the loop" .....	68
8.5	Le prototypage rapide en automatique .....	68
9.	Annexe .....	69
9.1	Annexe 1 : Résolution de l'équation polynomiale de Bezout.....	69
10.	Bibliographie.....	73
11.	Index.....	1







# 1. INTRODUCTION



Figure 1 : commande d'un systèmes de télémédecine par calculateur, vu par Claude Serre



## 2. RAPPELS ET PREREQUIS

Ce polycopié ne comporte pas de rappels sur les systèmes continus. S'adressant aux étudiants en deuxième année de l'ENSICA il suppose connu les enseignements de première année et de seconde année en automatique et traitement du signal.

En première année le cours de théorie du signal [2] a rendu familier les notions de :

- transformée de Fourier, transformée de Laplace
- corrélation, convolution

Le cours de traitement numérique du signal [3] a abordé en plus :

- la discrétisation et la quantification des signaux.
- la transformée en z et la transformée de Fourier discrète.
- les filtres numériques RIF et RII.

En automatique [4] les bases de la représentation et de l'analyse des systèmes continus ont été posées. On ne reviendra pas dans ce cours sur les notions de :

- modèles mathématique des systèmes linéaires à temps continus à temps invariant (LTI) : fonction de transfert, équation d'état, pôles, zéros et dynamique.
- commandabilité et observabilité.
- stabilité des systèmes en boucle ouverte : pôles, critères de Routh-Hurwitz.
- stabilité des systèmes bouclés : critère de Nyquist, critère de Bode.
- lieu des racines.

En deuxième année les notions suivantes sont acquises [5] :

- synthèse des correcteurs par méthodes fréquentielles (avance et retard de phase, PID)
- commande par retour d'état (placement de pôles), commande Linéaire Quadratique (LQG)
- reconstruction de l'état par Observateur, observateur optimal de Kalman.



## 3. SIGNAUX ET SYSTEMES DISCRETS

### 3.1 Signaux discrets

#### 3.1.1 Transformée en z

Pour une approche mathématiquement rigoureuse de représentation des signaux discrets par la transformée en z consulter [3] et sa bibliographie. Dans ce cours on se cantonne à une approche utilitariste des outils et méthodes mathématiques tout en essayant de ne pas contredire fondamentalement la rigueur mathématique.

Un signal discret  $s(k)$  est une séquence de nombres réels appelés échantillons. L'indice  $k$  est généralement associé à un instant d'échantillonnage :  $t = k.T_e$ . La période d'échantillonnage  $T_e$  est supposée constante. Par définition la transformée en z du signal  $s(k)$  est la fonction de la variable  $z$ , notée abusivement  $s(z)$  :

$$s(z) = Z(s(k)) = \sum_{k=0}^{\infty} s(k) \cdot z^{-k} \quad (1)$$

La transformée définie ici est plus précisément appelée transformée unilatérale puisque les termes correspondant à  $k < 0$  ne sont pas pris en compte. Ceci n'a pas d'importance puisque les signaux que nous utilisons sont systématiquement causaux<sup>1</sup> :  $s(k) = 0$  pour  $k < 0$ .

Il y a bien évidemment un lien avec la transformée de Laplace et la transformée de Fourier (voir [3])

La fonction de la variable complexe  $s(z)$  est une série et ne converge pas forcément pour toute valeur de  $z$ . En toute rigueur il faudrait associer le rayon de convergence à chaque fonction en  $z$  :  $\{s(z) \text{ converge pour } |z| > \alpha\}$ .

Les automaticiens pour qui la transformée en z n'est bien souvent qu'une façon bien commode de regrouper des termes de récurrence utilisent souvent la pseudo-variable<sup>2</sup>  $q$ .

A partir d'une transformée en z  $s(z)$  il est possible de retrouver la séquence  $s(k)$  correspondante en appliquant la transformation inverse, dont on ne donnera pas la formule ici. A la place de faire appel à cette formule compliquée il est possible d'utiliser d'autres méthodes pour retrouver la séquence  $s(k)$  lorsque la fonction  $s(z)$  est donnée sous forme de fraction rationnelle :

<sup>1</sup> Pour une justification du terme « causal » voir page 11

<sup>2</sup> Voir page 10

- division polynomiale
- décomposition de la fraction rationnelle en éléments simples puis utilisation de transformations élémentaires telles que celles données dans le Tableau 2

La propriété fondamentale de la transformée en z est qu'elle est linéaire (la transformée en z d'une composition linéaire de signaux est la composition linéaire de la transformée en z de chaque signal). Les autres propriétés fondamentales sont données dans le Tableau 2.

Tableau 1 : propriétés de la transformée en z

1	$s(z) = Z(s(k)) = \sum_{k=0}^{\infty} s(k) \cdot z^{-k}$
2	$Z(\alpha \cdot s_1(k) + \beta \cdot s_2(k)) = \alpha \cdot Z(s_1(k)) + \beta \cdot Z(s_2(k))$
3	$Z(k \cdot s(k)) = -z \cdot \frac{ds(z)}{dz}$
4	$Z(c^k \cdot s(k)) = s\left(\frac{z}{c}\right)$
5	$Z(s(k-1)) = s(k=-1) + z^{-1} \cdot s(z)$
6	$Z(s(k+1)) = z \cdot s(z) - z \cdot s(k=0)$
7	$s(k=0) = \lim_{z \rightarrow \infty} (s(z))$
8	$s(k \rightarrow \infty) = \lim_{z \rightarrow 1} \left( \frac{z-1}{z} \cdot s(z) \right)$

Remarque 1 : la propriété 5 ne prend pas en compte le fait que le signal est causal : en pratique  $s(k=-1) = 0$ .

Remarque 2 : la propriété 5 justifie le fait que le terme  $z^{-1}$  soit appelé "opérateur de retard unitaire", cette propriété est aussi appelée "théorème du retard".

Remarque 3 : les propriétés 7 et 8 sont données sous réserve de convergence.

Transformées inverses

### Exemple 1 : transformée inverse par division polynomiale

On considère un signal  $s(k)$  dont la transformée en z est donnée par la fraction rationnelle :

$$S(z) = \frac{0.1 \cdot z^2 + 0.1 \cdot z}{z^3 - 2.6 \cdot z^2 + 2.2 \cdot z - 0.6} \quad (3.1)$$

L'objectif est d'obtenir  $S(z)$  sous forme polynomiale :



$$S(z) = x_0 + x_1 \cdot z^{-1} + x_2 \cdot z^{-2} + \dots \text{etc.} \quad (3.2)$$

Cette suite est généralement infinie, la réponse temporelle est donnée par la succession des termes  $\{x_i\}$ . En traitant l'équation (3.1) comme une simple division l'algorithme de division classique (celui appris en CM2) permet d'obtenir par récurrence tous les termes de la suite (3.2) :

$0.1 \cdot z^2$	$+ 0.1 \cdot z$				$z^3 - 2.6 \cdot z^2 + 2.2 \cdot z - 0.6$
$0.1 \cdot z^2$	$- 0.26 \cdot z$	$+ 0.22$	$- 0.06 \cdot z^{-1}$		
0	$0.36 \cdot z$	$- 0.22$	$0.06 \cdot z^{-1}$		
	$0.36 \cdot z$	$- 0.936$	$0.792 \cdot z^{-1}$	$- 0.216 \cdot z^{-2}$	
0	$0.716$	$- 0.732 \cdot z^{-1}$	$\dots$	$\dots$	

Les éléments successifs de la suite  $s(k)$  sont donc  $s(k) = \{0, 0.1, 0.36, 0.716, \dots\}$ . Pour obtenir les termes suivants il faudrait continuer la division.

**Exemple 2 : transformée inverse par décomposition en éléments simples**

La deuxième méthode présentée consiste à décomposer la fraction rationnelle (3.1) en éléments simples puis d'utiliser les tables de transformées en  $z$  (Tableau 2). Pour notre exemple cela donne :

$$S(z) = \frac{0.1 \cdot z^2 + 0.1 \cdot z}{z^3 - 2.6 \cdot z^2 + 2.2 \cdot z - 0.6} = \frac{-z}{z-1} + \frac{0.5 \cdot z}{(z-1)^2} + \frac{z}{z-0.6}$$

La réponse temporelle est donc composée des trois éléments :

$$s(k) = -(1 \ 1 \ 1 \ \dots) + 0.5 \cdot (1 \ 2 \ 3 \ \dots) + (1 \ 0.6 \ 0.6^2 \ \dots)$$

$$\text{soit : } s(k) = \begin{cases} s(0) = -1 + 0.5 + 1 \\ s(2) = -1 + 0.5 \cdot 2 + 0.6 \cdot 0.5 \\ s(3) = -1 + 0.5 \cdot 3 + 0.6^2 \\ s(4) = \dots \end{cases}$$

**3.1.2 Signaux types**

Le Tableau 2 donne les transformée en  $z$  de certains signaux fondamentaux. La Figure 2 représente l'allure de quelques signaux types.

Tableau 2 : transformée en z élémentaires

$s(k)$	$S(z)$
signal temporel	Transformée en z
$\delta(k)$ : impulsion unité	1
$u(k)$ : échelon unité	$\frac{z}{z-1}$
$k \cdot u(k)$	$\frac{z}{(z-1)^2}$
$c^k \cdot u(k)$	$\frac{z}{z-c}$
$k \cdot a^k \cdot u(k)$	$\frac{a \cdot z}{(z-a)^2}$
$\sin(\omega \cdot k) \cdot u(k)$	$\frac{z \cdot \sin(\omega)}{z^2 - 2 \cdot z \cdot \sin(\omega) + 1}$
$\cos(\omega \cdot k) \cdot u(k)$	$\frac{z \cdot (z - \cos(\omega))}{z^2 - 2 \cdot z \cdot \sin(\omega) + 1}$
$a^k \cdot \sin(\omega \cdot k) \cdot u(k)$	$\frac{a \cdot z \cdot \sin(\omega)}{z^2 - 2 \cdot a \cdot \cos(\omega) \cdot z + a^2}$
$a^k \cdot \cos(\omega \cdot k) \cdot u(k)$	$\frac{z \cdot (z - a \cdot \cos(\omega))}{z^2 - 2 \cdot a \cdot \cos(\omega) \cdot z + a^2}$

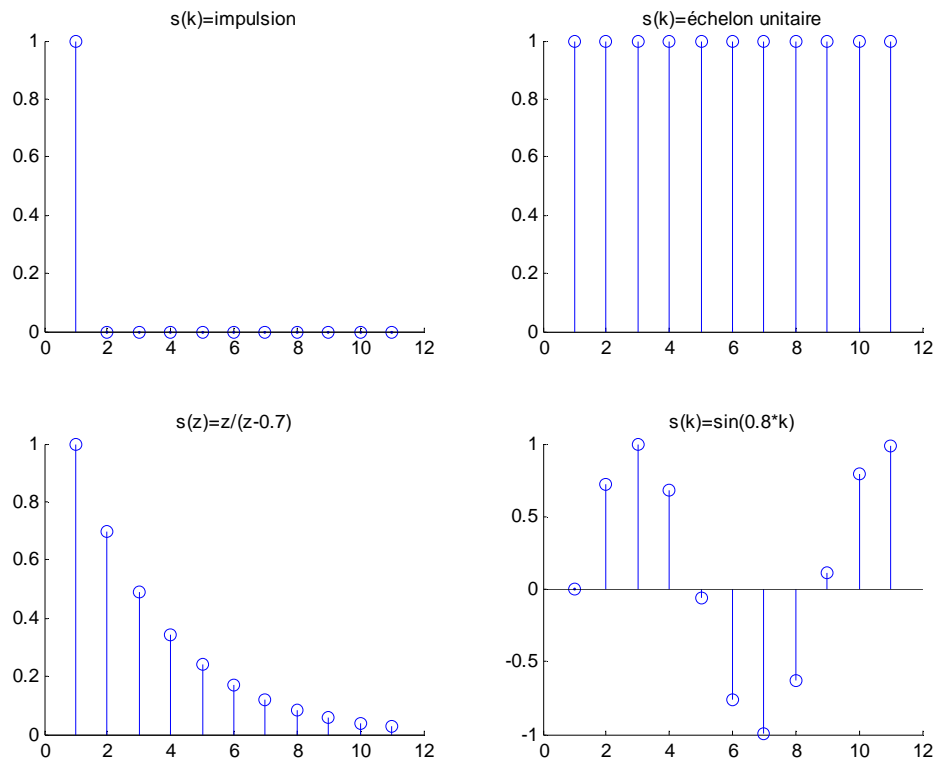


Figure 2 : réponses de quelques signaux types

## 3.2 Systèmes discrets

### 3.2.1 Fonction de transfert discrète, définition

On considère maintenant un système discret linéaire et invariant dans le temps (LTI) représenté sur la Figure 3.

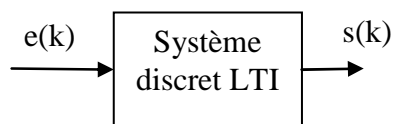


Figure 3 : système discret, notations

La relation entre la séquence d'entrée et la séquence de sortie est linéaire et définie par un ensemble de coefficients constants  $a_{1:n}$  et  $b_{1:m}$  :

$$\begin{aligned} s(k) + a_1 \cdot s(k-1) + a_2 \cdot s(k-2) + \dots + a_n \cdot s(k-n) = \\ b_0 \cdot e(k) + b_1 \cdot e(k-1) + b_2 \cdot e(k-2) + \dots + b_m \cdot e(k-m) \end{aligned} \quad (2)$$

On peut appliquer la transformée en  $z$  à chaque membre de l'équation puis en appliquant en récurrence la propriété 5 (théorème du retard) on obtient la fonction de transfert discrète du système :

$$\frac{s(z)}{e(z)} = F(z) = \frac{b_0 + b_1 \cdot z^{-1} + b_2 \cdot z^{-2} + \dots + b_m \cdot z^{-m}}{1 + a_1 \cdot z^{-1} + a_2 \cdot z^{-2} + \dots + a_n \cdot z^{-n}} \quad (3)$$

Remarque 1 : la fonction de transfert précédente est normalisée en prenant  $a_0 = 1$ .

Remarque 2 : on remarquera que la fonction de transfert est donnée en  $z^{-1}$  plutôt qu'en  $z$ . Cela est justifié par le fait que celle-ci traduit l'équation d'évolution du système physique causal (2).

Le script Matlab permettant de créer un objet Matlab correspondant à une fonction de transfert discrète est donné ci-dessous.

#### Matlab

```
% Création d'un système linéaire discret :
Te=0.1;
sysd = tf([b0 b1 b2],[1 a0 a2 a3],Te,'variable','z^-1');
```

On notera l'utilisation de l'argument 'variable', 'z^-1'.

### 3.2.2 Notation "opérateur de retard $q^{-1}$ "

La fonction de transfert est une fonction de la variable complexe  $z$ . Il faudrait donc en toute rigueur l'associer à son rayon de convergence. En pratique on ne s'embarrasse jamais de préciser cette zone de convergence. C'est pourquoi on trouve souvent la notation utilisant l'opérateur  $q^{-1}$  de décalage à la place de la variable complexe  $z^{-1}$  (4).  $F(q)$  n'est alors plus à proprement parler une fonction mais on lui appliquera les mêmes manipulations algébriques qu'à  $F(z)$ .

$$F(q) = \frac{b_0 + b_1 \cdot q^{-1} + b_2 \cdot q^{-2} + \dots + b_m \cdot q^{-m}}{1 + a_1 \cdot q^{-1} + a_2 \cdot q^{-2} + \dots + a_n \cdot q^{-n}} \quad (4)$$

Attention : selon les auteurs l'opérateur qui remplace  $z^{-1}$  est noté  $q$  (Matlab) ou  $q^{-1}$  (Landau [1]). Dans la suite du document c'est cette dernière convention qui sera appliquée.

L'utilisation de l'opérateur de retard  $q^{-1}$  à la place de la variable complexe  $z^{-1}$  permet de traiter du cas des fonctions de transfert à coefficients non constant, cas non prévu par la transformation en  $z$ .

### 3.2.3 Réponse impulsionnelle

Essayons d'appliquer au système un signal d'entrée particulier : une impulsion unitaire. La transformée en  $z$  du système est obtenue en multipliant la fonction de transfert par la transformée en  $z$  de l'impulsion unitaire : 1. On obtient :

$$s(z) = \frac{b_0 + b_1 \cdot z^{-1} + b_2 \cdot z^{-2} + \dots + b_m \cdot z^{-m}}{1 + a_1 \cdot z^{-1} + a_2 \cdot z^{-2} + \dots + a_n \cdot z^{-n}} \times 1 \quad (5)$$

Cela justifie le fait que la fonction de transfert du système peut aussi s'appeler "réponse impulsionnelle".

#### Matlab

```
% Réponse impulsionnelle d'un système :
[y,t] =impulse(sysd);
```

### 3.2.4 Causalité

Un système est dit causal si les sorties ne dépendent que du passé et non pas du futur. La conséquence est que la réponse impulsionnelle de  $F(z)$  ne peut pas comporter de terme non nuls pour  $k < 0$ . Cela justifie la confusion sémantique entre "système causal" et "signal causal", c'est-à-dire ne comportant pas de termes non nuls pour  $k < 0$ .

Attention : à la différence des systèmes continus un système physique (causal) discret peut très bien être modélisé avec un degré  $m$  supérieur au degré  $n$  : le système peut avoir plus de zéros que de pôles.

La transformée en  $z$  représentant des signaux échantillonnés elle peut tout aussi bien représenter les mesures prises à différents endroits d'une poutre. Le temps étant remplacé ici par une abscisse la notion de causalité ne s'applique pas ! Attention : beaucoup des propriétés de systèmes échantillonnés que l'on étudie sont liés à la causalité, prudence donc si l'on applique ces méthodes à d'autres domaines !

### 3.2.5 Réponses indicielle

Pour caractériser des systèmes on s'intéresse souvent à leur réponse impulsionnelle ou, plus souvent, à leur réponse indicielle. Pour calculer la réponse

à un échelon d'un système il faut donc multiplier la fonction de transfert par la transformée en z d'un échelon unitaire :

$$s(z) = \frac{b_0 + b_1 \cdot z^{-1} + b_2 \cdot z^{-2} + \dots + b_m \cdot z^{-m}}{1 + a_1 \cdot z^{-1} + a_2 \cdot z^{-2} + \dots + a_n \cdot z^{-n}} \times \frac{1}{1 - z^{-1}} \quad (6)$$

Il faudrait ensuite calculer la transformée en z inverse de s(z). En pratique, si l'on doit faire le calcul à la main on décompose en éléments simples puis on utilise des tables. Avec Matlab c'est plus simple. Voici par exemple le script pour obtenir les 20 premiers échantillons de la réponse indicielle d'un système :

**Matlab**

```
% Réponse indicielle d'un système :
[y,t] =step(sysd,20);
```

### 3.2.6 Régime stationnaire (gain statique)

Le gain statique est obtenu en appliquant une entrée indicielle et en calculant la valeur du régime permanent. En appliquant le théorème de la valeur finale il vient :

$$F_0 = \lim_{z \rightarrow 1} \left( \frac{z-1}{z} \cdot \frac{b_0 + b_1 \cdot z^{-1} + b_2 \cdot z^{-2} + \dots + b_m \cdot z^{-m}}{1 + a_1 \cdot z^{-1} + a_2 \cdot z^{-2} + \dots + a_n \cdot z^{-n}} \times \frac{1}{1 - z^{-1}} \right) \quad (7)$$

Soit :

$$F_0 = \frac{\sum_{i=0}^{i=m} b_i}{1 + \sum_{i=1}^{i=n} a_i} \quad (8)$$

Le gain statique est obtenu donc simplement en faisant le quotient de la somme des coefficients.

### 3.2.7 Pôles, zéros, stabilité

Comme pour les systèmes continus la dynamique et la stabilité des systèmes dépend des propriétés du dénominateur de la fonction de transfert.

$$F(z) = \frac{e_0 \cdot (1 + e_1 \cdot z^{-1}) \cdot (1 + e_2 \cdot z^{-1}) \cdots (1 + e_m \cdot z^{-1})}{(1 + c_1 \cdot z^{-1}) \cdot (1 + c_2 \cdot z^{-1}) \cdots (1 + c_n \cdot z^{-1})} \quad (9)$$

Les coefficients  $c_i$  sont les pôles de la fonction de transfert, les coefficients  $e_i$  sont les zéros. La fonction de transfert étant au départ composée de coefficients purement réels les coefficients  $c_i$  sont soit réels soit imaginaires par paires conjuguées.

Une fonction de transfert est donc aussi bien définie par la donnée de ses pôles et de ses zéros. Le script Matlab pour créer un système ainsi est donné ci-dessous :

#### Matlab

```
% Système discret défini par ses pôles et zéros :
Te=0.1;
sysd=zpk([c1 c2 c3],[e1 e2],e0,Te);
```

Si l'on décompose la fonction de transfert en éléments simples on obtient une somme d'éléments du type :

$$F(z) = \frac{d_1}{1 + c_1 \cdot z^{-1}} + \frac{d_2}{1 + c_2 \cdot z^{-1}} + \cdots + \frac{d_n}{1 + c_n \cdot z^{-1}} \quad (10)$$

Un système linéaire temps invariant est dit stable si sa réponse impulsionnelle tend asymptotiquement vers 0. Compte tenu de (10) cela nécessite que chaque pôle  $c_i$  soit de module inférieur à 1.

**Propriété** : un système discret LTI est stable si et seulement si tous ses pôles ont leur modules inférieur ou égal à 1.

#### Critère algébrique attestant de la stabilité : critère de Routh

On rappelle que le critère de Routh permet de prédire si les racines d'un polynôme sont toutes à partie réelle négative, donc stables. Énonçons le théorème de Routh dans le cas d'un polynôme  $P'(w)$  :

$$P'(w) = a_0 + a_1 \cdot w + a_2 \cdot w^2 + \cdots + a_n \cdot w^n \quad (11)$$

Dans un premier temps on construit le tableau :

$$\begin{array}{cccccc|c}
 \beta_{n,0} & \beta_{n,1} & \beta_{n,2} & \beta_{n,q-1} & \beta_{n,q} & 0 & \\
 \beta_{n-1,0} & \beta_{n-1,1} & \beta_{n-1,2} & \beta_{n-1,q-1} & \beta_{n-1,q-1} & 0 & \\
 \beta_{n-2,0} & \beta_{n-2,1} & \beta_{n-2,2} & \beta_{n-2,q-1} & \beta_{n-2,q-1} & & \\
 \\ 
 \beta_{2,0} & \beta_{2,1} & 0 & & & & \\
 \beta_{1,0} & 0 & & & & & \\
 \beta_{0,0} & 0 & & & & & 
 \end{array} \quad (12)$$

Puis on vérifie que tous les premiers éléments de la première colonne ont le même signe : cela garantit que tous les pôles de  $P'(w)$  ont leur partie réelle négative.

Les éléments du tableau sont construits par récurrence ligne par ligne en appliquant les formules suivantes :

Pour  $i = n$  et  $i = n - 1$  :

$$\begin{cases} \beta_{n,j} = a_{n-2j} \\ \beta_{n-1,j} = a_{n-1-2j} \end{cases} \quad (13)$$

Et pour  $i < n-1$  :

$$\beta_{i,j} = \frac{\beta_{i+1,0} \cdot \beta_{i+2,j+1} - \beta_{i+1,j+1} \cdot \beta_{i+2,0}}{\beta_{i+1,0}} \quad (14)$$

Dans le cas qui nous intéresse, un système échantillonné, on cherche plutôt à vérifier que les pôles du polynôme  $P(z)$  sont de module inférieur à un. L'idée est d'effectuer un changement de variable  $z \rightarrow w$  pour pouvoir appliquer le critère de Routh.

$$P'(w) = (1-w)^n \cdot P\left(\frac{1+w}{1-w}\right) \quad (15)$$

Si l'on effectue la transformation (15) alors le fait que  $P'(w)$  ait toutes ses racines à partie réelle négative garantit que  $P(z)$  a toutes ses racines dans le cercle unité. Pour vérifier la stabilité de  $P(z)$  il suffit donc d'effectuer le changement de variable (15) puis d'appliquer le critère de Routh conventionnel (12 13 14).

### Critère algébrique attestant de la stabilité : critère de Jury

Le critère de Jury s'apparente au critère de Routh. Nous le donnons pour un polynôme de degré 2, 3 et 4 du type :



$$P(z) = a_0 + a_1 \cdot z + a_2 \cdot z^2 + \dots + a_n \cdot z^n \quad (16)$$

Les conditions de stabilité sont données ci-dessous (ces conditions sont données dans le cas  $a_n > 0$ , il suffit de multiplier tous les termes par  $-1$  si cette condition n'est pas remplie par  $P(z)$ ) :

$$\begin{aligned}
 n = 2 & \begin{cases} a_0 + a_1 + a_2 > 0 \\ a_0 - a_1 + a_2 > 0 \\ -a_0 + a_2 > 0 \end{cases} \\
 n = 3 & \begin{cases} a_0 + a_1 + a_2 + a_3 > 0 \\ -a_0 + a_1 - a_2 + a_3 > 0 \\ -|a_0| + a_3 > 0 \\ a_0 a_2 - a_1 a_3 - a_0^2 + a_3^2 > 0 \end{cases} \\
 n = 4 & \begin{cases} a_0 + a_1 + a_2 + a_3 + a_4 > 0 \\ a_0 - a_1 + a_2 - a_3 + a_4 > 0 \\ a_4^2 - a_0^2 - |a_0 a_3 - a_1 a_4| > 0 \\ (a_0 - a_4)^2 (a_0 - a_2 + a_4) + (a_1 - a_3)(a_0 a_3 - a_1 a_4) > 0 \end{cases}
 \end{aligned} \quad (17)$$

### 3.2.8 Systèmes du premier ordre

On donne dans la Figure 4 les réponses indicielles de systèmes du premier ordre ainsi que la localisation des pôles dans le plan complexe. Le cercle de rayon unité représente la zone dans laquelle les pôles doivent se situer pour assurer la stabilité. On remarque sur les deux premières figures l'influence de la place du zéro par rapport au point  $(1,0)$  pour la rapidité.

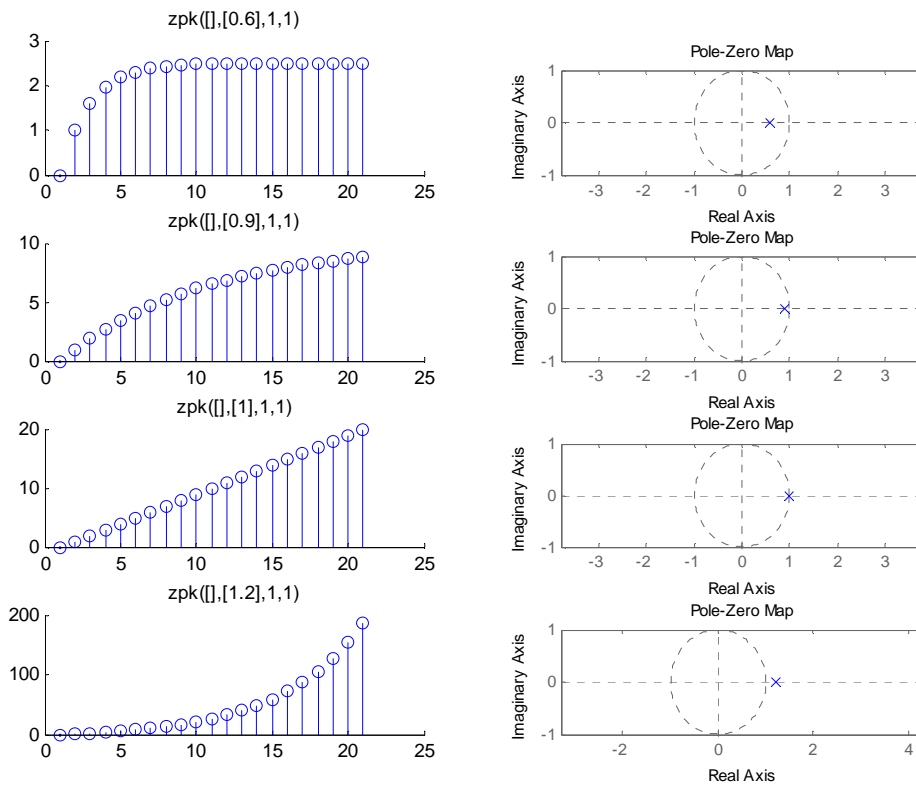


Figure 4 réponses indicielles de systèmes du premier ordre

### 3.2.9 Systèmes du second ordre

On donne dans la Figure 6 les réponses indicielles de systèmes du second ordre ainsi que la localisation des pôles dans le plan complexe. La rapidité et l'amortissement dépend de la position des pôles dans le cercle unité. La Figure 5 représente les courbes iso-amortissement et iso-pulsation des systèmes du second ordre.

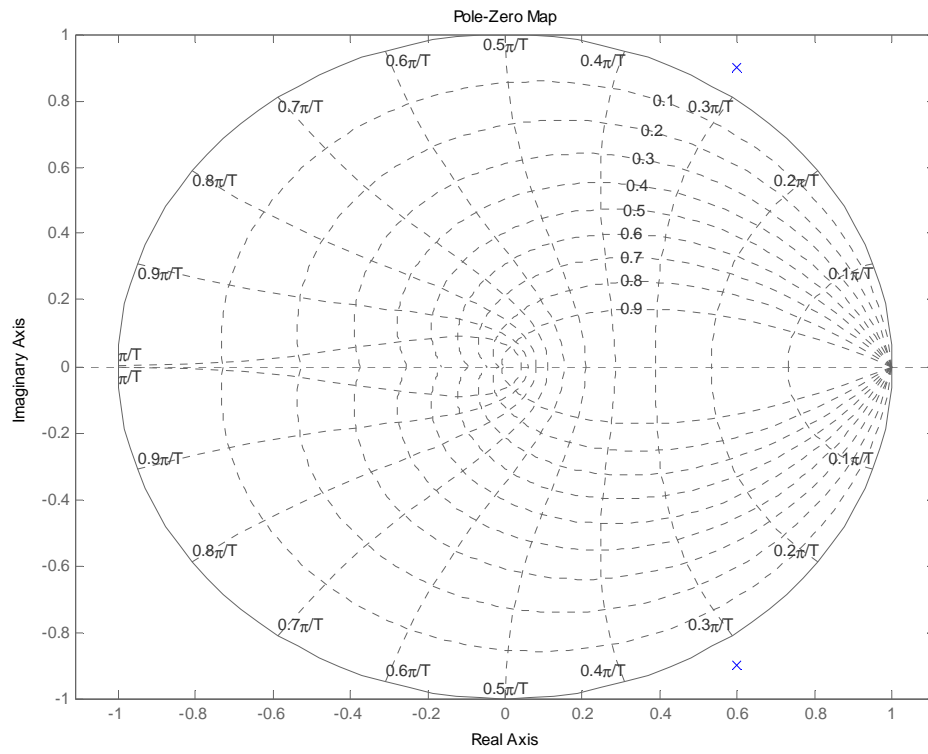


Figure 5 : courbes iso-amortissement et iso-pulsation des systèmes discrets du second ordre

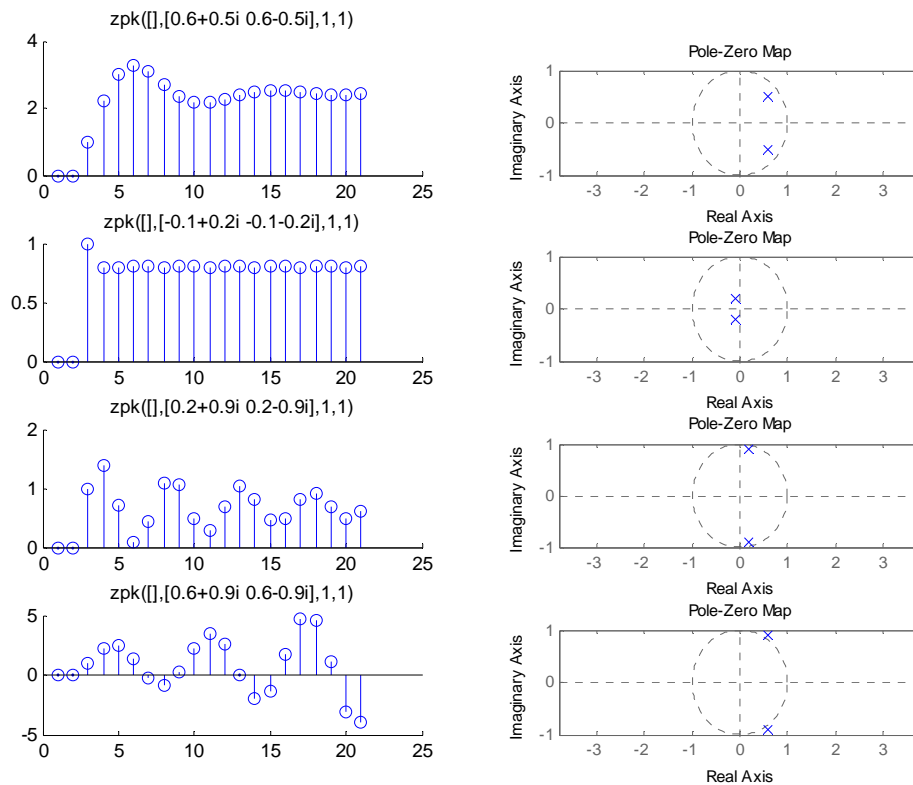


Figure 6 : réponses de systèmes du second ordre

### 3.2.10 Calcul itératif de la réponse d'un système

Si l'on dispose de la fonction de transfert d'un système et que l'on connaît les valeurs successives  $e(k)$  du signal d'entrée il est facile de calculer itérativement la sortie  $y(k)$  du système par l'équation de récurrence (2). Il faut initialiser par les  $n$  valeurs initiales de  $s(k)$ , prises généralement égales à 0.

### 3.2.11 Représentation par équation d'état

Un système discret peut aussi être représenté par un ensemble d'équations d'état comme dans le domaine continu (52). Cette représentation est naturellement adaptée aux systèmes multi-entrées multi-sorties. On définit alors un état interne  $x(k)$  ainsi qu'une fonction d'évolution  $f$  et une équation de mesure  $h$ .

$$\begin{cases} x(k+1) = f(x(k), u(k)) \\ s(k) = h(x(k), u(k)) \end{cases} \quad (18)$$

### 3.3 Réponse fréquentielle des systèmes échantillonnés

#### 3.3.1 Diagramme de Bode d'un système échantillonné

Lorsqu'on étudie la réponse fréquentielle des systèmes continus on considère des entrées particulières sinusoïdales :

$$\begin{cases} e(t) = \cos(2 \cdot \pi \cdot f \cdot t) + j \cdot \sin(2 \cdot \pi \cdot f \cdot t) = e^{j \cdot 2 \cdot \pi \cdot f \cdot t} \\ f \in [0 \quad \infty] \end{cases} \quad (19)$$

De même, pour des systèmes échantillonnés on considèrera des entrées sinusoïdales échantillonnées :

$$\begin{cases} e(k) = \cos(2 \cdot \pi \cdot f \cdot k) + j \cdot \sin(2 \cdot \pi \cdot f \cdot k) = e^{j \cdot 2 \cdot \pi \cdot f \cdot k} \\ f \in [0 \quad 1] \end{cases} \quad (20)$$

La sortie est aussi un signal échantillonné sinusoïdal de gain  $A(f)$  et de phase  $\Phi(f)$  variables et dépendant de  $f$ ; la fréquence réduite.

$$\begin{cases} s(k) = A(f) \cdot e^{j \cdot (2 \cdot \pi \cdot f \cdot k + \Phi(f))} \\ f \in [0 \quad 1] \end{cases} \quad (21)$$

Le nombre complexe, noté abusivement  $F(f)$ , d'amplitude  $A(f)$  et de phase  $\Phi(f)$  est obtenu à partir de  $F(z)$  en remplaçant  $z$  par  $e^{j \cdot 2 \cdot \pi \cdot f}$  :

$$\begin{cases} F(f) = F(z = e^{j \cdot 2 \cdot \pi \cdot f}) \\ f \in [0 \quad 1] \end{cases} \quad (22)$$

Le diagramme représentant le gain et la phase en fonction de la fréquence réduite s'appelle aussi un diagramme de Bode. L'échelle des fréquences est généralement linéaire de 0 à 0.5 (puisque le diagramme est nécessairement symétrique). La Figure 7 représente par exemple le diagramme de Bode d'un système du second ordre.

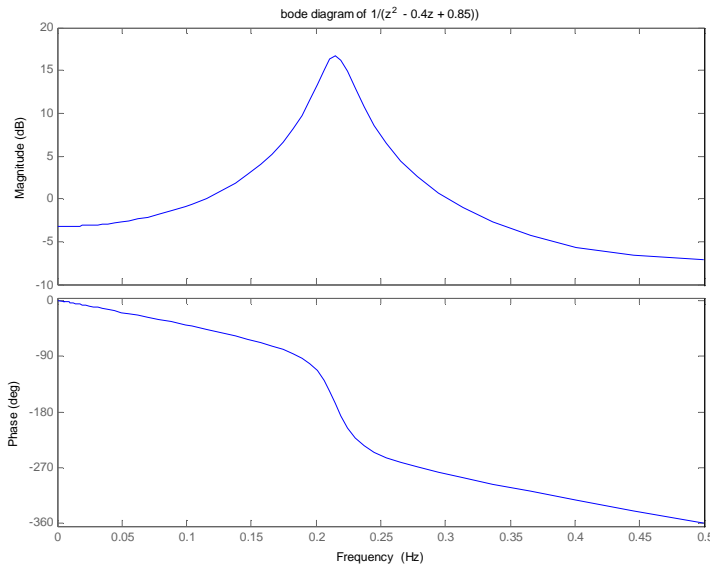


Figure 7 : exemple du diagramme de Bode d'un système discret du second ordre

### 3.3.2 Où l'on voit apparaître le théorème de Shannon

Le changement de variable  $z = e^{j2\pi f}$  n'est pas bijectif puisque  $e^{j2\pi f} = e^{j2\pi(f+k)}$  avec  $k$  entier. Si l'on fait le lien entre domaine continu et domaine fréquentiel on peut remarquer que le changement de variable  $z = e^{j2\pi f}$  fait correspondre une zone du plan complexe continu à une zone du plan complexe discret. La transformation est bijective si l'on se contente de la zone du plan complexe continu contenu entre les droites d'ordonnée 0 et 1.

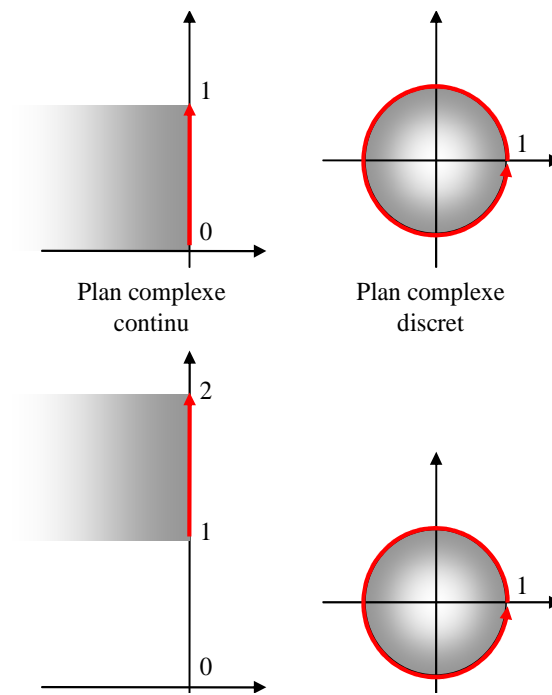


Figure 8 : correspondance entre plan complexe continu et plan complexe discret.





## 4. SYSTEMES CONTINUS ECHANTILLONNES

### 4.1 Structure des systèmes commandés par ordinateur

Les systèmes commandés par ordinateurs sont presque toujours des systèmes physiques continus par nature. Le ordinateur traite par contre les entrées et les sorties de manière discrète avec une période d'échantillonnage  $T_e$ . Les éléments importants, du point de vue de l'automaticien sont donc :

- le convertisseur analogique / numérique CAN ou du moins le système qui fournit séquentiellement les mesures au ordinateur à la période  $T_e$ .
- le ordinateur lui-même. Il doit effectuer l'acquisition des mesures, le traitement des informations, l'envoi des consignes dans un temps inférieur à  $T_e$ .
- le convertisseur numérique / analogique CNA.
- le système physique.

On suppose, pour faire simple et pour pouvoir utiliser la panoplie des outils mathématiques, que les systèmes sont échantillonnés à la même période  $T_e$  et au même moment.

La Figure 9 montre le système physique commandé par le ordinateur, dans sa version simpliste.

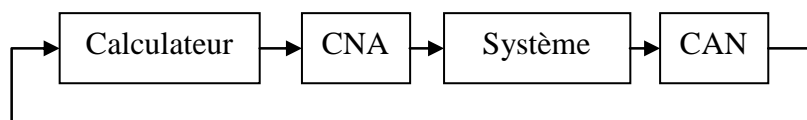


Figure 9 : système commandé par ordinateur

Au XXIème siècle les choses sont quand même parfois un petit peu plus compliquées :

- le ordinateur fournit bien souvent un signal de commande numérique (transporté par bus informatique USB, RS232, CAN, Ethernet, etc...) directement interprété par l'actionneur. On parle "d'actionneur intelligent".
- de même le "capteur intelligent" fournit aussi bien souvent la mesure sous forme d'un signal numérique.
- Les signaux sont transportés par des bus informatiques qui n'ont pas que ça à faire : il peut y avoir du retard.
- de même les capteurs et actionneurs sont rarement tous seuls, ce qui se traduit par des périodes d'échantillonnage différentes, et parfois non synchrones...

- enfin le ordinateur peut aussi avoir d'autres choses à faire...

Ceci étant dit nous allons pour le moment nous restreindre au cas représenté sur la Figure 9 avec des instants d'échantillonnage synchrones et régulièrement espacés d'un temps  $T_e$ .

## 4.2 Echantillonnage des systèmes continus

On dispose généralement de la fonction de transfert continue (notée en utilisant la variable complexe  $s$  ou  $p$ )  $F(s)$  du système continu. Le problème est d'obtenir la fonction de transfert discrète du système échantillonné de manière à pouvoir synthétiser un correcteur. Le système continu est associé au convertisseur numérique – analogique qu'il faut prendre en compte.

Les convertisseurs numérique analogiques peuvent généralement se représenter par des "échantillonneurs – bloqueurs d'ordre 0" ou BOZ. Cela signifie que la commande est maintenue à une valeur constante pendant toute la durée du temps d'échantillonnage.

Si l'on applique une commande discrète impulsionnelle à un BOZ il va maintenir la commande 1 pendant la durée  $T_e$  puis revenir à zéro (Figure 10).

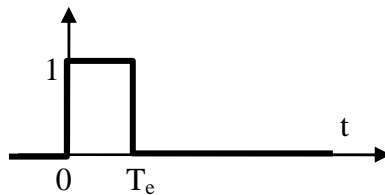


Figure 10 : réponse impulsionnelle d'un bloqueur d'ordre 0

La réponse impulsionnelle  $BOZ(t)$  est donc composée d'un échelon unitaire suivi d'un échelon unitaire de signe opposé et décalé d'un temps  $T_e$ . La transformée de Laplace de la réponse impulsionnelle du bloqueur est donnée par la formule (23) :

$$BOZ(s) = \frac{1}{s} - \frac{e^{-jT_e}}{s} \quad (23)$$

Du point de vue du ordinateur il faut donc prendre en compte le système continu de fonction de transfert  $F_{BOZ}(s)$  que l'on devra échantillonner :

$$F_{BOZ}(s) = \frac{1 - e^{-jT_e}}{s} \cdot F(s) \quad (24)$$

$$F_{\text{BOZ}}(z) = Z^{-1} \left( \frac{1 - e^{-jT_e}}{s} \cdot F(s) \right) \quad (25)$$

En pratique l'obtention de la transformée en z d'un système échantillonné par bloqueur d'ordre 0 se fait à l'aide de tables. On donne quelques exemples fondamentaux dans le Tableau 3 ci-dessous.

Tableau 3 : transformée en z de systèmes continus échantillonnés

F(s)	F <sub>BOZ</sub> (z)
1	1
$\frac{1}{s}$	$\frac{T_e \cdot z^{-1}}{1 - z^{-1}}$
$\frac{1}{1 + T \cdot s}$	$\begin{cases} \frac{b_1 \cdot z^{-1}}{1 + a_1 \cdot z^{-1}} \\ b_1 = 1 - e^{-T_e/T} \quad a_1 = -e^{-T_e/T} \end{cases}$
$\frac{e^{-Ls}}{1 + T \cdot s}$	$\begin{cases} \frac{b_1 \cdot z^{-1} + b_2 \cdot z^{-2}}{1 + a_1 \cdot z^{-1}} \quad a_1 = -e^{-T_e/T} \\ b_1 = 1 - e^{-(L-T_e)/T} \quad b_2 = e^{-T_e/T} (e^{L/T} - 1) \end{cases}$
$L < T_e$	

Si l'on dispose de Matlab c'est bien sûr encore plus simple. On utilise la fonction c2d avec l'argument 'zoh' pour "zero holder hold" (bloqueur d'ordre zéro).

#### Matlab

```
% Définition d'un système continu :
sysc=tf(1,[1 1]);
% Obtention de la transformée en z du système échantillonné à Te
Te=0.1;
sysd=c2d(sysc,Te,'zoh');
% Notation en z^-1
sysd.variable='z'
```

### 4.3 Effets de l'échantillonnage

Le convertisseur numérique a deux effets (néfastes) importants : l'effet de retard et l'effet de quantification.

L'effet de retard se voit dans (24) : on peut dire en première approximation que l'échantillonnage se traduit par un retard pur de  $T_e/2$ . Cela a donc, si ce temps

d'échantillonnage est non négligeable par rapport aux constantes de temps du système, un effet déstabilisant.

Pour illustrer l'effet de quantification supposons par exemple que l'on utilise un convertisseur analogique/numérique 8 bits 0-10V. Le plus petit incrément que le convertisseur peut distinguer est de  $10V/2^8 = 40mV$ . Si le signal utile varie physiquement entre 5V et 5,4V on n'aura que dix incréments utilisables... Ce phénomène n'entre pas dans la panoplie des phénomènes modélisables par la transformée en z. En pratique il est donc souvent nécessaire d'étudier le système et de synthétiser la commande numérique en négligeant ce phénomène puis de vérifier a posteriori qu'il n'a pas un effet trop néfaste.

#### 4.4 Où l'on revoit le théorème de Shannon

Le théorème de Shannon est bien connu en ce qui concerne l'échantillonnage de signaux discrets : on sait que le signal ne doit pas comporter de composante de fréquence supérieure à  $f_e/2$  si l'on ne veut pas observer de repliement. On doit aussi prendre des précautions si l'on veut travailler sur la fonction de transfert discrète correspondant à un système continu échantillonné.

On va traiter un exemple concret pour mettre en évidence le problème de repliement dans le cas de la commande des systèmes continus. On s'intéresse à deux systèmes continus du second ordre ayant des pôles de même partie réelle mais de parties imaginaires séparées de  $2\pi T_e$ . Le script Matlab ci-dessous montre la création des deux systèmes :

##### Matlab

```
% Définition de deux systèmes continus :
sys1=zpk([],[-1-2i -1+2i],2^2+1);
sys2=zpk([],[-1-(2+2*pi)*i -1+(2+2*pi)*i],1+(2+2*pi)^2);
```

On voit sur la Figure 11 que la réponse indicielle des deux systèmes est bien différente.

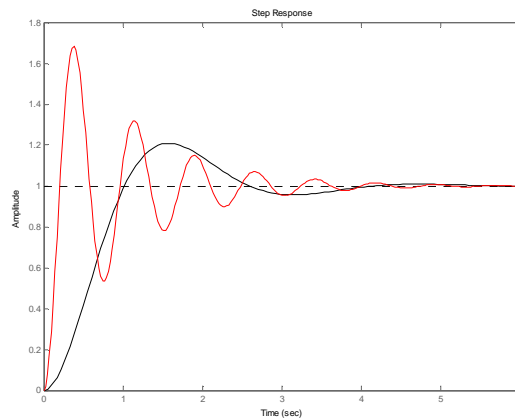


Figure 11 : réponse indicielle de deux systèmes du second ordre

Si l'on calcule la fonction de transfert échantillonnée des deux systèmes à la période de 1 seconde on obtient deux réponses indicielles très proches (représentés par des  $\circ$  sur la Figure 12). Conclusion : la période d'échantillonnage était mal choisie

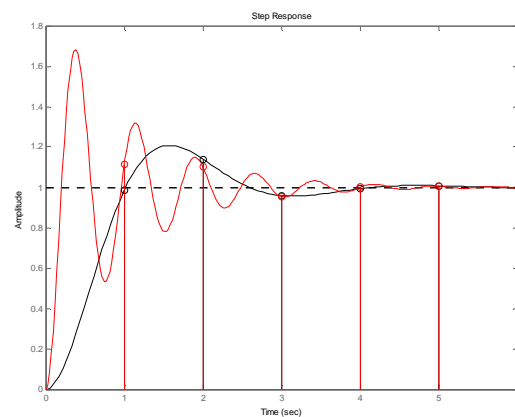


Figure 12 : réponse indicielle de deux systèmes du second ordre échantillonnés à 1s

En pratique la commande échantillonnée de systèmes continus doit être associée à certaines précautions :

- le système ne doit pas comporter de dynamiques propres situées dans les fréquences supérieures à  $f_c/2$  (exemple illustré dans ce chapitre).
- les signaux échantillonnés doivent être "nettoyés" des fréquences supérieures à  $f_c/2$  avec un filtre anti-repliement. (A moins que le signal soit suffisamment propre et ne comporte pas de composantes significatives au-delà de ces fréquences).



### Filtre anti-repliement

En toute rigueur chaque fois qu'un signal est échantillonné il faudrait faire précéder le convertisseur numérique d'un filtre analogique anti-repliement éliminant physiquement toutes les composantes du signal de fréquence supérieure à  $f_e/2$ .

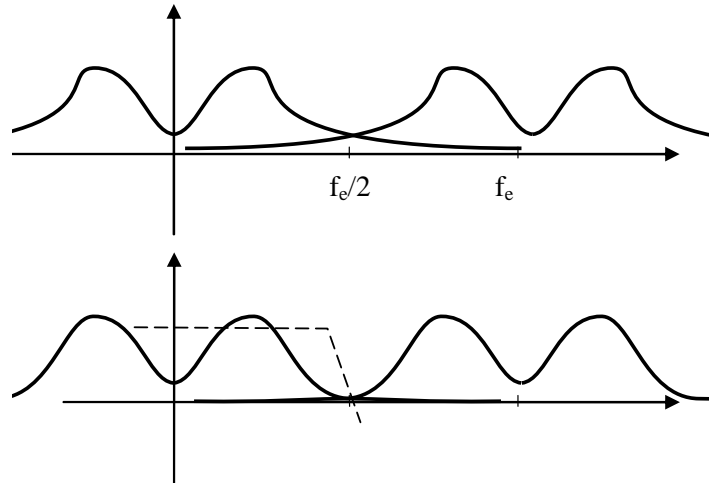


Figure 14 : Effet du filtre anti-repliement

Bien souvent la période d'échantillonnage de l'asservissement est relativement lente (quelques secondes) et il est alors impossible de réaliser un filtre de constante de temps aussi faible. On procède alors à un sur-échantillonnage : le signal est filtré, échantillonné à haute fréquence  $T'_e$  puis filtré numériquement et sous-échantillonné à la période  $T_e$  beaucoup plus grande que  $T'_e$  comme illustré sur la

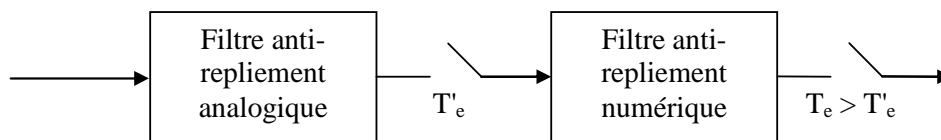


Figure 15 : Sur échantillonnage

Un simple filtre numérique de type moyenne mobile est souvent suffisant.

### 4.5 Discrétisation de fonction de transfert continues

Une manière classique pour réaliser un contrôle par ordinateur est de commencer par synthétiser un correcteur continu avec les outils de l'automatique classique puis de trouver un correcteur discret qui "approche" convenablement le correcteur continu.

#### 4.5.1 Approximation numérique d'équation différentielle

Supposons que le correcteur continu que l'on a synthétisé soit du type PD :

$$\text{PID}(s) = \frac{e(s)}{\varepsilon(s)} = k_p + k_D \cdot s \quad (26)$$

Il s'agit de calculer les commandes  $e(k)$  successives issues du correcteur en fonction des erreurs  $\varepsilon(k)$ . Une approche classique et intuitive est d'approximer la dérivée par une différence :

$$e(k) = k_p \cdot \varepsilon(k) + k_D \cdot \frac{\varepsilon(k) - \varepsilon(k-1)}{T_e} \quad (27)$$

Si on calcule la transformée en  $z$  de  $e(k)$  on obtient :

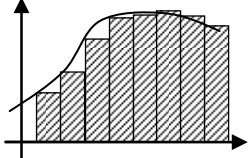
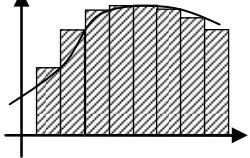
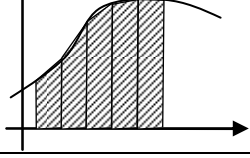
$$e(z) = k_p \cdot \varepsilon(z) + k_D \cdot \frac{\varepsilon(z) - z^{-1}\varepsilon(z)}{T_e} = \left( k_p + k_D \cdot \frac{1 - z^{-1}}{T_e} \right) \cdot \varepsilon(z) \quad (28)$$

On retrouve la fonction de transfert PD(s) dans laquelle on a effectué le changement de variable  $s = \frac{1-z^{-1}}{T_e}$

De manière plus générale, l'approximation de fonction de transfert revient à effectuer l'approximation numérique d'intégration. Selon la méthode d'intégration choisie on effectue des changements de variable différents.



Tableau 4 : méthodes d'intégration

Méthode	Changement de variable	
Euler progressif (Euler's forward method)	$s = \frac{1 - z^{-1}}{z^{-1} \cdot T_e}$	
Euler retrograde (Euler's backward method)	$s = \frac{1 - z^{-1}}{T_e}$	
Méthode des trapèzes, ou bilinéaire, ou de Tustin	$s = \frac{2}{T_e} \cdot \frac{1 - z^{-1}}{1 + z^{-1}}$	

En pratique le changement de variable  $s = \frac{1 - z^{-1}}{z^{-1} \cdot T_e}$  n'est pas sans risque. On donne sur la Figure 16 la correspondance entre la zone stable du plan complexe en  $s$  et la zone correspondante après la transformation (Euler progressif, Euler rétrograde, bilinéaire). On voit notamment qu'il est possible, partant d'un système continu stable, d'obtenir des pôles dans le domaine discret en dehors du cercle de stabilité si la transformation appliquée est celle d'Euler progressif.

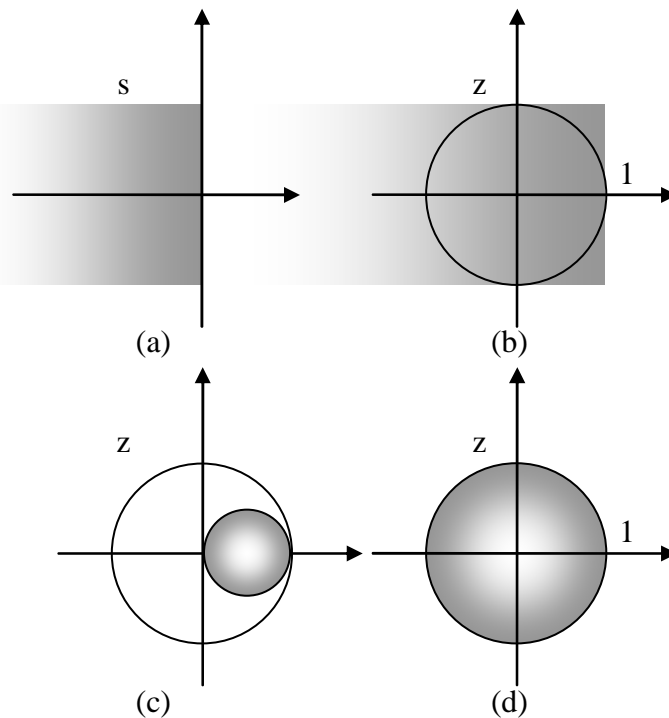


Figure 16 : correspondance des plans complexes. (a) : zone de stabilité dans le plan complexe continu en  $z$  ; (b) zone correspondante avec l'approximation d'Euler progressif ; (c) zone correspondante avec l'approximation d'Euler rétrograde ; (d) zone correspondante avec l'approximation bilinéaire.

La fonction `c2d` de Matlab permet d'obtenir l'équivalent discret d'une fonction de transfert continue avec la méthode d'intégration comme argument :

```

Matlab
sysc=tf([1 0],[1 2 5]);
Te=0.1;
% Méthode d'euler :
sysd1=c2d(sysc,Te,'zoh');
% Méthode des trapèzes :
sysd2=c2d(sysc,Te,'foh');
% Méthode bilinéaire :
sysd3=c2d(sysc,Te,'tustin');
    
```

#### 4.5.2 Superposition des réponses indicielles

Une autre méthode pour approximer une fonction de transfert discrète est de s'assurer que la réponse indicielle de sa version discrétisée correspond exactement lors des temps d'échantillonnage.

### 4.5.3 Correspondances pôles - zéros

La dernière méthode est de partir d'une décomposition de la fonction de transfert continue en pôles et zéros et d'effectuer la conversion pôles à pôles et zéros par zéros. La correspondance, selon que le pôle est réel ou complexe est donnée ci-dessous :

$$\begin{aligned} (s + a) &\rightarrow (z - e^{-a \cdot T_e}) \\ (s + a)^2 + b^2 &\rightarrow z^2 - 2 \cdot e^{-a \cdot T_e} \cdot \cos(b \cdot T_e) \cdot z + e^{-2 \cdot a \cdot T_e} \end{aligned} \quad (29)$$

### 4.5.4 Bilan

Aucune méthode n'est parfaite et on n'aura jamais en même temps :

- même réponse indicielle
- même réponse impulsionnelle
- approximation (Euler, Tustin, etc...) de l'équation différentielle
- correspondance des pôles et des zéros
- correspondance des réponses fréquentielles

Ceci étant dit le choix de la période d'échantillonnage est primordial. Si la fréquence d'échantillonnage est beaucoup plus faible que la dynamique de la fonction de transfert à approximer la différence entre chaque méthode sera minimale.



## 5. IDENTIFICATION DES SYSTEMES ECHANTILLONES

Dans le présent chapitre on donne les bases des méthodes d'identification. Pour une approche complète, voir le cours de troisième année "Identification avion" [6] et les ouvrages [1] et [7].

Même si l'on dispose d'un modèle théorique du processus que l'on souhaite commander il est presque toujours nécessaire de l'affiner en procédant à des essais sur le système réel. Bien souvent le seul modèle dont l'on dispose est ce modèle issu de l'expérimentation. Différentes méthodes s'appliquent selon l'objectif recherché et le niveau de précision exigé. On distinguera :

- les modèles paramétriques
- les modèles simples
- les modèles non paramétriques

### 5.1 Identification d'un modèle ARX par la méthode des moindres carrés récursifs

A partir de cet exemple, le plus simple qui soit, on pose les bases de toutes les méthodes d'identification.

On considère un système d'entrée  $u(t)$  et de sortie  $y(t)$  représenté par l'équation de récurrence suivante :

$$y(t) + a_1 \cdot y(t-1) + a_2 \cdot y(t-2) + \dots + a_n \cdot y(t-n) = b_0 \cdot u(t) + b_1 \cdot u(t-1) + b_2 \cdot u(t-2) + \dots + b_m \cdot u(t-m) \quad (30)$$

L'objectif est, à partir d'essais et de mesures, d'identifier les meilleures valeurs possibles pour les coefficients  $a_i$  et  $b_i$ .

L'équation (30) peut être réécrite pour mettre en évidence la *prédiction* de la sortie  $y(t)$  à partir des mesures passées :

$$y(t) = -a_1 \cdot y(t-1) - a_2 \cdot y(t-2) - \dots - a_n \cdot y(t-n) + b_0 \cdot u(t) + b_1 \cdot u(t-1) + b_2 \cdot u(t-2) + \dots + b_m \cdot u(t-m) \quad (31)$$

Soit :

$$y(t) = \varphi^T(t) \cdot \theta(t)$$

avec :

$$\begin{cases} \varphi^T(t) = [-y(t-1) \ \dots \ -y(t-n) \ , \ u(t-1) \ \dots \ u(t-m)] \\ \theta(t) = [a_1 \ \dots \ a_n \ , \ b_1 \ \dots \ b_m] \end{cases} \quad (32)$$

Le but est de trouver le meilleur ensemble de paramètres  $\theta(t)$  : celui qui, à partir d'un ensemble de mesures  $\varphi(t)$  permet de prédire avec la meilleure précision la sortie effectivement mesurée.

Supposons qu'à la suite d'une expérience on dispose d'un ensemble de mesures  $\{y(t), u(t)\}_{t=1:N}$  et que l'ensemble des paramètres  $\theta$  n'est pas bien connu. On construit la séquence des *sorties prédites* :

$$\hat{y}(t) = \varphi^T(t) \cdot \theta(t) \quad (33)$$

Le paramètre  $\theta$  est celui qui minimise une norme quadratique :

$$\theta = \arg \max \left( \frac{1}{N} \sum_{t=1}^N (y(t) - \hat{y}(t))^2 \right) = \arg \max \left( \frac{1}{N} \sum_{t=1}^N (y(t) - \varphi^T(t) \cdot \theta(t))^2 \right) \quad (34)$$

Compte tenu du fait que l'équation est linéaire au départ et que le terme à minimiser est du type quadratique la solution est simplement obtenue par une pseudo-inverse :

$$\theta = \left[ \sum_{t=1}^N (\varphi(t) \cdot \varphi^T(t)) \right]^{-1} \sum_{t=1}^N (\varphi(t) \cdot y(t)) \quad (35)$$

On obtient donc la solution classique des moindres carrés. La formule (35) est le plus souvent utilisée sous sa forme récursive :

$$\theta(t) = \theta(t-1) + F(t) \cdot \varphi(t) \cdot (y(t) - \theta^T(t-1) \cdot \varphi(t))$$

avec

$$F(t)^{-1} = F(t-1)^{-1} + \varphi(t) \cdot \varphi(t)^T \quad (36)$$

ou

$$F(t) = F(t-1) - \frac{F(t-1) \cdot \varphi(t) \cdot \varphi(t)^T \cdot F(t-1)}{1 + \varphi(t)^T \cdot F(t-1) \cdot \varphi(t)}$$

(On laisse au lecteur le plaisir d'effectuer lui-même les démonstrations, qu'il peut du reste trouver dans [1] et [7]).

## 5.2 L'identification paramétrique en général et en pratique

L'exemple précédent a permis de mettre en évidence tous les aspects de la méthodologie classique de l'identification :

- il a fallu choisir un ensemble de mesures  $\{y(t), u(t)\}_{t=1:N}$ . Le choix du signal d'excitation  $u(t)$  est en pratique primordial,
- il a fallu choisir une structure pour le modèle à identifier (ici les ordres  $n$  et  $m$ ),
- il a fallu choisir un critère (ici un critère quadratique sur l'erreur entre la sortie et la sortie prédite),
- il a fallu choisir un algorithme et le paramétrer (ici l'algorithme des moindres carrés récursifs qu'il a fallu initialiser)

Enfin un point non traité dans le chapitre précédent concerne la validation :

- il aurait fallu déterminer une méthode pour comparer différents modèles et choisir le meilleur

### 5.2.1 Choix du signal d'excitation, traitement des mesures

De manière générale le système que l'on doit identifier doit être excité dans tout le domaine fréquentiel qui nous intéresse. Le signal classiquement utilisé est une séquence pseudo binaire aléatoire (Pseudo Random Binary Signal) tel celui représenté sur la Figure 17.

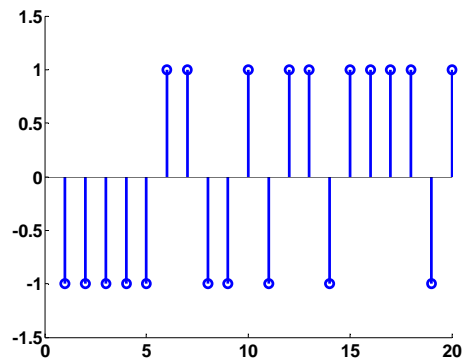


Figure 17 : exemple d'une séquence PRBS

Le signal d'excitation doit se superposer au signal de commande correspondant au point de fonctionnement étudié. Son amplitude doit être faible par rapport à l'amplitude des commandes du système (quelques %).

Une fois le système excité et les mesures effectuées il faut supprimer les valeurs moyennes des mesures.

### 5.2.2 Choix d'une structure de modèle

Le modèle étudié dans le chapitre précédent est de type ARX. On peut le noter sous forme polynomiale :

$$A \cdot y = B \cdot u \quad (37)$$

Ce modèle correspond en fait à un parti-pris sur la manière dont on traite les perturbations. En réalité on prend en compte un signal d'erreurs possibles  $e$  (de moyenne nulle) tels que :

$$A \cdot y = B \cdot u + e \quad (38)$$

Si l'on fait cette hypothèse de modélisation alors la méthode optimale est celle des moindres carrés. Malheureusement cette manière de prendre en compte les perturbations est bien restrictive. D'autres modélisations sont possibles, adaptées à d'autres algorithmes, par exemple le modèle OE :

$$y = \frac{B}{F} \cdot u + e \quad (39)$$

Il est aussi possible de "structurer" le modèle de bruit comme par exemple dans les modèles ARMAX ou BJ :

$$\begin{aligned} A \cdot y &= B \cdot u + C \cdot e \\ y &= \frac{B}{F} \cdot u + \frac{C}{D} \cdot e \end{aligned} \quad (40)$$

La structure du modèle étant choisie il faut choisir l'ordre de ses différents polynômes.

### 5.2.3 Choix d'un critère à minimiser

Dans la plupart des cas le critère à minimiser est l'erreur quadratique entre la sortie et la sortie prédite à un coup (one step ahead prediction). Ceci n'est bien évidemment pas restrictif et peut être généralisé à la prédiction à N coups.

### 5.2.4 Choix d'un algorithme

Même si chaque structure de modèle est généralement associée à un algorithme "optimal" celui-ci doit généralement se paramétrer.



Prenons par exemple l'algorithme des moindres carrés récursifs tel que présenté dans (36). Cette équation peut être modifiée en intégrant un "facteur d'oubli"  $\lambda$  :

$$\begin{aligned} \theta(t) &= \theta(t-1) + F(t) \cdot \varphi(t) \cdot (y(t) - \theta^T(t-1) \cdot \varphi(t)) \\ \text{avec} & \\ F(t)^{-1} &= \lambda \cdot F(t-1)^{-1} + \varphi(t) \cdot \varphi(t)^T \end{aligned} \quad (41)$$

Ce paramètre (typiquement entre 0.95 et 0.99) permet, comme son nom l'indique, "d'oublier" les anciennes mesures et de se concentrer sur les mesures les plus récentes.

### 5.2.5 Validation

Une fois l'algorithme d'identification appliqué et le modèle obtenu (ce qui prend généralement quelques secondes avec un ordinateur moderne) il reste à expertiser celui-ci. Selon la "qualité" du modèle obtenu on remet en cause l'une ou l'autre des étapes (signal d'excitation, structure du modèle, algorithme) et on réitère jusqu'à obtenir un modèle satisfaisant.

En pratique on essaie un certain nombre de modèles avant de choisir le "meilleur".

Le premier critère de qualité est la capacité du modèle à prédire effectivement la sortie (sortie prédite à un coup, sortie prédite à N coups). Il faut pour cela utiliser un jeu de mesures non utilisé pour l'identification. (En pratique le jeu de mesures initiales est coupé en deux : une partie pour l'identification et une partie pour la validation).

Le second critère de qualité est l'analyse du résidu : la composition spectrale de l'erreur de prédiction.

### 5.2.6 Utilisation de logiciels

En pratique il est commode d'utiliser un célèbre outil de calcul scientifique et sa toolboxe d'identification.

Comme les calculs proprement dit ne prennent que quelques secondes il est facile de réaliser un grand nombre d'identifications et de stocker les différents modèles pour les comparer ensuite. On est rapidement noyé sous le grand nombre de modèles possibles...

## 5.3 Identification "simple"

Il est souvent suffisant d'obtenir un modèle "simple" de type premier ordre et un simple essai indiciel convient souvent. Le système est excité avec un échelon unité et on cherche le meilleur modèle du premier ordre correspondant.

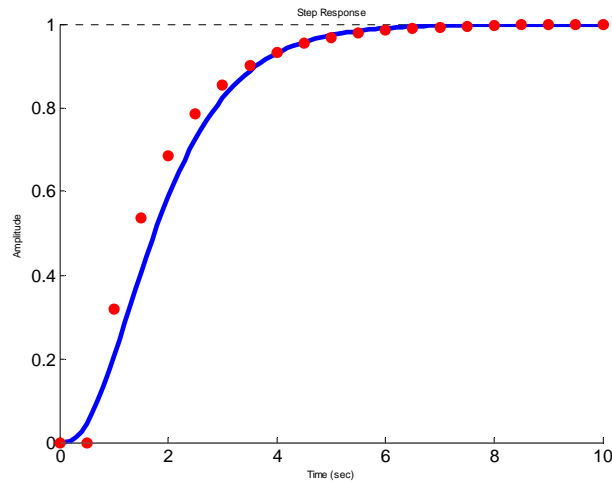


Figure 18 : identification sur un essai indiciel

Dans l'exemple de la Figure 18 on superpose le résultat d'une mesure (en bleu) avec la simulation d'un modèle discret avec une période de 0.5s. Le modèle cherché est du premier ordre avec un retard pur  $d$  :

$$y(z) = \frac{(1 + c_1) \cdot z^{-d}}{1 + c_1 \cdot z^{-1}} \cdot u(z) \quad (42)$$

Il s'agit de trouver l'entier  $d$  et le nombre réel  $c_1$  tels que la réponse indicelle du modèle corresponde pour le mieux aux mesures issues de l'essai. Ceci peut se formaliser par exemple de la manière suivante :

$$(c_1 \quad d_1) = \arg \min \sum_{t=1}^N (y_{\text{modèle}}(t) - y_{\text{mesure}}(t))^2 \quad (43)$$

Cette optimisation peut se faire à la main ou grâce à une procédure automatisée (avec le célèbre outil de calcul scientifique).

Dans le cas où le système comprend naturellement une intégration (par exemple la commande d'un moteur à courant continu par une tension et la mesure de l'angle de rotation) il est utile de prendre en compte explicitement cette intégration dans le modèle :

$$y(z) = \frac{1}{1 - z^{-1}} \cdot \frac{(1 + c_1) \cdot z^{-d}}{1 + c_1 \cdot z^{-1}} \cdot u(z) \quad (44)$$

La procédure est la même mais au lieu de prendre en compte la sortie  $y(t)$  on prend en compte la dérivée  $y'(t)$  :

$$y'(t) = y(t) - y(t-1) \quad (45)$$

## 5.4 Identification non paramétrique

Pour terminer ce panorama rapide sur les méthodes d'identification il faut parler des méthodes non paramétriques : celles qui permettent de tracer la réponse impulsionnelle ou la réponse fréquentielle du système.

### 5.4.1 Estimation de la réponse impulsionnelle

La réponse impulsionnelle d'un système peut être obtenue simplement en appliquant une entrée de type "impulsion" au système et d'en mesurer les conséquences.

La seconde méthode consiste à utiliser la méthode des moindres carrés (ou une méthode dérivée) comme vue au chapitre 5.1. La réponse impulsionnelle d'un système peut en effet être vue comme la fonction de transfert particulière où le dénominateur est de degré nul et le numérateur comporte un nombre infini (en pratique : grand) d'éléments ( $n = 0$  et  $m$  grand) :

$$y(t) = b_0 \cdot u(t) + b_1 \cdot u(t-1) + b_2 \cdot u(t-2) + \dots + b_m \cdot u(t-m) \quad (46)$$

La dernière méthode, qui sera développée ici est l'analyse d'intercorrélation.

Par définition l'intercorrélation  $R_{yu}$  entre deux signaux est :

$$R_{yu}(\tau) = E[(y(t) - E(y(t))) \cdot (u(t-\tau) - E(u(t-\tau)))] \quad (47)$$

Dans cette formule l'espérance mathématique peut être approchée grâce à la loi des grands nombres :

$$R_{yu}(\tau) \approx R_{yu}^N(\tau) = \lim_{N \rightarrow \infty} \frac{1}{N} \sum_{t=1}^N [(y(t) - m_y) \cdot (u(t-\tau) - m_u)] \quad (48)$$

On s'intéresse maintenant à un système tel que décrit par l'équation impulsionnelle (46) et on suppose que le signal d'entrée est un bruit blanc. Le signal d'entrée vérifie donc :

$$\begin{cases} \tau \neq 0 & R_{uu}(\tau) = 0 \\ \tau = 0 & R_{uu}(\tau) = R \end{cases} \quad (49)$$

Il vient alors naturellement :

$$b_k \approx \frac{1}{R} R_{yu}(k) \quad (50)$$

#### 5.4.2 Estimation de la réponse fréquentielle

On utilise ici le spectre d'intercorrélation. Par définition le spectre d'intercorrélation entre deux signaux est :

$$\Phi_{yu}(\omega) = \sum_{\tau=-\infty}^{\infty} R_{yu}(\tau) \cdot e^{-i\omega\tau} \quad (51)$$

On suppose que le signal d'entrée  $u(t)$  et le signal de sortie  $y(t)$  sont reliés par une fonction de transfert linéaire du type :

$$y(t) = F \cdot u(t) + v(t) \quad (52)$$

où  $v(t)$  est une perturbation indépendante de  $u(t)$ . On a alors les relations :

$$\Phi_{yu}(\omega) = F(j\omega) \cdot \Phi_{uu}(\omega) \quad (53)$$

Il reste à estimer les spectres d'intercorrélation pour remonter à la réponse fréquentielle  $F(j\omega)$  recherchée. Une idée naturelle est d'utiliser la relation (48) mais cela marche très mal dans presque tous les cas, même en choisissant  $N$  assez grand. L'amélioration est d'utiliser une "fenêtre" pour favoriser les valeurs faibles de  $\tau$  par rapport aux valeurs élevées. La solution classique utilisant la *fenêtre de Hamming* donne :

$$\Phi_{yu}^N(\omega) = \sum_{\tau=-\gamma}^{\gamma} R_{yu}^N(\tau) \cdot w_{\gamma}(\tau) \cdot e^{-i\omega\tau} \quad (54)$$

avec

$$\begin{cases} w_{\gamma}(\tau) = \frac{1}{2} \left( 1 + \cos\left(\frac{\pi \cdot \tau}{\gamma}\right) \right) & |\tau| < \gamma \\ w_{\gamma}(\tau) = 0 & |\tau| > \gamma \end{cases}$$

On utilise ensuite la relation (51) pour estimer la réponse fréquentielle :

$$F(j \cdot \omega) = \frac{\Phi_u^N(\omega)}{\Phi_{yu}^N(\omega)} \quad (55)$$



## 6. SYSTEMES BOUCLES

### 6.1 Fonction de transfert en boucle fermée

Un système et sa commande par ordinateur peut être représenté par un ensemble de fonctions de transfert formant un schéma-bloc plus ou moins compliqué (Figure 19).

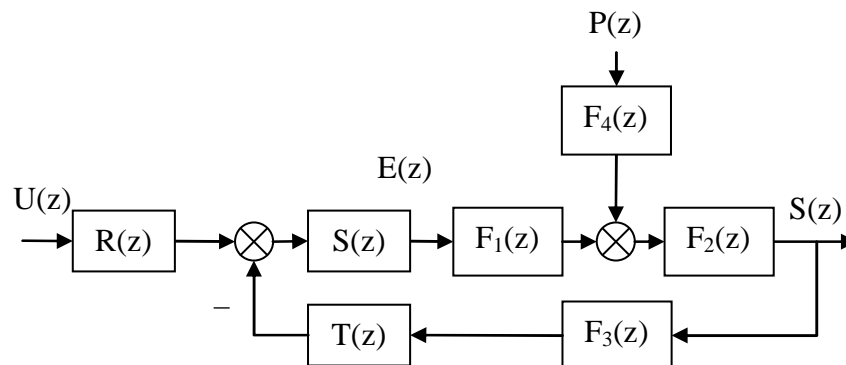


Figure 19 : système discret commandé par ordinateur. Le correcteur est représenté par les fonctions R,S,T, le système par  $F_1$  et  $F_2$ , le modèle de perturbations par  $F_4$  et le modèle des actionneurs par  $F_4$ .

Quelle que soit la structure d'un système asservi il est possible de se ramener à une forme canonique comprenant une fonction de transfert en boucle ouverte en combinant judicieusement les fonctions de transfert du procédé, du régulateur et des perturbations. La Figure 20 représente par exemple le système de la Figure 19 où  $F$  est une combinaison des fonctions de transfert du système, du régulateur, des perturbations et des capteurs.

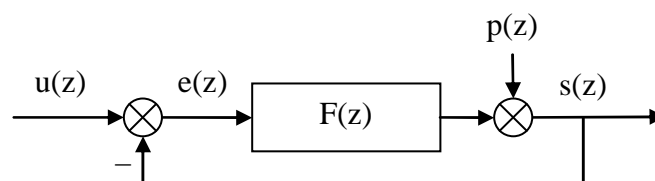


Figure 20 : système discret, notations.

Par défaut on considère une contre-réaction négative et une perturbation additive sur la sortie. La fonction de transfert  $F$  est appelée *fonction de transfert en boucle ouverte*. Comme pour les systèmes continus la *fonction de transfert en boucle fermée* est obtenue simplement à partir de  $F$  (58).

$$F(z) = \frac{s(z)}{e(z)} = \frac{B(z^{-1})}{A(z^{-1})} \quad (56)$$

$$G(z) = \frac{s(z)}{u(z)} = \frac{B(z^{-1})}{A(z^{-1}) + B(z^{-1})}$$

## 6.2 Erreur statique

L'erreur statique du système en boucle fermée, comme pour le système en boucle ouverte, est simplement le rapport des coefficients.

$$G_0 = \frac{\sum_{i=0}^{i=m} b_i}{1 + \sum_{i=0}^{i=m} b_i + \sum_{i=1}^{i=n} a_i} \quad (57)$$

Si l'on souhaite que l'asservissement confère un gain statique unitaire au système en boucle fermée (c'est-à-dire que l'on force la sortie  $s(k)$  à tendre vers la valeur de consigne  $u(k)$ ) il faut que ce rapport de coefficients soit égal à un. On montre facilement que cela est obtenu si la fonction de transfert  $F(z)$  comporte un intégrateur :

$$F(z) = \frac{s(z)}{e(z)} = \frac{1}{1 - z^{-1}} \cdot \frac{B(z)}{A'(z)} \quad (58)$$

## 6.3 Rejet de perturbations

La fonction de transfert entre la perturbation  $p(z)$  et la sortie  $y(z)$  est appelée fonction de sensibilité perturbation-sortie, notée  $S_{yp}$ .

$$S_{yp}(z) = \frac{s(z)}{p(z)} = \frac{A(z^{-1})}{A(z^{-1}) + B(z^{-1})} \quad (59)$$

Si l'on souhaite qu'une perturbation de type échelon soit parfaitement rejetée il faut que  $A(1) = 0$ , on retrouve donc la condition (58) à savoir que le système doit comporter une intégration dans la boucle directe. On peut être en outre être amené à spécifier que les perturbations ne soient pas amplifiées dans une certaine bande de fréquence. On rajoute donc une contrainte du type  $|S_{yp}(e^{j2\pi f})| < S_{max}$  pour  $f$  appartenant à un certain domaine de  $[0 \ 0.5]$ .



## 6.4 Stabilité des systèmes à simple gain de rétroaction

Les critères de Routh et de Jury (donnés dans le chapitre 3.2.7 p13) ou le calcul numérique des pôles permettent de déterminer la stabilité du système en boucle fermée si l'on connaît sa fonction de transfert.

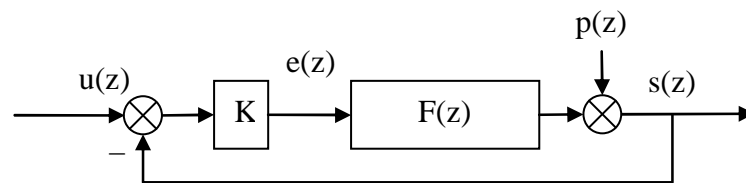


Figure 21 : système bouclé par un simple gain.

On s'intéresse ici au cas particulier où l'asservissement est réalisé par un simple gain de rétroaction  $K$  (Figure 21). On cherche la valeur de  $K$  qui amène le système en limite de stabilité. L'outil adapté ici est le lieu d'Evans ou lieu des racines (root locus) qui permet de visualiser l'évolution des pôles du système bouclé lorsque le gain  $K$  varie.

Prenons l'exemple d'un système comportant trois pôles et deux zéros. L'exemple est donné sous forme de script Matlab :

### Matlab

```
% Définition d'un système discret :
sys=zpk([-2.037 -0.1406],[0.6065 0.3679 0.3679],1,1)
%Tracé du lieu des racines
rlocus(sys);
```

Le script précédent produit le tracé de la Figure 22. On pourrait y lire que la limite de stabilité est obtenue pour  $K=0.2$  environ (détail de la Figure 23).

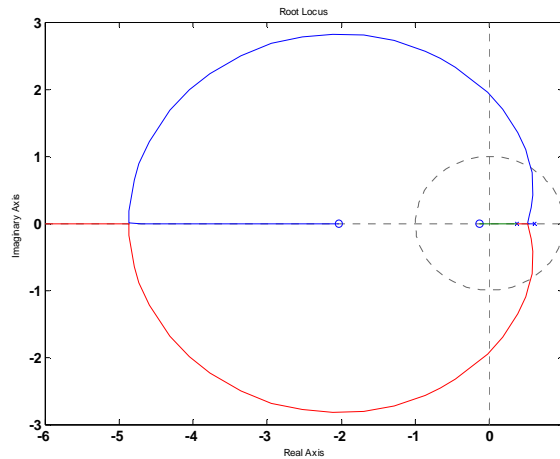


Figure 22 : lieu des racines

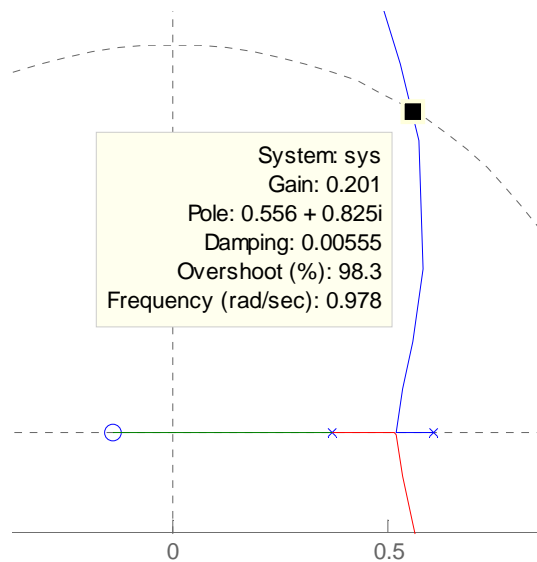


Figure 23 : lieu des racines : détail de la limite de stabilité

On voit sur la Figure 22 que le lieu des racines d'un système échantillonné ressemble à celui d'un système continu :

- on commence par tracer les pôles (x) et les zéros (o) du système en boucle ouverte.
- les pôles du système en boucle fermée décrivent un lieu qui part des pôles (pour  $K = 0$ ) et arrive aux zéros (pour  $K = \infty$ )
- lorsque il y a  $n-m$  plus de pôles que de zéros le système comporte  $n-m$  branches asymptotiques.

On ne décrira pas ici les diverses propriétés du lieu des racines qui permettent de le tracer à la main puisque il est facile à obtenir avec un ordinateur.

Il peut aussi être utilisé pour régler "à la main" le gain permettant d'obtenir un ou plusieurs modes dominants ayant certaines propriétés de fréquence et d'amortissement. On voit sur la Figure 24 que le gain 0.017 permet d'obtenir un pôle double d'amortissement 0.706 et de fréquence 0.65 rad/s et un pôle simple de fréquence 1.63 rad/s.

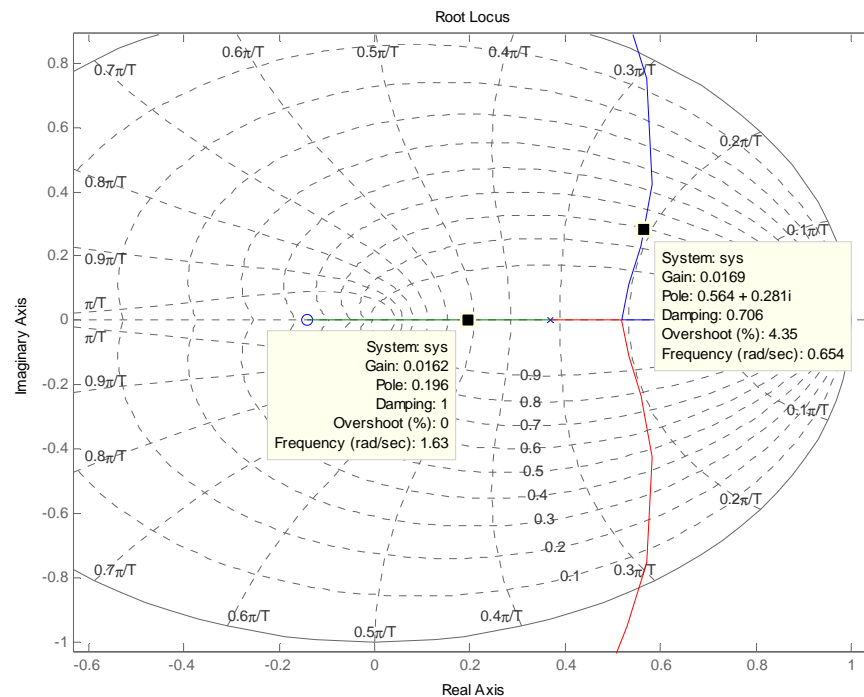


Figure 24 : utilisation du lieu des racines pour régler le gain de la boucle fermée

L'outil rltool de Matlab permet de nombreuses manipulations graphiques de ce type.

## 6.5 Analyse des systèmes bouclés dans le domaine fréquentiel : stabilité

Comme dans le cas continu il est possible de prévoir des propriétés, en particulier la stabilité, d'un système en boucle fermée à partir de l'examen de la fonction de transfert en boucle ouverte.

On ne redémontrera pas ici le théorème de Nyquist mais on se contente de rappeler l'énoncé.

On rappelle que le *lieu de Nyquist* est le lieu de  $F(j2\pi f)$  dans le plan complexe pour  $f$  variant de 0 à  $+\infty$ . A noter que le lieu pour la zone  $[0, 0.5]$  est symétrique de celui pour la zone  $[0.5, 1]$ . Pour cette raison on ne représente généralement que le lieu pour  $f$  appartenant à  $[0, 0.5]$ .

### 6.5.1 Théorème de Nyquist dans le cas où le système est stable en boucle ouverte (critère du revers)

Le système est stable en boucle fermée si le lieu de Nyquist de la fonction de transfert en boucle ouverte parcouru dans le sens des fréquences croissantes de 0 à 0.5 laisse le point critique (-1,0) à gauche.

La Figure 25 représente par exemple le lieu de Nyquist d'une fonction de transfert en boucle ouverte qui sera stable en boucle fermée.

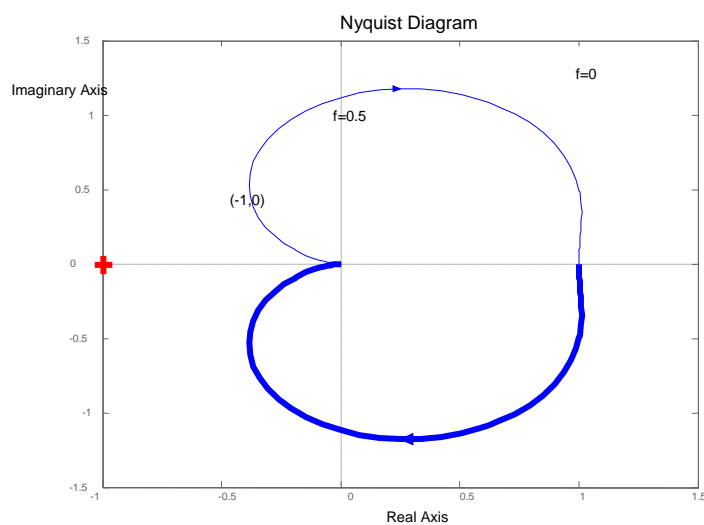


Figure 25 : lieu de Nyquist d'un système stable

### 6.5.2 Théorème de Nyquist dans le cas où le système est instable en boucle ouverte

Si le système en boucle ouverte comporte  $N$  pôles instables il sera stable en boucle fermée si le lieu de Nyquist de la fonction de transfert en boucle ouverte parcouru dans le sens des fréquences croissantes de  $-0.5$  à  $0.5$  entoure le point critique  $(-1,0)$   $N$  fois dans le sens des aiguilles d'une montre..

### 6.5.3 Interprétations du lieu de Nyquist

Comme dans le cas continu, la distance à laquelle on passe du point critique renseigne sur la robustesse du système bouclé. La distance sera exprimée de différente manière (Figure 26) :

- marge de module  $\Delta M$  : plus petite distance entre le lieu des racines et le point critique,

- marge de gain  $\Delta G$
- marge de phase  $\Delta P$

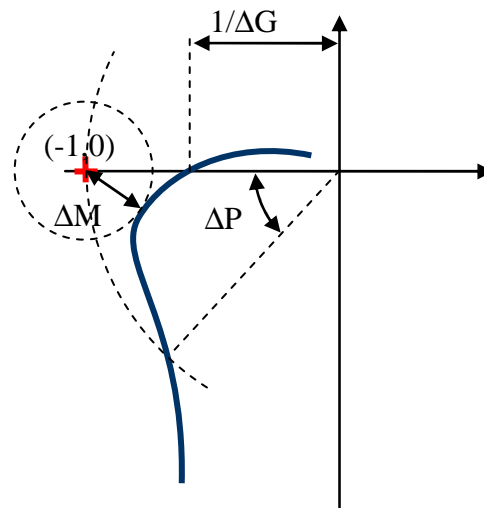


Figure 26 : Marges de stabilité

Les valeurs typiques de ces marges sont données ci-après :

- $\Delta G \geq 6\text{dB}$
- $\Delta M \geq -6\text{dB}$
- $30^\circ \leq \Delta P \leq 60^\circ$

La distance entre le point critique et le lieu des racines est la norme de  $1+F(j2\pi f)$ . On peut donc exprimer la marge de module en fonction de la fonction de sensibilité :

$$\begin{aligned} \Delta M &= \min |1 + F(j \cdot \omega)|_{\omega \in [0, \pi]} \\ \Delta M &= \min \left| 1 + \frac{B(j \cdot \omega)}{A(j \cdot \omega)} \right|_{\omega \in [0, \pi]} = \max \left| \frac{A(j \cdot \omega)}{A(j \cdot \omega) + B(j \cdot \omega)} \right|_{\omega \in [0, \pi]} \quad (60) \\ \Delta M &= \max |S_{yp}(j \cdot \omega)|_{\omega \in [0, \pi]}^{-1} \\ \Delta M_{\text{dB}} &= -\max |S_{yp}(j \cdot \omega)|_{\text{dB}} \end{aligned}$$

Si l'on trace l'évolution de  $S_{yp}$  en fonction de  $f$  on obtient donc la valeur de la marge de module.



# 7. METHODES DE REGULATION DES SYSTEMES ECHANTILLONES

## 7.1 Introduction

Les bases théoriques permettant de concevoir le correcteur pour un système échantillonné ont été posées dans les chapitres précédents. Ce chapitre fait le bilan des connaissances acquises et donne quelques solutions pour calculer des correcteurs à partir d'un cahier des charges précis.

Les systèmes que l'on cherche à commander sont généralement, par nature, continus. La réalisation d'un correcteur par un ordinateur implique que l'on génère une équation de commande discrète (généralement une équation récurrente). L'obtention de cette équation de commande peut se faire de deux manières :

- la première méthode consiste à d'abord synthétiser un correcteur continu (en utilisant les outils classiques de l'automatique et la simulation) et discrétiser ensuite ce correcteur. C'est l'objet du chapitre 7.2.2.
- la seconde méthode consiste à tenir la représentation discrète du système continu associé à son interface numérique (convertisseurs analogique-numérique et numérique-analogique) comme expliqué dans le chapitre 4.5 puis synthétiser directement le correcteur discret. Cela sera expliqué dans le chapitre 7.3 et suivants.

Ce choix peut être résumé par la figure suivante :

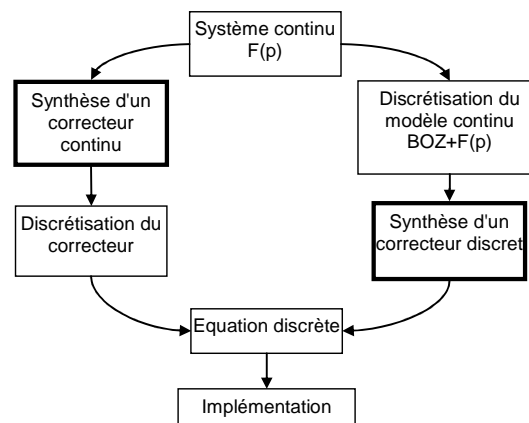


Figure 27 : Réalisation d'un correcteur discret.

Au-delà des affaires de goût, le choix de la méthode sera souvent dicté par l'application. En effet, la première méthode est d'autant meilleure que le correcteur discrétisé approche correctement le correcteur continu. Cela implique que le temps d'échantillonnage soit très faible par rapport aux constantes de temps du système

bouclé. Dans le cas où les contraintes temps réel sont fortes (capacités de calcul partagées par exemple) il sera préférable de synthétiser un correcteur discret.

## 7.2 Structure d'un système commandé par ordinateur, hypothèses

Par la suite on se ramènera toujours à un système tel que celui représenté sur la Figure 28 en cherchant un correcteur  $K(z)$  linéaire. On montre que quel que soit le correcteur linéaire on peut se ramener à la forme représenté sur la Figure 29 (forme RST).

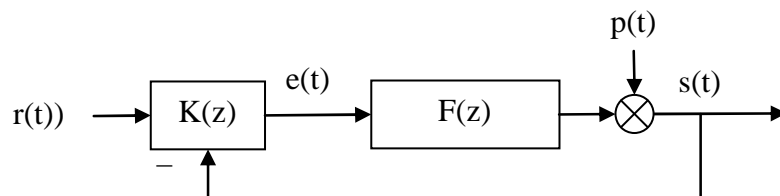


Figure 28 : système bouclé, forme générale.

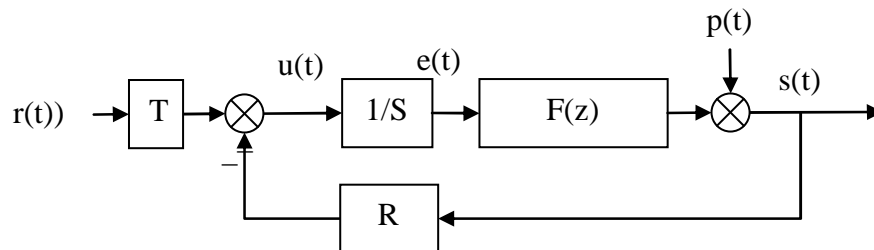


Figure 29 : système bouclé, forme canonique.

### 7.2.1 Discrétisation, choix de la période d'échantillonnage

Du point de vue du ordinateur, l'extérieur (le procédé) est représenté par la fonction de transfert  $F(z)$ . L'hypothèse importante est que la commande (sortie du ordinateur) et la mesure (entrée du ordinateur) sont échantillonnées de manière synchrone à la même période  $T_e$ .

Le choix judicieux de la fréquence d'échantillonnage est très important. En pratique on choisit cette fréquence en fonction de la bande passante du système en **boucle fermée** :

$$f_e = (6 \text{ à } 25) f^{BF} \quad (61)$$



### 7.2.2 Discrétisation d'un correcteur continu

La première méthode pour obtenir les équations d'un correcteur numérique est de faire sa synthèse avec les outils de l'automatique continue puis de discrétiser la fonction de transfert du correcteur continu obtenu (voir pour cela le chapitre 4.2 *Echantillonnage des systèmes continus*).

On va voir par la suite que la synthèse directe d'un correcteur discret offre plus de degré de liberté et permet en général de remplir un cahier des charges plus complet.

## 7.3 Régulateur PID numérique

Le régulateur le plus populaire est le régulateur PID. Nous allons voir dans un premier temps comment on génère un régulateur PID à partir des équations bien connues du régulateur PID continu et comment ces équations peuvent être généralisées pour obtenir plus de degrés de liberté pour le réglage.

### 7.3.1 Cas 1

Un régulateur PID continu a une fonction de transfert du type :

$$K_{\text{PID}}(s) = K \cdot \left( 1 + \frac{1}{T_i \cdot s} + \frac{T_d \cdot s}{1 + \frac{T_d}{N} \cdot s} \right) \quad (62)$$

La discrétisation de ce correcteur continu par la méthode d'Euler (Tableau 4 page 31) se fait en remplaçant la variable de Laplace  $s$  par le terme  $\frac{1-q^{-1}}{q^{-1} \cdot T_e}$ . On obtient :

$$K_{\text{PID}}(q) = K \cdot \left( 1 + \frac{1}{T_i \cdot \frac{1-q^{-1}}{T_e}} + \frac{T_d \cdot \frac{1-q^{-1}}{T_e}}{1 + \frac{T_d}{N} \cdot \frac{1-q^{-1}}{T_e}} \right) \quad (63)$$

En réarrangeant on obtient la forme générale RST (Figure 30):

$$K_{PID}(q) = K \cdot \frac{r_0 + r_1 \cdot q^{-1} + r_2 \cdot q^{-2}}{(1 - q^{-1}) \cdot (1 + s_1' \cdot q^{-1})}$$

$$\begin{cases} r_0 = K \cdot \left( 1 + \frac{T_e}{T_i} + \frac{N \cdot T_d}{T_d + N \cdot T_e} \right) \\ r_1 = -K \cdot \left( 1 + \frac{T_d}{T_d + N \cdot T_e} + \frac{T_e}{T_i} \cdot \frac{T_d}{T_d + N \cdot T_e} + \frac{N \cdot T_d}{T_d + N \cdot T_e} \right) \\ r_2 = K \cdot \left( \frac{T_d}{T_d + N \cdot T_e} \right) \\ s_1' = \frac{T_d}{T_d + N \cdot T_e} \end{cases} \quad (64)$$

Comme on le voit le correcteur obtenu comporte autant de degrés de liberté que le correcteur PID analogique. On reconnaît au dénominateur le terme  $(1 - q^{-1})$  en facteur qui assure l'effet d'intégration.

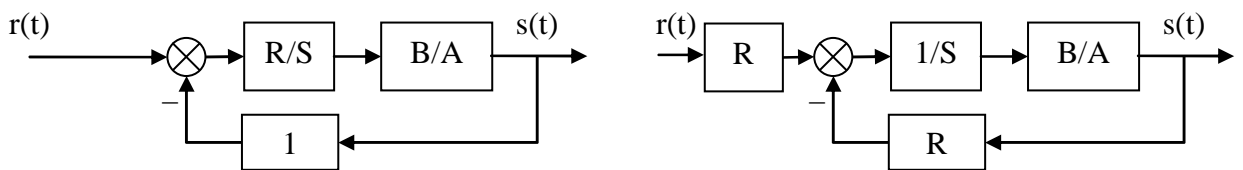


Figure 30 : PID sous forme RST

La fonction de transfert en boucle fermée obtenue est donnée par l'équation suivante :

$$H_{BF} = \frac{B \cdot R}{A \cdot S + B \cdot R} = \frac{B \cdot R}{P} \quad (65)$$

De manière générale le polynôme caractéristique  $P$  fixe les performances visées en boucle fermée. Le calcul de  $R$  et  $S$ ,  $P$  étant fixé et  $A$  et  $B$  étant connus, est réalisé en résolvant l'équation de Bézout  $P = A \cdot S + B \cdot R$ . (voir 9.1 Annexe 1 : Résolution de l'équation polynomiale de Bezout)

On voit que le numérateur  $B \cdot R$  est imposé une fois que  $R$  et  $S$  sont trouvés. Le régulateur introduit donc naturellement des zéros dans la boucle fermée.

Traisons l'exemple entièrement dans le cas où le système est modélisé comme un système du deuxième ordre avec faible retard. La fonction de transfert du procédé est donnée par :

$$H_{BO} = \frac{B}{A} = \frac{b_1 \cdot q^{-1} + b_2 \cdot q^{-2}}{1 + a_1 \cdot q^{-1} + a_2 \cdot q^{-2}} \quad (66)$$

Le correcteur PID rajoutant deux pôles, le polynôme caractéristique en boucle fermée spécifié doit posséder quatre pôles. Le polynôme caractéristique en boucle fermée doit donc être de la forme :

$$\begin{aligned} P &= 1 + p_1 \cdot q^{-1} + p_2 \cdot q^{-2} + p_2 \cdot q^{-3} + p_2 \cdot q^{-4} \\ P &= (1 + p_1' \cdot q^{-1} + p_2' \cdot q^{-2}) \cdot (1 + \alpha_1 \cdot q^{-1}) \cdot (1 + \alpha_2 \cdot q^{-1}) \end{aligned} \quad (67)$$

Et il doit être identifié au produit  $AS + BR$  avec  $R$  et  $S$  ayant les formes données par (64) :

$$\begin{aligned} &(1 + p_1' \cdot q^{-1} + p_2' \cdot q^{-2}) \cdot (1 + \alpha_1 \cdot q^{-1}) \cdot (1 + \alpha_2 \cdot q^{-1}) \\ &= (1 + a_1 \cdot q^{-1} + a_2 \cdot q^{-2}) \cdot (1 - q^{-1}) \cdot (1 + s_1' \cdot q^{-1}) \\ &+ (b_1 \cdot q^{-1} + b_2 \cdot q^{-2}) \cdot (r_0 + r_1 \cdot q^{-1} + r_2 \cdot q^{-2}) \end{aligned} \quad (68)$$

Les coefficients  $p_1'$  et  $p_2'$  définissent un modèle du second ordre aux caractéristiques (pulsation propre  $\omega_0$  et amortissement  $\sigma$  choisis et coefficients déduits conformément au Tableau 3 page 25). Les coefficients  $\alpha_1$  et  $\alpha_2$  correspondent à des pôles auxiliaires aperiodiques. Les choix typiques pour ces coefficients sont :

$$-0.05 < \alpha_1, \alpha_2 < -0.5 \quad (69)$$

On montre que la résolution de (68) est réalisée par une inversion matricielle simple :

$$M \cdot x = p \quad (70)$$

avec

$$\begin{cases} x^T = [1 & s_1' & r_0 & r_1 & r_2] \\ p^T = [1 & p_1 & p_2 & p_3 & p_4] \end{cases}$$

et

$$M = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ a_1' & 1 & b_1 & 0 & 0 \\ a_2' & a_1' & b_2 & b_1 & 0 \\ a_3' & a_2' & 0 & b_2 & b_1 \\ 0 & a_3' & 0 & 0 & b_2 \end{pmatrix} \quad (71)$$

On montre que M est inversible à la condition que A et B soient premiers entre eux, c'est-à-dire que A et B n'aient pas de pôles et de zéros en commun.

(Le problème peut aussi être résolu en utilisant la résolution de l'équation de Bézout).

### 7.3.2 Cas 2

Comme on l'a vu le correcteur PI numérique directement transposé du correcteur PID analogique décrit dans le chapitre 7.3.1 a l'inconvénient de ramener des zéros au dénominateur de la fonction de transfert en boucle fermée (équation 65). Il est possible, grâce à la flexibilité du régulateur numérique d'imposer au correcteur de ne pas ajouter de zéros en boucle fermée. La fonction de transfert désirée en boucle fermée sera donc de la forme :

$$H_{BF} = \frac{P(1)}{B(1)} \cdot \frac{B}{P} \quad (72)$$

Le facteur  $P(1)/B(1)$  garantit le gain statique unitaire du système en boucle fermée (voir page 12). Si le régulateur a la forme générale RST alors les polynômes R S et T doivent vérifier :

$$H_{BF} = \frac{B \cdot T}{A \cdot S + B \cdot R} = \frac{P(1) B}{B(1) P} \quad (73)$$

S et R sont obtenus par la résolution de la même équation que précédemment (68). Le polynôme T est obtenu par simple identification :  $T = R(1)$ .

La structure, sous forme de schéma-bloc est donnée ci-dessous :

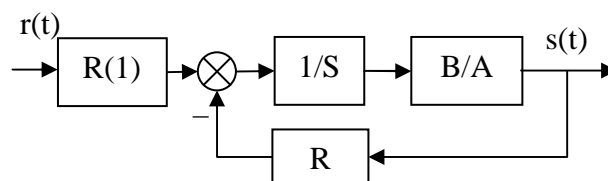


Figure 31 : PID2 sous forme RST

On montre que cette structure de PID, ramené en continu, correspond au PID canonique pour l'action intégrale. L'action proportionnelle et dérivée n'est appliquée que sur le signal y.

La structure du PID2 par rapport au PID1 a pour avantage de ne pas ramener de zéros à la boucle fermée. Cela se traduit par un dépassement moindre en transitoire lorsqu'on applique un échelon de commande. Par contre les deux systèmes réagissent de la même manière à un échelon de perturbation.

## 7.4 Placement de pôles

Les deux structures de PID décrites dans le chapitre précédent sont deux cas particulier d'une stratégie plus générale de placement de pôles.

La synthèse est plus générale que les PID présentés précédemment puisqu'il n'est pas nécessaire de faire d'hypothèses sur le degré de A et B, sur le retard, ni sur la présence ou non de zéros stables ou instables dans le procédé.

La structure du correcteur est la structure type RST telle que présentée dans la Figure 29. Pour la suite des calculs on posera pour la fonction de transfert les conventions suivantes :

$$H_{BO} = \frac{q^{-d} \cdot B(q^{-1})}{A(q^{-1})} \quad (74)$$

$$\begin{cases} A(q^{-1}) = 1 + a_1 \cdot q^{-1} + \dots + a_{n_A} \cdot q^{-n_A} \\ B(q^{-1}) = b_1 \cdot q^{-1} + \dots + b_{n_B} \cdot q^{-n_B} = q^{-1} \cdot B^*(q^{-1}) \end{cases}$$

Compte tenu de la forme RST du régulateur la fonction de transfert en boucle fermée vérifie :

$$H_{BF} = \frac{q^{-d} \cdot B(q^{-1}) \cdot T(q^{-1})}{P(q^{-1})} \quad (75)$$

$$P(q^{-1}) = A(q^{-1}) \cdot S(q^{-1}) + q^{-d} \cdot B(q^{-1}) \cdot R(q^{-1}) = 1 + p_1 \cdot q^{-1} + \dots + p_{n_P} \cdot q^{-n_P}$$

Le polynôme  $P(q^{-1})$  définit donc les pôles de la fonction de transfert en boucle fermée et traduit donc le comportement en *poursuite*.

De la même manière la fonction de transfert entre une perturbation et la sortie (fonction de sensibilité perturbation-sortie) est donnée par :

$$S_{yp} = \frac{A(q^{-1}) \cdot S(q^{-1})}{P(q^{-1})} \quad (76)$$

Ce même polynôme traduit donc aussi le comportement en *régulation*.

Les consignes en poursuite et en régulation peuvent être différentes comme cela est représenté par exemple sur la Figure 32.

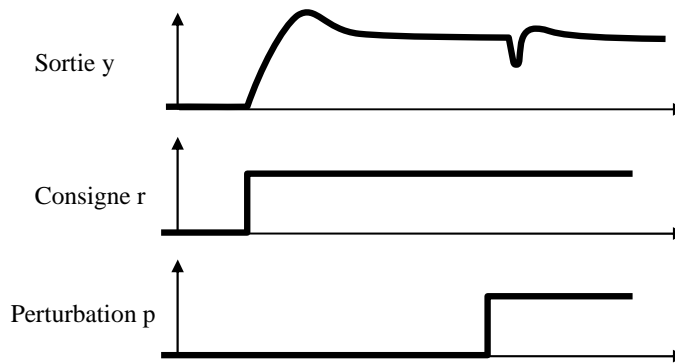


Figure 32 : Performances différentes en régulation et en poursuite

### 7.4.1 Choix des pôles en boucle fermée

De manière générale le comportement en boucle fermée est spécifié sous la forme d'une pulsation propre  $\omega_0$  et d'un amortissement  $\sigma$ , donc une fonction de transfert du second ordre. La spécification étant généralement donnée en continu il est nécessaire de traduire la valeur désirée de  $\omega_0$  et  $\sigma$  en un polynôme  $P_D(q^{-1})$  par discrétisation de la fonction de transfert continue (chapitre 4.5 page 29). Ce polynôme  $P_D$  traduit le comportement dominant du système bouclé.

Si l'on souhaite introduire un effet de filtrage supplémentaire dans certaines bandes de fréquence (réduction des bruits de mesure, adoucissement de la commande, robustesse) cela est décrit par un polynôme  $P_F$  comportant des pôles auxiliaires, plus rapides que les pôles de  $P_D$ .

Le polynôme caractéristique en boucle fermée est donc constitué de pôles dominants et de pôles auxiliaires :

$$P(q^{-1}) = P_D(q^{-1}) \cdot P_F(q^{-1}) \quad (77)$$

### 7.4.2 Régulation : calcul de R et S

Le polynôme  $P(q^{-1})$  étant fixé il s'agit de trouver R et S vérifiant :

$$P(q^{-1}) = A(q^{-1}) \cdot S(q^{-1}) + q^{-d} \cdot B(q^{-1}) \cdot R(q^{-1}) \quad (78)$$

Comme déjà vu au chapitre 7.3.1 on obtient une équation de Bézout qu'il s'agit de résoudre. (Voir aussi 9.1 Annexe 1 : Résolution de l'équation polynomiale de Bezout)

Il est parfois nécessaire de spécifier R et S avec des parties fixes  $H_R$  et  $H_S$ . Dans ce cas l'équation (78) est modifiée et on l'écrit sous la forme:

$$P(q^{-1}) = (A(q^{-1}) \cdot H_S(q^{-1})) \cdot S'(q^{-1}) + (q^{-d} \cdot B(q^{-1}) \cdot H_R(q^{-1})) \cdot R'(q^{-1}) \quad (79)$$

Les inconnues que l'on souhaite trouver sont  $S'$  et  $R'$ . Ceci est nécessaire dans les cas particuliers suivants :

- spécification d'erreur statique nulle,
- rejet d'une perturbation harmonique,
- blocage d'un signal,

Détaillons maintenant ces cas particuliers.

### Spécification d'une erreur statique nulle

On retrouve la nécessité d'avoir un terme intégrateur dans la boucle directe. La spécification se retrouve donc sur  $S$  :

$$S(q^{-1}) = (1 - q^{-1}) \cdot S'(q^{-1}) = H_S(q^{-1}) \cdot S'(q^{-1}) \quad (80)$$

### Rejet d'une perturbation harmonique

Si le système est perturbé par une perturbation harmonique à une fréquence fixe (par exemple 50Hz...) il est possible d'imposer à  $S$  d'avoir un module *nul* ou amorti à cette fréquence. Cela se traduit par la présence dans  $S$  d'une paire de zéros complexe à amortissement nul ou faible à cette fréquence :

$$S(q^{-1}) = (1 + \alpha \cdot q^{-1} + \beta \cdot q^{-2}) \cdot S'(q^{-1}) = H_S(q^{-1}) \cdot S'(q^{-1}) \quad (81)$$

### Blocage d'un signal

Pour certaines application on souhaite que le régulateur "laisse passer" certaines composantes du signal de sortie, c'est-à-dire qu'il ne réagisse pas à ces composantes. Cela se traduit par  $R=0$  à ces fréquences. On retrouve la même forme que précédemment :

$$R(q^{-1}) = (1 + \alpha \cdot q^{-1} + \beta \cdot q^{-2}) \cdot R'(q^{-1}) = H_R(q^{-1}) \cdot R'(q^{-1}) \quad (82)$$

### 7.4.3 Poursuite : calcul de T

Le schéma-bloc de l'asservissement est donné sur la Figure 33. Le terme  $T$  est décomposé en un terme  $T'$  et un terme  $B^*/A^*$ .

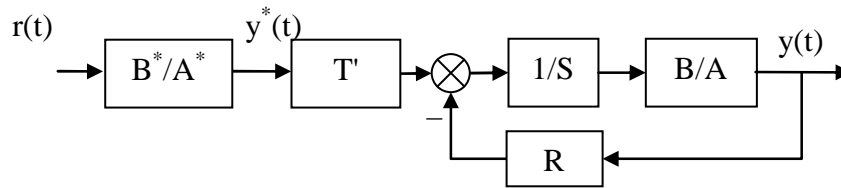


Figure 33 : placement de pôle

L'idée est de régler  $B^*$  et  $A^*$  de telle manière que le signal  $y^*$  ait la dynamique de poursuite désirée (par exemple un second ordre avec pulsation propre et amortissement donné) puis de régler  $T'$  de telle manière que le transfert entre  $y^*$  et  $y$  soit le plus "transparent" possible.

### Réglage de $T'$

Le transfert entre  $y^*$  et  $y$  est :

$$S_{yy^*}(q^{-1}) = T'(q^{-1}) \frac{q^{-d} \cdot B(q^{-1})}{P(q^{-1})} \quad (83)$$

Pour assurer le transfert le plus "transparent" possible il faut donc choisir

$$T'(q^{-1}) = \frac{P(q^{-1})}{B(1)} \quad (84)$$

Remarques :

- Si  $B(1)=0$  alors on remplace  $B(1)$  par 1 dans (84)
- De même dans (84)  $P(q^{-1})$  peut être remplacé par  $P_D$  correspondant à la dynamique dominante.

### Réglage de $B^*$ et $A^*$

$B^*$  et  $A^*$  sont choisis de telle manière que la fonction de transfert  $B^*/A^*$  corresponde à la dynamique de poursuite souhaitée.

## 7.5 Poursuite et régulation à objectifs indépendants

Cette méthode ressemble à la méthode de placement de pôles à ceci près que l'on va effectuer une simplification des zéros de  $B^*(q^{-1})$  : cela ne s'applique donc qu'aux systèmes avec des pôles stables. Plus précisément il faut qu'ils aient un amortissement suffisant ( $> 0,2$ ) de manière à ne pas être trop près du cercle unité.

La structure de l'asservissement est toujours celle donnée sur la Figure 33.



**Choix de P**

$P(q^{-1})$  est choisi de manière à simplifier  $B^*(q^{-1})$  (rappel : la forme de B est donnée dans l'équation 74). Comme indiqué dans le chapitre 7.4.2 page 60 il s'agit donc d'imposer les zéros de  $B(q^{-1})$  dans  $H_R(q^{-1})$  et  $H_S(q^{-1})$ .

**Choix de T**

Si l'on choisit  $T(q^{-1}) = P(q^{-1})$  alors le transfert entre  $y^*$  et  $y$  est :

$$S_{yy^*}(q^{-1}) = q^{-(d+1)} \quad (85)$$

**Réglage de  $B^*$  et  $A^*$** 

$B^*$  et  $A^*$  sont choisis de telle manière que la fonction de transfert  $B^*/A^*$  corresponde à la dynamique de poursuite souhaitée.



## 8. TECHNOLOGIE DE LA COMMANDE PAR ORDINATEUR

Du point de vue de l'automaticien le ordinateur est relié au système physique à piloter par un certain nombre d'intermédiaires plus ou moins gênants :

- les interfaces analogiques et numériques,
- le transport de l'information par bus et protocole,
- l'utilisation éventuelle de capteurs ou d'actionneurs « intelligents ».

Il ne s'agit pas ici de faire un exposé complet sur ces trois aspects mais de poser les problèmes qui se posent à l'automaticien lorsque le système qu'il doit commander est perturbé par ces éléments.

Enfin ce chapitre se termine sur les notions de « Hardware In the Loop » et de « prototypage rapide » qui sont partie intégrante des processus de réalisation de systèmes de contrôle.

### 8.1 Calculateurs et interfaces numériques / analogiques

Le premier problème à prendre en compte est la quantification de l'information. Du point de vue du ordinateur d'une part et du point de vue des convertisseurs d'autre part. Dans un premier temps, considérons les aspects performance de calcul lié à la nature du codage des nombres dans le ordinateur.

#### Les performances liées au types de variables

Prenons l'exemple où le concepteur de la loi de commande souhaite réaliser un correcteur de type proportionnel. Entre l'acquisition des mesures et l'écriture de la consigne il y aura la réalisation d'une de commande du type :

$$y = K \cdot (\text{consigne} - \text{mesure}) \quad (86)$$

En utilisant un langage de programmation évolué (par exemple le C) la ligne de code sera la transcription littérale de (86). Le programmeur devra décider de la définition et du type des variables, principalement du nombre d'octets utilisés pour stocker ses variables. Le Tableau 5 résume par exemple les types utilisés en langage C (norme ANSI)

Tableau 5 : Types ANSI C

Types entiers	char	8 bits
	short int	16 bits
	long int	32 bits
Types flottants (mantisse et exposant)	float	32 bits
	double	64 bits
	longdouble	128 bits

Une fois le code écrit en C l'exécution du programme sur un processeur cible ne peut se faire que par l'intermédiaire d'une compilation du programme. A chaque processeur est donc adapté une version de compilateur.

Les performances du programme (sa rapidité d'exécution) dépendront donc en partie de la qualité du compilateur. (Par exemple, une multiplication s'effectue t'elle bêtement par un décalage de tous les bits ou bien des solutions astucieuses sont-elles implémentées pour arrêter le décalage selon le contexte ?)

Le principal déterminant de la performance est cependant le processeur : son type (8 bits, 16 bits ou 64 bits) et son cadencement (en MHz ou GHz) sont les deux plus importants paramètres. Coder les variables sur 64 bits (double) dans le programme coûtera cher en temps de calcul si le processeur est un 8 bits. Le type de mémoire et leurs temps d'accès peuvent aussi être déterminants. Enfin, certains processeurs disposent d'un processeur spécifiquement dédié aux opérations mathématiques (coprocesseur mathématique).

Il n'est pas possible de dire exactement, pour tel type de compilateur associé à tel type de processeur, le temps que prendra par exemple une multiplication. En pratique on effectue une série de tests types (benchmark) et on affiche un temps moyen d'exécution.

En résumé, on peut être tenté, pour ne pas s'embêter, de coder toutes les variables en virgule flottante. Mais il faut bien savoir que cela a un coût en terme de temps de calcul, surtout si le processeur n'est pas adapté. Pour fixer les idées on peut dire qu'il y a un rapport 20 entre une addition réalisée en "short" et une addition réalisée en virgule flottante.

Enfin, l'utilisation de fonctions mathématiques (sinus, exponentielles, etc...) peut aussi avoir un coût très lourd si le travail n'est pas réalisé par un coprocesseur mathématique dédié. Le compilateur et sa librairie mathématique joue là encore un grand rôle.

### La quantification imposée par le processeur

Au-delà de l'aspect performance et temps de calcul, la quantification de l'information doit parfois aussi être prise en compte. Rappelons qu'un entier codé sur N bits peut prendre seulement  $2^N - 1$  valeurs différentes. Avant d'effectuer les calculs mathématiques dans le corps du programme il est donc judicieux d'appliquer un

facteur d'échelle de manière à utiliser l'amplitude maximale possible (entre 0 et  $2^N - 1$ ).

### **La quantification imposée par les conversions analogique et numérique**

Les convertisseurs analogique-numérique et numérique-analogique sont principalement caractérisés par leur amplitude de signal analogique d'entrée (généralement en volts, par exemple +/- 10V) et par leur résolution en bits (8 ou 10 bits pour les plus courants, jusqu'à 16 bits pour les convertisseurs haut de gamme).

Prenons l'exemple d'un capteur gyromètre transformant une vitesse de rotation en tension. Compte tenu de l'application le signal utile varie entre 2,4V et 2,6V. Si le signal mesuré est branché directement sur un convertisseur 0-10V 10bits alors un problème de conditionnement du signal se pose. En effet la quantification est de  $\frac{10V}{2^{10}-1} = 0,098V$ . Les 0,2V de signal utile seront donc discrétisés en  $\frac{0,2}{0,098} = 20$  intervalles seulement !

Si la discrétisation du signal utile est trop grossière cela peut se traduire par une diminution des performances de l'asservissement, voire une instabilité. Le système n'est en effet plus linéaire et il n'existe pas d'outils théoriques permettant de prévoir les performances du système bouclé.

## **8.2 Transport de l'information : les bus et les protocoles**

Pas question de faire ni même d'ébaucher un cours sur les bus et les protocoles ici. Mais on va quand même donner quelques indications. L'automaticien a l'habitude de simplifier le problème qu'il se pose en dessinant un schéma simplifié de son problème d'asservissement (du type de la Figure 20 page 45) pour pouvoir appliquer les théories de la commande. Cependant il arrive bien souvent que l'information transite à l'intérieur du système, soit entre différents ordinateurs effectuant différentes parties du travail soit entre ordinateur et périphériques, par l'intermédiaire de bus informatique. A chaque type de bus correspond un ou plusieurs protocoles qui peuvent poser des problèmes pour l'automaticien :

- les données sont-elles discrétisées ? (le protocole peut par exemple ne permettre de transmettre que des paquets de données de 16 bits),
- le bus induit-t'il un retard ? (le composant mettant en forme les données du ordinateur pour les publier sur le bus a certaines performances qui peuvent induire un délai)
- le bus est-il déterministe ? (Certains bus ne garantissent pas que les données arriveront en un temps donné),
- le bus est-il sûr ? (Certains bus ne garantissent pas non plus que les données arriveront bien)

### **8.3 Capteurs et actionneurs "intelligents", "smart sensors"**

De plus en plus souvent les capteurs et les actionneurs sont associés à leur électronique de puissance, électronique de commande et de conditionnement de signal. L'interface avec le capteur ou l'actionneur se fait donc via un protocole de communication (par exemple via un bus RS232).

L'intérêt est de véhiculer non pas des signaux analogiques, tensions ou courants, mais des signaux numériques beaucoup moins sensibles aux éventuelles perturbations.

On retrouve donc, entre le capteur ou l'actionneur et le ordinateur, la présence d'un bus et d'un protocole de communication.

### **8.4 La philosophie "hardware in the loop"**

Les systèmes embarqués sont de plus en plus complexes et touchent des domaines de plus en plus variés : automobile, satellites, aéronautique, UAV, systèmes d'arme, sous-marins, systèmes, etc... Les systèmes à piloter ainsi que les systèmes de commande sont de plus en plus complexes.

Les techniques de simulation « Hardware In the Loop » (HIL) sont de plus en plus utilisées dans le développement et dans le test de ces systèmes embarqués. L'idée est de développer entièrement le système de contrôle sur une plateforme de test en utilisant le maximum de composants réels et en ne remplaçant que le minimum par la simulation. Par exemple en aéronautique l'« Iron Bird » regroupe dans un hangar tous les systèmes réels de l'avion en cours de développement. Seule la mécanique du vol est simulée : les sorties des actionneurs sont mesurées, rentrées dans un modèle de simulation numérique, puis les mesures effectuées normalement par les capteurs sont calculées et restituées par le modèle.

L'intérêt est de diminuer de manière importante les coûts et les délais en économisant le plus grand nombre possible de tests sur le système réel en situation. Lors des premiers tests réels les techniques « Hardware In the Loop » permettent d'avoir effectué la plus grande part du développement des lois de commande.

### **8.5 Le prototypage rapide en automatique**

Un peu dans le même ordre d'idée que le « Hardware In the Loop » le prototypage rapide permet de réaliser la conception des lois de commande en simulation puis de transporter rapidement (en un simple click !) le code du contrôleur simulé vers la cible temps réel implémentée sur le système réel.

La société Mathworks [9] investit massivement dans ce domaine en ayant enrichi son logiciel de calcul scientifique (Matlab) avec un logiciel de simulation graphique très évolué (Simulink) puis en développant des outils de prototypage rapide associés (Real Time Workshop, Embedded Target For...).

## 9. ANNEXE

### 9.1 Annexe 1 : Résolution de l'équation polynomiale de Bezout

Il s'agit de trouver R et S vérifiant :

$$P(q^{-1}) = A(q^{-1}).S(q^{-1}) + B(q^{-1}).R(q^{-1})$$

Où P, A et B sont connus.

Dans le cas où P, A, B, R et S sont des nombres entiers la solution de l'algorithme est donnée par l'algorithme de division Euclidienne étendu [8].

Dans le cas où P, A, B, R et S sont des polynômes, l'algorithme est donné ci-dessous, programmé en matlab ([1]).

```
function [Rp,Sp,nrp,nsp]=bezoutd(A,B,Hs,Hr,P)
%function [Rp,Sp,nrp,nsp]=bezoutd(A,B,Hs,Hr,P)
%solves AHsSp+BHrRp=P by coefficient comparison. Delay and discretization delay need to be integrated in B
%(zeros at the beginning).
%
%inputs:
%A=[a0 a1 ... aNa] ... vector of model denominator coefficients A=a0 + a1z^(-1) + a2z^(-2) +...+ aNaz^(-Na)
%B=[b0 b1 ... bNb] ... vector of model numerator coefficients B=b0 + b1z^(-1) + b2z^(-2) +...+ bNbz^(-Nb)
%Hs=[hs0 hs1 ... hsNhs] ... vector of controller denominator fixed part Hs=hs0 + hs1z^(-1) +...+ hsNhsz^(-Nhs)
%Hr=[hr0 hr1 ... hrNhr] ... vector of controller denominator fixed part Hr=hr0 + hr1z^(-1) +...+ hrNhrz^(-Nhr)
%P=[p0 p1 ... pNp] ... vector of desired polynomial coefficients P=p0 + p1z^(-1) + p2z^(-2) +...+ pNpz^(-Np)
%outputs:
%Rp=[rp0 rp1 rp2 ...] ... vector of coefficients for resulted controller numerator
%Sp=[sp0 sp1 sp2 ...] ... vector of coefficients for resulted controller denominator
%nrp ... order of Rp
%nsp ... order of Sp
%
%written by: J. Langer, I.D. Landau, H. Prochazka
%7th june 2002

PRECISION=1e-16;

D=size(A);
if D(1)>1, A=A'; end;
```

```

D=size(B);
if D(1)>1, B=B'; end;
D=size(Hs);
if D(1)>1, Hs=Hs'; end;
if D(1)==0, Hs=1; end;
D=size(Hr);
if D(1)>1, Hr=Hr'; end;
if D(1)==0, Hr=1;end;
D=size(P);
if D(1)>1, P=P'; end;

na=length(A)-1;
nb=length(B)-1;
np=length(P)-1;
nhs=length(Hs)-1;
nhr=length(Hr)-1;

if (nhs>0), Ah=real(convz(A,Hs)); else Ah=A*Hs; end; % Ah = A * Hs
nah=length(Ah) -1;
if (nhr>0), Bh=real(convz(B,Hr)); else Bh=B*Hr; end;
nbh=length(Bh) -1;
if (np>nah+nbh-1), disp('Bezout error: too many poles');end;

% increase size of P using zeros if necessary
if (np<nah+nbh-1),
%
% set remaining poles onto a circle with radius rmin
P=[P zeros(1,nah+nbh-1-np)];
rootsPdes=roots(P);
nextra=nah+nbh-1-np;
rmin=1e-16;
angle=[0:nextra-1]'/nextra*2*pi;
j=sqrt(-1);
rootsPextra=rmin*exp(j*angle);
P=poly([rootsPdes;rootsPextra]);
np=nah+nbh-1;

end;
P,

nsp=nbh-1;
nrp=nah-1;

%matrix is smaller than vector PD

```



```

if (np>nah+nbh-1),
    disp('Order of model denominator is too low! Add a polynom of higher order to Hs or Hr. ');
end;

% ns=nsp+nhs
% nr=nrp+nhr

M=[];
for j=1:nsp+1,
    V=[];
    if (j>1), V=[V ; zeros(j-1,1)]; end;% zeros in front of Ah
    V=[V ; Ah'];% Ah
    if (j<=nsp), V=[V ; zeros(nsp+1-j,1)]; end;% zeros behind Ah
    if (length(V)~=nah+nbh), disp('bezoutb: error V'); end;
    M=[M V]; % add one column to M
end;

for j=1:nrp+1,
    V=[];
    if (j>1), V=[V ; zeros(j-1,1)]; end;
    V=[V ; Bh'];
    if (j<=nrp), V=[V ; zeros(nrp+1-j,1)]; end;
    if (length(V)~=nah+nbh), disp('bezoutb: error V'); end;
    M=[M V];
end;

D=size(M);
if (D(1)~=nah+nbh), disp('bezoutb: error size M row'); end;
if (D(2)~=nah+nbh), disp('bezoutb: error size M column'); end;

% make P column vector
P=P';

global M1;
M1=M;

X= M\P;
% coefficients are real values
X=real(X);

% make X row vector
X=X';

```

## Commande des systèmes par ordinateur

```
Sp=X(1:nsp+1);  
Rp=X(nsp+2:nsp+nrp+2);
```

## 10. BIBLIOGRAPHIE

- [1] I. D. Landau, "*Commande des systèmes*", Edition Lavoisier 2002.
- [2] C. Nouals, "Théorie du Signal", Notes de cours ENSICA
- [3] F. Vincent, "*Traitement Numérique du Signal*", Notes de cours ENSICA.
- [4] D. Arzelier, "*Représentation et analyse des systèmes, notes de cours*", Edition ENSICA. Voir aussi <http://www.laas.fr/~arzelier/>
- [5] J. Bordeneuve-Guibé, "*Commande des systèmes linéaires*", Notes de cours ENSICA.
- [6] Y. Briere, "*Identification*", Notes de cours ENSICA.
- [7] L. Ljung, "*System Identification, Theory for the User*", Prentice Hall 1987.
- [8] Wikipedia , équation de Bezout : [http://en.wikipedia.org/wiki/Bézout's\\_identity](http://en.wikipedia.org/wiki/Bézout's_identity)
- [9] <http://www.mathworks.com>



# 11. INDEX

<b>A</b>	
actionneur intelligent .....	21
amortissement .....	15
ANSI C .....	60
Approximation numérique d'équation différentielle .....	28
ARMAX .....	35
<b>B</b>	
BJ35 .....	
Blocage d'un signal .....	56
bloqueurs d'ordre 0 .....	22
BOZ .....	22
bus.....	21, 61
<b>C</b>	
CAN.....	21
capteur intelligent .....	21
causal .....	6, 9
Causalité .....	10
CNA.....	21
compilateur .....	60
convertisseur analogique / numérique.....	21
convertisseur numérique / analogique.....	21
convertisseurs analogique-numérique .....	61
coprocesseur mathématique .....	60
Correspondances pôles - zéros.....	31
<b>D</b>	
Diagramme de Bode .....	18
Discrétisation de fonction de transfert continues .....	27
division polynomiale.....	6
<b>E</b>	
échelon unitaire.....	11
équation d'état .....	17
Erreur statique.....	42
Estimation de la réponse fréquentielle .....	39
Estimation de la réponse impulsionnelle.....	38
Euler .....	29
<b>F</b>	
facteur d'échelle .....	61
facteur d'oubli .....	36
<i>fenêtre de Hamming</i> .....	39
filtre anti-repliement .....	25
fonction de sensibilité .....	42
fonction de transfert discrète.....	9
<b>G</b>	
gain statique.....	11
gain statique unitaire.....	42
<b>I</b>	
Identification.....	32
Identification non paramétrique .....	38
intercorrélation.....	38
<b>L</b>	
lieu des racines.....	43
lieu d'Evans.....	43
LTI.....	8
<b>M</b>	
marge de gain.....	47
marge de module.....	46
marge de phase .....	47
méthode des moindres carrés récursifs .....	32
Méthode des trapèzes.....	29
modèle ARX .....	32
<b>O</b>	
OE.....	35
one step ahead prediction.....	35
opérateur de retard .....	9
<b>P</b>	
période d'échantillonnage (choix) .....	26
Placement de pôles .....	54
Pôles .....	11
Poursuite .....	56
<i>prédiction</i> .....	32
protocoles.....	61
Pseudo Random Binary Signal .....	34
pulsation .....	15
<b>Q</b>	
quantification .....	23, 24, 60
<b>R</b>	
R,S,T.....	41
rapidité.....	15
rayon de convergence .....	5, 9
Régulation.....	55

Rejet de perturbations .....	42
Rejet d'une perturbation harmonique .....	56
réponse à un échelon.....	11
réponse fréquentielle.....	18
Réponse impulsionnelle.....	10
Réponses indicielle .....	10
retard.....	23
root locus .....	43

### **S**

signal discret .....	5
smart sensors.....	62
stabilité.....	11
Systèmes du premier ordre.....	14
Systèmes du second ordre.....	15

### **T**

Théorème de Nyquist.....	46
théorème de Shannon.....	19, 24
théorème du retard .....	6
Transformée en z .....	5
transformée unilatérale .....	5
Tustin.....	29

### **V**

virgule flottante.....	60
------------------------	----

### **Z**

zero holder hold .....	23
zéros.....	11
zoh .....	23

# Commande des systèmes par ordinateur

Yves Briere  
ENSICA

Notes de cours  
2007





# TABLE DES MATIERES

Table des matières .....	1
1. Introduction .....	1
2. Rappels et prérequis .....	3
3. Signaux et systèmes discrets .....	5
3.1 Signaux discrets .....	5
3.1.1 Transformée en z .....	5
3.1.2 Signaux types .....	7
3.2 Systèmes discrets .....	9
3.2.1 Fonction de transfert discrète, définition .....	9
3.2.2 Notation "opérateur de retard $q^{-1}$ " .....	10
3.2.3 Réponse impulsionnelle .....	11
3.2.4 Causalité .....	11
3.2.5 Réponses indicielle .....	11
3.2.6 Régime stationnaire (gain statique) .....	12
3.2.7 Pôles, zéros, stabilité .....	12
3.2.8 Systèmes du premier ordre .....	15
3.2.9 Systèmes du second ordre .....	16
3.2.10 Calcul itératif de la réponse d'un système .....	18
3.2.11 Représentation par équation d'état .....	18
3.3 Réponse fréquentielle des systèmes échantillonnés .....	19
3.3.1 Diagramme de Bode d'un système échantillonné .....	19
3.3.2 Où l'on voit apparaître le théorème de Shannon .....	20
4. Systèmes continus échantillonnés .....	23
4.1 Structure des systèmes commandés par ordinateur .....	23
4.2 Échantillonnage des systèmes continus .....	24
4.3 Effets de l'échantillonnage .....	25
4.4 Où l'on revoit le théorème de Shannon .....	26
4.5 Discrétisation de fonction de transfert continues .....	29
4.5.1 Approximation numérique d'équation différentielle .....	30
4.5.2 Superposition des réponses indicielles .....	32
4.5.3 Correspondances pôles - zéros .....	33
4.5.4 Bilan .....	33
5. Identification des systèmes échantillonnés .....	35
5.1 Identification d'un modèle ARX par la méthode des moindres carrés récurrents .....	35
5.2 L'identification paramétrique en général et en pratique .....	37

5.2.1	Choix du signal d'excitation, traitement des mesures .....	37
5.2.2	Choix d'une structure de modèle.....	38
5.2.3	Choix d'un critère à minimiser.....	38
5.2.4	Choix d'un algorithme.....	38
5.2.5	Validation.....	39
5.2.6	Utilisation de logiciels .....	39
5.3	Identification "simple" .....	39
5.4	Identification non paramétrique .....	41
5.4.1	Estimation de la réponse impulsionnelle .....	41
5.4.2	Estimation de la réponse fréquentielle.....	42
6.	Systèmes bouclés .....	45
6.1	Fonction de transfert en boucle fermée .....	45
6.2	Erreur statique .....	46
6.3	Rejet de perturbations.....	46
6.4	Stabilité des systèmes à simple gain de rétroaction.....	47
6.5	Analyse des systèmes bouclés dans le domaine fréquentiel : stabilité.....	49
6.5.1	Théorème de Nyquist dans le cas où le système est stable en boucle ouverte (critère du revers) .....	50
6.5.2	Théorème de Nyquist dans le cas où le système est instable en boucle ouverte .....	50
6.5.3	Interprétations du lieu de Nyquist.....	50
7.	Méthodes de régulation des systèmes échantillonnés .....	53
7.1	Introduction .....	53
7.2	Structure d'un système commandé par ordinateur, hypothèses.....	54
7.2.1	Discretisation, choix de la période d'échantillonnage.....	54
7.2.2	Discretisation d'un correcteur continu .....	55
7.3	Régulateur PID numérique .....	55
7.3.1	Cas 1 .....	55
7.3.2	Cas 2 .....	58
7.4	Placement de pôles .....	59
7.4.1	Choix des pôles en boucle fermée .....	60
7.4.2	Régulation : calcul de R et S.....	60
7.4.3	Poursuite : calcul de T .....	61
7.5	Poursuite et régulation à objectifs indépendants .....	62
8.	Technologie de la commande par ordinateur.....	65
8.1	Calculateurs et interfaces numériques / analogiques.....	65
8.2	Transport de l'information : les bus et les protocoles .....	67
8.3	Capteurs et actionneurs "intelligents", "smart sensors" .....	68
8.4	La philosophie "hardware in the loop" .....	68
8.5	Le prototypage rapide en automatique .....	68
9.	Annexe .....	69
9.1	Annexe 1 : Résolution de l'équation polynomiale de Bezout.....	69
10.	Bibliographie.....	73
11.	Index.....	1





# 1. INTRODUCTION



Figure 1 : commande d'un systèmes de télémédecine par calculateur, vu par Claude Serre



## 2. RAPPELS ET PREREQUIS

Ce polycopié ne comporte pas de rappels sur les systèmes continus. S'adressant aux étudiants en deuxième année de l'ENSICA il suppose connu les enseignements de première année et de seconde année en automatique et traitement du signal.

En première année le cours de théorie du signal [2] a rendu familier les notions de :

- transformée de Fourier, transformée de Laplace
- corrélation, convolution

Le cours de traitement numérique du signal [3] a abordé en plus :

- la discrétisation et la quantification des signaux.
- la transformée en z et la transformée de Fourier discrète.
- les filtres numériques RIF et RII.

En automatique [4] les bases de la représentation et de l'analyse des systèmes continus ont été posées. On ne reviendra pas dans ce cours sur les notions de :

- modèles mathématique des systèmes linéaires à temps continus à temps invariant (LTI) : fonction de transfert, équation d'état, pôles, zéros et dynamique.
- commandabilité et observabilité.
- stabilité des systèmes en boucle ouverte : pôles, critères de Routh-Hurwitz.
- stabilité des systèmes bouclés : critère de Nyquist, critère de Bode.
- lieu des racines.

En deuxième année les notions suivantes sont acquises [5] :

- synthèse des correcteurs par méthodes fréquentielles (avance et retard de phase, PID)
- commande par retour d'état (placement de pôles), commande Linéaire Quadratique (LQG)
- reconstruction de l'état par Observateur, observateur optimal de Kalman.





## 3. SIGNAUX ET SYSTEMES DISCRETS

### 3.1 Signaux discrets

#### 3.1.1 Transformée en z

Pour une approche mathématiquement rigoureuse de représentation des signaux discrets par la transformée en z consulter [3] et sa bibliographie. Dans ce cours on se cantonne à une approche utilitariste des outils et méthodes mathématiques tout en essayant de ne pas contredire fondamentalement la rigueur mathématique.

Un signal discret  $s(k)$  est une séquence de nombres réels appelés échantillons. L'indice  $k$  est généralement associé à un instant d'échantillonnage :  $t = k.T_e$ . La période d'échantillonnage  $T_e$  est supposée constante. Par définition la transformée en z du signal  $s(k)$  est la fonction de la variable  $z$ , notée abusivement  $s(z)$  :

$$s(z) = Z(s(k)) = \sum_{k=0}^{\infty} s(k) \cdot z^{-k} \quad (1)$$

La transformée définie ici est plus précisément appelée transformée unilatérale puisque les termes correspondant à  $k < 0$  ne sont pas pris en compte. Ceci n'a pas d'importance puisque les signaux que nous utilisons sont systématiquement causaux<sup>1</sup> :  $s(k) = 0$  pour  $k < 0$ .

Il y a bien évidemment un lien avec la transformée de Laplace et la transformée de Fourier (voir [3])

La fonction de la variable complexe  $s(z)$  est une série et ne converge pas forcément pour toute valeur de  $z$ . En toute rigueur il faudrait associer le rayon de convergence à chaque fonction en  $z$  :  $\{s(z) \text{ converge pour } |z| > \alpha\}$ .

Les automaticiens pour qui la transformée en z n'est bien souvent qu'une façon bien commode de regrouper des termes de récurrence utilisent souvent la pseudo-variable<sup>2</sup>  $q$ .

A partir d'une transformée en z  $s(z)$  il est possible de retrouver la séquence  $s(k)$  correspondante en appliquant la transformation inverse, dont on ne donnera pas la formule ici. A la place de faire appel à cette formule compliquée il est possible d'utiliser d'autres méthodes pour retrouver la séquence  $s(k)$  lorsque la fonction  $s(z)$  est donnée sous forme de fraction rationnelle :

<sup>1</sup> Pour une justification du terme « causal » voir page 11

<sup>2</sup> Voir page 10

- division polynomiale
- décomposition de la fraction rationnelle en éléments simples puis utilisation de transformations élémentaires telles que celles données dans le Tableau 2

La propriété fondamentale de la transformée en z est qu'elle est linéaire (la transformée en z d'une composition linéaire de signaux est la composition linéaire de la transformée en z de chaque signal). Les autres propriétés fondamentales sont données dans le Tableau 2.

Tableau 1 : propriétés de la transformée en z

1	$s(z) = Z(s(k)) = \sum_{k=0}^{\infty} s(k) \cdot z^{-k}$
2	$Z(\alpha \cdot s_1(k) + \beta \cdot s_2(k)) = \alpha \cdot Z(s_1(k)) + \beta \cdot Z(s_2(k))$
3	$Z(k \cdot s(k)) = -z \cdot \frac{ds(z)}{dz}$
4	$Z(c^k \cdot s(k)) = s\left(\frac{z}{c}\right)$
5	$Z(s(k-1)) = s(k=-1) + z^{-1} \cdot s(z)$
6	$Z(s(k+1)) = z \cdot s(z) - z \cdot s(k=0)$
7	$s(k=0) = \lim_{z \rightarrow \infty} (s(z))$
8	$s(k \rightarrow \infty) = \lim_{z \rightarrow 1} \left( \frac{z-1}{z} \cdot s(z) \right)$

Remarque 1 : la propriété 5 ne prend pas en compte le fait que le signal est causal : en pratique  $s(k=-1) = 0$ .

Remarque 2 : la propriété 5 justifie le fait que le terme  $z^{-1}$  soit appelé "opérateur de retard unitaire", cette propriété est aussi appelée "théorème du retard".

Remarque 3 : les propriétés 7 et 8 sont données sous réserve de convergence.

Transformées inverses

### Exemple 1 : transformée inverse par division polynomiale

On considère un signal  $s(k)$  dont la transformée en z est donnée par la fraction rationnelle :

$$S(z) = \frac{0.1 \cdot z^2 + 0.1 \cdot z}{z^3 - 2.6 \cdot z^2 + 2.2 \cdot z - 0.6} \quad (3.1)$$

L'objectif est d'obtenir  $S(z)$  sous forme polynomiale :

$$S(z) = x_0 + x_1 \cdot z^{-1} + x_2 \cdot z^{-2} + \dots \text{etc.} \tag{3.2}$$

Cette suite est généralement infinie, la réponse temporelle est donnée par la succession des termes  $\{x_i\}$ . En traitant l'équation (3.1) comme une simple division l'algorithme de division classique (celui appris en CM2) permet d'obtenir par récurrence tous les termes de la suite (3.2) :

$0.1 \cdot z^2$	$+ 0.1 \cdot z$				$z^3 - 2.6 \cdot z^2 + 2.2 \cdot z - 0.6$ $0.1 \cdot z^{-1} + 0.36 \cdot z^{-2} + 0.716 \cdot z^{-3}$
$0.1 \cdot z^2$	$- 0.26 \cdot z$	$+ 0.22$	$- 0.06 \cdot z^{-1}$		
$0$	$0.36 \cdot z$	$- 0.22$	$0.06 \cdot z^{-1}$		
	$0.36 \cdot z$	$- 0.936$	$0.792 \cdot z^{-1}$	$- 0.216 \cdot z^{-2}$	
$0$	$0.716$	$- 0.732 \cdot z^{-1}$	$\dots$	$\dots$	

Les éléments successifs de la suite  $s(k)$  sont donc  $s(k) = \{0, 0.1, 0.36, 0.716, \dots\}$ . Pour obtenir les termes suivants il faudrait continuer la division.

**Exemple 2 : transformée inverse par décomposition en éléments simples**

La deuxième méthode présentée consiste à décomposer la fraction rationnelle (3.1) en éléments simples puis d'utiliser les tables de transformées en  $z$  (Tableau 2). Pour notre exemple cela donne :

$$S(z) = \frac{0.1 \cdot z^2 + 0.1 \cdot z}{z^3 - 2.6 \cdot z^2 + 2.2 \cdot z - 0.6} = \frac{-z}{z-1} + \frac{0.5 \cdot z}{(z-1)^2} + \frac{z}{z-0.6}$$

La réponse temporelle est donc composée des trois éléments :

$$s(k) = -(1 \ 1 \ 1 \ \dots) + 0.5 \cdot (1 \ 2 \ 3 \ \dots) + (1 \ 0.6 \ 0.6^2 \ \dots)$$

$$\text{soit : } s(k) = \begin{cases} s(0) = -1 + 0.5 + 1 \\ s(2) = -1 + 0.5 \cdot 2 + 0.6 \cdot 0.5 \\ s(3) = -1 + 0.5 \cdot 3 + 0.6^2 \\ s(4) = \dots \end{cases}$$

**3.1.2 Signaux types**

Le Tableau 2 donne les transformée en  $z$  de certains signaux fondamentaux. La Figure 2 représente l'allure de quelques signaux types.

Tableau 2 : transformée en z élémentaires

$s(k)$	$S(z)$
signal temporel	Transformée en z
$\delta(k)$ : impulsion unité	1
$u(k)$ : échelon unité	$\frac{z}{z-1}$
$k \cdot u(k)$	$\frac{z}{(z-1)^2}$
$c^k \cdot u(k)$	$\frac{z}{z-c}$
$k \cdot a^k \cdot u(k)$	$\frac{a \cdot z}{(z-a)^2}$
$\sin(\omega \cdot k) \cdot u(k)$	$\frac{z \cdot \sin(\omega)}{z^2 - 2 \cdot z \cdot \sin(\omega) + 1}$
$\cos(\omega \cdot k) \cdot u(k)$	$\frac{z \cdot (z - \cos(\omega))}{z^2 - 2 \cdot z \cdot \sin(\omega) + 1}$
$a^k \cdot \sin(\omega \cdot k) \cdot u(k)$	$\frac{a \cdot z \cdot \sin(\omega)}{z^2 - 2 \cdot a \cdot \cos(\omega) \cdot z + a^2}$
$a^k \cdot \cos(\omega \cdot k) \cdot u(k)$	$\frac{z \cdot (z - a \cdot \cos(\omega))}{z^2 - 2 \cdot a \cdot \cos(\omega) \cdot z + a^2}$

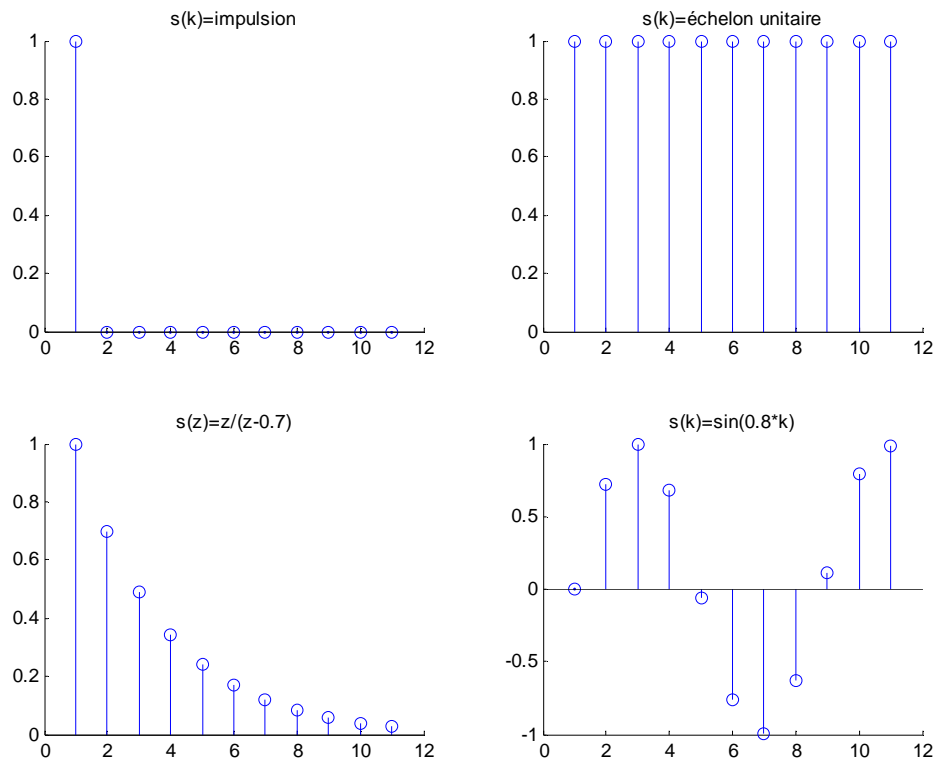


Figure 2 : réponses de quelques signaux types

## 3.2 Systèmes discrets

### 3.2.1 Fonction de transfert discrète, définition

On considère maintenant un système discret linéaire et invariant dans le temps (LTI) représenté sur la Figure 3.

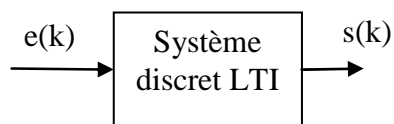


Figure 3 : système discret, notations

La relation entre la séquence d'entrée et la séquence de sortie est linéaire et définie par un ensemble de coefficients constants  $a_{1:n}$  et  $b_{1:m}$  :

$$\begin{aligned} s(k) + a_1 \cdot s(k-1) + a_2 \cdot s(k-2) + \dots + a_n \cdot s(k-n) = \\ b_0 \cdot e(k) + b_1 \cdot e(k-1) + b_2 \cdot e(k-2) + \dots + b_m \cdot e(k-m) \end{aligned} \quad (2)$$

On peut appliquer la transformée en  $z$  à chaque membre de l'équation puis en appliquant en récurrence la propriété 5 (théorème du retard) on obtient la fonction de transfert discrète du système :

$$\frac{s(z)}{e(z)} = F(z) = \frac{b_0 + b_1 \cdot z^{-1} + b_2 \cdot z^{-2} + \dots + b_m \cdot z^{-m}}{1 + a_1 \cdot z^{-1} + a_2 \cdot z^{-2} + \dots + a_n \cdot z^{-n}} \quad (3)$$

Remarque 1 : la fonction de transfert précédente est normalisée en prenant  $a_0 = 1$ .

Remarque 2 : on remarquera que la fonction de transfert est donnée en  $z^{-1}$  plutôt qu'en  $z$ . Cela est justifié par le fait que celle-ci traduit l'équation d'évolution du système physique causal (2).

Le script Matlab permettant de créer un objet Matlab correspondant à une fonction de transfert discrète est donné ci-dessous.

#### Matlab

```
% Création d'un système linéaire discret :
Te=0.1;
sysd = tf([b0 b1 b2],[1 a0 a2 a3],Te,'variable','z^-1');
```

On notera l'utilisation de l'argument 'variable', 'z^-1'.

### 3.2.2 Notation "opérateur de retard $q^{-1}$ "

La fonction de transfert est une fonction de la variable complexe  $z$ . Il faudrait donc en toute rigueur l'associer à son rayon de convergence. En pratique on ne s'embarrasse jamais de préciser cette zone de convergence. C'est pourquoi on trouve souvent la notation utilisant l'opérateur  $q^{-1}$  de décalage à la place de la variable complexe  $z^{-1}$  (4).  $F(q)$  n'est alors plus à proprement parler une fonction mais on lui appliquera les mêmes manipulations algébriques qu'à  $F(z)$ .

$$F(q) = \frac{b_0 + b_1 \cdot q^{-1} + b_2 \cdot q^{-2} + \dots + b_m \cdot q^{-m}}{1 + a_1 \cdot q^{-1} + a_2 \cdot q^{-2} + \dots + a_n \cdot q^{-n}} \quad (4)$$

Attention : selon les auteurs l'opérateur qui remplace  $z^{-1}$  est noté  $q$  (Matlab) ou  $q^{-1}$  (Landau [1]). Dans la suite du document c'est cette dernière convention qui sera appliquée.

L'utilisation de l'opérateur de retard  $q^{-1}$  à la place de la variable complexe  $z^{-1}$  permet de traiter du cas des fonctions de transfert à coefficients non constant, cas non prévu par la transformation en  $z$ .

### 3.2.3 Réponse impulsionnelle

Essayons d'appliquer au système un signal d'entrée particulier : une impulsion unitaire. La transformée en  $z$  du système est obtenue en multipliant la fonction de transfert par la transformée en  $z$  de l'impulsion unitaire : 1. On obtient :

$$s(z) = \frac{b_0 + b_1 \cdot z^{-1} + b_2 \cdot z^{-2} + \dots + b_m \cdot z^{-m}}{1 + a_1 \cdot z^{-1} + a_2 \cdot z^{-2} + \dots + a_n \cdot z^{-n}} \times 1 \quad (5)$$

Cela justifie le fait que la fonction de transfert du système peut aussi s'appeler "réponse impulsionnelle".

#### Matlab

```
% Réponse impulsionnelle d'un système :
[y,t] =impulse(sysd);
```

### 3.2.4 Causalité

Un système est dit causal si les sorties ne dépendent que du passé et non pas du futur. La conséquence est que la réponse impulsionnelle de  $F(z)$  ne peut pas comporter de terme non nuls pour  $k < 0$ . Cela justifie la confusion sémantique entre "système causal" et "signal causal", c'est-à-dire ne comportant pas de termes non nuls pour  $k < 0$ .

Attention : à la différence des systèmes continus un système physique (causal) discret peut très bien être modélisé avec un degré  $m$  supérieur au degré  $n$  : le système peut avoir plus de zéros que de pôles.

La transformée en  $z$  représentant des signaux échantillonnés elle peut tout aussi bien représenter les mesures prises à différents endroits d'une poutre. Le temps étant remplacé ici par une abscisse la notion de causalité ne s'applique pas ! Attention : beaucoup des propriétés de systèmes échantillonnés que l'on étudie sont liés à la causalité, prudence donc si l'on applique ces méthodes à d'autres domaines !

### 3.2.5 Réponses indicielle

Pour caractériser des systèmes on s'intéresse souvent à leur réponse impulsionnelle ou, plus souvent, à leur réponse indicielle. Pour calculer la réponse

à un échelon d'un système il faut donc multiplier la fonction de transfert par la transformée en z d'un échelon unitaire :

$$s(z) = \frac{b_0 + b_1 \cdot z^{-1} + b_2 \cdot z^{-2} + \dots + b_m \cdot z^{-m}}{1 + a_1 \cdot z^{-1} + a_2 \cdot z^{-2} + \dots + a_n \cdot z^{-n}} \times \frac{1}{1 - z^{-1}} \quad (6)$$

Il faudrait ensuite calculer la transformée en z inverse de s(z). En pratique, si l'on doit faire le calcul à la main on décompose en éléments simples puis on utilise des tables. Avec Matlab c'est plus simple. Voici par exemple le script pour obtenir les 20 premiers échantillons de la réponse indicielle d'un système :

**Matlab**

```
% Réponse indicielle d'un système :
[y,t] =step(sysd,20);
```

### 3.2.6 Régime stationnaire (gain statique)

Le gain statique est obtenu en appliquant une entrée indicielle et en calculant la valeur du régime permanent. En appliquant le théorème de la valeur finale il vient :

$$F_0 = \lim_{z \rightarrow 1} \left( \frac{z-1}{z} \cdot \frac{b_0 + b_1 \cdot z^{-1} + b_2 \cdot z^{-2} + \dots + b_m \cdot z^{-m}}{1 + a_1 \cdot z^{-1} + a_2 \cdot z^{-2} + \dots + a_n \cdot z^{-n}} \times \frac{1}{1 - z^{-1}} \right) \quad (7)$$

Soit :

$$F_0 = \frac{\sum_{i=0}^{i=m} b_i}{1 + \sum_{i=1}^{i=n} a_i} \quad (8)$$

Le gain statique est obtenu donc simplement en faisant le quotient de la somme des coefficients.

### 3.2.7 Pôles, zéros, stabilité

Comme pour les systèmes continus la dynamique et la stabilité des systèmes dépend des propriétés du dénominateur de la fonction de transfert.



$$F(z) = \frac{e_0 \cdot (1 + e_1 \cdot z^{-1}) \cdot (1 + e_2 \cdot z^{-1}) \cdots (1 + e_m \cdot z^{-1})}{(1 + c_1 \cdot z^{-1}) \cdot (1 + c_2 \cdot z^{-1}) \cdots (1 + c_n \cdot z^{-1})} \quad (9)$$

Les coefficients  $c_i$  sont les pôles de la fonction de transfert, les coefficients  $e_i$  sont les zéros. La fonction de transfert étant au départ composée de coefficients purement réels les coefficients  $c_i$  sont soit réels soit imaginaires par paires conjuguées.

Une fonction de transfert est donc aussi bien définie par la donnée de ses pôles et de ses zéros. Le script Matlab pour créer un système ainsi est donné ci-dessous :

#### Matlab

```
% Système discret défini par ses pôles et zéros :
Te=0.1;
sysd=zpk([c1 c2 c3],[e1 e2],e0,Te);
```

Si l'on décompose la fonction de transfert en éléments simples on obtient une somme d'éléments du type :

$$F(z) = \frac{d_1}{1 + c_1 \cdot z^{-1}} + \frac{d_2}{1 + c_2 \cdot z^{-1}} + \cdots + \frac{d_n}{1 + c_n \cdot z^{-1}} \quad (10)$$

Un système linéaire temps invariant est dit stable si sa réponse impulsionnelle tend asymptotiquement vers 0. Compte tenu de (10) cela nécessite que chaque pôle  $c_i$  soit de module inférieur à 1.

**Propriété** : un système discret LTI est stable si et seulement si tous ses pôles ont leur modules inférieur ou égal à 1.

#### Critère algébrique attestant de la stabilité : critère de Routh

On rappelle que le critère de Routh permet de prédire si les racines d'un polynôme sont toutes à partie réelle négative, donc stables. Énonçons le théorème de Routh dans le cas d'un polynôme  $P'(w)$  :

$$P'(w) = a_0 + a_1 \cdot w + a_2 \cdot w^2 + \cdots + a_n \cdot w^n \quad (11)$$

Dans un premier temps on construit le tableau :

$$\left| \begin{array}{cccccc} \beta_{n,0} & \beta_{n,1} & \beta_{n,2} & \beta_{n,q-1} & \beta_{n,q} & 0 \\ \beta_{n-1,0} & \beta_{n-1,1} & \beta_{n-1,2} & \beta_{n-1,q-1} & \beta_{n-1,q-1} & 0 \\ \beta_{n-2,0} & \beta_{n-2,1} & \beta_{n-2,2} & \beta_{n-2,q-1} & \beta_{n-2,q-1} & \\ \\ \beta_{2,0} & \beta_{2,1} & 0 & & & \\ \beta_{1,0} & 0 & & & & \\ \beta_{0,0} & 0 & & & & \end{array} \right| \quad (12)$$

Puis on vérifie que tous les premiers éléments de la première colonne ont le même signe : cela garantit que tous les pôles de  $P'(w)$  ont leur partie réelle négative.

Les éléments du tableau sont construits par récurrence ligne par ligne en appliquant les formules suivantes :

Pour  $i = n$  et  $i = n - 1$  :

$$\begin{cases} \beta_{n,j} = a_{n-2j} \\ \beta_{n-1,j} = a_{n-1-2j} \end{cases} \quad (13)$$

Et pour  $i < n-1$  :

$$\beta_{i,j} = \frac{\beta_{i+1,0} \cdot \beta_{i+2,j+1} - \beta_{i+1,j+1} \cdot \beta_{i+2,0}}{\beta_{i+1,0}} \quad (14)$$

Dans le cas qui nous intéresse, un système échantillonné, on cherche plutôt à vérifier que les pôles du polynôme  $P(z)$  sont de module inférieur à un. L'idée est d'effectuer un changement de variable  $z \rightarrow w$  pour pouvoir appliquer le critère de Routh.

$$P'(w) = (1-w)^n \cdot P\left(\frac{1+w}{1-w}\right) \quad (15)$$

Si l'on effectue la transformation (15) alors le fait que  $P'(w)$  ait toutes ses racines à partie réelle négative garantit que  $P(z)$  a toutes ses racines dans le cercle unité. Pour vérifier la stabilité de  $P(z)$  il suffit donc d'effectuer le changement de variable (15) puis d'appliquer le critère de Routh conventionnel (12 13 14).

### Critère algébrique attestant de la stabilité : critère de Jury

Le critère de Jury s'apparente au critère de Routh. Nous le donnons pour un polynôme de degré 2, 3 et 4 du type :

$$P(z) = a_0 + a_1 \cdot z + a_2 \cdot z^2 + \dots + a_n \cdot z^n \quad (16)$$

Les conditions de stabilité sont données ci-dessous (ces conditions sont données dans le cas  $a_n > 0$ , il suffit de multiplier tous les termes par  $-1$  si cette condition n'est pas remplie par  $P(z)$ ) :

$$\begin{aligned}
 n = 2 & \begin{cases} a_0 + a_1 + a_2 > 0 \\ a_0 - a_1 + a_2 > 0 \\ -a_0 + a_2 > 0 \end{cases} \\
 n = 3 & \begin{cases} a_0 + a_1 + a_2 + a_3 > 0 \\ -a_0 + a_1 - a_2 + a_3 > 0 \\ -|a_0| + a_3 > 0 \\ a_0 a_2 - a_1 a_3 - a_0^2 + a_3^2 > 0 \end{cases} \\
 n = 4 & \begin{cases} a_0 + a_1 + a_2 + a_3 + a_4 > 0 \\ a_0 - a_1 + a_2 - a_3 + a_4 > 0 \\ a_4^2 - a_0^2 - |a_0 a_3 - a_1 a_4| > 0 \\ (a_0 - a_4)^2 (a_0 - a_2 + a_4) + (a_1 - a_3)(a_0 a_3 - a_1 a_4) > 0 \end{cases}
 \end{aligned} \quad (17)$$

### 3.2.8 Systèmes du premier ordre

On donne dans la Figure 4 les réponses indicielles de systèmes du premier ordre ainsi que la localisation des pôles dans le plan complexe. Le cercle de rayon unité représente la zone dans laquelle les pôles doivent se situer pour assurer la stabilité. On remarque sur les deux premières figures l'influence de la place du zéro par rapport au point  $(1,0)$  pour la rapidité.

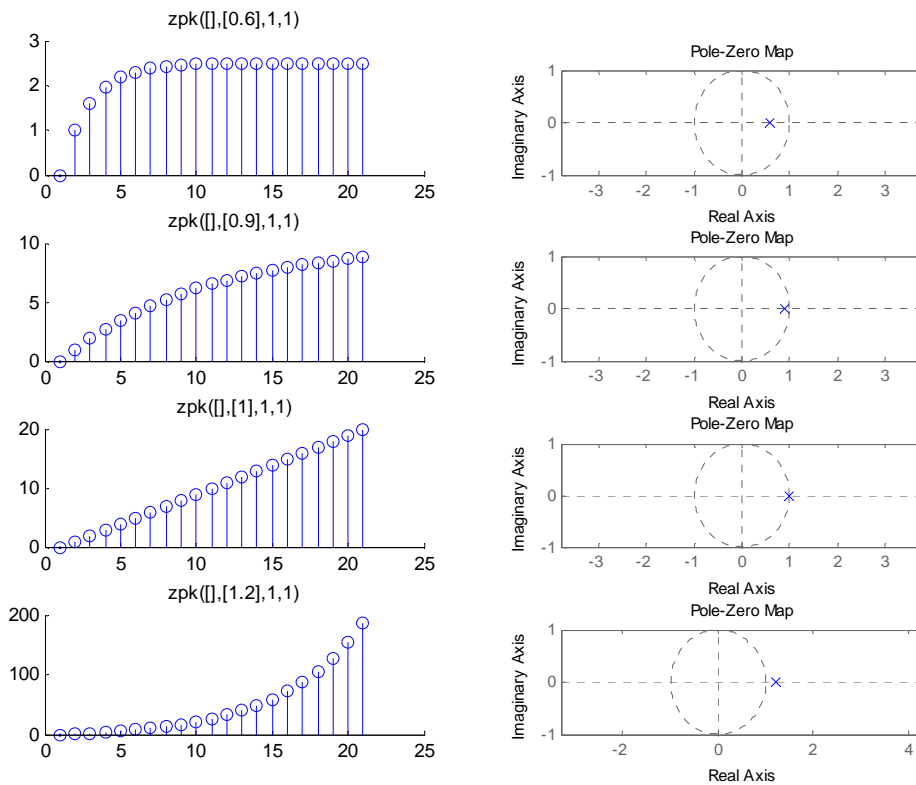


Figure 4 réponses indicielles de systèmes du premier ordre

### 3.2.9 Systèmes du second ordre

On donne dans la Figure 6 les réponses indicielles de systèmes du second ordre ainsi que la localisation des pôles dans le plan complexe. La rapidité et l'amortissement dépend de la position des pôles dans le cercle unité. La Figure 5 représente les courbes iso-amortissement et iso-pulsation des systèmes du second ordre.

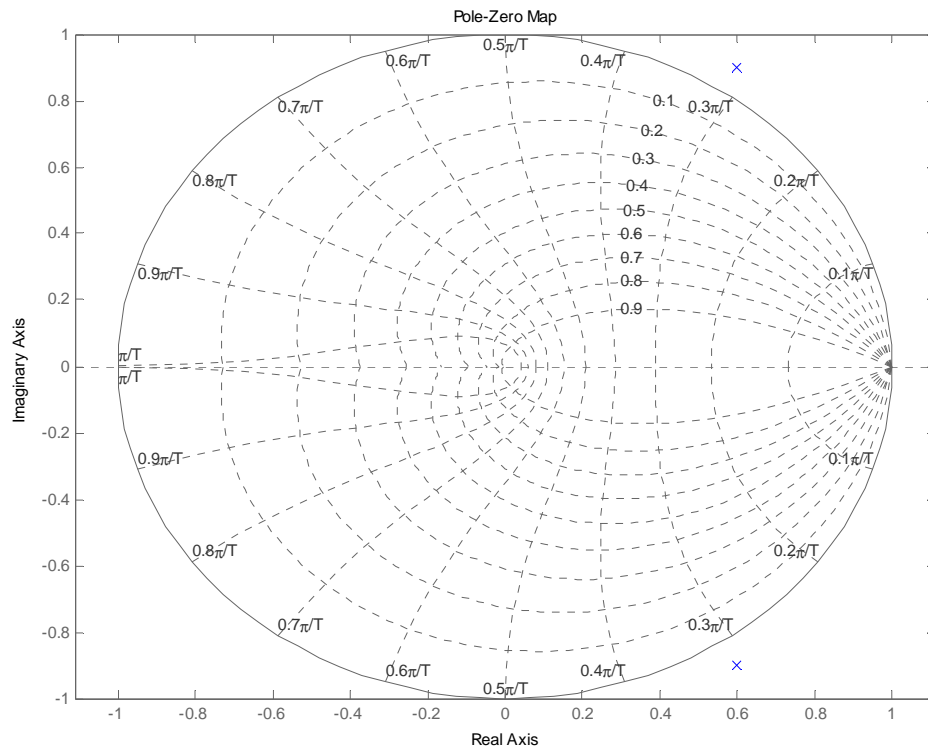


Figure 5 : courbes iso-amortissement et iso-pulsation des systèmes discrets du second ordre

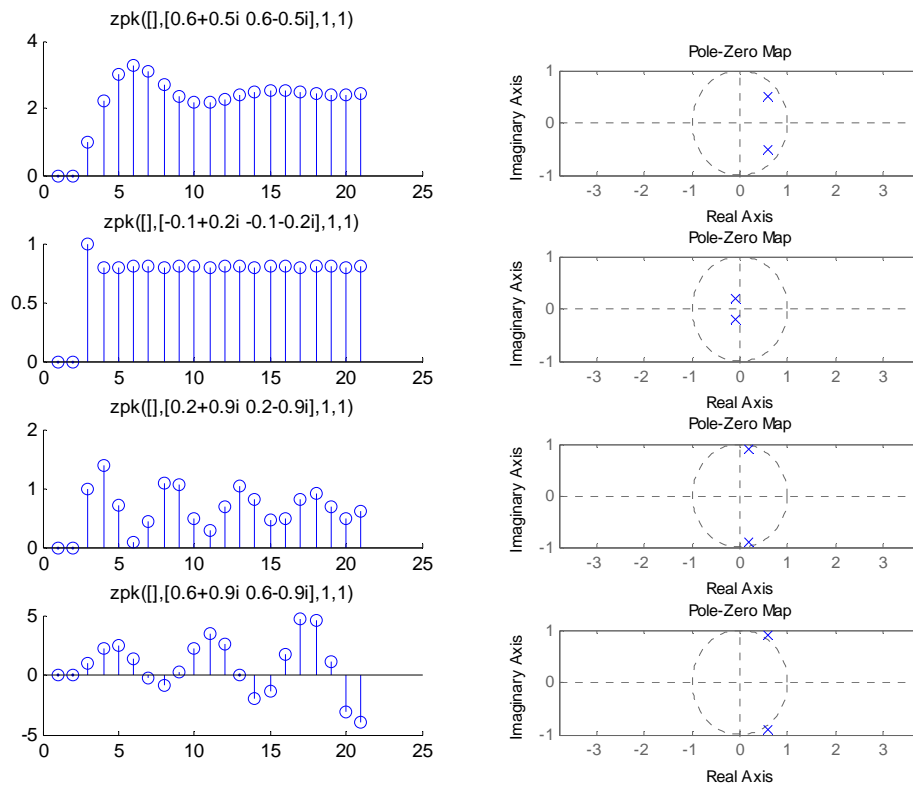


Figure 6 : réponses de systèmes du second ordre

### 3.2.10 Calcul itératif de la réponse d'un système

Si l'on dispose de la fonction de transfert d'un système et que l'on connaît les valeurs successives  $e(k)$  du signal d'entrée il est facile de calculer itérativement la sortie  $y(k)$  du système par l'équation de récurrence (2). Il faut initialiser par les  $n$  valeurs initiales de  $s(k)$ , prises généralement égales à 0.

### 3.2.11 Représentation par équation d'état

Un système discret peut aussi être représenté par un ensemble d'équations d'état comme dans le domaine continu (52). Cette représentation est naturellement adaptée aux systèmes multi-entrées multi-sorties. On définit alors un état interne  $x(k)$  ainsi qu'une fonction d'évolution  $f$  et une équation de mesure  $h$ .

$$\begin{cases} x(k+1) = f(x(k), u(k)) \\ s(k) = h(x(k), u(k)) \end{cases} \quad (18)$$

### 3.3 Réponse fréquentielle des systèmes échantillonnés

#### 3.3.1 Diagramme de Bode d'un système échantillonné

Lorsqu'on étudie la réponse fréquentielle des systèmes continus on considère des entrées particulières sinusoïdales :

$$\begin{cases} e(t) = \cos(2 \cdot \pi \cdot f \cdot t) + j \cdot \sin(2 \cdot \pi \cdot f \cdot t) = e^{j \cdot 2 \cdot \pi \cdot f \cdot t} \\ f \in [0 \quad \infty] \end{cases} \quad (19)$$

De même, pour des systèmes échantillonnés on considèrera des entrées sinusoïdales échantillonnées :

$$\begin{cases} e(k) = \cos(2 \cdot \pi \cdot f \cdot k) + j \cdot \sin(2 \cdot \pi \cdot f \cdot k) = e^{j \cdot 2 \cdot \pi \cdot f \cdot k} \\ f \in [0 \quad 1] \end{cases} \quad (20)$$

La sortie est aussi un signal échantillonné sinusoïdal de gain  $A(f)$  et de phase  $\Phi(f)$  variables et dépendant de  $f$ ; la fréquence réduite.

$$\begin{cases} s(k) = A(f) \cdot e^{j \cdot (2 \cdot \pi \cdot f \cdot k + \Phi(f))} \\ f \in [0 \quad 1] \end{cases} \quad (21)$$

Le nombre complexe, noté abusivement  $F(f)$ , d'amplitude  $A(f)$  et de phase  $\Phi(f)$  est obtenu à partir de  $F(z)$  en remplaçant  $z$  par  $e^{j \cdot 2 \cdot \pi \cdot f}$  :

$$\begin{cases} F(f) = F(z = e^{j \cdot 2 \cdot \pi \cdot f}) \\ f \in [0 \quad 1] \end{cases} \quad (22)$$

Le diagramme représentant le gain et la phase en fonction de la fréquence réduite s'appelle aussi un diagramme de Bode. L'échelle des fréquences est généralement linéaire de 0 à 0.5 (puisque le diagramme est nécessairement symétrique). La Figure 7 représente par exemple le diagramme de Bode d'un système du second ordre.

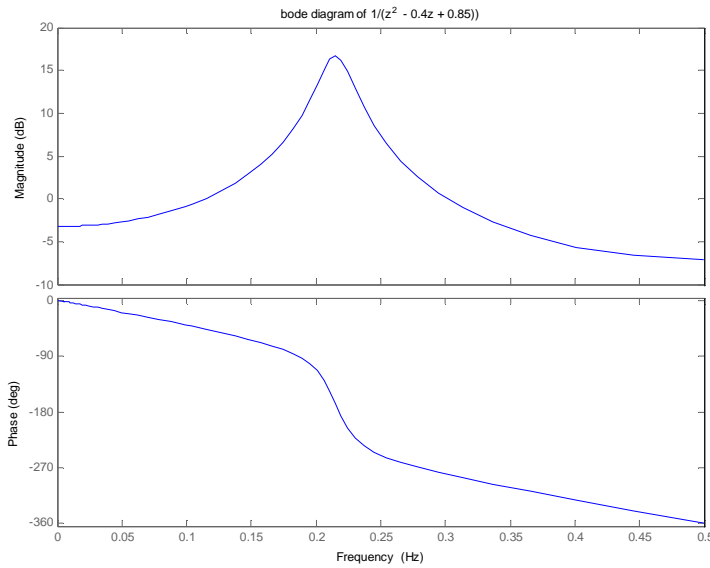


Figure 7 : exemple du diagramme de Bode d'un système discret du second ordre

### 3.3.2 Où l'on voit apparaître le théorème de Shannon

Le changement de variable  $z = e^{j2\pi f}$  n'est pas bijectif puisque  $e^{j2\pi f} = e^{j2\pi(f+k)}$  avec  $k$  entier. Si l'on fait le lien entre domaine continu et domaine fréquentiel on peut remarquer que le changement de variable  $z = e^{j2\pi f}$  fait correspondre une zone du plan complexe continu à une zone du plan complexe discret. La transformation est bijective si l'on se contente de la zone du plan complexe continu contenu entre les droites d'ordonnée 0 et 1.



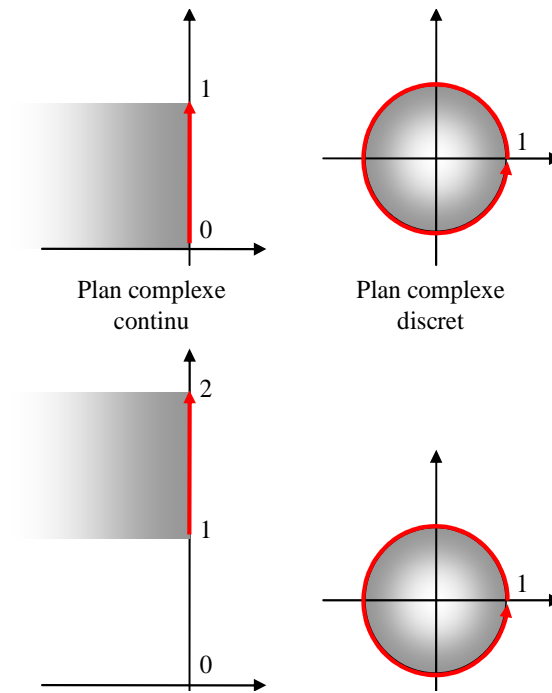


Figure 8 : correspondance entre plan complexe continu et plan complexe discret.



## 4. SYSTEMES CONTINUS ECHANTILLONNES

### 4.1 Structure des systèmes commandés par ordinateur

Les systèmes commandés par ordinateurs sont presque toujours des systèmes physiques continus par nature. Le ordinateur traite par contre les entrées et les sorties de manière discrète avec une période d'échantillonnage  $T_e$ . Les éléments importants, du point de vue de l'automaticien sont donc :

- le convertisseur analogique / numérique CAN ou du moins le système qui fournit séquentiellement les mesures au ordinateur à la période  $T_e$ .
- le ordinateur lui-même. Il doit effectuer l'acquisition des mesures, le traitement des informations, l'envoi des consignes dans un temps inférieur à  $T_e$ .
- le convertisseur numérique / analogique CNA.
- le système physique.

On suppose, pour faire simple et pour pouvoir utiliser la panoplie des outils mathématiques, que les systèmes sont échantillonnés à la même période  $T_e$  et au même moment.

La Figure 9 montre le système physique commandé par le ordinateur, dans sa version simplifiée.

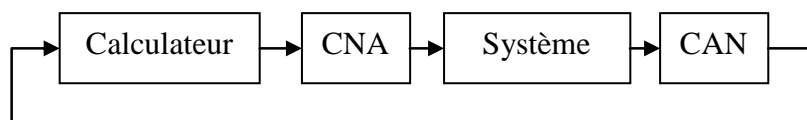


Figure 9 : système commandé par ordinateur

Au XXI<sup>ème</sup> siècle les choses sont quand même parfois un petit peu plus compliquées :

- le ordinateur fournit bien souvent un signal de commande numérique (transporté par bus informatique USB, RS232, CAN, Ethernet, etc...) directement interprété par l'actionneur. On parle "d'actionneur intelligent".
- de même le "capteur intelligent" fournit aussi bien souvent la mesure sous forme d'un signal numérique.
- Les signaux sont transportés par des bus informatiques qui n'ont pas que ça à faire : il peut y avoir du retard.
- de même les capteurs et actionneurs sont rarement tous seuls, ce qui se traduit par des périodes d'échantillonnage différentes, et parfois non synchrones...

- enfin le ordinateur peut aussi avoir d'autres choses à faire...

Ceci étant dit nous allons pour le moment nous restreindre au cas représenté sur la Figure 9 avec des instants d'échantillonnage synchrones et régulièrement espacés d'un temps  $T_e$ .

## 4.2 Echantillonnage des systèmes continus

On dispose généralement de la fonction de transfert continue (notée en utilisant la variable complexe  $s$  ou  $p$ )  $F(s)$  du système continu. Le problème est d'obtenir la fonction de transfert discrète du système échantillonné de manière à pouvoir synthétiser un correcteur. Le système continu est associé au convertisseur numérique – analogique qu'il faut prendre en compte.

Les convertisseurs numérique analogiques peuvent généralement se représenter par des "échantillonneurs – bloqueurs d'ordre 0" ou BOZ. Cela signifie que la commande est maintenue à une valeur constante pendant toute la durée du temps d'échantillonnage.

Si l'on applique une commande discrète impulsionnelle à un BOZ il va maintenir la commande 1 pendant la durée  $T_e$  puis revenir à zéro (Figure 10).

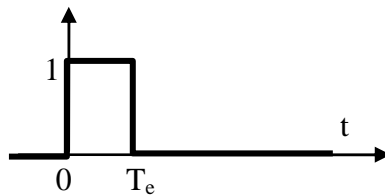


Figure 10 : réponse impulsionnelle d'un bloqueur d'ordre 0

La réponse impulsionnelle  $BOZ(t)$  est donc composée d'un échelon unitaire suivi d'un échelon unitaire de signe opposé et décalé d'un temps  $T_e$ . La transformée de Laplace de la réponse impulsionnelle du bloqueur est donnée par la formule (23) :

$$BOZ(s) = \frac{1}{s} - \frac{e^{-jT_e}}{s} \quad (23)$$

Du point de vue du ordinateur il faut donc prendre en compte le système continu de fonction de transfert  $F_{BOZ}(s)$  que l'on devra échantillonner :

$$F_{BOZ}(s) = \frac{1 - e^{-jT_e}}{s} \cdot F(s) \quad (24)$$

$$F_{\text{BOZ}}(z) = Z^{-1} \left( \frac{1 - e^{-jT_e}}{s} \cdot F(s) \right) \quad (25)$$

En pratique l'obtention de la transformée en z d'un système échantillonné par bloqueur d'ordre 0 se fait à l'aide de tables. On donne quelques exemples fondamentaux dans le Tableau 3 ci-dessous.

Tableau 3 : transformée en z de systèmes continus échantillonnés

F(s)	F <sub>BOZ</sub> (z)
1	1
$\frac{1}{s}$	$\frac{T_e \cdot z^{-1}}{1 - z^{-1}}$
$\frac{1}{1 + T \cdot s}$	$\begin{cases} \frac{b_1 \cdot z^{-1}}{1 + a_1 \cdot z^{-1}} \\ b_1 = 1 - e^{-T_e/T} \quad a_1 = -e^{-T_e/T} \end{cases}$
$\frac{e^{-Ls}}{1 + T \cdot s}$	$\begin{cases} \frac{b_1 \cdot z^{-1} + b_2 \cdot z^{-2}}{1 + a_1 \cdot z^{-1}} \quad a_1 = -e^{-T_e/T} \\ b_1 = 1 - e^{-(L-T_e)/T} \quad b_2 = e^{-T_e/T} (e^{L/T} - 1) \end{cases}$
$L < T_e$	

Si l'on dispose de Matlab c'est bien sûr encore plus simple. On utilise la fonction c2d avec l'argument 'zoh' pour "zero holder hold" (bloqueur d'ordre zéro).

#### Matlab

```
% Définition d'un système continu :
sysc=tf(1,[1 1]);
% Obtention de la transformée en z du système échantillonné à Te
Te=0.1;
sysd=c2d(sysc,Te,'zoh');
% Notation en z^-1
sysd.variable='z'
```

### 4.3 Effets de l'échantillonnage

Le convertisseur numérique a deux effets (néfastes) importants : l'effet de retard et l'effet de quantification.

L'effet de retard se voit dans (24) : on peut dire en première approximation que l'échantillonnage se traduit par un retard pur de  $T_e/2$ . Cela a donc, si ce temps

d'échantillonnage est non négligeable par rapport aux constantes de temps du système, un effet déstabilisant.

Pour illustrer l'effet de quantification supposons par exemple que l'on utilise un convertisseur analogique/numérique 8 bits 0-10V. Le plus petit incrément que le convertisseur peut distinguer est de  $10V/2^8 = 40mV$ . Si le signal utile varie physiquement entre 5V et 5,4V on n'aura que dix incréments utilisables... Ce phénomène n'entre pas dans la panoplie des phénomènes modélisables par la transformée en z. En pratique il est donc souvent nécessaire d'étudier le système et de synthétiser la commande numérique en négligeant ce phénomène puis de vérifier a posteriori qu'il n'a pas un effet trop néfaste.

#### 4.4 Où l'on revoit le théorème de Shannon

Le théorème de Shannon est bien connu en ce qui concerne l'échantillonnage de signaux discrets : on sait que le signal ne doit pas comporter de composante de fréquence supérieure à  $f_e/2$  si l'on ne veut pas observer de repliement. On doit aussi prendre des précautions si l'on veut travailler sur la fonction de transfert discrète correspondant à un système continu échantillonné.

On va traiter un exemple concret pour mettre en évidence le problème de repliement dans le cas de la commande des systèmes continus. On s'intéresse à deux systèmes continus du second ordre ayant des pôles de même partie réelle mais de parties imaginaires séparées de  $2\pi T_e$ . Le script Matlab ci-dessous montre la création des deux systèmes :

##### Matlab

```
% Définition de deux systèmes continus :  
sys1=zpk([],[-1-2i -1+2i],2^2+1);  
sys2=zpk([],[-1-(2+2*pi)*i -1+(2+2*pi)*i],1+(2+2*pi)^2);
```

On voit sur la Figure 11 que la réponse indicielle des deux systèmes est bien différente.

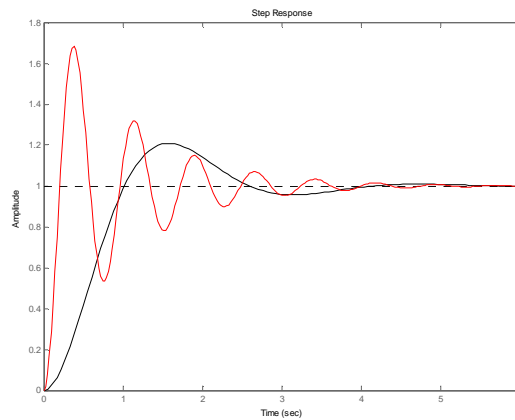


Figure 11 : réponse indicielle de deux systèmes du second ordre

Si l'on calcule la fonction de transfert échantillonnée les deux systèmes à la période de 1 seconde on obtient deux réponses indicielles très proches (représentés par des  $\circ$  sur la Figure 12). Conclusion : la période d'échantillonnage était mal choisie

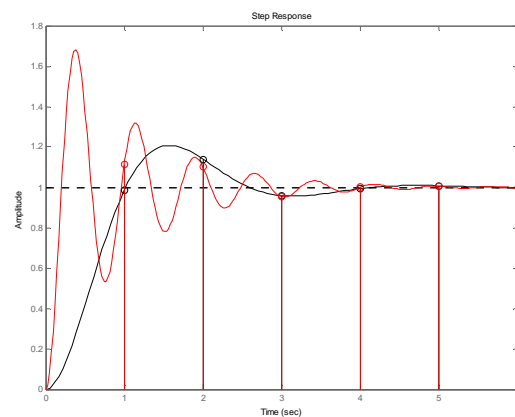


Figure 12 : réponse indicielle de deux systèmes du second ordre échantillonnés à 1s

En pratique la commande échantillonnée de systèmes continus doit être associée à certaines précautions :

- le système ne doit pas comporter de dynamiques propres situées dans les fréquences supérieures à  $f_c/2$  (exemple illustré dans ce chapitre).
- les signaux échantillonnés doivent être "nettoyés" des fréquences supérieures à  $f_c/2$  avec un filtre anti-repliement. (A moins que le signal soit suffisamment propre et ne comporte pas de composantes significatives au-delà de ces fréquences).

### Choix de la période d'échantillonnage

Le choix de la période d'échantillonnage est très important et dépend de la dynamique des signaux. Attention, les systèmes commandés en boucle fermée ont souvent une dynamique plus rapide que le système en boucle ouverte : il faut donc prendre en compte la dynamique en boucle fermée pour le choix de la fréquence d'échantillonnage.

En pratique on préconise de prendre une fréquence 5 à 25 fois supérieure à la bande passante souhaitée en boucle fermée.

On peut être tenté de prendre une fréquence d'échantillonnage très élevée si la technologie le permet (ordinateur et convertisseurs suffisamment rapide). Ceci peut amener rapidement à des problèmes numériques. Imaginons par exemple le cas d'un système ayant une dynamique très lente (quelques secondes) et un système de mesure codant les mesures sur 10 bits (cas classique). La Figure 13 illustre cet exemple. Le signal varie peu entre deux instants d'échantillonnage et les variations du signal sont faibles par rapport à la valeur de l'incrément numérique (figure du haut). Le calcul de la simple dérivée numérique se traduit donc par une succession de  $-1$   $0$  et  $+1$ .

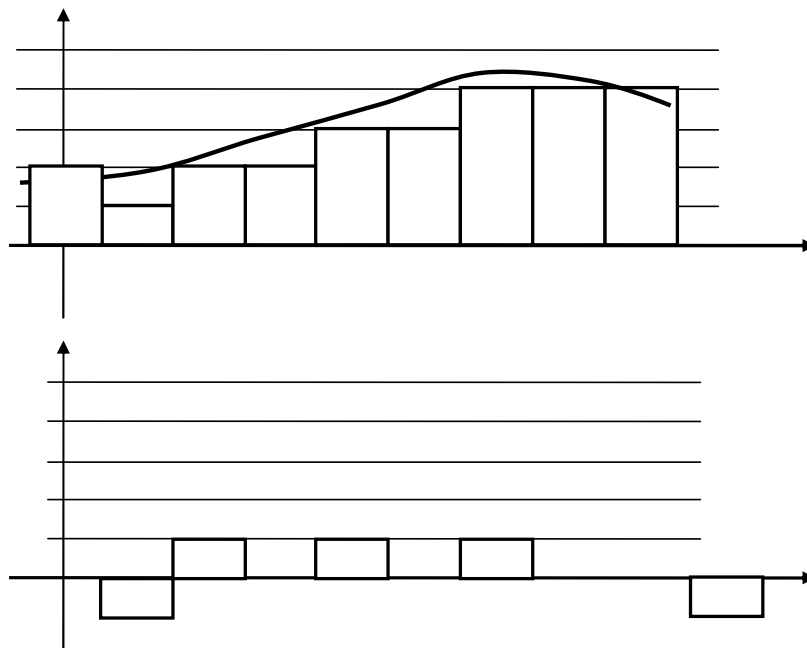


Figure 13 : période d'échantillonnage trop faible. En haut : signal échantillonné et discrétisé ; en bas : calcul de la dérivée numérique.



### Filtre anti-repliement

En toute rigueur chaque fois qu'un signal est échantillonné il faudrait faire précéder le convertisseur numérique d'un filtre analogique anti-repliement éliminant physiquement toutes les composantes du signal de fréquence supérieure à  $f_e/2$ .

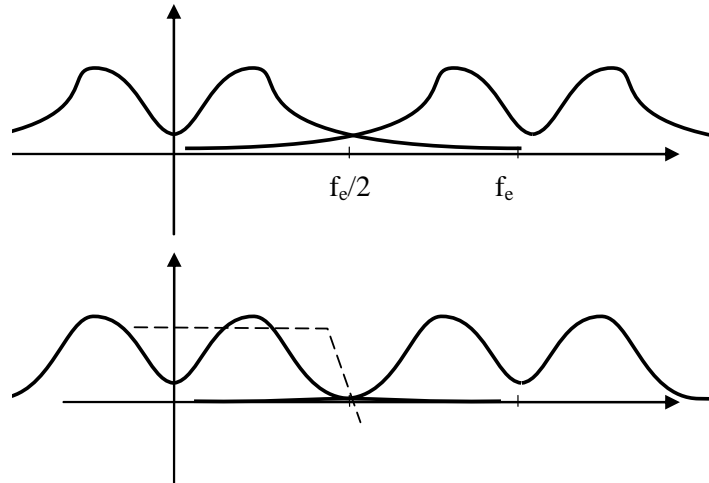


Figure 14 : Effet du filtre anti-repliement

Bien souvent la période d'échantillonnage de l'asservissement est relativement lente (quelques secondes) et il est alors impossible de réaliser un filtre de constante de temps aussi faible. On procède alors à un sur-échantillonnage : le signal est filtré, échantillonné à haute fréquence  $T'_e$  puis filtré numériquement et sous-échantillonné à la période  $T_e$  beaucoup plus grande que  $T'_e$  comme illustré sur la

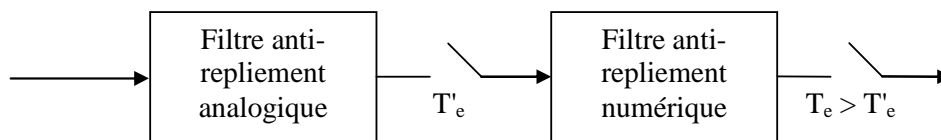


Figure 15 : Sur échantillonnage

Un simple filtre numérique de type moyenne mobile est souvent suffisant.

### 4.5 Discrétisation de fonction de transfert continues

Une manière classique pour réaliser un contrôle par ordinateur est de commencer par synthétiser un correcteur continu avec les outils de l'automatique classique puis de trouver un correcteur discret qui "approche" convenablement le correcteur continu.

#### 4.5.1 Approximation numérique d'équation différentielle

Supposons que le correcteur continu que l'on a synthétisé soit du type PD :

$$\text{PID}(s) = \frac{e(s)}{\varepsilon(s)} = k_p + k_D \cdot s \quad (26)$$

Il s'agit de calculer les commandes  $e(k)$  successives issues du correcteur en fonction des erreurs  $\varepsilon(k)$ . Une approche classique et intuitive est d'approximer la dérivée par une différence :

$$e(k) = k_p \cdot \varepsilon(k) + k_D \cdot \frac{\varepsilon(k) - \varepsilon(k-1)}{T_e} \quad (27)$$

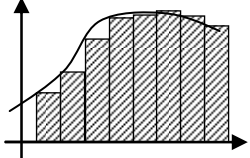
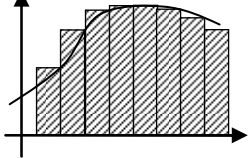
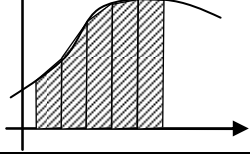
Si on calcule la transformée en  $z$  de  $e(k)$  on obtient :

$$e(z) = k_p \cdot \varepsilon(z) + k_D \cdot \frac{\varepsilon(z) - z^{-1}\varepsilon(z)}{T_e} = \left( k_p + k_D \cdot \frac{1 - z^{-1}}{T_e} \right) \cdot \varepsilon(z) \quad (28)$$

On retrouve la fonction de transfert PD(s) dans laquelle on a effectué le changement de variable  $s = \frac{1-z^{-1}}{T_e}$

De manière plus générale, l'approximation de fonction de transfert revient à effectuer l'approximation numérique d'intégration. Selon la méthode d'intégration choisie on effectue des changements de variable différents.

Tableau 4 : méthodes d'intégration

Méthode	Changement de variable	
Euler progressif (Euler's forward method)	$s = \frac{1 - z^{-1}}{z^{-1} \cdot T_e}$	
Euler rétrograde (Euler's backward method)	$s = \frac{1 - z^{-1}}{T_e}$	
Méthode des trapèzes, ou bilinéaire, ou de Tustin	$s = \frac{2}{T_e} \cdot \frac{1 - z^{-1}}{1 + z^{-1}}$	

En pratique le changement de variable  $s = \frac{1 - z^{-1}}{z^{-1} \cdot T_e}$  n'est pas sans risque. On donne sur la Figure 16 la correspondance entre la zone stable du plan complexe en  $s$  et la zone correspondante après la transformation (Euler progressif, Euler rétrograde, bilinéaire). On voit notamment qu'il est possible, partant d'un système continu stable, d'obtenir des pôles dans le domaine discret en dehors du cercle de stabilité si la transformation appliquée est celle d'Euler progressif.

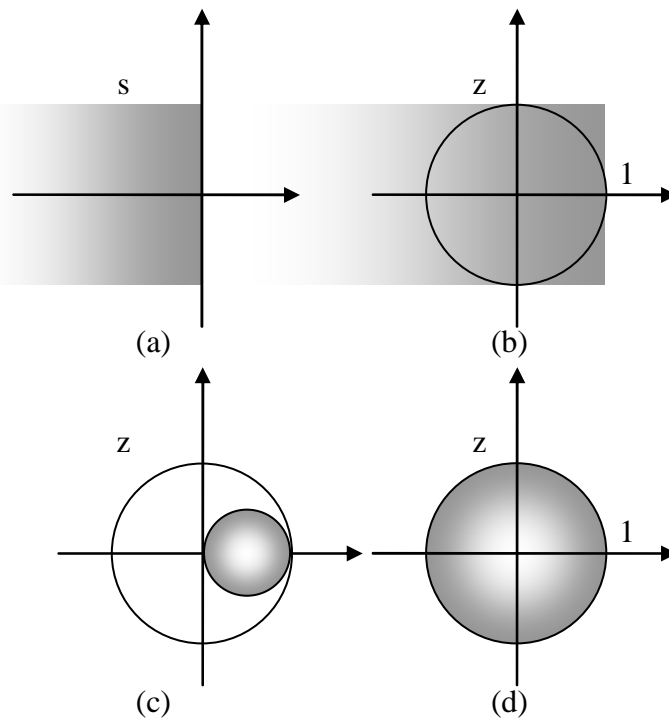


Figure 16 : correspondance des plans complexes. (a) : zone de stabilité dans le plan complexe continu en  $z$  ; (b) zone correspondante avec l'approximation d'Euler progressif ; (c) zone correspondante avec l'approximation d'Euler rétrograde ; (d) zone correspondante avec l'approximation bilinéaire.

La fonction `c2d` de Matlab permet d'obtenir l'équivalent discret d'une fonction de transfert continue avec la méthode d'intégration comme argument :

```

Matlab
sysc=tf([1 0],[1 2 5]);
Te=0.1;
% Méthode d'euler :
sysd1=c2d(sysc,Te,'zoh');
% Méthode des trapèzes :
sysd2=c2d(sysc,Te,'foh');
% Méthode bilinéaire :
sysd3=c2d(sysc,Te,'tustin');
    
```

#### 4.5.2 Superposition des réponses indicielles

Une autre méthode pour approximer une fonction de transfert discrète est de s'assurer que la réponse indicielle de sa version discrétisée correspond exactement lors des temps d'échantillonnage.

### 4.5.3 Correspondances pôles - zéros

La dernière méthode est de partir d'une décomposition de la fonction de transfert continue en pôles et zéros et d'effectuer la conversion pôles à pôles et zéros par zéros. La correspondance, selon que le pôle est réel ou complexe est donnée ci-dessous :

$$\begin{aligned} (s + a) &\rightarrow (z - e^{-a \cdot T_c}) \\ (s + a)^2 + b^2 &\rightarrow z^2 - 2 \cdot e^{-a \cdot T_c} \cdot \cos(b \cdot T_c) \cdot z + e^{-2 \cdot a \cdot T_c} \end{aligned} \quad (29)$$

### 4.5.4 Bilan

Aucune méthode n'est parfaite et on n'aura jamais en même temps :

- même réponse indicielle
- même réponse impulsionnelle
- approximation (Euler, Tustin, etc...) de l'équation différentielle
- correspondance des pôles et des zéros
- correspondance des réponses fréquentielles

Ceci étant dit le choix de la période d'échantillonnage est primordial. Si la fréquence d'échantillonnage est beaucoup plus faible que la dynamique de la fonction de transfert à approximer la différence entre chaque méthode sera minimale.



## 5. IDENTIFICATION DES SYSTEMES ECHANTILLONES

Dans le présent chapitre on donne les bases des méthodes d'identification. Pour une approche complète, voir le cours de troisième année "Identification avion" [6] et les ouvrages [1] et [7].

Même si l'on dispose d'un modèle théorique du processus que l'on souhaite commander il est presque toujours nécessaire de l'affiner en procédant à des essais sur le système réel. Bien souvent le seul modèle dont l'on dispose est ce modèle issu de l'expérimentation. Différentes méthodes s'appliquent selon l'objectif recherché et le niveau de précision exigé. On distinguera :

- les modèles paramétriques
- les modèles simples
- les modèles non paramétriques

### 5.1 Identification d'un modèle ARX par la méthode des moindres carrés récursifs

A partir de cet exemple, le plus simple qui soit, on pose les bases de toutes les méthodes d'identification.

On considère un système d'entrée  $u(t)$  et de sortie  $y(t)$  représenté par l'équation de récurrence suivante :

$$y(t) + a_1 \cdot y(t-1) + a_2 \cdot y(t-2) + \dots + a_n \cdot y(t-n) = b_0 \cdot u(t) + b_1 \cdot u(t-1) + b_2 \cdot u(t-2) + \dots + b_m \cdot u(t-m) \quad (30)$$

L'objectif est, à partir d'essais et de mesures, d'identifier les meilleures valeurs possibles pour les coefficients  $a_i$  et  $b_i$ .

L'équation (30) peut être réécrite pour mettre en évidence la *prédiction* de la sortie  $y(t)$  à partir des mesures passées :

$$y(t) = -a_1 \cdot y(t-1) - a_2 \cdot y(t-2) - \dots - a_n \cdot y(t-n) + b_0 \cdot u(t) + b_1 \cdot u(t-1) + b_2 \cdot u(t-2) + \dots + b_m \cdot u(t-m) \quad (31)$$

Soit :

$$y(t) = \varphi^T(t) \cdot \theta(t)$$

avec :

$$\begin{cases} \varphi^T(t) = [-y(t-1) \ \dots \ -y(t-n) \ , \ u(t-1) \ \dots \ u(t-m)] \\ \theta(t) = [a_1 \ \dots \ a_n \ , \ b_1 \ \dots \ b_m] \end{cases} \quad (32)$$

Le but est de trouver le meilleur ensemble de paramètres  $\theta(t)$  : celui qui, à partir d'un ensemble de mesures  $\varphi(t)$  permet de prédire avec la meilleure précision la sortie effectivement mesurée.

Supposons qu'à la suite d'une expérience on dispose d'un ensemble de mesures  $\{y(t), u(t)\}_{t=1:N}$  et que l'ensemble des paramètres  $\theta$  n'est pas bien connu. On construit la séquence des *sorties prédites* :

$$\hat{y}(t) = \varphi^T(t) \cdot \theta(t) \quad (33)$$

Le paramètre  $\theta$  est celui qui minimise une norme quadratique :

$$\theta = \arg \max \left( \frac{1}{N} \sum_{t=1}^N (y(t) - \hat{y}(t))^2 \right) = \arg \max \left( \frac{1}{N} \sum_{t=1}^N (y(t) - \varphi^T(t) \cdot \theta(t))^2 \right) \quad (34)$$

Compte tenu du fait que l'équation est linéaire au départ et que le terme à minimiser est du type quadratique la solution est simplement obtenue par une pseudo-inverse :

$$\theta = \left[ \sum_{t=1}^N (\varphi(t) \cdot \varphi^T(t)) \right]^{-1} \sum_{t=1}^N (\varphi(t) \cdot y(t)) \quad (35)$$

On obtient donc la solution classique des moindres carrés. La formule (35) est le plus souvent utilisée sous sa forme récursive :

$$\theta(t) = \theta(t-1) + F(t) \cdot \varphi(t) \cdot (y(t) - \theta^T(t-1) \cdot \varphi(t))$$

avec

$$F(t)^{-1} = F(t-1)^{-1} + \varphi(t) \cdot \varphi(t)^T \quad (36)$$

ou

$$F(t) = F(t-1) - \frac{F(t-1) \cdot \varphi(t) \cdot \varphi(t)^T \cdot F(t-1)}{1 + \varphi(t)^T \cdot F(t-1) \cdot \varphi(t)}$$

(On laisse au lecteur le plaisir d'effectuer lui-même les démonstrations, qu'il peut du reste trouver dans [1] et [7]).



## 5.2 L'identification paramétrique en général et en pratique

L'exemple précédent a permis de mettre en évidence tous les aspects de la méthodologie classique de l'identification :

- il a fallu choisir un ensemble de mesures  $\{y(t), u(t)\}_{t=1:N}$ . Le choix du signal d'excitation  $u(t)$  est en pratique primordial,
- il a fallu choisir une structure pour le modèle à identifier (ici les ordres  $n$  et  $m$ ),
- il a fallu choisir un critère (ici un critère quadratique sur l'erreur entre la sortie et la sortie prédite),
- il a fallu choisir un algorithme et le paramétrer (ici l'algorithme des moindres carrés récursifs qu'il a fallu initialiser)

Enfin un point non traité dans le chapitre précédent concerne la validation :

- il aurait fallu déterminer une méthode pour comparer différents modèles et choisir le meilleur

### 5.2.1 Choix du signal d'excitation, traitement des mesures

De manière générale le système que l'on doit identifier doit être excité dans tout le domaine fréquentiel qui nous intéresse. Le signal classiquement utilisé est une séquence pseudo binaire aléatoire (Pseudo Random Binary Signal) tel celui représenté sur la Figure 17.

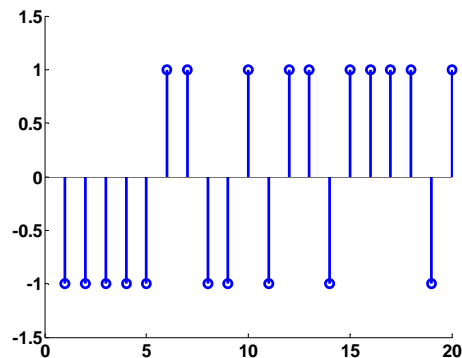


Figure 17 : exemple d'une séquence PRBS

Le signal d'excitation doit se superposer au signal de commande correspondant au point de fonctionnement étudié. Son amplitude doit être faible par rapport à l'amplitude des commandes du système (quelques %).

Une fois le système excité et les mesures effectuées il faut supprimer les valeurs moyennes des mesures.

### 5.2.2 Choix d'une structure de modèle

Le modèle étudié dans le chapitre précédent est de type ARX. On peut le noter sous forme polynomiale :

$$A \cdot y = B \cdot u \quad (37)$$

Ce modèle correspond en fait à un parti-pris sur la manière dont on traite les perturbations. En réalité on prend en compte un signal d'erreurs possibles  $e$  (de moyenne nulle) tels que :

$$A \cdot y = B \cdot u + e \quad (38)$$

Si l'on fait cette hypothèse de modélisation alors la méthode optimale est celle des moindres carrés. Malheureusement cette manière de prendre en compte les perturbations est bien restrictive. D'autres modélisations sont possibles, adaptées à d'autres algorithmes, par exemple le modèle OE :

$$y = \frac{B}{F} \cdot u + e \quad (39)$$

Il est aussi possible de "structurer" le modèle de bruit comme par exemple dans les modèles ARMAX ou BJ :

$$\begin{aligned} A \cdot y &= B \cdot u + C \cdot e \\ y &= \frac{B}{F} \cdot u + \frac{C}{D} \cdot e \end{aligned} \quad (40)$$

La structure du modèle étant choisie il faut choisir l'ordre de ses différents polynômes.

### 5.2.3 Choix d'un critère à minimiser

Dans la plupart des cas le critère à minimiser est l'erreur quadratique entre la sortie et la sortie prédite à un coup (one step ahead prediction). Ceci n'est bien évidemment pas restrictif et peut être généralisé à la prédiction à  $N$  coups.

### 5.2.4 Choix d'un algorithme

Même si chaque structure de modèle est généralement associée à un algorithme "optimal" celui-ci doit généralement se paramétrer.

Prenons par exemple l'algorithme des moindres carrés récursifs tel que présenté dans (36). Cette équation peut être modifiée en intégrant un "facteur d'oubli"  $\lambda$  :

$$\begin{aligned} \theta(t) &= \theta(t-1) + F(t) \cdot \varphi(t) \cdot (y(t) - \theta^T(t-1) \cdot \varphi(t)) \\ \text{avec} & \\ F(t)^{-1} &= \lambda \cdot F(t-1)^{-1} + \varphi(t) \cdot \varphi(t)^T \end{aligned} \quad (41)$$

Ce paramètre (typiquement entre 0.95 et 0.99) permet, comme son nom l'indique, "d'oublier" les anciennes mesures et de se concentrer sur les mesures les plus récentes.

### 5.2.5 Validation

Une fois l'algorithme d'identification appliqué et le modèle obtenu (ce qui prend généralement quelques secondes avec un ordinateur moderne) il reste à expertiser celui-ci. Selon la "qualité" du modèle obtenu on remet en cause l'une ou l'autre des étapes (signal d'excitation, structure du modèle, algorithme) et on réitère jusqu'à obtenir un modèle satisfaisant.

En pratique on essaie un certain nombre de modèles avant de choisir le "meilleur".

Le premier critère de qualité est la capacité du modèle à prédire effectivement la sortie (sortie prédite à un coup, sortie prédite à N coups). Il faut pour cela utiliser un jeu de mesures non utilisé pour l'identification. (En pratique le jeu de mesures initiales est coupé en deux : une partie pour l'identification et une partie pour la validation).

Le second critère de qualité est l'analyse du résidu : la composition spectrale de l'erreur de prédiction.

### 5.2.6 Utilisation de logiciels

En pratique il est commode d'utiliser un célèbre outil de calcul scientifique et sa toolboxe d'identification.

Comme les calculs proprement dit ne prennent que quelques secondes il est facile de réaliser un grand nombre d'identifications et de stocker les différents modèles pour les comparer ensuite. On est rapidement noyé sous le grand nombre de modèles possibles...

## 5.3 Identification "simple"

Il est souvent suffisant d'obtenir un modèle "simple" de type premier ordre et un simple essai indiciel convient souvent. Le système est excité avec un échelon unité et on cherche le meilleur modèle du premier ordre correspondant.

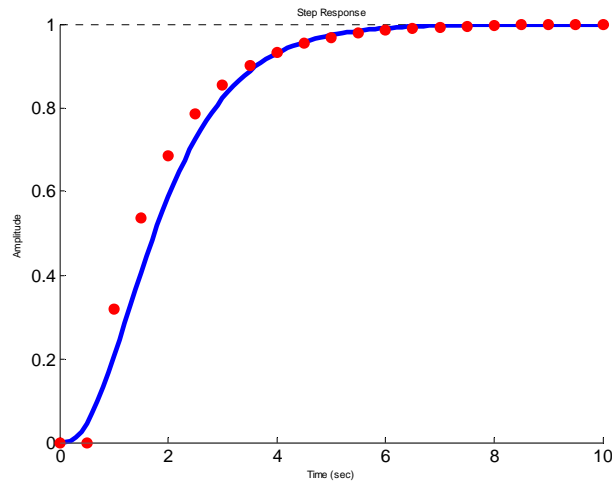


Figure 18 : identification sur un essai indiciel

Dans l'exemple de la Figure 18 on superpose le résultat d'une mesure (en bleu) avec la simulation d'un modèle discret avec une période de 0.5s. Le modèle cherché est du premier ordre avec un retard pur  $d$  :

$$y(z) = \frac{(1 + c_1) \cdot z^{-d}}{1 + c_1 \cdot z^{-1}} \cdot u(z) \quad (42)$$

Il s'agit de trouver l'entier  $d$  et le nombre réel  $c_1$  tels que la réponse indicielle du modèle corresponde pour le mieux aux mesures issues de l'essai. Ceci peut se formaliser par exemple de la manière suivante :

$$(c_1 \quad d_1) = \arg \min \sum_{t=1}^N (y_{\text{modèle}}(t) - y_{\text{mesure}}(t))^2 \quad (43)$$

Cette optimisation peut se faire à la main ou grâce à une procédure automatisée (avec le célèbre outil de calcul scientifique).

Dans le cas où le système comprend naturellement une intégration (par exemple la commande d'un moteur à courant continu par une tension et la mesure de l'angle de rotation) il est utile de prendre en compte explicitement cette intégration dans le modèle :

$$y(z) = \frac{1}{1 - z^{-1}} \cdot \frac{(1 + c_1) \cdot z^{-d}}{1 + c_1 \cdot z^{-1}} \cdot u(z) \quad (44)$$

La procédure est la même mais au lieu de prendre en compte la sortie  $y(t)$  on prend en compte la dérivée  $y'(t)$  :

$$y'(t) = y(t) - y(t-1) \quad (45)$$

## 5.4 Identification non paramétrique

Pour terminer ce panorama rapide sur les méthodes d'identification il faut parler des méthodes non paramétriques : celles qui permettent de tracer la réponse impulsionnelle ou la réponse fréquentielle du système.

### 5.4.1 Estimation de la réponse impulsionnelle

La réponse impulsionnelle d'un système peut être obtenue simplement en appliquant une entrée de type "impulsion" au système et d'en mesurer les conséquences.

La seconde méthode consiste à utiliser la méthode des moindres carrés (ou une méthode dérivée) comme vue au chapitre 5.1. La réponse impulsionnelle d'un système peut en effet être vue comme la fonction de transfert particulière où le dénominateur est de degré nul et le numérateur comporte un nombre infini (en pratique : grand) d'éléments ( $n = 0$  et  $m$  grand) :

$$y(t) = b_0 \cdot u(t) + b_1 \cdot u(t-1) + b_2 \cdot u(t-2) + \dots + b_m \cdot u(t-m) \quad (46)$$

La dernière méthode, qui sera développée ici est l'analyse d'intercorrélation.

Par définition l'intercorrélation  $R_{yu}$  entre deux signaux est :

$$R_{yu}(\tau) = E[(y(t) - E(y(t))) \cdot (u(t-\tau) - E(u(t-\tau)))] \quad (47)$$

Dans cette formule l'espérance mathématique peut être approchée grâce à la loi des grands nombres :

$$R_{yu}(\tau) \approx R_{yu}^N(\tau) = \lim_{N \rightarrow \infty} \frac{1}{N} \sum_{t=1}^N [(y(t) - m_y) \cdot (u(t-\tau) - m_u)] \quad (48)$$

On s'intéresse maintenant à un système tel que décrit par l'équation impulsionnelle (46) et on suppose que le signal d'entrée est un bruit blanc. Le signal d'entrée vérifie donc :

$$\begin{cases} \tau \neq 0 & R_{uu}(\tau) = 0 \\ \tau = 0 & R_{uu}(\tau) = R \end{cases} \quad (49)$$

Il vient alors naturellement :

$$b_k \approx \frac{1}{R} R_{yu}(k) \quad (50)$$

### 5.4.2 Estimation de la réponse fréquentielle

On utilise ici le spectre d'intercorrélation. Par définition le spectre d'intercorrélation entre deux signaux est :

$$\Phi_{yu}(\omega) = \sum_{\tau=-\infty}^{\infty} R_{yu}(\tau) \cdot e^{-i\omega\tau} \quad (51)$$

On suppose que le signal d'entrée  $u(t)$  et le signal de sortie  $y(t)$  sont reliés par une fonction de transfert linéaire du type :

$$y(t) = F \cdot u(t) + v(t) \quad (52)$$

où  $v(t)$  est une perturbation indépendante de  $u(t)$ . On a alors les relations :

$$\Phi_{yu}(\omega) = F(j\omega) \cdot \Phi_{uu}(\omega) \quad (53)$$

Il reste à estimer les spectres d'intercorrélation pour remonter à la réponse fréquentielle  $F(j\omega)$  recherchée. Une idée naturelle est d'utiliser la relation (48) mais cela marche très mal dans presque tous les cas, même en choisissant  $N$  assez grand. L'amélioration est d'utiliser une "fenêtre" pour favoriser les valeurs faibles de  $\tau$  par rapports aux valeurs élevées. La solution classique utilisant la *fenêtre de Hamming* donne :

$$\Phi_{yu}^N(\omega) = \sum_{\tau=-\gamma}^{\gamma} R_{yu}^N(\tau) \cdot w_{\gamma}(\tau) \cdot e^{-i\omega\tau} \quad (54)$$

avec

$$\begin{cases} w_{\gamma}(\tau) = \frac{1}{2} \left( 1 + \cos\left(\frac{\pi \cdot \tau}{\gamma}\right) \right) & |\tau| < \gamma \\ w_{\gamma}(\tau) = 0 & |\tau| > \gamma \end{cases}$$

On utilise ensuite la relation (51) pour estimer la réponse fréquentielle :

$$F(j \cdot \omega) = \frac{\Phi_u^N(\omega)}{\Phi_{yu}^N(\omega)} \quad (55)$$





## 6. SYSTEMES BOUCLES

### 6.1 Fonction de transfert en boucle fermée

Un système et sa commande par ordinateur peut être représenté par un ensemble de fonctions de transfert formant un schéma-bloc plus ou moins compliqué (Figure 19).

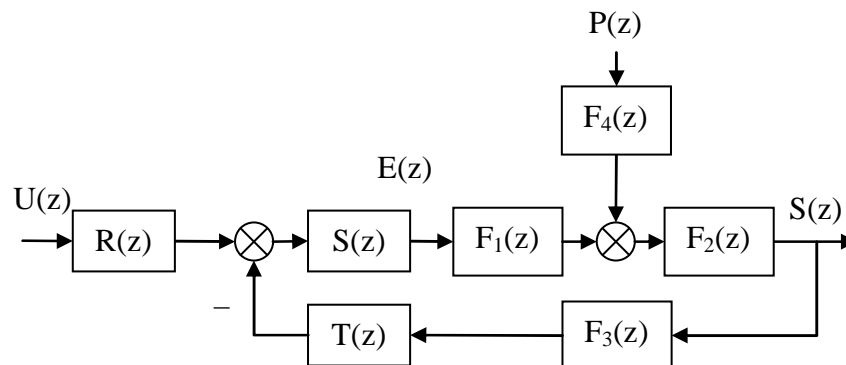


Figure 19 : système discret commandé par ordinateur. Le correcteur est représenté par les fonctions R,S,T, le système par  $F_1$  et  $F_2$ , le modèle de perturbations par  $F_4$  et le modèle des actionneurs par  $F_4$ .

Quelle que soit la structure d'un système asservi il est possible de se ramener à une forme canonique comprenant une fonction de transfert en boucle ouverte en combinant judicieusement les fonctions de transfert du procédé, du régulateur et des perturbations. La Figure 20 représente par exemple le système de la Figure 19 où  $F$  est une combinaison des fonctions de transfert du système, du régulateur, des perturbations et des capteurs.

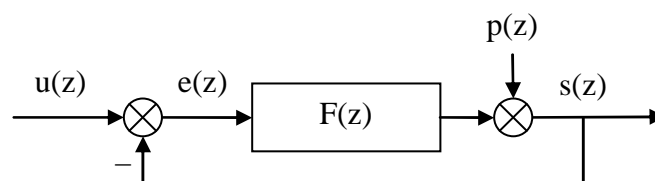


Figure 20 : système discret, notations.

Par défaut on considère une contre-réaction négative et une perturbation additive sur la sortie. La fonction de transfert  $F$  est appelée *fonction de transfert en boucle ouverte*. Comme pour les systèmes continus la *fonction de transfert en boucle fermée* est obtenue simplement à partir de  $F$  (58).

$$\begin{aligned}
 F(z) &= \frac{s(z)}{e(z)} = \frac{B(z^{-1})}{A(z^{-1})} \\
 G(z) &= \frac{s(z)}{u(z)} = \frac{B(z^{-1})}{A(z^{-1}) + B(z^{-1})}
 \end{aligned}
 \tag{56}$$

## 6.2 Erreur statique

L'erreur statique du système en boucle fermée, comme pour le système en boucle ouverte, est simplement le rapport des coefficients.

$$G_0 = \frac{\sum_{i=0}^{i=m} b_i}{1 + \sum_{i=0}^{i=m} b_i + \sum_{i=1}^{i=n} a_i}
 \tag{57}$$

Si l'on souhaite que l'asservissement confère un gain statique unitaire au système en boucle fermée (c'est-à-dire que l'on force la sortie  $s(k)$  à tendre vers la valeur de consigne  $u(k)$ ) il faut que ce rapport de coefficients soit égal à un. On montre facilement que cela est obtenu si la fonction de transfert  $F(z)$  comporte un intégrateur :

$$F(z) = \frac{s(z)}{e(z)} = \frac{1}{1 - z^{-1}} \cdot \frac{B(z)}{A'(z)}
 \tag{58}$$

## 6.3 Rejet de perturbations

La fonction de transfert entre la perturbation  $p(z)$  et la sortie  $y(z)$  est appelée fonction de sensibilité perturbation-sortie, notée  $S_{yp}$ .

$$S_{yp}(z) = \frac{s(z)}{p(z)} = \frac{A(z^{-1})}{A(z^{-1}) + B(z^{-1})}
 \tag{59}$$

Si l'on souhaite qu'une perturbation de type échelon soit parfaitement rejetée il faut que  $A(1) = 0$ , on retrouve donc la condition (58) à savoir que le système doit comporter une intégration dans la boucle directe. On peut être en outre être amené à spécifier que les perturbations ne soient pas amplifiées dans une certaine bande de fréquence. On rajoute donc une contrainte du type  $|S_{yp}(e^{j2\pi f})| < S_{\max}$  pour  $f$  appartenant à un certain domaine de  $[0 \ 0.5]$ .

## 6.4 Stabilité des systèmes à simple gain de rétroaction

Les critères de Routh et de Jury (donnés dans le chapitre 3.2.7 p13) ou le calcul numérique des pôles permettent de déterminer la stabilité du système en boucle fermée si l'on connaît sa fonction de transfert.

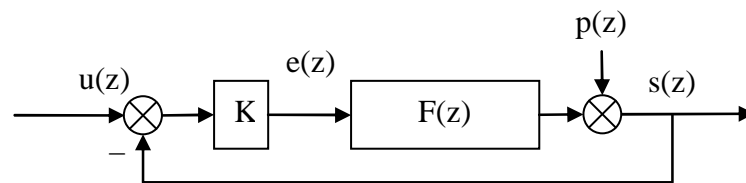


Figure 21 : système bouclé par un simple gain.

On s'intéresse ici au cas particulier où l'asservissement est réalisé par un simple gain de rétroaction  $K$  (Figure 21). On cherche la valeur de  $K$  qui amène le système en limite de stabilité. L'outil adapté ici est le lieu d'Evans ou lieu des racines (root locus) qui permet de visualiser l'évolution des pôles du système bouclé lorsque le gain  $K$  varie.

Prenons l'exemple d'un système comportant trois pôles et deux zéros. L'exemple est donné sous forme de script Matlab :

### Matlab

```
% Définition d'un système discret :
sys=zpk([-2.037 -0.1406],[0.6065 0.3679 0.3679],1,1)
%Tracé du lieu des racines
rlocus(sys);
```

Le script précédent produit le tracé de la Figure 22. On pourrait y lire que la limite de stabilité est obtenue pour  $K=0.2$  environ (détail de la Figure 23).

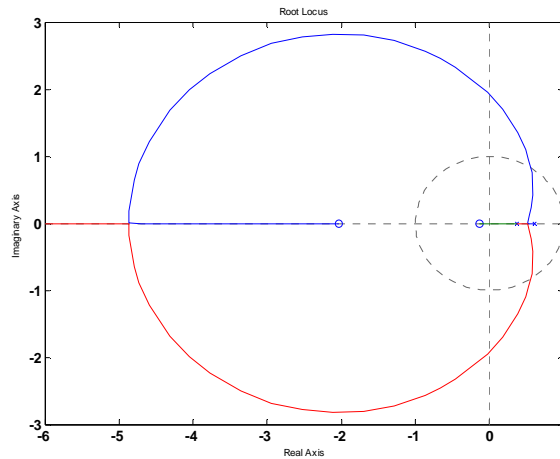


Figure 22 : lieu des racines

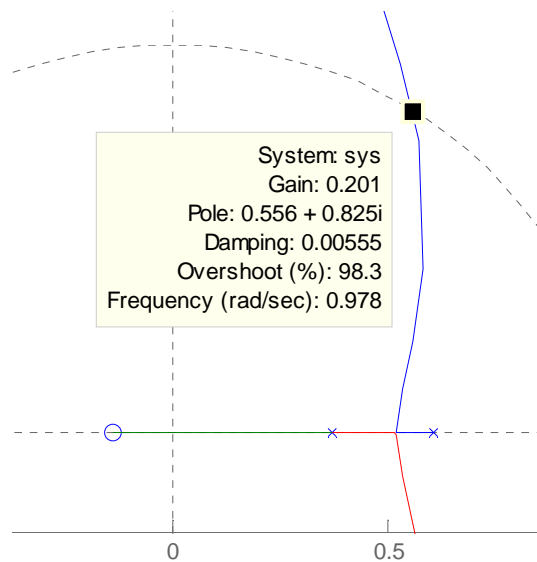


Figure 23 : lieu des racines : détail de la limite de stabilité

On voit sur la Figure 22 que le lieu des racines d'un système échantillonné ressemble à celui d'un système continu :

- on commence par tracer les pôles ( $\times$ ) et les zéros ( $o$ ) du système en boucle ouverte.
- les pôles du système en boucle fermé décrivent un lieu qui part des pôles (pour  $K = 0$ ) et arrive aux zéros (pour  $K = \infty$ )
- lorsque il y a  $n-m$  plus de pôles que de zéros le système comporte  $n-m$  branches asymptotiques.

On ne décrira pas ici les diverses propriétés du lieu des racines qui permettent de le tracer à la main puisque il est facile à obtenir avec un ordinateur.

Il peut aussi être utilisé pour régler "à la main" le gain permettant d'obtenir un ou plusieurs modes dominants ayant certaines propriétés de fréquence et d'amortissement. On voit sur la Figure 24 que le gain 0.017 permet d'obtenir un pôle double d'amortissement 0.706 et de fréquence 0.65 rad/s et un pôle simple de fréquence 1.63 rad/s.

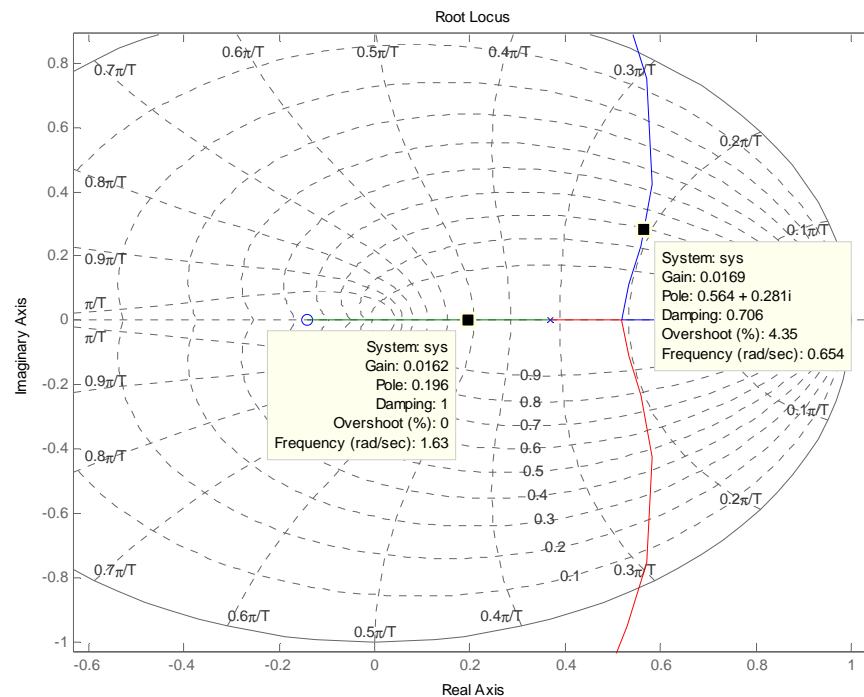


Figure 24 : utilisation du lieu des racines pour régler le gain de la boucle fermée

L'outil rltool de Matlab permet de nombreuses manipulations graphiques de ce type.

## 6.5 Analyse des systèmes bouclés dans le domaine fréquentiel : stabilité

Comme dans le cas continu il est possible de prévoir des propriétés, en particulier la stabilité, d'un système en boucle fermée à partir de l'examen de la fonction de transfert en boucle ouverte.

On ne redémontrera pas ici le théorème de Nyquist mais on se contente de rappeler l'énoncé.

On rappelle que le *lieu de Nyquist* est le lieu de  $F(j2\pi f)$  dans le plan complexe pour  $f$  variant de 0 à  $+\infty$ . A noter que le lieu pour la zone  $[0, 0.5]$  est symétrique de celui pour la zone  $[0.5, 1]$ . Pour cette raison on ne représente généralement que le lieu pour  $f$  appartenant à  $[0, 0.5]$ .

### 6.5.1 Théorème de Nyquist dans le cas où le système est stable en boucle ouverte (critère du revers)

Le système est stable en boucle fermée si le lieu de Nyquist de la fonction de transfert en boucle ouverte parcouru dans le sens des fréquences croissantes de 0 à 0.5 laisse le point critique  $(-1,0)$  à gauche.

La Figure 25 représente par exemple le lieu de Nyquist d'une fonction de transfert en boucle ouverte qui sera stable en boucle fermée.

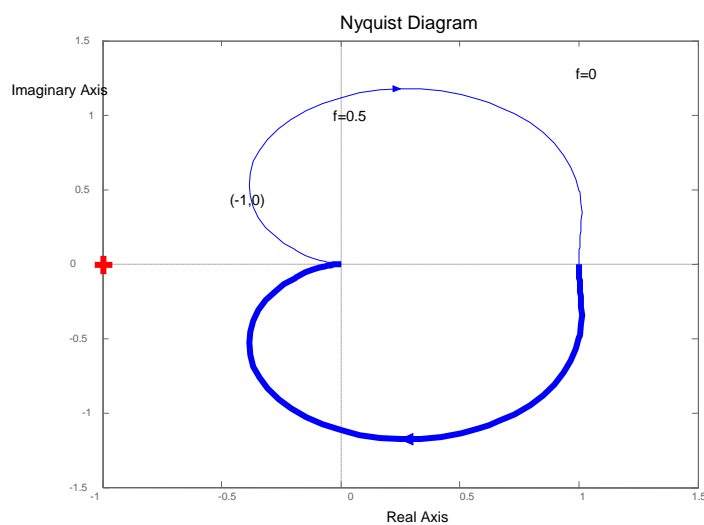


Figure 25 : lieu de Nyquist d'un système stable

### 6.5.2 Théorème de Nyquist dans le cas où le système est instable en boucle ouverte

Si le système en boucle ouverte comporte  $N$  pôles instables il sera stable en boucle fermée si le lieu de Nyquist de la fonction de transfert en boucle ouverte parcouru dans le sens des fréquences croissantes de  $-0.5$  à  $0.5$  entoure le point critique  $(-1,0)$   $N$  fois dans le sens des aiguilles d'une montre..

### 6.5.3 Interprétations du lieu de Nyquist

Comme dans le cas continu, la distance à laquelle on passe du point critique renseigne sur la robustesse du système bouclé. La distance sera exprimée de différente manière (Figure 26) :

- marge de module  $\Delta M$  : plus petite distance entre le lieu des racines et le point critique,

- marge de gain  $\Delta G$
- marge de phase  $\Delta P$

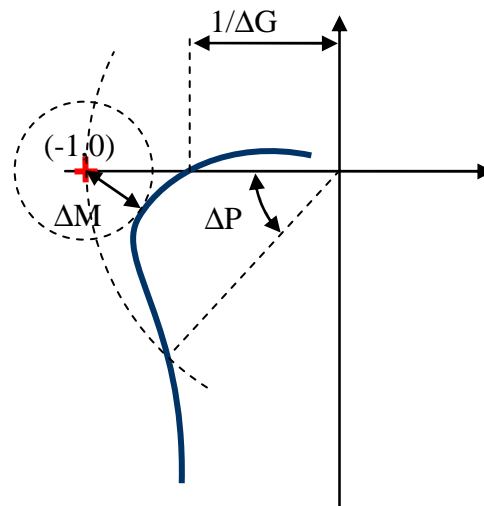


Figure 26 : Marges de stabilité

Les valeurs typiques de ces marges sont données ci-après :

- $\Delta G \geq 6\text{dB}$
- $\Delta M \geq -6\text{dB}$
- $30^\circ \leq \Delta P \leq 60^\circ$

La distance entre le point critique et le lieu des racines est la norme de  $1+F(j2\pi f)$ . On peut donc exprimer la marge de module en fonction de la fonction de sensibilité :

$$\begin{aligned} \Delta M &= \min |1 + F(j \cdot \omega)|_{\omega \in [0, \pi]} \\ \Delta M &= \min \left| 1 + \frac{B(j \cdot \omega)}{A(j \cdot \omega)} \right|_{\omega \in [0, \pi]} = \max \left| \frac{A(j \cdot \omega)}{A(j \cdot \omega) + B(j \cdot \omega)} \right|_{\omega \in [0, \pi]} \quad (60) \\ \Delta M &= \max |S_{yp}(j \cdot \omega)|_{\omega \in [0, \pi]}^{-1} \\ \Delta M_{\text{dB}} &= -\max |S_{yp}(j \cdot \omega)|_{\text{dB}} \end{aligned}$$

Si l'on trace l'évolution de  $S_{yp}$  en fonction de  $f$  on obtient donc la valeur de la marge de module.





# 7. METHODES DE REGULATION DES SYSTEMES ECHANTILLONES

## 7.1 Introduction

Les bases théoriques permettant de concevoir le correcteur pour un système échantillonné ont été posées dans les chapitres précédents. Ce chapitre fait le bilan des connaissances acquises et donne quelques solutions pour calculer des correcteurs à partir d'un cahier des charges précis.

Les systèmes que l'on cherche à commander sont généralement, par nature, continus. La réalisation d'un correcteur par un ordinateur implique que l'on génère une équation de commande discrète (généralement une équation récurrente). L'obtention de cette équation de commande peut se faire de deux manières :

- la première méthode consiste à d'abord synthétiser un correcteur continu (en utilisant les outils classiques de l'automatique et la simulation) et discrétiser ensuite ce correcteur. C'est l'objet du chapitre 7.2.2.
- la seconde méthode consiste à tenir la représentation discrète du système continu associé à son interface numérique (convertisseurs analogique-numérique et numérique-analogique) comme expliqué dans le chapitre 4.5 puis synthétiser directement le correcteur discret. Cela sera expliqué dans le chapitre 7.3 et suivants.

Ce choix peut être résumé par la figure suivante :

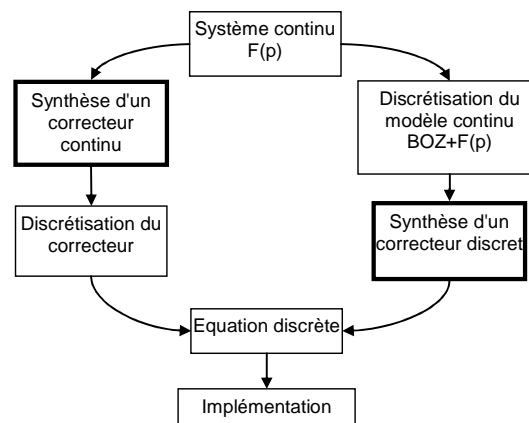


Figure 27 : Réalisation d'un correcteur discret.

Au-delà des affaires de goût, le choix de la méthode sera souvent dicté par l'application. En effet, la première méthode est d'autant meilleure que le correcteur discrétisé approche correctement le correcteur continu. Cela implique que le temps d'échantillonnage soit très faible par rapport aux constantes de temps du système

bouclé. Dans le cas où les contraintes temps réel sont fortes (capacités de calcul partagées par exemple) il sera préférable de synthétiser un correcteur discret.

## 7.2 Structure d'un système commandé par ordinateur, hypothèses

Par la suite on se ramènera toujours à un système tel que celui représenté sur la Figure 28 en cherchant un correcteur  $K(z)$  linéaire. On montre que quel que soit le correcteur linéaire on peut se ramener à la forme représenté sur la Figure 29 (forme RST).

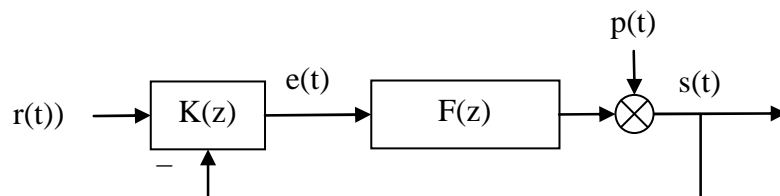


Figure 28 : système bouclé, forme générale.

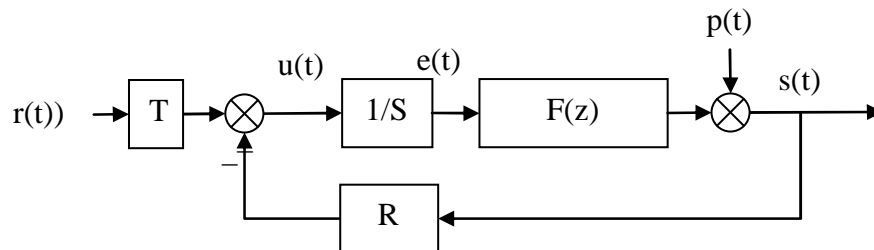


Figure 29 : système bouclé, forme canonique.

### 7.2.1 Discrétisation, choix de la période d'échantillonnage

Du point de vue du ordinateur, l'extérieur (le procédé) est représenté par la fonction de transfert  $F(z)$ . L'hypothèse importante est que la commande (sortie du ordinateur) et la mesure (entrée du ordinateur) sont échantillonnées de manière synchrone à la même période  $T_e$ .

Le choix judicieux de la fréquence d'échantillonnage est très important. En pratique on choisit cette fréquence en fonction de la bande passante du système en **boucle fermée** :

$$f_e = (6 \text{ à } 25) f^{BF} \quad (61)$$

### 7.2.2 Discrétisation d'un correcteur continu

La première méthode pour obtenir les équations d'un correcteur numérique est de faire sa synthèse avec les outils de l'automatique continue puis de discrétiser la fonction de transfert du correcteur continu obtenu (voir pour cela le chapitre 4.2 *Echantillonnage des systèmes continus*).

On va voir par la suite que la synthèse directe d'un correcteur discret offre plus de degré de liberté et permet en général de remplir un cahier des charges plus complet.

## 7.3 Régulateur PID numérique

Le régulateur le plus populaire est le régulateur PID. Nous allons voir dans un premier temps comment on génère un régulateur PID à partir des équations bien connues du régulateur PID continu et comment ces équations peuvent être généralisées pour obtenir plus de degrés de liberté pour le réglage.

### 7.3.1 Cas 1

Un régulateur PID continu a une fonction de transfert du type :

$$K_{\text{PID}}(s) = K \cdot \left( 1 + \frac{1}{T_i \cdot s} + \frac{T_d \cdot s}{1 + \frac{T_d}{N} \cdot s} \right) \quad (62)$$

La discrétisation de ce correcteur continu par la méthode d'Euler (Tableau 4 page 31) se fait en remplaçant la variable de Laplace  $s$  par le terme  $\frac{1-q^{-1}}{q^{-1} \cdot T_e}$ . On obtient :

$$K_{\text{PID}}(q) = K \cdot \left( 1 + \frac{1}{T_i \cdot \frac{1-q^{-1}}{T_e}} + \frac{T_d \cdot \frac{1-q^{-1}}{T_e}}{1 + \frac{T_d}{N} \cdot \frac{1-q^{-1}}{T_e}} \right) \quad (63)$$

En réarrangeant on obtient la forme générale RST (Figure 30):

$$K_{PID}(q) = K \cdot \frac{r_0 + r_1 \cdot q^{-1} + r_2 \cdot q^{-2}}{(1 - q^{-1}) \cdot (1 + s_1' \cdot q^{-1})}$$

$$\begin{cases} r_0 = K \cdot \left( 1 + \frac{T_e}{T_i} + \frac{N \cdot T_d}{T_d + N \cdot T_e} \right) \\ r_1 = -K \cdot \left( 1 + \frac{T_d}{T_d + N \cdot T_e} + \frac{T_e}{T_i} \cdot \frac{T_d}{T_d + N \cdot T_e} + \frac{N \cdot T_d}{T_d + N \cdot T_e} \right) \\ r_2 = K \cdot \left( \frac{T_d}{T_d + N \cdot T_e} \right) \\ s_1' = \frac{T_d}{T_d + N \cdot T_e} \end{cases} \quad (64)$$

Comme on le voit le correcteur obtenu comporte autant de degrés de liberté que le correcteur PID analogique. On reconnaît au dénominateur le terme  $(1 - q^{-1})$  en facteur qui assure l'effet d'intégration.

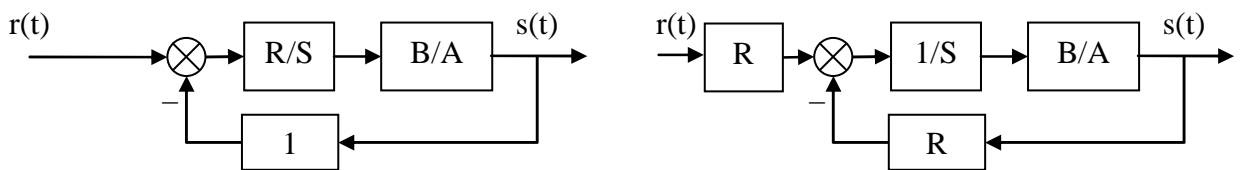


Figure 30 : PID sous forme RST

La fonction de transfert en boucle fermée obtenue est donnée par l'équation suivante :

$$H_{BF} = \frac{B \cdot R}{A \cdot S + B \cdot R} = \frac{B \cdot R}{P} \quad (65)$$

De manière générale le polynôme caractéristique  $P$  fixe les performances visées en boucle fermée. Le calcul de  $R$  et  $S$ ,  $P$  étant fixé et  $A$  et  $B$  étant connus, est réalisé en résolvant l'équation de Bézout  $P = A \cdot S + B \cdot R$ . (voir 9.1 Annexe 1 : Résolution de l'équation polynomiale de Bezout)

On voit que le numérateur  $B \cdot R$  est imposé une fois que  $R$  et  $S$  sont trouvés. Le régulateur introduit donc naturellement des zéros dans la boucle fermée.

Traisons l'exemple entièrement dans le cas où le système est modélisé comme un système du deuxième ordre avec faible retard. La fonction de transfert du procédé est donnée par :

$$H_{BO} = \frac{B}{A} = \frac{b_1 \cdot q^{-1} + b_2 \cdot q^{-2}}{1 + a_1 \cdot q^{-1} + a_2 \cdot q^{-2}} \quad (66)$$

Le correcteur PID rajoutant deux pôles, le polynôme caractéristique en boucle fermée spécifié doit posséder quatre pôles. Le polynôme caractéristique en boucle fermée doit donc être de la forme :

$$\begin{aligned} P &= 1 + p_1 \cdot q^{-1} + p_2 \cdot q^{-2} + p_2 \cdot q^{-3} + p_2 \cdot q^{-4} \\ P &= (1 + p_1' \cdot q^{-1} + p_2' \cdot q^{-2}) \cdot (1 + \alpha_1 \cdot q^{-1}) \cdot (1 + \alpha_2 \cdot q^{-1}) \end{aligned} \quad (67)$$

Et il doit être identifié au produit  $AS + BR$  avec  $R$  et  $S$  ayant les formes données par (64) :

$$\begin{aligned} &(1 + p_1' \cdot q^{-1} + p_2' \cdot q^{-2}) \cdot (1 + \alpha_1 \cdot q^{-1}) \cdot (1 + \alpha_2 \cdot q^{-1}) \\ &= (1 + a_1 \cdot q^{-1} + a_2 \cdot q^{-2}) \cdot (1 - q^{-1}) \cdot (1 + s_1' \cdot q^{-1}) \\ &+ (b_1 \cdot q^{-1} + b_2 \cdot q^{-2}) \cdot (r_0 + r_1 \cdot q^{-1} + r_2 \cdot q^{-2}) \end{aligned} \quad (68)$$

Les coefficients  $p_1'$  et  $p_2'$  définissent un modèle du second ordre aux caractéristiques (pulsation propre  $\omega_0$  et amortissement  $\sigma$  choisis et coefficients déduits conformément au Tableau 3 page 25). Les coefficients  $\alpha_1$  et  $\alpha_2$  correspondent à des pôles auxiliaires aperiodiques. Les choix typiques pour ces coefficients sont :

$$-0.05 < \alpha_1, \alpha_2 < -0.5 \quad (69)$$

On montre que la résolution de (68) est réalisée par une inversion matricielle simple :

$$M \cdot x = p \quad (70)$$

avec

$$\begin{cases} x^T = [1 & s_1' & r_0 & r_1 & r_2] \\ p^T = [1 & p_1 & p_2 & p_3 & p_4] \end{cases}$$

et

$$M = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ a_1' & 1 & b_1 & 0 & 0 \\ a_2' & a_1' & b_2 & b_1 & 0 \\ a_3' & a_2' & 0 & b_2 & b_1 \\ 0 & a_3' & 0 & 0 & b_2 \end{pmatrix} \quad (71)$$

On montre que M est inversible à la condition que A et B soient premiers entre eux, c'est-à-dire que A et B n'aient pas de pôles et de zéros en commun.

(Le problème peut aussi être résolu en utilisant la résolution de l'équation de Bézout).

### 7.3.2 Cas 2

Comme on l'a vu le correcteur PI numérique directement transposé du correcteur PID analogique décrit dans le chapitre 7.3.1 a l'inconvénient de ramener des zéros au dénominateur de la fonction de transfert en boucle fermée (équation 65). Il est possible, grâce à la flexibilité du régulateur numérique d'imposer au correcteur de ne pas ajouter de zéros en boucle fermée. La fonction de transfert désirée en boucle fermée sera donc de la forme :

$$H_{BF} = \frac{P(1)}{B(1)} \cdot \frac{B}{P} \quad (72)$$

Le facteur  $P(1)/B(1)$  garantit le gain statique unitaire du système en boucle fermée (voir page 12). Si le régulateur a la forme générale RST alors les polynômes R S et T doivent vérifier :

$$H_{BF} = \frac{B \cdot T}{A \cdot S + B \cdot R} = \frac{P(1) B}{B(1) P} \quad (73)$$

S et R sont obtenus par la résolution de la même équation que précédemment (68). Le polynôme T est obtenu par simple identification :  $T = R(1)$ .

La structure, sous forme de schéma-bloc est donnée ci-dessous :

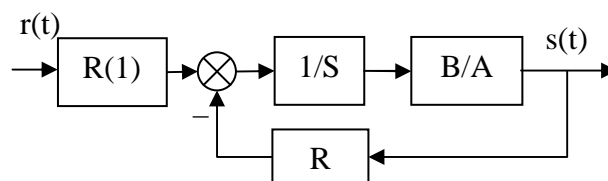


Figure 31 : PID2 sous forme RST

On montre que cette structure de PID, ramené en continu, correspond au PID canonique pour l'action intégrale. L'action proportionnelle et dérivée n'est appliquée que sur le signal  $y$ .

La structure du PID2 par rapport au PID1 a pour avantage de ne pas ramener de zéros à la boucle fermée. Cela se traduit par un dépassement moindre en transitoire lorsqu'on applique un échelon de commande. Par contre les deux systèmes réagissent de la même manière à un échelon de perturbation.

## 7.4 Placement de pôles

Les deux structures de PID décrites dans le chapitre précédent sont deux cas particulier d'une stratégie plus générale de placement de pôles.

La synthèse est plus générale que les PID présentés précédemment puisqu'il n'est pas nécessaire de faire d'hypothèses sur le degré de A et B, sur le retard, ni sur la présence ou non de zéros stables ou instables dans le procédé.

La structure du correcteur est la structure type RST telle que présentée dans la Figure 29. Pour la suite des calculs on posera pour la fonction de transfert les conventions suivantes :

$$H_{BO} = \frac{q^{-d} \cdot B(q^{-1})}{A(q^{-1})} \quad (74)$$

$$\begin{cases} A(q^{-1}) = 1 + a_1 \cdot q^{-1} + \dots + a_{n_A} \cdot q^{-n_A} \\ B(q^{-1}) = b_1 \cdot q^{-1} + \dots + b_{n_B} \cdot q^{-n_B} = q^{-1} \cdot B^*(q^{-1}) \end{cases}$$

Compte tenu de la forme RST du régulateur la fonction de transfert en boucle fermée vérifie :

$$H_{BF} = \frac{q^{-d} \cdot B(q^{-1}) \cdot T(q^{-1})}{P(q^{-1})} \quad (75)$$

$$P(q^{-1}) = A(q^{-1}) \cdot S(q^{-1}) + q^{-d} \cdot B(q^{-1}) \cdot R(q^{-1}) = 1 + p_1 \cdot q^{-1} + \dots + p_{n_P} \cdot q^{-n_P}$$

Le polynôme  $P(q^{-1})$  définit donc les pôles de la fonction de transfert en boucle fermée et traduit donc le comportement en *poursuite*.

De la même manière la fonction de transfert entre une perturbation et la sortie (fonction de sensibilité perturbation-sortie) est donnée par :

$$S_{yp} = \frac{A(q^{-1}) \cdot S(q^{-1})}{P(q^{-1})} \quad (76)$$

Ce même polynôme traduit donc aussi le comportement en *régulation*.

Les consignes en poursuite et en régulation peuvent être différentes comme cela est représenté par exemple sur la Figure 32.

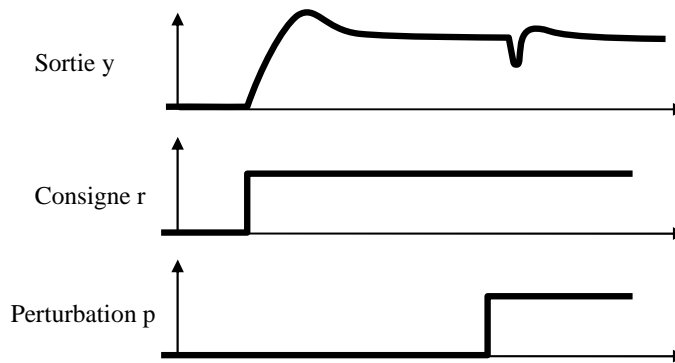


Figure 32 : Performances différentes en régulation et en poursuite

### 7.4.1 Choix des pôles en boucle fermée

De manière générale le comportement en boucle fermée est spécifié sous la forme d'une pulsation propre  $\omega_0$  et d'un amortissement  $\sigma$ , donc une fonction de transfert du second ordre. La spécification étant généralement donnée en continu il est nécessaire de traduire la valeur désirée de  $\omega_0$  et  $\sigma$  en un polynôme  $P_D(q^{-1})$  par discrétisation de la fonction de transfert continue (chapitre 4.5 page 29). Ce polynôme  $P_D$  traduit le comportement dominant du système bouclé.

Si l'on souhaite introduire un effet de filtrage supplémentaire dans certaines bandes de fréquence (réduction des bruits de mesure, adoucissement de la commande, robustesse) cela est décrit par un polynôme  $P_F$  comportant des pôles auxiliaires, plus rapides que les pôles de  $P_D$ .

Le polynôme caractéristique en boucle fermée est donc constitué de pôles dominants et de pôles auxiliaires :

$$P(q^{-1}) = P_D(q^{-1}) \cdot P_F(q^{-1}) \quad (77)$$

### 7.4.2 Régulation : calcul de R et S

Le polynôme  $P(q^{-1})$  étant fixé il s'agit de trouver R et S vérifiant :

$$P(q^{-1}) = A(q^{-1}) \cdot S(q^{-1}) + q^{-d} \cdot B(q^{-1}) \cdot R(q^{-1}) \quad (78)$$

Comme déjà vu au chapitre 7.3.1 on obtient une équation de Bézout qu'il s'agit de résoudre. (Voir aussi 9.1 Annexe 1 : Résolution de l'équation polynomiale de Bezout)

Il est parfois nécessaire de spécifier R et S avec des parties fixes  $H_R$  et  $H_S$ . Dans ce cas l'équation (78) est modifiée et on l'écrit sous la forme:



$$P(q^{-1}) = (A(q^{-1}) \cdot H_S(q^{-1})) \cdot S'(q^{-1}) + (q^{-d} \cdot B(q^{-1}) \cdot H_R(q^{-1})) \cdot R'(q^{-1}) \quad (79)$$

Les inconnues que l'on souhaite trouver sont  $S'$  et  $R'$ . Ceci est nécessaire dans les cas particuliers suivants :

- spécification d'erreur statique nulle,
- rejet d'une perturbation harmonique,
- blocage d'un signal,

Détaillons maintenant ces cas particuliers.

### Spécification d'une erreur statique nulle

On retrouve la nécessité d'avoir un terme intégrateur dans la boucle directe. La spécification se retrouve donc sur  $S$  :

$$S(q^{-1}) = (1 - q^{-1}) \cdot S'(q^{-1}) = H_S(q^{-1}) \cdot S'(q^{-1}) \quad (80)$$

### Rejet d'une perturbation harmonique

Si le système est perturbé par une perturbation harmonique à une fréquence fixe (par exemple 50Hz...) il est possible d'imposer à  $S$  d'avoir un module *nul* ou amorti à cette fréquence. Cela se traduit par la présence dans  $S$  d'une paire de zéros complexe à amortissement nul ou faible à cette fréquence :

$$S(q^{-1}) = (1 + \alpha \cdot q^{-1} + \beta \cdot q^{-2}) \cdot S'(q^{-1}) = H_S(q^{-1}) \cdot S'(q^{-1}) \quad (81)$$

### Blocage d'un signal

Pour certaines application on souhaite que le régulateur "laisse passer" certaines composantes du signal de sortie, c'est-à-dire qu'il ne réagisse pas à ces composantes. Cela se traduit par  $R=0$  à ces fréquences. On retrouve la même forme que précédemment :

$$R(q^{-1}) = (1 + \alpha \cdot q^{-1} + \beta \cdot q^{-2}) \cdot R'(q^{-1}) = H_R(q^{-1}) \cdot R'(q^{-1}) \quad (82)$$

### 7.4.3 Poursuite : calcul de T

Le schéma-bloc de l'asservissement est donné sur la Figure 33. Le terme  $T$  est décomposé en un terme  $T'$  et un terme  $B^*/A^*$ .

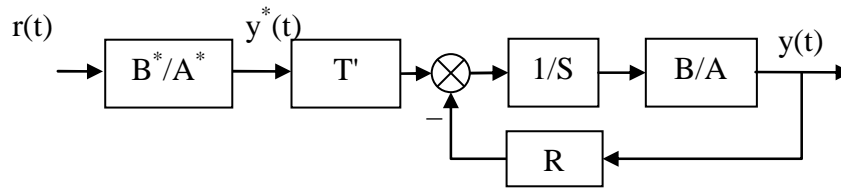


Figure 33 : placement de pôle

L'idée est de régler  $B^*$  et  $A^*$  de telle manière que le signal  $y^*$  ait la dynamique de poursuite désirée (par exemple un second ordre avec pulsation propre et amortissement donné) puis de régler  $T'$  de telle manière que le transfert entre  $y^*$  et  $y$  soit le plus "transparent" possible.

### Réglage de $T'$

Le transfert entre  $y^*$  et  $y$  est :

$$S_{yy^*}(q^{-1}) = T'(q^{-1}) \frac{q^{-d} \cdot B(q^{-1})}{P(q^{-1})} \quad (83)$$

Pour assurer le transfert le plus "transparent" possible il faut donc choisir

$$T'(q^{-1}) = \frac{P(q^{-1})}{B(1)} \quad (84)$$

Remarques :

- Si  $B(1)=0$  alors on remplace  $B(1)$  par 1 dans (84)
- De même dans (84)  $P(q^{-1})$  peut être remplacé par  $P_D$  correspondant à la dynamique dominante.

### Réglage de $B^*$ et $A^*$

$B^*$  et  $A^*$  sont choisis de telle manière que la fonction de transfert  $B^*/A^*$  corresponde à la dynamique de poursuite souhaitée.

## 7.5 Poursuite et régulation à objectifs indépendants

Cette méthode ressemble à la méthode de placement de pôles à ceci près que l'on va effectuer une simplification des zéros de  $B^*(q^{-1})$  : cela ne s'applique donc qu'aux systèmes avec des pôles stables. Plus précisément il faut qu'ils aient un amortissement suffisant ( $> 0,2$ ) de manière à ne pas être trop près du cercle unité.

La structure de l'asservissement est toujours celle donnée sur la Figure 33.

**Choix de P**

$P(q^{-1})$  est choisi de manière à simplifier  $B^*(q^{-1})$  (rappel : la forme de B est donnée dans l'équation 74). Comme indiqué dans le chapitre 7.4.2 page 60 il s'agit donc d'imposer les zéros de  $B(q^{-1})$  dans  $H_R(q^{-1})$  et  $H_S(q^{-1})$ .

**Choix de T**

Si l'on choisit  $T(q^{-1}) = P(q^{-1})$  alors le transfert entre  $y^*$  et  $y$  est :

$$S_{yy^*}(q^{-1}) = q^{-(d+1)} \quad (85)$$

**Réglage de  $B^*$  et  $A^*$** 

$B^*$  et  $A^*$  sont choisis de telle manière que la fonction de transfert  $B^*/A^*$  corresponde à la dynamique de poursuite souhaitée.



## 8. TECHNOLOGIE DE LA COMMANDE PAR ORDINATEUR

Du point de vue de l'automaticien le ordinateur est relié au système physique à piloter par un certain nombre d'intermédiaires plus ou moins gênants :

- les interfaces analogiques et numériques,
- le transport de l'information par bus et protocole,
- l'utilisation éventuelle de capteurs ou d'actionneurs « intelligents ».

Il ne s'agit pas ici de faire un exposé complet sur ces trois aspects mais de poser les problèmes qui se posent à l'automaticien lorsque le système qu'il doit commander est perturbé par ces éléments.

Enfin ce chapitre se termine sur les notions de « Hardware In the Loop » et de « prototypage rapide » qui sont partie intégrante des processus de réalisation de systèmes de contrôle.

### 8.1 Calculateurs et interfaces numériques / analogiques

Le premier problème à prendre en compte est la quantification de l'information. Du point de vue du ordinateur d'une part et du point de vue des convertisseurs d'autre part. Dans un premier temps, considérons les aspects performance de calcul lié à la nature du codage des nombres dans le ordinateur.

#### Les performances liées au types de variables

Prenons l'exemple où le concepteur de la loi de commande souhaite réaliser un correcteur de type proportionnel. Entre l'acquisition des mesures et l'écriture de la consigne il y aura la réalisation d'une de commande du type :

$$y = K \cdot (\text{consigne} - \text{mesure}) \quad (86)$$

En utilisant un langage de programmation évolué (par exemple le C) la ligne de code sera la transcription littérale de (86). Le programmeur devra décider de la définition et du type des variables, principalement du nombre d'octets utilisés pour stocker ses variables. Le Tableau 5 résume par exemple les types utilisés en langage C (norme ANSI)

Tableau 5 : Types ANSI C

Types entiers	char	8 bits
	short int	16 bits
	long int	32 bits
Types flottants (mantisse et exposant)	float	32 bits
	double	64 bits
	longdouble	128 bits

Une fois le code écrit en C l'exécution du programme sur un processeur cible ne peut se faire que par l'intermédiaire d'une compilation du programme. A chaque processeur est donc adapté une version de compilateur.

Les performances du programme (sa rapidité d'exécution) dépendront donc en partie de la qualité du compilateur. (Par exemple, une multiplication s'effectue t'elle bêtement par un décalage de tous les bits ou bien des solutions astucieuses sont-elles implémentées pour arrêter le décalage selon le contexte ?)

Le principal déterminant de la performance est cependant le processeur : son type (8 bits, 16 bits ou 64 bits) et son cadencement (en MHz ou GHz) sont les deux plus importants paramètres. Coder les variables sur 64 bits (double) dans le programme coûtera cher en temps de calcul si le processeur est un 8 bits. Le type de mémoire et leurs temps d'accès peuvent aussi être déterminants. Enfin, certains processeurs disposent d'un processeur spécifiquement dédié aux opérations mathématiques (coprocesseur mathématique).

Il n'est pas possible de dire exactement, pour tel type de compilateur associé à tel type de processeur, le temps que prendra par exemple une multiplication. En pratique on effectue une série de tests types (benchmark) et on affiche un temps moyen d'exécution.

En résumé, on peut être tenté, pour ne pas s'embêter, de coder toutes les variables en virgule flottante. Mais il faut bien savoir que cela a un coût en terme de temps de calcul, surtout si le processeur n'est pas adapté. Pour fixer les idées on peut dire qu'il y a un rapport 20 entre une addition réalisée en "short" et une addition réalisée en virgule flottante.

Enfin, l'utilisation de fonctions mathématiques (sinus, exponentielles, etc...) peut aussi avoir un coût très lourd si le travail n'est pas réalisé par un coprocesseur mathématique dédié. Le compilateur et sa librairie mathématique joue là encore un grand rôle.

### La quantification imposée par le processeur

Au-delà de l'aspect performance et temps de calcul, la quantification de l'information doit parfois aussi être prise en compte. Rappelons qu'un entier codé sur N bits peut prendre seulement  $2^N - 1$  valeurs différentes. Avant d'effectuer les calculs mathématiques dans le corps du programme il est donc judicieux d'appliquer un

facteur d'échelle de manière à utiliser l'amplitude maximale possible (entre 0 et  $2^N - 1$ ).

### **La quantification imposée par les conversions analogique et numérique**

Les convertisseurs analogique-numérique et numérique-analogique sont principalement caractérisés par leur amplitude de signal analogique d'entrée (généralement en volts, par exemple +/- 10V) et par leur résolution en bits (8 ou 10 bits pour les plus courants, jusqu'à 16 bits pour les convertisseurs haut de gamme).

Prenons l'exemple d'un capteur gyromètre transformant une vitesse de rotation en tension. Compte tenu de l'application le signal utile varie entre 2,4V et 2,6V. Si le signal mesuré est branché directement sur un convertisseur 0-10V 10bits alors un problème de conditionnement du signal se pose. En effet la quantification est de  $\frac{10V}{2^{10}-1} = 0,098V$ . Les 0,2V de signal utile seront donc discrétisés en  $\frac{0,2}{0,098} = 20$  intervalles seulement !

Si la discrétisation du signal utile est trop grossière cela peut se traduire par une diminution des performances de l'asservissement, voire une instabilité. Le système n'est en effet plus linéaire et il n'existe pas d'outils théoriques permettant de prévoir les performances du système bouclé.

## **8.2 Transport de l'information : les bus et les protocoles**

Pas question de faire ni même d'ébaucher un cours sur les bus et les protocoles ici. Mais on va quand même donner quelques indications. L'automaticien a l'habitude de simplifier le problème qu'il se pose en dessinant un schéma simplifié de son problème d'asservissement (du type de la Figure 20 page 45) pour pouvoir appliquer les théories de la commande. Cependant il arrive bien souvent que l'information transite à l'intérieur du système, soit entre différents ordinateurs effectuant différentes parties du travail soit entre ordinateur et périphériques, par l'intermédiaire de bus informatique. A chaque type de bus correspond un ou plusieurs protocoles qui peuvent poser des problèmes pour l'automaticien :

- les données sont-elles discrétisées ? (le protocole peut par exemple ne permettre de transmettre que des paquets de données de 16 bits),
- le bus induit-t'il un retard ? (le composant mettant en forme les données du ordinateur pour les publier sur le bus a certaines performances qui peuvent induire un délai)
- le bus est-il déterministe ? (Certains bus ne garantissent pas que les données arriveront en un temps donné),
- le bus est-il sûr ? (Certains bus ne garantissent pas non plus que les données arriveront bien)

### **8.3 Capteurs et actionneurs "intelligents", "smart sensors"**

De plus en plus souvent les capteurs et les actionneurs sont associés à leur électronique de puissance, électronique de commande et de conditionnement de signal. L'interface avec le capteur ou l'actionneur se fait donc via un protocole de communication (par exemple via un bus RS232).

L'intérêt est de véhiculer non pas des signaux analogiques, tensions ou courants, mais des signaux numériques beaucoup moins sensibles aux éventuelles perturbations.

On retrouve donc, entre le capteur ou l'actionneur et le ordinateur, la présence d'un bus et d'un protocole de communication.

### **8.4 La philosophie "hardware in the loop"**

Les systèmes embarqués sont de plus en plus complexes et touchent des domaines de plus en plus variés : automobile, satellites, aéronautique, UAV, systèmes d'arme, sous-marins, systèmes, etc... Les systèmes à piloter ainsi que les systèmes de commande sont de plus en plus complexes.

Les techniques de simulation « Hardware In the Loop » (HIL) sont de plus en plus utilisées dans le développement et dans le test de ces systèmes embarqués. L'idée est de développer entièrement le système de contrôle sur une plateforme de test en utilisant le maximum de composants réels et en ne remplaçant que le minimum par la simulation. Par exemple en aéronautique l'« Iron Bird » regroupe dans un hangar tous les systèmes réels de l'avion en cours de développement. Seule la mécanique du vol est simulée : les sorties des actionneurs sont mesurées, rentrées dans un modèle de simulation numérique, puis les mesures effectuées normalement par les capteurs sont calculées et restituées par le modèle.

L'intérêt est de diminuer de manière importante les coûts et les délais en économisant le plus grand nombre possible de tests sur le système réel en situation. Lors des premiers tests réels les techniques « Hardware In the Loop » permettent d'avoir effectué la plus grande part du développement des lois de commande.

### **8.5 Le prototypage rapide en automatique**

Un peu dans le même ordre d'idée que le « Hardware In the Loop » le prototypage rapide permet de réaliser la conception des lois de commande en simulation puis de transporter rapidement (en un simple click !) le code du contrôleur simulé vers la cible temps réel implémentée sur le système réel.

La société Mathworks [9] investit massivement dans ce domaine en ayant enrichi son logiciel de calcul scientifique (Matlab) avec un logiciel de simulation graphique très évolué (Simulink) puis en développant des outils de prototypage rapide associés (Real Time Workshop, Embedded Target For...).



## 9. ANNEXE

### 9.1 Annexe 1 : Résolution de l'équation polynomiale de Bezout

Il s'agit de trouver R et S vérifiant :

$$P(q^{-1}) = A(q^{-1}).S(q^{-1}) + B(q^{-1}).R(q^{-1})$$

Où P, A et B sont connus.

Dans le cas où P, A, B, R et S sont des nombres entiers la solution de l'algorithme est donnée par l'algorithme de division Euclidienne étendu [8].

Dans le cas où P, A, B, R et S sont des polynômes, l'algorithme est donné ci-dessous, programmé en matlab ([1]).

```
function [Rp,Sp,nrp,nsp]=bezoutd(A,B,Hs,Hr,P)
%function [Rp,Sp,nrp,nsp]=bezoutd(A,B,Hs,Hr,P)
%solves AHsSp+BHrRp=P by coefficient comparison. Delay and discretization delay need to be integrated in B
%(zeros at the beginning).
%
%inputs:
%A=[a0 a1 ... aNa] ... vector of model denominator coefficients A=a0 + a1z^(-1) + a2z^(-2) +...+ aNaz^(-Na)
%B=[b0 b1 ... bNb] ... vector of model numerator coefficients B=b0 + b1z^(-1) + b2z^(-2) +...+ bNbz^(-Nb)
%Hs=[hs0 hs1 ... hsNhs] ... vector of controller denominator fixed part Hs=hs0 + hs1z^(-1) +...+ hsNhsz^(-Nhs)
%Hr=[hr0 hr1 ... hrNhr] ... vector of controller denominator fixed part Hr=hr0 + hr1z^(-1) +...+ hrNhrz^(-Nhr)
%P=[p0 p1 ... pNp] ... vector of desired polynomial coefficients P=p0 + p1z^(-1) + p2z^(-2) +...+ pNpz^(-Np)
%outputs:
%Rp=[rp0 rp1 rp2 ...] ... vector of coefficients for resulted controller numerator
%Sp=[sp0 sp1 sp2 ...] ... vector of coefficients for resulted controller denominator
%nrp ... order of Rp
%nsp ... order of Sp
%
%written by: J. Langer, I.D. Landau, H. Prochazka
%7th june 2002

PRECISION=1e-16;

D=size(A);
if D(1)>1, A=A'; end;
```

```

D=size(B);
if D(1)>1, B=B'; end;
D=size(Hs);
if D(1)>1, Hs=Hs'; end;
if D(1)==0, Hs=1; end;
D=size(Hr);
if D(1)>1, Hr=Hr'; end;
if D(1)==0, Hr=1;end;
D=size(P);
if D(1)>1, P=P'; end;

na=length(A)-1;
nb=length(B)-1;
np=length(P)-1;
nhs=length(Hs)-1;
nhr=length(Hr)-1;

if (nhs>0), Ah=real(convz(A,Hs)); else Ah=A*Hs; end; % Ah = A * Hs
nah=length(Ah) -1;
if (nhr>0), Bh=real(convz(B,Hr)); else Bh=B*Hr; end;
nbh=length(Bh) -1;
if (np>nah+nbh-1), disp('Bezout error: too many poles');end;

% increase size of P using zeros if necessary
if (np<nah+nbh-1),
%
% set remaining poles onto a circle with radius rmin
P=[P zeros(1,nah+nbh-1-np)];
rootsPdes=roots(P);
nextra=nah+nbh-1-np;
rmin=1e-16;
angle=[0:nextra-1]'/nextra*2*pi;
j=sqrt(-1);
rootsPextra=rmin*exp(j*angle);
P=poly([rootsPdes;rootsPextra]);
np=nah+nbh-1;

end;
P,

nsp=nbh-1;
nrp=nah-1;

%matrix is smaller than vector PD

```

```

if (np>nah+nbh-1),
    disp('Order of model denominator is too low! Add a polynom of higher order to Hs or Hr. ');
end;

% ns=nsp+nhs
% nr=nrp+nhr

M=[];
for j=1:nsp+1,
    V=[];
    if (j>1), V=[V ; zeros(j-1,1)]; end;% zeros in front of Ah
    V=[V ; Ah'];% Ah
    if (j<=nsp), V=[V ; zeros(nsp+1-j,1)]; end;% zeros behind Ah
    if (length(V)~=nah+nbh), disp('bezoutb: error V'); end;
    M=[M V]; % add one column to M
end;

for j=1:nrp+1,
    V=[];
    if (j>1), V=[V ; zeros(j-1,1)]; end;
    V=[V ; Bh'];
    if (j<=nrp), V=[V ; zeros(nrp+1-j,1)]; end;
    if (length(V)~=nah+nbh), disp('bezoutb: error V'); end;
    M=[M V];
end;

D=size(M);
if (D(1)~=nah+nbh), disp('bezoutb: error size M row'); end;
if (D(2)~=nah+nbh), disp('bezoutb: error size M column'); end;

% make P column vector
P=P';

global M1;
M1=M;

X= M\P;
% coefficients are real values
X=real(X);

% make X row vector
X=X';

```

## Commande des systèmes par ordinateur

```
Sp=X(1:nsp+1);  
Rp=X(nsp+2:nsp+nrp+2);
```

## 10. BIBLIOGRAPHIE

- [1] I. D. Landau, "*Commande des systèmes*", Edition Lavoisier 2002.
- [2] C. Nouals, "Théorie du Signal", Notes de cours ENSICA
- [3] F. Vincent, "*Traitement Numérique du Signal*", Notes de cours ENSICA.
- [4] D. Arzelier, "*Représentation et analyse des systèmes, notes de cours*", Edition ENSICA. Voir aussi <http://www.laas.fr/~arzelier/>
- [5] J. Bordeneuve-Guibé, "*Commande des systèmes linéaires*", Notes de cours ENSICA.
- [6] Y. Briere, "*Identification*", Notes de cours ENSICA.
- [7] L. Ljung, "*System Identification, Theory for the User*", Prentice Hall 1987.
- [8] Wikipedia , équation de Bezout : [http://en.wikipedia.org/wiki/Bézout's\\_identity](http://en.wikipedia.org/wiki/Bézout's_identity)
- [9] <http://www.mathworks.com>



# 11. INDEX

<b>A</b>	
actionneur intelligent .....	21
amortissement .....	15
ANSI C .....	60
Approximation numérique d'équation différentielle .....	28
ARMAX .....	35
<b>B</b>	
BJ35 .....	
Blocage d'un signal .....	56
bloqueurs d'ordre 0 .....	22
BOZ .....	22
bus.....	21, 61
<b>C</b>	
CAN.....	21
capteur intelligent .....	21
causal .....	6, 9
Causalité .....	10
CNA.....	21
compilateur .....	60
convertisseur analogique / numérique.....	21
convertisseur numérique / analogique.....	21
convertisseurs analogique-numérique .....	61
coprocesseur mathématique .....	60
Correspondances pôles - zéros.....	31
<b>D</b>	
Diagramme de Bode .....	18
Discrétisation de fonction de transfert continues .....	27
division polynomiale.....	6
<b>E</b>	
échelon unitaire.....	11
équation d'état .....	17
Erreur statique.....	42
Estimation de la réponse fréquentielle .....	39
Estimation de la réponse impulsionnelle.....	38
Euler .....	29
<b>F</b>	
facteur d'échelle .....	61
facteur d'oubli .....	36
<i>fenêtre de Hamming</i> .....	39
filtre anti-repliement .....	25
fonction de sensibilité .....	42
fonction de transfert discrète.....	9
<b>G</b>	
gain statique.....	11
gain statique unitaire.....	42
<b>I</b>	
Identification.....	32
Identification non paramétrique .....	38
intercorrélation.....	38
<b>L</b>	
lieu des racines.....	43
lieu d'Evans.....	43
LTI.....	8
<b>M</b>	
marge de gain.....	47
marge de module.....	46
marge de phase .....	47
méthode des moindres carrés récursifs .....	32
Méthode des trapèzes.....	29
modèle ARX .....	32
<b>O</b>	
OE.....	35
one step ahead prediction.....	35
opérateur de retard .....	9
<b>P</b>	
période d'échantillonnage (choix) .....	26
Placement de pôles .....	54
Pôles .....	11
Poursuite .....	56
<i>prédiction</i> .....	32
protocoles.....	61
Pseudo Random Binary Signal .....	34
pulsation .....	15
<b>Q</b>	
quantification .....	23, 24, 60
<b>R</b>	
R,S,T.....	41
rapidité.....	15
rayon de convergence .....	5, 9
Régulation.....	55

Rejet de perturbations .....	42
Rejet d'une perturbation harmonique .....	56
réponse à un échelon.....	11
réponse fréquentielle.....	18
Réponse impulsionnelle.....	10
Réponses indicielle .....	10
retard.....	23
root locus .....	43

**S**

signal discret .....	5
smart sensors.....	62
stabilité.....	11
Systèmes du premier ordre.....	14
Systèmes du second ordre.....	15

**T**

Théorème de Nyquist.....	46
théorème de Shannon.....	19, 24
théorème du retard .....	6
Transformée en z .....	5
transformée unilatérale .....	5
Tustin.....	29

**V**

virgule flottante.....	60
------------------------	----

**Z**

zero holder hold .....	23
zéros.....	11
zoh .....	23



# Commande des systèmes par ordinateur

Yves Briere  
ENSICA

Notes de cours  
2007



# TABLE DES MATIERES

Table des matières .....	1
1. Introduction .....	1
2. Rappels et prérequis .....	3
3. Signaux et systèmes discrets .....	5
3.1 Signaux discrets .....	5
3.1.1 Transformée en z .....	5
3.1.2 Signaux types .....	7
3.2 Systèmes discrets .....	9
3.2.1 Fonction de transfert discrète, définition .....	9
3.2.2 Notation "opérateur de retard $q^{-1}$ " .....	10
3.2.3 Réponse impulsionnelle .....	11
3.2.4 Causalité .....	11
3.2.5 Réponses indicielle .....	11
3.2.6 Régime stationnaire (gain statique) .....	12
3.2.7 Pôles, zéros, stabilité .....	12
3.2.8 Systèmes du premier ordre .....	15
3.2.9 Systèmes du second ordre .....	16
3.2.10 Calcul itératif de la réponse d'un système .....	18
3.2.11 Représentation par équation d'état .....	18
3.3 Réponse fréquentielle des systèmes échantillonnés .....	19
3.3.1 Diagramme de Bode d'un système échantillonné .....	19
3.3.2 Où l'on voit apparaître le théorème de Shannon .....	20
4. Systèmes continus échantillonnés .....	23
4.1 Structure des systèmes commandés par ordinateur .....	23
4.2 Échantillonnage des systèmes continus .....	24
4.3 Effets de l'échantillonnage .....	25
4.4 Où l'on revoit le théorème de Shannon .....	26
4.5 Discrétisation de fonction de transfert continues .....	29
4.5.1 Approximation numérique d'équation différentielle .....	30
4.5.2 Superposition des réponses indicielles .....	32
4.5.3 Correspondances pôles - zéros .....	33
4.5.4 Bilan .....	33
5. Identification des systèmes échantillonnés .....	35
5.1 Identification d'un modèle ARX par la méthode des moindres carrés récurrents .....	35
5.2 L'identification paramétrique en général et en pratique .....	37

5.2.1	Choix du signal d'excitation, traitement des mesures .....	37
5.2.2	Choix d'une structure de modèle.....	38
5.2.3	Choix d'un critère à minimiser.....	38
5.2.4	Choix d'un algorithme.....	38
5.2.5	Validation.....	39
5.2.6	Utilisation de logiciels .....	39
5.3	Identification "simple" .....	39
5.4	Identification non paramétrique .....	41
5.4.1	Estimation de la réponse impulsionnelle .....	41
5.4.2	Estimation de la réponse fréquentielle.....	42
6.	Systèmes bouclés .....	45
6.1	Fonction de transfert en boucle fermée .....	45
6.2	Erreur statique .....	46
6.3	Rejet de perturbations.....	46
6.4	Stabilité des systèmes à simple gain de rétroaction.....	47
6.5	Analyse des systèmes bouclés dans le domaine fréquentiel : stabilité.....	49
6.5.1	Théorème de Nyquist dans le cas où le système est stable en boucle ouverte (critère du revers) .....	50
6.5.2	Théorème de Nyquist dans le cas où le système est instable en boucle ouverte .....	50
6.5.3	Interprétations du lieu de Nyquist.....	50
7.	Méthodes de régulation des systèmes échantillonnés .....	53
7.1	Introduction .....	53
7.2	Structure d'un système commandé par ordinateur, hypothèses.....	54
7.2.1	Discretisation, choix de la période d'échantillonnage.....	54
7.2.2	Discretisation d'un correcteur continu .....	55
7.3	Régulateur PID numérique .....	55
7.3.1	Cas 1 .....	55
7.3.2	Cas 2 .....	58
7.4	Placement de pôles .....	59
7.4.1	Choix des pôles en boucle fermée .....	60
7.4.2	Régulation : calcul de R et S.....	60
7.4.3	Poursuite : calcul de T .....	61
7.5	Poursuite et régulation à objectifs indépendants .....	62
8.	Technologie de la commande par ordinateur.....	65
8.1	Calculateurs et interfaces numériques / analogiques.....	65
8.2	Transport de l'information : les bus et les protocoles .....	67
8.3	Capteurs et actionneurs "intelligents", "smart sensors" .....	68
8.4	La philosophie "hardware in the loop" .....	68
8.5	Le prototypage rapide en automatique .....	68
9.	Annexe .....	69
9.1	Annexe 1 : Résolution de l'équation polynomiale de Bezout.....	69
10.	Bibliographie.....	73
11.	Index.....	1





# 1. INTRODUCTION



Figure 1 : commande d'un systèmes de télémédecine par ordinateur, vu par Claude Serre





## 2. RAPPELS ET PREREQUIS

Ce polycopié ne comporte pas de rappels sur les systèmes continus. S'adressant aux étudiants en deuxième année de l'ENSICA il suppose connu les enseignements de première année et de seconde année en automatique et traitement du signal.

En première année le cours de théorie du signal [2] a rendu familier les notions de :

- transformée de Fourier, transformée de Laplace
- corrélation, convolution

Le cours de traitement numérique du signal [3] a abordé en plus :

- la discrétisation et la quantification des signaux.
- la transformée en z et la transformée de Fourier discrète.
- les filtres numériques RIF et RII.

En automatique [4] les bases de la représentation et de l'analyse des systèmes continus ont été posées. On ne reviendra pas dans ce cours sur les notions de :

- modèles mathématique des systèmes linéaires à temps continus à temps invariant (LTI) : fonction de transfert, équation d'état, pôles, zéros et dynamique.
- commandabilité et observabilité.
- stabilité des systèmes en boucle ouverte : pôles, critères de Routh-Hurwitz.
- stabilité des systèmes bouclés : critère de Nyquist, critère de Bode.
- lieu des racines.

En deuxième année les notions suivantes sont acquises [5] :

- synthèse des correcteurs par méthodes fréquentielles (avance et retard de phase, PID)
- commande par retour d'état (placement de pôles), commande Linéaire Quadratique (LQG)
- reconstruction de l'état par Observateur, observateur optimal de Kalman.



## 3. SIGNAUX ET SYSTEMES DISCRETS

### 3.1 Signaux discrets

#### 3.1.1 Transformée en z

Pour une approche mathématiquement rigoureuse de représentation des signaux discrets par la transformée en z consulter [3] et sa bibliographie. Dans ce cours on se cantonne à une approche utilitariste des outils et méthodes mathématiques tout en essayant de ne pas contredire fondamentalement la rigueur mathématique.

Un signal discret  $s(k)$  est une séquence de nombres réels appelés échantillons. L'indice  $k$  est généralement associé à un instant d'échantillonnage :  $t = k.T_e$ . La période d'échantillonnage  $T_e$  est supposée constante. Par définition la transformée en z du signal  $s(k)$  est la fonction de la variable  $z$ , notée abusivement  $s(z)$  :

$$s(z) = Z(s(k)) = \sum_{k=0}^{\infty} s(k) \cdot z^{-k} \quad (1)$$

La transformée définie ici est plus précisément appelée transformée unilatérale puisque les termes correspondant à  $k < 0$  ne sont pas pris en compte. Ceci n'a pas d'importance puisque les signaux que nous utilisons sont systématiquement causaux<sup>1</sup> :  $s(k) = 0$  pour  $k < 0$ .

Il y a bien évidemment un lien avec la transformée de Laplace et la transformée de Fourier (voir [3])

La fonction de la variable complexe  $s(z)$  est une série et ne converge pas forcément pour toute valeur de  $z$ . En toute rigueur il faudrait associer le rayon de convergence à chaque fonction en  $z$  :  $\{s(z) \text{ converge pour } |z| > \alpha\}$ .

Les automaticiens pour qui la transformée en z n'est bien souvent qu'une façon bien commode de regrouper des termes de récurrence utilisent souvent la pseudo-variable<sup>2</sup>  $q$ .

A partir d'une transformée en z  $s(z)$  il est possible de retrouver la séquence  $s(k)$  correspondante en appliquant la transformation inverse, dont on ne donnera pas la formule ici. A la place de faire appel à cette formule compliquée il est possible d'utiliser d'autres méthodes pour retrouver la séquence  $s(k)$  lorsque la fonction  $s(z)$  est donnée sous forme de fraction rationnelle :

<sup>1</sup> Pour une justification du terme « causal » voir page 11

<sup>2</sup> Voir page 10

- division polynomiale
- décomposition de la fraction rationnelle en éléments simples puis utilisation de transformations élémentaires telles que celles données dans le Tableau 2

La propriété fondamentale de la transformée en z est qu'elle est linéaire (la transformée en z d'une composition linéaire de signaux est la composition linéaire de la transformée en z de chaque signal). Les autres propriétés fondamentales sont données dans le Tableau 2.

Tableau 1 : propriétés de la transformée en z

1	$s(z) = Z(s(k)) = \sum_{k=0}^{\infty} s(k) \cdot z^{-k}$
2	$Z(\alpha \cdot s_1(k) + \beta \cdot s_2(k)) = \alpha \cdot Z(s_1(k)) + \beta \cdot Z(s_2(k))$
3	$Z(k \cdot s(k)) = -z \cdot \frac{ds(z)}{dz}$
4	$Z(c^k \cdot s(k)) = s\left(\frac{z}{c}\right)$
5	$Z(s(k-1)) = s(k=-1) + z^{-1} \cdot s(z)$
6	$Z(s(k+1)) = z \cdot s(z) - z \cdot s(k=0)$
7	$s(k=0) = \lim_{z \rightarrow \infty} (s(z))$
8	$s(k \rightarrow \infty) = \lim_{z \rightarrow 1} \left( \frac{z-1}{z} \cdot s(z) \right)$

Remarque 1 : la propriété 5 ne prend pas en compte le fait que le signal est causal : en pratique  $s(k=-1) = 0$ .

Remarque 2 : la propriété 5 justifie le fait que le terme  $z^{-1}$  soit appelé "opérateur de retard unitaire", cette propriété est aussi appelée "théorème du retard".

Remarque 3 : les propriétés 7 et 8 sont données sous réserve de convergence.

Transformées inverses

### Exemple 1 : transformée inverse par division polynomiale

On considère un signal  $s(k)$  dont la transformée en z est donnée par la fraction rationnelle :

$$S(z) = \frac{0.1 \cdot z^2 + 0.1 \cdot z}{z^3 - 2.6 \cdot z^2 + 2.2 \cdot z - 0.6} \quad (3.1)$$

L'objectif est d'obtenir  $S(z)$  sous forme polynomiale :

$$S(z) = x_0 + x_1 \cdot z^{-1} + x_2 \cdot z^{-2} + \dots \text{etc.} \quad (3.2)$$

Cette suite est généralement infinie, la réponse temporelle est donnée par la succession des termes  $\{x_i\}$ . En traitant l'équation (3.1) comme une simple division l'algorithme de division classique (celui appris en CM2) permet d'obtenir par récurrence tous les termes de la suite (3.2) :

$0.1 \cdot z^2$	$+ 0.1 \cdot z$				$z^3 - 2.6 \cdot z^2 + 2.2 \cdot z - 0.6$ $0.1 \cdot z^{-1} + 0.36 \cdot z^{-2} + 0.716 \cdot z^{-3}$	
$0.1 \cdot z^2$	$- 0.26 \cdot z$	$+ 0.22$	$- 0.06 \cdot z^{-1}$			
$0$	$0.36 \cdot z$	$- 0.22$	$0.06 \cdot z^{-1}$			
	$0.36 \cdot z$	$- 0.936$	$0.792 \cdot z^{-1}$	$- 0.216 \cdot z^{-2}$		
$0$	$0.716$	$- 0.732 \cdot z^{-1}$	$\dots$	$\dots$		

Les éléments successifs de la suite  $s(k)$  sont donc  $s(k) = \{0, 0.1, 0.36, 0.716, \dots\}$ . Pour obtenir les termes suivants il faudrait continuer la division.

**Exemple 2 : transformée inverse par décomposition en éléments simples**

La deuxième méthode présentée consiste à décomposer la fraction rationnelle (3.1) en éléments simples puis d'utiliser les tables de transformées en  $z$  (Tableau 2). Pour notre exemple cela donne :

$$S(z) = \frac{0.1 \cdot z^2 + 0.1 \cdot z}{z^3 - 2.6 \cdot z^2 + 2.2 \cdot z - 0.6} = \frac{-z}{z-1} + \frac{0.5 \cdot z}{(z-1)^2} + \frac{z}{z-0.6}$$

La réponse temporelle est donc composée des trois éléments :

$$s(k) = -(1 \ 1 \ 1 \ \dots) + 0.5 \cdot (1 \ 2 \ 3 \ \dots) + (1 \ 0.6 \ 0.6^2 \ \dots)$$

$$\text{soit : } s(k) = \begin{cases} s(0) = -1 + 0.5 + 1 \\ s(2) = -1 + 0.5 \cdot 2 + 0.6 \cdot 0.5 \\ s(3) = -1 + 0.5 \cdot 3 + 0.6^2 \\ s(4) = \dots \end{cases}$$

**3.1.2 Signaux types**

Le Tableau 2 donne les transformée en  $z$  de certains signaux fondamentaux. La Figure 2 représente l'allure de quelques signaux types.

Tableau 2 : transformée en z élémentaires

$s(k)$	$S(z)$
signal temporel	Transformée en z
$\delta(k)$ : impulsion unité	1
$u(k)$ : échelon unité	$\frac{z}{z-1}$
$k \cdot u(k)$	$\frac{z}{(z-1)^2}$
$c^k \cdot u(k)$	$\frac{z}{z-c}$
$k \cdot a^k \cdot u(k)$	$\frac{a \cdot z}{(z-a)^2}$
$\sin(\omega \cdot k) \cdot u(k)$	$\frac{z \cdot \sin(\omega)}{z^2 - 2 \cdot z \cdot \sin(\omega) + 1}$
$\cos(\omega \cdot k) \cdot u(k)$	$\frac{z \cdot (z - \cos(\omega))}{z^2 - 2 \cdot z \cdot \sin(\omega) + 1}$
$a^k \cdot \sin(\omega \cdot k) \cdot u(k)$	$\frac{a \cdot z \cdot \sin(\omega)}{z^2 - 2 \cdot a \cdot \cos(\omega) \cdot z + a^2}$
$a^k \cdot \cos(\omega \cdot k) \cdot u(k)$	$\frac{z \cdot (z - a \cdot \cos(\omega))}{z^2 - 2 \cdot a \cdot \cos(\omega) \cdot z + a^2}$

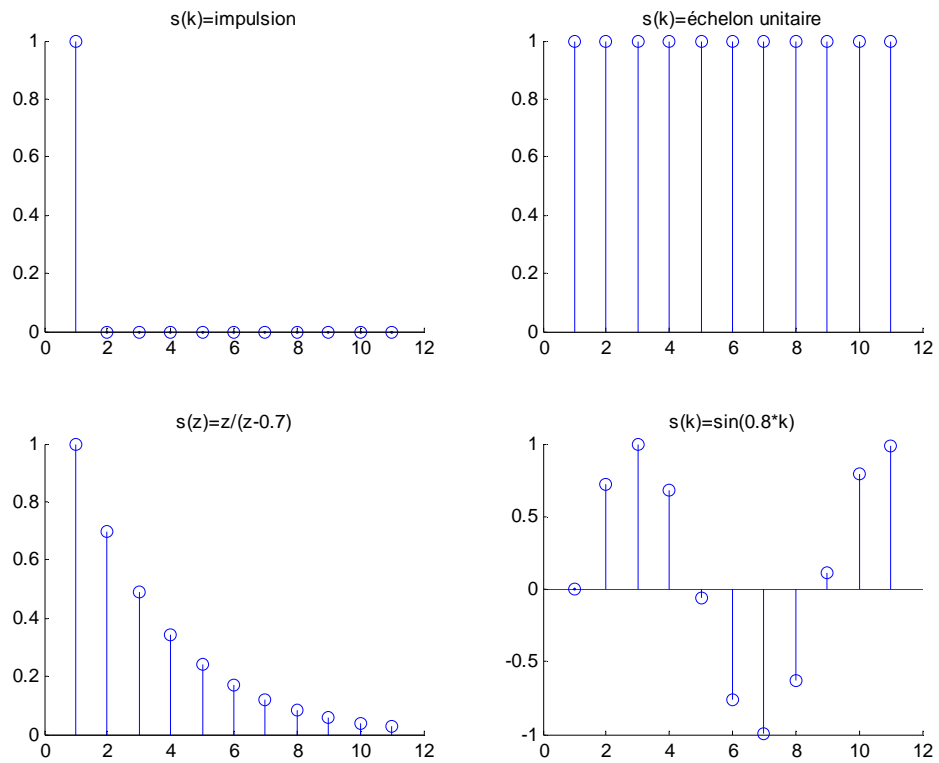


Figure 2 : réponses de quelques signaux types

## 3.2 Systèmes discrets

### 3.2.1 Fonction de transfert discrète, définition

On considère maintenant un système discret linéaire et invariant dans le temps (LTI) représenté sur la Figure 3.

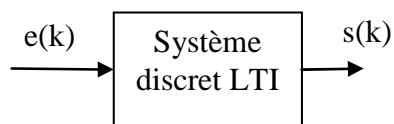


Figure 3 : système discret, notations

La relation entre la séquence d'entrée et la séquence de sortie est linéaire et définie par un ensemble de coefficients constants  $a_{1:n}$  et  $b_{1:m}$  :

$$\begin{aligned} s(k) + a_1 \cdot s(k-1) + a_2 \cdot s(k-2) + \dots + a_n \cdot s(k-n) = \\ b_0 \cdot e(k) + b_1 \cdot e(k-1) + b_2 \cdot e(k-2) + \dots + b_m \cdot e(k-m) \end{aligned} \quad (2)$$

On peut appliquer la transformée en  $z$  à chaque membre de l'équation puis en appliquant en récurrence la propriété 5 (théorème du retard) on obtient la fonction de transfert discrète du système :

$$\frac{s(z)}{e(z)} = F(z) = \frac{b_0 + b_1 \cdot z^{-1} + b_2 \cdot z^{-2} + \dots + b_m \cdot z^{-m}}{1 + a_1 \cdot z^{-1} + a_2 \cdot z^{-2} + \dots + a_n \cdot z^{-n}} \quad (3)$$

Remarque 1 : la fonction de transfert précédente est normalisée en prenant  $a_0 = 1$ .

Remarque 2 : on remarquera que la fonction de transfert est donnée en  $z^{-1}$  plutôt qu'en  $z$ . Cela est justifié par le fait que celle-ci traduit l'équation d'évolution du système physique causal (2).

Le script Matlab permettant de créer un objet Matlab correspondant à une fonction de transfert discrète est donné ci-dessous.

#### Matlab

```
% Création d'un système linéaire discret :
Te=0.1;
sysd = tf([b0 b1 b2],[1 a0 a2 a3],Te,'variable','z^-1');
```

On notera l'utilisation de l'argument 'variable', 'z^-1'.

### 3.2.2 Notation "opérateur de retard $q^{-1}$ "

La fonction de transfert est une fonction de la variable complexe  $z$ . Il faudrait donc en toute rigueur l'associer à son rayon de convergence. En pratique on ne s'embarrasse jamais de préciser cette zone de convergence. C'est pourquoi on trouve souvent la notation utilisant l'opérateur  $q^{-1}$  de décalage à la place de la variable complexe  $z^{-1}$  (4).  $F(q)$  n'est alors plus à proprement parler une fonction mais on lui appliquera les mêmes manipulations algébriques qu'à  $F(z)$ .

$$F(q) = \frac{b_0 + b_1 \cdot q^{-1} + b_2 \cdot q^{-2} + \dots + b_m \cdot q^{-m}}{1 + a_1 \cdot q^{-1} + a_2 \cdot q^{-2} + \dots + a_n \cdot q^{-n}} \quad (4)$$

Attention : selon les auteurs l'opérateur qui remplace  $z^{-1}$  est noté  $q$  (Matlab) ou  $q^{-1}$  (Landau [1]). Dans la suite du document c'est cette dernière convention qui sera appliquée.



L'utilisation de l'opérateur de retard  $q^{-1}$  à la place de la variable complexe  $z^{-1}$  permet de traiter du cas des fonctions de transfert à coefficients non constant, cas non prévu par la transformation en  $z$ .

### 3.2.3 Réponse impulsionnelle

Essayons d'appliquer au système un signal d'entrée particulier : une impulsion unitaire. La transformée en  $z$  du système est obtenue en multipliant la fonction de transfert par la transformée en  $z$  de l'impulsion unitaire : 1. On obtient :

$$s(z) = \frac{b_0 + b_1 \cdot z^{-1} + b_2 \cdot z^{-2} + \dots + b_m \cdot z^{-m}}{1 + a_1 \cdot z^{-1} + a_2 \cdot z^{-2} + \dots + a_n \cdot z^{-n}} \times 1 \quad (5)$$

Cela justifie le fait que la fonction de transfert du système peut aussi s'appeler "réponse impulsionnelle".

#### Matlab

```
% Réponse impulsionnelle d'un système :
[y,t] =impulse(sysd);
```

### 3.2.4 Causalité

Un système est dit causal si les sorties ne dépendent que du passé et non pas du futur. La conséquence est que la réponse impulsionnelle de  $F(z)$  ne peut pas comporter de terme non nuls pour  $k < 0$ . Cela justifie la confusion sémantique entre "système causal" et "signal causal", c'est-à-dire ne comportant pas de termes non nuls pour  $k < 0$ .

Attention : à la différence des systèmes continus un système physique (causal) discret peut très bien être modélisé avec un degré  $m$  supérieur au degré  $n$  : le système peut avoir plus de zéros que de pôles.

La transformée en  $z$  représentant des signaux échantillonnés elle peut tout aussi bien représenter les mesures prises à différents endroits d'une poutre. Le temps étant remplacé ici par une abscisse la notion de causalité ne s'applique pas ! Attention : beaucoup des propriétés de systèmes échantillonnés que l'on étudie sont liés à la causalité, prudence donc si l'on applique ces méthodes à d'autres domaines !

### 3.2.5 Réponses indicielle

Pour caractériser des systèmes on s'intéresse souvent à leur réponse impulsionnelle ou, plus souvent, à leur réponse indicielle. Pour calculer la réponse

à un échelon d'un système il faut donc multiplier la fonction de transfert par la transformée en z d'un échelon unitaire :

$$s(z) = \frac{b_0 + b_1 \cdot z^{-1} + b_2 \cdot z^{-2} + \dots + b_m \cdot z^{-m}}{1 + a_1 \cdot z^{-1} + a_2 \cdot z^{-2} + \dots + a_n \cdot z^{-n}} \times \frac{1}{1 - z^{-1}} \quad (6)$$

Il faudrait ensuite calculer la transformée en z inverse de s(z). En pratique, si l'on doit faire le calcul à la main on décompose en éléments simples puis on utilise des tables. Avec Matlab c'est plus simple. Voici par exemple le script pour obtenir les 20 premiers échantillons de la réponse indicielle d'un système :

**Matlab**

```
% Réponse indicielle d'un système :
[y,t] =step(sysd,20);
```

### 3.2.6 Régime stationnaire (gain statique)

Le gain statique est obtenu en appliquant une entrée indicielle et en calculant la valeur du régime permanent. En appliquant le théorème de la valeur finale il vient :

$$F_0 = \lim_{z \rightarrow 1} \left( \frac{z-1}{z} \cdot \frac{b_0 + b_1 \cdot z^{-1} + b_2 \cdot z^{-2} + \dots + b_m \cdot z^{-m}}{1 + a_1 \cdot z^{-1} + a_2 \cdot z^{-2} + \dots + a_n \cdot z^{-n}} \times \frac{1}{1 - z^{-1}} \right) \quad (7)$$

Soit :

$$F_0 = \frac{\sum_{i=0}^{i=m} b_i}{1 + \sum_{i=1}^{i=n} a_i} \quad (8)$$

Le gain statique est obtenu donc simplement en faisant le quotient de la somme des coefficients.

### 3.2.7 Pôles, zéros, stabilité

Comme pour les systèmes continus la dynamique et la stabilité des systèmes dépend des propriétés du dénominateur de la fonction de transfert.

$$F(z) = \frac{e_0 \cdot (1 + e_1 \cdot z^{-1}) \cdot (1 + e_2 \cdot z^{-1}) \cdots (1 + e_m \cdot z^{-1})}{(1 + c_1 \cdot z^{-1}) \cdot (1 + c_2 \cdot z^{-1}) \cdots (1 + c_n \cdot z^{-1})} \quad (9)$$

Les coefficients  $c_i$  sont les pôles de la fonction de transfert, les coefficients  $e_i$  sont les zéros. La fonction de transfert étant au départ composée de coefficients purement réels les coefficients  $c_i$  sont soit réels soit imaginaires par paires conjuguées.

Une fonction de transfert est donc aussi bien définie par la donnée de ses pôles et de ses zéros. Le script Matlab pour créer un système ainsi est donné ci-dessous :

#### Matlab

```
% Système discret défini par ses pôles et zéros :
Te=0.1;
sysd=zpk([c1 c2 c3],[e1 e2],e0,Te);
```

Si l'on décompose la fonction de transfert en éléments simples on obtient une somme d'éléments du type :

$$F(z) = \frac{d_1}{1 + c_1 \cdot z^{-1}} + \frac{d_2}{1 + c_2 \cdot z^{-1}} + \cdots + \frac{d_n}{1 + c_n \cdot z^{-1}} \quad (10)$$

Un système linéaire temps invariant est dit stable si sa réponse impulsionnelle tend asymptotiquement vers 0. Compte tenu de (10) cela nécessite que chaque pôle  $c_i$  soit de module inférieur à 1.

**Propriété** : un système discret LTI est stable si et seulement si tous ses pôles ont leur modules inférieur ou égal à 1.

#### Critère algébrique attestant de la stabilité : critère de Routh

On rappelle que le critère de Routh permet de prédire si les racines d'un polynôme sont toutes à partie réelle négative, donc stables. Énonçons le théorème de Routh dans le cas d'un polynôme  $P'(w)$  :

$$P'(w) = a_0 + a_1 \cdot w + a_2 \cdot w^2 + \cdots + a_n \cdot w^n \quad (11)$$

Dans un premier temps on construit le tableau :

$$\begin{array}{cccccc|c}
 \beta_{n,0} & \beta_{n,1} & \beta_{n,2} & \beta_{n,q-1} & \beta_{n,q} & 0 & \\
 \beta_{n-1,0} & \beta_{n-1,1} & \beta_{n-1,2} & \beta_{n-1,q-1} & \beta_{n-1,q-1} & 0 & \\
 \beta_{n-2,0} & \beta_{n-2,1} & \beta_{n-2,2} & \beta_{n-2,q-1} & \beta_{n-2,q-1} & & \\
 \\ 
 \beta_{2,0} & \beta_{2,1} & 0 & & & & \\
 \beta_{1,0} & 0 & & & & & \\
 \beta_{0,0} & 0 & & & & & 
 \end{array} \quad (12)$$

Puis on vérifie que tous les premiers éléments de la première colonne ont le même signe : cela garantit que tous les pôles de  $P'(w)$  ont leur partie réelle négative.

Les éléments du tableau sont construits par récurrence ligne par ligne en appliquant les formules suivantes :

Pour  $i = n$  et  $i = n - 1$  :

$$\begin{cases} \beta_{n,j} = a_{n-2j} \\ \beta_{n-1,j} = a_{n-1-2j} \end{cases} \quad (13)$$

Et pour  $i < n-1$  :

$$\beta_{i,j} = \frac{\beta_{i+1,0} \cdot \beta_{i+2,j+1} - \beta_{i+1,j+1} \cdot \beta_{i+2,0}}{\beta_{i+1,0}} \quad (14)$$

Dans le cas qui nous intéresse, un système échantillonné, on cherche plutôt à vérifier que les pôles du polynôme  $P(z)$  sont de module inférieur à un. L'idée est d'effectuer un changement de variable  $z \rightarrow w$  pour pouvoir appliquer le critère de Routh.

$$P'(w) = (1-w)^n \cdot P\left(\frac{1+w}{1-w}\right) \quad (15)$$

Si l'on effectue la transformation (15) alors le fait que  $P'(w)$  ait toutes ses racines à partie réelle négative garantit que  $P(z)$  a toutes ses racines dans le cercle unité. Pour vérifier la stabilité de  $P(z)$  il suffit donc d'effectuer le changement de variable (15) puis d'appliquer le critère de Routh conventionnel (12 13 14).

### Critère algébrique attestant de la stabilité : critère de Jury

Le critère de Jury s'apparente au critère de Routh. Nous le donnons pour un polynôme de degré 2, 3 et 4 du type :

$$P(z) = a_0 + a_1 \cdot z + a_2 \cdot z^2 + \dots + a_n \cdot z^n \quad (16)$$

Les conditions de stabilité sont données ci-dessous (ces conditions sont données dans le cas  $a_n > 0$ , il suffit de multiplier tous les termes par  $-1$  si cette condition n'est pas remplie par  $P(z)$ ) :

$$\begin{aligned}
 n = 2 & \begin{cases} a_0 + a_1 + a_2 > 0 \\ a_0 - a_1 + a_2 > 0 \\ -a_0 + a_2 > 0 \end{cases} \\
 n = 3 & \begin{cases} a_0 + a_1 + a_2 + a_3 > 0 \\ -a_0 + a_1 - a_2 + a_3 > 0 \\ -|a_0| + a_3 > 0 \\ a_0 a_2 - a_1 a_3 - a_0^2 + a_3^2 > 0 \end{cases} \\
 n = 4 & \begin{cases} a_0 + a_1 + a_2 + a_3 + a_4 > 0 \\ a_0 - a_1 + a_2 - a_3 + a_4 > 0 \\ a_4^2 - a_0^2 - |a_0 a_3 - a_1 a_4| > 0 \\ (a_0 - a_4)^2 (a_0 - a_2 + a_4) + (a_1 - a_3)(a_0 a_3 - a_1 a_4) > 0 \end{cases}
 \end{aligned} \quad (17)$$

### 3.2.8 Systèmes du premier ordre

On donne dans la Figure 4 les réponses indicielles de systèmes du premier ordre ainsi que la localisation des pôles dans le plan complexe. Le cercle de rayon unité représente la zone dans laquelle les pôles doivent se situer pour assurer la stabilité. On remarque sur les deux premières figures l'influence de la place du zéro par rapport au point  $(1,0)$  pour la rapidité.

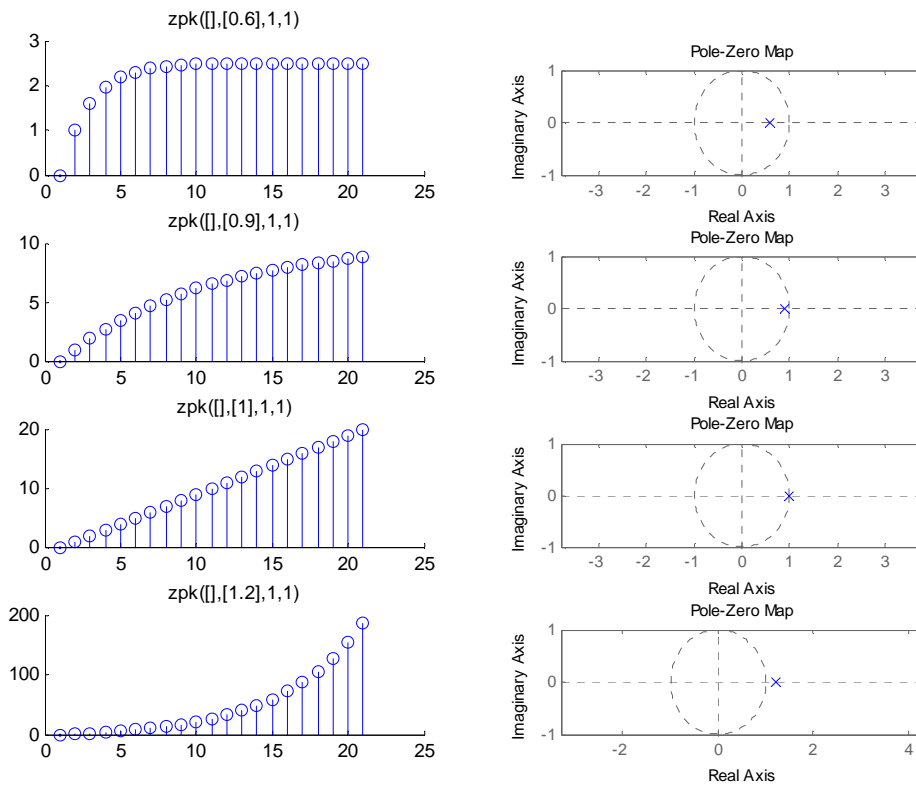


Figure 4 réponses indicielles de systèmes du premier ordre

### 3.2.9 Systèmes du second ordre

On donne dans la Figure 6 les réponses indicielles de systèmes du second ordre ainsi que la localisation des pôles dans le plan complexe. La rapidité et l'amortissement dépend de la position des pôles dans le cercle unité. La Figure 5 représente les courbes iso-amortissement et iso-pulsation des systèmes du second ordre.

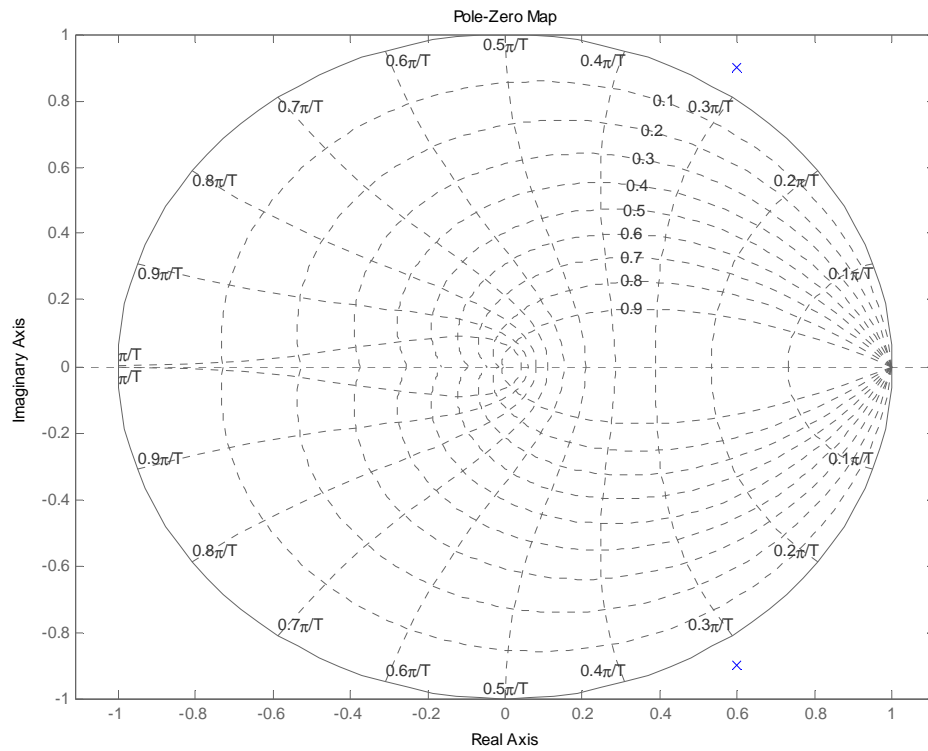


Figure 5 : courbes iso-amortissement et iso-pulsation des systèmes discrets du second ordre

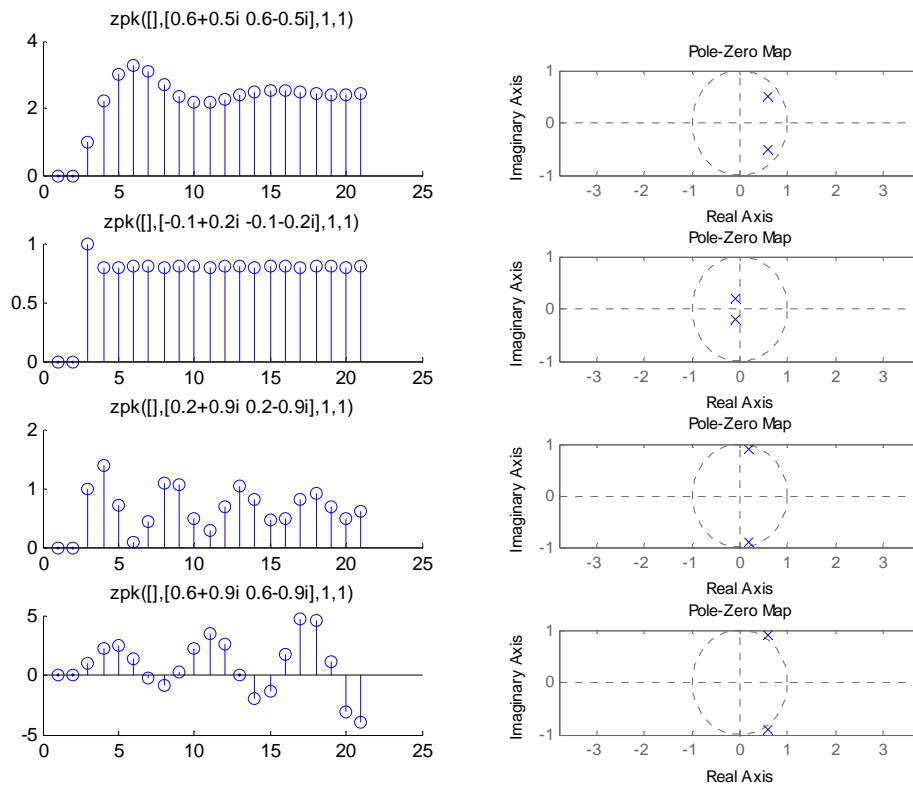


Figure 6 : réponses de systèmes du second ordre

### 3.2.10 Calcul itératif de la réponse d'un système

Si l'on dispose de la fonction de transfert d'un système et que l'on connaît les valeurs successives  $e(k)$  du signal d'entrée il est facile de calculer itérativement la sortie  $y(k)$  du système par l'équation de récurrence (2). Il faut initialiser par les  $n$  valeurs initiales de  $s(k)$ , prises généralement égales à 0.

### 3.2.11 Représentation par équation d'état

Un système discret peut aussi être représenté par un ensemble d'équations d'état comme dans le domaine continu (52). Cette représentation est naturellement adaptée aux systèmes multi-entrées multi-sorties. On définit alors un état interne  $x(k)$  ainsi qu'une fonction d'évolution  $f$  et une équation de mesure  $h$ .

$$\begin{cases} x(k+1) = f(x(k), u(k)) \\ s(k) = h(x(k), u(k)) \end{cases} \quad (18)$$



### 3.3 Réponse fréquentielle des systèmes échantillonnés

#### 3.3.1 Diagramme de Bode d'un système échantillonné

Lorsqu'on étudie la réponse fréquentielle des systèmes continus on considère des entrées particulières sinusoïdales :

$$\begin{cases} e(t) = \cos(2 \cdot \pi \cdot f \cdot t) + j \cdot \sin(2 \cdot \pi \cdot f \cdot t) = e^{j \cdot 2 \cdot \pi \cdot f \cdot t} \\ f \in [0 \quad \infty] \end{cases} \quad (19)$$

De même, pour des systèmes échantillonnés on considèrera des entrées sinusoïdales échantillonnées :

$$\begin{cases} e(k) = \cos(2 \cdot \pi \cdot f \cdot k) + j \cdot \sin(2 \cdot \pi \cdot f \cdot k) = e^{j \cdot 2 \cdot \pi \cdot f \cdot k} \\ f \in [0 \quad 1] \end{cases} \quad (20)$$

La sortie est aussi un signal échantillonné sinusoïdal de gain  $A(f)$  et de phase  $\Phi(f)$  variables et dépendant de  $f$ ; la fréquence réduite.

$$\begin{cases} s(k) = A(f) \cdot e^{j \cdot (2 \cdot \pi \cdot f \cdot k + \Phi(f))} \\ f \in [0 \quad 1] \end{cases} \quad (21)$$

Le nombre complexe, noté abusivement  $F(f)$ , d'amplitude  $A(f)$  et de phase  $\Phi(f)$  est obtenu à partir de  $F(z)$  en remplaçant  $z$  par  $e^{j \cdot 2 \cdot \pi \cdot f}$  :

$$\begin{cases} F(f) = F(z = e^{j \cdot 2 \cdot \pi \cdot f}) \\ f \in [0 \quad 1] \end{cases} \quad (22)$$

Le diagramme représentant le gain et la phase en fonction de la fréquence réduite s'appelle aussi un diagramme de Bode. L'échelle des fréquences est généralement linéaire de 0 à 0.5 (puisque le diagramme est nécessairement symétrique). La Figure 7 représente par exemple le diagramme de Bode d'un système du second ordre.

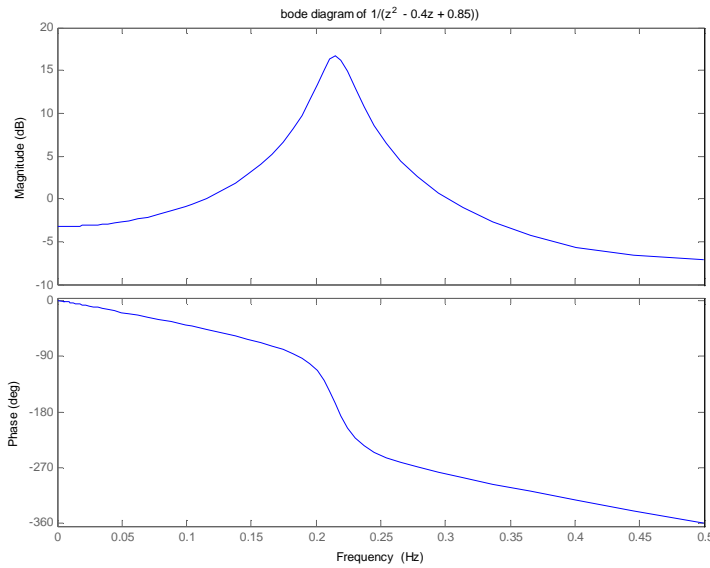


Figure 7 : exemple du diagramme de Bode d'un système discret du second ordre

### 3.3.2 Où l'on voit apparaître le théorème de Shannon

Le changement de variable  $z = e^{j2\pi f}$  n'est pas bijectif puisque  $e^{j2\pi f} = e^{j2\pi(f+k)}$  avec  $k$  entier. Si l'on fait le lien entre domaine continu et domaine fréquentiel on peut remarquer que le changement de variable  $z = e^{j2\pi f}$  fait correspondre une zone du plan complexe continu à une zone du plan complexe discret. La transformation est bijective si l'on se contente de la zone du plan complexe continu contenu entre les droites d'ordonnée 0 et 1.

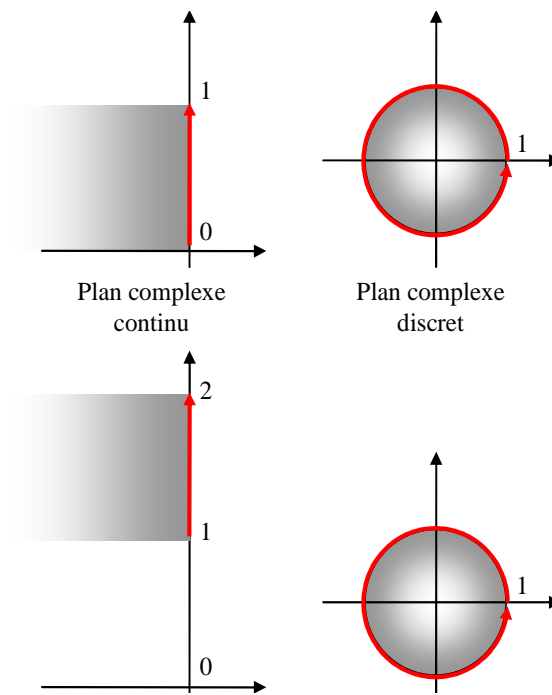


Figure 8 : correspondance entre plan complexe continu et plan complexe discret.



## 4. SYSTEMES CONTINUS ECHANTILLONNES

### 4.1 Structure des systèmes commandés par ordinateur

Les systèmes commandés par ordinateurs sont presque toujours des systèmes physiques continus par nature. Le ordinateur traite par contre les entrées et les sorties de manière discrète avec une période d'échantillonnage  $T_e$ . Les éléments importants, du point de vue de l'automaticien sont donc :

- le convertisseur analogique / numérique CAN ou du moins le système qui fournit séquentiellement les mesures au ordinateur à la période  $T_e$ .
- le ordinateur lui-même. Il doit effectuer l'acquisition des mesures, le traitement des informations, l'envoi des consignes dans un temps inférieur à  $T_e$ .
- le convertisseur numérique / analogique CNA.
- le système physique.

On suppose, pour faire simple et pour pouvoir utiliser la panoplie des outils mathématiques, que les systèmes sont échantillonnés à la même période  $T_e$  et au même moment.

La Figure 9 montre le système physique commandé par le ordinateur, dans sa version simpliste.

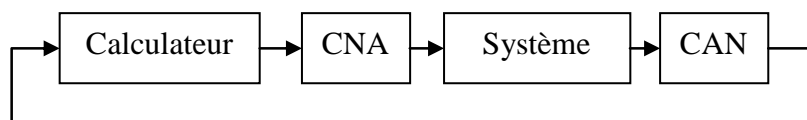


Figure 9 : système commandé par ordinateur

Au XXI<sup>ème</sup> siècle les choses sont quand même parfois un petit peu plus compliquées :

- le ordinateur fournit bien souvent un signal de commande numérique (transporté par bus informatique USB, RS232, CAN, Ethernet, etc...) directement interprété par l'actionneur. On parle "d'actionneur intelligent".
- de même le "capteur intelligent" fournit aussi bien souvent la mesure sous forme d'un signal numérique.
- Les signaux sont transportés par des bus informatiques qui n'ont pas que ça à faire : il peut y avoir du retard.
- de même les capteurs et actionneurs sont rarement tous seuls, ce qui se traduit par des périodes d'échantillonnage différentes, et parfois non synchrones...

- enfin le ordinateur peut aussi avoir d'autres choses à faire...

Ceci étant dit nous allons pour le moment nous restreindre au cas représenté sur la Figure 9 avec des instants d'échantillonnage synchrones et régulièrement espacés d'un temps  $T_e$ .

## 4.2 Echantillonnage des systèmes continus

On dispose généralement de la fonction de transfert continue (notée en utilisant la variable complexe  $s$  ou  $p$ )  $F(s)$  du système continu. Le problème est d'obtenir la fonction de transfert discrète du système échantillonné de manière à pouvoir synthétiser un correcteur. Le système continu est associé au convertisseur numérique – analogique qu'il faut prendre en compte.

Les convertisseurs numérique analogiques peuvent généralement se représenter par des "échantillonneurs – bloqueurs d'ordre 0" ou BOZ. Cela signifie que la commande est maintenue à une valeur constante pendant toute la durée du temps d'échantillonnage.

Si l'on applique une commande discrète impulsionnelle à un BOZ il va maintenir la commande 1 pendant la durée  $T_e$  puis revenir à zéro (Figure 10).

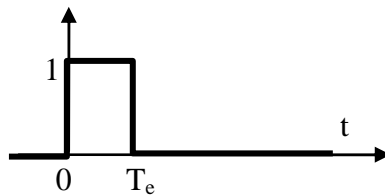


Figure 10 : réponse impulsionnelle d'un bloqueur d'ordre 0

La réponse impulsionnelle  $BOZ(t)$  est donc composée d'un échelon unitaire suivi d'un échelon unitaire de signe opposé et décalé d'un temps  $T_e$ . La transformée de Laplace de la réponse impulsionnelle du bloqueur est donnée par la formule (23) :

$$BOZ(s) = \frac{1}{s} - \frac{e^{-jT_e}}{s} \quad (23)$$

Du point de vue du ordinateur il faut donc prendre en compte le système continu de fonction de transfert  $F_{BOZ}(s)$  que l'on devra échantillonner :

$$F_{BOZ}(s) = \frac{1 - e^{-jT_e}}{s} \cdot F(s) \quad (24)$$

$$F_{\text{BOZ}}(z) = Z^{-1} \left( \frac{1 - e^{-jT_e}}{s} \cdot F(s) \right) \quad (25)$$

En pratique l'obtention de la transformée en z d'un système échantillonné par bloqueur d'ordre 0 se fait à l'aide de tables. On donne quelques exemples fondamentaux dans le Tableau 3 ci-dessous.

Tableau 3 : transformée en z de systèmes continus échantillonnés

F(s)	F <sub>BOZ</sub> (z)
1	1
$\frac{1}{s}$	$\frac{T_e \cdot z^{-1}}{1 - z^{-1}}$
$\frac{1}{1 + T \cdot s}$	$\begin{cases} \frac{b_1 \cdot z^{-1}}{1 + a_1 \cdot z^{-1}} \\ b_1 = 1 - e^{-T_e/T} \quad a_1 = -e^{-T_e/T} \end{cases}$
$\frac{e^{-Ls}}{1 + T \cdot s}$	$\begin{cases} \frac{b_1 \cdot z^{-1} + b_2 \cdot z^{-2}}{1 + a_1 \cdot z^{-1}} \quad a_1 = -e^{-T_e/T} \\ b_1 = 1 - e^{-(L-T_e)/T} \quad b_2 = e^{-T_e/T} (e^{L/T} - 1) \end{cases}$
$L < T_e$	

Si l'on dispose de Matlab c'est bien sûr encore plus simple. On utilise la fonction c2d avec l'argument 'zoh' pour "zero holder hold" (bloqueur d'ordre zéro).

#### Matlab

```
% Définition d'un système continu :
sysc=tf(1,[1 1]);
% Obtention de la transformée en z du système échantillonné à Te
Te=0.1;
sysd=c2d(sysc,Te,'zoh');
% Notation en z^-1
sysd.variable='z'
```

### 4.3 Effets de l'échantillonnage

Le convertisseur numérique a deux effets (néfastes) importants : l'effet de retard et l'effet de quantification.

L'effet de retard se voit dans (24) : on peut dire en première approximation que l'échantillonnage se traduit par un retard pur de  $T_e/2$ . Cela a donc, si ce temps

d'échantillonnage est non négligeable par rapport aux constantes de temps du système, un effet déstabilisant.

Pour illustrer l'effet de quantification supposons par exemple que l'on utilise un convertisseur analogique/numérique 8 bits 0-10V. Le plus petit incrément que le convertisseur peut distinguer est de  $10V/2^8 = 40mV$ . Si le signal utile varie physiquement entre 5V et 5,4V on n'aura que dix incréments utilisables... Ce phénomène n'entre pas dans la panoplie des phénomènes modélisables par la transformée en z. En pratique il est donc souvent nécessaire d'étudier le système et de synthétiser la commande numérique en négligeant ce phénomène puis de vérifier a posteriori qu'il n'a pas un effet trop néfaste.

#### 4.4 Où l'on revoit le théorème de Shannon

Le théorème de Shannon est bien connu en ce qui concerne l'échantillonnage de signaux discrets : on sait que le signal ne doit pas comporter de composante de fréquence supérieure à  $f_e/2$  si l'on ne veut pas observer de repliement. On doit aussi prendre des précautions si l'on veut travailler sur la fonction de transfert discrète correspondant à un système continu échantillonné.

On va traiter un exemple concret pour mettre en évidence le problème de repliement dans le cas de la commande des systèmes continus. On s'intéresse à deux systèmes continus du second ordre ayant des pôles de même partie réelle mais de parties imaginaires séparées de  $2\pi T_e$ . Le script Matlab ci-dessous montre la création des deux systèmes :

##### Matlab

```
% Définition de deux systèmes continus :
sys1=zpk([],[-1-2i -1+2i],2^2+1);
sys2=zpk([],[-1-(2+2*pi)*i -1+(2+2*pi)*i],1+(2+2*pi)^2);
```

On voit sur la Figure 11 que la réponse indicielle des deux systèmes est bien différente.



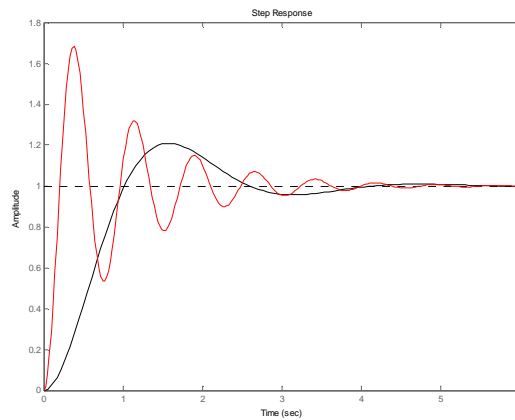


Figure 11 : réponse indicielle de deux systèmes du second ordre

Si l'on calcule la fonction de transfert échantillonnée les deux systèmes à la période de 1 seconde on obtient deux réponses indicielles très proches (représentés par des  $\circ$  sur la Figure 12). Conclusion : la période d'échantillonnage était mal choisie

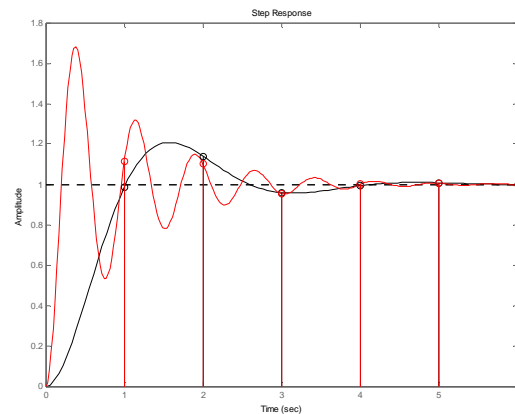


Figure 12 : réponse indicielle de deux systèmes du second ordre échantillonnés à 1s

En pratique la commande échantillonnée de systèmes continus doit être associée à certaines précautions :

- le système ne doit pas comporter de dynamiques propres situées dans les fréquences supérieures à  $f_c/2$  (exemple illustré dans ce chapitre).
- les signaux échantillonnés doivent être "nettoyés" des fréquences supérieures à  $f_c/2$  avec un filtre anti-repliement. (A moins que le signal soit suffisamment propre et ne comporte pas de composantes significatives au-delà de ces fréquences).

### Choix de la période d'échantillonnage

Le choix de la période d'échantillonnage est très important et dépend de la dynamique des signaux. Attention, les systèmes commandés en boucle fermée ont souvent une dynamique plus rapide que le système en boucle ouverte : il faut donc prendre en compte la dynamique en boucle fermée pour le choix de la fréquence d'échantillonnage.

En pratique on préconise de prendre une fréquence 5 à 25 fois supérieure à la bande passante souhaitée en boucle fermée.

On peut être tenté de prendre une fréquence d'échantillonnage très élevée si la technologie le permet (ordinateur et convertisseurs suffisamment rapide). Ceci peut amener rapidement à des problèmes numériques. Imaginons par exemple le cas d'un système ayant une dynamique très lente (quelques secondes) et un système de mesure codant les mesures sur 10 bits (cas classique). La Figure 13 illustre cet exemple. Le signal varie peu entre deux instants d'échantillonnage et les variations du signal sont faibles par rapport à la valeur de l'incrément numérique (figure du haut). Le calcul de la simple dérivée numérique se traduit donc par une succession de  $-1$   $0$  et  $+1$ .

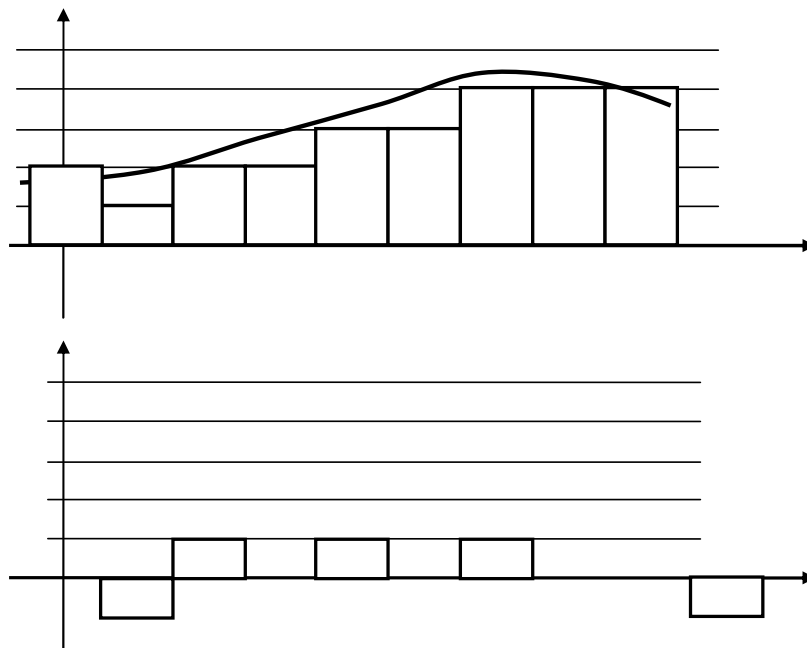


Figure 13 : période d'échantillonnage trop faible. En haut : signal échantillonné et discrétisé ; en bas : calcul de la dérivée numérique.

### Filtre anti-repliement

En toute rigueur chaque fois qu'un signal est échantillonné il faudrait faire précéder le convertisseur numérique d'un filtre analogique anti-repliement éliminant physiquement toutes les composantes du signal de fréquence supérieure à  $f_e/2$ .

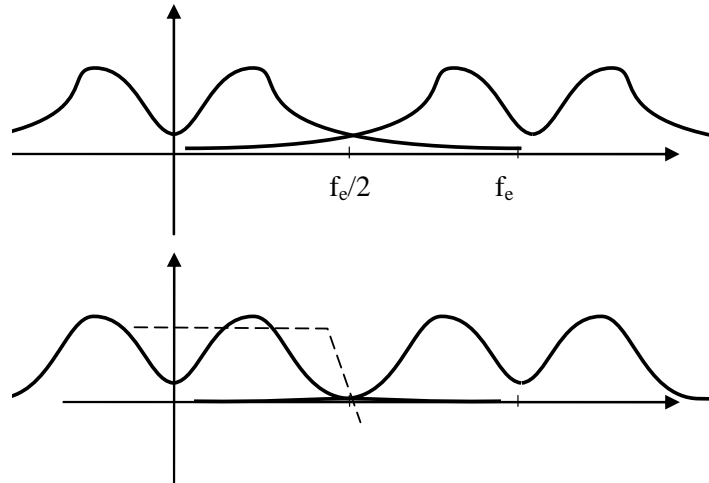


Figure 14 : Effet du filtre anti-repliement

Bien souvent la période d'échantillonnage de l'asservissement est relativement lente (quelques secondes) et il est alors impossible de réaliser un filtre de constante de temps aussi faible. On procède alors à un sur-échantillonnage : le signal est filtré, échantillonné à haute fréquence  $T'_e$  puis filtré numériquement et sous-échantillonné à la période  $T_e$  beaucoup plus grande que  $T'_e$  comme illustré sur la

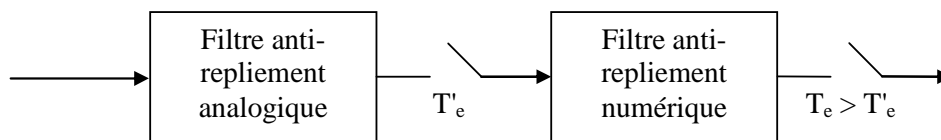


Figure 15 : Sur échantillonnage

Un simple filtre numérique de type moyenne mobile est souvent suffisant.

### 4.5 Discrétisation de fonction de transfert continues

Une manière classique pour réaliser un contrôle par ordinateur est de commencer par synthétiser un correcteur continu avec les outils de l'automatique classique puis de trouver un correcteur discret qui "approche" convenablement le correcteur continu.

#### 4.5.1 Approximation numérique d'équation différentielle

Supposons que le correcteur continu que l'on a synthétisé soit du type PD :

$$\text{PID}(s) = \frac{e(s)}{\varepsilon(s)} = k_p + k_D \cdot s \quad (26)$$

Il s'agit de calculer les commandes  $e(k)$  successives issues du correcteur en fonction des erreurs  $\varepsilon(k)$ . Une approche classique et intuitive est d'approximer la dérivée par une différence :

$$e(k) = k_p \cdot \varepsilon(k) + k_D \cdot \frac{\varepsilon(k) - \varepsilon(k-1)}{T_e} \quad (27)$$

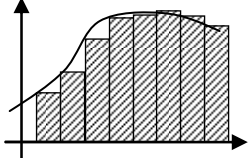
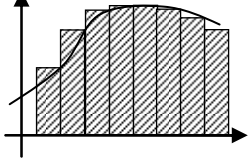
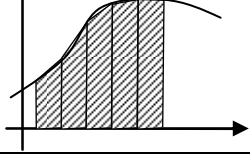
Si on calcule la transformée en  $z$  de  $e(k)$  on obtient :

$$e(z) = k_p \cdot \varepsilon(z) + k_D \cdot \frac{\varepsilon(z) - z^{-1}\varepsilon(z)}{T_e} = \left( k_p + k_D \cdot \frac{1 - z^{-1}}{T_e} \right) \cdot \varepsilon(z) \quad (28)$$

On retrouve la fonction de transfert PD(s) dans laquelle on a effectué le changement de variable  $s = \frac{1-z^{-1}}{T_e}$

De manière plus générale, l'approximation de fonction de transfert revient à effectuer l'approximation numérique d'intégration. Selon la méthode d'intégration choisie on effectue des changements de variable différents.

Tableau 4 : méthodes d'intégration

Méthode	Changement de variable	
Euler progressif (Euler's forward method)	$s = \frac{1 - z^{-1}}{z^{-1} \cdot T_e}$	
Euler rétrograde (Euler's backward method)	$s = \frac{1 - z^{-1}}{T_e}$	
Méthode des trapèzes, ou bilinéaire, ou de Tustin	$s = \frac{2}{T_e} \cdot \frac{1 - z^{-1}}{1 + z^{-1}}$	

En pratique le changement de variable  $s = \frac{1 - z^{-1}}{z^{-1} \cdot T_e}$  n'est pas sans risque. On donne sur la Figure 16 la correspondance entre la zone stable du plan complexe en  $s$  et la zone correspondante après la transformation (Euler progressif, Euler rétrograde, bilinéaire). On voit notamment qu'il est possible, partant d'un système continu stable, d'obtenir des pôles dans le domaine discret en dehors du cercle de stabilité si la transformation appliquée est celle d'Euler progressif.

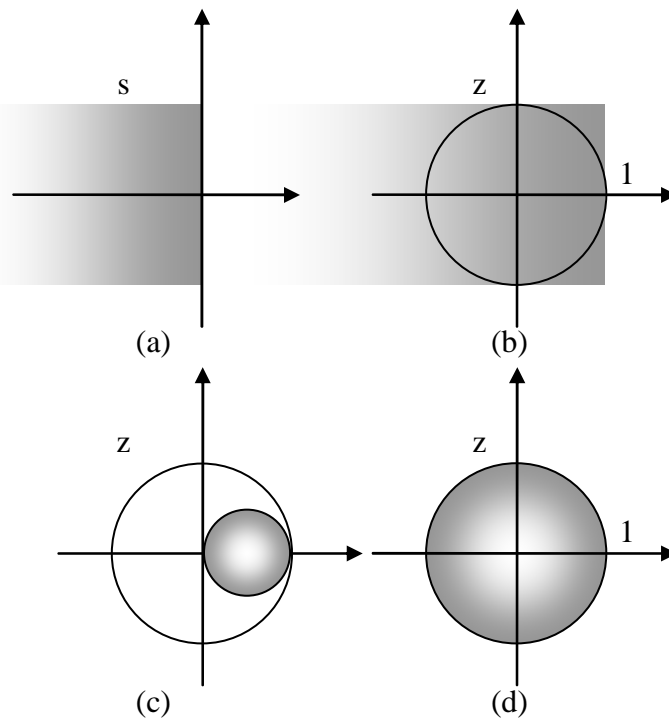


Figure 16 : correspondance des plans complexes. (a) : zone de stabilité dans le plan complexe continu en  $z$  ; (b) zone correspondante avec l'approximation d'Euler progressif ; (c) zone correspondante avec l'approximation d'Euler rétrograde ; (d) zone correspondante avec l'approximation bilinéaire.

La fonction `c2d` de Matlab permet d'obtenir l'équivalent discret d'une fonction de transfert continue avec la méthode d'intégration comme argument :

```

Matlab
sysc=tf([1 0],[1 2 5]);
Te=0.1;
% Méthode d'euler :
sysd1=c2d(sysc,Te,'zoh');
% Méthode des trapèzes :
sysd2=c2d(sysc,Te,'foh');
% Méthode bilinéaire :
sysd3=c2d(sysc,Te,'tustin');
    
```

#### 4.5.2 Superposition des réponses indicielles

Une autre méthode pour approximer une fonction de transfert discrète est de s'assurer que la réponse indicielle de sa version discrétisée correspond exactement lors des temps d'échantillonnage.

### 4.5.3 Correspondances pôles - zéros

La dernière méthode est de partir d'une décomposition de la fonction de transfert continue en pôles et zéros et d'effectuer la conversion pôles à pôles et zéros par zéros. La correspondance, selon que le pôle est réel ou complexe est donnée ci-dessous :

$$\begin{aligned} (s + a) &\rightarrow (z - e^{-a \cdot T_e}) \\ (s + a)^2 + b^2 &\rightarrow z^2 - 2 \cdot e^{-a \cdot T_e} \cdot \cos(b \cdot T_e) \cdot z + e^{-2 \cdot a \cdot T_e} \end{aligned} \quad (29)$$

### 4.5.4 Bilan

Aucune méthode n'est parfaite et on n'aura jamais en même temps :

- même réponse indicielle
- même réponse impulsionnelle
- approximation (Euler, Tustin, etc...) de l'équation différentielle
- correspondance des pôles et des zéros
- correspondance des réponses fréquentielles

Ceci étant dit le choix de la période d'échantillonnage est primordial. Si la fréquence d'échantillonnage est beaucoup plus faible que la dynamique de la fonction de transfert à approximer la différence entre chaque méthode sera minimale.





## 5. IDENTIFICATION DES SYSTEMES ECHANTILLONES

Dans le présent chapitre on donne les bases des méthodes d'identification. Pour une approche complète, voir le cours de troisième année "Identification avion" [6] et les ouvrages [1] et [7].

Même si l'on dispose d'un modèle théorique du processus que l'on souhaite commander il est presque toujours nécessaire de l'affiner en procédant à des essais sur le système réel. Bien souvent le seul modèle dont l'on dispose est ce modèle issu de l'expérimentation. Différentes méthodes s'appliquent selon l'objectif recherché et le niveau de précision exigé. On distinguera :

- les modèles paramétriques
- les modèles simples
- les modèles non paramétriques

### 5.1 Identification d'un modèle ARX par la méthode des moindres carrés récursifs

A partir de cet exemple, le plus simple qui soit, on pose les bases de toutes les méthodes d'identification.

On considère un système d'entrée  $u(t)$  et de sortie  $y(t)$  représenté par l'équation de récurrence suivante :

$$y(t) + a_1 \cdot y(t-1) + a_2 \cdot y(t-2) + \dots + a_n \cdot y(t-n) = b_0 \cdot u(t) + b_1 \cdot u(t-1) + b_2 \cdot u(t-2) + \dots + b_m \cdot u(t-m) \quad (30)$$

L'objectif est, à partir d'essais et de mesures, d'identifier les meilleures valeurs possibles pour les coefficients  $a_i$  et  $b_i$ .

L'équation (30) peut être réécrite pour mettre en évidence la *prédiction* de la sortie  $y(t)$  à partir des mesures passées :

$$y(t) = -a_1 \cdot y(t-1) - a_2 \cdot y(t-2) - \dots - a_n \cdot y(t-n) + b_0 \cdot u(t) + b_1 \cdot u(t-1) + b_2 \cdot u(t-2) + \dots + b_m \cdot u(t-m) \quad (31)$$

Soit :

$$y(t) = \varphi^T(t) \cdot \theta(t)$$

avec :

$$\begin{cases} \varphi^T(t) = [-y(t-1) \ \dots \ -y(t-n) \ , \ u(t-1) \ \dots \ u(t-m)] \\ \theta(t) = [a_1 \ \dots \ a_n \ , \ b_1 \ \dots \ b_m] \end{cases} \quad (32)$$

Le but est de trouver le meilleur ensemble de paramètres  $\theta(t)$  : celui qui, à partir d'un ensemble de mesures  $\varphi(t)$  permet de prédire avec la meilleure précision la sortie effectivement mesurée.

Supposons qu'à la suite d'une expérience on dispose d'un ensemble de mesures  $\{y(t), u(t)\}_{t=1:N}$  et que l'ensemble des paramètres  $\theta$  n'est pas bien connu. On construit la séquence des *sorties prédites* :

$$\hat{y}(t) = \varphi^T(t) \cdot \theta(t) \quad (33)$$

Le paramètre  $\theta$  est celui qui minimise une norme quadratique :

$$\theta = \arg \max \left( \frac{1}{N} \sum_{t=1}^N (y(t) - \hat{y}(t))^2 \right) = \arg \max \left( \frac{1}{N} \sum_{t=1}^N (y(t) - \varphi^T(t) \cdot \theta(t))^2 \right) \quad (34)$$

Compte tenu du fait que l'équation est linéaire au départ et que le terme à minimiser est du type quadratique la solution est simplement obtenue par une pseudo-inverse :

$$\theta = \left[ \sum_{t=1}^N (\varphi(t) \cdot \varphi^T(t)) \right]^{-1} \sum_{t=1}^N (\varphi(t) \cdot y(t)) \quad (35)$$

On obtient donc la solution classique des moindres carrés. La formule (35) est le plus souvent utilisée sous sa forme récursive :

$$\theta(t) = \theta(t-1) + F(t) \cdot \varphi(t) \cdot (y(t) - \theta^T(t-1) \cdot \varphi(t))$$

avec

$$F(t)^{-1} = F(t-1)^{-1} + \varphi(t) \cdot \varphi(t)^T \quad (36)$$

ou

$$F(t) = F(t-1) - \frac{F(t-1) \cdot \varphi(t) \cdot \varphi(t)^T \cdot F(t-1)}{1 + \varphi(t)^T \cdot F(t-1) \cdot \varphi(t)}$$

(On laisse au lecteur le plaisir d'effectuer lui-même les démonstrations, qu'il peut du reste trouver dans [1] et [7]).

## 5.2 L'identification paramétrique en général et en pratique

L'exemple précédent a permis de mettre en évidence tous les aspects de la méthodologie classique de l'identification :

- il a fallu choisir un ensemble de mesures  $\{y(t), u(t)\}_{t=1:N}$ . Le choix du signal d'excitation  $u(t)$  est en pratique primordial,
- il a fallu choisir une structure pour le modèle à identifier (ici les ordres  $n$  et  $m$ ),
- il a fallu choisir un critère (ici un critère quadratique sur l'erreur entre la sortie et la sortie prédite),
- il a fallu choisir un algorithme et le paramétrer (ici l'algorithme des moindres carrés récursifs qu'il a fallu initialiser)

Enfin un point non traité dans le chapitre précédent concerne la validation :

- il aurait fallu déterminer une méthode pour comparer différents modèles et choisir le meilleur

### 5.2.1 Choix du signal d'excitation, traitement des mesures

De manière générale le système que l'on doit identifier doit être excité dans tout le domaine fréquentiel qui nous intéresse. Le signal classiquement utilisé est une séquence pseudo binaire aléatoire (Pseudo Random Binary Signal) tel celui représenté sur la Figure 17.

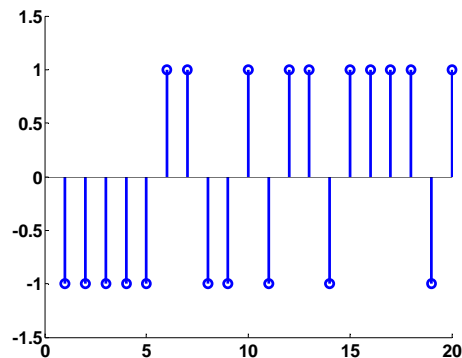


Figure 17 : exemple d'une séquence PRBS

Le signal d'excitation doit se superposer au signal de commande correspondant au point de fonctionnement étudié. Son amplitude doit être faible par rapport à l'amplitude des commandes du système (quelques %).

Une fois le système excité et les mesures effectuées il faut supprimer les valeurs moyennes des mesures.

### 5.2.2 Choix d'une structure de modèle

Le modèle étudié dans le chapitre précédent est de type ARX. On peut le noter sous forme polynomiale :

$$A \cdot y = B \cdot u \quad (37)$$

Ce modèle correspond en fait à un parti-pris sur la manière dont on traite les perturbations. En réalité on prend en compte un signal d'erreurs possibles  $e$  (de moyenne nulle) tels que :

$$A \cdot y = B \cdot u + e \quad (38)$$

Si l'on fait cette hypothèse de modélisation alors la méthode optimale est celle des moindres carrés. Malheureusement cette manière de prendre en compte les perturbations est bien restrictive. D'autres modélisations sont possibles, adaptées à d'autres algorithmes, par exemple le modèle OE :

$$y = \frac{B}{F} \cdot u + e \quad (39)$$

Il est aussi possible de "structurer" le modèle de bruit comme par exemple dans les modèles ARMAX ou BJ :

$$\begin{aligned} A \cdot y &= B \cdot u + C \cdot e \\ y &= \frac{B}{F} \cdot u + \frac{C}{D} \cdot e \end{aligned} \quad (40)$$

La structure du modèle étant choisie il faut choisir l'ordre de ses différents polynômes.

### 5.2.3 Choix d'un critère à minimiser

Dans la plupart des cas le critère à minimiser est l'erreur quadratique entre la sortie et la sortie prédite à un coup (one step ahead prediction). Ceci n'est bien évidemment pas restrictif et peut être généralisé à la prédiction à  $N$  coups.

### 5.2.4 Choix d'un algorithme

Même si chaque structure de modèle est généralement associée à un algorithme "optimal" celui-ci doit généralement se paramétrer.

Prenons par exemple l'algorithme des moindres carrés récursifs tel que présenté dans (36). Cette équation peut être modifiée en intégrant un "facteur d'oubli"  $\lambda$  :

$$\begin{aligned} \theta(t) &= \theta(t-1) + F(t) \cdot \varphi(t) \cdot (y(t) - \theta^T(t-1) \cdot \varphi(t)) \\ \text{avec} & \\ F(t)^{-1} &= \lambda \cdot F(t-1)^{-1} + \varphi(t) \cdot \varphi(t)^T \end{aligned} \quad (41)$$

Ce paramètre (typiquement entre 0.95 et 0.99) permet, comme son nom l'indique, "d'oublier" les anciennes mesures et de se concentrer sur les mesures les plus récentes.

### 5.2.5 Validation

Une fois l'algorithme d'identification appliqué et le modèle obtenu (ce qui prend généralement quelques secondes avec un ordinateur moderne) il reste à expertiser celui-ci. Selon la "qualité" du modèle obtenu on remet en cause l'une ou l'autre des étapes (signal d'excitation, structure du modèle, algorithme) et on réitère jusqu'à obtenir un modèle satisfaisant.

En pratique on essaie un certain nombre de modèles avant de choisir le "meilleur".

Le premier critère de qualité est la capacité du modèle à prédire effectivement la sortie (sortie prédite à un coup, sortie prédite à N coups). Il faut pour cela utiliser un jeu de mesures non utilisé pour l'identification. (En pratique le jeu de mesures initiales est coupé en deux : une partie pour l'identification et une partie pour la validation).

Le second critère de qualité est l'analyse du résidu : la composition spectrale de l'erreur de prédiction.

### 5.2.6 Utilisation de logiciels

En pratique il est commode d'utiliser un célèbre outil de calcul scientifique et sa toolboxe d'identification.

Comme les calculs proprement dit ne prennent que quelques secondes il est facile de réaliser un grand nombre d'identifications et de stocker les différents modèles pour les comparer ensuite. On est rapidement noyé sous le grand nombre de modèles possibles...

## 5.3 Identification "simple"

Il est souvent suffisant d'obtenir un modèle "simple" de type premier ordre et un simple essai indiciel convient souvent. Le système est excité avec un échelon unité et on cherche le meilleur modèle du premier ordre correspondant.

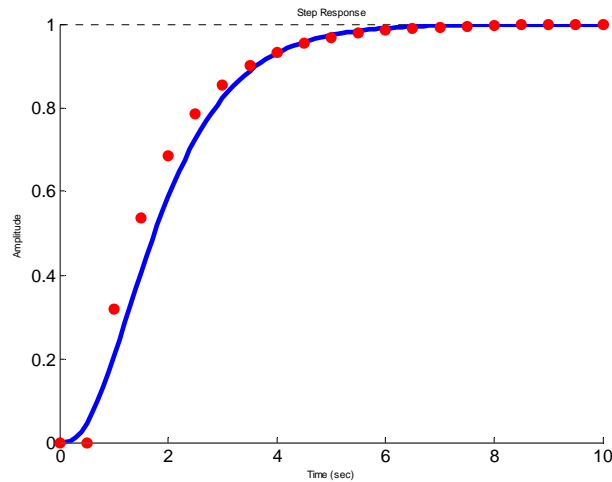


Figure 18 : identification sur un essai indiciel

Dans l'exemple de la Figure 18 on superpose le résultat d'une mesure (en bleu) avec la simulation d'un modèle discret avec une période de 0.5s. Le modèle cherché est du premier ordre avec un retard pur  $d$  :

$$y(z) = \frac{(1 + c_1) \cdot z^{-d}}{1 + c_1 \cdot z^{-1}} \cdot u(z) \quad (42)$$

Il s'agit de trouver l'entier  $d$  et le nombre réel  $c_1$  tels que la réponse indiciale du modèle corresponde pour le mieux aux mesures issues de l'essai. Ceci peut se formaliser par exemple de la manière suivante :

$$(c_1 \quad d_1) = \arg \min \sum_{t=1}^N (y_{\text{modèle}}(t) - y_{\text{mesure}}(t))^2 \quad (43)$$

Cette optimisation peut se faire à la main ou grâce à une procédure automatisée (avec le célèbre outil de calcul scientifique).

Dans le cas où le système comprend naturellement une intégration (par exemple la commande d'un moteur à courant continu par une tension et la mesure de l'angle de rotation) il est utile de prendre en compte explicitement cette intégration dans le modèle :

$$y(z) = \frac{1}{1 - z^{-1}} \cdot \frac{(1 + c_1) \cdot z^{-d}}{1 + c_1 \cdot z^{-1}} \cdot u(z) \quad (44)$$

La procédure est la même mais au lieu de prendre en compte la sortie  $y(t)$  on prend en compte la dérivée  $y'(t)$  :

$$y'(t) = y(t) - y(t-1) \quad (45)$$

## 5.4 Identification non paramétrique

Pour terminer ce panorama rapide sur les méthodes d'identification il faut parler des méthodes non paramétriques : celles qui permettent de tracer la réponse impulsionnelle ou la réponse fréquentielle du système.

### 5.4.1 Estimation de la réponse impulsionnelle

La réponse impulsionnelle d'un système peut être obtenue simplement en appliquant une entrée de type "impulsion" au système et d'en mesurer les conséquences.

La seconde méthode consiste à utiliser la méthode des moindres carrés (ou une méthode dérivée) comme vue au chapitre 5.1. La réponse impulsionnelle d'un système peut en effet être vue comme la fonction de transfert particulière où le dénominateur est de degré nul et le numérateur comporte un nombre infini (en pratique : grand) d'éléments ( $n = 0$  et  $m$  grand) :

$$y(t) = b_0 \cdot u(t) + b_1 \cdot u(t-1) + b_2 \cdot u(t-2) + \dots + b_m \cdot u(t-m) \quad (46)$$

La dernière méthode, qui sera développée ici est l'analyse d'intercorrélation.

Par définition l'intercorrélation  $R_{yu}$  entre deux signaux est :

$$R_{yu}(\tau) = E[(y(t) - E(y(t))) \cdot (u(t-\tau) - E(u(t-\tau)))] \quad (47)$$

Dans cette formule l'espérance mathématique peut être approchée grâce à la loi des grands nombres :

$$R_{yu}(\tau) \approx R_{yu}^N(\tau) = \lim_{N \rightarrow \infty} \frac{1}{N} \sum_{t=1}^N [(y(t) - m_y) \cdot (u(t-\tau) - m_u)] \quad (48)$$

On s'intéresse maintenant à un système tel que décrit par l'équation impulsionnelle (46) et on suppose que le signal d'entrée est un bruit blanc. Le signal d'entrée vérifie donc :

$$\begin{cases} \tau \neq 0 & R_{uu}(\tau) = 0 \\ \tau = 0 & R_{uu}(\tau) = R \end{cases} \quad (49)$$

Il vient alors naturellement :

$$b_k \approx \frac{1}{R} R_{yu}(k) \quad (50)$$

### 5.4.2 Estimation de la réponse fréquentielle

On utilise ici le spectre d'intercorrélation. Par définition le spectre d'intercorrélation entre deux signaux est :

$$\Phi_{yu}(\omega) = \sum_{\tau=-\infty}^{\infty} R_{yu}(\tau) \cdot e^{-i\omega\tau} \quad (51)$$

On suppose que le signal d'entrée  $u(t)$  et le signal de sortie  $y(t)$  sont reliés par une fonction de transfert linéaire du type :

$$y(t) = F \cdot u(t) + v(t) \quad (52)$$

où  $v(t)$  est une perturbation indépendante de  $u(t)$ . On a alors les relations :

$$\Phi_{yu}(\omega) = F(j\omega) \cdot \Phi_{uu}(\omega) \quad (53)$$

Il reste à estimer les spectres d'intercorrélation pour remonter à la réponse fréquentielle  $F(j\omega)$  recherchée. Une idée naturelle est d'utiliser la relation (48) mais cela marche très mal dans presque tous les cas, même en choisissant  $N$  assez grand. L'amélioration est d'utiliser une "fenêtre" pour favoriser les valeurs faibles de  $\tau$  par rapport aux valeurs élevées. La solution classique utilisant la *fenêtre de Hamming* donne :

$$\Phi_{yu}^N(\omega) = \sum_{\tau=-\gamma}^{\gamma} R_{yu}^N(\tau) \cdot w_{\gamma}(\tau) \cdot e^{-i\omega\tau} \quad (54)$$

avec

$$\begin{cases} w_{\gamma}(\tau) = \frac{1}{2} \left( 1 + \cos\left(\frac{\pi \cdot \tau}{\gamma}\right) \right) & |\tau| < \gamma \\ w_{\gamma}(\tau) = 0 & |\tau| > \gamma \end{cases}$$

On utilise ensuite la relation (51) pour estimer la réponse fréquentielle :



$$F(j \cdot \omega) = \frac{\Phi_u^N(\omega)}{\Phi_{yu}^N(\omega)} \quad (55)$$



## 6. SYSTEMES BOUCLES

### 6.1 Fonction de transfert en boucle fermée

Un système et sa commande par ordinateur peut être représenté par un ensemble de fonctions de transfert formant un schéma-bloc plus ou moins compliqué (Figure 19).

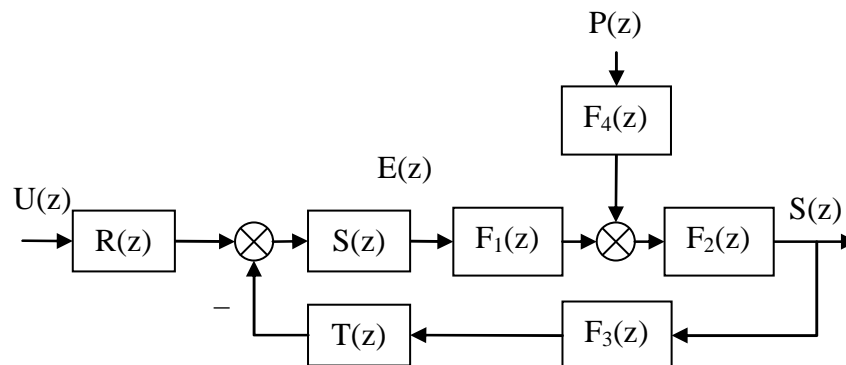


Figure 19 : système discret commandé par ordinateur. Le correcteur est représenté par les fonctions R,S,T, le système par  $F_1$  et  $F_2$ , le modèle de perturbations par  $F_4$  et le modèle des actionneurs par  $F_4$ .

Quelle que soit la structure d'un système asservi il est possible de se ramener à une forme canonique comprenant une fonction de transfert en boucle ouverte en combinant judicieusement les fonctions de transfert du procédé, du régulateur et des perturbations. La Figure 20 représente par exemple le système de la Figure 19 où  $F$  est une combinaison des fonctions de transfert du système, du régulateur, des perturbations et des capteurs.

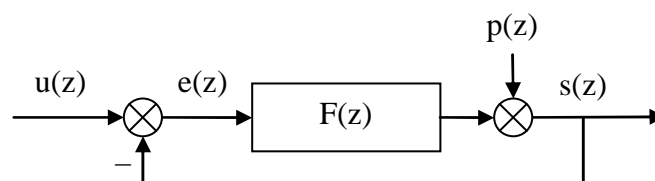


Figure 20 : système discret, notations.

Par défaut on considère une contre-réaction négative et une perturbation additive sur la sortie. La fonction de transfert  $F$  est appelée *fonction de transfert en boucle ouverte*. Comme pour les systèmes continus la *fonction de transfert en boucle fermée* est obtenue simplement à partir de  $F$  (58).

$$F(z) = \frac{s(z)}{e(z)} = \frac{B(z^{-1})}{A(z^{-1})} \quad (56)$$

$$G(z) = \frac{s(z)}{u(z)} = \frac{B(z^{-1})}{A(z^{-1}) + B(z^{-1})}$$

## 6.2 Erreur statique

L'erreur statique du système en boucle fermée, comme pour le système en boucle ouverte, est simplement le rapport des coefficients.

$$G_0 = \frac{\sum_{i=0}^{i=m} b_i}{1 + \sum_{i=0}^{i=m} b_i + \sum_{i=1}^{i=n} a_i} \quad (57)$$

Si l'on souhaite que l'asservissement confère un gain statique unitaire au système en boucle fermée (c'est-à-dire que l'on force la sortie  $s(k)$  à tendre vers la valeur de consigne  $u(k)$ ) il faut que ce rapport de coefficients soit égal à un. On montre facilement que cela est obtenu si la fonction de transfert  $F(z)$  comporte un intégrateur :

$$F(z) = \frac{s(z)}{e(z)} = \frac{1}{1 - z^{-1}} \cdot \frac{B(z)}{A'(z)} \quad (58)$$

## 6.3 Rejet de perturbations

La fonction de transfert entre la perturbation  $p(z)$  et la sortie  $y(z)$  est appelée fonction de sensibilité perturbation-sortie, notée  $S_{yp}$ .

$$S_{yp}(z) = \frac{s(z)}{p(z)} = \frac{A(z^{-1})}{A(z^{-1}) + B(z^{-1})} \quad (59)$$

Si l'on souhaite qu'une perturbation de type échelon soit parfaitement rejetée il faut que  $A(1) = 0$ , on retrouve donc la condition (58) à savoir que le système doit comporter une intégration dans la boucle directe. On peut être en outre être amené à spécifier que les perturbations ne soient pas amplifiées dans une certaine bande de fréquence. On rajoute donc une contrainte du type  $|S_{yp}(e^{j2\pi f})| < S_{max}$  pour  $f$  appartenant à un certain domaine de  $[0, 0.5]$ .

## 6.4 Stabilité des systèmes à simple gain de rétroaction

Les critères de Routh et de Jury (donnés dans le chapitre 3.2.7 p13) ou le calcul numérique des pôles permettent de déterminer la stabilité du système en boucle fermée si l'on connaît sa fonction de transfert.

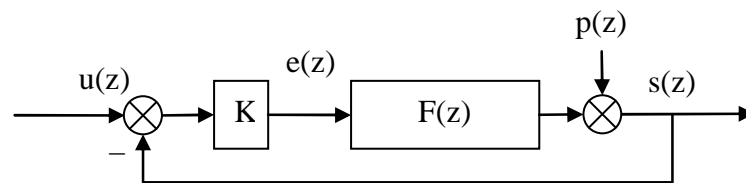


Figure 21 : système bouclé par un simple gain.

On s'intéresse ici au cas particulier où l'asservissement est réalisé par un simple gain de rétroaction  $K$  (Figure 21). On cherche la valeur de  $K$  qui amène le système en limite de stabilité. L'outil adapté ici est le lieu d'Evans ou lieu des racines (root locus) qui permet de visualiser l'évolution des pôles du système bouclé lorsque le gain  $K$  varie.

Prenons l'exemple d'un système comportant trois pôles et deux zéros. L'exemple est donné sous forme de script Matlab :

### Matlab

```
% Définition d'un système discret :
sys=zpk([-2.037 -0.1406],[0.6065 0.3679 0.3679],1,1)
%Tracé du lieu des racines
rlocus(sys);
```

Le script précédent produit le tracé de la Figure 22. On pourrait y lire que la limite de stabilité est obtenue pour  $K=0.2$  environ (détail de la Figure 23).

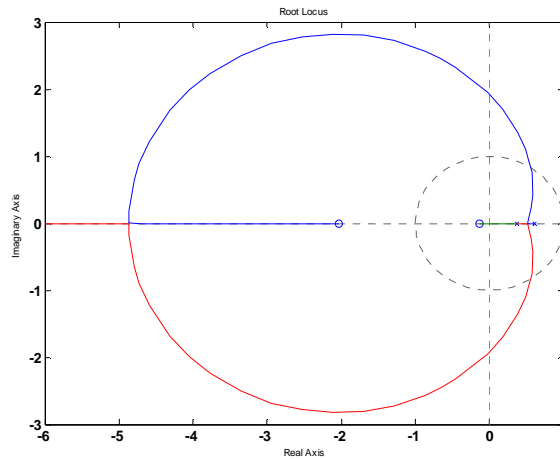


Figure 22 : lieu des racines

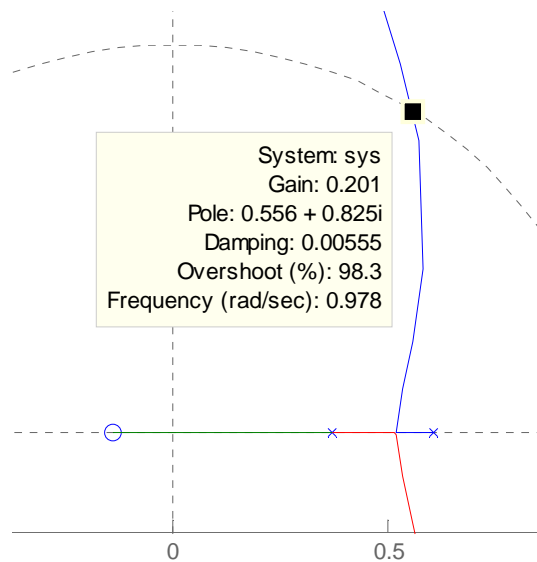


Figure 23 : lieu des racines : détail de la limite de stabilité

On voit sur la Figure 22 que le lieu des racines d'un système échantillonné ressemble à celui d'un système continu :

- on commence par tracer les pôles (×) et les zéros (o) du système en boucle ouverte.
- les pôles du système en boucle fermée décrivent un lieu qui part des pôles (pour  $K = 0$ ) et arrive aux zéros (pour  $K = \infty$ )
- lorsque il y a  $n-m$  plus de pôles que de zéros le système comporte  $n-m$  branches asymptotiques.

On ne décrira pas ici les diverses propriétés du lieu des racines qui permettent de le tracer à la main puisque il est facile à obtenir avec un ordinateur.

Il peut aussi être utilisé pour régler "à la main" le gain permettant d'obtenir un ou plusieurs modes dominants ayant certaines propriétés de fréquence et d'amortissement. On voit sur la Figure 24 que le gain 0.017 permet d'obtenir un pôle double d'amortissement 0.706 et de fréquence 0.65 rad/s et un pôle simple de fréquence 1.63 rad/s.

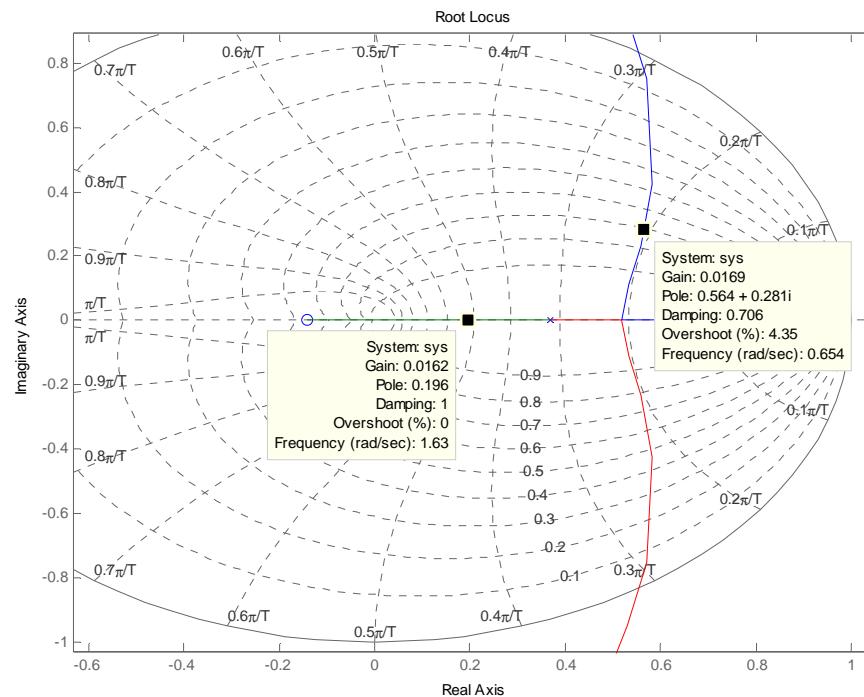


Figure 24 : utilisation du lieu des racines pour régler le gain de la boucle fermée

L'outil rltool de Matlab permet de nombreuses manipulations graphiques de ce type.

## 6.5 Analyse des systèmes bouclés dans le domaine fréquentiel : stabilité

Comme dans le cas continu il est possible de prévoir des propriétés, en particulier la stabilité, d'un système en boucle fermée à partir de l'examen de la fonction de transfert en boucle ouverte.

On ne redémontrera pas ici le théorème de Nyquist mais on se contente de rappeler l'énoncé.

On rappelle que le *lieu de Nyquist* est le lieu de  $F(j2\pi f)$  dans le plan complexe pour  $f$  variant de 0 à  $+\infty$ . A noter que le lieu pour la zone  $[0, 0.5]$  est symétrique de celui pour la zone  $[0.5, 1]$ . Pour cette raison on ne représente généralement que le lieu pour  $f$  appartenant à  $[0, 0.5]$ .

### 6.5.1 Théorème de Nyquist dans le cas où le système est stable en boucle ouverte (critère du revers)

Le système est stable en boucle fermée si le lieu de Nyquist de la fonction de transfert en boucle ouverte parcouru dans le sens des fréquences croissantes de 0 à 0.5 laisse le point critique  $(-1,0)$  à gauche.

La Figure 25 représente par exemple le lieu de Nyquist d'une fonction de transfert en boucle ouverte qui sera stable en boucle fermée.

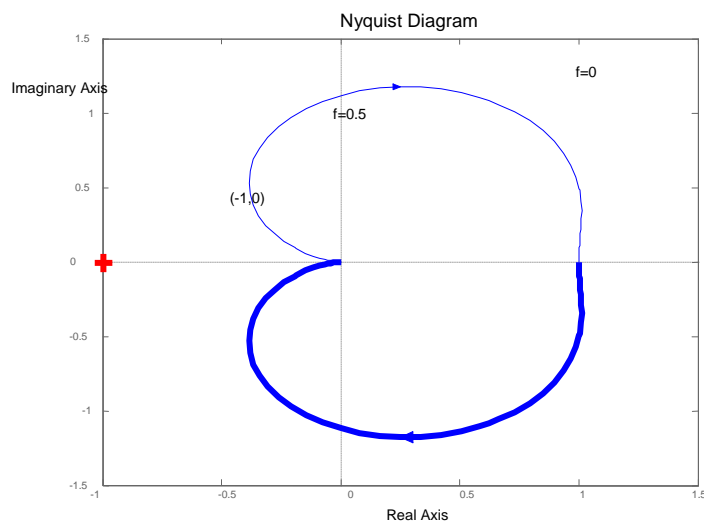


Figure 25 : lieu de Nyquist d'un système stable

### 6.5.2 Théorème de Nyquist dans le cas où le système est instable en boucle ouverte

Si le système en boucle ouverte comporte  $N$  pôles instables il sera stable en boucle fermée si le lieu de Nyquist de la fonction de transfert en boucle ouverte parcouru dans le sens des fréquences croissantes de  $-0.5$  à  $0.5$  entoure le point critique  $(-1,0)$   $N$  fois dans le sens des aiguilles d'une montre..

### 6.5.3 Interprétations du lieu de Nyquist

Comme dans le cas continu, la distance à laquelle on passe du point critique renseigne sur la robustesse du système bouclé. La distance sera exprimée de différente manière (Figure 26) :

- marge de module  $\Delta M$  : plus petite distance entre le lieu des racines et le point critique,



- marge de gain  $\Delta G$
- marge de phase  $\Delta P$

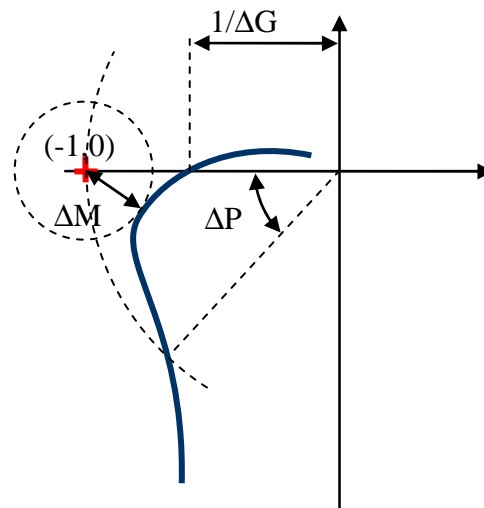


Figure 26 : Marges de stabilité

Les valeurs typiques de ces marges sont données ci-après :

- $\Delta G \geq 6\text{dB}$
- $\Delta M \geq -6\text{dB}$
- $30^\circ \leq \Delta P \leq 60^\circ$

La distance entre le point critique et le lieu des racines est la norme de  $1+F(j2\pi f)$ . On peut donc exprimer la marge de module en fonction de la fonction de sensibilité :

$$\begin{aligned} \Delta M &= \min |1 + F(j \cdot \omega)|_{\omega \in [0, \pi]} \\ \Delta M &= \min \left| 1 + \frac{B(j \cdot \omega)}{A(j \cdot \omega)} \right|_{\omega \in [0, \pi]} = \max \left| \frac{A(j \cdot \omega)}{A(j \cdot \omega) + B(j \cdot \omega)} \right|_{\omega \in [0, \pi]} \quad (60) \\ \Delta M &= \max |S_{yp}(j \cdot \omega)|_{\omega \in [0, \pi]}^{-1} \\ \Delta M_{\text{dB}} &= -\max |S_{yp}(j \cdot \omega)|_{\text{dB}} \end{aligned}$$

Si l'on trace l'évolution de  $S_{yp}$  en fonction de  $f$  on obtient donc la valeur de la marge de module.



# 7. METHODES DE REGULATION DES SYSTEMES ECHANTILLONES

## 7.1 Introduction

Les bases théoriques permettant de concevoir le correcteur pour un système échantillonné ont été posées dans les chapitres précédents. Ce chapitre fait le bilan des connaissances acquises et donne quelques solutions pour calculer des correcteurs à partir d'un cahier des charges précis.

Les systèmes que l'on cherche à commander sont généralement, par nature, continus. La réalisation d'un correcteur par un ordinateur implique que l'on génère une équation de commande discrète (généralement une équation récurrente). L'obtention de cette équation de commande peut se faire de deux manières :

- la première méthode consiste à d'abord synthétiser un correcteur continu (en utilisant les outils classiques de l'automatique et la simulation) et discrétiser ensuite ce correcteur. C'est l'objet du chapitre 7.2.2.
- la seconde méthode consiste à tenir la représentation discrète du système continu associé à son interface numérique (convertisseurs analogique-numérique et numérique-analogique) comme expliqué dans le chapitre 4.5 puis synthétiser directement le correcteur discret. Cela sera expliqué dans le chapitre 7.3 et suivants.

Ce choix peut être résumé par la figure suivante :

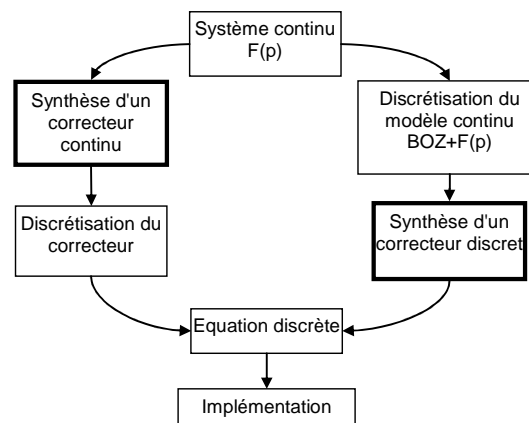


Figure 27 : Réalisation d'un correcteur discret.

Au-delà des affaires de goût, le choix de la méthode sera souvent dicté par l'application. En effet, la première méthode est d'autant meilleure que le correcteur discrétisé approche correctement le correcteur continu. Cela implique que le temps d'échantillonnage soit très faible par rapport aux constantes de temps du système

bouclé. Dans le cas où les contraintes temps réel sont fortes (capacités de calcul partagées par exemple) il sera préférable de synthétiser un correcteur discret.

## 7.2 Structure d'un système commandé par ordinateur, hypothèses

Par la suite on se ramènera toujours à un système tel que celui représenté sur la Figure 28 en cherchant un correcteur  $K(z)$  linéaire. On montre que quel que soit le correcteur linéaire on peut se ramener à la forme représenté sur la Figure 29 (forme RST).

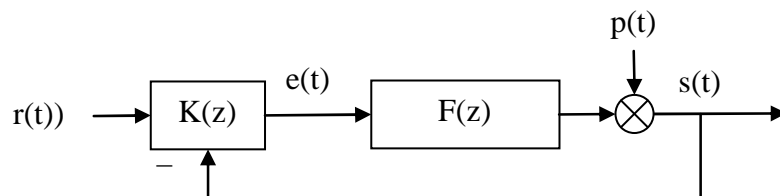


Figure 28 : système bouclé, forme générale.

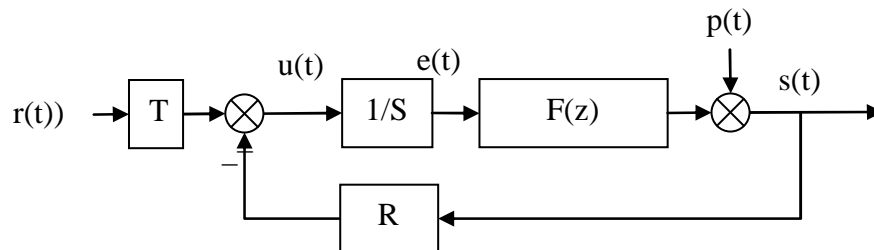


Figure 29 : système bouclé, forme canonique.

### 7.2.1 Discrétisation, choix de la période d'échantillonnage

Du point de vue du ordinateur, l'extérieur (le procédé) est représenté par la fonction de transfert  $F(z)$ . L'hypothèse importante est que la commande (sortie du ordinateur) et la mesure (entrée du ordinateur) sont échantillonnées de manière synchrone à la même période  $T_e$ .

Le choix judicieux de la fréquence d'échantillonnage est très important. En pratique on choisit cette fréquence en fonction de la bande passante du système en **boucle fermée** :

$$f_e = (6 \text{ à } 25) f^{BF} \quad (61)$$

### 7.2.2 Discrétisation d'un correcteur continu

La première méthode pour obtenir les équations d'un correcteur numérique est de faire sa synthèse avec les outils de l'automatique continue puis de discrétiser la fonction de transfert du correcteur continu obtenu (voir pour cela le chapitre 4.2 *Echantillonnage des systèmes continus*).

On va voir par la suite que la synthèse directe d'un correcteur discret offre plus de degré de liberté et permet en général de remplir un cahier des charges plus complet.

## 7.3 Régulateur PID numérique

Le régulateur le plus populaire est le régulateur PID. Nous allons voir dans un premier temps comment on génère un régulateur PID à partir des équations bien connues du régulateur PID continu et comment ces équations peuvent être généralisées pour obtenir plus de degrés de liberté pour le réglage.

### 7.3.1 Cas 1

Un régulateur PID continu a une fonction de transfert du type :

$$K_{\text{PID}}(s) = K \cdot \left( 1 + \frac{1}{T_i \cdot s} + \frac{T_d \cdot s}{1 + \frac{T_d}{N} \cdot s} \right) \quad (62)$$

La discrétisation de ce correcteur continu par la méthode d'Euler (Tableau 4 page 31) se fait en remplaçant la variable de Laplace  $s$  par le terme  $\frac{1-q^{-1}}{q^{-1} \cdot T_e}$ . On obtient :

$$K_{\text{PID}}(q) = K \cdot \left( 1 + \frac{1}{T_i \cdot \frac{1-q^{-1}}{T_e}} + \frac{T_d \cdot \frac{1-q^{-1}}{T_e}}{1 + \frac{T_d}{N} \cdot \frac{1-q^{-1}}{T_e}} \right) \quad (63)$$

En réarrangeant on obtient la forme générale RST (Figure 30):

$$K_{PID}(q) = K \cdot \frac{r_0 + r_1 \cdot q^{-1} + r_2 \cdot q^{-2}}{(1 - q^{-1}) \cdot (1 + s_1' \cdot q^{-1})}$$

$$\begin{cases} r_0 = K \cdot \left( 1 + \frac{T_e}{T_i} + \frac{N \cdot T_d}{T_d + N \cdot T_e} \right) \\ r_1 = -K \cdot \left( 1 + \frac{T_d}{T_d + N \cdot T_e} + \frac{T_e}{T_i} \cdot \frac{T_d}{T_d + N \cdot T_e} + \frac{N \cdot T_d}{T_d + N \cdot T_e} \right) \\ r_2 = K \cdot \left( \frac{T_d}{T_d + N \cdot T_e} \right) \\ s_1' = \frac{T_d}{T_d + N \cdot T_e} \end{cases} \quad (64)$$

Comme on le voit le correcteur obtenu comporte autant de degrés de liberté que le correcteur PID analogique. On reconnaît au dénominateur le terme  $(1 - q^{-1})$  en facteur qui assure l'effet d'intégration.

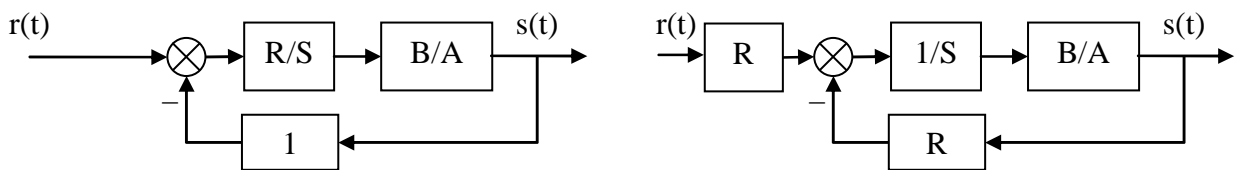


Figure 30 : PID sous forme RST

La fonction de transfert en boucle fermée obtenue est donnée par l'équation suivante :

$$H_{BF} = \frac{B \cdot R}{A \cdot S + B \cdot R} = \frac{B \cdot R}{P} \quad (65)$$

De manière générale le polynôme caractéristique  $P$  fixe les performances visées en boucle fermée. Le calcul de  $R$  et  $S$ ,  $P$  étant fixé et  $A$  et  $B$  étant connus, est réalisé en résolvant l'équation de Bézout  $P = A \cdot S + B \cdot R$ . (voir 9.1 Annexe 1 : Résolution de l'équation polynomiale de Bezout)

On voit que le numérateur  $B \cdot R$  est imposé une fois que  $R$  et  $S$  sont trouvés. Le régulateur introduit donc naturellement des zéros dans la boucle fermée.

Traisons l'exemple entièrement dans le cas où le système est modélisé comme un système du deuxième ordre avec faible retard. La fonction de transfert du procédé est donnée par :

$$H_{BO} = \frac{B}{A} = \frac{b_1 \cdot q^{-1} + b_2 \cdot q^{-2}}{1 + a_1 \cdot q^{-1} + a_2 \cdot q^{-2}} \quad (66)$$

Le correcteur PID rajoutant deux pôles, le polynôme caractéristique en boucle fermée spécifié doit posséder quatre pôles. Le polynôme caractéristique en boucle fermée doit donc être de la forme :

$$\begin{aligned} P &= 1 + p_1 \cdot q^{-1} + p_2 \cdot q^{-2} + p_2 \cdot q^{-3} + p_2 \cdot q^{-4} \\ P &= (1 + p_1' \cdot q^{-1} + p_2' \cdot q^{-2}) \cdot (1 + \alpha_1 \cdot q^{-1}) \cdot (1 + \alpha_2 \cdot q^{-1}) \end{aligned} \quad (67)$$

Et il doit être identifié au produit  $AS + BR$  avec  $R$  et  $S$  ayant les formes données par (64) :

$$\begin{aligned} &(1 + p_1' \cdot q^{-1} + p_2' \cdot q^{-2}) \cdot (1 + \alpha_1 \cdot q^{-1}) \cdot (1 + \alpha_2 \cdot q^{-1}) \\ &= (1 + a_1 \cdot q^{-1} + a_2 \cdot q^{-2}) \cdot (1 - q^{-1}) \cdot (1 + s_1' \cdot q^{-1}) \\ &+ (b_1 \cdot q^{-1} + b_2 \cdot q^{-2}) \cdot (r_0 + r_1 \cdot q^{-1} + r_2 \cdot q^{-2}) \end{aligned} \quad (68)$$

Les coefficients  $p_1'$  et  $p_2'$  définissent un modèle du second ordre aux caractéristiques (pulsation propre  $\omega_0$  et amortissement  $\sigma$  choisis et coefficients déduits conformément au Tableau 3 page 25). Les coefficients  $\alpha_1$  et  $\alpha_2$  correspondent à des pôles auxiliaires aperiodiques. Les choix typiques pour ces coefficients sont :

$$-0.05 < \alpha_1, \alpha_2 < -0.5 \quad (69)$$

On montre que la résolution de (68) est réalisée par une inversion matricielle simple :

$$M \cdot x = p \quad (70)$$

avec

$$\begin{cases} x^T = [1 & s_1' & r_0 & r_1 & r_2] \\ p^T = [1 & p_1 & p_2 & p_3 & p_4] \end{cases}$$

et

$$M = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ a_1' & 1 & b_1 & 0 & 0 \\ a_2' & a_1' & b_2 & b_1 & 0 \\ a_3' & a_2' & 0 & b_2 & b_1 \\ 0 & a_3' & 0 & 0 & b_2 \end{pmatrix} \quad (71)$$

On montre que M est inversible à la condition que A et B soient premiers entre eux, c'est-à-dire que A et B n'aient pas de pôles et de zéros en commun.

(Le problème peut aussi être résolu en utilisant la résolution de l'équation de Bézout).

### 7.3.2 Cas 2

Comme on l'a vu le correcteur PI numérique directement transposé du correcteur PID analogique décrit dans le chapitre 7.3.1 a l'inconvénient de ramener des zéros au dénominateur de la fonction de transfert en boucle fermée (équation 65). Il est possible, grâce à la flexibilité du régulateur numérique d'imposer au correcteur de ne pas ajouter de zéros en boucle fermée. La fonction de transfert désirée en boucle fermée sera donc de la forme :

$$H_{BF} = \frac{P(1)}{B(1)} \cdot \frac{B}{P} \quad (72)$$

Le facteur  $P(1)/B(1)$  garantit le gain statique unitaire du système en boucle fermée (voir page 12). Si le régulateur a la forme générale RST alors les polynômes R S et T doivent vérifier :

$$H_{BF} = \frac{B \cdot T}{A \cdot S + B \cdot R} = \frac{P(1) B}{B(1) P} \quad (73)$$

S et R sont obtenus par la résolution de la même équation que précédemment (68). Le polynôme T est obtenu par simple identification :  $T = R(1)$ .

La structure, sous forme de schéma-bloc est donnée ci-dessous :

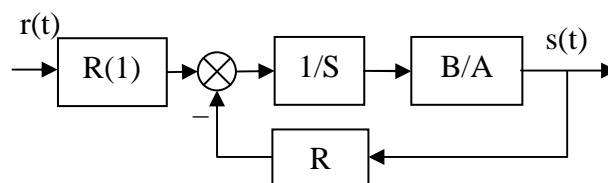


Figure 31 : PID2 sous forme RST

On montre que cette structure de PID, ramené en continu, correspond au PID canonique pour l'action intégrale. L'action proportionnelle et dérivée n'est appliquée que sur le signal  $y$ .

La structure du PID2 par rapport au PID1 a pour avantage de ne pas ramener de zéros à la boucle fermée. Cela se traduit par un dépassement moindre en transitoire lorsqu'on applique un échelon de commande. Par contre les deux systèmes réagissent de la même manière à un échelon de perturbation.



## 7.4 Placement de pôles

Les deux structures de PID décrites dans le chapitre précédent sont deux cas particulier d'une stratégie plus générale de placement de pôles.

La synthèse est plus générale que les PID présentés précédemment puisqu'il n'est pas nécessaire de faire d'hypothèses sur le degré de A et B, sur le retard, ni sur la présence ou non de zéros stables ou instables dans le procédé.

La structure du correcteur est la structure type RST telle que présentée dans la Figure 29. Pour la suite des calculs on posera pour la fonction de transfert les conventions suivantes :

$$H_{BO} = \frac{q^{-d} \cdot B(q^{-1})}{A(q^{-1})} \quad (74)$$

$$\begin{cases} A(q^{-1}) = 1 + a_1 \cdot q^{-1} + \dots + a_{n_A} \cdot q^{-n_A} \\ B(q^{-1}) = b_1 \cdot q^{-1} + \dots + b_{n_B} \cdot q^{-n_B} = q^{-1} \cdot B^*(q^{-1}) \end{cases}$$

Compte tenu de la forme RST du régulateur la fonction de transfert en boucle fermée vérifie :

$$H_{BF} = \frac{q^{-d} \cdot B(q^{-1}) \cdot T(q^{-1})}{P(q^{-1})} \quad (75)$$

$$P(q^{-1}) = A(q^{-1}) \cdot S(q^{-1}) + q^{-d} \cdot B(q^{-1}) \cdot R(q^{-1}) = 1 + p_1 \cdot q^{-1} + \dots + p_{n_P} \cdot q^{-n_P}$$

Le polynôme  $P(q^{-1})$  définit donc les pôles de la fonction de transfert en boucle fermée et traduit donc le comportement en *poursuite*.

De la même manière la fonction de transfert entre une perturbation et la sortie (fonction de sensibilité perturbation-sortie) est donnée par :

$$S_{yp} = \frac{A(q^{-1}) \cdot S(q^{-1})}{P(q^{-1})} \quad (76)$$

Ce même polynôme traduit donc aussi le comportement en *régulation*.

Les consignes en poursuite et en régulation peuvent être différentes comme cela est représenté par exemple sur la Figure 32.

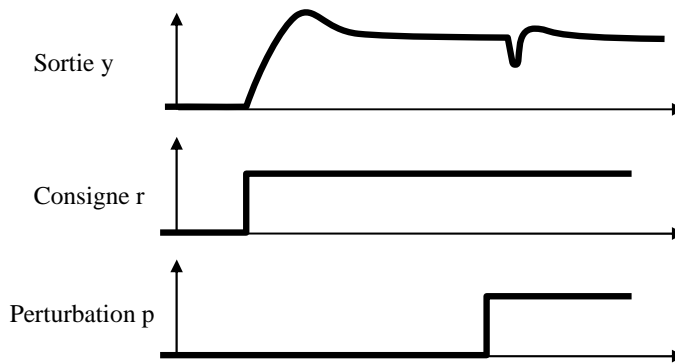


Figure 32 : Performances différentes en régulation et en poursuite

### 7.4.1 Choix des pôles en boucle fermée

De manière générale le comportement en boucle fermée est spécifié sous la forme d'une pulsation propre  $\omega_0$  et d'un amortissement  $\sigma$ , donc une fonction de transfert du second ordre. La spécification étant généralement donnée en continu il est nécessaire de traduire la valeur désirée de  $\omega_0$  et  $\sigma$  en un polynôme  $P_D(q^{-1})$  par discrétisation de la fonction de transfert continue (chapitre 4.5 page 29). Ce polynôme  $P_D$  traduit le comportement dominant du système bouclé.

Si l'on souhaite introduire un effet de filtrage supplémentaire dans certaines bandes de fréquence (réduction des bruits de mesure, adoucissement de la commande, robustesse) cela est décrit par un polynôme  $P_F$  comportant des pôles auxiliaires, plus rapides que les pôles de  $P_D$ .

Le polynôme caractéristique en boucle fermée est donc constitué de pôles dominants et de pôles auxiliaires :

$$P(q^{-1}) = P_D(q^{-1}) \cdot P_F(q^{-1}) \quad (77)$$

### 7.4.2 Régulation : calcul de R et S

Le polynôme  $P(q^{-1})$  étant fixé il s'agit de trouver R et S vérifiant :

$$P(q^{-1}) = A(q^{-1}) \cdot S(q^{-1}) + q^{-d} \cdot B(q^{-1}) \cdot R(q^{-1}) \quad (78)$$

Comme déjà vu au chapitre 7.3.1 on obtient une équation de Bézout qu'il s'agit de résoudre. (Voir aussi 9.1 Annexe 1 : Résolution de l'équation polynomiale de Bezout)

Il est parfois nécessaire de spécifier R et S avec des parties fixes  $H_R$  et  $H_S$ . Dans ce cas l'équation (78) est modifiée et on l'écrit sous la forme:

$$P(q^{-1}) = (A(q^{-1}) \cdot H_S(q^{-1})) \cdot S'(q^{-1}) + (q^{-d} \cdot B(q^{-1}) \cdot H_R(q^{-1})) \cdot R'(q^{-1}) \quad (79)$$

Les inconnues que l'on souhaite trouver sont  $S'$  et  $R'$ . Ceci est nécessaire dans les cas particuliers suivants :

- spécification d'erreur statique nulle,
- rejet d'une perturbation harmonique,
- blocage d'un signal,

Détaillons maintenant ces cas particuliers.

### Spécification d'une erreur statique nulle

On retrouve la nécessité d'avoir un terme intégrateur dans la boucle directe. La spécification se retrouve donc sur  $S$  :

$$S(q^{-1}) = (1 - q^{-1}) \cdot S'(q^{-1}) = H_S(q^{-1}) \cdot S'(q^{-1}) \quad (80)$$

### Rejet d'une perturbation harmonique

Si le système est perturbé par une perturbation harmonique à une fréquence fixe (par exemple 50Hz...) il est possible d'imposer à  $S$  d'avoir un module *nul* ou amorti à cette fréquence. Cela se traduit par la présence dans  $S$  d'une paire de zéros complexe à amortissement nul ou faible à cette fréquence :

$$S(q^{-1}) = (1 + \alpha \cdot q^{-1} + \beta \cdot q^{-2}) \cdot S'(q^{-1}) = H_S(q^{-1}) \cdot S'(q^{-1}) \quad (81)$$

### Blocage d'un signal

Pour certaines application on souhaite que le régulateur "laisse passer" certaines composantes du signal de sortie, c'est-à-dire qu'il ne réagisse pas à ces composantes. Cela se traduit par  $R=0$  à ces fréquences. On retrouve la même forme que précédemment :

$$R(q^{-1}) = (1 + \alpha \cdot q^{-1} + \beta \cdot q^{-2}) \cdot R'(q^{-1}) = H_R(q^{-1}) \cdot R'(q^{-1}) \quad (82)$$

### 7.4.3 Poursuite : calcul de T

Le schéma-bloc de l'asservissement est donné sur la Figure 33. Le terme  $T$  est décomposé en un terme  $T'$  et un terme  $B^*/A^*$ .

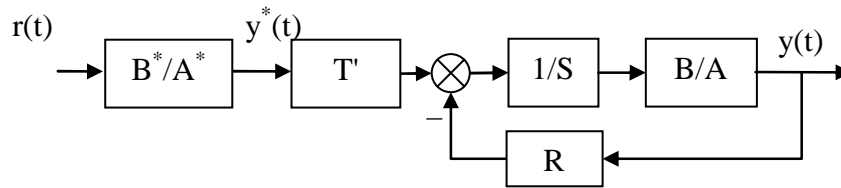


Figure 33 : placement de pôle

L'idée est de régler  $B^*$  et  $A^*$  de telle manière que le signal  $y^*$  ait la dynamique de poursuite désirée (par exemple un second ordre avec pulsation propre et amortissement donné) puis de régler  $T'$  de telle manière que le transfert entre  $y^*$  et  $y$  soit le plus "transparent" possible.

### Réglage de $T'$

Le transfert entre  $y^*$  et  $y$  est :

$$S_{yy^*}(q^{-1}) = T'(q^{-1}) \frac{q^{-d} \cdot B(q^{-1})}{P(q^{-1})} \quad (83)$$

Pour assurer le transfert le plus "transparent" possible il faut donc choisir

$$T'(q^{-1}) = \frac{P(q^{-1})}{B(1)} \quad (84)$$

Remarques :

- Si  $B(1)=0$  alors on remplace  $B(1)$  par 1 dans (84)
- De même dans (84)  $P(q^{-1})$  peut être remplacé par  $P_D$  correspondant à la dynamique dominante.

### Réglage de $B^*$ et $A^*$

$B^*$  et  $A^*$  sont choisis de telle manière que la fonction de transfert  $B^*/A^*$  corresponde à la dynamique de poursuite souhaitée.

## 7.5 Poursuite et régulation à objectifs indépendants

Cette méthode ressemble à la méthode de placement de pôles à ceci près que l'on va effectuer une simplification des zéros de  $B^*(q^{-1})$  : cela ne s'applique donc qu'aux systèmes avec des pôles stables. Plus précisément il faut qu'ils aient un amortissement suffisant ( $> 0,2$ ) de manière à ne pas être trop près du cercle unité.

La structure de l'asservissement est toujours celle donnée sur la Figure 33.

**Choix de P**

$P(q^{-1})$  est choisi de manière à simplifier  $B^*(q^{-1})$  (rappel : la forme de B est donnée dans l'équation 74). Comme indiqué dans le chapitre 7.4.2 page 60 il s'agit donc d'imposer les zéros de  $B(q^{-1})$  dans  $H_R(q^{-1})$  et  $H_S(q^{-1})$ .

**Choix de T**

Si l'on choisit  $T(q^{-1}) = P(q^{-1})$  alors le transfert entre  $y^*$  et  $y$  est :

$$S_{yy^*}(q^{-1}) = q^{-(d+1)} \quad (85)$$

**Réglage de  $B^*$  et  $A^*$** 

$B^*$  et  $A^*$  sont choisis de telle manière que la fonction de transfert  $B^*/A^*$  corresponde à la dynamique de poursuite souhaitée.



## 8. TECHNOLOGIE DE LA COMMANDE PAR ORDINATEUR

Du point de vue de l'automaticien le ordinateur est relié au système physique à piloter par un certain nombre d'intermédiaires plus ou moins gênants :

- les interfaces analogiques et numériques,
- le transport de l'information par bus et protocole,
- l'utilisation éventuelle de capteurs ou d'actionneurs « intelligents ».

Il ne s'agit pas ici de faire un exposé complet sur ces trois aspects mais de poser les problèmes qui se posent à l'automaticien lorsque le système qu'il doit commander est perturbé par ces éléments.

Enfin ce chapitre se termine sur les notions de « Hardware In the Loop » et de « prototypage rapide » qui sont partie intégrante des processus de réalisation de systèmes de contrôle.

### 8.1 Calculateurs et interfaces numériques / analogiques

Le premier problème à prendre en compte est la quantification de l'information. Du point de vue du ordinateur d'une part et du point de vue des convertisseurs d'autre part. Dans un premier temps, considérons les aspects performance de calcul lié à la nature du codage des nombres dans le ordinateur.

#### Les performances liées au types de variables

Prenons l'exemple où le concepteur de la loi de commande souhaite réaliser un correcteur de type proportionnel. Entre l'acquisition des mesures et l'écriture de la consigne il y aura la réalisation d'une de commande du type :

$$y = K \cdot (\text{consigne} - \text{mesure}) \quad (86)$$

En utilisant un langage de programmation évolué (par exemple le C) la ligne de code sera la transcription littérale de (86). Le programmeur devra décider de la définition et du type des variables, principalement du nombre d'octets utilisés pour stocker ses variables. Le Tableau 5 résume par exemple les types utilisés en langage C (norme ANSI)

Tableau 5 : Types ANSI C

Types entiers	char	8 bits
	short int	16 bits
	long int	32 bits
Types flottants (mantisse et exposant)	float	32 bits
	double	64 bits
	longdouble	128 bits

Une fois le code écrit en C l'exécution du programme sur un processeur cible ne peut se faire que par l'intermédiaire d'une compilation du programme. A chaque processeur est donc adapté une version de compilateur.

Les performances du programme (sa rapidité d'exécution) dépendront donc en partie de la qualité du compilateur. (Par exemple, une multiplication s'effectue t'elle bêtement par un décalage de tous les bits ou bien des solutions astucieuses sont-elles implémentées pour arrêter le décalage selon le contexte ?)

Le principal déterminant de la performance est cependant le processeur : son type (8 bits, 16 bits ou 64 bits) et son cadencement (en MHz ou GHz) sont les deux plus importants paramètres. Coder les variables sur 64 bits (double) dans le programme coûtera cher en temps de calcul si le processeur est un 8 bits. Le type de mémoire et leurs temps d'accès peuvent aussi être déterminants. Enfin, certains processeurs disposent d'un processeur spécifiquement dédié aux opérations mathématiques (coprocesseur mathématique).

Il n'est pas possible de dire exactement, pour tel type de compilateur associé à tel type de processeur, le temps que prendra par exemple une multiplication. En pratique on effectue une série de tests types (benchmark) et on affiche un temps moyen d'exécution.

En résumé, on peut être tenté, pour ne pas s'embêter, de coder toutes les variables en virgule flottante. Mais il faut bien savoir que cela a un coût en terme de temps de calcul, surtout si le processeur n'est pas adapté. Pour fixer les idées on peut dire qu'il y a un rapport 20 entre une addition réalisée en "short" et une addition réalisée en virgule flottante.

Enfin, l'utilisation de fonctions mathématiques (sinus, exponentielles, etc...) peut aussi avoir un coût très lourd si le travail n'est pas réalisé par un coprocesseur mathématique dédié. Le compilateur et sa librairie mathématique joue là encore un grand rôle.

### La quantification imposée par le processeur

Au-delà de l'aspect performance et temps de calcul, la quantification de l'information doit parfois aussi être prise en compte. Rappelons qu'un entier codé sur N bits peut prendre seulement  $2^N - 1$  valeurs différentes. Avant d'effectuer les calculs mathématiques dans le corps du programme il est donc judicieux d'appliquer un



facteur d'échelle de manière à utiliser l'amplitude maximale possible (entre 0 et  $2^N - 1$ ).

### **La quantification imposée par les conversions analogique et numérique**

Les convertisseurs analogique-numérique et numérique-analogique sont principalement caractérisés par leur amplitude de signal analogique d'entrée (généralement en volts, par exemple +/- 10V) et par leur résolution en bits (8 ou 10 bits pour les plus courants, jusqu'à 16 bits pour les convertisseurs haut de gamme).

Prenons l'exemple d'un capteur gyromètre transformant une vitesse de rotation en tension. Compte tenu de l'application le signal utile varie entre 2,4V et 2,6V. Si le signal mesuré est branché directement sur un convertisseur 0-10V 10bits alors un problème de conditionnement du signal se pose. En effet la quantification est de  $\frac{10V}{2^{10}-1} = 0,098V$ . Les 0,2V de signal utile seront donc discrétisés en  $\frac{0,2}{0,098} = 20$  intervalles seulement !

Si la discrétisation du signal utile est trop grossière cela peut se traduire par une diminution des performances de l'asservissement, voire une instabilité. Le système n'est en effet plus linéaire et il n'existe pas d'outils théoriques permettant de prévoir les performances du système bouclé.

## **8.2 Transport de l'information : les bus et les protocoles**

Pas question de faire ni même d'ébaucher un cours sur les bus et les protocoles ici. Mais on va quand même donner quelques indications. L'automaticien a l'habitude de simplifier le problème qu'il se pose en dessinant un schéma simplifié de son problème d'asservissement (du type de la Figure 20 page 45) pour pouvoir appliquer les théories de la commande. Cependant il arrive bien souvent que l'information transite à l'intérieur du système, soit entre différents ordinateurs effectuant différentes parties du travail soit entre ordinateur et périphériques, par l'intermédiaire de bus informatique. A chaque type de bus correspond un ou plusieurs protocoles qui peuvent poser des problèmes pour l'automaticien :

- les données sont-elles discrétisées ? (le protocole peut par exemple ne permettre de transmettre que des paquets de données de 16 bits),
- le bus induit-t'il un retard ? (le composant mettant en forme les données du ordinateur pour les publier sur le bus a certaines performances qui peuvent induire un délai)
- le bus est-il déterministe ? (Certains bus ne garantissent pas que les données arriveront en un temps donné),
- le bus est-il sûr ? (Certains bus ne garantissent pas non plus que les données arriveront bien)

### **8.3 Capteurs et actionneurs "intelligents", "smart sensors"**

De plus en plus souvent les capteurs et les actionneurs sont associés à leur électronique de puissance, électronique de commande et de conditionnement de signal. L'interface avec le capteur ou l'actionneur se fait donc via un protocole de communication (par exemple via un bus RS232).

L'intérêt est de véhiculer non pas des signaux analogiques, tensions ou courants, mais des signaux numériques beaucoup moins sensibles aux éventuelles perturbations.

On retrouve donc, entre le capteur ou l'actionneur et le ordinateur, la présence d'un bus et d'un protocole de communication.

### **8.4 La philosophie "hardware in the loop"**

Les systèmes embarqués sont de plus en plus complexes et touchent des domaines de plus en plus variés : automobile, satellites, aéronautique, UAV, systèmes d'arme, sous-marins, systèmes, etc... Les systèmes à piloter ainsi que les systèmes de commande sont de plus en plus complexes.

Les techniques de simulation « Hardware In the Loop » (HIL) sont de plus en plus utilisées dans le développement et dans le test de ces systèmes embarqués. L'idée est de développer entièrement le système de contrôle sur une plateforme de test en utilisant le maximum de composants réels et en ne remplaçant que le minimum par la simulation. Par exemple en aéronautique l'« Iron Bird » regroupe dans un hangar tous les systèmes réels de l'avion en cours de développement. Seule la mécanique du vol est simulée : les sorties des actionneurs sont mesurées, rentrées dans un modèle de simulation numérique, puis les mesures effectuées normalement par les capteurs sont calculées et restituées par le modèle.

L'intérêt est de diminuer de manière importante les coûts et les délais en économisant le plus grand nombre possible de tests sur le système réel en situation. Lors des premiers tests réels les techniques « Hardware In the Loop » permettent d'avoir effectué la plus grande part du développement des lois de commande.

### **8.5 Le prototypage rapide en automatique**

Un peu dans le même ordre d'idée que le « Hardware In the Loop » le prototypage rapide permet de réaliser la conception des lois de commande en simulation puis de transporter rapidement (en un simple click !) le code du contrôleur simulé vers la cible temps réel implémentée sur le système réel.

La société Mathworks [9] investit massivement dans ce domaine en ayant enrichi son logiciel de calcul scientifique (Matlab) avec un logiciel de simulation graphique très évolué (Simulink) puis en développant des outils de prototypage rapide associés (Real Time Workshop, Embedded Target For...).

## 9. ANNEXE

### 9.1 Annexe 1 : Résolution de l'équation polynomiale de Bezout

Il s'agit de trouver R et S vérifiant :

$$P(q^{-1}) = A(q^{-1}).S(q^{-1}) + B(q^{-1}).R(q^{-1})$$

Où P, A et B sont connus.

Dans le cas où P, A, B, R et S sont des nombres entiers la solution de l'algorithme est donnée par l'algorithme de division Euclidienne étendu [8].

Dans le cas où P, A, B, R et S sont des polynômes, l'algorithme est donné ci-dessous, programmé en matlab ([1]).

```
function [Rp,Sp,nrp,nsp]=bezoutd(A,B,Hs,Hr,P)
%function [Rp,Sp,nrp,nsp]=bezoutd(A,B,Hs,Hr,P)
%solves AHsSp+BHrRp=P by coefficient comparison. Delay and discretization delay need to be integrated in B
%(zeros at the beginning).
%
%inputs:
%A=[a0 a1 ... aNa] ... vector of model denominator coefficients A=a0 + a1z^(-1) + a2z^(-2) +...+ aNaz^(-Na)
%B=[b0 b1 ... bNb] ... vector of model numerator coefficients B=b0 + b1z^(-1) + b2z^(-2) +...+ bNbz^(-Nb)
%Hs=[hs0 hs1 ... hsNhs] ... vector of controller denominator fixed part Hs=hs0 + hs1z^(-1) +...+ hsNhsz^(-Nhs)
%Hr=[hr0 hr1 ... hrNhr] ... vector of controller denominator fixed part Hr=hr0 + hr1z^(-1) +...+ hrNhrz^(-Nhr)
%P=[p0 p1 ... pNp] ... vector of desired polynomial coefficients P=p0 + p1z^(-1) + p2z^(-2) +...+ pNpz^(-Np)
%outputs:
%Rp=[rp0 rp1 rp2 ...] ... vector of coefficients for resulted controller numerator
%Sp=[sp0 sp1 sp2 ...] ... vector of coefficients for resulted controller denominator
%nrp ... order of Rp
%nsp ... order of Sp
%
%written by: J. Langer, I.D. Landau, H. Prochazka
%7th june 2002

PRECISION=1e-16;

D=size(A);
if D(1)>1, A=A'; end;
```

```

D=size(B);
if D(1)>1, B=B'; end;
D=size(Hs);
if D(1)>1, Hs=Hs'; end;
if D(1)==0, Hs=1; end;
D=size(Hr);
if D(1)>1, Hr=Hr'; end;
if D(1)==0, Hr=1;end;
D=size(P);
if D(1)>1, P=P'; end;

na=length(A)-1;
nb=length(B)-1;
np=length(P)-1;
nhs=length(Hs)-1;
nhr=length(Hr)-1;

if (nhs>0), Ah=real(convz(A,Hs)); else Ah=A*Hs; end; % Ah = A * Hs
nah=length(Ah) -1;
if (nhr>0), Bh=real(convz(B,Hr)); else Bh=B*Hr; end;
nbh=length(Bh) -1;
if (np>nah+nbh-1), disp('Bezout error: too many poles');end;

% increase size of P using zeros if necessary
if (np<nah+nbh-1),
%
% set remaining poles onto a circle with radius rmin
P=[P zeros(1,nah+nbh-1-np)];
rootsPdes=roots(P);
nextra=nah+nbh-1-np;
rmin=1e-16;
angle=[0:nextra-1]'/nextra*2*pi;
j=sqrt(-1);
rootsPextra=rmin*exp(j*angle);
P=poly([rootsPdes;rootsPextra]);
np=nah+nbh-1;

end;
P,

nsp=nbh-1;
nrp=nah-1;

%matrix is smaller than vector PD

```

```

if (np>nah+nbh-1),
    disp('Order of model denominator is too low! Add a polynom of higher order to Hs or Hr. ');
end;

% ns=nsp+nhs
% nr=nrp+nhr

M=[];
for j=1:nsp+1,
    V=[];
    if (j>1), V=[V ; zeros(j-1,1)]; end;% zeros in front of Ah
    V=[V ; Ah'];% Ah
    if (j<=nsp), V=[V ; zeros(nsp+1-j,1)]; end;% zeros behind Ah
    if (length(V)~=nah+nbh), disp('bezoutb: error V'); end;
    M=[M V]; % add one column to M
end;

for j=1:nrp+1,
    V=[];
    if (j>1), V=[V ; zeros(j-1,1)]; end;
    V=[V ; Bh'];
    if (j<=nrp), V=[V ; zeros(nrp+1-j,1)]; end;
    if (length(V)~=nah+nbh), disp('bezoutb: error V'); end;
    M=[M V];
end;

D=size(M);
if (D(1)~=nah+nbh), disp('bezoutb: error size M row'); end;
if (D(2)~=nah+nbh), disp('bezoutb: error size M column'); end;

% make P column vector
P=P';

global M1;
M1=M;

X= M\P;
% coefficients are real values
X=real(X);

% make X row vector
X=X';

```

## Commande des systèmes par ordinateur

```
Sp=X(1:nsp+1);  
Rp=X(nsp+2:nsp+nrp+2);
```

## 10. BIBLIOGRAPHIE

- [1] I. D. Landau, "*Commande des systèmes*", Edition Lavoisier 2002.
- [2] C. Nouals, "Théorie du Signal", Notes de cours ENSICA
- [3] F. Vincent, "*Traitement Numérique du Signal*", Notes de cours ENSICA.
- [4] D. Arzelier, "*Représentation et analyse des systèmes, notes de cours*", Edition ENSICA. Voir aussi <http://www.laas.fr/~arzelier/>
- [5] J. Bordeneuve-Guibé, "*Commande des systèmes linéaires*", Notes de cours ENSICA.
- [6] Y. Briere, "*Identification*", Notes de cours ENSICA.
- [7] L. Ljung, "*System Identification, Theory for the User*", Prentice Hall 1987.
- [8] Wikipedia , équation de Bezout : [http://en.wikipedia.org/wiki/Bézout's\\_identity](http://en.wikipedia.org/wiki/Bézout's_identity)
- [9] <http://www.mathworks.com>





# 11. INDEX

<b>A</b>	
actionneur intelligent .....	21
amortissement .....	15
ANSI C .....	60
Approximation numérique d'équation différentielle .....	28
ARMAX .....	35
<b>B</b>	
BJ35 .....	
Blocage d'un signal .....	56
bloqueurs d'ordre 0 .....	22
BOZ .....	22
bus.....	21, 61
<b>C</b>	
CAN.....	21
capteur intelligent .....	21
causal .....	6, 9
Causalité .....	10
CNA.....	21
compilateur .....	60
convertisseur analogique / numérique.....	21
convertisseur numérique / analogique.....	21
convertisseurs analogique-numérique .....	61
coprocesseur mathématique .....	60
Correspondances pôles - zéros.....	31
<b>D</b>	
Diagramme de Bode .....	18
Discrétisation de fonction de transfert continues .....	27
division polynomiale.....	6
<b>E</b>	
échelon unitaire.....	11
équation d'état .....	17
Erreur statique.....	42
Estimation de la réponse fréquentielle .....	39
Estimation de la réponse impulsionnelle.....	38
Euler .....	29
<b>F</b>	
facteur d'échelle .....	61
facteur d'oubli .....	36
<i>fenêtre de Hamming</i> .....	39
filtre anti-repliement .....	25
fonction de sensibilité .....	42
fonction de transfert discrète.....	9
<b>G</b>	
gain statique.....	11
gain statique unitaire.....	42
<b>I</b>	
Identification.....	32
Identification non paramétrique .....	38
intercorrélation.....	38
<b>L</b>	
lieu des racines.....	43
lieu d'Evans.....	43
LTI.....	8
<b>M</b>	
marge de gain.....	47
marge de module.....	46
marge de phase .....	47
méthode des moindres carrés récursifs .....	32
Méthode des trapèzes.....	29
modèle ARX .....	32
<b>O</b>	
OE.....	35
one step ahead prediction.....	35
opérateur de retard .....	9
<b>P</b>	
période d'échantillonnage (choix) .....	26
Placement de pôles .....	54
Pôles .....	11
Poursuite .....	56
<i>prédiction</i> .....	32
protocoles.....	61
Pseudo Random Binary Signal .....	34
pulsation .....	15
<b>Q</b>	
quantification .....	23, 24, 60
<b>R</b>	
R,S,T.....	41
rapidité.....	15
rayon de convergence .....	5, 9
Régulation.....	55

Rejet de perturbations .....	42
Rejet d'une perturbation harmonique .....	56
réponse à un échelon.....	11
réponse fréquentielle.....	18
Réponse impulsionnelle.....	10
Réponses indicielle .....	10
retard.....	23
root locus .....	43

### **S**

signal discret .....	5
smart sensors.....	62
stabilité.....	11
Systèmes du premier ordre.....	14
Systèmes du second ordre.....	15

### **T**

Théorème de Nyquist.....	46
théorème de Shannon.....	19, 24
théorème du retard .....	6
Transformée en z .....	5
transformée unilatérale .....	5
Tustin.....	29

### **V**

virgule flottante.....	60
------------------------	----

### **Z**

zero holder hold .....	23
zéros.....	11
zoh .....	23

# Commande des systèmes par ordinateur

Yves Briere  
ENSICA

Notes de cours  
2007



# TABLE DES MATIERES

Table des matières .....	1
1. Introduction .....	1
2. Rappels et prérequis .....	3
3. Signaux et systèmes discrets .....	5
3.1 Signaux discrets .....	5
3.1.1 Transformée en z .....	5
3.1.2 Signaux types .....	7
3.2 Systèmes discrets .....	9
3.2.1 Fonction de transfert discrète, définition .....	9
3.2.2 Notation "opérateur de retard $q^{-1}$ " .....	10
3.2.3 Réponse impulsionnelle .....	11
3.2.4 Causalité .....	11
3.2.5 Réponses indicielle .....	11
3.2.6 Régime stationnaire (gain statique) .....	12
3.2.7 Pôles, zéros, stabilité .....	12
3.2.8 Systèmes du premier ordre .....	15
3.2.9 Systèmes du second ordre .....	16
3.2.10 Calcul itératif de la réponse d'un système .....	18
3.2.11 Représentation par équation d'état .....	18
3.3 Réponse fréquentielle des systèmes échantillonnés .....	19
3.3.1 Diagramme de Bode d'un système échantillonné .....	19
3.3.2 Où l'on voit apparaître le théorème de Shannon .....	20
4. Systèmes continus échantillonnés .....	23
4.1 Structure des systèmes commandés par ordinateur .....	23
4.2 Échantillonnage des systèmes continus .....	24
4.3 Effets de l'échantillonnage .....	25
4.4 Où l'on revoit le théorème de Shannon .....	26
4.5 Discrétisation de fonction de transfert continues .....	29
4.5.1 Approximation numérique d'équation différentielle .....	30
4.5.2 Superposition des réponses indicielles .....	32
4.5.3 Correspondances pôles - zéros .....	33
4.5.4 Bilan .....	33
5. Identification des systèmes échantillonnés .....	35
5.1 Identification d'un modèle ARX par la méthode des moindres carrés récurrents .....	35
5.2 L'identification paramétrique en général et en pratique .....	37

5.2.1	Choix du signal d'excitation, traitement des mesures .....	37
5.2.2	Choix d'une structure de modèle.....	38
5.2.3	Choix d'un critère à minimiser.....	38
5.2.4	Choix d'un algorithme.....	38
5.2.5	Validation.....	39
5.2.6	Utilisation de logiciels .....	39
5.3	Identification "simple" .....	39
5.4	Identification non paramétrique .....	41
5.4.1	Estimation de la réponse impulsionnelle .....	41
5.4.2	Estimation de la réponse fréquentielle.....	42
6.	Systèmes bouclés .....	45
6.1	Fonction de transfert en boucle fermée .....	45
6.2	Erreur statique .....	46
6.3	Rejet de perturbations.....	46
6.4	Stabilité des systèmes à simple gain de rétroaction.....	47
6.5	Analyse des systèmes bouclés dans le domaine fréquentiel : stabilité.....	49
6.5.1	Théorème de Nyquist dans le cas où le système est stable en boucle ouverte (critère du revers) .....	50
6.5.2	Théorème de Nyquist dans le cas où le système est instable en boucle ouverte .....	50
6.5.3	Interprétations du lieu de Nyquist.....	50
7.	Méthodes de régulation des systèmes échantillonnés .....	53
7.1	Introduction .....	53
7.2	Structure d'un système commandé par ordinateur, hypothèses.....	54
7.2.1	Discretisation, choix de la période d'échantillonnage.....	54
7.2.2	Discretisation d'un correcteur continu .....	55
7.3	Régulateur PID numérique .....	55
7.3.1	Cas 1 .....	55
7.3.2	Cas 2 .....	58
7.4	Placement de pôles .....	59
7.4.1	Choix des pôles en boucle fermée .....	60
7.4.2	Régulation : calcul de R et S.....	60
7.4.3	Poursuite : calcul de T .....	61
7.5	Poursuite et régulation à objectifs indépendants .....	62
8.	Technologie de la commande par ordinateur.....	65
8.1	Calculateurs et interfaces numériques / analogiques.....	65
8.2	Transport de l'information : les bus et les protocoles .....	67
8.3	Capteurs et actionneurs "intelligents", "smart sensors" .....	68
8.4	La philosophie "hardware in the loop" .....	68
8.5	Le prototypage rapide en automatique .....	68
9.	Annexe .....	69
9.1	Annexe 1 : Résolution de l'équation polynomiale de Bezout.....	69
10.	Bibliographie.....	73
11.	Index.....	1







# 1. INTRODUCTION



Figure 1 : commande d'un systèmes de télémédecine par calculateur, vu par Claude Serre



## 2. RAPPELS ET PREREQUIS

Ce polycopié ne comporte pas de rappels sur les systèmes continus. S'adressant aux étudiants en deuxième année de l'ENSICA il suppose connu les enseignements de première année et de seconde année en automatique et traitement du signal.

En première année le cours de théorie du signal [2] a rendu familier les notions de :

- transformée de Fourier, transformée de Laplace
- corrélation, convolution

Le cours de traitement numérique du signal [3] a abordé en plus :

- la discrétisation et la quantification des signaux.
- la transformée en z et la transformée de Fourier discrète.
- les filtres numériques RIF et RII.

En automatique [4] les bases de la représentation et de l'analyse des systèmes continus ont été posées. On ne reviendra pas dans ce cours sur les notions de :

- modèles mathématique des systèmes linéaires à temps continus à temps invariant (LTI) : fonction de transfert, équation d'état, pôles, zéros et dynamique.
- commandabilité et observabilité.
- stabilité des systèmes en boucle ouverte : pôles, critères de Routh-Hurwitz.
- stabilité des systèmes bouclés : critère de Nyquist, critère de Bode.
- lieu des racines.

En deuxième année les notions suivantes sont acquises [5] :

- synthèse des correcteurs par méthodes fréquentielles (avance et retard de phase, PID)
- commande par retour d'état (placement de pôles), commande Linéaire Quadratique (LQG)
- reconstruction de l'état par Observateur, observateur optimal de Kalman.



## 3. SIGNAUX ET SYSTEMES DISCRETS

### 3.1 Signaux discrets

#### 3.1.1 Transformée en z

Pour une approche mathématiquement rigoureuse de représentation des signaux discrets par la transformée en z consulter [3] et sa bibliographie. Dans ce cours on se cantonne à une approche utilitariste des outils et méthodes mathématiques tout en essayant de ne pas contredire fondamentalement la rigueur mathématique.

Un signal discret  $s(k)$  est une séquence de nombres réels appelés échantillons. L'indice  $k$  est généralement associé à un instant d'échantillonnage :  $t = k.T_e$ . La période d'échantillonnage  $T_e$  est supposée constante. Par définition la transformée en z du signal  $s(k)$  est la fonction de la variable  $z$ , notée abusivement  $s(z)$  :

$$s(z) = Z(s(k)) = \sum_{k=0}^{\infty} s(k) \cdot z^{-k} \quad (1)$$

La transformée définie ici est plus précisément appelée transformée unilatérale puisque les termes correspondant à  $k < 0$  ne sont pas pris en compte. Ceci n'a pas d'importance puisque les signaux que nous utilisons sont systématiquement causaux<sup>1</sup> :  $s(k) = 0$  pour  $k < 0$ .

Il y a bien évidemment un lien avec la transformée de Laplace et la transformée de Fourier (voir [3])

La fonction de la variable complexe  $s(z)$  est une série et ne converge pas forcément pour toute valeur de  $z$ . En toute rigueur il faudrait associer le rayon de convergence à chaque fonction en  $z$  :  $\{s(z) \text{ converge pour } |z| > \alpha\}$ .

Les automaticiens pour qui la transformée en z n'est bien souvent qu'une façon bien commode de regrouper des termes de récurrence utilisent souvent la pseudo-variable<sup>2</sup>  $q$ .

A partir d'une transformée en z  $s(z)$  il est possible de retrouver la séquence  $s(k)$  correspondante en appliquant la transformation inverse, dont on ne donnera pas la formule ici. A la place de faire appel à cette formule compliquée il est possible d'utiliser d'autres méthodes pour retrouver la séquence  $s(k)$  lorsque la fonction  $s(z)$  est donnée sous forme de fraction rationnelle :

<sup>1</sup> Pour une justification du terme « causal » voir page 11

<sup>2</sup> Voir page 10

- division polynomiale
- décomposition de la fraction rationnelle en éléments simples puis utilisation de transformations élémentaires telles que celles données dans le Tableau 2

La propriété fondamentale de la transformée en z est qu'elle est linéaire (la transformée en z d'une composition linéaire de signaux est la composition linéaire de la transformée en z de chaque signal). Les autres propriétés fondamentales sont données dans le Tableau 2.

Tableau 1 : propriétés de la transformée en z

1	$s(z) = Z(s(k)) = \sum_{k=0}^{\infty} s(k) \cdot z^{-k}$
2	$Z(\alpha \cdot s_1(k) + \beta \cdot s_2(k)) = \alpha \cdot Z(s_1(k)) + \beta \cdot Z(s_2(k))$
3	$Z(k \cdot s(k)) = -z \cdot \frac{ds(z)}{dz}$
4	$Z(c^k \cdot s(k)) = s\left(\frac{z}{c}\right)$
5	$Z(s(k-1)) = s(k=-1) + z^{-1} \cdot s(z)$
6	$Z(s(k+1)) = z \cdot s(z) - z \cdot s(k=0)$
7	$s(k=0) = \lim_{z \rightarrow \infty} (s(z))$
8	$s(k \rightarrow \infty) = \lim_{z \rightarrow 1} \left( \frac{z-1}{z} \cdot s(z) \right)$

Remarque 1 : la propriété 5 ne prend pas en compte le fait que le signal est causal : en pratique  $s(k=-1) = 0$ .

Remarque 2 : la propriété 5 justifie le fait que le terme  $z^{-1}$  soit appelé "opérateur de retard unitaire", cette propriété est aussi appelée "théorème du retard".

Remarque 3 : les propriétés 7 et 8 sont données sous réserve de convergence.

Transformées inverses

### Exemple 1 : transformée inverse par division polynomiale

On considère un signal  $s(k)$  dont la transformée en z est donnée par la fraction rationnelle :

$$S(z) = \frac{0.1 \cdot z^2 + 0.1 \cdot z}{z^3 - 2.6 \cdot z^2 + 2.2 \cdot z - 0.6} \quad (3.1)$$

L'objectif est d'obtenir  $S(z)$  sous forme polynomiale :

$$S(z) = x_0 + x_1 \cdot z^{-1} + x_2 \cdot z^{-2} + \dots \text{etc.} \quad (3.2)$$

Cette suite est généralement infinie, la réponse temporelle est donnée par la succession des termes  $\{x_i\}$ . En traitant l'équation (3.1) comme une simple division l'algorithme de division classique (celui appris en CM2) permet d'obtenir par récurrence tous les termes de la suite (3.2) :

$0.1 \cdot z^2$	$+ 0.1 \cdot z$			$z^3 - 2.6 \cdot z^2 + 2.2 \cdot z - 0.6$
$0.1 \cdot z^2$	$- 0.26 \cdot z$	$+ 0.22$	$- 0.06 \cdot z^{-1}$	$0.1 \cdot z^{-1} + 0.36 \cdot z^{-2} + 0.716 \cdot z^{-3}$
0	$0.36 \cdot z$	$- 0.22$	$0.06 \cdot z^{-1}$	
	$0.36 \cdot z$	$- 0.936$	$0.792 \cdot z^{-1}$	$- 0.216 \cdot z^{-2}$
0	$0.716$	$- 0.732 \cdot z^{-1}$	$\dots$	$\dots$

Les éléments successifs de la suite  $s(k)$  sont donc  $s(k) = \{0, 0.1, 0.36, 0.716, \dots\}$ . Pour obtenir les termes suivants il faudrait continuer la division.

**Exemple 2 : transformée inverse par décomposition en éléments simples**

La deuxième méthode présentée consiste à décomposer la fraction rationnelle (3.1) en éléments simples puis d'utiliser les tables de transformées en  $z$  (Tableau 2). Pour notre exemple cela donne :

$$S(z) = \frac{0.1 \cdot z^2 + 0.1 \cdot z}{z^3 - 2.6 \cdot z^2 + 2.2 \cdot z - 0.6} = \frac{-z}{z-1} + \frac{0.5 \cdot z}{(z-1)^2} + \frac{z}{z-0.6}$$

La réponse temporelle est donc composée des trois éléments :

$$s(k) = -(1 \ 1 \ 1 \ \dots) + 0.5 \cdot (1 \ 2 \ 3 \ \dots) + (1 \ 0.6 \ 0.6^2 \ \dots)$$

$$\text{soit : } s(k) = \begin{cases} s(0) = -1 + 0.5 + 1 \\ s(2) = -1 + 0.5 \cdot 2 + 0.6 \cdot 0.5 \\ s(3) = -1 + 0.5 \cdot 3 + 0.6^2 \\ s(4) = \dots \end{cases}$$

**3.1.2 Signaux types**

Le Tableau 2 donne les transformée en  $z$  de certains signaux fondamentaux. La Figure 2 représente l'allure de quelques signaux types.

Tableau 2 : transformée en z élémentaires

$s(k)$	$S(z)$
signal temporel	Transformée en z
$\delta(k)$ : impulsion unité	1
$u(k)$ : échelon unité	$\frac{z}{z-1}$
$k \cdot u(k)$	$\frac{z}{(z-1)^2}$
$c^k \cdot u(k)$	$\frac{z}{z-c}$
$k \cdot a^k \cdot u(k)$	$\frac{a \cdot z}{(z-a)^2}$
$\sin(\omega \cdot k) \cdot u(k)$	$\frac{z \cdot \sin(\omega)}{z^2 - 2 \cdot z \cdot \sin(\omega) + 1}$
$\cos(\omega \cdot k) \cdot u(k)$	$\frac{z \cdot (z - \cos(\omega))}{z^2 - 2 \cdot z \cdot \sin(\omega) + 1}$
$a^k \cdot \sin(\omega \cdot k) \cdot u(k)$	$\frac{a \cdot z \cdot \sin(\omega)}{z^2 - 2 \cdot a \cdot \cos(\omega) \cdot z + a^2}$
$a^k \cdot \cos(\omega \cdot k) \cdot u(k)$	$\frac{z \cdot (z - a \cdot \cos(\omega))}{z^2 - 2 \cdot a \cdot \cos(\omega) \cdot z + a^2}$



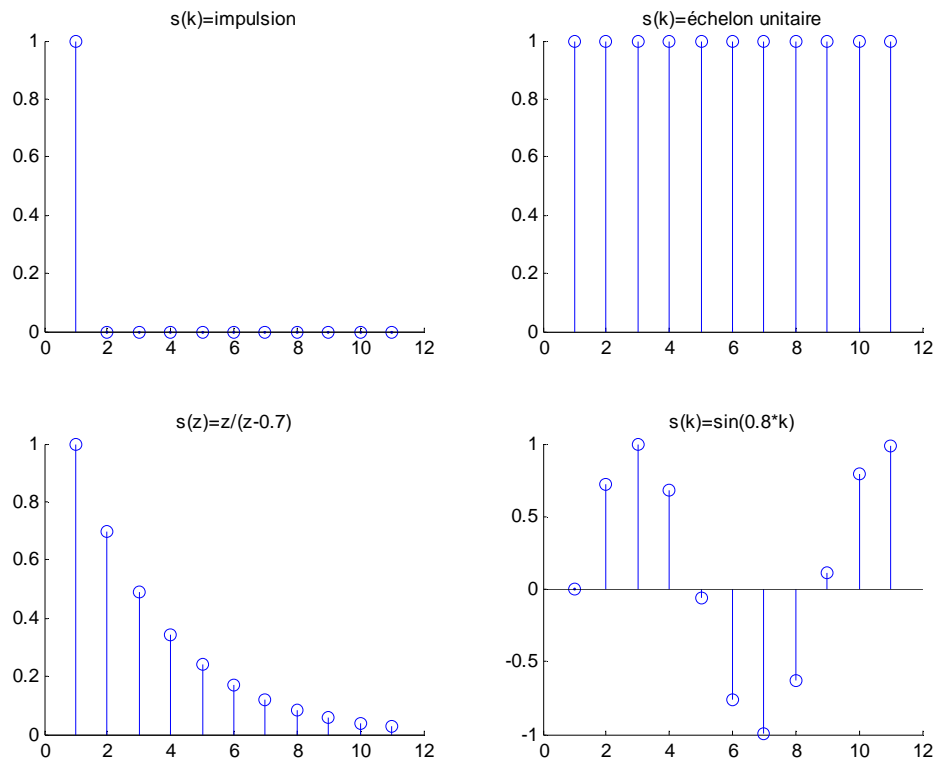


Figure 2 : réponses de quelques signaux types

### 3.2 Systèmes discrets

#### 3.2.1 Fonction de transfert discrète, définition

On considère maintenant un système discret linéaire et invariant dans le temps (LTI) représenté sur la Figure 3.

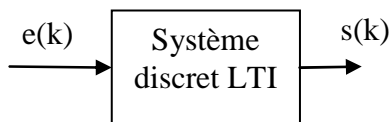


Figure 3 : système discret, notations

La relation entre la séquence d'entrée et la séquence de sortie est linéaire et définie par un ensemble de coefficients constants  $a_{1:n}$  et  $b_{1:m}$  :

$$\begin{aligned} s(k) + a_1 \cdot s(k-1) + a_2 \cdot s(k-2) + \dots + a_n \cdot s(k-n) = \\ b_0 \cdot e(k) + b_1 \cdot e(k-1) + b_2 \cdot e(k-2) + \dots + b_m \cdot e(k-m) \end{aligned} \quad (2)$$

On peut appliquer la transformée en  $z$  à chaque membre de l'équation puis en appliquant en récurrence la propriété 5 (théorème du retard) on obtient la fonction de transfert discrète du système :

$$\frac{s(z)}{e(z)} = F(z) = \frac{b_0 + b_1 \cdot z^{-1} + b_2 \cdot z^{-2} + \dots + b_m \cdot z^{-m}}{1 + a_1 \cdot z^{-1} + a_2 \cdot z^{-2} + \dots + a_n \cdot z^{-n}} \quad (3)$$

Remarque 1 : la fonction de transfert précédente est normalisée en prenant  $a_0 = 1$ .

Remarque 2 : on remarquera que la fonction de transfert est donnée en  $z^{-1}$  plutôt qu'en  $z$ . Cela est justifié par le fait que celle-ci traduit l'équation d'évolution du système physique causal (2).

Le script Matlab permettant de créer un objet Matlab correspondant à une fonction de transfert discrète est donné ci-dessous.

#### Matlab

```
% Création d'un système linéaire discret :
Te=0.1;
sysd = tf([b0 b1 b2],[1 a0 a2 a3],Te,'variable','z^-1');
```

On notera l'utilisation de l'argument 'variable', 'z^-1'.

### 3.2.2 Notation "opérateur de retard $q^{-1}$ "

La fonction de transfert est une fonction de la variable complexe  $z$ . Il faudrait donc en toute rigueur l'associer à son rayon de convergence. En pratique on ne s'embarrasse jamais de préciser cette zone de convergence. C'est pourquoi on trouve souvent la notation utilisant l'opérateur  $q^{-1}$  de décalage à la place de la variable complexe  $z^{-1}$  (4).  $F(q)$  n'est alors plus à proprement parler une fonction mais on lui appliquera les mêmes manipulations algébriques qu'à  $F(z)$ .

$$F(q) = \frac{b_0 + b_1 \cdot q^{-1} + b_2 \cdot q^{-2} + \dots + b_m \cdot q^{-m}}{1 + a_1 \cdot q^{-1} + a_2 \cdot q^{-2} + \dots + a_n \cdot q^{-n}} \quad (4)$$

Attention : selon les auteurs l'opérateur qui remplace  $z^{-1}$  est noté  $q$  (Matlab) ou  $q^{-1}$  (Landau [1]). Dans la suite du document c'est cette dernière convention qui sera appliquée.

L'utilisation de l'opérateur de retard  $q^{-1}$  à la place de la variable complexe  $z^{-1}$  permet de traiter du cas des fonctions de transfert à coefficients non constant, cas non prévu par la transformation en  $z$ .

### 3.2.3 Réponse impulsionnelle

Essayons d'appliquer au système un signal d'entrée particulier : une impulsion unitaire. La transformée en  $z$  du système est obtenue en multipliant la fonction de transfert par la transformée en  $z$  de l'impulsion unitaire : 1. On obtient :

$$s(z) = \frac{b_0 + b_1 \cdot z^{-1} + b_2 \cdot z^{-2} + \dots + b_m \cdot z^{-m}}{1 + a_1 \cdot z^{-1} + a_2 \cdot z^{-2} + \dots + a_n \cdot z^{-n}} \times 1 \quad (5)$$

Cela justifie le fait que la fonction de transfert du système peut aussi s'appeler "réponse impulsionnelle".

#### Matlab

```
% Réponse impulsionnelle d'un système :
[y,t] =impulse(sysd);
```

### 3.2.4 Causalité

Un système est dit causal si les sorties ne dépendent que du passé et non pas du futur. La conséquence est que la réponse impulsionnelle de  $F(z)$  ne peut pas comporter de terme non nuls pour  $k < 0$ . Cela justifie la confusion sémantique entre "système causal" et "signal causal", c'est-à-dire ne comportant pas de termes non nuls pour  $k < 0$ .

Attention : à la différence des systèmes continus un système physique (causal) discret peut très bien être modélisé avec un degré  $m$  supérieur au degré  $n$  : le système peut avoir plus de zéros que de pôles.

La transformée en  $z$  représentant des signaux échantillonnés elle peut tout aussi bien représenter les mesures prises à différents endroits d'une poutre. Le temps étant remplacé ici par une abscisse la notion de causalité ne s'applique pas ! Attention : beaucoup des propriétés de systèmes échantillonnés que l'on étudie sont liés à la causalité, prudence donc si l'on applique ces méthodes à d'autres domaines !

### 3.2.5 Réponses indicielle

Pour caractériser des systèmes on s'intéresse souvent à leur réponse impulsionnelle ou, plus souvent, à leur réponse indicielle. Pour calculer la réponse

à un échelon d'un système il faut donc multiplier la fonction de transfert par la transformée en z d'un échelon unitaire :

$$s(z) = \frac{b_0 + b_1 \cdot z^{-1} + b_2 \cdot z^{-2} + \dots + b_m \cdot z^{-m}}{1 + a_1 \cdot z^{-1} + a_2 \cdot z^{-2} + \dots + a_n \cdot z^{-n}} \times \frac{1}{1 - z^{-1}} \quad (6)$$

Il faudrait ensuite calculer la transformée en z inverse de s(z). En pratique, si l'on doit faire le calcul à la main on décompose en éléments simples puis on utilise des tables. Avec Matlab c'est plus simple. Voici par exemple le script pour obtenir les 20 premiers échantillons de la réponse indicielle d'un système :

**Matlab**

```
% Réponse indicielle d'un système :
[y,t] =step(sysd,20);
```

**3.2.6 Régime stationnaire (gain statique)**

Le gain statique est obtenu en appliquant une entrée indicielle et en calculant la valeur du régime permanent. En appliquant le théorème de la valeur finale il vient :

$$F_0 = \lim_{z \rightarrow 1} \left( \frac{z-1}{z} \cdot \frac{b_0 + b_1 \cdot z^{-1} + b_2 \cdot z^{-2} + \dots + b_m \cdot z^{-m}}{1 + a_1 \cdot z^{-1} + a_2 \cdot z^{-2} + \dots + a_n \cdot z^{-n}} \times \frac{1}{1 - z^{-1}} \right) \quad (7)$$

Soit :

$$F_0 = \frac{\sum_{i=0}^{i=m} b_i}{1 + \sum_{i=1}^{i=n} a_i} \quad (8)$$

Le gain statique est obtenu donc simplement en faisant le quotient de la somme des coefficients.

**3.2.7 Pôles, zéros, stabilité**

Comme pour les systèmes continus la dynamique et la stabilité des systèmes dépend des propriétés du dénominateur de la fonction de transfert.

$$F(z) = \frac{e_0 \cdot (1 + e_1 \cdot z^{-1}) \cdot (1 + e_2 \cdot z^{-1}) \cdots (1 + e_m \cdot z^{-1})}{(1 + c_1 \cdot z^{-1}) \cdot (1 + c_2 \cdot z^{-1}) \cdots (1 + c_n \cdot z^{-1})} \quad (9)$$

Les coefficients  $c_i$  sont les pôles de la fonction de transfert, les coefficients  $e_i$  sont les zéros. La fonction de transfert étant au départ composée de coefficients purement réels les coefficients  $c_i$  sont soit réels soit imaginaires par paires conjuguées.

Une fonction de transfert est donc aussi bien définie par la donnée de ses pôles et de ses zéros. Le script Matlab pour créer un système ainsi est donné ci-dessous :

#### Matlab

```
% Système discret défini par ses pôles et zéros :
Te=0.1;
sysd=zpk([c1 c2 c3],[e1 e2],e0,Te);
```

Si l'on décompose la fonction de transfert en éléments simples on obtient une somme d'éléments du type :

$$F(z) = \frac{d_1}{1 + c_1 \cdot z^{-1}} + \frac{d_2}{1 + c_2 \cdot z^{-1}} + \cdots + \frac{d_n}{1 + c_n \cdot z^{-1}} \quad (10)$$

Un système linéaire temps invariant est dit stable si sa réponse impulsionnelle tend asymptotiquement vers 0. Compte tenu de (10) cela nécessite que chaque pôle  $c_i$  soit de module inférieur à 1.

**Propriété** : un système discret LTI est stable si et seulement si tous ses pôles ont leur modules inférieur ou égal à 1.

#### Critère algébrique attestant de la stabilité : critère de Routh

On rappelle que le critère de Routh permet de prédire si les racines d'un polynôme sont toutes à partie réelle négative, donc stables. Énonçons le théorème de Routh dans le cas d'un polynôme  $P'(w)$  :

$$P'(w) = a_0 + a_1 \cdot w + a_2 \cdot w^2 + \cdots + a_n \cdot w^n \quad (11)$$

Dans un premier temps on construit le tableau :

$$\begin{array}{cccccc|c}
 \beta_{n,0} & \beta_{n,1} & \beta_{n,2} & \beta_{n,q-1} & \beta_{n,q} & 0 & \\
 \beta_{n-1,0} & \beta_{n-1,1} & \beta_{n-1,2} & \beta_{n-1,q-1} & \beta_{n-1,q-1} & 0 & \\
 \beta_{n-2,0} & \beta_{n-2,1} & \beta_{n-2,2} & \beta_{n-2,q-1} & \beta_{n-2,q-1} & & \\
 & & & & & & \\
 \beta_{2,0} & \beta_{2,1} & 0 & & & & \\
 \beta_{1,0} & 0 & & & & & \\
 \beta_{0,0} & 0 & & & & & 
 \end{array} \quad (12)$$

Puis on vérifie que tous les premiers éléments de la première colonne ont le même signe : cela garantit que tous les pôles de  $P'(w)$  ont leur partie réelle négative.

Les éléments du tableau sont construits par récurrence ligne par ligne en appliquant les formules suivantes :

Pour  $i = n$  et  $i = n - 1$  :

$$\begin{cases} \beta_{n,j} = a_{n-2j} \\ \beta_{n-1,j} = a_{n-1-2j} \end{cases} \quad (13)$$

Et pour  $i < n-1$  :

$$\beta_{i,j} = \frac{\beta_{i+1,0} \cdot \beta_{i+2,j+1} - \beta_{i+1,j+1} \cdot \beta_{i+2,0}}{\beta_{i+1,0}} \quad (14)$$

Dans le cas qui nous intéresse, un système échantillonné, on cherche plutôt à vérifier que les pôles du polynôme  $P(z)$  sont de module inférieur à un. L'idée est d'effectuer un changement de variable  $z \rightarrow w$  pour pouvoir appliquer le critère de Routh.

$$P'(w) = (1-w)^n \cdot P\left(\frac{1+w}{1-w}\right) \quad (15)$$

Si l'on effectue la transformation (15) alors le fait que  $P'(w)$  ait toutes ses racines à partie réelle négative garantit que  $P(z)$  a toutes ses racines dans le cercle unité. Pour vérifier la stabilité de  $P(z)$  il suffit donc d'effectuer le changement de variable (15) puis d'appliquer le critère de Routh conventionnel (12 13 14).

### Critère algébrique attestant de la stabilité : critère de Jury

Le critère de Jury s'apparente au critère de Routh. Nous le donnons pour un polynôme de degré 2, 3 et 4 du type :

$$P(z) = a_0 + a_1 \cdot z + a_2 \cdot z^2 + \dots + a_n \cdot z^n \quad (16)$$

Les conditions de stabilité sont données ci-dessous (ces conditions sont données dans le cas  $a_n > 0$ , il suffit de multiplier tous les termes par  $-1$  si cette condition n'est pas remplie par  $P(z)$ ) :

$$\begin{aligned}
 n = 2 & \begin{cases} a_0 + a_1 + a_2 > 0 \\ a_0 - a_1 + a_2 > 0 \\ -a_0 + a_2 > 0 \end{cases} \\
 n = 3 & \begin{cases} a_0 + a_1 + a_2 + a_3 > 0 \\ -a_0 + a_1 - a_2 + a_3 > 0 \\ -|a_0| + a_3 > 0 \\ a_0 a_2 - a_1 a_3 - a_0^2 + a_3^2 > 0 \end{cases} \\
 n = 4 & \begin{cases} a_0 + a_1 + a_2 + a_3 + a_4 > 0 \\ a_0 - a_1 + a_2 - a_3 + a_4 > 0 \\ a_4^2 - a_0^2 - |a_0 a_3 - a_1 a_4| > 0 \\ (a_0 - a_4)^2 (a_0 - a_2 + a_4) + (a_1 - a_3)(a_0 a_3 - a_1 a_4) > 0 \end{cases}
 \end{aligned} \quad (17)$$

### 3.2.8 Systèmes du premier ordre

On donne dans la Figure 4 les réponses indicielles de systèmes du premier ordre ainsi que la localisation des pôles dans le plan complexe. Le cercle de rayon unité représente la zone dans laquelle les pôles doivent se situer pour assurer la stabilité. On remarque sur les deux premières figures l'influence de la place du zéro par rapport au point  $(1,0)$  pour la rapidité.

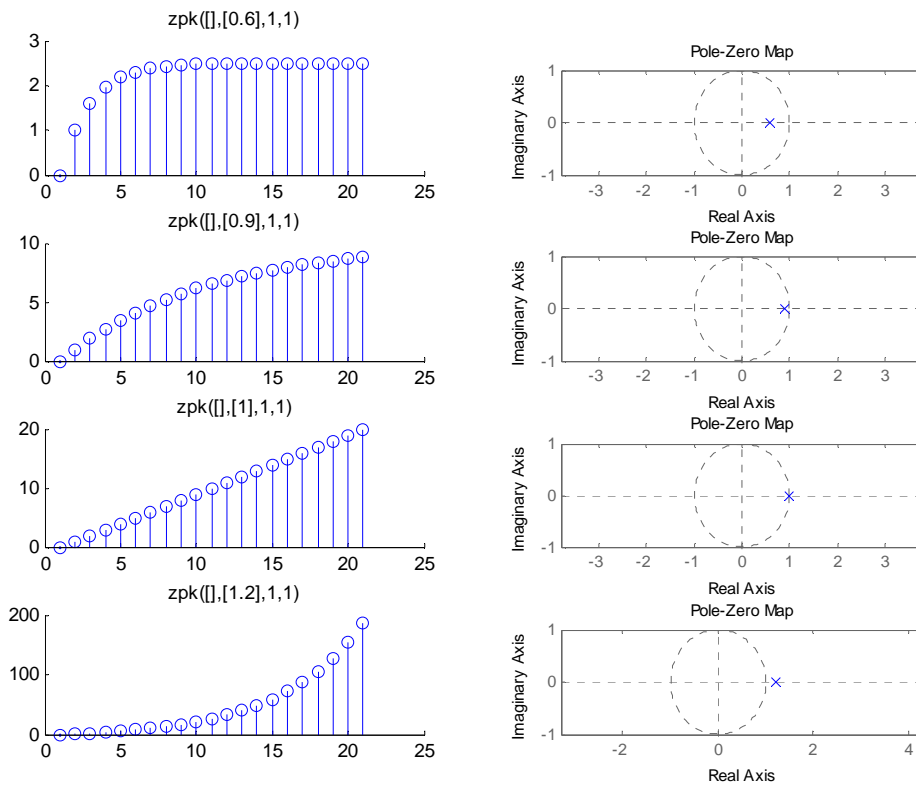


Figure 4 réponses indicielles de systèmes du premier ordre

### 3.2.9 Systèmes du second ordre

On donne dans la Figure 6 les réponses indicielles de systèmes du second ordre ainsi que la localisation des pôles dans le plan complexe. La rapidité et l'amortissement dépend de la position des pôles dans le cercle unité. La Figure 5 représente les courbes iso-amortissement et iso-pulsation des systèmes du second ordre.



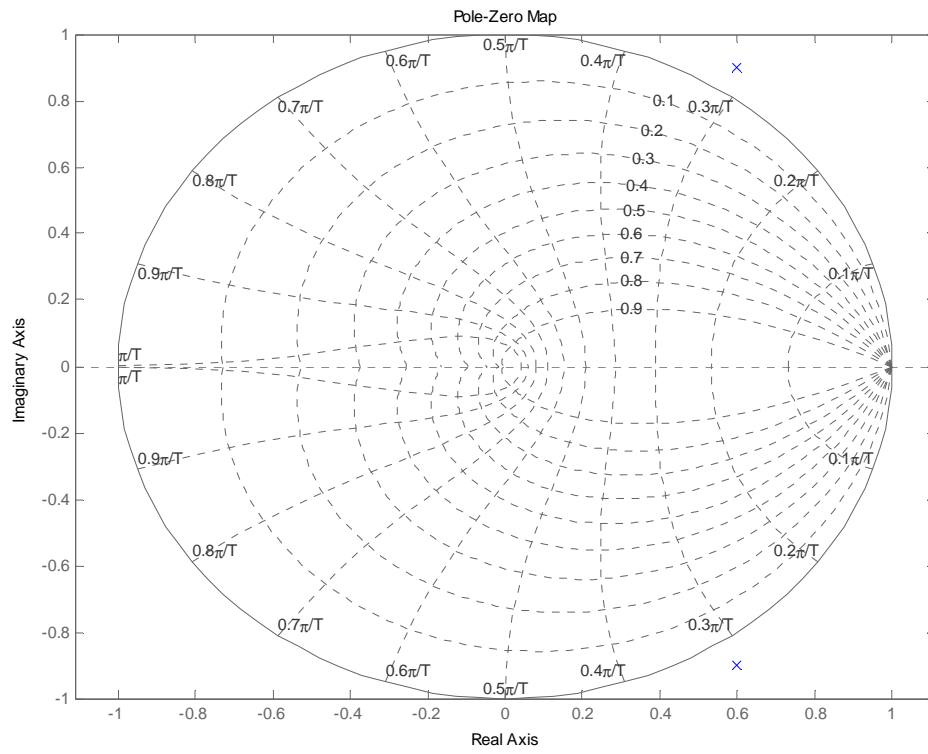


Figure 5 : courbes iso-amortissement et iso-pulsation des systèmes discrets du second ordre

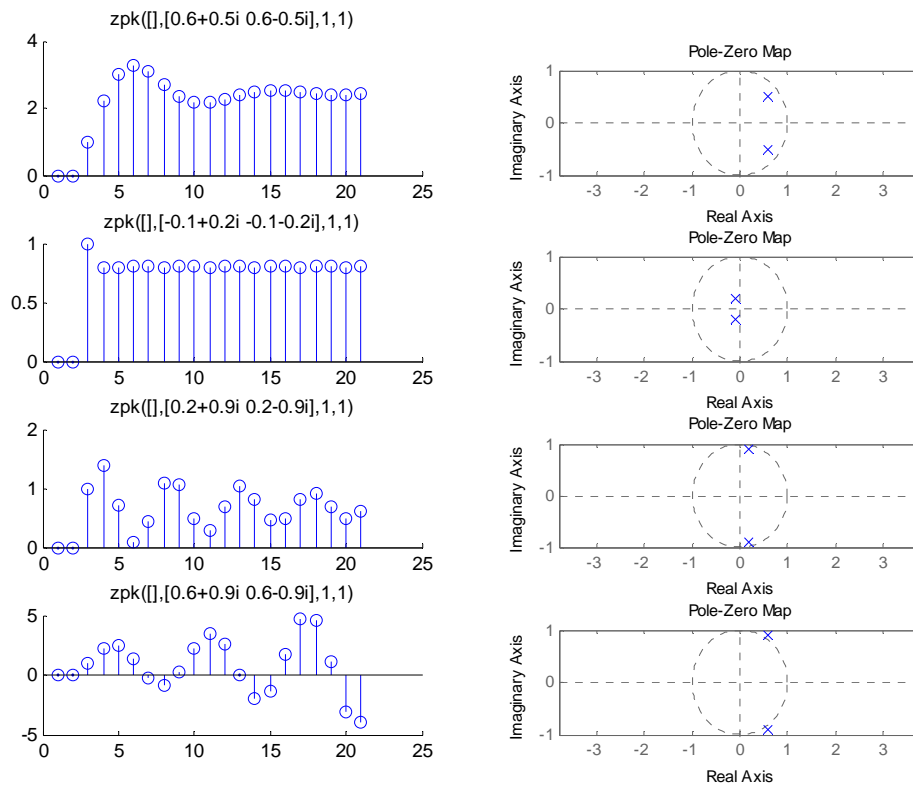


Figure 6 : réponses de systèmes du second ordre

### 3.2.10 Calcul itératif de la réponse d'un système

Si l'on dispose de la fonction de transfert d'un système et que l'on connaît les valeurs successives  $e(k)$  du signal d'entrée il est facile de calculer itérativement la sortie  $y(k)$  du système par l'équation de récurrence (2). Il faut initialiser par les  $n$  valeurs initiales de  $s(k)$ , prises généralement égales à 0.

### 3.2.11 Représentation par équation d'état

Un système discret peut aussi être représenté par un ensemble d'équations d'état comme dans le domaine continu (52). Cette représentation est naturellement adaptée aux systèmes multi-entrées multi-sorties. On définit alors un état interne  $x(k)$  ainsi qu'une fonction d'évolution  $f$  et une équation de mesure  $h$ .

$$\begin{cases} x(k+1) = f(x(k), u(k)) \\ s(k) = h(x(k), u(k)) \end{cases} \quad (18)$$

### 3.3 Réponse fréquentielle des systèmes échantillonnés

#### 3.3.1 Diagramme de Bode d'un système échantillonné

Lorsqu'on étudie la réponse fréquentielle des systèmes continus on considère des entrées particulières sinusoïdales :

$$\begin{cases} e(t) = \cos(2 \cdot \pi \cdot f \cdot t) + j \cdot \sin(2 \cdot \pi \cdot f \cdot t) = e^{j \cdot 2 \cdot \pi \cdot f \cdot t} \\ f \in [0 \quad \infty] \end{cases} \quad (19)$$

De même, pour des systèmes échantillonnés on considèrera des entrées sinusoïdales échantillonnées :

$$\begin{cases} e(k) = \cos(2 \cdot \pi \cdot f \cdot k) + j \cdot \sin(2 \cdot \pi \cdot f \cdot k) = e^{j \cdot 2 \cdot \pi \cdot f \cdot k} \\ f \in [0 \quad 1] \end{cases} \quad (20)$$

La sortie est aussi un signal échantillonné sinusoïdal de gain  $A(f)$  et de phase  $\Phi(f)$  variables et dépendant de  $f$ ; la fréquence réduite.

$$\begin{cases} s(k) = A(f) \cdot e^{j \cdot (2 \cdot \pi \cdot f \cdot k + \Phi(f))} \\ f \in [0 \quad 1] \end{cases} \quad (21)$$

Le nombre complexe, noté abusivement  $F(f)$ , d'amplitude  $A(f)$  et de phase  $\Phi(f)$  est obtenu à partir de  $F(z)$  en remplaçant  $z$  par  $e^{j \cdot 2 \cdot \pi \cdot f}$  :

$$\begin{cases} F(f) = F(z = e^{j \cdot 2 \cdot \pi \cdot f}) \\ f \in [0 \quad 1] \end{cases} \quad (22)$$

Le diagramme représentant le gain et la phase en fonction de la fréquence réduite s'appelle aussi un diagramme de Bode. L'échelle des fréquences est généralement linéaire de 0 à 0.5 (puisque le diagramme est nécessairement symétrique). La Figure 7 représente par exemple le diagramme de Bode d'un système du second ordre.

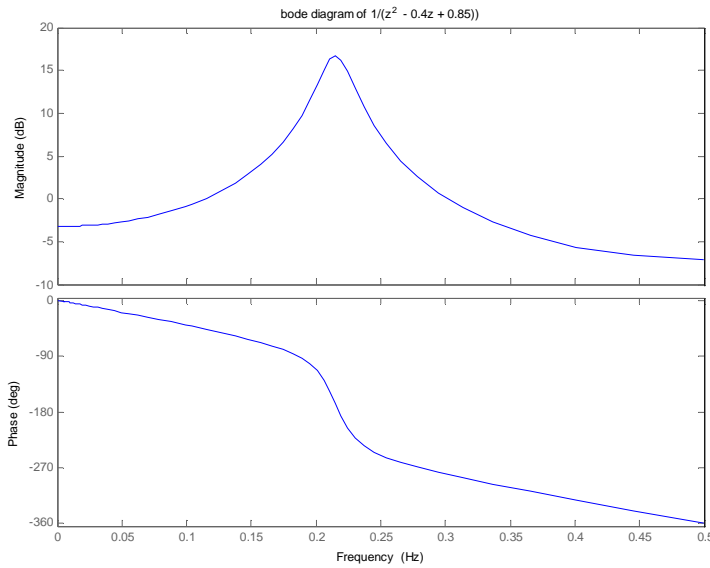


Figure 7 : exemple du diagramme de Bode d'un système discret du second ordre

### 3.3.2 Où l'on voit apparaître le théorème de Shannon

Le changement de variable  $z = e^{j2\pi f}$  n'est pas bijectif puisque  $e^{j2\pi f} = e^{j2\pi(f+k)}$  avec  $k$  entier. Si l'on fait le lien entre domaine continu et domaine fréquentiel on peut remarquer que le changement de variable  $z = e^{j2\pi f}$  fait correspondre une zone du plan complexe continu à une zone du plan complexe discret. La transformation est bijective si l'on se contente de la zone du plan complexe continu contenu entre les droites d'ordonnée 0 et 1.

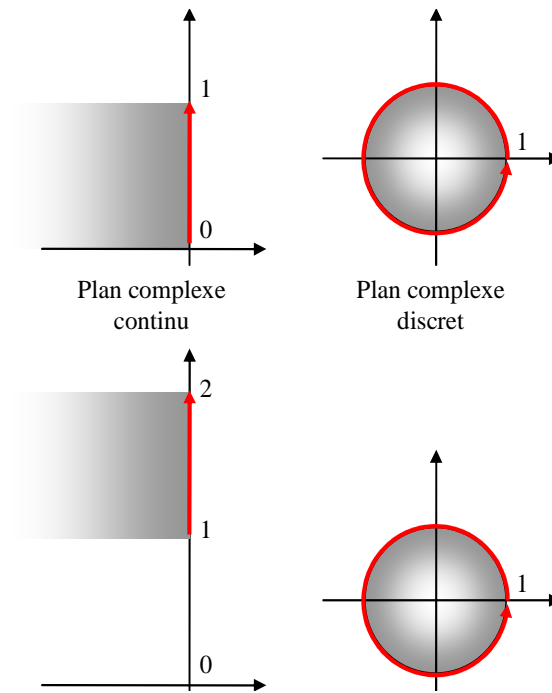


Figure 8 : correspondance entre plan complexe continu et plan complexe discret.



## 4. SYSTEMES CONTINUS ECHANTILLONNES

### 4.1 Structure des systèmes commandés par ordinateur

Les systèmes commandés par ordinateurs sont presque toujours des systèmes physiques continus par nature. Le ordinateur traite par contre les entrées et les sorties de manière discrète avec une période d'échantillonnage  $T_e$ . Les éléments importants, du point de vue de l'automaticien sont donc :

- le convertisseur analogique / numérique CAN ou du moins le système qui fournit séquentiellement les mesures au ordinateur à la période  $T_e$ .
- le ordinateur lui-même. Il doit effectuer l'acquisition des mesures, le traitement des informations, l'envoi des consignes dans un temps inférieur à  $T_e$ .
- le convertisseur numérique / analogique CNA.
- le système physique.

On suppose, pour faire simple et pour pouvoir utiliser la panoplie des outils mathématiques, que les systèmes sont échantillonnés à la même période  $T_e$  et au même moment.

La Figure 9 montre le système physique commandé par le ordinateur, dans sa version simpliste.

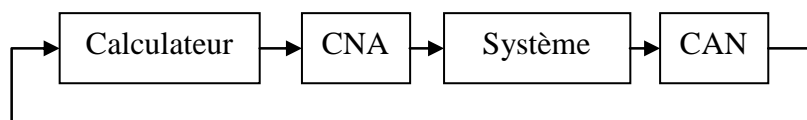


Figure 9 : système commandé par ordinateur

Au XXIème siècle les choses sont quand même parfois un petit peu plus compliquées :

- le ordinateur fournit bien souvent un signal de commande numérique (transporté par bus informatique USB, RS232, CAN, Ethernet, etc...) directement interprété par l'actionneur. On parle "d'actionneur intelligent".
- de même le "capteur intelligent" fournit aussi bien souvent la mesure sous forme d'un signal numérique.
- Les signaux sont transportés par des bus informatiques qui n'ont pas que ça à faire : il peut y avoir du retard.
- de même les capteurs et actionneurs sont rarement tous seuls, ce qui se traduit par des périodes d'échantillonnage différentes, et parfois non synchrones...

- enfin le ordinateur peut aussi avoir d'autres choses à faire...

Ceci étant dit nous allons pour le moment nous restreindre au cas représenté sur la Figure 9 avec des instants d'échantillonnage synchrones et régulièrement espacés d'un temps  $T_e$ .

## 4.2 Echantillonnage des systèmes continus

On dispose généralement de la fonction de transfert continue (notée en utilisant la variable complexe  $s$  ou  $p$ )  $F(s)$  du système continu. Le problème est d'obtenir la fonction de transfert discrète du système échantillonné de manière à pouvoir synthétiser un correcteur. Le système continu est associé au convertisseur numérique – analogique qu'il faut prendre en compte.

Les convertisseurs numérique analogiques peuvent généralement se représenter par des "échantillonneurs – bloqueurs d'ordre 0" ou BOZ. Cela signifie que la commande est maintenue à une valeur constante pendant toute la durée du temps d'échantillonnage.

Si l'on applique une commande discrète impulsionnelle à un BOZ il va maintenir la commande 1 pendant la durée  $T_e$  puis revenir à zéro (Figure 10).

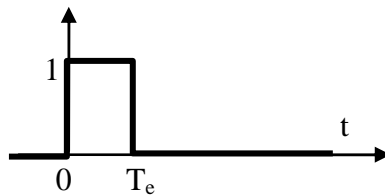


Figure 10 : réponse impulsionnelle d'un bloqueur d'ordre 0

La réponse impulsionnelle  $BOZ(t)$  est donc composée d'un échelon unitaire suivi d'un échelon unitaire de signe opposé et décalé d'un temps  $T_e$ . La transformée de Laplace de la réponse impulsionnelle du bloqueur est donnée par la formule (23) :

$$BOZ(s) = \frac{1}{s} - \frac{e^{-jT_e}}{s} \quad (23)$$

Du point de vue du ordinateur il faut donc prendre en compte le système continu de fonction de transfert  $F_{BOZ}(s)$  que l'on devra échantillonner :

$$F_{BOZ}(s) = \frac{1 - e^{-jT_e}}{s} \cdot F(s) \quad (24)$$



$$F_{\text{BOZ}}(z) = Z^{-1} \left( \frac{1 - e^{-jT_e}}{s} \cdot F(s) \right) \quad (25)$$

En pratique l'obtention de la transformée en z d'un système échantillonné par bloqueur d'ordre 0 se fait à l'aide de tables. On donne quelques exemples fondamentaux dans le Tableau 3 ci-dessous.

Tableau 3 : transformée en z de systèmes continus échantillonnés

F(s)	F <sub>BOZ</sub> (z)
1	1
$\frac{1}{s}$	$\frac{T_e \cdot z^{-1}}{1 - z^{-1}}$
$\frac{1}{1 + T \cdot s}$	$\begin{cases} \frac{b_1 \cdot z^{-1}}{1 + a_1 \cdot z^{-1}} \\ b_1 = 1 - e^{-T_e/T} \quad a_1 = -e^{-T_e/T} \end{cases}$
$\frac{e^{-Ls}}{1 + T \cdot s}$	$\begin{cases} \frac{b_1 \cdot z^{-1} + b_2 \cdot z^{-2}}{1 + a_1 \cdot z^{-1}} \quad a_1 = -e^{-T_e/T} \\ b_1 = 1 - e^{-(L-T_e)/T} \quad b_2 = e^{-T_e/T} (e^{L/T} - 1) \end{cases}$
$L < T_e$	

Si l'on dispose de Matlab c'est bien sûr encore plus simple. On utilise la fonction c2d avec l'argument 'zoh' pour "zero holder hold" (bloqueur d'ordre zéro).

#### Matlab

```
% Définition d'un système continu :
sysc=tf(1,[1 1]);
% Obtention de la transformée en z du système échantillonné à Te
Te=0.1;
sysd=c2d(sysc,Te,'zoh');
% Notation en z^-1
sysd.variable='z'
```

### 4.3 Effets de l'échantillonnage

Le convertisseur numérique a deux effets (néfastes) importants : l'effet de retard et l'effet de quantification.

L'effet de retard se voit dans (24) : on peut dire en première approximation que l'échantillonnage se traduit par un retard pur de  $T_e/2$ . Cela a donc, si ce temps

d'échantillonnage est non négligeable par rapport aux constantes de temps du système, un effet déstabilisant.

Pour illustrer l'effet de quantification supposons par exemple que l'on utilise un convertisseur analogique/numérique 8 bits 0-10V. Le plus petit incrément que le convertisseur peut distinguer est de  $10V/2^8 = 40mV$ . Si le signal utile varie physiquement entre 5V et 5,4V on n'aura que dix incréments utilisables... Ce phénomène n'entre pas dans la panoplie des phénomènes modélisables par la transformée en z. En pratique il est donc souvent nécessaire d'étudier le système et de synthétiser la commande numérique en négligeant ce phénomène puis de vérifier a posteriori qu'il n'a pas un effet trop néfaste.

#### 4.4 Où l'on revoit le théorème de Shannon

Le théorème de Shannon est bien connu en ce qui concerne l'échantillonnage de signaux discrets : on sait que le signal ne doit pas comporter de composante de fréquence supérieure à  $f_e/2$  si l'on ne veut pas observer de repliement. On doit aussi prendre des précautions si l'on veut travailler sur la fonction de transfert discrète correspondant à un système continu échantillonné.

On va traiter un exemple concret pour mettre en évidence le problème de repliement dans le cas de la commande des systèmes continus. On s'intéresse à deux systèmes continus du second ordre ayant des pôles de même partie réelle mais de parties imaginaires séparées de  $2\pi T_e$ . Le script Matlab ci-dessous montre la création des deux systèmes :

##### Matlab

```
% Définition de deux systèmes continus :
sys1=zpk([],[-1-2i -1+2i],2^2+1);
sys2=zpk([],[-1-(2+2*pi)*i -1+(2+2*pi)*i],1+(2+2*pi)^2);
```

On voit sur la Figure 11 que la réponse indicielle des deux systèmes est bien différente.

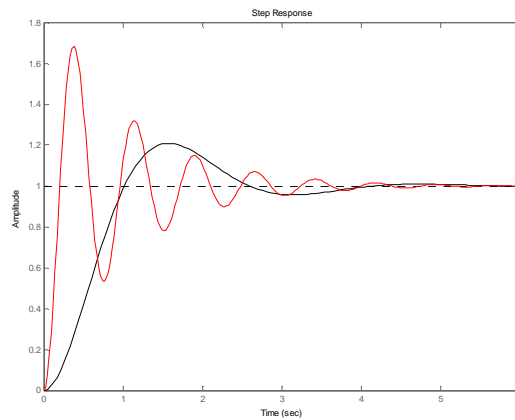


Figure 11 : réponse indicielle de deux systèmes du second ordre

Si l'on calcule la fonction de transfert échantillonnée les deux systèmes à la période de 1 seconde on obtient deux réponses indicielles très proches (représentés par des  $\circ$  sur la Figure 12). Conclusion : la période d'échantillonnage était mal choisie

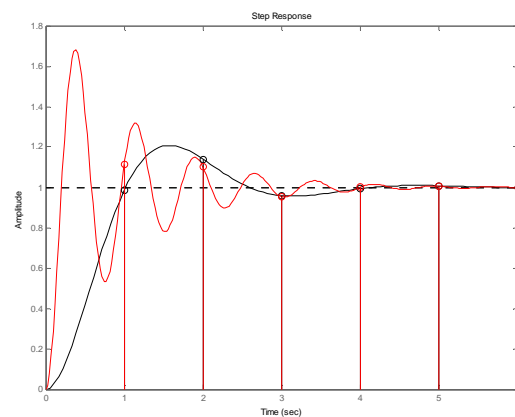


Figure 12 : réponse indicielle de deux systèmes du second ordre échantillonnés à 1s

En pratique la commande échantillonnée de systèmes continus doit être associée à certaines précautions :

- le système ne doit pas comporter de dynamiques propres situées dans les fréquences supérieures à  $f_c/2$  (exemple illustré dans ce chapitre).
- les signaux échantillonnés doivent être "nettoyés" des fréquences supérieures à  $f_c/2$  avec un filtre anti-repliement. (A moins que le signal soit suffisamment propre et ne comporte pas de composantes significatives au-delà de ces fréquences).

### Choix de la période d'échantillonnage

Le choix de la période d'échantillonnage est très important et dépend de la dynamique des signaux. Attention, les systèmes commandés en boucle fermée ont souvent une dynamique plus rapide que le système en boucle ouverte : il faut donc prendre en compte la dynamique en boucle fermée pour le choix de la fréquence d'échantillonnage.

En pratique on préconise de prendre une fréquence 5 à 25 fois supérieure à la bande passante souhaitée en boucle fermée.

On peut être tenté de prendre une fréquence d'échantillonnage très élevée si la technologie le permet (ordinateur et convertisseurs suffisamment rapide). Ceci peut amener rapidement à des problèmes numériques. Imaginons par exemple le cas d'un système ayant une dynamique très lente (quelques secondes) et un système de mesure codant les mesures sur 10 bits (cas classique). La Figure 13 illustre cet exemple. Le signal varie peu entre deux instants d'échantillonnage et les variations du signal sont faibles par rapport à la valeur de l'incrément numérique (figure du haut). Le calcul de la simple dérivée numérique se traduit donc par une succession de  $-1$   $0$  et  $+1$ .

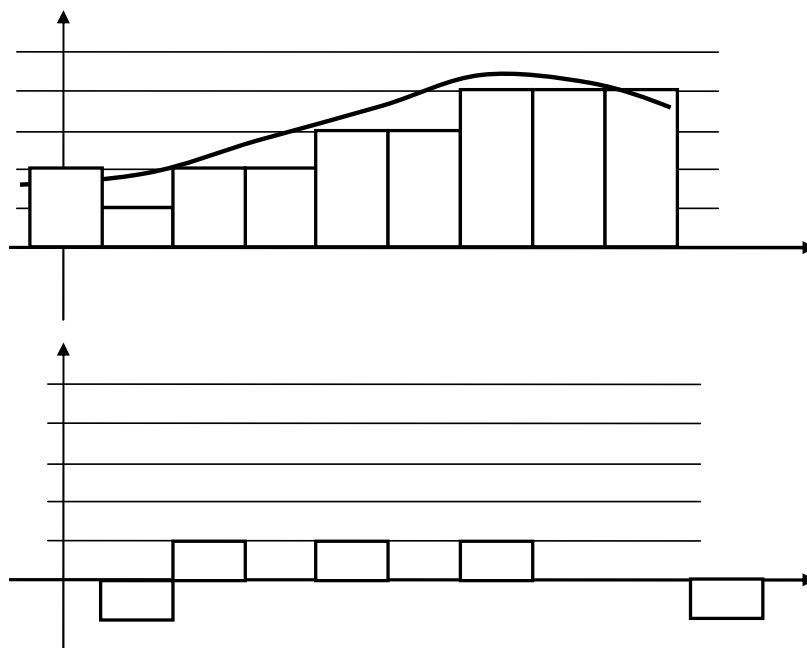


Figure 13 : période d'échantillonnage trop faible. En haut : signal échantillonné et discrétisé ; en bas : calcul de la dérivée numérique.

### Filtre anti-repliement

En toute rigueur chaque fois qu'un signal est échantillonné il faudrait faire précéder le convertisseur numérique d'un filtre analogique anti-repliement éliminant physiquement toutes les composantes du signal de fréquence supérieure à  $f_e/2$ .

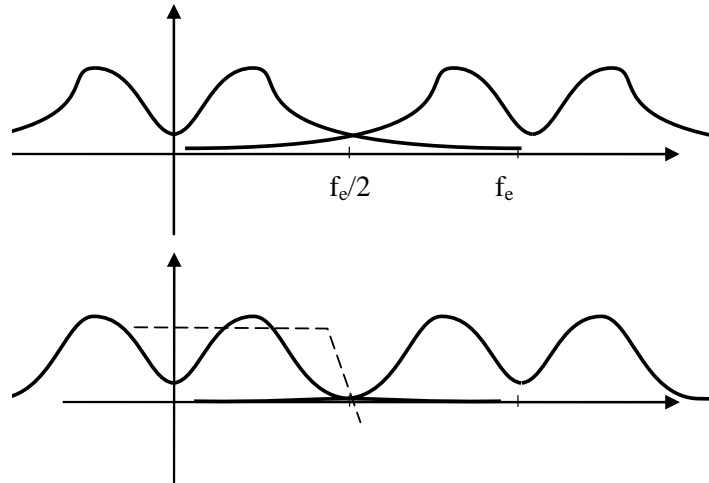


Figure 14 : Effet du filtre anti-repliement

Bien souvent la période d'échantillonnage de l'asservissement est relativement lente (quelques secondes) et il est alors impossible de réaliser un filtre de constante de temps aussi faible. On procède alors à un sur-échantillonnage : le signal est filtré, échantillonné à haute fréquence  $T'_e$  puis filtré numériquement et sous-échantillonné à la période  $T_e$  beaucoup plus grande que  $T'_e$  comme illustré sur la

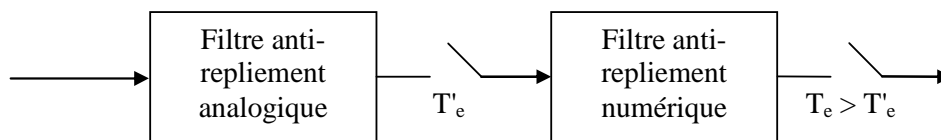


Figure 15 : Sur échantillonnage

Un simple filtre numérique de type moyenne mobile est souvent suffisant.

### 4.5 Discrétisation de fonction de transfert continues

Une manière classique pour réaliser un contrôle par ordinateur est de commencer par synthétiser un correcteur continu avec les outils de l'automatique classique puis de trouver un correcteur discret qui "approche" convenablement le correcteur continu.

#### 4.5.1 Approximation numérique d'équation différentielle

Supposons que le correcteur continu que l'on a synthétisé soit du type PD :

$$\text{PID}(s) = \frac{e(s)}{\varepsilon(s)} = k_p + k_D \cdot s \quad (26)$$

Il s'agit de calculer les commandes  $e(k)$  successives issues du correcteur en fonction des erreurs  $\varepsilon(k)$ . Une approche classique et intuitive est d'approximer la dérivée par une différence :

$$e(k) = k_p \cdot \varepsilon(k) + k_D \cdot \frac{\varepsilon(k) - \varepsilon(k-1)}{T_e} \quad (27)$$

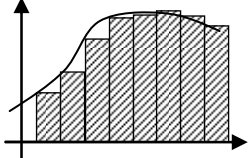
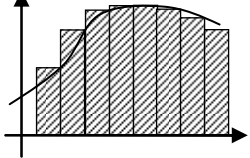
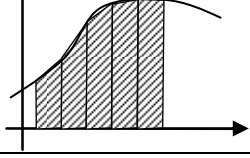
Si on calcule la transformée en  $z$  de  $e(k)$  on obtient :

$$e(z) = k_p \cdot \varepsilon(z) + k_D \cdot \frac{\varepsilon(z) - z^{-1}\varepsilon(z)}{T_e} = \left( k_p + k_D \cdot \frac{1 - z^{-1}}{T_e} \right) \cdot \varepsilon(z) \quad (28)$$

On retrouve la fonction de transfert PD(s) dans laquelle on a effectué le changement de variable  $s = \frac{1-z^{-1}}{T_e}$

De manière plus générale, l'approximation de fonction de transfert revient à effectuer l'approximation numérique d'intégration. Selon la méthode d'intégration choisie on effectue des changements de variable différents.

Tableau 4 : méthodes d'intégration

Méthode	Changement de variable	
Euler progressif (Euler's forward method)	$s = \frac{1 - z^{-1}}{z^{-1} \cdot T_e}$	
Euler rétrograde (Euler's backward method)	$s = \frac{1 - z^{-1}}{T_e}$	
Méthode des trapèzes, ou bilinéaire, ou de Tustin	$s = \frac{2}{T_e} \cdot \frac{1 - z^{-1}}{1 + z^{-1}}$	

En pratique le changement de variable  $s = \frac{1 - z^{-1}}{z^{-1} \cdot T_e}$  n'est pas sans risque. On donne sur la Figure 16 la correspondance entre la zone stable du plan complexe en  $s$  et la zone correspondante après la transformation (Euler progressif, Euler rétrograde, bilinéaire). On voit notamment qu'il est possible, partant d'un système continu stable, d'obtenir des pôles dans le domaine discret en dehors du cercle de stabilité si la transformation appliquée est celle d'Euler progressif.

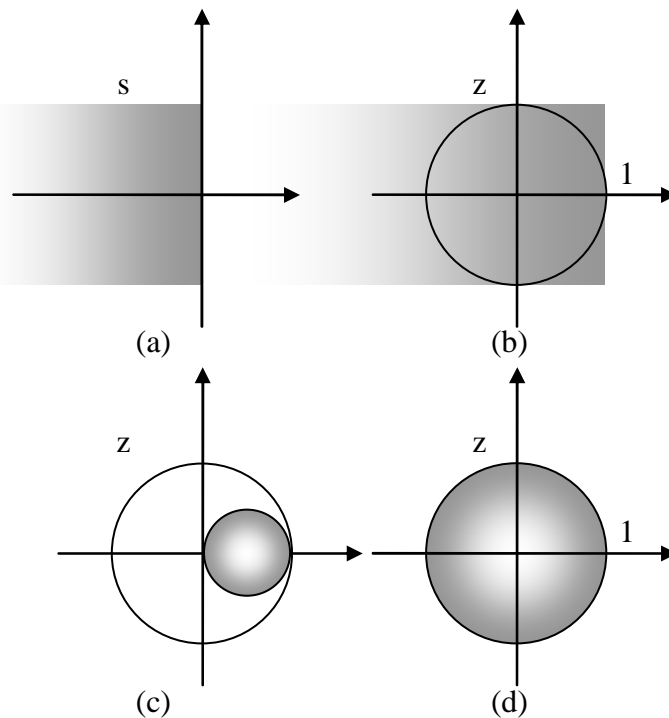


Figure 16 : correspondance des plans complexes. (a) : zone de stabilité dans le plan complexe continu en  $z$  ; (b) zone correspondante avec l'approximation d'Euler progressif ; (c) zone correspondante avec l'approximation d'Euler rétrograde ; (d) zone correspondante avec l'approximation bilinéaire.

La fonction `c2d` de Matlab permet d'obtenir l'équivalent discret d'une fonction de transfert continue avec la méthode d'intégration comme argument :

```

Matlab
sysc=tf([1 0],[1 2 5]);
Te=0.1;
% Méthode d'euler :
sysd1=c2d(sysc,Te,'zoh');
% Méthode des trapèzes :
sysd2=c2d(sysc,Te,'foh');
% Méthode bilinéaire :
sysd3=c2d(sysc,Te,'tustin');
    
```

#### 4.5.2 Superposition des réponses indicielles

Une autre méthode pour approximer une fonction de transfert discrète est de s'assurer que la réponse indicielle de sa version discrétisée correspond exactement lors des temps d'échantillonnage.



### 4.5.3 Correspondances pôles - zéros

La dernière méthode est de partir d'une décomposition de la fonction de transfert continue en pôles et zéros et d'effectuer la conversion pôles à pôles et zéros par zéros. La correspondance, selon que le pôle est réel ou complexe est donnée ci-dessous :

$$\begin{aligned} (s + a) &\rightarrow (z - e^{-a \cdot T_e}) \\ (s + a)^2 + b^2 &\rightarrow z^2 - 2 \cdot e^{-a \cdot T_e} \cdot \cos(b \cdot T_e) \cdot z + e^{-2 \cdot a \cdot T_e} \end{aligned} \quad (29)$$

### 4.5.4 Bilan

Aucune méthode n'est parfaite et on n'aura jamais en même temps :

- même réponse indicielle
- même réponse impulsionnelle
- approximation (Euler, Tustin, etc...) de l'équation différentielle
- correspondance des pôles et des zéros
- correspondance des réponses fréquentielles

Ceci étant dit le choix de la période d'échantillonnage est primordial. Si la fréquence d'échantillonnage est beaucoup plus faible que la dynamique de la fonction de transfert à approximer la différence entre chaque méthode sera minimale.



## 5. IDENTIFICATION DES SYSTEMES ECHANTILLONES

Dans le présent chapitre on donne les bases des méthodes d'identification. Pour une approche complète, voir le cours de troisième année "Identification avion" [6] et les ouvrages [1] et [7].

Même si l'on dispose d'un modèle théorique du processus que l'on souhaite commander il est presque toujours nécessaire de l'affiner en procédant à des essais sur le système réel. Bien souvent le seul modèle dont l'on dispose est ce modèle issu de l'expérimentation. Différentes méthodes s'appliquent selon l'objectif recherché et le niveau de précision exigé. On distinguera :

- les modèles paramétriques
- les modèles simples
- les modèles non paramétriques

### 5.1 Identification d'un modèle ARX par la méthode des moindres carrés récursifs

A partir de cet exemple, le plus simple qui soit, on pose les bases de toutes les méthodes d'identification.

On considère un système d'entrée  $u(t)$  et de sortie  $y(t)$  représenté par l'équation de récurrence suivante :

$$y(t) + a_1 \cdot y(t-1) + a_2 \cdot y(t-2) + \dots + a_n \cdot y(t-n) = b_0 \cdot u(t) + b_1 \cdot u(t-1) + b_2 \cdot u(t-2) + \dots + b_m \cdot u(t-m) \quad (30)$$

L'objectif est, à partir d'essais et de mesures, d'identifier les meilleures valeurs possibles pour les coefficients  $a_i$  et  $b_i$ .

L'équation (30) peut être réécrite pour mettre en évidence la *prédiction* de la sortie  $y(t)$  à partir des mesures passées :

$$y(t) = -a_1 \cdot y(t-1) - a_2 \cdot y(t-2) - \dots - a_n \cdot y(t-n) + b_0 \cdot u(t) + b_1 \cdot u(t-1) + b_2 \cdot u(t-2) + \dots + b_m \cdot u(t-m) \quad (31)$$

Soit :

$$y(t) = \varphi^T(t) \cdot \theta(t)$$

avec :

$$\begin{cases} \varphi^T(t) = [-y(t-1) \ \dots \ -y(t-n) \ , \ u(t-1) \ \dots \ u(t-m)] \\ \theta(t) = [a_1 \ \dots \ a_n \ , \ b_1 \ \dots \ b_m] \end{cases} \quad (32)$$

Le but est de trouver le meilleur ensemble de paramètres  $\theta(t)$  : celui qui, à partir d'un ensemble de mesures  $\varphi(t)$  permet de prédire avec la meilleure précision la sortie effectivement mesurée.

Supposons qu'à la suite d'une expérience on dispose d'un ensemble de mesures  $\{y(t), u(t)\}_{t=1:N}$  et que l'ensemble des paramètres  $\theta$  n'est pas bien connu. On construit la séquence des *sorties prédites* :

$$\hat{y}(t) = \varphi^T(t) \cdot \theta(t) \quad (33)$$

Le paramètre  $\theta$  est celui qui minimise une norme quadratique :

$$\theta = \arg \max \left( \frac{1}{N} \sum_{t=1}^N (y(t) - \hat{y}(t))^2 \right) = \arg \max \left( \frac{1}{N} \sum_{t=1}^N (y(t) - \varphi^T(t) \cdot \theta(t))^2 \right) \quad (34)$$

Compte tenu du fait que l'équation est linéaire au départ et que le terme à minimiser est du type quadratique la solution est simplement obtenue par une pseudo-inverse :

$$\theta = \left[ \sum_{t=1}^N (\varphi(t) \cdot \varphi^T(t)) \right]^{-1} \sum_{t=1}^N (\varphi(t) \cdot y(t)) \quad (35)$$

On obtient donc la solution classique des moindres carrés. La formule (35) est le plus souvent utilisée sous sa forme récursive :

$$\theta(t) = \theta(t-1) + F(t) \cdot \varphi(t) \cdot (y(t) - \theta^T(t-1) \cdot \varphi(t))$$

avec

$$F(t)^{-1} = F(t-1)^{-1} + \varphi(t) \cdot \varphi(t)^T \quad (36)$$

ou

$$F(t) = F(t-1) - \frac{F(t-1) \cdot \varphi(t) \cdot \varphi(t)^T \cdot F(t-1)}{1 + \varphi(t)^T \cdot F(t-1) \cdot \varphi(t)}$$

(On laisse au lecteur le plaisir d'effectuer lui-même les démonstrations, qu'il peut du reste trouver dans [1] et [7]).

## 5.2 L'identification paramétrique en général et en pratique

L'exemple précédent a permis de mettre en évidence tous les aspects de la méthodologie classique de l'identification :

- il a fallu choisir un ensemble de mesures  $\{y(t), u(t)\}_{t=1:N}$ . Le choix du signal d'excitation  $u(t)$  est en pratique primordial,
- il a fallu choisir une structure pour le modèle à identifier (ici les ordres  $n$  et  $m$ ),
- il a fallu choisir un critère (ici un critère quadratique sur l'erreur entre la sortie et la sortie prédite),
- il a fallu choisir un algorithme et le paramétrer (ici l'algorithme des moindres carrés récursifs qu'il a fallu initialiser)

Enfin un point non traité dans le chapitre précédent concerne la validation :

- il aurait fallu déterminer une méthode pour comparer différents modèles et choisir le meilleur

### 5.2.1 Choix du signal d'excitation, traitement des mesures

De manière générale le système que l'on doit identifier doit être excité dans tout le domaine fréquentiel qui nous intéresse. Le signal classiquement utilisé est une séquence pseudo binaire aléatoire (Pseudo Random Binary Signal) tel celui représenté sur la Figure 17.

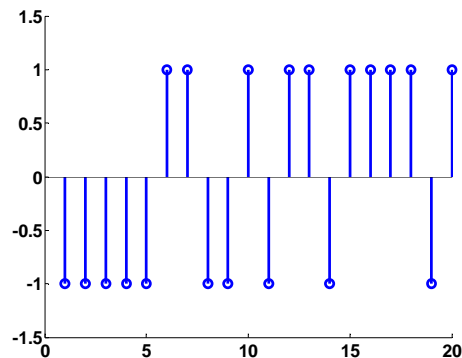


Figure 17 : exemple d'une séquence PRBS

Le signal d'excitation doit se superposer au signal de commande correspondant au point de fonctionnement étudié. Son amplitude doit être faible par rapport à l'amplitude des commandes du système (quelques %).

Une fois le système excité et les mesures effectuées il faut supprimer les valeurs moyennes des mesures.

### 5.2.2 Choix d'une structure de modèle

Le modèle étudié dans le chapitre précédent est de type ARX. On peut le noter sous forme polynomiale :

$$A \cdot y = B \cdot u \quad (37)$$

Ce modèle correspond en fait à un parti-pris sur la manière dont on traite les perturbations. En réalité on prend en compte un signal d'erreurs possibles  $e$  (de moyenne nulle) tels que :

$$A \cdot y = B \cdot u + e \quad (38)$$

Si l'on fait cette hypothèse de modélisation alors la méthode optimale est celle des moindres carrés. Malheureusement cette manière de prendre en compte les perturbations est bien restrictive. D'autres modélisations sont possibles, adaptées à d'autres algorithmes, par exemple le modèle OE :

$$y = \frac{B}{F} \cdot u + e \quad (39)$$

Il est aussi possible de "structurer" le modèle de bruit comme par exemple dans les modèles ARMAX ou BJ :

$$\begin{aligned} A \cdot y &= B \cdot u + C \cdot e \\ y &= \frac{B}{F} \cdot u + \frac{C}{D} \cdot e \end{aligned} \quad (40)$$

La structure du modèle étant choisie il faut choisir l'ordre de ses différents polynômes.

### 5.2.3 Choix d'un critère à minimiser

Dans la plupart des cas le critère à minimiser est l'erreur quadratique entre la sortie et la sortie prédite à un coup (one step ahead prediction). Ceci n'est bien évidemment pas restrictif et peut être généralisé à la prédiction à N coups.

### 5.2.4 Choix d'un algorithme

Même si chaque structure de modèle est généralement associée à un algorithme "optimal" celui-ci doit généralement se paramétrer.

Prenons par exemple l'algorithme des moindres carrés récursifs tel que présenté dans (36). Cette équation peut être modifiée en intégrant un "facteur d'oubli"  $\lambda$  :

$$\begin{aligned} \theta(t) &= \theta(t-1) + F(t) \cdot \varphi(t) \cdot (y(t) - \theta^T(t-1) \cdot \varphi(t)) \\ \text{avec} & \\ F(t)^{-1} &= \lambda \cdot F(t-1)^{-1} + \varphi(t) \cdot \varphi(t)^T \end{aligned} \quad (41)$$

Ce paramètre (typiquement entre 0.95 et 0.99) permet, comme son nom l'indique, "d'oublier" les anciennes mesures et de se concentrer sur les mesures les plus récentes.

### 5.2.5 Validation

Une fois l'algorithme d'identification appliqué et le modèle obtenu (ce qui prend généralement quelques secondes avec un ordinateur moderne) il reste à expertiser celui-ci. Selon la "qualité" du modèle obtenu on remet en cause l'une ou l'autre des étapes (signal d'excitation, structure du modèle, algorithme) et on réitère jusqu'à obtenir un modèle satisfaisant.

En pratique on essaie un certain nombre de modèles avant de choisir le "meilleur".

Le premier critère de qualité est la capacité du modèle à prédire effectivement la sortie (sortie prédite à un coup, sortie prédite à N coups). Il faut pour cela utiliser un jeu de mesures non utilisé pour l'identification. (En pratique le jeu de mesures initiales est coupé en deux : une partie pour l'identification et une partie pour la validation).

Le second critère de qualité est l'analyse du résidu : la composition spectrale de l'erreur de prédiction.

### 5.2.6 Utilisation de logiciels

En pratique il est commode d'utiliser un célèbre outil de calcul scientifique et sa toolboxe d'identification.

Comme les calculs proprement dit ne prennent que quelques secondes il est facile de réaliser un grand nombre d'identifications et de stocker les différents modèles pour les comparer ensuite. On est rapidement noyé sous le grand nombre de modèles possibles...

## 5.3 Identification "simple"

Il est souvent suffisant d'obtenir un modèle "simple" de type premier ordre et un simple essai indiciel convient souvent. Le système est excité avec un échelon unité et on cherche le meilleur modèle du premier ordre correspondant.

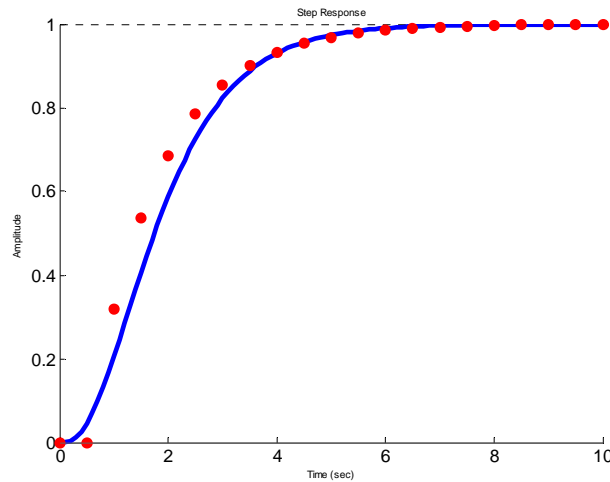


Figure 18 : identification sur un essai indiciel

Dans l'exemple de la Figure 18 on superpose le résultat d'une mesure (en bleu) avec la simulation d'un modèle discret avec une période de 0.5s. Le modèle cherché est du premier ordre avec un retard pur  $d$  :

$$y(z) = \frac{(1 + c_1) \cdot z^{-d}}{1 + c_1 \cdot z^{-1}} \cdot u(z) \quad (42)$$

Il s'agit de trouver l'entier  $d$  et le nombre réel  $c_1$  tels que la réponse indicelle du modèle corresponde pour le mieux aux mesures issues de l'essai. Ceci peut se formaliser par exemple de la manière suivante :

$$(c_1 \quad d_1) = \arg \min \sum_{t=1}^N (y_{\text{modèle}}(t) - y_{\text{mesure}}(t))^2 \quad (43)$$

Cette optimisation peut se faire à la main ou grâce à une procédure automatisée (avec le célèbre outil de calcul scientifique).

Dans le cas où le système comprend naturellement une intégration (par exemple la commande d'un moteur à courant continu par une tension et la mesure de l'angle de rotation) il est utile de prendre en compte explicitement cette intégration dans le modèle :

$$y(z) = \frac{1}{1 - z^{-1}} \cdot \frac{(1 + c_1) \cdot z^{-d}}{1 + c_1 \cdot z^{-1}} \cdot u(z) \quad (44)$$



La procédure est la même mais au lieu de prendre en compte la sortie  $y(t)$  on prend en compte la dérivée  $y'(t)$  :

$$y'(t) = y(t) - y(t-1) \quad (45)$$

## 5.4 Identification non paramétrique

Pour terminer ce panorama rapide sur les méthodes d'identification il faut parler des méthodes non paramétriques : celles qui permettent de tracer la réponse impulsionnelle ou la réponse fréquentielle du système.

### 5.4.1 Estimation de la réponse impulsionnelle

La réponse impulsionnelle d'un système peut être obtenue simplement en appliquant une entrée de type "impulsion" au système et d'en mesurer les conséquences.

La seconde méthode consiste à utiliser la méthode des moindres carrés (ou une méthode dérivée) comme vue au chapitre 5.1. La réponse impulsionnelle d'un système peut en effet être vue comme la fonction de transfert particulière où le dénominateur est de degré nul et le numérateur comporte un nombre infini (en pratique : grand) d'éléments ( $n = 0$  et  $m$  grand) :

$$y(t) = b_0 \cdot u(t) + b_1 \cdot u(t-1) + b_2 \cdot u(t-2) + \dots + b_m \cdot u(t-m) \quad (46)$$

La dernière méthode, qui sera développée ici est l'analyse d'intercorrélation.

Par définition l'intercorrélation  $R_{yu}$  entre deux signaux est :

$$R_{yu}(\tau) = E[(y(t) - E(y(t))) \cdot (u(t-\tau) - E(u(t-\tau)))] \quad (47)$$

Dans cette formule l'espérance mathématique peut être approchée grâce à la loi des grands nombres :

$$R_{yu}(\tau) \approx R_{yu}^N(\tau) = \lim_{N \rightarrow \infty} \frac{1}{N} \sum_{t=1}^N [(y(t) - m_y) \cdot (u(t-\tau) - m_u)] \quad (48)$$

On s'intéresse maintenant à un système tel que décrit par l'équation impulsionnelle (46) et on suppose que le signal d'entrée est un bruit blanc. Le signal d'entrée vérifie donc :

$$\begin{cases} \tau \neq 0 & R_{uu}(\tau) = 0 \\ \tau = 0 & R_{uu}(\tau) = R \end{cases} \quad (49)$$

Il vient alors naturellement :

$$b_k \approx \frac{1}{R} R_{yu}(k) \quad (50)$$

### 5.4.2 Estimation de la réponse fréquentielle

On utilise ici le spectre d'intercorrélation. Par définition le spectre d'intercorrélation entre deux signaux est :

$$\Phi_{yu}(\omega) = \sum_{\tau=-\infty}^{\infty} R_{yu}(\tau) \cdot e^{-i\omega\tau} \quad (51)$$

On suppose que le signal d'entrée  $u(t)$  et le signal de sortie  $y(t)$  sont reliés par une fonction de transfert linéaire du type :

$$y(t) = F \cdot u(t) + v(t) \quad (52)$$

où  $v(t)$  est une perturbation indépendante de  $u(t)$ . On a alors les relations :

$$\Phi_{yu}(\omega) = F(j\omega) \cdot \Phi_{uu}(\omega) \quad (53)$$

Il reste à estimer les spectres d'intercorrélation pour remonter à la réponse fréquentielle  $F(j\omega)$  recherchée. Une idée naturelle est d'utiliser la relation (48) mais cela marche très mal dans presque tous les cas, même en choisissant  $N$  assez grand. L'amélioration est d'utiliser une "fenêtre" pour favoriser les valeurs faibles de  $\tau$  par rapport aux valeurs élevées. La solution classique utilisant la *fenêtre de Hamming* donne :

$$\Phi_{yu}^N(\omega) = \sum_{\tau=-\gamma}^{\gamma} R_{yu}^N(\tau) \cdot w_{\gamma}(\tau) \cdot e^{-i\omega\tau} \quad (54)$$

avec

$$\begin{cases} w_{\gamma}(\tau) = \frac{1}{2} \left( 1 + \cos\left(\frac{\pi \cdot \tau}{\gamma}\right) \right) & |\tau| < \gamma \\ w_{\gamma}(\tau) = 0 & |\tau| > \gamma \end{cases}$$

On utilise ensuite la relation (51) pour estimer la réponse fréquentielle :

$$F(j \cdot \omega) = \frac{\Phi_u^N(\omega)}{\Phi_{yu}^N(\omega)} \quad (55)$$



## 6. SYSTEMES BOUCLES

### 6.1 Fonction de transfert en boucle fermée

Un système et sa commande par ordinateur peut être représenté par un ensemble de fonctions de transfert formant un schéma-bloc plus ou moins compliqué (Figure 19).

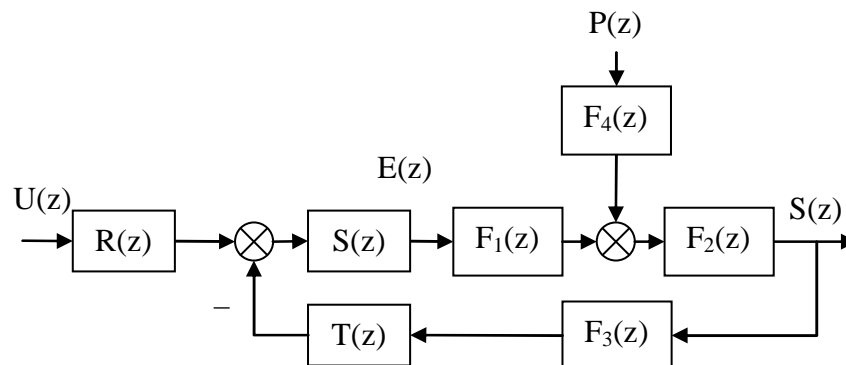


Figure 19 : système discret commandé par ordinateur. Le correcteur est représenté par les fonctions R,S,T, le système par  $F_1$  et  $F_2$ , le modèle de perturbations par  $F_4$  et le modèle des actionneurs par  $F_4$ .

Quelle que soit la structure d'un système asservi il est possible de se ramener à une forme canonique comprenant une fonction de transfert en boucle ouverte en combinant judicieusement les fonctions de transfert du procédé, du régulateur et des perturbations. La Figure 20 représente par exemple le système de la Figure 19 où  $F$  est une combinaison des fonctions de transfert du système, du régulateur, des perturbations et des capteurs.

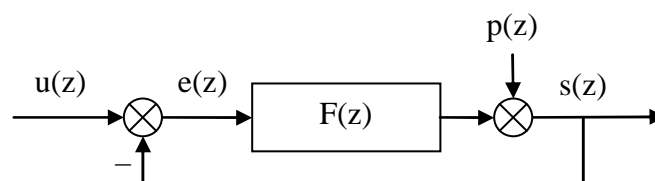


Figure 20 : système discret, notations.

Par défaut on considère une contre-réaction négative et une perturbation additive sur la sortie. La fonction de transfert  $F$  est appelée *fonction de transfert en boucle ouverte*. Comme pour les systèmes continus la *fonction de transfert en boucle fermée* est obtenue simplement à partir de  $F$  (58).

$$F(z) = \frac{s(z)}{e(z)} = \frac{B(z^{-1})}{A(z^{-1})} \quad (56)$$

$$G(z) = \frac{s(z)}{u(z)} = \frac{B(z^{-1})}{A(z^{-1}) + B(z^{-1})}$$

## 6.2 Erreur statique

L'erreur statique du système en boucle fermée, comme pour le système en boucle ouverte, est simplement le rapport des coefficients.

$$G_0 = \frac{\sum_{i=0}^{i=m} b_i}{1 + \sum_{i=0}^{i=m} b_i + \sum_{i=1}^{i=n} a_i} \quad (57)$$

Si l'on souhaite que l'asservissement confère un gain statique unitaire au système en boucle fermée (c'est-à-dire que l'on force la sortie  $s(k)$  à tendre vers la valeur de consigne  $u(k)$ ) il faut que ce rapport de coefficients soit égal à un. On montre facilement que cela est obtenu si la fonction de transfert  $F(z)$  comporte un intégrateur :

$$F(z) = \frac{s(z)}{e(z)} = \frac{1}{1 - z^{-1}} \cdot \frac{B(z)}{A'(z)} \quad (58)$$

## 6.3 Rejet de perturbations

La fonction de transfert entre la perturbation  $p(z)$  et la sortie  $y(z)$  est appelée fonction de sensibilité perturbation-sortie, notée  $S_{yp}$ .

$$S_{yp}(z) = \frac{s(z)}{p(z)} = \frac{A(z^{-1})}{A(z^{-1}) + B(z^{-1})} \quad (59)$$

Si l'on souhaite qu'une perturbation de type échelon soit parfaitement rejetée il faut que  $A(1) = 0$ , on retrouve donc la condition (58) à savoir que le système doit comporter une intégration dans la boucle directe. On peut être en outre être amené à spécifier que les perturbations ne soient pas amplifiées dans une certaine bande de fréquence. On rajoute donc une contrainte du type  $|S_{yp}(e^{j2\pi f})| < S_{\max}$  pour  $f$  appartenant à un certain domaine de  $[0 \ 0.5]$ .

## 6.4 Stabilité des systèmes à simple gain de rétroaction

Les critères de Routh et de Jury (donnés dans le chapitre 3.2.7 p13) ou le calcul numérique des pôles permettent de déterminer la stabilité du système en boucle fermée si l'on connaît sa fonction de transfert.

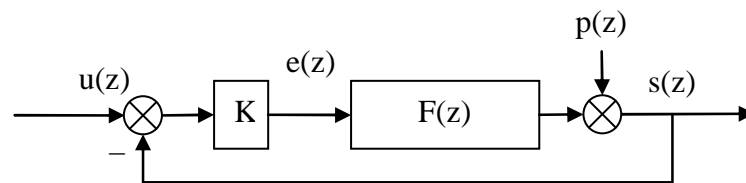


Figure 21 : système bouclé par un simple gain.

On s'intéresse ici au cas particulier où l'asservissement est réalisé par un simple gain de rétroaction  $K$  (Figure 21). On cherche la valeur de  $K$  qui amène le système en limite de stabilité. L'outil adapté ici est le lieu d'Evans ou lieu des racines (root locus) qui permet de visualiser l'évolution des pôles du système bouclé lorsque le gain  $K$  varie.

Prenons l'exemple d'un système comportant trois pôles et deux zéros. L'exemple est donné sous forme de script Matlab :

### Matlab

```
% Définition d'un système discret :
sys=zpk([-2.037 -0.1406],[0.6065 0.3679 0.3679],1,1)
%Tracé du lieu des racines
rlocus(sys);
```

Le script précédent produit le tracé de la Figure 22. On pourrait y lire que la limite de stabilité est obtenue pour  $K=0.2$  environ (détail de la Figure 23).

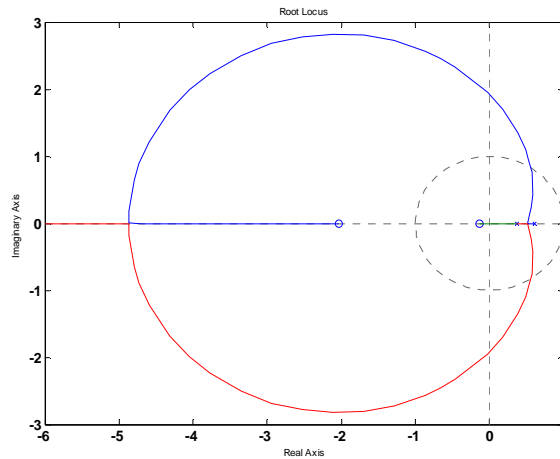


Figure 22 : lieu des racines

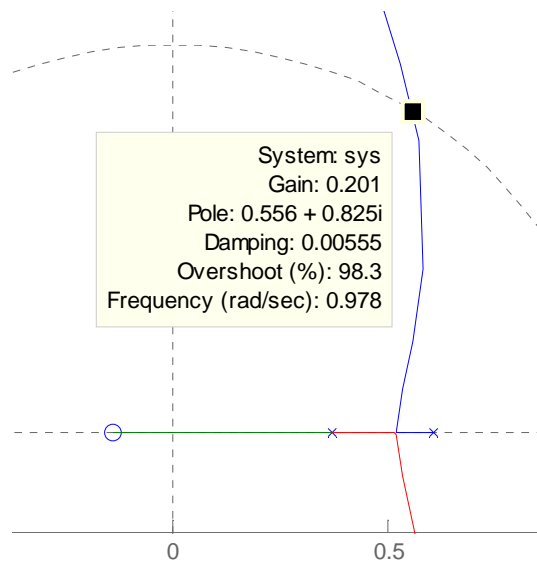


Figure 23 : lieu des racines : détail de la limite de stabilité

On voit sur la Figure 22 que le lieu des racines d'un système échantillonné ressemble à celui d'un système continu :

- on commence par tracer les pôles ( $\times$ ) et les zéros ( $o$ ) du système en boucle ouverte.
- les pôles du système en boucle fermée décrivent un lieu qui part des pôles (pour  $K = 0$ ) et arrive aux zéros (pour  $K = \infty$ )
- lorsque il y a  $n-m$  plus de pôles que de zéros le système comporte  $n-m$  branches asymptotiques.

On ne décrira pas ici les diverses propriétés du lieu des racines qui permettent de le tracer à la main puisque il est facile à obtenir avec un ordinateur.



Il peut aussi être utilisé pour régler "à la main" le gain permettant d'obtenir un ou plusieurs modes dominants ayant certaines propriétés de fréquence et d'amortissement. On voit sur la Figure 24 que le gain 0.017 permet d'obtenir un pôle double d'amortissement 0.706 et de fréquence 0.65 rad/s et un pôle simple de fréquence 1.63 rad/s.

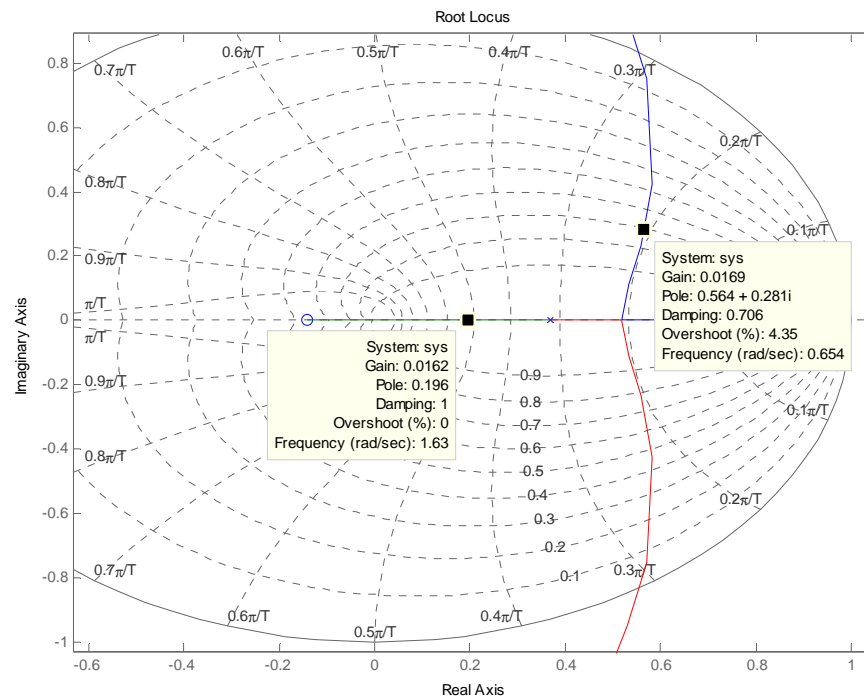


Figure 24 : utilisation du lieu des racines pour régler le gain de la boucle fermée

L'outil rltool de Matlab permet de nombreuses manipulations graphiques de ce type.

## 6.5 Analyse des systèmes bouclés dans le domaine fréquentiel : stabilité

Comme dans le cas continu il est possible de prévoir des propriétés, en particulier la stabilité, d'un système en boucle fermée à partir de l'examen de la fonction de transfert en boucle ouverte.

On ne redémontrera pas ici le théorème de Nyquist mais on se contente de rappeler l'énoncé.

On rappelle que le *lieu de Nyquist* est le lieu de  $F(j2\pi f)$  dans le plan complexe pour  $f$  variant de 0 à  $+\infty$ . A noter que le lieu pour la zone  $[0, 0.5]$  est symétrique de celui pour la zone  $[0.5, 1]$ . Pour cette raison on ne représente généralement que le lieu pour  $f$  appartenant à  $[0, 0.5]$ .

### 6.5.1 Théorème de Nyquist dans le cas où le système est stable en boucle ouverte (critère du revers)

Le système est stable en boucle fermée si le lieu de Nyquist de la fonction de transfert en boucle ouverte parcouru dans le sens des fréquences croissantes de 0 à 0.5 laisse le point critique (-1,0) à gauche.

La Figure 25 représente par exemple le lieu de Nyquist d'une fonction de transfert en boucle ouverte qui sera stable en boucle fermée.

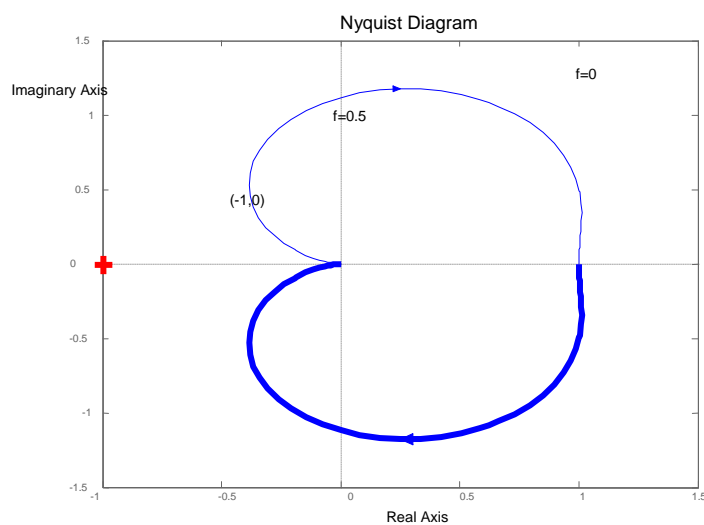


Figure 25 : lieu de Nyquist d'un système stable

### 6.5.2 Théorème de Nyquist dans le cas où le système est instable en boucle ouverte

Si le système en boucle ouverte comporte  $N$  pôles instables il sera stable en boucle fermée si le lieu de Nyquist de la fonction de transfert en boucle ouverte parcouru dans le sens des fréquences croissantes de  $-0.5$  à  $0.5$  entoure le point critique  $(-1,0)$   $N$  fois dans le sens des aiguilles d'une montre..

### 6.5.3 Interprétations du lieu de Nyquist

Comme dans le cas continu, la distance à laquelle on passe du point critique renseigne sur la robustesse du système bouclé. La distance sera exprimée de différente manière (Figure 26) :

- marge de module  $\Delta M$  : plus petite distance entre le lieu des racines et le point critique,

- marge de gain  $\Delta G$
- marge de phase  $\Delta P$

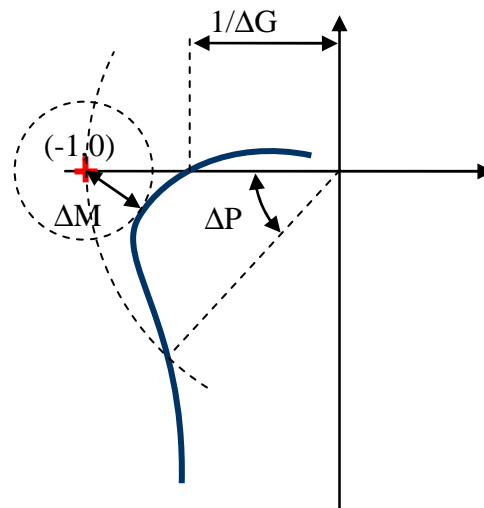


Figure 26 : Marges de stabilité

Les valeurs typiques de ces marges sont données ci-après :

- $\Delta G \geq 6\text{dB}$
- $\Delta M \geq -6\text{dB}$
- $30^\circ \leq \Delta P \leq 60^\circ$

La distance entre le point critique et le lieu des racines est la norme de  $1+F(j2\pi f)$ . On peut donc exprimer la marge de module en fonction de la fonction de sensibilité :

$$\begin{aligned} \Delta M &= \min |1 + F(j \cdot \omega)|_{\omega \in [0, \pi]} \\ \Delta M &= \min \left| 1 + \frac{B(j \cdot \omega)}{A(j \cdot \omega)} \right|_{\omega \in [0, \pi]} = \max \left| \frac{A(j \cdot \omega)}{A(j \cdot \omega) + B(j \cdot \omega)} \right|_{\omega \in [0, \pi]} \quad (60) \\ \Delta M &= \max |S_{yp}(j \cdot \omega)|_{\omega \in [0, \pi]}^{-1} \\ \Delta M_{\text{dB}} &= -\max |S_{yp}(j \cdot \omega)|_{\text{dB}} \end{aligned}$$

Si l'on trace l'évolution de  $S_{yp}$  en fonction de  $f$  on obtient donc la valeur de la marge de module.



# 7. METHODES DE REGULATION DES SYSTEMES ECHANTILLONES

## 7.1 Introduction

Les bases théoriques permettant de concevoir le correcteur pour un système échantillonné ont été posées dans les chapitres précédents. Ce chapitre fait le bilan des connaissances acquises et donne quelques solutions pour calculer des correcteurs à partir d'un cahier des charges précis.

Les systèmes que l'on cherche à commander sont généralement, par nature, continus. La réalisation d'un correcteur par un ordinateur implique que l'on génère une équation de commande discrète (généralement une équation récurrente). L'obtention de cette équation de commande peut se faire de deux manières :

- la première méthode consiste à d'abord synthétiser un correcteur continu (en utilisant les outils classiques de l'automatique et la simulation) et discrétiser ensuite ce correcteur. C'est l'objet du chapitre 7.2.2.
- la seconde méthode consiste à tenir la représentation discrète du système continu associé à son interface numérique (convertisseurs analogique-numérique et numérique-analogique) comme expliqué dans le chapitre 4.5 puis synthétiser directement le correcteur discret. Cela sera expliqué dans le chapitre 7.3 et suivants.

Ce choix peut être résumé par la figure suivante :

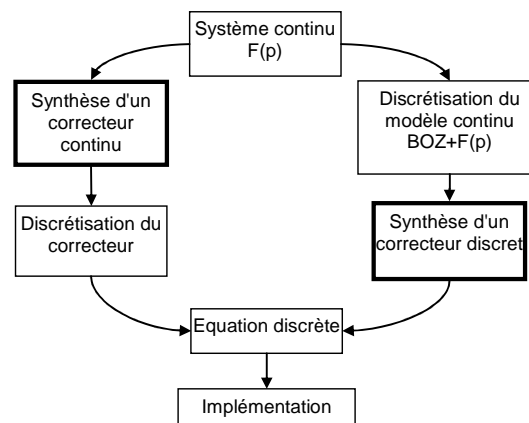


Figure 27 : Réalisation d'un correcteur discret.

Au-delà des affaires de goût, le choix de la méthode sera souvent dicté par l'application. En effet, la première méthode est d'autant meilleure que le correcteur discrétisé approche correctement le correcteur continu. Cela implique que le temps d'échantillonnage soit très faible par rapport aux constantes de temps du système

bouclé. Dans le cas où les contraintes temps réel sont fortes (capacités de calcul partagées par exemple) il sera préférable de synthétiser un correcteur discret.

## 7.2 Structure d'un système commandé par ordinateur, hypothèses

Par la suite on se ramènera toujours à un système tel que celui représenté sur la Figure 28 en cherchant un correcteur  $K(z)$  linéaire. On montre que quel que soit le correcteur linéaire on peut se ramener à la forme représenté sur la Figure 29 (forme RST).

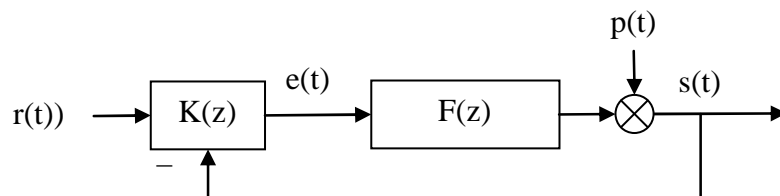


Figure 28 : système bouclé, forme générale.

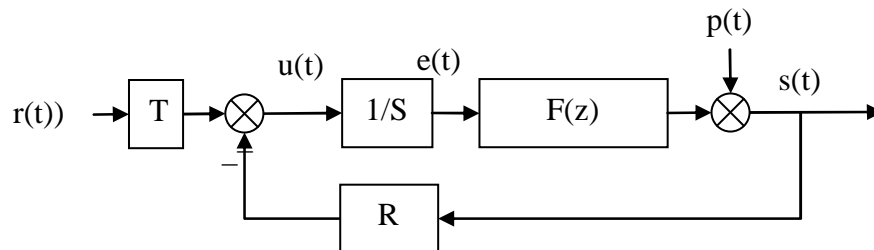


Figure 29 : système bouclé, forme canonique.

### 7.2.1 Discrétisation, choix de la période d'échantillonnage

Du point de vue du ordinateur, l'extérieur (le procédé) est représenté par la fonction de transfert  $F(z)$ . L'hypothèse importante est que la commande (sortie du ordinateur) et la mesure (entrée du ordinateur) sont échantillonnées de manière synchrone à la même période  $T_e$ .

Le choix judicieux de la fréquence d'échantillonnage est très important. En pratique on choisit cette fréquence en fonction de la bande passante du système en **boucle fermée** :

$$f_e = (6 \text{ à } 25) f^{BF} \quad (61)$$

### 7.2.2 Discrétisation d'un correcteur continu

La première méthode pour obtenir les équations d'un correcteur numérique est de faire sa synthèse avec les outils de l'automatique continue puis de discrétiser la fonction de transfert du correcteur continu obtenu (voir pour cela le chapitre 4.2 *Echantillonnage des systèmes continus*).

On va voir par la suite que la synthèse directe d'un correcteur discret offre plus de degré de liberté et permet en général de remplir un cahier des charges plus complet.

## 7.3 Régulateur PID numérique

Le régulateur le plus populaire est le régulateur PID. Nous allons voir dans un premier temps comment on génère un régulateur PID à partir des équations bien connues du régulateur PID continu et comment ces équations peuvent être généralisées pour obtenir plus de degrés de liberté pour le réglage.

### 7.3.1 Cas 1

Un régulateur PID continu a une fonction de transfert du type :

$$K_{\text{PID}}(s) = K \cdot \left( 1 + \frac{1}{T_i \cdot s} + \frac{T_d \cdot s}{1 + \frac{T_d}{N} \cdot s} \right) \quad (62)$$

La discrétisation de ce correcteur continu par la méthode d'Euler (Tableau 4 page 31) se fait en remplaçant la variable de Laplace  $s$  par le terme  $\frac{1-q^{-1}}{q^{-1} \cdot T_e}$ . On obtient :

$$K_{\text{PID}}(q) = K \cdot \left( 1 + \frac{1}{T_i \cdot \frac{1-q^{-1}}{T_e}} + \frac{T_d \cdot \frac{1-q^{-1}}{T_e}}{1 + \frac{T_d}{N} \cdot \frac{1-q^{-1}}{T_e}} \right) \quad (63)$$

En réarrangeant on obtient la forme générale RST (Figure 30):

$$K_{PID}(q) = K \cdot \frac{r_0 + r_1 \cdot q^{-1} + r_2 \cdot q^{-2}}{(1 - q^{-1}) \cdot (1 + s_1' \cdot q^{-1})}$$

$$\begin{cases} r_0 = K \cdot \left( 1 + \frac{T_e}{T_i} + \frac{N \cdot T_d}{T_d + N \cdot T_e} \right) \\ r_1 = -K \cdot \left( 1 + \frac{T_d}{T_d + N \cdot T_e} + \frac{T_e}{T_i} \cdot \frac{T_d}{T_d + N \cdot T_e} + \frac{N \cdot T_d}{T_d + N \cdot T_e} \right) \\ r_2 = K \cdot \left( \frac{T_d}{T_d + N \cdot T_e} \right) \\ s_1' = \frac{T_d}{T_d + N \cdot T_e} \end{cases} \quad (64)$$

Comme on le voit le correcteur obtenu comporte autant de degrés de liberté que le correcteur PID analogique. On reconnaît au dénominateur le terme  $(1 - q^{-1})$  en facteur qui assure l'effet d'intégration.

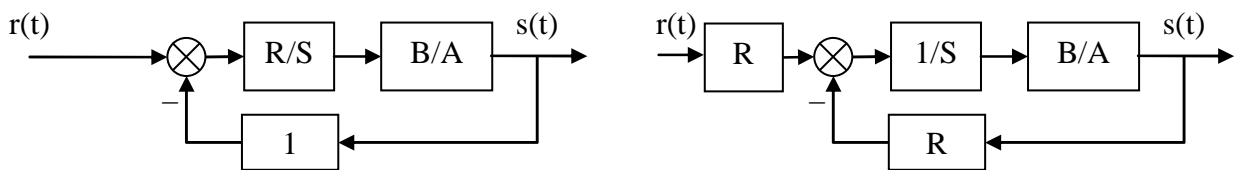


Figure 30 : PID sous forme RST

La fonction de transfert en boucle fermée obtenue est donnée par l'équation suivante :

$$H_{BF} = \frac{B \cdot R}{A \cdot S + B \cdot R} = \frac{B \cdot R}{P} \quad (65)$$

De manière générale le polynôme caractéristique  $P$  fixe les performances visées en boucle fermée. Le calcul de  $R$  et  $S$ ,  $P$  étant fixé et  $A$  et  $B$  étant connus, est réalisé en résolvant l'équation de Bézout  $P = A \cdot S + B \cdot R$ . (voir 9.1 Annexe 1 : Résolution de l'équation polynomiale de Bezout)

On voit que le numérateur  $B \cdot R$  est imposé une fois que  $R$  et  $S$  sont trouvés. Le régulateur introduit donc naturellement des zéros dans la boucle fermée.

Traisons l'exemple entièrement dans le cas où le système est modélisé comme un système du deuxième ordre avec faible retard. La fonction de transfert du procédé est donnée par :



$$H_{BO} = \frac{B}{A} = \frac{b_1 \cdot q^{-1} + b_2 \cdot q^{-2}}{1 + a_1 \cdot q^{-1} + a_2 \cdot q^{-2}} \quad (66)$$

Le correcteur PID rajoutant deux pôles, le polynôme caractéristique en boucle fermée spécifié doit posséder quatre pôles. Le polynôme caractéristique en boucle fermée doit donc être de la forme :

$$\begin{aligned} P &= 1 + p_1 \cdot q^{-1} + p_2 \cdot q^{-2} + p_2 \cdot q^{-3} + p_2 \cdot q^{-4} \\ P &= (1 + p_1' \cdot q^{-1} + p_2' \cdot q^{-2}) \cdot (1 + \alpha_1 \cdot q^{-1}) \cdot (1 + \alpha_2 \cdot q^{-1}) \end{aligned} \quad (67)$$

Et il doit être identifié au produit  $AS + BR$  avec  $R$  et  $S$  ayant les formes données par (64) :

$$\begin{aligned} &(1 + p_1' \cdot q^{-1} + p_2' \cdot q^{-2}) \cdot (1 + \alpha_1 \cdot q^{-1}) \cdot (1 + \alpha_2 \cdot q^{-1}) \\ &= (1 + a_1 \cdot q^{-1} + a_2 \cdot q^{-2}) \cdot (1 - q^{-1}) \cdot (1 + s_1' \cdot q^{-1}) \\ &+ (b_1 \cdot q^{-1} + b_2 \cdot q^{-2}) \cdot (r_0 + r_1 \cdot q^{-1} + r_2 \cdot q^{-2}) \end{aligned} \quad (68)$$

Les coefficients  $p_1'$  et  $p_2'$  définissent un modèle du second ordre aux caractéristiques (pulsation propre  $\omega_0$  et amortissement  $\sigma$  choisis et coefficients déduits conformément au Tableau 3 page 25). Les coefficients  $\alpha_1$  et  $\alpha_2$  correspondent à des pôles auxiliaires aperiodiques. Les choix typiques pour ces coefficients sont :

$$-0.05 < \alpha_1, \alpha_2 < -0.5 \quad (69)$$

On montre que la résolution de (68) est réalisée par une inversion matricielle simple :

$$M \cdot x = p \quad (70)$$

avec

$$\begin{cases} x^T = [1 & s_1' & r_0 & r_1 & r_2] \\ p^T = [1 & p_1 & p_2 & p_3 & p_4] \end{cases}$$

et

$$M = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ a_1' & 1 & b_1 & 0 & 0 \\ a_2' & a_1' & b_2 & b_1 & 0 \\ a_3' & a_2' & 0 & b_2 & b_1 \\ 0 & a_3' & 0 & 0 & b_2 \end{pmatrix} \quad (71)$$

On montre que M est inversible à la condition que A et B soient premiers entre eux, c'est-à-dire que A et B n'aient pas de pôles et de zéros en commun.

(Le problème peut aussi être résolu en utilisant la résolution de l'équation de Bézout).

### 7.3.2 Cas 2

Comme on l'a vu le correcteur PI numérique directement transposé du correcteur PID analogique décrit dans le chapitre 7.3.1 a l'inconvénient de ramener des zéros au dénominateur de la fonction de transfert en boucle fermée (équation 65). Il est possible, grâce à la flexibilité du régulateur numérique d'imposer au correcteur de ne pas ajouter de zéros en boucle fermée. La fonction de transfert désirée en boucle fermée sera donc de la forme :

$$H_{BF} = \frac{P(1)}{B(1)} \cdot \frac{B}{P} \quad (72)$$

Le facteur  $P(1)/B(1)$  garantit le gain statique unitaire du système en boucle fermée (voir page 12). Si le régulateur a la forme générale RST alors les polynômes R S et T doivent vérifier :

$$H_{BF} = \frac{B \cdot T}{A \cdot S + B \cdot R} = \frac{P(1) B}{B(1) P} \quad (73)$$

S et R sont obtenus par la résolution de la même équation que précédemment (68). Le polynôme T est obtenu par simple identification :  $T = R(1)$ .

La structure, sous forme de schéma-bloc est donnée ci-dessous :

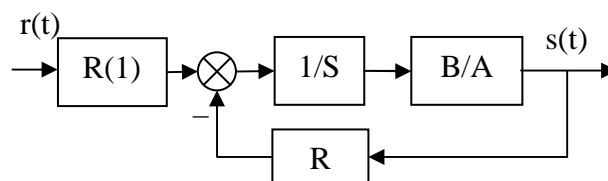


Figure 31 : PID2 sous forme RST

On montre que cette structure de PID, ramené en continu, correspond au PID canonique pour l'action intégrale. L'action proportionnelle et dérivée n'est appliquée que sur le signal y.

La structure du PID2 par rapport au PID1 a pour avantage de ne pas ramener de zéros à la boucle fermée. Cela se traduit par un dépassement moindre en transitoire lorsqu'on applique un échelon de commande. Par contre les deux systèmes réagissent de la même manière à un échelon de perturbation.

## 7.4 Placement de pôles

Les deux structures de PID décrites dans le chapitre précédent sont deux cas particulier d'une stratégie plus générale de placement de pôles.

La synthèse est plus générale que les PID présentés précédemment puisqu'il n'est pas nécessaire de faire d'hypothèses sur le degré de A et B, sur le retard, ni sur la présence ou non de zéros stables ou instables dans le procédé.

La structure du correcteur est la structure type RST telle que présentée dans la Figure 29. Pour la suite des calculs on posera pour la fonction de transfert les conventions suivantes :

$$H_{BO} = \frac{q^{-d} \cdot B(q^{-1})}{A(q^{-1})} \quad (74)$$

$$\begin{cases} A(q^{-1}) = 1 + a_1 \cdot q^{-1} + \dots + a_{n_A} \cdot q^{-n_A} \\ B(q^{-1}) = b_1 \cdot q^{-1} + \dots + b_{n_B} \cdot q^{-n_B} = q^{-1} \cdot B^*(q^{-1}) \end{cases}$$

Compte tenu de la forme RST du régulateur la fonction de transfert en boucle fermée vérifie :

$$H_{BF} = \frac{q^{-d} \cdot B(q^{-1}) \cdot T(q^{-1})}{P(q^{-1})} \quad (75)$$

$$P(q^{-1}) = A(q^{-1}) \cdot S(q^{-1}) + q^{-d} \cdot B(q^{-1}) \cdot R(q^{-1}) = 1 + p_1 \cdot q^{-1} + \dots + p_{n_P} \cdot q^{-n_P}$$

Le polynôme  $P(q^{-1})$  définit donc les pôles de la fonction de transfert en boucle fermée et traduit donc le comportement en *poursuite*.

De la même manière la fonction de transfert entre une perturbation et la sortie (fonction de sensibilité perturbation-sortie) est donnée par :

$$S_{yp} = \frac{A(q^{-1}) \cdot S(q^{-1})}{P(q^{-1})} \quad (76)$$

Ce même polynôme traduit donc aussi le comportement en *régulation*.

Les consignes en poursuite et en régulation peuvent être différentes comme cela est représenté par exemple sur la Figure 32.

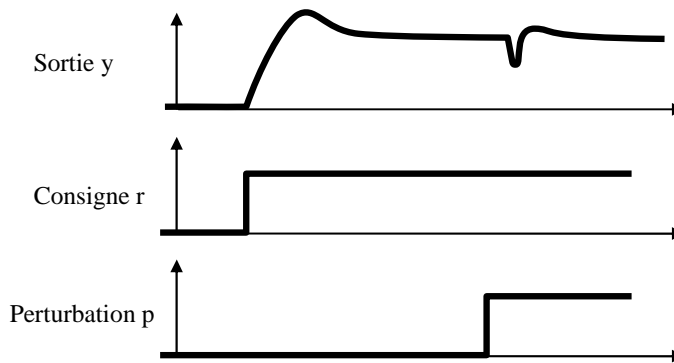


Figure 32 : Performances différentes en régulation et en poursuite

### 7.4.1 Choix des pôles en boucle fermée

De manière générale le comportement en boucle fermée est spécifié sous la forme d'une pulsation propre  $\omega_0$  et d'un amortissement  $\sigma$ , donc une fonction de transfert du second ordre. La spécification étant généralement donnée en continu il est nécessaire de traduire la valeur désirée de  $\omega_0$  et  $\sigma$  en un polynôme  $P_D(q^{-1})$  par discrétisation de la fonction de transfert continue (chapitre 4.5 page 29). Ce polynôme  $P_D$  traduit le comportement dominant du système bouclé.

Si l'on souhaite introduire un effet de filtrage supplémentaire dans certaines bandes de fréquence (réduction des bruits de mesure, adoucissement de la commande, robustesse) cela est décrit par un polynôme  $P_F$  comportant des pôles auxiliaires, plus rapides que les pôles de  $P_D$ .

Le polynôme caractéristique en boucle fermée est donc constitué de pôles dominants et de pôles auxiliaires :

$$P(q^{-1}) = P_D(q^{-1}) \cdot P_F(q^{-1}) \quad (77)$$

### 7.4.2 Régulation : calcul de R et S

Le polynôme  $P(q^{-1})$  étant fixé il s'agit de trouver R et S vérifiant :

$$P(q^{-1}) = A(q^{-1}) \cdot S(q^{-1}) + q^{-d} \cdot B(q^{-1}) \cdot R(q^{-1}) \quad (78)$$

Comme déjà vu au chapitre 7.3.1 on obtient une équation de Bézout qu'il s'agit de résoudre. (Voir aussi 9.1 Annexe 1 : Résolution de l'équation polynomiale de Bezout)

Il est parfois nécessaire de spécifier R et S avec des parties fixes  $H_R$  et  $H_S$ . Dans ce cas l'équation (78) est modifiée et on l'écrit sous la forme:

$$P(q^{-1}) = (A(q^{-1}) \cdot H_S(q^{-1})) \cdot S'(q^{-1}) + (q^{-d} \cdot B(q^{-1}) \cdot H_R(q^{-1})) \cdot R'(q^{-1}) \quad (79)$$

Les inconnues que l'on souhaite trouver sont  $S'$  et  $R'$ . Ceci est nécessaire dans les cas particuliers suivants :

- spécification d'erreur statique nulle,
- rejet d'une perturbation harmonique,
- blocage d'un signal,

Détaillons maintenant ces cas particuliers.

### Spécification d'une erreur statique nulle

On retrouve la nécessité d'avoir un terme intégrateur dans la boucle directe. La spécification se retrouve donc sur  $S$  :

$$S(q^{-1}) = (1 - q^{-1}) \cdot S'(q^{-1}) = H_S(q^{-1}) \cdot S'(q^{-1}) \quad (80)$$

### Rejet d'une perturbation harmonique

Si le système est perturbé par une perturbation harmonique à une fréquence fixe (par exemple 50Hz...) il est possible d'imposer à  $S$  d'avoir un module *nul* ou amorti à cette fréquence. Cela se traduit par la présence dans  $S$  d'une paire de zéros complexe à amortissement nul ou faible à cette fréquence :

$$S(q^{-1}) = (1 + \alpha \cdot q^{-1} + \beta \cdot q^{-2}) \cdot S'(q^{-1}) = H_S(q^{-1}) \cdot S'(q^{-1}) \quad (81)$$

### Blocage d'un signal

Pour certaines application on souhaite que le régulateur "laisse passer" certaines composantes du signal de sortie, c'est-à-dire qu'il ne réagisse pas à ces composantes. Cela se traduit par  $R=0$  à ces fréquences. On retrouve la même forme que précédemment :

$$R(q^{-1}) = (1 + \alpha \cdot q^{-1} + \beta \cdot q^{-2}) \cdot R'(q^{-1}) = H_R(q^{-1}) \cdot R'(q^{-1}) \quad (82)$$

### 7.4.3 Poursuite : calcul de T

Le schéma-bloc de l'asservissement est donné sur la Figure 33. Le terme  $T$  est décomposé en un terme  $T'$  et un terme  $B^*/A^*$ .

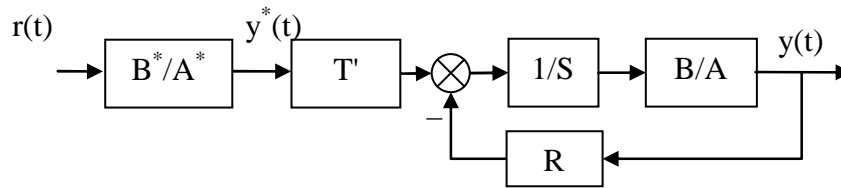


Figure 33 : placement de pôle

L'idée est de régler  $B^*$  et  $A^*$  de telle manière que le signal  $y^*$  ait la dynamique de poursuite désirée (par exemple un second ordre avec pulsation propre et amortissement donné) puis de régler  $T'$  de telle manière que le transfert entre  $y^*$  et  $y$  soit le plus "transparent" possible.

### Réglage de $T'$

Le transfert entre  $y^*$  et  $y$  est :

$$S_{yy^*}(q^{-1}) = T'(q^{-1}) \frac{q^{-d} \cdot B(q^{-1})}{P(q^{-1})} \quad (83)$$

Pour assurer le transfert le plus "transparent" possible il faut donc choisir

$$T'(q^{-1}) = \frac{P(q^{-1})}{B(1)} \quad (84)$$

Remarques :

- Si  $B(1)=0$  alors on remplace  $B(1)$  par 1 dans (84)
- De même dans (84)  $P(q^{-1})$  peut être remplacé par  $P_D$  correspondant à la dynamique dominante.

### Réglage de $B^*$ et $A^*$

$B^*$  et  $A^*$  sont choisis de telle manière que la fonction de transfert  $B^*/A^*$  corresponde à la dynamique de poursuite souhaitée.

## 7.5 Poursuite et régulation à objectifs indépendants

Cette méthode ressemble à la méthode de placement de pôles à ceci près que l'on va effectuer une simplification des zéros de  $B^*(q^{-1})$  : cela ne s'applique donc qu'aux systèmes avec des pôles stables. Plus précisément il faut qu'ils aient un amortissement suffisant ( $> 0,2$ ) de manière à ne pas être trop près du cercle unité.

La structure de l'asservissement est toujours celle donnée sur la Figure 33.

**Choix de P**

$P(q^{-1})$  est choisi de manière à simplifier  $B^*(q^{-1})$  (rappel : la forme de B est donnée dans l'équation 74). Comme indiqué dans le chapitre 7.4.2 page 60 il s'agit donc d'imposer les zéros de  $B(q^{-1})$  dans  $H_R(q^{-1})$  et  $H_S(q^{-1})$ .

**Choix de T**

Si l'on choisit  $T(q^{-1}) = P(q^{-1})$  alors le transfert entre  $y^*$  et  $y$  est :

$$S_{yy^*}(q^{-1}) = q^{-(d+1)} \quad (85)$$

**Réglage de  $B^*$  et  $A^*$** 

$B^*$  et  $A^*$  sont choisis de telle manière que la fonction de transfert  $B^*/A^*$  corresponde à la dynamique de poursuite souhaitée.





## 8. TECHNOLOGIE DE LA COMMANDE PAR ORDINATEUR

Du point de vue de l'automaticien le ordinateur est relié au système physique à piloter par un certain nombre d'intermédiaires plus ou moins gênants :

- les interfaces analogiques et numériques,
- le transport de l'information par bus et protocole,
- l'utilisation éventuelle de capteurs ou d'actionneurs « intelligents ».

Il ne s'agit pas ici de faire un exposé complet sur ces trois aspects mais de poser les problèmes qui se posent à l'automaticien lorsque le système qu'il doit commander est perturbé par ces éléments.

Enfin ce chapitre se termine sur les notions de « Hardware In the Loop » et de « prototypage rapide » qui sont partie intégrante des processus de réalisation de systèmes de contrôle.

### 8.1 Calculateurs et interfaces numériques / analogiques

Le premier problème à prendre en compte est la quantification de l'information. Du point de vue du ordinateur d'une part et du point de vue des convertisseurs d'autre part. Dans un premier temps, considérons les aspects performance de calcul lié à la nature du codage des nombres dans le ordinateur.

#### Les performances liées au types de variables

Prenons l'exemple où le concepteur de la loi de commande souhaite réaliser un correcteur de type proportionnel. Entre l'acquisition des mesures et l'écriture de la consigne il y aura la réalisation d'une de commande du type :

$$y = K \cdot (\text{consigne} - \text{mesure}) \quad (86)$$

En utilisant un langage de programmation évolué (par exemple le C) la ligne de code sera la transcription littérale de (86). Le programmeur devra décider de la définition et du type des variables, principalement du nombre d'octets utilisés pour stocker ses variables. Le Tableau 5 résume par exemple les types utilisés en langage C (norme ANSI)

Tableau 5 : Types ANSI C

Types entiers	char	8 bits
	short int	16 bits
	long int	32 bits
Types flottants (mantisse et exposant)	float	32 bits
	double	64 bits
	longdouble	128 bits

Une fois le code écrit en C l'exécution du programme sur un processeur cible ne peut se faire que par l'intermédiaire d'une compilation du programme. A chaque processeur est donc adapté une version de compilateur.

Les performances du programme (sa rapidité d'exécution) dépendront donc en partie de la qualité du compilateur. (Par exemple, une multiplication s'effectue t'elle bêtement par un décalage de tous les bits ou bien des solutions astucieuses sont-elles implémentées pour arrêter le décalage selon le contexte ?)

Le principal déterminant de la performance est cependant le processeur : son type (8 bits, 16 bits ou 64 bits) et son cadencement (en MHz ou GHz) sont les deux plus importants paramètres. Coder les variables sur 64 bits (double) dans le programme coûtera cher en temps de calcul si le processeur est un 8 bits. Le type de mémoire et leurs temps d'accès peuvent aussi être déterminants. Enfin, certains processeurs disposent d'un processeur spécifiquement dédié aux opérations mathématiques (coprocesseur mathématique).

Il n'est pas possible de dire exactement, pour tel type de compilateur associé à tel type de processeur, le temps que prendra par exemple une multiplication. En pratique on effectue une série de tests types (benchmark) et on affiche un temps moyen d'exécution.

En résumé, on peut être tenté, pour ne pas s'embêter, de coder toutes les variables en virgule flottante. Mais il faut bien savoir que cela a un coût en terme de temps de calcul, surtout si le processeur n'est pas adapté. Pour fixer les idées on peut dire qu'il y a un rapport 20 entre une addition réalisée en "short" et une addition réalisée en virgule flottante.

Enfin, l'utilisation de fonctions mathématiques (sinus, exponentielles, etc...) peut aussi avoir un coût très lourd si le travail n'est pas réalisé par un coprocesseur mathématique dédié. Le compilateur et sa librairie mathématique joue là encore un grand rôle.

### La quantification imposée par le processeur

Au-delà de l'aspect performance et temps de calcul, la quantification de l'information doit parfois aussi être prise en compte. Rappelons qu'un entier codé sur N bits peut prendre seulement  $2^N - 1$  valeurs différentes. Avant d'effectuer les calculs mathématiques dans le corps du programme il est donc judicieux d'appliquer un

facteur d'échelle de manière à utiliser l'amplitude maximale possible (entre 0 et  $2^N - 1$ ).

### **La quantification imposée par les conversions analogique et numérique**

Les convertisseurs analogique-numérique et numérique-analogique sont principalement caractérisés par leur amplitude de signal analogique d'entrée (généralement en volts, par exemple +/- 10V) et par leur résolution en bits (8 ou 10 bits pour les plus courants, jusqu'à 16 bits pour les convertisseurs haut de gamme).

Prenons l'exemple d'un capteur gyromètre transformant une vitesse de rotation en tension. Compte tenu de l'application le signal utile varie entre 2,4V et 2,6V. Si le signal mesuré est branché directement sur un convertisseur 0-10V 10bits alors un problème de conditionnement du signal se pose. En effet la quantification est de  $\frac{10V}{2^{10}-1} = 0,098V$ . Les 0,2V de signal utile seront donc discrétisés en  $\frac{0,2}{0,098} = 20$  intervalles seulement !

Si la discrétisation du signal utile est trop grossière cela peut se traduire par une diminution des performances de l'asservissement, voire une instabilité. Le système n'est en effet plus linéaire et il n'existe pas d'outils théoriques permettant de prévoir les performances du système bouclé.

## **8.2 Transport de l'information : les bus et les protocoles**

Pas question de faire ni même d'ébaucher un cours sur les bus et les protocoles ici. Mais on va quand même donner quelques indications. L'automaticien a l'habitude de simplifier le problème qu'il se pose en dessinant un schéma simplifié de son problème d'asservissement (du type de la Figure 20 page 45) pour pouvoir appliquer les théories de la commande. Cependant il arrive bien souvent que l'information transite à l'intérieur du système, soit entre différents ordinateurs effectuant différentes parties du travail soit entre ordinateur et périphériques, par l'intermédiaire de bus informatique. A chaque type de bus correspond un ou plusieurs protocoles qui peuvent poser des problèmes pour l'automaticien :

- les données sont-elles discrétisées ? (le protocole peut par exemple ne permettre de transmettre que des paquets de données de 16 bits),
- le bus induit-t'il un retard ? (le composant mettant en forme les données du ordinateur pour les publier sur le bus a certaines performances qui peuvent induire un délai)
- le bus est-il déterministe ? (Certains bus ne garantissent pas que les données arriveront en un temps donné),
- le bus est-il sûr ? (Certains bus ne garantissent pas non plus que les données arriveront bien)

### **8.3 Capteurs et actionneurs "intelligents", "smart sensors"**

De plus en plus souvent les capteurs et les actionneurs sont associés à leur électronique de puissance, électronique de commande et de conditionnement de signal. L'interface avec le capteur ou l'actionneur se fait donc via un protocole de communication (par exemple via un bus RS232).

L'intérêt est de véhiculer non pas des signaux analogiques, tensions ou courants, mais des signaux numériques beaucoup moins sensibles aux éventuelles perturbations.

On retrouve donc, entre le capteur ou l'actionneur et le ordinateur, la présence d'un bus et d'un protocole de communication.

### **8.4 La philosophie "hardware in the loop"**

Les systèmes embarqués sont de plus en plus complexes et touchent des domaines de plus en plus variés : automobile, satellites, aéronautique, UAV, systèmes d'arme, sous-marins, systèmes, etc... Les systèmes à piloter ainsi que les systèmes de commande sont de plus en plus complexes.

Les techniques de simulation « Hardware In the Loop » (HIL) sont de plus en plus utilisées dans le développement et dans le test de ces systèmes embarqués. L'idée est de développer entièrement le système de contrôle sur une plateforme de test en utilisant le maximum de composants réels et en ne remplaçant que le minimum par la simulation. Par exemple en aéronautique l'« Iron Bird » regroupe dans un hangar tous les systèmes réels de l'avion en cours de développement. Seule la mécanique du vol est simulée : les sorties des actionneurs sont mesurées, rentrées dans un modèle de simulation numérique, puis les mesures effectuées normalement par les capteurs sont calculées et restituées par le modèle.

L'intérêt est de diminuer de manière importante les coûts et les délais en économisant le plus grand nombre possible de tests sur le système réel en situation. Lors des premiers tests réels les techniques « Hardware In the Loop » permettent d'avoir effectué la plus grande part du développement des lois de commande.

### **8.5 Le prototypage rapide en automatique**

Un peu dans le même ordre d'idée que le « Hardware In the Loop » le prototypage rapide permet de réaliser la conception des lois de commande en simulation puis de transporter rapidement (en un simple click !) le code du contrôleur simulé vers la cible temps réel implémentée sur le système réel.

La société Mathworks [9] investit massivement dans ce domaine en ayant enrichi son logiciel de calcul scientifique (Matlab) avec un logiciel de simulation graphique très évolué (Simulink) puis en développant des outils de prototypage rapide associés (Real Time Workshop, Embedded Target For...).

## 9. ANNEXE

### 9.1 Annexe 1 : Résolution de l'équation polynomiale de Bezout

Il s'agit de trouver R et S vérifiant :

$$P(q^{-1}) = A(q^{-1}).S(q^{-1}) + B(q^{-1}).R(q^{-1})$$

Où P, A et B sont connus.

Dans le cas où P, A, B, R et S sont des nombres entiers la solution de l'algorithme est donnée par l'algorithme de division Euclidienne étendu [8].

Dans le cas où P, A, B, R et S sont des polynômes, l'algorithme est donné ci-dessous, programmé en matlab ([1]).

```
function [Rp,Sp,nrp,nsp]=bezoutd(A,B,Hs,Hr,P)
%function [Rp,Sp,nrp,nsp]=bezoutd(A,B,Hs,Hr,P)
%solves AHsSp+BHrRp=P by coefficient comparison. Delay and discretization delay need to be integrated in B
%(zeros at the beginning).
%
%inputs:
%A=[a0 a1 ... aNa] ... vector of model denominator coefficients A=a0 + a1z^(-1) + a2z^(-2) +...+ aNaz^(-Na)
%B=[b0 b1 ... bNb] ... vector of model numerator coefficients B=b0 + b1z^(-1) + b2z^(-2) +...+ bNbz^(-Nb)
%Hs=[hs0 hs1 ... hsNhs] ... vector of controller denominator fixed part Hs=hs0 + hs1z^(-1) +...+ hsNhsz^(-Nhs)
%Hr=[hr0 hr1 ... hrNhr] ... vector of controller denominator fixed part Hr=hr0 + hr1z^(-1) +...+ hrNhrz^(-Nhr)
%P=[p0 p1 ... pNp] ... vector of desired polynomial coefficients P=p0 + p1z^(-1) + p2z^(-2) +...+ pNpz^(-Np)
%outputs:
%Rp=[rp0 rp1 rp2 ...] ... vector of coefficients for resulted controller numerator
%Sp=[sp0 sp1 sp2 ...] ... vector of coefficients for resulted controller denominator
%nrp ... order of Rp
%nsp ... order of Sp
%
%written by: J. Langer, I.D. Landau, H. Prochazka
%7th june 2002

PRECISION=1e-16;

D=size(A);
if D(1)>1, A=A'; end;
```

```

D=size(B);
if D(1)>1, B=B'; end;
D=size(Hs);
if D(1)>1, Hs=Hs'; end;
if D(1)==0, Hs=1; end;
D=size(Hr);
if D(1)>1, Hr=Hr'; end;
if D(1)==0, Hr=1;end;
D=size(P);
if D(1)>1, P=P'; end;

na=length(A)-1;
nb=length(B)-1;
np=length(P)-1;
nhs=length(Hs)-1;
nhr=length(Hr)-1;

if (nhs>0), Ah=real(convz(A,Hs)); else Ah=A*Hs; end; % Ah = A * Hs
nah=length(Ah) -1;
if (nhr>0), Bh=real(convz(B,Hr)); else Bh=B*Hr; end;
nbh=length(Bh) -1;
if (np>nah+nbh-1), disp('Bezout error: too many poles');end;

% increase size of P using zeros if necessary
if (np<nah+nbh-1),
%
% set remaining poles onto a circle with radius rmin
P=[P zeros(1,nah+nbh-1-np)];
rootsPdes=roots(P);
nextra=nah+nbh-1-np;
rmin=1e-16;
angle=[0:nextra-1]'/nextra*2*pi;
j=sqrt(-1);
rootsPextra=rmin*exp(j*angle);
P=poly([rootsPdes;rootsPextra]);
np=nah+nbh-1;

end;
P,

nsp=nbh-1;
nrp=nah-1;

%matrix is smaller than vector PD

```

```

if (np>nah+nbh-1),
    disp('Order of model denominator is too low! Add a polynom of higher order to Hs or Hr. ');
end;

% ns=nsp+nhs
% nr=nrp+nhr

M=[];
for j=1:nsp+1,
    V=[];
    if (j>1), V=[V ; zeros(j-1,1)]; end;% zeros in front of Ah
    V=[V ; Ah'];% Ah
    if (j<=nsp), V=[V ; zeros(nsp+1-j,1)]; end;% zeros behind Ah
    if (length(V)~=nah+nbh), disp('bezoutb: error V'); end;
    M=[M V]; % add one column to M
end;

for j=1:nrp+1,
    V=[];
    if (j>1), V=[V ; zeros(j-1,1)]; end;
    V=[V ; Bh'];
    if (j<=nrp), V=[V ; zeros(nrp+1-j,1)]; end;
    if (length(V)~=nah+nbh), disp('bezoutb: error V'); end;
    M=[M V];
end;

D=size(M);
if (D(1)~=nah+nbh), disp('bezoutb: error size M row'); end;
if (D(2)~=nah+nbh), disp('bezoutb: error size M column'); end;

% make P column vector
P=P';

global M1;
M1=M;

X= M\P;
% coefficients are real values
X=real(X);

% make X row vector
X=X';

```

## Commande des systèmes par ordinateur

```
Sp=X(1:nsp+1);  
Rp=X(nsp+2:nsp+nrp+2);
```



## 10. BIBLIOGRAPHIE

- [1] I. D. Landau, "*Commande des systèmes*", Edition Lavoisier 2002.
- [2] C. Nouals, "Théorie du Signal", Notes de cours ENSICA
- [3] F. Vincent, "*Traitement Numérique du Signal*", Notes de cours ENSICA.
- [4] D. Arzelier, "*Représentation et analyse des systèmes, notes de cours*", Edition ENSICA. Voir aussi <http://www.laas.fr/~arzelier/>
- [5] J. Bordeneuve-Guibé, "*Commande des systèmes linéaires*", Notes de cours ENSICA.
- [6] Y. Briere, "*Identification*", Notes de cours ENSICA.
- [7] L. Ljung, "*System Identification, Theory for the User*", Prentice Hall 1987.
- [8] Wikipedia , équation de Bezout : [http://en.wikipedia.org/wiki/Bézout's\\_identity](http://en.wikipedia.org/wiki/Bézout's_identity)
- [9] <http://www.mathworks.com>



# 11. INDEX

<b>A</b>	
actionneur intelligent .....	21
amortissement .....	15
ANSI C .....	60
Approximation numérique d'équation différentielle .....	28
ARMAX .....	35
<b>B</b>	
BJ35 .....	
Blocage d'un signal .....	56
bloqueurs d'ordre 0 .....	22
BOZ .....	22
bus.....	21, 61
<b>C</b>	
CAN.....	21
capteur intelligent .....	21
causal .....	6, 9
Causalité .....	10
CNA.....	21
compilateur .....	60
convertisseur analogique / numérique.....	21
convertisseur numérique / analogique.....	21
convertisseurs analogique-numérique .....	61
coprocesseur mathématique .....	60
Correspondances pôles - zéros.....	31
<b>D</b>	
Diagramme de Bode .....	18
Discrétisation de fonction de transfert continues .....	27
division polynomiale.....	6
<b>E</b>	
échelon unitaire.....	11
équation d'état .....	17
Erreur statique.....	42
Estimation de la réponse fréquentielle .....	39
Estimation de la réponse impulsionnelle.....	38
Euler .....	29
<b>F</b>	
facteur d'échelle .....	61
facteur d'oubli .....	36
<i>fenêtre de Hamming</i> .....	39
filtre anti-repliement .....	25
fonction de sensibilité .....	42
fonction de transfert discrète.....	9
<b>G</b>	
gain statique.....	11
gain statique unitaire.....	42
<b>I</b>	
Identification.....	32
Identification non paramétrique .....	38
intercorrélation.....	38
<b>L</b>	
lieu des racines.....	43
lieu d'Evans.....	43
LTI.....	8
<b>M</b>	
marge de gain.....	47
marge de module.....	46
marge de phase .....	47
méthode des moindres carrés récursifs .....	32
Méthode des trapèzes.....	29
modèle ARX .....	32
<b>O</b>	
OE.....	35
one step ahead prediction.....	35
opérateur de retard .....	9
<b>P</b>	
période d'échantillonnage (choix) .....	26
Placement de pôles .....	54
Pôles .....	11
Poursuite .....	56
<i>prédiction</i> .....	32
protocoles.....	61
Pseudo Random Binary Signal .....	34
pulsation .....	15
<b>Q</b>	
quantification .....	23, 24, 60
<b>R</b>	
R,S,T.....	41
rapidité.....	15
rayon de convergence .....	5, 9
Régulation.....	55

Rejet de perturbations .....	42
Rejet d'une perturbation harmonique .....	56
réponse à un échelon.....	11
réponse fréquentielle.....	18
Réponse impulsionnelle.....	10
Réponses indicielle .....	10
retard.....	23
root locus .....	43

### S

signal discret .....	5
smart sensors.....	62
stabilité.....	11
Systèmes du premier ordre.....	14
Systèmes du second ordre.....	15

### T

Théorème de Nyquist.....	46
théorème de Shannon.....	19, 24
théorème du retard .....	6
Transformée en z .....	5
transformée unilatérale .....	5
Tustin.....	29

### V

virgule flottante.....	60
------------------------	----

### Z

zero holder hold .....	23
zéros.....	11
zoh .....	23

# Commande des systèmes par ordinateur

Yves Briere  
ENSICA

Notes de cours  
2007



# TABLE DES MATIERES

Table des matières .....	1
1. Introduction .....	1
2. Rappels et prérequis .....	3
3. Signaux et systèmes discrets .....	5
3.1 Signaux discrets .....	5
3.1.1 Transformée en z .....	5
3.1.2 Signaux types .....	7
3.2 Systèmes discrets .....	9
3.2.1 Fonction de transfert discrète, définition .....	9
3.2.2 Notation "opérateur de retard $q^{-1}$ " .....	10
3.2.3 Réponse impulsionnelle .....	11
3.2.4 Causalité .....	11
3.2.5 Réponses indicielle .....	11
3.2.6 Régime stationnaire (gain statique) .....	12
3.2.7 Pôles, zéros, stabilité .....	12
3.2.8 Systèmes du premier ordre .....	15
3.2.9 Systèmes du second ordre .....	16
3.2.10 Calcul itératif de la réponse d'un système .....	18
3.2.11 Représentation par équation d'état .....	18
3.3 Réponse fréquentielle des systèmes échantillonnés .....	19
3.3.1 Diagramme de Bode d'un système échantillonné .....	19
3.3.2 Où l'on voit apparaître le théorème de Shannon .....	20
4. Systèmes continus échantillonnés .....	23
4.1 Structure des systèmes commandés par ordinateur .....	23
4.2 Échantillonnage des systèmes continus .....	24
4.3 Effets de l'échantillonnage .....	25
4.4 Où l'on revoit le théorème de Shannon .....	26
4.5 Discrétisation de fonction de transfert continues .....	29
4.5.1 Approximation numérique d'équation différentielle .....	30
4.5.2 Superposition des réponses indicielles .....	32
4.5.3 Correspondances pôles - zéros .....	33
4.5.4 Bilan .....	33
5. Identification des systèmes échantillonnés .....	35
5.1 Identification d'un modèle ARX par la méthode des moindres carrés récurrents .....	35
5.2 L'identification paramétrique en général et en pratique .....	37

5.2.1	Choix du signal d'excitation, traitement des mesures .....	37
5.2.2	Choix d'une structure de modèle.....	38
5.2.3	Choix d'un critère à minimiser.....	38
5.2.4	Choix d'un algorithme.....	38
5.2.5	Validation.....	39
5.2.6	Utilisation de logiciels .....	39
5.3	Identification "simple" .....	39
5.4	Identification non paramétrique .....	41
5.4.1	Estimation de la réponse impulsionnelle .....	41
5.4.2	Estimation de la réponse fréquentielle.....	42
6.	Systèmes bouclés .....	45
6.1	Fonction de transfert en boucle fermée .....	45
6.2	Erreur statique .....	46
6.3	Rejet de perturbations.....	46
6.4	Stabilité des systèmes à simple gain de rétroaction.....	47
6.5	Analyse des systèmes bouclés dans le domaine fréquentiel : stabilité.....	49
6.5.1	Théorème de Nyquist dans le cas où le système est stable en boucle ouverte (critère du revers) .....	50
6.5.2	Théorème de Nyquist dans le cas où le système est instable en boucle ouverte .....	50
6.5.3	Interprétations du lieu de Nyquist.....	50
7.	Méthodes de régulation des systèmes échantillonnés .....	53
7.1	Introduction .....	53
7.2	Structure d'un système commandé par ordinateur, hypothèses.....	54
7.2.1	Discretisation, choix de la période d'échantillonnage.....	54
7.2.2	Discretisation d'un correcteur continu .....	55
7.3	Régulateur PID numérique .....	55
7.3.1	Cas 1 .....	55
7.3.2	Cas 2 .....	58
7.4	Placement de pôles .....	59
7.4.1	Choix des pôles en boucle fermée .....	60
7.4.2	Régulation : calcul de R et S.....	60
7.4.3	Poursuite : calcul de T .....	61
7.5	Poursuite et régulation à objectifs indépendants .....	62
8.	Technologie de la commande par ordinateur.....	65
8.1	Calculateurs et interfaces numériques / analogiques.....	65
8.2	Transport de l'information : les bus et les protocoles .....	67
8.3	Capteurs et actionneurs "intelligents", "smart sensors" .....	68
8.4	La philosophie "hardware in the loop" .....	68
8.5	Le prototypage rapide en automatique .....	68
9.	Annexe .....	69
9.1	Annexe 1 : Résolution de l'équation polynomiale de Bezout.....	69
10.	Bibliographie.....	73
11.	Index.....	1







# 1. INTRODUCTION



Figure 1 : commande d'un systèmes de télémédecine par calculateur, vu par Claude Serre



## 2. RAPPELS ET PREREQUIS

Ce polycopié ne comporte pas de rappels sur les systèmes continus. S'adressant aux étudiants en deuxième année de l'ENSICA il suppose connu les enseignements de première année et de seconde année en automatique et traitement du signal.

En première année le cours de théorie du signal [2] a rendu familier les notions de :

- transformée de Fourier, transformée de Laplace
- corrélation, convolution

Le cours de traitement numérique du signal [3] a abordé en plus :

- la discrétisation et la quantification des signaux.
- la transformée en z et la transformée de Fourier discrète.
- les filtres numériques RIF et RII.

En automatique [4] les bases de la représentation et de l'analyse des systèmes continus ont été posées. On ne reviendra pas dans ce cours sur les notions de :

- modèles mathématique des systèmes linéaires à temps continus à temps invariant (LTI) : fonction de transfert, équation d'état, pôles, zéros et dynamique.
- commandabilité et observabilité.
- stabilité des systèmes en boucle ouverte : pôles, critères de Routh-Hurwitz.
- stabilité des systèmes bouclés : critère de Nyquist, critère de Bode.
- lieu des racines.

En deuxième année les notions suivantes sont acquises [5] :

- synthèse des correcteurs par méthodes fréquentielles (avance et retard de phase, PID)
- commande par retour d'état (placement de pôles), commande Linéaire Quadratique (LQG)
- reconstruction de l'état par Observateur, observateur optimal de Kalman.



## 3. SIGNAUX ET SYSTEMES DISCRETS

### 3.1 Signaux discrets

#### 3.1.1 Transformée en z

Pour une approche mathématiquement rigoureuse de représentation des signaux discrets par la transformée en z consulter [3] et sa bibliographie. Dans ce cours on se cantonne à une approche utilitariste des outils et méthodes mathématiques tout en essayant de ne pas contredire fondamentalement la rigueur mathématique.

Un signal discret  $s(k)$  est une séquence de nombres réels appelés échantillons. L'indice  $k$  est généralement associé à un instant d'échantillonnage :  $t = k.T_e$ . La période d'échantillonnage  $T_e$  est supposée constante. Par définition la transformée en z du signal  $s(k)$  est la fonction de la variable  $z$ , notée abusivement  $s(z)$  :

$$s(z) = Z(s(k)) = \sum_{k=0}^{\infty} s(k) \cdot z^{-k} \quad (1)$$

La transformée définie ici est plus précisément appelée transformée unilatérale puisque les termes correspondant à  $k < 0$  ne sont pas pris en compte. Ceci n'a pas d'importance puisque les signaux que nous utilisons sont systématiquement causaux<sup>1</sup> :  $s(k) = 0$  pour  $k < 0$ .

Il y a bien évidemment un lien avec la transformée de Laplace et la transformée de Fourier (voir [3])

La fonction de la variable complexe  $s(z)$  est une série et ne converge pas forcément pour toute valeur de  $z$ . En toute rigueur il faudrait associer le rayon de convergence à chaque fonction en  $z$  :  $\{s(z) \text{ converge pour } |z| > \alpha\}$ .

Les automaticiens pour qui la transformée en z n'est bien souvent qu'une façon bien commode de regrouper des termes de récurrence utilisent souvent la pseudo-variable<sup>2</sup>  $q$ .

A partir d'une transformée en z  $s(z)$  il est possible de retrouver la séquence  $s(k)$  correspondante en appliquant la transformation inverse, dont on ne donnera pas la formule ici. A la place de faire appel à cette formule compliquée il est possible d'utiliser d'autres méthodes pour retrouver la séquence  $s(k)$  lorsque la fonction  $s(z)$  est donnée sous forme de fraction rationnelle :

<sup>1</sup> Pour une justification du terme « causal » voir page 11

<sup>2</sup> Voir page 10

- division polynomiale
- décomposition de la fraction rationnelle en éléments simples puis utilisation de transformations élémentaires telles que celles données dans le Tableau 2

La propriété fondamentale de la transformée en z est qu'elle est linéaire (la transformée en z d'une composition linéaire de signaux est la composition linéaire de la transformée en z de chaque signal). Les autres propriétés fondamentales sont données dans le Tableau 2.

Tableau 1 : propriétés de la transformée en z

1	$s(z) = Z(s(k)) = \sum_{k=0}^{\infty} s(k) \cdot z^{-k}$
2	$Z(\alpha \cdot s_1(k) + \beta \cdot s_2(k)) = \alpha \cdot Z(s_1(k)) + \beta \cdot Z(s_2(k))$
3	$Z(k \cdot s(k)) = -z \cdot \frac{ds(z)}{dz}$
4	$Z(c^k \cdot s(k)) = s\left(\frac{z}{c}\right)$
5	$Z(s(k-1)) = s(k=-1) + z^{-1} \cdot s(z)$
6	$Z(s(k+1)) = z \cdot s(z) - z \cdot s(k=0)$
7	$s(k=0) = \lim_{z \rightarrow \infty} (s(z))$
8	$s(k \rightarrow \infty) = \lim_{z \rightarrow 1} \left( \frac{z-1}{z} \cdot s(z) \right)$

Remarque 1 : la propriété 5 ne prend pas en compte le fait que le signal est causal : en pratique  $s(k=-1) = 0$ .

Remarque 2 : la propriété 5 justifie le fait que le terme  $z^{-1}$  soit appelé "opérateur de retard unitaire", cette propriété est aussi appelée "théorème du retard".

Remarque 3 : les propriétés 7 et 8 sont données sous réserve de convergence.

Transformées inverses

### Exemple 1 : transformée inverse par division polynomiale

On considère un signal  $s(k)$  dont la transformée en z est donnée par la fraction rationnelle :

$$S(z) = \frac{0.1 \cdot z^2 + 0.1 \cdot z}{z^3 - 2.6 \cdot z^2 + 2.2 \cdot z - 0.6} \quad (3.1)$$

L'objectif est d'obtenir  $S(z)$  sous forme polynomiale :



$$S(z) = x_0 + x_1 \cdot z^{-1} + x_2 \cdot z^{-2} + \dots \text{etc.} \tag{3.2}$$

Cette suite est généralement infinie, la réponse temporelle est donnée par la succession des termes  $\{x_i\}$ . En traitant l'équation (3.1) comme une simple division l'algorithme de division classique (celui appris en CM2) permet d'obtenir par récurrence tous les termes de la suite (3.2) :

$0.1 \cdot z^2$	$+ 0.1 \cdot z$				$z^3 - 2.6 \cdot z^2 + 2.2 \cdot z - 0.6$
$0.1 \cdot z^2$	$- 0.26 \cdot z$	$+ 0.22$	$- 0.06 \cdot z^{-1}$		
0	$0.36 \cdot z$	$- 0.22$	$0.06 \cdot z^{-1}$		
	$0.36 \cdot z$	$- 0.936$	$0.792 \cdot z^{-1}$	$- 0.216 \cdot z^{-2}$	
0	$0.716$	$- 0.732 \cdot z^{-1}$	$\dots$	$\dots$	

Les éléments successifs de la suite  $s(k)$  sont donc  $s(k) = \{0, 0.1, 0.36, 0.716, \dots\}$ . Pour obtenir les termes suivants il faudrait continuer la division.

**Exemple 2 : transformée inverse par décomposition en éléments simples**

La deuxième méthode présentée consiste à décomposer la fraction rationnelle (3.1) en éléments simples puis d'utiliser les tables de transformées en  $z$  (Tableau 2). Pour notre exemple cela donne :

$$S(z) = \frac{0.1 \cdot z^2 + 0.1 \cdot z}{z^3 - 2.6 \cdot z^2 + 2.2 \cdot z - 0.6} = \frac{-z}{z-1} + \frac{0.5 \cdot z}{(z-1)^2} + \frac{z}{z-0.6}$$

La réponse temporelle est donc composée des trois éléments :

$$s(k) = -(1 \ 1 \ 1 \ \dots) + 0.5 \cdot (1 \ 2 \ 3 \ \dots) + (1 \ 0.6 \ 0.6^2 \ \dots)$$

$$\text{soit : } s(k) = \begin{cases} s(0) = -1 + 0.5 + 1 \\ s(2) = -1 + 0.5 \cdot 2 + 0.6 \cdot 0.5 \\ s(3) = -1 + 0.5 \cdot 3 + 0.6^2 \\ s(4) = \dots \end{cases}$$

**3.1.2 Signaux types**

Le Tableau 2 donne les transformée en  $z$  de certains signaux fondamentaux. La Figure 2 représente l'allure de quelques signaux types.

Tableau 2 : transformée en z élémentaires

$s(k)$	$S(z)$
signal temporel	Transformée en z
$\delta(k)$ : impulsion unité	1
$u(k)$ : échelon unité	$\frac{z}{z-1}$
$k \cdot u(k)$	$\frac{z}{(z-1)^2}$
$c^k \cdot u(k)$	$\frac{z}{z-c}$
$k \cdot a^k \cdot u(k)$	$\frac{a \cdot z}{(z-a)^2}$
$\sin(\omega \cdot k) \cdot u(k)$	$\frac{z \cdot \sin(\omega)}{z^2 - 2 \cdot z \cdot \sin(\omega) + 1}$
$\cos(\omega \cdot k) \cdot u(k)$	$\frac{z \cdot (z - \cos(\omega))}{z^2 - 2 \cdot z \cdot \sin(\omega) + 1}$
$a^k \cdot \sin(\omega \cdot k) \cdot u(k)$	$\frac{a \cdot z \cdot \sin(\omega)}{z^2 - 2 \cdot a \cdot \cos(\omega) \cdot z + a^2}$
$a^k \cdot \cos(\omega \cdot k) \cdot u(k)$	$\frac{z \cdot (z - a \cdot \cos(\omega))}{z^2 - 2 \cdot a \cdot \cos(\omega) \cdot z + a^2}$

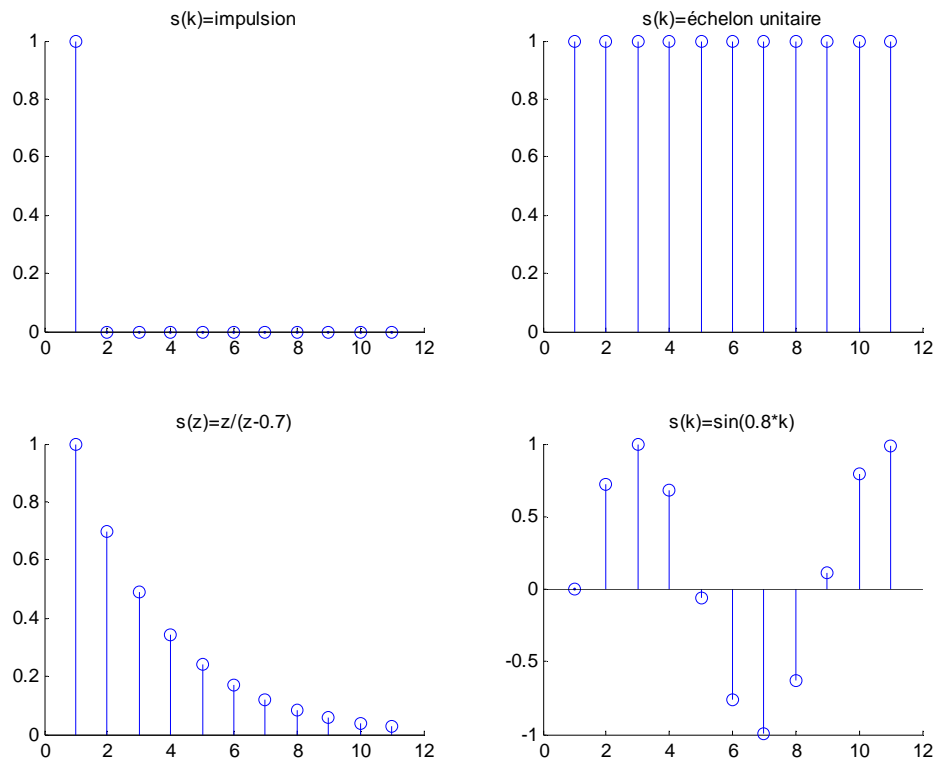


Figure 2 : réponses de quelques signaux types

## 3.2 Systèmes discrets

### 3.2.1 Fonction de transfert discrète, définition

On considère maintenant un système discret linéaire et invariant dans le temps (LTI) représenté sur la Figure 3.

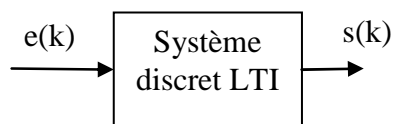


Figure 3 : système discret, notations

La relation entre la séquence d'entrée et la séquence de sortie est linéaire et définie par un ensemble de coefficients constants  $a_{1:n}$  et  $b_{1:m}$  :

$$\begin{aligned} s(k) + a_1 \cdot s(k-1) + a_2 \cdot s(k-2) + \dots + a_n \cdot s(k-n) = \\ b_0 \cdot e(k) + b_1 \cdot e(k-1) + b_2 \cdot e(k-2) + \dots + b_m \cdot e(k-m) \end{aligned} \quad (2)$$

On peut appliquer la transformée en  $z$  à chaque membre de l'équation puis en appliquant en récurrence la propriété 5 (théorème du retard) on obtient la fonction de transfert discrète du système :

$$\frac{s(z)}{e(z)} = F(z) = \frac{b_0 + b_1 \cdot z^{-1} + b_2 \cdot z^{-2} + \dots + b_m \cdot z^{-m}}{1 + a_1 \cdot z^{-1} + a_2 \cdot z^{-2} + \dots + a_n \cdot z^{-n}} \quad (3)$$

Remarque 1 : la fonction de transfert précédente est normalisée en prenant  $a_0 = 1$ .

Remarque 2 : on remarquera que la fonction de transfert est donnée en  $z^{-1}$  plutôt qu'en  $z$ . Cela est justifié par le fait que celle-ci traduit l'équation d'évolution du système physique causal (2).

Le script Matlab permettant de créer un objet Matlab correspondant à une fonction de transfert discrète est donné ci-dessous.

#### Matlab

```
% Création d'un système linéaire discret :
Te=0.1;
sysd = tf([b0 b1 b2],[1 a0 a2 a3],Te,'variable','z^-1');
```

On notera l'utilisation de l'argument 'variable', 'z^-1'.

### 3.2.2 Notation "opérateur de retard $q^{-1}$ "

La fonction de transfert est une fonction de la variable complexe  $z$ . Il faudrait donc en toute rigueur l'associer à son rayon de convergence. En pratique on ne s'embarrasse jamais de préciser cette zone de convergence. C'est pourquoi on trouve souvent la notation utilisant l'opérateur  $q^{-1}$  de décalage à la place de la variable complexe  $z^{-1}$  (4).  $F(q)$  n'est alors plus à proprement parler une fonction mais on lui appliquera les mêmes manipulations algébriques qu'à  $F(z)$ .

$$F(q) = \frac{b_0 + b_1 \cdot q^{-1} + b_2 \cdot q^{-2} + \dots + b_m \cdot q^{-m}}{1 + a_1 \cdot q^{-1} + a_2 \cdot q^{-2} + \dots + a_n \cdot q^{-n}} \quad (4)$$

Attention : selon les auteurs l'opérateur qui remplace  $z^{-1}$  est noté  $q$  (Matlab) ou  $q^{-1}$  (Landau [1]). Dans la suite du document c'est cette dernière convention qui sera appliquée.

L'utilisation de l'opérateur de retard  $q^{-1}$  à la place de la variable complexe  $z^{-1}$  permet de traiter du cas des fonctions de transfert à coefficients non constant, cas non prévu par la transformation en  $z$ .

### 3.2.3 Réponse impulsionnelle

Essayons d'appliquer au système un signal d'entrée particulier : une impulsion unitaire. La transformée en  $z$  du système est obtenue en multipliant la fonction de transfert par la transformée en  $z$  de l'impulsion unitaire : 1. On obtient :

$$s(z) = \frac{b_0 + b_1 \cdot z^{-1} + b_2 \cdot z^{-2} + \dots + b_m \cdot z^{-m}}{1 + a_1 \cdot z^{-1} + a_2 \cdot z^{-2} + \dots + a_n \cdot z^{-n}} \times 1 \quad (5)$$

Cela justifie le fait que la fonction de transfert du système peut aussi s'appeler "réponse impulsionnelle".

#### Matlab

```
% Réponse impulsionnelle d'un système :
[y,t] =impulse(sysd);
```

### 3.2.4 Causalité

Un système est dit causal si les sorties ne dépendent que du passé et non pas du futur. La conséquence est que la réponse impulsionnelle de  $F(z)$  ne peut pas comporter de terme non nuls pour  $k < 0$ . Cela justifie la confusion sémantique entre "système causal" et "signal causal", c'est-à-dire ne comportant pas de termes non nuls pour  $k < 0$ .

Attention : à la différence des systèmes continus un système physique (causal) discret peut très bien être modélisé avec un degré  $m$  supérieur au degré  $n$  : le système peut avoir plus de zéros que de pôles.

La transformée en  $z$  représentant des signaux échantillonnés elle peut tout aussi bien représenter les mesures prises à différents endroits d'une poutre. Le temps étant remplacé ici par une abscisse la notion de causalité ne s'applique pas ! Attention : beaucoup des propriétés de systèmes échantillonnés que l'on étudie sont liés à la causalité, prudence donc si l'on applique ces méthodes à d'autres domaines !

### 3.2.5 Réponses indicielle

Pour caractériser des systèmes on s'intéresse souvent à leur réponse impulsionnelle ou, plus souvent, à leur réponse indicielle. Pour calculer la réponse

à un échelon d'un système il faut donc multiplier la fonction de transfert par la transformée en z d'un échelon unitaire :

$$s(z) = \frac{b_0 + b_1 \cdot z^{-1} + b_2 \cdot z^{-2} + \dots + b_m \cdot z^{-m}}{1 + a_1 \cdot z^{-1} + a_2 \cdot z^{-2} + \dots + a_n \cdot z^{-n}} \times \frac{1}{1 - z^{-1}} \quad (6)$$

Il faudrait ensuite calculer la transformée en z inverse de s(z). En pratique, si l'on doit faire le calcul à la main on décompose en éléments simples puis on utilise des tables. Avec Matlab c'est plus simple. Voici par exemple le script pour obtenir les 20 premiers échantillons de la réponse indicielle d'un système :

**Matlab**

```
% Réponse indicielle d'un système :
[y,t] =step(sysd,20);
```

### 3.2.6 Régime stationnaire (gain statique)

Le gain statique est obtenu en appliquant une entrée indicielle et en calculant la valeur du régime permanent. En appliquant le théorème de la valeur finale il vient :

$$F_0 = \lim_{z \rightarrow 1} \left( \frac{z-1}{z} \cdot \frac{b_0 + b_1 \cdot z^{-1} + b_2 \cdot z^{-2} + \dots + b_m \cdot z^{-m}}{1 + a_1 \cdot z^{-1} + a_2 \cdot z^{-2} + \dots + a_n \cdot z^{-n}} \times \frac{1}{1 - z^{-1}} \right) \quad (7)$$

Soit :

$$F_0 = \frac{\sum_{i=0}^{i=m} b_i}{1 + \sum_{i=1}^{i=n} a_i} \quad (8)$$

Le gain statique est obtenu donc simplement en faisant le quotient de la somme des coefficients.

### 3.2.7 Pôles, zéros, stabilité

Comme pour les systèmes continus la dynamique et la stabilité des systèmes dépend des propriétés du dénominateur de la fonction de transfert.

$$F(z) = \frac{e_0 \cdot (1 + e_1 \cdot z^{-1}) \cdot (1 + e_2 \cdot z^{-1}) \cdots (1 + e_m \cdot z^{-1})}{(1 + c_1 \cdot z^{-1}) \cdot (1 + c_2 \cdot z^{-1}) \cdots (1 + c_n \cdot z^{-1})} \quad (9)$$

Les coefficients  $c_i$  sont les pôles de la fonction de transfert, les coefficients  $e_i$  sont les zéros. La fonction de transfert étant au départ composée de coefficients purement réels les coefficients  $c_i$  sont soit réels soit imaginaires par paires conjuguées.

Une fonction de transfert est donc aussi bien définie par la donnée de ses pôles et de ses zéros. Le script Matlab pour créer un système ainsi est donné ci-dessous :

#### Matlab

```
% Système discret défini par ses pôles et zéros :
Te=0.1;
sysd=zpk([c1 c2 c3],[e1 e2],e0,Te);
```

Si l'on décompose la fonction de transfert en éléments simples on obtient une somme d'éléments du type :

$$F(z) = \frac{d_1}{1 + c_1 \cdot z^{-1}} + \frac{d_2}{1 + c_2 \cdot z^{-1}} + \cdots + \frac{d_n}{1 + c_n \cdot z^{-1}} \quad (10)$$

Un système linéaire temps invariant est dit stable si sa réponse impulsionnelle tend asymptotiquement vers 0. Compte tenu de (10) cela nécessite que chaque pôle  $c_i$  soit de module inférieur à 1.

**Propriété** : un système discret LTI est stable si et seulement si tous ses pôles ont leur modules inférieur ou égal à 1.

#### Critère algébrique attestant de la stabilité : critère de Routh

On rappelle que le critère de Routh permet de prédire si les racines d'un polynôme sont toutes à partie réelle négative, donc stables. Énonçons le théorème de Routh dans le cas d'un polynôme  $P'(w)$  :

$$P'(w) = a_0 + a_1 \cdot w + a_2 \cdot w^2 + \cdots + a_n \cdot w^n \quad (11)$$

Dans un premier temps on construit le tableau :

$$\left| \begin{array}{cccccc} \beta_{n,0} & \beta_{n,1} & \beta_{n,2} & \beta_{n,q-1} & \beta_{n,q} & 0 \\ \beta_{n-1,0} & \beta_{n-1,1} & \beta_{n-1,2} & \beta_{n-1,q-1} & \beta_{n-1,q-1} & 0 \\ \beta_{n-2,0} & \beta_{n-2,1} & \beta_{n-2,2} & \beta_{n-2,q-1} & \beta_{n-2,q-1} & \\ \\ \beta_{2,0} & \beta_{2,1} & 0 & & & \\ \beta_{1,0} & 0 & & & & \\ \beta_{0,0} & 0 & & & & \end{array} \right| \quad (12)$$

Puis on vérifie que tous les premiers éléments de la première colonne ont le même signe : cela garantit que tous les pôles de  $P'(w)$  ont leur partie réelle négative.

Les éléments du tableau sont construits par récurrence ligne par ligne en appliquant les formules suivantes :

Pour  $i = n$  et  $i = n - 1$  :

$$\begin{cases} \beta_{n,j} = a_{n-2j} \\ \beta_{n-1,j} = a_{n-1-2j} \end{cases} \quad (13)$$

Et pour  $i < n-1$  :

$$\beta_{i,j} = \frac{\beta_{i+1,0} \cdot \beta_{i+2,j+1} - \beta_{i+1,j+1} \cdot \beta_{i+2,0}}{\beta_{i+1,0}} \quad (14)$$

Dans le cas qui nous intéresse, un système échantillonné, on cherche plutôt à vérifier que les pôles du polynôme  $P(z)$  sont de module inférieur à un. L'idée est d'effectuer un changement de variable  $z \rightarrow w$  pour pouvoir appliquer le critère de Routh.

$$P'(w) = (1-w)^n \cdot P\left(\frac{1+w}{1-w}\right) \quad (15)$$

Si l'on effectue la transformation (15) alors le fait que  $P'(w)$  ait toutes ses racines à partie réelle négative garantit que  $P(z)$  a toutes ses racines dans le cercle unité. Pour vérifier la stabilité de  $P(z)$  il suffit donc d'effectuer le changement de variable (15) puis d'appliquer le critère de Routh conventionnel (12 13 14).

### Critère algébrique attestant de la stabilité : critère de Jury

Le critère de Jury s'apparente au critère de Routh. Nous le donnons pour un polynôme de degré 2, 3 et 4 du type :



$$P(z) = a_0 + a_1 \cdot z + a_2 \cdot z^2 + \dots + a_n \cdot z^n \quad (16)$$

Les conditions de stabilité sont données ci-dessous (ces conditions sont données dans le cas  $a_n > 0$ , il suffit de multiplier tous les termes par  $-1$  si cette condition n'est pas remplie par  $P(z)$ ) :

$$\begin{aligned}
 n = 2 & \begin{cases} a_0 + a_1 + a_2 > 0 \\ a_0 - a_1 + a_2 > 0 \\ -a_0 + a_2 > 0 \end{cases} \\
 n = 3 & \begin{cases} a_0 + a_1 + a_2 + a_3 > 0 \\ -a_0 + a_1 - a_2 + a_3 > 0 \\ -|a_0| + a_3 > 0 \\ a_0 a_2 - a_1 a_3 - a_0^2 + a_3^2 > 0 \end{cases} \\
 n = 4 & \begin{cases} a_0 + a_1 + a_2 + a_3 + a_4 > 0 \\ a_0 - a_1 + a_2 - a_3 + a_4 > 0 \\ a_4^2 - a_0^2 - |a_0 a_3 - a_1 a_4| > 0 \\ (a_0 - a_4)^2 (a_0 - a_2 + a_4) + (a_1 - a_3)(a_0 a_3 - a_1 a_4) > 0 \end{cases}
 \end{aligned} \quad (17)$$

### 3.2.8 Systèmes du premier ordre

On donne dans la Figure 4 les réponses indicielles de systèmes du premier ordre ainsi que la localisation des pôles dans le plan complexe. Le cercle de rayon unité représente la zone dans laquelle les pôles doivent se situer pour assurer la stabilité. On remarque sur les deux premières figures l'influence de la place du zéro par rapport au point  $(1,0)$  pour la rapidité.

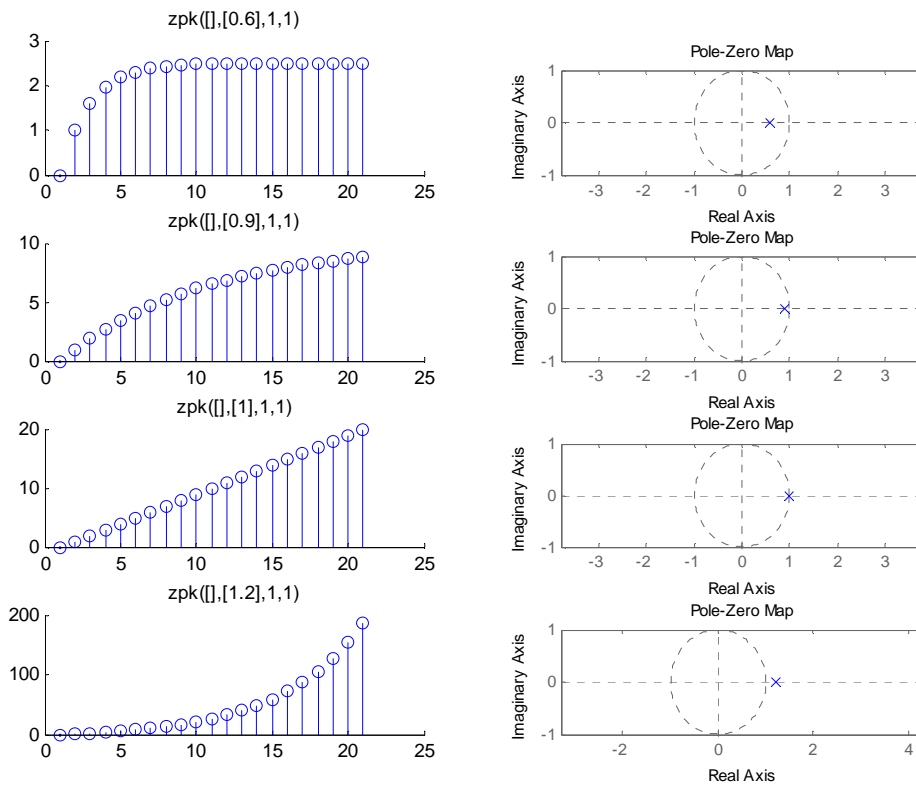


Figure 4 réponses indicielles de systèmes du premier ordre

### 3.2.9 Systèmes du second ordre

On donne dans la Figure 6 les réponses indicielles de systèmes du second ordre ainsi que la localisation des pôles dans le plan complexe. La rapidité et l'amortissement dépend de la position des pôles dans le cercle unité. La Figure 5 représente les courbes iso-amortissement et iso-pulsation des systèmes du second ordre.

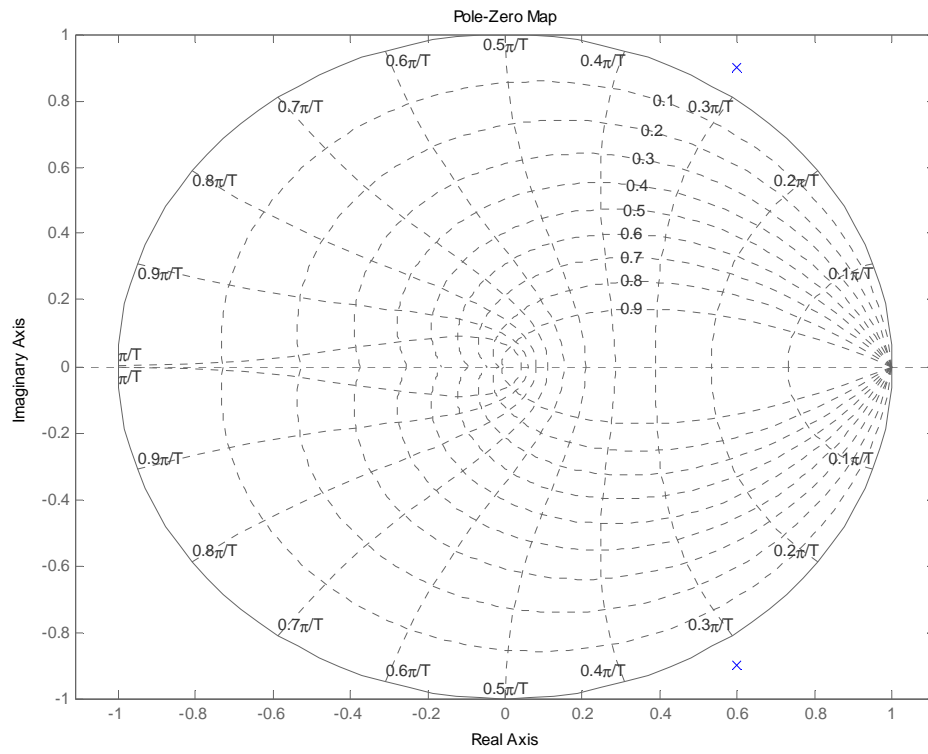


Figure 5 : courbes iso-amortissement et iso-pulsation des systèmes discrets du second ordre

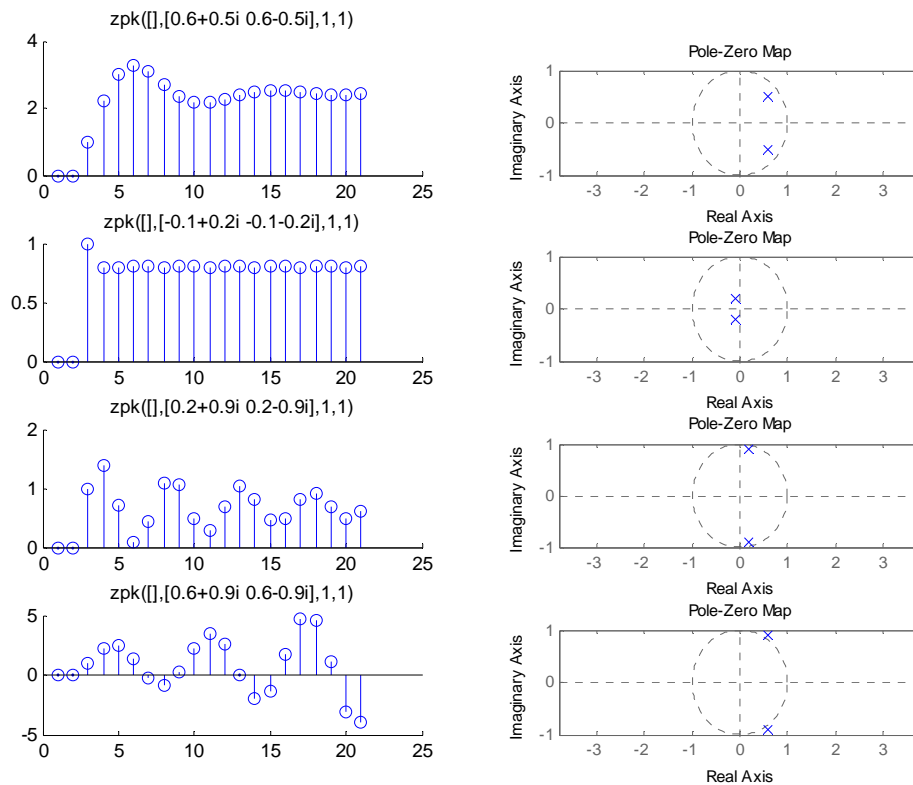


Figure 6 : réponses de systèmes du second ordre

### 3.2.10 Calcul itératif de la réponse d'un système

Si l'on dispose de la fonction de transfert d'un système et que l'on connaît les valeurs successives  $e(k)$  du signal d'entrée il est facile de calculer itérativement la sortie  $y(k)$  du système par l'équation de récurrence (2). Il faut initialiser par les  $n$  valeurs initiales de  $s(k)$ , prises généralement égales à 0.

### 3.2.11 Représentation par équation d'état

Un système discret peut aussi être représenté par un ensemble d'équations d'état comme dans le domaine continu (52). Cette représentation est naturellement adaptée aux systèmes multi-entrées multi-sorties. On définit alors un état interne  $x(k)$  ainsi qu'une fonction d'évolution  $f$  et une équation de mesure  $h$ .

$$\begin{cases} x(k+1) = f(x(k), u(k)) \\ s(k) = h(x(k), u(k)) \end{cases} \quad (18)$$

### 3.3 Réponse fréquentielle des systèmes échantillonnés

#### 3.3.1 Diagramme de Bode d'un système échantillonné

Lorsqu'on étudie la réponse fréquentielle des systèmes continus on considère des entrées particulières sinusoïdales :

$$\begin{cases} e(t) = \cos(2 \cdot \pi \cdot f \cdot t) + j \cdot \sin(2 \cdot \pi \cdot f \cdot t) = e^{j \cdot 2 \cdot \pi \cdot f \cdot t} \\ f \in [0 \quad \infty] \end{cases} \quad (19)$$

De même, pour des systèmes échantillonnés on considèrera des entrées sinusoïdales échantillonnées :

$$\begin{cases} e(k) = \cos(2 \cdot \pi \cdot f \cdot k) + j \cdot \sin(2 \cdot \pi \cdot f \cdot k) = e^{j \cdot 2 \cdot \pi \cdot f \cdot k} \\ f \in [0 \quad 1] \end{cases} \quad (20)$$

La sortie est aussi un signal échantillonné sinusoïdal de gain  $A(f)$  et de phase  $\Phi(f)$  variables et dépendant de  $f$ ; la fréquence réduite.

$$\begin{cases} s(k) = A(f) \cdot e^{j \cdot (2 \cdot \pi \cdot f \cdot k + \Phi(f))} \\ f \in [0 \quad 1] \end{cases} \quad (21)$$

Le nombre complexe, noté abusivement  $F(f)$ , d'amplitude  $A(f)$  et de phase  $\Phi(f)$  est obtenu à partir de  $F(z)$  en remplaçant  $z$  par  $e^{j \cdot 2 \cdot \pi \cdot f}$  :

$$\begin{cases} F(f) = F(z = e^{j \cdot 2 \cdot \pi \cdot f}) \\ f \in [0 \quad 1] \end{cases} \quad (22)$$

Le diagramme représentant le gain et la phase en fonction de la fréquence réduite s'appelle aussi un diagramme de Bode. L'échelle des fréquences est généralement linéaire de 0 à 0.5 (puisque le diagramme est nécessairement symétrique). La Figure 7 représente par exemple le diagramme de Bode d'un système du second ordre.

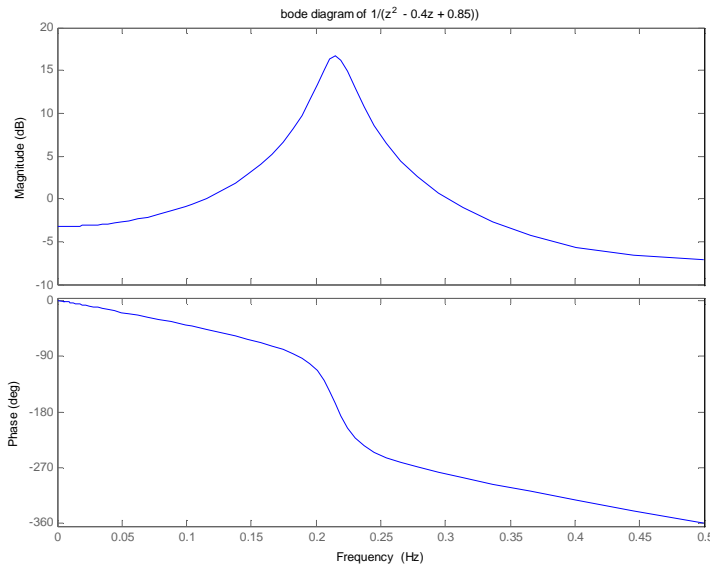


Figure 7 : exemple du diagramme de Bode d'un système discret du second ordre

### 3.3.2 Où l'on voit apparaître le théorème de Shannon

Le changement de variable  $z = e^{j2\pi f}$  n'est pas bijectif puisque  $e^{j2\pi f} = e^{j2\pi(f+k)}$  avec  $k$  entier. Si l'on fait le lien entre domaine continu et domaine fréquentiel on peut remarquer que le changement de variable  $z = e^{j2\pi f}$  fait correspondre une zone du plan complexe continu à une zone du plan complexe discret. La transformation est bijective si l'on se contente de la zone du plan complexe continu contenu entre les droites d'ordonnée 0 et 1.

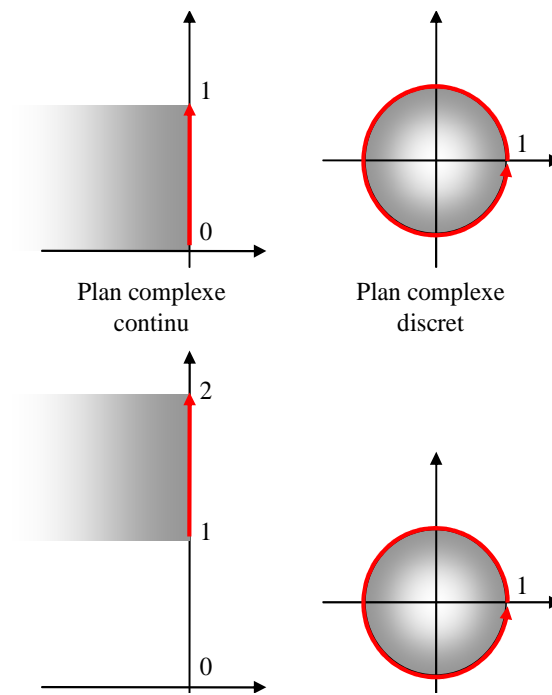


Figure 8 : correspondance entre plan complexe continu et plan complexe discret.





## 4. SYSTEMES CONTINUS ECHANTILLONNES

### 4.1 Structure des systèmes commandés par ordinateur

Les systèmes commandés par ordinateurs sont presque toujours des systèmes physiques continus par nature. Le ordinateur traite par contre les entrées et les sorties de manière discrète avec une période d'échantillonnage  $T_e$ . Les éléments importants, du point de vue de l'automaticien sont donc :

- le convertisseur analogique / numérique CAN ou du moins le système qui fournit séquentiellement les mesures au ordinateur à la période  $T_e$ .
- le ordinateur lui-même. Il doit effectuer l'acquisition des mesures, le traitement des informations, l'envoi des consignes dans un temps inférieur à  $T_e$ .
- le convertisseur numérique / analogique CNA.
- le système physique.

On suppose, pour faire simple et pour pouvoir utiliser la panoplie des outils mathématiques, que les systèmes sont échantillonnés à la même période  $T_e$  et au même moment.

La Figure 9 montre le système physique commandé par le ordinateur, dans sa version simpliste.

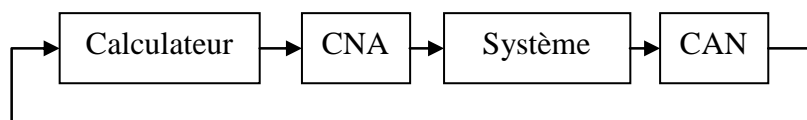


Figure 9 : système commandé par ordinateur

Au XXIème siècle les choses sont quand même parfois un petit peu plus compliquées :

- le ordinateur fournit bien souvent un signal de commande numérique (transporté par bus informatique USB, RS232, CAN, Ethernet, etc...) directement interprété par l'actionneur. On parle "d'actionneur intelligent".
- de même le "capteur intelligent" fournit aussi bien souvent la mesure sous forme d'un signal numérique.
- Les signaux sont transportés par des bus informatiques qui n'ont pas que ça à faire : il peut y avoir du retard.
- de même les capteurs et actionneurs sont rarement tous seuls, ce qui se traduit par des périodes d'échantillonnage différentes, et parfois non synchrones...

- enfin le ordinateur peut aussi avoir d'autres choses à faire...

Ceci étant dit nous allons pour le moment nous restreindre au cas représenté sur la Figure 9 avec des instants d'échantillonnage synchrones et régulièrement espacés d'un temps  $T_e$ .

## 4.2 Echantillonnage des systèmes continus

On dispose généralement de la fonction de transfert continue (notée en utilisant la variable complexe  $s$  ou  $p$ )  $F(s)$  du système continu. Le problème est d'obtenir la fonction de transfert discrète du système échantillonné de manière à pouvoir synthétiser un correcteur. Le système continu est associé au convertisseur numérique – analogique qu'il faut prendre en compte.

Les convertisseurs numérique analogiques peuvent généralement se représenter par des "échantillonneurs – bloqueurs d'ordre 0" ou BOZ. Cela signifie que la commande est maintenue à une valeur constante pendant toute la durée du temps d'échantillonnage.

Si l'on applique une commande discrète impulsionnelle à un BOZ il va maintenir la commande 1 pendant la durée  $T_e$  puis revenir à zéro (Figure 10).

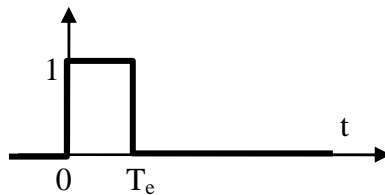


Figure 10 : réponse impulsionnelle d'un bloqueur d'ordre 0

La réponse impulsionnelle  $BOZ(t)$  est donc composée d'un échelon unitaire suivi d'un échelon unitaire de signe opposé et décalé d'un temps  $T_e$ . La transformée de Laplace de la réponse impulsionnelle du bloqueur est donnée par la formule (23) :

$$BOZ(s) = \frac{1}{s} - \frac{e^{-jT_e}}{s} \quad (23)$$

Du point de vue du ordinateur il faut donc prendre en compte le système continu de fonction de transfert  $F_{BOZ}(s)$  que l'on devra échantillonner :

$$F_{BOZ}(s) = \frac{1 - e^{-jT_e}}{s} \cdot F(s) \quad (24)$$

$$F_{\text{BOZ}}(z) = Z^{-1} \left( \frac{1 - e^{-jT_e}}{s} \cdot F(s) \right) \quad (25)$$

En pratique l'obtention de la transformée en z d'un système échantillonné par bloqueur d'ordre 0 se fait à l'aide de tables. On donne quelques exemples fondamentaux dans le Tableau 3 ci-dessous.

Tableau 3 : transformée en z de systèmes continus échantillonnés

F(s)	F <sub>BOZ</sub> (z)
1	1
$\frac{1}{s}$	$\frac{T_e \cdot z^{-1}}{1 - z^{-1}}$
$\frac{1}{1 + T \cdot s}$	$\begin{cases} \frac{b_1 \cdot z^{-1}}{1 + a_1 \cdot z^{-1}} \\ b_1 = 1 - e^{-T_e/T} \quad a_1 = -e^{-T_e/T} \end{cases}$
$\frac{e^{-Ls}}{1 + T \cdot s}$	$\begin{cases} \frac{b_1 \cdot z^{-1} + b_2 \cdot z^{-2}}{1 + a_1 \cdot z^{-1}} \quad a_1 = -e^{-T_e/T} \\ b_1 = 1 - e^{-(L-T_e)/T} \quad b_2 = e^{-T_e/T} (e^{L/T} - 1) \end{cases}$
$L < T_e$	

Si l'on dispose de Matlab c'est bien sûr encore plus simple. On utilise la fonction c2d avec l'argument 'zoh' pour "zero holder hold" (bloqueur d'ordre zéro).

#### Matlab

```
% Définition d'un système continu :
sysc=tf(1,[1 1]);
% Obtention de la transformée en z du système échantillonné à Te
Te=0.1;
sysd=c2d(sysc,Te,'zoh');
% Notation en z^-1
sysd.variable='z'
```

### 4.3 Effets de l'échantillonnage

Le convertisseur numérique a deux effets (néfastes) importants : l'effet de retard et l'effet de quantification.

L'effet de retard se voit dans (24) : on peut dire en première approximation que l'échantillonnage se traduit par un retard pur de  $T_e/2$ . Cela a donc, si ce temps

d'échantillonnage est non négligeable par rapport aux constantes de temps du système, un effet déstabilisant.

Pour illustrer l'effet de quantification supposons par exemple que l'on utilise un convertisseur analogique/numérique 8 bits 0-10V. Le plus petit incrément que le convertisseur peut distinguer est de  $10V/2^8 = 40mV$ . Si le signal utile varie physiquement entre 5V et 5,4V on n'aura que dix incréments utilisables... Ce phénomène n'entre pas dans la panoplie des phénomènes modélisables par la transformée en z. En pratique il est donc souvent nécessaire d'étudier le système et de synthétiser la commande numérique en négligeant ce phénomène puis de vérifier a posteriori qu'il n'a pas un effet trop néfaste.

#### 4.4 Où l'on revoit le théorème de Shannon

Le théorème de Shannon est bien connu en ce qui concerne l'échantillonnage de signaux discrets : on sait que le signal ne doit pas comporter de composante de fréquence supérieure à  $f_e/2$  si l'on ne veut pas observer de repliement. On doit aussi prendre des précautions si l'on veut travailler sur la fonction de transfert discrète correspondant à un système continu échantillonné.

On va traiter un exemple concret pour mettre en évidence le problème de repliement dans le cas de la commande des systèmes continus. On s'intéresse à deux systèmes continus du second ordre ayant des pôles de même partie réelle mais de parties imaginaires séparées de  $2\pi T_e$ . Le script Matlab ci-dessous montre la création des deux systèmes :

##### Matlab

```
% Définition de deux systèmes continus :  
sys1=zpk([],[-1-2i -1+2i],2^2+1);  
sys2=zpk([],[-1-(2+2*pi)*i -1+(2+2*pi)*i],1+(2+2*pi)^2);
```

On voit sur la Figure 11 que la réponse indicielle des deux systèmes est bien différente.

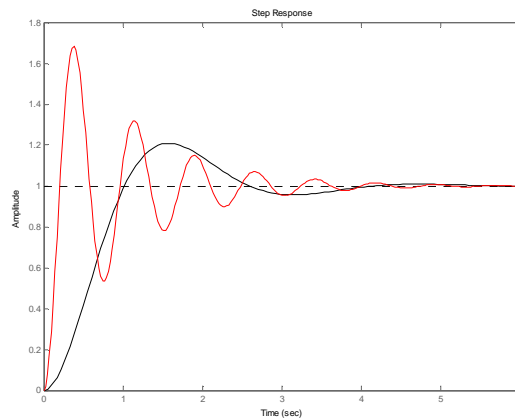


Figure 11 : réponse indicielle de deux systèmes du second ordre

Si l'on calcule la fonction de transfert échantillonnée des deux systèmes à la période de 1 seconde on obtient deux réponses indicielles très proches (représentés par des  $\circ$  sur la Figure 12). Conclusion : la période d'échantillonnage était mal choisie

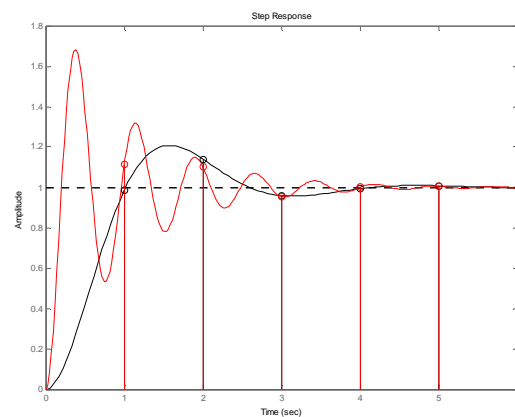


Figure 12 : réponse indicielle de deux systèmes du second ordre échantillonnés à 1s

En pratique la commande échantillonnée de systèmes continus doit être associée à certaines précautions :

- le système ne doit pas comporter de dynamiques propres situées dans les fréquences supérieures à  $f_c/2$  (exemple illustré dans ce chapitre).
- les signaux échantillonnés doivent être "nettoyés" des fréquences supérieures à  $f_c/2$  avec un filtre anti-repliement. (A moins que le signal soit suffisamment propre et ne comporte pas de composantes significatives au-delà de ces fréquences).

### Choix de la période d'échantillonnage

Le choix de la période d'échantillonnage est très important et dépend de la dynamique des signaux. Attention, les systèmes commandés en boucle fermée ont souvent une dynamique plus rapide que le système en boucle ouverte : il faut donc prendre en compte la dynamique en boucle fermée pour le choix de la fréquence d'échantillonnage.

En pratique on préconise de prendre une fréquence 5 à 25 fois supérieure à la bande passante souhaitée en boucle fermée.

On peut être tenté de prendre une fréquence d'échantillonnage très élevée si la technologie le permet (ordinateur et convertisseurs suffisamment rapide). Ceci peut amener rapidement à des problèmes numériques. Imaginons par exemple le cas d'un système ayant une dynamique très lente (quelques secondes) et un système de mesure codant les mesures sur 10 bits (cas classique). La Figure 13 illustre cet exemple. Le signal varie peu entre deux instants d'échantillonnage et les variations du signal sont faibles par rapport à la valeur de l'incrément numérique (figure du haut). Le calcul de la simple dérivée numérique se traduit donc par une succession de  $-1$   $0$  et  $+1$ .

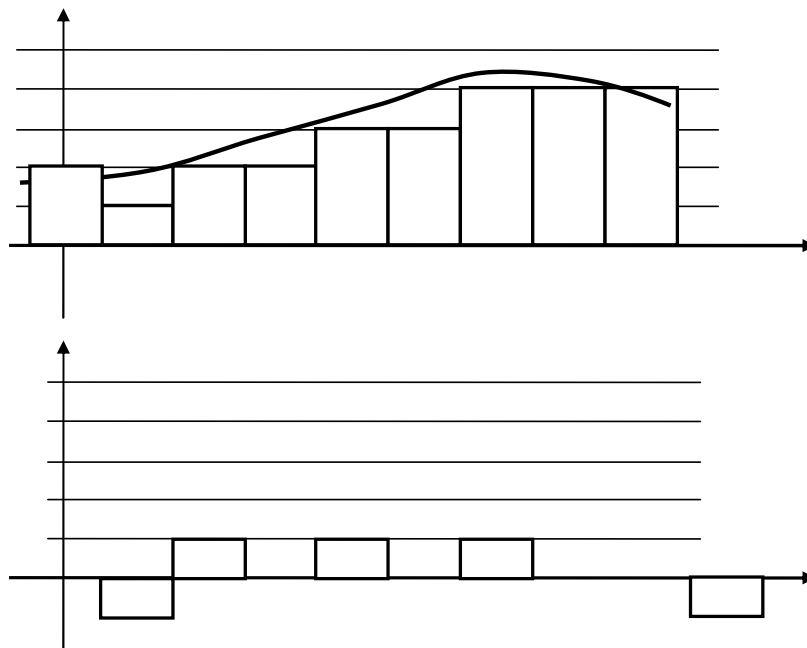


Figure 13 : période d'échantillonnage trop faible. En haut : signal échantillonné et discrétisé ; en bas : calcul de la dérivée numérique.

### Filtre anti-repliement

En toute rigueur chaque fois qu'un signal est échantillonné il faudrait faire précéder le convertisseur numérique d'un filtre analogique anti-repliement éliminant physiquement toutes les composantes du signal de fréquence supérieure à  $f_e/2$ .

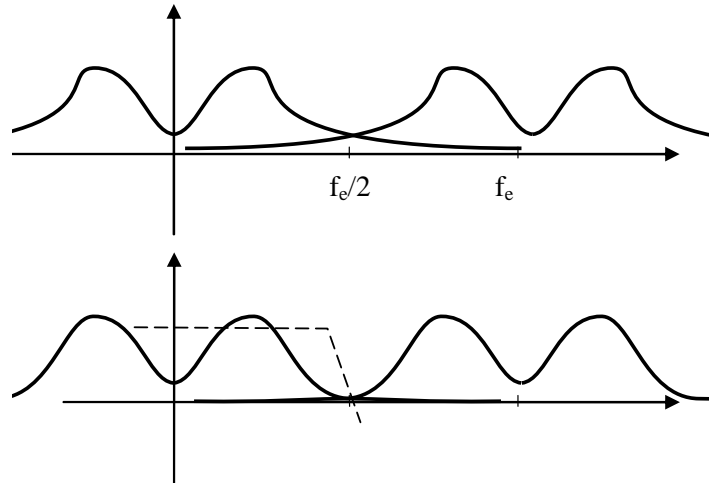


Figure 14 : Effet du filtre anti-repliement

Bien souvent la période d'échantillonnage de l'asservissement est relativement lente (quelques secondes) et il est alors impossible de réaliser un filtre de constante de temps aussi faible. On procède alors à un sur-échantillonnage : le signal est filtré, échantillonné à haute fréquence  $T'_e$  puis filtré numériquement et sous-échantillonné à la période  $T_e$  beaucoup plus grande que  $T'_e$  comme illustré sur la

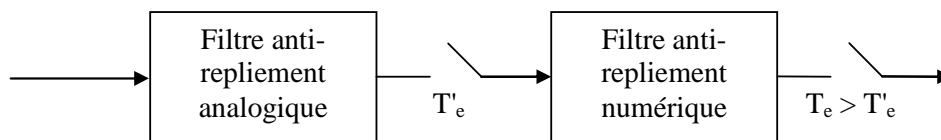


Figure 15 : Sur échantillonnage

Un simple filtre numérique de type moyenne mobile est souvent suffisant.

### 4.5 Discrétisation de fonction de transfert continues

Une manière classique pour réaliser un contrôle par ordinateur est de commencer par synthétiser un correcteur continu avec les outils de l'automatique classique puis de trouver un correcteur discret qui "approche" convenablement le correcteur continu.

#### 4.5.1 Approximation numérique d'équation différentielle

Supposons que le correcteur continu que l'on a synthétisé soit du type PD :

$$\text{PID}(s) = \frac{e(s)}{\varepsilon(s)} = k_p + k_D \cdot s \quad (26)$$

Il s'agit de calculer les commandes  $e(k)$  successives issues du correcteur en fonction des erreurs  $\varepsilon(k)$ . Une approche classique et intuitive est d'approximer la dérivée par une différence :

$$e(k) = k_p \cdot \varepsilon(k) + k_D \cdot \frac{\varepsilon(k) - \varepsilon(k-1)}{T_e} \quad (27)$$

Si on calcule la transformée en  $z$  de  $e(k)$  on obtient :

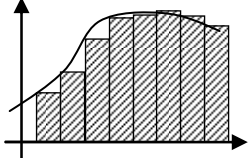
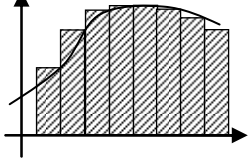
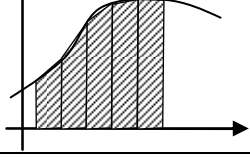
$$e(z) = k_p \cdot \varepsilon(z) + k_D \cdot \frac{\varepsilon(z) - z^{-1}\varepsilon(z)}{T_e} = \left( k_p + k_D \cdot \frac{1 - z^{-1}}{T_e} \right) \cdot \varepsilon(z) \quad (28)$$

On retrouve la fonction de transfert PD(s) dans laquelle on a effectué le changement de variable  $s = \frac{1-z^{-1}}{T_e}$

De manière plus générale, l'approximation de fonction de transfert revient à effectuer l'approximation numérique d'intégration. Selon la méthode d'intégration choisie on effectue des changements de variable différents.



Tableau 4 : méthodes d'intégration

Méthode	Changement de variable	
Euler progressif (Euler's forward method)	$s = \frac{1 - z^{-1}}{z^{-1} \cdot T_e}$	
Euler retrograde (Euler's backward method)	$s = \frac{1 - z^{-1}}{T_e}$	
Méthode des trapèzes, ou bilinéaire, ou de Tustin	$s = \frac{2}{T_e} \cdot \frac{1 - z^{-1}}{1 + z^{-1}}$	

En pratique le changement de variable  $s = \frac{1 - z^{-1}}{z^{-1} \cdot T_e}$  n'est pas sans risque. On donne sur la Figure 16 la correspondance entre la zone stable du plan complexe en  $s$  et la zone correspondante après la transformation (Euler progressif, Euler rétrograde, bilinéaire). On voit notamment qu'il est possible, partant d'un système continu stable, d'obtenir des pôles dans le domaine discret en dehors du cercle de stabilité si la transformation appliquée est celle d'Euler progressif.

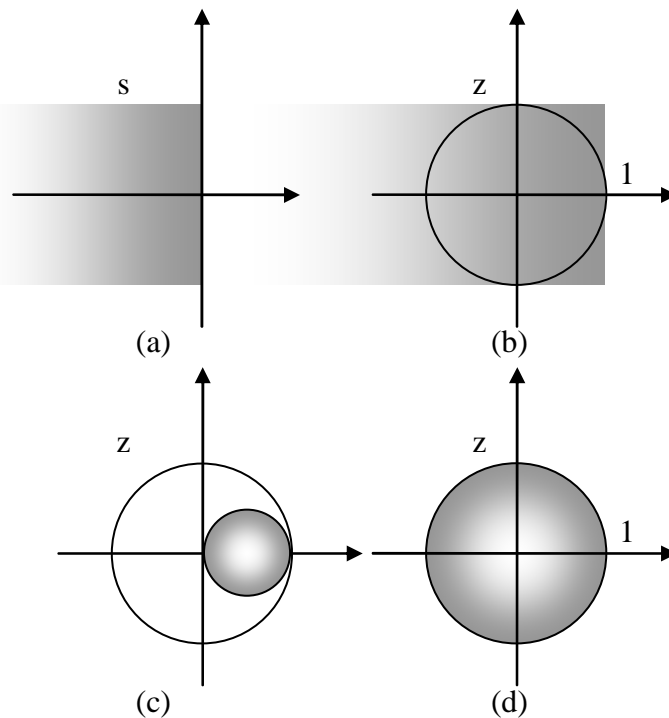


Figure 16 : correspondance des plans complexes. (a) : zone de stabilité dans le plan complexe continu en  $z$  ; (b) zone correspondante avec l'approximation d'Euler progressif ; (c) zone correspondante avec l'approximation d'Euler rétrograde ; (d) zone correspondante avec l'approximation bilinéaire.

La fonction `c2d` de Matlab permet d'obtenir l'équivalent discret d'une fonction de transfert continue avec la méthode d'intégration comme argument :

```

Matlab
sysc=tf([1 0],[1 2 5]);
Te=0.1;
% Méthode d'euler :
sysd1=c2d(sysc,Te,'zoh');
% Méthode des trapèzes :
sysd2=c2d(sysc,Te,'foh');
% Méthode bilinéaire :
sysd3=c2d(sysc,Te,'tustin');
    
```

#### 4.5.2 Superposition des réponses indicielles

Une autre méthode pour approximer une fonction de transfert discrète est de s'assurer que la réponse indicielle de sa version discrétisée correspond exactement lors des temps d'échantillonnage.

### 4.5.3 Correspondances pôles - zéros

La dernière méthode est de partir d'une décomposition de la fonction de transfert continue en pôles et zéros et d'effectuer la conversion pôles à pôles et zéros par zéros. La correspondance, selon que le pôle est réel ou complexe est donnée ci-dessous :

$$\begin{aligned} (s + a) &\rightarrow (z - e^{-a \cdot T_c}) \\ (s + a)^2 + b^2 &\rightarrow z^2 - 2 \cdot e^{-a \cdot T_c} \cdot \cos(b \cdot T_c) \cdot z + e^{-2 \cdot a \cdot T_c} \end{aligned} \quad (29)$$

### 4.5.4 Bilan

Aucune méthode n'est parfaite et on n'aura jamais en même temps :

- même réponse indicielle
- même réponse impulsionnelle
- approximation (Euler, Tustin, etc...) de l'équation différentielle
- correspondance des pôles et des zéros
- correspondance des réponses fréquentielles

Ceci étant dit le choix de la période d'échantillonnage est primordial. Si la fréquence d'échantillonnage est beaucoup plus faible que la dynamique de la fonction de transfert à approximer la différence entre chaque méthode sera minimale.



## 5. IDENTIFICATION DES SYSTEMES ECHANTILLONES

Dans le présent chapitre on donne les bases des méthodes d'identification. Pour une approche complète, voir le cours de troisième année "Identification avion" [6] et les ouvrages [1] et [7].

Même si l'on dispose d'un modèle théorique du processus que l'on souhaite commander il est presque toujours nécessaire de l'affiner en procédant à des essais sur le système réel. Bien souvent le seul modèle dont l'on dispose est ce modèle issu de l'expérimentation. Différentes méthodes s'appliquent selon l'objectif recherché et le niveau de précision exigé. On distinguera :

- les modèles paramétriques
- les modèles simples
- les modèles non paramétriques

### 5.1 Identification d'un modèle ARX par la méthode des moindres carrés récursifs

A partir de cet exemple, le plus simple qui soit, on pose les bases de toutes les méthodes d'identification.

On considère un système d'entrée  $u(t)$  et de sortie  $y(t)$  représenté par l'équation de récurrence suivante :

$$y(t) + a_1 \cdot y(t-1) + a_2 \cdot y(t-2) + \dots + a_n \cdot y(t-n) = b_0 \cdot u(t) + b_1 \cdot u(t-1) + b_2 \cdot u(t-2) + \dots + b_m \cdot u(t-m) \quad (30)$$

L'objectif est, à partir d'essais et de mesures, d'identifier les meilleures valeurs possibles pour les coefficients  $a_i$  et  $b_i$ .

L'équation (30) peut être réécrite pour mettre en évidence la *prédiction* de la sortie  $y(t)$  à partir des mesures passées :

$$y(t) = -a_1 \cdot y(t-1) - a_2 \cdot y(t-2) - \dots - a_n \cdot y(t-n) + b_0 \cdot u(t) + b_1 \cdot u(t-1) + b_2 \cdot u(t-2) + \dots + b_m \cdot u(t-m) \quad (31)$$

Soit :

$$y(t) = \varphi^T(t) \cdot \theta(t)$$

avec :

$$\begin{cases} \varphi^T(t) = [-y(t-1) \ \dots \ -y(t-n) \ , \ u(t-1) \ \dots \ u(t-m)] \\ \theta(t) = [a_1 \ \dots \ a_n \ , \ b_1 \ \dots \ b_m] \end{cases} \quad (32)$$

Le but est de trouver le meilleur ensemble de paramètres  $\theta(t)$  : celui qui, à partir d'un ensemble de mesures  $\varphi(t)$  permet de prédire avec la meilleure précision la sortie effectivement mesurée.

Supposons qu'à la suite d'une expérience on dispose d'un ensemble de mesures  $\{y(t), u(t)\}_{t=1:N}$  et que l'ensemble des paramètres  $\theta$  n'est pas bien connu. On construit la séquence des *sorties prédites* :

$$\hat{y}(t) = \varphi^T(t) \cdot \theta(t) \quad (33)$$

Le paramètre  $\theta$  est celui qui minimise une norme quadratique :

$$\theta = \arg \max \left( \frac{1}{N} \sum_{t=1}^N (y(t) - \hat{y}(t))^2 \right) = \arg \max \left( \frac{1}{N} \sum_{t=1}^N (y(t) - \varphi^T(t) \cdot \theta(t))^2 \right) \quad (34)$$

Compte tenu du fait que l'équation est linéaire au départ et que le terme à minimiser est du type quadratique la solution est simplement obtenue par une pseudo-inverse :

$$\theta = \left[ \sum_{t=1}^N (\varphi(t) \cdot \varphi^T(t)) \right]^{-1} \sum_{t=1}^N (\varphi(t) \cdot y(t)) \quad (35)$$

On obtient donc la solution classique des moindres carrés. La formule (35) est le plus souvent utilisée sous sa forme récursive :

$$\theta(t) = \theta(t-1) + F(t) \cdot \varphi(t) \cdot (y(t) - \theta^T(t-1) \cdot \varphi(t))$$

avec

$$F(t)^{-1} = F(t-1)^{-1} + \varphi(t) \cdot \varphi(t)^T \quad (36)$$

ou

$$F(t) = F(t-1) - \frac{F(t-1) \cdot \varphi(t) \cdot \varphi(t)^T \cdot F(t-1)}{1 + \varphi(t)^T \cdot F(t-1) \cdot \varphi(t)}$$

(On laisse au lecteur le plaisir d'effectuer lui-même les démonstrations, qu'il peut du reste trouver dans [1] et [7]).

## 5.2 L'identification paramétrique en général et en pratique

L'exemple précédent a permis de mettre en évidence tous les aspects de la méthodologie classique de l'identification :

- il a fallu choisir un ensemble de mesures  $\{y(t), u(t)\}_{t=1:N}$ . Le choix du signal d'excitation  $u(t)$  est en pratique primordial,
- il a fallu choisir une structure pour le modèle à identifier (ici les ordres  $n$  et  $m$ ),
- il a fallu choisir un critère (ici un critère quadratique sur l'erreur entre la sortie et la sortie prédite),
- il a fallu choisir un algorithme et le paramétrer (ici l'algorithme des moindres carrés récursifs qu'il a fallu initialiser)

Enfin un point non traité dans le chapitre précédent concerne la validation :

- il aurait fallu déterminer une méthode pour comparer différents modèles et choisir le meilleur

### 5.2.1 Choix du signal d'excitation, traitement des mesures

De manière générale le système que l'on doit identifier doit être excité dans tout le domaine fréquentiel qui nous intéresse. Le signal classiquement utilisé est une séquence pseudo binaire aléatoire (Pseudo Random Binary Signal) tel celui représenté sur la Figure 17.

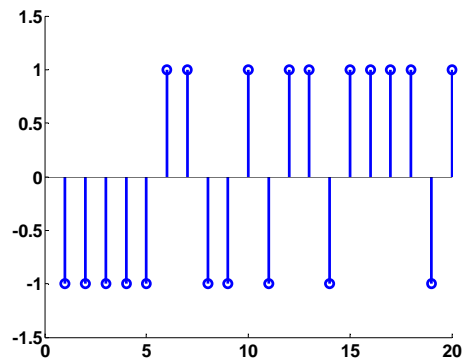


Figure 17 : exemple d'une séquence PRBS

Le signal d'excitation doit se superposer au signal de commande correspondant au point de fonctionnement étudié. Son amplitude doit être faible par rapport à l'amplitude des commandes du système (quelques %).

Une fois le système excité et les mesures effectuées il faut supprimer les valeurs moyennes des mesures.

### 5.2.2 Choix d'une structure de modèle

Le modèle étudié dans le chapitre précédent est de type ARX. On peut le noter sous forme polynomiale :

$$A \cdot y = B \cdot u \quad (37)$$

Ce modèle correspond en fait à un parti-pris sur la manière dont on traite les perturbations. En réalité on prend en compte un signal d'erreurs possibles  $e$  (de moyenne nulle) tels que :

$$A \cdot y = B \cdot u + e \quad (38)$$

Si l'on fait cette hypothèse de modélisation alors la méthode optimale est celle des moindres carrés. Malheureusement cette manière de prendre en compte les perturbations est bien restrictive. D'autres modélisations sont possibles, adaptées à d'autres algorithmes, par exemple le modèle OE :

$$y = \frac{B}{F} \cdot u + e \quad (39)$$

Il est aussi possible de "structurer" le modèle de bruit comme par exemple dans les modèles ARMAX ou BJ :

$$A \cdot y = B \cdot u + C \cdot e$$

$$y = \frac{B}{F} \cdot u + \frac{C}{D} \cdot e \quad (40)$$

La structure du modèle étant choisie il faut choisir l'ordre de ses différents polynômes.

### 5.2.3 Choix d'un critère à minimiser

Dans la plupart des cas le critère à minimiser est l'erreur quadratique entre la sortie et la sortie prédite à un coup (one step ahead prediction). Ceci n'est bien évidemment pas restrictif et peut être généralisé à la prédiction à N coups.

### 5.2.4 Choix d'un algorithme

Même si chaque structure de modèle est généralement associée à un algorithme "optimal" celui-ci doit généralement se paramétrer.



Prenons par exemple l'algorithme des moindres carrés récursifs tel que présenté dans (36). Cette équation peut être modifiée en intégrant un "facteur d'oubli"  $\lambda$  :

$$\begin{aligned} \theta(t) &= \theta(t-1) + F(t) \cdot \varphi(t) \cdot (y(t) - \theta^T(t-1) \cdot \varphi(t)) \\ \text{avec} & \\ F(t)^{-1} &= \lambda \cdot F(t-1)^{-1} + \varphi(t) \cdot \varphi(t)^T \end{aligned} \quad (41)$$

Ce paramètre (typiquement entre 0.95 et 0.99) permet, comme son nom l'indique, "d'oublier" les anciennes mesures et de se concentrer sur les mesures les plus récentes.

### 5.2.5 Validation

Une fois l'algorithme d'identification appliqué et le modèle obtenu (ce qui prend généralement quelques secondes avec un ordinateur moderne) il reste à expertiser celui-ci. Selon la "qualité" du modèle obtenu on remet en cause l'une ou l'autre des étapes (signal d'excitation, structure du modèle, algorithme) et on réitère jusqu'à obtenir un modèle satisfaisant.

En pratique on essaie un certain nombre de modèles avant de choisir le "meilleur".

Le premier critère de qualité est la capacité du modèle à prédire effectivement la sortie (sortie prédite à un coup, sortie prédite à N coups). Il faut pour cela utiliser un jeu de mesures non utilisé pour l'identification. (En pratique le jeu de mesures initiales est coupé en deux : une partie pour l'identification et une partie pour la validation).

Le second critère de qualité est l'analyse du résidu : la composition spectrale de l'erreur de prédiction.

### 5.2.6 Utilisation de logiciels

En pratique il est commode d'utiliser un célèbre outil de calcul scientifique et sa toolboxe d'identification.

Comme les calculs proprement dit ne prennent que quelques secondes il est facile de réaliser un grand nombre d'identifications et de stocker les différents modèles pour les comparer ensuite. On est rapidement noyé sous le grand nombre de modèles possibles...

## 5.3 Identification "simple"

Il est souvent suffisant d'obtenir un modèle "simple" de type premier ordre et un simple essai indiciel convient souvent. Le système est excité avec un échelon unité et on cherche le meilleur modèle du premier ordre correspondant.

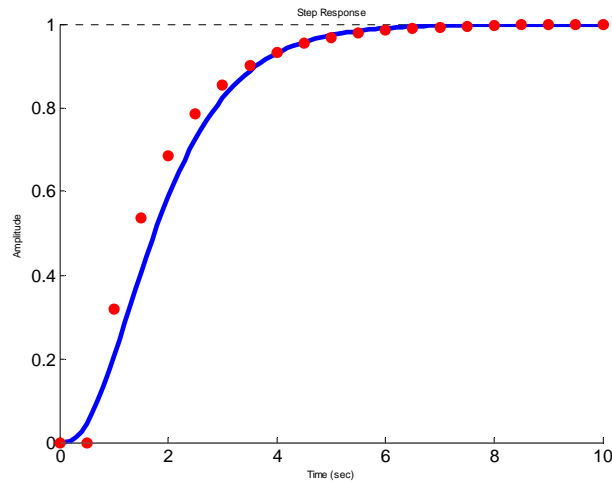


Figure 18 : identification sur un essai indiciel

Dans l'exemple de la Figure 18 on superpose le résultat d'une mesure (en bleu) avec la simulation d'un modèle discret avec une période de 0.5s. Le modèle cherché est du premier ordre avec un retard pur  $d$  :

$$y(z) = \frac{(1 + c_1) \cdot z^{-d}}{1 + c_1 \cdot z^{-1}} \cdot u(z) \quad (42)$$

Il s'agit de trouver l'entier  $d$  et le nombre réel  $c_1$  tels que la réponse indicielle du modèle corresponde pour le mieux aux mesures issues de l'essai. Ceci peut se formaliser par exemple de la manière suivante :

$$(c_1 \quad d_1) = \arg \min \sum_{t=1}^N (y_{\text{modèle}}(t) - y_{\text{mesure}}(t))^2 \quad (43)$$

Cette optimisation peut se faire à la main ou grâce à une procédure automatisée (avec le célèbre outil de calcul scientifique).

Dans le cas où le système comprend naturellement une intégration (par exemple la commande d'un moteur à courant continu par une tension et la mesure de l'angle de rotation) il est utile de prendre en compte explicitement cette intégration dans le modèle :

$$y(z) = \frac{1}{1 - z^{-1}} \cdot \frac{(1 + c_1) \cdot z^{-d}}{1 + c_1 \cdot z^{-1}} \cdot u(z) \quad (44)$$

La procédure est la même mais au lieu de prendre en compte la sortie  $y(t)$  on prend en compte la dérivée  $y'(t)$  :

$$y'(t) = y(t) - y(t-1) \quad (45)$$

## 5.4 Identification non paramétrique

Pour terminer ce panorama rapide sur les méthodes d'identification il faut parler des méthodes non paramétriques : celles qui permettent de tracer la réponse impulsionnelle ou la réponse fréquentielle du système.

### 5.4.1 Estimation de la réponse impulsionnelle

La réponse impulsionnelle d'un système peut être obtenue simplement en appliquant une entrée de type "impulsion" au système et d'en mesurer les conséquences.

La seconde méthode consiste à utiliser la méthode des moindres carrés (ou une méthode dérivée) comme vue au chapitre 5.1. La réponse impulsionnelle d'un système peut en effet être vue comme la fonction de transfert particulière où le dénominateur est de degré nul et le numérateur comporte un nombre infini (en pratique : grand) d'éléments ( $n = 0$  et  $m$  grand) :

$$y(t) = b_0 \cdot u(t) + b_1 \cdot u(t-1) + b_2 \cdot u(t-2) + \dots + b_m \cdot u(t-m) \quad (46)$$

La dernière méthode, qui sera développée ici est l'analyse d'intercorrélation.

Par définition l'intercorrélation  $R_{yu}$  entre deux signaux est :

$$R_{yu}(\tau) = E[(y(t) - E(y(t))) \cdot (u(t-\tau) - E(u(t-\tau)))] \quad (47)$$

Dans cette formule l'espérance mathématique peut être approchée grâce à la loi des grands nombres :

$$R_{yu}(\tau) \approx R_{yu}^N(\tau) = \lim_{N \rightarrow \infty} \frac{1}{N} \sum_{t=1}^N [(y(t) - m_y) \cdot (u(t-\tau) - m_u)] \quad (48)$$

On s'intéresse maintenant à un système tel que décrit par l'équation impulsionnelle (46) et on suppose que le signal d'entrée est un bruit blanc. Le signal d'entrée vérifie donc :

$$\begin{cases} \tau \neq 0 & R_{uu}(\tau) = 0 \\ \tau = 0 & R_{uu}(\tau) = R \end{cases} \quad (49)$$

Il vient alors naturellement :

$$b_k \approx \frac{1}{R} R_{yu}(k) \quad (50)$$

#### 5.4.2 Estimation de la réponse fréquentielle

On utilise ici le spectre d'intercorrélation. Par définition le spectre d'intercorrélation entre deux signaux est :

$$\Phi_{yu}(\omega) = \sum_{\tau=-\infty}^{\infty} R_{yu}(\tau) \cdot e^{-i\omega\tau} \quad (51)$$

On suppose que le signal d'entrée  $u(t)$  et le signal de sortie  $y(t)$  sont reliés par une fonction de transfert linéaire du type :

$$y(t) = F \cdot u(t) + v(t) \quad (52)$$

où  $v(t)$  est une perturbation indépendante de  $u(t)$ . On a alors les relations :

$$\Phi_{yu}(\omega) = F(j\omega) \cdot \Phi_{uu}(\omega) \quad (53)$$

Il reste à estimer les spectres d'intercorrélation pour remonter à la réponse fréquentielle  $F(j\omega)$  recherchée. Une idée naturelle est d'utiliser la relation (48) mais cela marche très mal dans presque tous les cas, même en choisissant  $N$  assez grand. L'amélioration est d'utiliser une "fenêtre" pour favoriser les valeurs faibles de  $\tau$  par rapport aux valeurs élevées. La solution classique utilisant la *fenêtre de Hamming* donne :

$$\Phi_{yu}^N(\omega) = \sum_{\tau=-\gamma}^{\gamma} R_{yu}^N(\tau) \cdot w_{\gamma}(\tau) \cdot e^{-i\omega\tau} \quad (54)$$

avec

$$\begin{cases} w_{\gamma}(\tau) = \frac{1}{2} \left( 1 + \cos\left(\frac{\pi \cdot \tau}{\gamma}\right) \right) & |\tau| < \gamma \\ w_{\gamma}(\tau) = 0 & |\tau| > \gamma \end{cases}$$

On utilise ensuite la relation (51) pour estimer la réponse fréquentielle :

$$F(j \cdot \omega) = \frac{\Phi_u^N(\omega)}{\Phi_{yu}^N(\omega)} \quad (55)$$



## 6. SYSTEMES BOUCLES

### 6.1 Fonction de transfert en boucle fermée

Un système et sa commande par ordinateur peut être représenté par un ensemble de fonctions de transfert formant un schéma-bloc plus ou moins compliqué (Figure 19).

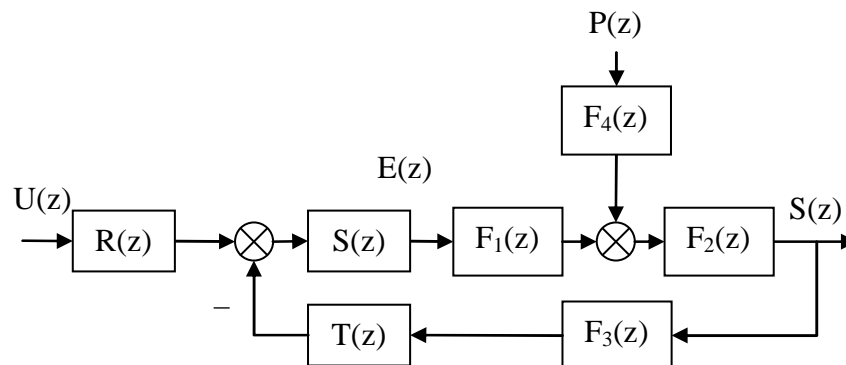


Figure 19 : système discret commandé par ordinateur. Le correcteur est représenté par les fonctions R,S,T, le système par  $F_1$  et  $F_2$ , le modèle de perturbations par  $F_4$  et le modèle des actionneurs par  $F_4$ .

Quelle que soit la structure d'un système asservi il est possible de se ramener à une forme canonique comprenant une fonction de transfert en boucle ouverte en combinant judicieusement les fonctions de transfert du procédé, du régulateur et des perturbations. La Figure 20 représente par exemple le système de la Figure 19 où  $F$  est une combinaison des fonctions de transfert du système, du régulateur, des perturbations et des capteurs.

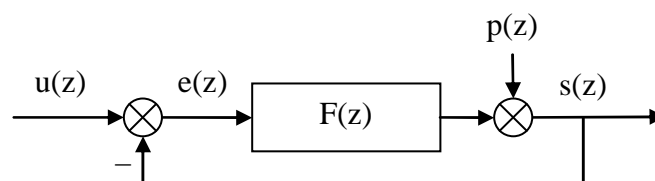


Figure 20 : système discret, notations.

Par défaut on considère une contre-réaction négative et une perturbation additive sur la sortie. La fonction de transfert  $F$  est appelée *fonction de transfert en boucle ouverte*. Comme pour les systèmes continus la *fonction de transfert en boucle fermée* est obtenue simplement à partir de  $F$  (58).

$$\begin{aligned}
 F(z) &= \frac{s(z)}{e(z)} = \frac{B(z^{-1})}{A(z^{-1})} \\
 G(z) &= \frac{s(z)}{u(z)} = \frac{B(z^{-1})}{A(z^{-1}) + B(z^{-1})}
 \end{aligned}
 \tag{56}$$

## 6.2 Erreur statique

L'erreur statique du système en boucle fermée, comme pour le système en boucle ouverte, est simplement le rapport des coefficients.

$$G_0 = \frac{\sum_{i=0}^{i=m} b_i}{1 + \sum_{i=0}^{i=m} b_i + \sum_{i=1}^{i=n} a_i}
 \tag{57}$$

Si l'on souhaite que l'asservissement confère un gain statique unitaire au système en boucle fermée (c'est-à-dire que l'on force la sortie  $s(k)$  à tendre vers la valeur de consigne  $u(k)$ ) il faut que ce rapport de coefficients soit égal à un. On montre facilement que cela est obtenu si la fonction de transfert  $F(z)$  comporte un intégrateur :

$$F(z) = \frac{s(z)}{e(z)} = \frac{1}{1 - z^{-1}} \cdot \frac{B(z)}{A'(z)}
 \tag{58}$$

## 6.3 Rejet de perturbations

La fonction de transfert entre la perturbation  $p(z)$  et la sortie  $y(z)$  est appelée fonction de sensibilité perturbation-sortie, notée  $S_{yp}$ .

$$S_{yp}(z) = \frac{s(z)}{p(z)} = \frac{A(z^{-1})}{A(z^{-1}) + B(z^{-1})}
 \tag{59}$$

Si l'on souhaite qu'une perturbation de type échelon soit parfaitement rejetée il faut que  $A(1) = 0$ , on retrouve donc la condition (58) à savoir que le système doit comporter une intégration dans la boucle directe. On peut être en outre être amené à spécifier que les perturbations ne soient pas amplifiées dans une certaine bande de fréquence. On rajoute donc une contrainte du type  $|S_{yp}(e^{j2\pi f})| < S_{\max}$  pour  $f$  appartenant à un certain domaine de  $[0 \ 0.5]$ .



## 6.4 Stabilité des systèmes à simple gain de rétroaction

Les critères de Routh et de Jury (donnés dans le chapitre 3.2.7 p13) ou le calcul numérique des pôles permettent de déterminer la stabilité du système en boucle fermée si l'on connaît sa fonction de transfert.

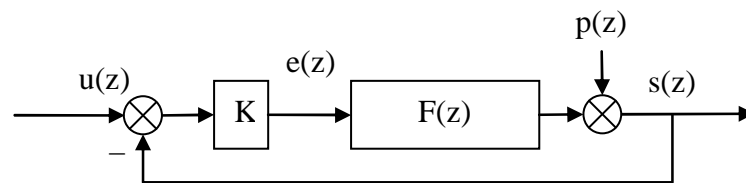


Figure 21 : système bouclé par un simple gain.

On s'intéresse ici au cas particulier où l'asservissement est réalisé par un simple gain de rétroaction  $K$  (Figure 21). On cherche la valeur de  $K$  qui amène le système en limite de stabilité. L'outil adapté ici est le lieu d'Evans ou lieu des racines (root locus) qui permet de visualiser l'évolution des pôles du système bouclé lorsque le gain  $K$  varie.

Prenons l'exemple d'un système comportant trois pôles et deux zéros. L'exemple est donné sous forme de script Matlab :

### Matlab

```
% Définition d'un système discret :
sys=zpk([-2.037 -0.1406],[0.6065 0.3679 0.3679],1,1)
%Tracé du lieu des racines
rlocus(sys);
```

Le script précédent produit le tracé de la Figure 22. On pourrait y lire que la limite de stabilité est obtenue pour  $K=0.2$  environ (détail de la Figure 23).

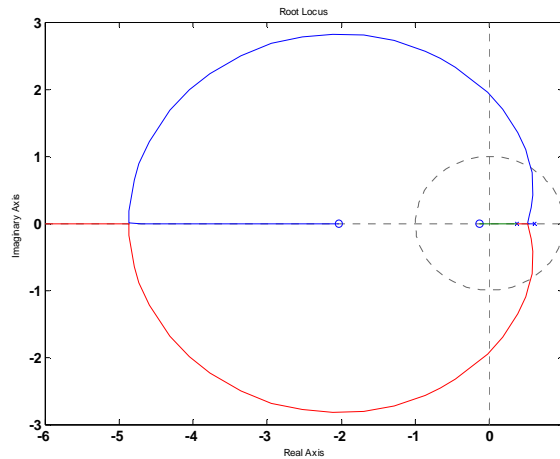


Figure 22 : lieu des racines

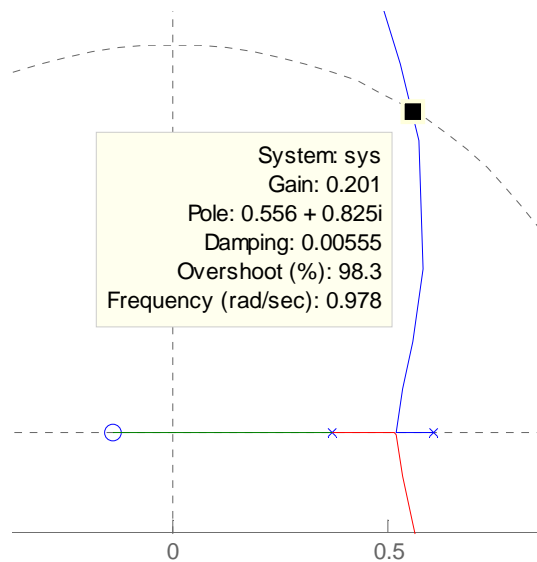


Figure 23 : lieu des racines : détail de la limite de stabilité

On voit sur la Figure 22 que le lieu des racines d'un système échantillonné ressemble à celui d'un système continu :

- on commence par tracer les pôles (×) et les zéros (o) du système en boucle ouverte.
- les pôles du système en boucle fermée décrivent un lieu qui part des pôles (pour  $K = 0$ ) et arrive aux zéros (pour  $K = \infty$ )
- lorsque il y a  $n-m$  plus de pôles que de zéros le système comporte  $n-m$  branches asymptotiques.

On ne décrira pas ici les diverses propriétés du lieu des racines qui permettent de le tracer à la main puisque il est facile à obtenir avec un ordinateur.

Il peut aussi être utilisé pour régler "à la main" le gain permettant d'obtenir un ou plusieurs modes dominants ayant certaines propriétés de fréquence et d'amortissement. On voit sur la Figure 24 que le gain 0.017 permet d'obtenir un pôle double d'amortissement 0.706 et de fréquence 0.65 rad/s et un pôle simple de fréquence 1.63 rad/s.

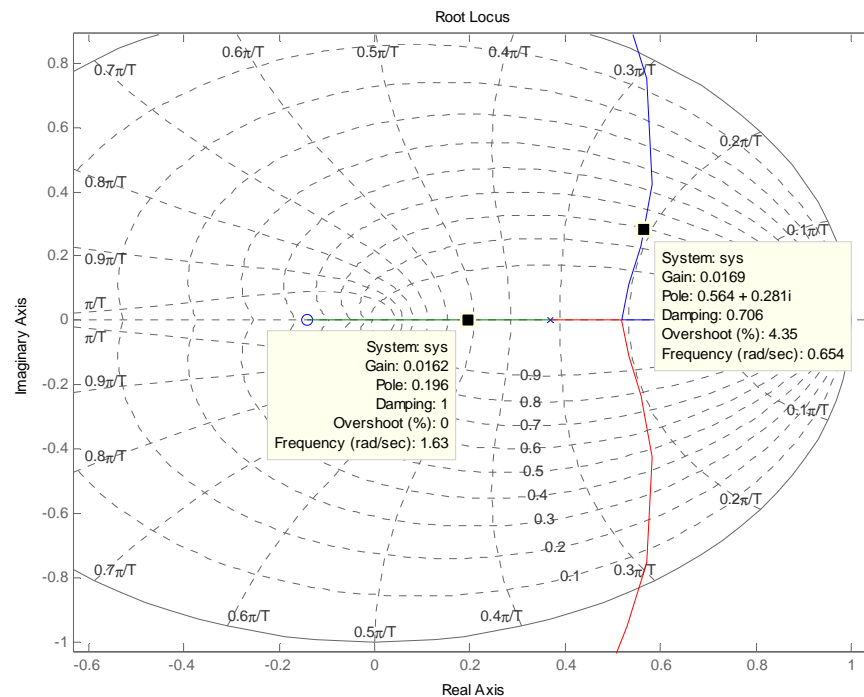


Figure 24 : utilisation du lieu des racines pour régler le gain de la boucle fermée

L'outil rltool de Matlab permet de nombreuses manipulations graphiques de ce type.

## 6.5 Analyse des systèmes bouclés dans le domaine fréquentiel : stabilité

Comme dans le cas continu il est possible de prévoir des propriétés, en particulier la stabilité, d'un système en boucle fermée à partir de l'examen de la fonction de transfert en boucle ouverte.

On ne redémontrera pas ici le théorème de Nyquist mais on se contente de rappeler l'énoncé.

On rappelle que le *lieu de Nyquist* est le lieu de  $F(j2\pi f)$  dans le plan complexe pour  $f$  variant de 0 à  $+\infty$ . A noter que le lieu pour la zone  $[0, 0.5]$  est symétrique de celui pour la zone  $[0.5, 1]$ . Pour cette raison on ne représente généralement que le lieu pour  $f$  appartenant à  $[0, 0.5]$ .

### 6.5.1 Théorème de Nyquist dans le cas où le système est stable en boucle ouverte (critère du revers)

Le système est stable en boucle fermée si le lieu de Nyquist de la fonction de transfert en boucle ouverte parcouru dans le sens des fréquences croissantes de 0 à 0.5 laisse le point critique  $(-1,0)$  à gauche.

La Figure 25 représente par exemple le lieu de Nyquist d'une fonction de transfert en boucle ouverte qui sera stable en boucle fermée.

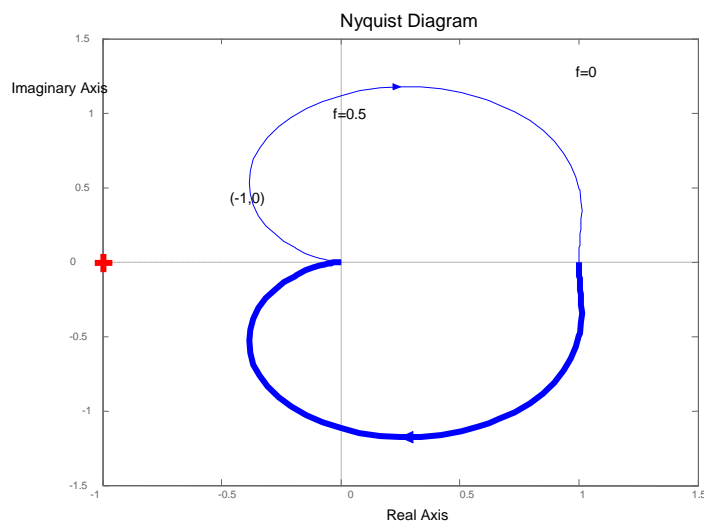


Figure 25 : lieu de Nyquist d'un système stable

### 6.5.2 Théorème de Nyquist dans le cas où le système est instable en boucle ouverte

Si le système en boucle ouverte comporte  $N$  pôles instables il sera stable en boucle fermée si le lieu de Nyquist de la fonction de transfert en boucle ouverte parcouru dans le sens des fréquences croissantes de  $-0.5$  à  $0.5$  entoure le point critique  $(-1,0)$   $N$  fois dans le sens des aiguilles d'une montre..

### 6.5.3 Interprétations du lieu de Nyquist

Comme dans le cas continu, la distance à laquelle on passe du point critique renseigne sur la robustesse du système bouclé. La distance sera exprimée de différente manière (Figure 26) :

- marge de module  $\Delta M$  : plus petite distance entre le lieu des racines et le point critique,

- marge de gain  $\Delta G$
- marge de phase  $\Delta P$

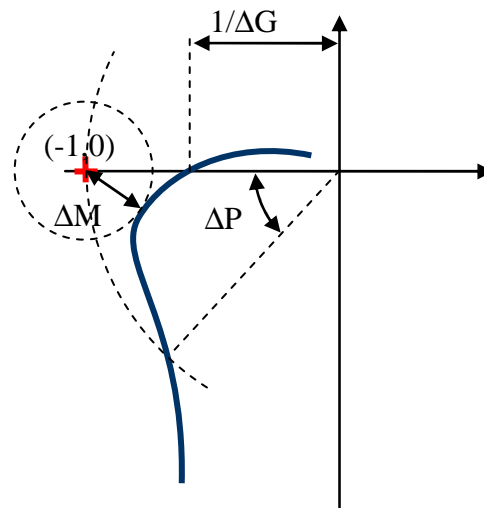


Figure 26 : Marges de stabilité

Les valeurs typiques de ces marges sont données ci-après :

- $\Delta G \geq 6\text{dB}$
- $\Delta M \geq -6\text{dB}$
- $30^\circ \leq \Delta P \leq 60^\circ$

La distance entre le point critique et le lieu des racines est la norme de  $1+F(j2\pi f)$ . On peut donc exprimer la marge de module en fonction de la fonction de sensibilité :

$$\begin{aligned} \Delta M &= \min |1 + F(j \cdot \omega)|_{\omega \in [0, \pi]} \\ \Delta M &= \min \left| 1 + \frac{B(j \cdot \omega)}{A(j \cdot \omega)} \right|_{\omega \in [0, \pi]} = \max \left| \frac{A(j \cdot \omega)}{A(j \cdot \omega) + B(j \cdot \omega)} \right|_{\omega \in [0, \pi]} \quad (60) \\ \Delta M &= \max |S_{yp}(j \cdot \omega)|_{\omega \in [0, \pi]}^{-1} \\ \Delta M_{\text{dB}} &= -\max |S_{yp}(j \cdot \omega)|_{\text{dB}} \end{aligned}$$

Si l'on trace l'évolution de  $S_{yp}$  en fonction de  $f$  on obtient donc la valeur de la marge de module.



# 7. METHODES DE REGULATION DES SYSTEMES ECHANTILLONES

## 7.1 Introduction

Les bases théoriques permettant de concevoir le correcteur pour un système échantillonné ont été posées dans les chapitres précédents. Ce chapitre fait le bilan des connaissances acquises et donne quelques solutions pour calculer des correcteurs à partir d'un cahier des charges précis.

Les systèmes que l'on cherche à commander sont généralement, par nature, continus. La réalisation d'un correcteur par un ordinateur implique que l'on génère une équation de commande discrète (généralement une équation récurrente). L'obtention de cette équation de commande peut se faire de deux manières :

- la première méthode consiste à d'abord synthétiser un correcteur continu (en utilisant les outils classiques de l'automatique et la simulation) et discrétiser ensuite ce correcteur. C'est l'objet du chapitre 7.2.2.
- la seconde méthode consiste à tenir la représentation discrète du système continu associé à son interface numérique (convertisseurs analogique-numérique et numérique-analogique) comme expliqué dans le chapitre 4.5 puis synthétiser directement le correcteur discret. Cela sera expliqué dans le chapitre 7.3 et suivants.

Ce choix peut être résumé par la figure suivante :

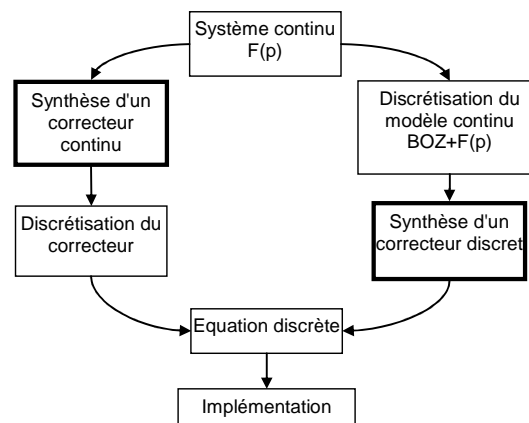


Figure 27 : Réalisation d'un correcteur discret.

Au-delà des affaires de goût, le choix de la méthode sera souvent dicté par l'application. En effet, la première méthode est d'autant meilleure que le correcteur discrétisé approche correctement le correcteur continu. Cela implique que le temps d'échantillonnage soit très faible par rapport aux constantes de temps du système

bouclé. Dans le cas où les contraintes temps réel sont fortes (capacités de calcul partagées par exemple) il sera préférable de synthétiser un correcteur discret.

## 7.2 Structure d'un système commandé par ordinateur, hypothèses

Par la suite on se ramènera toujours à un système tel que celui représenté sur la Figure 28 en cherchant un correcteur  $K(z)$  linéaire. On montre que quel que soit le correcteur linéaire on peut se ramener à la forme représenté sur la Figure 29 (forme RST).

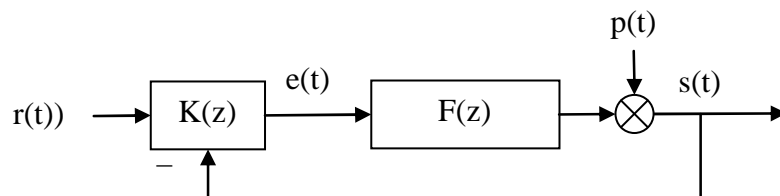


Figure 28 : système bouclé, forme générale.

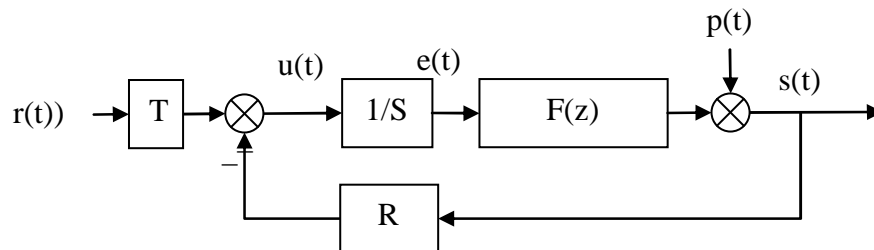


Figure 29 : système bouclé, forme canonique.

### 7.2.1 Discrétisation, choix de la période d'échantillonnage

Du point de vue du ordinateur, l'extérieur (le procédé) est représenté par la fonction de transfert  $F(z)$ . L'hypothèse importante est que la commande (sortie du ordinateur) et la mesure (entrée du ordinateur) sont échantillonnées de manière synchrone à la même période  $T_e$ .

Le choix judicieux de la fréquence d'échantillonnage est très important. En pratique on choisit cette fréquence en fonction de la bande passante du système en **boucle fermée** :

$$f_e = (6 \text{ à } 25) f^{BF} \quad (61)$$



### 7.2.2 Discrétisation d'un correcteur continu

La première méthode pour obtenir les équations d'un correcteur numérique est de faire sa synthèse avec les outils de l'automatique continue puis de discrétiser la fonction de transfert du correcteur continu obtenu (voir pour cela le chapitre 4.2 *Echantillonnage des systèmes continus*).

On va voir par la suite que la synthèse directe d'un correcteur discret offre plus de degré de liberté et permet en général de remplir un cahier des charges plus complet.

## 7.3 Régulateur PID numérique

Le régulateur le plus populaire est le régulateur PID. Nous allons voir dans un premier temps comment on génère un régulateur PID à partir des équations bien connues du régulateur PID continu et comment ces équations peuvent être généralisées pour obtenir plus de degrés de liberté pour le réglage.

### 7.3.1 Cas 1

Un régulateur PID continu a une fonction de transfert du type :

$$K_{\text{PID}}(s) = K \cdot \left( 1 + \frac{1}{T_i \cdot s} + \frac{T_d \cdot s}{1 + \frac{T_d}{N} \cdot s} \right) \quad (62)$$

La discrétisation de ce correcteur continu par la méthode d'Euler (Tableau 4 page 31) se fait en remplaçant la variable de Laplace  $s$  par le terme  $\frac{1-q^{-1}}{q^{-1} \cdot T_e}$ . On obtient :

$$K_{\text{PID}}(q) = K \cdot \left( 1 + \frac{1}{T_i \cdot \frac{1-q^{-1}}{T_e}} + \frac{T_d \cdot \frac{1-q^{-1}}{T_e}}{1 + \frac{T_d}{N} \cdot \frac{1-q^{-1}}{T_e}} \right) \quad (63)$$

En réarrangeant on obtient la forme générale RST (Figure 30):

$$K_{PID}(q) = K \cdot \frac{r_0 + r_1 \cdot q^{-1} + r_2 \cdot q^{-2}}{(1 - q^{-1}) \cdot (1 + s_1' \cdot q^{-1})}$$

$$\begin{cases} r_0 = K \cdot \left( 1 + \frac{T_e}{T_i} + \frac{N \cdot T_d}{T_d + N \cdot T_e} \right) \\ r_1 = -K \cdot \left( 1 + \frac{T_d}{T_d + N \cdot T_e} + \frac{T_e}{T_i} \cdot \frac{T_d}{T_d + N \cdot T_e} + \frac{N \cdot T_d}{T_d + N \cdot T_e} \right) \\ r_2 = K \cdot \left( \frac{T_d}{T_d + N \cdot T_e} \right) \\ s_1' = \frac{T_d}{T_d + N \cdot T_e} \end{cases} \quad (64)$$

Comme on le voit le correcteur obtenu comporte autant de degrés de liberté que le correcteur PID analogique. On reconnaît au dénominateur le terme  $(1 - q^{-1})$  en facteur qui assure l'effet d'intégration.

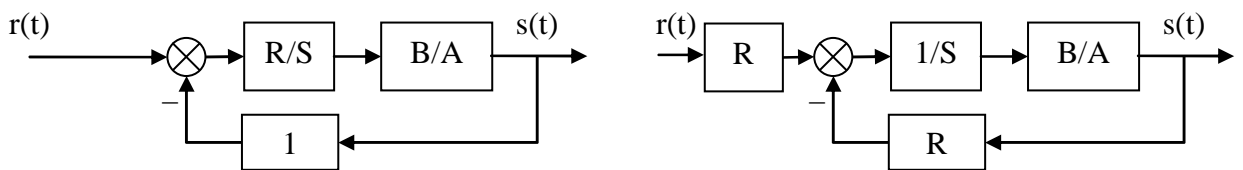


Figure 30 : PID sous forme RST

La fonction de transfert en boucle fermée obtenue est donnée par l'équation suivante :

$$H_{BF} = \frac{B \cdot R}{A \cdot S + B \cdot R} = \frac{B \cdot R}{P} \quad (65)$$

De manière générale le polynôme caractéristique  $P$  fixe les performances visées en boucle fermée. Le calcul de  $R$  et  $S$ ,  $P$  étant fixé et  $A$  et  $B$  étant connus, est réalisé en résolvant l'équation de Bézout  $P = A \cdot S + B \cdot R$ . (voir 9.1 Annexe 1 : Résolution de l'équation polynomiale de Bezout)

On voit que le numérateur  $B \cdot R$  est imposé une fois que  $R$  et  $S$  sont trouvés. Le régulateur introduit donc naturellement des zéros dans la boucle fermée.

Traisons l'exemple entièrement dans le cas où le système est modélisé comme un système du deuxième ordre avec faible retard. La fonction de transfert du procédé est donnée par :

$$H_{BO} = \frac{B}{A} = \frac{b_1 \cdot q^{-1} + b_2 \cdot q^{-2}}{1 + a_1 \cdot q^{-1} + a_2 \cdot q^{-2}} \quad (66)$$

Le correcteur PID rajoutant deux pôles, le polynôme caractéristique en boucle fermée spécifié doit posséder quatre pôles. Le polynôme caractéristique en boucle fermée doit donc être de la forme :

$$\begin{aligned} P &= 1 + p_1 \cdot q^{-1} + p_2 \cdot q^{-2} + p_2 \cdot q^{-3} + p_2 \cdot q^{-4} \\ P &= (1 + p_1' \cdot q^{-1} + p_2' \cdot q^{-2}) \cdot (1 + \alpha_1 \cdot q^{-1}) \cdot (1 + \alpha_2 \cdot q^{-1}) \end{aligned} \quad (67)$$

Et il doit être identifié au produit  $AS + BR$  avec  $R$  et  $S$  ayant les formes données par (64) :

$$\begin{aligned} &(1 + p_1' \cdot q^{-1} + p_2' \cdot q^{-2}) \cdot (1 + \alpha_1 \cdot q^{-1}) \cdot (1 + \alpha_2 \cdot q^{-1}) \\ &= (1 + a_1 \cdot q^{-1} + a_2 \cdot q^{-2}) \cdot (1 - q^{-1}) \cdot (1 + s_1' \cdot q^{-1}) \\ &+ (b_1 \cdot q^{-1} + b_2 \cdot q^{-2}) \cdot (r_0 + r_1 \cdot q^{-1} + r_2 \cdot q^{-2}) \end{aligned} \quad (68)$$

Les coefficients  $p_1'$  et  $p_2'$  définissent un modèle du second ordre aux caractéristiques (pulsation propre  $\omega_0$  et amortissement  $\sigma$  choisis et coefficients déduits conformément au Tableau 3 page 25). Les coefficients  $\alpha_1$  et  $\alpha_2$  correspondent à des pôles auxiliaires aperiodiques. Les choix typiques pour ces coefficients sont :

$$-0.05 < \alpha_1, \alpha_2 < -0.5 \quad (69)$$

On montre que la résolution de (68) est réalisée par une inversion matricielle simple :

$$\begin{aligned} M \cdot x &= p \\ \text{avec} & \quad (70) \\ \begin{cases} x^T = [1 & s_1' & r_0 & r_1 & r_2] \\ p^T = [1 & p_1 & p_2 & p_3 & p_4] \end{cases} \end{aligned}$$

et

$$M = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ a_1' & 1 & b_1 & 0 & 0 \\ a_2' & a_1' & b_2 & b_1 & 0 \\ a_3' & a_2' & 0 & b_2 & b_1 \\ 0 & a_3' & 0 & 0 & b_2 \end{pmatrix} \quad (71)$$

On montre que M est inversible à la condition que A et B soient premiers entre eux, c'est-à-dire que A et B n'aient pas de pôles et de zéros en commun.

(Le problème peut aussi être résolu en utilisant la résolution de l'équation de Bézout).

### 7.3.2 Cas 2

Comme on l'a vu le correcteur PI numérique directement transposé du correcteur PID analogique décrit dans le chapitre 7.3.1 a l'inconvénient de ramener des zéros au dénominateur de la fonction de transfert en boucle fermée (équation 65). Il est possible, grâce à la flexibilité du régulateur numérique d'imposer au correcteur de ne pas ajouter de zéros en boucle fermée. La fonction de transfert désirée en boucle fermée sera donc de la forme :

$$H_{BF} = \frac{P(1)}{B(1)} \cdot \frac{B}{P} \quad (72)$$

Le facteur  $P(1)/B(1)$  garantit le gain statique unitaire du système en boucle fermée (voir page 12). Si le régulateur a la forme générale RST alors les polynômes R S et T doivent vérifier :

$$H_{BF} = \frac{B \cdot T}{A \cdot S + B \cdot R} = \frac{P(1) B}{B(1) P} \quad (73)$$

S et R sont obtenus par la résolution de la même équation que précédemment (68). Le polynôme T est obtenu par simple identification :  $T = R(1)$ .

La structure, sous forme de schéma-bloc est donnée ci-dessous :

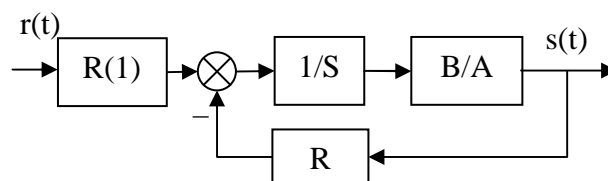


Figure 31 : PID2 sous forme RST

On montre que cette structure de PID, ramené en continu, correspond au PID canonique pour l'action intégrale. L'action proportionnelle et dérivée n'est appliquée que sur le signal y.

La structure du PID2 par rapport au PID1 a pour avantage de ne pas ramener de zéros à la boucle fermée. Cela se traduit par un dépassement moindre en transitoire lorsqu'on applique un échelon de commande. Par contre les deux systèmes réagissent de la même manière à un échelon de perturbation.

## 7.4 Placement de pôles

Les deux structures de PID décrites dans le chapitre précédent sont deux cas particulier d'une stratégie plus générale de placement de pôles.

La synthèse est plus générale que les PID présentés précédemment puisqu'il n'est pas nécessaire de faire d'hypothèses sur le degré de A et B, sur le retard, ni sur la présence ou non de zéros stables ou instables dans le procédé.

La structure du correcteur est la structure type RST telle que présentée dans la Figure 29. Pour la suite des calculs on posera pour la fonction de transfert les conventions suivantes :

$$H_{BO} = \frac{q^{-d} \cdot B(q^{-1})}{A(q^{-1})} \quad (74)$$

$$\begin{cases} A(q^{-1}) = 1 + a_1 \cdot q^{-1} + \dots + a_{n_A} \cdot q^{-n_A} \\ B(q^{-1}) = b_1 \cdot q^{-1} + \dots + b_{n_B} \cdot q^{-n_B} = q^{-1} \cdot B^*(q^{-1}) \end{cases}$$

Compte tenu de la forme RST du régulateur la fonction de transfert en boucle fermée vérifie :

$$H_{BF} = \frac{q^{-d} \cdot B(q^{-1}) \cdot T(q^{-1})}{P(q^{-1})} \quad (75)$$

$$P(q^{-1}) = A(q^{-1}) \cdot S(q^{-1}) + q^{-d} \cdot B(q^{-1}) \cdot R(q^{-1}) = 1 + p_1 \cdot q^{-1} + \dots + p_{n_P} \cdot q^{-n_P}$$

Le polynôme  $P(q^{-1})$  définit donc les pôles de la fonction de transfert en boucle fermée et traduit donc le comportement en *poursuite*.

De la même manière la fonction de transfert entre une perturbation et la sortie (fonction de sensibilité perturbation-sortie) est donnée par :

$$S_{yp} = \frac{A(q^{-1}) \cdot S(q^{-1})}{P(q^{-1})} \quad (76)$$

Ce même polynôme traduit donc aussi le comportement en *régulation*.

Les consignes en poursuite et en régulation peuvent être différentes comme cela est représenté par exemple sur la Figure 32.

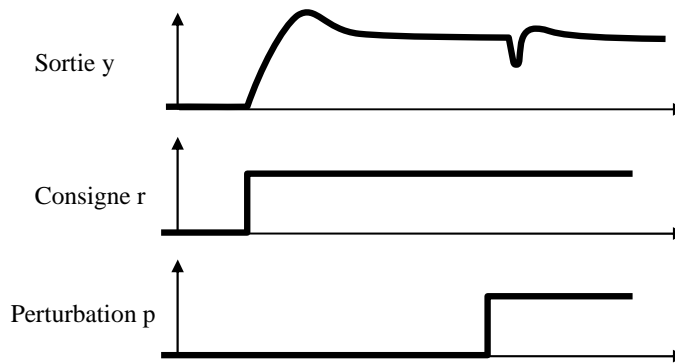


Figure 32 : Performances différentes en régulation et en poursuite

### 7.4.1 Choix des pôles en boucle fermée

De manière générale le comportement en boucle fermée est spécifié sous la forme d'une pulsation propre  $\omega_0$  et d'un amortissement  $\sigma$ , donc une fonction de transfert du second ordre. La spécification étant généralement donnée en continu il est nécessaire de traduire la valeur désirée de  $\omega_0$  et  $\sigma$  en un polynôme  $P_D(q^{-1})$  par discrétisation de la fonction de transfert continue (chapitre 4.5 page 29). Ce polynôme  $P_D$  traduit le comportement dominant du système bouclé.

Si l'on souhaite introduire un effet de filtrage supplémentaire dans certaines bandes de fréquence (réduction des bruits de mesure, adoucissement de la commande, robustesse) cela est décrit par un polynôme  $P_F$  comportant des pôles auxiliaires, plus rapides que les pôles de  $P_D$ .

Le polynôme caractéristique en boucle fermée est donc constitué de pôles dominants et de pôles auxiliaires :

$$P(q^{-1}) = P_D(q^{-1}) \cdot P_F(q^{-1}) \quad (77)$$

### 7.4.2 Régulation : calcul de R et S

Le polynôme  $P(q^{-1})$  étant fixé il s'agit de trouver R et S vérifiant :

$$P(q^{-1}) = A(q^{-1}) \cdot S(q^{-1}) + q^{-d} \cdot B(q^{-1}) \cdot R(q^{-1}) \quad (78)$$

Comme déjà vu au chapitre 7.3.1 on obtient une équation de Bézout qu'il s'agit de résoudre. (Voir aussi 9.1 Annexe 1 : Résolution de l'équation polynomiale de Bezout)

Il est parfois nécessaire de spécifier R et S avec des parties fixes  $H_R$  et  $H_S$ . Dans ce cas l'équation (78) est modifiée et on l'écrit sous la forme:

$$P(q^{-1}) = (A(q^{-1}) \cdot H_S(q^{-1})) \cdot S'(q^{-1}) + (q^{-d} \cdot B(q^{-1}) \cdot H_R(q^{-1})) \cdot R'(q^{-1}) \quad (79)$$

Les inconnues que l'on souhaite trouver sont  $S'$  et  $R'$ . Ceci est nécessaire dans les cas particuliers suivants :

- spécification d'erreur statique nulle,
- rejet d'une perturbation harmonique,
- blocage d'un signal,

Détaillons maintenant ces cas particuliers.

### Spécification d'une erreur statique nulle

On retrouve la nécessité d'avoir un terme intégrateur dans la boucle directe. La spécification se retrouve donc sur  $S$  :

$$S(q^{-1}) = (1 - q^{-1}) \cdot S'(q^{-1}) = H_S(q^{-1}) \cdot S'(q^{-1}) \quad (80)$$

### Rejet d'une perturbation harmonique

Si le système est perturbé par une perturbation harmonique à une fréquence fixe (par exemple 50Hz...) il est possible d'imposer à  $S$  d'avoir un module *nul* ou amorti à cette fréquence. Cela se traduit par la présence dans  $S$  d'une paire de zéros complexe à amortissement nul ou faible à cette fréquence :

$$S(q^{-1}) = (1 + \alpha \cdot q^{-1} + \beta \cdot q^{-2}) \cdot S'(q^{-1}) = H_S(q^{-1}) \cdot S'(q^{-1}) \quad (81)$$

### Blocage d'un signal

Pour certaines application on souhaite que le régulateur "laisse passer" certaines composantes du signal de sortie, c'est-à-dire qu'il ne réagisse pas à ces composantes. Cela se traduit par  $R=0$  à ces fréquences. On retrouve la même forme que précédemment :

$$R(q^{-1}) = (1 + \alpha \cdot q^{-1} + \beta \cdot q^{-2}) \cdot R'(q^{-1}) = H_R(q^{-1}) \cdot R'(q^{-1}) \quad (82)$$

### 7.4.3 Poursuite : calcul de T

Le schéma-bloc de l'asservissement est donné sur la Figure 33. Le terme  $T$  est décomposé en un terme  $T'$  et un terme  $B^*/A^*$ .

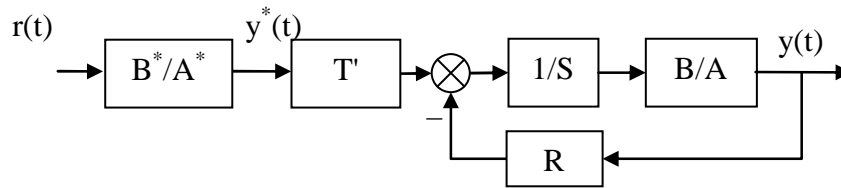


Figure 33 : placement de pôle

L'idée est de régler  $B^*$  et  $A^*$  de telle manière que le signal  $y^*$  ait la dynamique de poursuite désirée (par exemple un second ordre avec pulsation propre et amortissement donné) puis de régler  $T'$  de telle manière que le transfert entre  $y^*$  et  $y$  soit le plus "transparent" possible.

### Réglage de $T'$

Le transfert entre  $y^*$  et  $y$  est :

$$S_{yy^*}(q^{-1}) = T'(q^{-1}) \frac{q^{-d} \cdot B(q^{-1})}{P(q^{-1})} \quad (83)$$

Pour assurer le transfert le plus "transparent" possible il faut donc choisir

$$T'(q^{-1}) = \frac{P(q^{-1})}{B(1)} \quad (84)$$

Remarques :

- Si  $B(1)=0$  alors on remplace  $B(1)$  par 1 dans (84)
- De même dans (84)  $P(q^{-1})$  peut être remplacé par  $P_D$  correspondant à la dynamique dominante.

### Réglage de $B^*$ et $A^*$

$B^*$  et  $A^*$  sont choisis de telle manière que la fonction de transfert  $B^*/A^*$  corresponde à la dynamique de poursuite souhaitée.

## 7.5 Poursuite et régulation à objectifs indépendants

Cette méthode ressemble à la méthode de placement de pôles à ceci près que l'on va effectuer une simplification des zéros de  $B^*(q^{-1})$  : cela ne s'applique donc qu'aux systèmes avec des pôles stables. Plus précisément il faut qu'ils aient un amortissement suffisant ( $> 0,2$ ) de manière à ne pas être trop près du cercle unité.

La structure de l'asservissement est toujours celle donnée sur la Figure 33.



**Choix de P**

$P(q^{-1})$  est choisi de manière à simplifier  $B^*(q^{-1})$  (rappel : la forme de B est donnée dans l'équation 74). Comme indiqué dans le chapitre 7.4.2 page 60 il s'agit donc d'imposer les zéros de  $B(q^{-1})$  dans  $H_R(q^{-1})$  et  $H_S(q^{-1})$ .

**Choix de T**

Si l'on choisit  $T(q^{-1}) = P(q^{-1})$  alors le transfert entre  $y^*$  et  $y$  est :

$$S_{yy^*}(q^{-1}) = q^{-(d+1)} \quad (85)$$

**Réglage de  $B^*$  et  $A^*$** 

$B^*$  et  $A^*$  sont choisis de telle manière que la fonction de transfert  $B^*/A^*$  corresponde à la dynamique de poursuite souhaitée.



## 8. TECHNOLOGIE DE LA COMMANDE PAR ORDINATEUR

Du point de vue de l'automaticien le ordinateur est relié au système physique à piloter par un certain nombre d'intermédiaires plus ou moins gênants :

- les interfaces analogiques et numériques,
- le transport de l'information par bus et protocole,
- l'utilisation éventuelle de capteurs ou d'actionneurs « intelligents ».

Il ne s'agit pas ici de faire un exposé complet sur ces trois aspects mais de poser les problèmes qui se posent à l'automaticien lorsque le système qu'il doit commander est perturbé par ces éléments.

Enfin ce chapitre se termine sur les notions de « Hardware In the Loop » et de « prototypage rapide » qui sont partie intégrante des processus de réalisation de systèmes de contrôle.

### 8.1 Calculateurs et interfaces numériques / analogiques

Le premier problème à prendre en compte est la quantification de l'information. Du point de vue du ordinateur d'une part et du point de vue des convertisseurs d'autre part. Dans un premier temps, considérons les aspects performance de calcul lié à la nature du codage des nombres dans le ordinateur.

#### Les performances liées au types de variables

Prenons l'exemple où le concepteur de la loi de commande souhaite réaliser un correcteur de type proportionnel. Entre l'acquisition des mesures et l'écriture de la consigne il y aura la réalisation d'une de commande du type :

$$y = K \cdot (\text{consigne} - \text{mesure}) \quad (86)$$

En utilisant un langage de programmation évolué (par exemple le C) la ligne de code sera la transcription littérale de (86). Le programmeur devra décider de la définition et du type des variables, principalement du nombre d'octets utilisés pour stocker ses variables. Le Tableau 5 résume par exemple les types utilisés en langage C (norme ANSI)

Tableau 5 : Types ANSI C

Types entiers	char	8 bits
	short int	16 bits
	long int	32 bits
Types flottants (mantisse et exposant)	float	32 bits
	double	64 bits
	longdouble	128 bits

Une fois le code écrit en C l'exécution du programme sur un processeur cible ne peut se faire que par l'intermédiaire d'une compilation du programme. A chaque processeur est donc adapté une version de compilateur.

Les performances du programme (sa rapidité d'exécution) dépendront donc en partie de la qualité du compilateur. (Par exemple, une multiplication s'effectue t'elle bêtement par un décalage de tous les bits ou bien des solutions astucieuses sont-elles implémentées pour arrêter le décalage selon le contexte ?)

Le principal déterminant de la performance est cependant le processeur : son type (8 bits, 16 bits ou 64 bits) et son cadencement (en MHz ou GHz) sont les deux plus importants paramètres. Coder les variables sur 64 bits (double) dans le programme coûtera cher en temps de calcul si le processeur est un 8 bits. Le type de mémoire et leurs temps d'accès peuvent aussi être déterminants. Enfin, certains processeurs disposent d'un processeur spécifiquement dédié aux opérations mathématiques (coprocesseur mathématique).

Il n'est pas possible de dire exactement, pour tel type de compilateur associé à tel type de processeur, le temps que prendra par exemple une multiplication. En pratique on effectue une série de tests types (benchmark) et on affiche un temps moyen d'exécution.

En résumé, on peut être tenté, pour ne pas s'embêter, de coder toutes les variables en virgule flottante. Mais il faut bien savoir que cela a un coût en terme de temps de calcul, surtout si le processeur n'est pas adapté. Pour fixer les idées on peut dire qu'il y a un rapport 20 entre une addition réalisée en "short" et une addition réalisée en virgule flottante.

Enfin, l'utilisation de fonctions mathématiques (sinus, exponentielles, etc...) peut aussi avoir un coût très lourd si le travail n'est pas réalisé par un coprocesseur mathématique dédié. Le compilateur et sa librairie mathématique joue là encore un grand rôle.

### La quantification imposée par le processeur

Au-delà de l'aspect performance et temps de calcul, la quantification de l'information doit parfois aussi être prise en compte. Rappelons qu'un entier codé sur N bits peut prendre seulement  $2^N - 1$  valeurs différentes. Avant d'effectuer les calculs mathématiques dans le corps du programme il est donc judicieux d'appliquer un

facteur d'échelle de manière à utiliser l'amplitude maximale possible (entre 0 et  $2^N - 1$ ).

### **La quantification imposée par les conversions analogique et numérique**

Les convertisseurs analogique-numérique et numérique-analogique sont principalement caractérisés par leur amplitude de signal analogique d'entrée (généralement en volts, par exemple +/- 10V) et par leur résolution en bits (8 ou 10 bits pour les plus courants, jusqu'à 16 bits pour les convertisseurs haut de gamme).

Prenons l'exemple d'un capteur gyromètre transformant une vitesse de rotation en tension. Compte tenu de l'application le signal utile varie entre 2,4V et 2,6V. Si le signal mesuré est branché directement sur un convertisseur 0-10V 10bits alors un problème de conditionnement du signal se pose. En effet la quantification est de  $\frac{10V}{2^{10}-1} = 0,098V$ . Les 0,2V de signal utile seront donc discrétisés en  $\frac{0,2}{0,098} = 20$  intervalles seulement !

Si la discrétisation du signal utile est trop grossière cela peut se traduire par une diminution des performances de l'asservissement, voire une instabilité. Le système n'est en effet plus linéaire et il n'existe pas d'outils théoriques permettant de prévoir les performances du système bouclé.

## **8.2 Transport de l'information : les bus et les protocoles**

Pas question de faire ni même d'ébaucher un cours sur les bus et les protocoles ici. Mais on va quand même donner quelques indications. L'automaticien a l'habitude de simplifier le problème qu'il se pose en dessinant un schéma simplifié de son problème d'asservissement (du type de la Figure 20 page 45) pour pouvoir appliquer les théories de la commande. Cependant il arrive bien souvent que l'information transite à l'intérieur du système, soit entre différents ordinateurs effectuant différentes parties du travail soit entre ordinateur et périphériques, par l'intermédiaire de bus informatique. A chaque type de bus correspond un ou plusieurs protocoles qui peuvent poser des problèmes pour l'automaticien :

- les données sont-elles discrétisées ? (le protocole peut par exemple ne permettre de transmettre que des paquets de données de 16 bits),
- le bus induit-t'il un retard ? (le composant mettant en forme les données du ordinateur pour les publier sur le bus a certaines performances qui peuvent induire un délai)
- le bus est-il déterministe ? (Certains bus ne garantissent pas que les données arriveront en un temps donné),
- le bus est-il sûr ? (Certains bus ne garantissent pas non plus que les données arriveront bien)

### **8.3 Capteurs et actionneurs "intelligents", "smart sensors"**

De plus en plus souvent les capteurs et les actionneurs sont associés à leur électronique de puissance, électronique de commande et de conditionnement de signal. L'interface avec le capteur ou l'actionneur se fait donc via un protocole de communication (par exemple via un bus RS232).

L'intérêt est de véhiculer non pas des signaux analogiques, tensions ou courants, mais des signaux numériques beaucoup moins sensibles aux éventuelles perturbations.

On retrouve donc, entre le capteur ou l'actionneur et le ordinateur, la présence d'un bus et d'un protocole de communication.

### **8.4 La philosophie "hardware in the loop"**

Les systèmes embarqués sont de plus en plus complexes et touchent des domaines de plus en plus variés : automobile, satellites, aéronautique, UAV, systèmes d'arme, sous-marins, systèmes, etc... Les systèmes à piloter ainsi que les systèmes de commande sont de plus en plus complexes.

Les techniques de simulation « Hardware In the Loop » (HIL) sont de plus en plus utilisées dans le développement et dans le test de ces systèmes embarqués. L'idée est de développer entièrement le système de contrôle sur une plateforme de test en utilisant le maximum de composants réels et en ne remplaçant que le minimum par la simulation. Par exemple en aéronautique l'« Iron Bird » regroupe dans un hangar tous les systèmes réels de l'avion en cours de développement. Seule la mécanique du vol est simulée : les sorties des actionneurs sont mesurées, rentrées dans un modèle de simulation numérique, puis les mesures effectuées normalement par les capteurs sont calculées et restituées par le modèle.

L'intérêt est de diminuer de manière importante les coûts et les délais en économisant le plus grand nombre possible de tests sur le système réel en situation. Lors des premiers tests réels les techniques « Hardware In the Loop » permettent d'avoir effectué la plus grande part du développement des lois de commande.

### **8.5 Le prototypage rapide en automatique**

Un peu dans le même ordre d'idée que le « Hardware In the Loop » le prototypage rapide permet de réaliser la conception des lois de commande en simulation puis de transporter rapidement (en un simple click !) le code du contrôleur simulé vers la cible temps réel implémentée sur le système réel.

La société Mathworks [9] investit massivement dans ce domaine en ayant enrichi son logiciel de calcul scientifique (Matlab) avec un logiciel de simulation graphique très évolué (Simulink) puis en développant des outils de prototypage rapide associés (Real Time Workshop, Embedded Target For...).

## 9. ANNEXE

### 9.1 Annexe 1 : Résolution de l'équation polynomiale de Bezout

Il s'agit de trouver R et S vérifiant :

$$P(q^{-1}) = A(q^{-1}).S(q^{-1}) + B(q^{-1}).R(q^{-1})$$

Où P, A et B sont connus.

Dans le cas où P, A, B, R et S sont des nombres entiers la solution de l'algorithme est donnée par l'algorithme de division Euclidienne étendu [8].

Dans le cas où P, A, B, R et S sont des polynômes, l'algorithme est donné ci-dessous, programmé en matlab ([1]).

```
function [Rp,Sp,nrp,nsp]=bezoutd(A,B,Hs,Hr,P)
%function [Rp,Sp,nrp,nsp]=bezoutd(A,B,Hs,Hr,P)
%solves AHsSp+BHRp=P by coefficient comparison. Delay and discretization delay need to be integrated in B
%(zeros at the beginning).
%
%inputs:
%A=[a0 a1 ... aNa] ... vector of model denominator coefficients A=a0 + a1z^(-1) + a2z^(-2) +...+ aNaz^(-Na)
%B=[b0 b1 ... bNb] ... vector of model numerator coefficients B=b0 + b1z^(-1) + b2z^(-2) +...+ bNbz^(-Nb)
%Hs=[hs0 hs1 ... hsNhs] ... vector of controller denominator fixed part Hs=hs0 + hs1z^(-1) +...+ hsNhsz^(-Nhs)
%Hr=[hr0 hr1 ... hrNhr] ... vector of controller denominator fixed part Hr=hr0 + hr1z^(-1) +...+ hrNhrz^(-Nhr)
%P=[p0 p1 ... pNp] ... vector of desired polynomial coefficients P=p0 + p1z^(-1) + p2z^(-2) +...+ pNpz^(-Np)
%outputs:
%Rp=[rp0 rp1 rp2 ...] ... vector of coefficients for resulted controller numerator
%Sp=[sp0 sp1 sp2 ...] ... vector of coefficients for resulted controller denominator
%nrp ... order of Rp
%nsp ... order of Sp
%
%written by: J. Langer, I.D. Landau, H. Prochazka
%7th june 2002

PRECISION=1e-16;

D=size(A);
if D(1)>1, A=A'; end;
```

```

D=size(B);
if D(1)>1, B=B'; end;
D=size(Hs);
if D(1)>1, Hs=Hs'; end;
if D(1)==0, Hs=1; end;
D=size(Hr);
if D(1)>1, Hr=Hr'; end;
if D(1)==0, Hr=1;end;
D=size(P);
if D(1)>1, P=P'; end;

na=length(A)-1;
nb=length(B)-1;
np=length(P)-1;
nhs=length(Hs)-1;
nhr=length(Hr)-1;

if (nhs>0), Ah=real(convz(A,Hs)); else Ah=A*Hs; end; % Ah = A * Hs
nah=length(Ah) -1;
if (nhr>0), Bh=real(convz(B,Hr)); else Bh=B*Hr; end;
nbh=length(Bh) -1;
if (np>nah+nbh-1), disp('Bezout error: too many poles');end;

% increase size of P using zeros if necessary
if (np<nah+nbh-1),
%
% set remaining poles onto a circle with radius rmin
P=[P zeros(1,nah+nbh-1-np)];
rootsPdes=roots(P);
nextra=nah+nbh-1-np;
rmin=1e-16;
angle=[0:nextra-1]'/nextra*2*pi;
j=sqrt(-1);
rootsPextra=rmin*exp(j*angle);
P=poly([rootsPdes;rootsPextra]);
np=nah+nbh-1;

end;
P,

nsp=nbh-1;
nrp=nah-1;

%matrix is smaller than vector PD

```



```

if (np>nah+nbh-1),
    disp('Order of model denominator is too low! Add a polynom of higher order to Hs or Hr. ');
end;

% ns=nsp+nhs
% nr=nrp+nhr

M=[];
for j=1:nsp+1,
    V=[];
    if (j>1), V=[V ; zeros(j-1,1)]; end;% zeros in front of Ah
    V=[V ; Ah'];% Ah
    if (j<=nsp), V=[V ; zeros(nsp+1-j,1)]; end;% zeros behind Ah
    if (length(V)~=nah+nbh), disp('bezoutb: error V'); end;
    M=[M V]; % add one column to M
end;

for j=1:nrp+1,
    V=[];
    if (j>1), V=[V ; zeros(j-1,1)]; end;
    V=[V ; Bh'];
    if (j<=nrp), V=[V ; zeros(nrp+1-j,1)]; end;
    if (length(V)~=nah+nbh), disp('bezoutb: error V'); end;
    M=[M V];
end;

D=size(M);
if (D(1)~=nah+nbh), disp('bezoutb: error size M row'); end;
if (D(2)~=nah+nbh), disp('bezoutb: error size M column'); end;

% make P column vector
P=P';

global M1;
M1=M;

X= M\P;
% coefficients are real values
X=real(X);

% make X row vector
X=X';

```

## Commande des systèmes par ordinateur

```
Sp=X(1:nsp+1);  
Rp=X(nsp+2:nsp+nrp+2);
```

## 10. BIBLIOGRAPHIE

- [1] I. D. Landau, "*Commande des systèmes*", Edition Lavoisier 2002.
- [2] C. Nouals, "Théorie du Signal", Notes de cours ENSICA
- [3] F. Vincent, "*Traitement Numérique du Signal*", Notes de cours ENSICA.
- [4] D. Arzelier, "*Représentation et analyse des systèmes, notes de cours*", Edition ENSICA. Voir aussi <http://www.laas.fr/~arzelier/>
- [5] J. Bordeneuve-Guibé, "*Commande des systèmes linéaires*", Notes de cours ENSICA.
- [6] Y. Briere, "*Identification*", Notes de cours ENSICA.
- [7] L. Ljung, "*System Identification, Theory for the User*", Prentice Hall 1987.
- [8] Wikipedia , équation de Bezout : [http://en.wikipedia.org/wiki/Bézout's\\_identity](http://en.wikipedia.org/wiki/Bézout's_identity)
- [9] <http://www.mathworks.com>



# 11. INDEX

<b>A</b>	
actionneur intelligent .....	21
amortissement .....	15
ANSI C .....	60
Approximation numérique d'équation différentielle .....	28
ARMAX .....	35
<b>B</b>	
BJ35 .....	
Blocage d'un signal .....	56
bloqueurs d'ordre 0 .....	22
BOZ .....	22
bus.....	21, 61
<b>C</b>	
CAN.....	21
capteur intelligent .....	21
causal .....	6, 9
Causalité .....	10
CNA.....	21
compilateur .....	60
convertisseur analogique / numérique.....	21
convertisseur numérique / analogique.....	21
convertisseurs analogique-numérique .....	61
coprocesseur mathématique .....	60
Correspondances pôles - zéros.....	31
<b>D</b>	
Diagramme de Bode .....	18
Discrétisation de fonction de transfert continues .....	27
division polynomiale.....	6
<b>E</b>	
échelon unitaire.....	11
équation d'état .....	17
Erreur statique.....	42
Estimation de la réponse fréquentielle .....	39
Estimation de la réponse impulsionnelle.....	38
Euler .....	29
<b>F</b>	
facteur d'échelle .....	61
facteur d'oubli .....	36
<i>fenêtre de Hamming</i> .....	39
filtre anti-repliement .....	25
fonction de sensibilité .....	42
fonction de transfert discrète.....	9
<b>G</b>	
gain statique.....	11
gain statique unitaire.....	42
<b>I</b>	
Identification.....	32
Identification non paramétrique .....	38
intercorrélation.....	38
<b>L</b>	
lieu des racines.....	43
lieu d'Evans.....	43
LTI.....	8
<b>M</b>	
marge de gain.....	47
marge de module.....	46
marge de phase .....	47
méthode des moindres carrés récursifs .....	32
Méthode des trapèzes.....	29
modèle ARX .....	32
<b>O</b>	
OE.....	35
one step ahead prediction.....	35
opérateur de retard .....	9
<b>P</b>	
période d'échantillonnage (choix) .....	26
Placement de pôles .....	54
Pôles .....	11
Poursuite .....	56
<i>prédiction</i> .....	32
protocoles.....	61
Pseudo Random Binary Signal .....	34
pulsation .....	15
<b>Q</b>	
quantification .....	23, 24, 60
<b>R</b>	
R,S,T.....	41
rapidité.....	15
rayon de convergence .....	5, 9
Régulation.....	55

Rejet de perturbations .....	42
Rejet d'une perturbation harmonique .....	56
réponse à un échelon.....	11
réponse fréquentielle.....	18
Réponse impulsionnelle.....	10
Réponses indicielle .....	10
retard.....	23
root locus .....	43

### S

signal discret .....	5
smart sensors.....	62
stabilité.....	11
Systèmes du premier ordre.....	14
Systèmes du second ordre.....	15

### T

Théorème de Nyquist.....	46
théorème de Shannon.....	19, 24
théorème du retard .....	6
Transformée en z .....	5
transformée unilatérale .....	5
Tustin.....	29

### V

virgule flottante.....	60
------------------------	----

### Z

zero holder hold .....	23
zéros.....	11
zoh .....	23

# Commande des systèmes par ordinateur

Yves Briere  
ENSICA

Notes de cours  
2007





# TABLE DES MATIERES

Table des matières .....	1
1. Introduction .....	1
2. Rappels et prérequis .....	3
3. Signaux et systèmes discrets .....	5
3.1 Signaux discrets .....	5
3.1.1 Transformée en z .....	5
3.1.2 Signaux types .....	7
3.2 Systèmes discrets .....	9
3.2.1 Fonction de transfert discrète, définition .....	9
3.2.2 Notation "opérateur de retard $q^{-1}$ " .....	10
3.2.3 Réponse impulsionnelle .....	11
3.2.4 Causalité .....	11
3.2.5 Réponses indicielle .....	11
3.2.6 Régime stationnaire (gain statique) .....	12
3.2.7 Pôles, zéros, stabilité .....	12
3.2.8 Systèmes du premier ordre .....	15
3.2.9 Systèmes du second ordre .....	16
3.2.10 Calcul itératif de la réponse d'un système .....	18
3.2.11 Représentation par équation d'état .....	18
3.3 Réponse fréquentielle des systèmes échantillonnés .....	19
3.3.1 Diagramme de Bode d'un système échantillonné .....	19
3.3.2 Où l'on voit apparaître le théorème de Shannon .....	20
4. Systèmes continus échantillonnés .....	23
4.1 Structure des systèmes commandés par ordinateur .....	23
4.2 Échantillonnage des systèmes continus .....	24
4.3 Effets de l'échantillonnage .....	25
4.4 Où l'on revoit le théorème de Shannon .....	26
4.5 Discrétisation de fonction de transfert continues .....	29
4.5.1 Approximation numérique d'équation différentielle .....	30
4.5.2 Superposition des réponses indicielles .....	32
4.5.3 Correspondances pôles - zéros .....	33
4.5.4 Bilan .....	33
5. Identification des systèmes échantillonnés .....	35
5.1 Identification d'un modèle ARX par la méthode des moindres carrés récurrents .....	35
5.2 L'identification paramétrique en général et en pratique .....	37

5.2.1	Choix du signal d'excitation, traitement des mesures .....	37
5.2.2	Choix d'une structure de modèle.....	38
5.2.3	Choix d'un critère à minimiser.....	38
5.2.4	Choix d'un algorithme.....	38
5.2.5	Validation.....	39
5.2.6	Utilisation de logiciels .....	39
5.3	Identification "simple" .....	39
5.4	Identification non paramétrique .....	41
5.4.1	Estimation de la réponse impulsionnelle .....	41
5.4.2	Estimation de la réponse fréquentielle.....	42
6.	Systèmes bouclés .....	45
6.1	Fonction de transfert en boucle fermée .....	45
6.2	Erreur statique .....	46
6.3	Rejet de perturbations.....	46
6.4	Stabilité des systèmes à simple gain de rétroaction.....	47
6.5	Analyse des systèmes bouclés dans le domaine fréquentiel : stabilité.....	49
6.5.1	Théorème de Nyquist dans le cas où le système est stable en boucle ouverte (critère du revers) .....	50
6.5.2	Théorème de Nyquist dans le cas où le système est instable en boucle ouverte .....	50
6.5.3	Interprétations du lieu de Nyquist.....	50
7.	Méthodes de régulation des systèmes échantillonnés .....	53
7.1	Introduction .....	53
7.2	Structure d'un système commandé par ordinateur, hypothèses.....	54
7.2.1	Discretisation, choix de la période d'échantillonnage.....	54
7.2.2	Discretisation d'un correcteur continu .....	55
7.3	Régulateur PID numérique .....	55
7.3.1	Cas 1 .....	55
7.3.2	Cas 2 .....	58
7.4	Placement de pôles .....	59
7.4.1	Choix des pôles en boucle fermée .....	60
7.4.2	Régulation : calcul de R et S.....	60
7.4.3	Poursuite : calcul de T .....	61
7.5	Poursuite et régulation à objectifs indépendants .....	62
8.	Technologie de la commande par ordinateur.....	65
8.1	Calculateurs et interfaces numériques / analogiques.....	65
8.2	Transport de l'information : les bus et les protocoles .....	67
8.3	Capteurs et actionneurs "intelligents", "smart sensors" .....	68
8.4	La philosophie "hardware in the loop" .....	68
8.5	Le prototypage rapide en automatique .....	68
9.	Annexe .....	69
9.1	Annexe 1 : Résolution de l'équation polynomiale de Bezout.....	69
10.	Bibliographie.....	73
11.	Index.....	1





# 1. INTRODUCTION



Figure 1 : commande d'un systèmes de télémédecine par ordinateur, vu par Claude Serre



## 2. RAPPELS ET PREREQUIS

Ce polycopié ne comporte pas de rappels sur les systèmes continus. S'adressant aux étudiants en deuxième année de l'ENSICA il suppose connu les enseignements de première année et de seconde année en automatique et traitement du signal.

En première année le cours de théorie du signal [2] a rendu familier les notions de :

- transformée de Fourier, transformée de Laplace
- corrélation, convolution

Le cours de traitement numérique du signal [3] a abordé en plus :

- la discrétisation et la quantification des signaux.
- la transformée en z et la transformée de Fourier discrète.
- les filtres numériques RIF et RII.

En automatique [4] les bases de la représentation et de l'analyse des systèmes continus ont été posées. On ne reviendra pas dans ce cours sur les notions de :

- modèles mathématique des systèmes linéaires à temps continus à temps invariant (LTI) : fonction de transfert, équation d'état, pôles, zéros et dynamique.
- commandabilité et observabilité.
- stabilité des systèmes en boucle ouverte : pôles, critères de Routh-Hurwitz.
- stabilité des systèmes bouclés : critère de Nyquist, critère de Bode.
- lieu des racines.

En deuxième année les notions suivantes sont acquises [5] :

- synthèse des correcteurs par méthodes fréquentielles (avance et retard de phase, PID)
- commande par retour d'état (placement de pôles), commande Linéaire Quadratique (LQG)
- reconstruction de l'état par Observateur, observateur optimal de Kalman.





## 3. SIGNAUX ET SYSTEMES DISCRETS

### 3.1 Signaux discrets

#### 3.1.1 Transformée en z

Pour une approche mathématiquement rigoureuse de représentation des signaux discrets par la transformée en z consulter [3] et sa bibliographie. Dans ce cours on se cantonne à une approche utilitariste des outils et méthodes mathématiques tout en essayant de ne pas contredire fondamentalement la rigueur mathématique.

Un signal discret  $s(k)$  est une séquence de nombres réels appelés échantillons. L'indice  $k$  est généralement associé à un instant d'échantillonnage :  $t = k.T_e$ . La période d'échantillonnage  $T_e$  est supposée constante. Par définition la transformée en z du signal  $s(k)$  est la fonction de la variable  $z$ , notée abusivement  $s(z)$  :

$$s(z) = Z(s(k)) = \sum_{k=0}^{\infty} s(k) \cdot z^{-k} \quad (1)$$

La transformée définie ici est plus précisément appelée transformée unilatérale puisque les termes correspondant à  $k < 0$  ne sont pas pris en compte. Ceci n'a pas d'importance puisque les signaux que nous utilisons sont systématiquement causaux<sup>1</sup> :  $s(k) = 0$  pour  $k < 0$ .

Il y a bien évidemment un lien avec la transformée de Laplace et la transformée de Fourier (voir [3])

La fonction de la variable complexe  $s(z)$  est une série et ne converge pas forcément pour toute valeur de  $z$ . En toute rigueur il faudrait associer le rayon de convergence à chaque fonction en  $z$  :  $\{s(z) \text{ converge pour } |z| > \alpha\}$ .

Les automaticiens pour qui la transformée en z n'est bien souvent qu'une façon bien commode de regrouper des termes de récurrence utilisent souvent la pseudo-variable<sup>2</sup>  $q$ .

A partir d'une transformée en z  $s(z)$  il est possible de retrouver la séquence  $s(k)$  correspondante en appliquant la transformation inverse, dont on ne donnera pas la formule ici. A la place de faire appel à cette formule compliquée il est possible d'utiliser d'autres méthodes pour retrouver la séquence  $s(k)$  lorsque la fonction  $s(z)$  est donnée sous forme de fraction rationnelle :

<sup>1</sup> Pour une justification du terme « causal » voir page 11

<sup>2</sup> Voir page 10

- division polynomiale
- décomposition de la fraction rationnelle en éléments simples puis utilisation de transformations élémentaires telles que celles données dans le Tableau 2

La propriété fondamentale de la transformée en  $z$  est qu'elle est linéaire (la transformée en  $z$  d'une composition linéaire de signaux est la composition linéaire de la transformée en  $z$  de chaque signal). Les autres propriétés fondamentales sont données dans le Tableau 2.

Tableau 1 : propriétés de la transformée en  $z$

1	$s(z) = Z(s(k)) = \sum_{k=0}^{\infty} s(k) \cdot z^{-k}$
2	$Z(\alpha \cdot s_1(k) + \beta \cdot s_2(k)) = \alpha \cdot Z(s_1(k)) + \beta \cdot Z(s_2(k))$
3	$Z(k \cdot s(k)) = -z \cdot \frac{ds(z)}{dz}$
4	$Z(c^k \cdot s(k)) = s\left(\frac{z}{c}\right)$
5	$Z(s(k-1)) = s(k=-1) + z^{-1} \cdot s(z)$
6	$Z(s(k+1)) = z \cdot s(z) - z \cdot s(k=0)$
7	$s(k=0) = \lim_{z \rightarrow \infty} (s(z))$
8	$s(k \rightarrow \infty) = \lim_{z \rightarrow 1} \left( \frac{z-1}{z} \cdot s(z) \right)$

Remarque 1 : la propriété 5 ne prend pas en compte le fait que le signal est causal : en pratique  $s(k=-1) = 0$ .

Remarque 2 : la propriété 5 justifie le fait que le terme  $z^{-1}$  soit appelé "opérateur de retard unitaire", cette propriété est aussi appelée "théorème du retard".

Remarque 3 : les propriétés 7 et 8 sont données sous réserve de convergence.

Transformées inverses

### Exemple 1 : transformée inverse par division polynomiale

On considère un signal  $s(k)$  dont la transformée en  $z$  est donnée par la fraction rationnelle :

$$S(z) = \frac{0.1 \cdot z^2 + 0.1 \cdot z}{z^3 - 2.6 \cdot z^2 + 2.2 \cdot z - 0.6} \quad (3.1)$$

L'objectif est d'obtenir  $S(z)$  sous forme polynomiale :

$$S(z) = x_0 + x_1 \cdot z^{-1} + x_2 \cdot z^{-2} + \dots \text{etc.} \tag{3.2}$$

Cette suite est généralement infinie, la réponse temporelle est donnée par la succession des termes  $\{x_i\}$ . En traitant l'équation (3.1) comme une simple division l'algorithme de division classique (celui appris en CM2) permet d'obtenir par récurrence tous les termes de la suite (3.2) :

$0.1 \cdot z^2$	$+ 0.1 \cdot z$				$z^3 - 2.6 \cdot z^2 + 2.2 \cdot z - 0.6$
$0.1 \cdot z^2$	$- 0.26 \cdot z$	$+ 0.22$	$- 0.06 \cdot z^{-1}$		
$0$	$0.36 \cdot z$	$- 0.22$	$0.06 \cdot z^{-1}$		
	$0.36 \cdot z$	$- 0.936$	$0.792 \cdot z^{-1}$	$- 0.216 \cdot z^{-2}$	
$0$	$0.716$	$- 0.732 \cdot z^{-1}$	$\dots$	$\dots$	

Les éléments successifs de la suite  $s(k)$  sont donc  $s(k) = \{0, 0.1, 0.36, 0.716, \dots\}$ . Pour obtenir les termes suivants il faudrait continuer la division.

**Exemple 2 : transformée inverse par décomposition en éléments simples**

La deuxième méthode présentée consiste à décomposer la fraction rationnelle (3.1) en éléments simples puis d'utiliser les tables de transformées en  $z$  (Tableau 2). Pour notre exemple cela donne :

$$S(z) = \frac{0.1 \cdot z^2 + 0.1 \cdot z}{z^3 - 2.6 \cdot z^2 + 2.2 \cdot z - 0.6} = \frac{-z}{z-1} + \frac{0.5 \cdot z}{(z-1)^2} + \frac{z}{z-0.6}$$

La réponse temporelle est donc composée des trois éléments :

$$s(k) = -(1 \ 1 \ 1 \ \dots) + 0.5 \cdot (1 \ 2 \ 3 \ \dots) + (1 \ 0.6 \ 0.6^2 \ \dots)$$

$$\text{soit : } s(k) = \begin{cases} s(0) = -1 + 0.5 + 1 \\ s(2) = -1 + 0.5 \cdot 2 + 0.6 \cdot 0.5 \\ s(3) = -1 + 0.5 \cdot 3 + 0.6^2 \\ s(4) = \dots \end{cases}$$

**3.1.2 Signaux types**

Le Tableau 2 donne les transformée en  $z$  de certains signaux fondamentaux. La Figure 2 représente l'allure de quelques signaux types.

Tableau 2 : transformée en z élémentaires

$s(k)$	$S(z)$
signal temporel	Transformée en z
$\delta(k)$ : impulsion unité	1
$u(k)$ : échelon unité	$\frac{z}{z-1}$
$k \cdot u(k)$	$\frac{z}{(z-1)^2}$
$c^k \cdot u(k)$	$\frac{z}{z-c}$
$k \cdot a^k \cdot u(k)$	$\frac{a \cdot z}{(z-a)^2}$
$\sin(\omega \cdot k) \cdot u(k)$	$\frac{z \cdot \sin(\omega)}{z^2 - 2 \cdot z \cdot \sin(\omega) + 1}$
$\cos(\omega \cdot k) \cdot u(k)$	$\frac{z \cdot (z - \cos(\omega))}{z^2 - 2 \cdot z \cdot \sin(\omega) + 1}$
$a^k \cdot \sin(\omega \cdot k) \cdot u(k)$	$\frac{a \cdot z \cdot \sin(\omega)}{z^2 - 2 \cdot a \cdot \cos(\omega) \cdot z + a^2}$
$a^k \cdot \cos(\omega \cdot k) \cdot u(k)$	$\frac{z \cdot (z - a \cdot \cos(\omega))}{z^2 - 2 \cdot a \cdot \cos(\omega) \cdot z + a^2}$

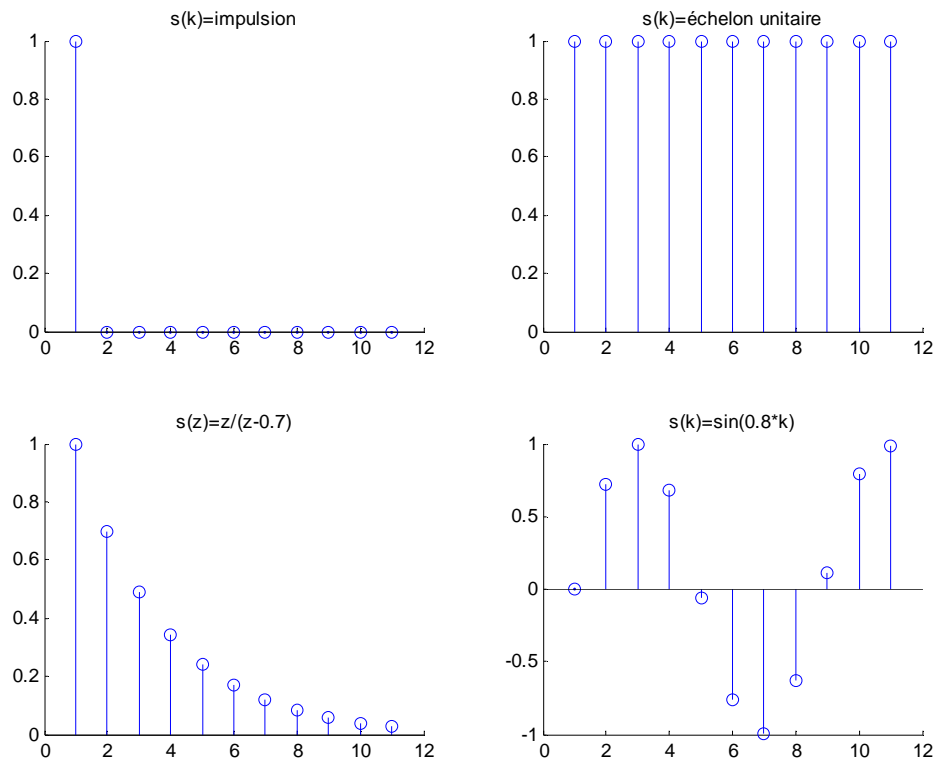


Figure 2 : réponses de quelques signaux types

## 3.2 Systèmes discrets

### 3.2.1 Fonction de transfert discrète, définition

On considère maintenant un système discret linéaire et invariant dans le temps (LTI) représenté sur la Figure 3.

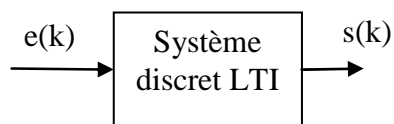


Figure 3 : système discret, notations

La relation entre la séquence d'entrée et la séquence de sortie est linéaire et définie par un ensemble de coefficients constants  $a_{1:n}$  et  $b_{1:m}$  :

$$\begin{aligned} s(k) + a_1 \cdot s(k-1) + a_2 \cdot s(k-2) + \dots + a_n \cdot s(k-n) = \\ b_0 \cdot e(k) + b_1 \cdot e(k-1) + b_2 \cdot e(k-2) + \dots + b_m \cdot e(k-m) \end{aligned} \quad (2)$$

On peut appliquer la transformée en  $z$  à chaque membre de l'équation puis en appliquant en récurrence la propriété 5 (théorème du retard) on obtient la fonction de transfert discrète du système :

$$\frac{s(z)}{e(z)} = F(z) = \frac{b_0 + b_1 \cdot z^{-1} + b_2 \cdot z^{-2} + \dots + b_m \cdot z^{-m}}{1 + a_1 \cdot z^{-1} + a_2 \cdot z^{-2} + \dots + a_n \cdot z^{-n}} \quad (3)$$

Remarque 1 : la fonction de transfert précédente est normalisée en prenant  $a_0 = 1$ .

Remarque 2 : on remarquera que la fonction de transfert est donnée en  $z^{-1}$  plutôt qu'en  $z$ . Cela est justifié par le fait que celle-ci traduit l'équation d'évolution du système physique causal (2).

Le script Matlab permettant de créer un objet Matlab correspondant à une fonction de transfert discrète est donné ci-dessous.

#### Matlab

```
% Création d'un système linéaire discret :
Te=0.1;
sysd = tf([b0 b1 b2],[1 a0 a2 a3],Te,'variable','z^-1');
```

On notera l'utilisation de l'argument 'variable', 'z^-1'.

### 3.2.2 Notation "opérateur de retard $q^{-1}$ "

La fonction de transfert est une fonction de la variable complexe  $z$ . Il faudrait donc en toute rigueur l'associer à son rayon de convergence. En pratique on ne s'embarrasse jamais de préciser cette zone de convergence. C'est pourquoi on trouve souvent la notation utilisant l'opérateur  $q^{-1}$  de décalage à la place de la variable complexe  $z^{-1}$  (4).  $F(q)$  n'est alors plus à proprement parler une fonction mais on lui appliquera les mêmes manipulations algébriques qu'à  $F(z)$ .

$$F(q) = \frac{b_0 + b_1 \cdot q^{-1} + b_2 \cdot q^{-2} + \dots + b_m \cdot q^{-m}}{1 + a_1 \cdot q^{-1} + a_2 \cdot q^{-2} + \dots + a_n \cdot q^{-n}} \quad (4)$$

Attention : selon les auteurs l'opérateur qui remplace  $z^{-1}$  est noté  $q$  (Matlab) ou  $q^{-1}$  (Landau [1]). Dans la suite du document c'est cette dernière convention qui sera appliquée.

L'utilisation de l'opérateur de retard  $q^{-1}$  à la place de la variable complexe  $z^{-1}$  permet de traiter du cas des fonctions de transfert à coefficients non constant, cas non prévu par la transformation en  $z$ .

### 3.2.3 Réponse impulsionnelle

Essayons d'appliquer au système un signal d'entrée particulier : une impulsion unitaire. La transformée en  $z$  du système est obtenue en multipliant la fonction de transfert par la transformée en  $z$  de l'impulsion unitaire : 1. On obtient :

$$s(z) = \frac{b_0 + b_1 \cdot z^{-1} + b_2 \cdot z^{-2} + \dots + b_m \cdot z^{-m}}{1 + a_1 \cdot z^{-1} + a_2 \cdot z^{-2} + \dots + a_n \cdot z^{-n}} \times 1 \quad (5)$$

Cela justifie le fait que la fonction de transfert du système peut aussi s'appeler "réponse impulsionnelle".

#### Matlab

```
% Réponse impulsionnelle d'un système :
[y,t] =impulse(sysd);
```

### 3.2.4 Causalité

Un système est dit causal si les sorties ne dépendent que du passé et non pas du futur. La conséquence est que la réponse impulsionnelle de  $F(z)$  ne peut pas comporter de terme non nuls pour  $k < 0$ . Cela justifie la confusion sémantique entre "système causal" et "signal causal", c'est-à-dire ne comportant pas de termes non nuls pour  $k < 0$ .

Attention : à la différence des systèmes continus un système physique (causal) discret peut très bien être modélisé avec un degré  $m$  supérieur au degré  $n$  : le système peut avoir plus de zéros que de pôles.

La transformée en  $z$  représentant des signaux échantillonnés elle peut tout aussi bien représenter les mesures prises à différents endroits d'une poutre. Le temps étant remplacé ici par une abscisse la notion de causalité ne s'applique pas ! Attention : beaucoup des propriétés de systèmes échantillonnés que l'on étudie sont liés à la causalité, prudence donc si l'on applique ces méthodes à d'autres domaines !

### 3.2.5 Réponses indicielle

Pour caractériser des systèmes on s'intéresse souvent à leur réponse impulsionnelle ou, plus souvent, à leur réponse indicielle. Pour calculer la réponse

à un échelon d'un système il faut donc multiplier la fonction de transfert par la transformée en z d'un échelon unitaire :

$$s(z) = \frac{b_0 + b_1 \cdot z^{-1} + b_2 \cdot z^{-2} + \dots + b_m \cdot z^{-m}}{1 + a_1 \cdot z^{-1} + a_2 \cdot z^{-2} + \dots + a_n \cdot z^{-n}} \times \frac{1}{1 - z^{-1}} \quad (6)$$

Il faudrait ensuite calculer la transformée en z inverse de s(z). En pratique, si l'on doit faire le calcul à la main on décompose en éléments simples puis on utilise des tables. Avec Matlab c'est plus simple. Voici par exemple le script pour obtenir les 20 premiers échantillons de la réponse indicielle d'un système :

**Matlab**

```
% Réponse indicielle d'un système :
[y,t] =step(sysd,20);
```

### 3.2.6 Régime stationnaire (gain statique)

Le gain statique est obtenu en appliquant une entrée indicielle et en calculant la valeur du régime permanent. En appliquant le théorème de la valeur finale il vient :

$$F_0 = \lim_{z \rightarrow 1} \left( \frac{z-1}{z} \cdot \frac{b_0 + b_1 \cdot z^{-1} + b_2 \cdot z^{-2} + \dots + b_m \cdot z^{-m}}{1 + a_1 \cdot z^{-1} + a_2 \cdot z^{-2} + \dots + a_n \cdot z^{-n}} \times \frac{1}{1 - z^{-1}} \right) \quad (7)$$

Soit :

$$F_0 = \frac{\sum_{i=0}^{i=m} b_i}{1 + \sum_{i=1}^{i=n} a_i} \quad (8)$$

Le gain statique est obtenu donc simplement en faisant le quotient de la somme des coefficients.

### 3.2.7 Pôles, zéros, stabilité

Comme pour les systèmes continus la dynamique et la stabilité des systèmes dépend des propriétés du dénominateur de la fonction de transfert.



$$F(z) = \frac{e_0 \cdot (1 + e_1 \cdot z^{-1}) \cdot (1 + e_2 \cdot z^{-1}) \cdots (1 + e_m \cdot z^{-1})}{(1 + c_1 \cdot z^{-1}) \cdot (1 + c_2 \cdot z^{-1}) \cdots (1 + c_n \cdot z^{-1})} \quad (9)$$

Les coefficients  $c_i$  sont les pôles de la fonction de transfert, les coefficients  $e_i$  sont les zéros. La fonction de transfert étant au départ composée de coefficients purement réels les coefficients  $c_i$  sont soit réels soit imaginaires par paires conjuguées.

Une fonction de transfert est donc aussi bien définie par la donnée de ses pôles et de ses zéros. Le script Matlab pour créer un système ainsi est donné ci-dessous :

#### Matlab

```
% Système discret défini par ses pôles et zéros :
Te=0.1;
sysd=zpk([c1 c2 c3],[e1 e2],e0,Te);
```

Si l'on décompose la fonction de transfert en éléments simples on obtient une somme d'éléments du type :

$$F(z) = \frac{d_1}{1 + c_1 \cdot z^{-1}} + \frac{d_2}{1 + c_2 \cdot z^{-1}} + \cdots + \frac{d_n}{1 + c_n \cdot z^{-1}} \quad (10)$$

Un système linéaire temps invariant est dit stable si sa réponse impulsionnelle tend asymptotiquement vers 0. Compte tenu de (10) cela nécessite que chaque pôle  $c_i$  soit de module inférieur à 1.

**Propriété** : un système discret LTI est stable si et seulement si tous ses pôles ont leur modules inférieur ou égal à 1.

#### Critère algébrique attestant de la stabilité : critère de Routh

On rappelle que le critère de Routh permet de prédire si les racines d'un polynôme sont toutes à partie réelle négative, donc stables. Énonçons le théorème de Routh dans le cas d'un polynôme  $P'(w)$  :

$$P'(w) = a_0 + a_1 \cdot w + a_2 \cdot w^2 + \cdots + a_n \cdot w^n \quad (11)$$

Dans un premier temps on construit le tableau :

$$\left| \begin{array}{cccccc} \beta_{n,0} & \beta_{n,1} & \beta_{n,2} & \beta_{n,q-1} & \beta_{n,q} & 0 \\ \beta_{n-1,0} & \beta_{n-1,1} & \beta_{n-1,2} & \beta_{n-1,q-1} & \beta_{n-1,q-1} & 0 \\ \beta_{n-2,0} & \beta_{n-2,1} & \beta_{n-2,2} & \beta_{n-2,q-1} & \beta_{n-2,q-1} & \\ \\ \beta_{2,0} & \beta_{2,1} & 0 & & & \\ \beta_{1,0} & 0 & & & & \\ \beta_{0,0} & 0 & & & & \end{array} \right| \quad (12)$$

Puis on vérifie que tous les premiers éléments de la première colonne ont le même signe : cela garantit que tous les pôles de  $P'(w)$  ont leur partie réelle négative.

Les éléments du tableau sont construits par récurrence ligne par ligne en appliquant les formules suivantes :

Pour  $i = n$  et  $i = n - 1$  :

$$\begin{cases} \beta_{n,j} = a_{n-2j} \\ \beta_{n-1,j} = a_{n-1-2j} \end{cases} \quad (13)$$

Et pour  $i < n-1$  :

$$\beta_{i,j} = \frac{\beta_{i+1,0} \cdot \beta_{i+2,j+1} - \beta_{i+1,j+1} \cdot \beta_{i+2,0}}{\beta_{i+1,0}} \quad (14)$$

Dans le cas qui nous intéresse, un système échantillonné, on cherche plutôt à vérifier que les pôles du polynôme  $P(z)$  sont de module inférieur à un. L'idée est d'effectuer un changement de variable  $z \rightarrow w$  pour pouvoir appliquer le critère de Routh.

$$P'(w) = (1-w)^n \cdot P\left(\frac{1+w}{1-w}\right) \quad (15)$$

Si l'on effectue la transformation (15) alors le fait que  $P'(w)$  ait toutes ses racines à partie réelle négative garantit que  $P(z)$  a toutes ses racines dans le cercle unité. Pour vérifier la stabilité de  $P(z)$  il suffit donc d'effectuer le changement de variable (15) puis d'appliquer le critère de Routh conventionnel (12 13 14).

### Critère algébrique attestant de la stabilité : critère de Jury

Le critère de Jury s'apparente au critère de Routh. Nous le donnons pour un polynôme de degré 2, 3 et 4 du type :

$$P(z) = a_0 + a_1 \cdot z + a_2 \cdot z^2 + \dots + a_n \cdot z^n \quad (16)$$

Les conditions de stabilité sont données ci-dessous (ces conditions sont données dans le cas  $a_n > 0$ , il suffit de multiplier tous les termes par  $-1$  si cette condition n'est pas remplie par  $P(z)$ ) :

$$\begin{aligned}
 n = 2 & \begin{cases} a_0 + a_1 + a_2 > 0 \\ a_0 - a_1 + a_2 > 0 \\ -a_0 + a_2 > 0 \end{cases} \\
 n = 3 & \begin{cases} a_0 + a_1 + a_2 + a_3 > 0 \\ -a_0 + a_1 - a_2 + a_3 > 0 \\ -|a_0| + a_3 > 0 \\ a_0 a_2 - a_1 a_3 - a_0^2 + a_3^2 > 0 \end{cases} \\
 n = 4 & \begin{cases} a_0 + a_1 + a_2 + a_3 + a_4 > 0 \\ a_0 - a_1 + a_2 - a_3 + a_4 > 0 \\ a_4^2 - a_0^2 - |a_0 a_3 - a_1 a_4| > 0 \\ (a_0 - a_4)^2 (a_0 - a_2 + a_4) + (a_1 - a_3)(a_0 a_3 - a_1 a_4) > 0 \end{cases}
 \end{aligned} \quad (17)$$

### 3.2.8 Systèmes du premier ordre

On donne dans la Figure 4 les réponses indicielles de systèmes du premier ordre ainsi que la localisation des pôles dans le plan complexe. Le cercle de rayon unité représente la zone dans laquelle les pôles doivent se situer pour assurer la stabilité. On remarque sur les deux premières figures l'influence de la place du zéro par rapport au point  $(1,0)$  pour la rapidité.

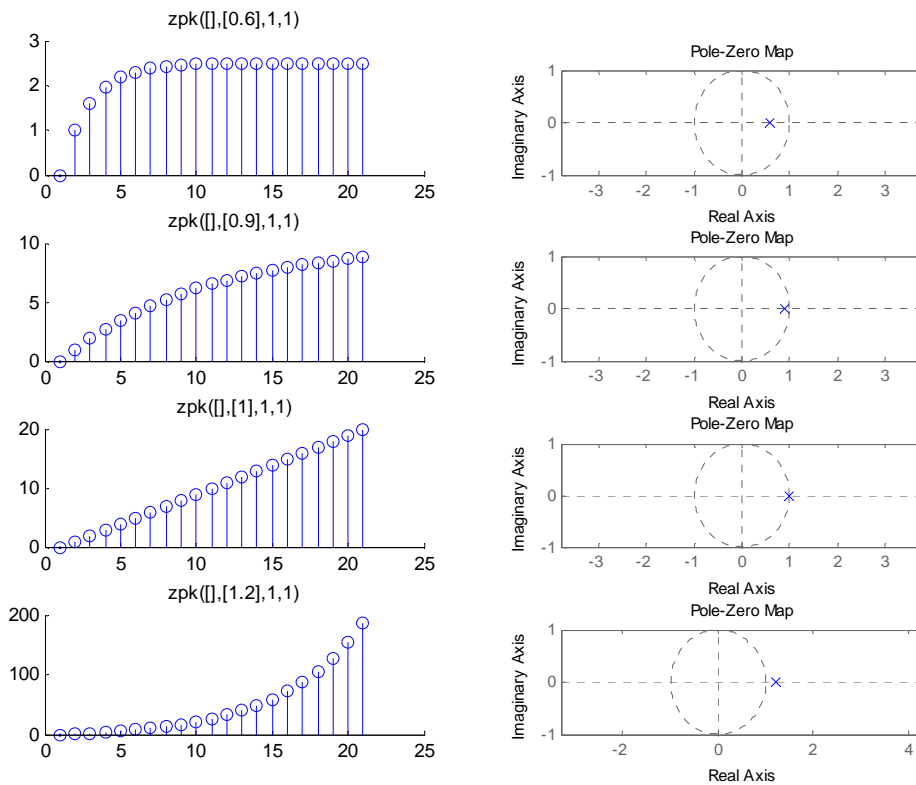


Figure 4 réponses indicielles de systèmes du premier ordre

### 3.2.9 Systèmes du second ordre

On donne dans la Figure 6 les réponses indicielles de systèmes du second ordre ainsi que la localisation des pôles dans le plan complexe. La rapidité et l'amortissement dépend de la position des pôles dans le cercle unité. La Figure 5 représente les courbes iso-amortissement et iso-pulsation des systèmes du second ordre.

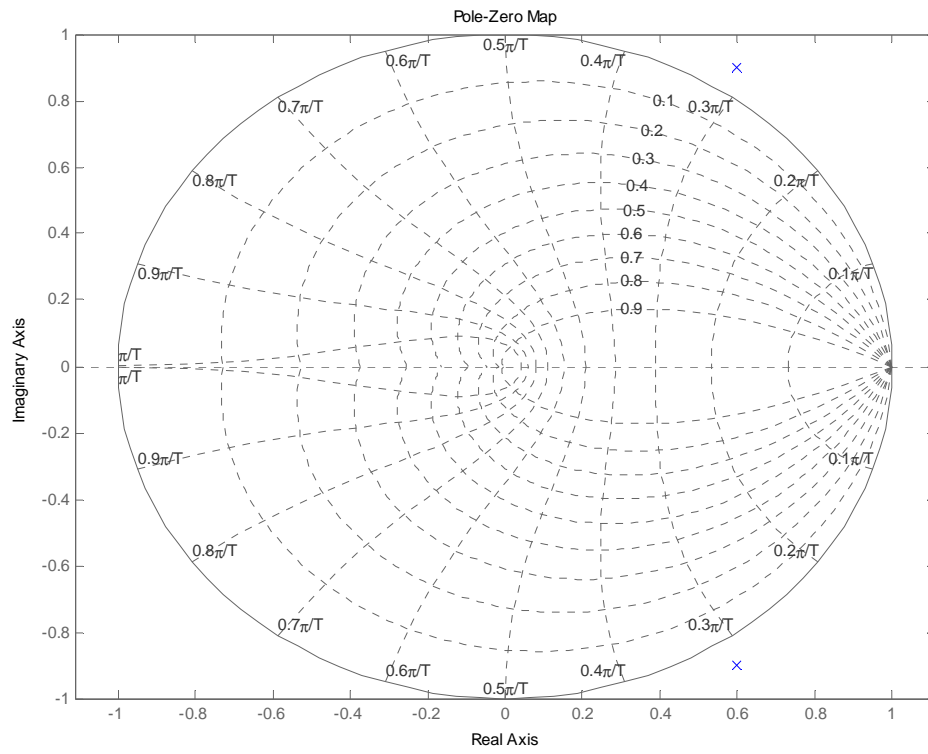


Figure 5 : courbes iso-amortissement et iso-pulsation des systèmes discrets du second ordre

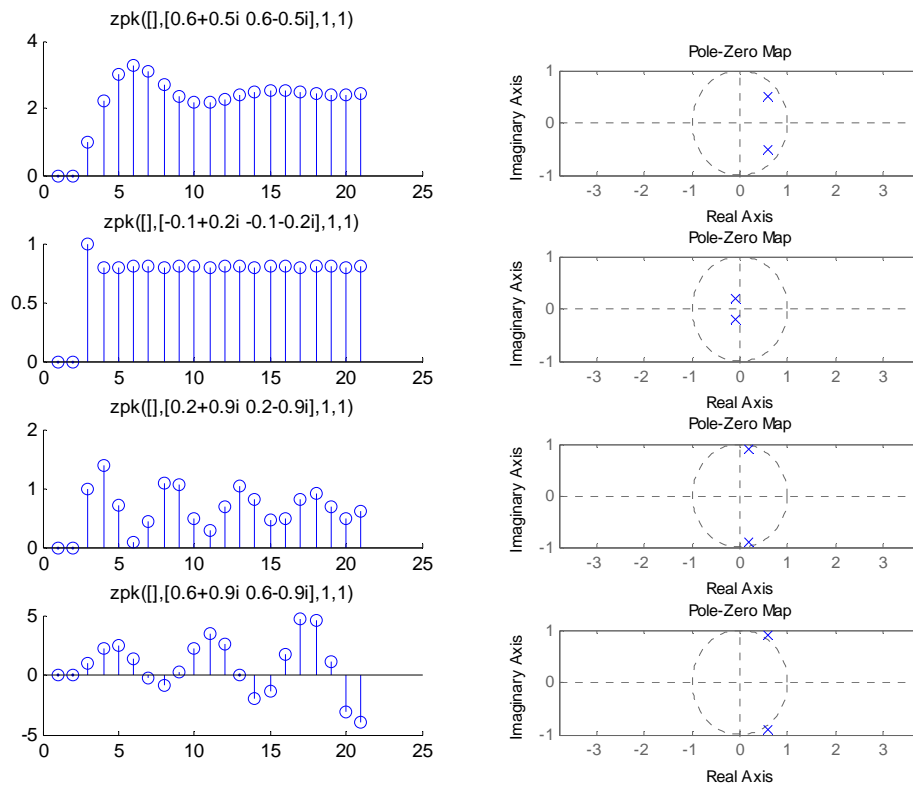


Figure 6 : réponses de systèmes du second ordre

### 3.2.10 Calcul itératif de la réponse d'un système

Si l'on dispose de la fonction de transfert d'un système et que l'on connaît les valeurs successives  $e(k)$  du signal d'entrée il est facile de calculer itérativement la sortie  $y(k)$  du système par l'équation de récurrence (2). Il faut initialiser par les  $n$  valeurs initiales de  $s(k)$ , prises généralement égales à 0.

### 3.2.11 Représentation par équation d'état

Un système discret peut aussi être représenté par un ensemble d'équations d'état comme dans le domaine continu (52). Cette représentation est naturellement adaptée aux systèmes multi-entrées multi-sorties. On définit alors un état interne  $x(k)$  ainsi qu'une fonction d'évolution  $f$  et une équation de mesure  $h$ .

$$\begin{cases} x(k+1) = f(x(k), u(k)) \\ s(k) = h(x(k), u(k)) \end{cases} \quad (18)$$

### 3.3 Réponse fréquentielle des systèmes échantillonnés

#### 3.3.1 Diagramme de Bode d'un système échantillonné

Lorsqu'on étudie la réponse fréquentielle des systèmes continus on considère des entrées particulières sinusoïdales :

$$\begin{cases} e(t) = \cos(2 \cdot \pi \cdot f \cdot t) + j \cdot \sin(2 \cdot \pi \cdot f \cdot t) = e^{j \cdot 2 \cdot \pi \cdot f \cdot t} \\ f \in [0 \quad \infty] \end{cases} \quad (19)$$

De même, pour des systèmes échantillonnés on considèrera des entrées sinusoïdales échantillonnées :

$$\begin{cases} e(k) = \cos(2 \cdot \pi \cdot f \cdot k) + j \cdot \sin(2 \cdot \pi \cdot f \cdot k) = e^{j \cdot 2 \cdot \pi \cdot f \cdot k} \\ f \in [0 \quad 1] \end{cases} \quad (20)$$

La sortie est aussi un signal échantillonné sinusoïdal de gain  $A(f)$  et de phase  $\Phi(f)$  variables et dépendant de  $f$ ; la fréquence réduite.

$$\begin{cases} s(k) = A(f) \cdot e^{j \cdot (2 \cdot \pi \cdot f \cdot k + \Phi(f))} \\ f \in [0 \quad 1] \end{cases} \quad (21)$$

Le nombre complexe, noté abusivement  $F(f)$ , d'amplitude  $A(f)$  et de phase  $\Phi(f)$  est obtenu à partir de  $F(z)$  en remplaçant  $z$  par  $e^{j \cdot 2 \cdot \pi \cdot f}$  :

$$\begin{cases} F(f) = F(z = e^{j \cdot 2 \cdot \pi \cdot f}) \\ f \in [0 \quad 1] \end{cases} \quad (22)$$

Le diagramme représentant le gain et la phase en fonction de la fréquence réduite s'appelle aussi un diagramme de Bode. L'échelle des fréquences est généralement linéaire de 0 à 0.5 (puisque le diagramme est nécessairement symétrique). La Figure 7 représente par exemple le diagramme de Bode d'un système du second ordre.

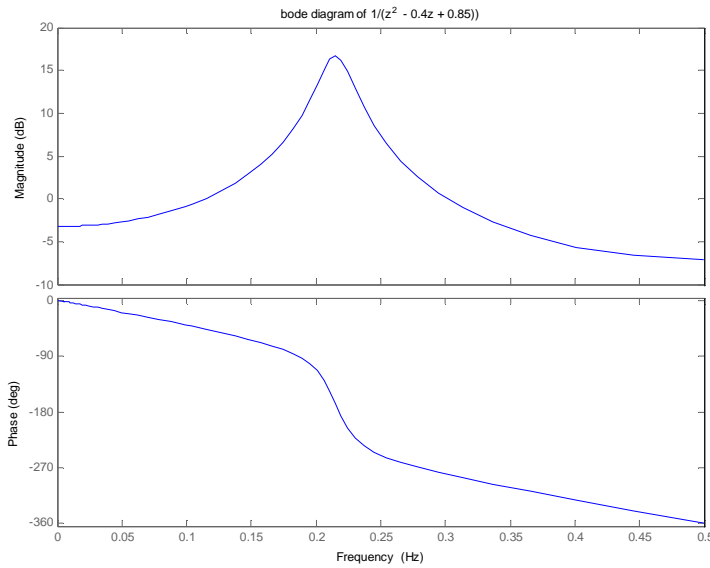


Figure 7 : exemple du diagramme de Bode d'un système discret du second ordre

### 3.3.2 Où l'on voit apparaître le théorème de Shannon

Le changement de variable  $z = e^{j2\pi f}$  n'est pas bijectif puisque  $e^{j2\pi f} = e^{j2\pi(f+k)}$  avec  $k$  entier. Si l'on fait le lien entre domaine continu et domaine fréquentiel on peut remarquer que le changement de variable  $z = e^{j2\pi f}$  fait correspondre une zone du plan complexe continu à une zone du plan complexe discret. La transformation est bijective si l'on se contente de la zone du plan complexe continu contenu entre les droites d'ordonnée 0 et 1.



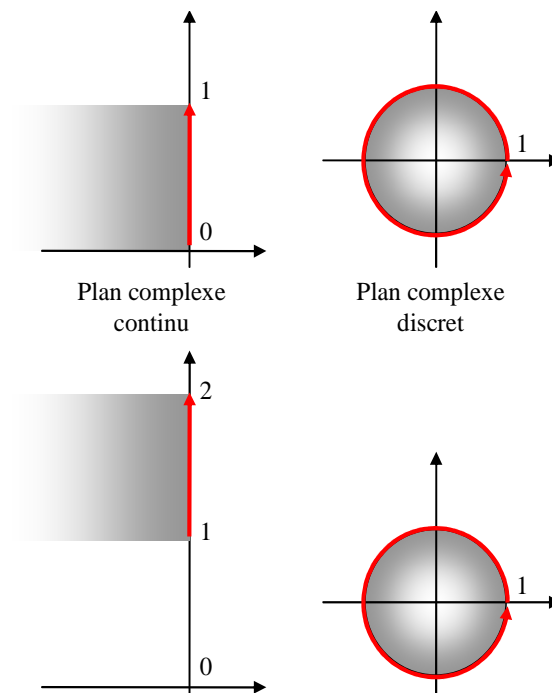


Figure 8 : correspondance entre plan complexe continu et plan complexe discret.



## 4. SYSTEMES CONTINUS ECHANTILLONNES

### 4.1 Structure des systèmes commandés par ordinateur

Les systèmes commandés par ordinateurs sont presque toujours des systèmes physiques continus par nature. Le ordinateur traite par contre les entrées et les sorties de manière discrète avec une période d'échantillonnage  $T_e$ . Les éléments importants, du point de vue de l'automaticien sont donc :

- le convertisseur analogique / numérique CAN ou du moins le système qui fournit séquentiellement les mesures au ordinateur à la période  $T_e$ .
- le ordinateur lui-même. Il doit effectuer l'acquisition des mesures, le traitement des informations, l'envoi des consignes dans un temps inférieur à  $T_e$ .
- le convertisseur numérique / analogique CNA.
- le système physique.

On suppose, pour faire simple et pour pouvoir utiliser la panoplie des outils mathématiques, que les systèmes sont échantillonnés à la même période  $T_e$  et au même moment.

La Figure 9 montre le système physique commandé par le ordinateur, dans sa version simpliste.

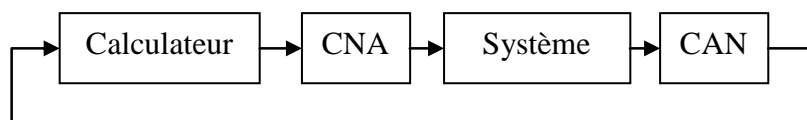


Figure 9 : système commandé par ordinateur

Au XXIème siècle les choses sont quand même parfois un petit peu plus compliquées :

- le ordinateur fournit bien souvent un signal de commande numérique (transporté par bus informatique USB, RS232, CAN, Ethernet, etc...) directement interprété par l'actionneur. On parle "d'actionneur intelligent".
- de même le "capteur intelligent" fournit aussi bien souvent la mesure sous forme d'un signal numérique.
- Les signaux sont transportés par des bus informatiques qui n'ont pas que ça à faire : il peut y avoir du retard.
- de même les capteurs et actionneurs sont rarement tous seuls, ce qui se traduit par des périodes d'échantillonnage différentes, et parfois non synchrones...

- enfin le ordinateur peut aussi avoir d'autres choses à faire...

Ceci étant dit nous allons pour le moment nous restreindre au cas représenté sur la Figure 9 avec des instants d'échantillonnage synchrones et régulièrement espacés d'un temps  $T_e$ .

## 4.2 Echantillonnage des systèmes continus

On dispose généralement de la fonction de transfert continue (notée en utilisant la variable complexe  $s$  ou  $p$ )  $F(s)$  du système continu. Le problème est d'obtenir la fonction de transfert discrète du système échantillonné de manière à pouvoir synthétiser un correcteur. Le système continu est associé au convertisseur numérique – analogique qu'il faut prendre en compte.

Les convertisseurs numérique analogiques peuvent généralement se représenter par des "échantillonneurs – bloqueurs d'ordre 0" ou BOZ. Cela signifie que la commande est maintenue à une valeur constante pendant toute la durée du temps d'échantillonnage.

Si l'on applique une commande discrète impulsionnelle à un BOZ il va maintenir la commande 1 pendant la durée  $T_e$  puis revenir à zéro (Figure 10).

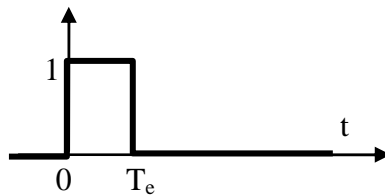


Figure 10 : réponse impulsionnelle d'un bloqueur d'ordre 0

La réponse impulsionnelle  $BOZ(t)$  est donc composée d'un échelon unitaire suivi d'un échelon unitaire de signe opposé et décalé d'un temps  $T_e$ . La transformée de Laplace de la réponse impulsionnelle du bloqueur est donnée par la formule (23) :

$$BOZ(s) = \frac{1}{s} - \frac{e^{-jT_e}}{s} \quad (23)$$

Du point de vue du ordinateur il faut donc prendre en compte le système continu de fonction de transfert  $F_{BOZ}(s)$  que l'on devra échantillonner :

$$F_{BOZ}(s) = \frac{1 - e^{-jT_e}}{s} \cdot F(s) \quad (24)$$

$$F_{\text{BOZ}}(z) = Z^{-1} \left( \frac{1 - e^{-jT_e}}{s} \cdot F(s) \right) \quad (25)$$

En pratique l'obtention de la transformée en z d'un système échantillonné par bloqueur d'ordre 0 se fait à l'aide de tables. On donne quelques exemples fondamentaux dans le Tableau 3 ci-dessous.

Tableau 3 : transformée en z de systèmes continus échantillonnés

F(s)	F <sub>BOZ</sub> (z)
1	1
$\frac{1}{s}$	$\frac{T_e \cdot z^{-1}}{1 - z^{-1}}$
$\frac{1}{1 + T \cdot s}$	$\begin{cases} \frac{b_1 \cdot z^{-1}}{1 + a_1 \cdot z^{-1}} \\ b_1 = 1 - e^{-T_e/T} \quad a_1 = -e^{-T_e/T} \end{cases}$
$\frac{e^{-Ls}}{1 + T \cdot s}$	$\begin{cases} \frac{b_1 \cdot z^{-1} + b_2 \cdot z^{-2}}{1 + a_1 \cdot z^{-1}} \quad a_1 = -e^{-T_e/T} \\ b_1 = 1 - e^{-(L-T_e)/T} \quad b_2 = e^{-T_e/T} (e^{L/T} - 1) \end{cases}$
$L < T_e$	

Si l'on dispose de Matlab c'est bien sûr encore plus simple. On utilise la fonction c2d avec l'argument 'zoh' pour "zero holder hold" (bloqueur d'ordre zéro).

#### Matlab

```
% Définition d'un système continu :
sysc=tf(1,[1 1]);
% Obtention de la transformée en z du système échantillonné à Te
Te=0.1;
sysd=c2d(sysc,Te,'zoh');
% Notation en z^-1
sysd.variable='z'
```

### 4.3 Effets de l'échantillonnage

Le convertisseur numérique a deux effets (néfastes) importants : l'effet de retard et l'effet de quantification.

L'effet de retard se voit dans (24) : on peut dire en première approximation que l'échantillonnage se traduit par un retard pur de  $T_e/2$ . Cela a donc, si ce temps

d'échantillonnage est non négligeable par rapport aux constantes de temps du système, un effet déstabilisant.

Pour illustrer l'effet de quantification supposons par exemple que l'on utilise un convertisseur analogique/numérique 8 bits 0-10V. Le plus petit incrément que le convertisseur peut distinguer est de  $10V/2^8 = 40mV$ . Si le signal utile varie physiquement entre 5V et 5,4V on n'aura que dix incréments utilisables... Ce phénomène n'entre pas dans la panoplie des phénomènes modélisables par la transformée en z. En pratique il est donc souvent nécessaire d'étudier le système et de synthétiser la commande numérique en négligeant ce phénomène puis de vérifier a posteriori qu'il n'a pas un effet trop néfaste.

#### 4.4 Où l'on revoit le théorème de Shannon

Le théorème de Shannon est bien connu en ce qui concerne l'échantillonnage de signaux discrets : on sait que le signal ne doit pas comporter de composante de fréquence supérieure à  $f_e/2$  si l'on ne veut pas observer de repliement. On doit aussi prendre des précautions si l'on veut travailler sur la fonction de transfert discrète correspondant à un système continu échantillonné.

On va traiter un exemple concret pour mettre en évidence le problème de repliement dans le cas de la commande des systèmes continus. On s'intéresse à deux systèmes continus du second ordre ayant des pôles de même partie réelle mais de parties imaginaires séparées de  $2\pi T_e$ . Le script Matlab ci-dessous montre la création des deux systèmes :

##### Matlab

```
% Définition de deux systèmes continus :  
sys1=zpk([],[-1-2i -1+2i],2^2+1);  
sys2=zpk([],[-1-(2+2*pi)*i -1+(2+2*pi)*i],1+(2+2*pi)^2);
```

On voit sur la Figure 11 que la réponse indicielle des deux systèmes est bien différente.

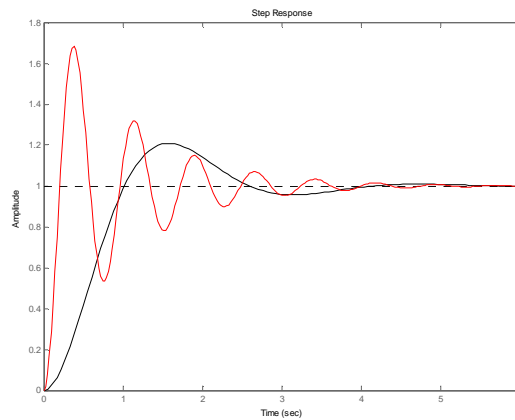


Figure 11 : réponse indicielle de deux systèmes du second ordre

Si l'on calcule la fonction de transfert échantillonnée les deux systèmes à la période de 1 seconde on obtient deux réponses indicielles très proches (représentés par des  $\circ$  sur la Figure 12). Conclusion : la période d'échantillonnage était mal choisie

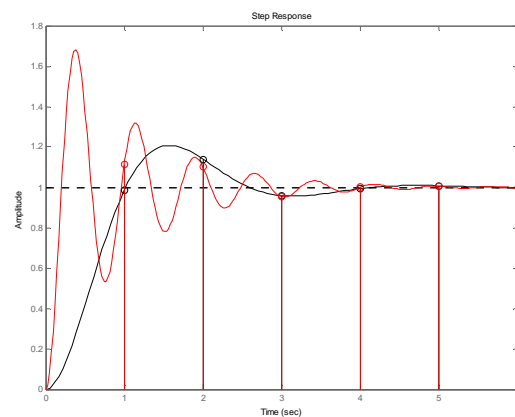


Figure 12 : réponse indicielle de deux systèmes du second ordre échantillonnés à 1s

En pratique la commande échantillonnée de systèmes continus doit être associée à certaines précautions :

- le système ne doit pas comporter de dynamiques propres situées dans les fréquences supérieures à  $f_c/2$  (exemple illustré dans ce chapitre).
- les signaux échantillonnés doivent être "nettoyés" des fréquences supérieures à  $f_c/2$  avec un filtre anti-repliement. (A moins que le signal soit suffisamment propre et ne comporte pas de composantes significatives au-delà de ces fréquences).

### Choix de la période d'échantillonnage

Le choix de la période d'échantillonnage est très important et dépend de la dynamique des signaux. Attention, les systèmes commandés en boucle fermée ont souvent une dynamique plus rapide que le système en boucle ouverte : il faut donc prendre en compte la dynamique en boucle fermée pour le choix de la fréquence d'échantillonnage.

En pratique on préconise de prendre une fréquence 5 à 25 fois supérieure à la bande passante souhaitée en boucle fermée.

On peut être tenté de prendre une fréquence d'échantillonnage très élevée si la technologie le permet (ordinateur et convertisseurs suffisamment rapide). Ceci peut amener rapidement à des problèmes numériques. Imaginons par exemple le cas d'un système ayant une dynamique très lente (quelques secondes) et un système de mesure codant les mesures sur 10 bits (cas classique). La Figure 13 illustre cet exemple. Le signal varie peu entre deux instants d'échantillonnage et les variations du signal sont faibles par rapport à la valeur de l'incrément numérique (figure du haut). Le calcul de la simple dérivée numérique se traduit donc par une succession de  $-1$   $0$  et  $+1$ .

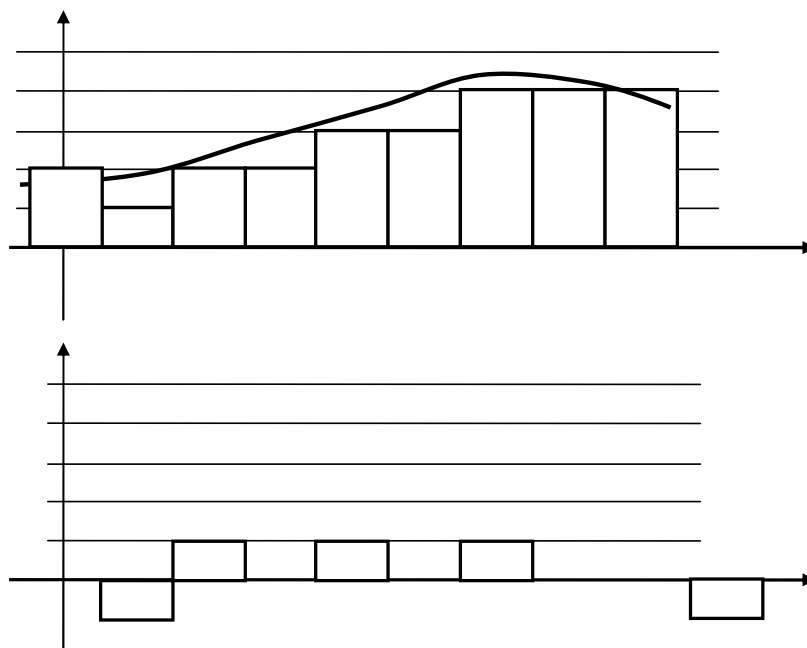


Figure 13 : période d'échantillonnage trop faible. En haut : signal échantillonné et discrétisé ; en bas : calcul de la dérivée numérique.



### Filtre anti-repliement

En toute rigueur chaque fois qu'un signal est échantillonné il faudrait faire précéder le convertisseur numérique d'un filtre analogique anti-repliement éliminant physiquement toutes les composantes du signal de fréquence supérieure à  $f_e/2$ .

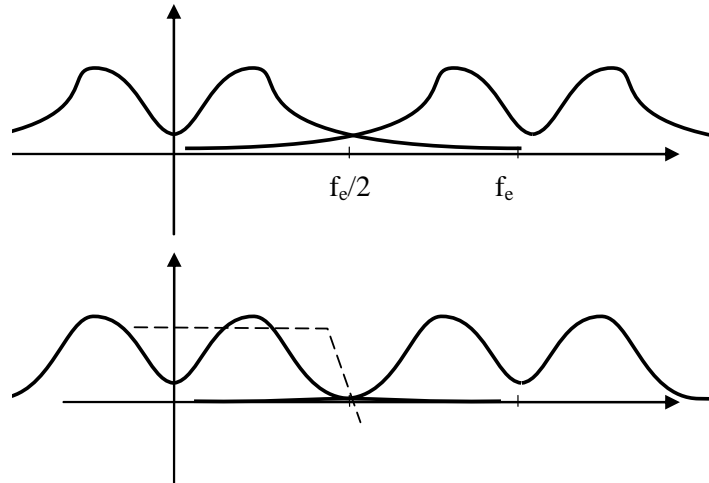


Figure 14 : Effet du filtre anti-repliement

Bien souvent la période d'échantillonnage de l'asservissement est relativement lente (quelques secondes) et il est alors impossible de réaliser un filtre de constante de temps aussi faible. On procède alors à un sur-échantillonnage : le signal est filtré, échantillonné à haute fréquence  $T'_e$  puis filtré numériquement et sous-échantillonné à la période  $T_e$  beaucoup plus grande que  $T'_e$  comme illustré sur la

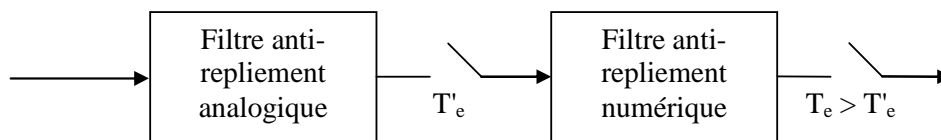


Figure 15 : Sur échantillonnage

Un simple filtre numérique de type moyenne mobile est souvent suffisant.

### 4.5 Discrétisation de fonction de transfert continues

Une manière classique pour réaliser un contrôle par ordinateur est de commencer par synthétiser un correcteur continu avec les outils de l'automatique classique puis de trouver un correcteur discret qui "approche" convenablement le correcteur continu.

#### 4.5.1 Approximation numérique d'équation différentielle

Supposons que le correcteur continu que l'on a synthétisé soit du type PD :

$$\text{PID}(s) = \frac{e(s)}{\varepsilon(s)} = k_p + k_D \cdot s \quad (26)$$

Il s'agit de calculer les commandes  $e(k)$  successives issues du correcteur en fonction des erreurs  $\varepsilon(k)$ . Une approche classique et intuitive est d'approximer la dérivée par une différence :

$$e(k) = k_p \cdot \varepsilon(k) + k_D \cdot \frac{\varepsilon(k) - \varepsilon(k-1)}{T_e} \quad (27)$$

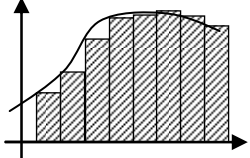
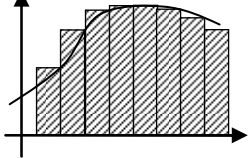
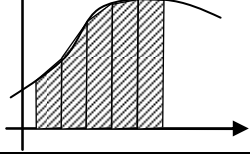
Si on calcule la transformée en  $z$  de  $e(k)$  on obtient :

$$e(z) = k_p \cdot \varepsilon(z) + k_D \cdot \frac{\varepsilon(z) - z^{-1}\varepsilon(z)}{T_e} = \left( k_p + k_D \cdot \frac{1 - z^{-1}}{T_e} \right) \cdot \varepsilon(z) \quad (28)$$

On retrouve la fonction de transfert PD(s) dans laquelle on a effectué le changement de variable  $s = \frac{1-z^{-1}}{T_e}$

De manière plus générale, l'approximation de fonction de transfert revient à effectuer l'approximation numérique d'intégration. Selon la méthode d'intégration choisie on effectue des changements de variable différents.

Tableau 4 : méthodes d'intégration

Méthode	Changement de variable	
Euler progressif (Euler's forward method)	$s = \frac{1 - z^{-1}}{z^{-1} \cdot T_e}$	
Euler rétrograde (Euler's backward method)	$s = \frac{1 - z^{-1}}{T_e}$	
Méthode des trapèzes, ou bilinéaire, ou de Tustin	$s = \frac{2}{T_e} \cdot \frac{1 - z^{-1}}{1 + z^{-1}}$	

En pratique le changement de variable  $s = \frac{1 - z^{-1}}{z^{-1} \cdot T_e}$  n'est pas sans risque. On donne sur la Figure 16 la correspondance entre la zone stable du plan complexe en  $s$  et la zone correspondante après la transformation (Euler progressif, Euler rétrograde, bilinéaire). On voit notamment qu'il est possible, partant d'un système continu stable, d'obtenir des pôles dans le domaine discret en dehors du cercle de stabilité si la transformation appliquée est celle d'Euler progressif.

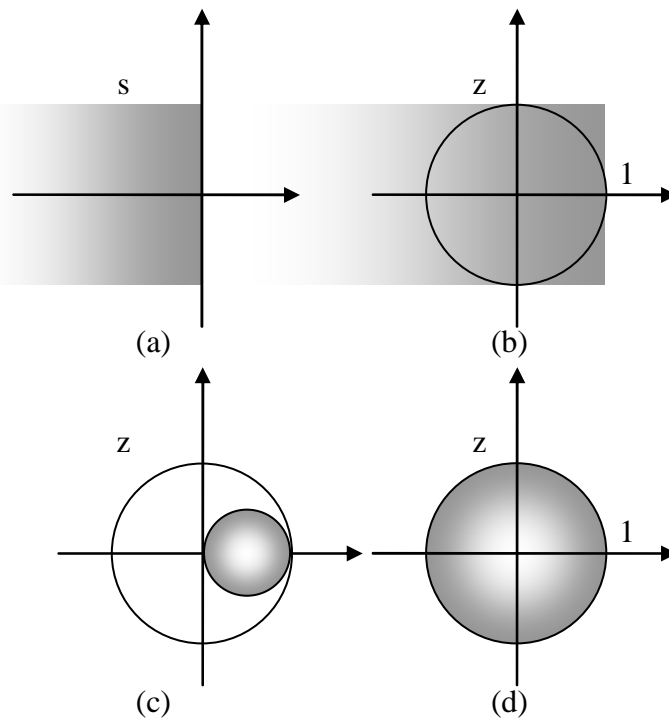


Figure 16 : correspondance des plans complexes. (a) : zone de stabilité dans le plan complexe continu en  $z$  ; (b) zone correspondante avec l'approximation d'Euler progressif ; (c) zone correspondante avec l'approximation d'Euler rétrograde ; (d) zone correspondante avec l'approximation bilinéaire.

La fonction `c2d` de Matlab permet d'obtenir l'équivalent discret d'une fonction de transfert continue avec la méthode d'intégration comme argument :

```

Matlab
sysc=tf([1 0],[1 2 5]);
Te=0.1;
% Méthode d'euler :
sysd1=c2d(sysc,Te,'zoh');
% Méthode des trapèzes :
sysd2=c2d(sysc,Te,'foh');
% Méthode bilinéaire :
sysd3=c2d(sysc,Te,'tustin');
    
```

#### 4.5.2 Superposition des réponses indicielles

Une autre méthode pour approximer une fonction de transfert discrète est de s'assurer que la réponse indicielle de sa version discrétisée correspond exactement lors des temps d'échantillonnage.

### 4.5.3 Correspondances pôles - zéros

La dernière méthode est de partir d'une décomposition de la fonction de transfert continue en pôles et zéros et d'effectuer la conversion pôles à pôles et zéros par zéros. La correspondance, selon que le pôle est réel ou complexe est donnée ci-dessous :

$$\begin{aligned} (s + a) &\rightarrow (z - e^{-a \cdot T_e}) \\ (s + a)^2 + b^2 &\rightarrow z^2 - 2 \cdot e^{-a \cdot T_e} \cdot \cos(b \cdot T_e) \cdot z + e^{-2 \cdot a \cdot T_e} \end{aligned} \quad (29)$$

### 4.5.4 Bilan

Aucune méthode n'est parfaite et on n'aura jamais en même temps :

- même réponse indicielle
- même réponse impulsionnelle
- approximation (Euler, Tustin, etc...) de l'équation différentielle
- correspondance des pôles et des zéros
- correspondance des réponses fréquentielles

Ceci étant dit le choix de la période d'échantillonnage est primordial. Si la fréquence d'échantillonnage est beaucoup plus faible que la dynamique de la fonction de transfert à approximer la différence entre chaque méthode sera minimale.



## 5. IDENTIFICATION DES SYSTEMES ECHANTILLONES

Dans le présent chapitre on donne les bases des méthodes d'identification. Pour une approche complète, voir le cours de troisième année "Identification avion" [6] et les ouvrages [1] et [7].

Même si l'on dispose d'un modèle théorique du processus que l'on souhaite commander il est presque toujours nécessaire de l'affiner en procédant à des essais sur le système réel. Bien souvent le seul modèle dont l'on dispose est ce modèle issu de l'expérimentation. Différentes méthodes s'appliquent selon l'objectif recherché et le niveau de précision exigé. On distinguera :

- les modèles paramétriques
- les modèles simples
- les modèles non paramétriques

### 5.1 Identification d'un modèle ARX par la méthode des moindres carrés récursifs

A partir de cet exemple, le plus simple qui soit, on pose les bases de toutes les méthodes d'identification.

On considère un système d'entrée  $u(t)$  et de sortie  $y(t)$  représenté par l'équation de récurrence suivante :

$$y(t) + a_1 \cdot y(t-1) + a_2 \cdot y(t-2) + \dots + a_n \cdot y(t-n) = b_0 \cdot u(t) + b_1 \cdot u(t-1) + b_2 \cdot u(t-2) + \dots + b_m \cdot u(t-m) \quad (30)$$

L'objectif est, à partir d'essais et de mesures, d'identifier les meilleures valeurs possibles pour les coefficients  $a_i$  et  $b_i$ .

L'équation (30) peut être réécrite pour mettre en évidence la *prédiction* de la sortie  $y(t)$  à partir des mesures passées :

$$y(t) = -a_1 \cdot y(t-1) - a_2 \cdot y(t-2) - \dots - a_n \cdot y(t-n) + b_0 \cdot u(t) + b_1 \cdot u(t-1) + b_2 \cdot u(t-2) + \dots + b_m \cdot u(t-m) \quad (31)$$

Soit :

$$y(t) = \varphi^T(t) \cdot \theta(t)$$

avec :

$$\begin{cases} \varphi^T(t) = [-y(t-1) \ \dots \ -y(t-n) \ , \ u(t-1) \ \dots \ u(t-m)] \\ \theta(t) = [a_1 \ \dots \ a_n \ , \ b_1 \ \dots \ b_m] \end{cases} \quad (32)$$

Le but est de trouver le meilleur ensemble de paramètres  $\theta(t)$  : celui qui, à partir d'un ensemble de mesures  $\varphi(t)$  permet de prédire avec la meilleure précision la sortie effectivement mesurée.

Supposons qu'à la suite d'une expérience on dispose d'un ensemble de mesures  $\{y(t), u(t)\}_{t=1:N}$  et que l'ensemble des paramètres  $\theta$  n'est pas bien connu. On construit la séquence des *sorties prédites* :

$$\hat{y}(t) = \varphi^T(t) \cdot \theta(t) \quad (33)$$

Le paramètre  $\theta$  est celui qui minimise une norme quadratique :

$$\theta = \arg \max \left( \frac{1}{N} \sum_{t=1}^N (y(t) - \hat{y}(t))^2 \right) = \arg \max \left( \frac{1}{N} \sum_{t=1}^N (y(t) - \varphi^T(t) \cdot \theta(t))^2 \right) \quad (34)$$

Compte tenu du fait que l'équation est linéaire au départ et que le terme à minimiser est du type quadratique la solution est simplement obtenue par une pseudo-inverse :

$$\theta = \left[ \sum_{t=1}^N (\varphi(t) \cdot \varphi^T(t)) \right]^{-1} \sum_{t=1}^N (\varphi(t) \cdot y(t)) \quad (35)$$

On obtient donc la solution classique des moindres carrés. La formule (35) est le plus souvent utilisée sous sa forme récursive :

$$\theta(t) = \theta(t-1) + F(t) \cdot \varphi(t) \cdot (y(t) - \theta^T(t-1) \cdot \varphi(t))$$

avec

$$F(t)^{-1} = F(t-1)^{-1} + \varphi(t) \cdot \varphi(t)^T \quad (36)$$

ou

$$F(t) = F(t-1) - \frac{F(t-1) \cdot \varphi(t) \cdot \varphi(t)^T \cdot F(t-1)}{1 + \varphi(t)^T \cdot F(t-1) \cdot \varphi(t)}$$

(On laisse au lecteur le plaisir d'effectuer lui-même les démonstrations, qu'il peut du reste trouver dans [1] et [7]).



## 5.2 L'identification paramétrique en général et en pratique

L'exemple précédent a permis de mettre en évidence tous les aspects de la méthodologie classique de l'identification :

- il a fallu choisir un ensemble de mesures  $\{y(t), u(t)\}_{t=1:N}$ . Le choix du signal d'excitation  $u(t)$  est en pratique primordial,
- il a fallu choisir une structure pour le modèle à identifier (ici les ordres  $n$  et  $m$ ),
- il a fallu choisir un critère (ici un critère quadratique sur l'erreur entre la sortie et la sortie prédite),
- il a fallu choisir un algorithme et le paramétrer (ici l'algorithme des moindres carrés récursifs qu'il a fallu initialiser)

Enfin un point non traité dans le chapitre précédent concerne la validation :

- il aurait fallu déterminer une méthode pour comparer différents modèles et choisir le meilleur

### 5.2.1 Choix du signal d'excitation, traitement des mesures

De manière générale le système que l'on doit identifier doit être excité dans tout le domaine fréquentiel qui nous intéresse. Le signal classiquement utilisé est une séquence pseudo binaire aléatoire (Pseudo Random Binary Signal) tel celui représenté sur la Figure 17.

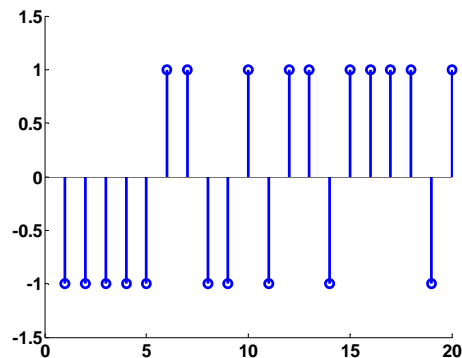


Figure 17 : exemple d'une séquence PRBS

Le signal d'excitation doit se superposer au signal de commande correspondant au point de fonctionnement étudié. Son amplitude doit être faible par rapport à l'amplitude des commandes du système (quelques %).

Une fois le système excité et les mesures effectuées il faut supprimer les valeurs moyennes des mesures.

### 5.2.2 Choix d'une structure de modèle

Le modèle étudié dans le chapitre précédent est de type ARX. On peut le noter sous forme polynomiale :

$$A \cdot y = B \cdot u \quad (37)$$

Ce modèle correspond en fait à un parti-pris sur la manière dont on traite les perturbations. En réalité on prend en compte un signal d'erreurs possibles  $e$  (de moyenne nulle) tels que :

$$A \cdot y = B \cdot u + e \quad (38)$$

Si l'on fait cette hypothèse de modélisation alors la méthode optimale est celle des moindres carrés. Malheureusement cette manière de prendre en compte les perturbations est bien restrictive. D'autres modélisations sont possibles, adaptées à d'autres algorithmes, par exemple le modèle OE :

$$y = \frac{B}{F} \cdot u + e \quad (39)$$

Il est aussi possible de "structurer" le modèle de bruit comme par exemple dans les modèles ARMAX ou BJ :

$$\begin{aligned} A \cdot y &= B \cdot u + C \cdot e \\ y &= \frac{B}{F} \cdot u + \frac{C}{D} \cdot e \end{aligned} \quad (40)$$

La structure du modèle étant choisie il faut choisir l'ordre de ses différents polynômes.

### 5.2.3 Choix d'un critère à minimiser

Dans la plupart des cas le critère à minimiser est l'erreur quadratique entre la sortie et la sortie prédite à un coup (one step ahead prediction). Ceci n'est bien évidemment pas restrictif et peut être généralisé à la prédiction à N coups.

### 5.2.4 Choix d'un algorithme

Même si chaque structure de modèle est généralement associée à un algorithme "optimal" celui-ci doit généralement se paramétrer.

Prenons par exemple l'algorithme des moindres carrés récursifs tel que présenté dans (36). Cette équation peut être modifiée en intégrant un "facteur d'oubli"  $\lambda$  :

$$\begin{aligned} \theta(t) &= \theta(t-1) + F(t) \cdot \varphi(t) \cdot (y(t) - \theta^T(t-1) \cdot \varphi(t)) \\ \text{avec} & \\ F(t)^{-1} &= \lambda \cdot F(t-1)^{-1} + \varphi(t) \cdot \varphi(t)^T \end{aligned} \quad (41)$$

Ce paramètre (typiquement entre 0.95 et 0.99) permet, comme son nom l'indique, "d'oublier" les anciennes mesures et de se concentrer sur les mesures les plus récentes.

### 5.2.5 Validation

Une fois l'algorithme d'identification appliqué et le modèle obtenu (ce qui prend généralement quelques secondes avec un ordinateur moderne) il reste à expertiser celui-ci. Selon la "qualité" du modèle obtenu on remet en cause l'une ou l'autre des étapes (signal d'excitation, structure du modèle, algorithme) et on réitère jusqu'à obtenir un modèle satisfaisant.

En pratique on essaie un certain nombre de modèles avant de choisir le "meilleur".

Le premier critère de qualité est la capacité du modèle à prédire effectivement la sortie (sortie prédite à un coup, sortie prédite à N coups). Il faut pour cela utiliser un jeu de mesures non utilisé pour l'identification. (En pratique le jeu de mesures initiales est coupé en deux : une partie pour l'identification et une partie pour la validation).

Le second critère de qualité est l'analyse du résidu : la composition spectrale de l'erreur de prédiction.

### 5.2.6 Utilisation de logiciels

En pratique il est commode d'utiliser un célèbre outil de calcul scientifique et sa toolboxe d'identification.

Comme les calculs proprement dit ne prennent que quelques secondes il est facile de réaliser un grand nombre d'identifications et de stocker les différents modèles pour les comparer ensuite. On est rapidement noyé sous le grand nombre de modèles possibles...

## 5.3 Identification "simple"

Il est souvent suffisant d'obtenir un modèle "simple" de type premier ordre et un simple essai indiciel convient souvent. Le système est excité avec un échelon unité et on cherche le meilleur modèle du premier ordre correspondant.

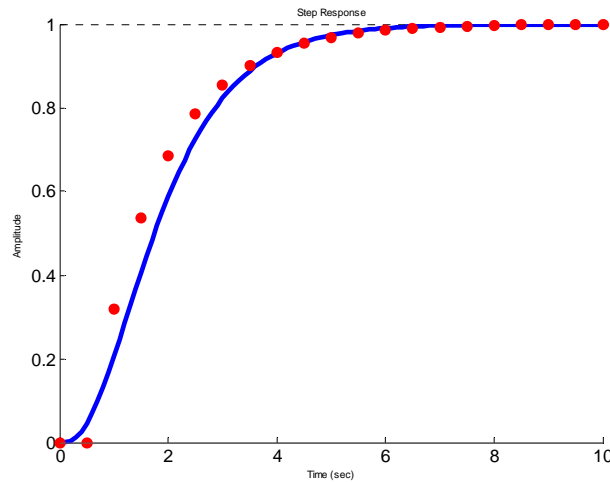


Figure 18 : identification sur un essai indiciel

Dans l'exemple de la Figure 18 on superpose le résultat d'une mesure (en bleu) avec la simulation d'un modèle discret avec une période de 0.5s. Le modèle cherché est du premier ordre avec un retard pur  $d$  :

$$y(z) = \frac{(1 + c_1) \cdot z^{-d}}{1 + c_1 \cdot z^{-1}} \cdot u(z) \quad (42)$$

Il s'agit de trouver l'entier  $d$  et le nombre réel  $c_1$  tels que la réponse indiciale du modèle corresponde pour le mieux aux mesures issues de l'essai. Ceci peut se formaliser par exemple de la manière suivante :

$$(c_1 \quad d_1) = \arg \min \sum_{t=1}^N (y_{\text{modèle}}(t) - y_{\text{mesure}}(t))^2 \quad (43)$$

Cette optimisation peut se faire à la main ou grâce à une procédure automatisée (avec le célèbre outil de calcul scientifique).

Dans le cas où le système comprend naturellement une intégration (par exemple la commande d'un moteur à courant continu par une tension et la mesure de l'angle de rotation) il est utile de prendre en compte explicitement cette intégration dans le modèle :

$$y(z) = \frac{1}{1 - z^{-1}} \cdot \frac{(1 + c_1) \cdot z^{-d}}{1 + c_1 \cdot z^{-1}} \cdot u(z) \quad (44)$$

La procédure est la même mais au lieu de prendre en compte la sortie  $y(t)$  on prend en compte la dérivée  $y'(t)$  :

$$y'(t) = y(t) - y(t-1) \quad (45)$$

## 5.4 Identification non paramétrique

Pour terminer ce panorama rapide sur les méthodes d'identification il faut parler des méthodes non paramétriques : celles qui permettent de tracer la réponse impulsionnelle ou la réponse fréquentielle du système.

### 5.4.1 Estimation de la réponse impulsionnelle

La réponse impulsionnelle d'un système peut être obtenue simplement en appliquant une entrée de type "impulsion" au système et d'en mesurer les conséquences.

La seconde méthode consiste à utiliser la méthode des moindres carrés (ou une méthode dérivée) comme vue au chapitre 5.1. La réponse impulsionnelle d'un système peut en effet être vue comme la fonction de transfert particulière où le dénominateur est de degré nul et le numérateur comporte un nombre infini (en pratique : grand) d'éléments ( $n = 0$  et  $m$  grand) :

$$y(t) = b_0 \cdot u(t) + b_1 \cdot u(t-1) + b_2 \cdot u(t-2) + \dots + b_m \cdot u(t-m) \quad (46)$$

La dernière méthode, qui sera développée ici est l'analyse d'intercorrélation.

Par définition l'intercorrélation  $R_{yu}$  entre deux signaux est :

$$R_{yu}(\tau) = E[(y(t) - E(y(t))) \cdot (u(t-\tau) - E(u(t-\tau)))] \quad (47)$$

Dans cette formule l'espérance mathématique peut être approchée grâce à la loi des grands nombres :

$$R_{yu}(\tau) \approx R_{yu}^N(\tau) = \lim_{N \rightarrow \infty} \frac{1}{N} \sum_{t=1}^N [(y(t) - m_y) \cdot (u(t-\tau) - m_u)] \quad (48)$$

On s'intéresse maintenant à un système tel que décrit par l'équation impulsionnelle (46) et on suppose que le signal d'entrée est un bruit blanc. Le signal d'entrée vérifie donc :

$$\begin{cases} \tau \neq 0 & R_{uu}(\tau) = 0 \\ \tau = 0 & R_{uu}(\tau) = R \end{cases} \quad (49)$$

Il vient alors naturellement :

$$b_k \approx \frac{1}{R} R_{yu}(k) \quad (50)$$

#### 5.4.2 Estimation de la réponse fréquentielle

On utilise ici le spectre d'intercorrélation. Par définition le spectre d'intercorrélation entre deux signaux est :

$$\Phi_{yu}(\omega) = \sum_{\tau=-\infty}^{\infty} R_{yu}(\tau) \cdot e^{-i\omega\tau} \quad (51)$$

On suppose que le signal d'entrée  $u(t)$  et le signal de sortie  $y(t)$  sont reliés par une fonction de transfert linéaire du type :

$$y(t) = F \cdot u(t) + v(t) \quad (52)$$

où  $v(t)$  est une perturbation indépendante de  $u(t)$ . On a alors les relations :

$$\Phi_{yu}(\omega) = F(j\omega) \cdot \Phi_{uu}(\omega) \quad (53)$$

Il reste à estimer les spectres d'intercorrélation pour remonter à la réponse fréquentielle  $F(j\omega)$  recherchée. Une idée naturelle est d'utiliser la relation (48) mais cela marche très mal dans presque tous les cas, même en choisissant  $N$  assez grand. L'amélioration est d'utiliser une "fenêtre" pour favoriser les valeurs faibles de  $\tau$  par rapport aux valeurs élevées. La solution classique utilisant la *fenêtre de Hamming* donne :

$$\Phi_{yu}^N(\omega) = \sum_{\tau=-\gamma}^{\gamma} R_{yu}^N(\tau) \cdot w_{\gamma}(\tau) \cdot e^{-i\omega\tau} \quad (54)$$

avec

$$\begin{cases} w_{\gamma}(\tau) = \frac{1}{2} \left( 1 + \cos\left(\frac{\pi \cdot \tau}{\gamma}\right) \right) & |\tau| < \gamma \\ w_{\gamma}(\tau) = 0 & |\tau| > \gamma \end{cases}$$

On utilise ensuite la relation (51) pour estimer la réponse fréquentielle :

$$F(j \cdot \omega) = \frac{\Phi_u^N(\omega)}{\Phi_{yu}^N(\omega)} \quad (55)$$





## 6. SYSTEMES BOUCLES

### 6.1 Fonction de transfert en boucle fermée

Un système et sa commande par ordinateur peut être représenté par un ensemble de fonctions de transfert formant un schéma-bloc plus ou moins compliqué (Figure 19).

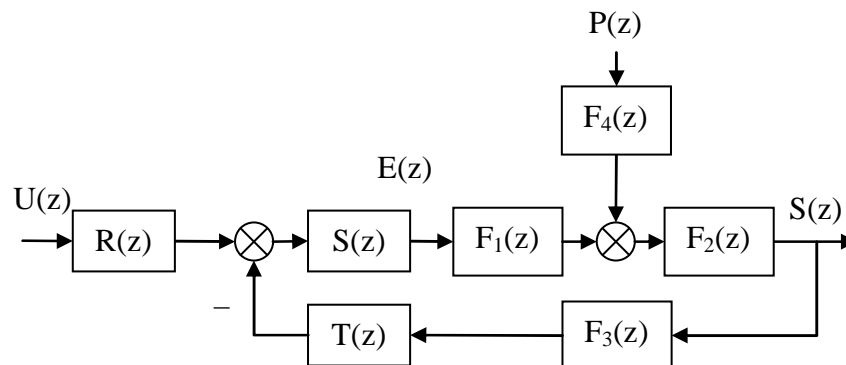


Figure 19 : système discret commandé par ordinateur. Le correcteur est représenté par les fonctions R,S,T, le système par  $F_1$  et  $F_2$ , le modèle de perturbations par  $F_4$  et le modèle des actionneurs par  $F_4$ .

Quelle que soit la structure d'un système asservi il est possible de se ramener à une forme canonique comprenant une fonction de transfert en boucle ouverte en combinant judicieusement les fonctions de transfert du procédé, du régulateur et des perturbations. La Figure 20 représente par exemple le système de la Figure 19 où  $F$  est une combinaison des fonctions de transfert du système, du régulateur, des perturbations et des capteurs.

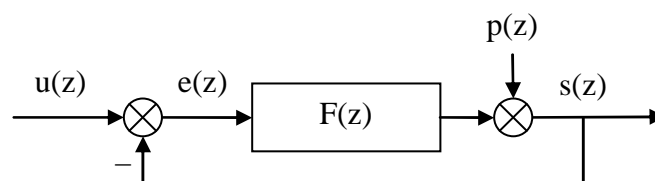


Figure 20 : système discret, notations.

Par défaut on considère une contre-réaction négative et une perturbation additive sur la sortie. La fonction de transfert  $F$  est appelée *fonction de transfert en boucle ouverte*. Comme pour les systèmes continus la *fonction de transfert en boucle fermée* est obtenue simplement à partir de  $F$  (58).

$$\begin{aligned}
 F(z) &= \frac{s(z)}{e(z)} = \frac{B(z^{-1})}{A(z^{-1})} \\
 G(z) &= \frac{s(z)}{u(z)} = \frac{B(z^{-1})}{A(z^{-1}) + B(z^{-1})}
 \end{aligned}
 \tag{56}$$

## 6.2 Erreur statique

L'erreur statique du système en boucle fermée, comme pour le système en boucle ouverte, est simplement le rapport des coefficients.

$$G_0 = \frac{\sum_{i=0}^{i=m} b_i}{1 + \sum_{i=0}^{i=m} b_i + \sum_{i=1}^{i=n} a_i}
 \tag{57}$$

Si l'on souhaite que l'asservissement confère un gain statique unitaire au système en boucle fermée (c'est-à-dire que l'on force la sortie  $s(k)$  à tendre vers la valeur de consigne  $u(k)$ ) il faut que ce rapport de coefficients soit égal à un. On montre facilement que cela est obtenu si la fonction de transfert  $F(z)$  comporte un intégrateur :

$$F(z) = \frac{s(z)}{e(z)} = \frac{1}{1 - z^{-1}} \cdot \frac{B(z)}{A'(z)}
 \tag{58}$$

## 6.3 Rejet de perturbations

La fonction de transfert entre la perturbation  $p(z)$  et la sortie  $y(z)$  est appelée fonction de sensibilité perturbation-sortie, notée  $S_{yp}$ .

$$S_{yp}(z) = \frac{s(z)}{p(z)} = \frac{A(z^{-1})}{A(z^{-1}) + B(z^{-1})}
 \tag{59}$$

Si l'on souhaite qu'une perturbation de type échelon soit parfaitement rejetée il faut que  $A(1) = 0$ , on retrouve donc la condition (58) à savoir que le système doit comporter une intégration dans la boucle directe. On peut être en outre être amené à spécifier que les perturbations ne soient pas amplifiées dans une certaine bande de fréquence. On rajoute donc une contrainte du type  $|S_{yp}(e^{j2\pi f})| < S_{\max}$  pour  $f$  appartenant à un certain domaine de  $[0 \ 0.5]$ .

## 6.4 Stabilité des systèmes à simple gain de rétroaction

Les critères de Routh et de Jury (donnés dans le chapitre 3.2.7 p13) ou le calcul numérique des pôles permettent de déterminer la stabilité du système en boucle fermée si l'on connaît sa fonction de transfert.

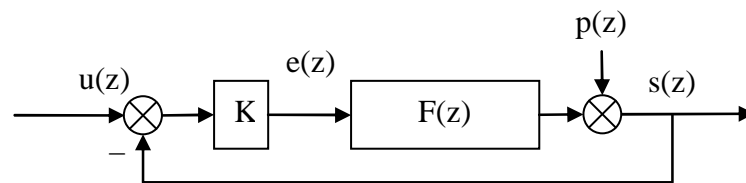


Figure 21 : système bouclé par un simple gain.

On s'intéresse ici au cas particulier où l'asservissement est réalisé par un simple gain de rétroaction  $K$  (Figure 21). On cherche la valeur de  $K$  qui amène le système en limite de stabilité. L'outil adapté ici est le lieu d'Evans ou lieu des racines (root locus) qui permet de visualiser l'évolution des pôles du système bouclé lorsque le gain  $K$  varie.

Prenons l'exemple d'un système comportant trois pôles et deux zéros. L'exemple est donné sous forme de script Matlab :

### Matlab

```
% Définition d'un système discret :
sys=zpk([-2.037 -0.1406],[0.6065 0.3679 0.3679],1,1)
%Tracé du lieu des racines
rlocus(sys);
```

Le script précédent produit le tracé de la Figure 22. On pourrait y lire que la limite de stabilité est obtenue pour  $K=0.2$  environ (détail de la Figure 23).

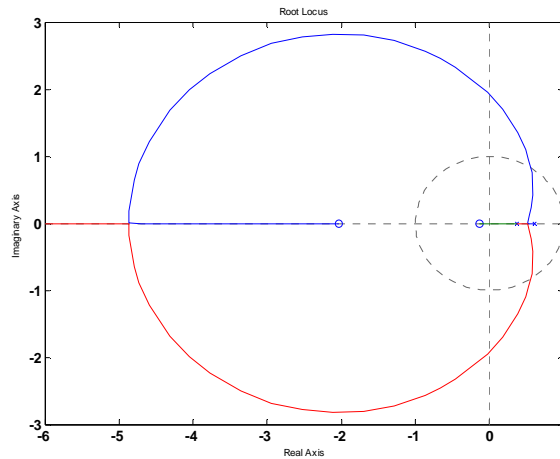


Figure 22 : lieu des racines

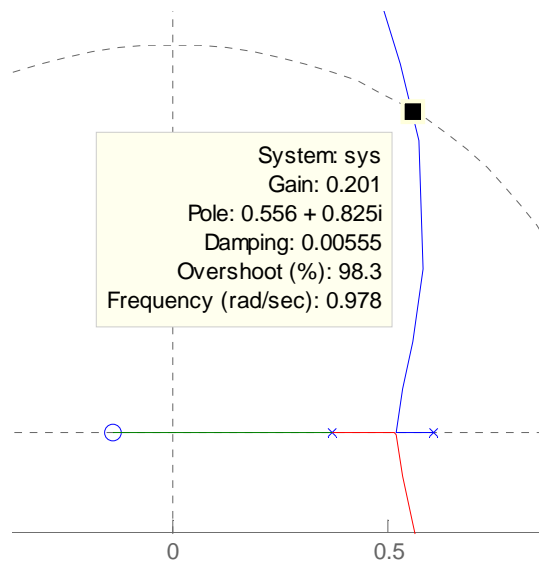


Figure 23 : lieu des racines : détail de la limite de stabilité

On voit sur la Figure 22 que le lieu des racines d'un système échantillonné ressemble à celui d'un système continu :

- on commence par tracer les pôles ( $\times$ ) et les zéros ( $o$ ) du système en boucle ouverte.
- les pôles du système en boucle fermé décrivent un lieu qui part des pôles (pour  $K = 0$ ) et arrive aux zéros (pour  $K = \infty$ )
- lorsque il y a  $n-m$  plus de pôles que de zéros le système comporte  $n-m$  branches asymptotiques.

On ne décrira pas ici les diverses propriétés du lieu des racines qui permettent de le tracer à la main puisque il est facile à obtenir avec un ordinateur.

Il peut aussi être utilisé pour régler "à la main" le gain permettant d'obtenir un ou plusieurs modes dominants ayant certaines propriétés de fréquence et d'amortissement. On voit sur la Figure 24 que le gain 0.017 permet d'obtenir un pôle double d'amortissement 0.706 et de fréquence 0.65 rad/s et un pôle simple de fréquence 1.63 rad/s.

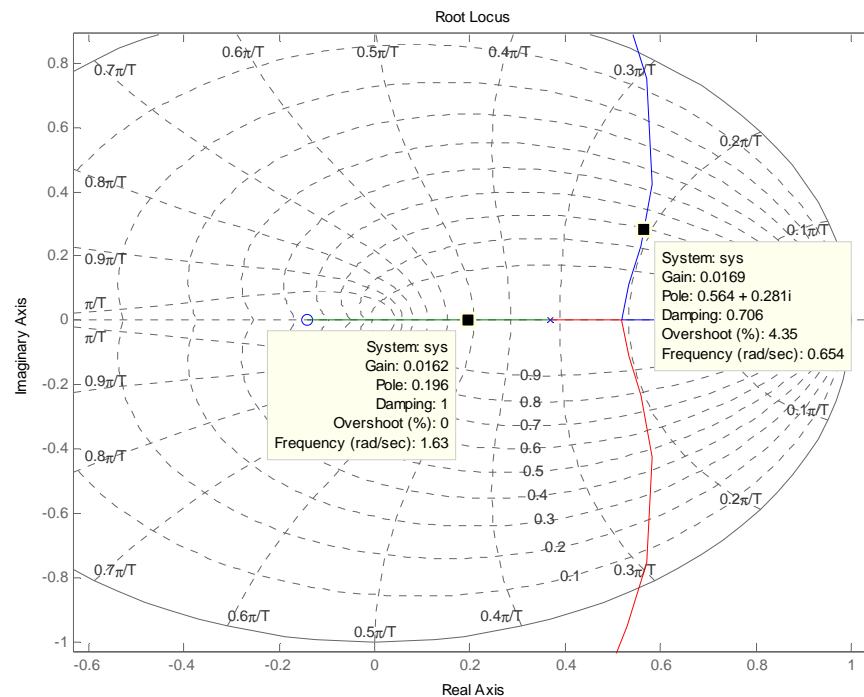


Figure 24 : utilisation du lieu des racines pour régler le gain de la boucle fermée

L'outil rltool de Matlab permet de nombreuses manipulations graphiques de ce type.

## 6.5 Analyse des systèmes bouclés dans le domaine fréquentiel : stabilité

Comme dans le cas continu il est possible de prévoir des propriétés, en particulier la stabilité, d'un système en boucle fermée à partir de l'examen de la fonction de transfert en boucle ouverte.

On ne redémontrera pas ici le théorème de Nyquist mais on se contente de rappeler l'énoncé.

On rappelle que le *lieu de Nyquist* est le lieu de  $F(j2\pi f)$  dans le plan complexe pour  $f$  variant de 0 à  $+\infty$ . A noter que le lieu pour la zone  $[0, 0.5]$  est symétrique de celui pour la zone  $[0.5, 1]$ . Pour cette raison on ne représente généralement que le lieu pour  $f$  appartenant à  $[0, 0.5]$ .

### 6.5.1 Théorème de Nyquist dans le cas où le système est stable en boucle ouverte (critère du revers)

Le système est stable en boucle fermée si le lieu de Nyquist de la fonction de transfert en boucle ouverte parcouru dans le sens des fréquences croissantes de 0 à 0.5 laisse le point critique  $(-1,0)$  à gauche.

La Figure 25 représente par exemple le lieu de Nyquist d'une fonction de transfert en boucle ouverte qui sera stable en boucle fermée.

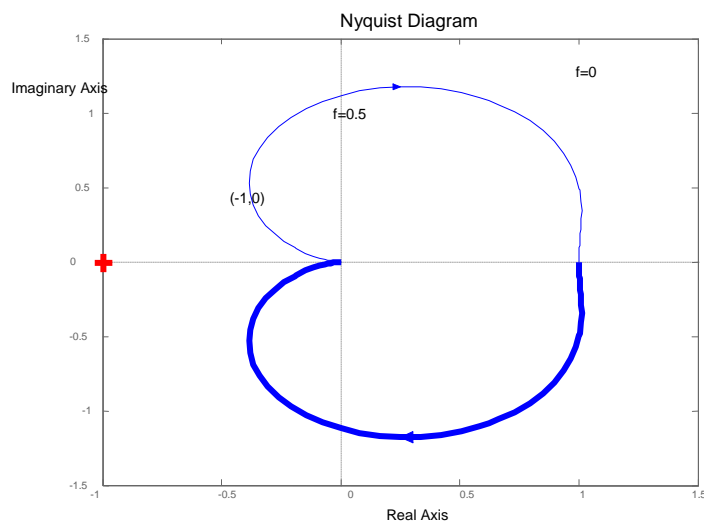


Figure 25 : lieu de Nyquist d'un système stable

### 6.5.2 Théorème de Nyquist dans le cas où le système est instable en boucle ouverte

Si le système en boucle ouverte comporte  $N$  pôles instables il sera stable en boucle fermée si le lieu de Nyquist de la fonction de transfert en boucle ouverte parcouru dans le sens des fréquences croissantes de  $-0.5$  à  $0.5$  entoure le point critique  $(-1,0)$   $N$  fois dans le sens des aiguilles d'une montre..

### 6.5.3 Interprétations du lieu de Nyquist

Comme dans le cas continu, la distance à laquelle on passe du point critique renseigne sur la robustesse du système bouclé. La distance sera exprimée de différente manière (Figure 26) :

- marge de module  $\Delta M$  : plus petite distance entre le lieu des racines et le point critique,

- marge de gain  $\Delta G$
- marge de phase  $\Delta P$

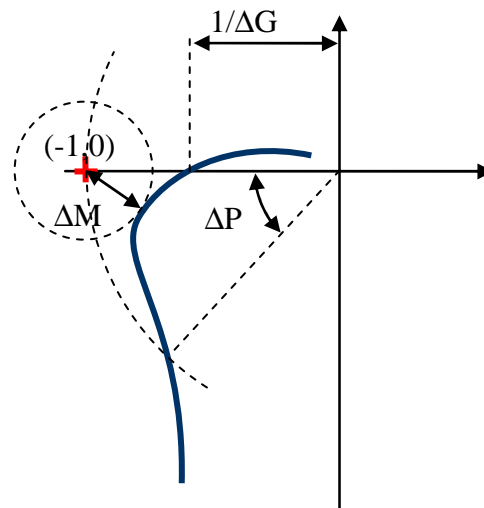


Figure 26 : Marges de stabilité

Les valeurs typiques de ces marges sont données ci-après :

- $\Delta G \geq 6\text{dB}$
- $\Delta M \geq -6\text{dB}$
- $30^\circ \leq \Delta P \leq 60^\circ$

La distance entre le point critique et le lieu des racines est la norme de  $1+F(j2\pi f)$ . On peut donc exprimer la marge de module en fonction de la fonction de sensibilité :

$$\begin{aligned} \Delta M &= \min |1 + F(j \cdot \omega)|_{\omega \in [0, \pi]} \\ \Delta M &= \min \left| 1 + \frac{B(j \cdot \omega)}{A(j \cdot \omega)} \right|_{\omega \in [0, \pi]} = \max \left| \frac{A(j \cdot \omega)}{A(j \cdot \omega) + B(j \cdot \omega)} \right|_{\omega \in [0, \pi]} \quad (60) \\ \Delta M &= \max |S_{yp}(j \cdot \omega)|_{\omega \in [0, \pi]}^{-1} \\ \Delta M_{\text{dB}} &= -\max |S_{yp}(j \cdot \omega)|_{\text{dB}} \end{aligned}$$

Si l'on trace l'évolution de  $S_{yp}$  en fonction de  $f$  on obtient donc la valeur de la marge de module.





# 7. METHODES DE REGULATION DES SYSTEMES ECHANTILLONES

## 7.1 Introduction

Les bases théoriques permettant de concevoir le correcteur pour un système échantillonné ont été posées dans les chapitres précédents. Ce chapitre fait le bilan des connaissances acquises et donne quelques solutions pour calculer des correcteurs à partir d'un cahier des charges précis.

Les systèmes que l'on cherche à commander sont généralement, par nature, continus. La réalisation d'un correcteur par un ordinateur implique que l'on génère une équation de commande discrète (généralement une équation récurrente). L'obtention de cette équation de commande peut se faire de deux manières :

- la première méthode consiste à d'abord synthétiser un correcteur continu (en utilisant les outils classiques de l'automatique et la simulation) et discrétiser ensuite ce correcteur. C'est l'objet du chapitre 7.2.2.
- la seconde méthode consiste à tenir la représentation discrète du système continu associé à son interface numérique (convertisseurs analogique-numérique et numérique-analogique) comme expliqué dans le chapitre 4.5 puis synthétiser directement le correcteur discret. Cela sera expliqué dans le chapitre 7.3 et suivants.

Ce choix peut être résumé par la figure suivante :

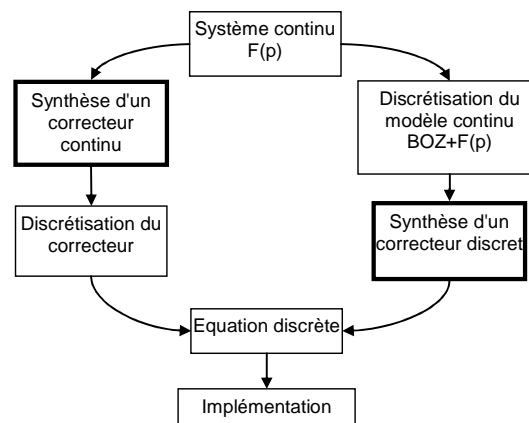


Figure 27 : Réalisation d'un correcteur discret.

Au-delà des affaires de goût, le choix de la méthode sera souvent dicté par l'application. En effet, la première méthode est d'autant meilleure que le correcteur discrétisé approche correctement le correcteur continu. Cela implique que le temps d'échantillonnage soit très faible par rapport aux constantes de temps du système

bouclé. Dans le cas où les contraintes temps réel sont fortes (capacités de calcul partagées par exemple) il sera préférable de synthétiser un correcteur discret.

## 7.2 Structure d'un système commandé par ordinateur, hypothèses

Par la suite on se ramènera toujours à un système tel que celui représenté sur la Figure 28 en cherchant un correcteur  $K(z)$  linéaire. On montre que quel que soit le correcteur linéaire on peut se ramener à la forme représenté sur la Figure 29 (forme RST).

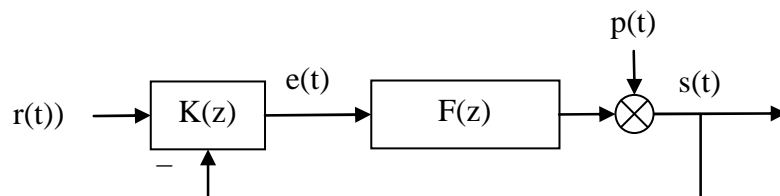


Figure 28 : système bouclé, forme générale.

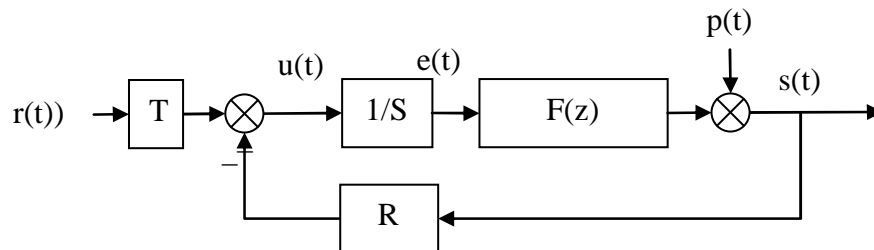


Figure 29 : système bouclé, forme canonique.

### 7.2.1 Discrétisation, choix de la période d'échantillonnage

Du point de vue du ordinateur, l'extérieur (le procédé) est représenté par la fonction de transfert  $F(z)$ . L'hypothèse importante est que la commande (sortie du ordinateur) et la mesure (entrée du ordinateur) sont échantillonnées de manière synchrone à la même période  $T_e$ .

Le choix judicieux de la fréquence d'échantillonnage est très important. En pratique on choisit cette fréquence en fonction de la bande passante du système en **boucle fermée** :

$$f_e = (6 \text{ à } 25) f^{BF} \quad (61)$$

### 7.2.2 Discrétisation d'un correcteur continu

La première méthode pour obtenir les équations d'un correcteur numérique est de faire sa synthèse avec les outils de l'automatique continue puis de discrétiser la fonction de transfert du correcteur continu obtenu (voir pour cela le chapitre 4.2 *Echantillonnage des systèmes continus*).

On va voir par la suite que la synthèse directe d'un correcteur discret offre plus de degré de liberté et permet en général de remplir un cahier des charges plus complet.

## 7.3 Régulateur PID numérique

Le régulateur le plus populaire est le régulateur PID. Nous allons voir dans un premier temps comment on génère un régulateur PID à partir des équations bien connues du régulateur PID continu et comment ces équations peuvent être généralisées pour obtenir plus de degrés de liberté pour le réglage.

### 7.3.1 Cas 1

Un régulateur PID continu a une fonction de transfert du type :

$$K_{\text{PID}}(s) = K \cdot \left( 1 + \frac{1}{T_i \cdot s} + \frac{T_d \cdot s}{1 + \frac{T_d}{N} \cdot s} \right) \quad (62)$$

La discrétisation de ce correcteur continu par la méthode d'Euler (Tableau 4 page 31) se fait en remplaçant la variable de Laplace  $s$  par le terme  $\frac{1-q^{-1}}{q^{-1} \cdot T_e}$ . On obtient :

$$K_{\text{PID}}(q) = K \cdot \left( 1 + \frac{1}{T_i \cdot \frac{1-q^{-1}}{T_e}} + \frac{T_d \cdot \frac{1-q^{-1}}{T_e}}{1 + \frac{T_d}{N} \cdot \frac{1-q^{-1}}{T_e}} \right) \quad (63)$$

En réarrangeant on obtient la forme générale RST (Figure 30):

$$K_{PID}(q) = K \cdot \frac{r_0 + r_1 \cdot q^{-1} + r_2 \cdot q^{-2}}{(1 - q^{-1}) \cdot (1 + s_1' \cdot q^{-1})}$$

$$\begin{cases} r_0 = K \cdot \left( 1 + \frac{T_e}{T_i} + \frac{N \cdot T_d}{T_d + N \cdot T_e} \right) \\ r_1 = -K \cdot \left( 1 + \frac{T_d}{T_d + N \cdot T_e} + \frac{T_e}{T_i} \cdot \frac{T_d}{T_d + N \cdot T_e} + \frac{N \cdot T_d}{T_d + N \cdot T_e} \right) \\ r_2 = K \cdot \left( \frac{T_d}{T_d + N \cdot T_e} \right) \\ s_1' = \frac{T_d}{T_d + N \cdot T_e} \end{cases} \quad (64)$$

Comme on le voit le correcteur obtenu comporte autant de degrés de liberté que le correcteur PID analogique. On reconnaît au dénominateur le terme  $(1 - q^{-1})$  en facteur qui assure l'effet d'intégration.

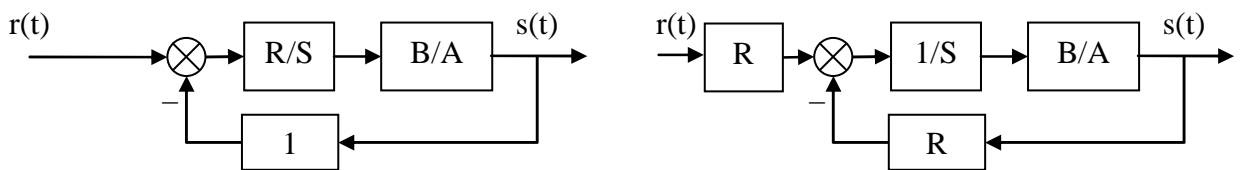


Figure 30 : PID sous forme RST

La fonction de transfert en boucle fermée obtenue est donnée par l'équation suivante :

$$H_{BF} = \frac{B \cdot R}{A \cdot S + B \cdot R} = \frac{B \cdot R}{P} \quad (65)$$

De manière générale le polynôme caractéristique  $P$  fixe les performances visées en boucle fermée. Le calcul de  $R$  et  $S$ ,  $P$  étant fixé et  $A$  et  $B$  étant connus, est réalisé en résolvant l'équation de Bézout  $P = A \cdot S + B \cdot R$ . (voir 9.1 Annexe 1 : Résolution de l'équation polynomiale de Bezout)

On voit que le numérateur  $B \cdot R$  est imposé une fois que  $R$  et  $S$  sont trouvés. Le régulateur introduit donc naturellement des zéros dans la boucle fermée.

Traisons l'exemple entièrement dans le cas où le système est modélisé comme un système du deuxième ordre avec faible retard. La fonction de transfert du procédé est donnée par :

$$H_{BO} = \frac{B}{A} = \frac{b_1 \cdot q^{-1} + b_2 \cdot q^{-2}}{1 + a_1 \cdot q^{-1} + a_2 \cdot q^{-2}} \quad (66)$$

Le correcteur PID rajoutant deux pôles, le polynôme caractéristique en boucle fermée spécifié doit posséder quatre pôles. Le polynôme caractéristique en boucle fermée doit donc être de la forme :

$$\begin{aligned} P &= 1 + p_1 \cdot q^{-1} + p_2 \cdot q^{-2} + p_2 \cdot q^{-3} + p_2 \cdot q^{-4} \\ P &= (1 + p_1' \cdot q^{-1} + p_2' \cdot q^{-2}) \cdot (1 + \alpha_1 \cdot q^{-1}) \cdot (1 + \alpha_2 \cdot q^{-1}) \end{aligned} \quad (67)$$

Et il doit être identifié au produit  $AS + BR$  avec  $R$  et  $S$  ayant les formes données par (64) :

$$\begin{aligned} &(1 + p_1' \cdot q^{-1} + p_2' \cdot q^{-2}) \cdot (1 + \alpha_1 \cdot q^{-1}) \cdot (1 + \alpha_2 \cdot q^{-1}) \\ &= (1 + a_1 \cdot q^{-1} + a_2 \cdot q^{-2}) \cdot (1 - q^{-1}) \cdot (1 + s_1' \cdot q^{-1}) \\ &+ (b_1 \cdot q^{-1} + b_2 \cdot q^{-2}) \cdot (r_0 + r_1 \cdot q^{-1} + r_2 \cdot q^{-2}) \end{aligned} \quad (68)$$

Les coefficients  $p_1'$  et  $p_2'$  définissent un modèle du second ordre aux caractéristiques (pulsation propre  $\omega_0$  et amortissement  $\sigma$  choisis et coefficients déduits conformément au Tableau 3 page 25). Les coefficients  $\alpha_1$  et  $\alpha_2$  correspondent à des pôles auxiliaires aperiodiques. Les choix typiques pour ces coefficients sont :

$$-0.05 < \alpha_1, \alpha_2 < -0.5 \quad (69)$$

On montre que la résolution de (68) est réalisée par une inversion matricielle simple :

$$M \cdot x = p \quad (70)$$

avec

$$\begin{cases} x^T = [1 & s_1' & r_0 & r_1 & r_2] \\ p^T = [1 & p_1 & p_2 & p_3 & p_4] \end{cases}$$

et

$$M = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ a_1' & 1 & b_1 & 0 & 0 \\ a_2' & a_1' & b_2 & b_1 & 0 \\ a_3' & a_2' & 0 & b_2 & b_1 \\ 0 & a_3' & 0 & 0 & b_2 \end{pmatrix} \quad (71)$$

On montre que M est inversible à la condition que A et B soient premiers entre eux, c'est-à-dire que A et B n'aient pas de pôles et de zéros en commun.

(Le problème peut aussi être résolu en utilisant la résolution de l'équation de Bézout).

### 7.3.2 Cas 2

Comme on l'a vu le correcteur PI numérique directement transposé du correcteur PID analogique décrit dans le chapitre 7.3.1 a l'inconvénient de ramener des zéros au dénominateur de la fonction de transfert en boucle fermée (équation 65). Il est possible, grâce à la flexibilité du régulateur numérique d'imposer au correcteur de ne pas ajouter de zéros en boucle fermée. La fonction de transfert désirée en boucle fermée sera donc de la forme :

$$H_{BF} = \frac{P(1)}{B(1)} \cdot \frac{B}{P} \quad (72)$$

Le facteur  $P(1)/B(1)$  garantit le gain statique unitaire du système en boucle fermée (voir page 12). Si le régulateur a la forme générale RST alors les polynômes R S et T doivent vérifier :

$$H_{BF} = \frac{B \cdot T}{A \cdot S + B \cdot R} = \frac{P(1)}{B(1)} \frac{B}{P} \quad (73)$$

S et R sont obtenus par la résolution de la même équation que précédemment (68). Le polynôme T est obtenu par simple identification :  $T = R(1)$ .

La structure, sous forme de schéma-bloc est donnée ci-dessous :

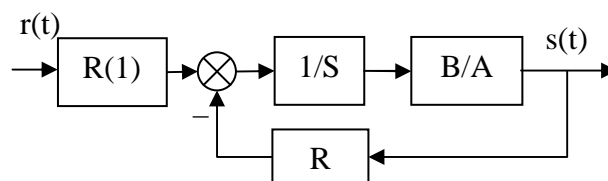


Figure 31 : PID2 sous forme RST

On montre que cette structure de PID, ramené en continu, correspond au PID canonique pour l'action intégrale. L'action proportionnelle et dérivée n'est appliquée que sur le signal  $y$ .

La structure du PID2 par rapport au PID1 a pour avantage de ne pas ramener de zéros à la boucle fermée. Cela se traduit par un dépassement moindre en transitoire lorsqu'on applique un échelon de commande. Par contre les deux systèmes réagissent de la même manière à un échelon de perturbation.

## 7.4 Placement de pôles

Les deux structures de PID décrites dans le chapitre précédent sont deux cas particulier d'une stratégie plus générale de placement de pôles.

La synthèse est plus générale que les PID présentés précédemment puisqu'il n'est pas nécessaire de faire d'hypothèses sur le degré de A et B, sur le retard, ni sur la présence ou non de zéros stables ou instables dans le procédé.

La structure du correcteur est la structure type RST telle que présentée dans la Figure 29. Pour la suite des calculs on posera pour la fonction de transfert les conventions suivantes :

$$H_{BO} = \frac{q^{-d} \cdot B(q^{-1})}{A(q^{-1})} \quad (74)$$

$$\begin{cases} A(q^{-1}) = 1 + a_1 \cdot q^{-1} + \dots + a_{n_A} \cdot q^{-n_A} \\ B(q^{-1}) = b_1 \cdot q^{-1} + \dots + b_{n_B} \cdot q^{-n_B} = q^{-1} \cdot B^*(q^{-1}) \end{cases}$$

Compte tenu de la forme RST du régulateur la fonction de transfert en boucle fermée vérifie :

$$H_{BF} = \frac{q^{-d} \cdot B(q^{-1}) \cdot T(q^{-1})}{P(q^{-1})} \quad (75)$$

$$P(q^{-1}) = A(q^{-1}) \cdot S(q^{-1}) + q^{-d} \cdot B(q^{-1}) \cdot R(q^{-1}) = 1 + p_1 \cdot q^{-1} + \dots + p_{n_P} \cdot q^{-n_P}$$

Le polynôme  $P(q^{-1})$  définit donc les pôles de la fonction de transfert en boucle fermée et traduit donc le comportement en *poursuite*.

De la même manière la fonction de transfert entre une perturbation et la sortie (fonction de sensibilité perturbation-sortie) est donnée par :

$$S_{yp} = \frac{A(q^{-1}) \cdot S(q^{-1})}{P(q^{-1})} \quad (76)$$

Ce même polynôme traduit donc aussi le comportement en *régulation*.

Les consignes en poursuite et en régulation peuvent être différentes comme cela est représenté par exemple sur la Figure 32.

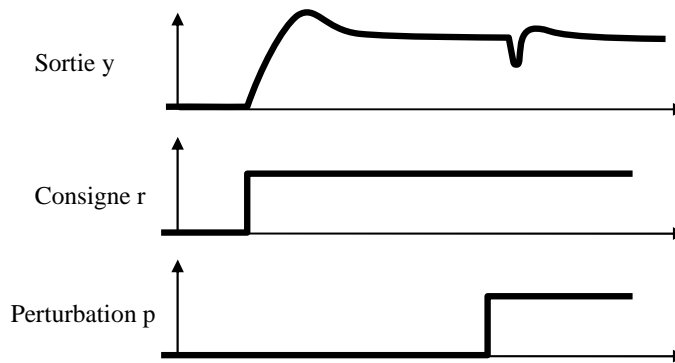


Figure 32 : Performances différentes en régulation et en poursuite

### 7.4.1 Choix des pôles en boucle fermée

De manière générale le comportement en boucle fermée est spécifié sous la forme d'une pulsation propre  $\omega_0$  et d'un amortissement  $\sigma$ , donc une fonction de transfert du second ordre. La spécification étant généralement donnée en continu il est nécessaire de traduire la valeur désirée de  $\omega_0$  et  $\sigma$  en un polynôme  $P_D(q^{-1})$  par discrétisation de la fonction de transfert continue (chapitre 4.5 page 29). Ce polynôme  $P_D$  traduit le comportement dominant du système bouclé.

Si l'on souhaite introduire un effet de filtrage supplémentaire dans certaines bandes de fréquence (réduction des bruits de mesure, adoucissement de la commande, robustesse) cela est décrit par un polynôme  $P_F$  comportant des pôles auxiliaires, plus rapides que les pôles de  $P_D$ .

Le polynôme caractéristique en boucle fermée est donc constitué de pôles dominants et de pôles auxiliaires :

$$P(q^{-1}) = P_D(q^{-1}) \cdot P_F(q^{-1}) \quad (77)$$

### 7.4.2 Régulation : calcul de R et S

Le polynôme  $P(q^{-1})$  étant fixé il s'agit de trouver R et S vérifiant :

$$P(q^{-1}) = A(q^{-1}) \cdot S(q^{-1}) + q^{-d} \cdot B(q^{-1}) \cdot R(q^{-1}) \quad (78)$$

Comme déjà vu au chapitre 7.3.1 on obtient une équation de Bézout qu'il s'agit de résoudre. (Voir aussi 9.1 Annexe 1 : Résolution de l'équation polynomiale de Bezout)

Il est parfois nécessaire de spécifier R et S avec des parties fixes  $H_R$  et  $H_S$ . Dans ce cas l'équation (78) est modifiée et on l'écrit sous la forme:



$$P(q^{-1}) = (A(q^{-1}) \cdot H_S(q^{-1})) \cdot S'(q^{-1}) + (q^{-d} \cdot B(q^{-1}) \cdot H_R(q^{-1})) \cdot R'(q^{-1}) \quad (79)$$

Les inconnues que l'on souhaite trouver sont  $S'$  et  $R'$ . Ceci est nécessaire dans les cas particuliers suivants :

- spécification d'erreur statique nulle,
- rejet d'une perturbation harmonique,
- blocage d'un signal,

Détaillons maintenant ces cas particuliers.

### Spécification d'une erreur statique nulle

On retrouve la nécessité d'avoir un terme intégrateur dans la boucle directe. La spécification se retrouve donc sur  $S$  :

$$S(q^{-1}) = (1 - q^{-1}) \cdot S'(q^{-1}) = H_S(q^{-1}) \cdot S'(q^{-1}) \quad (80)$$

### Rejet d'une perturbation harmonique

Si le système est perturbé par une perturbation harmonique à une fréquence fixe (par exemple 50Hz...) il est possible d'imposer à  $S$  d'avoir un module *nul* ou amorti à cette fréquence. Cela se traduit par la présence dans  $S$  d'une paire de zéros complexe à amortissement nul ou faible à cette fréquence :

$$S(q^{-1}) = (1 + \alpha \cdot q^{-1} + \beta \cdot q^{-2}) \cdot S'(q^{-1}) = H_S(q^{-1}) \cdot S'(q^{-1}) \quad (81)$$

### Blocage d'un signal

Pour certaines application on souhaite que le régulateur "laisse passer" certaines composantes du signal de sortie, c'est-à-dire qu'il ne réagisse pas à ces composantes. Cela se traduit par  $R=0$  à ces fréquences. On retrouve la même forme que précédemment :

$$R(q^{-1}) = (1 + \alpha \cdot q^{-1} + \beta \cdot q^{-2}) \cdot R'(q^{-1}) = H_R(q^{-1}) \cdot R'(q^{-1}) \quad (82)$$

### 7.4.3 Poursuite : calcul de T

Le schéma-bloc de l'asservissement est donné sur la Figure 33. Le terme  $T$  est décomposé en un terme  $T'$  et un terme  $B^*/A^*$ .

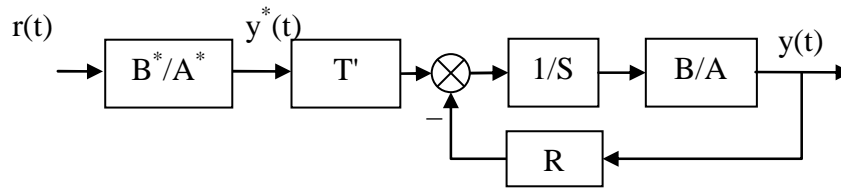


Figure 33 : placement de pôle

L'idée est de régler  $B^*$  et  $A^*$  de telle manière que le signal  $y^*$  ait la dynamique de poursuite désirée (par exemple un second ordre avec pulsation propre et amortissement donné) puis de régler  $T'$  de telle manière que le transfert entre  $y^*$  et  $y$  soit le plus "transparent" possible.

### Réglage de $T'$

Le transfert entre  $y^*$  et  $y$  est :

$$S_{yy^*}(q^{-1}) = T'(q^{-1}) \frac{q^{-d} \cdot B(q^{-1})}{P(q^{-1})} \quad (83)$$

Pour assurer le transfert le plus "transparent" possible il faut donc choisir

$$T'(q^{-1}) = \frac{P(q^{-1})}{B(1)} \quad (84)$$

Remarques :

- Si  $B(1)=0$  alors on remplace  $B(1)$  par 1 dans (84)
- De même dans (84)  $P(q^{-1})$  peut être remplacé par  $P_D$  correspondant à la dynamique dominante.

### Réglage de $B^*$ et $A^*$

$B^*$  et  $A^*$  sont choisis de telle manière que la fonction de transfert  $B^*/A^*$  corresponde à la dynamique de poursuite souhaitée.

## 7.5 Poursuite et régulation à objectifs indépendants

Cette méthode ressemble à la méthode de placement de pôles à ceci près que l'on va effectuer une simplification des zéros de  $B^*(q^{-1})$  : cela ne s'applique donc qu'aux systèmes avec des pôles stables. Plus précisément il faut qu'ils aient un amortissement suffisant ( $> 0,2$ ) de manière à ne pas être trop près du cercle unité.

La structure de l'asservissement est toujours celle donnée sur la Figure 33.

**Choix de P**

$P(q^{-1})$  est choisi de manière à simplifier  $B^*(q^{-1})$  (rappel : la forme de B est donnée dans l'équation 74). Comme indiqué dans le chapitre 7.4.2 page 60 il s'agit donc d'imposer les zéros de  $B(q^{-1})$  dans  $H_R(q^{-1})$  et  $H_S(q^{-1})$ .

**Choix de T**

Si l'on choisit  $T(q^{-1}) = P(q^{-1})$  alors le transfert entre  $y^*$  et  $y$  est :

$$S_{yy^*}(q^{-1}) = q^{-(d+1)} \quad (85)$$

**Réglage de  $B^*$  et  $A^*$** 

$B^*$  et  $A^*$  sont choisis de telle manière que la fonction de transfert  $B^*/A^*$  corresponde à la dynamique de poursuite souhaitée.



## 8. TECHNOLOGIE DE LA COMMANDE PAR ORDINATEUR

Du point de vue de l'automaticien le ordinateur est relié au système physique à piloter par un certain nombre d'intermédiaires plus ou moins gênants :

- les interfaces analogiques et numériques,
- le transport de l'information par bus et protocole,
- l'utilisation éventuelle de capteurs ou d'actionneurs « intelligents ».

Il ne s'agit pas ici de faire un exposé complet sur ces trois aspects mais de poser les problèmes qui se posent à l'automaticien lorsque le système qu'il doit commander est perturbé par ces éléments.

Enfin ce chapitre se termine sur les notions de « Hardware In the Loop » et de « prototypage rapide » qui sont partie intégrante des processus de réalisation de systèmes de contrôle.

### 8.1 Calculateurs et interfaces numériques / analogiques

Le premier problème à prendre en compte est la quantification de l'information. Du point de vue du ordinateur d'une part et du point de vue des convertisseurs d'autre part. Dans un premier temps, considérons les aspects performance de calcul lié à la nature du codage des nombres dans le ordinateur.

#### Les performances liées au types de variables

Prenons l'exemple où le concepteur de la loi de commande souhaite réaliser un correcteur de type proportionnel. Entre l'acquisition des mesures et l'écriture de la consigne il y aura la réalisation d'une de commande du type :

$$y = K \cdot (\text{consigne} - \text{mesure}) \quad (86)$$

En utilisant un langage de programmation évolué (par exemple le C) la ligne de code sera la transcription littérale de (86). Le programmeur devra décider de la définition et du type des variables, principalement du nombre d'octets utilisés pour stocker ses variables. Le Tableau 5 résume par exemple les types utilisés en langage C (norme ANSI)

Tableau 5 : Types ANSI C

Types entiers	char	8 bits
	short int	16 bits
	long int	32 bits
Types flottants (mantisse et exposant)	float	32 bits
	double	64 bits
	longdouble	128 bits

Une fois le code écrit en C l'exécution du programme sur un processeur cible ne peut se faire que par l'intermédiaire d'une compilation du programme. A chaque processeur est donc adapté une version de compilateur.

Les performances du programme (sa rapidité d'exécution) dépendront donc en partie de la qualité du compilateur. (Par exemple, une multiplication s'effectue t'elle bêtement par un décalage de tous les bits ou bien des solutions astucieuses sont-elles implémentées pour arrêter le décalage selon le contexte ?)

Le principal déterminant de la performance est cependant le processeur : son type (8 bits, 16 bits ou 64 bits) et son cadencement (en MHz ou GHz) sont les deux plus importants paramètres. Coder les variables sur 64 bits (double) dans le programme coûtera cher en temps de calcul si le processeur est un 8 bits. Le type de mémoire et leurs temps d'accès peuvent aussi être déterminants. Enfin, certains processeurs disposent d'un processeur spécifiquement dédié aux opérations mathématiques (coprocesseur mathématique).

Il n'est pas possible de dire exactement, pour tel type de compilateur associé à tel type de processeur, le temps que prendra par exemple une multiplication. En pratique on effectue une série de tests types (benchmark) et on affiche un temps moyen d'exécution.

En résumé, on peut être tenté, pour ne pas s'embêter, de coder toutes les variables en virgule flottante. Mais il faut bien savoir que cela a un coût en terme de temps de calcul, surtout si le processeur n'est pas adapté. Pour fixer les idées on peut dire qu'il y a un rapport 20 entre une addition réalisée en "short" et une addition réalisée en virgule flottante.

Enfin, l'utilisation de fonctions mathématiques (sinus, exponentielles, etc...) peut aussi avoir un coût très lourd si le travail n'est pas réalisé par un coprocesseur mathématique dédié. Le compilateur et sa librairie mathématique joue là encore un grand rôle.

### La quantification imposée par le processeur

Au-delà de l'aspect performance et temps de calcul, la quantification de l'information doit parfois aussi être prise en compte. Rappelons qu'un entier codé sur N bits peut prendre seulement  $2^N - 1$  valeurs différentes. Avant d'effectuer les calculs mathématiques dans le corps du programme il est donc judicieux d'appliquer un

facteur d'échelle de manière à utiliser l'amplitude maximale possible (entre 0 et  $2^N - 1$ ).

### **La quantification imposée par les conversions analogique et numérique**

Les convertisseurs analogique-numérique et numérique-analogique sont principalement caractérisés par leur amplitude de signal analogique d'entrée (généralement en volts, par exemple +/- 10V) et par leur résolution en bits (8 ou 10 bits pour les plus courants, jusqu'à 16 bits pour les convertisseurs haut de gamme).

Prenons l'exemple d'un capteur gyromètre transformant une vitesse de rotation en tension. Compte tenu de l'application le signal utile varie entre 2,4V et 2,6V. Si le signal mesuré est branché directement sur un convertisseur 0-10V 10bits alors un problème de conditionnement du signal se pose. En effet la quantification est de  $\frac{10V}{2^{10}-1} = 0,098V$ . Les 0,2V de signal utile seront donc discrétisés en  $\frac{0,2}{0,098} = 20$  intervalles seulement !

Si la discrétisation du signal utile est trop grossière cela peut se traduire par une diminution des performances de l'asservissement, voire une instabilité. Le système n'est en effet plus linéaire et il n'existe pas d'outils théoriques permettant de prévoir les performances du système bouclé.

## **8.2 Transport de l'information : les bus et les protocoles**

Pas question de faire ni même d'ébaucher un cours sur les bus et les protocoles ici. Mais on va quand même donner quelques indications. L'automaticien a l'habitude de simplifier le problème qu'il se pose en dessinant un schéma simplifié de son problème d'asservissement (du type de la Figure 20 page 45) pour pouvoir appliquer les théories de la commande. Cependant il arrive bien souvent que l'information transite à l'intérieur du système, soit entre différents ordinateurs effectuant différentes parties du travail soit entre ordinateur et périphériques, par l'intermédiaire de bus informatique. A chaque type de bus correspond un ou plusieurs protocoles qui peuvent poser des problèmes pour l'automaticien :

- les données sont-elles discrétisées ? (le protocole peut par exemple ne permettre de transmettre que des paquets de données de 16 bits),
- le bus induit-t'il un retard ? (le composant mettant en forme les données du ordinateur pour les publier sur le bus a certaines performances qui peuvent induire un délai)
- le bus est-il déterministe ? (Certains bus ne garantissent pas que les données arriveront en un temps donné),
- le bus est-il sûr ? (Certains bus ne garantissent pas non plus que les données arriveront bien)

### **8.3 Capteurs et actionneurs "intelligents", "smart sensors"**

De plus en plus souvent les capteurs et les actionneurs sont associés à leur électronique de puissance, électronique de commande et de conditionnement de signal. L'interface avec le capteur ou l'actionneur se fait donc via un protocole de communication (par exemple via un bus RS232).

L'intérêt est de véhiculer non pas des signaux analogiques, tensions ou courants, mais des signaux numériques beaucoup moins sensibles aux éventuelles perturbations.

On retrouve donc, entre le capteur ou l'actionneur et le ordinateur, la présence d'un bus et d'un protocole de communication.

### **8.4 La philosophie "hardware in the loop"**

Les systèmes embarqués sont de plus en plus complexes et touchent des domaines de plus en plus variés : automobile, satellites, aéronautique, UAV, systèmes d'arme, sous-marins, systèmes, etc... Les systèmes à piloter ainsi que les systèmes de commande sont de plus en plus complexes.

Les techniques de simulation « Hardware In the Loop » (HIL) sont de plus en plus utilisées dans le développement et dans le test de ces systèmes embarqués. L'idée est de développer entièrement le système de contrôle sur une plateforme de test en utilisant le maximum de composants réels et en ne remplaçant que le minimum par la simulation. Par exemple en aéronautique l'« Iron Bird » regroupe dans un hangar tous les systèmes réels de l'avion en cours de développement. Seule la mécanique du vol est simulée : les sorties des actionneurs sont mesurées, rentrées dans un modèle de simulation numérique, puis les mesures effectuées normalement par les capteurs sont calculées et restituées par le modèle.

L'intérêt est de diminuer de manière importante les coûts et les délais en économisant le plus grand nombre possible de tests sur le système réel en situation. Lors des premiers tests réels les techniques « Hardware In the Loop » permettent d'avoir effectué la plus grande part du développement des lois de commande.

### **8.5 Le prototypage rapide en automatique**

Un peu dans le même ordre d'idée que le « Hardware In the Loop » le prototypage rapide permet de réaliser la conception des lois de commande en simulation puis de transporter rapidement (en un simple click !) le code du contrôleur simulé vers la cible temps réel implémentée sur le système réel.

La société Mathworks [9] investit massivement dans ce domaine en ayant enrichi son logiciel de calcul scientifique (Matlab) avec un logiciel de simulation graphique très évolué (Simulink) puis en développant des outils de prototypage rapide associés (Real Time Workshop, Embedded Target For...).



## 9. ANNEXE

### 9.1 Annexe 1 : Résolution de l'équation polynomiale de Bezout

Il s'agit de trouver R et S vérifiant :

$$P(q^{-1}) = A(q^{-1}).S(q^{-1}) + B(q^{-1}).R(q^{-1})$$

Où P, A et B sont connus.

Dans le cas où P, A, B, R et S sont des nombres entiers la solution de l'algorithme est donnée par l'algorithme de division Euclidienne étendu [8].

Dans le cas où P, A, B, R et S sont des polynômes, l'algorithme est donné ci-dessous, programmé en matlab ([1]).

```
function [Rp,Sp,nrp,nsp]=bezoutd(A,B,Hs,Hr,P)
%function [Rp,Sp,nrp,nsp]=bezoutd(A,B,Hs,Hr,P)
%solves AHsSp+BHrRp=P by coefficient comparison. Delay and discretization delay need to be integrated in B
%(zeros at the beginning).
%
%inputs:
%A=[a0 a1 ... aNa] ... vector of model denominator coefficients A=a0 + a1z^(-1) + a2z^(-2) +...+ aNaz^(-Na)
%B=[b0 b1 ... bNb] ... vector of model numerator coefficients B=b0 + b1z^(-1) + b2z^(-2) +...+ bNbz^(-Nb)
%Hs=[hs0 hs1 ... hsNhs] ... vector of controller denominator fixed part Hs=hs0 + hs1z^(-1) +...+ hsNhsz^(-Nhs)
%Hr=[hr0 hr1 ... hrNhr] ... vector of controller denominator fixed part Hr=hr0 + hr1z^(-1) +...+ hrNhrz^(-Nhr)
%P=[p0 p1 ... pNp] ... vector of desired polynomial coefficients P=p0 + p1z^(-1) + p2z^(-2) +...+ pNpz^(-Np)
%outputs:
%Rp=[rp0 rp1 rp2 ...] ... vector of coefficients for resulted controller numerator
%Sp=[sp0 sp1 sp2 ...] ... vector of coefficients for resulted controller denominator
%nrp ... order of Rp
%nsp ... order of Sp
%
%written by: J. Langer, I.D. Landau, H. Prochazka
%7th june 2002

PRECISION=1e-16;

D=size(A);
if D(1)>1, A=A'; end;
```

```

D=size(B);
if D(1)>1, B=B'; end;
D=size(Hs);
if D(1)>1, Hs=Hs'; end;
if D(1)==0, Hs=1; end;
D=size(Hr);
if D(1)>1, Hr=Hr'; end;
if D(1)==0, Hr=1;end;
D=size(P);
if D(1)>1, P=P'; end;

na=length(A)-1;
nb=length(B)-1;
np=length(P)-1;
nhs=length(Hs)-1;
nhr=length(Hr)-1;

if (nhs>0), Ah=real(convz(A,Hs)); else Ah=A*Hs; end; % Ah = A * Hs
nah=length(Ah) -1;
if (nhr>0), Bh=real(convz(B,Hr)); else Bh=B*Hr; end;
nbh=length(Bh) -1;
if (np>nah+nbh-1), disp('Bezout error: too many poles');end;

% increase size of P using zeros if necessary
if (np<nah+nbh-1),
%
% set remaining poles onto a circle with radius rmin
P=[P zeros(1,nah+nbh-1-np)];
rootsPdes=roots(P);
nextra=nah+nbh-1-np;
rmin=1e-16;
angle=[0:nextra-1]'/nextra*2*pi;
j=sqrt(-1);
rootsPextra=rmin*exp(j*angle);
P=poly([rootsPdes;rootsPextra]);
np=nah+nbh-1;

end;
P,

nsp=nbh-1;
nrp=nah-1;

%matrix is smaller than vector PD

```

```

if (np>nah+nbh-1),
    disp('Order of model denominator is too low! Add a polynom of higher order to Hs or Hr. ');
end;

% ns=nsp+nhs
% nr=nrp+nhr

M=[];
for j=1:nsp+1,
    V=[];
    if (j>1), V=[V ; zeros(j-1,1)]; end;% zeros in front of Ah
    V=[V ; Ah'];% Ah
    if (j<=nsp), V=[V ; zeros(nsp+1-j,1)]; end;% zeros behind Ah
    if (length(V)~=nah+nbh), disp('bezoutb: error V'); end;
    M=[M V]; % add one column to M
end;

for j=1:nrp+1,
    V=[];
    if (j>1), V=[V ; zeros(j-1,1)]; end;
    V=[V ; Bh'];
    if (j<=nrp), V=[V ; zeros(nrp+1-j,1)]; end;
    if (length(V)~=nah+nbh), disp('bezoutb: error V'); end;
    M=[M V];
end;

D=size(M);
if (D(1)~=nah+nbh), disp('bezoutb: error size M row'); end;
if (D(2)~=nah+nbh), disp('bezoutb: error size M column'); end;

% make P column vector
P=P';

global M1;
M1=M;

X= M\P;
% coefficients are real values
X=real(X);

% make X row vector
X=X';

```

## Commande des systèmes par ordinateur

```
Sp=X(1:nsp+1);  
Rp=X(nsp+2:nsp+nrp+2);
```

## 10. BIBLIOGRAPHIE

- [1] I. D. Landau, "*Commande des systèmes*", Edition Lavoisier 2002.
- [2] C. Nouals, "Théorie du Signal", Notes de cours ENSICA
- [3] F. Vincent, "*Traitement Numérique du Signal*", Notes de cours ENSICA.
- [4] D. Arzelier, "*Représentation et analyse des systèmes, notes de cours*", Edition ENSICA. Voir aussi <http://www.laas.fr/~arzelier/>
- [5] J. Bordeneuve-Guibé, "*Commande des systèmes linéaires*", Notes de cours ENSICA.
- [6] Y. Briere, "*Identification*", Notes de cours ENSICA.
- [7] L. Ljung, "*System Identification, Theory for the User*", Prentice Hall 1987.
- [8] Wikipedia , équation de Bezout : [http://en.wikipedia.org/wiki/Bézout's\\_identity](http://en.wikipedia.org/wiki/Bézout's_identity)
- [9] <http://www.mathworks.com>



# 11. INDEX

<b>A</b>	
actionneur intelligent .....	21
amortissement .....	15
ANSI C .....	60
Approximation numérique d'équation différentielle .....	28
ARMAX .....	35
<b>B</b>	
BJ35 .....	
Blocage d'un signal .....	56
bloqueurs d'ordre 0 .....	22
BOZ .....	22
bus.....	21, 61
<b>C</b>	
CAN.....	21
capteur intelligent .....	21
causal .....	6, 9
Causalité .....	10
CNA.....	21
compilateur .....	60
convertisseur analogique / numérique.....	21
convertisseur numérique / analogique.....	21
convertisseurs analogique-numérique .....	61
coprocesseur mathématique .....	60
Correspondances pôles - zéros.....	31
<b>D</b>	
Diagramme de Bode .....	18
Discrétisation de fonction de transfert continues .....	27
division polynomiale.....	6
<b>E</b>	
échelon unitaire.....	11
équation d'état .....	17
Erreur statique.....	42
Estimation de la réponse fréquentielle .....	39
Estimation de la réponse impulsionnelle.....	38
Euler .....	29
<b>F</b>	
facteur d'échelle .....	61
facteur d'oubli .....	36
<i>fenêtre de Hamming</i> .....	39
filtre anti-repliement .....	25
fonction de sensibilité .....	42
fonction de transfert discrète.....	9
<b>G</b>	
gain statique.....	11
gain statique unitaire.....	42
<b>I</b>	
Identification.....	32
Identification non paramétrique .....	38
intercorrélation.....	38
<b>L</b>	
lieu des racines.....	43
lieu d'Evans.....	43
LTI.....	8
<b>M</b>	
marge de gain.....	47
marge de module.....	46
marge de phase .....	47
méthode des moindres carrés récursifs .....	32
Méthode des trapèzes.....	29
modèle ARX .....	32
<b>O</b>	
OE.....	35
one step ahead prediction.....	35
opérateur de retard .....	9
<b>P</b>	
période d'échantillonnage (choix) .....	26
Placement de pôles .....	54
Pôles .....	11
Poursuite .....	56
<i>prédiction</i> .....	32
protocoles.....	61
Pseudo Random Binary Signal .....	34
pulsation .....	15
<b>Q</b>	
quantification .....	23, 24, 60
<b>R</b>	
R,S,T.....	41
rapidité.....	15
rayon de convergence .....	5, 9
Régulation.....	55

Rejet de perturbations .....	42
Rejet d'une perturbation harmonique .....	56
réponse à un échelon.....	11
réponse fréquentielle.....	18
Réponse impulsionnelle.....	10
Réponses indicielle .....	10
retard.....	23
root locus .....	43

### S

signal discret .....	5
smart sensors.....	62
stabilité.....	11
Systèmes du premier ordre.....	14
Systèmes du second ordre.....	15

### T

Théorème de Nyquist.....	46
théorème de Shannon.....	19, 24
théorème du retard .....	6
Transformée en z .....	5
transformée unilatérale .....	5
Tustin.....	29

### V

virgule flottante.....	60
------------------------	----

### Z

zero holder hold .....	23
zéros.....	11
zoh .....	23



# Commande des systèmes par ordinateur

Yves Briere  
ENSICA

Notes de cours  
2007



# TABLE DES MATIERES

Table des matières .....	1
1. Introduction .....	1
2. Rappels et prérequis .....	3
3. Signaux et systèmes discrets .....	5
3.1 Signaux discrets .....	5
3.1.1 Transformée en z .....	5
3.1.2 Signaux types .....	7
3.2 Systèmes discrets .....	9
3.2.1 Fonction de transfert discrète, définition .....	9
3.2.2 Notation "opérateur de retard $q^{-1}$ " .....	10
3.2.3 Réponse impulsionnelle .....	11
3.2.4 Causalité .....	11
3.2.5 Réponses indicielle .....	11
3.2.6 Régime stationnaire (gain statique) .....	12
3.2.7 Pôles, zéros, stabilité .....	12
3.2.8 Systèmes du premier ordre .....	15
3.2.9 Systèmes du second ordre .....	16
3.2.10 Calcul itératif de la réponse d'un système .....	18
3.2.11 Représentation par équation d'état .....	18
3.3 Réponse fréquentielle des systèmes échantillonnés .....	19
3.3.1 Diagramme de Bode d'un système échantillonné .....	19
3.3.2 Où l'on voit apparaître le théorème de Shannon .....	20
4. Systèmes continus échantillonnés .....	23
4.1 Structure des systèmes commandés par ordinateur .....	23
4.2 Échantillonnage des systèmes continus .....	24
4.3 Effets de l'échantillonnage .....	25
4.4 Où l'on revoit le théorème de Shannon .....	26
4.5 Discrétisation de fonction de transfert continues .....	29
4.5.1 Approximation numérique d'équation différentielle .....	30
4.5.2 Superposition des réponses indicielles .....	32
4.5.3 Correspondances pôles - zéros .....	33
4.5.4 Bilan .....	33
5. Identification des systèmes échantillonnés .....	35
5.1 Identification d'un modèle ARX par la méthode des moindres carrés récurrents .....	35
5.2 L'identification paramétrique en général et en pratique .....	37

5.2.1	Choix du signal d'excitation, traitement des mesures .....	37
5.2.2	Choix d'une structure de modèle.....	38
5.2.3	Choix d'un critère à minimiser.....	38
5.2.4	Choix d'un algorithme.....	38
5.2.5	Validation.....	39
5.2.6	Utilisation de logiciels .....	39
5.3	Identification "simple" .....	39
5.4	Identification non paramétrique .....	41
5.4.1	Estimation de la réponse impulsionnelle .....	41
5.4.2	Estimation de la réponse fréquentielle.....	42
6.	Systèmes bouclés .....	45
6.1	Fonction de transfert en boucle fermée .....	45
6.2	Erreur statique .....	46
6.3	Rejet de perturbations.....	46
6.4	Stabilité des systèmes à simple gain de rétroaction.....	47
6.5	Analyse des systèmes bouclés dans le domaine fréquentiel : stabilité.....	49
6.5.1	Théorème de Nyquist dans le cas où le système est stable en boucle ouverte (critère du revers) .....	50
6.5.2	Théorème de Nyquist dans le cas où le système est instable en boucle ouverte .....	50
6.5.3	Interprétations du lieu de Nyquist.....	50
7.	Méthodes de régulation des systèmes échantillonnés .....	53
7.1	Introduction .....	53
7.2	Structure d'un système commandé par ordinateur, hypothèses.....	54
7.2.1	Discretisation, choix de la période d'échantillonnage.....	54
7.2.2	Discretisation d'un correcteur continu .....	55
7.3	Régulateur PID numérique .....	55
7.3.1	Cas 1 .....	55
7.3.2	Cas 2 .....	58
7.4	Placement de pôles .....	59
7.4.1	Choix des pôles en boucle fermée .....	60
7.4.2	Régulation : calcul de R et S.....	60
7.4.3	Poursuite : calcul de T .....	61
7.5	Poursuite et régulation à objectifs indépendants .....	62
8.	Technologie de la commande par ordinateur.....	65
8.1	Calculateurs et interfaces numériques / analogiques.....	65
8.2	Transport de l'information : les bus et les protocoles .....	67
8.3	Capteurs et actionneurs "intelligents", "smart sensors" .....	68
8.4	La philosophie "hardware in the loop" .....	68
8.5	Le prototypage rapide en automatique .....	68
9.	Annexe .....	69
9.1	Annexe 1 : Résolution de l'équation polynomiale de Bezout.....	69
10.	Bibliographie.....	73
11.	Index.....	1





# 1. INTRODUCTION



Figure 1 : commande d'un systèmes de télémédecine par calculateur, vu par Claude Serre





## 2. RAPPELS ET PREREQUIS

Ce polycopié ne comporte pas de rappels sur les systèmes continus. S'adressant aux étudiants en deuxième année de l'ENSICA il suppose connu les enseignements de première année et de seconde année en automatique et traitement du signal.

En première année le cours de théorie du signal [2] a rendu familier les notions de :

- transformée de Fourier, transformée de Laplace
- corrélation, convolution

Le cours de traitement numérique du signal [3] a abordé en plus :

- la discrétisation et la quantification des signaux.
- la transformée en z et la transformée de Fourier discrète.
- les filtres numériques RIF et RII.

En automatique [4] les bases de la représentation et de l'analyse des systèmes continus ont été posées. On ne reviendra pas dans ce cours sur les notions de :

- modèles mathématique des systèmes linéaires à temps continus à temps invariant (LTI) : fonction de transfert, équation d'état, pôles, zéros et dynamique.
- commandabilité et observabilité.
- stabilité des systèmes en boucle ouverte : pôles, critères de Routh-Hurwitz.
- stabilité des systèmes bouclés : critère de Nyquist, critère de Bode.
- lieu des racines.

En deuxième année les notions suivantes sont acquises [5] :

- synthèse des correcteurs par méthodes fréquentielles (avance et retard de phase, PID)
- commande par retour d'état (placement de pôles), commande Linéaire Quadratique (LQG)
- reconstruction de l'état par Observateur, observateur optimal de Kalman.



## 3. SIGNAUX ET SYSTEMES DISCRETS

### 3.1 Signaux discrets

#### 3.1.1 Transformée en z

Pour une approche mathématiquement rigoureuse de représentation des signaux discrets par la transformée en z consulter [3] et sa bibliographie. Dans ce cours on se cantonne à une approche utilitariste des outils et méthodes mathématiques tout en essayant de ne pas contredire fondamentalement la rigueur mathématique.

Un signal discret  $s(k)$  est une séquence de nombres réels appelés échantillons. L'indice  $k$  est généralement associé à un instant d'échantillonnage :  $t = k.T_e$ . La période d'échantillonnage  $T_e$  est supposée constante. Par définition la transformée en z du signal  $s(k)$  est la fonction de la variable  $z$ , notée abusivement  $s(z)$  :

$$s(z) = Z(s(k)) = \sum_{k=0}^{\infty} s(k) \cdot z^{-k} \quad (1)$$

La transformée définie ici est plus précisément appelée transformée unilatérale puisque les termes correspondant à  $k < 0$  ne sont pas pris en compte. Ceci n'a pas d'importance puisque les signaux que nous utilisons sont systématiquement causaux<sup>1</sup> :  $s(k) = 0$  pour  $k < 0$ .

Il y a bien évidemment un lien avec la transformée de Laplace et la transformée de Fourier (voir [3])

La fonction de la variable complexe  $s(z)$  est une série et ne converge pas forcément pour toute valeur de  $z$ . En toute rigueur il faudrait associer le rayon de convergence à chaque fonction en  $z$  :  $\{s(z) \text{ converge pour } |z| > \alpha\}$ .

Les automaticiens pour qui la transformée en z n'est bien souvent qu'une façon bien commode de regrouper des termes de récurrence utilisent souvent la pseudo-variable<sup>2</sup>  $q$ .

A partir d'une transformée en z  $s(z)$  il est possible de retrouver la séquence  $s(k)$  correspondante en appliquant la transformation inverse, dont on ne donnera pas la formule ici. A la place de faire appel à cette formule compliquée il est possible d'utiliser d'autres méthodes pour retrouver la séquence  $s(k)$  lorsque la fonction  $s(z)$  est donnée sous forme de fraction rationnelle :

<sup>1</sup> Pour une justification du terme « causal » voir page 11

<sup>2</sup> Voir page 10

- division polynomiale
- décomposition de la fraction rationnelle en éléments simples puis utilisation de transformations élémentaires telles que celles données dans le Tableau 2

La propriété fondamentale de la transformée en z est qu'elle est linéaire (la transformée en z d'une composition linéaire de signaux est la composition linéaire de la transformée en z de chaque signal). Les autres propriétés fondamentales sont données dans le Tableau 2.

Tableau 1 : propriétés de la transformée en z

1	$s(z) = Z(s(k)) = \sum_{k=0}^{\infty} s(k) \cdot z^{-k}$
2	$Z(\alpha \cdot s_1(k) + \beta \cdot s_2(k)) = \alpha \cdot Z(s_1(k)) + \beta \cdot Z(s_2(k))$
3	$Z(k \cdot s(k)) = -z \cdot \frac{ds(z)}{dz}$
4	$Z(c^k \cdot s(k)) = s\left(\frac{z}{c}\right)$
5	$Z(s(k-1)) = s(k=-1) + z^{-1} \cdot s(z)$
6	$Z(s(k+1)) = z \cdot s(z) - z \cdot s(k=0)$
7	$s(k=0) = \lim_{z \rightarrow \infty} (s(z))$
8	$s(k \rightarrow \infty) = \lim_{z \rightarrow 1} \left( \frac{z-1}{z} \cdot s(z) \right)$

Remarque 1 : la propriété 5 ne prend pas en compte le fait que le signal est causal : en pratique  $s(k=-1) = 0$ .

Remarque 2 : la propriété 5 justifie le fait que le terme  $z^{-1}$  soit appelé "opérateur de retard unitaire", cette propriété est aussi appelée "théorème du retard".

Remarque 3 : les propriétés 7 et 8 sont données sous réserve de convergence.

Transformées inverses

### Exemple 1 : transformée inverse par division polynomiale

On considère un signal  $s(k)$  dont la transformée en z est donnée par la fraction rationnelle :

$$S(z) = \frac{0.1 \cdot z^2 + 0.1 \cdot z}{z^3 - 2.6 \cdot z^2 + 2.2 \cdot z - 0.6} \quad (3.1)$$

L'objectif est d'obtenir  $S(z)$  sous forme polynomiale :

$$S(z) = x_0 + x_1 \cdot z^{-1} + x_2 \cdot z^{-2} + \dots \text{etc.} \tag{3.2}$$

Cette suite est généralement infinie, la réponse temporelle est donnée par la succession des termes  $\{x_i\}$ . En traitant l'équation (3.1) comme une simple division l'algorithme de division classique (celui appris en CM2) permet d'obtenir par récurrence tous les termes de la suite (3.2) :

$0.1 \cdot z^2$	$+ 0.1 \cdot z$				$z^3 - 2.6 \cdot z^2 + 2.2 \cdot z - 0.6$ $0.1 \cdot z^{-1} + 0.36 \cdot z^{-2} + 0.716 \cdot z^{-3}$	
$0.1 \cdot z^2$	$- 0.26 \cdot z$	$+ 0.22$	$- 0.06 \cdot z^{-1}$			
$0$	$0.36 \cdot z$	$- 0.22$	$0.06 \cdot z^{-1}$			
	$0.36 \cdot z$	$- 0.936$	$0.792 \cdot z^{-1}$	$- 0.216 \cdot z^{-2}$		
$0$	$0.716$	$- 0.732 \cdot z^{-1}$	$\dots$	$\dots$		

Les éléments successifs de la suite  $s(k)$  sont donc  $s(k) = \{0, 0.1, 0.36, 0.716, \dots\}$ . Pour obtenir les termes suivants il faudrait continuer la division.

**Exemple 2 : transformée inverse par décomposition en éléments simples**

La deuxième méthode présentée consiste à décomposer la fraction rationnelle (3.1) en éléments simples puis d'utiliser les tables de transformées en  $z$  (Tableau 2). Pour notre exemple cela donne :

$$S(z) = \frac{0.1 \cdot z^2 + 0.1 \cdot z}{z^3 - 2.6 \cdot z^2 + 2.2 \cdot z - 0.6} = \frac{-z}{z-1} + \frac{0.5 \cdot z}{(z-1)^2} + \frac{z}{z-0.6}$$

La réponse temporelle est donc composée des trois éléments :

$$s(k) = -(1 \ 1 \ 1 \ \dots) + 0.5 \cdot (1 \ 2 \ 3 \ \dots) + (1 \ 0.6 \ 0.6^2 \ \dots)$$

$$\text{soit : } s(k) = \begin{cases} s(0) = -1 + 0.5 + 1 \\ s(2) = -1 + 0.5 \cdot 2 + 0.6 \cdot 0.5 \\ s(3) = -1 + 0.5 \cdot 3 + 0.6^2 \\ s(4) = \dots \end{cases}$$

**3.1.2 Signaux types**

Le Tableau 2 donne les transformée en  $z$  de certains signaux fondamentaux. La Figure 2 représente l'allure de quelques signaux types.

Tableau 2 : transformée en z élémentaires

$s(k)$	$S(z)$
signal temporel	Transformée en z
$\delta(k)$ : impulsion unité	1
$u(k)$ : échelon unité	$\frac{z}{z-1}$
$k \cdot u(k)$	$\frac{z}{(z-1)^2}$
$c^k \cdot u(k)$	$\frac{z}{z-c}$
$k \cdot a^k \cdot u(k)$	$\frac{a \cdot z}{(z-a)^2}$
$\sin(\omega \cdot k) \cdot u(k)$	$\frac{z \cdot \sin(\omega)}{z^2 - 2 \cdot z \cdot \sin(\omega) + 1}$
$\cos(\omega \cdot k) \cdot u(k)$	$\frac{z \cdot (z - \cos(\omega))}{z^2 - 2 \cdot z \cdot \sin(\omega) + 1}$
$a^k \cdot \sin(\omega \cdot k) \cdot u(k)$	$\frac{a \cdot z \cdot \sin(\omega)}{z^2 - 2 \cdot a \cdot \cos(\omega) \cdot z + a^2}$
$a^k \cdot \cos(\omega \cdot k) \cdot u(k)$	$\frac{z \cdot (z - a \cdot \cos(\omega))}{z^2 - 2 \cdot a \cdot \cos(\omega) \cdot z + a^2}$

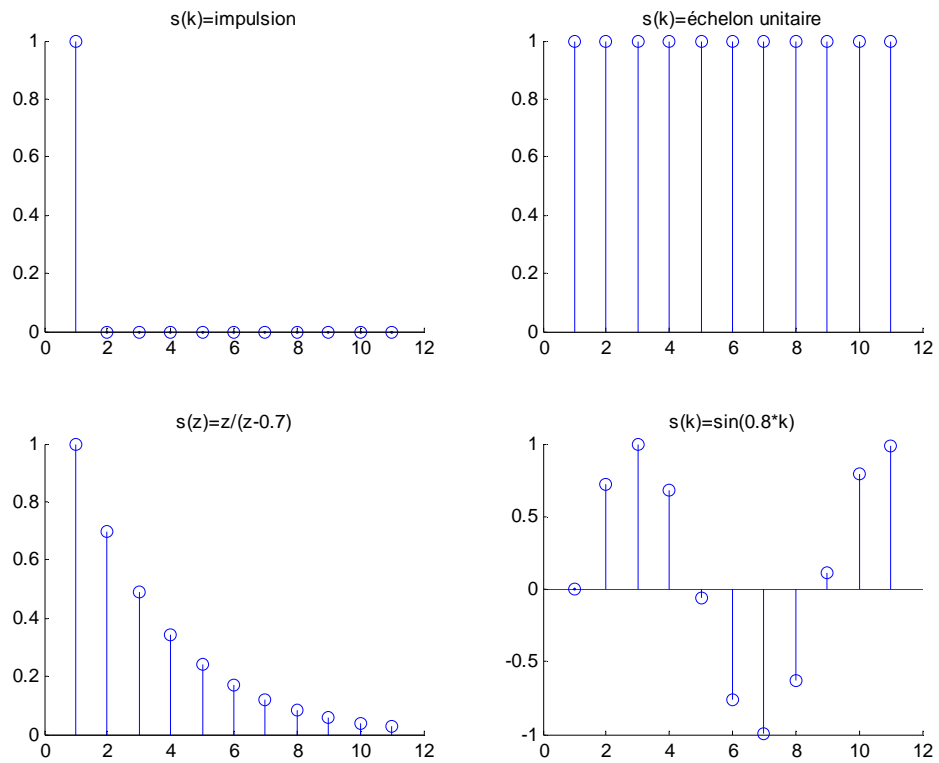


Figure 2 : réponses de quelques signaux types

### 3.2 Systèmes discrets

#### 3.2.1 Fonction de transfert discrète, définition

On considère maintenant un système discret linéaire et invariant dans le temps (LTI) représenté sur la Figure 3.

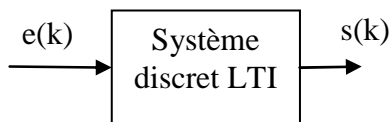


Figure 3 : système discret, notations

La relation entre la séquence d'entrée et la séquence de sortie est linéaire et définie par un ensemble de coefficients constants  $a_{1:n}$  et  $b_{1:m}$  :

$$\begin{aligned} s(k) + a_1 \cdot s(k-1) + a_2 \cdot s(k-2) + \dots + a_n \cdot s(k-n) = \\ b_0 \cdot e(k) + b_1 \cdot e(k-1) + b_2 \cdot e(k-2) + \dots + b_m \cdot e(k-m) \end{aligned} \quad (2)$$

On peut appliquer la transformée en  $z$  à chaque membre de l'équation puis en appliquant en récurrence la propriété 5 (théorème du retard) on obtient la fonction de transfert discrète du système :

$$\frac{s(z)}{e(z)} = F(z) = \frac{b_0 + b_1 \cdot z^{-1} + b_2 \cdot z^{-2} + \dots + b_m \cdot z^{-m}}{1 + a_1 \cdot z^{-1} + a_2 \cdot z^{-2} + \dots + a_n \cdot z^{-n}} \quad (3)$$

Remarque 1 : la fonction de transfert précédente est normalisée en prenant  $a_0 = 1$ .

Remarque 2 : on remarquera que la fonction de transfert est donnée en  $z^{-1}$  plutôt qu'en  $z$ . Cela est justifié par le fait que celle-ci traduit l'équation d'évolution du système physique causal (2).

Le script Matlab permettant de créer un objet Matlab correspondant à une fonction de transfert discrète est donné ci-dessous.

#### Matlab

```
% Création d'un système linéaire discret :
Te=0.1;
sysd = tf([b0 b1 b2],[1 a0 a2 a3],Te,'variable','z^-1');
```

On notera l'utilisation de l'argument 'variable', 'z^-1'.

### 3.2.2 Notation "opérateur de retard $q^{-1}$ "

La fonction de transfert est une fonction de la variable complexe  $z$ . Il faudrait donc en toute rigueur l'associer à son rayon de convergence. En pratique on ne s'embarrasse jamais de préciser cette zone de convergence. C'est pourquoi on trouve souvent la notation utilisant l'opérateur  $q^{-1}$  de décalage à la place de la variable complexe  $z^{-1}$  (4).  $F(q)$  n'est alors plus à proprement parler une fonction mais on lui appliquera les mêmes manipulations algébriques qu'à  $F(z)$ .

$$F(q) = \frac{b_0 + b_1 \cdot q^{-1} + b_2 \cdot q^{-2} + \dots + b_m \cdot q^{-m}}{1 + a_1 \cdot q^{-1} + a_2 \cdot q^{-2} + \dots + a_n \cdot q^{-n}} \quad (4)$$

Attention : selon les auteurs l'opérateur qui remplace  $z^{-1}$  est noté  $q$  (Matlab) ou  $q^{-1}$  (Landau [1]). Dans la suite du document c'est cette dernière convention qui sera appliquée.



L'utilisation de l'opérateur de retard  $q^{-1}$  à la place de la variable complexe  $z^{-1}$  permet de traiter du cas des fonctions de transfert à coefficients non constant, cas non prévu par la transformation en  $z$ .

### 3.2.3 Réponse impulsionnelle

Essayons d'appliquer au système un signal d'entrée particulier : une impulsion unitaire. La transformée en  $z$  du système est obtenue en multipliant la fonction de transfert par la transformée en  $z$  de l'impulsion unitaire : 1. On obtient :

$$s(z) = \frac{b_0 + b_1 \cdot z^{-1} + b_2 \cdot z^{-2} + \dots + b_m \cdot z^{-m}}{1 + a_1 \cdot z^{-1} + a_2 \cdot z^{-2} + \dots + a_n \cdot z^{-n}} \times 1 \quad (5)$$

Cela justifie le fait que la fonction de transfert du système peut aussi s'appeler "réponse impulsionnelle".

#### Matlab

```
% Réponse impulsionnelle d'un système :
[y,t] =impulse(sysd);
```

### 3.2.4 Causalité

Un système est dit causal si les sorties ne dépendent que du passé et non pas du futur. La conséquence est que la réponse impulsionnelle de  $F(z)$  ne peut pas comporter de terme non nuls pour  $k < 0$ . Cela justifie la confusion sémantique entre "système causal" et "signal causal", c'est-à-dire ne comportant pas de termes non nuls pour  $k < 0$ .

Attention : à la différence des systèmes continus un système physique (causal) discret peut très bien être modélisé avec un degré  $m$  supérieur au degré  $n$  : le système peut avoir plus de zéros que de pôles.

La transformée en  $z$  représentant des signaux échantillonnés elle peut tout aussi bien représenter les mesures prises à différents endroits d'une poutre. Le temps étant remplacé ici par une abscisse la notion de causalité ne s'applique pas ! Attention : beaucoup des propriétés de systèmes échantillonnés que l'on étudie sont liés à la causalité, prudence donc si l'on applique ces méthodes à d'autres domaines !

### 3.2.5 Réponses indicielle

Pour caractériser des systèmes on s'intéresse souvent à leur réponse impulsionnelle ou, plus souvent, à leur réponse indicielle. Pour calculer la réponse

à un échelon d'un système il faut donc multiplier la fonction de transfert par la transformée en z d'un échelon unitaire :

$$s(z) = \frac{b_0 + b_1 \cdot z^{-1} + b_2 \cdot z^{-2} + \dots + b_m \cdot z^{-m}}{1 + a_1 \cdot z^{-1} + a_2 \cdot z^{-2} + \dots + a_n \cdot z^{-n}} \times \frac{1}{1 - z^{-1}} \quad (6)$$

Il faudrait ensuite calculer la transformée en z inverse de s(z). En pratique, si l'on doit faire le calcul à la main on décompose en éléments simples puis on utilise des tables. Avec Matlab c'est plus simple. Voici par exemple le script pour obtenir les 20 premiers échantillons de la réponse indicielle d'un système :

**Matlab**

```
% Réponse indicielle d'un système :
[y,t] =step(sysd,20);
```

### 3.2.6 Régime stationnaire (gain statique)

Le gain statique est obtenu en appliquant une entrée indicielle et en calculant la valeur du régime permanent. En appliquant le théorème de la valeur finale il vient :

$$F_0 = \lim_{z \rightarrow 1} \left( \frac{z-1}{z} \cdot \frac{b_0 + b_1 \cdot z^{-1} + b_2 \cdot z^{-2} + \dots + b_m \cdot z^{-m}}{1 + a_1 \cdot z^{-1} + a_2 \cdot z^{-2} + \dots + a_n \cdot z^{-n}} \times \frac{1}{1 - z^{-1}} \right) \quad (7)$$

Soit :

$$F_0 = \frac{\sum_{i=0}^{i=m} b_i}{1 + \sum_{i=1}^{i=n} a_i} \quad (8)$$

Le gain statique est obtenu donc simplement en faisant le quotient de la somme des coefficients.

### 3.2.7 Pôles, zéros, stabilité

Comme pour les systèmes continus la dynamique et la stabilité des systèmes dépend des propriétés du dénominateur de la fonction de transfert.

$$F(z) = \frac{e_0 \cdot (1 + e_1 \cdot z^{-1}) \cdot (1 + e_2 \cdot z^{-1}) \cdots (1 + e_m \cdot z^{-1})}{(1 + c_1 \cdot z^{-1}) \cdot (1 + c_2 \cdot z^{-1}) \cdots (1 + c_n \cdot z^{-1})} \quad (9)$$

Les coefficients  $c_i$  sont les pôles de la fonction de transfert, les coefficients  $e_i$  sont les zéros. La fonction de transfert étant au départ composée de coefficients purement réels les coefficients  $c_i$  sont soit réels soit imaginaires par paires conjuguées.

Une fonction de transfert est donc aussi bien définie par la donnée de ses pôles et de ses zéros. Le script Matlab pour créer un système ainsi est donné ci-dessous :

#### Matlab

```
% Système discret défini par ses pôles et zéros :
Te=0.1;
sysd=zpk([c1 c2 c3],[e1 e2],e0,Te);
```

Si l'on décompose la fonction de transfert en éléments simples on obtient une somme d'éléments du type :

$$F(z) = \frac{d_1}{1 + c_1 \cdot z^{-1}} + \frac{d_2}{1 + c_2 \cdot z^{-1}} + \cdots + \frac{d_n}{1 + c_n \cdot z^{-1}} \quad (10)$$

Un système linéaire temps invariant est dit stable si sa réponse impulsionnelle tend asymptotiquement vers 0. Compte tenu de (10) cela nécessite que chaque pôle  $c_i$  soit de module inférieur à 1.

**Propriété** : un système discret LTI est stable si et seulement si tous ses pôles ont leur modules inférieur ou égal à 1.

#### Critère algébrique attestant de la stabilité : critère de Routh

On rappelle que le critère de Routh permet de prédire si les racines d'un polynôme sont toutes à partie réelle négative, donc stables. Énonçons le théorème de Routh dans le cas d'un polynôme  $P'(w)$  :

$$P'(w) = a_0 + a_1 \cdot w + a_2 \cdot w^2 + \cdots + a_n \cdot w^n \quad (11)$$

Dans un premier temps on construit le tableau :

$$\begin{array}{cccccc|c} \beta_{n,0} & \beta_{n,1} & \beta_{n,2} & \beta_{n,q-1} & \beta_{n,q} & 0 & \\ \beta_{n-1,0} & \beta_{n-1,1} & \beta_{n-1,2} & \beta_{n-1,q-1} & \beta_{n-1,q-1} & 0 & \\ \beta_{n-2,0} & \beta_{n-2,1} & \beta_{n-2,2} & \beta_{n-2,q-1} & \beta_{n-2,q-1} & & \\ & \beta_{2,0} & \beta_{2,1} & 0 & & & \\ & \beta_{1,0} & 0 & & & & \\ & \beta_{0,0} & 0 & & & & \end{array} \quad (12)$$

Puis on vérifie que tous les premiers éléments de la première colonne ont le même signe : cela garantit que tous les pôles de  $P'(w)$  ont leur partie réelle négative.

Les éléments du tableau sont construits par récurrence ligne par ligne en appliquant les formules suivantes :

Pour  $i = n$  et  $i = n - 1$  :

$$\begin{cases} \beta_{n,j} = a_{n-2j} \\ \beta_{n-1,j} = a_{n-1-2j} \end{cases} \quad (13)$$

Et pour  $i < n-1$  :

$$\beta_{i,j} = \frac{\beta_{i+1,0} \cdot \beta_{i+2,j+1} - \beta_{i+1,j+1} \cdot \beta_{i+2,0}}{\beta_{i+1,0}} \quad (14)$$

Dans le cas qui nous intéresse, un système échantillonné, on cherche plutôt à vérifier que les pôles du polynôme  $P(z)$  sont de module inférieur à un. L'idée est d'effectuer un changement de variable  $z \rightarrow w$  pour pouvoir appliquer le critère de Routh.

$$P'(w) = (1-w)^n \cdot P\left(\frac{1+w}{1-w}\right) \quad (15)$$

Si l'on effectue la transformation (15) alors le fait que  $P'(w)$  ait toutes ses racines à partie réelle négative garantit que  $P(z)$  a toutes ses racines dans le cercle unité. Pour vérifier la stabilité de  $P(z)$  il suffit donc d'effectuer le changement de variable (15) puis d'appliquer le critère de Routh conventionnel (12 13 14).

### Critère algébrique attestant de la stabilité : critère de Jury

Le critère de Jury s'apparente au critère de Routh. Nous le donnons pour un polynôme de degré 2, 3 et 4 du type :

$$P(z) = a_0 + a_1 \cdot z + a_2 \cdot z^2 + \dots + a_n \cdot z^n \quad (16)$$

Les conditions de stabilité sont données ci-dessous (ces conditions sont données dans le cas  $a_n > 0$ , il suffit de multiplier tous les termes par  $-1$  si cette condition n'est pas remplie par  $P(z)$ ) :

$$\begin{aligned}
 n = 2 & \begin{cases} a_0 + a_1 + a_2 > 0 \\ a_0 - a_1 + a_2 > 0 \\ -a_0 + a_2 > 0 \end{cases} \\
 n = 3 & \begin{cases} a_0 + a_1 + a_2 + a_3 > 0 \\ -a_0 + a_1 - a_2 + a_3 > 0 \\ -|a_0| + a_3 > 0 \\ a_0 a_2 - a_1 a_3 - a_0^2 + a_3^2 > 0 \end{cases} \\
 n = 4 & \begin{cases} a_0 + a_1 + a_2 + a_3 + a_4 > 0 \\ a_0 - a_1 + a_2 - a_3 + a_4 > 0 \\ a_4^2 - a_0^2 - |a_0 a_3 - a_1 a_4| > 0 \\ (a_0 - a_4)^2 (a_0 - a_2 + a_4) + (a_1 - a_3)(a_0 a_3 - a_1 a_4) > 0 \end{cases}
 \end{aligned} \quad (17)$$

### 3.2.8 Systèmes du premier ordre

On donne dans la Figure 4 les réponses indicielles de systèmes du premier ordre ainsi que la localisation des pôles dans le plan complexe. Le cercle de rayon unité représente la zone dans laquelle les pôles doivent se situer pour assurer la stabilité. On remarque sur les deux premières figures l'influence de la place du zéro par rapport au point  $(1,0)$  pour la rapidité.

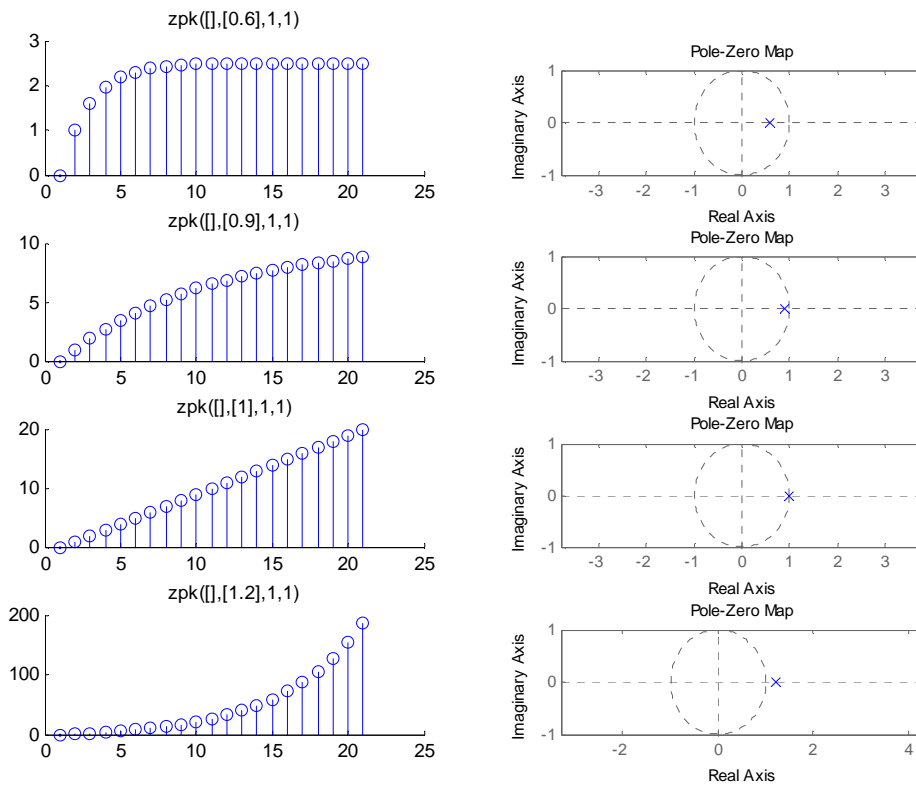


Figure 4 réponses indicielles de systèmes du premier ordre

### 3.2.9 Systèmes du second ordre

On donne dans la Figure 6 les réponses indicielles de systèmes du second ordre ainsi que la localisation des pôles dans le plan complexe. La rapidité et l'amortissement dépend de la position des pôles dans le cercle unité. La Figure 5 représente les courbes iso-amortissement et iso-pulsation des systèmes du second ordre.

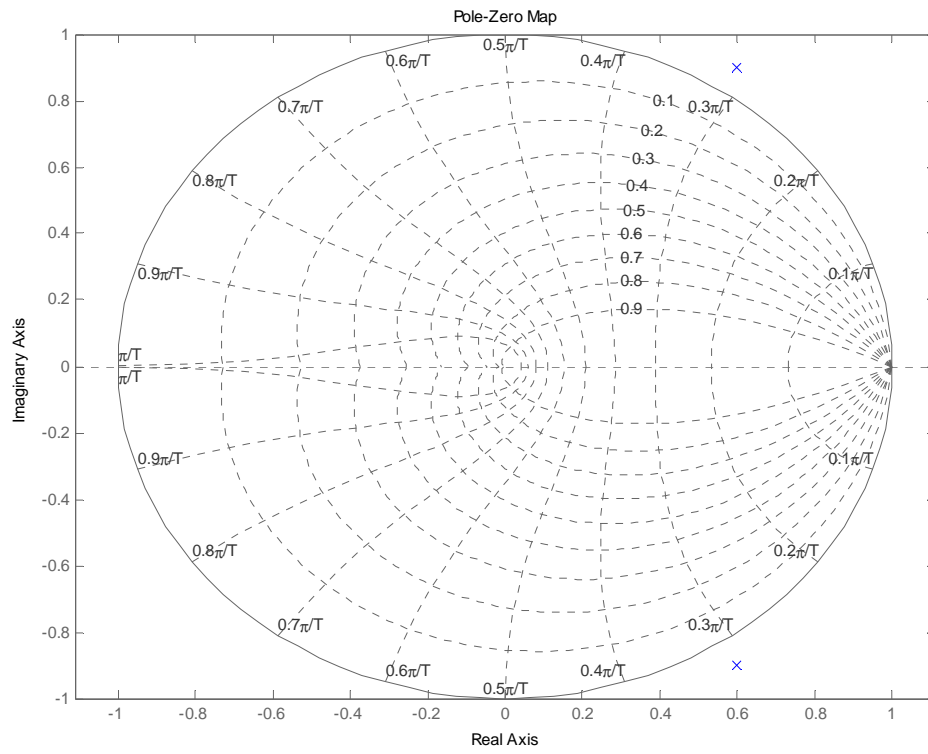


Figure 5 : courbes iso-amortissement et iso-pulsation des systèmes discrets du second ordre

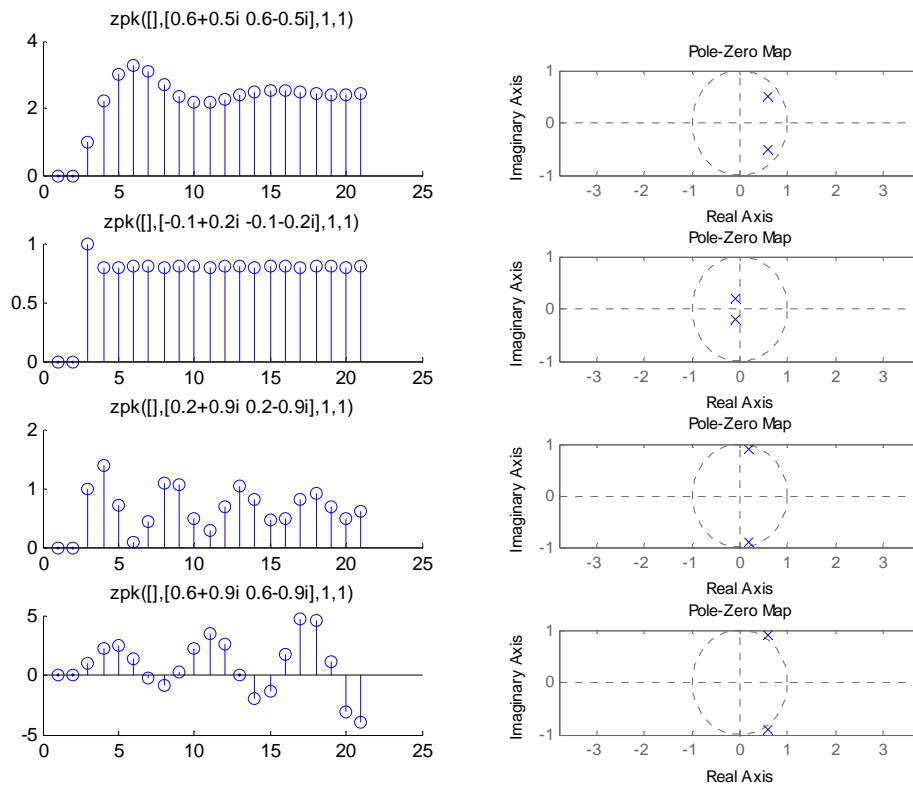


Figure 6 : réponses de systèmes du second ordre

### 3.2.10 Calcul itératif de la réponse d'un système

Si l'on dispose de la fonction de transfert d'un système et que l'on connaît les valeurs successives  $e(k)$  du signal d'entrée il est facile de calculer itérativement la sortie  $y(k)$  du système par l'équation de récurrence (2). Il faut initialiser par les  $n$  valeurs initiales de  $s(k)$ , prises généralement égales à 0.

### 3.2.11 Représentation par équation d'état

Un système discret peut aussi être représenté par un ensemble d'équations d'état comme dans le domaine continu (52). Cette représentation est naturellement adaptée aux systèmes multi-entrées multi-sorties. On définit alors un état interne  $x(k)$  ainsi qu'une fonction d'évolution  $f$  et une équation de mesure  $h$ .

$$\begin{cases} x(k+1) = f(x(k), u(k)) \\ s(k) = h(x(k), u(k)) \end{cases} \quad (18)$$



### 3.3 Réponse fréquentielle des systèmes échantillonnés

#### 3.3.1 Diagramme de Bode d'un système échantillonné

Lorsqu'on étudie la réponse fréquentielle des systèmes continus on considère des entrées particulières sinusoïdales :

$$\begin{cases} e(t) = \cos(2 \cdot \pi \cdot f \cdot t) + j \cdot \sin(2 \cdot \pi \cdot f \cdot t) = e^{j \cdot 2 \cdot \pi \cdot f \cdot t} \\ f \in [0 \quad \infty] \end{cases} \quad (19)$$

De même, pour des systèmes échantillonnés on considèrera des entrées sinusoïdales échantillonnées :

$$\begin{cases} e(k) = \cos(2 \cdot \pi \cdot f \cdot k) + j \cdot \sin(2 \cdot \pi \cdot f \cdot k) = e^{j \cdot 2 \cdot \pi \cdot f \cdot k} \\ f \in [0 \quad 1] \end{cases} \quad (20)$$

La sortie est aussi un signal échantillonné sinusoïdal de gain  $A(f)$  et de phase  $\Phi(f)$  variables et dépendant de  $f$ ; la fréquence réduite.

$$\begin{cases} s(k) = A(f) \cdot e^{j \cdot (2 \cdot \pi \cdot f \cdot k + \Phi(f))} \\ f \in [0 \quad 1] \end{cases} \quad (21)$$

Le nombre complexe, noté abusivement  $F(f)$ , d'amplitude  $A(f)$  et de phase  $\Phi(f)$  est obtenu à partir de  $F(z)$  en remplaçant  $z$  par  $e^{j \cdot 2 \cdot \pi \cdot f}$  :

$$\begin{cases} F(f) = F(z = e^{j \cdot 2 \cdot \pi \cdot f}) \\ f \in [0 \quad 1] \end{cases} \quad (22)$$

Le diagramme représentant le gain et la phase en fonction de la fréquence réduite s'appelle aussi un diagramme de Bode. L'échelle des fréquences est généralement linéaire de 0 à 0.5 (puisque le diagramme est nécessairement symétrique). La Figure 7 représente par exemple le diagramme de Bode d'un système du second ordre.

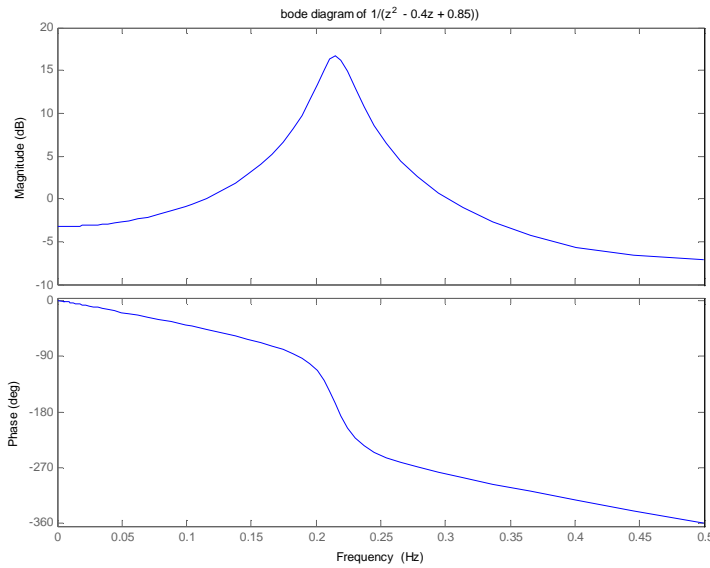


Figure 7 : exemple du diagramme de Bode d'un système discret du second ordre

### 3.3.2 Où l'on voit apparaître le théorème de Shannon

Le changement de variable  $z = e^{j2\pi f}$  n'est pas bijectif puisque  $e^{j2\pi f} = e^{j2\pi(f+k)}$  avec  $k$  entier. Si l'on fait le lien entre domaine continu et domaine fréquentiel on peut remarquer que le changement de variable  $z = e^{j2\pi f}$  fait correspondre une zone du plan complexe continu à une zone du plan complexe discret. La transformation est bijective si l'on se contente de la zone du plan complexe continu contenu entre les droites d'ordonnée 0 et 1.

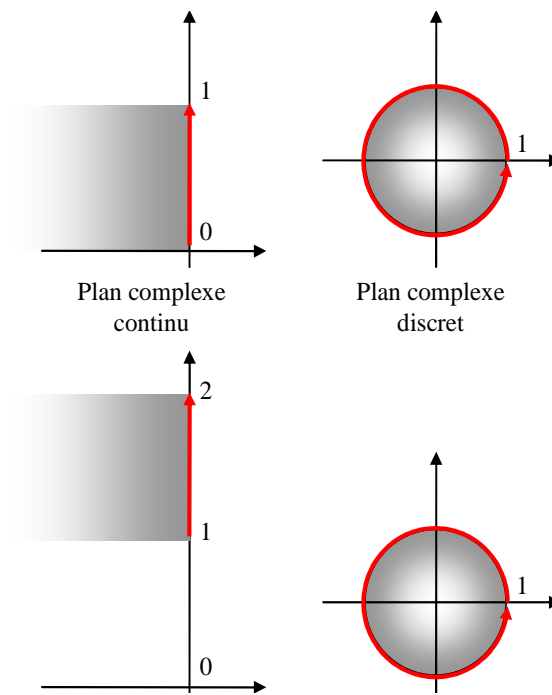


Figure 8 : correspondance entre plan complexe continu et plan complexe discret.



## 4. SYSTEMES CONTINUS ECHANTILLONNES

### 4.1 Structure des systèmes commandés par ordinateur

Les systèmes commandés par ordinateurs sont presque toujours des systèmes physiques continus par nature. Le ordinateur traite par contre les entrées et les sorties de manière discrète avec une période d'échantillonnage  $T_e$ . Les éléments importants, du point de vue de l'automaticien sont donc :

- le convertisseur analogique / numérique CAN ou du moins le système qui fournit séquentiellement les mesures au ordinateur à la période  $T_e$ .
- le ordinateur lui-même. Il doit effectuer l'acquisition des mesures, le traitement des informations, l'envoi des consignes dans un temps inférieur à  $T_e$ .
- le convertisseur numérique / analogique CNA.
- le système physique.

On suppose, pour faire simple et pour pouvoir utiliser la panoplie des outils mathématiques, que les systèmes sont échantillonnés à la même période  $T_e$  et au même moment.

La Figure 9 montre le système physique commandé par le ordinateur, dans sa version simpliste.

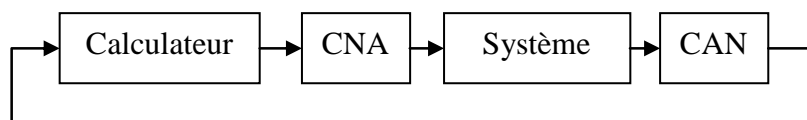


Figure 9 : système commandé par ordinateur

Au XXIème siècle les choses sont quand même parfois un petit peu plus compliquées :

- le ordinateur fournit bien souvent un signal de commande numérique (transporté par bus informatique USB, RS232, CAN, Ethernet, etc...) directement interprété par l'actionneur. On parle "d'actionneur intelligent".
- de même le "capteur intelligent" fournit aussi bien souvent la mesure sous forme d'un signal numérique.
- Les signaux sont transportés par des bus informatiques qui n'ont pas que ça à faire : il peut y avoir du retard.
- de même les capteurs et actionneurs sont rarement tous seuls, ce qui se traduit par des périodes d'échantillonnage différentes, et parfois non synchrones...

- enfin le ordinateur peut aussi avoir d'autres choses à faire...

Ceci étant dit nous allons pour le moment nous restreindre au cas représenté sur la Figure 9 avec des instants d'échantillonnage synchrones et régulièrement espacés d'un temps  $T_e$ .

## 4.2 Echantillonnage des systèmes continus

On dispose généralement de la fonction de transfert continue (notée en utilisant la variable complexe  $s$  ou  $p$ )  $F(s)$  du système continu. Le problème est d'obtenir la fonction de transfert discrète du système échantillonné de manière à pouvoir synthétiser un correcteur. Le système continu est associé au convertisseur numérique – analogique qu'il faut prendre en compte.

Les convertisseurs numérique analogiques peuvent généralement se représenter par des "échantillonneurs – bloqueurs d'ordre 0" ou BOZ. Cela signifie que la commande est maintenue à une valeur constante pendant toute la durée du temps d'échantillonnage.

Si l'on applique une commande discrète impulsionnelle à un BOZ il va maintenir la commande 1 pendant la durée  $T_e$  puis revenir à zéro (Figure 10).

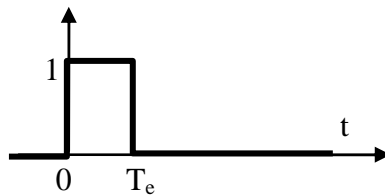


Figure 10 : réponse impulsionnelle d'un bloqueur d'ordre 0

La réponse impulsionnelle  $BOZ(t)$  est donc composée d'un échelon unitaire suivi d'un échelon unitaire de signe opposé et décalé d'un temps  $T_e$ . La transformée de Laplace de la réponse impulsionnelle du bloqueur est donnée par la formule (23) :

$$BOZ(s) = \frac{1}{s} - \frac{e^{-jT_e}}{s} \quad (23)$$

Du point de vue du ordinateur il faut donc prendre en compte le système continu de fonction de transfert  $F_{BOZ}(s)$  que l'on devra échantillonner :

$$F_{BOZ}(s) = \frac{1 - e^{-jT_e}}{s} \cdot F(s) \quad (24)$$

$$F_{\text{BOZ}}(z) = Z^{-1} \left( \frac{1 - e^{-jT_e}}{s} \cdot F(s) \right) \quad (25)$$

En pratique l'obtention de la transformée en z d'un système échantillonné par bloqueur d'ordre 0 se fait à l'aide de tables. On donne quelques exemples fondamentaux dans le Tableau 3 ci-dessous.

Tableau 3 : transformée en z de systèmes continus échantillonnés

F(s)	F <sub>BOZ</sub> (z)
1	1
$\frac{1}{s}$	$\frac{T_e \cdot z^{-1}}{1 - z^{-1}}$
$\frac{1}{1 + T \cdot s}$	$\begin{cases} \frac{b_1 \cdot z^{-1}}{1 + a_1 \cdot z^{-1}} \\ b_1 = 1 - e^{-T_e/T} \quad a_1 = -e^{-T_e/T} \end{cases}$
$\frac{e^{-Ls}}{1 + T \cdot s}$	$\begin{cases} \frac{b_1 \cdot z^{-1} + b_2 \cdot z^{-2}}{1 + a_1 \cdot z^{-1}} \quad a_1 = -e^{-T_e/T} \\ b_1 = 1 - e^{-(L-T_e)/T} \quad b_2 = e^{-T_e/T} (e^{L/T} - 1) \end{cases}$
$L < T_e$	

Si l'on dispose de Matlab c'est bien sûr encore plus simple. On utilise la fonction c2d avec l'argument 'zoh' pour "zero holder hold" (bloqueur d'ordre zéro).

#### Matlab

```
% Définition d'un système continu :
sysc=tf(1,[1 1]);
% Obtention de la transformée en z du système échantillonné à Te
Te=0.1;
sysd=c2d(sysc,Te,'zoh');
% Notation en z^-1
sysd.variable='z'
```

### 4.3 Effets de l'échantillonnage

Le convertisseur numérique a deux effets (néfastes) importants : l'effet de retard et l'effet de quantification.

L'effet de retard se voit dans (24) : on peut dire en première approximation que l'échantillonnage se traduit par un retard pur de  $T_e/2$ . Cela a donc, si ce temps

d'échantillonnage est non négligeable par rapport aux constantes de temps du système, un effet déstabilisant.

Pour illustrer l'effet de quantification supposons par exemple que l'on utilise un convertisseur analogique/numérique 8 bits 0-10V. Le plus petit incrément que le convertisseur peut distinguer est de  $10V/2^8 = 40mV$ . Si le signal utile varie physiquement entre 5V et 5,4V on n'aura que dix incréments utilisables... Ce phénomène n'entre pas dans la panoplie des phénomènes modélisables par la transformée en z. En pratique il est donc souvent nécessaire d'étudier le système et de synthétiser la commande numérique en négligeant ce phénomène puis de vérifier a posteriori qu'il n'a pas un effet trop néfaste.

#### 4.4 Où l'on revoit le théorème de Shannon

Le théorème de Shannon est bien connu en ce qui concerne l'échantillonnage de signaux discrets : on sait que le signal ne doit pas comporter de composante de fréquence supérieure à  $f_e/2$  si l'on ne veut pas observer de repliement. On doit aussi prendre des précautions si l'on veut travailler sur la fonction de transfert discrète correspondant à un système continu échantillonné.

On va traiter un exemple concret pour mettre en évidence le problème de repliement dans le cas de la commande des systèmes continus. On s'intéresse à deux systèmes continus du second ordre ayant des pôles de même partie réelle mais de parties imaginaires séparées de  $2\pi T_e$ . Le script Matlab ci-dessous montre la création des deux systèmes :

##### Matlab

```
% Définition de deux systèmes continus :
sys1=zpk([],[-1-2i -1+2i],2^2+1);
sys2=zpk([],[-1-(2+2*pi)*i -1+(2+2*pi)*i],1+(2+2*pi)^2);
```

On voit sur la Figure 11 que la réponse indicielle des deux systèmes est bien différente.



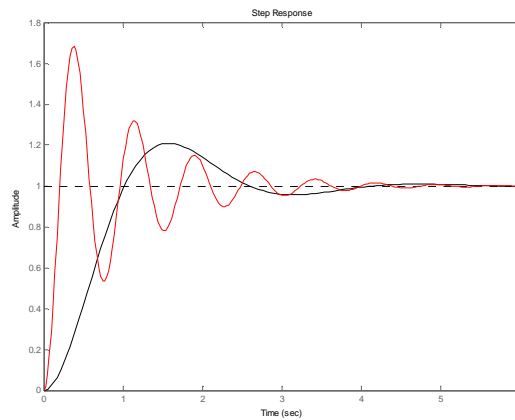


Figure 11 : réponse indicielle de deux systèmes du second ordre

Si l'on calcule la fonction de transfert échantillonnée les deux systèmes à la période de 1 seconde on obtient deux réponses indicielles très proches (représentés par des  $\circ$  sur la Figure 12). Conclusion : la période d'échantillonnage était mal choisie

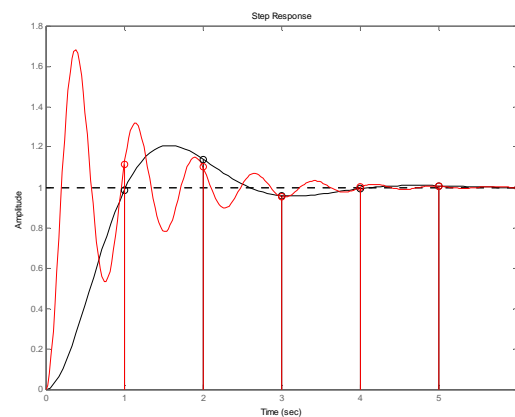


Figure 12 : réponse indicielle de deux systèmes du second ordre échantillonnés à 1s

En pratique la commande échantillonnée de systèmes continus doit être associée à certaines précautions :

- le système ne doit pas comporter de dynamiques propres situées dans les fréquences supérieures à  $f_c/2$  (exemple illustré dans ce chapitre).
- les signaux échantillonnés doivent être "nettoyés" des fréquences supérieures à  $f_c/2$  avec un filtre anti-repliement. (A moins que le signal soit suffisamment propre et ne comporte pas de composantes significatives au-delà de ces fréquences).

### Choix de la période d'échantillonnage

Le choix de la période d'échantillonnage est très important et dépend de la dynamique des signaux. Attention, les systèmes commandés en boucle fermée ont souvent une dynamique plus rapide que le système en boucle ouverte : il faut donc prendre en compte la dynamique en boucle fermée pour le choix de la fréquence d'échantillonnage.

En pratique on préconise de prendre une fréquence 5 à 25 fois supérieure à la bande passante souhaitée en boucle fermée.

On peut être tenté de prendre une fréquence d'échantillonnage très élevée si la technologie le permet (ordinateur et convertisseurs suffisamment rapide). Ceci peut amener rapidement à des problèmes numériques. Imaginons par exemple le cas d'un système ayant une dynamique très lente (quelques secondes) et un système de mesure codant les mesures sur 10 bits (cas classique). La Figure 13 illustre cet exemple. Le signal varie peu entre deux instants d'échantillonnage et les variations du signal sont faibles par rapport à la valeur de l'incrément numérique (figure du haut). Le calcul de la simple dérivée numérique se traduit donc par une succession de  $-1$   $0$  et  $+1$ .

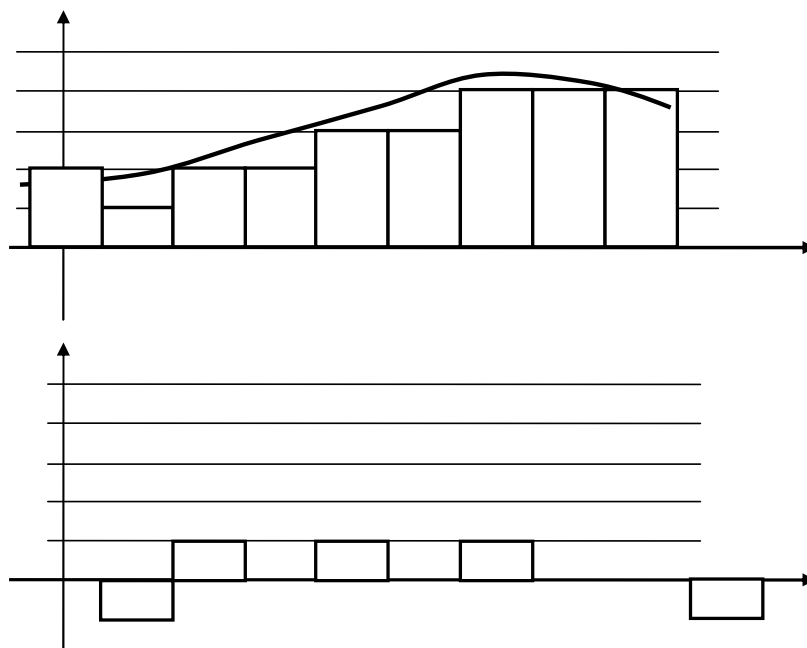


Figure 13 : période d'échantillonnage trop faible. En haut : signal échantillonné et discrétisé ; en bas : calcul de la dérivée numérique.

### Filtre anti-repliement

En toute rigueur chaque fois qu'un signal est échantillonné il faudrait faire précéder le convertisseur numérique d'un filtre analogique anti-repliement éliminant physiquement toutes les composantes du signal de fréquence supérieure à  $f_e/2$ .

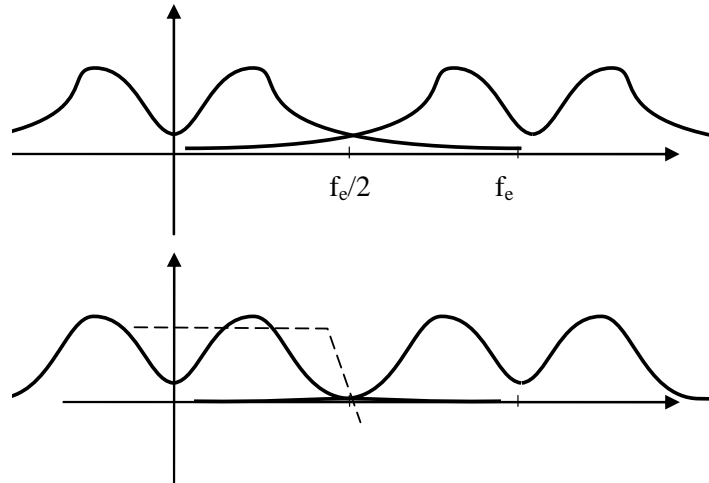


Figure 14 : Effet du filtre anti-repliement

Bien souvent la période d'échantillonnage de l'asservissement est relativement lente (quelques secondes) et il est alors impossible de réaliser un filtre de constante de temps aussi faible. On procède alors à un sur-échantillonnage : le signal est filtré, échantillonné à haute fréquence  $T'_e$  puis filtré numériquement et sous-échantillonné à la période  $T_e$  beaucoup plus grande que  $T'_e$  comme illustré sur la

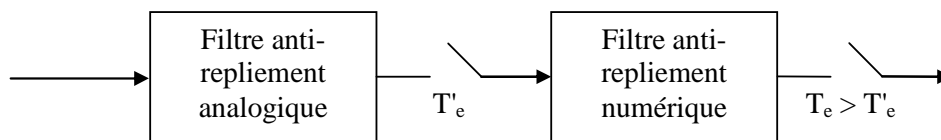


Figure 15 : Sur échantillonnage

Un simple filtre numérique de type moyenne mobile est souvent suffisant.

### 4.5 Discrétisation de fonction de transfert continues

Une manière classique pour réaliser un contrôle par ordinateur est de commencer par synthétiser un correcteur continu avec les outils de l'automatique classique puis de trouver un correcteur discret qui "approche" convenablement le correcteur continu.

#### 4.5.1 Approximation numérique d'équation différentielle

Supposons que le correcteur continu que l'on a synthétisé soit du type PD :

$$\text{PID}(s) = \frac{e(s)}{\varepsilon(s)} = k_p + k_D \cdot s \quad (26)$$

Il s'agit de calculer les commandes  $e(k)$  successives issues du correcteur en fonction des erreurs  $\varepsilon(k)$ . Une approche classique et intuitive est d'approximer la dérivée par une différence :

$$e(k) = k_p \cdot \varepsilon(k) + k_D \cdot \frac{\varepsilon(k) - \varepsilon(k-1)}{T_e} \quad (27)$$

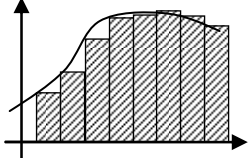
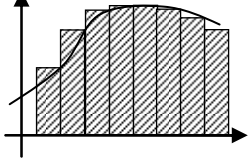
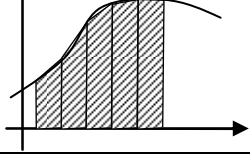
Si on calcule la transformée en  $z$  de  $e(k)$  on obtient :

$$e(z) = k_p \cdot \varepsilon(z) + k_D \cdot \frac{\varepsilon(z) - z^{-1}\varepsilon(z)}{T_e} = \left( k_p + k_D \cdot \frac{1 - z^{-1}}{T_e} \right) \cdot \varepsilon(z) \quad (28)$$

On retrouve la fonction de transfert PD(s) dans laquelle on a effectué le changement de variable  $s = \frac{1-z^{-1}}{T_e}$

De manière plus générale, l'approximation de fonction de transfert revient à effectuer l'approximation numérique d'intégration. Selon la méthode d'intégration choisie on effectue des changements de variable différents.

Tableau 4 : méthodes d'intégration

Méthode	Changement de variable	
Euler progressif (Euler's forward method)	$s = \frac{1 - z^{-1}}{z^{-1} \cdot T_e}$	
Euler rétrograde (Euler's backward method)	$s = \frac{1 - z^{-1}}{T_e}$	
Méthode des trapèzes, ou bilinéaire, ou de Tustin	$s = \frac{2}{T_e} \cdot \frac{1 - z^{-1}}{1 + z^{-1}}$	

En pratique le changement de variable  $s = \frac{1 - z^{-1}}{z^{-1} \cdot T_e}$  n'est pas sans risque. On donne sur la Figure 16 la correspondance entre la zone stable du plan complexe en  $s$  et la zone correspondante après la transformation (Euler progressif, Euler rétrograde, bilinéaire). On voit notamment qu'il est possible, partant d'un système continu stable, d'obtenir des pôles dans le domaine discret en dehors du cercle de stabilité si la transformation appliquée est celle d'Euler progressif.

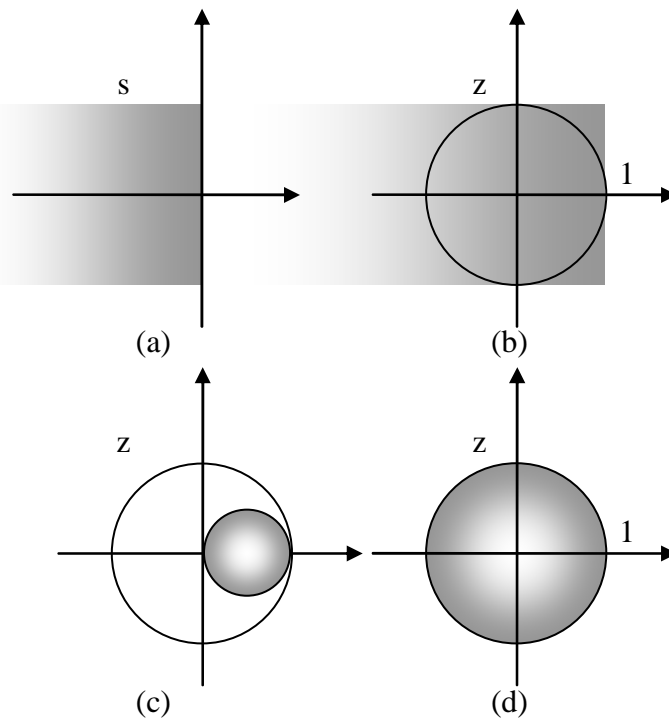


Figure 16 : correspondance des plans complexes. (a) : zone de stabilité dans le plan complexe continu en  $z$  ; (b) zone correspondante avec l'approximation d'Euler progressif ; (c) zone correspondante avec l'approximation d'Euler rétrograde ; (d) zone correspondante avec l'approximation bilinéaire.

La fonction `c2d` de Matlab permet d'obtenir l'équivalent discret d'une fonction de transfert continue avec la méthode d'intégration comme argument :

```

Matlab
sysc=tf([1 0],[1 2 5]);
Te=0.1;
% Méthode d'euler :
sysd1=c2d(sysc,Te,'zoh');
% Méthode des trapèzes :
sysd2=c2d(sysc,Te,'foh');
% Méthode bilinéaire :
sysd3=c2d(sysc,Te,'tustin');
    
```

#### 4.5.2 Superposition des réponses indicielles

Une autre méthode pour approximer une fonction de transfert discrète est de s'assurer que la réponse indicielle de sa version discrétisée correspond exactement lors des temps d'échantillonnage.

### 4.5.3 Correspondances pôles - zéros

La dernière méthode est de partir d'une décomposition de la fonction de transfert continue en pôles et zéros et d'effectuer la conversion pôles à pôles et zéros par zéros. La correspondance, selon que le pôle est réel ou complexe est donnée ci-dessous :

$$\begin{aligned} (s + a) &\rightarrow (z - e^{-a \cdot T_c}) \\ (s + a)^2 + b^2 &\rightarrow z^2 - 2 \cdot e^{-a \cdot T_c} \cdot \cos(b \cdot T_c) \cdot z + e^{-2 \cdot a \cdot T_c} \end{aligned} \quad (29)$$

### 4.5.4 Bilan

Aucune méthode n'est parfaite et on n'aura jamais en même temps :

- même réponse indicielle
- même réponse impulsionnelle
- approximation (Euler, Tustin, etc...) de l'équation différentielle
- correspondance des pôles et des zéros
- correspondance des réponses fréquentielles

Ceci étant dit le choix de la période d'échantillonnage est primordial. Si la fréquence d'échantillonnage est beaucoup plus faible que la dynamique de la fonction de transfert à approximer la différence entre chaque méthode sera minimale.





## 5. IDENTIFICATION DES SYSTEMES ECHANTILLONES

Dans le présent chapitre on donne les bases des méthodes d'identification. Pour une approche complète, voir le cours de troisième année "Identification avion" [6] et les ouvrages [1] et [7].

Même si l'on dispose d'un modèle théorique du processus que l'on souhaite commander il est presque toujours nécessaire de l'affiner en procédant à des essais sur le système réel. Bien souvent le seul modèle dont l'on dispose est ce modèle issu de l'expérimentation. Différentes méthodes s'appliquent selon l'objectif recherché et le niveau de précision exigé. On distinguera :

- les modèles paramétriques
- les modèles simples
- les modèles non paramétriques

### 5.1 Identification d'un modèle ARX par la méthode des moindres carrés récursifs

A partir de cet exemple, le plus simple qui soit, on pose les bases de toutes les méthodes d'identification.

On considère un système d'entrée  $u(t)$  et de sortie  $y(t)$  représenté par l'équation de récurrence suivante :

$$y(t) + a_1 \cdot y(t-1) + a_2 \cdot y(t-2) + \dots + a_n \cdot y(t-n) = b_0 \cdot u(t) + b_1 \cdot u(t-1) + b_2 \cdot u(t-2) + \dots + b_m \cdot u(t-m) \quad (30)$$

L'objectif est, à partir d'essais et de mesures, d'identifier les meilleures valeurs possibles pour les coefficients  $a_i$  et  $b_i$ .

L'équation (30) peut être réécrite pour mettre en évidence la *prédiction* de la sortie  $y(t)$  à partir des mesures passées :

$$y(t) = -a_1 \cdot y(t-1) - a_2 \cdot y(t-2) - \dots - a_n \cdot y(t-n) + b_0 \cdot u(t) + b_1 \cdot u(t-1) + b_2 \cdot u(t-2) + \dots + b_m \cdot u(t-m) \quad (31)$$

Soit :

$$y(t) = \varphi^T(t) \cdot \theta(t)$$

avec :

$$\begin{cases} \varphi^T(t) = [-y(t-1) \ \dots \ -y(t-n) \ , \ u(t-1) \ \dots \ u(t-m)] \\ \theta(t) = [a_1 \ \dots \ a_n \ , \ b_1 \ \dots \ b_m] \end{cases} \quad (32)$$

Le but est de trouver le meilleur ensemble de paramètres  $\theta(t)$  : celui qui, à partir d'un ensemble de mesures  $\varphi(t)$  permet de prédire avec la meilleure précision la sortie effectivement mesurée.

Supposons qu'à la suite d'une expérience on dispose d'un ensemble de mesures  $\{y(t), u(t)\}_{t=1:N}$  et que l'ensemble des paramètres  $\theta$  n'est pas bien connu. On construit la séquence des *sorties prédites* :

$$\hat{y}(t) = \varphi^T(t) \cdot \theta(t) \quad (33)$$

Le paramètre  $\theta$  est celui qui minimise une norme quadratique :

$$\theta = \arg \max \left( \frac{1}{N} \sum_{t=1}^N (y(t) - \hat{y}(t))^2 \right) = \arg \max \left( \frac{1}{N} \sum_{t=1}^N (y(t) - \varphi^T(t) \cdot \theta(t))^2 \right) \quad (34)$$

Compte tenu du fait que l'équation est linéaire au départ et que le terme à minimiser est du type quadratique la solution est simplement obtenue par une pseudo-inverse :

$$\theta = \left[ \sum_{t=1}^N (\varphi(t) \cdot \varphi^T(t)) \right]^{-1} \sum_{t=1}^N (\varphi(t) \cdot y(t)) \quad (35)$$

On obtient donc la solution classique des moindres carrés. La formule (35) est le plus souvent utilisée sous sa forme récursive :

$$\theta(t) = \theta(t-1) + F(t) \cdot \varphi(t) \cdot (y(t) - \theta^T(t-1) \cdot \varphi(t))$$

avec

$$F(t)^{-1} = F(t-1)^{-1} + \varphi(t) \cdot \varphi(t)^T \quad (36)$$

ou

$$F(t) = F(t-1) - \frac{F(t-1) \cdot \varphi(t) \cdot \varphi(t)^T \cdot F(t-1)}{1 + \varphi(t)^T \cdot F(t-1) \cdot \varphi(t)}$$

(On laisse au lecteur le plaisir d'effectuer lui-même les démonstrations, qu'il peut du reste trouver dans [1] et [7]).

## 5.2 L'identification paramétrique en général et en pratique

L'exemple précédent a permis de mettre en évidence tous les aspects de la méthodologie classique de l'identification :

- il a fallu choisir un ensemble de mesures  $\{y(t), u(t)\}_{t=1:N}$ . Le choix du signal d'excitation  $u(t)$  est en pratique primordial,
- il a fallu choisir une structure pour le modèle à identifier (ici les ordres  $n$  et  $m$ ),
- il a fallu choisir un critère (ici un critère quadratique sur l'erreur entre la sortie et la sortie prédite),
- il a fallu choisir un algorithme et le paramétrer (ici l'algorithme des moindres carrés récursifs qu'il a fallu initialiser)

Enfin un point non traité dans le chapitre précédent concerne la validation :

- il aurait fallu déterminer une méthode pour comparer différents modèles et choisir le meilleur

### 5.2.1 Choix du signal d'excitation, traitement des mesures

De manière générale le système que l'on doit identifier doit être excité dans tout le domaine fréquentiel qui nous intéresse. Le signal classiquement utilisé est une séquence pseudo binaire aléatoire (Pseudo Random Binary Signal) tel celui représenté sur la Figure 17.

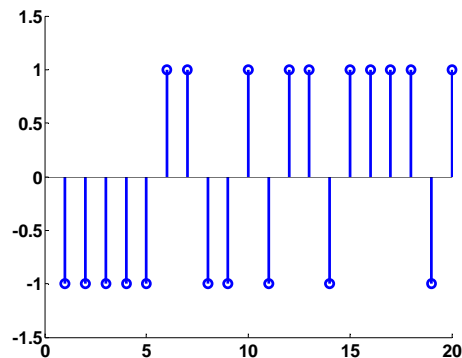


Figure 17 : exemple d'une séquence PRBS

Le signal d'excitation doit se superposer au signal de commande correspondant au point de fonctionnement étudié. Son amplitude doit être faible par rapport à l'amplitude des commandes du système (quelques %).

Une fois le système excité et les mesures effectuées il faut supprimer les valeurs moyennes des mesures.

### 5.2.2 Choix d'une structure de modèle

Le modèle étudié dans le chapitre précédent est de type ARX. On peut le noter sous forme polynomiale :

$$A \cdot y = B \cdot u \quad (37)$$

Ce modèle correspond en fait à un parti-pris sur la manière dont on traite les perturbations. En réalité on prend en compte un signal d'erreurs possibles  $e$  (de moyenne nulle) tels que :

$$A \cdot y = B \cdot u + e \quad (38)$$

Si l'on fait cette hypothèse de modélisation alors la méthode optimale est celle des moindres carrés. Malheureusement cette manière de prendre en compte les perturbations est bien restrictive. D'autres modélisations sont possibles, adaptées à d'autres algorithmes, par exemple le modèle OE :

$$y = \frac{B}{F} \cdot u + e \quad (39)$$

Il est aussi possible de "structurer" le modèle de bruit comme par exemple dans les modèles ARMAX ou BJ :

$$\begin{aligned} A \cdot y &= B \cdot u + C \cdot e \\ y &= \frac{B}{F} \cdot u + \frac{C}{D} \cdot e \end{aligned} \quad (40)$$

La structure du modèle étant choisie il faut choisir l'ordre de ses différents polynômes.

### 5.2.3 Choix d'un critère à minimiser

Dans la plupart des cas le critère à minimiser est l'erreur quadratique entre la sortie et la sortie prédite à un coup (one step ahead prediction). Ceci n'est bien évidemment pas restrictif et peut être généralisé à la prédiction à N coups.

### 5.2.4 Choix d'un algorithme

Même si chaque structure de modèle est généralement associée à un algorithme "optimal" celui-ci doit généralement se paramétrer.

Prenons par exemple l'algorithme des moindres carrés récursifs tel que présenté dans (36). Cette équation peut être modifiée en intégrant un "facteur d'oubli"  $\lambda$  :

$$\begin{aligned} \theta(t) &= \theta(t-1) + F(t) \cdot \varphi(t) \cdot (y(t) - \theta^T(t-1) \cdot \varphi(t)) \\ \text{avec} & \\ F(t)^{-1} &= \lambda \cdot F(t-1)^{-1} + \varphi(t) \cdot \varphi(t)^T \end{aligned} \quad (41)$$

Ce paramètre (typiquement entre 0.95 et 0.99) permet, comme son nom l'indique, "d'oublier" les anciennes mesures et de se concentrer sur les mesures les plus récentes.

### 5.2.5 Validation

Une fois l'algorithme d'identification appliqué et le modèle obtenu (ce qui prend généralement quelques secondes avec un ordinateur moderne) il reste à expertiser celui-ci. Selon la "qualité" du modèle obtenu on remet en cause l'une ou l'autre des étapes (signal d'excitation, structure du modèle, algorithme) et on réitère jusqu'à obtenir un modèle satisfaisant.

En pratique on essaie un certain nombre de modèles avant de choisir le "meilleur".

Le premier critère de qualité est la capacité du modèle à prédire effectivement la sortie (sortie prédite à un coup, sortie prédite à N coups). Il faut pour cela utiliser un jeu de mesures non utilisé pour l'identification. (En pratique le jeu de mesures initiales est coupé en deux : une partie pour l'identification et une partie pour la validation).

Le second critère de qualité est l'analyse du résidu : la composition spectrale de l'erreur de prédiction.

### 5.2.6 Utilisation de logiciels

En pratique il est commode d'utiliser un célèbre outil de calcul scientifique et sa toolboxe d'identification.

Comme les calculs proprement dit ne prennent que quelques secondes il est facile de réaliser un grand nombre d'identifications et de stocker les différents modèles pour les comparer ensuite. On est rapidement noyé sous le grand nombre de modèles possibles...

## 5.3 Identification "simple"

Il est souvent suffisant d'obtenir un modèle "simple" de type premier ordre et un simple essai indiciel convient souvent. Le système est excité avec un échelon unité et on cherche le meilleur modèle du premier ordre correspondant.

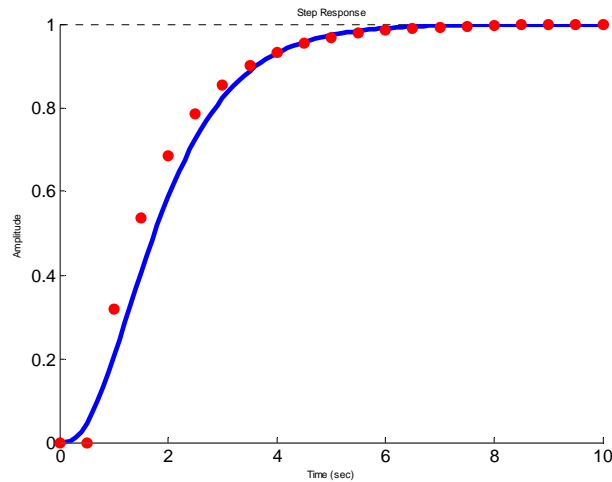


Figure 18 : identification sur un essai indiciel

Dans l'exemple de la Figure 18 on superpose le résultat d'une mesure (en bleu) avec la simulation d'un modèle discret avec une période de 0.5s. Le modèle cherché est du premier ordre avec un retard pur  $d$  :

$$y(z) = \frac{(1 + c_1) \cdot z^{-d}}{1 + c_1 \cdot z^{-1}} \cdot u(z) \quad (42)$$

Il s'agit de trouver l'entier  $d$  et le nombre réel  $c_1$  tels que la réponse indicielle du modèle corresponde pour le mieux aux mesures issues de l'essai. Ceci peut se formaliser par exemple de la manière suivante :

$$(c_1 \quad d_1) = \arg \min \sum_{t=1}^N (y_{\text{modèle}}(t) - y_{\text{mesure}}(t))^2 \quad (43)$$

Cette optimisation peut se faire à la main ou grâce à une procédure automatisée (avec le célèbre outil de calcul scientifique).

Dans le cas où le système comprend naturellement une intégration (par exemple la commande d'un moteur à courant continu par une tension et la mesure de l'angle de rotation) il est utile de prendre en compte explicitement cette intégration dans le modèle :

$$y(z) = \frac{1}{1 - z^{-1}} \cdot \frac{(1 + c_1) \cdot z^{-d}}{1 + c_1 \cdot z^{-1}} \cdot u(z) \quad (44)$$

La procédure est la même mais au lieu de prendre en compte la sortie  $y(t)$  on prend en compte la dérivée  $y'(t)$  :

$$y'(t) = y(t) - y(t-1) \quad (45)$$

## 5.4 Identification non paramétrique

Pour terminer ce panorama rapide sur les méthodes d'identification il faut parler des méthodes non paramétriques : celles qui permettent de tracer la réponse impulsionnelle ou la réponse fréquentielle du système.

### 5.4.1 Estimation de la réponse impulsionnelle

La réponse impulsionnelle d'un système peut être obtenue simplement en appliquant une entrée de type "impulsion" au système et d'en mesurer les conséquences.

La seconde méthode consiste à utiliser la méthode des moindres carrés (ou une méthode dérivée) comme vue au chapitre 5.1. La réponse impulsionnelle d'un système peut en effet être vue comme la fonction de transfert particulière où le dénominateur est de degré nul et le numérateur comporte un nombre infini (en pratique : grand) d'éléments ( $n = 0$  et  $m$  grand) :

$$y(t) = b_0 \cdot u(t) + b_1 \cdot u(t-1) + b_2 \cdot u(t-2) + \dots + b_m \cdot u(t-m) \quad (46)$$

La dernière méthode, qui sera développée ici est l'analyse d'intercorrélation.

Par définition l'intercorrélation  $R_{yu}$  entre deux signaux est :

$$R_{yu}(\tau) = E[(y(t) - E(y(t))) \cdot (u(t-\tau) - E(u(t-\tau)))] \quad (47)$$

Dans cette formule l'espérance mathématique peut être approchée grâce à la loi des grands nombres :

$$R_{yu}(\tau) \approx R_{yu}^N(\tau) = \lim_{N \rightarrow \infty} \frac{1}{N} \sum_{t=1}^N [(y(t) - m_y) \cdot (u(t-\tau) - m_u)] \quad (48)$$

On s'intéresse maintenant à un système tel que décrit par l'équation impulsionnelle (46) et on suppose que le signal d'entrée est un bruit blanc. Le signal d'entrée vérifie donc :

$$\begin{cases} \tau \neq 0 & R_{uu}(\tau) = 0 \\ \tau = 0 & R_{uu}(\tau) = R \end{cases} \quad (49)$$

Il vient alors naturellement :

$$b_k \approx \frac{1}{R} R_{yu}(k) \quad (50)$$

#### 5.4.2 Estimation de la réponse fréquentielle

On utilise ici le spectre d'intercorrélation. Par définition le spectre d'intercorrélation entre deux signaux est :

$$\Phi_{yu}(\omega) = \sum_{\tau=-\infty}^{\infty} R_{yu}(\tau) \cdot e^{-i\omega\tau} \quad (51)$$

On suppose que le signal d'entrée  $u(t)$  et le signal de sortie  $y(t)$  sont reliés par une fonction de transfert linéaire du type :

$$y(t) = F \cdot u(t) + v(t) \quad (52)$$

où  $v(t)$  est une perturbation indépendante de  $u(t)$ . On a alors les relations :

$$\Phi_{yu}(\omega) = F(j\omega) \cdot \Phi_{uu}(\omega) \quad (53)$$

Il reste à estimer les spectres d'intercorrélation pour remonter à la réponse fréquentielle  $F(j\omega)$  recherchée. Une idée naturelle est d'utiliser la relation (48) mais cela marche très mal dans presque tous les cas, même en choisissant  $N$  assez grand. L'amélioration est d'utiliser une "fenêtre" pour favoriser les valeurs faibles de  $\tau$  par rapport aux valeurs élevées. La solution classique utilisant la *fenêtre de Hamming* donne :

$$\Phi_{yu}^N(\omega) = \sum_{\tau=-\gamma}^{\gamma} R_{yu}^N(\tau) \cdot w_{\gamma}(\tau) \cdot e^{-i\omega\tau} \quad (54)$$

avec

$$\begin{cases} w_{\gamma}(\tau) = \frac{1}{2} \left( 1 + \cos\left(\frac{\pi \cdot \tau}{\gamma}\right) \right) & |\tau| < \gamma \\ w_{\gamma}(\tau) = 0 & |\tau| > \gamma \end{cases}$$

On utilise ensuite la relation (51) pour estimer la réponse fréquentielle :



$$F(j \cdot \omega) = \frac{\Phi_u^N(\omega)}{\Phi_{yu}^N(\omega)} \quad (55)$$



## 6. SYSTEMES BOUCLES

### 6.1 Fonction de transfert en boucle fermée

Un système et sa commande par ordinateur peut être représenté par un ensemble de fonctions de transfert formant un schéma-bloc plus ou moins compliqué (Figure 19).

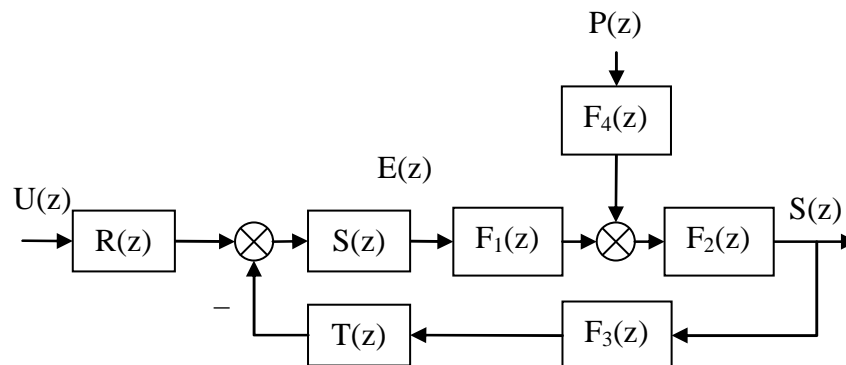


Figure 19 : système discret commandé par ordinateur. Le correcteur est représenté par les fonctions R,S,T, le système par  $F_1$  et  $F_2$ , le modèle de perturbations par  $F_4$  et le modèle des actionneurs par  $F_4$ .

Quelle que soit la structure d'un système asservi il est possible de se ramener à une forme canonique comprenant une fonction de transfert en boucle ouverte en combinant judicieusement les fonctions de transfert du procédé, du régulateur et des perturbations. La Figure 20 représente par exemple le système de la Figure 19 où  $F$  est une combinaison des fonctions de transfert du système, du régulateur, des perturbations et des capteurs.

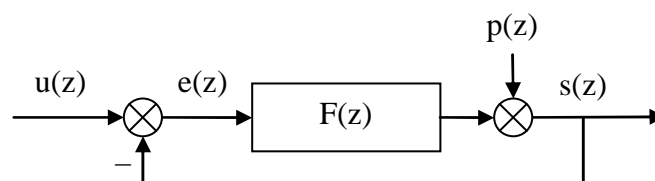


Figure 20 : système discret, notations.

Par défaut on considère une contre-réaction négative et une perturbation additive sur la sortie. La fonction de transfert  $F$  est appelée *fonction de transfert en boucle ouverte*. Comme pour les systèmes continus la *fonction de transfert en boucle fermée* est obtenue simplement à partir de  $F$  (58).

$$\begin{aligned}
 F(z) &= \frac{s(z)}{e(z)} = \frac{B(z^{-1})}{A(z^{-1})} \\
 G(z) &= \frac{s(z)}{u(z)} = \frac{B(z^{-1})}{A(z^{-1}) + B(z^{-1})}
 \end{aligned}
 \tag{56}$$

## 6.2 Erreur statique

L'erreur statique du système en boucle fermée, comme pour le système en boucle ouverte, est simplement le rapport des coefficients.

$$G_0 = \frac{\sum_{i=0}^{i=m} b_i}{1 + \sum_{i=0}^{i=m} b_i + \sum_{i=1}^{i=n} a_i}
 \tag{57}$$

Si l'on souhaite que l'asservissement confère un gain statique unitaire au système en boucle fermée (c'est-à-dire que l'on force la sortie  $s(k)$  à tendre vers la valeur de consigne  $u(k)$ ) il faut que ce rapport de coefficients soit égal à un. On montre facilement que cela est obtenu si la fonction de transfert  $F(z)$  comporte un intégrateur :

$$F(z) = \frac{s(z)}{e(z)} = \frac{1}{1 - z^{-1}} \cdot \frac{B(z)}{A'(z)}
 \tag{58}$$

## 6.3 Rejet de perturbations

La fonction de transfert entre la perturbation  $p(z)$  et la sortie  $y(z)$  est appelée fonction de sensibilité perturbation-sortie, notée  $S_{yp}$ .

$$S_{yp}(z) = \frac{s(z)}{p(z)} = \frac{A(z^{-1})}{A(z^{-1}) + B(z^{-1})}
 \tag{59}$$

Si l'on souhaite qu'une perturbation de type échelon soit parfaitement rejetée il faut que  $A(1) = 0$ , on retrouve donc la condition (58) à savoir que le système doit comporter une intégration dans la boucle directe. On peut être en outre être amené à spécifier que les perturbations ne soient pas amplifiées dans une certaine bande de fréquence. On rajoute donc une contrainte du type  $|S_{yp}(e^{j2\pi f})| < S_{\max}$  pour  $f$  appartenant à un certain domaine de  $[0 \ 0.5]$ .

## 6.4 Stabilité des systèmes à simple gain de rétroaction

Les critères de Routh et de Jury (donnés dans le chapitre 3.2.7 p13) ou le calcul numérique des pôles permettent de déterminer la stabilité du système en boucle fermée si l'on connaît sa fonction de transfert.

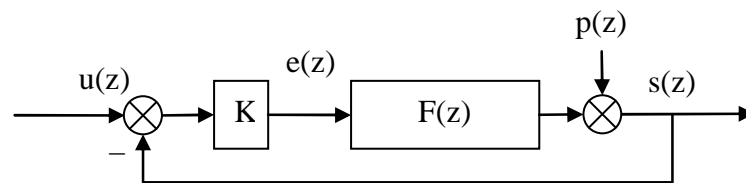


Figure 21 : système bouclé par un simple gain.

On s'intéresse ici au cas particulier où l'asservissement est réalisé par un simple gain de rétroaction  $K$  (Figure 21). On cherche la valeur de  $K$  qui amène le système en limite de stabilité. L'outil adapté ici est le lieu d'Evans ou lieu des racines (root locus) qui permet de visualiser l'évolution des pôles du système bouclé lorsque le gain  $K$  varie.

Prenons l'exemple d'un système comportant trois pôles et deux zéros. L'exemple est donné sous forme de script Matlab :

### Matlab

```
% Définition d'un système discret :
sys=zpk([-2.037 -0.1406],[0.6065 0.3679 0.3679],1,1)
%Tracé du lieu des racines
rlocus(sys);
```

Le script précédent produit le tracé de la Figure 22. On pourrait y lire que la limite de stabilité est obtenue pour  $K=0.2$  environ (détail de la Figure 23).

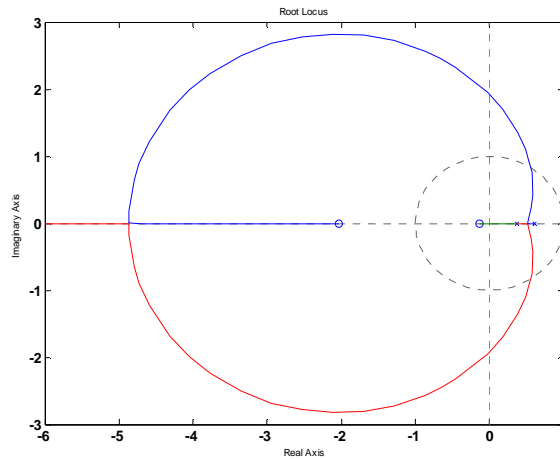


Figure 22 : lieu des racines

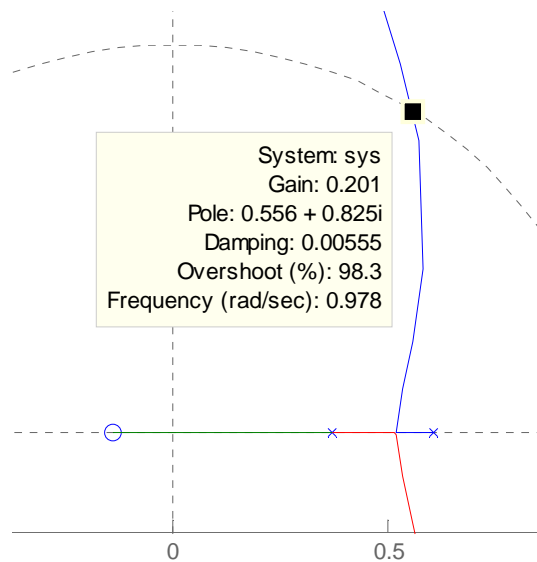


Figure 23 : lieu des racines : détail de la limite de stabilité

On voit sur la Figure 22 que le lieu des racines d'un système échantillonné ressemble à celui d'un système continu :

- on commence par tracer les pôles ( $\times$ ) et les zéros ( $o$ ) du système en boucle ouverte.
- les pôles du système en boucle fermé décrivent un lieu qui part des pôles (pour  $K = 0$ ) et arrive aux zéros (pour  $K = \infty$ )
- lorsque il y a  $n-m$  plus de pôles que de zéros le système comporte  $n-m$  branches asymptotiques.

On ne décrira pas ici les diverses propriétés du lieu des racines qui permettent de le tracer à la main puisque il est facile à obtenir avec un ordinateur.

Il peut aussi être utilisé pour régler "à la main" le gain permettant d'obtenir un ou plusieurs modes dominants ayant certaines propriétés de fréquence et d'amortissement. On voit sur la Figure 24 que le gain 0.017 permet d'obtenir un pôle double d'amortissement 0.706 et de fréquence 0.65 rad/s et un pôle simple de fréquence 1.63 rad/s.

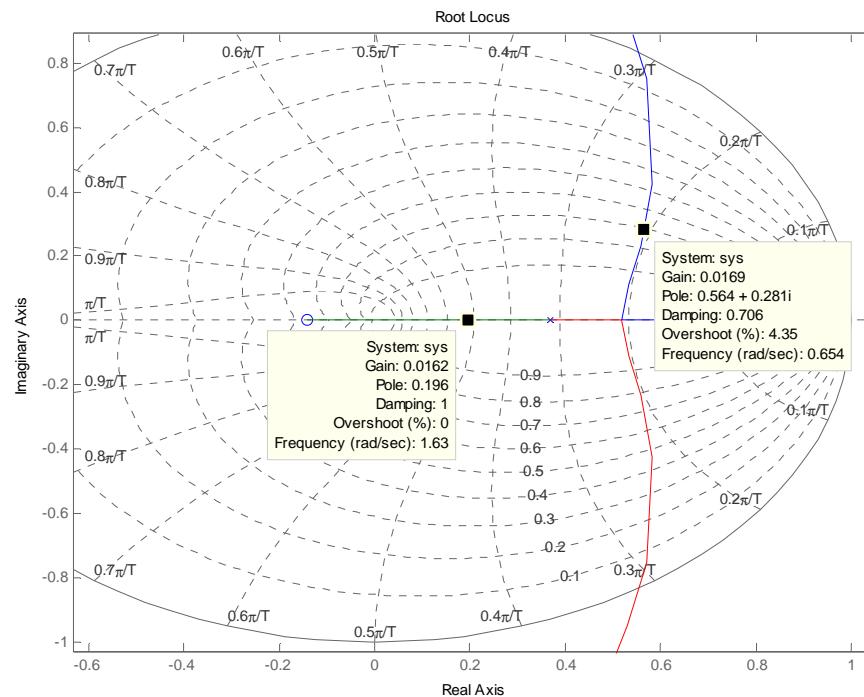


Figure 24 : utilisation du lieu des racines pour régler le gain de la boucle fermée

L'outil rltool de Matlab permet de nombreuses manipulations graphiques de ce type.

## 6.5 Analyse des systèmes bouclés dans le domaine fréquentiel : stabilité

Comme dans le cas continu il est possible de prévoir des propriétés, en particulier la stabilité, d'un système en boucle fermée à partir de l'examen de la fonction de transfert en boucle ouverte.

On ne redémontrera pas ici le théorème de Nyquist mais on se contente de rappeler l'énoncé.

On rappelle que le *lieu de Nyquist* est le lieu de  $F(j2\pi f)$  dans le plan complexe pour  $f$  variant de 0 à  $+\infty$ . A noter que le lieu pour la zone  $[0, 0.5]$  est symétrique de celui pour la zone  $[0.5, 1]$ . Pour cette raison on ne représente généralement que le lieu pour  $f$  appartenant à  $[0, 0.5]$ .

### 6.5.1 Théorème de Nyquist dans le cas où le système est stable en boucle ouverte (critère du revers)

Le système est stable en boucle fermée si le lieu de Nyquist de la fonction de transfert en boucle ouverte parcouru dans le sens des fréquences croissantes de 0 à 0.5 laisse le point critique  $(-1,0)$  à gauche.

La Figure 25 représente par exemple le lieu de Nyquist d'une fonction de transfert en boucle ouverte qui sera stable en boucle fermée.

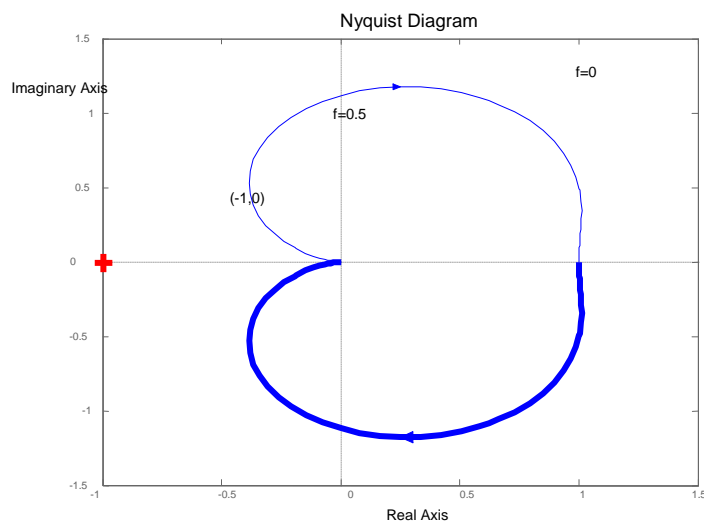


Figure 25 : lieu de Nyquist d'un système stable

### 6.5.2 Théorème de Nyquist dans le cas où le système est instable en boucle ouverte

Si le système en boucle ouverte comporte  $N$  pôles instables il sera stable en boucle fermée si le lieu de Nyquist de la fonction de transfert en boucle ouverte parcouru dans le sens des fréquences croissantes de  $-0.5$  à  $0.5$  entoure le point critique  $(-1,0)$   $N$  fois dans le sens des aiguilles d'une montre..

### 6.5.3 Interprétations du lieu de Nyquist

Comme dans le cas continu, la distance à laquelle on passe du point critique renseigne sur la robustesse du système bouclé. La distance sera exprimée de différente manière (Figure 26) :

- marge de module  $\Delta M$  : plus petite distance entre le lieu des racines et le point critique,



- marge de gain  $\Delta G$
- marge de phase  $\Delta P$

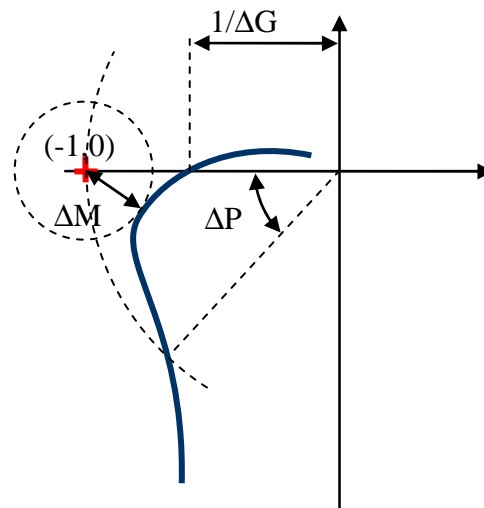


Figure 26 : Marges de stabilité

Les valeurs typiques de ces marges sont données ci-après :

- $\Delta G \geq 6\text{dB}$
- $\Delta M \geq -6\text{dB}$
- $30^\circ \leq \Delta P \leq 60^\circ$

La distance entre le point critique et le lieu des racines est la norme de  $1+F(j2\pi f)$ . On peut donc exprimer la marge de module en fonction de la fonction de sensibilité :

$$\begin{aligned} \Delta M &= \min |1 + F(j \cdot \omega)|_{\omega \in [0, \pi]} \\ \Delta M &= \min \left| 1 + \frac{B(j \cdot \omega)}{A(j \cdot \omega)} \right|_{\omega \in [0, \pi]} = \max \left| \frac{A(j \cdot \omega)}{A(j \cdot \omega) + B(j \cdot \omega)} \right|_{\omega \in [0, \pi]} \quad (60) \\ \Delta M &= \max |S_{yp}(j \cdot \omega)|_{\omega \in [0, \pi]}^{-1} \\ \Delta M_{\text{dB}} &= -\max |S_{yp}(j \cdot \omega)|_{\text{dB}} \end{aligned}$$

Si l'on trace l'évolution de  $S_{yp}$  en fonction de  $f$  on obtient donc la valeur de la marge de module.



# 7. METHODES DE REGULATION DES SYSTEMES ECHANTILLONES

## 7.1 Introduction

Les bases théoriques permettant de concevoir le correcteur pour un système échantillonné ont été posées dans les chapitres précédents. Ce chapitre fait le bilan des connaissances acquises et donne quelques solutions pour calculer des correcteurs à partir d'un cahier des charges précis.

Les systèmes que l'on cherche à commander sont généralement, par nature, continus. La réalisation d'un correcteur par un ordinateur implique que l'on génère une équation de commande discrète (généralement une équation récurrente). L'obtention de cette équation de commande peut se faire de deux manières :

- la première méthode consiste à d'abord synthétiser un correcteur continu (en utilisant les outils classiques de l'automatique et la simulation) et discrétiser ensuite ce correcteur. C'est l'objet du chapitre 7.2.2.
- la seconde méthode consiste à tenir la représentation discrète du système continu associé à son interface numérique (convertisseurs analogique-numérique et numérique-analogique) comme expliqué dans le chapitre 4.5 puis synthétiser directement le correcteur discret. Cela sera expliqué dans le chapitre 7.3 et suivants.

Ce choix peut être résumé par la figure suivante :

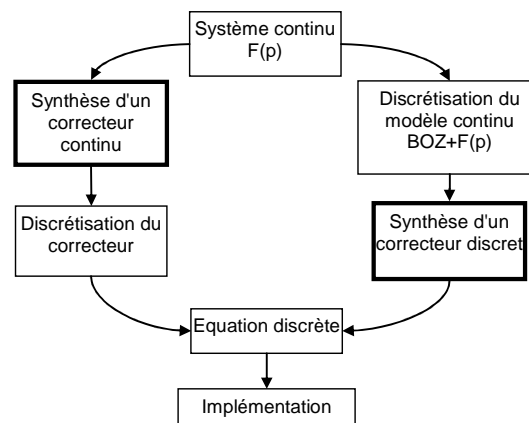


Figure 27 : Réalisation d'un correcteur discret.

Au-delà des affaires de goût, le choix de la méthode sera souvent dicté par l'application. En effet, la première méthode est d'autant meilleure que le correcteur discrétisé approche correctement le correcteur continu. Cela implique que le temps d'échantillonnage soit très faible par rapport aux constantes de temps du système

bouclé. Dans le cas où les contraintes temps réel sont fortes (capacités de calcul partagées par exemple) il sera préférable de synthétiser un correcteur discret.

## 7.2 Structure d'un système commandé par ordinateur, hypothèses

Par la suite on se ramènera toujours à un système tel que celui représenté sur la Figure 28 en cherchant un correcteur  $K(z)$  linéaire. On montre que quel que soit le correcteur linéaire on peut se ramener à la forme représenté sur la Figure 29 (forme RST).

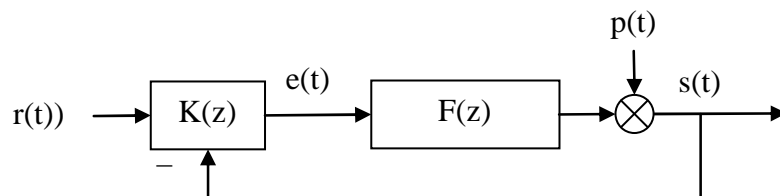


Figure 28 : système bouclé, forme générale.

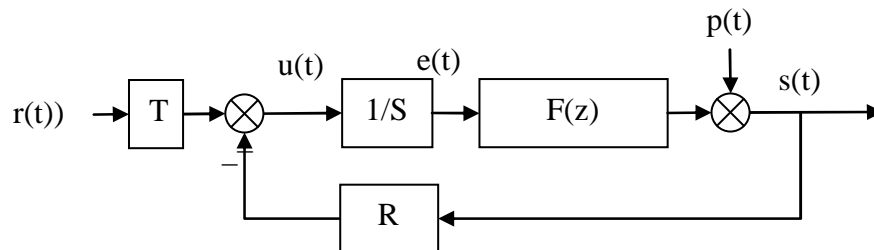


Figure 29 : système bouclé, forme canonique.

### 7.2.1 Discrétisation, choix de la période d'échantillonnage

Du point de vue du ordinateur, l'extérieur (le procédé) est représenté par la fonction de transfert  $F(z)$ . L'hypothèse importante est que la commande (sortie du ordinateur) et la mesure (entrée du ordinateur) sont échantillonnées de manière synchrone à la même période  $T_e$ .

Le choix judicieux de la fréquence d'échantillonnage est très important. En pratique on choisit cette fréquence en fonction de la bande passante du système en **boucle fermée** :

$$f_e = (6 \text{ à } 25) f^{BF} \quad (61)$$

### 7.2.2 Discrétisation d'un correcteur continu

La première méthode pour obtenir les équations d'un correcteur numérique est de faire sa synthèse avec les outils de l'automatique continue puis de discrétiser la fonction de transfert du correcteur continu obtenu (voir pour cela le chapitre 4.2 *Echantillonnage des systèmes continus*).

On va voir par la suite que la synthèse directe d'un correcteur discret offre plus de degré de liberté et permet en général de remplir un cahier des charges plus complet.

## 7.3 Régulateur PID numérique

Le régulateur le plus populaire est le régulateur PID. Nous allons voir dans un premier temps comment on génère un régulateur PID à partir des équations bien connues du régulateur PID continu et comment ces équations peuvent être généralisées pour obtenir plus de degrés de liberté pour le réglage.

### 7.3.1 Cas 1

Un régulateur PID continu a une fonction de transfert du type :

$$K_{\text{PID}}(s) = K \cdot \left( 1 + \frac{1}{T_i \cdot s} + \frac{T_d \cdot s}{1 + \frac{T_d}{N} \cdot s} \right) \quad (62)$$

La discrétisation de ce correcteur continu par la méthode d'Euler (Tableau 4 page 31) se fait en remplaçant la variable de Laplace  $s$  par le terme  $\frac{1-q^{-1}}{q^{-1} \cdot T_e}$ . On obtient :

$$K_{\text{PID}}(q) = K \cdot \left( 1 + \frac{1}{T_i \cdot \frac{1-q^{-1}}{T_e}} + \frac{T_d \cdot \frac{1-q^{-1}}{T_e}}{1 + \frac{T_d}{N} \cdot \frac{1-q^{-1}}{T_e}} \right) \quad (63)$$

En réarrangeant on obtient la forme générale RST (Figure 30):

$$K_{PID}(q) = K \cdot \frac{r_0 + r_1 \cdot q^{-1} + r_2 \cdot q^{-2}}{(1 - q^{-1}) \cdot (1 + s_1' \cdot q^{-1})}$$

$$\begin{cases} r_0 = K \cdot \left( 1 + \frac{T_e}{T_i} + \frac{N \cdot T_d}{T_d + N \cdot T_e} \right) \\ r_1 = -K \cdot \left( 1 + \frac{T_d}{T_d + N \cdot T_e} + \frac{T_e}{T_i} \cdot \frac{T_d}{T_d + N \cdot T_e} + \frac{N \cdot T_d}{T_d + N \cdot T_e} \right) \\ r_2 = K \cdot \left( \frac{T_d}{T_d + N \cdot T_e} \right) \\ s_1' = \frac{T_d}{T_d + N \cdot T_e} \end{cases} \quad (64)$$

Comme on le voit le correcteur obtenu comporte autant de degrés de liberté que le correcteur PID analogique. On reconnaît au dénominateur le terme  $(1 - q^{-1})$  en facteur qui assure l'effet d'intégration.

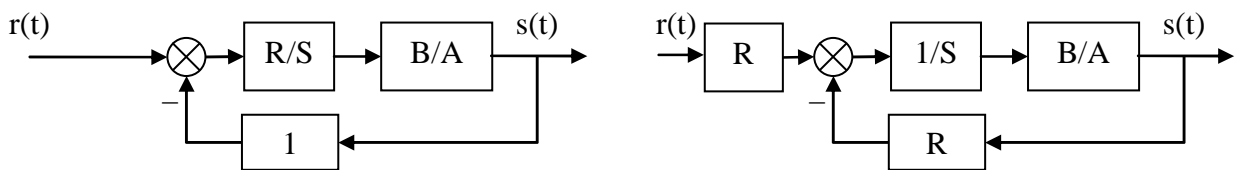


Figure 30 : PID sous forme RST

La fonction de transfert en boucle fermée obtenue est donnée par l'équation suivante :

$$H_{BF} = \frac{B \cdot R}{A \cdot S + B \cdot R} = \frac{B \cdot R}{P} \quad (65)$$

De manière générale le polynôme caractéristique  $P$  fixe les performances visées en boucle fermée. Le calcul de  $R$  et  $S$ ,  $P$  étant fixé et  $A$  et  $B$  étant connus, est réalisé en résolvant l'équation de Bézout  $P = A \cdot S + B \cdot R$ . (voir 9.1 Annexe 1 : Résolution de l'équation polynomiale de Bezout)

On voit que le numérateur  $B \cdot R$  est imposé une fois que  $R$  et  $S$  sont trouvés. Le régulateur introduit donc naturellement des zéros dans la boucle fermée.

Traisons l'exemple entièrement dans le cas où le système est modélisé comme un système du deuxième ordre avec faible retard. La fonction de transfert du procédé est donnée par :

$$H_{BO} = \frac{B}{A} = \frac{b_1 \cdot q^{-1} + b_2 \cdot q^{-2}}{1 + a_1 \cdot q^{-1} + a_2 \cdot q^{-2}} \quad (66)$$

Le correcteur PID rajoutant deux pôles, le polynôme caractéristique en boucle fermée spécifié doit posséder quatre pôles. Le polynôme caractéristique en boucle fermée doit donc être de la forme :

$$\begin{aligned} P &= 1 + p_1 \cdot q^{-1} + p_2 \cdot q^{-2} + p_2 \cdot q^{-3} + p_2 \cdot q^{-4} \\ P &= (1 + p_1' \cdot q^{-1} + p_2' \cdot q^{-2}) \cdot (1 + \alpha_1 \cdot q^{-1}) \cdot (1 + \alpha_2 \cdot q^{-1}) \end{aligned} \quad (67)$$

Et il doit être identifié au produit  $AS + BR$  avec  $R$  et  $S$  ayant les formes données par (64) :

$$\begin{aligned} &(1 + p_1' \cdot q^{-1} + p_2' \cdot q^{-2}) \cdot (1 + \alpha_1 \cdot q^{-1}) \cdot (1 + \alpha_2 \cdot q^{-1}) \\ &= (1 + a_1 \cdot q^{-1} + a_2 \cdot q^{-2}) \cdot (1 - q^{-1}) \cdot (1 + s_1' \cdot q^{-1}) \\ &+ (b_1 \cdot q^{-1} + b_2 \cdot q^{-2}) \cdot (r_0 + r_1 \cdot q^{-1} + r_2 \cdot q^{-2}) \end{aligned} \quad (68)$$

Les coefficients  $p_1'$  et  $p_2'$  définissent un modèle du second ordre aux caractéristiques (pulsation propre  $\omega_0$  et amortissement  $\sigma$  choisis et coefficients déduits conformément au Tableau 3 page 25). Les coefficients  $\alpha_1$  et  $\alpha_2$  correspondent à des pôles auxiliaires aperiodiques. Les choix typiques pour ces coefficients sont :

$$-0.05 < \alpha_1, \alpha_2 < -0.5 \quad (69)$$

On montre que la résolution de (68) est réalisée par une inversion matricielle simple :

$$M \cdot x = p \quad (70)$$

avec

$$\begin{cases} x^T = [1 & s_1' & r_0 & r_1 & r_2] \\ p^T = [1 & p_1 & p_2 & p_3 & p_4] \end{cases}$$

et

$$M = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ a_1' & 1 & b_1 & 0 & 0 \\ a_2' & a_1' & b_2 & b_1 & 0 \\ a_3' & a_2' & 0 & b_2 & b_1 \\ 0 & a_3' & 0 & 0 & b_2 \end{pmatrix} \quad (71)$$

On montre que M est inversible à la condition que A et B soient premiers entre eux, c'est-à-dire que A et B n'aient pas de pôles et de zéros en commun.

(Le problème peut aussi être résolu en utilisant la résolution de l'équation de Bézout).

### 7.3.2 Cas 2

Comme on l'a vu le correcteur PI numérique directement transposé du correcteur PID analogique décrit dans le chapitre 7.3.1 a l'inconvénient de ramener des zéros au dénominateur de la fonction de transfert en boucle fermée (équation 65). Il est possible, grâce à la flexibilité du régulateur numérique d'imposer au correcteur de ne pas ajouter de zéros en boucle fermée. La fonction de transfert désirée en boucle fermée sera donc de la forme :

$$H_{BF} = \frac{P(1)}{B(1)} \cdot \frac{B}{P} \quad (72)$$

Le facteur  $P(1)/B(1)$  garantit le gain statique unitaire du système en boucle fermée (voir page 12). Si le régulateur a la forme générale RST alors les polynômes R S et T doivent vérifier :

$$H_{BF} = \frac{B \cdot T}{A \cdot S + B \cdot R} = \frac{P(1)}{B(1)} \frac{B}{P} \quad (73)$$

S et R sont obtenus par la résolution de la même équation que précédemment (68). Le polynôme T est obtenu par simple identification :  $T = R(1)$ .

La structure, sous forme de schéma-bloc est donnée ci-dessous :

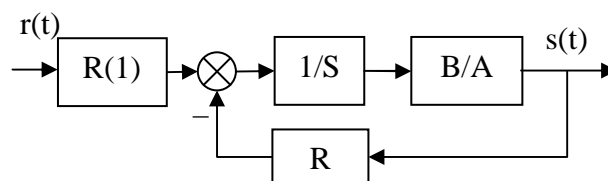


Figure 31 : PID2 sous forme RST

On montre que cette structure de PID, ramené en continu, correspond au PID canonique pour l'action intégrale. L'action proportionnelle et dérivée n'est appliquée que sur le signal  $y$ .

La structure du PID2 par rapport au PID1 a pour avantage de ne pas ramener de zéros à la boucle fermée. Cela se traduit par un dépassement moindre en transitoire lorsqu'on applique un échelon de commande. Par contre les deux systèmes réagissent de la même manière à un échelon de perturbation.



## 7.4 Placement de pôles

Les deux structures de PID décrites dans le chapitre précédent sont deux cas particulier d'une stratégie plus générale de placement de pôles.

La synthèse est plus générale que les PID présentés précédemment puisqu'il n'est pas nécessaire de faire d'hypothèses sur le degré de A et B, sur le retard, ni sur la présence ou non de zéros stables ou instables dans le procédé.

La structure du correcteur est la structure type RST telle que présentée dans la Figure 29. Pour la suite des calculs on posera pour la fonction de transfert les conventions suivantes :

$$H_{BO} = \frac{q^{-d} \cdot B(q^{-1})}{A(q^{-1})} \quad (74)$$

$$\begin{cases} A(q^{-1}) = 1 + a_1 \cdot q^{-1} + \dots + a_{n_A} \cdot q^{-n_A} \\ B(q^{-1}) = b_1 \cdot q^{-1} + \dots + b_{n_B} \cdot q^{-n_B} = q^{-1} \cdot B^*(q^{-1}) \end{cases}$$

Compte tenu de la forme RST du régulateur la fonction de transfert en boucle fermée vérifie :

$$H_{BF} = \frac{q^{-d} \cdot B(q^{-1}) \cdot T(q^{-1})}{P(q^{-1})} \quad (75)$$

$$P(q^{-1}) = A(q^{-1}) \cdot S(q^{-1}) + q^{-d} \cdot B(q^{-1}) \cdot R(q^{-1}) = 1 + p_1 \cdot q^{-1} + \dots + p_{n_P} \cdot q^{-n_P}$$

Le polynôme  $P(q^{-1})$  définit donc les pôles de la fonction de transfert en boucle fermée et traduit donc le comportement en *poursuite*.

De la même manière la fonction de transfert entre une perturbation et la sortie (fonction de sensibilité perturbation-sortie) est donnée par :

$$S_{yp} = \frac{A(q^{-1}) \cdot S(q^{-1})}{P(q^{-1})} \quad (76)$$

Ce même polynôme traduit donc aussi le comportement en *régulation*.

Les consignes en poursuite et en régulation peuvent être différentes comme cela est représenté par exemple sur la Figure 32.

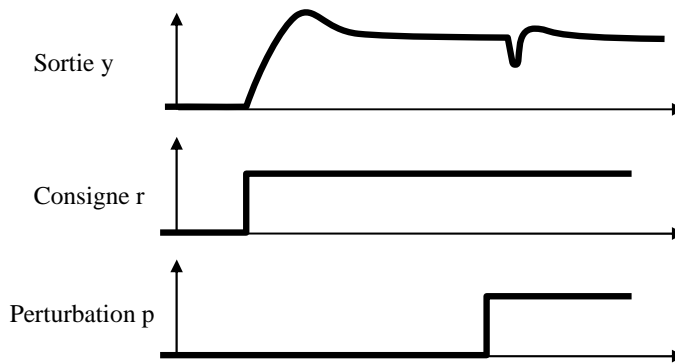


Figure 32 : Performances différentes en régulation et en poursuite

### 7.4.1 Choix des pôles en boucle fermée

De manière générale le comportement en boucle fermée est spécifié sous la forme d'une pulsation propre  $\omega_0$  et d'un amortissement  $\sigma$ , donc une fonction de transfert du second ordre. La spécification étant généralement donnée en continu il est nécessaire de traduire la valeur désirée de  $\omega_0$  et  $\sigma$  en un polynôme  $P_D(q^{-1})$  par discrétisation de la fonction de transfert continue (chapitre 4.5 page 29). Ce polynôme  $P_D$  traduit le comportement dominant du système bouclé.

Si l'on souhaite introduire un effet de filtrage supplémentaire dans certaines bandes de fréquence (réduction des bruits de mesure, adoucissement de la commande, robustesse) cela est décrit par un polynôme  $P_F$  comportant des pôles auxiliaires, plus rapides que les pôles de  $P_D$ .

Le polynôme caractéristique en boucle fermée est donc constitué de pôles dominants et de pôles auxiliaires :

$$P(q^{-1}) = P_D(q^{-1}) \cdot P_F(q^{-1}) \quad (77)$$

### 7.4.2 Régulation : calcul de R et S

Le polynôme  $P(q^{-1})$  étant fixé il s'agit de trouver R et S vérifiant :

$$P(q^{-1}) = A(q^{-1}) \cdot S(q^{-1}) + q^{-d} \cdot B(q^{-1}) \cdot R(q^{-1}) \quad (78)$$

Comme déjà vu au chapitre 7.3.1 on obtient une équation de Bézout qu'il s'agit de résoudre. (Voir aussi 9.1 Annexe 1 : Résolution de l'équation polynomiale de Bezout)

Il est parfois nécessaire de spécifier R et S avec des parties fixes  $H_R$  et  $H_S$ . Dans ce cas l'équation (78) est modifiée et on l'écrit sous la forme:

$$P(q^{-1}) = (A(q^{-1}) \cdot H_S(q^{-1})) \cdot S'(q^{-1}) + (q^{-d} \cdot B(q^{-1}) \cdot H_R(q^{-1})) \cdot R'(q^{-1}) \quad (79)$$

Les inconnues que l'on souhaite trouver sont  $S'$  et  $R'$ . Ceci est nécessaire dans les cas particuliers suivants :

- spécification d'erreur statique nulle,
- rejet d'une perturbation harmonique,
- blocage d'un signal,

Détaillons maintenant ces cas particuliers.

### Spécification d'une erreur statique nulle

On retrouve la nécessité d'avoir un terme intégrateur dans la boucle directe. La spécification se retrouve donc sur  $S$  :

$$S(q^{-1}) = (1 - q^{-1}) \cdot S'(q^{-1}) = H_S(q^{-1}) \cdot S'(q^{-1}) \quad (80)$$

### Rejet d'une perturbation harmonique

Si le système est perturbé par une perturbation harmonique à une fréquence fixe (par exemple 50Hz...) il est possible d'imposer à  $S$  d'avoir un module *nul* ou amorti à cette fréquence. Cela se traduit par la présence dans  $S$  d'une paire de zéros complexe à amortissement nul ou faible à cette fréquence :

$$S(q^{-1}) = (1 + \alpha \cdot q^{-1} + \beta \cdot q^{-2}) \cdot S'(q^{-1}) = H_S(q^{-1}) \cdot S'(q^{-1}) \quad (81)$$

### Blocage d'un signal

Pour certaines application on souhaite que le régulateur "laisse passer" certaines composantes du signal de sortie, c'est-à-dire qu'il ne réagisse pas à ces composantes. Cela se traduit par  $R=0$  à ces fréquences. On retrouve la même forme que précédemment :

$$R(q^{-1}) = (1 + \alpha \cdot q^{-1} + \beta \cdot q^{-2}) \cdot R'(q^{-1}) = H_R(q^{-1}) \cdot R'(q^{-1}) \quad (82)$$

### 7.4.3 Poursuite : calcul de T

Le schéma-bloc de l'asservissement est donné sur la Figure 33. Le terme  $T$  est décomposé en un terme  $T'$  et un terme  $B^*/A^*$ .

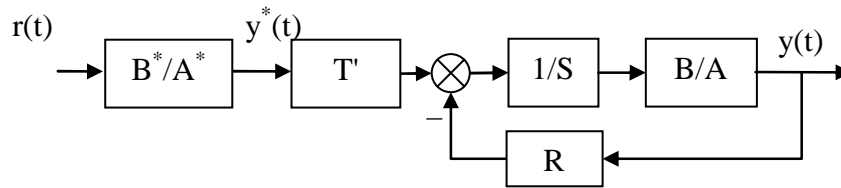


Figure 33 : placement de pôle

L'idée est de régler  $B^*$  et  $A^*$  de telle manière que le signal  $y^*$  ait la dynamique de poursuite désirée (par exemple un second ordre avec pulsation propre et amortissement donné) puis de régler  $T'$  de telle manière que le transfert entre  $y^*$  et  $y$  soit le plus "transparent" possible.

### Réglage de $T'$

Le transfert entre  $y^*$  et  $y$  est :

$$S_{yy^*}(q^{-1}) = T'(q^{-1}) \frac{q^{-d} \cdot B(q^{-1})}{P(q^{-1})} \quad (83)$$

Pour assurer le transfert le plus "transparent" possible il faut donc choisir

$$T'(q^{-1}) = \frac{P(q^{-1})}{B(1)} \quad (84)$$

Remarques :

- Si  $B(1)=0$  alors on remplace  $B(1)$  par 1 dans (84)
- De même dans (84)  $P(q^{-1})$  peut être remplacé par  $P_D$  correspondant à la dynamique dominante.

### Réglage de $B^*$ et $A^*$

$B^*$  et  $A^*$  sont choisis de telle manière que la fonction de transfert  $B^*/A^*$  corresponde à la dynamique de poursuite souhaitée.

## 7.5 Poursuite et régulation à objectifs indépendants

Cette méthode ressemble à la méthode de placement de pôles à ceci près que l'on va effectuer une simplification des zéros de  $B^*(q^{-1})$  : cela ne s'applique donc qu'aux systèmes avec des pôles stables. Plus précisément il faut qu'ils aient un amortissement suffisant ( $> 0,2$ ) de manière à ne pas être trop près du cercle unité.

La structure de l'asservissement est toujours celle donnée sur la Figure 33.

**Choix de P**

$P(q^{-1})$  est choisi de manière à simplifier  $B^*(q^{-1})$  (rappel : la forme de B est donnée dans l'équation 74). Comme indiqué dans le chapitre 7.4.2 page 60 il s'agit donc d'imposer les zéros de  $B(q^{-1})$  dans  $H_R(q^{-1})$  et  $H_S(q^{-1})$ .

**Choix de T**

Si l'on choisit  $T(q^{-1}) = P(q^{-1})$  alors le transfert entre  $y^*$  et  $y$  est :

$$S_{yy^*}(q^{-1}) = q^{-(d+1)} \quad (85)$$

**Réglage de  $B^*$  et  $A^*$** 

$B^*$  et  $A^*$  sont choisis de telle manière que la fonction de transfert  $B^*/A^*$  corresponde à la dynamique de poursuite souhaitée.



## 8. TECHNOLOGIE DE LA COMMANDE PAR ORDINATEUR

Du point de vue de l'automaticien le ordinateur est relié au système physique à piloter par un certain nombre d'intermédiaires plus ou moins gênants :

- les interfaces analogiques et numériques,
- le transport de l'information par bus et protocole,
- l'utilisation éventuelle de capteurs ou d'actionneurs « intelligents ».

Il ne s'agit pas ici de faire un exposé complet sur ces trois aspects mais de poser les problèmes qui se posent à l'automaticien lorsque le système qu'il doit commander est perturbé par ces éléments.

Enfin ce chapitre se termine sur les notions de « Hardware In the Loop » et de « prototypage rapide » qui sont partie intégrante des processus de réalisation de systèmes de contrôle.

### 8.1 Calculateurs et interfaces numériques / analogiques

Le premier problème à prendre en compte est la quantification de l'information. Du point de vue du ordinateur d'une part et du point de vue des convertisseurs d'autre part. Dans un premier temps, considérons les aspects performance de calcul lié à la nature du codage des nombres dans le ordinateur.

#### Les performances liées au types de variables

Prenons l'exemple où le concepteur de la loi de commande souhaite réaliser un correcteur de type proportionnel. Entre l'acquisition des mesures et l'écriture de la consigne il y aura la réalisation d'une de commande du type :

$$y = K \cdot (\text{consigne} - \text{mesure}) \quad (86)$$

En utilisant un langage de programmation évolué (par exemple le C) la ligne de code sera la transcription littérale de (86). Le programmeur devra décider de la définition et du type des variables, principalement du nombre d'octets utilisés pour stocker ses variables. Le Tableau 5 résume par exemple les types utilisés en langage C (norme ANSI)

Tableau 5 : Types ANSI C

Types entiers	char	8 bits
	short int	16 bits
	long int	32 bits
Types flottants (mantisse et exposant)	float	32 bits
	double	64 bits
	longdouble	128 bits

Une fois le code écrit en C l'exécution du programme sur un processeur cible ne peut se faire que par l'intermédiaire d'une compilation du programme. A chaque processeur est donc adapté une version de compilateur.

Les performances du programme (sa rapidité d'exécution) dépendront donc en partie de la qualité du compilateur. (Par exemple, une multiplication s'effectue t'elle bêtement par un décalage de tous les bits ou bien des solutions astucieuses sont-elles implémentées pour arrêter le décalage selon le contexte ?)

Le principal déterminant de la performance est cependant le processeur : son type (8 bits, 16 bits ou 64 bits) et son cadencement (en MHz ou GHz) sont les deux plus importants paramètres. Coder les variables sur 64 bits (double) dans le programme coûtera cher en temps de calcul si le processeur est un 8 bits. Le type de mémoire et leurs temps d'accès peuvent aussi être déterminants. Enfin, certains processeurs disposent d'un processeur spécifiquement dédié aux opérations mathématiques (coprocesseur mathématique).

Il n'est pas possible de dire exactement, pour tel type de compilateur associé à tel type de processeur, le temps que prendra par exemple une multiplication. En pratique on effectue une série de tests types (benchmark) et on affiche un temps moyen d'exécution.

En résumé, on peut être tenté, pour ne pas s'embêter, de coder toutes les variables en virgule flottante. Mais il faut bien savoir que cela a un coût en terme de temps de calcul, surtout si le processeur n'est pas adapté. Pour fixer les idées on peut dire qu'il y a un rapport 20 entre une addition réalisée en "short" et une addition réalisée en virgule flottante.

Enfin, l'utilisation de fonctions mathématiques (sinus, exponentielles, etc...) peut aussi avoir un coût très lourd si le travail n'est pas réalisé par un coprocesseur mathématique dédié. Le compilateur et sa librairie mathématique joue là encore un grand rôle.

### La quantification imposée par le processeur

Au-delà de l'aspect performance et temps de calcul, la quantification de l'information doit parfois aussi être prise en compte. Rappelons qu'un entier codé sur N bits peut prendre seulement  $2^N - 1$  valeurs différentes. Avant d'effectuer les calculs mathématiques dans le corps du programme il est donc judicieux d'appliquer un



facteur d'échelle de manière à utiliser l'amplitude maximale possible (entre 0 et  $2^N - 1$ ).

### **La quantification imposée par les conversions analogique et numérique**

Les convertisseurs analogique-numérique et numérique-analogique sont principalement caractérisés par leur amplitude de signal analogique d'entrée (généralement en volts, par exemple +/- 10V) et par leur résolution en bits (8 ou 10 bits pour les plus courants, jusqu'à 16 bits pour les convertisseurs haut de gamme).

Prenons l'exemple d'un capteur gyromètre transformant une vitesse de rotation en tension. Compte tenu de l'application le signal utile varie entre 2,4V et 2,6V. Si le signal mesuré est branché directement sur un convertisseur 0-10V 10bits alors un problème de conditionnement du signal se pose. En effet la quantification est de  $\frac{10V}{2^{10}-1} = 0,098V$ . Les 0,2V de signal utile seront donc discrétisés en  $\frac{0,2}{0,098} = 20$  intervalles seulement !

Si la discrétisation du signal utile est trop grossière cela peut se traduire par une diminution des performances de l'asservissement, voire une instabilité. Le système n'est en effet plus linéaire et il n'existe pas d'outils théoriques permettant de prévoir les performances du système bouclé.

## **8.2 Transport de l'information : les bus et les protocoles**

Pas question de faire ni même d'ébaucher un cours sur les bus et les protocoles ici. Mais on va quand même donner quelques indications. L'automaticien a l'habitude de simplifier le problème qu'il se pose en dessinant un schéma simplifié de son problème d'asservissement (du type de la Figure 20 page 45) pour pouvoir appliquer les théories de la commande. Cependant il arrive bien souvent que l'information transite à l'intérieur du système, soit entre différents calculateurs effectuant différentes parties du travail soit entre ordinateur et périphériques, par l'intermédiaire de bus informatique. A chaque type de bus correspond un ou plusieurs protocoles qui peuvent poser des problèmes pour l'automaticien :

- les données sont-elles discrétisées ? (le protocole peut par exemple ne permettre de transmettre que des paquets de données de 16 bits),
- le bus induit-t-il un retard ? (le composant mettant en forme les données du ordinateur pour les publier sur le bus a certaines performances qui peuvent induire un délai)
- le bus est-il déterministe ? (Certains bus ne garantissent pas que les données arriveront en un temps donné),
- le bus est-il sûr ? (Certains bus ne garantissent pas non plus que les données arriveront bien)

### **8.3 Capteurs et actionneurs "intelligents", "smart sensors"**

De plus en plus souvent les capteurs et les actionneurs sont associés à leur électronique de puissance, électronique de commande et de conditionnement de signal. L'interface avec le capteur ou l'actionneur se fait donc via un protocole de communication (par exemple via un bus RS232).

L'intérêt est de véhiculer non pas des signaux analogiques, tensions ou courants, mais des signaux numériques beaucoup moins sensibles aux éventuelles perturbations.

On retrouve donc, entre le capteur ou l'actionneur et le ordinateur, la présence d'un bus et d'un protocole de communication.

### **8.4 La philosophie "hardware in the loop"**

Les systèmes embarqués sont de plus en plus complexes et touchent des domaines de plus en plus variés : automobile, satellites, aéronautique, UAV, systèmes d'arme, sous-marins, systèmes, etc... Les systèmes à piloter ainsi que les systèmes de commande sont de plus en plus complexes.

Les techniques de simulation « Hardware In the Loop » (HIL) sont de plus en plus utilisées dans le développement et dans le test de ces systèmes embarqués. L'idée est de développer entièrement le système de contrôle sur une plateforme de test en utilisant le maximum de composants réels et en ne remplaçant que le minimum par la simulation. Par exemple en aéronautique l'« Iron Bird » regroupe dans un hangar tous les systèmes réels de l'avion en cours de développement. Seule la mécanique du vol est simulée : les sorties des actionneurs sont mesurées, rentrées dans un modèle de simulation numérique, puis les mesures effectuées normalement par les capteurs sont calculées et restituées par le modèle.

L'intérêt est de diminuer de manière importante les coûts et les délais en économisant le plus grand nombre possible de tests sur le système réel en situation. Lors des premiers tests réels les techniques « Hardware In the Loop » permettent d'avoir effectué la plus grande part du développement des lois de commande.

### **8.5 Le prototypage rapide en automatique**

Un peu dans le même ordre d'idée que le « Hardware In the Loop » le prototypage rapide permet de réaliser la conception des lois de commande en simulation puis de transporter rapidement (en un simple click !) le code du contrôleur simulé vers la cible temps réel implémentée sur le système réel.

La société Mathworks [9] investit massivement dans ce domaine en ayant enrichi son logiciel de calcul scientifique (Matlab) avec un logiciel de simulation graphique très évolué (Simulink) puis en développant des outils de prototypage rapide associés (Real Time Workshop, Embedded Target For...).

## 9. ANNEXE

### 9.1 Annexe 1 : Résolution de l'équation polynomiale de Bezout

Il s'agit de trouver R et S vérifiant :

$$P(q^{-1}) = A(q^{-1}).S(q^{-1}) + B(q^{-1}).R(q^{-1})$$

Où P, A et B sont connus.

Dans le cas où P, A, B, R et S sont des nombres entiers la solution de l'algorithme est donnée par l'algorithme de division Euclidienne étendu [8].

Dans le cas où P, A, B, R et S sont des polynômes, l'algorithme est donné ci-dessous, programmé en matlab ([1]).

```
function [Rp,Sp,nrp,nsp]=bezoutd(A,B,Hs,Hr,P)
%function [Rp,Sp,nrp,nsp]=bezoutd(A,B,Hs,Hr,P)
%solves AHsSp+BHrRp=P by coefficient comparison. Delay and discretization delay need to be integrated in B
%(zeros at the beginning).
%
%inputs:
%A=[a0 a1 ... aNa] ... vector of model denominator coefficients A=a0 + a1z^(-1) + a2z^(-2) +...+ aNaz^(-Na)
%B=[b0 b1 ... bNb] ... vector of model numerator coefficients B=b0 + b1z^(-1) + b2z^(-2) +...+ bNbz^(-Nb)
%Hs=[hs0 hs1 ... hsNhs] ... vector of controller denominator fixed part Hs=hs0 + hs1z^(-1) +...+ hsNhsz^(-Nhs)
%Hr=[hr0 hr1 ... hrNhr] ... vector of controller denominator fixed part Hr=hr0 + hr1z^(-1) +...+ hrNhrz^(-Nhr)
%P=[p0 p1 ... pNp] ... vector of desired polynomial coefficients P=p0 + p1z^(-1) + p2z^(-2) +...+ pNpz^(-Np)
%outputs:
%Rp=[rp0 rp1 rp2 ...] ... vector of coefficients for resulted controller numerator
%Sp=[sp0 sp1 sp2 ...] ... vector of coefficients for resulted controller denominator
%nrp ... order of Rp
%nsp ... order of Sp
%
%written by: J. Langer, I.D. Landau, H. Prochazka
%7th june 2002

PRECISION=1e-16;

D=size(A);
if D(1)>1, A=A'; end;
```

```

D=size(B);
if D(1)>1, B=B'; end;
D=size(Hs);
if D(1)>1, Hs=Hs'; end;
if D(1)==0, Hs=1; end;
D=size(Hr);
if D(1)>1, Hr=Hr'; end;
if D(1)==0, Hr=1;end;
D=size(P);
if D(1)>1, P=P'; end;

na=length(A)-1;
nb=length(B)-1;
np=length(P)-1;
nhs=length(Hs)-1;
nhr=length(Hr)-1;

if (nhs>0), Ah=real(convz(A,Hs)); else Ah=A*Hs; end; % Ah = A * Hs
nah=length(Ah) -1;
if (nhr>0), Bh=real(convz(B,Hr)); else Bh=B*Hr; end;
nbh=length(Bh) -1;
if (np>nah+nbh-1), disp('Bezout error: too many poles');end;

% increase size of P using zeros if necessary
if (np<nah+nbh-1),
%
% set remaining poles onto a circle with radius rmin
P=[P zeros(1,nah+nbh-1-np)];
rootsPdes=roots(P);
nextra=nah+nbh-1-np;
rmin=1e-16;
angle=[0:nextra-1]'/nextra*2*pi;
j=sqrt(-1);
rootsPextra=rmin*exp(j*angle);
P=poly([rootsPdes;rootsPextra]);
np=nah+nbh-1;

end;
P,

nsp=nbh-1;
nrp=nah-1;

%matrix is smaller than vector PD

```

```

if (np>nah+nbh-1),
    disp('Order of model denominator is too low! Add a polynom of higher order to Hs or Hr. ');
end;

% ns=nsp+nhs
% nr=nrp+nhr

M=[];
for j=1:nsp+1,
    V=[];
    if (j>1), V=[V ; zeros(j-1,1)]; end;% zeros in front of Ah
    V=[V ; Ah'];% Ah
    if (j<=nsp), V=[V ; zeros(nsp+1-j,1)]; end;% zeros behind Ah
    if (length(V)~=nah+nbh), disp('bezoutb: error V'); end;
    M=[M V]; % add one column to M
end;

for j=1:nrp+1,
    V=[];
    if (j>1), V=[V ; zeros(j-1,1)]; end;
    V=[V ; Bh'];
    if (j<=nrp), V=[V ; zeros(nrp+1-j,1)]; end;
    if (length(V)~=nah+nbh), disp('bezoutb: error V'); end;
    M=[M V];
end;

D=size(M);
if (D(1)~=nah+nbh), disp('bezoutb: error size M row'); end;
if (D(2)~=nah+nbh), disp('bezoutb: error size M column'); end;

% make P column vector
P=P';

global M1;
M1=M;

X= M\P;
% coefficients are real values
X=real(X);

% make X row vector
X=X';

```

## Commande des systèmes par ordinateur

```
Sp=X(1:nsp+1);  
Rp=X(nsp+2:nsp+nrp+2);
```

## 10. BIBLIOGRAPHIE

- [1] I. D. Landau, "*Commande des systèmes*", Edition Lavoisier 2002.
- [2] C. Nouals, "Théorie du Signal", Notes de cours ENSICA
- [3] F. Vincent, "*Traitement Numérique du Signal*", Notes de cours ENSICA.
- [4] D. Arzelier, "*Représentation et analyse des systèmes, notes de cours*", Edition ENSICA. Voir aussi <http://www.laas.fr/~arzelier/>
- [5] J. Bordeneuve-Guibé, "*Commande des systèmes linéaires*", Notes de cours ENSICA.
- [6] Y. Briere, "*Identification*", Notes de cours ENSICA.
- [7] L. Ljung, "*System Identification, Theory for the User*", Prentice Hall 1987.
- [8] Wikipedia , équation de Bezout : [http://en.wikipedia.org/wiki/Bézout's\\_identity](http://en.wikipedia.org/wiki/Bézout's_identity)
- [9] <http://www.mathworks.com>





# 11. INDEX

<b>A</b>	
actionneur intelligent .....	21
amortissement .....	15
ANSI C .....	60
Approximation numérique d'équation différentielle .....	28
ARMAX .....	35
<b>B</b>	
BJ35 .....	
Blocage d'un signal .....	56
bloqueurs d'ordre 0 .....	22
BOZ .....	22
bus.....	21, 61
<b>C</b>	
CAN.....	21
capteur intelligent .....	21
causal .....	6, 9
Causalité .....	10
CNA.....	21
compilateur .....	60
convertisseur analogique / numérique.....	21
convertisseur numérique / analogique.....	21
convertisseurs analogique-numérique .....	61
coprocesseur mathématique .....	60
Correspondances pôles - zéros.....	31
<b>D</b>	
Diagramme de Bode .....	18
Discrétisation de fonction de transfert continues .....	27
division polynomiale.....	6
<b>E</b>	
échelon unitaire.....	11
équation d'état .....	17
Erreur statique.....	42
Estimation de la réponse fréquentielle .....	39
Estimation de la réponse impulsionnelle.....	38
Euler .....	29
<b>F</b>	
facteur d'échelle .....	61
facteur d'oubli .....	36
<i>fenêtre de Hamming</i> .....	39
filtre anti-repliement .....	25
fonction de sensibilité .....	42
fonction de transfert discrète.....	9
<b>G</b>	
gain statique.....	11
gain statique unitaire.....	42
<b>I</b>	
Identification.....	32
Identification non paramétrique .....	38
intercorrélation.....	38
<b>L</b>	
lieu des racines.....	43
lieu d'Evans.....	43
LTI.....	8
<b>M</b>	
marge de gain.....	47
marge de module.....	46
marge de phase .....	47
méthode des moindres carrés récursifs .....	32
Méthode des trapèzes.....	29
modèle ARX .....	32
<b>O</b>	
OE.....	35
one step ahead prediction.....	35
opérateur de retard .....	9
<b>P</b>	
période d'échantillonnage (choix) .....	26
Placement de pôles .....	54
Pôles .....	11
Poursuite .....	56
<i>prédiction</i> .....	32
protocoles.....	61
Pseudo Random Binary Signal .....	34
pulsation .....	15
<b>Q</b>	
quantification .....	23, 24, 60
<b>R</b>	
R,S,T.....	41
rapidité.....	15
rayon de convergence .....	5, 9
Régulation.....	55

Rejet de perturbations .....	42
Rejet d'une perturbation harmonique .....	56
réponse à un échelon.....	11
réponse fréquentielle.....	18
Réponse impulsionnelle.....	10
Réponses indicielle .....	10
retard.....	23
root locus .....	43

**S**

signal discret .....	5
smart sensors.....	62
stabilité.....	11
Systèmes du premier ordre.....	14
Systèmes du second ordre.....	15

**T**

Théorème de Nyquist.....	46
théorème de Shannon.....	19, 24
théorème du retard .....	6
Transformée en z .....	5
transformée unilatérale .....	5
Tustin.....	29

**V**

virgule flottante.....	60
------------------------	----

**Z**

zero holder hold .....	23
zéros.....	11
zoh .....	23

# Commande des systèmes par ordinateur

Yves Briere  
ENSICA

Notes de cours  
2007



# TABLE DES MATIERES

Table des matières .....	1
1. Introduction .....	1
2. Rappels et prérequis .....	3
3. Signaux et systèmes discrets .....	5
3.1 Signaux discrets .....	5
3.1.1 Transformée en z .....	5
3.1.2 Signaux types .....	7
3.2 Systèmes discrets .....	9
3.2.1 Fonction de transfert discrète, définition .....	9
3.2.2 Notation "opérateur de retard $q^{-1}$ " .....	10
3.2.3 Réponse impulsionnelle .....	11
3.2.4 Causalité .....	11
3.2.5 Réponses indicielle .....	11
3.2.6 Régime stationnaire (gain statique) .....	12
3.2.7 Pôles, zéros, stabilité .....	12
3.2.8 Systèmes du premier ordre .....	15
3.2.9 Systèmes du second ordre .....	16
3.2.10 Calcul itératif de la réponse d'un système .....	18
3.2.11 Représentation par équation d'état .....	18
3.3 Réponse fréquentielle des systèmes échantillonnés .....	19
3.3.1 Diagramme de Bode d'un système échantillonné .....	19
3.3.2 Où l'on voit apparaître le théorème de Shannon .....	20
4. Systèmes continus échantillonnés .....	23
4.1 Structure des systèmes commandés par ordinateur .....	23
4.2 Échantillonnage des systèmes continus .....	24
4.3 Effets de l'échantillonnage .....	25
4.4 Où l'on revoit le théorème de Shannon .....	26
4.5 Discrétisation de fonction de transfert continues .....	29
4.5.1 Approximation numérique d'équation différentielle .....	30
4.5.2 Superposition des réponses indicielles .....	32
4.5.3 Correspondances pôles - zéros .....	33
4.5.4 Bilan .....	33
5. Identification des systèmes échantillonnés .....	35
5.1 Identification d'un modèle ARX par la méthode des moindres carrés récurifs .....	35
5.2 L'identification paramétrique en général et en pratique .....	37

5.2.1	Choix du signal d'excitation, traitement des mesures .....	37
5.2.2	Choix d'une structure de modèle.....	38
5.2.3	Choix d'un critère à minimiser.....	38
5.2.4	Choix d'un algorithme.....	38
5.2.5	Validation.....	39
5.2.6	Utilisation de logiciels .....	39
5.3	Identification "simple" .....	39
5.4	Identification non paramétrique .....	41
5.4.1	Estimation de la réponse impulsionnelle .....	41
5.4.2	Estimation de la réponse fréquentielle.....	42
6.	Systèmes bouclés .....	45
6.1	Fonction de transfert en boucle fermée .....	45
6.2	Erreur statique .....	46
6.3	Rejet de perturbations.....	46
6.4	Stabilité des systèmes à simple gain de rétroaction.....	47
6.5	Analyse des systèmes bouclés dans le domaine fréquentiel : stabilité.....	49
6.5.1	Théorème de Nyquist dans le cas où le système est stable en boucle ouverte (critère du revers) .....	50
6.5.2	Théorème de Nyquist dans le cas où le système est instable en boucle ouverte .....	50
6.5.3	Interprétations du lieu de Nyquist.....	50
7.	Méthodes de régulation des systèmes échantillonnés .....	53
7.1	Introduction .....	53
7.2	Structure d'un système commandé par ordinateur, hypothèses.....	54
7.2.1	Discretisation, choix de la période d'échantillonnage.....	54
7.2.2	Discretisation d'un correcteur continu .....	55
7.3	Régulateur PID numérique .....	55
7.3.1	Cas 1 .....	55
7.3.2	Cas 2 .....	58
7.4	Placement de pôles .....	59
7.4.1	Choix des pôles en boucle fermée .....	60
7.4.2	Régulation : calcul de R et S.....	60
7.4.3	Poursuite : calcul de T .....	61
7.5	Poursuite et régulation à objectifs indépendants .....	62
8.	Technologie de la commande par ordinateur.....	65
8.1	Calculateurs et interfaces numériques / analogiques.....	65
8.2	Transport de l'information : les bus et les protocoles .....	67
8.3	Capteurs et actionneurs "intelligents", "smart sensors" .....	68
8.4	La philosophie "hardware in the loop" .....	68
8.5	Le prototypage rapide en automatique .....	68
9.	Annexe .....	69
9.1	Annexe 1 : Résolution de l'équation polynomiale de Bezout.....	69
10.	Bibliographie.....	73
11.	Index.....	1







# 1. INTRODUCTION



Figure 1 : commande d'un systèmes de télémédecine par calculateur, vu par Claude Serre



## 2. RAPPELS ET PREREQUIS

Ce polycopié ne comporte pas de rappels sur les systèmes continus. S'adressant aux étudiants en deuxième année de l'ENSICA il suppose connu les enseignements de première année et de seconde année en automatique et traitement du signal.

En première année le cours de théorie du signal [2] a rendu familier les notions de :

- transformée de Fourier, transformée de Laplace
- corrélation, convolution

Le cours de traitement numérique du signal [3] a abordé en plus :

- la discrétisation et la quantification des signaux.
- la transformée en z et la transformée de Fourier discrète.
- les filtres numériques RIF et RII.

En automatique [4] les bases de la représentation et de l'analyse des systèmes continus ont été posées. On ne reviendra pas dans ce cours sur les notions de :

- modèles mathématique des systèmes linéaires à temps continus à temps invariant (LTI) : fonction de transfert, équation d'état, pôles, zéros et dynamique.
- commandabilité et observabilité.
- stabilité des systèmes en boucle ouverte : pôles, critères de Routh-Hurwitz.
- stabilité des systèmes bouclés : critère de Nyquist, critère de Bode.
- lieu des racines.

En deuxième année les notions suivantes sont acquises [5] :

- synthèse des correcteurs par méthodes fréquentielles (avance et retard de phase, PID)
- commande par retour d'état (placement de pôles), commande Linéaire Quadratique (LQG)
- reconstruction de l'état par Observateur, observateur optimal de Kalman.



## 3. SIGNAUX ET SYSTEMES DISCRETS

### 3.1 Signaux discrets

#### 3.1.1 Transformée en z

Pour une approche mathématiquement rigoureuse de représentation des signaux discrets par la transformée en z consulter [3] et sa bibliographie. Dans ce cours on se cantonne à une approche utilitariste des outils et méthodes mathématiques tout en essayant de ne pas contredire fondamentalement la rigueur mathématique.

Un signal discret  $s(k)$  est une séquence de nombres réels appelés échantillons. L'indice  $k$  est généralement associé à un instant d'échantillonnage :  $t = k.T_e$ . La période d'échantillonnage  $T_e$  est supposée constante. Par définition la transformée en z du signal  $s(k)$  est la fonction de la variable  $z$ , notée abusivement  $s(z)$  :

$$s(z) = Z(s(k)) = \sum_{k=0}^{\infty} s(k) \cdot z^{-k} \quad (1)$$

La transformée définie ici est plus précisément appelée transformée unilatérale puisque les termes correspondant à  $k < 0$  ne sont pas pris en compte. Ceci n'a pas d'importance puisque les signaux que nous utilisons sont systématiquement causaux<sup>1</sup> :  $s(k) = 0$  pour  $k < 0$ .

Il y a bien évidemment un lien avec la transformée de Laplace et la transformée de Fourier (voir [3])

La fonction de la variable complexe  $s(z)$  est une série et ne converge pas forcément pour toute valeur de  $z$ . En toute rigueur il faudrait associer le rayon de convergence à chaque fonction en  $z$  :  $\{s(z) \text{ converge pour } |z| > \alpha\}$ .

Les automaticiens pour qui la transformée en z n'est bien souvent qu'une façon bien commode de regrouper des termes de récurrence utilisent souvent la pseudo-variable<sup>2</sup>  $q$ .

A partir d'une transformée en z  $s(z)$  il est possible de retrouver la séquence  $s(k)$  correspondante en appliquant la transformation inverse, dont on ne donnera pas la formule ici. A la place de faire appel à cette formule compliquée il est possible d'utiliser d'autres méthodes pour retrouver la séquence  $s(k)$  lorsque la fonction  $s(z)$  est donnée sous forme de fraction rationnelle :

<sup>1</sup> Pour une justification du terme « causal » voir page 11

<sup>2</sup> Voir page 10

- division polynomiale
- décomposition de la fraction rationnelle en éléments simples puis utilisation de transformations élémentaires telles que celles données dans le Tableau 2

La propriété fondamentale de la transformée en z est qu'elle est linéaire (la transformée en z d'une composition linéaire de signaux est la composition linéaire de la transformée en z de chaque signal). Les autres propriétés fondamentales sont données dans le Tableau 2.

Tableau 1 : propriétés de la transformée en z

1	$s(z) = Z(s(k)) = \sum_{k=0}^{\infty} s(k) \cdot z^{-k}$
2	$Z(\alpha \cdot s_1(k) + \beta \cdot s_2(k)) = \alpha \cdot Z(s_1(k)) + \beta \cdot Z(s_2(k))$
3	$Z(k \cdot s(k)) = -z \cdot \frac{ds(z)}{dz}$
4	$Z(c^k \cdot s(k)) = s\left(\frac{z}{c}\right)$
5	$Z(s(k-1)) = s(k=-1) + z^{-1} \cdot s(z)$
6	$Z(s(k+1)) = z \cdot s(z) - z \cdot s(k=0)$
7	$s(k=0) = \lim_{z \rightarrow \infty} (s(z))$
8	$s(k \rightarrow \infty) = \lim_{z \rightarrow 1} \left( \frac{z-1}{z} \cdot s(z) \right)$

Remarque 1 : la propriété 5 ne prend pas en compte le fait que le signal est causal : en pratique  $s(k=-1) = 0$ .

Remarque 2 : la propriété 5 justifie le fait que le terme  $z^{-1}$  soit appelé "opérateur de retard unitaire", cette propriété est aussi appelée "théorème du retard".

Remarque 3 : les propriétés 7 et 8 sont données sous réserve de convergence.

Transformées inverses

### Exemple 1 : transformée inverse par division polynomiale

On considère un signal  $s(k)$  dont la transformée en z est donnée par la fraction rationnelle :

$$S(z) = \frac{0.1 \cdot z^2 + 0.1 \cdot z}{z^3 - 2.6 \cdot z^2 + 2.2 \cdot z - 0.6} \quad (3.1)$$

L'objectif est d'obtenir  $S(z)$  sous forme polynomiale :

$$S(z) = x_0 + x_1 \cdot z^{-1} + x_2 \cdot z^{-2} + \dots \text{etc.} \quad (3.2)$$

Cette suite est généralement infinie, la réponse temporelle est donnée par la succession des termes  $\{x_i\}$ . En traitant l'équation (3.1) comme une simple division l'algorithme de division classique (celui appris en CM2) permet d'obtenir par récurrence tous les termes de la suite (3.2) :

$0.1 \cdot z^2$	$+ 0.1 \cdot z$				$z^3 - 2.6 \cdot z^2 + 2.2 \cdot z - 0.6$
$0.1 \cdot z^2$	$- 0.26 \cdot z$	$+ 0.22$	$- 0.06 \cdot z^{-1}$		
$0$	$0.36 \cdot z$	$- 0.22$	$0.06 \cdot z^{-1}$		
	$0.36 \cdot z$	$- 0.936$	$0.792 \cdot z^{-1}$	$- 0.216 \cdot z^{-2}$	
$0$	$0.716$	$- 0.732 \cdot z^{-1}$	$\dots$	$\dots$	

Les éléments successifs de la suite  $s(k)$  sont donc  $s(k) = \{0, 0.1, 0.36, 0.716, \dots\}$ . Pour obtenir les termes suivants il faudrait continuer la division.

**Exemple 2 : transformée inverse par décomposition en éléments simples**

La deuxième méthode présentée consiste à décomposer la fraction rationnelle (3.1) en éléments simples puis d'utiliser les tables de transformées en  $z$  (Tableau 2). Pour notre exemple cela donne :

$$S(z) = \frac{0.1 \cdot z^2 + 0.1 \cdot z}{z^3 - 2.6 \cdot z^2 + 2.2 \cdot z - 0.6} = \frac{-z}{z-1} + \frac{0.5 \cdot z}{(z-1)^2} + \frac{z}{z-0.6}$$

La réponse temporelle est donc composée des trois éléments :

$$s(k) = -(1 \ 1 \ 1 \ \dots) + 0.5 \cdot (1 \ 2 \ 3 \ \dots) + (1 \ 0.6 \ 0.6^2 \ \dots)$$

$$\text{soit : } s(k) = \begin{cases} s(0) = -1 + 0.5 + 1 \\ s(2) = -1 + 0.5 \cdot 2 + 0.6 \cdot 0.5 \\ s(3) = -1 + 0.5 \cdot 3 + 0.6^2 \\ s(4) = \dots \end{cases}$$

**3.1.2 Signaux types**

Le Tableau 2 donne les transformée en  $z$  de certains signaux fondamentaux. La Figure 2 représente l'allure de quelques signaux types.

Tableau 2 : transformée en z élémentaires

$s(k)$	$S(z)$
signal temporel	Transformée en z
$\delta(k)$ : impulsion unité	1
$u(k)$ : échelon unité	$\frac{z}{z-1}$
$k \cdot u(k)$	$\frac{z}{(z-1)^2}$
$c^k \cdot u(k)$	$\frac{z}{z-c}$
$k \cdot a^k \cdot u(k)$	$\frac{a \cdot z}{(z-a)^2}$
$\sin(\omega \cdot k) \cdot u(k)$	$\frac{z \cdot \sin(\omega)}{z^2 - 2 \cdot z \cdot \sin(\omega) + 1}$
$\cos(\omega \cdot k) \cdot u(k)$	$\frac{z \cdot (z - \cos(\omega))}{z^2 - 2 \cdot z \cdot \sin(\omega) + 1}$
$a^k \cdot \sin(\omega \cdot k) \cdot u(k)$	$\frac{a \cdot z \cdot \sin(\omega)}{z^2 - 2 \cdot a \cdot \cos(\omega) \cdot z + a^2}$
$a^k \cdot \cos(\omega \cdot k) \cdot u(k)$	$\frac{z \cdot (z - a \cdot \cos(\omega))}{z^2 - 2 \cdot a \cdot \cos(\omega) \cdot z + a^2}$



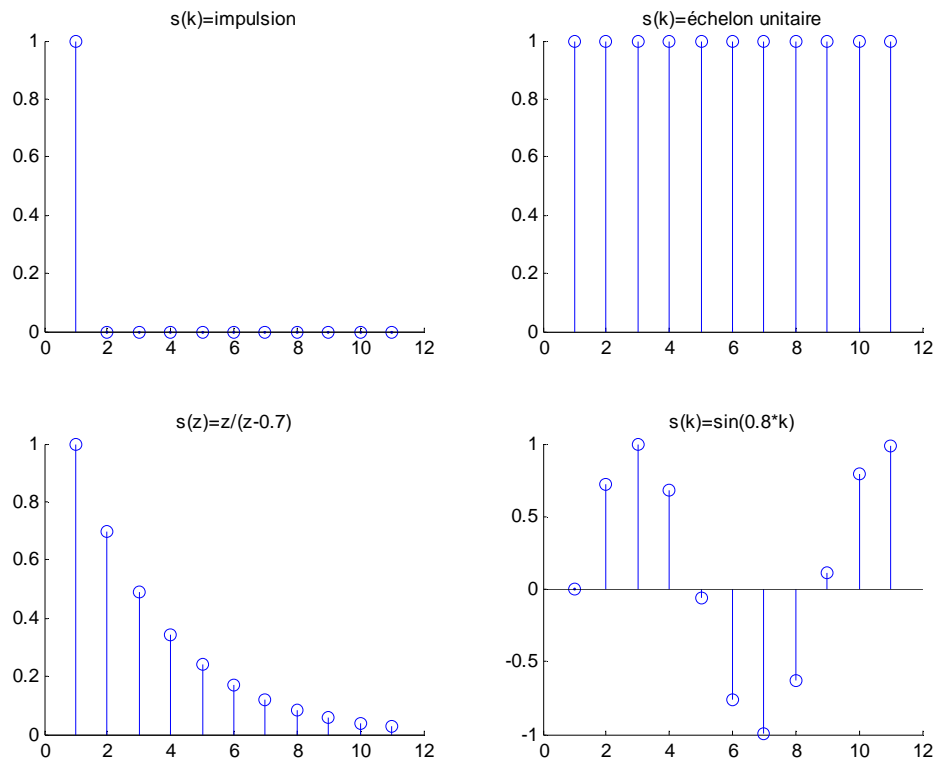


Figure 2 : réponses de quelques signaux types

### 3.2 Systèmes discrets

#### 3.2.1 Fonction de transfert discrète, définition

On considère maintenant un système discret linéaire et invariant dans le temps (LTI) représenté sur la Figure 3.

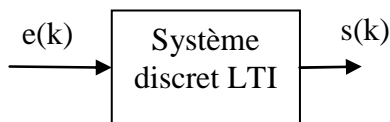


Figure 3 : système discret, notations

La relation entre la séquence d'entrée et la séquence de sortie est linéaire et définie par un ensemble de coefficients constants  $a_{1:n}$  et  $b_{1:m}$  :

$$\begin{aligned} s(k) + a_1 \cdot s(k-1) + a_2 \cdot s(k-2) + \dots + a_n \cdot s(k-n) = \\ b_0 \cdot e(k) + b_1 \cdot e(k-1) + b_2 \cdot e(k-2) + \dots + b_m \cdot e(k-m) \end{aligned} \quad (2)$$

On peut appliquer la transformée en  $z$  à chaque membre de l'équation puis en appliquant en récurrence la propriété 5 (théorème du retard) on obtient la fonction de transfert discrète du système :

$$\frac{s(z)}{e(z)} = F(z) = \frac{b_0 + b_1 \cdot z^{-1} + b_2 \cdot z^{-2} + \dots + b_m \cdot z^{-m}}{1 + a_1 \cdot z^{-1} + a_2 \cdot z^{-2} + \dots + a_n \cdot z^{-n}} \quad (3)$$

Remarque 1 : la fonction de transfert précédente est normalisée en prenant  $a_0 = 1$ .

Remarque 2 : on remarquera que la fonction de transfert est donnée en  $z^{-1}$  plutôt qu'en  $z$ . Cela est justifié par le fait que celle-ci traduit l'équation d'évolution du système physique causal (2).

Le script Matlab permettant de créer un objet Matlab correspondant à une fonction de transfert discrète est donné ci-dessous.

#### Matlab

```
% Création d'un système linéaire discret :
Te=0.1;
sysd = tf([b0 b1 b2],[1 a0 a2 a3],Te,'variable','z^-1');
```

On notera l'utilisation de l'argument 'variable', 'z^-1'.

### 3.2.2 Notation "opérateur de retard $q^{-1}$ "

La fonction de transfert est une fonction de la variable complexe  $z$ . Il faudrait donc en toute rigueur l'associer à son rayon de convergence. En pratique on ne s'embarrasse jamais de préciser cette zone de convergence. C'est pourquoi on trouve souvent la notation utilisant l'opérateur  $q^{-1}$  de décalage à la place de la variable complexe  $z^{-1}$  (4).  $F(q)$  n'est alors plus à proprement parler une fonction mais on lui appliquera les mêmes manipulations algébriques qu'à  $F(z)$ .

$$F(q) = \frac{b_0 + b_1 \cdot q^{-1} + b_2 \cdot q^{-2} + \dots + b_m \cdot q^{-m}}{1 + a_1 \cdot q^{-1} + a_2 \cdot q^{-2} + \dots + a_n \cdot q^{-n}} \quad (4)$$

Attention : selon les auteurs l'opérateur qui remplace  $z^{-1}$  est noté  $q$  (Matlab) ou  $q^{-1}$  (Landau [1]). Dans la suite du document c'est cette dernière convention qui sera appliquée.

L'utilisation de l'opérateur de retard  $q^{-1}$  à la place de la variable complexe  $z^{-1}$  permet de traiter du cas des fonctions de transfert à coefficients non constant, cas non prévu par la transformation en  $z$ .

### 3.2.3 Réponse impulsionnelle

Essayons d'appliquer au système un signal d'entrée particulier : une impulsion unitaire. La transformée en  $z$  du système est obtenue en multipliant la fonction de transfert par la transformée en  $z$  de l'impulsion unitaire : 1. On obtient :

$$s(z) = \frac{b_0 + b_1 \cdot z^{-1} + b_2 \cdot z^{-2} + \dots + b_m \cdot z^{-m}}{1 + a_1 \cdot z^{-1} + a_2 \cdot z^{-2} + \dots + a_n \cdot z^{-n}} \times 1 \quad (5)$$

Cela justifie le fait que la fonction de transfert du système peut aussi s'appeler "réponse impulsionnelle".

#### Matlab

```
% Réponse impulsionnelle d'un système :
[y,t] =impulse(sysd);
```

### 3.2.4 Causalité

Un système est dit causal si les sorties ne dépendent que du passé et non pas du futur. La conséquence est que la réponse impulsionnelle de  $F(z)$  ne peut pas comporter de terme non nuls pour  $k < 0$ . Cela justifie la confusion sémantique entre "système causal" et "signal causal", c'est-à-dire ne comportant pas de termes non nuls pour  $k < 0$ .

Attention : à la différence des systèmes continus un système physique (causal) discret peut très bien être modélisé avec un degré  $m$  supérieur au degré  $n$  : le système peut avoir plus de zéros que de pôles.

La transformée en  $z$  représentant des signaux échantillonnés elle peut tout aussi bien représenter les mesures prises à différents endroits d'une poutre. Le temps étant remplacé ici par une abscisse la notion de causalité ne s'applique pas ! Attention : beaucoup des propriétés de systèmes échantillonnés que l'on étudie sont liés à la causalité, prudence donc si l'on applique ces méthodes à d'autres domaines !

### 3.2.5 Réponses indicielle

Pour caractériser des systèmes on s'intéresse souvent à leur réponse impulsionnelle ou, plus souvent, à leur réponse indicielle. Pour calculer la réponse

à un échelon d'un système il faut donc multiplier la fonction de transfert par la transformée en z d'un échelon unitaire :

$$s(z) = \frac{b_0 + b_1 \cdot z^{-1} + b_2 \cdot z^{-2} + \dots + b_m \cdot z^{-m}}{1 + a_1 \cdot z^{-1} + a_2 \cdot z^{-2} + \dots + a_n \cdot z^{-n}} \times \frac{1}{1 - z^{-1}} \quad (6)$$

Il faudrait ensuite calculer la transformée en z inverse de s(z). En pratique, si l'on doit faire le calcul à la main on décompose en éléments simples puis on utilise des tables. Avec Matlab c'est plus simple. Voici par exemple le script pour obtenir les 20 premiers échantillons de la réponse indicielle d'un système :

**Matlab**

```
% Réponse indicielle d'un système :
[y,t] =step(sysd,20);
```

### 3.2.6 Régime stationnaire (gain statique)

Le gain statique est obtenu en appliquant une entrée indicielle et en calculant la valeur du régime permanent. En appliquant le théorème de la valeur finale il vient :

$$F_0 = \lim_{z \rightarrow 1} \left( \frac{z-1}{z} \cdot \frac{b_0 + b_1 \cdot z^{-1} + b_2 \cdot z^{-2} + \dots + b_m \cdot z^{-m}}{1 + a_1 \cdot z^{-1} + a_2 \cdot z^{-2} + \dots + a_n \cdot z^{-n}} \times \frac{1}{1 - z^{-1}} \right) \quad (7)$$

Soit :

$$F_0 = \frac{\sum_{i=0}^{i=m} b_i}{1 + \sum_{i=1}^{i=n} a_i} \quad (8)$$

Le gain statique est obtenu donc simplement en faisant le quotient de la somme des coefficients.

### 3.2.7 Pôles, zéros, stabilité

Comme pour les systèmes continus la dynamique et la stabilité des systèmes dépend des propriétés du dénominateur de la fonction de transfert.

$$F(z) = \frac{e_0 \cdot (1 + e_1 \cdot z^{-1}) \cdot (1 + e_2 \cdot z^{-1}) \cdots (1 + e_m \cdot z^{-1})}{(1 + c_1 \cdot z^{-1}) \cdot (1 + c_2 \cdot z^{-1}) \cdots (1 + c_n \cdot z^{-1})} \quad (9)$$

Les coefficients  $c_i$  sont les pôles de la fonction de transfert, les coefficients  $e_i$  sont les zéros. La fonction de transfert étant au départ composée de coefficients purement réels les coefficients  $c_i$  sont soit réels soit imaginaires par paires conjuguées.

Une fonction de transfert est donc aussi bien définie par la donnée de ses pôles et de ses zéros. Le script Matlab pour créer un système ainsi est donné ci-dessous :

#### Matlab

```
% Système discret défini par ses pôles et zéros :
Te=0.1;
sysd=zpk([c1 c2 c3],[e1 e2],e0,Te);
```

Si l'on décompose la fonction de transfert en éléments simples on obtient une somme d'éléments du type :

$$F(z) = \frac{d_1}{1 + c_1 \cdot z^{-1}} + \frac{d_2}{1 + c_2 \cdot z^{-1}} + \cdots + \frac{d_n}{1 + c_n \cdot z^{-1}} \quad (10)$$

Un système linéaire temps invariant est dit stable si sa réponse impulsionnelle tend asymptotiquement vers 0. Compte tenu de (10) cela nécessite que chaque pôle  $c_i$  soit de module inférieur à 1.

**Propriété** : un système discret LTI est stable si et seulement si tous ses pôles ont leur modules inférieur ou égal à 1.

#### Critère algébrique attestant de la stabilité : critère de Routh

On rappelle que le critère de Routh permet de prédire si les racines d'un polynôme sont toutes à partie réelle négative, donc stables. Énonçons le théorème de Routh dans le cas d'un polynôme  $P'(w)$  :

$$P'(w) = a_0 + a_1 \cdot w + a_2 \cdot w^2 + \cdots + a_n \cdot w^n \quad (11)$$

Dans un premier temps on construit le tableau :

$$\begin{array}{cccccc|c}
 \beta_{n,0} & \beta_{n,1} & \beta_{n,2} & \beta_{n,q-1} & \beta_{n,q} & 0 & \\
 \beta_{n-1,0} & \beta_{n-1,1} & \beta_{n-1,2} & \beta_{n-1,q-1} & \beta_{n-1,q-1} & 0 & \\
 \beta_{n-2,0} & \beta_{n-2,1} & \beta_{n-2,2} & \beta_{n-2,q-1} & \beta_{n-2,q-1} & & \\
 & & & & & & \\
 \beta_{2,0} & \beta_{2,1} & 0 & & & & \\
 \beta_{1,0} & 0 & & & & & \\
 \beta_{0,0} & 0 & & & & & 
 \end{array} \quad (12)$$

Puis on vérifie que tous les premiers éléments de la première colonne ont le même signe : cela garantit que tous les pôles de  $P'(w)$  ont leur partie réelle négative.

Les éléments du tableau sont construits par récurrence ligne par ligne en appliquant les formules suivantes :

Pour  $i = n$  et  $i = n - 1$  :

$$\begin{cases} \beta_{n,j} = a_{n-2j} \\ \beta_{n-1,j} = a_{n-1-2j} \end{cases} \quad (13)$$

Et pour  $i < n-1$  :

$$\beta_{i,j} = \frac{\beta_{i+1,0} \cdot \beta_{i+2,j+1} - \beta_{i+1,j+1} \cdot \beta_{i+2,0}}{\beta_{i+1,0}} \quad (14)$$

Dans le cas qui nous intéresse, un système échantillonné, on cherche plutôt à vérifier que les pôles du polynôme  $P(z)$  sont de module inférieur à un. L'idée est d'effectuer un changement de variable  $z \rightarrow w$  pour pouvoir appliquer le critère de Routh.

$$P'(w) = (1-w)^n \cdot P\left(\frac{1+w}{1-w}\right) \quad (15)$$

Si l'on effectue la transformation (15) alors le fait que  $P'(w)$  ait toutes ses racines à partie réelle négative garantit que  $P(z)$  a toutes ses racines dans le cercle unité. Pour vérifier la stabilité de  $P(z)$  il suffit donc d'effectuer le changement de variable (15) puis d'appliquer le critère de Routh conventionnel (12 13 14).

### Critère algébrique attestant de la stabilité : critère de Jury

Le critère de Jury s'apparente au critère de Routh. Nous le donnons pour un polynôme de degré 2, 3 et 4 du type :

$$P(z) = a_0 + a_1 \cdot z + a_2 \cdot z^2 + \dots + a_n \cdot z^n \quad (16)$$

Les conditions de stabilité sont données ci-dessous (ces conditions sont données dans le cas  $a_n > 0$ , il suffit de multiplier tous les termes par  $-1$  si cette condition n'est pas remplie par  $P(z)$ ) :

$$\begin{aligned}
 n = 2 & \begin{cases} a_0 + a_1 + a_2 > 0 \\ a_0 - a_1 + a_2 > 0 \\ -a_0 + a_2 > 0 \end{cases} \\
 n = 3 & \begin{cases} a_0 + a_1 + a_2 + a_3 > 0 \\ -a_0 + a_1 - a_2 + a_3 > 0 \\ -|a_0| + a_3 > 0 \\ a_0 a_2 - a_1 a_3 - a_0^2 + a_3^2 > 0 \end{cases} \\
 n = 4 & \begin{cases} a_0 + a_1 + a_2 + a_3 + a_4 > 0 \\ a_0 - a_1 + a_2 - a_3 + a_4 > 0 \\ a_4^2 - a_0^2 - |a_0 a_3 - a_1 a_4| > 0 \\ (a_0 - a_4)^2 (a_0 - a_2 + a_4) + (a_1 - a_3)(a_0 a_3 - a_1 a_4) > 0 \end{cases}
 \end{aligned} \quad (17)$$

### 3.2.8 Systèmes du premier ordre

On donne dans la Figure 4 les réponses indicielles de systèmes du premier ordre ainsi que la localisation des pôles dans le plan complexe. Le cercle de rayon unité représente la zone dans laquelle les pôles doivent se situer pour assurer la stabilité. On remarque sur les deux premières figures l'influence de la place du zéro par rapport au point  $(1,0)$  pour la rapidité.

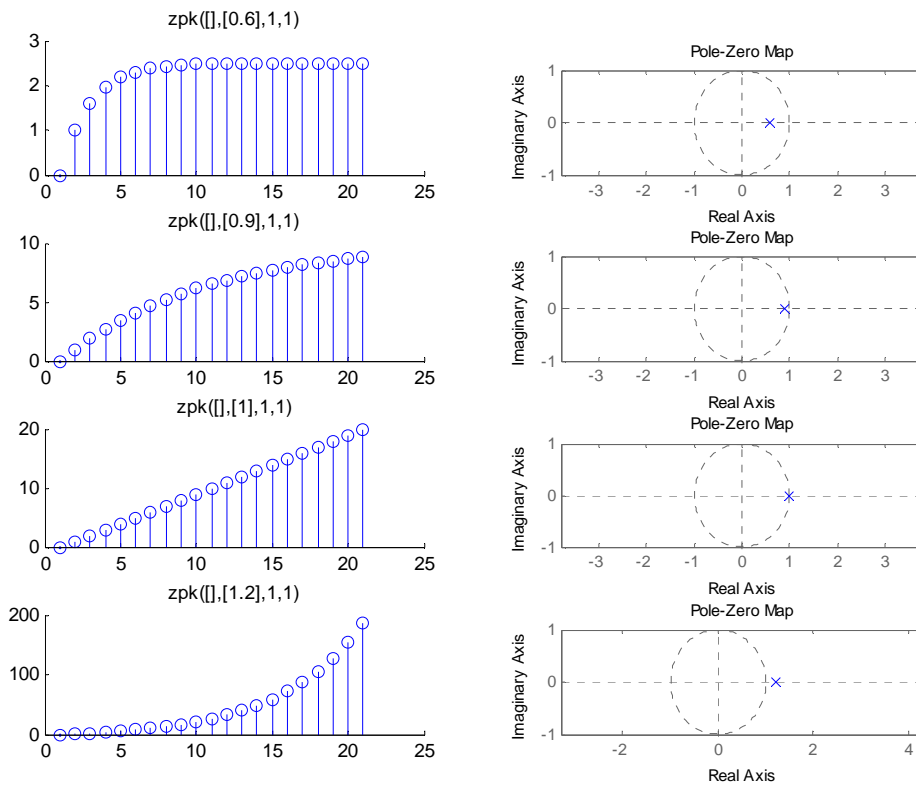


Figure 4 réponses indicielles de systèmes du premier ordre

### 3.2.9 Systèmes du second ordre

On donne dans la Figure 6 les réponses indicielles de systèmes du second ordre ainsi que la localisation des pôles dans le plan complexe. La rapidité et l'amortissement dépend de la position des pôles dans le cercle unité. La Figure 5 représente les courbes iso-amortissement et iso-pulsation des systèmes du second ordre.



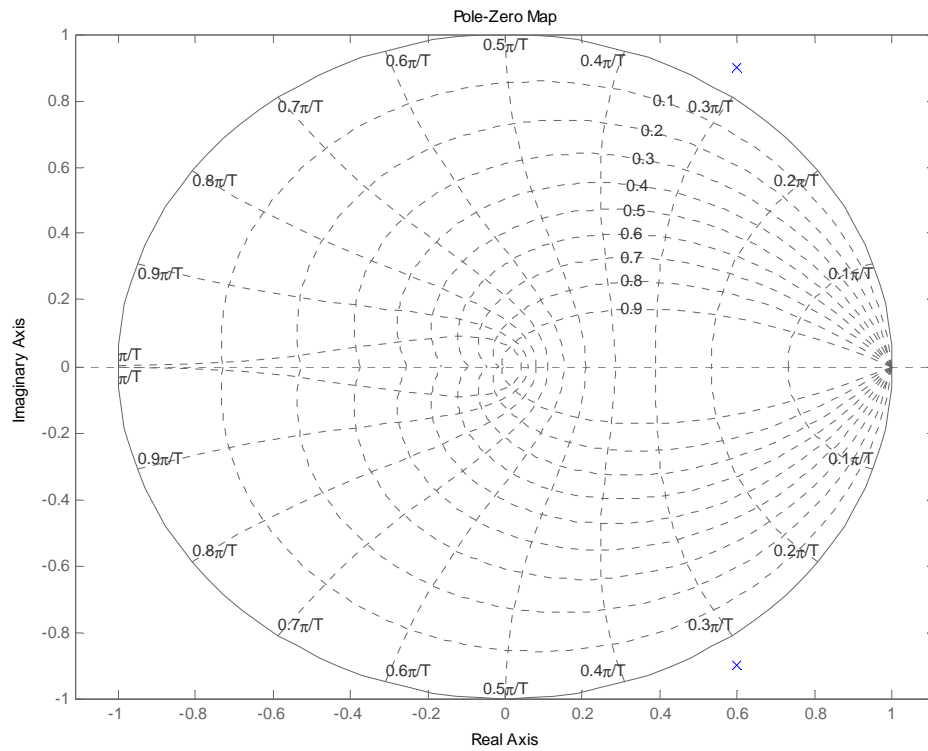


Figure 5 : courbes iso-amortissement et iso-pulsation des systèmes discrets du second ordre

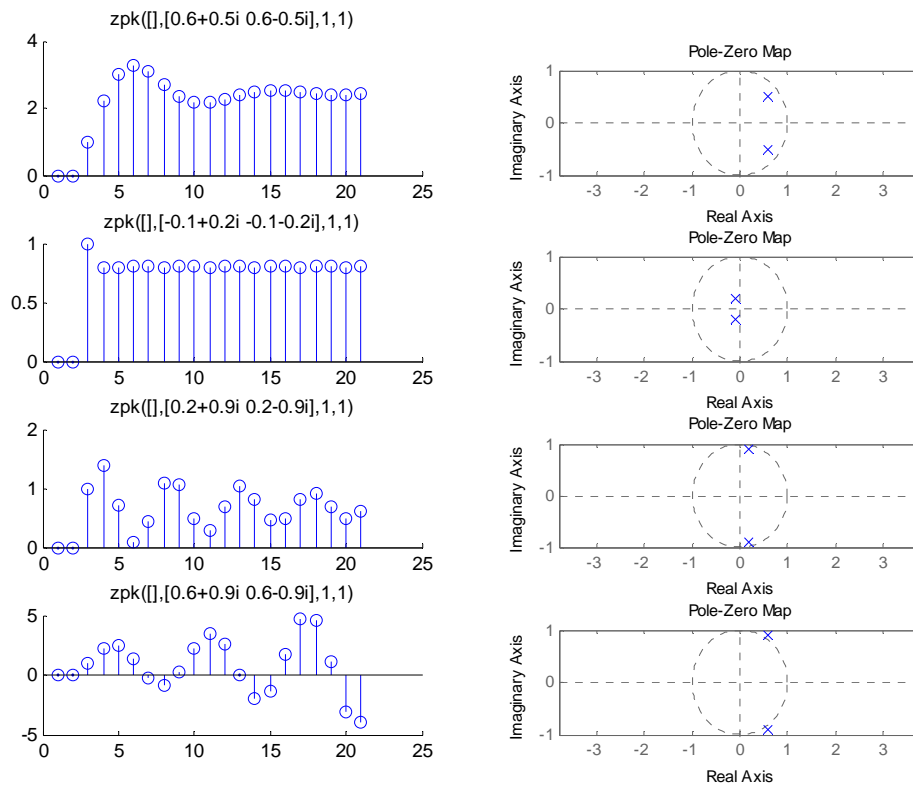


Figure 6 : réponses de systèmes du second ordre

### 3.2.10 Calcul itératif de la réponse d'un système

Si l'on dispose de la fonction de transfert d'un système et que l'on connaît les valeurs successives  $e(k)$  du signal d'entrée il est facile de calculer itérativement la sortie  $y(k)$  du système par l'équation de récurrence (2). Il faut initialiser par les  $n$  valeurs initiales de  $s(k)$ , prises généralement égales à 0.

### 3.2.11 Représentation par équation d'état

Un système discret peut aussi être représenté par un ensemble d'équations d'état comme dans le domaine continu (52). Cette représentation est naturellement adaptée aux systèmes multi-entrées multi-sorties. On définit alors un état interne  $x(k)$  ainsi qu'une fonction d'évolution  $f$  et une équation de mesure  $h$ .

$$\begin{cases} x(k+1) = f(x(k), u(k)) \\ s(k) = h(x(k), u(k)) \end{cases} \quad (18)$$

### 3.3 Réponse fréquentielle des systèmes échantillonnés

#### 3.3.1 Diagramme de Bode d'un système échantillonné

Lorsqu'on étudie la réponse fréquentielle des systèmes continus on considère des entrées particulières sinusoïdales :

$$\begin{cases} e(t) = \cos(2 \cdot \pi \cdot f \cdot t) + j \cdot \sin(2 \cdot \pi \cdot f \cdot t) = e^{j \cdot 2 \cdot \pi \cdot f \cdot t} \\ f \in [0 \quad \infty] \end{cases} \quad (19)$$

De même, pour des systèmes échantillonnés on considèrera des entrées sinusoïdales échantillonnées :

$$\begin{cases} e(k) = \cos(2 \cdot \pi \cdot f \cdot k) + j \cdot \sin(2 \cdot \pi \cdot f \cdot k) = e^{j \cdot 2 \cdot \pi \cdot f \cdot k} \\ f \in [0 \quad 1] \end{cases} \quad (20)$$

La sortie est aussi un signal échantillonné sinusoïdal de gain  $A(f)$  et de phase  $\Phi(f)$  variables et dépendant de  $f$ ; la fréquence réduite.

$$\begin{cases} s(k) = A(f) \cdot e^{j \cdot (2 \cdot \pi \cdot f \cdot k + \Phi(f))} \\ f \in [0 \quad 1] \end{cases} \quad (21)$$

Le nombre complexe, noté abusivement  $F(f)$ , d'amplitude  $A(f)$  et de phase  $\Phi(f)$  est obtenu à partir de  $F(z)$  en remplaçant  $z$  par  $e^{j \cdot 2 \cdot \pi \cdot f}$  :

$$\begin{cases} F(f) = F(z = e^{j \cdot 2 \cdot \pi \cdot f}) \\ f \in [0 \quad 1] \end{cases} \quad (22)$$

Le diagramme représentant le gain et la phase en fonction de la fréquence réduite s'appelle aussi un diagramme de Bode. L'échelle des fréquences est généralement linéaire de 0 à 0.5 (puisque le diagramme est nécessairement symétrique). La Figure 7 représente par exemple le diagramme de Bode d'un système du second ordre.

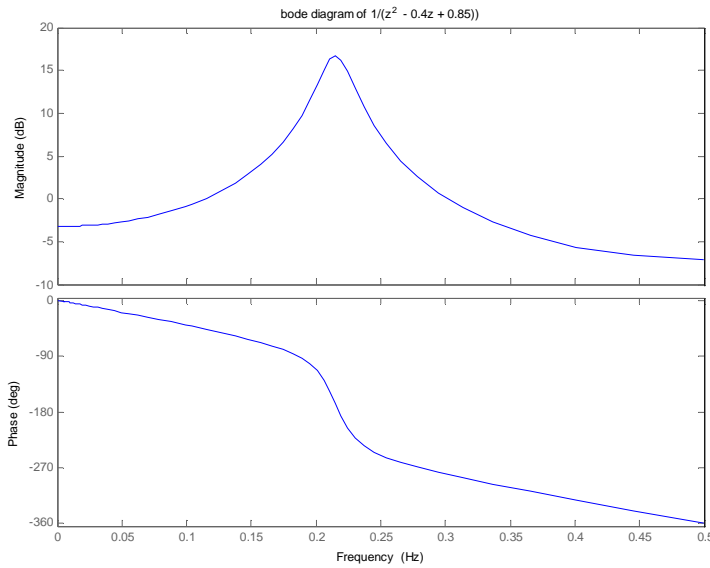


Figure 7 : exemple du diagramme de Bode d'un système discret du second ordre

### 3.3.2 Où l'on voit apparaître le théorème de Shannon

Le changement de variable  $z = e^{j2\pi f}$  n'est pas bijectif puisque  $e^{j2\pi f} = e^{j2\pi(f+k)}$  avec  $k$  entier. Si l'on fait le lien entre domaine continu et domaine fréquentiel on peut remarquer que le changement de variable  $z = e^{j2\pi f}$  fait correspondre une zone du plan complexe continu à une zone du plan complexe discret. La transformation est bijective si l'on se contente de la zone du plan complexe continu contenu entre les droites d'ordonnée 0 et 1.

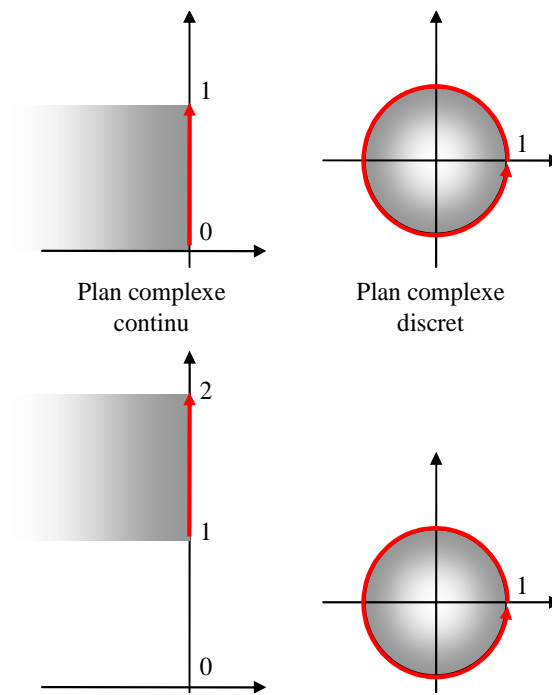


Figure 8 : correspondance entre plan complexe continu et plan complexe discret.



## 4. SYSTEMES CONTINUS ECHANTILLONNES

### 4.1 Structure des systèmes commandés par ordinateur

Les systèmes commandés par ordinateurs sont presque toujours des systèmes physiques continus par nature. Le ordinateur traite par contre les entrées et les sorties de manière discrète avec une période d'échantillonnage  $T_e$ . Les éléments importants, du point de vue de l'automaticien sont donc :

- le convertisseur analogique / numérique CAN ou du moins le système qui fournit séquentiellement les mesures au ordinateur à la période  $T_e$ .
- le ordinateur lui-même. Il doit effectuer l'acquisition des mesures, le traitement des informations, l'envoi des consignes dans un temps inférieur à  $T_e$ .
- le convertisseur numérique / analogique CNA.
- le système physique.

On suppose, pour faire simple et pour pouvoir utiliser la panoplie des outils mathématiques, que les systèmes sont échantillonnés à la même période  $T_e$  et au même moment.

La Figure 9 montre le système physique commandé par le ordinateur, dans sa version simpliste.

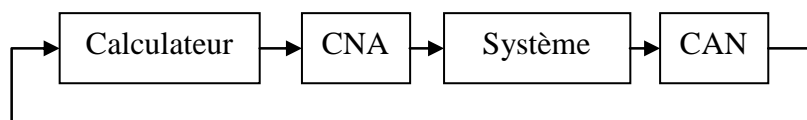


Figure 9 : système commandé par ordinateur

Au XXI<sup>ème</sup> siècle les choses sont quand même parfois un petit peu plus compliquées :

- le ordinateur fournit bien souvent un signal de commande numérique (transporté par bus informatique USB, RS232, CAN, Ethernet, etc...) directement interprété par l'actionneur. On parle "d'actionneur intelligent".
- de même le "capteur intelligent" fournit aussi bien souvent la mesure sous forme d'un signal numérique.
- Les signaux sont transportés par des bus informatiques qui n'ont pas que ça à faire : il peut y avoir du retard.
- de même les capteurs et actionneurs sont rarement tous seuls, ce qui se traduit par des périodes d'échantillonnage différentes, et parfois non synchrones...

- enfin le ordinateur peut aussi avoir d'autres choses à faire...

Ceci étant dit nous allons pour le moment nous restreindre au cas représenté sur la Figure 9 avec des instants d'échantillonnage synchrones et régulièrement espacés d'un temps  $T_e$ .

## 4.2 Echantillonnage des systèmes continus

On dispose généralement de la fonction de transfert continue (notée en utilisant la variable complexe  $s$  ou  $p$ )  $F(s)$  du système continu. Le problème est d'obtenir la fonction de transfert discrète du système échantillonné de manière à pouvoir synthétiser un correcteur. Le système continu est associé au convertisseur numérique – analogique qu'il faut prendre en compte.

Les convertisseurs numérique analogiques peuvent généralement se représenter par des "échantillonneurs – bloqueurs d'ordre 0" ou BOZ. Cela signifie que la commande est maintenue à une valeur constante pendant toute la durée du temps d'échantillonnage.

Si l'on applique une commande discrète impulsionnelle à un BOZ il va maintenir la commande 1 pendant la durée  $T_e$  puis revenir à zéro (Figure 10).

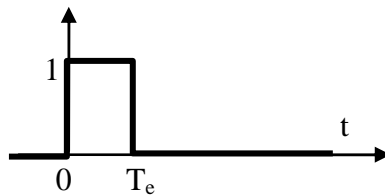


Figure 10 : réponse impulsionnelle d'un bloqueur d'ordre 0

La réponse impulsionnelle  $BOZ(t)$  est donc composée d'un échelon unitaire suivi d'un échelon unitaire de signe opposé et décalé d'un temps  $T_e$ . La transformée de Laplace de la réponse impulsionnelle du bloqueur est donnée par la formule (23) :

$$BOZ(s) = \frac{1}{s} - \frac{e^{-jT_e}}{s} \quad (23)$$

Du point de vue du ordinateur il faut donc prendre en compte le système continu de fonction de transfert  $F_{BOZ}(s)$  que l'on devra échantillonner :

$$F_{BOZ}(s) = \frac{1 - e^{-jT_e}}{s} \cdot F(s) \quad (24)$$



$$F_{\text{BOZ}}(z) = Z^{-1} \left( \frac{1 - e^{-jT_e}}{s} \cdot F(s) \right) \quad (25)$$

En pratique l'obtention de la transformée en z d'un système échantillonné par bloqueur d'ordre 0 se fait à l'aide de tables. On donne quelques exemples fondamentaux dans le Tableau 3 ci-dessous.

Tableau 3 : transformée en z de systèmes continus échantillonnés

F(s)	F <sub>BOZ</sub> (z)
1	1
$\frac{1}{s}$	$\frac{T_e \cdot z^{-1}}{1 - z^{-1}}$
$\frac{1}{1 + T \cdot s}$	$\begin{cases} \frac{b_1 \cdot z^{-1}}{1 + a_1 \cdot z^{-1}} \\ b_1 = 1 - e^{-T_e/T} \quad a_1 = -e^{-T_e/T} \end{cases}$
$\frac{e^{-Ls}}{1 + T \cdot s}$	$\begin{cases} \frac{b_1 \cdot z^{-1} + b_2 \cdot z^{-2}}{1 + a_1 \cdot z^{-1}} \quad a_1 = -e^{-T_e/T} \\ b_1 = 1 - e^{-(L-T_e)/T} \quad b_2 = e^{-T_e/T} (e^{L/T} - 1) \end{cases}$
$L < T_e$	

Si l'on dispose de Matlab c'est bien sûr encore plus simple. On utilise la fonction c2d avec l'argument 'zoh' pour "zero holder hold" (bloqueur d'ordre zéro).

#### Matlab

```
% Définition d'un système continu :
sysc=tf(1,[1 1]);
% Obtention de la transformée en z du système échantillonné à Te
Te=0.1;
sysd=c2d(sysc,Te,'zoh');
% Notation en z^-1
sysd.variable='z'
```

### 4.3 Effets de l'échantillonnage

Le convertisseur numérique a deux effets (néfastes) importants : l'effet de retard et l'effet de quantification.

L'effet de retard se voit dans (24) : on peut dire en première approximation que l'échantillonnage se traduit par un retard pur de  $T_e/2$ . Cela a donc, si ce temps

d'échantillonnage est non négligeable par rapport aux constantes de temps du système, un effet déstabilisant.

Pour illustrer l'effet de quantification supposons par exemple que l'on utilise un convertisseur analogique/numérique 8 bits 0-10V. Le plus petit incrément que le convertisseur peut distinguer est de  $10V/2^8 = 40mV$ . Si le signal utile varie physiquement entre 5V et 5,4V on n'aura que dix incréments utilisables... Ce phénomène n'entre pas dans la panoplie des phénomènes modélisables par la transformée en z. En pratique il est donc souvent nécessaire d'étudier le système et de synthétiser la commande numérique en négligeant ce phénomène puis de vérifier a posteriori qu'il n'a pas un effet trop néfaste.

#### 4.4 Où l'on revoit le théorème de Shannon

Le théorème de Shannon est bien connu en ce qui concerne l'échantillonnage de signaux discrets : on sait que le signal ne doit pas comporter de composante de fréquence supérieure à  $f_e/2$  si l'on ne veut pas observer de repliement. On doit aussi prendre des précautions si l'on veut travailler sur la fonction de transfert discrète correspondant à un système continu échantillonné.

On va traiter un exemple concret pour mettre en évidence le problème de repliement dans le cas de la commande des systèmes continus. On s'intéresse à deux systèmes continus du second ordre ayant des pôles de même partie réelle mais de parties imaginaires séparées de  $2\pi T_e$ . Le script Matlab ci-dessous montre la création des deux systèmes :

##### Matlab

```
% Définition de deux systèmes continus :
sys1=zpk([],[-1-2i -1+2i],2^2+1);
sys2=zpk([],[-1-(2+2*pi)*i -1+(2+2*pi)*i],1+(2+2*pi)^2);
```

On voit sur la Figure 11 que la réponse indicielle des deux systèmes est bien différente.

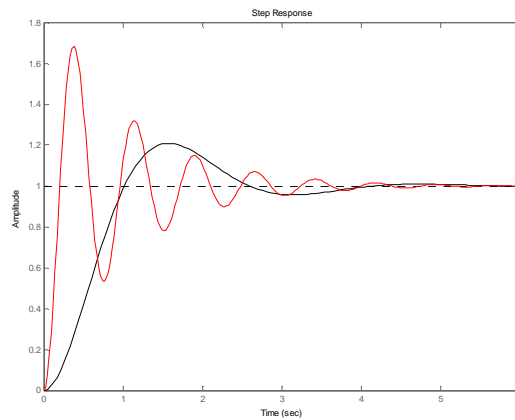


Figure 11 : réponse indicielle de deux systèmes du second ordre

Si l'on calcule la fonction de transfert échantillonnée les deux systèmes à la période de 1 seconde on obtient deux réponses indicielles très proches (représentés par des  $\circ$  sur la Figure 12). Conclusion : la période d'échantillonnage était mal choisie

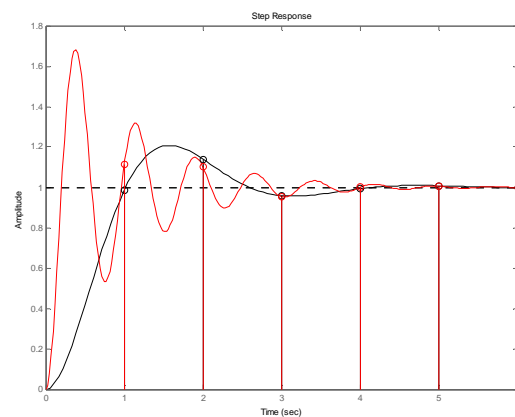


Figure 12 : réponse indicielle de deux systèmes du second ordre échantillonnés à 1s

En pratique la commande échantillonnée de systèmes continus doit être associée à certaines précautions :

- le système ne doit pas comporter de dynamiques propres situées dans les fréquences supérieures à  $f_c/2$  (exemple illustré dans ce chapitre).
- les signaux échantillonnés doivent être "nettoyés" des fréquences supérieures à  $f_c/2$  avec un filtre anti-repliement. (A moins que le signal soit suffisamment propre et ne comporte pas de composantes significatives au-delà de ces fréquences).

### Choix de la période d'échantillonnage

Le choix de la période d'échantillonnage est très important et dépend de la dynamique des signaux. Attention, les systèmes commandés en boucle fermée ont souvent une dynamique plus rapide que le système en boucle ouverte : il faut donc prendre en compte la dynamique en boucle fermée pour le choix de la fréquence d'échantillonnage.

En pratique on préconise de prendre une fréquence 5 à 25 fois supérieure à la bande passante souhaitée en boucle fermée.

On peut être tenté de prendre une fréquence d'échantillonnage très élevée si la technologie le permet (ordinateur et convertisseurs suffisamment rapide). Ceci peut amener rapidement à des problèmes numériques. Imaginons par exemple le cas d'un système ayant une dynamique très lente (quelques secondes) et un système de mesure codant les mesures sur 10 bits (cas classique). La Figure 13 illustre cet exemple. Le signal varie peu entre deux instants d'échantillonnage et les variations du signal sont faibles par rapport à la valeur de l'incrément numérique (figure du haut). Le calcul de la simple dérivée numérique se traduit donc par une succession de  $-1$   $0$  et  $+1$ .

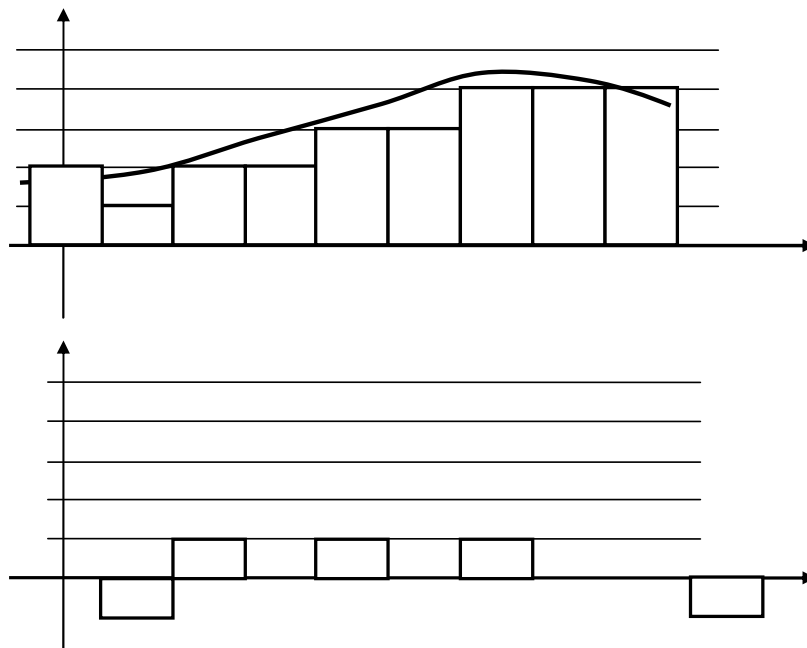


Figure 13 : période d'échantillonnage trop faible. En haut : signal échantillonné et discrétisé ; en bas : calcul de la dérivée numérique.

### Filtre anti-repliement

En toute rigueur chaque fois qu'un signal est échantillonné il faudrait faire précéder le convertisseur numérique d'un filtre analogique anti-repliement éliminant physiquement toutes les composantes du signal de fréquence supérieure à  $f_e/2$ .

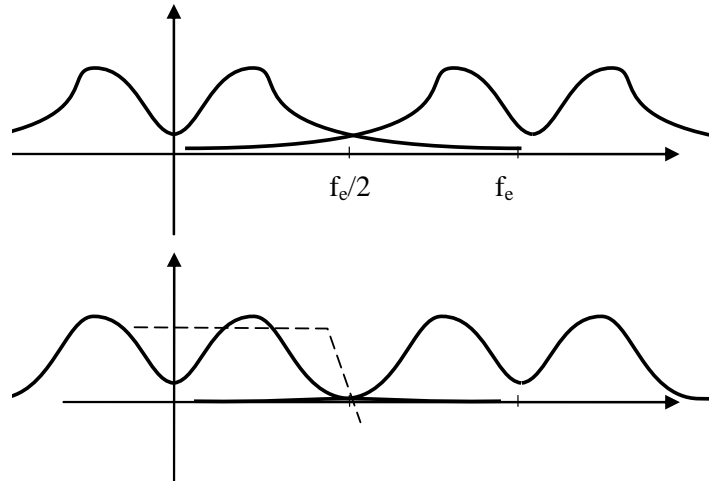


Figure 14 : Effet du filtre anti-repliement

Bien souvent la période d'échantillonnage de l'asservissement est relativement lente (quelques secondes) et il est alors impossible de réaliser un filtre de constante de temps aussi faible. On procède alors à un sur-échantillonnage : le signal est filtré, échantillonné à haute fréquence  $T'_e$  puis filtré numériquement et sous-échantillonné à la période  $T_e$  beaucoup plus grande que  $T'_e$  comme illustré sur la

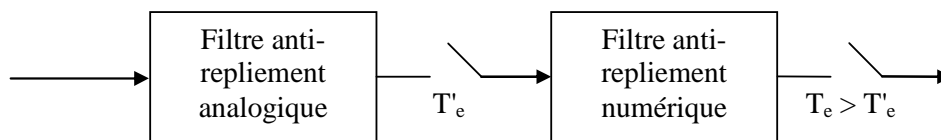


Figure 15 : Sur échantillonnage

Un simple filtre numérique de type moyenne mobile est souvent suffisant.

### 4.5 Discrétisation de fonction de transfert continues

Une manière classique pour réaliser un contrôle par ordinateur est de commencer par synthétiser un correcteur continu avec les outils de l'automatique classique puis de trouver un correcteur discret qui "approche" convenablement le correcteur continu.

#### 4.5.1 Approximation numérique d'équation différentielle

Supposons que le correcteur continu que l'on a synthétisé soit du type PD :

$$\text{PID}(s) = \frac{e(s)}{\varepsilon(s)} = k_p + k_D \cdot s \quad (26)$$

Il s'agit de calculer les commandes  $e(k)$  successives issues du correcteur en fonction des erreurs  $\varepsilon(k)$ . Une approche classique et intuitive est d'approximer la dérivée par une différence :

$$e(k) = k_p \cdot \varepsilon(k) + k_D \cdot \frac{\varepsilon(k) - \varepsilon(k-1)}{T_e} \quad (27)$$

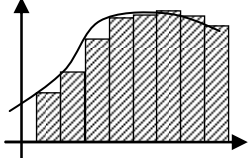
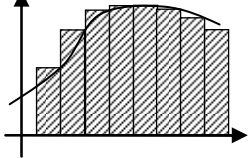
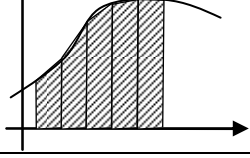
Si on calcule la transformée en  $z$  de  $e(k)$  on obtient :

$$e(z) = k_p \cdot \varepsilon(z) + k_D \cdot \frac{\varepsilon(z) - z^{-1}\varepsilon(z)}{T_e} = \left( k_p + k_D \cdot \frac{1 - z^{-1}}{T_e} \right) \cdot \varepsilon(z) \quad (28)$$

On retrouve la fonction de transfert PD(s) dans laquelle on a effectué le changement de variable  $s = \frac{1-z^{-1}}{T_e}$

De manière plus générale, l'approximation de fonction de transfert revient à effectuer l'approximation numérique d'intégration. Selon la méthode d'intégration choisie on effectue des changements de variable différents.

Tableau 4 : méthodes d'intégration

Méthode	Changement de variable	
Euler progressif (Euler's forward method)	$s = \frac{1 - z^{-1}}{z^{-1} \cdot T_e}$	
Euler retrograde (Euler's backward method)	$s = \frac{1 - z^{-1}}{T_e}$	
Méthode des trapèzes, ou bilinéaire, ou de Tustin	$s = \frac{2}{T_e} \cdot \frac{1 - z^{-1}}{1 + z^{-1}}$	

En pratique le changement de variable  $s = \frac{1 - z^{-1}}{z^{-1} \cdot T_e}$  n'est pas sans risque. On donne sur la Figure 16 la correspondance entre la zone stable du plan complexe en  $s$  et la zone correspondante après la transformation (Euler progressif, Euler rétrograde, bilinéaire). On voit notamment qu'il est possible, partant d'un système continu stable, d'obtenir des pôles dans le domaine discret en dehors du cercle de stabilité si la transformation appliquée est celle d'Euler progressif.

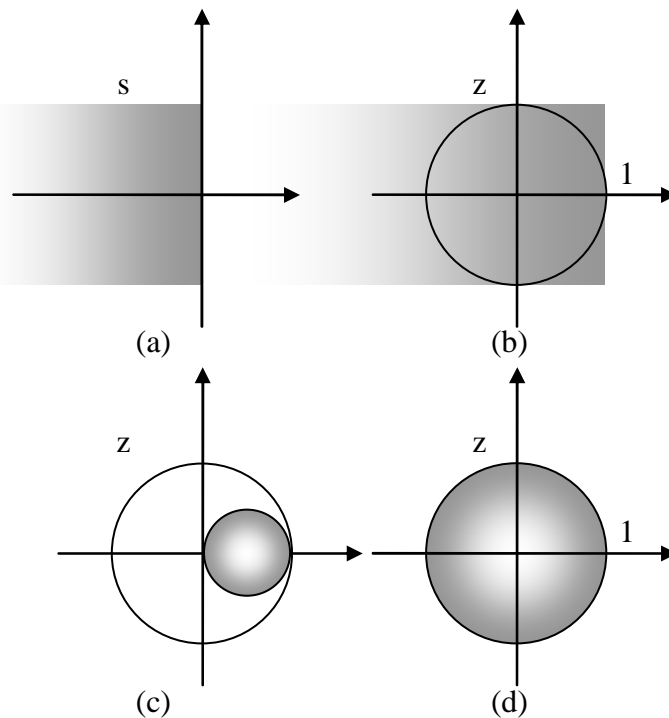


Figure 16 : correspondance des plans complexes. (a) : zone de stabilité dans le plan complexe continu en  $z$  ; (b) zone correspondante avec l'approximation d'Euler progressif ; (c) zone correspondante avec l'approximation d'Euler rétrograde ; (d) zone correspondante avec l'approximation bilinéaire.

La fonction `c2d` de Matlab permet d'obtenir l'équivalent discret d'une fonction de transfert continue avec la méthode d'intégration comme argument :

```

Matlab
sysc=tf([1 0],[1 2 5]);
Te=0.1;
% Méthode d'euler :
sysd1=c2d(sysc,Te,'zoh');
% Méthode des trapèzes :
sysd2=c2d(sysc,Te,'foh');
% Méthode bilinéaire :
sysd3=c2d(sysc,Te,'tustin');
    
```

#### 4.5.2 Superposition des réponses indicielles

Une autre méthode pour approximer une fonction de transfert discrète est de s'assurer que la réponse indicielle de sa version discrétisée correspond exactement lors des temps d'échantillonnage.



### 4.5.3 Correspondances pôles - zéros

La dernière méthode est de partir d'une décomposition de la fonction de transfert continue en pôles et zéros et d'effectuer la conversion pôles à pôles et zéros par zéros. La correspondance, selon que le pôle est réel ou complexe est donnée ci-dessous :

$$\begin{aligned} (s + a) &\rightarrow (z - e^{-a \cdot T_e}) \\ (s + a)^2 + b^2 &\rightarrow z^2 - 2 \cdot e^{-a \cdot T_e} \cdot \cos(b \cdot T_e) \cdot z + e^{-2 \cdot a \cdot T_e} \end{aligned} \quad (29)$$

### 4.5.4 Bilan

Aucune méthode n'est parfaite et on n'aura jamais en même temps :

- même réponse indicielle
- même réponse impulsionnelle
- approximation (Euler, Tustin, etc...) de l'équation différentielle
- correspondance des pôles et des zéros
- correspondance des réponses fréquentielles

Ceci étant dit le choix de la période d'échantillonnage est primordial. Si la fréquence d'échantillonnage est beaucoup plus faible que la dynamique de la fonction de transfert à approximer la différence entre chaque méthode sera minimale.



## 5. IDENTIFICATION DES SYSTEMES ECHANTILLONES

Dans le présent chapitre on donne les bases des méthodes d'identification. Pour une approche complète, voir le cours de troisième année "Identification avion" [6] et les ouvrages [1] et [7].

Même si l'on dispose d'un modèle théorique du processus que l'on souhaite commander il est presque toujours nécessaire de l'affiner en procédant à des essais sur le système réel. Bien souvent le seul modèle dont l'on dispose est ce modèle issu de l'expérimentation. Différentes méthodes s'appliquent selon l'objectif recherché et le niveau de précision exigé. On distinguera :

- les modèles paramétriques
- les modèles simples
- les modèles non paramétriques

### 5.1 Identification d'un modèle ARX par la méthode des moindres carrés récursifs

A partir de cet exemple, le plus simple qui soit, on pose les bases de toutes les méthodes d'identification.

On considère un système d'entrée  $u(t)$  et de sortie  $y(t)$  représenté par l'équation de récurrence suivante :

$$y(t) + a_1 \cdot y(t-1) + a_2 \cdot y(t-2) + \dots + a_n \cdot y(t-n) = b_0 \cdot u(t) + b_1 \cdot u(t-1) + b_2 \cdot u(t-2) + \dots + b_m \cdot u(t-m) \quad (30)$$

L'objectif est, à partir d'essais et de mesures, d'identifier les meilleures valeurs possibles pour les coefficients  $a_i$  et  $b_i$ .

L'équation (30) peut être réécrite pour mettre en évidence la *prédiction* de la sortie  $y(t)$  à partir des mesures passées :

$$y(t) = -a_1 \cdot y(t-1) - a_2 \cdot y(t-2) - \dots - a_n \cdot y(t-n) + b_0 \cdot u(t) + b_1 \cdot u(t-1) + b_2 \cdot u(t-2) + \dots + b_m \cdot u(t-m) \quad (31)$$

Soit :

$$y(t) = \varphi^T(t) \cdot \theta(t)$$

avec :

$$\begin{cases} \varphi^T(t) = [-y(t-1) \ \dots \ -y(t-n) \ , \ u(t-1) \ \dots \ u(t-m)] \\ \theta(t) = [a_1 \ \dots \ a_n \ , \ b_1 \ \dots \ b_m] \end{cases} \quad (32)$$

Le but est de trouver le meilleur ensemble de paramètres  $\theta(t)$  : celui qui, à partir d'un ensemble de mesures  $\varphi(t)$  permet de prédire avec la meilleure précision la sortie effectivement mesurée.

Supposons qu'à la suite d'une expérience on dispose d'un ensemble de mesures  $\{y(t), u(t)\}_{t=1:N}$  et que l'ensemble des paramètres  $\theta$  n'est pas bien connu. On construit la séquence des *sorties prédites* :

$$\hat{y}(t) = \varphi^T(t) \cdot \theta(t) \quad (33)$$

Le paramètre  $\theta$  est celui qui minimise une norme quadratique :

$$\theta = \arg \max \left( \frac{1}{N} \sum_{t=1}^N (y(t) - \hat{y}(t))^2 \right) = \arg \max \left( \frac{1}{N} \sum_{t=1}^N (y(t) - \varphi^T(t) \cdot \theta(t))^2 \right) \quad (34)$$

Compte tenu du fait que l'équation est linéaire au départ et que le terme à minimiser est du type quadratique la solution est simplement obtenue par une pseudo-inverse :

$$\theta = \left[ \sum_{t=1}^N (\varphi(t) \cdot \varphi^T(t)) \right]^{-1} \sum_{t=1}^N (\varphi(t) \cdot y(t)) \quad (35)$$

On obtient donc la solution classique des moindres carrés. La formule (35) est le plus souvent utilisée sous sa forme récursive :

$$\theta(t) = \theta(t-1) + F(t) \cdot \varphi(t) \cdot (y(t) - \theta^T(t-1) \cdot \varphi(t))$$

avec

$$F(t)^{-1} = F(t-1)^{-1} + \varphi(t) \cdot \varphi(t)^T \quad (36)$$

ou

$$F(t) = F(t-1) - \frac{F(t-1) \cdot \varphi(t) \cdot \varphi(t)^T \cdot F(t-1)}{1 + \varphi(t)^T \cdot F(t-1) \cdot \varphi(t)}$$

(On laisse au lecteur le plaisir d'effectuer lui-même les démonstrations, qu'il peut du reste trouver dans [1] et [7]).

## 5.2 L'identification paramétrique en général et en pratique

L'exemple précédent a permis de mettre en évidence tous les aspects de la méthodologie classique de l'identification :

- il a fallu choisir un ensemble de mesures  $\{y(t), u(t)\}_{t=1:N}$ . Le choix du signal d'excitation  $u(t)$  est en pratique primordial,
- il a fallu choisir une structure pour le modèle à identifier (ici les ordres  $n$  et  $m$ ),
- il a fallu choisir un critère (ici un critère quadratique sur l'erreur entre la sortie et la sortie prédite),
- il a fallu choisir un algorithme et le paramétrer (ici l'algorithme des moindres carrés récursifs qu'il a fallu initialiser)

Enfin un point non traité dans le chapitre précédent concerne la validation :

- il aurait fallu déterminer une méthode pour comparer différents modèles et choisir le meilleur

### 5.2.1 Choix du signal d'excitation, traitement des mesures

De manière générale le système que l'on doit identifier doit être excité dans tout le domaine fréquentiel qui nous intéresse. Le signal classiquement utilisé est une séquence pseudo binaire aléatoire (Pseudo Random Binary Signal) tel celui représenté sur la Figure 17.

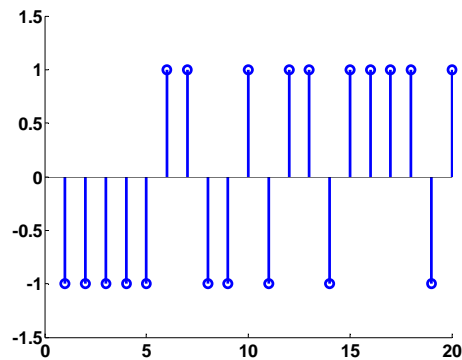


Figure 17 : exemple d'une séquence PRBS

Le signal d'excitation doit se superposer au signal de commande correspondant au point de fonctionnement étudié. Son amplitude doit être faible par rapport à l'amplitude des commandes du système (quelques %).

Une fois le système excité et les mesures effectuées il faut supprimer les valeurs moyennes des mesures.

### 5.2.2 Choix d'une structure de modèle

Le modèle étudié dans le chapitre précédent est de type ARX. On peut le noter sous forme polynomiale :

$$A \cdot y = B \cdot u \quad (37)$$

Ce modèle correspond en fait à un parti-pris sur la manière dont on traite les perturbations. En réalité on prend en compte un signal d'erreurs possibles  $e$  (de moyenne nulle) tels que :

$$A \cdot y = B \cdot u + e \quad (38)$$

Si l'on fait cette hypothèse de modélisation alors la méthode optimale est celle des moindres carrés. Malheureusement cette manière de prendre en compte les perturbations est bien restrictive. D'autres modélisations sont possibles, adaptées à d'autres algorithmes, par exemple le modèle OE :

$$y = \frac{B}{F} \cdot u + e \quad (39)$$

Il est aussi possible de "structurer" le modèle de bruit comme par exemple dans les modèles ARMAX ou BJ :

$$\begin{aligned} A \cdot y &= B \cdot u + C \cdot e \\ y &= \frac{B}{F} \cdot u + \frac{C}{D} \cdot e \end{aligned} \quad (40)$$

La structure du modèle étant choisie il faut choisir l'ordre de ses différents polynômes.

### 5.2.3 Choix d'un critère à minimiser

Dans la plupart des cas le critère à minimiser est l'erreur quadratique entre la sortie et la sortie prédite à un coup (one step ahead prediction). Ceci n'est bien évidemment pas restrictif et peut être généralisé à la prédiction à  $N$  coups.

### 5.2.4 Choix d'un algorithme

Même si chaque structure de modèle est généralement associée à un algorithme "optimal" celui-ci doit généralement se paramétrer.

Prenons par exemple l'algorithme des moindres carrés récursifs tel que présenté dans (36). Cette équation peut être modifiée en intégrant un "facteur d'oubli"  $\lambda$  :

$$\begin{aligned} \theta(t) &= \theta(t-1) + F(t) \cdot \varphi(t) \cdot (y(t) - \theta^T(t-1) \cdot \varphi(t)) \\ \text{avec} & \\ F(t)^{-1} &= \lambda \cdot F(t-1)^{-1} + \varphi(t) \cdot \varphi(t)^T \end{aligned} \quad (41)$$

Ce paramètre (typiquement entre 0.95 et 0.99) permet, comme son nom l'indique, "d'oublier" les anciennes mesures et de se concentrer sur les mesures les plus récentes.

### 5.2.5 Validation

Une fois l'algorithme d'identification appliqué et le modèle obtenu (ce qui prend généralement quelques secondes avec un ordinateur moderne) il reste à expertiser celui-ci. Selon la "qualité" du modèle obtenu on remet en cause l'une ou l'autre des étapes (signal d'excitation, structure du modèle, algorithme) et on réitère jusqu'à obtenir un modèle satisfaisant.

En pratique on essaie un certain nombre de modèles avant de choisir le "meilleur".

Le premier critère de qualité est la capacité du modèle à prédire effectivement la sortie (sortie prédite à un coup, sortie prédite à N coups). Il faut pour cela utiliser un jeu de mesures non utilisé pour l'identification. (En pratique le jeu de mesures initiales est coupé en deux : une partie pour l'identification et une partie pour la validation).

Le second critère de qualité est l'analyse du résidu : la composition spectrale de l'erreur de prédiction.

### 5.2.6 Utilisation de logiciels

En pratique il est commode d'utiliser un célèbre outil de calcul scientifique et sa toolboxe d'identification.

Comme les calculs proprement dit ne prennent que quelques secondes il est facile de réaliser un grand nombre d'identifications et de stocker les différents modèles pour les comparer ensuite. On est rapidement noyé sous le grand nombre de modèles possibles...

## 5.3 Identification "simple"

Il est souvent suffisant d'obtenir un modèle "simple" de type premier ordre et un simple essai indiciel convient souvent. Le système est excité avec un échelon unité et on cherche le meilleur modèle du premier ordre correspondant.

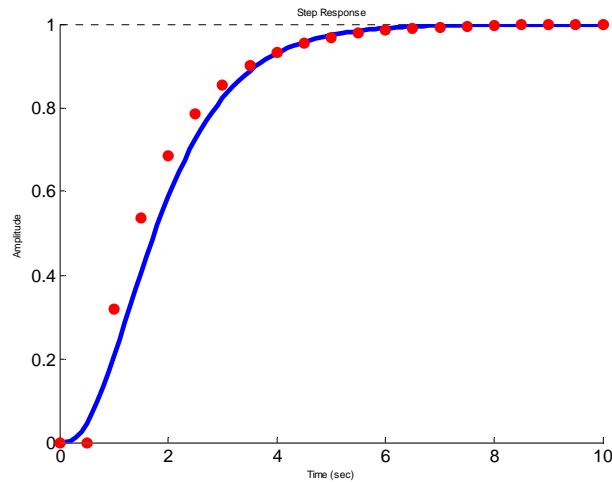


Figure 18 : identification sur un essai indiciel

Dans l'exemple de la Figure 18 on superpose le résultat d'une mesure (en bleu) avec la simulation d'un modèle discret avec une période de 0.5s. Le modèle cherché est du premier ordre avec un retard pur  $d$  :

$$y(z) = \frac{(1 + c_1) \cdot z^{-d}}{1 + c_1 \cdot z^{-1}} \cdot u(z) \quad (42)$$

Il s'agit de trouver l'entier  $d$  et le nombre réel  $c_1$  tels que la réponse indicielle du modèle corresponde pour le mieux aux mesures issues de l'essai. Ceci peut se formaliser par exemple de la manière suivante :

$$(c_1 \quad d_1) = \arg \min \sum_{t=1}^N (y_{\text{modèle}}(t) - y_{\text{mesure}}(t))^2 \quad (43)$$

Cette optimisation peut se faire à la main ou grâce à une procédure automatisée (avec le célèbre outil de calcul scientifique).

Dans le cas où le système comprend naturellement une intégration (par exemple la commande d'un moteur à courant continu par une tension et la mesure de l'angle de rotation) il est utile de prendre en compte explicitement cette intégration dans le modèle :

$$y(z) = \frac{1}{1 - z^{-1}} \cdot \frac{(1 + c_1) \cdot z^{-d}}{1 + c_1 \cdot z^{-1}} \cdot u(z) \quad (44)$$



La procédure est la même mais au lieu de prendre en compte la sortie  $y(t)$  on prend en compte la dérivée  $y'(t)$  :

$$y'(t) = y(t) - y(t-1) \quad (45)$$

## 5.4 Identification non paramétrique

Pour terminer ce panorama rapide sur les méthodes d'identification il faut parler des méthodes non paramétriques : celles qui permettent de tracer la réponse impulsionnelle ou la réponse fréquentielle du système.

### 5.4.1 Estimation de la réponse impulsionnelle

La réponse impulsionnelle d'un système peut être obtenue simplement en appliquant une entrée de type "impulsion" au système et d'en mesurer les conséquences.

La seconde méthode consiste à utiliser la méthode des moindres carrés (ou une méthode dérivée) comme vue au chapitre 5.1. La réponse impulsionnelle d'un système peut en effet être vue comme la fonction de transfert particulière où le dénominateur est de degré nul et le numérateur comporte un nombre infini (en pratique : grand) d'éléments ( $n = 0$  et  $m$  grand) :

$$y(t) = b_0 \cdot u(t) + b_1 \cdot u(t-1) + b_2 \cdot u(t-2) + \dots + b_m \cdot u(t-m) \quad (46)$$

La dernière méthode, qui sera développée ici est l'analyse d'intercorrélation.

Par définition l'intercorrélation  $R_{yu}$  entre deux signaux est :

$$R_{yu}(\tau) = E[(y(t) - E(y(t))) \cdot (u(t-\tau) - E(u(t-\tau)))] \quad (47)$$

Dans cette formule l'espérance mathématique peut être approchée grâce à la loi des grands nombres :

$$R_{yu}(\tau) \approx R_{yu}^N(\tau) = \lim_{N \rightarrow \infty} \frac{1}{N} \sum_{t=1}^N [(y(t) - m_y) \cdot (u(t-\tau) - m_u)] \quad (48)$$

On s'intéresse maintenant à un système tel que décrit par l'équation impulsionnelle (46) et on suppose que le signal d'entrée est un bruit blanc. Le signal d'entrée vérifie donc :

$$\begin{cases} \tau \neq 0 & R_{uu}(\tau) = 0 \\ \tau = 0 & R_{uu}(\tau) = R \end{cases} \quad (49)$$

Il vient alors naturellement :

$$b_k \approx \frac{1}{R} R_{yu}(k) \quad (50)$$

#### 5.4.2 Estimation de la réponse fréquentielle

On utilise ici le spectre d'intercorrélation. Par définition le spectre d'intercorrélation entre deux signaux est :

$$\Phi_{yu}(\omega) = \sum_{\tau=-\infty}^{\infty} R_{yu}(\tau) \cdot e^{-i\omega\tau} \quad (51)$$

On suppose que le signal d'entrée  $u(t)$  et le signal de sortie  $y(t)$  sont reliés par une fonction de transfert linéaire du type :

$$y(t) = F \cdot u(t) + v(t) \quad (52)$$

où  $v(t)$  est une perturbation indépendante de  $u(t)$ . On a alors les relations :

$$\Phi_{yu}(\omega) = F(j\omega) \cdot \Phi_{uu}(\omega) \quad (53)$$

Il reste à estimer les spectres d'intercorrélation pour remonter à la réponse fréquentielle  $F(j\omega)$  recherchée. Une idée naturelle est d'utiliser la relation (48) mais cela marche très mal dans presque tous les cas, même en choisissant  $N$  assez grand. L'amélioration est d'utiliser une "fenêtre" pour favoriser les valeurs faibles de  $\tau$  par rapport aux valeurs élevées. La solution classique utilisant la *fenêtre de Hamming* donne :

$$\Phi_{yu}^N(\omega) = \sum_{\tau=-\gamma}^{\gamma} R_{yu}^N(\tau) \cdot w_{\gamma}(\tau) \cdot e^{-i\omega\tau} \quad (54)$$

avec

$$\begin{cases} w_{\gamma}(\tau) = \frac{1}{2} \left( 1 + \cos\left(\frac{\pi \cdot \tau}{\gamma}\right) \right) & |\tau| < \gamma \\ w_{\gamma}(\tau) = 0 & |\tau| > \gamma \end{cases}$$

On utilise ensuite la relation (51) pour estimer la réponse fréquentielle :

$$F(j \cdot \omega) = \frac{\Phi_u^N(\omega)}{\Phi_{yu}^N(\omega)} \quad (55)$$



## 6. SYSTEMES BOUCLES

### 6.1 Fonction de transfert en boucle fermée

Un système et sa commande par ordinateur peut être représenté par un ensemble de fonctions de transfert formant un schéma-bloc plus ou moins compliqué (Figure 19).

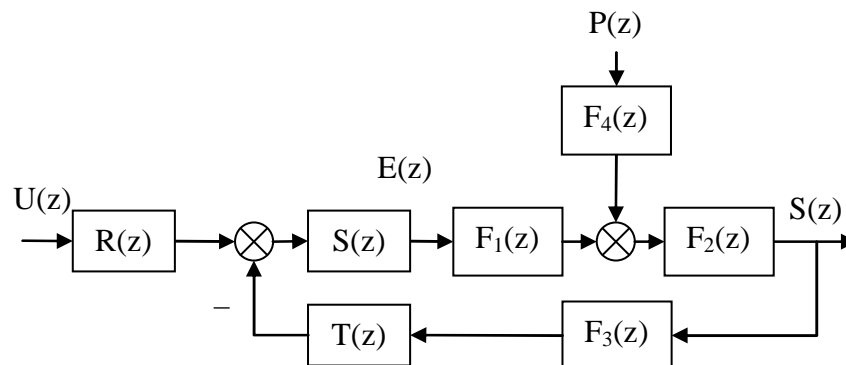


Figure 19 : système discret commandé par ordinateur. Le correcteur est représenté par les fonctions R,S,T, le système par  $F_1$  et  $F_2$ , le modèle de perturbations par  $F_4$  et le modèle des actionneurs par  $F_4$ .

Quelle que soit la structure d'un système asservi il est possible de se ramener à une forme canonique comprenant une fonction de transfert en boucle ouverte en combinant judicieusement les fonctions de transfert du procédé, du régulateur et des perturbations. La Figure 20 représente par exemple le système de la Figure 19 où  $F$  est une combinaison des fonctions de transfert du système, du régulateur, des perturbations et des capteurs.

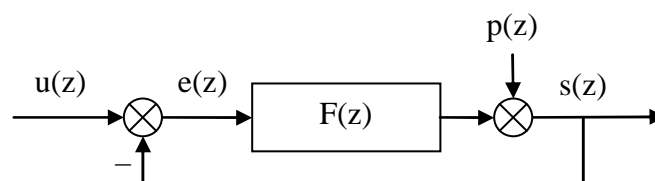


Figure 20 : système discret, notations.

Par défaut on considère une contre-réaction négative et une perturbation additive sur la sortie. La fonction de transfert  $F$  est appelée *fonction de transfert en boucle ouverte*. Comme pour les systèmes continus la *fonction de transfert en boucle fermée* est obtenue simplement à partir de  $F$  (58).

$$F(z) = \frac{s(z)}{e(z)} = \frac{B(z^{-1})}{A(z^{-1})} \quad (56)$$

$$G(z) = \frac{s(z)}{u(z)} = \frac{B(z^{-1})}{A(z^{-1}) + B(z^{-1})}$$

## 6.2 Erreur statique

L'erreur statique du système en boucle fermée, comme pour le système en boucle ouverte, est simplement le rapport des coefficients.

$$G_0 = \frac{\sum_{i=0}^{i=m} b_i}{1 + \sum_{i=0}^{i=m} b_i + \sum_{i=1}^{i=n} a_i} \quad (57)$$

Si l'on souhaite que l'asservissement confère un gain statique unitaire au système en boucle fermée (c'est-à-dire que l'on force la sortie  $s(k)$  à tendre vers la valeur de consigne  $u(k)$ ) il faut que ce rapport de coefficients soit égal à un. On montre facilement que cela est obtenu si la fonction de transfert  $F(z)$  comporte un intégrateur :

$$F(z) = \frac{s(z)}{e(z)} = \frac{1}{1 - z^{-1}} \cdot \frac{B(z)}{A'(z)} \quad (58)$$

## 6.3 Rejet de perturbations

La fonction de transfert entre la perturbation  $p(z)$  et la sortie  $y(z)$  est appelée fonction de sensibilité perturbation-sortie, notée  $S_{yp}$ .

$$S_{yp}(z) = \frac{s(z)}{p(z)} = \frac{A(z^{-1})}{A(z^{-1}) + B(z^{-1})} \quad (59)$$

Si l'on souhaite qu'une perturbation de type échelon soit parfaitement rejetée il faut que  $A(1) = 0$ , on retrouve donc la condition (58) à savoir que le système doit comporter une intégration dans la boucle directe. On peut être en outre être amené à spécifier que les perturbations ne soient pas amplifiées dans une certaine bande de fréquence. On rajoute donc une contrainte du type  $|S_{yp}(e^{j2\pi f})| < S_{\max}$  pour  $f$  appartenant à un certain domaine de  $[0 \ 0.5]$ .

## 6.4 Stabilité des systèmes à simple gain de rétroaction

Les critères de Routh et de Jury (donnés dans le chapitre 3.2.7 p13) ou le calcul numérique des pôles permettent de déterminer la stabilité du système en boucle fermée si l'on connaît sa fonction de transfert.

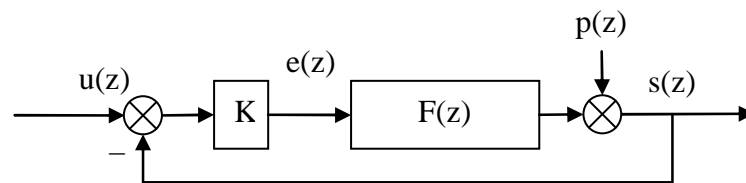


Figure 21 : système bouclé par un simple gain.

On s'intéresse ici au cas particulier où l'asservissement est réalisé par un simple gain de rétroaction  $K$  (Figure 21). On cherche la valeur de  $K$  qui amène le système en limite de stabilité. L'outil adapté ici est le lieu d'Evans ou lieu des racines (root locus) qui permet de visualiser l'évolution des pôles du système bouclé lorsque le gain  $K$  varie.

Prenons l'exemple d'un système comportant trois pôles et deux zéros. L'exemple est donné sous forme de script Matlab :

### Matlab

```
% Définition d'un système discret :
sys=zpk([-2.037 -0.1406],[0.6065 0.3679 0.3679],1,1)
%Tracé du lieu des racines
rlocus(sys);
```

Le script précédent produit le tracé de la Figure 22. On pourrait y lire que la limite de stabilité est obtenue pour  $K=0.2$  environ (détail de la Figure 23).

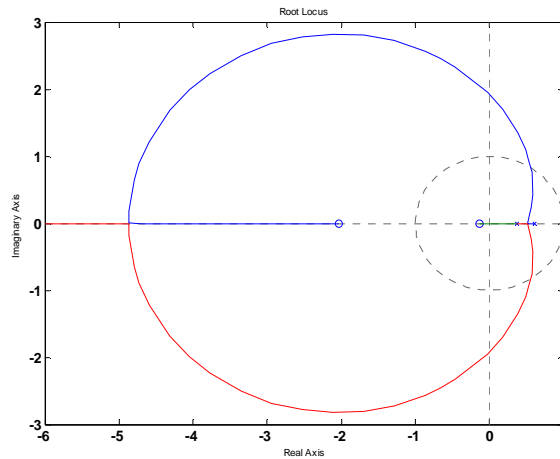


Figure 22 : lieu des racines

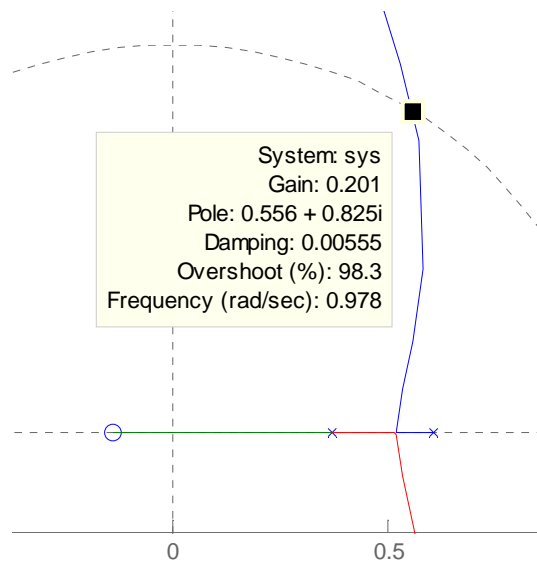


Figure 23 : lieu des racines : détail de la limite de stabilité

On voit sur la Figure 22 que le lieu des racines d'un système échantillonné ressemble à celui d'un système continu :

- on commence par tracer les pôles ( $\times$ ) et les zéros ( $o$ ) du système en boucle ouverte.
- les pôles du système en boucle fermée décrivent un lieu qui part des pôles (pour  $K = 0$ ) et arrive aux zéros (pour  $K = \infty$ )
- lorsque il y a  $n-m$  plus de pôles que de zéros le système comporte  $n-m$  branches asymptotiques.

On ne décrira pas ici les diverses propriétés du lieu des racines qui permettent de le tracer à la main puisque il est facile à obtenir avec un ordinateur.



Il peut aussi être utilisé pour régler "à la main" le gain permettant d'obtenir un ou plusieurs modes dominants ayant certaines propriétés de fréquence et d'amortissement. On voit sur la Figure 24 que le gain 0.017 permet d'obtenir un pôle double d'amortissement 0.706 et de fréquence 0.65 rad/s et un pôle simple de fréquence 1.63 rad/s.

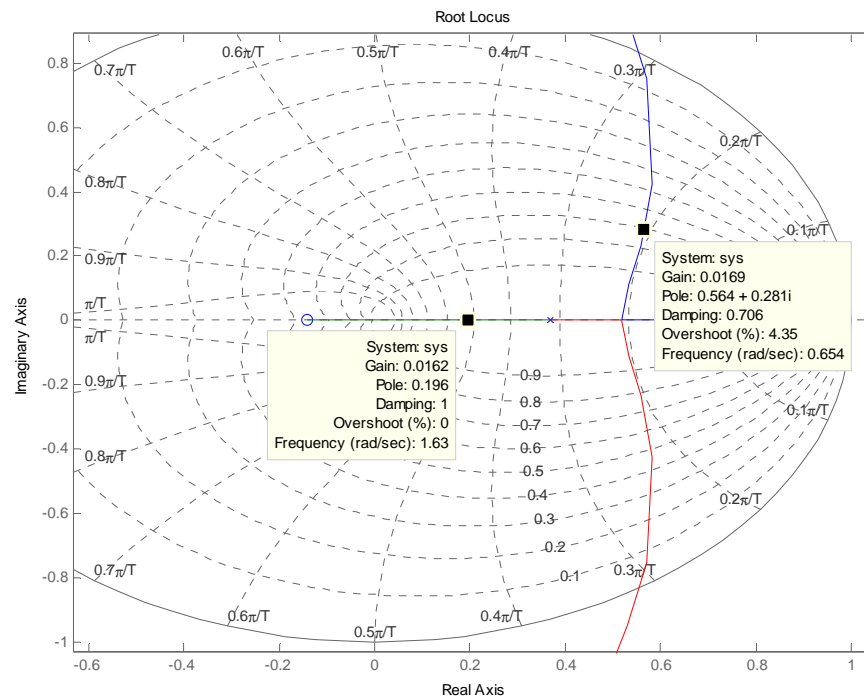


Figure 24 : utilisation du lieu des racines pour régler le gain de la boucle fermée

L'outil rltool de Matlab permet de nombreuses manipulations graphiques de ce type.

## 6.5 Analyse des systèmes bouclés dans le domaine fréquentiel : stabilité

Comme dans le cas continu il est possible de prévoir des propriétés, en particulier la stabilité, d'un système en boucle fermée à partir de l'examen de la fonction de transfert en boucle ouverte.

On ne redémontrera pas ici le théorème de Nyquist mais on se contente de rappeler l'énoncé.

On rappelle que le *lieu de Nyquist* est le lieu de  $F(j2\pi f)$  dans le plan complexe pour  $f$  variant de 0 à  $+\infty$ . A noter que le lieu pour la zone  $[0, 0.5]$  est symétrique de celui pour la zone  $[0.5, 1]$ . Pour cette raison on ne représente généralement que le lieu pour  $f$  appartenant à  $[0, 0.5]$ .

### 6.5.1 Théorème de Nyquist dans le cas où le système est stable en boucle ouverte (critère du revers)

Le système est stable en boucle fermée si le lieu de Nyquist de la fonction de transfert en boucle ouverte parcouru dans le sens des fréquences croissantes de 0 à 0.5 laisse le point critique (-1,0) à gauche.

La Figure 25 représente par exemple le lieu de Nyquist d'une fonction de transfert en boucle ouverte qui sera stable en boucle fermée.

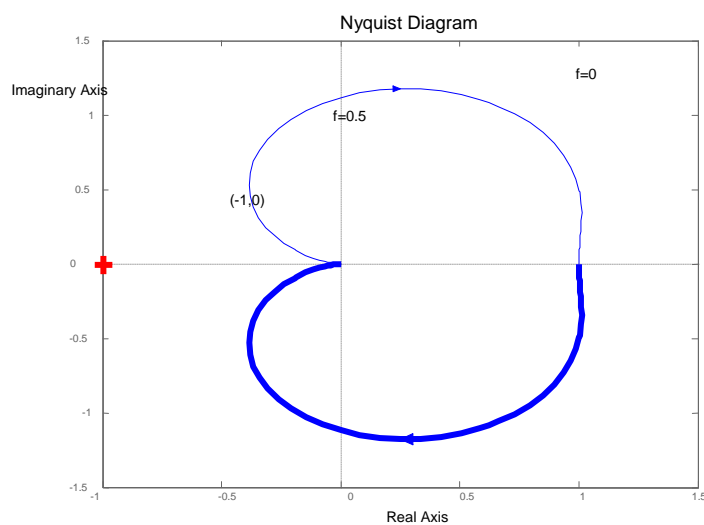


Figure 25 : lieu de Nyquist d'un système stable

### 6.5.2 Théorème de Nyquist dans le cas où le système est instable en boucle ouverte

Si le système en boucle ouverte comporte  $N$  pôles instables il sera stable en boucle fermée si le lieu de Nyquist de la fonction de transfert en boucle ouverte parcouru dans le sens des fréquences croissantes de  $-0.5$  à  $0.5$  entoure le point critique (-1,0)  $N$  fois dans le sens des aiguilles d'une montre..

### 6.5.3 Interprétations du lieu de Nyquist

Comme dans le cas continu, la distance à laquelle on passe du point critique renseigne sur la robustesse du système bouclé. La distance sera exprimée de différente manière (Figure 26) :

- marge de module  $\Delta M$  : plus petite distance entre le lieu des racines et le point critique,

- marge de gain  $\Delta G$
- marge de phase  $\Delta P$

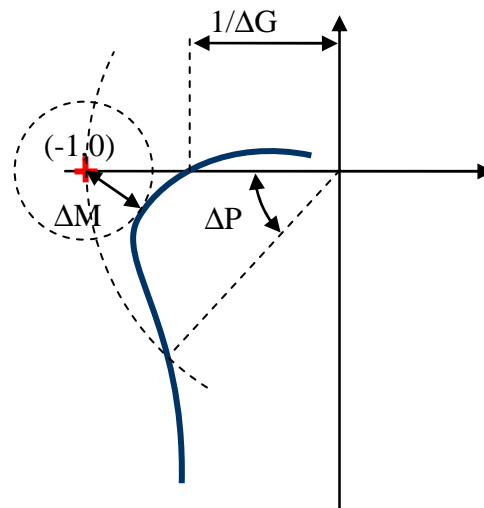


Figure 26 : Marges de stabilité

Les valeurs typiques de ces marges sont données ci-après :

- $\Delta G \geq 6\text{dB}$
- $\Delta M \geq -6\text{dB}$
- $30^\circ \leq \Delta P \leq 60^\circ$

La distance entre le point critique et le lieu des racines est la norme de  $1+F(j2\pi f)$ . On peut donc exprimer la marge de module en fonction de la fonction de sensibilité :

$$\begin{aligned} \Delta M &= \min |1 + F(j \cdot \omega)|_{\omega \in [0, \pi]} \\ \Delta M &= \min \left| 1 + \frac{B(j \cdot \omega)}{A(j \cdot \omega)} \right|_{\omega \in [0, \pi]} = \max \left| \frac{A(j \cdot \omega)}{A(j \cdot \omega) + B(j \cdot \omega)} \right|_{\omega \in [0, \pi]} \quad (60) \\ \Delta M &= \max |S_{yp}(j \cdot \omega)|_{\omega \in [0, \pi]}^{-1} \\ \Delta M_{\text{dB}} &= -\max |S_{yp}(j \cdot \omega)|_{\text{dB}} \end{aligned}$$

Si l'on trace l'évolution de  $S_{yp}$  en fonction de  $f$  on obtient donc la valeur de la marge de module.



# 7. METHODES DE REGULATION DES SYSTEMES ECHANTILLONES

## 7.1 Introduction

Les bases théoriques permettant de concevoir le correcteur pour un système échantillonné ont été posées dans les chapitres précédents. Ce chapitre fait le bilan des connaissances acquises et donne quelques solutions pour calculer des correcteurs à partir d'un cahier des charges précis.

Les systèmes que l'on cherche à commander sont généralement, par nature, continus. La réalisation d'un correcteur par un ordinateur implique que l'on génère une équation de commande discrète (généralement une équation récurrente). L'obtention de cette équation de commande peut se faire de deux manières :

- la première méthode consiste à d'abord synthétiser un correcteur continu (en utilisant les outils classiques de l'automatique et la simulation) et discrétiser ensuite ce correcteur. C'est l'objet du chapitre 7.2.2.
- la seconde méthode consiste à tenir la représentation discrète du système continu associé à son interface numérique (convertisseurs analogique-numérique et numérique-analogique) comme expliqué dans le chapitre 4.5 puis synthétiser directement le correcteur discret. Cela sera expliqué dans le chapitre 7.3 et suivants.

Ce choix peut être résumé par la figure suivante :

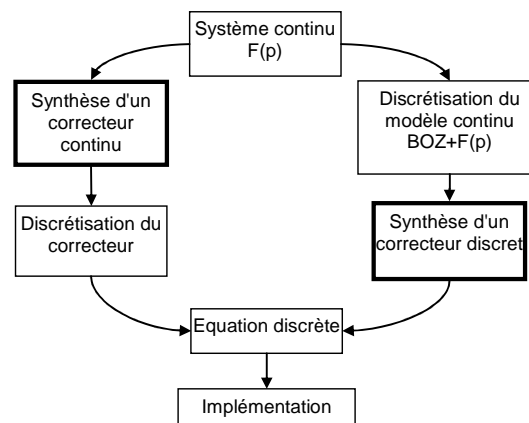


Figure 27 : Réalisation d'un correcteur discret.

Au-delà des affaires de goût, le choix de la méthode sera souvent dicté par l'application. En effet, la première méthode est d'autant meilleure que le correcteur discrétisé approche correctement le correcteur continu. Cela implique que le temps d'échantillonnage soit très faible par rapport aux constantes de temps du système

bouclé. Dans le cas où les contraintes temps réel sont fortes (capacités de calcul partagées par exemple) il sera préférable de synthétiser un correcteur discret.

## 7.2 Structure d'un système commandé par ordinateur, hypothèses

Par la suite on se ramènera toujours à un système tel que celui représenté sur la Figure 28 en cherchant un correcteur  $K(z)$  linéaire. On montre que quel que soit le correcteur linéaire on peut se ramener à la forme représenté sur la Figure 29 (forme RST).

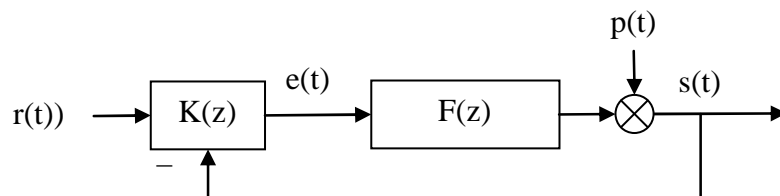


Figure 28 : système bouclé, forme générale.

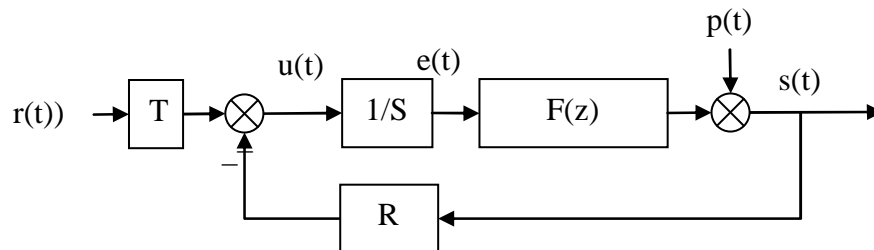


Figure 29 : système bouclé, forme canonique.

### 7.2.1 Discrétisation, choix de la période d'échantillonnage

Du point de vue du ordinateur, l'extérieur (le procédé) est représenté par la fonction de transfert  $F(z)$ . L'hypothèse importante est que la commande (sortie du ordinateur) et la mesure (entrée du ordinateur) sont échantillonnées de manière synchrone à la même période  $T_e$ .

Le choix judicieux de la fréquence d'échantillonnage est très important. En pratique on choisit cette fréquence en fonction de la bande passante du système en **boucle fermée** :

$$f_e = (6 \text{ à } 25) f^{BF} \quad (61)$$

### 7.2.2 Discrétisation d'un correcteur continu

La première méthode pour obtenir les équations d'un correcteur numérique est de faire sa synthèse avec les outils de l'automatique continue puis de discrétiser la fonction de transfert du correcteur continu obtenu (voir pour cela le chapitre 4.2 *Echantillonnage des systèmes continus*).

On va voir par la suite que la synthèse directe d'un correcteur discret offre plus de degré de liberté et permet en général de remplir un cahier des charges plus complet.

## 7.3 Régulateur PID numérique

Le régulateur le plus populaire est le régulateur PID. Nous allons voir dans un premier temps comment on génère un régulateur PID à partir des équations bien connues du régulateur PID continu et comment ces équations peuvent être généralisées pour obtenir plus de degrés de liberté pour le réglage.

### 7.3.1 Cas 1

Un régulateur PID continu a une fonction de transfert du type :

$$K_{\text{PID}}(s) = K \cdot \left( 1 + \frac{1}{T_i \cdot s} + \frac{T_d \cdot s}{1 + \frac{T_d}{N} \cdot s} \right) \quad (62)$$

La discrétisation de ce correcteur continu par la méthode d'Euler (Tableau 4 page 31) se fait en remplaçant la variable de Laplace  $s$  par le terme  $\frac{1-q^{-1}}{q^{-1} \cdot T_e}$ . On obtient :

$$K_{\text{PID}}(q) = K \cdot \left( 1 + \frac{1}{T_i \cdot \frac{1-q^{-1}}{T_e}} + \frac{T_d \cdot \frac{1-q^{-1}}{T_e}}{1 + \frac{T_d}{N} \cdot \frac{1-q^{-1}}{T_e}} \right) \quad (63)$$

En réarrangeant on obtient la forme générale RST (Figure 30):

$$K_{PID}(q) = K \cdot \frac{r_0 + r_1 \cdot q^{-1} + r_2 \cdot q^{-2}}{(1 - q^{-1}) \cdot (1 + s_1' \cdot q^{-1})}$$

$$\begin{cases} r_0 = K \cdot \left( 1 + \frac{T_e}{T_i} + \frac{N \cdot T_d}{T_d + N \cdot T_e} \right) \\ r_1 = -K \cdot \left( 1 + \frac{T_d}{T_d + N \cdot T_e} + \frac{T_e}{T_i} \cdot \frac{T_d}{T_d + N \cdot T_e} + \frac{N \cdot T_d}{T_d + N \cdot T_e} \right) \\ r_2 = K \cdot \left( \frac{T_d}{T_d + N \cdot T_e} \right) \\ s_1' = \frac{T_d}{T_d + N \cdot T_e} \end{cases} \quad (64)$$

Comme on le voit le correcteur obtenu comporte autant de degrés de liberté que le correcteur PID analogique. On reconnaît au dénominateur le terme  $(1 - q^{-1})$  en facteur qui assure l'effet d'intégration.

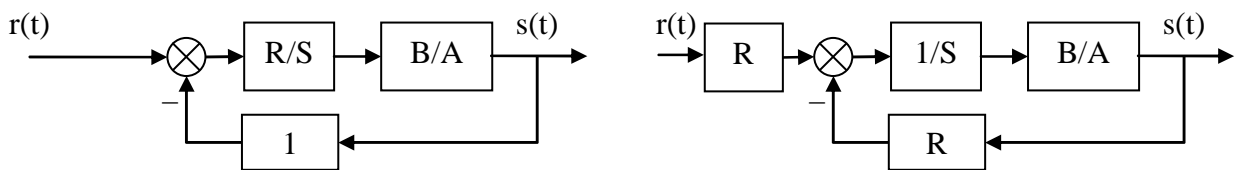


Figure 30 : PID sous forme RST

La fonction de transfert en boucle fermée obtenue est donnée par l'équation suivante :

$$H_{BF} = \frac{B \cdot R}{A \cdot S + B \cdot R} = \frac{B \cdot R}{P} \quad (65)$$

De manière générale le polynôme caractéristique  $P$  fixe les performances visées en boucle fermée. Le calcul de  $R$  et  $S$ ,  $P$  étant fixé et  $A$  et  $B$  étant connus, est réalisé en résolvant l'équation de Bézout  $P = A \cdot S + B \cdot R$ . (voir 9.1 Annexe 1 : Résolution de l'équation polynomiale de Bezout)

On voit que le numérateur  $B \cdot R$  est imposé une fois que  $R$  et  $S$  sont trouvés. Le régulateur introduit donc naturellement des zéros dans la boucle fermée.

Traisons l'exemple entièrement dans le cas où le système est modélisé comme un système du deuxième ordre avec faible retard. La fonction de transfert du procédé est donnée par :



$$H_{BO} = \frac{B}{A} = \frac{b_1 \cdot q^{-1} + b_2 \cdot q^{-2}}{1 + a_1 \cdot q^{-1} + a_2 \cdot q^{-2}} \quad (66)$$

Le correcteur PID rajoutant deux pôles, le polynôme caractéristique en boucle fermée spécifié doit posséder quatre pôles. Le polynôme caractéristique en boucle fermée doit donc être de la forme :

$$\begin{aligned} P &= 1 + p_1 \cdot q^{-1} + p_2 \cdot q^{-2} + p_2 \cdot q^{-3} + p_2 \cdot q^{-4} \\ P &= (1 + p_1' \cdot q^{-1} + p_2' \cdot q^{-2}) \cdot (1 + \alpha_1 \cdot q^{-1}) \cdot (1 + \alpha_2 \cdot q^{-1}) \end{aligned} \quad (67)$$

Et il doit être identifié au produit  $AS + BR$  avec  $R$  et  $S$  ayant les formes données par (64) :

$$\begin{aligned} &(1 + p_1' \cdot q^{-1} + p_2' \cdot q^{-2}) \cdot (1 + \alpha_1 \cdot q^{-1}) \cdot (1 + \alpha_2 \cdot q^{-1}) \\ &= (1 + a_1 \cdot q^{-1} + a_2 \cdot q^{-2}) \cdot (1 - q^{-1}) \cdot (1 + s_1' \cdot q^{-1}) \\ &+ (b_1 \cdot q^{-1} + b_2 \cdot q^{-2}) \cdot (r_0 + r_1 \cdot q^{-1} + r_2 \cdot q^{-2}) \end{aligned} \quad (68)$$

Les coefficients  $p_1'$  et  $p_2'$  définissent un modèle du second ordre aux caractéristiques (pulsation propre  $\omega_0$  et amortissement  $\sigma$  choisis et coefficients déduits conformément au Tableau 3 page 25). Les coefficients  $\alpha_1$  et  $\alpha_2$  correspondent à des pôles auxiliaires aperiodiques. Les choix typiques pour ces coefficients sont :

$$-0.05 < \alpha_1, \alpha_2 < -0.5 \quad (69)$$

On montre que la résolution de (68) est réalisée par une inversion matricielle simple :

$$M \cdot x = p \quad (70)$$

avec

$$\begin{cases} x^T = [1 & s_1' & r_0 & r_1 & r_2] \\ p^T = [1 & p_1 & p_2 & p_3 & p_4] \end{cases}$$

et

$$M = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ a_1' & 1 & b_1 & 0 & 0 \\ a_2' & a_1' & b_2 & b_1 & 0 \\ a_3' & a_2' & 0 & b_2 & b_1 \\ 0 & a_3' & 0 & 0 & b_2 \end{pmatrix} \quad (71)$$

On montre que M est inversible à la condition que A et B soient premiers entre eux, c'est-à-dire que A et B n'aient pas de pôles et de zéros en commun.

(Le problème peut aussi être résolu en utilisant la résolution de l'équation de Bézout).

### 7.3.2 Cas 2

Comme on l'a vu le correcteur PI numérique directement transposé du correcteur PID analogique décrit dans le chapitre 7.3.1 a l'inconvénient de ramener des zéros au dénominateur de la fonction de transfert en boucle fermée (équation 65). Il est possible, grâce à la flexibilité du régulateur numérique d'imposer au correcteur de ne pas ajouter de zéros en boucle fermée. La fonction de transfert désirée en boucle fermée sera donc de la forme :

$$H_{BF} = \frac{P(1)}{B(1)} \cdot \frac{B}{P} \quad (72)$$

Le facteur  $P(1)/B(1)$  garantit le gain statique unitaire du système en boucle fermée (voir page 12). Si le régulateur a la forme générale RST alors les polynômes R S et T doivent vérifier :

$$H_{BF} = \frac{B \cdot T}{A \cdot S + B \cdot R} = \frac{P(1) B}{B(1) P} \quad (73)$$

S et R sont obtenus par la résolution de la même équation que précédemment (68). Le polynôme T est obtenu par simple identification :  $T = R(1)$ .

La structure, sous forme de schéma-bloc est donnée ci-dessous :

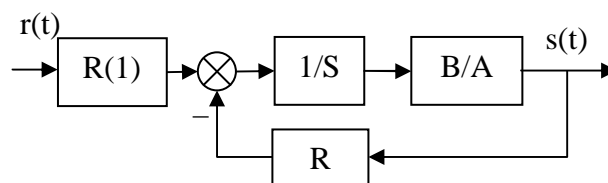


Figure 31 : PID2 sous forme RST

On montre que cette structure de PID, ramené en continu, correspond au PID canonique pour l'action intégrale. L'action proportionnelle et dérivée n'est appliquée que sur le signal y.

La structure du PID2 par rapport au PID1 a pour avantage de ne pas ramener de zéros à la boucle fermée. Cela se traduit par un dépassement moindre en transitoire lorsqu'on applique un échelon de commande. Par contre les deux systèmes réagissent de la même manière à un échelon de perturbation.

## 7.4 Placement de pôles

Les deux structures de PID décrites dans le chapitre précédent sont deux cas particulier d'une stratégie plus générale de placement de pôles.

La synthèse est plus générale que les PID présentés précédemment puisqu'il n'est pas nécessaire de faire d'hypothèses sur le degré de A et B, sur le retard, ni sur la présence ou non de zéros stables ou instables dans le procédé.

La structure du correcteur est la structure type RST telle que présentée dans la Figure 29. Pour la suite des calculs on posera pour la fonction de transfert les conventions suivantes :

$$H_{BO} = \frac{q^{-d} \cdot B(q^{-1})}{A(q^{-1})} \quad (74)$$

$$\begin{cases} A(q^{-1}) = 1 + a_1 \cdot q^{-1} + \dots + a_{n_A} \cdot q^{-n_A} \\ B(q^{-1}) = b_1 \cdot q^{-1} + \dots + b_{n_B} \cdot q^{-n_B} = q^{-1} \cdot B^*(q^{-1}) \end{cases}$$

Compte tenu de la forme RST du régulateur la fonction de transfert en boucle fermée vérifie :

$$H_{BF} = \frac{q^{-d} \cdot B(q^{-1}) \cdot T(q^{-1})}{P(q^{-1})} \quad (75)$$

$$P(q^{-1}) = A(q^{-1}) \cdot S(q^{-1}) + q^{-d} \cdot B(q^{-1}) \cdot R(q^{-1}) = 1 + p_1 \cdot q^{-1} + \dots + p_{n_P} \cdot q^{-n_P}$$

Le polynôme  $P(q^{-1})$  définit donc les pôles de la fonction de transfert en boucle fermée et traduit donc le comportement en *poursuite*.

De la même manière la fonction de transfert entre une perturbation et la sortie (fonction de sensibilité perturbation-sortie) est donnée par :

$$S_{yp} = \frac{A(q^{-1}) \cdot S(q^{-1})}{P(q^{-1})} \quad (76)$$

Ce même polynôme traduit donc aussi le comportement en *régulation*.

Les consignes en poursuite et en régulation peuvent être différentes comme cela est représenté par exemple sur la Figure 32.

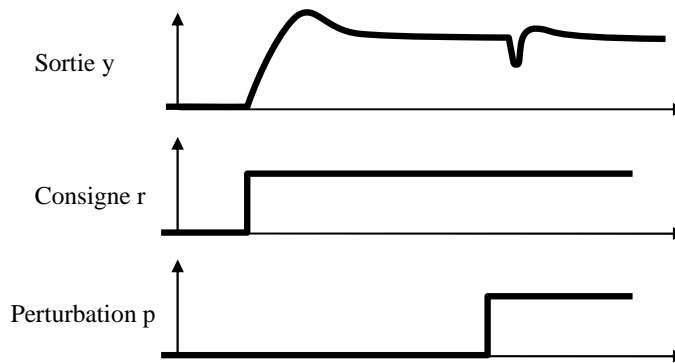


Figure 32 : Performances différentes en régulation et en poursuite

### 7.4.1 Choix des pôles en boucle fermée

De manière générale le comportement en boucle fermée est spécifié sous la forme d'une pulsation propre  $\omega_0$  et d'un amortissement  $\sigma$ , donc une fonction de transfert du second ordre. La spécification étant généralement donnée en continu il est nécessaire de traduire la valeur désirée de  $\omega_0$  et  $\sigma$  en un polynôme  $P_D(q^{-1})$  par discrétisation de la fonction de transfert continue (chapitre 4.5 page 29). Ce polynôme  $P_D$  traduit le comportement dominant du système bouclé.

Si l'on souhaite introduire un effet de filtrage supplémentaire dans certaines bandes de fréquence (réduction des bruits de mesure, adoucissement de la commande, robustesse) cela est décrit par un polynôme  $P_F$  comportant des pôles auxiliaires, plus rapides que les pôles de  $P_D$ .

Le polynôme caractéristique en boucle fermée est donc constitué de pôles dominants et de pôles auxiliaires :

$$P(q^{-1}) = P_D(q^{-1}) \cdot P_F(q^{-1}) \quad (77)$$

### 7.4.2 Régulation : calcul de R et S

Le polynôme  $P(q^{-1})$  étant fixé il s'agit de trouver R et S vérifiant :

$$P(q^{-1}) = A(q^{-1}) \cdot S(q^{-1}) + q^{-d} \cdot B(q^{-1}) \cdot R(q^{-1}) \quad (78)$$

Comme déjà vu au chapitre 7.3.1 on obtient une équation de Bézout qu'il s'agit de résoudre. (Voir aussi 9.1 Annexe 1 : Résolution de l'équation polynomiale de Bezout)

Il est parfois nécessaire de spécifier R et S avec des parties fixes  $H_R$  et  $H_S$ . Dans ce cas l'équation (78) est modifiée et on l'écrit sous la forme:

$$P(q^{-1}) = (A(q^{-1}) \cdot H_S(q^{-1})) \cdot S'(q^{-1}) + (q^{-d} \cdot B(q^{-1}) \cdot H_R(q^{-1})) \cdot R'(q^{-1}) \quad (79)$$

Les inconnues que l'on souhaite trouver sont  $S'$  et  $R'$ . Ceci est nécessaire dans les cas particuliers suivants :

- spécification d'erreur statique nulle,
- rejet d'une perturbation harmonique,
- blocage d'un signal,

Détaillons maintenant ces cas particuliers.

### Spécification d'une erreur statique nulle

On retrouve la nécessité d'avoir un terme intégrateur dans la boucle directe. La spécification se retrouve donc sur  $S$  :

$$S(q^{-1}) = (1 - q^{-1}) \cdot S'(q^{-1}) = H_S(q^{-1}) \cdot S'(q^{-1}) \quad (80)$$

### Rejet d'une perturbation harmonique

Si le système est perturbé par une perturbation harmonique à une fréquence fixe (par exemple 50Hz...) il est possible d'imposer à  $S$  d'avoir un module *nul* ou amorti à cette fréquence. Cela se traduit par la présence dans  $S$  d'une paire de zéros complexe à amortissement nul ou faible à cette fréquence :

$$S(q^{-1}) = (1 + \alpha \cdot q^{-1} + \beta \cdot q^{-2}) \cdot S'(q^{-1}) = H_S(q^{-1}) \cdot S'(q^{-1}) \quad (81)$$

### Blocage d'un signal

Pour certaines application on souhaite que le régulateur "laisse passer" certaines composantes du signal de sortie, c'est-à-dire qu'il ne réagisse pas à ces composantes. Cela se traduit par  $R=0$  à ces fréquences. On retrouve la même forme que précédemment :

$$R(q^{-1}) = (1 + \alpha \cdot q^{-1} + \beta \cdot q^{-2}) \cdot R'(q^{-1}) = H_R(q^{-1}) \cdot R'(q^{-1}) \quad (82)$$

### 7.4.3 Poursuite : calcul de T

Le schéma-bloc de l'asservissement est donné sur la Figure 33. Le terme  $T$  est décomposé en un terme  $T'$  et un terme  $B^*/A^*$ .

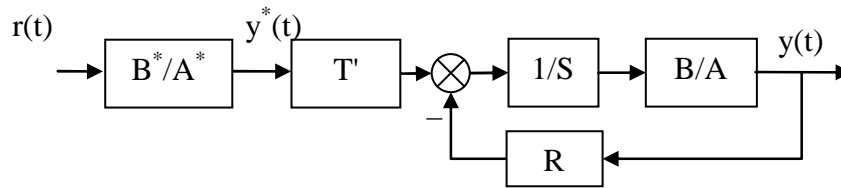


Figure 33 : placement de pôle

L'idée est de régler  $B^*$  et  $A^*$  de telle manière que le signal  $y^*$  ait la dynamique de poursuite désirée (par exemple un second ordre avec pulsation propre et amortissement donné) puis de régler  $T'$  de telle manière que le transfert entre  $y^*$  et  $y$  soit le plus "transparent" possible.

### Réglage de $T'$

Le transfert entre  $y^*$  et  $y$  est :

$$S_{yy^*}(q^{-1}) = T'(q^{-1}) \frac{q^{-d} \cdot B(q^{-1})}{P(q^{-1})} \quad (83)$$

Pour assurer le transfert le plus "transparent" possible il faut donc choisir

$$T'(q^{-1}) = \frac{P(q^{-1})}{B(1)} \quad (84)$$

Remarques :

- Si  $B(1)=0$  alors on remplace  $B(1)$  par 1 dans (84)
- De même dans (84)  $P(q^{-1})$  peut être remplacé par  $P_D$  correspondant à la dynamique dominante.

### Réglage de $B^*$ et $A^*$

$B^*$  et  $A^*$  sont choisis de telle manière que la fonction de transfert  $B^*/A^*$  corresponde à la dynamique de poursuite souhaitée.

## 7.5 Poursuite et régulation à objectifs indépendants

Cette méthode ressemble à la méthode de placement de pôles à ceci près que l'on va effectuer une simplification des zéros de  $B^*(q^{-1})$  : cela ne s'applique donc qu'aux systèmes avec des pôles stables. Plus précisément il faut qu'ils aient un amortissement suffisant ( $> 0,2$ ) de manière à ne pas être trop près du cercle unité.

La structure de l'asservissement est toujours celle donnée sur la Figure 33.

**Choix de P**

$P(q^{-1})$  est choisi de manière à simplifier  $B^*(q^{-1})$  (rappel : la forme de B est donnée dans l'équation 74). Comme indiqué dans le chapitre 7.4.2 page 60 il s'agit donc d'imposer les zéros de  $B(q^{-1})$  dans  $H_R(q^{-1})$  et  $H_S(q^{-1})$ .

**Choix de T**

Si l'on choisit  $T(q^{-1}) = P(q^{-1})$  alors le transfert entre  $y^*$  et  $y$  est :

$$S_{yy^*}(q^{-1}) = q^{-(d+1)} \quad (85)$$

**Réglage de  $B^*$  et  $A^*$** 

$B^*$  et  $A^*$  sont choisis de telle manière que la fonction de transfert  $B^*/A^*$  corresponde à la dynamique de poursuite souhaitée.





## 8. TECHNOLOGIE DE LA COMMANDE PAR ORDINATEUR

Du point de vue de l'automaticien le ordinateur est relié au système physique à piloter par un certain nombre d'intermédiaires plus ou moins gênants :

- les interfaces analogiques et numériques,
- le transport de l'information par bus et protocole,
- l'utilisation éventuelle de capteurs ou d'actionneurs « intelligents ».

Il ne s'agit pas ici de faire un exposé complet sur ces trois aspects mais de poser les problèmes qui se posent à l'automaticien lorsque le système qu'il doit commander est perturbé par ces éléments.

Enfin ce chapitre se termine sur les notions de « Hardware In the Loop » et de « prototypage rapide » qui sont partie intégrante des processus de réalisation de systèmes de contrôle.

### 8.1 Calculateurs et interfaces numériques / analogiques

Le premier problème à prendre en compte est la quantification de l'information. Du point de vue du ordinateur d'une part et du point de vue des convertisseurs d'autre part. Dans un premier temps, considérons les aspects performance de calcul lié à la nature du codage des nombres dans le ordinateur.

#### Les performances liées au types de variables

Prenons l'exemple où le concepteur de la loi de commande souhaite réaliser un correcteur de type proportionnel. Entre l'acquisition des mesures et l'écriture de la consigne il y aura la réalisation d'une de commande du type :

$$y = K \cdot (\text{consigne} - \text{mesure}) \quad (86)$$

En utilisant un langage de programmation évolué (par exemple le C) la ligne de code sera la transcription littérale de (86). Le programmeur devra décider de la définition et du type des variables, principalement du nombre d'octets utilisés pour stocker ses variables. Le Tableau 5 résume par exemple les types utilisés en langage C (norme ANSI)

Tableau 5 : Types ANSI C

Types entiers	char	8 bits
	short int	16 bits
	long int	32 bits
Types flottants (mantisse et exposant)	float	32 bits
	double	64 bits
	longdouble	128 bits

Une fois le code écrit en C l'exécution du programme sur un processeur cible ne peut se faire que par l'intermédiaire d'une compilation du programme. A chaque processeur est donc adapté une version de compilateur.

Les performances du programme (sa rapidité d'exécution) dépendront donc en partie de la qualité du compilateur. (Par exemple, une multiplication s'effectue t'elle bêtement par un décalage de tous les bits ou bien des solutions astucieuses sont-elles implémentées pour arrêter le décalage selon le contexte ?)

Le principal déterminant de la performance est cependant le processeur : son type (8 bits, 16 bits ou 64 bits) et son cadencement (en MHz ou GHz) sont les deux plus importants paramètres. Coder les variables sur 64 bits (double) dans le programme coûtera cher en temps de calcul si le processeur est un 8 bits. Le type de mémoire et leurs temps d'accès peuvent aussi être déterminants. Enfin, certains processeurs disposent d'un processeur spécifiquement dédié aux opérations mathématiques (coprocesseur mathématique).

Il n'est pas possible de dire exactement, pour tel type de compilateur associé à tel type de processeur, le temps que prendra par exemple une multiplication. En pratique on effectue une série de tests types (benchmark) et on affiche un temps moyen d'exécution.

En résumé, on peut être tenté, pour ne pas s'embêter, de coder toutes les variables en virgule flottante. Mais il faut bien savoir que cela a un coût en terme de temps de calcul, surtout si le processeur n'est pas adapté. Pour fixer les idées on peut dire qu'il y a un rapport 20 entre une addition réalisée en "short" et une addition réalisée en virgule flottante.

Enfin, l'utilisation de fonctions mathématiques (sinus, exponentielles, etc...) peut aussi avoir un coût très lourd si le travail n'est pas réalisé par un coprocesseur mathématique dédié. Le compilateur et sa librairie mathématique joue là encore un grand rôle.

### La quantification imposée par le processeur

Au-delà de l'aspect performance et temps de calcul, la quantification de l'information doit parfois aussi être prise en compte. Rappelons qu'un entier codé sur N bits peut prendre seulement  $2^N - 1$  valeurs différentes. Avant d'effectuer les calculs mathématiques dans le corps du programme il est donc judicieux d'appliquer un

facteur d'échelle de manière à utiliser l'amplitude maximale possible (entre 0 et  $2^N - 1$ ).

### **La quantification imposée par les conversions analogique et numérique**

Les convertisseurs analogique-numérique et numérique-analogique sont principalement caractérisés par leur amplitude de signal analogique d'entrée (généralement en volts, par exemple +/- 10V) et par leur résolution en bits (8 ou 10 bits pour les plus courants, jusqu'à 16 bits pour les convertisseurs haut de gamme).

Prenons l'exemple d'un capteur gyromètre transformant une vitesse de rotation en tension. Compte tenu de l'application le signal utile varie entre 2,4V et 2,6V. Si le signal mesuré est branché directement sur un convertisseur 0-10V 10bits alors un problème de conditionnement du signal se pose. En effet la quantification est de  $\frac{10V}{2^{10}-1} = 0,098V$ . Les 0,2V de signal utile seront donc discrétisés en  $\frac{0,2}{0,098} = 20$  intervalles seulement !

Si la discrétisation du signal utile est trop grossière cela peut se traduire par une diminution des performances de l'asservissement, voire une instabilité. Le système n'est en effet plus linéaire et il n'existe pas d'outils théoriques permettant de prévoir les performances du système bouclé.

## **8.2 Transport de l'information : les bus et les protocoles**

Pas question de faire ni même d'ébaucher un cours sur les bus et les protocoles ici. Mais on va quand même donner quelques indications. L'automaticien a l'habitude de simplifier le problème qu'il se pose en dessinant un schéma simplifié de son problème d'asservissement (du type de la Figure 20 page 45) pour pouvoir appliquer les théories de la commande. Cependant il arrive bien souvent que l'information transite à l'intérieur du système, soit entre différents ordinateurs effectuant différentes parties du travail soit entre ordinateur et périphériques, par l'intermédiaire de bus informatique. A chaque type de bus correspond un ou plusieurs protocoles qui peuvent poser des problèmes pour l'automaticien :

- les données sont-elles discrétisées ? (le protocole peut par exemple ne permettre de transmettre que des paquets de données de 16 bits),
- le bus induit-t'il un retard ? (le composant mettant en forme les données du ordinateur pour les publier sur le bus a certaines performances qui peuvent induire un délai)
- le bus est-il déterministe ? (Certains bus ne garantissent pas que les données arriveront en un temps donné),
- le bus est-il sûr ? (Certains bus ne garantissent pas non plus que les données arriveront bien)

### **8.3 Capteurs et actionneurs "intelligents", "smart sensors"**

De plus en plus souvent les capteurs et les actionneurs sont associés à leur électronique de puissance, électronique de commande et de conditionnement de signal. L'interface avec le capteur ou l'actionneur se fait donc via un protocole de communication (par exemple via un bus RS232).

L'intérêt est de véhiculer non pas des signaux analogiques, tensions ou courants, mais des signaux numériques beaucoup moins sensibles aux éventuelles perturbations.

On retrouve donc, entre le capteur ou l'actionneur et le ordinateur, la présence d'un bus et d'un protocole de communication.

### **8.4 La philosophie "hardware in the loop"**

Les systèmes embarqués sont de plus en plus complexes et touchent des domaines de plus en plus variés : automobile, satellites, aéronautique, UAV, systèmes d'arme, sous-marins, systèmes, etc... Les systèmes à piloter ainsi que les systèmes de commande sont de plus en plus complexes.

Les techniques de simulation « Hardware In the Loop » (HIL) sont de plus en plus utilisées dans le développement et dans le test de ces systèmes embarqués. L'idée est de développer entièrement le système de contrôle sur une plateforme de test en utilisant le maximum de composants réels et en ne remplaçant que le minimum par la simulation. Par exemple en aéronautique l'« Iron Bird » regroupe dans un hangar tous les systèmes réels de l'avion en cours de développement. Seule la mécanique du vol est simulée : les sorties des actionneurs sont mesurées, rentrées dans un modèle de simulation numérique, puis les mesures effectuées normalement par les capteurs sont calculées et restituées par le modèle.

L'intérêt est de diminuer de manière importante les coûts et les délais en économisant le plus grand nombre possible de tests sur le système réel en situation. Lors des premiers tests réels les techniques « Hardware In the Loop » permettent d'avoir effectué la plus grande part du développement des lois de commande.

### **8.5 Le prototypage rapide en automatique**

Un peu dans le même ordre d'idée que le « Hardware In the Loop » le prototypage rapide permet de réaliser la conception des lois de commande en simulation puis de transporter rapidement (en un simple click !) le code du contrôleur simulé vers la cible temps réel implémentée sur le système réel.

La société Mathworks [9] investit massivement dans ce domaine en ayant enrichi son logiciel de calcul scientifique (Matlab) avec un logiciel de simulation graphique très évolué (Simulink) puis en développant des outils de prototypage rapide associés (Real Time Workshop, Embedded Target For...).

## 9. ANNEXE

### 9.1 Annexe 1 : Résolution de l'équation polynomiale de Bezout

Il s'agit de trouver R et S vérifiant :

$$P(q^{-1}) = A(q^{-1}).S(q^{-1}) + B(q^{-1}).R(q^{-1})$$

Où P, A et B sont connus.

Dans le cas où P, A, B, R et S sont des nombres entiers la solution de l'algorithme est donnée par l'algorithme de division Euclidienne étendu [8].

Dans le cas où P, A, B, R et S sont des polynômes, l'algorithme est donné ci-dessous, programmé en matlab ([1]).

```
function [Rp,Sp,nrp,nsp]=bezoutd(A,B,Hs,Hr,P)
%function [Rp,Sp,nrp,nsp]=bezoutd(A,B,Hs,Hr,P)
%solves AHsSp+BHrRp=P by coefficient comparison. Delay and discretization delay need to be integrated in B
%(zeros at the beginning).
%
%inputs:
%A=[a0 a1 ... aNa] ... vector of model denominator coefficients A=a0 + a1z^(-1) + a2z^(-2) +...+ aNaz^(-Na)
%B=[b0 b1 ... bNb] ... vector of model numerator coefficients B=b0 + b1z^(-1) + b2z^(-2) +...+ bNbz^(-Nb)
%Hs=[hs0 hs1 ... hsNhs] ... vector of controller denominator fixed part Hs=hs0 + hs1z^(-1) +...+ hsNhsz^(-Nhs)
%Hr=[hr0 hr1 ... hrNhr] ... vector of controller denominator fixed part Hr=hr0 + hr1z^(-1) +...+ hrNhrz^(-Nhr)
%P=[p0 p1 ... pNp] ... vector of desired polynomial coefficients P=p0 + p1z^(-1) + p2z^(-2) +...+ pNpz^(-Np)
%outputs:
%Rp=[rp0 rp1 rp2 ...] ... vector of coefficients for resulted controller numerator
%Sp=[sp0 sp1 sp2 ...] ... vector of coefficients for resulted controller denominator
%nrp ... order of Rp
%nsp ... order of Sp
%
%written by: J. Langer, I.D. Landau, H. Prochazka
%7th june 2002

PRECISION=1e-16;

D=size(A);
if D(1)>1, A=A'; end;
```

```

D=size(B);
if D(1)>1, B=B'; end;
D=size(Hs);
if D(1)>1, Hs=Hs'; end;
if D(1)==0, Hs=1; end;
D=size(Hr);
if D(1)>1, Hr=Hr'; end;
if D(1)==0, Hr=1;end;
D=size(P);
if D(1)>1, P=P'; end;

na=length(A)-1;
nb=length(B)-1;
np=length(P)-1;
nhs=length(Hs)-1;
nhr=length(Hr)-1;

if (nhs>0), Ah=real(convz(A,Hs)); else Ah=A*Hs; end; % Ah = A * Hs
nah=length(Ah) -1;
if (nhr>0), Bh=real(convz(B,Hr)); else Bh=B*Hr; end;
nbh=length(Bh) -1;
if (np>nah+nbh-1), disp('Bezout error: too many poles');end;

% increase size of P using zeros if necessary
if (np<nah+nbh-1),
%
% set remaining poles onto a circle with radius rmin
P=[P zeros(1,nah+nbh-1-np)];
rootsPdes=roots(P);
nextra=nah+nbh-1-np;
rmin=1e-16;
angle=[0:nextra-1]'/nextra*2*pi;
j=sqrt(-1);
rootsPextra=rmin*exp(j*angle);
P=poly([rootsPdes;rootsPextra]);
np=nah+nbh-1;

end;
P,

nsp=nbh-1;
nrp=nah-1;

%matrix is smaller than vector PD

```

```

if (np>nah+nbh-1),
    disp('Order of model denominator is too low! Add a polynom of higher order to Hs or Hr. ');
end;

% ns=nsp+nhs
% nr=nrp+nhr

M=[];
for j=1:nsp+1,
    V=[];
    if (j>1), V=[V ; zeros(j-1,1)]; end;% zeros in front of Ah
    V=[V ; Ah'];% Ah
    if (j<=nsp), V=[V ; zeros(nsp+1-j,1)]; end;% zeros behind Ah
    if (length(V)~=nah+nbh), disp('bezoutb: error V'); end;
    M=[M V]; % add one column to M
end;

for j=1:nrp+1,
    V=[];
    if (j>1), V=[V ; zeros(j-1,1)]; end;
    V=[V ; Bh'];
    if (j<=nrp), V=[V ; zeros(nrp+1-j,1)]; end;
    if (length(V)~=nah+nbh), disp('bezoutb: error V'); end;
    M=[M V];
end;

D=size(M);
if (D(1)~=nah+nbh), disp('bezoutb: error size M row'); end;
if (D(2)~=nah+nbh), disp('bezoutb: error size M column'); end;

% make P column vector
P=P';

global M1;
M1=M;

X= M\P;
% coefficients are real values
X=real(X);

% make X row vector
X=X';

```

## Commande des systèmes par ordinateur

```
Sp=X(1:nsp+1);  
Rp=X(nsp+2:nsp+nrp+2);
```



## 10. BIBLIOGRAPHIE

- [1] I. D. Landau, "*Commande des systèmes*", Edition Lavoisier 2002.
- [2] C. Nouals, "Théorie du Signal", Notes de cours ENSICA
- [3] F. Vincent, "*Traitement Numérique du Signal*", Notes de cours ENSICA.
- [4] D. Arzelier, "*Représentation et analyse des systèmes, notes de cours*", Edition ENSICA. Voir aussi <http://www.laas.fr/~arzelier/>
- [5] J. Bordeneuve-Guibé, "*Commande des systèmes linéaires*", Notes de cours ENSICA.
- [6] Y. Briere, "*Identification*", Notes de cours ENSICA.
- [7] L. Ljung, "*System Identification, Theory for the User*", Prentice Hall 1987.
- [8] Wikipedia , équation de Bezout : [http://en.wikipedia.org/wiki/Bézout's\\_identity](http://en.wikipedia.org/wiki/Bézout's_identity)
- [9] <http://www.mathworks.com>



# 11. INDEX

<b>A</b>	
actionneur intelligent .....	21
amortissement .....	15
ANSI C .....	60
Approximation numérique d'équation différentielle .....	28
ARMAX .....	35
<b>B</b>	
BJ35 .....	
Blocage d'un signal .....	56
bloqueurs d'ordre 0 .....	22
BOZ .....	22
bus.....	21, 61
<b>C</b>	
CAN.....	21
capteur intelligent .....	21
causal .....	6, 9
Causalité .....	10
CNA.....	21
compilateur .....	60
convertisseur analogique / numérique.....	21
convertisseur numérique / analogique.....	21
convertisseurs analogique-numérique .....	61
coprocesseur mathématique .....	60
Correspondances pôles - zéros.....	31
<b>D</b>	
Diagramme de Bode .....	18
Discrétisation de fonction de transfert continues .....	27
division polynomiale.....	6
<b>E</b>	
échelon unitaire.....	11
équation d'état .....	17
Erreur statique.....	42
Estimation de la réponse fréquentielle .....	39
Estimation de la réponse impulsionnelle.....	38
Euler .....	29
<b>F</b>	
facteur d'échelle .....	61
facteur d'oubli .....	36
<i>fenêtre de Hamming</i> .....	39
filtre anti-repliement .....	25
fonction de sensibilité .....	42
fonction de transfert discrète.....	9
<b>G</b>	
gain statique.....	11
gain statique unitaire.....	42
<b>I</b>	
Identification.....	32
Identification non paramétrique .....	38
intercorrélation.....	38
<b>L</b>	
lieu des racines.....	43
lieu d'Evans.....	43
LTI.....	8
<b>M</b>	
marge de gain.....	47
marge de module.....	46
marge de phase .....	47
méthode des moindres carrés récursifs .....	32
Méthode des trapèzes.....	29
modèle ARX .....	32
<b>O</b>	
OE.....	35
one step ahead prediction.....	35
opérateur de retard .....	9
<b>P</b>	
période d'échantillonnage (choix) .....	26
Placement de pôles .....	54
Pôles .....	11
Poursuite .....	56
<i>prédiction</i> .....	32
protocoles.....	61
Pseudo Random Binary Signal .....	34
pulsation .....	15
<b>Q</b>	
quantification .....	23, 24, 60
<b>R</b>	
R,S,T.....	41
rapidité.....	15
rayon de convergence .....	5, 9
Régulation.....	55

Rejet de perturbations .....	42
Rejet d'une perturbation harmonique .....	56
réponse à un échelon.....	11
réponse fréquentielle.....	18
Réponse impulsionnelle.....	10
Réponses indicielle .....	10
retard.....	23
root locus .....	43

### S

signal discret .....	5
smart sensors.....	62
stabilité.....	11
Systèmes du premier ordre.....	14
Systèmes du second ordre.....	15

### T

Théorème de Nyquist.....	46
théorème de Shannon.....	19, 24
théorème du retard .....	6
Transformée en z .....	5
transformée unilatérale .....	5
Tustin.....	29

### V

virgule flottante.....	60
------------------------	----

### Z

zero holder hold .....	23
zéros.....	11
zoh .....	23

# Commande des systèmes par ordinateur

Yves Briere  
ENSICA

Notes de cours  
2007



# TABLE DES MATIERES

Table des matières .....	1
1. Introduction .....	1
2. Rappels et prérequis .....	3
3. Signaux et systèmes discrets .....	5
3.1 Signaux discrets .....	5
3.1.1 Transformée en z .....	5
3.1.2 Signaux types .....	7
3.2 Systèmes discrets .....	9
3.2.1 Fonction de transfert discrète, définition .....	9
3.2.2 Notation "opérateur de retard $q^{-1}$ " .....	10
3.2.3 Réponse impulsionnelle .....	11
3.2.4 Causalité .....	11
3.2.5 Réponses indicielle .....	11
3.2.6 Régime stationnaire (gain statique) .....	12
3.2.7 Pôles, zéros, stabilité .....	12
3.2.8 Systèmes du premier ordre .....	15
3.2.9 Systèmes du second ordre .....	16
3.2.10 Calcul itératif de la réponse d'un système .....	18
3.2.11 Représentation par équation d'état .....	18
3.3 Réponse fréquentielle des systèmes échantillonnés .....	19
3.3.1 Diagramme de Bode d'un système échantillonné .....	19
3.3.2 Où l'on voit apparaître le théorème de Shannon .....	20
4. Systèmes continus échantillonnés .....	23
4.1 Structure des systèmes commandés par ordinateur .....	23
4.2 Échantillonnage des systèmes continus .....	24
4.3 Effets de l'échantillonnage .....	25
4.4 Où l'on revoit le théorème de Shannon .....	26
4.5 Discrétisation de fonction de transfert continues .....	29
4.5.1 Approximation numérique d'équation différentielle .....	30
4.5.2 Superposition des réponses indicielles .....	32
4.5.3 Correspondances pôles - zéros .....	33
4.5.4 Bilan .....	33
5. Identification des systèmes échantillonnés .....	35
5.1 Identification d'un modèle ARX par la méthode des moindres carrés récurifs .....	35
5.2 L'identification paramétrique en général et en pratique .....	37

5.2.1	Choix du signal d'excitation, traitement des mesures .....	37
5.2.2	Choix d'une structure de modèle.....	38
5.2.3	Choix d'un critère à minimiser.....	38
5.2.4	Choix d'un algorithme.....	38
5.2.5	Validation.....	39
5.2.6	Utilisation de logiciels .....	39
5.3	Identification "simple" .....	39
5.4	Identification non paramétrique .....	41
5.4.1	Estimation de la réponse impulsionnelle .....	41
5.4.2	Estimation de la réponse fréquentielle.....	42
6.	Systèmes bouclés .....	45
6.1	Fonction de transfert en boucle fermée .....	45
6.2	Erreur statique .....	46
6.3	Rejet de perturbations.....	46
6.4	Stabilité des systèmes à simple gain de rétroaction.....	47
6.5	Analyse des systèmes bouclés dans le domaine fréquentiel : stabilité.....	49
6.5.1	Théorème de Nyquist dans le cas où le système est stable en boucle ouverte (critère du revers) .....	50
6.5.2	Théorème de Nyquist dans le cas où le système est instable en boucle ouverte .....	50
6.5.3	Interprétations du lieu de Nyquist.....	50
7.	Méthodes de régulation des systèmes échantillonnés .....	53
7.1	Introduction .....	53
7.2	Structure d'un système commandé par ordinateur, hypothèses.....	54
7.2.1	Discretisation, choix de la période d'échantillonnage.....	54
7.2.2	Discretisation d'un correcteur continu .....	55
7.3	Régulateur PID numérique .....	55
7.3.1	Cas 1 .....	55
7.3.2	Cas 2 .....	58
7.4	Placement de pôles .....	59
7.4.1	Choix des pôles en boucle fermée .....	60
7.4.2	Régulation : calcul de R et S.....	60
7.4.3	Poursuite : calcul de T .....	61
7.5	Poursuite et régulation à objectifs indépendants .....	62
8.	Technologie de la commande par ordinateur.....	65
8.1	Calculateurs et interfaces numériques / analogiques.....	65
8.2	Transport de l'information : les bus et les protocoles .....	67
8.3	Capteurs et actionneurs "intelligents", "smart sensors" .....	68
8.4	La philosophie "hardware in the loop" .....	68
8.5	Le prototypage rapide en automatique .....	68
9.	Annexe .....	69
9.1	Annexe 1 : Résolution de l'équation polynomiale de Bezout.....	69
10.	Bibliographie.....	73
11.	Index.....	1







# 1. INTRODUCTION



Figure 1 : commande d'un systèmes de télémédecine par ordinateur, vu par Claude Serre



## 2. RAPPELS ET PREREQUIS

Ce polycopié ne comporte pas de rappels sur les systèmes continus. S'adressant aux étudiants en deuxième année de l'ENSICA il suppose connu les enseignements de première année et de seconde année en automatique et traitement du signal.

En première année le cours de théorie du signal [2] a rendu familier les notions de :

- transformée de Fourier, transformée de Laplace
- corrélation, convolution

Le cours de traitement numérique du signal [3] a abordé en plus :

- la discrétisation et la quantification des signaux.
- la transformée en z et la transformée de Fourier discrète.
- les filtres numériques RIF et RII.

En automatique [4] les bases de la représentation et de l'analyse des systèmes continus ont été posées. On ne reviendra pas dans ce cours sur les notions de :

- modèles mathématique des systèmes linéaires à temps continus à temps invariant (LTI) : fonction de transfert, équation d'état, pôles, zéros et dynamique.
- commandabilité et observabilité.
- stabilité des systèmes en boucle ouverte : pôles, critères de Routh-Hurwitz.
- stabilité des systèmes bouclés : critère de Nyquist, critère de Bode.
- lieu des racines.

En deuxième année les notions suivantes sont acquises [5] :

- synthèse des correcteurs par méthodes fréquentielles (avance et retard de phase, PID)
- commande par retour d'état (placement de pôles), commande Linéaire Quadratique (LQG)
- reconstruction de l'état par Observateur, observateur optimal de Kalman.



## 3. SIGNAUX ET SYSTEMES DISCRETS

### 3.1 Signaux discrets

#### 3.1.1 Transformée en z

Pour une approche mathématiquement rigoureuse de représentation des signaux discrets par la transformée en z consulter [3] et sa bibliographie. Dans ce cours on se cantonne à une approche utilitariste des outils et méthodes mathématiques tout en essayant de ne pas contredire fondamentalement la rigueur mathématique.

Un signal discret  $s(k)$  est une séquence de nombres réels appelés échantillons. L'indice  $k$  est généralement associé à un instant d'échantillonnage :  $t = k.T_e$ . La période d'échantillonnage  $T_e$  est supposée constante. Par définition la transformée en z du signal  $s(k)$  est la fonction de la variable  $z$ , notée abusivement  $s(z)$  :

$$s(z) = Z(s(k)) = \sum_{k=0}^{\infty} s(k) \cdot z^{-k} \quad (1)$$

La transformée définie ici est plus précisément appelée transformée unilatérale puisque les termes correspondant à  $k < 0$  ne sont pas pris en compte. Ceci n'a pas d'importance puisque les signaux que nous utilisons sont systématiquement causaux<sup>1</sup> :  $s(k) = 0$  pour  $k < 0$ .

Il y a bien évidemment un lien avec la transformée de Laplace et la transformée de Fourier (voir [3])

La fonction de la variable complexe  $s(z)$  est une série et ne converge pas forcément pour toute valeur de  $z$ . En toute rigueur il faudrait associer le rayon de convergence à chaque fonction en  $z$  :  $\{s(z) \text{ converge pour } |z| > \alpha\}$ .

Les automaticiens pour qui la transformée en z n'est bien souvent qu'une façon bien commode de regrouper des termes de récurrence utilisent souvent la pseudo-variable<sup>2</sup>  $q$ .

A partir d'une transformée en z  $s(z)$  il est possible de retrouver la séquence  $s(k)$  correspondante en appliquant la transformation inverse, dont on ne donnera pas la formule ici. A la place de faire appel à cette formule compliquée il est possible d'utiliser d'autres méthodes pour retrouver la séquence  $s(k)$  lorsque la fonction  $s(z)$  est donnée sous forme de fraction rationnelle :

<sup>1</sup> Pour une justification du terme « causal » voir page 11

<sup>2</sup> Voir page 10

- division polynomiale
- décomposition de la fraction rationnelle en éléments simples puis utilisation de transformations élémentaires telles que celles données dans le Tableau 2

La propriété fondamentale de la transformée en z est qu'elle est linéaire (la transformée en z d'une composition linéaire de signaux est la composition linéaire de la transformée en z de chaque signal). Les autres propriétés fondamentales sont données dans le Tableau 2.

Tableau 1 : propriétés de la transformée en z

1	$s(z) = Z(s(k)) = \sum_{k=0}^{\infty} s(k) \cdot z^{-k}$
2	$Z(\alpha \cdot s_1(k) + \beta \cdot s_2(k)) = \alpha \cdot Z(s_1(k)) + \beta \cdot Z(s_2(k))$
3	$Z(k \cdot s(k)) = -z \cdot \frac{ds(z)}{dz}$
4	$Z(c^k \cdot s(k)) = s\left(\frac{z}{c}\right)$
5	$Z(s(k-1)) = s(k=-1) + z^{-1} \cdot s(z)$
6	$Z(s(k+1)) = z \cdot s(z) - z \cdot s(k=0)$
7	$s(k=0) = \lim_{z \rightarrow \infty} (s(z))$
8	$s(k \rightarrow \infty) = \lim_{z \rightarrow 1} \left( \frac{z-1}{z} \cdot s(z) \right)$

Remarque 1 : la propriété 5 ne prend pas en compte le fait que le signal est causal : en pratique  $s(k=-1) = 0$ .

Remarque 2 : la propriété 5 justifie le fait que le terme  $z^{-1}$  soit appelé "opérateur de retard unitaire", cette propriété est aussi appelée "théorème du retard".

Remarque 3 : les propriétés 7 et 8 sont données sous réserve de convergence.

Transformées inverses

### Exemple 1 : transformée inverse par division polynomiale

On considère un signal  $s(k)$  dont la transformée en z est donnée par la fraction rationnelle :

$$S(z) = \frac{0.1 \cdot z^2 + 0.1 \cdot z}{z^3 - 2.6 \cdot z^2 + 2.2 \cdot z - 0.6} \quad (3.1)$$

L'objectif est d'obtenir  $S(z)$  sous forme polynomiale :



$$S(z) = x_0 + x_1 \cdot z^{-1} + x_2 \cdot z^{-2} + \dots \text{etc.} \tag{3.2}$$

Cette suite est généralement infinie, la réponse temporelle est donnée par la succession des termes  $\{x_i\}$ . En traitant l'équation (3.1) comme une simple division l'algorithme de division classique (celui appris en CM2) permet d'obtenir par récurrence tous les termes de la suite (3.2) :

$0.1 \cdot z^2$	$+ 0.1 \cdot z$				$z^3 - 2.6 \cdot z^2 + 2.2 \cdot z - 0.6$
$0.1 \cdot z^2$	$- 0.26 \cdot z$	$+ 0.22$	$- 0.06 \cdot z^{-1}$		
0	$0.36 \cdot z$	$- 0.22$	$0.06 \cdot z^{-1}$		
	$0.36 \cdot z$	$- 0.936$	$0.792 \cdot z^{-1}$	$- 0.216 \cdot z^{-2}$	
0	$0.716$	$- 0.732 \cdot z^{-1}$	$\dots$	$\dots$	

Les éléments successifs de la suite  $s(k)$  sont donc  $s(k) = \{0, 0.1, 0.36, 0.716, \dots\}$ . Pour obtenir les termes suivants il faudrait continuer la division.

**Exemple 2 : transformée inverse par décomposition en éléments simples**

La deuxième méthode présentée consiste à décomposer la fraction rationnelle (3.1) en éléments simples puis d'utiliser les tables de transformées en  $z$  (Tableau 2). Pour notre exemple cela donne :

$$S(z) = \frac{0.1 \cdot z^2 + 0.1 \cdot z}{z^3 - 2.6 \cdot z^2 + 2.2 \cdot z - 0.6} = \frac{-z}{z-1} + \frac{0.5 \cdot z}{(z-1)^2} + \frac{z}{z-0.6}$$

La réponse temporelle est donc composée des trois éléments :

$$s(k) = -(1 \ 1 \ 1 \ \dots) + 0.5 \cdot (1 \ 2 \ 3 \ \dots) + (1 \ 0.6 \ 0.6^2 \ \dots)$$

$$\text{soit : } s(k) = \begin{cases} s(0) = -1 + 0.5 + 1 \\ s(2) = -1 + 0.5 \cdot 2 + 0.6 \cdot 0.5 \\ s(3) = -1 + 0.5 \cdot 3 + 0.6^2 \\ s(4) = \dots \end{cases}$$

**3.1.2 Signaux types**

Le Tableau 2 donne les transformée en  $z$  de certains signaux fondamentaux. La Figure 2 représente l'allure de quelques signaux types.

Tableau 2 : transformée en z élémentaires

$s(k)$	$S(z)$
signal temporel	Transformée en z
$\delta(k)$ : impulsion unité	1
$u(k)$ : échelon unité	$\frac{z}{z-1}$
$k \cdot u(k)$	$\frac{z}{(z-1)^2}$
$c^k \cdot u(k)$	$\frac{z}{z-c}$
$k \cdot a^k \cdot u(k)$	$\frac{a \cdot z}{(z-a)^2}$
$\sin(\omega \cdot k) \cdot u(k)$	$\frac{z \cdot \sin(\omega)}{z^2 - 2 \cdot z \cdot \sin(\omega) + 1}$
$\cos(\omega \cdot k) \cdot u(k)$	$\frac{z \cdot (z - \cos(\omega))}{z^2 - 2 \cdot z \cdot \sin(\omega) + 1}$
$a^k \cdot \sin(\omega \cdot k) \cdot u(k)$	$\frac{a \cdot z \cdot \sin(\omega)}{z^2 - 2 \cdot a \cdot \cos(\omega) \cdot z + a^2}$
$a^k \cdot \cos(\omega \cdot k) \cdot u(k)$	$\frac{z \cdot (z - a \cdot \cos(\omega))}{z^2 - 2 \cdot a \cdot \cos(\omega) \cdot z + a^2}$

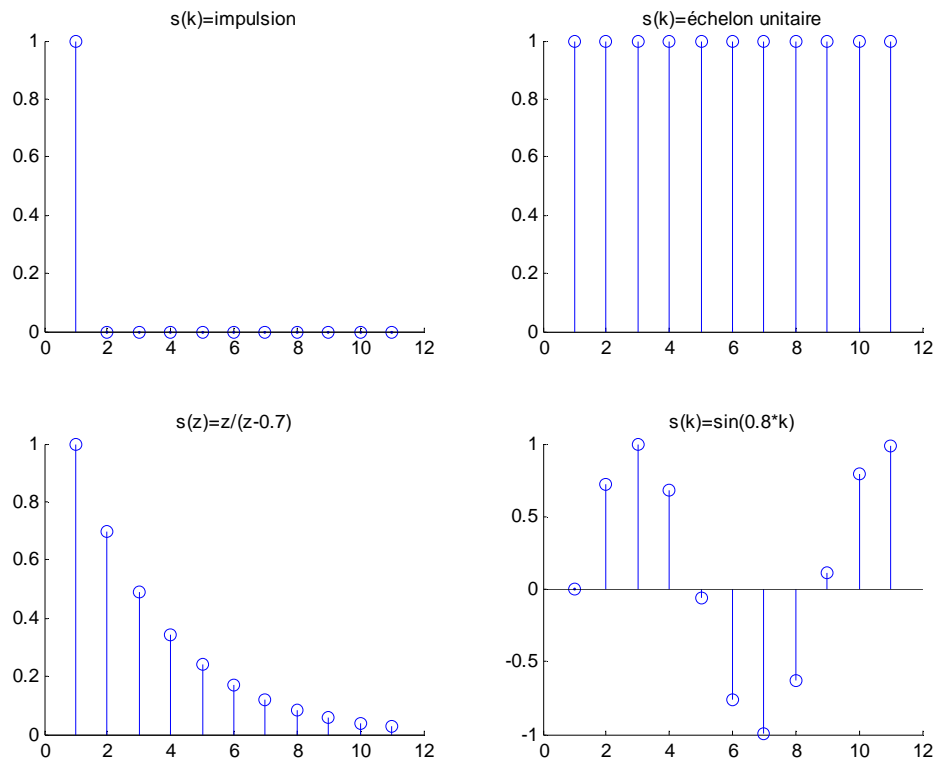


Figure 2 : réponses de quelques signaux types

### 3.2 Systèmes discrets

#### 3.2.1 Fonction de transfert discrète, définition

On considère maintenant un système discret linéaire et invariant dans le temps (LTI) représenté sur la Figure 3.

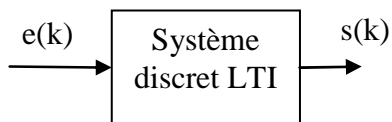


Figure 3 : système discret, notations

La relation entre la séquence d'entrée et la séquence de sortie est linéaire et définie par un ensemble de coefficients constants  $a_{1:n}$  et  $b_{1:m}$  :

$$\begin{aligned} s(k) + a_1 \cdot s(k-1) + a_2 \cdot s(k-2) + \dots + a_n \cdot s(k-n) = \\ b_0 \cdot e(k) + b_1 \cdot e(k-1) + b_2 \cdot e(k-2) + \dots + b_m \cdot e(k-m) \end{aligned} \quad (2)$$

On peut appliquer la transformée en  $z$  à chaque membre de l'équation puis en appliquant en récurrence la propriété 5 (théorème du retard) on obtient la fonction de transfert discrète du système :

$$\frac{s(z)}{e(z)} = F(z) = \frac{b_0 + b_1 \cdot z^{-1} + b_2 \cdot z^{-2} + \dots + b_m \cdot z^{-m}}{1 + a_1 \cdot z^{-1} + a_2 \cdot z^{-2} + \dots + a_n \cdot z^{-n}} \quad (3)$$

Remarque 1 : la fonction de transfert précédente est normalisée en prenant  $a_0 = 1$ .

Remarque 2 : on remarquera que la fonction de transfert est donnée en  $z^{-1}$  plutôt qu'en  $z$ . Cela est justifié par le fait que celle-ci traduit l'équation d'évolution du système physique causal (2).

Le script Matlab permettant de créer un objet Matlab correspondant à une fonction de transfert discrète est donné ci-dessous.

#### Matlab

```
% Création d'un système linéaire discret :
Te=0.1;
sysd = tf([b0 b1 b2],[1 a0 a2 a3],Te,'variable','z^-1');
```

On notera l'utilisation de l'argument 'variable', 'z^-1'.

### 3.2.2 Notation "opérateur de retard $q^{-1}$ "

La fonction de transfert est une fonction de la variable complexe  $z$ . Il faudrait donc en toute rigueur l'associer à son rayon de convergence. En pratique on ne s'embarrasse jamais de préciser cette zone de convergence. C'est pourquoi on trouve souvent la notation utilisant l'opérateur  $q^{-1}$  de décalage à la place de la variable complexe  $z^{-1}$  (4).  $F(q)$  n'est alors plus à proprement parler une fonction mais on lui appliquera les mêmes manipulations algébriques qu'à  $F(z)$ .

$$F(q) = \frac{b_0 + b_1 \cdot q^{-1} + b_2 \cdot q^{-2} + \dots + b_m \cdot q^{-m}}{1 + a_1 \cdot q^{-1} + a_2 \cdot q^{-2} + \dots + a_n \cdot q^{-n}} \quad (4)$$

Attention : selon les auteurs l'opérateur qui remplace  $z^{-1}$  est noté  $q$  (Matlab) ou  $q^{-1}$  (Landau [1]). Dans la suite du document c'est cette dernière convention qui sera appliquée.

L'utilisation de l'opérateur de retard  $q^{-1}$  à la place de la variable complexe  $z^{-1}$  permet de traiter du cas des fonctions de transfert à coefficients non constant, cas non prévu par la transformation en  $z$ .

### 3.2.3 Réponse impulsionnelle

Essayons d'appliquer au système un signal d'entrée particulier : une impulsion unitaire. La transformée en  $z$  du système est obtenue en multipliant la fonction de transfert par la transformée en  $z$  de l'impulsion unitaire : 1. On obtient :

$$s(z) = \frac{b_0 + b_1 \cdot z^{-1} + b_2 \cdot z^{-2} + \dots + b_m \cdot z^{-m}}{1 + a_1 \cdot z^{-1} + a_2 \cdot z^{-2} + \dots + a_n \cdot z^{-n}} \times 1 \quad (5)$$

Cela justifie le fait que la fonction de transfert du système peut aussi s'appeler "réponse impulsionnelle".

#### Matlab

```
% Réponse impulsionnelle d'un système :
[y,t] =impulse(sysd);
```

### 3.2.4 Causalité

Un système est dit causal si les sorties ne dépendent que du passé et non pas du futur. La conséquence est que la réponse impulsionnelle de  $F(z)$  ne peut pas comporter de terme non nuls pour  $k < 0$ . Cela justifie la confusion sémantique entre "système causal" et "signal causal", c'est-à-dire ne comportant pas de termes non nuls pour  $k < 0$ .

Attention : à la différence des systèmes continus un système physique (causal) discret peut très bien être modélisé avec un degré  $m$  supérieur au degré  $n$  : le système peut avoir plus de zéros que de pôles.

La transformée en  $z$  représentant des signaux échantillonnés elle peut tout aussi bien représenter les mesures prises à différents endroits d'une poutre. Le temps étant remplacé ici par une abscisse la notion de causalité ne s'applique pas ! Attention : beaucoup des propriétés de systèmes échantillonnés que l'on étudie sont liés à la causalité, prudence donc si l'on applique ces méthodes à d'autres domaines !

### 3.2.5 Réponses indicielle

Pour caractériser des systèmes on s'intéresse souvent à leur réponse impulsionnelle ou, plus souvent, à leur réponse indicielle. Pour calculer la réponse

à un échelon d'un système il faut donc multiplier la fonction de transfert par la transformée en z d'un échelon unitaire :

$$s(z) = \frac{b_0 + b_1 \cdot z^{-1} + b_2 \cdot z^{-2} + \dots + b_m \cdot z^{-m}}{1 + a_1 \cdot z^{-1} + a_2 \cdot z^{-2} + \dots + a_n \cdot z^{-n}} \times \frac{1}{1 - z^{-1}} \quad (6)$$

Il faudrait ensuite calculer la transformée en z inverse de s(z). En pratique, si l'on doit faire le calcul à la main on décompose en éléments simples puis on utilise des tables. Avec Matlab c'est plus simple. Voici par exemple le script pour obtenir les 20 premiers échantillons de la réponse indicielle d'un système :

**Matlab**

```
% Réponse indicielle d'un système :
[y,t] =step(sysd,20);
```

### 3.2.6 Régime stationnaire (gain statique)

Le gain statique est obtenu en appliquant une entrée indicielle et en calculant la valeur du régime permanent. En appliquant le théorème de la valeur finale il vient :

$$F_0 = \lim_{z \rightarrow 1} \left( \frac{z-1}{z} \cdot \frac{b_0 + b_1 \cdot z^{-1} + b_2 \cdot z^{-2} + \dots + b_m \cdot z^{-m}}{1 + a_1 \cdot z^{-1} + a_2 \cdot z^{-2} + \dots + a_n \cdot z^{-n}} \times \frac{1}{1 - z^{-1}} \right) \quad (7)$$

Soit :

$$F_0 = \frac{\sum_{i=0}^{i=m} b_i}{1 + \sum_{i=1}^{i=n} a_i} \quad (8)$$

Le gain statique est obtenu donc simplement en faisant le quotient de la somme des coefficients.

### 3.2.7 Pôles, zéros, stabilité

Comme pour les systèmes continus la dynamique et la stabilité des systèmes dépend des propriétés du dénominateur de la fonction de transfert.

$$F(z) = \frac{e_0 \cdot (1 + e_1 \cdot z^{-1}) \cdot (1 + e_2 \cdot z^{-1}) \cdots (1 + e_m \cdot z^{-1})}{(1 + c_1 \cdot z^{-1}) \cdot (1 + c_2 \cdot z^{-1}) \cdots (1 + c_n \cdot z^{-1})} \quad (9)$$

Les coefficients  $c_i$  sont les pôles de la fonction de transfert, les coefficients  $e_i$  sont les zéros. La fonction de transfert étant au départ composée de coefficients purement réels les coefficients  $c_i$  sont soit réels soit imaginaires par paires conjuguées.

Une fonction de transfert est donc aussi bien définie par la donnée de ses pôles et de ses zéros. Le script Matlab pour créer un système ainsi est donné ci-dessous :

#### Matlab

```
% Système discret défini par ses pôles et zéros :
Te=0.1;
sysd=zpk([c1 c2 c3],[e1 e2],e0,Te);
```

Si l'on décompose la fonction de transfert en éléments simples on obtient une somme d'éléments du type :

$$F(z) = \frac{d_1}{1 + c_1 \cdot z^{-1}} + \frac{d_2}{1 + c_2 \cdot z^{-1}} + \cdots + \frac{d_n}{1 + c_n \cdot z^{-1}} \quad (10)$$

Un système linéaire temps invariant est dit stable si sa réponse impulsionnelle tend asymptotiquement vers 0. Compte tenu de (10) cela nécessite que chaque pôle  $c_i$  soit de module inférieur à 1.

**Propriété** : un système discret LTI est stable si et seulement si tous ses pôles ont leur modules inférieur ou égal à 1.

#### Critère algébrique attestant de la stabilité : critère de Routh

On rappelle que le critère de Routh permet de prédire si les racines d'un polynôme sont toutes à partie réelle négative, donc stables. Énonçons le théorème de Routh dans le cas d'un polynôme  $P'(w)$  :

$$P'(w) = a_0 + a_1 \cdot w + a_2 \cdot w^2 + \cdots + a_n \cdot w^n \quad (11)$$

Dans un premier temps on construit le tableau :

$$\begin{array}{cccccc|c}
 \beta_{n,0} & \beta_{n,1} & \beta_{n,2} & \beta_{n,q-1} & \beta_{n,q} & 0 & \\
 \beta_{n-1,0} & \beta_{n-1,1} & \beta_{n-1,2} & \beta_{n-1,q-1} & \beta_{n-1,q-1} & 0 & \\
 \beta_{n-2,0} & \beta_{n-2,1} & \beta_{n-2,2} & \beta_{n-2,q-1} & \beta_{n-2,q-1} & & \\
 & & & & & & \\
 \beta_{2,0} & \beta_{2,1} & 0 & & & & \\
 \beta_{1,0} & 0 & & & & & \\
 \beta_{0,0} & 0 & & & & & 
 \end{array} \quad (12)$$

Puis on vérifie que tous les premiers éléments de la première colonne ont le même signe : cela garantit que tous les pôles de  $P'(w)$  ont leur partie réelle négative.

Les éléments du tableau sont construits par récurrence ligne par ligne en appliquant les formules suivantes :

Pour  $i = n$  et  $i = n - 1$  :

$$\begin{cases} \beta_{n,j} = a_{n-2j} \\ \beta_{n-1,j} = a_{n-1-2j} \end{cases} \quad (13)$$

Et pour  $i < n-1$  :

$$\beta_{i,j} = \frac{\beta_{i+1,0} \cdot \beta_{i+2,j+1} - \beta_{i+1,j+1} \cdot \beta_{i+2,0}}{\beta_{i+1,0}} \quad (14)$$

Dans le cas qui nous intéresse, un système échantillonné, on cherche plutôt à vérifier que les pôles du polynôme  $P(z)$  sont de module inférieur à un. L'idée est d'effectuer un changement de variable  $z \rightarrow w$  pour pouvoir appliquer le critère de Routh.

$$P'(w) = (1-w)^n \cdot P\left(\frac{1+w}{1-w}\right) \quad (15)$$

Si l'on effectue la transformation (15) alors le fait que  $P'(w)$  ait toutes ses racines à partie réelle négative garantit que  $P(z)$  a toutes ses racines dans le cercle unité. Pour vérifier la stabilité de  $P(z)$  il suffit donc d'effectuer le changement de variable (15) puis d'appliquer le critère de Routh conventionnel (12 13 14).

### Critère algébrique attestant de la stabilité : critère de Jury

Le critère de Jury s'apparente au critère de Routh. Nous le donnons pour un polynôme de degré 2, 3 et 4 du type :



$$P(z) = a_0 + a_1 \cdot z + a_2 \cdot z^2 + \dots + a_n \cdot z^n \quad (16)$$

Les conditions de stabilité sont données ci-dessous (ces conditions sont données dans le cas  $a_n > 0$ , il suffit de multiplier tous les termes par  $-1$  si cette condition n'est pas remplie par  $P(z)$ ) :

$$\begin{aligned}
 n = 2 & \begin{cases} a_0 + a_1 + a_2 > 0 \\ a_0 - a_1 + a_2 > 0 \\ -a_0 + a_2 > 0 \end{cases} \\
 n = 3 & \begin{cases} a_0 + a_1 + a_2 + a_3 > 0 \\ -a_0 + a_1 - a_2 + a_3 > 0 \\ -|a_0| + a_3 > 0 \\ a_0 a_2 - a_1 a_3 - a_0^2 + a_3^2 > 0 \end{cases} \\
 n = 4 & \begin{cases} a_0 + a_1 + a_2 + a_3 + a_4 > 0 \\ a_0 - a_1 + a_2 - a_3 + a_4 > 0 \\ a_4^2 - a_0^2 - |a_0 a_3 - a_1 a_4| > 0 \\ (a_0 - a_4)^2 (a_0 - a_2 + a_4) + (a_1 - a_3)(a_0 a_3 - a_1 a_4) > 0 \end{cases}
 \end{aligned} \quad (17)$$

### 3.2.8 Systèmes du premier ordre

On donne dans la Figure 4 les réponses indicielles de systèmes du premier ordre ainsi que la localisation des pôles dans le plan complexe. Le cercle de rayon unité représente la zone dans laquelle les pôles doivent se situer pour assurer la stabilité. On remarque sur les deux premières figures l'influence de la place du zéro par rapport au point  $(1,0)$  pour la rapidité.

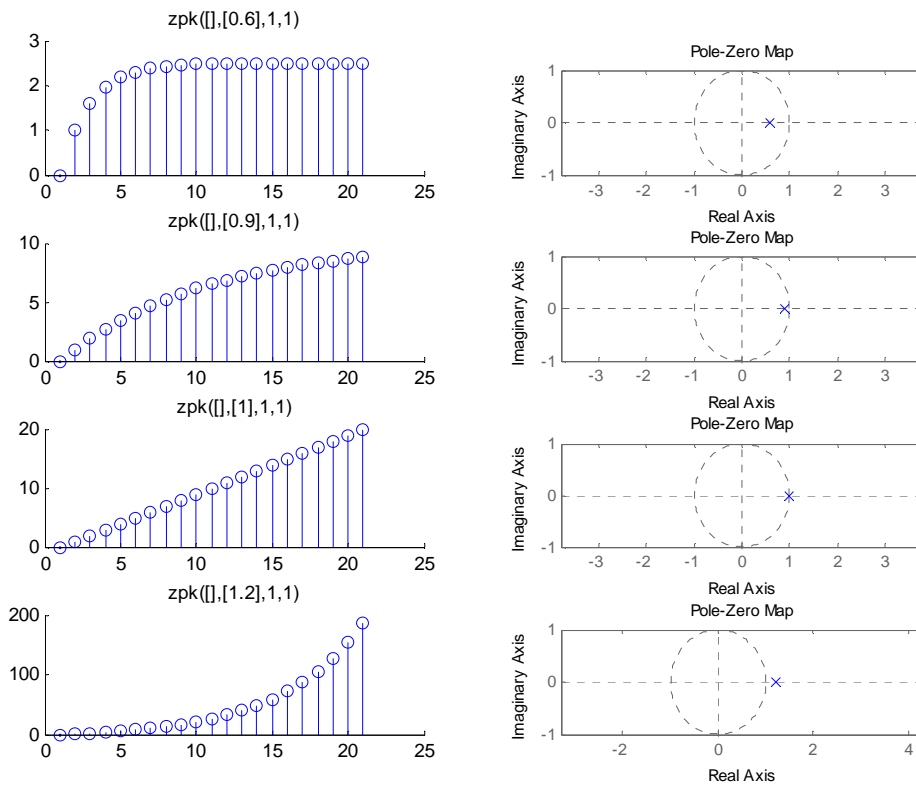


Figure 4 réponses indicielles de systèmes du premier ordre

### 3.2.9 Systèmes du second ordre

On donne dans la Figure 6 les réponses indicielles de systèmes du second ordre ainsi que la localisation des pôles dans le plan complexe. La rapidité et l'amortissement dépend de la position des pôles dans le cercle unité. La Figure 5 représente les courbes iso-amortissement et iso-pulsation des systèmes du second ordre.

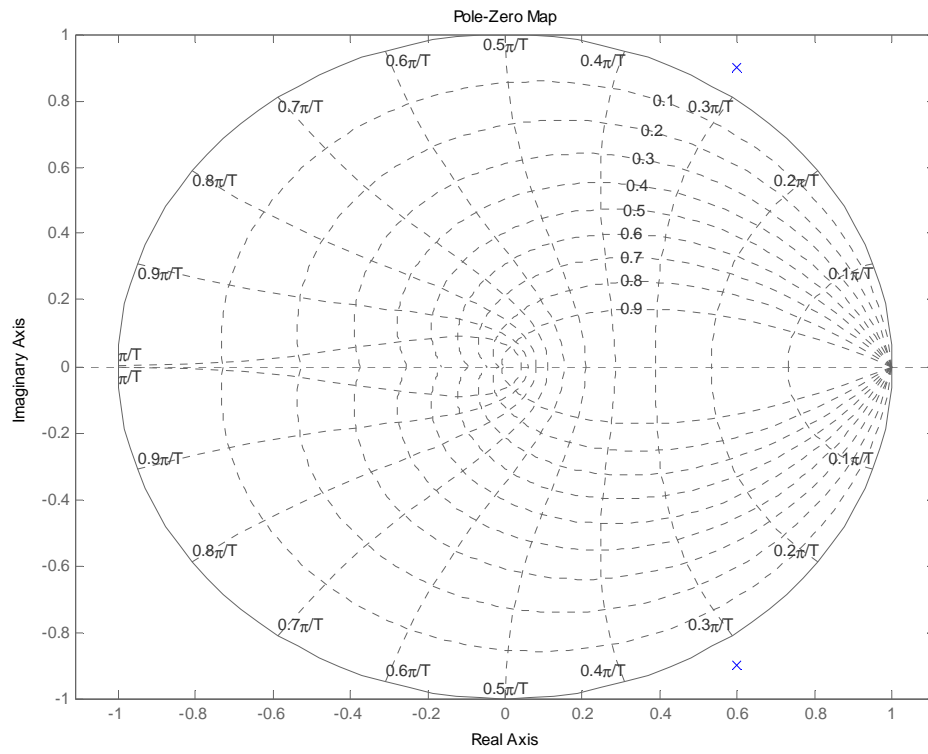


Figure 5 : courbes iso-amortissement et iso-pulsation des systèmes discrets du second ordre

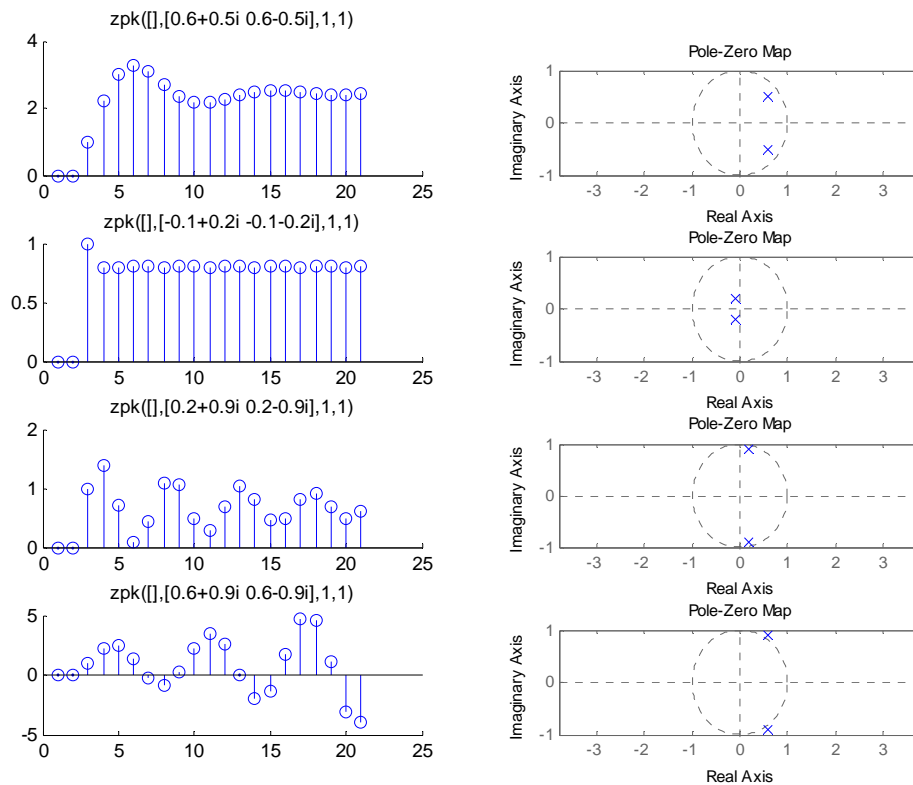


Figure 6 : réponses de systèmes du second ordre

### 3.2.10 Calcul itératif de la réponse d'un système

Si l'on dispose de la fonction de transfert d'un système et que l'on connaît les valeurs successives  $e(k)$  du signal d'entrée il est facile de calculer itérativement la sortie  $y(k)$  du système par l'équation de récurrence (2). Il faut initialiser par les  $n$  valeurs initiales de  $s(k)$ , prises généralement égales à 0.

### 3.2.11 Représentation par équation d'état

Un système discret peut aussi être représenté par un ensemble d'équations d'état comme dans le domaine continu (52). Cette représentation est naturellement adaptée aux systèmes multi-entrées multi-sorties. On définit alors un état interne  $x(k)$  ainsi qu'une fonction d'évolution  $f$  et une équation de mesure  $h$ .

$$\begin{cases} x(k+1) = f(x(k), u(k)) \\ s(k) = h(x(k), u(k)) \end{cases} \quad (18)$$

### 3.3 Réponse fréquentielle des systèmes échantillonnés

#### 3.3.1 Diagramme de Bode d'un système échantillonné

Lorsqu'on étudie la réponse fréquentielle des systèmes continus on considère des entrées particulières sinusoïdales :

$$\begin{cases} e(t) = \cos(2 \cdot \pi \cdot f \cdot t) + j \cdot \sin(2 \cdot \pi \cdot f \cdot t) = e^{j \cdot 2 \cdot \pi \cdot f \cdot t} \\ f \in [0 \quad \infty] \end{cases} \quad (19)$$

De même, pour des systèmes échantillonnés on considèrera des entrées sinusoïdales échantillonnées :

$$\begin{cases} e(k) = \cos(2 \cdot \pi \cdot f \cdot k) + j \cdot \sin(2 \cdot \pi \cdot f \cdot k) = e^{j \cdot 2 \cdot \pi \cdot f \cdot k} \\ f \in [0 \quad 1] \end{cases} \quad (20)$$

La sortie est aussi un signal échantillonné sinusoïdal de gain  $A(f)$  et de phase  $\Phi(f)$  variables et dépendant de  $f$ ; la fréquence réduite.

$$\begin{cases} s(k) = A(f) \cdot e^{j \cdot (2 \cdot \pi \cdot f \cdot k + \Phi(f))} \\ f \in [0 \quad 1] \end{cases} \quad (21)$$

Le nombre complexe, noté abusivement  $F(f)$ , d'amplitude  $A(f)$  et de phase  $\Phi(f)$  est obtenu à partir de  $F(z)$  en remplaçant  $z$  par  $e^{j \cdot 2 \cdot \pi \cdot f}$  :

$$\begin{cases} F(f) = F(z = e^{j \cdot 2 \cdot \pi \cdot f}) \\ f \in [0 \quad 1] \end{cases} \quad (22)$$

Le diagramme représentant le gain et la phase en fonction de la fréquence réduite s'appelle aussi un diagramme de Bode. L'échelle des fréquences est généralement linéaire de 0 à 0.5 (puisque le diagramme est nécessairement symétrique). La Figure 7 représente par exemple le diagramme de Bode d'un système du second ordre.

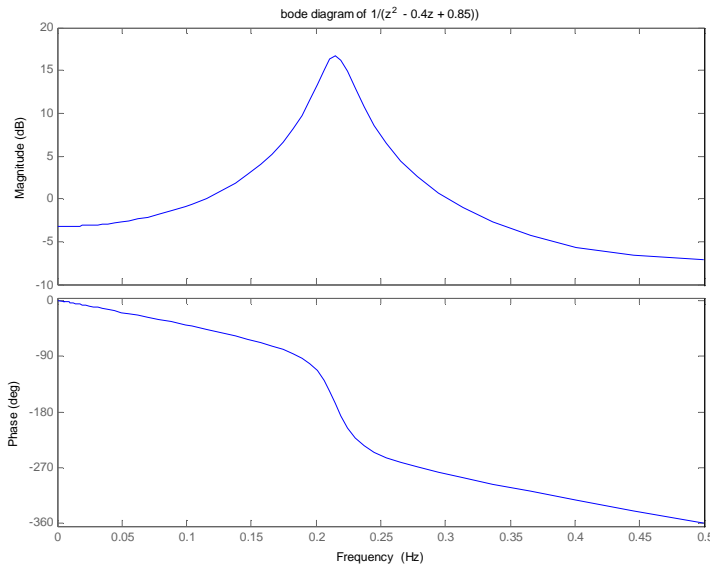


Figure 7 : exemple du diagramme de Bode d'un système discret du second ordre

### 3.3.2 Où l'on voit apparaître le théorème de Shannon

Le changement de variable  $z = e^{j2\pi f}$  n'est pas bijectif puisque  $e^{j2\pi f} = e^{j2\pi(f+k)}$  avec  $k$  entier. Si l'on fait le lien entre domaine continu et domaine fréquentiel on peut remarquer que le changement de variable  $z = e^{j2\pi f}$  fait correspondre une zone du plan complexe continu à une zone du plan complexe discret. La transformation est bijective si l'on se contente de la zone du plan complexe continu contenu entre les droites d'ordonnée 0 et 1.

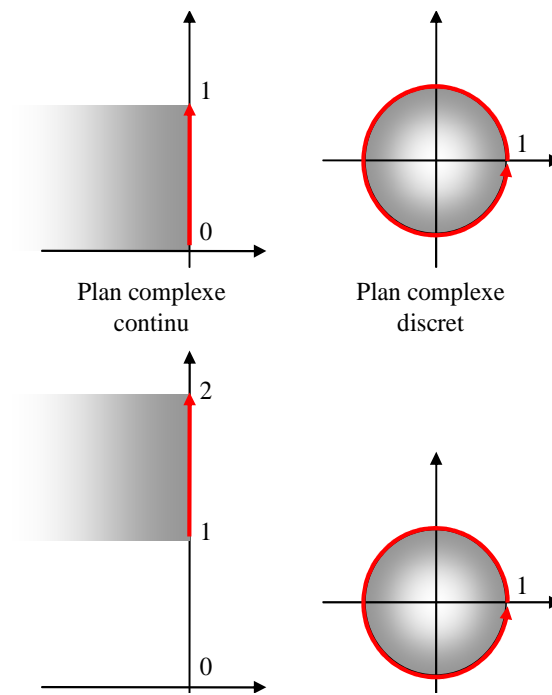


Figure 8 : correspondance entre plan complexe continu et plan complexe discret.





## 4. SYSTEMES CONTINUS ECHANTILLONNES

### 4.1 Structure des systèmes commandés par ordinateur

Les systèmes commandés par ordinateurs sont presque toujours des systèmes physiques continus par nature. Le ordinateur traite par contre les entrées et les sorties de manière discrète avec une période d'échantillonnage  $T_e$ . Les éléments importants, du point de vue de l'automaticien sont donc :

- le convertisseur analogique / numérique CAN ou du moins le système qui fournit séquentiellement les mesures au ordinateur à la période  $T_e$ .
- le ordinateur lui-même. Il doit effectuer l'acquisition des mesures, le traitement des informations, l'envoi des consignes dans un temps inférieur à  $T_e$ .
- le convertisseur numérique / analogique CNA.
- le système physique.

On suppose, pour faire simple et pour pouvoir utiliser la panoplie des outils mathématiques, que les systèmes sont échantillonnés à la même période  $T_e$  et au même moment.

La Figure 9 montre le système physique commandé par le ordinateur, dans sa version simpliste.

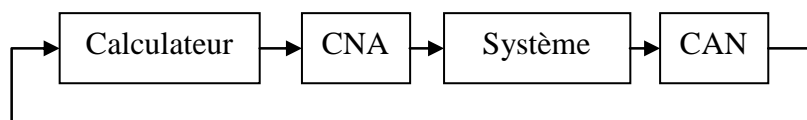


Figure 9 : système commandé par ordinateur

Au XXIème siècle les choses sont quand même parfois un petit peu plus compliquées :

- le ordinateur fournit bien souvent un signal de commande numérique (transporté par bus informatique USB, RS232, CAN, Ethernet, etc...) directement interprété par l'actionneur. On parle "d'actionneur intelligent".
- de même le "capteur intelligent" fournit aussi bien souvent la mesure sous forme d'un signal numérique.
- Les signaux sont transportés par des bus informatiques qui n'ont pas que ça à faire : il peut y avoir du retard.
- de même les capteurs et actionneurs sont rarement tous seuls, ce qui se traduit par des périodes d'échantillonnage différentes, et parfois non synchrones...

- enfin le ordinateur peut aussi avoir d'autres choses à faire...

Ceci étant dit nous allons pour le moment nous restreindre au cas représenté sur la Figure 9 avec des instants d'échantillonnage synchrones et régulièrement espacés d'un temps  $T_e$ .

## 4.2 Echantillonnage des systèmes continus

On dispose généralement de la fonction de transfert continue (notée en utilisant la variable complexe  $s$  ou  $p$ )  $F(s)$  du système continu. Le problème est d'obtenir la fonction de transfert discrète du système échantillonné de manière à pouvoir synthétiser un correcteur. Le système continu est associé au convertisseur numérique – analogique qu'il faut prendre en compte.

Les convertisseurs numérique analogiques peuvent généralement se représenter par des "échantillonneurs – bloqueurs d'ordre 0" ou BOZ. Cela signifie que la commande est maintenue à une valeur constante pendant toute la durée du temps d'échantillonnage.

Si l'on applique une commande discrète impulsionnelle à un BOZ il va maintenir la commande 1 pendant la durée  $T_e$  puis revenir à zéro (Figure 10).

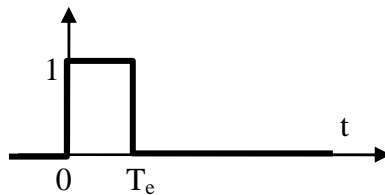


Figure 10 : réponse impulsionnelle d'un bloqueur d'ordre 0

La réponse impulsionnelle  $BOZ(t)$  est donc composée d'un échelon unitaire suivi d'un échelon unitaire de signe opposé et décalé d'un temps  $T_e$ . La transformée de Laplace de la réponse impulsionnelle du bloqueur est donnée par la formule (23) :

$$BOZ(s) = \frac{1}{s} - \frac{e^{-jT_e}}{s} \quad (23)$$

Du point de vue du ordinateur il faut donc prendre en compte le système continu de fonction de transfert  $F_{BOZ}(s)$  que l'on devra échantillonner :

$$F_{BOZ}(s) = \frac{1 - e^{-jT_e}}{s} \cdot F(s) \quad (24)$$

$$F_{\text{BOZ}}(z) = Z^{-1} \left( \frac{1 - e^{-jT_e}}{s} \cdot F(s) \right) \quad (25)$$

En pratique l'obtention de la transformée en z d'un système échantillonné par bloqueur d'ordre 0 se fait à l'aide de tables. On donne quelques exemples fondamentaux dans le Tableau 3 ci-dessous.

Tableau 3 : transformée en z de systèmes continus échantillonnés

F(s)	F <sub>BOZ</sub> (z)
1	1
$\frac{1}{s}$	$\frac{T_e \cdot z^{-1}}{1 - z^{-1}}$
$\frac{1}{1 + T \cdot s}$	$\begin{cases} \frac{b_1 \cdot z^{-1}}{1 + a_1 \cdot z^{-1}} \\ b_1 = 1 - e^{-T_e/T} \quad a_1 = -e^{-T_e/T} \end{cases}$
$\frac{e^{-Ls}}{1 + T \cdot s}$	$\begin{cases} \frac{b_1 \cdot z^{-1} + b_2 \cdot z^{-2}}{1 + a_1 \cdot z^{-1}} \quad a_1 = -e^{-T_e/T} \\ b_1 = 1 - e^{-(L-T_e)/T} \quad b_2 = e^{-T_e/T} (e^{L/T} - 1) \end{cases}$
$L < T_e$	

Si l'on dispose de Matlab c'est bien sûr encore plus simple. On utilise la fonction c2d avec l'argument 'zoh' pour "zero holder hold" (bloqueur d'ordre zéro).

#### Matlab

```
% Définition d'un système continu :
sysc=tf(1,[1 1]);
% Obtention de la transformée en z du système échantillonné à Te
Te=0.1;
sysd=c2d(sysc,Te,'zoh');
% Notation en z^-1
sysd.variable='z'
```

### 4.3 Effets de l'échantillonnage

Le convertisseur numérique a deux effets (néfastes) importants : l'effet de retard et l'effet de quantification.

L'effet de retard se voit dans (24) : on peut dire en première approximation que l'échantillonnage se traduit par un retard pur de  $T_e/2$ . Cela a donc, si ce temps

d'échantillonnage est non négligeable par rapport aux constantes de temps du système, un effet déstabilisant.

Pour illustrer l'effet de quantification supposons par exemple que l'on utilise un convertisseur analogique/numérique 8 bits 0-10V. Le plus petit incrément que le convertisseur peut distinguer est de  $10V/2^8 = 40mV$ . Si le signal utile varie physiquement entre 5V et 5,4V on n'aura que dix incréments utilisables... Ce phénomène n'entre pas dans la panoplie des phénomènes modélisables par la transformée en z. En pratique il est donc souvent nécessaire d'étudier le système et de synthétiser la commande numérique en négligeant ce phénomène puis de vérifier a posteriori qu'il n'a pas un effet trop néfaste.

#### 4.4 Où l'on revoit le théorème de Shannon

Le théorème de Shannon est bien connu en ce qui concerne l'échantillonnage de signaux discrets : on sait que le signal ne doit pas comporter de composante de fréquence supérieure à  $f_e/2$  si l'on ne veut pas observer de repliement. On doit aussi prendre des précautions si l'on veut travailler sur la fonction de transfert discrète correspondant à un système continu échantillonné.

On va traiter un exemple concret pour mettre en évidence le problème de repliement dans le cas de la commande des systèmes continus. On s'intéresse à deux systèmes continus du second ordre ayant des pôles de même partie réelle mais de parties imaginaires séparées de  $2\pi T_e$ . Le script Matlab ci-dessous montre la création des deux systèmes :

##### Matlab

```
% Définition de deux systèmes continus :
sys1=zpk([],[-1-2i -1+2i],2^2+1);
sys2=zpk([],[-1-(2+2*pi)*i -1+(2+2*pi)*i],1+(2+2*pi)^2);
```

On voit sur la Figure 11 que la réponse indicielle des deux systèmes est bien différente.

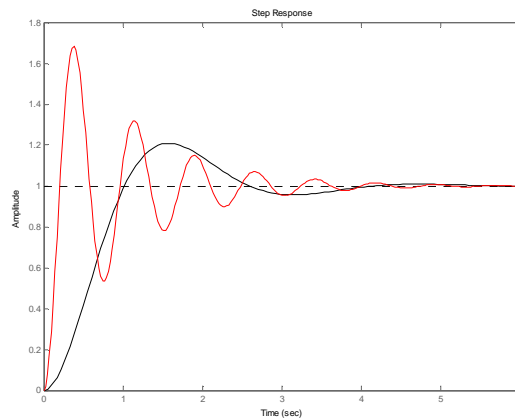


Figure 11 : réponse indicielle de deux systèmes du second ordre

Si l'on calcule la fonction de transfert échantillonnée les deux systèmes à la période de 1 seconde on obtient deux réponses indicielles très proches (représentés par des  $\circ$  sur la Figure 12). Conclusion : la période d'échantillonnage était mal choisie

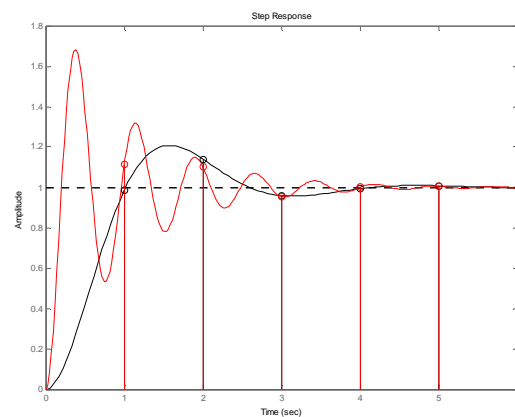


Figure 12 : réponse indicielle de deux systèmes du second ordre échantillonnés à 1s

En pratique la commande échantillonnée de systèmes continus doit être associée à certaines précautions :

- le système ne doit pas comporter de dynamiques propres situées dans les fréquences supérieures à  $f_c/2$  (exemple illustré dans ce chapitre).
- les signaux échantillonnés doivent être "nettoyés" des fréquences supérieures à  $f_c/2$  avec un filtre anti-repliement. (A moins que le signal soit suffisamment propre et ne comporte pas de composantes significatives au-delà de ces fréquences).

### Choix de la période d'échantillonnage

Le choix de la période d'échantillonnage est très important et dépend de la dynamique des signaux. Attention, les systèmes commandés en boucle fermée ont souvent une dynamique plus rapide que le système en boucle ouverte : il faut donc prendre en compte la dynamique en boucle fermée pour le choix de la fréquence d'échantillonnage.

En pratique on préconise de prendre une fréquence 5 à 25 fois supérieure à la bande passante souhaitée en boucle fermée.

On peut être tenté de prendre une fréquence d'échantillonnage très élevée si la technologie le permet (ordinateur et convertisseurs suffisamment rapide). Ceci peut amener rapidement à des problèmes numériques. Imaginons par exemple le cas d'un système ayant une dynamique très lente (quelques secondes) et un système de mesure codant les mesures sur 10 bits (cas classique). La Figure 13 illustre cet exemple. Le signal varie peu entre deux instants d'échantillonnage et les variations du signal sont faibles par rapport à la valeur de l'incrément numérique (figure du haut). Le calcul de la simple dérivée numérique se traduit donc par une succession de  $-1$   $0$  et  $+1$ .

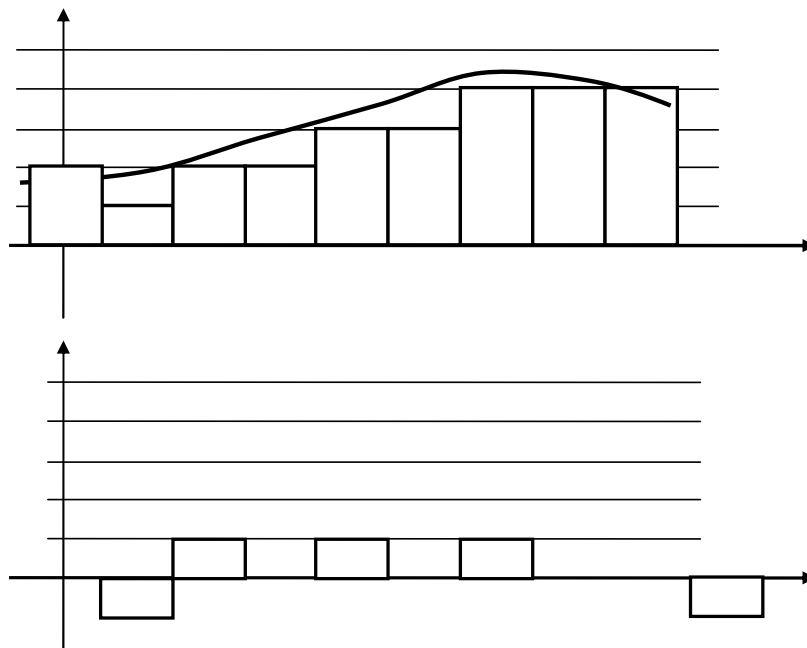


Figure 13 : période d'échantillonnage trop faible. En haut : signal échantillonné et discrétisé ; en bas : calcul de la dérivée numérique.

### Filtre anti-repliement

En toute rigueur chaque fois qu'un signal est échantillonné il faudrait faire précéder le convertisseur numérique d'un filtre analogique anti-repliement éliminant physiquement toutes les composantes du signal de fréquence supérieure à  $f_e/2$ .

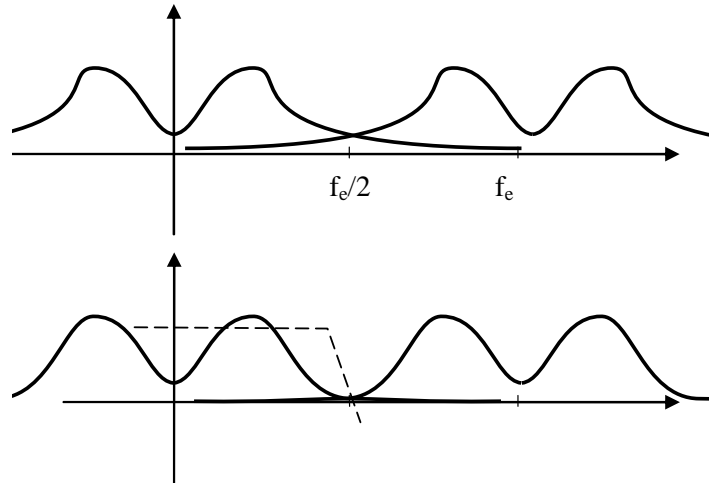


Figure 14 : Effet du filtre anti-repliement

Bien souvent la période d'échantillonnage de l'asservissement est relativement lente (quelques secondes) et il est alors impossible de réaliser un filtre de constante de temps aussi faible. On procède alors à un sur-échantillonnage : le signal est filtré, échantillonné à haute fréquence  $T'_e$  puis filtré numériquement et sous-échantillonné à la période  $T_e$  beaucoup plus grande que  $T'_e$  comme illustré sur la

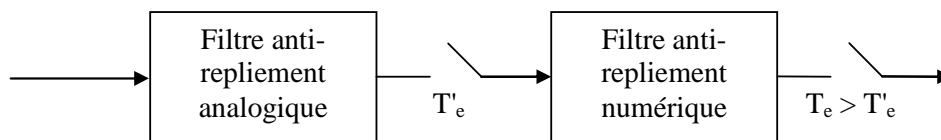


Figure 15 : Sur échantillonnage

Un simple filtre numérique de type moyenne mobile est souvent suffisant.

### 4.5 Discrétisation de fonction de transfert continues

Une manière classique pour réaliser un contrôle par ordinateur est de commencer par synthétiser un correcteur continu avec les outils de l'automatique classique puis de trouver un correcteur discret qui "approche" convenablement le correcteur continu.

#### 4.5.1 Approximation numérique d'équation différentielle

Supposons que le correcteur continu que l'on a synthétisé soit du type PD :

$$\text{PID}(s) = \frac{e(s)}{\varepsilon(s)} = k_p + k_D \cdot s \quad (26)$$

Il s'agit de calculer les commandes  $e(k)$  successives issues du correcteur en fonction des erreurs  $\varepsilon(k)$ . Une approche classique et intuitive est d'approximer la dérivée par une différence :

$$e(k) = k_p \cdot \varepsilon(k) + k_D \cdot \frac{\varepsilon(k) - \varepsilon(k-1)}{T_e} \quad (27)$$

Si on calcule la transformée en  $z$  de  $e(k)$  on obtient :

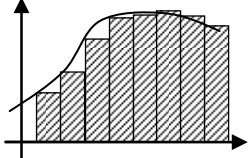
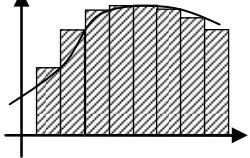
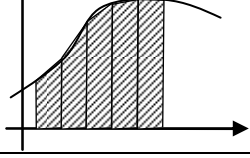
$$e(z) = k_p \cdot \varepsilon(z) + k_D \cdot \frac{\varepsilon(z) - z^{-1}\varepsilon(z)}{T_e} = \left( k_p + k_D \cdot \frac{1 - z^{-1}}{T_e} \right) \cdot \varepsilon(z) \quad (28)$$

On retrouve la fonction de transfert PD(s) dans laquelle on a effectué le changement de variable  $s = \frac{1-z^{-1}}{T_e}$

De manière plus générale, l'approximation de fonction de transfert revient à effectuer l'approximation numérique d'intégration. Selon la méthode d'intégration choisie on effectue des changements de variable différents.



Tableau 4 : méthodes d'intégration

Méthode	Changement de variable	
Euler progressif (Euler's forward method)	$s = \frac{1 - z^{-1}}{z^{-1} \cdot T_e}$	
Euler rétrograde (Euler's backward method)	$s = \frac{1 - z^{-1}}{T_e}$	
Méthode des trapèzes, ou bilinéaire, ou de Tustin	$s = \frac{2}{T_e} \cdot \frac{1 - z^{-1}}{1 + z^{-1}}$	

En pratique le changement de variable  $s = \frac{1 - z^{-1}}{z^{-1} \cdot T_e}$  n'est pas sans risque. On donne sur la Figure 16 la correspondance entre la zone stable du plan complexe en  $s$  et la zone correspondante après la transformation (Euler progressif, Euler rétrograde, bilinéaire). On voit notamment qu'il est possible, partant d'un système continu stable, d'obtenir des pôles dans le domaine discret en dehors du cercle de stabilité si la transformation appliquée est celle d'Euler progressif.

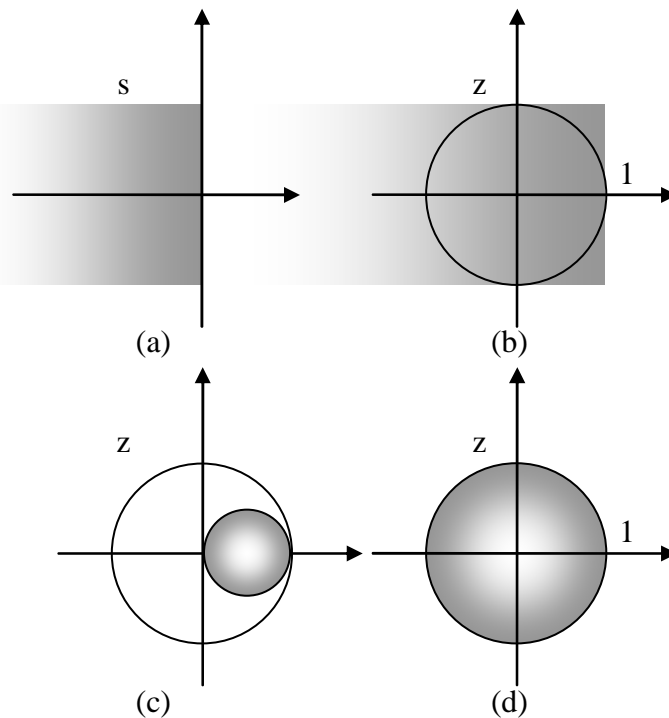


Figure 16 : correspondance des plans complexes. (a) : zone de stabilité dans le plan complexe continu en  $z$  ; (b) zone correspondante avec l'approximation d'Euler progressif ; (c) zone correspondante avec l'approximation d'Euler rétrograde ; (d) zone correspondante avec l'approximation bilinéaire.

La fonction `c2d` de Matlab permet d'obtenir l'équivalent discret d'une fonction de transfert continue avec la méthode d'intégration comme argument :

```

Matlab
sysc=tf([1 0],[1 2 5]);
Te=0.1;
% Méthode d'euler :
sysd1=c2d(sysc,Te,'zoh');
% Méthode des trapèzes :
sysd2=c2d(sysc,Te,'foh');
% Méthode bilinéaire :
sysd3=c2d(sysc,Te,'tustin');
    
```

#### 4.5.2 Superposition des réponses indicielles

Une autre méthode pour approximer une fonction de transfert discrète est de s'assurer que la réponse indicielle de sa version discrétisée correspond exactement lors des temps d'échantillonnage.

### 4.5.3 Correspondances pôles - zéros

La dernière méthode est de partir d'une décomposition de la fonction de transfert continue en pôles et zéros et d'effectuer la conversion pôles à pôles et zéros par zéros. La correspondance, selon que le pôle est réel ou complexe est donnée ci-dessous :

$$\begin{aligned} (s + a) &\rightarrow (z - e^{-a \cdot T_e}) \\ (s + a)^2 + b^2 &\rightarrow z^2 - 2 \cdot e^{-a \cdot T_e} \cdot \cos(b \cdot T_e) \cdot z + e^{-2 \cdot a \cdot T_e} \end{aligned} \quad (29)$$

### 4.5.4 Bilan

Aucune méthode n'est parfaite et on n'aura jamais en même temps :

- même réponse indicielle
- même réponse impulsionnelle
- approximation (Euler, Tustin, etc...) de l'équation différentielle
- correspondance des pôles et des zéros
- correspondance des réponses fréquentielles

Ceci étant dit le choix de la période d'échantillonnage est primordial. Si la fréquence d'échantillonnage est beaucoup plus faible que la dynamique de la fonction de transfert à approximer la différence entre chaque méthode sera minimale.



## 5. IDENTIFICATION DES SYSTEMES ECHANTILLONES

Dans le présent chapitre on donne les bases des méthodes d'identification. Pour une approche complète, voir le cours de troisième année "Identification avion" [6] et les ouvrages [1] et [7].

Même si l'on dispose d'un modèle théorique du processus que l'on souhaite commander il est presque toujours nécessaire de l'affiner en procédant à des essais sur le système réel. Bien souvent le seul modèle dont l'on dispose est ce modèle issu de l'expérimentation. Différentes méthodes s'appliquent selon l'objectif recherché et le niveau de précision exigé. On distinguera :

- les modèles paramétriques
- les modèles simples
- les modèles non paramétriques

### 5.1 Identification d'un modèle ARX par la méthode des moindres carrés récursifs

A partir de cet exemple, le plus simple qui soit, on pose les bases de toutes les méthodes d'identification.

On considère un système d'entrée  $u(t)$  et de sortie  $y(t)$  représenté par l'équation de récurrence suivante :

$$y(t) + a_1 \cdot y(t-1) + a_2 \cdot y(t-2) + \dots + a_n \cdot y(t-n) = b_0 \cdot u(t) + b_1 \cdot u(t-1) + b_2 \cdot u(t-2) + \dots + b_m \cdot u(t-m) \quad (30)$$

L'objectif est, à partir d'essais et de mesures, d'identifier les meilleures valeurs possibles pour les coefficients  $a_i$  et  $b_i$ .

L'équation (30) peut être réécrite pour mettre en évidence la *prédiction* de la sortie  $y(t)$  à partir des mesures passées :

$$y(t) = -a_1 \cdot y(t-1) - a_2 \cdot y(t-2) - \dots - a_n \cdot y(t-n) + b_0 \cdot u(t) + b_1 \cdot u(t-1) + b_2 \cdot u(t-2) + \dots + b_m \cdot u(t-m) \quad (31)$$

Soit :

$$y(t) = \varphi^T(t) \cdot \theta(t)$$

avec :

$$\begin{cases} \varphi^T(t) = [-y(t-1) \ \dots \ -y(t-n) \ , \ u(t-1) \ \dots \ u(t-m)] \\ \theta(t) = [a_1 \ \dots \ a_n \ , \ b_1 \ \dots \ b_m] \end{cases} \quad (32)$$

Le but est de trouver le meilleur ensemble de paramètres  $\theta(t)$  : celui qui, à partir d'un ensemble de mesures  $\varphi(t)$  permet de prédire avec la meilleure précision la sortie effectivement mesurée.

Supposons qu'à la suite d'une expérience on dispose d'un ensemble de mesures  $\{y(t), u(t)\}_{t=1:N}$  et que l'ensemble des paramètres  $\theta$  n'est pas bien connu. On construit la séquence des *sorties prédites* :

$$\hat{y}(t) = \varphi^T(t) \cdot \theta(t) \quad (33)$$

Le paramètre  $\theta$  est celui qui minimise une norme quadratique :

$$\theta = \arg \max \left( \frac{1}{N} \sum_{t=1}^N (y(t) - \hat{y}(t))^2 \right) = \arg \max \left( \frac{1}{N} \sum_{t=1}^N (y(t) - \varphi^T(t) \cdot \theta(t))^2 \right) \quad (34)$$

Compte tenu du fait que l'équation est linéaire au départ et que le terme à minimiser est du type quadratique la solution est simplement obtenue par une pseudo-inverse :

$$\theta = \left[ \sum_{t=1}^N (\varphi(t) \cdot \varphi^T(t)) \right]^{-1} \sum_{t=1}^N (\varphi(t) \cdot y(t)) \quad (35)$$

On obtient donc la solution classique des moindres carrés. La formule (35) est le plus souvent utilisée sous sa forme récursive :

$$\theta(t) = \theta(t-1) + F(t) \cdot \varphi(t) \cdot (y(t) - \theta^T(t-1) \cdot \varphi(t))$$

avec

$$F(t)^{-1} = F(t-1)^{-1} + \varphi(t) \cdot \varphi(t)^T \quad (36)$$

ou

$$F(t) = F(t-1) - \frac{F(t-1) \cdot \varphi(t) \cdot \varphi(t)^T \cdot F(t-1)}{1 + \varphi(t)^T \cdot F(t-1) \cdot \varphi(t)}$$

(On laisse au lecteur le plaisir d'effectuer lui-même les démonstrations, qu'il peut du reste trouver dans [1] et [7]).

## 5.2 L'identification paramétrique en général et en pratique

L'exemple précédent a permis de mettre en évidence tous les aspects de la méthodologie classique de l'identification :

- il a fallu choisir un ensemble de mesures  $\{y(t), u(t)\}_{t=1:N}$ . Le choix du signal d'excitation  $u(t)$  est en pratique primordial,
- il a fallu choisir une structure pour le modèle à identifier (ici les ordres  $n$  et  $m$ ),
- il a fallu choisir un critère (ici un critère quadratique sur l'erreur entre la sortie et la sortie prédite),
- il a fallu choisir un algorithme et le paramétrer (ici l'algorithme des moindres carrés récursifs qu'il a fallu initialiser)

Enfin un point non traité dans le chapitre précédent concerne la validation :

- il aurait fallu déterminer une méthode pour comparer différents modèles et choisir le meilleur

### 5.2.1 Choix du signal d'excitation, traitement des mesures

De manière générale le système que l'on doit identifier doit être excité dans tout le domaine fréquentiel qui nous intéresse. Le signal classiquement utilisé est une séquence pseudo binaire aléatoire (Pseudo Random Binary Signal) tel celui représenté sur la Figure 17.

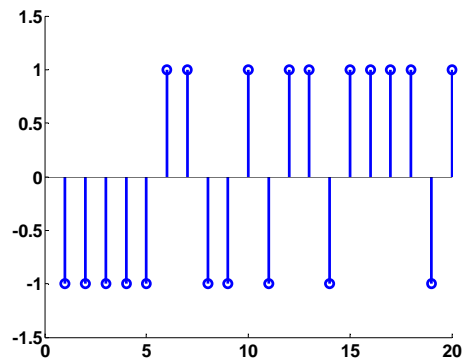


Figure 17 : exemple d'une séquence PRBS

Le signal d'excitation doit se superposer au signal de commande correspondant au point de fonctionnement étudié. Son amplitude doit être faible par rapport à l'amplitude des commandes du système (quelques %).

Une fois le système excité et les mesures effectuées il faut supprimer les valeurs moyennes des mesures.

### 5.2.2 Choix d'une structure de modèle

Le modèle étudié dans le chapitre précédent est de type ARX. On peut le noter sous forme polynomiale :

$$A \cdot y = B \cdot u \quad (37)$$

Ce modèle correspond en fait à un parti-pris sur la manière dont on traite les perturbations. En réalité on prend en compte un signal d'erreurs possibles  $e$  (de moyenne nulle) tels que :

$$A \cdot y = B \cdot u + e \quad (38)$$

Si l'on fait cette hypothèse de modélisation alors la méthode optimale est celle des moindres carrés. Malheureusement cette manière de prendre en compte les perturbations est bien restrictive. D'autres modélisations sont possibles, adaptées à d'autres algorithmes, par exemple le modèle OE :

$$y = \frac{B}{F} \cdot u + e \quad (39)$$

Il est aussi possible de "structurer" le modèle de bruit comme par exemple dans les modèles ARMAX ou BJ :

$$\begin{aligned} A \cdot y &= B \cdot u + C \cdot e \\ y &= \frac{B}{F} \cdot u + \frac{C}{D} \cdot e \end{aligned} \quad (40)$$

La structure du modèle étant choisie il faut choisir l'ordre de ses différents polynômes.

### 5.2.3 Choix d'un critère à minimiser

Dans la plupart des cas le critère à minimiser est l'erreur quadratique entre la sortie et la sortie prédite à un coup (one step ahead prediction). Ceci n'est bien évidemment pas restrictif et peut être généralisé à la prédiction à  $N$  coups.

### 5.2.4 Choix d'un algorithme

Même si chaque structure de modèle est généralement associée à un algorithme "optimal" celui-ci doit généralement se paramétrer.



Prenons par exemple l'algorithme des moindres carrés récursifs tel que présenté dans (36). Cette équation peut être modifiée en intégrant un "facteur d'oubli"  $\lambda$  :

$$\begin{aligned} \theta(t) &= \theta(t-1) + F(t) \cdot \varphi(t) \cdot (y(t) - \theta^T(t-1) \cdot \varphi(t)) \\ \text{avec} & \\ F(t)^{-1} &= \lambda \cdot F(t-1)^{-1} + \varphi(t) \cdot \varphi(t)^T \end{aligned} \quad (41)$$

Ce paramètre (typiquement entre 0.95 et 0.99) permet, comme son nom l'indique, "d'oublier" les anciennes mesures et de se concentrer sur les mesures les plus récentes.

### 5.2.5 Validation

Une fois l'algorithme d'identification appliqué et le modèle obtenu (ce qui prend généralement quelques secondes avec un ordinateur moderne) il reste à expertiser celui-ci. Selon la "qualité" du modèle obtenu on remet en cause l'une ou l'autre des étapes (signal d'excitation, structure du modèle, algorithme) et on réitère jusqu'à obtenir un modèle satisfaisant.

En pratique on essaie un certain nombre de modèles avant de choisir le "meilleur".

Le premier critère de qualité est la capacité du modèle à prédire effectivement la sortie (sortie prédite à un coup, sortie prédite à N coups). Il faut pour cela utiliser un jeu de mesures non utilisé pour l'identification. (En pratique le jeu de mesures initiales est coupé en deux : une partie pour l'identification et une partie pour la validation).

Le second critère de qualité est l'analyse du résidu : la composition spectrale de l'erreur de prédiction.

### 5.2.6 Utilisation de logiciels

En pratique il est commode d'utiliser un célèbre outil de calcul scientifique et sa toolboxe d'identification.

Comme les calculs proprement dit ne prennent que quelques secondes il est facile de réaliser un grand nombre d'identifications et de stocker les différents modèles pour les comparer ensuite. On est rapidement noyé sous le grand nombre de modèles possibles...

## 5.3 Identification "simple"

Il est souvent suffisant d'obtenir un modèle "simple" de type premier ordre et un simple essai indiciel convient souvent. Le système est excité avec un échelon unité et on cherche le meilleur modèle du premier ordre correspondant.

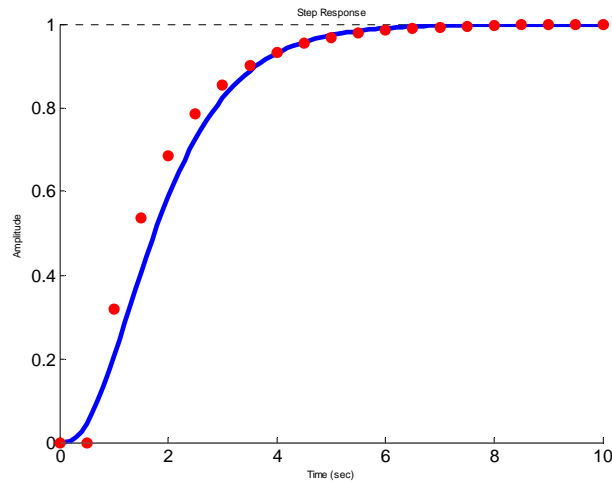


Figure 18 : identification sur un essai indiciel

Dans l'exemple de la Figure 18 on superpose le résultat d'une mesure (en bleu) avec la simulation d'un modèle discret avec une période de 0.5s. Le modèle cherché est du premier ordre avec un retard pur  $d$  :

$$y(z) = \frac{(1 + c_1) \cdot z^{-d}}{1 + c_1 \cdot z^{-1}} \cdot u(z) \quad (42)$$

Il s'agit de trouver l'entier  $d$  et le nombre réel  $c_1$  tels que la réponse indiciale du modèle corresponde pour le mieux aux mesures issues de l'essai. Ceci peut se formaliser par exemple de la manière suivante :

$$(c_1 \quad d_1) = \arg \min \sum_{t=1}^N (y_{\text{modèle}}(t) - y_{\text{mesure}}(t))^2 \quad (43)$$

Cette optimisation peut se faire à la main ou grâce à une procédure automatisée (avec le célèbre outil de calcul scientifique).

Dans le cas où le système comprend naturellement une intégration (par exemple la commande d'un moteur à courant continu par une tension et la mesure de l'angle de rotation) il est utile de prendre en compte explicitement cette intégration dans le modèle :

$$y(z) = \frac{1}{1 - z^{-1}} \cdot \frac{(1 + c_1) \cdot z^{-d}}{1 + c_1 \cdot z^{-1}} \cdot u(z) \quad (44)$$

La procédure est la même mais au lieu de prendre en compte la sortie  $y(t)$  on prend en compte la dérivée  $y'(t)$  :

$$y'(t) = y(t) - y(t-1) \quad (45)$$

## 5.4 Identification non paramétrique

Pour terminer ce panorama rapide sur les méthodes d'identification il faut parler des méthodes non paramétriques : celles qui permettent de tracer la réponse impulsionnelle ou la réponse fréquentielle du système.

### 5.4.1 Estimation de la réponse impulsionnelle

La réponse impulsionnelle d'un système peut être obtenue simplement en appliquant une entrée de type "impulsion" au système et d'en mesurer les conséquences.

La seconde méthode consiste à utiliser la méthode des moindres carrés (ou une méthode dérivée) comme vue au chapitre 5.1. La réponse impulsionnelle d'un système peut en effet être vue comme la fonction de transfert particulière où le dénominateur est de degré nul et le numérateur comporte un nombre infini (en pratique : grand) d'éléments ( $n = 0$  et  $m$  grand) :

$$y(t) = b_0 \cdot u(t) + b_1 \cdot u(t-1) + b_2 \cdot u(t-2) + \dots + b_m \cdot u(t-m) \quad (46)$$

La dernière méthode, qui sera développée ici est l'analyse d'intercorrélation.

Par définition l'intercorrélation  $R_{yu}$  entre deux signaux est :

$$R_{yu}(\tau) = E[(y(t) - E(y(t))) \cdot (u(t-\tau) - E(u(t-\tau)))] \quad (47)$$

Dans cette formule l'espérance mathématique peut être approchée grâce à la loi des grands nombres :

$$R_{yu}(\tau) \approx R_{yu}^N(\tau) = \lim_{N \rightarrow \infty} \frac{1}{N} \sum_{t=1}^N [(y(t) - m_y) \cdot (u(t-\tau) - m_u)] \quad (48)$$

On s'intéresse maintenant à un système tel que décrit par l'équation impulsionnelle (46) et on suppose que le signal d'entrée est un bruit blanc. Le signal d'entrée vérifie donc :

$$\begin{cases} \tau \neq 0 & R_{uu}(\tau) = 0 \\ \tau = 0 & R_{uu}(\tau) = R \end{cases} \quad (49)$$

Il vient alors naturellement :

$$b_k \approx \frac{1}{R} R_{yu}(k) \quad (50)$$

### 5.4.2 Estimation de la réponse fréquentielle

On utilise ici le spectre d'intercorrélation. Par définition le spectre d'intercorrélation entre deux signaux est :

$$\Phi_{yu}(\omega) = \sum_{\tau=-\infty}^{\infty} R_{yu}(\tau) \cdot e^{-i\omega\tau} \quad (51)$$

On suppose que le signal d'entrée  $u(t)$  et le signal de sortie  $y(t)$  sont reliés par une fonction de transfert linéaire du type :

$$y(t) = F \cdot u(t) + v(t) \quad (52)$$

où  $v(t)$  est une perturbation indépendante de  $u(t)$ . On a alors les relations :

$$\Phi_{yu}(\omega) = F(j\omega) \cdot \Phi_{uu}(\omega) \quad (53)$$

Il reste à estimer les spectres d'intercorrélation pour remonter à la réponse fréquentielle  $F(j\omega)$  recherchée. Une idée naturelle est d'utiliser la relation (48) mais cela marche très mal dans presque tous les cas, même en choisissant  $N$  assez grand. L'amélioration est d'utiliser une "fenêtre" pour favoriser les valeurs faibles de  $\tau$  par rapport aux valeurs élevées. La solution classique utilisant la *fenêtre de Hamming* donne :

$$\Phi_{yu}^N(\omega) = \sum_{\tau=-\gamma}^{\gamma} R_{yu}^N(\tau) \cdot w_{\gamma}(\tau) \cdot e^{-i\omega\tau} \quad (54)$$

avec

$$\begin{cases} w_{\gamma}(\tau) = \frac{1}{2} \left( 1 + \cos\left(\frac{\pi \cdot \tau}{\gamma}\right) \right) & |\tau| < \gamma \\ w_{\gamma}(\tau) = 0 & |\tau| > \gamma \end{cases}$$

On utilise ensuite la relation (51) pour estimer la réponse fréquentielle :

$$F(j \cdot \omega) = \frac{\Phi_u^N(\omega)}{\Phi_{yu}^N(\omega)} \quad (55)$$



## 6. SYSTEMES BOUCLES

### 6.1 Fonction de transfert en boucle fermée

Un système et sa commande par ordinateur peut être représenté par un ensemble de fonctions de transfert formant un schéma-bloc plus ou moins compliqué (Figure 19).

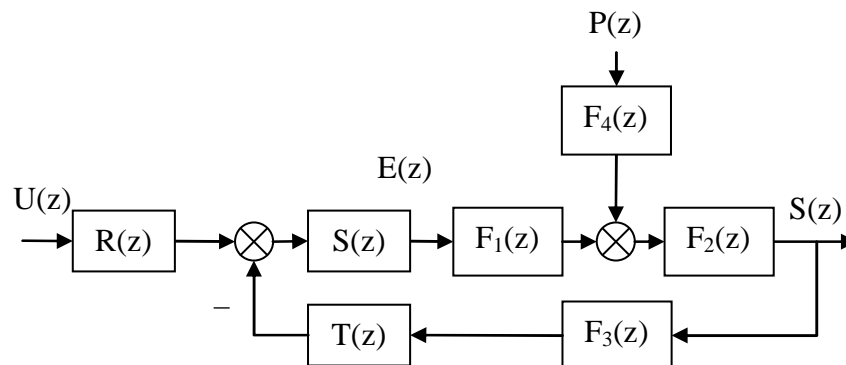


Figure 19 : système discret commandé par ordinateur. Le correcteur est représenté par les fonctions R,S,T, le système par  $F_1$  et  $F_2$ , le modèle de perturbations par  $F_4$  et le modèle des actionneurs par  $F_4$ .

Quelle que soit la structure d'un système asservi il est possible de se ramener à une forme canonique comprenant une fonction de transfert en boucle ouverte en combinant judicieusement les fonctions de transfert du procédé, du régulateur et des perturbations. La Figure 20 représente par exemple le système de la Figure 19 où  $F$  est une combinaison des fonctions de transfert du système, du régulateur, des perturbations et des capteurs.

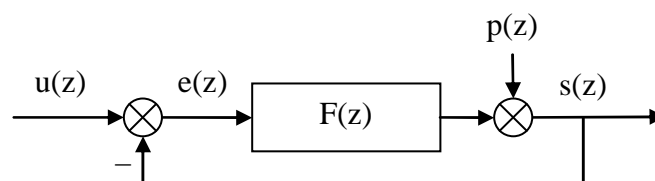


Figure 20 : système discret, notations.

Par défaut on considère une contre-réaction négative et une perturbation additive sur la sortie. La fonction de transfert  $F$  est appelée *fonction de transfert en boucle ouverte*. Comme pour les systèmes continus la *fonction de transfert en boucle fermée* est obtenue simplement à partir de  $F$  (58).

$$\begin{aligned}
 F(z) &= \frac{s(z)}{e(z)} = \frac{B(z^{-1})}{A(z^{-1})} \\
 G(z) &= \frac{s(z)}{u(z)} = \frac{B(z^{-1})}{A(z^{-1}) + B(z^{-1})}
 \end{aligned}
 \tag{56}$$

## 6.2 Erreur statique

L'erreur statique du système en boucle fermée, comme pour le système en boucle ouverte, est simplement le rapport des coefficients.

$$G_0 = \frac{\sum_{i=0}^{i=m} b_i}{1 + \sum_{i=0}^{i=m} b_i + \sum_{i=1}^{i=n} a_i}
 \tag{57}$$

Si l'on souhaite que l'asservissement confère un gain statique unitaire au système en boucle fermée (c'est-à-dire que l'on force la sortie  $s(k)$  à tendre vers la valeur de consigne  $u(k)$ ) il faut que ce rapport de coefficients soit égal à un. On montre facilement que cela est obtenu si la fonction de transfert  $F(z)$  comporte un intégrateur :

$$F(z) = \frac{s(z)}{e(z)} = \frac{1}{1 - z^{-1}} \cdot \frac{B(z)}{A'(z)}
 \tag{58}$$

## 6.3 Rejet de perturbations

La fonction de transfert entre la perturbation  $p(z)$  et la sortie  $y(z)$  est appelée fonction de sensibilité perturbation-sortie, notée  $S_{yp}$ .

$$S_{yp}(z) = \frac{s(z)}{p(z)} = \frac{A(z^{-1})}{A(z^{-1}) + B(z^{-1})}
 \tag{59}$$

Si l'on souhaite qu'une perturbation de type échelon soit parfaitement rejetée il faut que  $A(1) = 0$ , on retrouve donc la condition (58) à savoir que le système doit comporter une intégration dans la boucle directe. On peut être en outre être amené à spécifier que les perturbations ne soient pas amplifiées dans une certaine bande de fréquence. On rajoute donc une contrainte du type  $|S_{yp}(e^{j2\pi f})| < S_{\max}$  pour  $f$  appartenant à un certain domaine de  $[0 \ 0.5]$ .



## 6.4 Stabilité des systèmes à simple gain de rétroaction

Les critères de Routh et de Jury (donnés dans le chapitre 3.2.7 p13) ou le calcul numérique des pôles permettent de déterminer la stabilité du système en boucle fermée si l'on connaît sa fonction de transfert.

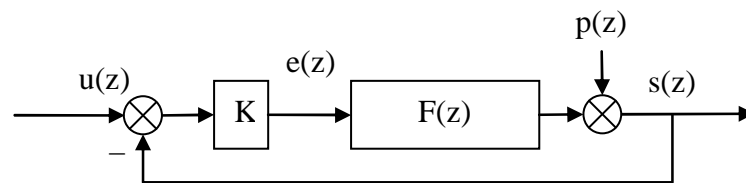


Figure 21 : système bouclé par un simple gain.

On s'intéresse ici au cas particulier où l'asservissement est réalisé par un simple gain de rétroaction  $K$  (Figure 21). On cherche la valeur de  $K$  qui amène le système en limite de stabilité. L'outil adapté ici est le lieu d'Evans ou lieu des racines (root locus) qui permet de visualiser l'évolution des pôles du système bouclé lorsque le gain  $K$  varie.

Prenons l'exemple d'un système comportant trois pôles et deux zéros. L'exemple est donné sous forme de script Matlab :

### Matlab

```
% Définition d'un système discret :
sys=zpk([-2.037 -0.1406],[0.6065 0.3679 0.3679],1,1)
%Tracé du lieu des racines
rlocus(sys);
```

Le script précédent produit le tracé de la Figure 22. On pourrait y lire que la limite de stabilité est obtenue pour  $K=0.2$  environ (détail de la Figure 23).

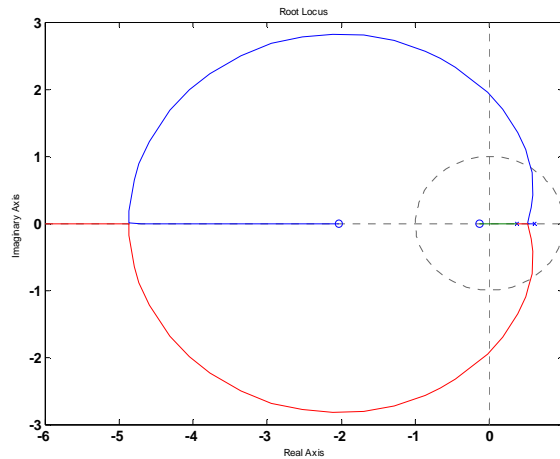


Figure 22 : lieu des racines

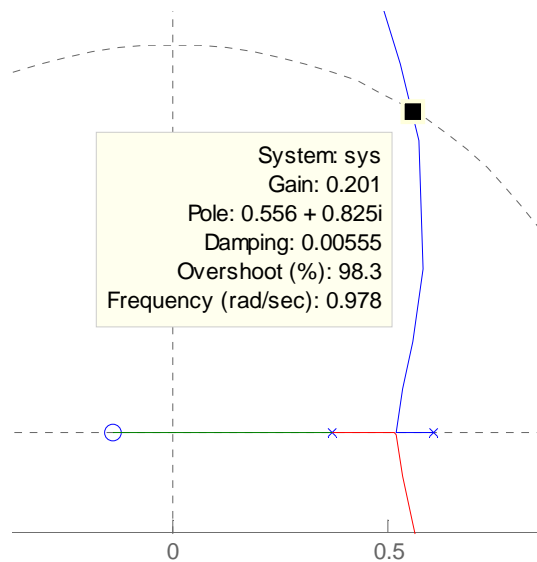


Figure 23 : lieu des racines : détail de la limite de stabilité

On voit sur la Figure 22 que le lieu des racines d'un système échantillonné ressemble à celui d'un système continu :

- on commence par tracer les pôles ( $\times$ ) et les zéros ( $o$ ) du système en boucle ouverte.
- les pôles du système en boucle fermé décrivent un lieu qui part des pôles (pour  $K = 0$ ) et arrive aux zéros (pour  $K = \infty$ )
- lorsque il y a  $n-m$  plus de pôles que de zéros le système comporte  $n-m$  branches asymptotiques.

On ne décrira pas ici les diverses propriétés du lieu des racines qui permettent de le tracer à la main puisque il est facile à obtenir avec un ordinateur.

Il peut aussi être utilisé pour régler "à la main" le gain permettant d'obtenir un ou plusieurs modes dominants ayant certaines propriétés de fréquence et d'amortissement. On voit sur la Figure 24 que le gain 0.017 permet d'obtenir un pôle double d'amortissement 0.706 et de fréquence 0.65 rad/s et un pôle simple de fréquence 1.63 rad/s.

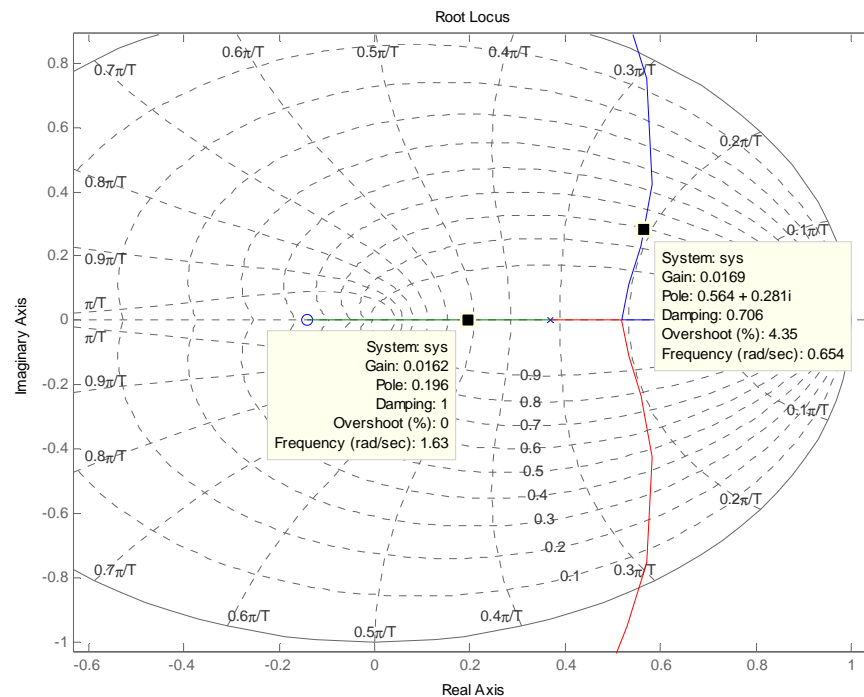


Figure 24 : utilisation du lieu des racines pour régler le gain de la boucle fermée

L'outil rltool de Matlab permet de nombreuses manipulations graphiques de ce type.

## 6.5 Analyse des systèmes bouclés dans le domaine fréquentiel : stabilité

Comme dans le cas continu il est possible de prévoir des propriétés, en particulier la stabilité, d'un système en boucle fermée à partir de l'examen de la fonction de transfert en boucle ouverte.

On ne redémontrera pas ici le théorème de Nyquist mais on se contente de rappeler l'énoncé.

On rappelle que le *lieu de Nyquist* est le lieu de  $F(j2\pi f)$  dans le plan complexe pour  $f$  variant de 0 à  $+\infty$ . A noter que le lieu pour la zone  $[0, 0.5]$  est symétrique de celui pour la zone  $[0.5, 1]$ . Pour cette raison on ne représente généralement que le lieu pour  $f$  appartenant à  $[0, 0.5]$ .

### 6.5.1 Théorème de Nyquist dans le cas où le système est stable en boucle ouverte (critère du revers)

Le système est stable en boucle fermée si le lieu de Nyquist de la fonction de transfert en boucle ouverte parcouru dans le sens des fréquences croissantes de 0 à 0.5 laisse le point critique (-1,0) à gauche.

La Figure 25 représente par exemple le lieu de Nyquist d'une fonction de transfert en boucle ouverte qui sera stable en boucle fermée.

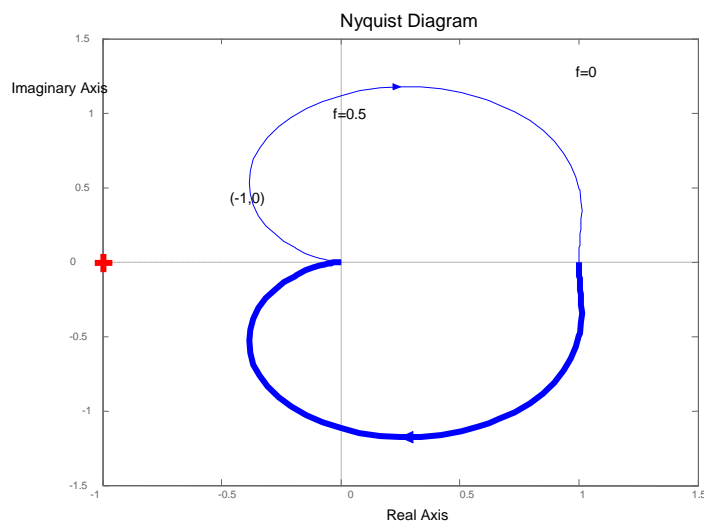


Figure 25 : lieu de Nyquist d'un système stable

### 6.5.2 Théorème de Nyquist dans le cas où le système est instable en boucle ouverte

Si le système en boucle ouverte comporte  $N$  pôles instables il sera stable en boucle fermée si le lieu de Nyquist de la fonction de transfert en boucle ouverte parcouru dans le sens des fréquences croissantes de  $-0.5$  à  $0.5$  entoure le point critique  $(-1,0)$   $N$  fois dans le sens des aiguilles d'une montre..

### 6.5.3 Interprétations du lieu de Nyquist

Comme dans le cas continu, la distance à laquelle on passe du point critique renseigne sur la robustesse du système bouclé. La distance sera exprimée de différente manière (Figure 26) :

- marge de module  $\Delta M$  : plus petite distance entre le lieu des racines et le point critique,

- marge de gain  $\Delta G$
- marge de phase  $\Delta P$

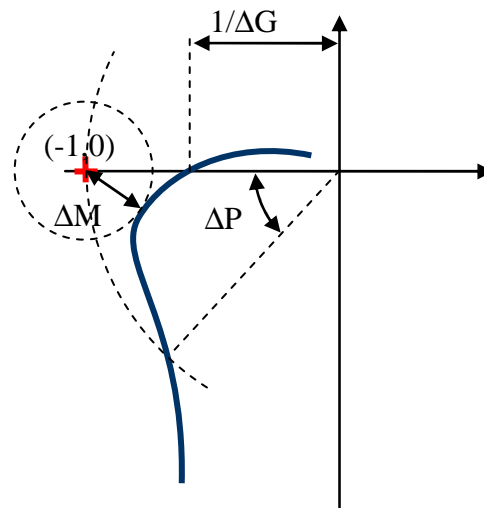


Figure 26 : Marges de stabilité

Les valeurs typiques de ces marges sont données ci-après :

- $\Delta G \geq 6\text{dB}$
- $\Delta M \geq -6\text{dB}$
- $30^\circ \leq \Delta P \leq 60^\circ$

La distance entre le point critique et le lieu des racines est la norme de  $1+F(j2\pi f)$ . On peut donc exprimer la marge de module en fonction de la fonction de sensibilité :

$$\begin{aligned} \Delta M &= \min |1 + F(j \cdot \omega)|_{\omega \in [0, \pi]} \\ \Delta M &= \min \left| 1 + \frac{B(j \cdot \omega)}{A(j \cdot \omega)} \right|_{\omega \in [0, \pi]} = \max \left| \frac{A(j \cdot \omega)}{A(j \cdot \omega) + B(j \cdot \omega)} \right|_{\omega \in [0, \pi]} \quad (60) \\ \Delta M &= \max |S_{yp}(j \cdot \omega)|_{\omega \in [0, \pi]}^{-1} \\ \Delta M_{\text{dB}} &= -\max |S_{yp}(j \cdot \omega)|_{\text{dB}} \end{aligned}$$

Si l'on trace l'évolution de  $S_{yp}$  en fonction de  $f$  on obtient donc la valeur de la marge de module.



# 7. METHODES DE REGULATION DES SYSTEMES ECHANTILLONES

## 7.1 Introduction

Les bases théoriques permettant de concevoir le correcteur pour un système échantillonné ont été posées dans les chapitres précédents. Ce chapitre fait le bilan des connaissances acquises et donne quelques solutions pour calculer des correcteurs à partir d'un cahier des charges précis.

Les systèmes que l'on cherche à commander sont généralement, par nature, continus. La réalisation d'un correcteur par un ordinateur implique que l'on génère une équation de commande discrète (généralement une équation récurrente). L'obtention de cette équation de commande peut se faire de deux manières :

- la première méthode consiste à d'abord synthétiser un correcteur continu (en utilisant les outils classiques de l'automatique et la simulation) et discrétiser ensuite ce correcteur. C'est l'objet du chapitre 7.2.2.
- la seconde méthode consiste à tenir la représentation discrète du système continu associé à son interface numérique (convertisseurs analogique-numérique et numérique-analogique) comme expliqué dans le chapitre 4.5 puis synthétiser directement le correcteur discret. Cela sera expliqué dans le chapitre 7.3 et suivants.

Ce choix peut être résumé par la figure suivante :

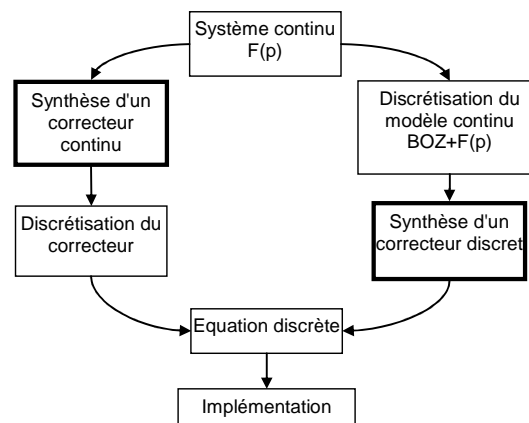


Figure 27 : Réalisation d'un correcteur discret.

Au-delà des affaires de goût, le choix de la méthode sera souvent dicté par l'application. En effet, la première méthode est d'autant meilleure que le correcteur discrétisé approche correctement le correcteur continu. Cela implique que le temps d'échantillonnage soit très faible par rapport aux constantes de temps du système

bouclé. Dans le cas où les contraintes temps réel sont fortes (capacités de calcul partagées par exemple) il sera préférable de synthétiser un correcteur discret.

## 7.2 Structure d'un système commandé par ordinateur, hypothèses

Par la suite on se ramènera toujours à un système tel que celui représenté sur la Figure 28 en cherchant un correcteur  $K(z)$  linéaire. On montre que quel que soit le correcteur linéaire on peut se ramener à la forme représenté sur la Figure 29 (forme RST).

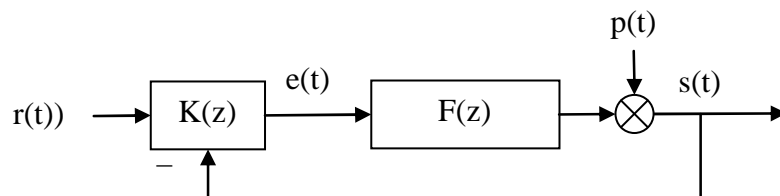


Figure 28 : système bouclé, forme générale.

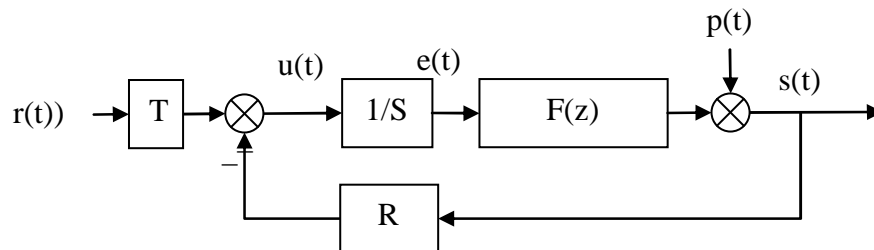


Figure 29 : système bouclé, forme canonique.

### 7.2.1 Discrétisation, choix de la période d'échantillonnage

Du point de vue du ordinateur, l'extérieur (le procédé) est représenté par la fonction de transfert  $F(z)$ . L'hypothèse importante est que la commande (sortie du ordinateur) et la mesure (entrée du ordinateur) sont échantillonnées de manière synchrone à la même période  $T_e$ .

Le choix judicieux de la fréquence d'échantillonnage est très important. En pratique on choisit cette fréquence en fonction de la bande passante du système en **boucle fermée** :

$$f_e = (6 \text{ à } 25) f^{BF} \quad (61)$$



### 7.2.2 Discrétisation d'un correcteur continu

La première méthode pour obtenir les équations d'un correcteur numérique est de faire sa synthèse avec les outils de l'automatique continue puis de discrétiser la fonction de transfert du correcteur continu obtenu (voir pour cela le chapitre 4.2 *Echantillonnage des systèmes continus*).

On va voir par la suite que la synthèse directe d'un correcteur discret offre plus de degré de liberté et permet en général de remplir un cahier des charges plus complet.

## 7.3 Régulateur PID numérique

Le régulateur le plus populaire est le régulateur PID. Nous allons voir dans un premier temps comment on génère un régulateur PID à partir des équations bien connues du régulateur PID continu et comment ces équations peuvent être généralisées pour obtenir plus de degrés de liberté pour le réglage.

### 7.3.1 Cas 1

Un régulateur PID continu a une fonction de transfert du type :

$$K_{\text{PID}}(s) = K \cdot \left( 1 + \frac{1}{T_i \cdot s} + \frac{T_d \cdot s}{1 + \frac{T_d}{N} \cdot s} \right) \quad (62)$$

La discrétisation de ce correcteur continu par la méthode d'Euler (Tableau 4 page 31) se fait en remplaçant la variable de Laplace  $s$  par le terme  $\frac{1-q^{-1}}{q^{-1} \cdot T_e}$ . On obtient :

$$K_{\text{PID}}(q) = K \cdot \left( 1 + \frac{1}{T_i \cdot \frac{1-q^{-1}}{T_e}} + \frac{T_d \cdot \frac{1-q^{-1}}{T_e}}{1 + \frac{T_d}{N} \cdot \frac{1-q^{-1}}{T_e}} \right) \quad (63)$$

En réarrangeant on obtient la forme générale RST (Figure 30):

$$K_{PID}(q) = K \cdot \frac{r_0 + r_1 \cdot q^{-1} + r_2 \cdot q^{-2}}{(1 - q^{-1}) \cdot (1 + s_1' \cdot q^{-1})}$$

$$\begin{cases} r_0 = K \cdot \left( 1 + \frac{T_e}{T_i} + \frac{N \cdot T_d}{T_d + N \cdot T_e} \right) \\ r_1 = -K \cdot \left( 1 + \frac{T_d}{T_d + N \cdot T_e} + \frac{T_e}{T_i} \cdot \frac{T_d}{T_d + N \cdot T_e} + \frac{N \cdot T_d}{T_d + N \cdot T_e} \right) \\ r_2 = K \cdot \left( \frac{T_d}{T_d + N \cdot T_e} \right) \\ s_1' = \frac{T_d}{T_d + N \cdot T_e} \end{cases} \quad (64)$$

Comme on le voit le correcteur obtenu comporte autant de degrés de liberté que le correcteur PID analogique. On reconnaît au dénominateur le terme  $(1 - q^{-1})$  en facteur qui assure l'effet d'intégration.

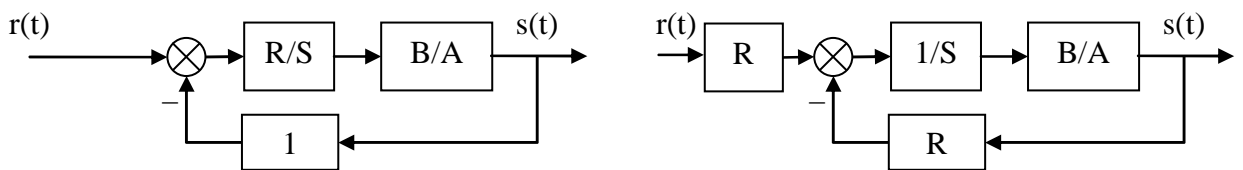


Figure 30 : PID sous forme RST

La fonction de transfert en boucle fermée obtenue est donnée par l'équation suivante :

$$H_{BF} = \frac{B \cdot R}{A \cdot S + B \cdot R} = \frac{B \cdot R}{P} \quad (65)$$

De manière générale le polynôme caractéristique  $P$  fixe les performances visées en boucle fermée. Le calcul de  $R$  et  $S$ ,  $P$  étant fixé et  $A$  et  $B$  étant connus, est réalisé en résolvant l'équation de Bézout  $P = A \cdot S + B \cdot R$ . (voir 9.1 Annexe 1 : Résolution de l'équation polynomiale de Bezout)

On voit que le numérateur  $B \cdot R$  est imposé une fois que  $R$  et  $S$  sont trouvés. Le régulateur introduit donc naturellement des zéros dans la boucle fermée.

Traisons l'exemple entièrement dans le cas où le système est modélisé comme un système du deuxième ordre avec faible retard. La fonction de transfert du procédé est donnée par :

$$H_{BO} = \frac{B}{A} = \frac{b_1 \cdot q^{-1} + b_2 \cdot q^{-2}}{1 + a_1 \cdot q^{-1} + a_2 \cdot q^{-2}} \quad (66)$$

Le correcteur PID rajoutant deux pôles, le polynôme caractéristique en boucle fermée spécifié doit posséder quatre pôles. Le polynôme caractéristique en boucle fermée doit donc être de la forme :

$$\begin{aligned} P &= 1 + p_1 \cdot q^{-1} + p_2 \cdot q^{-2} + p_2 \cdot q^{-3} + p_2 \cdot q^{-4} \\ P &= (1 + p_1' \cdot q^{-1} + p_2' \cdot q^{-2}) \cdot (1 + \alpha_1 \cdot q^{-1}) \cdot (1 + \alpha_2 \cdot q^{-1}) \end{aligned} \quad (67)$$

Et il doit être identifié au produit  $AS + BR$  avec  $R$  et  $S$  ayant les formes données par (64) :

$$\begin{aligned} &(1 + p_1' \cdot q^{-1} + p_2' \cdot q^{-2}) \cdot (1 + \alpha_1 \cdot q^{-1}) \cdot (1 + \alpha_2 \cdot q^{-1}) \\ &= (1 + a_1 \cdot q^{-1} + a_2 \cdot q^{-2}) \cdot (1 - q^{-1}) \cdot (1 + s_1' \cdot q^{-1}) \\ &+ (b_1 \cdot q^{-1} + b_2 \cdot q^{-2}) \cdot (r_0 + r_1 \cdot q^{-1} + r_2 \cdot q^{-2}) \end{aligned} \quad (68)$$

Les coefficients  $p_1'$  et  $p_2'$  définissent un modèle du second ordre aux caractéristiques (pulsation propre  $\omega_0$  et amortissement  $\sigma$  choisis et coefficients déduits conformément au Tableau 3 page 25). Les coefficients  $\alpha_1$  et  $\alpha_2$  correspondent à des pôles auxiliaires aperiodiques. Les choix typiques pour ces coefficients sont :

$$-0.05 < \alpha_1, \alpha_2 < -0.5 \quad (69)$$

On montre que la résolution de (68) est réalisée par une inversion matricielle simple :

$$M \cdot x = p \quad (70)$$

avec

$$\begin{cases} x^T = [1 & s_1' & r_0 & r_1 & r_2] \\ p^T = [1 & p_1 & p_2 & p_3 & p_4] \end{cases}$$

et

$$M = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ a_1' & 1 & b_1 & 0 & 0 \\ a_2' & a_1' & b_2 & b_1 & 0 \\ a_3' & a_2' & 0 & b_2 & b_1 \\ 0 & a_3' & 0 & 0 & b_2 \end{pmatrix} \quad (71)$$

On montre que M est inversible à la condition que A et B soient premiers entre eux, c'est-à-dire que A et B n'aient pas de pôles et de zéros en commun.

(Le problème peut aussi être résolu en utilisant la résolution de l'équation de Bézout).

### 7.3.2 Cas 2

Comme on l'a vu le correcteur PI numérique directement transposé du correcteur PID analogique décrit dans le chapitre 7.3.1 a l'inconvénient de ramener des zéros au dénominateur de la fonction de transfert en boucle fermée (équation 65). Il est possible, grâce à la flexibilité du régulateur numérique d'imposer au correcteur de ne pas ajouter de zéros en boucle fermée. La fonction de transfert désirée en boucle fermée sera donc de la forme :

$$H_{BF} = \frac{P(1)}{B(1)} \cdot \frac{B}{P} \quad (72)$$

Le facteur  $P(1)/B(1)$  garantit le gain statique unitaire du système en boucle fermée (voir page 12). Si le régulateur a la forme générale RST alors les polynômes R S et T doivent vérifier :

$$H_{BF} = \frac{B \cdot T}{A \cdot S + B \cdot R} = \frac{P(1) B}{B(1) P} \quad (73)$$

S et R sont obtenus par la résolution de la même équation que précédemment (68). Le polynôme T est obtenu par simple identification :  $T = R(1)$ .

La structure, sous forme de schéma-bloc est donnée ci-dessous :

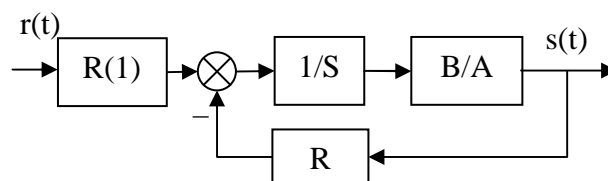


Figure 31 : PID2 sous forme RST

On montre que cette structure de PID, ramené en continu, correspond au PID canonique pour l'action intégrale. L'action proportionnelle et dérivée n'est appliquée que sur le signal y.

La structure du PID2 par rapport au PID1 a pour avantage de ne pas ramener de zéros à la boucle fermée. Cela se traduit par un dépassement moindre en transitoire lorsqu'on applique un échelon de commande. Par contre les deux systèmes réagissent de la même manière à un échelon de perturbation.

## 7.4 Placement de pôles

Les deux structures de PID décrites dans le chapitre précédent sont deux cas particulier d'une stratégie plus générale de placement de pôles.

La synthèse est plus générale que les PID présentés précédemment puisqu'il n'est pas nécessaire de faire d'hypothèses sur le degré de A et B, sur le retard, ni sur la présence ou non de zéros stables ou instables dans le procédé.

La structure du correcteur est la structure type RST telle que présentée dans la Figure 29. Pour la suite des calculs on posera pour la fonction de transfert les conventions suivantes :

$$H_{BO} = \frac{q^{-d} \cdot B(q^{-1})}{A(q^{-1})} \quad (74)$$

$$\begin{cases} A(q^{-1}) = 1 + a_1 \cdot q^{-1} + \dots + a_{n_A} \cdot q^{-n_A} \\ B(q^{-1}) = b_1 \cdot q^{-1} + \dots + b_{n_B} \cdot q^{-n_B} = q^{-1} \cdot B^*(q^{-1}) \end{cases}$$

Compte tenu de la forme RST du régulateur la fonction de transfert en boucle fermée vérifie :

$$H_{BF} = \frac{q^{-d} \cdot B(q^{-1}) \cdot T(q^{-1})}{P(q^{-1})} \quad (75)$$

$$P(q^{-1}) = A(q^{-1}) \cdot S(q^{-1}) + q^{-d} \cdot B(q^{-1}) \cdot R(q^{-1}) = 1 + p_1 \cdot q^{-1} + \dots + p_{n_p} \cdot q^{-n_p}$$

Le polynôme  $P(q^{-1})$  définit donc les pôles de la fonction de transfert en boucle fermée et traduit donc le comportement en *poursuite*.

De la même manière la fonction de transfert entre une perturbation et la sortie (fonction de sensibilité perturbation-sortie) est donnée par :

$$S_{yp} = \frac{A(q^{-1}) \cdot S(q^{-1})}{P(q^{-1})} \quad (76)$$

Ce même polynôme traduit donc aussi le comportement en *régulation*.

Les consignes en poursuite et en régulation peuvent être différentes comme cela est représenté par exemple sur la Figure 32.

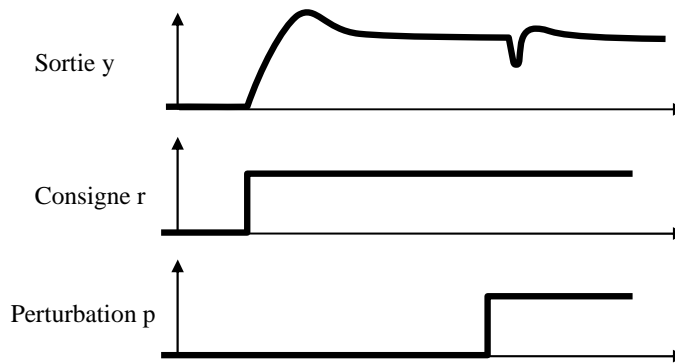


Figure 32 : Performances différentes en régulation et en poursuite

### 7.4.1 Choix des pôles en boucle fermée

De manière générale le comportement en boucle fermée est spécifié sous la forme d'une pulsation propre  $\omega_0$  et d'un amortissement  $\sigma$ , donc une fonction de transfert du second ordre. La spécification étant généralement donnée en continu il est nécessaire de traduire la valeur désirée de  $\omega_0$  et  $\sigma$  en un polynôme  $P_D(q^{-1})$  par discrétisation de la fonction de transfert continue (chapitre 4.5 page 29). Ce polynôme  $P_D$  traduit le comportement dominant du système bouclé.

Si l'on souhaite introduire un effet de filtrage supplémentaire dans certaines bandes de fréquence (réduction des bruits de mesure, adoucissement de la commande, robustesse) cela est décrit par un polynôme  $P_F$  comportant des pôles auxiliaires, plus rapides que les pôles de  $P_D$ .

Le polynôme caractéristique en boucle fermée est donc constitué de pôles dominants et de pôles auxiliaires :

$$P(q^{-1}) = P_D(q^{-1}) \cdot P_F(q^{-1}) \quad (77)$$

### 7.4.2 Régulation : calcul de R et S

Le polynôme  $P(q^{-1})$  étant fixé il s'agit de trouver R et S vérifiant :

$$P(q^{-1}) = A(q^{-1}) \cdot S(q^{-1}) + q^{-d} \cdot B(q^{-1}) \cdot R(q^{-1}) \quad (78)$$

Comme déjà vu au chapitre 7.3.1 on obtient une équation de Bézout qu'il s'agit de résoudre. (Voir aussi 9.1 Annexe 1 : Résolution de l'équation polynomiale de Bezout)

Il est parfois nécessaire de spécifier R et S avec des parties fixes  $H_R$  et  $H_S$ . Dans ce cas l'équation (78) est modifiée et on l'écrit sous la forme:

$$P(q^{-1}) = (A(q^{-1}) \cdot H_S(q^{-1})) \cdot S'(q^{-1}) + (q^{-d} \cdot B(q^{-1}) \cdot H_R(q^{-1})) \cdot R'(q^{-1}) \quad (79)$$

Les inconnues que l'on souhaite trouver sont  $S'$  et  $R'$ . Ceci est nécessaire dans les cas particuliers suivants :

- spécification d'erreur statique nulle,
- rejet d'une perturbation harmonique,
- blocage d'un signal,

Détaillons maintenant ces cas particuliers.

### Spécification d'une erreur statique nulle

On retrouve la nécessité d'avoir un terme intégrateur dans la boucle directe. La spécification se retrouve donc sur  $S$  :

$$S(q^{-1}) = (1 - q^{-1}) \cdot S'(q^{-1}) = H_S(q^{-1}) \cdot S'(q^{-1}) \quad (80)$$

### Rejet d'une perturbation harmonique

Si le système est perturbé par une perturbation harmonique à une fréquence fixe (par exemple 50Hz...) il est possible d'imposer à  $S$  d'avoir un module *nul* ou amorti à cette fréquence. Cela se traduit par la présence dans  $S$  d'une paire de zéros complexe à amortissement nul ou faible à cette fréquence :

$$S(q^{-1}) = (1 + \alpha \cdot q^{-1} + \beta \cdot q^{-2}) \cdot S'(q^{-1}) = H_S(q^{-1}) \cdot S'(q^{-1}) \quad (81)$$

### Blocage d'un signal

Pour certaines application on souhaite que le régulateur "laisse passer" certaines composantes du signal de sortie, c'est-à-dire qu'il ne réagisse pas à ces composantes. Cela se traduit par  $R=0$  à ces fréquences. On retrouve la même forme que précédemment :

$$R(q^{-1}) = (1 + \alpha \cdot q^{-1} + \beta \cdot q^{-2}) \cdot R'(q^{-1}) = H_R(q^{-1}) \cdot R'(q^{-1}) \quad (82)$$

### 7.4.3 Poursuite : calcul de T

Le schéma-bloc de l'asservissement est donné sur la Figure 33. Le terme  $T$  est décomposé en un terme  $T'$  et un terme  $B^*/A^*$ .

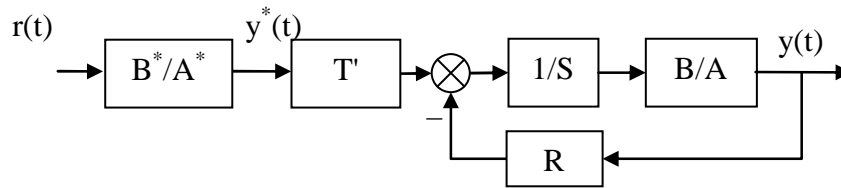


Figure 33 : placement de pôle

L'idée est de régler  $B^*$  et  $A^*$  de telle manière que le signal  $y^*$  ait la dynamique de poursuite désirée (par exemple un second ordre avec pulsation propre et amortissement donné) puis de régler  $T'$  de telle manière que le transfert entre  $y^*$  et  $y$  soit le plus "transparent" possible.

### Réglage de $T'$

Le transfert entre  $y^*$  et  $y$  est :

$$S_{yy^*}(q^{-1}) = T'(q^{-1}) \frac{q^{-d} \cdot B(q^{-1})}{P(q^{-1})} \quad (83)$$

Pour assurer le transfert le plus "transparent" possible il faut donc choisir

$$T'(q^{-1}) = \frac{P(q^{-1})}{B(1)} \quad (84)$$

Remarques :

- Si  $B(1)=0$  alors on remplace  $B(1)$  par 1 dans (84)
- De même dans (84)  $P(q^{-1})$  peut être remplacé par  $P_D$  correspondant à la dynamique dominante.

### Réglage de $B^*$ et $A^*$

$B^*$  et  $A^*$  sont choisis de telle manière que la fonction de transfert  $B^*/A^*$  corresponde à la dynamique de poursuite souhaitée.

## 7.5 Poursuite et régulation à objectifs indépendants

Cette méthode ressemble à la méthode de placement de pôles à ceci près que l'on va effectuer une simplification des zéros de  $B^*(q^{-1})$  : cela ne s'applique donc qu'aux systèmes avec des pôles stables. Plus précisément il faut qu'ils aient un amortissement suffisant ( $> 0,2$ ) de manière à ne pas être trop près du cercle unité.

La structure de l'asservissement est toujours celle donnée sur la Figure 33.



**Choix de P**

$P(q^{-1})$  est choisi de manière à simplifier  $B^*(q^{-1})$  (rappel : la forme de B est donnée dans l'équation 74). Comme indiqué dans le chapitre 7.4.2 page 60 il s'agit donc d'imposer les zéros de  $B(q^{-1})$  dans  $H_R(q^{-1})$  et  $H_S(q^{-1})$ .

**Choix de T**

Si l'on choisit  $T(q^{-1}) = P(q^{-1})$  alors le transfert entre  $y^*$  et  $y$  est :

$$S_{yy^*}(q^{-1}) = q^{-(d+1)} \quad (85)$$

**Réglage de  $B^*$  et  $A^*$** 

$B^*$  et  $A^*$  sont choisis de telle manière que la fonction de transfert  $B^*/A^*$  corresponde à la dynamique de poursuite souhaitée.



## 8. TECHNOLOGIE DE LA COMMANDE PAR ORDINATEUR

Du point de vue de l'automaticien le ordinateur est relié au système physique à piloter par un certain nombre d'intermédiaires plus ou moins gênants :

- les interfaces analogiques et numériques,
- le transport de l'information par bus et protocole,
- l'utilisation éventuelle de capteurs ou d'actionneurs « intelligents ».

Il ne s'agit pas ici de faire un exposé complet sur ces trois aspects mais de poser les problèmes qui se posent à l'automaticien lorsque le système qu'il doit commander est perturbé par ces éléments.

Enfin ce chapitre se termine sur les notions de « Hardware In the Loop » et de « prototypage rapide » qui sont partie intégrante des processus de réalisation de systèmes de contrôle.

### 8.1 Calculateurs et interfaces numériques / analogiques

Le premier problème à prendre en compte est la quantification de l'information. Du point de vue du ordinateur d'une part et du point de vue des convertisseurs d'autre part. Dans un premier temps, considérons les aspects performance de calcul lié à la nature du codage des nombres dans le ordinateur.

#### Les performances liées au types de variables

Prenons l'exemple où le concepteur de la loi de commande souhaite réaliser un correcteur de type proportionnel. Entre l'acquisition des mesures et l'écriture de la consigne il y aura la réalisation d'une de commande du type :

$$y = K \cdot (\text{consigne} - \text{mesure}) \quad (86)$$

En utilisant un langage de programmation évolué (par exemple le C) la ligne de code sera la transcription littérale de (86). Le programmeur devra décider de la définition et du type des variables, principalement du nombre d'octets utilisés pour stocker ses variables. Le Tableau 5 résume par exemple les types utilisés en langage C (norme ANSI)

Tableau 5 : Types ANSI C

Types entiers	char	8 bits
	short int	16 bits
	long int	32 bits
Types flottants (mantisse et exposant)	float	32 bits
	double	64 bits
	longdouble	128 bits

Une fois le code écrit en C l'exécution du programme sur un processeur cible ne peut se faire que par l'intermédiaire d'une compilation du programme. A chaque processeur est donc adapté une version de compilateur.

Les performances du programme (sa rapidité d'exécution) dépendront donc en partie de la qualité du compilateur. (Par exemple, une multiplication s'effectue t'elle bêtement par un décalage de tous les bits ou bien des solutions astucieuses sont-elles implémentées pour arrêter le décalage selon le contexte ?)

Le principal déterminant de la performance est cependant le processeur : son type (8 bits, 16 bits ou 64 bits) et son cadencement (en MHz ou GHz) sont les deux plus importants paramètres. Coder les variables sur 64 bits (double) dans le programme coûtera cher en temps de calcul si le processeur est un 8 bits. Le type de mémoire et leurs temps d'accès peuvent aussi être déterminants. Enfin, certains processeurs disposent d'un processeur spécifiquement dédié aux opérations mathématiques (coprocesseur mathématique).

Il n'est pas possible de dire exactement, pour tel type de compilateur associé à tel type de processeur, le temps que prendra par exemple une multiplication. En pratique on effectue une série de tests types (benchmark) et on affiche un temps moyen d'exécution.

En résumé, on peut être tenté, pour ne pas s'embêter, de coder toutes les variables en virgule flottante. Mais il faut bien savoir que cela a un coût en terme de temps de calcul, surtout si le processeur n'est pas adapté. Pour fixer les idées on peut dire qu'il y a un rapport 20 entre une addition réalisée en "short" et une addition réalisée en virgule flottante.

Enfin, l'utilisation de fonctions mathématiques (sinus, exponentielles, etc...) peut aussi avoir un coût très lourd si le travail n'est pas réalisé par un coprocesseur mathématique dédié. Le compilateur et sa librairie mathématique joue là encore un grand rôle.

### La quantification imposée par le processeur

Au-delà de l'aspect performance et temps de calcul, la quantification de l'information doit parfois aussi être prise en compte. Rappelons qu'un entier codé sur N bits peut prendre seulement  $2^N - 1$  valeurs différentes. Avant d'effectuer les calculs mathématiques dans le corps du programme il est donc judicieux d'appliquer un

facteur d'échelle de manière à utiliser l'amplitude maximale possible (entre 0 et  $2^N - 1$ ).

### **La quantification imposée par les conversions analogique et numérique**

Les convertisseurs analogique-numérique et numérique-analogique sont principalement caractérisés par leur amplitude de signal analogique d'entrée (généralement en volts, par exemple +/- 10V) et par leur résolution en bits (8 ou 10 bits pour les plus courants, jusqu'à 16 bits pour les convertisseurs haut de gamme).

Prenons l'exemple d'un capteur gyromètre transformant une vitesse de rotation en tension. Compte tenu de l'application le signal utile varie entre 2,4V et 2,6V. Si le signal mesuré est branché directement sur un convertisseur 0-10V 10bits alors un problème de conditionnement du signal se pose. En effet la quantification est de  $\frac{10V}{2^{10}-1} = 0,098V$ . Les 0,2V de signal utile seront donc discrétisés en  $\frac{0,2}{0,098} = 20$  intervalles seulement !

Si la discrétisation du signal utile est trop grossière cela peut se traduire par une diminution des performances de l'asservissement, voire une instabilité. Le système n'est en effet plus linéaire et il n'existe pas d'outils théoriques permettant de prévoir les performances du système bouclé.

## **8.2 Transport de l'information : les bus et les protocoles**

Pas question de faire ni même d'ébaucher un cours sur les bus et les protocoles ici. Mais on va quand même donner quelques indications. L'automaticien a l'habitude de simplifier le problème qu'il se pose en dessinant un schéma simplifié de son problème d'asservissement (du type de la Figure 20 page 45) pour pouvoir appliquer les théories de la commande. Cependant il arrive bien souvent que l'information transite à l'intérieur du système, soit entre différents ordinateurs effectuant différentes parties du travail soit entre ordinateur et périphériques, par l'intermédiaire de bus informatique. A chaque type de bus correspond un ou plusieurs protocoles qui peuvent poser des problèmes pour l'automaticien :

- les données sont-elles discrétisées ? (le protocole peut par exemple ne permettre de transmettre que des paquets de données de 16 bits),
- le bus induit-t'il un retard ? (le composant mettant en forme les données du ordinateur pour les publier sur le bus a certaines performances qui peuvent induire un délai)
- le bus est-il déterministe ? (Certains bus ne garantissent pas que les données arriveront en un temps donné),
- le bus est-il sûr ? (Certains bus ne garantissent pas non plus que les données arriveront bien)

### **8.3 Capteurs et actionneurs "intelligents", "smart sensors"**

De plus en plus souvent les capteurs et les actionneurs sont associés à leur électronique de puissance, électronique de commande et de conditionnement de signal. L'interface avec le capteur ou l'actionneur se fait donc via un protocole de communication (par exemple via un bus RS232).

L'intérêt est de véhiculer non pas des signaux analogiques, tensions ou courants, mais des signaux numériques beaucoup moins sensibles aux éventuelles perturbations.

On retrouve donc, entre le capteur ou l'actionneur et le ordinateur, la présence d'un bus et d'un protocole de communication.

### **8.4 La philosophie "hardware in the loop"**

Les systèmes embarqués sont de plus en plus complexes et touchent des domaines de plus en plus variés : automobile, satellites, aéronautique, UAV, systèmes d'arme, sous-marins, systèmes, etc... Les systèmes à piloter ainsi que les systèmes de commande sont de plus en plus complexes.

Les techniques de simulation « Hardware In the Loop » (HIL) sont de plus en plus utilisées dans le développement et dans le test de ces systèmes embarqués. L'idée est de développer entièrement le système de contrôle sur une plateforme de test en utilisant le maximum de composants réels et en ne remplaçant que le minimum par la simulation. Par exemple en aéronautique l'« Iron Bird » regroupe dans un hangar tous les systèmes réels de l'avion en cours de développement. Seule la mécanique du vol est simulée : les sorties des actionneurs sont mesurées, rentrées dans un modèle de simulation numérique, puis les mesures effectuées normalement par les capteurs sont calculées et restituées par le modèle.

L'intérêt est de diminuer de manière importante les coûts et les délais en économisant le plus grand nombre possible de tests sur le système réel en situation. Lors des premiers tests réels les techniques « Hardware In the Loop » permettent d'avoir effectué la plus grande part du développement des lois de commande.

### **8.5 Le prototypage rapide en automatique**

Un peu dans le même ordre d'idée que le « Hardware In the Loop » le prototypage rapide permet de réaliser la conception des lois de commande en simulation puis de transporter rapidement (en un simple click !) le code du contrôleur simulé vers la cible temps réel implémentée sur le système réel.

La société Mathworks [9] investit massivement dans ce domaine en ayant enrichi son logiciel de calcul scientifique (Matlab) avec un logiciel de simulation graphique très évolué (Simulink) puis en développant des outils de prototypage rapide associés (Real Time Workshop, Embedded Target For...).

## 9. ANNEXE

### 9.1 Annexe 1 : Résolution de l'équation polynomiale de Bezout

Il s'agit de trouver R et S vérifiant :

$$P(q^{-1}) = A(q^{-1}).S(q^{-1}) + B(q^{-1}).R(q^{-1})$$

Où P, A et B sont connus.

Dans le cas où P, A, B, R et S sont des nombres entiers la solution de l'algorithme est donnée par l'algorithme de division Euclidienne étendu [8].

Dans le cas où P, A, B, R et S sont des polynômes, l'algorithme est donné ci-dessous, programmé en matlab ([1]).

```
function [Rp,Sp,nrp,nsp]=bezoutd(A,B,Hs,Hr,P)
%function [Rp,Sp,nrp,nsp]=bezoutd(A,B,Hs,Hr,P)
%solves AHsSp+BHrRp=P by coefficient comparison. Delay and discretization delay need to be integrated in B
%(zeros at the beginning).
%
%inputs:
%A=[a0 a1 ... aNa] ... vector of model denominator coefficients A=a0 + a1z^(-1) + a2z^(-2) +...+ aNaz^(-Na)
%B=[b0 b1 ... bNb] ... vector of model numerator coefficients B=b0 + b1z^(-1) + b2z^(-2) +...+ bNbz^(-Nb)
%Hs=[hs0 hs1 ... hsNhs] ... vector of controller denominator fixed part Hs=hs0 + hs1z^(-1) +...+ hsNhsz^(-Nhs)
%Hr=[hr0 hr1 ... hrNhr] ... vector of controller denominator fixed part Hr=hr0 + hr1z^(-1) +...+ hrNhrz^(-Nhr)
%P=[p0 p1 ... pNp] ... vector of desired polynomial coefficients P=p0 + p1z^(-1) + p2z^(-2) +...+ pNpz^(-Np)
%outputs:
%Rp=[rp0 rp1 rp2 ...] ... vector of coefficients for resulted controller numerator
%Sp=[sp0 sp1 sp2 ...] ... vector of coefficients for resulted controller denominator
%nrp ... order of Rp
%nsp ... order of Sp
%
%written by: J. Langer, I.D. Landau, H. Prochazka
%7th june 2002

PRECISION=1e-16;

D=size(A);
if D(1)>1, A=A'; end;
```

```

D=size(B);
if D(1)>1, B=B'; end;
D=size(Hs);
if D(1)>1, Hs=Hs'; end;
if D(1)==0, Hs=1; end;
D=size(Hr);
if D(1)>1, Hr=Hr'; end;
if D(1)==0, Hr=1;end;
D=size(P);
if D(1)>1, P=P'; end;

na=length(A)-1;
nb=length(B)-1;
np=length(P)-1;
nhs=length(Hs)-1;
nhr=length(Hr)-1;

if (nhs>0), Ah=real(convz(A,Hs)); else Ah=A*Hs; end; % Ah = A * Hs
nah=length(Ah) -1;
if (nhr>0), Bh=real(convz(B,Hr)); else Bh=B*Hr; end;
nbh=length(Bh) -1;
if (np>nah+nbh-1), disp('Bezout error: too many poles');end;

% increase size of P using zeros if necessary
if (np<nah+nbh-1),
%
% set remaining poles onto a circle with radius rmin
P=[P zeros(1,nah+nbh-1-np)];
rootsPdes=roots(P);
nextra=nah+nbh-1-np;
rmin=1e-16;
angle=[0:nextra-1]'/nextra*2*pi;
j=sqrt(-1);
rootsPextra=rmin*exp(j*angle);
P=poly([rootsPdes;rootsPextra]);
np=nah+nbh-1;

end;
P,

nsp=nbh-1;
nrp=nah-1;

%matrix is smaller than vector PD

```



```

if (np>nah+nbh-1),
    disp('Order of model denominator is too low! Add a polynom of higher order to Hs or Hr. ');
end;

% ns=nsp+nhs
% nr=nrp+nhr

M=[];
for j=1:nsp+1,
    V=[];
    if (j>1), V=[V ; zeros(j-1,1)]; end;% zeros in front of Ah
    V=[V ; Ah'];% Ah
    if (j<=nsp), V=[V ; zeros(nsp+1-j,1)]; end;% zeros behind Ah
    if (length(V)~=nah+nbh), disp('bezoutb: error V'); end;
    M=[M V]; % add one column to M
end;

for j=1:nrp+1,
    V=[];
    if (j>1), V=[V ; zeros(j-1,1)]; end;
    V=[V ; Bh'];
    if (j<=nrp), V=[V ; zeros(nrp+1-j,1)]; end;
    if (length(V)~=nah+nbh), disp('bezoutb: error V'); end;
    M=[M V];
end;

D=size(M);
if (D(1)~=nah+nbh), disp('bezoutb: error size M row'); end;
if (D(2)~=nah+nbh), disp('bezoutb: error size M column'); end;

% make P column vector
P=P';

global M1;
M1=M;

X= M\P;
% coefficients are real values
X=real(X);

% make X row vector
X=X';

```

## Commande des systèmes par ordinateur

```
Sp=X(1:nsp+1);  
Rp=X(nsp+2:nsp+nrp+2);
```

## 10. BIBLIOGRAPHIE

- [1] I. D. Landau, "*Commande des systèmes*", Edition Lavoisier 2002.
- [2] C. Nouals, "Théorie du Signal", Notes de cours ENSICA
- [3] F. Vincent, "*Traitement Numérique du Signal*", Notes de cours ENSICA.
- [4] D. Arzelier, "*Représentation et analyse des systèmes, notes de cours*", Edition ENSICA. Voir aussi <http://www.laas.fr/~arzelier/>
- [5] J. Bordeneuve-Guibé, "*Commande des systèmes linéaires*", Notes de cours ENSICA.
- [6] Y. Briere, "*Identification*", Notes de cours ENSICA.
- [7] L. Ljung, "*System Identification, Theory for the User*", Prentice Hall 1987.
- [8] Wikipedia , équation de Bezout : [http://en.wikipedia.org/wiki/Bézout's\\_identity](http://en.wikipedia.org/wiki/Bézout's_identity)
- [9] <http://www.mathworks.com>



# 11. INDEX

<b>A</b>	
actionneur intelligent .....	21
amortissement .....	15
ANSI C .....	60
Approximation numérique d'équation différentielle .....	28
ARMAX .....	35
<b>B</b>	
BJ35 .....	
Blocage d'un signal .....	56
bloqueurs d'ordre 0 .....	22
BOZ .....	22
bus.....	21, 61
<b>C</b>	
CAN.....	21
capteur intelligent .....	21
causal .....	6, 9
Causalité .....	10
CNA.....	21
compilateur .....	60
convertisseur analogique / numérique.....	21
convertisseur numérique / analogique.....	21
convertisseurs analogique-numérique .....	61
coprocesseur mathématique .....	60
Correspondances pôles - zéros.....	31
<b>D</b>	
Diagramme de Bode .....	18
Discrétisation de fonction de transfert continues .....	27
division polynomiale.....	6
<b>E</b>	
échelon unitaire.....	11
équation d'état .....	17
Erreur statique.....	42
Estimation de la réponse fréquentielle .....	39
Estimation de la réponse impulsionnelle.....	38
Euler .....	29
<b>F</b>	
facteur d'échelle .....	61
facteur d'oubli .....	36
<i>fenêtre de Hamming</i> .....	39
filtre anti-repliement .....	25
fonction de sensibilité .....	42
fonction de transfert discrète.....	9
<b>G</b>	
gain statique.....	11
gain statique unitaire.....	42
<b>I</b>	
Identification.....	32
Identification non paramétrique .....	38
intercorrélation.....	38
<b>L</b>	
lieu des racines.....	43
lieu d'Evans.....	43
LTI.....	8
<b>M</b>	
marge de gain.....	47
marge de module.....	46
marge de phase .....	47
méthode des moindres carrés récursifs .....	32
Méthode des trapèzes.....	29
modèle ARX .....	32
<b>O</b>	
OE.....	35
one step ahead prediction.....	35
opérateur de retard .....	9
<b>P</b>	
période d'échantillonnage (choix) .....	26
Placement de pôles .....	54
Pôles .....	11
Poursuite .....	56
<i>prédiction</i> .....	32
protocoles.....	61
Pseudo Random Binary Signal .....	34
pulsation .....	15
<b>Q</b>	
quantification .....	23, 24, 60
<b>R</b>	
R,S,T.....	41
rapidité.....	15
rayon de convergence .....	5, 9
Régulation.....	55

Rejet de perturbations .....	42
Rejet d'une perturbation harmonique .....	56
réponse à un échelon.....	11
réponse fréquentielle.....	18
Réponse impulsionnelle.....	10
Réponses indicielle .....	10
retard.....	23
root locus .....	43

### S

signal discret .....	5
smart sensors.....	62
stabilité.....	11
Systèmes du premier ordre.....	14
Systèmes du second ordre.....	15

### T

Théorème de Nyquist.....	46
théorème de Shannon.....	19, 24
théorème du retard .....	6
Transformée en z .....	5
transformée unilatérale .....	5
Tustin.....	29

### V

virgule flottante.....	60
------------------------	----

### Z

zero holder hold .....	23
zéros.....	11
zoh .....	23

# Commande des systèmes par ordinateur

Yves Briere  
ENSICA

Notes de cours  
2007





# TABLE DES MATIERES

Table des matières .....	1
1. Introduction .....	1
2. Rappels et prérequis .....	3
3. Signaux et systèmes discrets .....	5
3.1 Signaux discrets .....	5
3.1.1 Transformée en z .....	5
3.1.2 Signaux types .....	7
3.2 Systèmes discrets .....	9
3.2.1 Fonction de transfert discrète, définition .....	9
3.2.2 Notation "opérateur de retard $q^{-1}$ " .....	10
3.2.3 Réponse impulsionnelle .....	11
3.2.4 Causalité .....	11
3.2.5 Réponses indicielle .....	11
3.2.6 Régime stationnaire (gain statique) .....	12
3.2.7 Pôles, zéros, stabilité .....	12
3.2.8 Systèmes du premier ordre .....	15
3.2.9 Systèmes du second ordre .....	16
3.2.10 Calcul itératif de la réponse d'un système .....	18
3.2.11 Représentation par équation d'état .....	18
3.3 Réponse fréquentielle des systèmes échantillonnés .....	19
3.3.1 Diagramme de Bode d'un système échantillonné .....	19
3.3.2 Où l'on voit apparaître le théorème de Shannon .....	20
4. Systèmes continus échantillonnés .....	23
4.1 Structure des systèmes commandés par ordinateur .....	23
4.2 Échantillonnage des systèmes continus .....	24
4.3 Effets de l'échantillonnage .....	25
4.4 Où l'on revoit le théorème de Shannon .....	26
4.5 Discrétisation de fonction de transfert continues .....	29
4.5.1 Approximation numérique d'équation différentielle .....	30
4.5.2 Superposition des réponses indicielles .....	32
4.5.3 Correspondances pôles - zéros .....	33
4.5.4 Bilan .....	33
5. Identification des systèmes échantillonnés .....	35
5.1 Identification d'un modèle ARX par la méthode des moindres carrés récurrents .....	35
5.2 L'identification paramétrique en général et en pratique .....	37

5.2.1	Choix du signal d'excitation, traitement des mesures .....	37
5.2.2	Choix d'une structure de modèle.....	38
5.2.3	Choix d'un critère à minimiser.....	38
5.2.4	Choix d'un algorithme.....	38
5.2.5	Validation.....	39
5.2.6	Utilisation de logiciels .....	39
5.3	Identification "simple" .....	39
5.4	Identification non paramétrique .....	41
5.4.1	Estimation de la réponse impulsionnelle .....	41
5.4.2	Estimation de la réponse fréquentielle.....	42
6.	Systèmes bouclés .....	45
6.1	Fonction de transfert en boucle fermée .....	45
6.2	Erreur statique .....	46
6.3	Rejet de perturbations.....	46
6.4	Stabilité des systèmes à simple gain de rétroaction.....	47
6.5	Analyse des systèmes bouclés dans le domaine fréquentiel : stabilité.....	49
6.5.1	Théorème de Nyquist dans le cas où le système est stable en boucle ouverte (critère du revers) .....	50
6.5.2	Théorème de Nyquist dans le cas où le système est instable en boucle ouverte .....	50
6.5.3	Interprétations du lieu de Nyquist.....	50
7.	Méthodes de régulation des systèmes échantillonnés .....	53
7.1	Introduction .....	53
7.2	Structure d'un système commandé par ordinateur, hypothèses.....	54
7.2.1	Discretisation, choix de la période d'échantillonnage.....	54
7.2.2	Discretisation d'un correcteur continu .....	55
7.3	Régulateur PID numérique .....	55
7.3.1	Cas 1 .....	55
7.3.2	Cas 2 .....	58
7.4	Placement de pôles .....	59
7.4.1	Choix des pôles en boucle fermée .....	60
7.4.2	Régulation : calcul de R et S.....	60
7.4.3	Poursuite : calcul de T .....	61
7.5	Poursuite et régulation à objectifs indépendants .....	62
8.	Technologie de la commande par ordinateur.....	65
8.1	Calculateurs et interfaces numériques / analogiques.....	65
8.2	Transport de l'information : les bus et les protocoles .....	67
8.3	Capteurs et actionneurs "intelligents", "smart sensors" .....	68
8.4	La philosophie "hardware in the loop" .....	68
8.5	Le prototypage rapide en automatique .....	68
9.	Annexe .....	69
9.1	Annexe 1 : Résolution de l'équation polynomiale de Bezout.....	69
10.	Bibliographie.....	73
11.	Index.....	1





# 1. INTRODUCTION



Figure 1 : commande d'un systèmes de télémédecine par ordinateur, vu par Claude Serre



## 2. RAPPELS ET PREREQUIS

Ce polycopié ne comporte pas de rappels sur les systèmes continus. S'adressant aux étudiants en deuxième année de l'ENSICA il suppose connu les enseignements de première année et de seconde année en automatique et traitement du signal.

En première année le cours de théorie du signal [2] a rendu familier les notions de :

- transformée de Fourier, transformée de Laplace
- corrélation, convolution

Le cours de traitement numérique du signal [3] a abordé en plus :

- la discrétisation et la quantification des signaux.
- la transformée en z et la transformée de Fourier discrète.
- les filtres numériques RIF et RII.

En automatique [4] les bases de la représentation et de l'analyse des systèmes continus ont été posées. On ne reviendra pas dans ce cours sur les notions de :

- modèles mathématique des systèmes linéaires à temps continus à temps invariant (LTI) : fonction de transfert, équation d'état, pôles, zéros et dynamique.
- commandabilité et observabilité.
- stabilité des systèmes en boucle ouverte : pôles, critères de Routh-Hurwitz.
- stabilité des systèmes bouclés : critère de Nyquist, critère de Bode.
- lieu des racines.

En deuxième année les notions suivantes sont acquises [5] :

- synthèse des correcteurs par méthodes fréquentielles (avance et retard de phase, PID)
- commande par retour d'état (placement de pôles), commande Linéaire Quadratique (LQG)
- reconstruction de l'état par Observateur, observateur optimal de Kalman.





## 3. SIGNAUX ET SYSTEMES DISCRETS

### 3.1 Signaux discrets

#### 3.1.1 Transformée en z

Pour une approche mathématiquement rigoureuse de représentation des signaux discrets par la transformée en z consulter [3] et sa bibliographie. Dans ce cours on se cantonne à une approche utilitariste des outils et méthodes mathématiques tout en essayant de ne pas contredire fondamentalement la rigueur mathématique.

Un signal discret  $s(k)$  est une séquence de nombres réels appelés échantillons. L'indice  $k$  est généralement associé à un instant d'échantillonnage :  $t = k.T_e$ . La période d'échantillonnage  $T_e$  est supposée constante. Par définition la transformée en z du signal  $s(k)$  est la fonction de la variable  $z$ , notée abusivement  $s(z)$  :

$$s(z) = Z(s(k)) = \sum_{k=0}^{\infty} s(k) \cdot z^{-k} \quad (1)$$

La transformée définie ici est plus précisément appelée transformée unilatérale puisque les termes correspondant à  $k < 0$  ne sont pas pris en compte. Ceci n'a pas d'importance puisque les signaux que nous utilisons sont systématiquement causaux<sup>1</sup> :  $s(k) = 0$  pour  $k < 0$ .

Il y a bien évidemment un lien avec la transformée de Laplace et la transformée de Fourier (voir [3])

La fonction de la variable complexe  $s(z)$  est une série et ne converge pas forcément pour toute valeur de  $z$ . En toute rigueur il faudrait associer le rayon de convergence à chaque fonction en  $z$  :  $\{s(z) \text{ converge pour } |z| > \alpha\}$ .

Les automaticiens pour qui la transformée en z n'est bien souvent qu'une façon bien commode de regrouper des termes de récurrence utilisent souvent la pseudo-variable<sup>2</sup>  $q$ .

A partir d'une transformée en z  $s(z)$  il est possible de retrouver la séquence  $s(k)$  correspondante en appliquant la transformation inverse, dont on ne donnera pas la formule ici. A la place de faire appel à cette formule compliquée il est possible d'utiliser d'autres méthodes pour retrouver la séquence  $s(k)$  lorsque la fonction  $s(z)$  est donnée sous forme de fraction rationnelle :

<sup>1</sup> Pour une justification du terme « causal » voir page 11

<sup>2</sup> Voir page 10

- division polynomiale
- décomposition de la fraction rationnelle en éléments simples puis utilisation de transformations élémentaires telles que celles données dans le Tableau 2

La propriété fondamentale de la transformée en z est qu'elle est linéaire (la transformée en z d'une composition linéaire de signaux est la composition linéaire de la transformée en z de chaque signal). Les autres propriétés fondamentales sont données dans le Tableau 2.

Tableau 1 : propriétés de la transformée en z

1	$s(z) = Z(s(k)) = \sum_{k=0}^{\infty} s(k) \cdot z^{-k}$
2	$Z(\alpha \cdot s_1(k) + \beta \cdot s_2(k)) = \alpha \cdot Z(s_1(k)) + \beta \cdot Z(s_2(k))$
3	$Z(k \cdot s(k)) = -z \cdot \frac{ds(z)}{dz}$
4	$Z(c^k \cdot s(k)) = s\left(\frac{z}{c}\right)$
5	$Z(s(k-1)) = s(k=-1) + z^{-1} \cdot s(z)$
6	$Z(s(k+1)) = z \cdot s(z) - z \cdot s(k=0)$
7	$s(k=0) = \lim_{z \rightarrow \infty} (s(z))$
8	$s(k \rightarrow \infty) = \lim_{z \rightarrow 1} \left( \frac{z-1}{z} \cdot s(z) \right)$

Remarque 1 : la propriété 5 ne prend pas en compte le fait que le signal est causal : en pratique  $s(k=-1) = 0$ .

Remarque 2 : la propriété 5 justifie le fait que le terme  $z^{-1}$  soit appelé "opérateur de retard unitaire", cette propriété est aussi appelée "théorème du retard".

Remarque 3 : les propriétés 7 et 8 sont données sous réserve de convergence.

Transformées inverses

### Exemple 1 : transformée inverse par division polynomiale

On considère un signal  $s(k)$  dont la transformée en z est donnée par la fraction rationnelle :

$$S(z) = \frac{0.1 \cdot z^2 + 0.1 \cdot z}{z^3 - 2.6 \cdot z^2 + 2.2 \cdot z - 0.6} \quad (3.1)$$

L'objectif est d'obtenir  $S(z)$  sous forme polynomiale :

$$S(z) = x_0 + x_1 \cdot z^{-1} + x_2 \cdot z^{-2} + \dots \text{etc.} \quad (3.2)$$

Cette suite est généralement infinie, la réponse temporelle est donnée par la succession des termes  $\{x_i\}$ . En traitant l'équation (3.1) comme une simple division l'algorithme de division classique (celui appris en CM2) permet d'obtenir par récurrence tous les termes de la suite (3.2) :

$0.1 \cdot z^2$	$+ 0.1 \cdot z$				$z^3 - 2.6 \cdot z^2 + 2.2 \cdot z - 0.6$
$0.1 \cdot z^2$	$- 0.26 \cdot z$	$+ 0.22$	$- 0.06 \cdot z^{-1}$	$0.1 \cdot z^{-1} + 0.36 \cdot z^{-2} + 0.716 \cdot z^{-3}$	
0	$0.36 \cdot z$	$- 0.22$	$0.06 \cdot z^{-1}$		
	$0.36 \cdot z$	$- 0.936$	$0.792 \cdot z^{-1}$	$- 0.216 \cdot z^{-2}$	
0	$0.716$	$- 0.732 \cdot z^{-1}$	$\dots$	$\dots$	

Les éléments successifs de la suite  $s(k)$  sont donc  $s(k) = \{0, 0.1, 0.36, 0.716, \dots\}$ . Pour obtenir les termes suivants il faudrait continuer la division.

**Exemple 2 : transformée inverse par décomposition en éléments simples**

La deuxième méthode présentée consiste à décomposer la fraction rationnelle (3.1) en éléments simples puis d'utiliser les tables de transformées en  $z$  (Tableau 2). Pour notre exemple cela donne :

$$S(z) = \frac{0.1 \cdot z^2 + 0.1 \cdot z}{z^3 - 2.6 \cdot z^2 + 2.2 \cdot z - 0.6} = \frac{-z}{z-1} + \frac{0.5 \cdot z}{(z-1)^2} + \frac{z}{z-0.6}$$

La réponse temporelle est donc composée des trois éléments :

$$s(k) = -(1 \ 1 \ 1 \ \dots) + 0.5 \cdot (1 \ 2 \ 3 \ \dots) + (1 \ 0.6 \ 0.6^2 \ \dots)$$

$$\text{soit : } s(k) = \begin{cases} s(0) = -1 + 0.5 + 1 \\ s(2) = -1 + 0.5 \cdot 2 + 0.6 \cdot 0.5 \\ s(3) = -1 + 0.5 \cdot 3 + 0.6^2 \\ s(4) = \dots \end{cases}$$

**3.1.2 Signaux types**

Le Tableau 2 donne les transformée en  $z$  de certains signaux fondamentaux. La Figure 2 représente l'allure de quelques signaux types.

Tableau 2 : transformée en z élémentaires

$s(k)$	$S(z)$
signal temporel	Transformée en z
$\delta(k)$ : impulsion unité	1
$u(k)$ : échelon unité	$\frac{z}{z-1}$
$k \cdot u(k)$	$\frac{z}{(z-1)^2}$
$c^k \cdot u(k)$	$\frac{z}{z-c}$
$k \cdot a^k \cdot u(k)$	$\frac{a \cdot z}{(z-a)^2}$
$\sin(\omega \cdot k) \cdot u(k)$	$\frac{z \cdot \sin(\omega)}{z^2 - 2 \cdot z \cdot \sin(\omega) + 1}$
$\cos(\omega \cdot k) \cdot u(k)$	$\frac{z \cdot (z - \cos(\omega))}{z^2 - 2 \cdot z \cdot \sin(\omega) + 1}$
$a^k \cdot \sin(\omega \cdot k) \cdot u(k)$	$\frac{a \cdot z \cdot \sin(\omega)}{z^2 - 2 \cdot a \cdot \cos(\omega) \cdot z + a^2}$
$a^k \cdot \cos(\omega \cdot k) \cdot u(k)$	$\frac{z \cdot (z - a \cdot \cos(\omega))}{z^2 - 2 \cdot a \cdot \cos(\omega) \cdot z + a^2}$

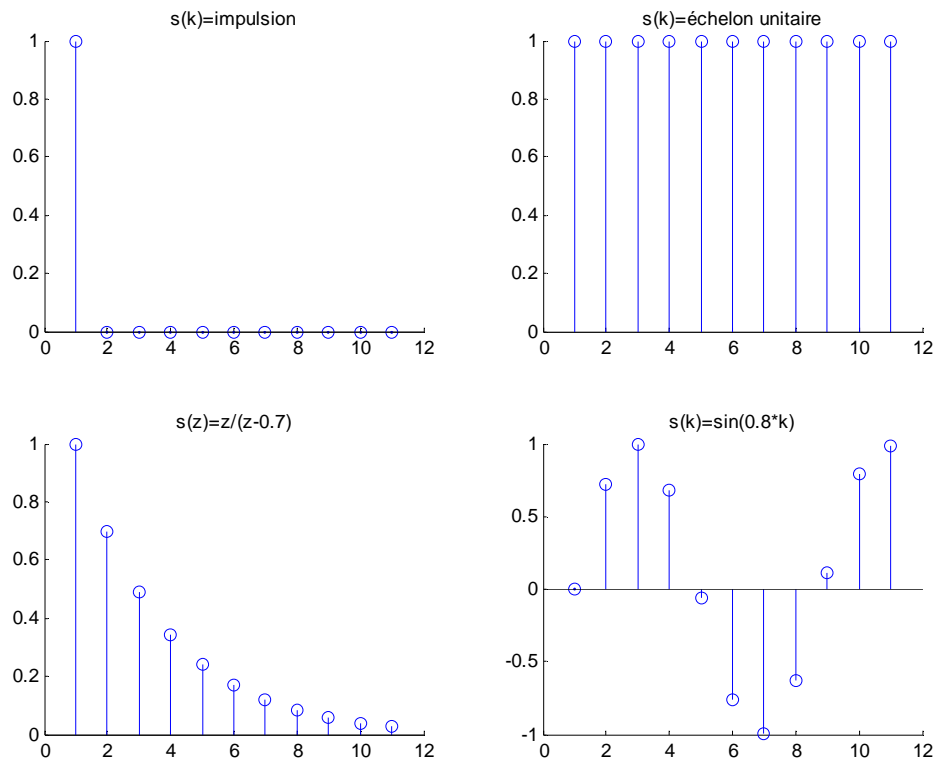


Figure 2 : réponses de quelques signaux types

## 3.2 Systèmes discrets

### 3.2.1 Fonction de transfert discrète, définition

On considère maintenant un système discret linéaire et invariant dans le temps (LTI) représenté sur la Figure 3.

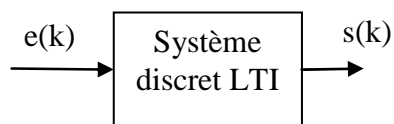


Figure 3 : système discret, notations

La relation entre la séquence d'entrée et la séquence de sortie est linéaire et définie par un ensemble de coefficients constants  $a_{1:n}$  et  $b_{1:m}$  :

$$\begin{aligned} s(k) + a_1 \cdot s(k-1) + a_2 \cdot s(k-2) + \dots + a_n \cdot s(k-n) = \\ b_0 \cdot e(k) + b_1 \cdot e(k-1) + b_2 \cdot e(k-2) + \dots + b_m \cdot e(k-m) \end{aligned} \quad (2)$$

On peut appliquer la transformée en  $z$  à chaque membre de l'équation puis en appliquant en récurrence la propriété 5 (théorème du retard) on obtient la fonction de transfert discrète du système :

$$\frac{s(z)}{e(z)} = F(z) = \frac{b_0 + b_1 \cdot z^{-1} + b_2 \cdot z^{-2} + \dots + b_m \cdot z^{-m}}{1 + a_1 \cdot z^{-1} + a_2 \cdot z^{-2} + \dots + a_n \cdot z^{-n}} \quad (3)$$

Remarque 1 : la fonction de transfert précédente est normalisée en prenant  $a_0 = 1$ .

Remarque 2 : on remarquera que la fonction de transfert est donnée en  $z^{-1}$  plutôt qu'en  $z$ . Cela est justifié par le fait que celle-ci traduit l'équation d'évolution du système physique causal (2).

Le script Matlab permettant de créer un objet Matlab correspondant à une fonction de transfert discrète est donné ci-dessous.

#### Matlab

```
% Création d'un système linéaire discret :
Te=0.1;
sysd = tf([b0 b1 b2],[1 a0 a2 a3],Te,'variable','z^-1');
```

On notera l'utilisation de l'argument 'variable', 'z^-1'.

### 3.2.2 Notation "opérateur de retard $q^{-1}$ "

La fonction de transfert est une fonction de la variable complexe  $z$ . Il faudrait donc en toute rigueur l'associer à son rayon de convergence. En pratique on ne s'embarrasse jamais de préciser cette zone de convergence. C'est pourquoi on trouve souvent la notation utilisant l'opérateur  $q^{-1}$  de décalage à la place de la variable complexe  $z^{-1}$  (4).  $F(q)$  n'est alors plus à proprement parler une fonction mais on lui appliquera les mêmes manipulations algébriques qu'à  $F(z)$ .

$$F(q) = \frac{b_0 + b_1 \cdot q^{-1} + b_2 \cdot q^{-2} + \dots + b_m \cdot q^{-m}}{1 + a_1 \cdot q^{-1} + a_2 \cdot q^{-2} + \dots + a_n \cdot q^{-n}} \quad (4)$$

Attention : selon les auteurs l'opérateur qui remplace  $z^{-1}$  est noté  $q$  (Matlab) ou  $q^{-1}$  (Landau [1]). Dans la suite du document c'est cette dernière convention qui sera appliquée.

L'utilisation de l'opérateur de retard  $q^{-1}$  à la place de la variable complexe  $z^{-1}$  permet de traiter du cas des fonctions de transfert à coefficients non constant, cas non prévu par la transformation en  $z$ .

### 3.2.3 Réponse impulsionnelle

Essayons d'appliquer au système un signal d'entrée particulier : une impulsion unitaire. La transformée en  $z$  du système est obtenue en multipliant la fonction de transfert par la transformée en  $z$  de l'impulsion unitaire : 1. On obtient :

$$s(z) = \frac{b_0 + b_1 \cdot z^{-1} + b_2 \cdot z^{-2} + \dots + b_m \cdot z^{-m}}{1 + a_1 \cdot z^{-1} + a_2 \cdot z^{-2} + \dots + a_n \cdot z^{-n}} \times 1 \quad (5)$$

Cela justifie le fait que la fonction de transfert du système peut aussi s'appeler "réponse impulsionnelle".

#### Matlab

```
% Réponse impulsionnelle d'un système :
[y,t] =impulse(sysd);
```

### 3.2.4 Causalité

Un système est dit causal si les sorties ne dépendent que du passé et non pas du futur. La conséquence est que la réponse impulsionnelle de  $F(z)$  ne peut pas comporter de terme non nuls pour  $k < 0$ . Cela justifie la confusion sémantique entre "système causal" et "signal causal", c'est-à-dire ne comportant pas de termes non nuls pour  $k < 0$ .

Attention : à la différence des systèmes continus un système physique (causal) discret peut très bien être modélisé avec un degré  $m$  supérieur au degré  $n$  : le système peut avoir plus de zéros que de pôles.

La transformée en  $z$  représentant des signaux échantillonnés elle peut tout aussi bien représenter les mesures prises à différents endroits d'une poutre. Le temps étant remplacé ici par une abscisse la notion de causalité ne s'applique pas ! Attention : beaucoup des propriétés de systèmes échantillonnés que l'on étudie sont liés à la causalité, prudence donc si l'on applique ces méthodes à d'autres domaines !

### 3.2.5 Réponses indicielle

Pour caractériser des systèmes on s'intéresse souvent à leur réponse impulsionnelle ou, plus souvent, à leur réponse indicielle. Pour calculer la réponse

à un échelon d'un système il faut donc multiplier la fonction de transfert par la transformée en z d'un échelon unitaire :

$$s(z) = \frac{b_0 + b_1 \cdot z^{-1} + b_2 \cdot z^{-2} + \dots + b_m \cdot z^{-m}}{1 + a_1 \cdot z^{-1} + a_2 \cdot z^{-2} + \dots + a_n \cdot z^{-n}} \times \frac{1}{1 - z^{-1}} \quad (6)$$

Il faudrait ensuite calculer la transformée en z inverse de s(z). En pratique, si l'on doit faire le calcul à la main on décompose en éléments simples puis on utilise des tables. Avec Matlab c'est plus simple. Voici par exemple le script pour obtenir les 20 premiers échantillons de la réponse indicielle d'un système :

**Matlab**

```
% Réponse indicielle d'un système :
[y,t] =step(sysd,20);
```

### 3.2.6 Régime stationnaire (gain statique)

Le gain statique est obtenu en appliquant une entrée indicielle et en calculant la valeur du régime permanent. En appliquant le théorème de la valeur finale il vient :

$$F_0 = \lim_{z \rightarrow 1} \left( \frac{z-1}{z} \cdot \frac{b_0 + b_1 \cdot z^{-1} + b_2 \cdot z^{-2} + \dots + b_m \cdot z^{-m}}{1 + a_1 \cdot z^{-1} + a_2 \cdot z^{-2} + \dots + a_n \cdot z^{-n}} \times \frac{1}{1 - z^{-1}} \right) \quad (7)$$

Soit :

$$F_0 = \frac{\sum_{i=0}^{i=m} b_i}{1 + \sum_{i=1}^{i=n} a_i} \quad (8)$$

Le gain statique est obtenu donc simplement en faisant le quotient de la somme des coefficients.

### 3.2.7 Pôles, zéros, stabilité

Comme pour les systèmes continus la dynamique et la stabilité des systèmes dépend des propriétés du dénominateur de la fonction de transfert.



$$F(z) = \frac{e_0 \cdot (1 + e_1 \cdot z^{-1}) \cdot (1 + e_2 \cdot z^{-1}) \cdots (1 + e_m \cdot z^{-1})}{(1 + c_1 \cdot z^{-1}) \cdot (1 + c_2 \cdot z^{-1}) \cdots (1 + c_n \cdot z^{-1})} \quad (9)$$

Les coefficients  $c_i$  sont les pôles de la fonction de transfert, les coefficients  $e_i$  sont les zéros. La fonction de transfert étant au départ composée de coefficients purement réels les coefficients  $c_i$  sont soit réels soit imaginaires par paires conjuguées.

Une fonction de transfert est donc aussi bien définie par la donnée de ses pôles et de ses zéros. Le script Matlab pour créer un système ainsi est donné ci-dessous :

#### Matlab

```
% Système discret défini par ses pôles et zéros :
Te=0.1;
sysd=zpk([c1 c2 c3],[e1 e2],e0,Te);
```

Si l'on décompose la fonction de transfert en éléments simples on obtient une somme d'éléments du type :

$$F(z) = \frac{d_1}{1 + c_1 \cdot z^{-1}} + \frac{d_2}{1 + c_2 \cdot z^{-1}} + \cdots + \frac{d_n}{1 + c_n \cdot z^{-1}} \quad (10)$$

Un système linéaire temps invariant est dit stable si sa réponse impulsionnelle tend asymptotiquement vers 0. Compte tenu de (10) cela nécessite que chaque pôle  $c_i$  soit de module inférieur à 1.

**Propriété** : un système discret LTI est stable si et seulement si tous ses pôles ont leur modules inférieur ou égal à 1.

#### Critère algébrique attestant de la stabilité : critère de Routh

On rappelle que le critère de Routh permet de prédire si les racines d'un polynôme sont toutes à partie réelle négative, donc stables. Énonçons le théorème de Routh dans le cas d'un polynôme  $P'(w)$  :

$$P'(w) = a_0 + a_1 \cdot w + a_2 \cdot w^2 + \cdots + a_n \cdot w^n \quad (11)$$

Dans un premier temps on construit le tableau :

$$\left| \begin{array}{cccccc} \beta_{n,0} & \beta_{n,1} & \beta_{n,2} & \beta_{n,q-1} & \beta_{n,q} & 0 \\ \beta_{n-1,0} & \beta_{n-1,1} & \beta_{n-1,2} & \beta_{n-1,q-1} & \beta_{n-1,q-1} & 0 \\ \beta_{n-2,0} & \beta_{n-2,1} & \beta_{n-2,2} & \beta_{n-2,q-1} & \beta_{n-2,q-1} & \\ \\ \beta_{2,0} & \beta_{2,1} & 0 & & & \\ \beta_{1,0} & 0 & & & & \\ \beta_{0,0} & 0 & & & & \end{array} \right| \quad (12)$$

Puis on vérifie que tous les premiers éléments de la première colonne ont le même signe : cela garantit que tous les pôles de  $P'(w)$  ont leur partie réelle négative.

Les éléments du tableau sont construits par récurrence ligne par ligne en appliquant les formules suivantes :

Pour  $i = n$  et  $i = n - 1$  :

$$\begin{cases} \beta_{n,j} = a_{n-2j} \\ \beta_{n-1,j} = a_{n-1-2j} \end{cases} \quad (13)$$

Et pour  $i < n-1$  :

$$\beta_{i,j} = \frac{\beta_{i+1,0} \cdot \beta_{i+2,j+1} - \beta_{i+1,j+1} \cdot \beta_{i+2,0}}{\beta_{i+1,0}} \quad (14)$$

Dans le cas qui nous intéresse, un système échantillonné, on cherche plutôt à vérifier que les pôles du polynôme  $P(z)$  sont de module inférieur à un. L'idée est d'effectuer un changement de variable  $z \rightarrow w$  pour pouvoir appliquer le critère de Routh.

$$P'(w) = (1-w)^n \cdot P\left(\frac{1+w}{1-w}\right) \quad (15)$$

Si l'on effectue la transformation (15) alors le fait que  $P'(w)$  ait toutes ses racines à partie réelle négative garantit que  $P(z)$  a toutes ses racines dans le cercle unité. Pour vérifier la stabilité de  $P(z)$  il suffit donc d'effectuer le changement de variable (15) puis d'appliquer le critère de Routh conventionnel (12 13 14).

### Critère algébrique attestant de la stabilité : critère de Jury

Le critère de Jury s'apparente au critère de Routh. Nous le donnons pour un polynôme de degré 2, 3 et 4 du type :

$$P(z) = a_0 + a_1 \cdot z + a_2 \cdot z^2 + \dots + a_n \cdot z^n \quad (16)$$

Les conditions de stabilité sont données ci-dessous (ces conditions sont données dans le cas  $a_n > 0$ , il suffit de multiplier tous les termes par  $-1$  si cette condition n'est pas remplie par  $P(z)$ ) :

$$\begin{aligned}
 n = 2 & \begin{cases} a_0 + a_1 + a_2 > 0 \\ a_0 - a_1 + a_2 > 0 \\ -a_0 + a_2 > 0 \end{cases} \\
 n = 3 & \begin{cases} a_0 + a_1 + a_2 + a_3 > 0 \\ -a_0 + a_1 - a_2 + a_3 > 0 \\ -|a_0| + a_3 > 0 \\ a_0 a_2 - a_1 a_3 - a_0^2 + a_3^2 > 0 \end{cases} \\
 n = 4 & \begin{cases} a_0 + a_1 + a_2 + a_3 + a_4 > 0 \\ a_0 - a_1 + a_2 - a_3 + a_4 > 0 \\ a_4^2 - a_0^2 - |a_0 a_3 - a_1 a_4| > 0 \\ (a_0 - a_4)^2 (a_0 - a_2 + a_4) + (a_1 - a_3)(a_0 a_3 - a_1 a_4) > 0 \end{cases}
 \end{aligned} \quad (17)$$

### 3.2.8 Systèmes du premier ordre

On donne dans la Figure 4 les réponses indicielles de systèmes du premier ordre ainsi que la localisation des pôles dans le plan complexe. Le cercle de rayon unité représente la zone dans laquelle les pôles doivent se situer pour assurer la stabilité. On remarque sur les deux premières figures l'influence de la place du zéro par rapport au point  $(1,0)$  pour la rapidité.

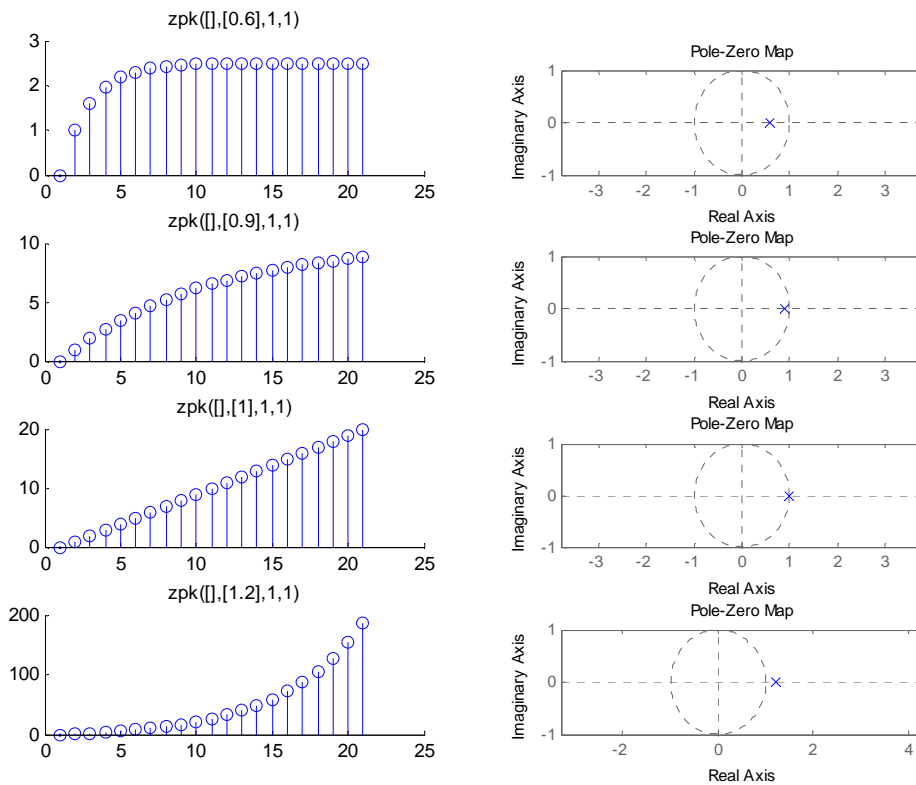


Figure 4 réponses indicielles de systèmes du premier ordre

### 3.2.9 Systèmes du second ordre

On donne dans la Figure 6 les réponses indicielles de systèmes du second ordre ainsi que la localisation des pôles dans le plan complexe. La rapidité et l'amortissement dépend de la position des pôles dans le cercle unité. La Figure 5 représente les courbes iso-amortissement et iso-pulsation des systèmes du second ordre.

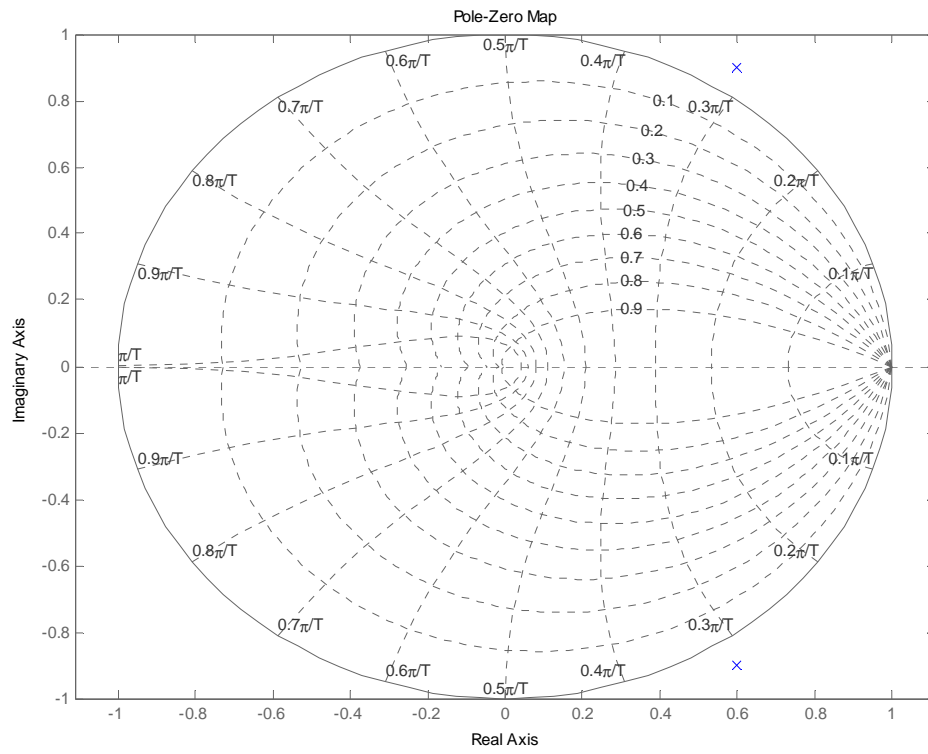


Figure 5 : courbes iso-amortissement et iso-pulsation des systèmes discrets du second ordre

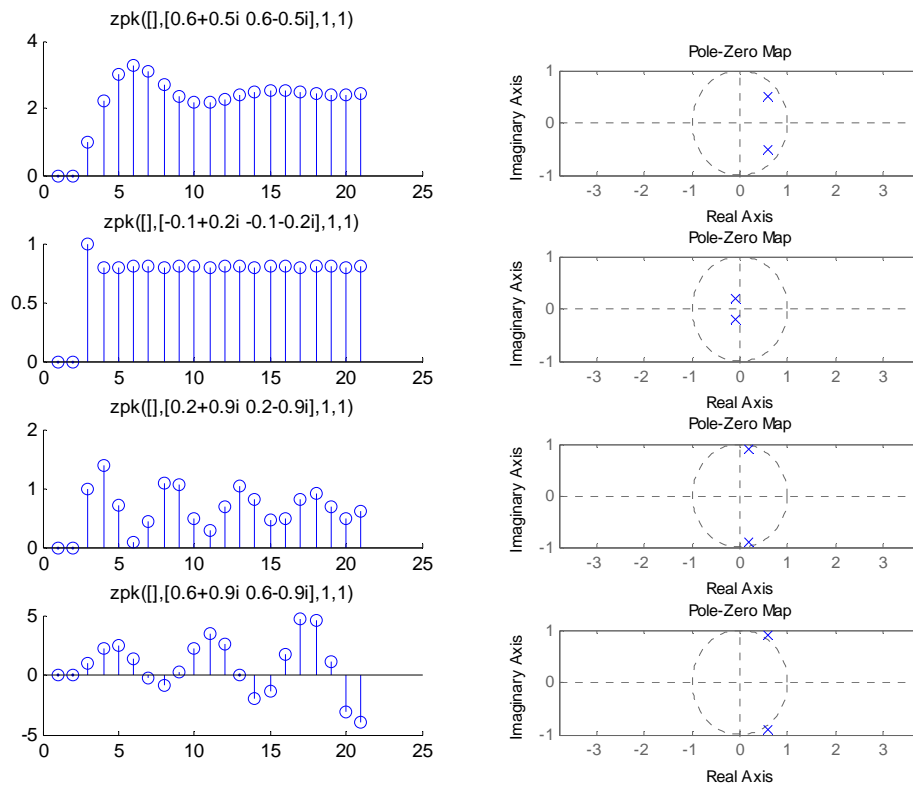


Figure 6 : réponses de systèmes du second ordre

### 3.2.10 Calcul itératif de la réponse d'un système

Si l'on dispose de la fonction de transfert d'un système et que l'on connaît les valeurs successives  $e(k)$  du signal d'entrée il est facile de calculer itérativement la sortie  $y(k)$  du système par l'équation de récurrence (2). Il faut initialiser par les  $n$  valeurs initiales de  $s(k)$ , prises généralement égales à 0.

### 3.2.11 Représentation par équation d'état

Un système discret peut aussi être représenté par un ensemble d'équations d'état comme dans le domaine continu (52). Cette représentation est naturellement adaptée aux systèmes multi-entrées multi-sorties. On définit alors un état interne  $x(k)$  ainsi qu'une fonction d'évolution  $f$  et une équation de mesure  $h$ .

$$\begin{cases} x(k+1) = f(x(k), u(k)) \\ s(k) = h(x(k), u(k)) \end{cases} \quad (18)$$

### 3.3 Réponse fréquentielle des systèmes échantillonnés

#### 3.3.1 Diagramme de Bode d'un système échantillonné

Lorsqu'on étudie la réponse fréquentielle des systèmes continus on considère des entrées particulières sinusoïdales :

$$\begin{cases} e(t) = \cos(2 \cdot \pi \cdot f \cdot t) + j \cdot \sin(2 \cdot \pi \cdot f \cdot t) = e^{j \cdot 2 \cdot \pi \cdot f \cdot t} \\ f \in [0 \quad \infty] \end{cases} \quad (19)$$

De même, pour des systèmes échantillonnés on considèrera des entrées sinusoïdales échantillonnées :

$$\begin{cases} e(k) = \cos(2 \cdot \pi \cdot f \cdot k) + j \cdot \sin(2 \cdot \pi \cdot f \cdot k) = e^{j \cdot 2 \cdot \pi \cdot f \cdot k} \\ f \in [0 \quad 1] \end{cases} \quad (20)$$

La sortie est aussi un signal échantillonné sinusoïdal de gain  $A(f)$  et de phase  $\Phi(f)$  variables et dépendant de  $f$ ; la fréquence réduite.

$$\begin{cases} s(k) = A(f) \cdot e^{j \cdot (2 \cdot \pi \cdot f \cdot k + \Phi(f))} \\ f \in [0 \quad 1] \end{cases} \quad (21)$$

Le nombre complexe, noté abusivement  $F(f)$ , d'amplitude  $A(f)$  et de phase  $\Phi(f)$  est obtenu à partir de  $F(z)$  en remplaçant  $z$  par  $e^{j \cdot 2 \cdot \pi \cdot f}$  :

$$\begin{cases} F(f) = F(z = e^{j \cdot 2 \cdot \pi \cdot f}) \\ f \in [0 \quad 1] \end{cases} \quad (22)$$

Le diagramme représentant le gain et la phase en fonction de la fréquence réduite s'appelle aussi un diagramme de Bode. L'échelle des fréquences est généralement linéaire de 0 à 0.5 (puisque le diagramme est nécessairement symétrique). La Figure 7 représente par exemple le diagramme de Bode d'un système du second ordre.

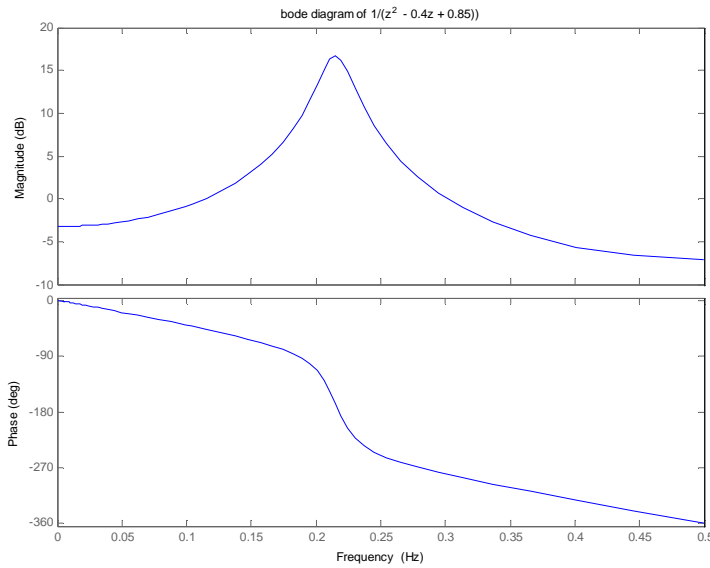


Figure 7 : exemple du diagramme de Bode d'un système discret du second ordre

### 3.3.2 Où l'on voit apparaître le théorème de Shannon

Le changement de variable  $z = e^{j2\pi f}$  n'est pas bijectif puisque  $e^{j2\pi f} = e^{j2\pi(f+k)}$  avec  $k$  entier. Si l'on fait le lien entre domaine continu et domaine fréquentiel on peut remarquer que le changement de variable  $z = e^{j2\pi f}$  fait correspondre une zone du plan complexe continu à une zone du plan complexe discret. La transformation est bijective si l'on se contente de la zone du plan complexe continu contenu entre les droites d'ordonnée 0 et 1.



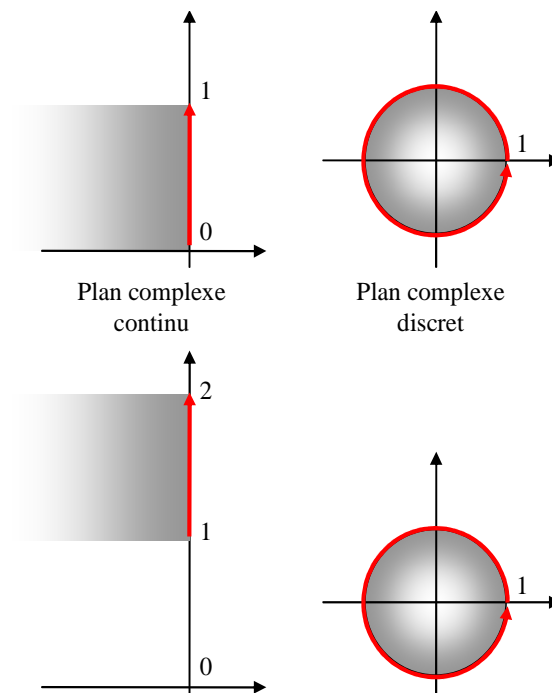


Figure 8 : correspondance entre plan complexe continu et plan complexe discret.



## 4. SYSTEMES CONTINUS ECHANTILLONNES

### 4.1 Structure des systèmes commandés par ordinateur

Les systèmes commandés par ordinateurs sont presque toujours des systèmes physiques continus par nature. Le ordinateur traite par contre les entrées et les sorties de manière discrète avec une période d'échantillonnage  $T_e$ . Les éléments importants, du point de vue de l'automaticien sont donc :

- le convertisseur analogique / numérique CAN ou du moins le système qui fournit séquentiellement les mesures au ordinateur à la période  $T_e$ .
- le ordinateur lui-même. Il doit effectuer l'acquisition des mesures, le traitement des informations, l'envoi des consignes dans un temps inférieur à  $T_e$ .
- le convertisseur numérique / analogique CNA.
- le système physique.

On suppose, pour faire simple et pour pouvoir utiliser la panoplie des outils mathématiques, que les systèmes sont échantillonnés à la même période  $T_e$  et au même moment.

La Figure 9 montre le système physique commandé par le ordinateur, dans sa version simpliste.

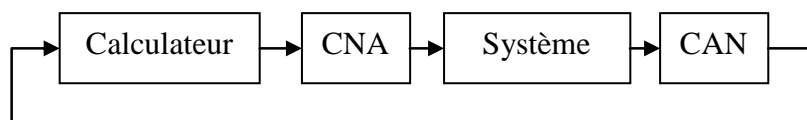


Figure 9 : système commandé par ordinateur

Au XXI<sup>ème</sup> siècle les choses sont quand même parfois un petit peu plus compliquées :

- le ordinateur fournit bien souvent un signal de commande numérique (transporté par bus informatique USB, RS232, CAN, Ethernet, etc...) directement interprété par l'actionneur. On parle "d'actionneur intelligent".
- de même le "capteur intelligent" fournit aussi bien souvent la mesure sous forme d'un signal numérique.
- Les signaux sont transportés par des bus informatiques qui n'ont pas que ça à faire : il peut y avoir du retard.
- de même les capteurs et actionneurs sont rarement tous seuls, ce qui se traduit par des périodes d'échantillonnage différentes, et parfois non synchrones...

- enfin le ordinateur peut aussi avoir d'autres choses à faire...

Ceci étant dit nous allons pour le moment nous restreindre au cas représenté sur la Figure 9 avec des instants d'échantillonnage synchrones et régulièrement espacés d'un temps  $T_e$ .

## 4.2 Echantillonnage des systèmes continus

On dispose généralement de la fonction de transfert continue (notée en utilisant la variable complexe  $s$  ou  $p$ )  $F(s)$  du système continu. Le problème est d'obtenir la fonction de transfert discrète du système échantillonné de manière à pouvoir synthétiser un correcteur. Le système continu est associé au convertisseur numérique – analogique qu'il faut prendre en compte.

Les convertisseurs numérique analogiques peuvent généralement se représenter par des "échantillonneurs – bloqueurs d'ordre 0" ou BOZ. Cela signifie que la commande est maintenue à une valeur constante pendant toute la durée du temps d'échantillonnage.

Si l'on applique une commande discrète impulsionnelle à un BOZ il va maintenir la commande 1 pendant la durée  $T_e$  puis revenir à zéro (Figure 10).

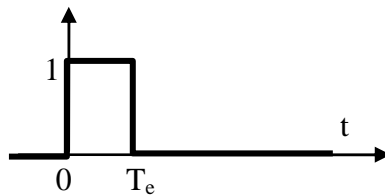


Figure 10 : réponse impulsionnelle d'un bloqueur d'ordre 0

La réponse impulsionnelle  $BOZ(t)$  est donc composée d'un échelon unitaire suivi d'un échelon unitaire de signe opposé et décalé d'un temps  $T_e$ . La transformée de Laplace de la réponse impulsionnelle du bloqueur est donnée par la formule (23) :

$$BOZ(s) = \frac{1}{s} - \frac{e^{-jT_e}}{s} \quad (23)$$

Du point de vue du ordinateur il faut donc prendre en compte le système continu de fonction de transfert  $F_{BOZ}(s)$  que l'on devra échantillonner :

$$F_{BOZ}(s) = \frac{1 - e^{-jT_e}}{s} \cdot F(s) \quad (24)$$

$$F_{\text{BOZ}}(z) = Z^{-1} \left( \frac{1 - e^{-jT_e}}{s} \cdot F(s) \right) \quad (25)$$

En pratique l'obtention de la transformée en z d'un système échantillonné par bloqueur d'ordre 0 se fait à l'aide de tables. On donne quelques exemples fondamentaux dans le Tableau 3 ci-dessous.

Tableau 3 : transformée en z de systèmes continus échantillonnés

F(s)	F <sub>BOZ</sub> (z)
1	1
$\frac{1}{s}$	$\frac{T_e \cdot z^{-1}}{1 - z^{-1}}$
$\frac{1}{1 + T \cdot s}$	$\begin{cases} \frac{b_1 \cdot z^{-1}}{1 + a_1 \cdot z^{-1}} \\ b_1 = 1 - e^{-T_e/T} \quad a_1 = -e^{-T_e/T} \end{cases}$
$\frac{e^{-Ls}}{1 + T \cdot s}$	$\begin{cases} \frac{b_1 \cdot z^{-1} + b_2 \cdot z^{-2}}{1 + a_1 \cdot z^{-1}} \quad a_1 = -e^{-T_e/T} \\ b_1 = 1 - e^{-(L-T_e)/T} \quad b_2 = e^{-T_e/T} (e^{L/T} - 1) \end{cases}$
$L < T_e$	

Si l'on dispose de Matlab c'est bien sûr encore plus simple. On utilise la fonction c2d avec l'argument 'zoh' pour "zero holder hold" (bloqueur d'ordre zéro).

#### Matlab

```
% Définition d'un système continu :
sysc=tf(1,[1 1]);
% Obtention de la transformée en z du système échantillonné à Te
Te=0.1;
sysd=c2d(sysc,Te,'zoh');
% Notation en z^-1
sysd.variable='z'
```

### 4.3 Effets de l'échantillonnage

Le convertisseur numérique a deux effets (néfastes) importants : l'effet de retard et l'effet de quantification.

L'effet de retard se voit dans (24) : on peut dire en première approximation que l'échantillonnage se traduit par un retard pur de  $T_e/2$ . Cela a donc, si ce temps

d'échantillonnage est non négligeable par rapport aux constantes de temps du système, un effet déstabilisant.

Pour illustrer l'effet de quantification supposons par exemple que l'on utilise un convertisseur analogique/numérique 8 bits 0-10V. Le plus petit incrément que le convertisseur peut distinguer est de  $10V/2^8 = 40mV$ . Si le signal utile varie physiquement entre 5V et 5,4V on n'aura que dix incréments utilisables... Ce phénomène n'entre pas dans la panoplie des phénomènes modélisables par la transformée en z. En pratique il est donc souvent nécessaire d'étudier le système et de synthétiser la commande numérique en négligeant ce phénomène puis de vérifier a posteriori qu'il n'a pas un effet trop néfaste.

#### 4.4 Où l'on revoit le théorème de Shannon

Le théorème de Shannon est bien connu en ce qui concerne l'échantillonnage de signaux discrets : on sait que le signal ne doit pas comporter de composante de fréquence supérieure à  $f_e/2$  si l'on ne veut pas observer de repliement. On doit aussi prendre des précautions si l'on veut travailler sur la fonction de transfert discrète correspondant à un système continu échantillonné.

On va traiter un exemple concret pour mettre en évidence le problème de repliement dans le cas de la commande des systèmes continus. On s'intéresse à deux systèmes continus du second ordre ayant des pôles de même partie réelle mais de parties imaginaires séparées de  $2\pi T_e$ . Le script Matlab ci-dessous montre la création des deux systèmes :

##### Matlab

```
% Définition de deux systèmes continus :  
sys1=zpk([],[-1-2i -1+2i],2^2+1);  
sys2=zpk([],[-1-(2+2*pi)*i -1+(2+2*pi)*i],1+(2+2*pi)^2);
```

On voit sur la Figure 11 que la réponse indicielle des deux systèmes est bien différente.

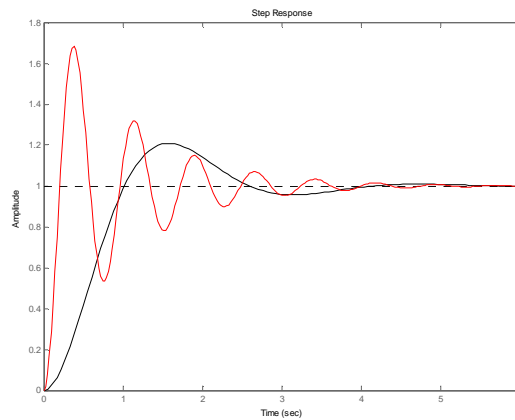


Figure 11 : réponse indicielle de deux systèmes du second ordre

Si l'on calcule la fonction de transfert échantillonnée des deux systèmes à la période de 1 seconde on obtient deux réponses indicielles très proches (représentés par des  $\circ$  sur la Figure 12). Conclusion : la période d'échantillonnage était mal choisie

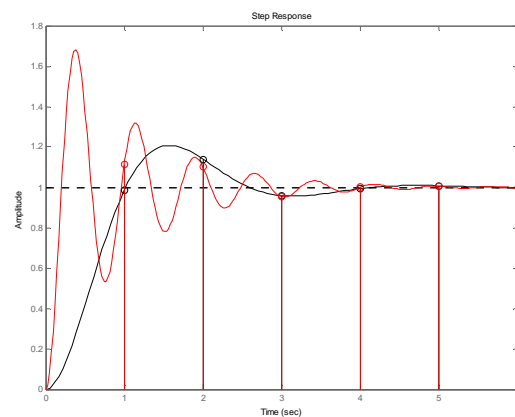


Figure 12 : réponse indicielle de deux systèmes du second ordre échantillonnés à 1s

En pratique la commande échantillonnée de systèmes continus doit être associée à certaines précautions :

- le système ne doit pas comporter de dynamiques propres situées dans les fréquences supérieures à  $f_c/2$  (exemple illustré dans ce chapitre).
- les signaux échantillonnés doivent être "nettoyés" des fréquences supérieures à  $f_c/2$  avec un filtre anti-repliement. (A moins que le signal soit suffisamment propre et ne comporte pas de composantes significatives au-delà de ces fréquences).

### Choix de la période d'échantillonnage

Le choix de la période d'échantillonnage est très important et dépend de la dynamique des signaux. Attention, les systèmes commandés en boucle fermée ont souvent une dynamique plus rapide que le système en boucle ouverte : il faut donc prendre en compte la dynamique en boucle fermée pour le choix de la fréquence d'échantillonnage.

En pratique on préconise de prendre une fréquence 5 à 25 fois supérieure à la bande passante souhaitée en boucle fermée.

On peut être tenté de prendre une fréquence d'échantillonnage très élevée si la technologie le permet (ordinateur et convertisseurs suffisamment rapide). Ceci peut amener rapidement à des problèmes numériques. Imaginons par exemple le cas d'un système ayant une dynamique très lente (quelques secondes) et un système de mesure codant les mesures sur 10 bits (cas classique). La Figure 13 illustre cet exemple. Le signal varie peu entre deux instants d'échantillonnage et les variations du signal sont faibles par rapport à la valeur de l'incrément numérique (figure du haut). Le calcul de la simple dérivée numérique se traduit donc par une succession de  $-1$   $0$  et  $+1$ .

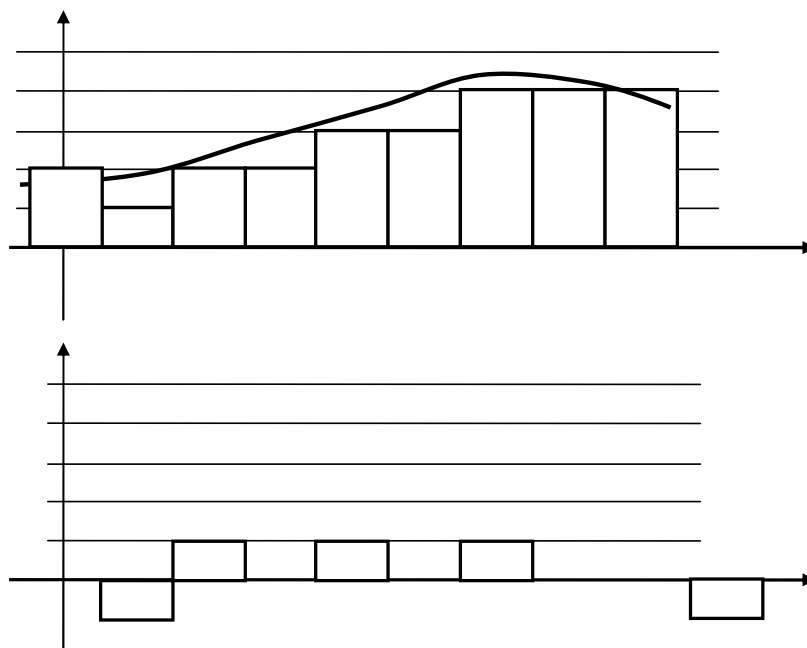


Figure 13 : période d'échantillonnage trop faible. En haut : signal échantillonné et discrétisé ; en bas : calcul de la dérivée numérique.



### Filtre anti-repliement

En toute rigueur chaque fois qu'un signal est échantillonné il faudrait faire précéder le convertisseur numérique d'un filtre analogique anti-repliement éliminant physiquement toutes les composantes du signal de fréquence supérieure à  $f_e/2$ .

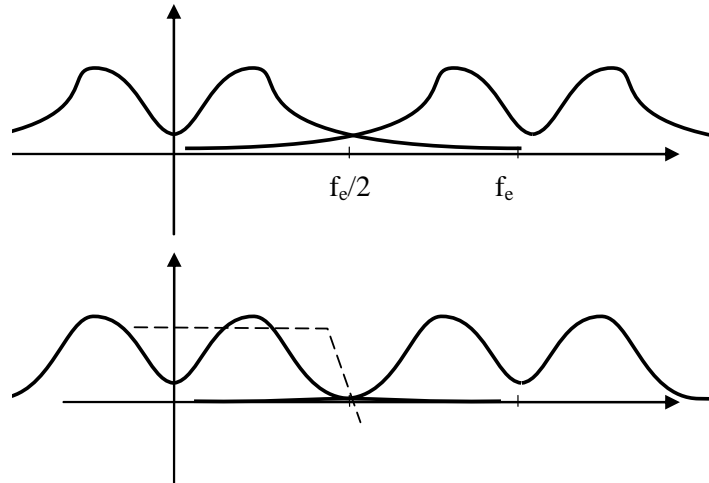


Figure 14 : Effet du filtre anti-repliement

Bien souvent la période d'échantillonnage de l'asservissement est relativement lente (quelques secondes) et il est alors impossible de réaliser un filtre de constante de temps aussi faible. On procède alors à un sur-échantillonnage : le signal est filtré, échantillonné à haute fréquence  $T'_e$  puis filtré numériquement et sous-échantillonné à la période  $T_e$  beaucoup plus grande que  $T'_e$  comme illustré sur la

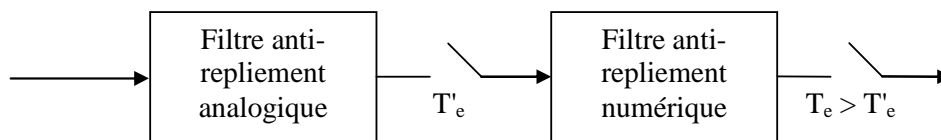


Figure 15 : Sur échantillonnage

Un simple filtre numérique de type moyenne mobile est souvent suffisant.

### 4.5 Discrétisation de fonction de transfert continues

Une manière classique pour réaliser un contrôle par ordinateur est de commencer par synthétiser un correcteur continu avec les outils de l'automatique classique puis de trouver un correcteur discret qui "approche" convenablement le correcteur continu.

#### 4.5.1 Approximation numérique d'équation différentielle

Supposons que le correcteur continu que l'on a synthétisé soit du type PD :

$$\text{PID}(s) = \frac{e(s)}{\varepsilon(s)} = k_p + k_D \cdot s \quad (26)$$

Il s'agit de calculer les commandes  $e(k)$  successives issues du correcteur en fonction des erreurs  $\varepsilon(k)$ . Une approche classique et intuitive est d'approximer la dérivée par une différence :

$$e(k) = k_p \cdot \varepsilon(k) + k_D \cdot \frac{\varepsilon(k) - \varepsilon(k-1)}{T_e} \quad (27)$$

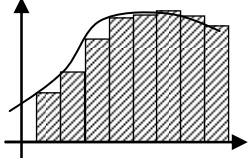
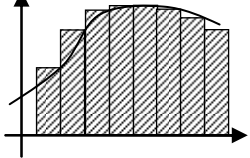
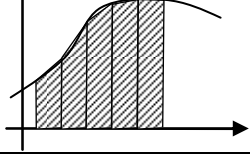
Si on calcule la transformée en  $z$  de  $e(k)$  on obtient :

$$e(z) = k_p \cdot \varepsilon(z) + k_D \cdot \frac{\varepsilon(z) - z^{-1}\varepsilon(z)}{T_e} = \left( k_p + k_D \cdot \frac{1 - z^{-1}}{T_e} \right) \cdot \varepsilon(z) \quad (28)$$

On retrouve la fonction de transfert PD(s) dans laquelle on a effectué le changement de variable  $s = \frac{1-z^{-1}}{T_e}$

De manière plus générale, l'approximation de fonction de transfert revient à effectuer l'approximation numérique d'intégration. Selon la méthode d'intégration choisie on effectue des changements de variable différents.

Tableau 4 : méthodes d'intégration

Méthode	Changement de variable	
Euler progressif (Euler's forward method)	$s = \frac{1 - z^{-1}}{z^{-1} \cdot T_e}$	
Euler rétrograde (Euler's backward method)	$s = \frac{1 - z^{-1}}{T_e}$	
Méthode des trapèzes, ou bilinéaire, ou de Tustin	$s = \frac{2}{T_e} \cdot \frac{1 - z^{-1}}{1 + z^{-1}}$	

En pratique le changement de variable  $s = \frac{1 - z^{-1}}{z^{-1} \cdot T_e}$  n'est pas sans risque. On donne sur la Figure 16 la correspondance entre la zone stable du plan complexe en  $s$  et la zone correspondante après la transformation (Euler progressif, Euler rétrograde, bilinéaire). On voit notamment qu'il est possible, partant d'un système continu stable, d'obtenir des pôles dans le domaine discret en dehors du cercle de stabilité si la transformation appliquée est celle d'Euler progressif.

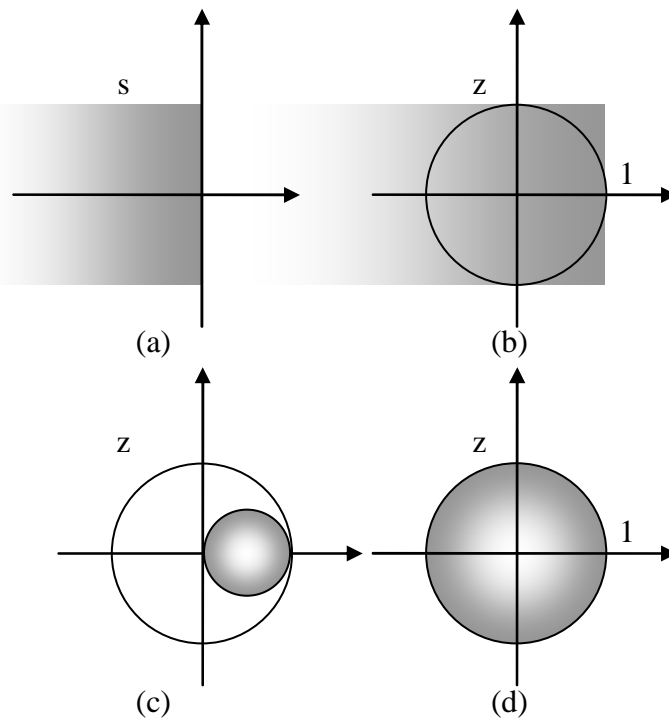


Figure 16 : correspondance des plans complexes. (a) : zone de stabilité dans le plan complexe continu en  $z$  ; (b) zone correspondante avec l'approximation d'Euler progressif ; (c) zone correspondante avec l'approximation d'Euler rétrograde ; (d) zone correspondante avec l'approximation bilinéaire.

La fonction `c2d` de Matlab permet d'obtenir l'équivalent discret d'une fonction de transfert continue avec la méthode d'intégration comme argument :

```

Matlab
sysc=tf([1 0],[1 2 5]);
Te=0.1;
% Méthode d'euler :
sysd1=c2d(sysc,Te,'zoh');
% Méthode des trapèzes :
sysd2=c2d(sysc,Te,'foh');
% Méthode bilinéaire :
sysd3=c2d(sysc,Te,'tustin');
    
```

#### 4.5.2 Superposition des réponses indicielles

Une autre méthode pour approximer une fonction de transfert discrète est de s'assurer que la réponse indicelle de sa version discrétisée correspond exactement lors des temps d'échantillonnage.

### 4.5.3 Correspondances pôles - zéros

La dernière méthode est de partir d'une décomposition de la fonction de transfert continue en pôles et zéros et d'effectuer la conversion pôles à pôles et zéros par zéros. La correspondance, selon que le pôle est réel ou complexe est donnée ci-dessous :

$$\begin{aligned} (s + a) &\rightarrow (z - e^{-a \cdot T_c}) \\ (s + a)^2 + b^2 &\rightarrow z^2 - 2 \cdot e^{-a \cdot T_c} \cdot \cos(b \cdot T_c) \cdot z + e^{-2 \cdot a \cdot T_c} \end{aligned} \quad (29)$$

### 4.5.4 Bilan

Aucune méthode n'est parfaite et on n'aura jamais en même temps :

- même réponse indicielle
- même réponse impulsionnelle
- approximation (Euler, Tustin, etc...) de l'équation différentielle
- correspondance des pôles et des zéros
- correspondance des réponses fréquentielles

Ceci étant dit le choix de la période d'échantillonnage est primordial. Si la fréquence d'échantillonnage est beaucoup plus faible que la dynamique de la fonction de transfert à approximer la différence entre chaque méthode sera minimale.



## 5. IDENTIFICATION DES SYSTEMES ECHANTILLONES

Dans le présent chapitre on donne les bases des méthodes d'identification. Pour une approche complète, voir le cours de troisième année "Identification avion" [6] et les ouvrages [1] et [7].

Même si l'on dispose d'un modèle théorique du processus que l'on souhaite commander il est presque toujours nécessaire de l'affiner en procédant à des essais sur le système réel. Bien souvent le seul modèle dont l'on dispose est ce modèle issu de l'expérimentation. Différentes méthodes s'appliquent selon l'objectif recherché et le niveau de précision exigé. On distinguera :

- les modèles paramétriques
- les modèles simples
- les modèles non paramétriques

### 5.1 Identification d'un modèle ARX par la méthode des moindres carrés récursifs

A partir de cet exemple, le plus simple qui soit, on pose les bases de toutes les méthodes d'identification.

On considère un système d'entrée  $u(t)$  et de sortie  $y(t)$  représenté par l'équation de récurrence suivante :

$$y(t) + a_1 \cdot y(t-1) + a_2 \cdot y(t-2) + \dots + a_n \cdot y(t-n) = b_0 \cdot u(t) + b_1 \cdot u(t-1) + b_2 \cdot u(t-2) + \dots + b_m \cdot u(t-m) \quad (30)$$

L'objectif est, à partir d'essais et de mesures, d'identifier les meilleures valeurs possibles pour les coefficients  $a_i$  et  $b_i$ .

L'équation (30) peut être réécrite pour mettre en évidence la *prédiction* de la sortie  $y(t)$  à partir des mesures passées :

$$y(t) = -a_1 \cdot y(t-1) - a_2 \cdot y(t-2) - \dots - a_n \cdot y(t-n) + b_0 \cdot u(t) + b_1 \cdot u(t-1) + b_2 \cdot u(t-2) + \dots + b_m \cdot u(t-m) \quad (31)$$

Soit :

$$y(t) = \varphi^T(t) \cdot \theta(t)$$

avec :

$$\begin{cases} \varphi^T(t) = [-y(t-1) \ \dots \ -y(t-n) \ , \ u(t-1) \ \dots \ u(t-m)] \\ \theta(t) = [a_1 \ \dots \ a_n \ , \ b_1 \ \dots \ b_m] \end{cases} \quad (32)$$

Le but est de trouver le meilleur ensemble de paramètres  $\theta(t)$  : celui qui, à partir d'un ensemble de mesures  $\varphi(t)$  permet de prédire avec la meilleure précision la sortie effectivement mesurée.

Supposons qu'à la suite d'une expérience on dispose d'un ensemble de mesures  $\{y(t), u(t)\}_{t=1:N}$  et que l'ensemble des paramètres  $\theta$  n'est pas bien connu. On construit la séquence des *sorties prédites* :

$$\hat{y}(t) = \varphi^T(t) \cdot \theta(t) \quad (33)$$

Le paramètre  $\theta$  est celui qui minimise une norme quadratique :

$$\theta = \arg \max \left( \frac{1}{N} \sum_{t=1}^N (y(t) - \hat{y}(t))^2 \right) = \arg \max \left( \frac{1}{N} \sum_{t=1}^N (y(t) - \varphi^T(t) \cdot \theta(t))^2 \right) \quad (34)$$

Compte tenu du fait que l'équation est linéaire au départ et que le terme à minimiser est du type quadratique la solution est simplement obtenue par une pseudo-inverse :

$$\theta = \left[ \sum_{t=1}^N (\varphi(t) \cdot \varphi^T(t)) \right]^{-1} \sum_{t=1}^N (\varphi(t) \cdot y(t)) \quad (35)$$

On obtient donc la solution classique des moindres carrés. La formule (35) est le plus souvent utilisée sous sa forme récursive :

$$\theta(t) = \theta(t-1) + F(t) \cdot \varphi(t) \cdot (y(t) - \theta^T(t-1) \cdot \varphi(t))$$

avec

$$F(t)^{-1} = F(t-1)^{-1} + \varphi(t) \cdot \varphi(t)^T \quad (36)$$

ou

$$F(t) = F(t-1) - \frac{F(t-1) \cdot \varphi(t) \cdot \varphi(t)^T \cdot F(t-1)}{1 + \varphi(t)^T \cdot F(t-1) \cdot \varphi(t)}$$

(On laisse au lecteur le plaisir d'effectuer lui-même les démonstrations, qu'il peut du reste trouver dans [1] et [7]).



## 5.2 L'identification paramétrique en général et en pratique

L'exemple précédent a permis de mettre en évidence tous les aspects de la méthodologie classique de l'identification :

- il a fallu choisir un ensemble de mesures  $\{y(t), u(t)\}_{t=1:N}$ . Le choix du signal d'excitation  $u(t)$  est en pratique primordial,
- il a fallu choisir une structure pour le modèle à identifier (ici les ordres  $n$  et  $m$ ),
- il a fallu choisir un critère (ici un critère quadratique sur l'erreur entre la sortie et la sortie prédite),
- il a fallu choisir un algorithme et le paramétrer (ici l'algorithme des moindres carrés récursifs qu'il a fallu initialiser)

Enfin un point non traité dans le chapitre précédent concerne la validation :

- il aurait fallu déterminer une méthode pour comparer différents modèles et choisir le meilleur

### 5.2.1 Choix du signal d'excitation, traitement des mesures

De manière générale le système que l'on doit identifier doit être excité dans tout le domaine fréquentiel qui nous intéresse. Le signal classiquement utilisé est une séquence pseudo binaire aléatoire (Pseudo Random Binary Signal) tel celui représenté sur la Figure 17.

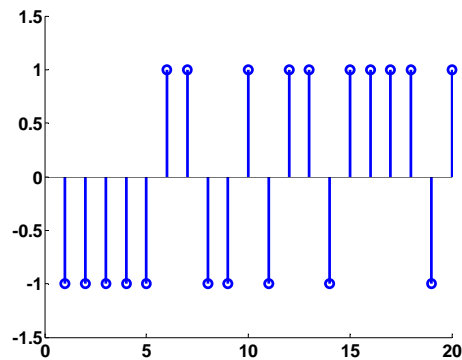


Figure 17 : exemple d'une séquence PRBS

Le signal d'excitation doit se superposer au signal de commande correspondant au point de fonctionnement étudié. Son amplitude doit être faible par rapport à l'amplitude des commandes du système (quelques %).

Une fois le système excité et les mesures effectuées il faut supprimer les valeurs moyennes des mesures.

### 5.2.2 Choix d'une structure de modèle

Le modèle étudié dans le chapitre précédent est de type ARX. On peut le noter sous forme polynomiale :

$$A \cdot y = B \cdot u \quad (37)$$

Ce modèle correspond en fait à un parti-pris sur la manière dont on traite les perturbations. En réalité on prend en compte un signal d'erreurs possibles  $e$  (de moyenne nulle) tels que :

$$A \cdot y = B \cdot u + e \quad (38)$$

Si l'on fait cette hypothèse de modélisation alors la méthode optimale est celle des moindres carrés. Malheureusement cette manière de prendre en compte les perturbations est bien restrictive. D'autres modélisations sont possibles, adaptées à d'autres algorithmes, par exemple le modèle OE :

$$y = \frac{B}{F} \cdot u + e \quad (39)$$

Il est aussi possible de "structurer" le modèle de bruit comme par exemple dans les modèles ARMAX ou BJ :

$$\begin{aligned} A \cdot y &= B \cdot u + C \cdot e \\ y &= \frac{B}{F} \cdot u + \frac{C}{D} \cdot e \end{aligned} \quad (40)$$

La structure du modèle étant choisie il faut choisir l'ordre de ses différents polynômes.

### 5.2.3 Choix d'un critère à minimiser

Dans la plupart des cas le critère à minimiser est l'erreur quadratique entre la sortie et la sortie prédite à un coup (one step ahead prediction). Ceci n'est bien évidemment pas restrictif et peut être généralisé à la prédiction à  $N$  coups.

### 5.2.4 Choix d'un algorithme

Même si chaque structure de modèle est généralement associée à un algorithme "optimal" celui-ci doit généralement se paramétrer.

Prenons par exemple l'algorithme des moindres carrés récursifs tel que présenté dans (36). Cette équation peut être modifiée en intégrant un "facteur d'oubli"  $\lambda$  :

$$\begin{aligned} \theta(t) &= \theta(t-1) + F(t) \cdot \varphi(t) \cdot (y(t) - \theta^T(t-1) \cdot \varphi(t)) \\ \text{avec} & \\ F(t)^{-1} &= \lambda \cdot F(t-1)^{-1} + \varphi(t) \cdot \varphi(t)^T \end{aligned} \quad (41)$$

Ce paramètre (typiquement entre 0.95 et 0.99) permet, comme son nom l'indique, "d'oublier" les anciennes mesures et de se concentrer sur les mesures les plus récentes.

### 5.2.5 Validation

Une fois l'algorithme d'identification appliqué et le modèle obtenu (ce qui prend généralement quelques secondes avec un ordinateur moderne) il reste à expertiser celui-ci. Selon la "qualité" du modèle obtenu on remet en cause l'une ou l'autre des étapes (signal d'excitation, structure du modèle, algorithme) et on réitère jusqu'à obtenir un modèle satisfaisant.

En pratique on essaie un certain nombre de modèles avant de choisir le "meilleur".

Le premier critère de qualité est la capacité du modèle à prédire effectivement la sortie (sortie prédite à un coup, sortie prédite à N coups). Il faut pour cela utiliser un jeu de mesures non utilisé pour l'identification. (En pratique le jeu de mesures initiales est coupé en deux : une partie pour l'identification et une partie pour la validation).

Le second critère de qualité est l'analyse du résidu : la composition spectrale de l'erreur de prédiction.

### 5.2.6 Utilisation de logiciels

En pratique il est commode d'utiliser un célèbre outil de calcul scientifique et sa toolboxe d'identification.

Comme les calculs proprement dit ne prennent que quelques secondes il est facile de réaliser un grand nombre d'identifications et de stocker les différents modèles pour les comparer ensuite. On est rapidement noyé sous le grand nombre de modèles possibles...

## 5.3 Identification "simple"

Il est souvent suffisant d'obtenir un modèle "simple" de type premier ordre et un simple essai indiciel convient souvent. Le système est excité avec un échelon unité et on cherche le meilleur modèle du premier ordre correspondant.

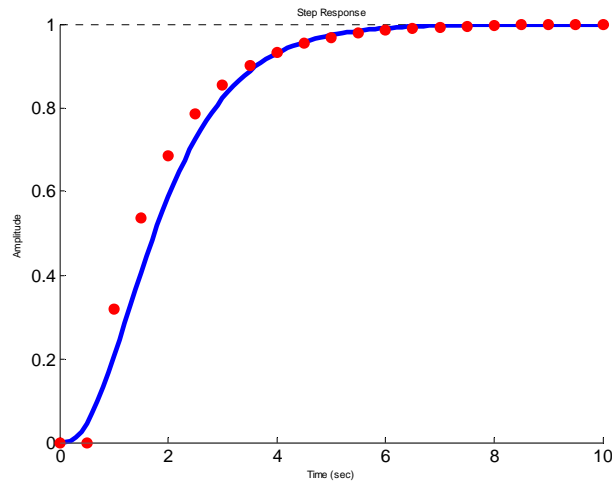


Figure 18 : identification sur un essai indiciel

Dans l'exemple de la Figure 18 on superpose le résultat d'une mesure (en bleu) avec la simulation d'un modèle discret avec une période de 0.5s. Le modèle cherché est du premier ordre avec un retard pur  $d$  :

$$y(z) = \frac{(1 + c_1) \cdot z^{-d}}{1 + c_1 \cdot z^{-1}} \cdot u(z) \quad (42)$$

Il s'agit de trouver l'entier  $d$  et le nombre réel  $c_1$  tels que la réponse indicelle du modèle corresponde pour le mieux aux mesures issues de l'essai. Ceci peut se formaliser par exemple de la manière suivante :

$$(c_1 \quad d_1) = \arg \min \sum_{t=1}^N (y_{\text{modèle}}(t) - y_{\text{mesure}}(t))^2 \quad (43)$$

Cette optimisation peut se faire à la main ou grâce à une procédure automatisée (avec le célèbre outil de calcul scientifique).

Dans le cas où le système comprend naturellement une intégration (par exemple la commande d'un moteur à courant continu par une tension et la mesure de l'angle de rotation) il est utile de prendre en compte explicitement cette intégration dans le modèle :

$$y(z) = \frac{1}{1 - z^{-1}} \cdot \frac{(1 + c_1) \cdot z^{-d}}{1 + c_1 \cdot z^{-1}} \cdot u(z) \quad (44)$$

La procédure est la même mais au lieu de prendre en compte la sortie  $y(t)$  on prend en compte la dérivée  $y'(t)$  :

$$y'(t) = y(t) - y(t-1) \quad (45)$$

## 5.4 Identification non paramétrique

Pour terminer ce panorama rapide sur les méthodes d'identification il faut parler des méthodes non paramétriques : celles qui permettent de tracer la réponse impulsionnelle ou la réponse fréquentielle du système.

### 5.4.1 Estimation de la réponse impulsionnelle

La réponse impulsionnelle d'un système peut être obtenue simplement en appliquant une entrée de type "impulsion" au système et d'en mesurer les conséquences.

La seconde méthode consiste à utiliser la méthode des moindres carrés (ou une méthode dérivée) comme vue au chapitre 5.1. La réponse impulsionnelle d'un système peut en effet être vue comme la fonction de transfert particulière où le dénominateur est de degré nul et le numérateur comporte un nombre infini (en pratique : grand) d'éléments ( $n = 0$  et  $m$  grand) :

$$y(t) = b_0 \cdot u(t) + b_1 \cdot u(t-1) + b_2 \cdot u(t-2) + \dots + b_m \cdot u(t-m) \quad (46)$$

La dernière méthode, qui sera développée ici est l'analyse d'intercorrélation.

Par définition l'intercorrélation  $R_{yu}$  entre deux signaux est :

$$R_{yu}(\tau) = E[(y(t) - E(y(t))) \cdot (u(t-\tau) - E(u(t-\tau)))] \quad (47)$$

Dans cette formule l'espérance mathématique peut être approchée grâce à la loi des grands nombres :

$$R_{yu}(\tau) \approx R_{yu}^N(\tau) = \lim_{N \rightarrow \infty} \frac{1}{N} \sum_{t=1}^N [(y(t) - m_y) \cdot (u(t-\tau) - m_u)] \quad (48)$$

On s'intéresse maintenant à un système tel que décrit par l'équation impulsionnelle (46) et on suppose que le signal d'entrée est un bruit blanc. Le signal d'entrée vérifie donc :

$$\begin{cases} \tau \neq 0 & R_{uu}(\tau) = 0 \\ \tau = 0 & R_{uu}(\tau) = R \end{cases} \quad (49)$$

Il vient alors naturellement :

$$b_k \approx \frac{1}{R} R_{yu}(k) \quad (50)$$

### 5.4.2 Estimation de la réponse fréquentielle

On utilise ici le spectre d'intercorrélation. Par définition le spectre d'intercorrélation entre deux signaux est :

$$\Phi_{yu}(\omega) = \sum_{\tau=-\infty}^{\infty} R_{yu}(\tau) \cdot e^{-i\omega\tau} \quad (51)$$

On suppose que le signal d'entrée  $u(t)$  et le signal de sortie  $y(t)$  sont reliés par une fonction de transfert linéaire du type :

$$y(t) = F \cdot u(t) + v(t) \quad (52)$$

où  $v(t)$  est une perturbation indépendante de  $u(t)$ . On a alors les relations :

$$\Phi_{yu}(\omega) = F(j\omega) \cdot \Phi_{uu}(\omega) \quad (53)$$

Il reste à estimer les spectres d'intercorrélation pour remonter à la réponse fréquentielle  $F(j\omega)$  recherchée. Une idée naturelle est d'utiliser la relation (48) mais cela marche très mal dans presque tous les cas, même en choisissant  $N$  assez grand. L'amélioration est d'utiliser une "fenêtre" pour favoriser les valeurs faibles de  $\tau$  par rapport aux valeurs élevées. La solution classique utilisant la *fenêtre de Hamming* donne :

$$\Phi_{yu}^N(\omega) = \sum_{\tau=-\gamma}^{\gamma} R_{yu}^N(\tau) \cdot w_{\gamma}(\tau) \cdot e^{-i\omega\tau} \quad (54)$$

avec

$$\begin{cases} w_{\gamma}(\tau) = \frac{1}{2} \left( 1 + \cos\left(\frac{\pi \cdot \tau}{\gamma}\right) \right) & |\tau| < \gamma \\ w_{\gamma}(\tau) = 0 & |\tau| > \gamma \end{cases}$$

On utilise ensuite la relation (51) pour estimer la réponse fréquentielle :

$$F(j \cdot \omega) = \frac{\Phi_u^N(\omega)}{\Phi_{yu}^N(\omega)} \quad (55)$$





## 6. SYSTEMES BOUCLES

### 6.1 Fonction de transfert en boucle fermée

Un système et sa commande par ordinateur peut être représenté par un ensemble de fonctions de transfert formant un schéma-bloc plus ou moins compliqué (Figure 19).

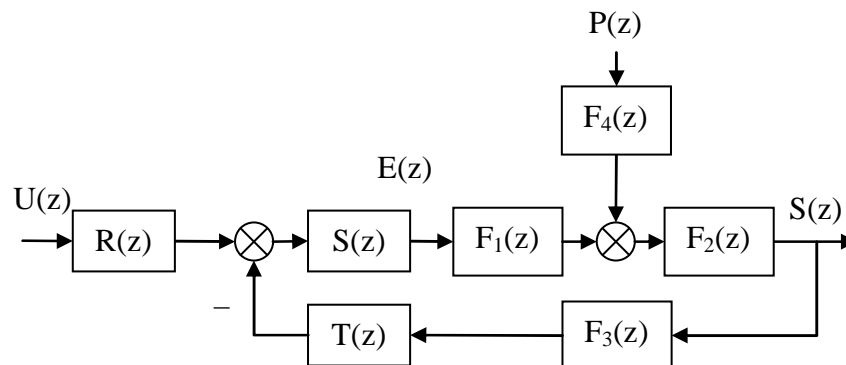


Figure 19 : système discret commandé par ordinateur. Le correcteur est représenté par les fonctions R,S,T, le système par  $F_1$  et  $F_2$ , le modèle de perturbations par  $F_4$  et le modèle des actionneurs par  $F_4$ .

Quelle que soit la structure d'un système asservi il est possible de se ramener à une forme canonique comprenant une fonction de transfert en boucle ouverte en combinant judicieusement les fonctions de transfert du procédé, du régulateur et des perturbations. La Figure 20 représente par exemple le système de la Figure 19 où  $F$  est une combinaison des fonctions de transfert du système, du régulateur, des perturbations et des capteurs.

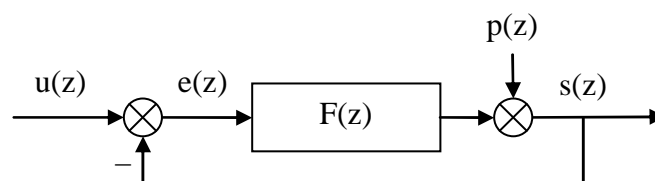


Figure 20 : système discret, notations.

Par défaut on considère une contre-réaction négative et une perturbation additive sur la sortie. La fonction de transfert  $F$  est appelée *fonction de transfert en boucle ouverte*. Comme pour les systèmes continus la *fonction de transfert en boucle fermée* est obtenue simplement à partir de  $F$  (58).

$$\begin{aligned}
 F(z) &= \frac{s(z)}{e(z)} = \frac{B(z^{-1})}{A(z^{-1})} \\
 G(z) &= \frac{s(z)}{u(z)} = \frac{B(z^{-1})}{A(z^{-1}) + B(z^{-1})}
 \end{aligned}
 \tag{56}$$

## 6.2 Erreur statique

L'erreur statique du système en boucle fermée, comme pour le système en boucle ouverte, est simplement le rapport des coefficients.

$$G_0 = \frac{\sum_{i=0}^{i=m} b_i}{1 + \sum_{i=0}^{i=m} b_i + \sum_{i=1}^{i=n} a_i}
 \tag{57}$$

Si l'on souhaite que l'asservissement confère un gain statique unitaire au système en boucle fermée (c'est-à-dire que l'on force la sortie  $s(k)$  à tendre vers la valeur de consigne  $u(k)$ ) il faut que ce rapport de coefficients soit égal à un. On montre facilement que cela est obtenu si la fonction de transfert  $F(z)$  comporte un intégrateur :

$$F(z) = \frac{s(z)}{e(z)} = \frac{1}{1 - z^{-1}} \cdot \frac{B(z)}{A'(z)}
 \tag{58}$$

## 6.3 Rejet de perturbations

La fonction de transfert entre la perturbation  $p(z)$  et la sortie  $y(z)$  est appelée fonction de sensibilité perturbation-sortie, notée  $S_{yp}$ .

$$S_{yp}(z) = \frac{s(z)}{p(z)} = \frac{A(z^{-1})}{A(z^{-1}) + B(z^{-1})}
 \tag{59}$$

Si l'on souhaite qu'une perturbation de type échelon soit parfaitement rejetée il faut que  $A(1) = 0$ , on retrouve donc la condition (58) à savoir que le système doit comporter une intégration dans la boucle directe. On peut être en outre être amené à spécifier que les perturbations ne soient pas amplifiées dans une certaine bande de fréquence. On rajoute donc une contrainte du type  $|S_{yp}(e^{j2\pi f})| < S_{max}$  pour  $f$  appartenant à un certain domaine de  $[0 \ 0.5]$ .

## 6.4 Stabilité des systèmes à simple gain de rétroaction

Les critères de Routh et de Jury (donnés dans le chapitre 3.2.7 p13) ou le calcul numérique des pôles permettent de déterminer la stabilité du système en boucle fermée si l'on connaît sa fonction de transfert.

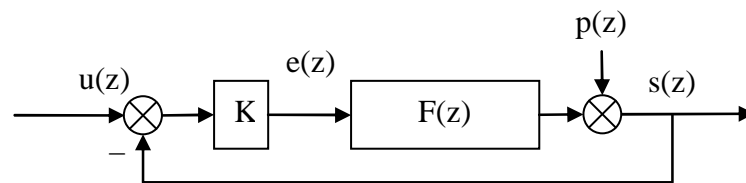


Figure 21 : système bouclé par un simple gain.

On s'intéresse ici au cas particulier où l'asservissement est réalisé par un simple gain de rétroaction  $K$  (Figure 21). On cherche la valeur de  $K$  qui amène le système en limite de stabilité. L'outil adapté ici est le lieu d'Evans ou lieu des racines (root locus) qui permet de visualiser l'évolution des pôles du système bouclé lorsque le gain  $K$  varie.

Prenons l'exemple d'un système comportant trois pôles et deux zéros. L'exemple est donné sous forme de script Matlab :

### Matlab

```
% Définition d'un système discret :
sys=zpk([-2.037 -0.1406],[0.6065 0.3679 0.3679],1,1)
%Tracé du lieu des racines
rlocus(sys);
```

Le script précédent produit le tracé de la Figure 22. On pourrait y lire que la limite de stabilité est obtenue pour  $K=0.2$  environ (détail de la Figure 23).

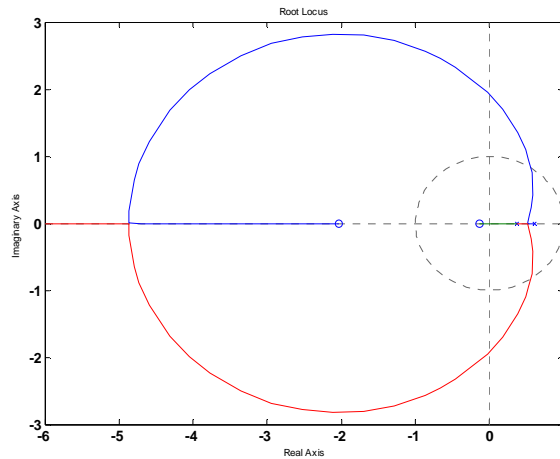


Figure 22 : lieu des racines

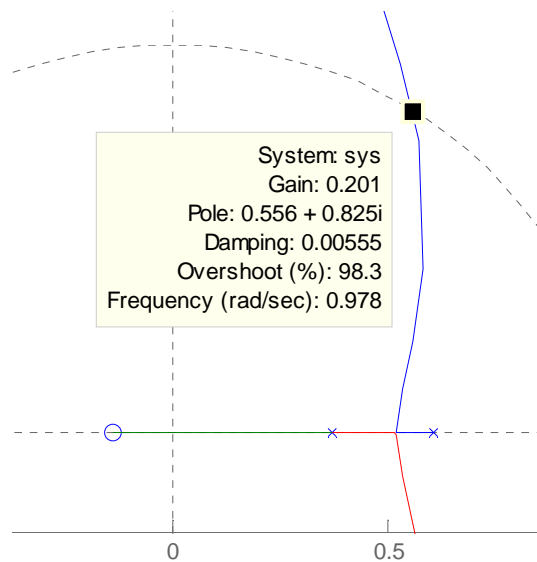


Figure 23 : lieu des racines : détail de la limite de stabilité

On voit sur la Figure 22 que le lieu des racines d'un système échantillonné ressemble à celui d'un système continu :

- on commence par tracer les pôles (×) et les zéros (o) du système en boucle ouverte.
- les pôles du système en boucle fermée décrivent un lieu qui part des pôles (pour  $K = 0$ ) et arrive aux zéros (pour  $K = \infty$ )
- lorsque il y a  $n-m$  plus de pôles que de zéros le système comporte  $n-m$  branches asymptotiques.

On ne décrira pas ici les diverses propriétés du lieu des racines qui permettent de le tracer à la main puisque il est facile à obtenir avec un ordinateur.

Il peut aussi être utilisé pour régler "à la main" le gain permettant d'obtenir un ou plusieurs modes dominants ayant certaines propriétés de fréquence et d'amortissement. On voit sur la Figure 24 que le gain 0.017 permet d'obtenir un pôle double d'amortissement 0.706 et de fréquence 0.65 rad/s et un pôle simple de fréquence 1.63 rad/s.

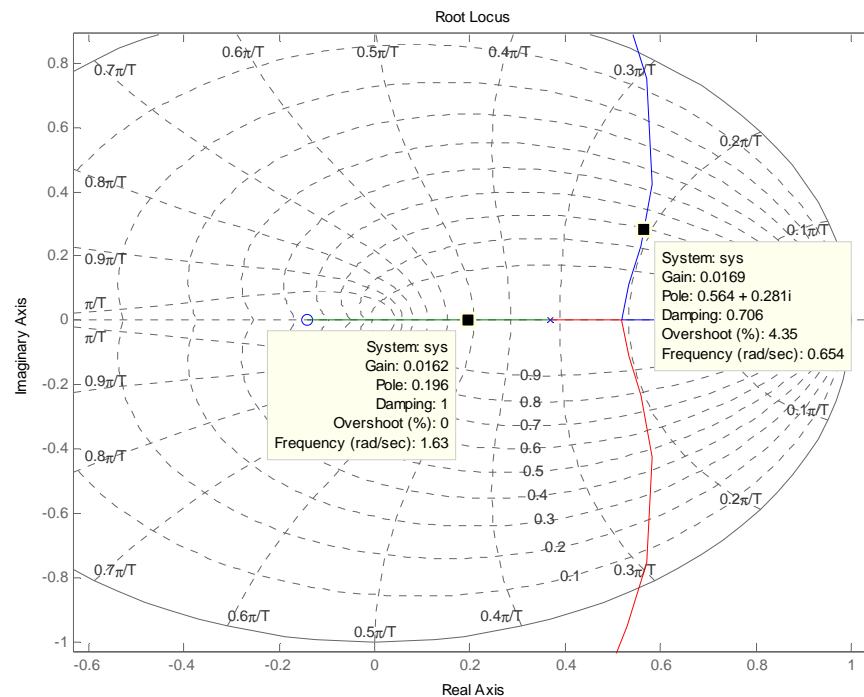


Figure 24 : utilisation du lieu des racines pour régler le gain de la boucle fermée

L'outil rltool de Matlab permet de nombreuses manipulations graphiques de ce type.

## 6.5 Analyse des systèmes bouclés dans le domaine fréquentiel : stabilité

Comme dans le cas continu il est possible de prévoir des propriétés, en particulier la stabilité, d'un système en boucle fermée à partir de l'examen de la fonction de transfert en boucle ouverte.

On ne redémontrera pas ici le théorème de Nyquist mais on se contente de rappeler l'énoncé.

On rappelle que le *lieu de Nyquist* est le lieu de  $F(j2\pi f)$  dans le plan complexe pour  $f$  variant de 0 à  $+\infty$ . A noter que le lieu pour la zone  $[0, 0.5]$  est symétrique de celui pour la zone  $[0.5, 1]$ . Pour cette raison on ne représente généralement que le lieu pour  $f$  appartenant à  $[0, 0.5]$ .

### 6.5.1 Théorème de Nyquist dans le cas où le système est stable en boucle ouverte (critère du revers)

Le système est stable en boucle fermée si le lieu de Nyquist de la fonction de transfert en boucle ouverte parcouru dans le sens des fréquences croissantes de 0 à 0.5 laisse le point critique  $(-1,0)$  à gauche.

La Figure 25 représente par exemple le lieu de Nyquist d'une fonction de transfert en boucle ouverte qui sera stable en boucle fermée.

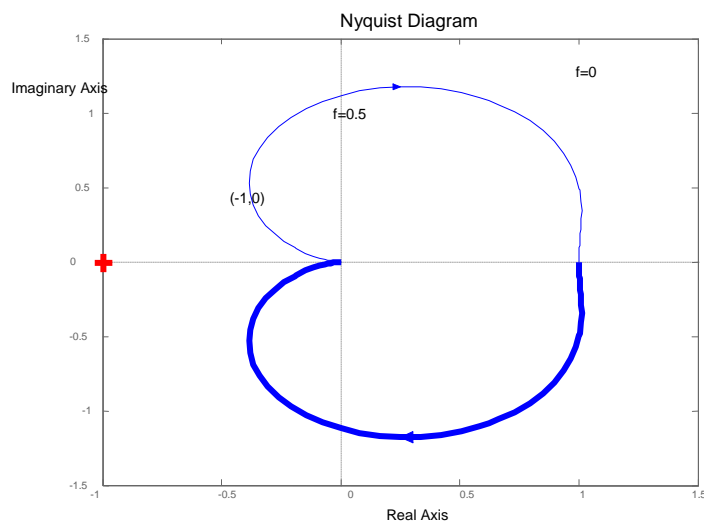


Figure 25 : lieu de Nyquist d'un système stable

### 6.5.2 Théorème de Nyquist dans le cas où le système est instable en boucle ouverte

Si le système en boucle ouverte comporte  $N$  pôles instables il sera stable en boucle fermée si le lieu de Nyquist de la fonction de transfert en boucle ouverte parcouru dans le sens des fréquences croissantes de  $-0.5$  à  $0.5$  entoure le point critique  $(-1,0)$   $N$  fois dans le sens des aiguilles d'une montre..

### 6.5.3 Interprétations du lieu de Nyquist

Comme dans le cas continu, la distance à laquelle on passe du point critique renseigne sur la robustesse du système bouclé. La distance sera exprimée de différente manière (Figure 26) :

- marge de module  $\Delta M$  : plus petite distance entre le lieu des racines et le point critique,

- marge de gain  $\Delta G$
- marge de phase  $\Delta P$

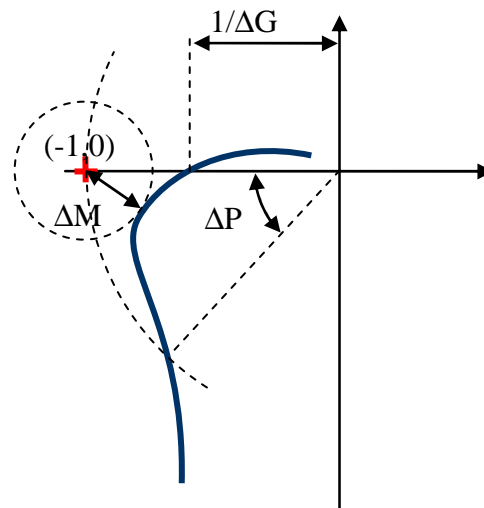


Figure 26 : Marges de stabilité

Les valeurs typiques de ces marges sont données ci-après :

- $\Delta G \geq 6\text{dB}$
- $\Delta M \geq -6\text{dB}$
- $30^\circ \leq \Delta P \leq 60^\circ$

La distance entre le point critique et le lieu des racines est la norme de  $1+F(j2\pi f)$ . On peut donc exprimer la marge de module en fonction de la fonction de sensibilité :

$$\begin{aligned} \Delta M &= \min |1 + F(j \cdot \omega)|_{\omega \in [0, \pi]} \\ \Delta M &= \min \left| 1 + \frac{B(j \cdot \omega)}{A(j \cdot \omega)} \right|_{\omega \in [0, \pi]} = \max \left| \frac{A(j \cdot \omega)}{A(j \cdot \omega) + B(j \cdot \omega)} \right|_{\omega \in [0, \pi]} \quad (60) \\ \Delta M &= \max |S_{yp}(j \cdot \omega)|_{\omega \in [0, \pi]}^{-1} \\ \Delta M_{\text{dB}} &= -\max |S_{yp}(j \cdot \omega)|_{\text{dB}} \end{aligned}$$

Si l'on trace l'évolution de  $S_{yp}$  en fonction de  $f$  on obtient donc la valeur de la marge de module.





# 7. METHODES DE REGULATION DES SYSTEMES ECHANTILLONES

## 7.1 Introduction

Les bases théoriques permettant de concevoir le correcteur pour un système échantillonné ont été posées dans les chapitres précédents. Ce chapitre fait le bilan des connaissances acquises et donne quelques solutions pour calculer des correcteurs à partir d'un cahier des charges précis.

Les systèmes que l'on cherche à commander sont généralement, par nature, continus. La réalisation d'un correcteur par un ordinateur implique que l'on génère une équation de commande discrète (généralement une équation récurrente). L'obtention de cette équation de commande peut se faire de deux manières :

- la première méthode consiste à d'abord synthétiser un correcteur continu (en utilisant les outils classiques de l'automatique et la simulation) et discrétiser ensuite ce correcteur. C'est l'objet du chapitre 7.2.2.
- la seconde méthode consiste à tenir la représentation discrète du système continu associé à son interface numérique (convertisseurs analogique-numérique et numérique-analogique) comme expliqué dans le chapitre 4.5 puis synthétiser directement le correcteur discret. Cela sera expliqué dans le chapitre 7.3 et suivants.

Ce choix peut être résumé par la figure suivante :

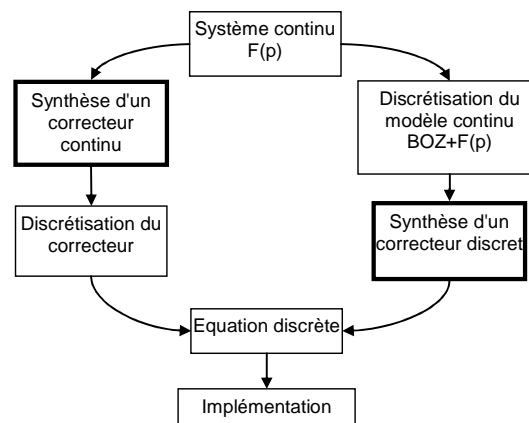


Figure 27 : Réalisation d'un correcteur discret.

Au-delà des affaires de goût, le choix de la méthode sera souvent dicté par l'application. En effet, la première méthode est d'autant meilleure que le correcteur discrétisé approche correctement le correcteur continu. Cela implique que le temps d'échantillonnage soit très faible par rapport aux constantes de temps du système

bouclé. Dans le cas où les contraintes temps réel sont fortes (capacités de calcul partagées par exemple) il sera préférable de synthétiser un correcteur discret.

## 7.2 Structure d'un système commandé par ordinateur, hypothèses

Par la suite on se ramènera toujours à un système tel que celui représenté sur la Figure 28 en cherchant un correcteur  $K(z)$  linéaire. On montre que quel que soit le correcteur linéaire on peut se ramener à la forme représenté sur la Figure 29 (forme RST).

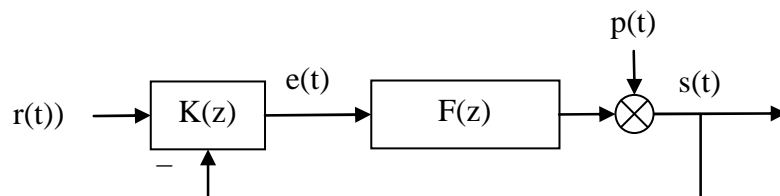


Figure 28 : système bouclé, forme générale.

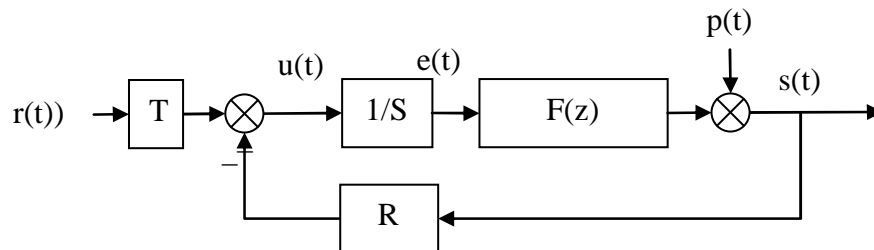


Figure 29 : système bouclé, forme canonique.

### 7.2.1 Discrétisation, choix de la période d'échantillonnage

Du point de vue du ordinateur, l'extérieur (le procédé) est représenté par la fonction de transfert  $F(z)$ . L'hypothèse importante est que la commande (sortie du ordinateur) et la mesure (entrée du ordinateur) sont échantillonnées de manière synchrone à la même période  $T_e$ .

Le choix judicieux de la fréquence d'échantillonnage est très important. En pratique on choisit cette fréquence en fonction de la bande passante du système en **boucle fermée** :

$$f_e = (6 \text{ à } 25) f^{BF} \quad (61)$$

### 7.2.2 Discrétisation d'un correcteur continu

La première méthode pour obtenir les équations d'un correcteur numérique est de faire sa synthèse avec les outils de l'automatique continue puis de discrétiser la fonction de transfert du correcteur continu obtenu (voir pour cela le chapitre 4.2 *Echantillonnage des systèmes continus*).

On va voir par la suite que la synthèse directe d'un correcteur discret offre plus de degré de liberté et permet en général de remplir un cahier des charges plus complet.

## 7.3 Régulateur PID numérique

Le régulateur le plus populaire est le régulateur PID. Nous allons voir dans un premier temps comment on génère un régulateur PID à partir des équations bien connues du régulateur PID continu et comment ces équations peuvent être généralisées pour obtenir plus de degrés de liberté pour le réglage.

### 7.3.1 Cas 1

Un régulateur PID continu a une fonction de transfert du type :

$$K_{\text{PID}}(s) = K \cdot \left( 1 + \frac{1}{T_i \cdot s} + \frac{T_d \cdot s}{1 + \frac{T_d}{N} \cdot s} \right) \quad (62)$$

La discrétisation de ce correcteur continu par la méthode d'Euler (Tableau 4 page 31) se fait en remplaçant la variable de Laplace  $s$  par le terme  $\frac{1-q^{-1}}{q^{-1} \cdot T_e}$ . On obtient :

$$K_{\text{PID}}(q) = K \cdot \left( 1 + \frac{1}{T_i \cdot \frac{1-q^{-1}}{T_e}} + \frac{T_d \cdot \frac{1-q^{-1}}{T_e}}{1 + \frac{T_d}{N} \cdot \frac{1-q^{-1}}{T_e}} \right) \quad (63)$$

En réarrangeant on obtient la forme générale RST (Figure 30):

$$K_{PID}(q) = K \cdot \frac{r_0 + r_1 \cdot q^{-1} + r_2 \cdot q^{-2}}{(1 - q^{-1}) \cdot (1 + s_1' \cdot q^{-1})}$$

$$\begin{cases} r_0 = K \cdot \left( 1 + \frac{T_e}{T_i} + \frac{N \cdot T_d}{T_d + N \cdot T_e} \right) \\ r_1 = -K \cdot \left( 1 + \frac{T_d}{T_d + N \cdot T_e} + \frac{T_e}{T_i} \cdot \frac{T_d}{T_d + N \cdot T_e} + \frac{N \cdot T_d}{T_d + N \cdot T_e} \right) \\ r_2 = K \cdot \left( \frac{T_d}{T_d + N \cdot T_e} \right) \\ s_1' = \frac{T_d}{T_d + N \cdot T_e} \end{cases} \quad (64)$$

Comme on le voit le correcteur obtenu comporte autant de degrés de liberté que le correcteur PID analogique. On reconnaît au dénominateur le terme  $(1 - q^{-1})$  en facteur qui assure l'effet d'intégration.

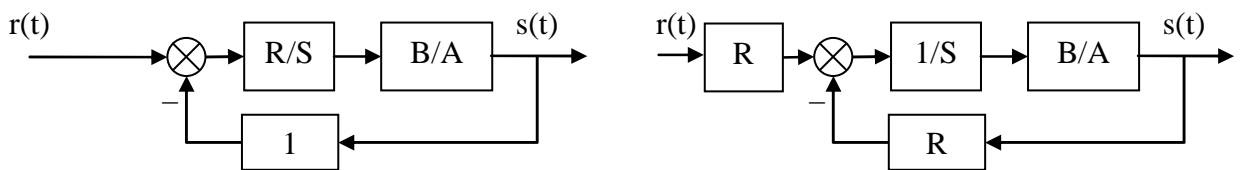


Figure 30 : PID sous forme RST

La fonction de transfert en boucle fermée obtenue est donnée par l'équation suivante :

$$H_{BF} = \frac{B \cdot R}{A \cdot S + B \cdot R} = \frac{B \cdot R}{P} \quad (65)$$

De manière générale le polynôme caractéristique  $P$  fixe les performances visées en boucle fermée. Le calcul de  $R$  et  $S$ ,  $P$  étant fixé et  $A$  et  $B$  étant connus, est réalisé en résolvant l'équation de Bézout  $P = A \cdot S + B \cdot R$ . (voir 9.1 Annexe 1 : Résolution de l'équation polynomiale de Bezout)

On voit que le numérateur  $B \cdot R$  est imposé une fois que  $R$  et  $S$  sont trouvés. Le régulateur introduit donc naturellement des zéros dans la boucle fermée.

Traisons l'exemple entièrement dans le cas où le système est modélisé comme un système du deuxième ordre avec faible retard. La fonction de transfert du procédé est donnée par :

$$H_{BO} = \frac{B}{A} = \frac{b_1 \cdot q^{-1} + b_2 \cdot q^{-2}}{1 + a_1 \cdot q^{-1} + a_2 \cdot q^{-2}} \quad (66)$$

Le correcteur PID rajoutant deux pôles, le polynôme caractéristique en boucle fermée spécifié doit posséder quatre pôles. Le polynôme caractéristique en boucle fermée doit donc être de la forme :

$$\begin{aligned} P &= 1 + p_1 \cdot q^{-1} + p_2 \cdot q^{-2} + p_2 \cdot q^{-3} + p_2 \cdot q^{-4} \\ P &= (1 + p_1' \cdot q^{-1} + p_2' \cdot q^{-2}) \cdot (1 + \alpha_1 \cdot q^{-1}) \cdot (1 + \alpha_2 \cdot q^{-1}) \end{aligned} \quad (67)$$

Et il doit être identifié au produit  $AS + BR$  avec  $R$  et  $S$  ayant les formes données par (64) :

$$\begin{aligned} &(1 + p_1' \cdot q^{-1} + p_2' \cdot q^{-2}) \cdot (1 + \alpha_1 \cdot q^{-1}) \cdot (1 + \alpha_2 \cdot q^{-1}) \\ &= (1 + a_1 \cdot q^{-1} + a_2 \cdot q^{-2}) \cdot (1 - q^{-1}) \cdot (1 + s_1' \cdot q^{-1}) \\ &+ (b_1 \cdot q^{-1} + b_2 \cdot q^{-2}) \cdot (r_0 + r_1 \cdot q^{-1} + r_2 \cdot q^{-2}) \end{aligned} \quad (68)$$

Les coefficients  $p_1'$  et  $p_2'$  définissent un modèle du second ordre aux caractéristiques (pulsation propre  $\omega_0$  et amortissement  $\sigma$  choisis et coefficients déduits conformément au Tableau 3 page 25). Les coefficients  $\alpha_1$  et  $\alpha_2$  correspondent à des pôles auxiliaires aperiodiques. Les choix typiques pour ces coefficients sont :

$$-0.05 < \alpha_1, \alpha_2 < -0.5 \quad (69)$$

On montre que la résolution de (68) est réalisée par une inversion matricielle simple :

$$M \cdot x = p \quad (70)$$

avec

$$\begin{cases} x^T = [1 & s_1' & r_0 & r_1 & r_2] \\ p^T = [1 & p_1 & p_2 & p_3 & p_4] \end{cases}$$

et

$$M = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ a_1' & 1 & b_1 & 0 & 0 \\ a_2' & a_1' & b_2 & b_1 & 0 \\ a_3' & a_2' & 0 & b_2 & b_1 \\ 0 & a_3' & 0 & 0 & b_2 \end{pmatrix} \quad (71)$$

On montre que M est inversible à la condition que A et B soient premiers entre eux, c'est-à-dire que A et B n'aient pas de pôles et de zéros en commun.

(Le problème peut aussi être résolu en utilisant la résolution de l'équation de Bézout).

### 7.3.2 Cas 2

Comme on l'a vu le correcteur PI numérique directement transposé du correcteur PID analogique décrit dans le chapitre 7.3.1 a l'inconvénient de ramener des zéros au dénominateur de la fonction de transfert en boucle fermée (équation 65). Il est possible, grâce à la flexibilité du régulateur numérique d'imposer au correcteur de ne pas ajouter de zéros en boucle fermée. La fonction de transfert désirée en boucle fermée sera donc de la forme :

$$H_{BF} = \frac{P(1)}{B(1)} \cdot \frac{B}{P} \quad (72)$$

Le facteur  $P(1)/B(1)$  garantit le gain statique unitaire du système en boucle fermée (voir page 12). Si le régulateur a la forme générale RST alors les polynômes R S et T doivent vérifier :

$$H_{BF} = \frac{B \cdot T}{A \cdot S + B \cdot R} = \frac{P(1) B}{B(1) P} \quad (73)$$

S et R sont obtenus par la résolution de la même équation que précédemment (68). Le polynôme T est obtenu par simple identification :  $T = R(1)$ .

La structure, sous forme de schéma-bloc est donnée ci-dessous :

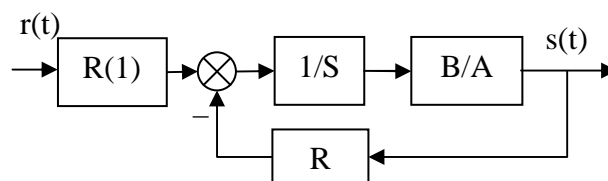


Figure 31 : PID2 sous forme RST

On montre que cette structure de PID, ramené en continu, correspond au PID canonique pour l'action intégrale. L'action proportionnelle et dérivée n'est appliquée que sur le signal y.

La structure du PID2 par rapport au PID1 a pour avantage de ne pas ramener de zéros à la boucle fermée. Cela se traduit par un dépassement moindre en transitoire lorsqu'on applique un échelon de commande. Par contre les deux systèmes réagissent de la même manière à un échelon de perturbation.

## 7.4 Placement de pôles

Les deux structures de PID décrites dans le chapitre précédent sont deux cas particulier d'une stratégie plus générale de placement de pôles.

La synthèse est plus générale que les PID présentés précédemment puisqu'il n'est pas nécessaire de faire d'hypothèses sur le degré de A et B, sur le retard, ni sur la présence ou non de zéros stables ou instables dans le procédé.

La structure du correcteur est la structure type RST telle que présentée dans la Figure 29. Pour la suite des calculs on posera pour la fonction de transfert les conventions suivantes :

$$H_{BO} = \frac{q^{-d} \cdot B(q^{-1})}{A(q^{-1})} \quad (74)$$

$$\begin{cases} A(q^{-1}) = 1 + a_1 \cdot q^{-1} + \dots + a_{n_A} \cdot q^{-n_A} \\ B(q^{-1}) = b_1 \cdot q^{-1} + \dots + b_{n_B} \cdot q^{-n_B} = q^{-1} \cdot B^*(q^{-1}) \end{cases}$$

Compte tenu de la forme RST du régulateur la fonction de transfert en boucle fermée vérifie :

$$H_{BF} = \frac{q^{-d} \cdot B(q^{-1}) \cdot T(q^{-1})}{P(q^{-1})} \quad (75)$$

$$P(q^{-1}) = A(q^{-1}) \cdot S(q^{-1}) + q^{-d} \cdot B(q^{-1}) \cdot R(q^{-1}) = 1 + p_1 \cdot q^{-1} + \dots + p_{n_p} \cdot q^{-n_p}$$

Le polynôme  $P(q^{-1})$  définit donc les pôles de la fonction de transfert en boucle fermée et traduit donc le comportement en *poursuite*.

De la même manière la fonction de transfert entre une perturbation et la sortie (fonction de sensibilité perturbation-sortie) est donnée par :

$$S_{yp} = \frac{A(q^{-1}) \cdot S(q^{-1})}{P(q^{-1})} \quad (76)$$

Ce même polynôme traduit donc aussi le comportement en *régulation*.

Les consignes en poursuite et en régulation peuvent être différentes comme cela est représenté par exemple sur la Figure 32.

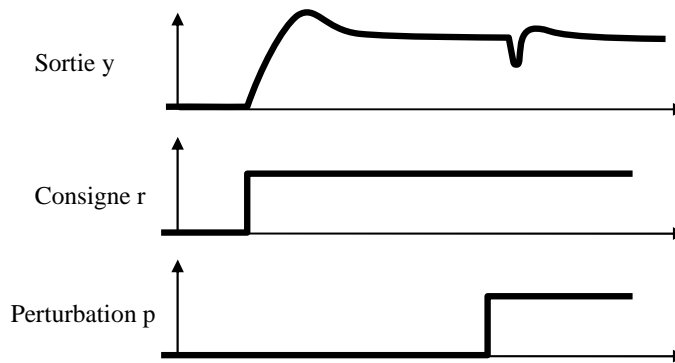


Figure 32 : Performances différentes en régulation et en poursuite

### 7.4.1 Choix des pôles en boucle fermée

De manière générale le comportement en boucle fermée est spécifié sous la forme d'une pulsation propre  $\omega_0$  et d'un amortissement  $\sigma$ , donc une fonction de transfert du second ordre. La spécification étant généralement donnée en continu il est nécessaire de traduire la valeur désirée de  $\omega_0$  et  $\sigma$  en un polynôme  $P_D(q^{-1})$  par discrétisation de la fonction de transfert continue (chapitre 4.5 page 29). Ce polynôme  $P_D$  traduit le comportement dominant du système bouclé.

Si l'on souhaite introduire un effet de filtrage supplémentaire dans certaines bandes de fréquence (réduction des bruits de mesure, adoucissement de la commande, robustesse) cela est décrit par un polynôme  $P_F$  comportant des pôles auxiliaires, plus rapides que les pôles de  $P_D$ .

Le polynôme caractéristique en boucle fermée est donc constitué de pôles dominants et de pôles auxiliaires :

$$P(q^{-1}) = P_D(q^{-1}) \cdot P_F(q^{-1}) \quad (77)$$

### 7.4.2 Régulation : calcul de R et S

Le polynôme  $P(q^{-1})$  étant fixé il s'agit de trouver R et S vérifiant :

$$P(q^{-1}) = A(q^{-1}) \cdot S(q^{-1}) + q^{-d} \cdot B(q^{-1}) \cdot R(q^{-1}) \quad (78)$$

Comme déjà vu au chapitre 7.3.1 on obtient une équation de Bézout qu'il s'agit de résoudre. (Voir aussi 9.1 Annexe 1 : Résolution de l'équation polynomiale de Bezout)

Il est parfois nécessaire de spécifier R et S avec des parties fixes  $H_R$  et  $H_S$ . Dans ce cas l'équation (78) est modifiée et on l'écrit sous la forme:



$$P(q^{-1}) = (A(q^{-1}) \cdot H_S(q^{-1})) \cdot S'(q^{-1}) + (q^{-d} \cdot B(q^{-1}) \cdot H_R(q^{-1})) \cdot R'(q^{-1}) \quad (79)$$

Les inconnues que l'on souhaite trouver sont  $S'$  et  $R'$ . Ceci est nécessaire dans les cas particuliers suivants :

- spécification d'erreur statique nulle,
- rejet d'une perturbation harmonique,
- blocage d'un signal,

Détaillons maintenant ces cas particuliers.

### Spécification d'une erreur statique nulle

On retrouve la nécessité d'avoir un terme intégrateur dans la boucle directe. La spécification se retrouve donc sur  $S$  :

$$S(q^{-1}) = (1 - q^{-1}) \cdot S'(q^{-1}) = H_S(q^{-1}) \cdot S'(q^{-1}) \quad (80)$$

### Rejet d'une perturbation harmonique

Si le système est perturbé par une perturbation harmonique à une fréquence fixe (par exemple 50Hz...) il est possible d'imposer à  $S$  d'avoir un module *nul* ou amorti à cette fréquence. Cela se traduit par la présence dans  $S$  d'une paire de zéros complexe à amortissement nul ou faible à cette fréquence :

$$S(q^{-1}) = (1 + \alpha \cdot q^{-1} + \beta \cdot q^{-2}) \cdot S'(q^{-1}) = H_S(q^{-1}) \cdot S'(q^{-1}) \quad (81)$$

### Blocage d'un signal

Pour certaines application on souhaite que le régulateur "laisse passer" certaines composantes du signal de sortie, c'est-à-dire qu'il ne réagisse pas à ces composantes. Cela se traduit par  $R=0$  à ces fréquences. On retrouve la même forme que précédemment :

$$R(q^{-1}) = (1 + \alpha \cdot q^{-1} + \beta \cdot q^{-2}) \cdot R'(q^{-1}) = H_R(q^{-1}) \cdot R'(q^{-1}) \quad (82)$$

### 7.4.3 Poursuite : calcul de T

Le schéma-bloc de l'asservissement est donné sur la Figure 33. Le terme  $T$  est décomposé en un terme  $T'$  et un terme  $B^*/A^*$ .

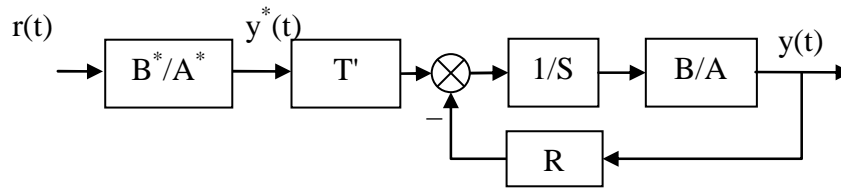


Figure 33 : placement de pôle

L'idée est de régler  $B^*$  et  $A^*$  de telle manière que le signal  $y^*$  ait la dynamique de poursuite désirée (par exemple un second ordre avec pulsation propre et amortissement donné) puis de régler  $T'$  de telle manière que le transfert entre  $y^*$  et  $y$  soit le plus "transparent" possible.

### Réglage de $T'$

Le transfert entre  $y^*$  et  $y$  est :

$$S_{y^*y}(q^{-1}) = T'(q^{-1}) \frac{q^{-d} \cdot B(q^{-1})}{P(q^{-1})} \quad (83)$$

Pour assurer le transfert le plus "transparent" possible il faut donc choisir

$$T'(q^{-1}) = \frac{P(q^{-1})}{B(1)} \quad (84)$$

Remarques :

- Si  $B(1)=0$  alors on remplace  $B(1)$  par 1 dans (84)
- De même dans (84)  $P(q^{-1})$  peut être remplacé par  $P_D$  correspondant à la dynamique dominante.

### Réglage de $B^*$ et $A^*$

$B^*$  et  $A^*$  sont choisis de telle manière que la fonction de transfert  $B^*/A^*$  corresponde à la dynamique de poursuite souhaitée.

## 7.5 Poursuite et régulation à objectifs indépendants

Cette méthode ressemble à la méthode de placement de pôles à ceci près que l'on va effectuer une simplification des zéros de  $B^*(q^{-1})$  : cela ne s'applique donc qu'aux systèmes avec des pôles stables. Plus précisément il faut qu'ils aient un amortissement suffisant ( $> 0,2$ ) de manière à ne pas être trop près du cercle unité.

La structure de l'asservissement est toujours celle donnée sur la Figure 33.

**Choix de P**

$P(q^{-1})$  est choisi de manière à simplifier  $B^*(q^{-1})$  (rappel : la forme de B est donnée dans l'équation 74). Comme indiqué dans le chapitre 7.4.2 page 60 il s'agit donc d'imposer les zéros de  $B(q^{-1})$  dans  $H_R(q^{-1})$  et  $H_S(q^{-1})$ .

**Choix de T**

Si l'on choisit  $T(q^{-1}) = P(q^{-1})$  alors le transfert entre  $y^*$  et  $y$  est :

$$S_{yy^*}(q^{-1}) = q^{-(d+1)} \quad (85)$$

**Réglage de  $B^*$  et  $A^*$** 

$B^*$  et  $A^*$  sont choisis de telle manière que la fonction de transfert  $B^*/A^*$  corresponde à la dynamique de poursuite souhaitée.



## 8. TECHNOLOGIE DE LA COMMANDE PAR ORDINATEUR

Du point de vue de l'automaticien le ordinateur est relié au système physique à piloter par un certain nombre d'intermédiaires plus ou moins gênants :

- les interfaces analogiques et numériques,
- le transport de l'information par bus et protocole,
- l'utilisation éventuelle de capteurs ou d'actionneurs « intelligents ».

Il ne s'agit pas ici de faire un exposé complet sur ces trois aspects mais de poser les problèmes qui se posent à l'automaticien lorsque le système qu'il doit commander est perturbé par ces éléments.

Enfin ce chapitre se termine sur les notions de « Hardware In the Loop » et de « prototypage rapide » qui sont partie intégrante des processus de réalisation de systèmes de contrôle.

### 8.1 Calculateurs et interfaces numériques / analogiques

Le premier problème à prendre en compte est la quantification de l'information. Du point de vue du ordinateur d'une part et du point de vue des convertisseurs d'autre part. Dans un premier temps, considérons les aspects performance de calcul lié à la nature du codage des nombres dans le ordinateur.

#### Les performances liées au types de variables

Prenons l'exemple où le concepteur de la loi de commande souhaite réaliser un correcteur de type proportionnel. Entre l'acquisition des mesures et l'écriture de la consigne il y aura la réalisation d'une de commande du type :

$$y = K \cdot (\text{consigne} - \text{mesure}) \quad (86)$$

En utilisant un langage de programmation évolué (par exemple le C) la ligne de code sera la transcription littérale de (86). Le programmeur devra décider de la définition et du type des variables, principalement du nombre d'octets utilisés pour stocker ses variables. Le Tableau 5 résume par exemple les types utilisés en langage C (norme ANSI)

Tableau 5 : Types ANSI C

Types entiers	char	8 bits
	short int	16 bits
	long int	32 bits
Types flottants (mantisse et exposant)	float	32 bits
	double	64 bits
	longdouble	128 bits

Une fois le code écrit en C l'exécution du programme sur un processeur cible ne peut se faire que par l'intermédiaire d'une compilation du programme. A chaque processeur est donc adapté une version de compilateur.

Les performances du programme (sa rapidité d'exécution) dépendront donc en partie de la qualité du compilateur. (Par exemple, une multiplication s'effectue t'elle bêtement par un décalage de tous les bits ou bien des solutions astucieuses sont-elles implémentées pour arrêter le décalage selon le contexte ?)

Le principal déterminant de la performance est cependant le processeur : son type (8 bits, 16 bits ou 64 bits) et son cadencement (en MHz ou GHz) sont les deux plus importants paramètres. Coder les variables sur 64 bits (double) dans le programme coûtera cher en temps de calcul si le processeur est un 8 bits. Le type de mémoire et leurs temps d'accès peuvent aussi être déterminants. Enfin, certains processeurs disposent d'un processeur spécifiquement dédié aux opérations mathématiques (coprocesseur mathématique).

Il n'est pas possible de dire exactement, pour tel type de compilateur associé à tel type de processeur, le temps que prendra par exemple une multiplication. En pratique on effectue une série de tests types (benchmark) et on affiche un temps moyen d'exécution.

En résumé, on peut être tenté, pour ne pas s'embêter, de coder toutes les variables en virgule flottante. Mais il faut bien savoir que cela a un coût en terme de temps de calcul, surtout si le processeur n'est pas adapté. Pour fixer les idées on peut dire qu'il y a un rapport 20 entre une addition réalisée en "short" et une addition réalisée en virgule flottante.

Enfin, l'utilisation de fonctions mathématiques (sinus, exponentielles, etc...) peut aussi avoir un coût très lourd si le travail n'est pas réalisé par un coprocesseur mathématique dédié. Le compilateur et sa librairie mathématique joue là encore un grand rôle.

### La quantification imposée par le processeur

Au-delà de l'aspect performance et temps de calcul, la quantification de l'information doit parfois aussi être prise en compte. Rappelons qu'un entier codé sur N bits peut prendre seulement  $2^N - 1$  valeurs différentes. Avant d'effectuer les calculs mathématiques dans le corps du programme il est donc judicieux d'appliquer un

facteur d'échelle de manière à utiliser l'amplitude maximale possible (entre 0 et  $2^N - 1$ ).

### **La quantification imposée par les conversions analogique et numérique**

Les convertisseurs analogique-numérique et numérique-analogique sont principalement caractérisés par leur amplitude de signal analogique d'entrée (généralement en volts, par exemple +/- 10V) et par leur résolution en bits (8 ou 10 bits pour les plus courants, jusqu'à 16 bits pour les convertisseurs haut de gamme).

Prenons l'exemple d'un capteur gyromètre transformant une vitesse de rotation en tension. Compte tenu de l'application le signal utile varie entre 2,4V et 2,6V. Si le signal mesuré est branché directement sur un convertisseur 0-10V 10bits alors un problème de conditionnement du signal se pose. En effet la quantification est de  $\frac{10V}{2^{10}-1} = 0,098V$ . Les 0,2V de signal utile seront donc discrétisés en  $\frac{0,2}{0,098} = 20$  intervalles seulement !

Si la discrétisation du signal utile est trop grossière cela peut se traduire par une diminution des performances de l'asservissement, voire une instabilité. Le système n'est en effet plus linéaire et il n'existe pas d'outils théoriques permettant de prévoir les performances du système bouclé.

## **8.2 Transport de l'information : les bus et les protocoles**

Pas question de faire ni même d'ébaucher un cours sur les bus et les protocoles ici. Mais on va quand même donner quelques indications. L'automaticien a l'habitude de simplifier le problème qu'il se pose en dessinant un schéma simplifié de son problème d'asservissement (du type de la Figure 20 page 45) pour pouvoir appliquer les théories de la commande. Cependant il arrive bien souvent que l'information transite à l'intérieur du système, soit entre différents ordinateurs effectuant différentes parties du travail soit entre ordinateur et périphériques, par l'intermédiaire de bus informatique. A chaque type de bus correspond un ou plusieurs protocoles qui peuvent poser des problèmes pour l'automaticien :

- les données sont-elles discrétisées ? (le protocole peut par exemple ne permettre de transmettre que des paquets de données de 16 bits),
- le bus induit-t'il un retard ? (le composant mettant en forme les données du ordinateur pour les publier sur le bus a certaines performances qui peuvent induire un délai)
- le bus est-il déterministe ? (Certains bus ne garantissent pas que les données arriveront en un temps donné),
- le bus est-il sûr ? (Certains bus ne garantissent pas non plus que les données arriveront bien)

### **8.3 Capteurs et actionneurs "intelligents", "smart sensors"**

De plus en plus souvent les capteurs et les actionneurs sont associés à leur électronique de puissance, électronique de commande et de conditionnement de signal. L'interface avec le capteur ou l'actionneur se fait donc via un protocole de communication (par exemple via un bus RS232).

L'intérêt est de véhiculer non pas des signaux analogiques, tensions ou courants, mais des signaux numériques beaucoup moins sensibles aux éventuelles perturbations.

On retrouve donc, entre le capteur ou l'actionneur et le ordinateur, la présence d'un bus et d'un protocole de communication.

### **8.4 La philosophie "hardware in the loop"**

Les systèmes embarqués sont de plus en plus complexes et touchent des domaines de plus en plus variés : automobile, satellites, aéronautique, UAV, systèmes d'arme, sous-marins, systèmes, etc... Les systèmes à piloter ainsi que les systèmes de commande sont de plus en plus complexes.

Les techniques de simulation « Hardware In the Loop » (HIL) sont de plus en plus utilisées dans le développement et dans le test de ces systèmes embarqués. L'idée est de développer entièrement le système de contrôle sur une plateforme de test en utilisant le maximum de composants réels et en ne remplaçant que le minimum par la simulation. Par exemple en aéronautique l'« Iron Bird » regroupe dans un hangar tous les systèmes réels de l'avion en cours de développement. Seule la mécanique du vol est simulée : les sorties des actionneurs sont mesurées, rentrées dans un modèle de simulation numérique, puis les mesures effectuées normalement par les capteurs sont calculées et restituées par le modèle.

L'intérêt est de diminuer de manière importante les coûts et les délais en économisant le plus grand nombre possible de tests sur le système réel en situation. Lors des premiers tests réels les techniques « Hardware In the Loop » permettent d'avoir effectué la plus grande part du développement des lois de commande.

### **8.5 Le prototypage rapide en automatique**

Un peu dans le même ordre d'idée que le « Hardware In the Loop » le prototypage rapide permet de réaliser la conception des lois de commande en simulation puis de transporter rapidement (en un simple click !) le code du contrôleur simulé vers la cible temps réel implémentée sur le système réel.

La société Mathworks [9] investit massivement dans ce domaine en ayant enrichi son logiciel de calcul scientifique (Matlab) avec un logiciel de simulation graphique très évolué (Simulink) puis en développant des outils de prototypage rapide associés (Real Time Workshop, Embedded Target For...).



## 9. ANNEXE

### 9.1 Annexe 1 : Résolution de l'équation polynomiale de Bezout

Il s'agit de trouver R et S vérifiant :

$$P(q^{-1}) = A(q^{-1}).S(q^{-1}) + B(q^{-1}).R(q^{-1})$$

Où P, A et B sont connus.

Dans le cas où P, A, B, R et S sont des nombres entiers la solution de l'algorithme est donnée par l'algorithme de division Euclidienne étendu [8].

Dans le cas où P, A, B, R et S sont des polynômes, l'algorithme est donné ci-dessous, programmé en matlab ([1]).

```
function [Rp,Sp,nrp,nsp]=bezoutd(A,B,Hs,Hr,P)
%function [Rp,Sp,nrp,nsp]=bezoutd(A,B,Hs,Hr,P)
%solves AHsSp+BHrRp=P by coefficient comparison. Delay and discretization delay need to be integrated in B
%(zeros at the beginning).
%
%inputs:
%A=[a0 a1 ... aNa] ... vector of model denominator coefficients A=a0 + a1z^(-1) + a2z^(-2) +...+ aNaz^(-Na)
%B=[b0 b1 ... bNb] ... vector of model numerator coefficients B=b0 + b1z^(-1) + b2z^(-2) +...+ bNbz^(-Nb)
%Hs=[hs0 hs1 ... hsNhs] ... vector of controller denominator fixed part Hs=hs0 + hs1z^(-1) +...+ hsNhsz^(-Nhs)
%Hr=[hr0 hr1 ... hrNhr] ... vector of controller denominator fixed part Hr=hr0 + hr1z^(-1) +...+ hrNhrz^(-Nhr)
%P=[p0 p1 ... pNp] ... vector of desired polynomial coefficients P=p0 + p1z^(-1) + p2z^(-2) +...+ pNpz^(-Np)
%outputs:
%Rp=[rp0 rp1 rp2 ...] ... vector of coefficients for resulted controller numerator
%Sp=[sp0 sp1 sp2 ...] ... vector of coefficients for resulted controller denominator
%nrp ... order of Rp
%nsp ... order of Sp
%
%written by: J. Langer, I.D. Landau, H. Prochazka
%7th june 2002

PRECISION=1e-16;

D=size(A);
if D(1)>1, A=A'; end;
```

```

D=size(B);
if D(1)>1, B=B'; end;
D=size(Hs);
if D(1)>1, Hs=Hs'; end;
if D(1)==0, Hs=1; end;
D=size(Hr);
if D(1)>1, Hr=Hr'; end;
if D(1)==0, Hr=1;end;
D=size(P);
if D(1)>1, P=P'; end;

na=length(A)-1;
nb=length(B)-1;
np=length(P)-1;
nhs=length(Hs)-1;
nhr=length(Hr)-1;

if (nhs>0), Ah=real(convz(A,Hs)); else Ah=A*Hs; end; % Ah = A * Hs
nah=length(Ah) -1;
if (nhr>0), Bh=real(convz(B,Hr)); else Bh=B*Hr; end;
nbh=length(Bh) -1;
if (np>nah+nbh-1), disp('Bezout error: too many poles');end;

% increase size of P using zeros if necessary
if (np<nah+nbh-1),
%
% set remaining poles onto a circle with radius rmin
P=[P zeros(1,nah+nbh-1-np)];
rootsPdes=roots(P);
nextra=nah+nbh-1-np;
rmin=1e-16;
angle=[0:nextra-1]'/nextra*2*pi;
j=sqrt(-1);
rootsPextra=rmin*exp(j*angle);
P=poly([rootsPdes;rootsPextra]);
np=nah+nbh-1;

end;
P,

nsp=nbh-1;
nrp=nah-1;

%matrix is smaller than vector PD

```

```

if (np>nah+nbh-1),
    disp('Order of model denominator is too low! Add a polynom of higher order to Hs or Hr. ');
end;

% ns=nsp+nhs
% nr=nrp+nhr

M=[];
for j=1:nsp+1,
    V=[];
    if (j>1), V=[V ; zeros(j-1,1)]; end;% zeros in front of Ah
    V=[V ; Ah'];% Ah
    if (j<=nsp), V=[V ; zeros(nsp+1-j,1)]; end;% zeros behind Ah
    if (length(V)~=nah+nbh), disp('bezoutb: error V'); end;
    M=[M V]; % add one column to M
end;

for j=1:nrp+1,
    V=[];
    if (j>1), V=[V ; zeros(j-1,1)]; end;
    V=[V ; Bh'];
    if (j<=nrp), V=[V ; zeros(nrp+1-j,1)]; end;
    if (length(V)~=nah+nbh), disp('bezoutb: error V'); end;
    M=[M V];
end;

D=size(M);
if (D(1)~=nah+nbh), disp('bezoutb: error size M row'); end;
if (D(2)~=nah+nbh), disp('bezoutb: error size M column'); end;

% make P column vector
P=P';

global M1;
M1=M;

X= M\P;
% coefficients are real values
X=real(X);

% make X row vector
X=X';

```

## Commande des systèmes par ordinateur

```
Sp=X(1:nsp+1);  
Rp=X(nsp+2:nsp+nrp+2);
```

## 10. BIBLIOGRAPHIE

- [1] I. D. Landau, "*Commande des systèmes*", Edition Lavoisier 2002.
- [2] C. Nouals, "Théorie du Signal", Notes de cours ENSICA
- [3] F. Vincent, "*Traitement Numérique du Signal*", Notes de cours ENSICA.
- [4] D. Arzelier, "*Représentation et analyse des systèmes, notes de cours*", Edition ENSICA. Voir aussi <http://www.laas.fr/~arzelier/>
- [5] J. Bordeneuve-Guibé, "*Commande des systèmes linéaires*", Notes de cours ENSICA.
- [6] Y. Briere, "*Identification*", Notes de cours ENSICA.
- [7] L. Ljung, "*System Identification, Theory for the User*", Prentice Hall 1987.
- [8] Wikipedia , équation de Bezout : [http://en.wikipedia.org/wiki/Bézout's\\_identity](http://en.wikipedia.org/wiki/Bézout's_identity)
- [9] <http://www.mathworks.com>



# 11. INDEX

<b>A</b>	
actionneur intelligent .....	21
amortissement .....	15
ANSI C .....	60
Approximation numérique d'équation différentielle .....	28
ARMAX .....	35
<b>B</b>	
BJ35 .....	
Blocage d'un signal .....	56
bloqueurs d'ordre 0 .....	22
BOZ .....	22
bus.....	21, 61
<b>C</b>	
CAN.....	21
capteur intelligent .....	21
causal .....	6, 9
Causalité .....	10
CNA.....	21
compilateur .....	60
convertisseur analogique / numérique.....	21
convertisseur numérique / analogique.....	21
convertisseurs analogique-numérique .....	61
coprocesseur mathématique .....	60
Correspondances pôles - zéros.....	31
<b>D</b>	
Diagramme de Bode .....	18
Discrétisation de fonction de transfert continues .....	27
division polynomiale.....	6
<b>E</b>	
échelon unitaire.....	11
équation d'état .....	17
Erreur statique.....	42
Estimation de la réponse fréquentielle .....	39
Estimation de la réponse impulsionnelle.....	38
Euler .....	29
<b>F</b>	
facteur d'échelle .....	61
facteur d'oubli .....	36
<i>fenêtre de Hamming</i> .....	39
filtre anti-repliement .....	25
fonction de sensibilité .....	42
fonction de transfert discrète.....	9
<b>G</b>	
gain statique.....	11
gain statique unitaire.....	42
<b>I</b>	
Identification.....	32
Identification non paramétrique .....	38
intercorrélation.....	38
<b>L</b>	
lieu des racines.....	43
lieu d'Evans.....	43
LTI.....	8
<b>M</b>	
marge de gain.....	47
marge de module.....	46
marge de phase .....	47
méthode des moindres carrés récursifs .....	32
Méthode des trapèzes.....	29
modèle ARX .....	32
<b>O</b>	
OE.....	35
one step ahead prediction.....	35
opérateur de retard .....	9
<b>P</b>	
période d'échantillonnage (choix) .....	26
Placement de pôles .....	54
Pôles .....	11
Poursuite .....	56
<i>prédiction</i> .....	32
protocoles.....	61
Pseudo Random Binary Signal .....	34
pulsation .....	15
<b>Q</b>	
quantification .....	23, 24, 60
<b>R</b>	
R,S,T.....	41
rapidité.....	15
rayon de convergence .....	5, 9
Régulation.....	55

Rejet de perturbations .....	42
Rejet d'une perturbation harmonique .....	56
réponse à un échelon.....	11
réponse fréquentielle.....	18
Réponse impulsionnelle.....	10
Réponses indicielle .....	10
retard.....	23
root locus .....	43

### S

signal discret .....	5
smart sensors.....	62
stabilité.....	11
Systèmes du premier ordre.....	14
Systèmes du second ordre.....	15

### T

Théorème de Nyquist.....	46
théorème de Shannon.....	19, 24
théorème du retard .....	6
Transformée en z .....	5
transformée unilatérale .....	5
Tustin.....	29

### V

virgule flottante.....	60
------------------------	----

### Z

zero holder hold .....	23
zéros.....	11
zoh .....	23



# Commande des systèmes par ordinateur

Yves Briere  
ENSICA

Notes de cours  
2007



# TABLE DES MATIERES

Table des matières .....	1
1. Introduction .....	1
2. Rappels et prérequis .....	3
3. Signaux et systèmes discrets .....	5
3.1 Signaux discrets .....	5
3.1.1 Transformée en z .....	5
3.1.2 Signaux types .....	7
3.2 Systèmes discrets .....	9
3.2.1 Fonction de transfert discrète, définition .....	9
3.2.2 Notation "opérateur de retard $q^{-1}$ " .....	10
3.2.3 Réponse impulsionnelle .....	11
3.2.4 Causalité .....	11
3.2.5 Réponses indicielle .....	11
3.2.6 Régime stationnaire (gain statique) .....	12
3.2.7 Pôles, zéros, stabilité .....	12
3.2.8 Systèmes du premier ordre .....	15
3.2.9 Systèmes du second ordre .....	16
3.2.10 Calcul itératif de la réponse d'un système .....	18
3.2.11 Représentation par équation d'état .....	18
3.3 Réponse fréquentielle des systèmes échantillonnés .....	19
3.3.1 Diagramme de Bode d'un système échantillonné .....	19
3.3.2 Où l'on voit apparaître le théorème de Shannon .....	20
4. Systèmes continus échantillonnés .....	23
4.1 Structure des systèmes commandés par ordinateur .....	23
4.2 Échantillonnage des systèmes continus .....	24
4.3 Effets de l'échantillonnage .....	25
4.4 Où l'on revoit le théorème de Shannon .....	26
4.5 Discrétisation de fonction de transfert continues .....	29
4.5.1 Approximation numérique d'équation différentielle .....	30
4.5.2 Superposition des réponses indicielles .....	32
4.5.3 Correspondances pôles - zéros .....	33
4.5.4 Bilan .....	33
5. Identification des systèmes échantillonnés .....	35
5.1 Identification d'un modèle ARX par la méthode des moindres carrés récurrents .....	35
5.2 L'identification paramétrique en général et en pratique .....	37

5.2.1	Choix du signal d'excitation, traitement des mesures .....	37
5.2.2	Choix d'une structure de modèle.....	38
5.2.3	Choix d'un critère à minimiser.....	38
5.2.4	Choix d'un algorithme.....	38
5.2.5	Validation.....	39
5.2.6	Utilisation de logiciels .....	39
5.3	Identification "simple" .....	39
5.4	Identification non paramétrique .....	41
5.4.1	Estimation de la réponse impulsionnelle .....	41
5.4.2	Estimation de la réponse fréquentielle.....	42
6.	Systèmes bouclés .....	45
6.1	Fonction de transfert en boucle fermée .....	45
6.2	Erreur statique .....	46
6.3	Rejet de perturbations.....	46
6.4	Stabilité des systèmes à simple gain de rétroaction.....	47
6.5	Analyse des systèmes bouclés dans le domaine fréquentiel : stabilité.....	49
6.5.1	Théorème de Nyquist dans le cas où le système est stable en boucle ouverte (critère du revers) .....	50
6.5.2	Théorème de Nyquist dans le cas où le système est instable en boucle ouverte .....	50
6.5.3	Interprétations du lieu de Nyquist.....	50
7.	Méthodes de régulation des systèmes échantillonnés .....	53
7.1	Introduction .....	53
7.2	Structure d'un système commandé par ordinateur, hypothèses.....	54
7.2.1	Discretisation, choix de la période d'échantillonnage.....	54
7.2.2	Discretisation d'un correcteur continu .....	55
7.3	Régulateur PID numérique .....	55
7.3.1	Cas 1 .....	55
7.3.2	Cas 2 .....	58
7.4	Placement de pôles .....	59
7.4.1	Choix des pôles en boucle fermée .....	60
7.4.2	Régulation : calcul de R et S.....	60
7.4.3	Poursuite : calcul de T .....	61
7.5	Poursuite et régulation à objectifs indépendants .....	62
8.	Technologie de la commande par ordinateur.....	65
8.1	Calculateurs et interfaces numériques / analogiques.....	65
8.2	Transport de l'information : les bus et les protocoles .....	67
8.3	Capteurs et actionneurs "intelligents", "smart sensors" .....	68
8.4	La philosophie "hardware in the loop" .....	68
8.5	Le prototypage rapide en automatique .....	68
9.	Annexe .....	69
9.1	Annexe 1 : Résolution de l'équation polynomiale de Bezout.....	69
10.	Bibliographie.....	73
11.	Index.....	1





# 1. INTRODUCTION



Figure 1 : commande d'un systèmes de télémédecine par calculateur, vu par Claude Serre





## 2. RAPPELS ET PREREQUIS

Ce polycopié ne comporte pas de rappels sur les systèmes continus. S'adressant aux étudiants en deuxième année de l'ENSICA il suppose connu les enseignements de première année et de seconde année en automatique et traitement du signal.

En première année le cours de théorie du signal [2] a rendu familier les notions de :

- transformée de Fourier, transformée de Laplace
- corrélation, convolution

Le cours de traitement numérique du signal [3] a abordé en plus :

- la discrétisation et la quantification des signaux.
- la transformée en z et la transformée de Fourier discrète.
- les filtres numériques RIF et RII.

En automatique [4] les bases de la représentation et de l'analyse des systèmes continus ont été posées. On ne reviendra pas dans ce cours sur les notions de :

- modèles mathématique des systèmes linéaires à temps continus à temps invariant (LTI) : fonction de transfert, équation d'état, pôles, zéros et dynamique.
- commandabilité et observabilité.
- stabilité des systèmes en boucle ouverte : pôles, critères de Routh-Hurwitz.
- stabilité des systèmes bouclés : critère de Nyquist, critère de Bode.
- lieu des racines.

En deuxième année les notions suivantes sont acquises [5] :

- synthèse des correcteurs par méthodes fréquentielles (avance et retard de phase, PID)
- commande par retour d'état (placement de pôles), commande Linéaire Quadratique (LQG)
- reconstruction de l'état par Observateur, observateur optimal de Kalman.



## 3. SIGNAUX ET SYSTEMES DISCRETS

### 3.1 Signaux discrets

#### 3.1.1 Transformée en z

Pour une approche mathématiquement rigoureuse de représentation des signaux discrets par la transformée en z consulter [3] et sa bibliographie. Dans ce cours on se cantonne à une approche utilitariste des outils et méthodes mathématiques tout en essayant de ne pas contredire fondamentalement la rigueur mathématique.

Un signal discret  $s(k)$  est une séquence de nombres réels appelés échantillons. L'indice  $k$  est généralement associé à un instant d'échantillonnage :  $t = k.T_e$ . La période d'échantillonnage  $T_e$  est supposée constante. Par définition la transformée en z du signal  $s(k)$  est la fonction de la variable  $z$ , notée abusivement  $s(z)$  :

$$s(z) = Z(s(k)) = \sum_{k=0}^{\infty} s(k) \cdot z^{-k} \quad (1)$$

La transformée définie ici est plus précisément appelée transformée unilatérale puisque les termes correspondant à  $k < 0$  ne sont pas pris en compte. Ceci n'a pas d'importance puisque les signaux que nous utilisons sont systématiquement causaux<sup>1</sup> :  $s(k) = 0$  pour  $k < 0$ .

Il y a bien évidemment un lien avec la transformée de Laplace et la transformée de Fourier (voir [3])

La fonction de la variable complexe  $s(z)$  est une série et ne converge pas forcément pour toute valeur de  $z$ . En toute rigueur il faudrait associer le rayon de convergence à chaque fonction en  $z$  :  $\{s(z) \text{ converge pour } |z| > \alpha\}$ .

Les automaticiens pour qui la transformée en z n'est bien souvent qu'une façon bien commode de regrouper des termes de récurrence utilisent souvent la pseudo-variable<sup>2</sup>  $q$ .

A partir d'une transformée en z  $s(z)$  il est possible de retrouver la séquence  $s(k)$  correspondante en appliquant la transformation inverse, dont on ne donnera pas la formule ici. A la place de faire appel à cette formule compliquée il est possible d'utiliser d'autres méthodes pour retrouver la séquence  $s(k)$  lorsque la fonction  $s(z)$  est donnée sous forme de fraction rationnelle :

<sup>1</sup> Pour une justification du terme « causal » voir page 11

<sup>2</sup> Voir page 10

- division polynomiale
- décomposition de la fraction rationnelle en éléments simples puis utilisation de transformations élémentaires telles que celles données dans le Tableau 2

La propriété fondamentale de la transformée en z est qu'elle est linéaire (la transformée en z d'une composition linéaire de signaux est la composition linéaire de la transformée en z de chaque signal). Les autres propriétés fondamentales sont données dans le Tableau 2.

Tableau 1 : propriétés de la transformée en z

1	$s(z) = Z(s(k)) = \sum_{k=0}^{\infty} s(k) \cdot z^{-k}$
2	$Z(\alpha \cdot s_1(k) + \beta \cdot s_2(k)) = \alpha \cdot Z(s_1(k)) + \beta \cdot Z(s_2(k))$
3	$Z(k \cdot s(k)) = -z \cdot \frac{ds(z)}{dz}$
4	$Z(c^k \cdot s(k)) = s\left(\frac{z}{c}\right)$
5	$Z(s(k-1)) = s(k=-1) + z^{-1} \cdot s(z)$
6	$Z(s(k+1)) = z \cdot s(z) - z \cdot s(k=0)$
7	$s(k=0) = \lim_{z \rightarrow \infty} (s(z))$
8	$s(k \rightarrow \infty) = \lim_{z \rightarrow 1} \left( \frac{z-1}{z} \cdot s(z) \right)$

Remarque 1 : la propriété 5 ne prend pas en compte le fait que le signal est causal : en pratique  $s(k=-1) = 0$ .

Remarque 2 : la propriété 5 justifie le fait que le terme  $z^{-1}$  soit appelé "opérateur de retard unitaire", cette propriété est aussi appelée "théorème du retard".

Remarque 3 : les propriétés 7 et 8 sont données sous réserve de convergence.

Transformées inverses

### Exemple 1 : transformée inverse par division polynomiale

On considère un signal  $s(k)$  dont la transformée en z est donnée par la fraction rationnelle :

$$S(z) = \frac{0.1 \cdot z^2 + 0.1 \cdot z}{z^3 - 2.6 \cdot z^2 + 2.2 \cdot z - 0.6} \quad (3.1)$$

L'objectif est d'obtenir  $S(z)$  sous forme polynomiale :

$$S(z) = x_0 + x_1 \cdot z^{-1} + x_2 \cdot z^{-2} + \dots \text{etc.} \quad (3.2)$$

Cette suite est généralement infinie, la réponse temporelle est donnée par la succession des termes  $\{x_i\}$ . En traitant l'équation (3.1) comme une simple division l'algorithme de division classique (celui appris en CM2) permet d'obtenir par récurrence tous les termes de la suite (3.2) :

$0.1 \cdot z^2$	$+ 0.1 \cdot z$				$z^3 - 2.6 \cdot z^2 + 2.2 \cdot z - 0.6$
$0.1 \cdot z^2$	$- 0.26 \cdot z$	$+ 0.22$	$- 0.06 \cdot z^{-1}$	$0.1 \cdot z^{-1} + 0.36 \cdot z^{-2} + 0.716 \cdot z^{-3}$	
0	$0.36 \cdot z$	$- 0.22$	$0.06 \cdot z^{-1}$		
	$0.36 \cdot z$	$- 0.936$	$0.792 \cdot z^{-1}$	$- 0.216 \cdot z^{-2}$	
0	$0.716$	$- 0.732 \cdot z^{-1}$	$\dots$	$\dots$	

Les éléments successifs de la suite  $s(k)$  sont donc  $s(k) = \{0, 0.1, 0.36, 0.716, \dots\}$ . Pour obtenir les termes suivants il faudrait continuer la division.

**Exemple 2 : transformée inverse par décomposition en éléments simples**

La deuxième méthode présentée consiste à décomposer la fraction rationnelle (3.1) en éléments simples puis d'utiliser les tables de transformées en  $z$  (Tableau 2). Pour notre exemple cela donne :

$$S(z) = \frac{0.1 \cdot z^2 + 0.1 \cdot z}{z^3 - 2.6 \cdot z^2 + 2.2 \cdot z - 0.6} = \frac{-z}{z-1} + \frac{0.5 \cdot z}{(z-1)^2} + \frac{z}{z-0.6}$$

La réponse temporelle est donc composée des trois éléments :

$$s(k) = -(1 \ 1 \ 1 \ \dots) + 0.5 \cdot (1 \ 2 \ 3 \ \dots) + (1 \ 0.6 \ 0.6^2 \ \dots)$$

$$\text{soit : } s(k) = \begin{cases} s(0) = -1 + 0.5 + 1 \\ s(2) = -1 + 0.5 \cdot 2 + 0.6 \cdot 0.5 \\ s(3) = -1 + 0.5 \cdot 3 + 0.6^2 \\ s(4) = \dots \end{cases}$$

**3.1.2 Signaux types**

Le Tableau 2 donne les transformée en  $z$  de certains signaux fondamentaux. La Figure 2 représente l'allure de quelques signaux types.

Tableau 2 : transformée en z élémentaires

$s(k)$	$S(z)$
signal temporel	Transformée en z
$\delta(k)$ : impulsion unité	1
$u(k)$ : échelon unité	$\frac{z}{z-1}$
$k \cdot u(k)$	$\frac{z}{(z-1)^2}$
$c^k \cdot u(k)$	$\frac{z}{z-c}$
$k \cdot a^k \cdot u(k)$	$\frac{a \cdot z}{(z-a)^2}$
$\sin(\omega \cdot k) \cdot u(k)$	$\frac{z \cdot \sin(\omega)}{z^2 - 2 \cdot z \cdot \sin(\omega) + 1}$
$\cos(\omega \cdot k) \cdot u(k)$	$\frac{z \cdot (z - \cos(\omega))}{z^2 - 2 \cdot z \cdot \sin(\omega) + 1}$
$a^k \cdot \sin(\omega \cdot k) \cdot u(k)$	$\frac{a \cdot z \cdot \sin(\omega)}{z^2 - 2 \cdot a \cdot \cos(\omega) \cdot z + a^2}$
$a^k \cdot \cos(\omega \cdot k) \cdot u(k)$	$\frac{z \cdot (z - a \cdot \cos(\omega))}{z^2 - 2 \cdot a \cdot \cos(\omega) \cdot z + a^2}$

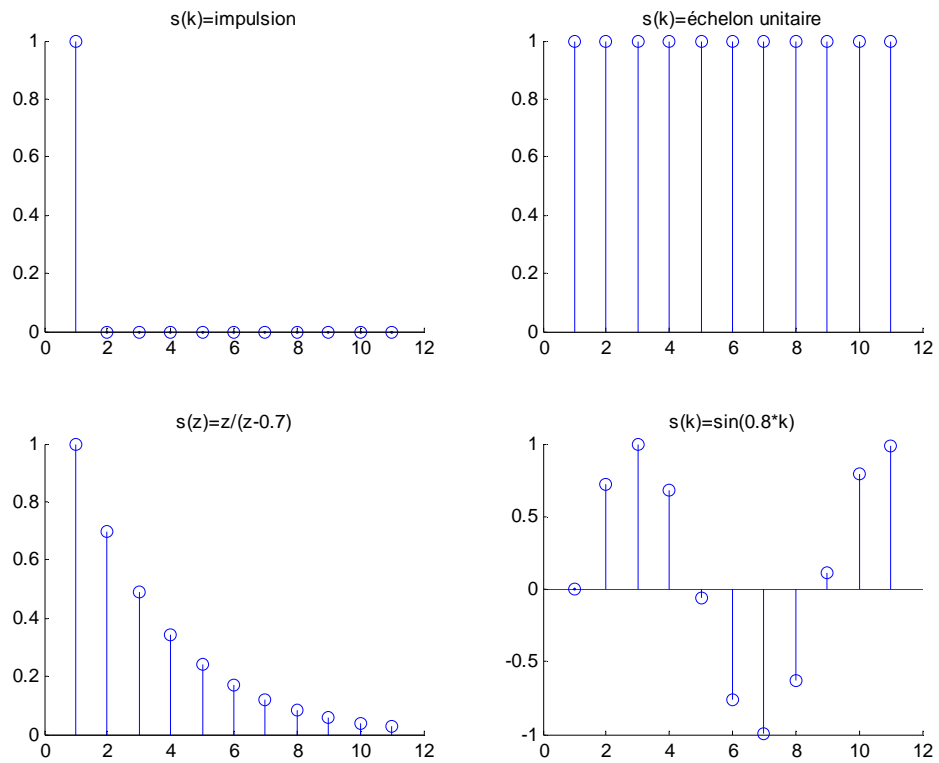


Figure 2 : réponses de quelques signaux types

### 3.2 Systèmes discrets

#### 3.2.1 Fonction de transfert discrète, définition

On considère maintenant un système discret linéaire et invariant dans le temps (LTI) représenté sur la Figure 3.

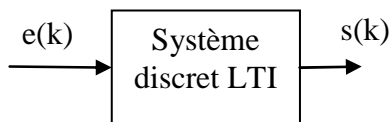


Figure 3 : système discret, notations

La relation entre la séquence d'entrée et la séquence de sortie est linéaire et définie par un ensemble de coefficients constants  $a_{1:n}$  et  $b_{1:m}$  :

$$\begin{aligned} s(k) + a_1 \cdot s(k-1) + a_2 \cdot s(k-2) + \dots + a_n \cdot s(k-n) = \\ b_0 \cdot e(k) + b_1 \cdot e(k-1) + b_2 \cdot e(k-2) + \dots + b_m \cdot e(k-m) \end{aligned} \quad (2)$$

On peut appliquer la transformée en  $z$  à chaque membre de l'équation puis en appliquant en récurrence la propriété 5 (théorème du retard) on obtient la fonction de transfert discrète du système :

$$\frac{s(z)}{e(z)} = F(z) = \frac{b_0 + b_1 \cdot z^{-1} + b_2 \cdot z^{-2} + \dots + b_m \cdot z^{-m}}{1 + a_1 \cdot z^{-1} + a_2 \cdot z^{-2} + \dots + a_n \cdot z^{-n}} \quad (3)$$

Remarque 1 : la fonction de transfert précédente est normalisée en prenant  $a_0 = 1$ .

Remarque 2 : on remarquera que la fonction de transfert est donnée en  $z^{-1}$  plutôt qu'en  $z$ . Cela est justifié par le fait que celle-ci traduit l'équation d'évolution du système physique causal (2).

Le script Matlab permettant de créer un objet Matlab correspondant à une fonction de transfert discrète est donné ci-dessous.

#### Matlab

```
% Création d'un système linéaire discret :
Te=0.1;
sysd = tf([b0 b1 b2],[1 a0 a2 a3],Te,'variable','z^-1');
```

On notera l'utilisation de l'argument 'variable', 'z^-1'.

### 3.2.2 Notation "opérateur de retard $q^{-1}$ "

La fonction de transfert est une fonction de la variable complexe  $z$ . Il faudrait donc en toute rigueur l'associer à son rayon de convergence. En pratique on ne s'embarrasse jamais de préciser cette zone de convergence. C'est pourquoi on trouve souvent la notation utilisant l'opérateur  $q^{-1}$  de décalage à la place de la variable complexe  $z^{-1}$  (4).  $F(q)$  n'est alors plus à proprement parler une fonction mais on lui appliquera les mêmes manipulations algébriques qu'à  $F(z)$ .

$$F(q) = \frac{b_0 + b_1 \cdot q^{-1} + b_2 \cdot q^{-2} + \dots + b_m \cdot q^{-m}}{1 + a_1 \cdot q^{-1} + a_2 \cdot q^{-2} + \dots + a_n \cdot q^{-n}} \quad (4)$$

Attention : selon les auteurs l'opérateur qui remplace  $z^{-1}$  est noté  $q$  (Matlab) ou  $q^{-1}$  (Landau [1]). Dans la suite du document c'est cette dernière convention qui sera appliquée.



L'utilisation de l'opérateur de retard  $q^{-1}$  à la place de la variable complexe  $z^{-1}$  permet de traiter du cas des fonctions de transfert à coefficients non constant, cas non prévu par la transformation en  $z$ .

### 3.2.3 Réponse impulsionnelle

Essayons d'appliquer au système un signal d'entrée particulier : une impulsion unitaire. La transformée en  $z$  du système est obtenue en multipliant la fonction de transfert par la transformée en  $z$  de l'impulsion unitaire : 1. On obtient :

$$s(z) = \frac{b_0 + b_1 \cdot z^{-1} + b_2 \cdot z^{-2} + \dots + b_m \cdot z^{-m}}{1 + a_1 \cdot z^{-1} + a_2 \cdot z^{-2} + \dots + a_n \cdot z^{-n}} \times 1 \quad (5)$$

Cela justifie le fait que la fonction de transfert du système peut aussi s'appeler "réponse impulsionnelle".

#### Matlab

```
% Réponse impulsionnelle d'un système :
[y,t] =impulse(sysd);
```

### 3.2.4 Causalité

Un système est dit causal si les sorties ne dépendent que du passé et non pas du futur. La conséquence est que la réponse impulsionnelle de  $F(z)$  ne peut pas comporter de terme non nuls pour  $k < 0$ . Cela justifie la confusion sémantique entre "système causal" et "signal causal", c'est-à-dire ne comportant pas de termes non nuls pour  $k < 0$ .

Attention : à la différence des systèmes continus un système physique (causal) discret peut très bien être modélisé avec un degré  $m$  supérieur au degré  $n$  : le système peut avoir plus de zéros que de pôles.

La transformée en  $z$  représentant des signaux échantillonnés elle peut tout aussi bien représenter les mesures prises à différents endroits d'une poutre. Le temps étant remplacé ici par une abscisse la notion de causalité ne s'applique pas ! Attention : beaucoup des propriétés de systèmes échantillonnés que l'on étudie sont liés à la causalité, prudence donc si l'on applique ces méthodes à d'autres domaines !

### 3.2.5 Réponses indicielle

Pour caractériser des systèmes on s'intéresse souvent à leur réponse impulsionnelle ou, plus souvent, à leur réponse indicielle. Pour calculer la réponse

à un échelon d'un système il faut donc multiplier la fonction de transfert par la transformée en z d'un échelon unitaire :

$$s(z) = \frac{b_0 + b_1 \cdot z^{-1} + b_2 \cdot z^{-2} + \dots + b_m \cdot z^{-m}}{1 + a_1 \cdot z^{-1} + a_2 \cdot z^{-2} + \dots + a_n \cdot z^{-n}} \times \frac{1}{1 - z^{-1}} \quad (6)$$

Il faudrait ensuite calculer la transformée en z inverse de s(z). En pratique, si l'on doit faire le calcul à la main on décompose en éléments simples puis on utilise des tables. Avec Matlab c'est plus simple. Voici par exemple le script pour obtenir les 20 premiers échantillons de la réponse indicielle d'un système :

**Matlab**

```
% Réponse indicielle d'un système :
[y,t] =step(sysd,20);
```

### 3.2.6 Régime stationnaire (gain statique)

Le gain statique est obtenu en appliquant une entrée indicielle et en calculant la valeur du régime permanent. En appliquant le théorème de la valeur finale il vient :

$$F_0 = \lim_{z \rightarrow 1} \left( \frac{z-1}{z} \cdot \frac{b_0 + b_1 \cdot z^{-1} + b_2 \cdot z^{-2} + \dots + b_m \cdot z^{-m}}{1 + a_1 \cdot z^{-1} + a_2 \cdot z^{-2} + \dots + a_n \cdot z^{-n}} \times \frac{1}{1 - z^{-1}} \right) \quad (7)$$

Soit :

$$F_0 = \frac{\sum_{i=0}^{i=m} b_i}{1 + \sum_{i=1}^{i=n} a_i} \quad (8)$$

Le gain statique est obtenu donc simplement en faisant le quotient de la somme des coefficients.

### 3.2.7 Pôles, zéros, stabilité

Comme pour les systèmes continus la dynamique et la stabilité des systèmes dépend des propriétés du dénominateur de la fonction de transfert.

$$F(z) = \frac{e_0 \cdot (1 + e_1 \cdot z^{-1}) \cdot (1 + e_2 \cdot z^{-1}) \cdots (1 + e_m \cdot z^{-1})}{(1 + c_1 \cdot z^{-1}) \cdot (1 + c_2 \cdot z^{-1}) \cdots (1 + c_n \cdot z^{-1})} \quad (9)$$

Les coefficients  $c_i$  sont les pôles de la fonction de transfert, les coefficients  $e_i$  sont les zéros. La fonction de transfert étant au départ composée de coefficients purement réels les coefficients  $c_i$  sont soit réels soit imaginaires par paires conjuguées.

Une fonction de transfert est donc aussi bien définie par la donnée de ses pôles et de ses zéros. Le script Matlab pour créer un système ainsi est donné ci-dessous :

#### Matlab

```
% Système discret défini par ses pôles et zéros :
Te=0.1;
sysd=zpk([c1 c2 c3],[e1 e2],e0,Te);
```

Si l'on décompose la fonction de transfert en éléments simples on obtient une somme d'éléments du type :

$$F(z) = \frac{d_1}{1 + c_1 \cdot z^{-1}} + \frac{d_2}{1 + c_2 \cdot z^{-1}} + \cdots + \frac{d_n}{1 + c_n \cdot z^{-1}} \quad (10)$$

Un système linéaire temps invariant est dit stable si sa réponse impulsionnelle tend asymptotiquement vers 0. Compte tenu de (10) cela nécessite que chaque pôle  $c_i$  soit de module inférieur à 1.

**Propriété** : un système discret LTI est stable si et seulement si tous ses pôles ont leur modules inférieur ou égal à 1.

#### Critère algébrique attestant de la stabilité : critère de Routh

On rappelle que le critère de Routh permet de prédire si les racines d'un polynôme sont toutes à partie réelle négative, donc stables. Énonçons le théorème de Routh dans le cas d'un polynôme  $P'(w)$  :

$$P'(w) = a_0 + a_1 \cdot w + a_2 \cdot w^2 + \cdots + a_n \cdot w^n \quad (11)$$

Dans un premier temps on construit le tableau :

$$\left| \begin{array}{cccccc} \beta_{n,0} & \beta_{n,1} & \beta_{n,2} & \beta_{n,q-1} & \beta_{n,q} & 0 \\ \beta_{n-1,0} & \beta_{n-1,1} & \beta_{n-1,2} & \beta_{n-1,q-1} & \beta_{n-1,q-1} & 0 \\ \beta_{n-2,0} & \beta_{n-2,1} & \beta_{n-2,2} & \beta_{n-2,q-1} & \beta_{n-2,q-1} & \\ \\ \beta_{2,0} & \beta_{2,1} & 0 & & & \\ \beta_{1,0} & 0 & & & & \\ \beta_{0,0} & 0 & & & & \end{array} \right| \quad (12)$$

Puis on vérifie que tous les premiers éléments de la première colonne ont le même signe : cela garantit que tous les pôles de  $P'(w)$  ont leur partie réelle négative.

Les éléments du tableau sont construits par récurrence ligne par ligne en appliquant les formules suivantes :

Pour  $i = n$  et  $i = n - 1$  :

$$\begin{cases} \beta_{n,j} = a_{n-2j} \\ \beta_{n-1,j} = a_{n-1-2j} \end{cases} \quad (13)$$

Et pour  $i < n-1$  :

$$\beta_{i,j} = \frac{\beta_{i+1,0} \cdot \beta_{i+2,j+1} - \beta_{i+1,j+1} \cdot \beta_{i+2,0}}{\beta_{i+1,0}} \quad (14)$$

Dans le cas qui nous intéresse, un système échantillonné, on cherche plutôt à vérifier que les pôles du polynôme  $P(z)$  sont de module inférieur à un. L'idée est d'effectuer un changement de variable  $z \rightarrow w$  pour pouvoir appliquer le critère de Routh.

$$P'(w) = (1-w)^n \cdot P\left(\frac{1+w}{1-w}\right) \quad (15)$$

Si l'on effectue la transformation (15) alors le fait que  $P'(w)$  ait toutes ses racines à partie réelle négative garantit que  $P(z)$  a toutes ses racines dans le cercle unité. Pour vérifier la stabilité de  $P(z)$  il suffit donc d'effectuer le changement de variable (15) puis d'appliquer le critère de Routh conventionnel (12 13 14).

### Critère algébrique attestant de la stabilité : critère de Jury

Le critère de Jury s'apparente au critère de Routh. Nous le donnons pour un polynôme de degré 2, 3 et 4 du type :

$$P(z) = a_0 + a_1 \cdot z + a_2 \cdot z^2 + \dots + a_n \cdot z^n \quad (16)$$

Les conditions de stabilité sont données ci-dessous (ces conditions sont données dans le cas  $a_n > 0$ , il suffit de multiplier tous les termes par  $-1$  si cette condition n'est pas remplie par  $P(z)$ ) :

$$\begin{aligned}
 n = 2 & \begin{cases} a_0 + a_1 + a_2 > 0 \\ a_0 - a_1 + a_2 > 0 \\ -a_0 + a_2 > 0 \end{cases} \\
 n = 3 & \begin{cases} a_0 + a_1 + a_2 + a_3 > 0 \\ -a_0 + a_1 - a_2 + a_3 > 0 \\ -|a_0| + a_3 > 0 \\ a_0 a_2 - a_1 a_3 - a_0^2 + a_3^2 > 0 \end{cases} \\
 n = 4 & \begin{cases} a_0 + a_1 + a_2 + a_3 + a_4 > 0 \\ a_0 - a_1 + a_2 - a_3 + a_4 > 0 \\ a_4^2 - a_0^2 - |a_0 a_3 - a_1 a_4| > 0 \\ (a_0 - a_4)^2 (a_0 - a_2 + a_4) + (a_1 - a_3)(a_0 a_3 - a_1 a_4) > 0 \end{cases}
 \end{aligned} \quad (17)$$

### 3.2.8 Systèmes du premier ordre

On donne dans la Figure 4 les réponses indicielles de systèmes du premier ordre ainsi que la localisation des pôles dans le plan complexe. Le cercle de rayon unité représente la zone dans laquelle les pôles doivent se situer pour assurer la stabilité. On remarque sur les deux premières figures l'influence de la place du zéro par rapport au point  $(1,0)$  pour la rapidité.

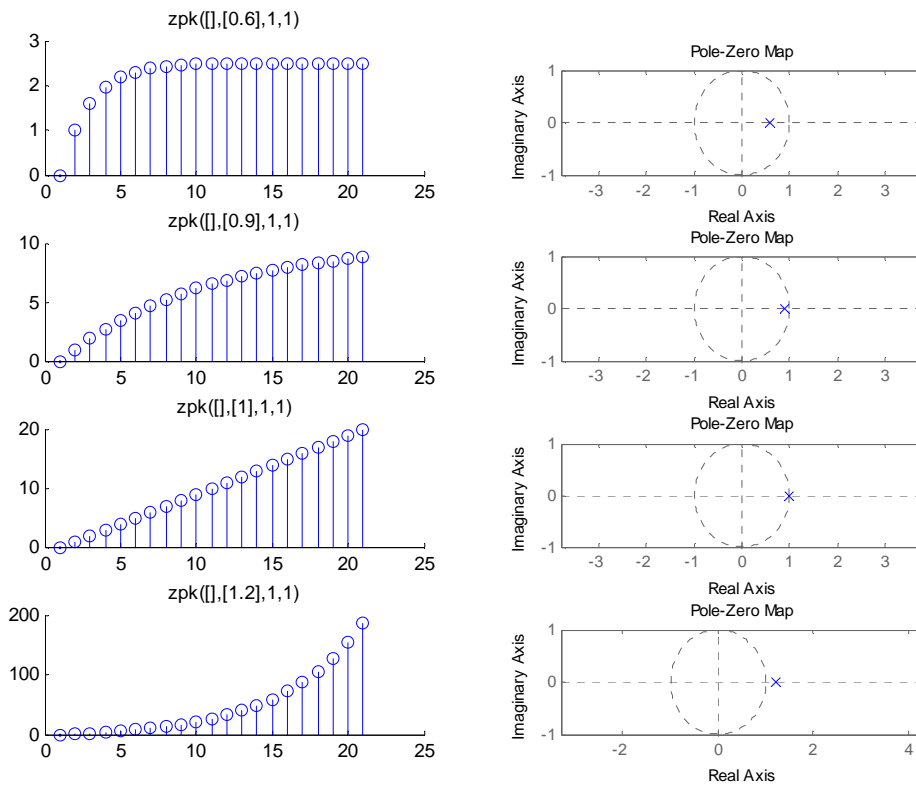


Figure 4 réponses indicielles de systèmes du premier ordre

### 3.2.9 Systèmes du second ordre

On donne dans la Figure 6 les réponses indicielles de systèmes du second ordre ainsi que la localisation des pôles dans le plan complexe. La rapidité et l'amortissement dépend de la position des pôles dans le cercle unité. La Figure 5 représente les courbes iso-amortissement et iso-pulsation des systèmes du second ordre.

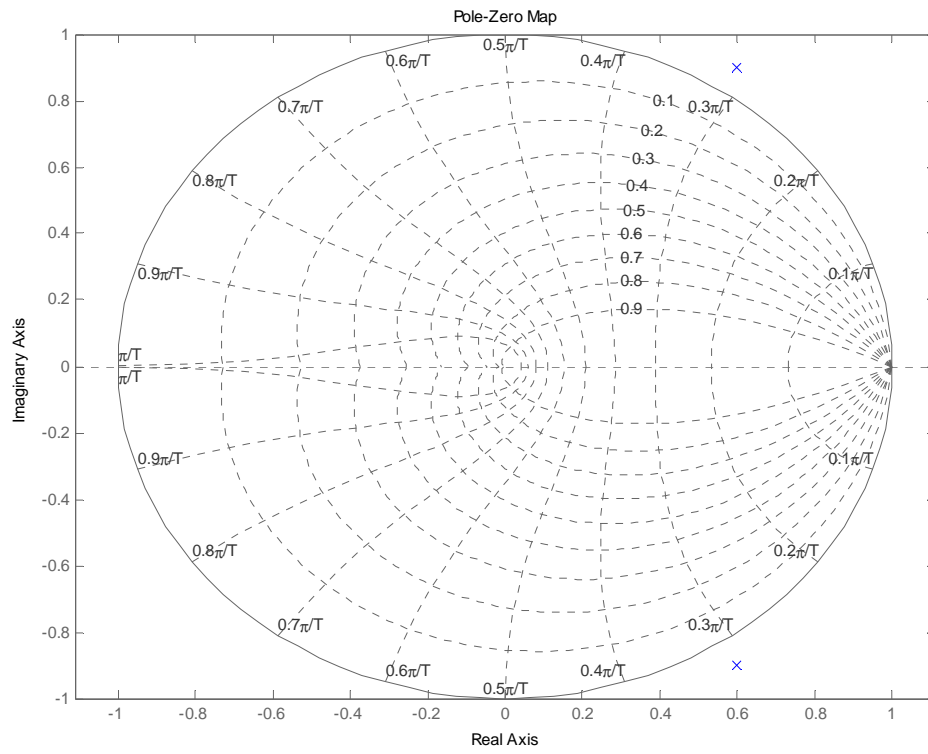


Figure 5 : courbes iso-amortissement et iso-pulsation des systèmes discrets du second ordre

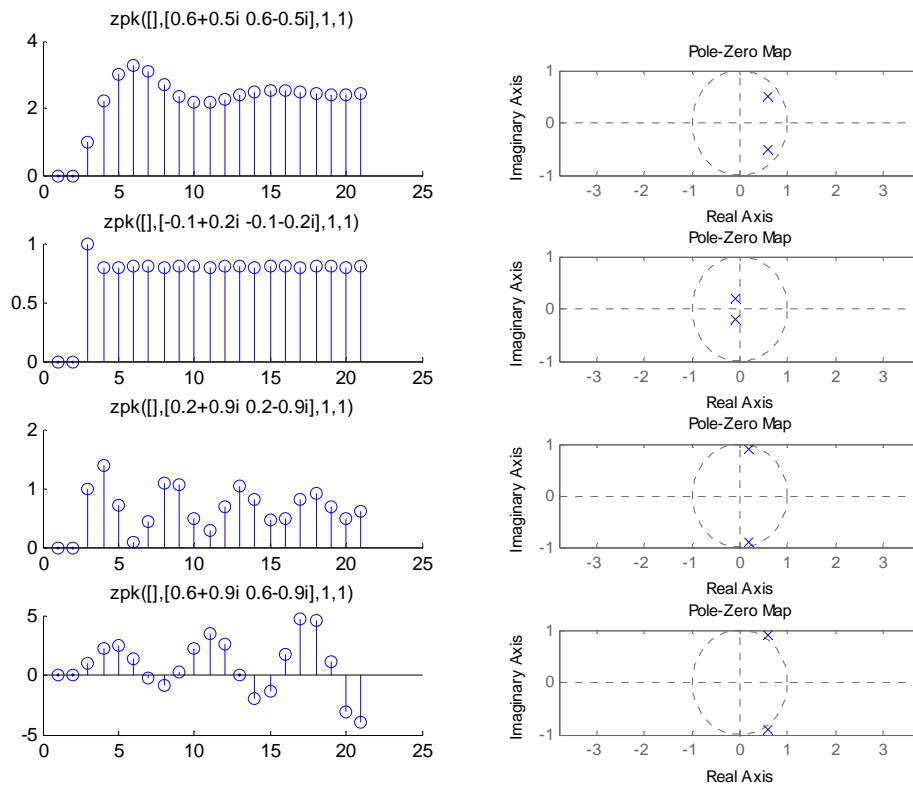


Figure 6 : réponses de systèmes du second ordre

### 3.2.10 Calcul itératif de la réponse d'un système

Si l'on dispose de la fonction de transfert d'un système et que l'on connaît les valeurs successives  $e(k)$  du signal d'entrée il est facile de calculer itérativement la sortie  $y(k)$  du système par l'équation de récurrence (2). Il faut initialiser par les  $n$  valeurs initiales de  $s(k)$ , prises généralement égales à 0.

### 3.2.11 Représentation par équation d'état

Un système discret peut aussi être représenté par un ensemble d'équations d'état comme dans le domaine continu (52). Cette représentation est naturellement adaptée aux systèmes multi-entrées multi-sorties. On définit alors un état interne  $x(k)$  ainsi qu'une fonction d'évolution  $f$  et une équation de mesure  $h$ .

$$\begin{cases} x(k+1) = f(x(k), u(k)) \\ s(k) = h(x(k), u(k)) \end{cases} \quad (18)$$



### 3.3 Réponse fréquentielle des systèmes échantillonnés

#### 3.3.1 Diagramme de Bode d'un système échantillonné

Lorsqu'on étudie la réponse fréquentielle des systèmes continus on considère des entrées particulières sinusoïdales :

$$\begin{cases} e(t) = \cos(2 \cdot \pi \cdot f \cdot t) + j \cdot \sin(2 \cdot \pi \cdot f \cdot t) = e^{j \cdot 2 \cdot \pi \cdot f \cdot t} \\ f \in [0 \quad \infty] \end{cases} \quad (19)$$

De même, pour des systèmes échantillonnés on considèrera des entrées sinusoïdales échantillonnées :

$$\begin{cases} e(k) = \cos(2 \cdot \pi \cdot f \cdot k) + j \cdot \sin(2 \cdot \pi \cdot f \cdot k) = e^{j \cdot 2 \cdot \pi \cdot f \cdot k} \\ f \in [0 \quad 1] \end{cases} \quad (20)$$

La sortie est aussi un signal échantillonné sinusoïdal de gain  $A(f)$  et de phase  $\Phi(f)$  variables et dépendant de  $f$ ; la fréquence réduite.

$$\begin{cases} s(k) = A(f) \cdot e^{j \cdot (2 \cdot \pi \cdot f \cdot k + \Phi(f))} \\ f \in [0 \quad 1] \end{cases} \quad (21)$$

Le nombre complexe, noté abusivement  $F(f)$ , d'amplitude  $A(f)$  et de phase  $\Phi(f)$  est obtenu à partir de  $F(z)$  en remplaçant  $z$  par  $e^{j \cdot 2 \cdot \pi \cdot f}$  :

$$\begin{cases} F(f) = F(z = e^{j \cdot 2 \cdot \pi \cdot f}) \\ f \in [0 \quad 1] \end{cases} \quad (22)$$

Le diagramme représentant le gain et la phase en fonction de la fréquence réduite s'appelle aussi un diagramme de Bode. L'échelle des fréquences est généralement linéaire de 0 à 0.5 (puisque le diagramme est nécessairement symétrique). La Figure 7 représente par exemple le diagramme de Bode d'un système du second ordre.

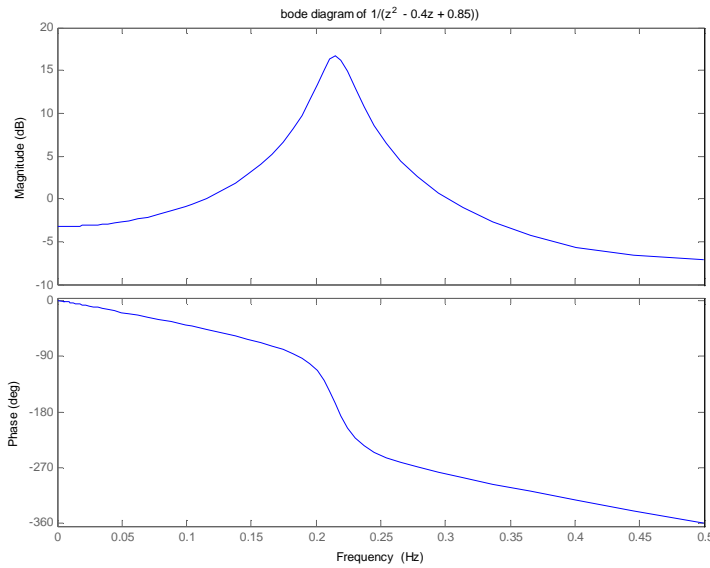


Figure 7 : exemple du diagramme de Bode d'un système discret du second ordre

### 3.3.2 Où l'on voit apparaître le théorème de Shannon

Le changement de variable  $z = e^{j2\pi f}$  n'est pas bijectif puisque  $e^{j2\pi f} = e^{j2\pi(f+k)}$  avec  $k$  entier. Si l'on fait le lien entre domaine continu et domaine fréquentiel on peut remarquer que le changement de variable  $z = e^{j2\pi f}$  fait correspondre une zone du plan complexe continu à une zone du plan complexe discret. La transformation est bijective si l'on se contente de la zone du plan complexe continu contenu entre les droites d'ordonnée 0 et 1.

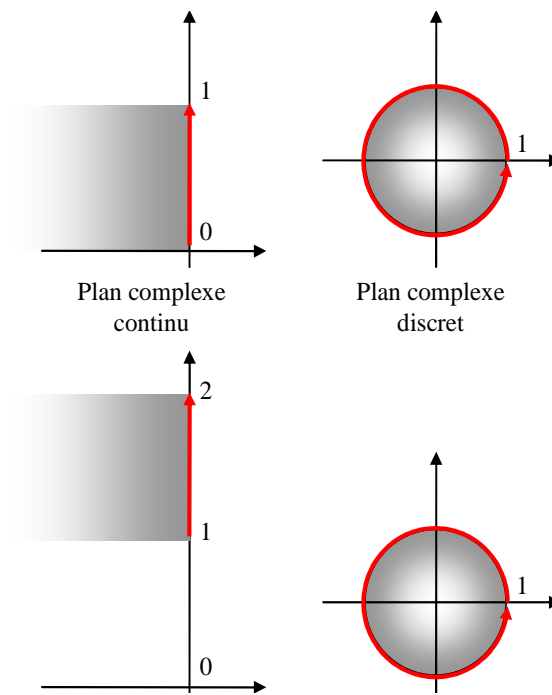


Figure 8 : correspondance entre plan complexe continu et plan complexe discret.



## 4. SYSTEMES CONTINUS ECHANTILLONNES

### 4.1 Structure des systèmes commandés par ordinateur

Les systèmes commandés par ordinateurs sont presque toujours des systèmes physiques continus par nature. Le ordinateur traite par contre les entrées et les sorties de manière discrète avec une période d'échantillonnage  $T_e$ . Les éléments importants, du point de vue de l'automaticien sont donc :

- le convertisseur analogique / numérique CAN ou du moins le système qui fournit séquentiellement les mesures au ordinateur à la période  $T_e$ .
- le ordinateur lui-même. Il doit effectuer l'acquisition des mesures, le traitement des informations, l'envoi des consignes dans un temps inférieur à  $T_e$ .
- le convertisseur numérique / analogique CNA.
- le système physique.

On suppose, pour faire simple et pour pouvoir utiliser la panoplie des outils mathématiques, que les systèmes sont échantillonnés à la même période  $T_e$  et au même moment.

La Figure 9 montre le système physique commandé par le ordinateur, dans sa version simpliste.

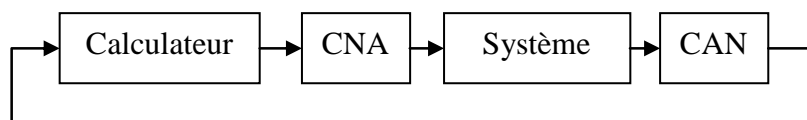


Figure 9 : système commandé par ordinateur

Au XXI<sup>ème</sup> siècle les choses sont quand même parfois un petit peu plus compliquées :

- le ordinateur fournit bien souvent un signal de commande numérique (transporté par bus informatique USB, RS232, CAN, Ethernet, etc...) directement interprété par l'actionneur. On parle "d'actionneur intelligent".
- de même le "capteur intelligent" fournit aussi bien souvent la mesure sous forme d'un signal numérique.
- Les signaux sont transportés par des bus informatiques qui n'ont pas que ça à faire : il peut y avoir du retard.
- de même les capteurs et actionneurs sont rarement tous seuls, ce qui se traduit par des périodes d'échantillonnage différentes, et parfois non synchrones...

- enfin le ordinateur peut aussi avoir d'autres choses à faire...

Ceci étant dit nous allons pour le moment nous restreindre au cas représenté sur la Figure 9 avec des instants d'échantillonnage synchrones et régulièrement espacés d'un temps  $T_e$ .

## 4.2 Echantillonnage des systèmes continus

On dispose généralement de la fonction de transfert continue (notée en utilisant la variable complexe  $s$  ou  $p$ )  $F(s)$  du système continu. Le problème est d'obtenir la fonction de transfert discrète du système échantillonné de manière à pouvoir synthétiser un correcteur. Le système continu est associé au convertisseur numérique – analogique qu'il faut prendre en compte.

Les convertisseurs numérique analogiques peuvent généralement se représenter par des "échantillonneurs – bloqueurs d'ordre 0" ou BOZ. Cela signifie que la commande est maintenue à une valeur constante pendant toute la durée du temps d'échantillonnage.

Si l'on applique une commande discrète impulsionnelle à un BOZ il va maintenir la commande 1 pendant la durée  $T_e$  puis revenir à zéro (Figure 10).

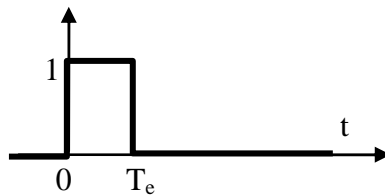


Figure 10 : réponse impulsionnelle d'un bloqueur d'ordre 0

La réponse impulsionnelle  $BOZ(t)$  est donc composée d'un échelon unitaire suivi d'un échelon unitaire de signe opposé et décalé d'un temps  $T_e$ . La transformée de Laplace de la réponse impulsionnelle du bloqueur est donnée par la formule (23) :

$$BOZ(s) = \frac{1}{s} - \frac{e^{-jT_e}}{s} \quad (23)$$

Du point de vue du ordinateur il faut donc prendre en compte le système continu de fonction de transfert  $F_{BOZ}(s)$  que l'on devra échantillonner :

$$F_{BOZ}(s) = \frac{1 - e^{-jT_e}}{s} \cdot F(s) \quad (24)$$

$$F_{\text{BOZ}}(z) = Z^{-1} \left( \frac{1 - e^{-jT_e}}{s} \cdot F(s) \right) \quad (25)$$

En pratique l'obtention de la transformée en z d'un système échantillonné par bloqueur d'ordre 0 se fait à l'aide de tables. On donne quelques exemples fondamentaux dans le Tableau 3 ci-dessous.

Tableau 3 : transformée en z de systèmes continus échantillonnés

F(s)	F <sub>BOZ</sub> (z)
1	1
$\frac{1}{s}$	$\frac{T_e \cdot z^{-1}}{1 - z^{-1}}$
$\frac{1}{1 + T \cdot s}$	$\begin{cases} \frac{b_1 \cdot z^{-1}}{1 + a_1 \cdot z^{-1}} \\ b_1 = 1 - e^{-T_e/T} \quad a_1 = -e^{-T_e/T} \end{cases}$
$\frac{e^{-Ls}}{1 + T \cdot s}$	$\begin{cases} \frac{b_1 \cdot z^{-1} + b_2 \cdot z^{-2}}{1 + a_1 \cdot z^{-1}} \quad a_1 = -e^{-T_e/T} \\ b_1 = 1 - e^{-(L-T_e)/T} \quad b_2 = e^{-T_e/T} (e^{L/T} - 1) \end{cases}$
$L < T_e$	

Si l'on dispose de Matlab c'est bien sûr encore plus simple. On utilise la fonction c2d avec l'argument 'zoh' pour "zero holder hold" (bloqueur d'ordre zéro).

#### Matlab

```
% Définition d'un système continu :
sysc=tf(1,[1 1]);
% Obtention de la transformée en z du système échantillonné à Te
Te=0.1;
sysd=c2d(sysc,Te,'zoh');
% Notation en z^-1
sysd.variable='z'
```

### 4.3 Effets de l'échantillonnage

Le convertisseur numérique a deux effets (néfastes) importants : l'effet de retard et l'effet de quantification.

L'effet de retard se voit dans (24) : on peut dire en première approximation que l'échantillonnage se traduit par un retard pur de  $T_e/2$ . Cela a donc, si ce temps

d'échantillonnage est non négligeable par rapport aux constantes de temps du système, un effet déstabilisant.

Pour illustrer l'effet de quantification supposons par exemple que l'on utilise un convertisseur analogique/numérique 8 bits 0-10V. Le plus petit incrément que le convertisseur peut distinguer est de  $10V/2^8 = 40mV$ . Si le signal utile varie physiquement entre 5V et 5,4V on n'aura que dix incréments utilisables... Ce phénomène n'entre pas dans la panoplie des phénomènes modélisables par la transformée en z. En pratique il est donc souvent nécessaire d'étudier le système et de synthétiser la commande numérique en négligeant ce phénomène puis de vérifier a posteriori qu'il n'a pas un effet trop néfaste.

#### 4.4 Où l'on revoit le théorème de Shannon

Le théorème de Shannon est bien connu en ce qui concerne l'échantillonnage de signaux discrets : on sait que le signal ne doit pas comporter de composante de fréquence supérieure à  $f_e/2$  si l'on ne veut pas observer de repliement. On doit aussi prendre des précautions si l'on veut travailler sur la fonction de transfert discrète correspondant à un système continu échantillonné.

On va traiter un exemple concret pour mettre en évidence le problème de repliement dans le cas de la commande des systèmes continus. On s'intéresse à deux systèmes continus du second ordre ayant des pôles de même partie réelle mais de parties imaginaires séparées de  $2\pi T_e$ . Le script Matlab ci-dessous montre la création des deux systèmes :

##### Matlab

```
% Définition de deux systèmes continus :  
sys1=zpk([],[-1-2i -1+2i],2^2+1);  
sys2=zpk([],[-1-(2+2*pi)*i -1+(2+2*pi)*i],1+(2+2*pi)^2);
```

On voit sur la Figure 11 que la réponse indicielle des deux systèmes est bien différente.



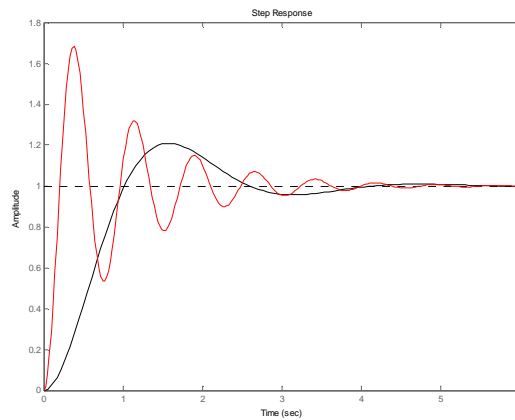


Figure 11 : réponse indicielle de deux systèmes du second ordre

Si l'on calcule la fonction de transfert échantillonnée les deux systèmes à la période de 1 seconde on obtient deux réponses indicielles très proches (représentés par des  $\circ$  sur la Figure 12). Conclusion : la période d'échantillonnage était mal choisie

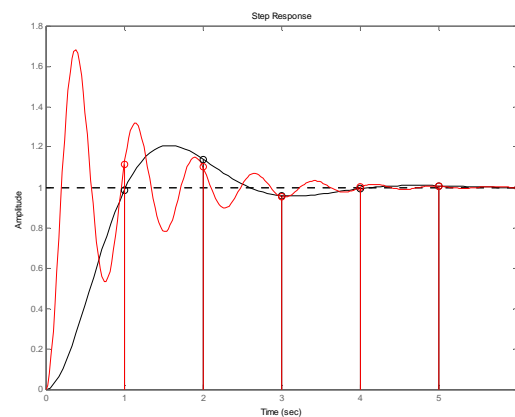


Figure 12 : réponse indicielle de deux systèmes du second ordre échantillonnés à 1s

En pratique la commande échantillonnée de systèmes continus doit être associée à certaines précautions :

- le système ne doit pas comporter de dynamiques propres situées dans les fréquences supérieures à  $f_c/2$  (exemple illustré dans ce chapitre).
- les signaux échantillonnés doivent être "nettoyés" des fréquences supérieures à  $f_c/2$  avec un filtre anti-repliement. (A moins que le signal soit suffisamment propre et ne comporte pas de composantes significatives au-delà de ces fréquences).

### Choix de la période d'échantillonnage

Le choix de la période d'échantillonnage est très important et dépend de la dynamique des signaux. Attention, les systèmes commandés en boucle fermée ont souvent une dynamique plus rapide que le système en boucle ouverte : il faut donc prendre en compte la dynamique en boucle fermée pour le choix de la fréquence d'échantillonnage.

En pratique on préconise de prendre une fréquence 5 à 25 fois supérieure à la bande passante souhaitée en boucle fermée.

On peut être tenté de prendre une fréquence d'échantillonnage très élevée si la technologie le permet (ordinateur et convertisseurs suffisamment rapide). Ceci peut amener rapidement à des problèmes numériques. Imaginons par exemple le cas d'un système ayant une dynamique très lente (quelques secondes) et un système de mesure codant les mesures sur 10 bits (cas classique). La Figure 13 illustre cet exemple. Le signal varie peu entre deux instants d'échantillonnage et les variations du signal sont faibles par rapport à la valeur de l'incrément numérique (figure du haut). Le calcul de la simple dérivée numérique se traduit donc par une succession de  $-1$   $0$  et  $+1$ .

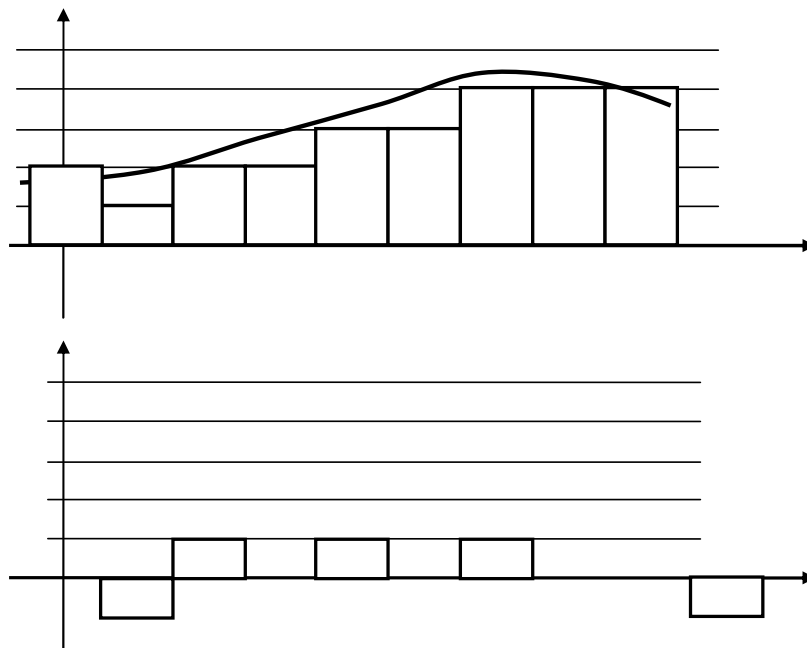


Figure 13 : période d'échantillonnage trop faible. En haut : signal échantillonné et discrétisé ; en bas : calcul de la dérivée numérique.

### Filtre anti-repliement

En toute rigueur chaque fois qu'un signal est échantillonné il faudrait faire précéder le convertisseur numérique d'un filtre analogique anti-repliement éliminant physiquement toutes les composantes du signal de fréquence supérieure à  $f_e/2$ .

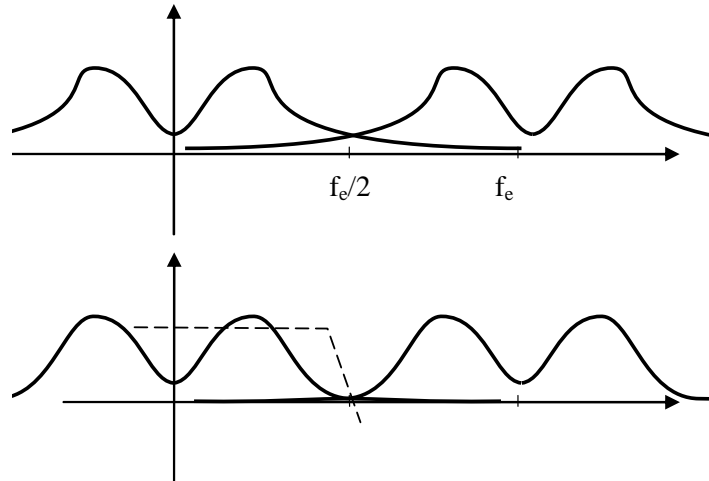


Figure 14 : Effet du filtre anti-repliement

Bien souvent la période d'échantillonnage de l'asservissement est relativement lente (quelques secondes) et il est alors impossible de réaliser un filtre de constante de temps aussi faible. On procède alors à un sur-échantillonnage : le signal est filtré, échantillonné à haute fréquence  $T'_e$  puis filtré numériquement et sous-échantillonné à la période  $T_e$  beaucoup plus grande que  $T'_e$  comme illustré sur la

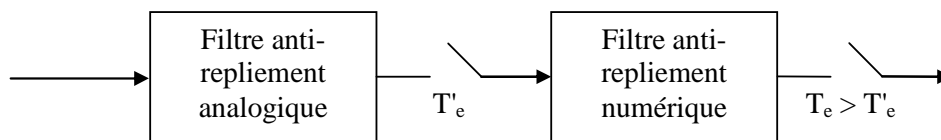


Figure 15 : Sur échantillonnage

Un simple filtre numérique de type moyenne mobile est souvent suffisant.

### 4.5 Discrétisation de fonction de transfert continues

Une manière classique pour réaliser un contrôle par ordinateur est de commencer par synthétiser un correcteur continu avec les outils de l'automatique classique puis de trouver un correcteur discret qui "approche" convenablement le correcteur continu.

#### 4.5.1 Approximation numérique d'équation différentielle

Supposons que le correcteur continu que l'on a synthétisé soit du type PD :

$$\text{PID}(s) = \frac{e(s)}{\varepsilon(s)} = k_p + k_D \cdot s \quad (26)$$

Il s'agit de calculer les commandes  $e(k)$  successives issues du correcteur en fonction des erreurs  $\varepsilon(k)$ . Une approche classique et intuitive est d'approximer la dérivée par une différence :

$$e(k) = k_p \cdot \varepsilon(k) + k_D \cdot \frac{\varepsilon(k) - \varepsilon(k-1)}{T_e} \quad (27)$$

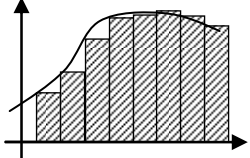
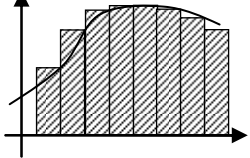
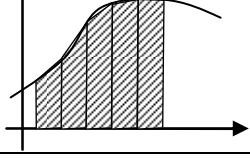
Si on calcule la transformée en  $z$  de  $e(k)$  on obtient :

$$e(z) = k_p \cdot \varepsilon(z) + k_D \cdot \frac{\varepsilon(z) - z^{-1}\varepsilon(z)}{T_e} = \left( k_p + k_D \cdot \frac{1 - z^{-1}}{T_e} \right) \cdot \varepsilon(z) \quad (28)$$

On retrouve la fonction de transfert PD(s) dans laquelle on a effectué le changement de variable  $s = \frac{1-z^{-1}}{T_e}$

De manière plus générale, l'approximation de fonction de transfert revient à effectuer l'approximation numérique d'intégration. Selon la méthode d'intégration choisie on effectue des changements de variable différents.

Tableau 4 : méthodes d'intégration

Méthode	Changement de variable	
Euler progressif (Euler's forward method)	$s = \frac{1 - z^{-1}}{z^{-1} \cdot T_e}$	
Euler rétrograde (Euler's backward method)	$s = \frac{1 - z^{-1}}{T_e}$	
Méthode des trapèzes, ou bilinéaire, ou de Tustin	$s = \frac{2}{T_e} \cdot \frac{1 - z^{-1}}{1 + z^{-1}}$	

En pratique le changement de variable  $s = \frac{1 - z^{-1}}{z^{-1} \cdot T_e}$  n'est pas sans risque. On donne sur la Figure 16 la correspondance entre la zone stable du plan complexe en  $s$  et la zone correspondante après la transformation (Euler progressif, Euler rétrograde, bilinéaire). On voit notamment qu'il est possible, partant d'un système continu stable, d'obtenir des pôles dans le domaine discret en dehors du cercle de stabilité si la transformation appliquée est celle d'Euler progressif.

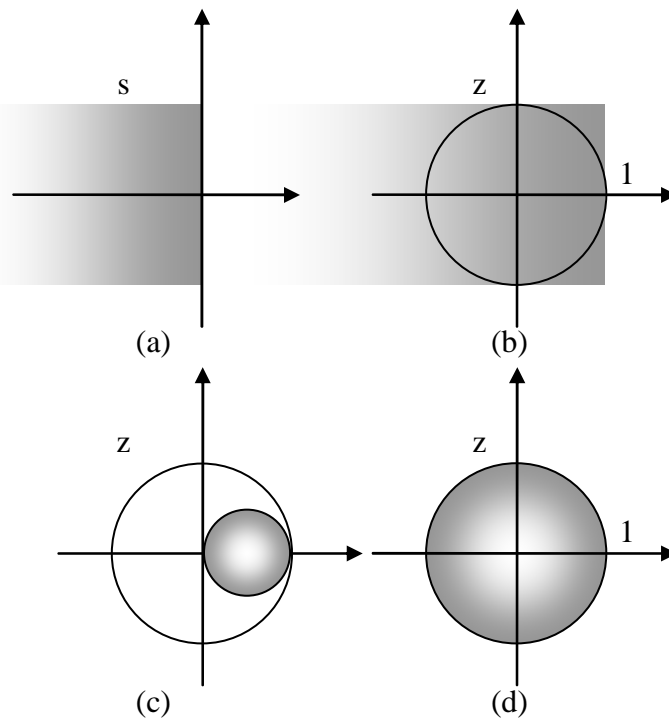


Figure 16 : correspondance des plans complexes. (a) : zone de stabilité dans le plan complexe continu en  $z$  ; (b) zone correspondante avec l'approximation d'Euler progressif ; (c) zone correspondante avec l'approximation d'Euler rétrograde ; (d) zone correspondante avec l'approximation bilinéaire.

La fonction `c2d` de Matlab permet d'obtenir l'équivalent discret d'une fonction de transfert continue avec la méthode d'intégration comme argument :

```

Matlab
sysc=tf([1 0],[1 2 5]);
Te=0.1;
% Méthode d'euler :
sysd1=c2d(sysc,Te,'zoh');
% Méthode des trapèzes :
sysd2=c2d(sysc,Te,'foh');
% Méthode bilinéaire :
sysd3=c2d(sysc,Te,'tustin');
    
```

#### 4.5.2 Superposition des réponses indicielles

Une autre méthode pour approximer une fonction de transfert discrète est de s'assurer que la réponse indicielle de sa version discrétisée correspond exactement lors des temps d'échantillonnage.

### 4.5.3 Correspondances pôles - zéros

La dernière méthode est de partir d'une décomposition de la fonction de transfert continue en pôles et zéros et d'effectuer la conversion pôles à pôles et zéros par zéros. La correspondance, selon que le pôle est réel ou complexe est donnée ci-dessous :

$$\begin{aligned} (s + a) &\rightarrow (z - e^{-a \cdot T_e}) \\ (s + a)^2 + b^2 &\rightarrow z^2 - 2 \cdot e^{-a \cdot T_e} \cdot \cos(b \cdot T_e) \cdot z + e^{-2 \cdot a \cdot T_e} \end{aligned} \quad (29)$$

### 4.5.4 Bilan

Aucune méthode n'est parfaite et on n'aura jamais en même temps :

- même réponse indicielle
- même réponse impulsionnelle
- approximation (Euler, Tustin, etc...) de l'équation différentielle
- correspondance des pôles et des zéros
- correspondance des réponses fréquentielles

Ceci étant dit le choix de la période d'échantillonnage est primordial. Si la fréquence d'échantillonnage est beaucoup plus faible que la dynamique de la fonction de transfert à approximer la différence entre chaque méthode sera minimale.





## 5. IDENTIFICATION DES SYSTEMES ECHANTILLONES

Dans le présent chapitre on donne les bases des méthodes d'identification. Pour une approche complète, voir le cours de troisième année "Identification avion" [6] et les ouvrages [1] et [7].

Même si l'on dispose d'un modèle théorique du processus que l'on souhaite commander il est presque toujours nécessaire de l'affiner en procédant à des essais sur le système réel. Bien souvent le seul modèle dont l'on dispose est ce modèle issu de l'expérimentation. Différentes méthodes s'appliquent selon l'objectif recherché et le niveau de précision exigé. On distinguera :

- les modèles paramétriques
- les modèles simples
- les modèles non paramétriques

### 5.1 Identification d'un modèle ARX par la méthode des moindres carrés récursifs

A partir de cet exemple, le plus simple qui soit, on pose les bases de toutes les méthodes d'identification.

On considère un système d'entrée  $u(t)$  et de sortie  $y(t)$  représenté par l'équation de récurrence suivante :

$$y(t) + a_1 \cdot y(t-1) + a_2 \cdot y(t-2) + \dots + a_n \cdot y(t-n) = b_0 \cdot u(t) + b_1 \cdot u(t-1) + b_2 \cdot u(t-2) + \dots + b_m \cdot u(t-m) \quad (30)$$

L'objectif est, à partir d'essais et de mesures, d'identifier les meilleures valeurs possibles pour les coefficients  $a_i$  et  $b_i$ .

L'équation (30) peut être réécrite pour mettre en évidence la *prédiction* de la sortie  $y(t)$  à partir des mesures passées :

$$y(t) = -a_1 \cdot y(t-1) - a_2 \cdot y(t-2) - \dots - a_n \cdot y(t-n) + b_0 \cdot u(t) + b_1 \cdot u(t-1) + b_2 \cdot u(t-2) + \dots + b_m \cdot u(t-m) \quad (31)$$

Soit :

$$y(t) = \varphi^T(t) \cdot \theta(t)$$

avec :

$$\begin{cases} \varphi^T(t) = [-y(t-1) \ \dots \ -y(t-n) \ , \ u(t-1) \ \dots \ u(t-m)] \\ \theta(t) = [a_1 \ \dots \ a_n \ , \ b_1 \ \dots \ b_m] \end{cases} \quad (32)$$

Le but est de trouver le meilleur ensemble de paramètres  $\theta(t)$  : celui qui, à partir d'un ensemble de mesures  $\varphi(t)$  permet de prédire avec la meilleure précision la sortie effectivement mesurée.

Supposons qu'à la suite d'une expérience on dispose d'un ensemble de mesures  $\{y(t), u(t)\}_{t=1:N}$  et que l'ensemble des paramètres  $\theta$  n'est pas bien connu. On construit la séquence des *sorties prédites* :

$$\hat{y}(t) = \varphi^T(t) \cdot \theta(t) \quad (33)$$

Le paramètre  $\theta$  est celui qui minimise une norme quadratique :

$$\theta = \arg \max \left( \frac{1}{N} \sum_{t=1}^N (y(t) - \hat{y}(t))^2 \right) = \arg \max \left( \frac{1}{N} \sum_{t=1}^N (y(t) - \varphi^T(t) \cdot \theta(t))^2 \right) \quad (34)$$

Compte tenu du fait que l'équation est linéaire au départ et que le terme à minimiser est du type quadratique la solution est simplement obtenue par une pseudo-inverse :

$$\theta = \left[ \sum_{t=1}^N (\varphi(t) \cdot \varphi^T(t)) \right]^{-1} \sum_{t=1}^N (\varphi(t) \cdot y(t)) \quad (35)$$

On obtient donc la solution classique des moindres carrés. La formule (35) est le plus souvent utilisée sous sa forme récursive :

$$\theta(t) = \theta(t-1) + F(t) \cdot \varphi(t) \cdot (y(t) - \theta^T(t-1) \cdot \varphi(t))$$

avec

$$F(t)^{-1} = F(t-1)^{-1} + \varphi(t) \cdot \varphi(t)^T \quad (36)$$

ou

$$F(t) = F(t-1) - \frac{F(t-1) \cdot \varphi(t) \cdot \varphi(t)^T \cdot F(t-1)}{1 + \varphi(t)^T \cdot F(t-1) \cdot \varphi(t)}$$

(On laisse au lecteur le plaisir d'effectuer lui-même les démonstrations, qu'il peut du reste trouver dans [1] et [7]).

## 5.2 L'identification paramétrique en général et en pratique

L'exemple précédent a permis de mettre en évidence tous les aspects de la méthodologie classique de l'identification :

- il a fallu choisir un ensemble de mesures  $\{y(t), u(t)\}_{t=1:N}$ . Le choix du signal d'excitation  $u(t)$  est en pratique primordial,
- il a fallu choisir une structure pour le modèle à identifier (ici les ordres  $n$  et  $m$ ),
- il a fallu choisir un critère (ici un critère quadratique sur l'erreur entre la sortie et la sortie prédite),
- il a fallu choisir un algorithme et le paramétrer (ici l'algorithme des moindres carrés récursifs qu'il a fallu initialiser)

Enfin un point non traité dans le chapitre précédent concerne la validation :

- il aurait fallu déterminer une méthode pour comparer différents modèles et choisir le meilleur

### 5.2.1 Choix du signal d'excitation, traitement des mesures

De manière générale le système que l'on doit identifier doit être excité dans tout le domaine fréquentiel qui nous intéresse. Le signal classiquement utilisé est une séquence pseudo binaire aléatoire (Pseudo Random Binary Signal) tel celui représenté sur la Figure 17.

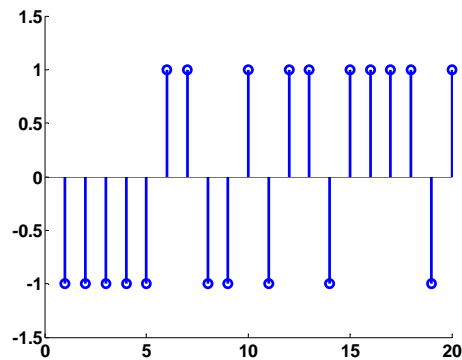


Figure 17 : exemple d'une séquence PRBS

Le signal d'excitation doit se superposer au signal de commande correspondant au point de fonctionnement étudié. Son amplitude doit être faible par rapport à l'amplitude des commandes du système (quelques %).

Une fois le système excité et les mesures effectuées il faut supprimer les valeurs moyennes des mesures.

### 5.2.2 Choix d'une structure de modèle

Le modèle étudié dans le chapitre précédent est de type ARX. On peut le noter sous forme polynomiale :

$$A \cdot y = B \cdot u \quad (37)$$

Ce modèle correspond en fait à un parti-pris sur la manière dont on traite les perturbations. En réalité on prend en compte un signal d'erreurs possibles  $e$  (de moyenne nulle) tels que :

$$A \cdot y = B \cdot u + e \quad (38)$$

Si l'on fait cette hypothèse de modélisation alors la méthode optimale est celle des moindres carrés. Malheureusement cette manière de prendre en compte les perturbations est bien restrictive. D'autres modélisations sont possibles, adaptées à d'autres algorithmes, par exemple le modèle OE :

$$y = \frac{B}{F} \cdot u + e \quad (39)$$

Il est aussi possible de "structurer" le modèle de bruit comme par exemple dans les modèles ARMAX ou BJ :

$$\begin{aligned} A \cdot y &= B \cdot u + C \cdot e \\ y &= \frac{B}{F} \cdot u + \frac{C}{D} \cdot e \end{aligned} \quad (40)$$

La structure du modèle étant choisie il faut choisir l'ordre de ses différents polynômes.

### 5.2.3 Choix d'un critère à minimiser

Dans la plupart des cas le critère à minimiser est l'erreur quadratique entre la sortie et la sortie prédite à un coup (one step ahead prediction). Ceci n'est bien évidemment pas restrictif et peut être généralisé à la prédiction à N coups.

### 5.2.4 Choix d'un algorithme

Même si chaque structure de modèle est généralement associée à un algorithme "optimal" celui-ci doit généralement se paramétrer.

Prenons par exemple l'algorithme des moindres carrés récursifs tel que présenté dans (36). Cette équation peut être modifiée en intégrant un "facteur d'oubli"  $\lambda$  :

$$\begin{aligned} \theta(t) &= \theta(t-1) + F(t) \cdot \varphi(t) \cdot (y(t) - \theta^T(t-1) \cdot \varphi(t)) \\ \text{avec} & \\ F(t)^{-1} &= \lambda \cdot F(t-1)^{-1} + \varphi(t) \cdot \varphi(t)^T \end{aligned} \quad (41)$$

Ce paramètre (typiquement entre 0.95 et 0.99) permet, comme son nom l'indique, "d'oublier" les anciennes mesures et de se concentrer sur les mesures les plus récentes.

### 5.2.5 Validation

Une fois l'algorithme d'identification appliqué et le modèle obtenu (ce qui prend généralement quelques secondes avec un ordinateur moderne) il reste à expertiser celui-ci. Selon la "qualité" du modèle obtenu on remet en cause l'une ou l'autre des étapes (signal d'excitation, structure du modèle, algorithme) et on réitère jusqu'à obtenir un modèle satisfaisant.

En pratique on essaie un certain nombre de modèles avant de choisir le "meilleur".

Le premier critère de qualité est la capacité du modèle à prédire effectivement la sortie (sortie prédite à un coup, sortie prédite à N coups). Il faut pour cela utiliser un jeu de mesures non utilisé pour l'identification. (En pratique le jeu de mesures initiales est coupé en deux : une partie pour l'identification et une partie pour la validation).

Le second critère de qualité est l'analyse du résidu : la composition spectrale de l'erreur de prédiction.

### 5.2.6 Utilisation de logiciels

En pratique il est commode d'utiliser un célèbre outil de calcul scientifique et sa toolboxe d'identification.

Comme les calculs proprement dit ne prennent que quelques secondes il est facile de réaliser un grand nombre d'identifications et de stocker les différents modèles pour les comparer ensuite. On est rapidement noyé sous le grand nombre de modèles possibles...

## 5.3 Identification "simple"

Il est souvent suffisant d'obtenir un modèle "simple" de type premier ordre et un simple essai indiciel convient souvent. Le système est excité avec un échelon unité et on cherche le meilleur modèle du premier ordre correspondant.

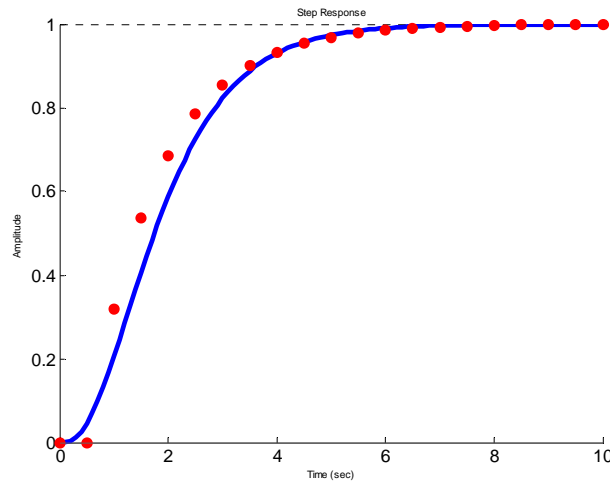


Figure 18 : identification sur un essai indiciel

Dans l'exemple de la Figure 18 on superpose le résultat d'une mesure (en bleu) avec la simulation d'un modèle discret avec une période de 0.5s. Le modèle cherché est du premier ordre avec un retard pur  $d$  :

$$y(z) = \frac{(1 + c_1) \cdot z^{-d}}{1 + c_1 \cdot z^{-1}} \cdot u(z) \quad (42)$$

Il s'agit de trouver l'entier  $d$  et le nombre réel  $c_1$  tels que la réponse indiciale du modèle corresponde pour le mieux aux mesures issues de l'essai. Ceci peut se formaliser par exemple de la manière suivante :

$$(c_1 \quad d_1) = \arg \min \sum_{t=1}^N (y_{\text{modèle}}(t) - y_{\text{mesure}}(t))^2 \quad (43)$$

Cette optimisation peut se faire à la main ou grâce à une procédure automatisée (avec le célèbre outil de calcul scientifique).

Dans le cas où le système comprend naturellement une intégration (par exemple la commande d'un moteur à courant continu par une tension et la mesure de l'angle de rotation) il est utile de prendre en compte explicitement cette intégration dans le modèle :

$$y(z) = \frac{1}{1 - z^{-1}} \cdot \frac{(1 + c_1) \cdot z^{-d}}{1 + c_1 \cdot z^{-1}} \cdot u(z) \quad (44)$$

La procédure est la même mais au lieu de prendre en compte la sortie  $y(t)$  on prend en compte la dérivée  $y'(t)$  :

$$y'(t) = y(t) - y(t-1) \quad (45)$$

## 5.4 Identification non paramétrique

Pour terminer ce panorama rapide sur les méthodes d'identification il faut parler des méthodes non paramétriques : celles qui permettent de tracer la réponse impulsionnelle ou la réponse fréquentielle du système.

### 5.4.1 Estimation de la réponse impulsionnelle

La réponse impulsionnelle d'un système peut être obtenue simplement en appliquant une entrée de type "impulsion" au système et d'en mesurer les conséquences.

La seconde méthode consiste à utiliser la méthode des moindres carrés (ou une méthode dérivée) comme vue au chapitre 5.1. La réponse impulsionnelle d'un système peut en effet être vue comme la fonction de transfert particulière où le dénominateur est de degré nul et le numérateur comporte un nombre infini (en pratique : grand) d'éléments ( $n = 0$  et  $m$  grand) :

$$y(t) = b_0 \cdot u(t) + b_1 \cdot u(t-1) + b_2 \cdot u(t-2) + \dots + b_m \cdot u(t-m) \quad (46)$$

La dernière méthode, qui sera développée ici est l'analyse d'intercorrélation.

Par définition l'intercorrélation  $R_{yu}$  entre deux signaux est :

$$R_{yu}(\tau) = E[(y(t) - E(y(t))) \cdot (u(t-\tau) - E(u(t-\tau)))] \quad (47)$$

Dans cette formule l'espérance mathématique peut être approchée grâce à la loi des grands nombres :

$$R_{yu}(\tau) \approx R_{yu}^N(\tau) = \lim_{N \rightarrow \infty} \frac{1}{N} \sum_{t=1}^N [(y(t) - m_y) \cdot (u(t-\tau) - m_u)] \quad (48)$$

On s'intéresse maintenant à un système tel que décrit par l'équation impulsionnelle (46) et on suppose que le signal d'entrée est un bruit blanc. Le signal d'entrée vérifie donc :

$$\begin{cases} \tau \neq 0 & R_{uu}(\tau) = 0 \\ \tau = 0 & R_{uu}(\tau) = R \end{cases} \quad (49)$$

Il vient alors naturellement :

$$b_k \approx \frac{1}{R} R_{yu}(k) \quad (50)$$

### 5.4.2 Estimation de la réponse fréquentielle

On utilise ici le spectre d'intercorrélation. Par définition le spectre d'intercorrélation entre deux signaux est :

$$\Phi_{yu}(\omega) = \sum_{\tau=-\infty}^{\infty} R_{yu}(\tau) \cdot e^{-i\omega\tau} \quad (51)$$

On suppose que le signal d'entrée  $u(t)$  et le signal de sortie  $y(t)$  sont reliés par une fonction de transfert linéaire du type :

$$y(t) = F \cdot u(t) + v(t) \quad (52)$$

où  $v(t)$  est une perturbation indépendante de  $u(t)$ . On a alors les relations :

$$\Phi_{yu}(\omega) = F(j\omega) \cdot \Phi_{uu}(\omega) \quad (53)$$

Il reste à estimer les spectres d'intercorrélation pour remonter à la réponse fréquentielle  $F(j\omega)$  recherchée. Une idée naturelle est d'utiliser la relation (48) mais cela marche très mal dans presque tous les cas, même en choisissant  $N$  assez grand. L'amélioration est d'utiliser une "fenêtre" pour favoriser les valeurs faibles de  $\tau$  par rapport aux valeurs élevées. La solution classique utilisant la *fenêtre de Hamming* donne :

$$\Phi_{yu}^N(\omega) = \sum_{\tau=-\gamma}^{\gamma} R_{yu}^N(\tau) \cdot w_{\gamma}(\tau) \cdot e^{-i\omega\tau} \quad (54)$$

avec

$$\begin{cases} w_{\gamma}(\tau) = \frac{1}{2} \left( 1 + \cos\left(\frac{\pi \cdot \tau}{\gamma}\right) \right) & |\tau| < \gamma \\ w_{\gamma}(\tau) = 0 & |\tau| > \gamma \end{cases}$$

On utilise ensuite la relation (51) pour estimer la réponse fréquentielle :



$$F(j \cdot \omega) = \frac{\Phi_u^N(\omega)}{\Phi_{yu}^N(\omega)} \quad (55)$$



## 6. SYSTEMES BOUCLES

### 6.1 Fonction de transfert en boucle fermée

Un système et sa commande par ordinateur peut être représenté par un ensemble de fonctions de transfert formant un schéma-bloc plus ou moins compliqué (Figure 19).

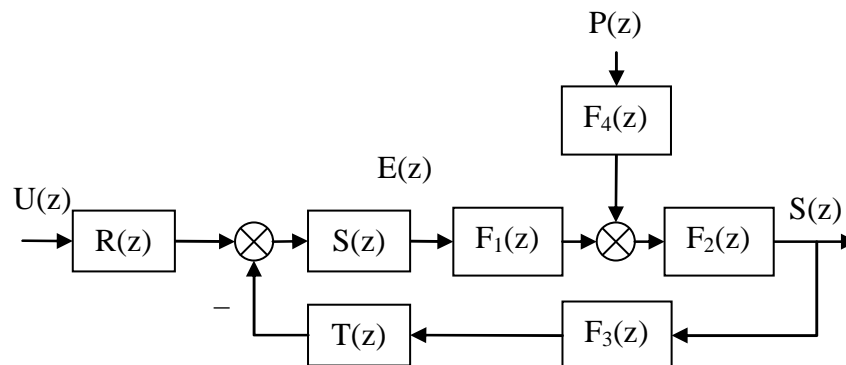


Figure 19 : système discret commandé par ordinateur. Le correcteur est représenté par les fonctions R,S,T, le système par  $F_1$  et  $F_2$ , le modèle de perturbations par  $F_4$  et le modèle des actionneurs par  $F_4$ .

Quelle que soit la structure d'un système asservi il est possible de se ramener à une forme canonique comprenant une fonction de transfert en boucle ouverte en combinant judicieusement les fonctions de transfert du procédé, du régulateur et des perturbations. La Figure 20 représente par exemple le système de la Figure 19 où  $F$  est une combinaison des fonctions de transfert du système, du régulateur, des perturbations et des capteurs.

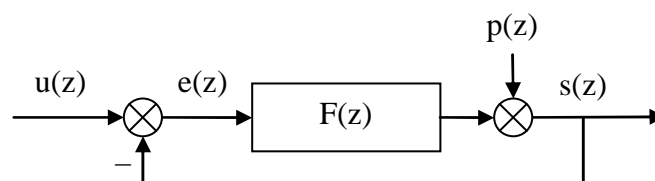


Figure 20 : système discret, notations.

Par défaut on considère une contre-réaction négative et une perturbation additive sur la sortie. La fonction de transfert  $F$  est appelée *fonction de transfert en boucle ouverte*. Comme pour les systèmes continus la *fonction de transfert en boucle fermée* est obtenue simplement à partir de  $F$  (58).

$$\begin{aligned}
 F(z) &= \frac{s(z)}{e(z)} = \frac{B(z^{-1})}{A(z^{-1})} \\
 G(z) &= \frac{s(z)}{u(z)} = \frac{B(z^{-1})}{A(z^{-1}) + B(z^{-1})}
 \end{aligned}
 \tag{56}$$

## 6.2 Erreur statique

L'erreur statique du système en boucle fermée, comme pour le système en boucle ouverte, est simplement le rapport des coefficients.

$$G_0 = \frac{\sum_{i=0}^{i=m} b_i}{1 + \sum_{i=0}^{i=m} b_i + \sum_{i=1}^{i=n} a_i}
 \tag{57}$$

Si l'on souhaite que l'asservissement confère un gain statique unitaire au système en boucle fermée (c'est-à-dire que l'on force la sortie  $s(k)$  à tendre vers la valeur de consigne  $u(k)$ ) il faut que ce rapport de coefficients soit égal à un. On montre facilement que cela est obtenu si la fonction de transfert  $F(z)$  comporte un intégrateur :

$$F(z) = \frac{s(z)}{e(z)} = \frac{1}{1 - z^{-1}} \cdot \frac{B(z)}{A'(z)}
 \tag{58}$$

## 6.3 Rejet de perturbations

La fonction de transfert entre la perturbation  $p(z)$  et la sortie  $y(z)$  est appelée fonction de sensibilité perturbation-sortie, notée  $S_{yp}$ .

$$S_{yp}(z) = \frac{s(z)}{p(z)} = \frac{A(z^{-1})}{A(z^{-1}) + B(z^{-1})}
 \tag{59}$$

Si l'on souhaite qu'une perturbation de type échelon soit parfaitement rejetée il faut que  $A(1) = 0$ , on retrouve donc la condition (58) à savoir que le système doit comporter une intégration dans la boucle directe. On peut être en outre être amené à spécifier que les perturbations ne soient pas amplifiées dans une certaine bande de fréquence. On rajoute donc une contrainte du type  $|S_{yp}(e^{j2\pi f})| < S_{max}$  pour  $f$  appartenant à un certain domaine de  $[0 \ 0.5]$ .

## 6.4 Stabilité des systèmes à simple gain de rétroaction

Les critères de Routh et de Jury (donnés dans le chapitre 3.2.7 p13) ou le calcul numérique des pôles permettent de déterminer la stabilité du système en boucle fermée si l'on connaît sa fonction de transfert.

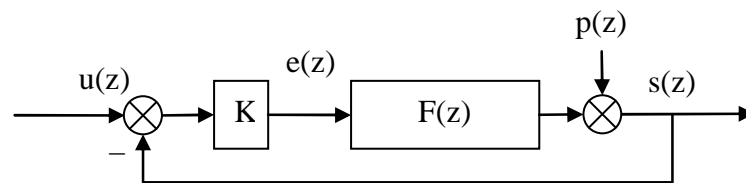


Figure 21 : système bouclé par un simple gain.

On s'intéresse ici au cas particulier où l'asservissement est réalisé par un simple gain de rétroaction  $K$  (Figure 21). On cherche la valeur de  $K$  qui amène le système en limite de stabilité. L'outil adapté ici est le lieu d'Evans ou lieu des racines (root locus) qui permet de visualiser l'évolution des pôles du système bouclé lorsque le gain  $K$  varie.

Prenons l'exemple d'un système comportant trois pôles et deux zéros. L'exemple est donné sous forme de script Matlab :

### Matlab

```
% Définition d'un système discret :
sys=zpk([-2.037 -0.1406],[0.6065 0.3679 0.3679],1,1)
%Tracé du lieu des racines
rlocus(sys);
```

Le script précédent produit le tracé de la Figure 22. On pourrait y lire que la limite de stabilité est obtenue pour  $K=0.2$  environ (détail de la Figure 23).

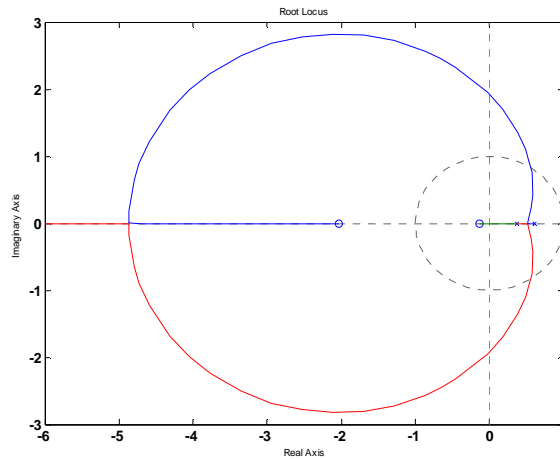


Figure 22 : lieu des racines

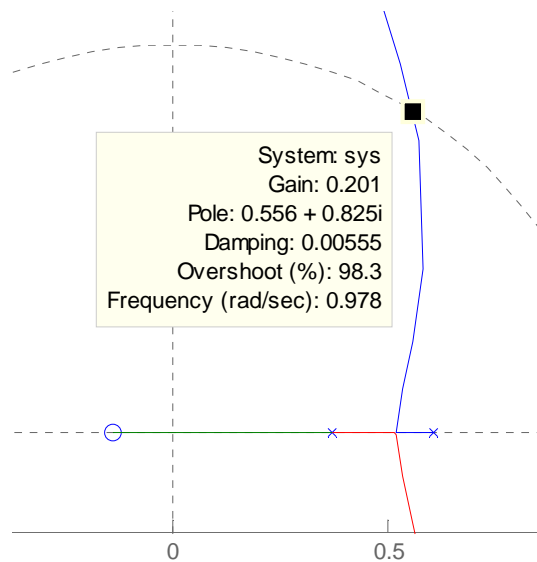


Figure 23 : lieu des racines : détail de la limite de stabilité

On voit sur la Figure 22 que le lieu des racines d'un système échantillonné ressemble à celui d'un système continu :

- on commence par tracer les pôles ( $\times$ ) et les zéros ( $o$ ) du système en boucle ouverte.
- les pôles du système en boucle fermé décrivent un lieu qui part des pôles (pour  $K = 0$ ) et arrive aux zéros (pour  $K = \infty$ )
- lorsque il y a  $n-m$  plus de pôles que de zéros le système comporte  $n-m$  branches asymptotiques.

On ne décrira pas ici les diverses propriétés du lieu des racines qui permettent de le tracer à la main puisque il est facile à obtenir avec un ordinateur.

Il peut aussi être utilisé pour régler "à la main" le gain permettant d'obtenir un ou plusieurs modes dominants ayant certaines propriétés de fréquence et d'amortissement. On voit sur la Figure 24 que le gain 0.017 permet d'obtenir un pôle double d'amortissement 0.706 et de fréquence 0.65 rad/s et un pôle simple de fréquence 1.63 rad/s.

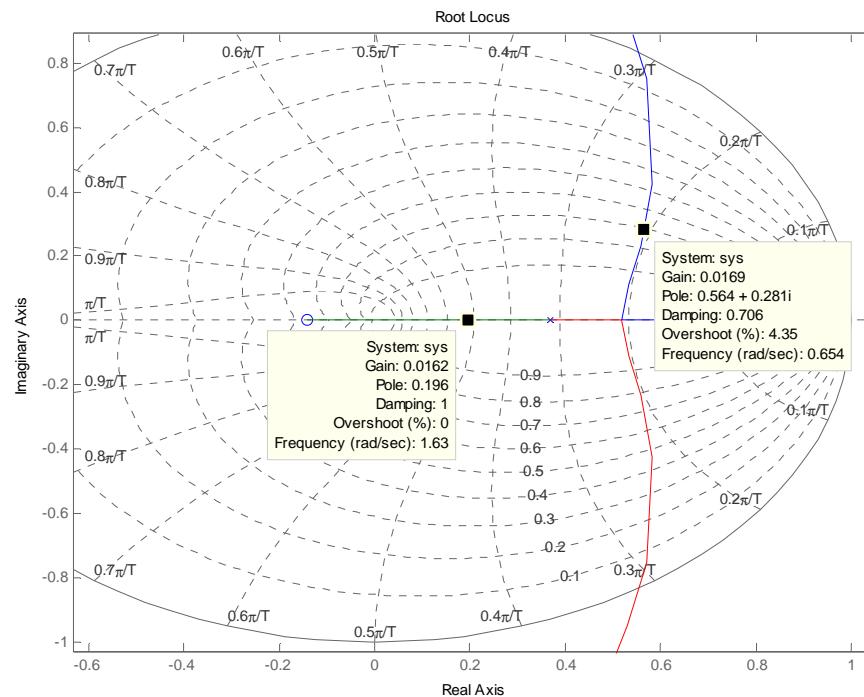


Figure 24 : utilisation du lieu des racines pour régler le gain de la boucle fermée

L'outil rltool de Matlab permet de nombreuses manipulations graphiques de ce type.

## 6.5 Analyse des systèmes bouclés dans le domaine fréquentiel : stabilité

Comme dans le cas continu il est possible de prévoir des propriétés, en particulier la stabilité, d'un système en boucle fermée à partir de l'examen de la fonction de transfert en boucle ouverte.

On ne redémontrera pas ici le théorème de Nyquist mais on se contente de rappeler l'énoncé.

On rappelle que le *lieu de Nyquist* est le lieu de  $F(j2\pi f)$  dans le plan complexe pour  $f$  variant de 0 à  $+\infty$ . A noter que le lieu pour la zone  $[0, 0.5]$  est symétrique de celui pour la zone  $[0.5, 1]$ . Pour cette raison on ne représente généralement que le lieu pour  $f$  appartenant à  $[0, 0.5]$ .

### 6.5.1 Théorème de Nyquist dans le cas où le système est stable en boucle ouverte (critère du revers)

Le système est stable en boucle fermée si le lieu de Nyquist de la fonction de transfert en boucle ouverte parcouru dans le sens des fréquences croissantes de 0 à 0.5 laisse le point critique (-1,0) à gauche.

La Figure 25 représente par exemple le lieu de Nyquist d'une fonction de transfert en boucle ouverte qui sera stable en boucle fermée.

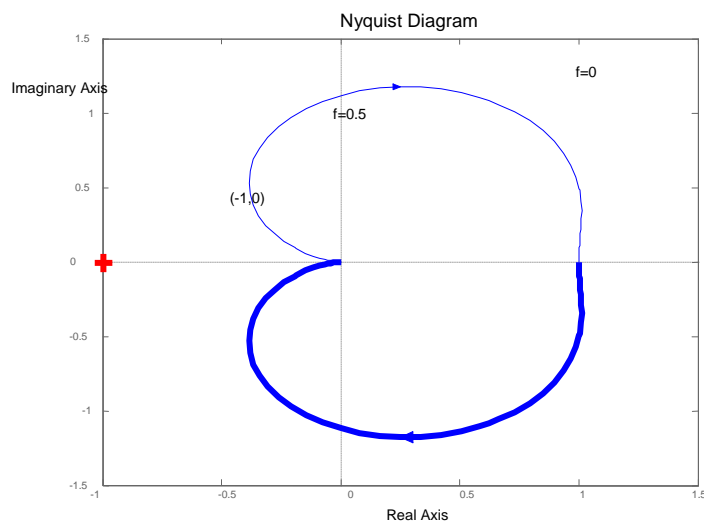


Figure 25 : lieu de Nyquist d'un système stable

### 6.5.2 Théorème de Nyquist dans le cas où le système est instable en boucle ouverte

Si le système en boucle ouverte comporte  $N$  pôles instables il sera stable en boucle fermée si le lieu de Nyquist de la fonction de transfert en boucle ouverte parcouru dans le sens des fréquences croissantes de  $-0.5$  à  $0.5$  entoure le point critique  $(-1,0)$   $N$  fois dans le sens des aiguilles d'une montre..

### 6.5.3 Interprétations du lieu de Nyquist

Comme dans le cas continu, la distance à laquelle on passe du point critique renseigne sur la robustesse du système bouclé. La distance sera exprimée de différente manière (Figure 26) :

- marge de module  $\Delta M$  : plus petite distance entre le lieu des racines et le point critique,



- marge de gain  $\Delta G$
- marge de phase  $\Delta P$

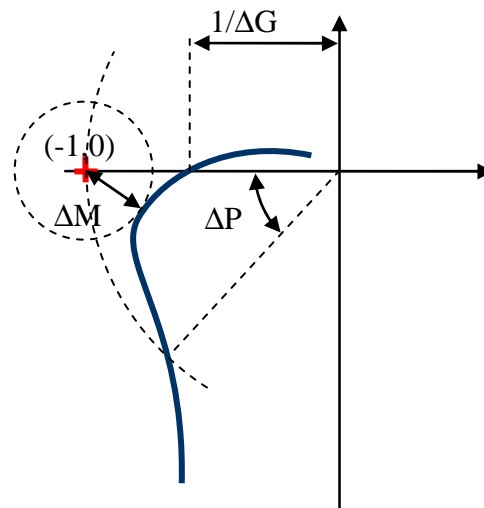


Figure 26 : Marges de stabilité

Les valeurs typiques de ces marges sont données ci-après :

- $\Delta G \geq 6\text{dB}$
- $\Delta M \geq -6\text{dB}$
- $30^\circ \leq \Delta P \leq 60^\circ$

La distance entre le point critique et le lieu des racines est la norme de  $1+F(j2\pi f)$ . On peut donc exprimer la marge de module en fonction de la fonction de sensibilité :

$$\begin{aligned} \Delta M &= \min |1 + F(j \cdot \omega)|_{\omega \in [0, \pi]} \\ \Delta M &= \min \left| 1 + \frac{B(j \cdot \omega)}{A(j \cdot \omega)} \right|_{\omega \in [0, \pi]} = \max \left| \frac{A(j \cdot \omega)}{A(j \cdot \omega) + B(j \cdot \omega)} \right|_{\omega \in [0, \pi]} \quad (60) \\ \Delta M &= \max |S_{yp}(j \cdot \omega)|_{\omega \in [0, \pi]}^{-1} \\ \Delta M_{\text{dB}} &= -\max |S_{yp}(j \cdot \omega)|_{\text{dB}} \end{aligned}$$

Si l'on trace l'évolution de  $S_{yp}$  en fonction de  $f$  on obtient donc la valeur de la marge de module.



# 7. METHODES DE REGULATION DES SYSTEMES ECHANTILLONES

## 7.1 Introduction

Les bases théoriques permettant de concevoir le correcteur pour un système échantillonné ont été posées dans les chapitres précédents. Ce chapitre fait le bilan des connaissances acquises et donne quelques solutions pour calculer des correcteurs à partir d'un cahier des charges précis.

Les systèmes que l'on cherche à commander sont généralement, par nature, continus. La réalisation d'un correcteur par un ordinateur implique que l'on génère une équation de commande discrète (généralement une équation récurrente). L'obtention de cette équation de commande peut se faire de deux manières :

- la première méthode consiste à d'abord synthétiser un correcteur continu (en utilisant les outils classiques de l'automatique et la simulation) et discrétiser ensuite ce correcteur. C'est l'objet du chapitre 7.2.2.
- la seconde méthode consiste à tenir la représentation discrète du système continu associé à son interface numérique (convertisseurs analogique-numérique et numérique-analogique) comme expliqué dans le chapitre 4.5 puis synthétiser directement le correcteur discret. Cela sera expliqué dans le chapitre 7.3 et suivants.

Ce choix peut être résumé par la figure suivante :

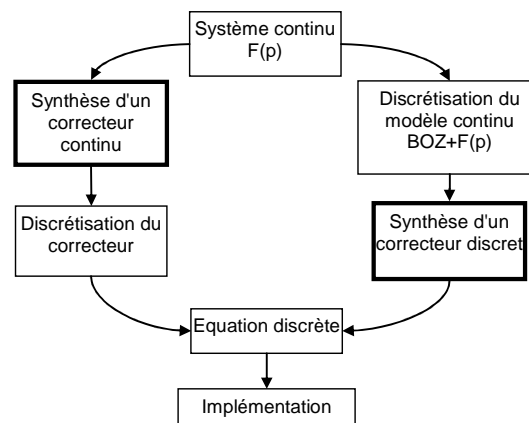


Figure 27 : Réalisation d'un correcteur discret.

Au-delà des affaires de goût, le choix de la méthode sera souvent dicté par l'application. En effet, la première méthode est d'autant meilleure que le correcteur discrétisé approche correctement le correcteur continu. Cela implique que le temps d'échantillonnage soit très faible par rapport aux constantes de temps du système

bouclé. Dans le cas où les contraintes temps réel sont fortes (capacités de calcul partagées par exemple) il sera préférable de synthétiser un correcteur discret.

## 7.2 Structure d'un système commandé par ordinateur, hypothèses

Par la suite on se ramènera toujours à un système tel que celui représenté sur la Figure 28 en cherchant un correcteur  $K(z)$  linéaire. On montre que quel que soit le correcteur linéaire on peut se ramener à la forme représenté sur la Figure 29 (forme RST).

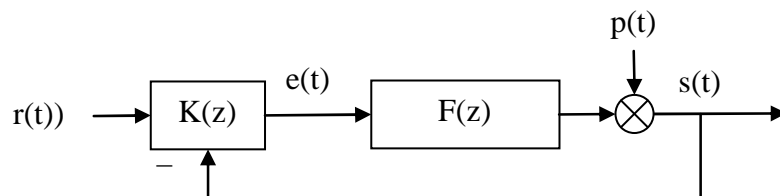


Figure 28 : système bouclé, forme générale.

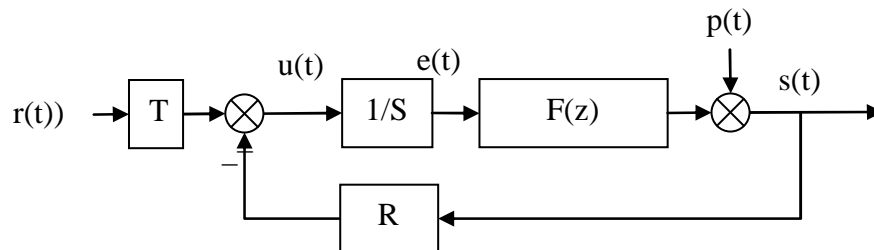


Figure 29 : système bouclé, forme canonique.

### 7.2.1 Discrétisation, choix de la période d'échantillonnage

Du point de vue du ordinateur, l'extérieur (le procédé) est représenté par la fonction de transfert  $F(z)$ . L'hypothèse importante est que la commande (sortie du ordinateur) et la mesure (entrée du ordinateur) sont échantillonné de manière synchrone à la même période  $T_e$ .

Le choix judicieux de la fréquence d'échantillonnage est très important. En pratique on choisit cette fréquence en fonction de la bande passante du système en **boucle fermée** :

$$f_e = (6 \text{ à } 25) f^{BF} \quad (61)$$

### 7.2.2 Discrétisation d'un correcteur continu

La première méthode pour obtenir les équations d'un correcteur numérique est de faire sa synthèse avec les outils de l'automatique continue puis de discrétiser la fonction de transfert du correcteur continu obtenu (voir pour cela le chapitre 4.2 *Echantillonnage des systèmes continus*).

On va voir par la suite que la synthèse directe d'un correcteur discret offre plus de degré de liberté et permet en général de remplir un cahier des charges plus complet.

## 7.3 Régulateur PID numérique

Le régulateur le plus populaire est le régulateur PID. Nous allons voir dans un premier temps comment on génère un régulateur PID à partir des équations bien connues du régulateur PID continu et comment ces équations peuvent être généralisées pour obtenir plus de degrés de liberté pour le réglage.

### 7.3.1 Cas 1

Un régulateur PID continu a une fonction de transfert du type :

$$K_{\text{PID}}(s) = K \cdot \left( 1 + \frac{1}{T_i \cdot s} + \frac{T_d \cdot s}{1 + \frac{T_d}{N} \cdot s} \right) \quad (62)$$

La discrétisation de ce correcteur continu par la méthode d'Euler (Tableau 4 page 31) se fait en remplaçant la variable de Laplace  $s$  par le terme  $\frac{1-q^{-1}}{q^{-1} \cdot T_e}$ . On obtient :

$$K_{\text{PID}}(q) = K \cdot \left( 1 + \frac{1}{T_i \cdot \frac{1-q^{-1}}{T_e}} + \frac{T_d \cdot \frac{1-q^{-1}}{T_e}}{1 + \frac{T_d}{N} \cdot \frac{1-q^{-1}}{T_e}} \right) \quad (63)$$

En réarrangeant on obtient la forme générale RST (Figure 30):

$$K_{PID}(q) = K \cdot \frac{r_0 + r_1 \cdot q^{-1} + r_2 \cdot q^{-2}}{(1 - q^{-1}) \cdot (1 + s_1' \cdot q^{-1})}$$

$$\begin{cases} r_0 = K \cdot \left( 1 + \frac{T_e}{T_i} + \frac{N \cdot T_d}{T_d + N \cdot T_e} \right) \\ r_1 = -K \cdot \left( 1 + \frac{T_d}{T_d + N \cdot T_e} + \frac{T_e}{T_i} \cdot \frac{T_d}{T_d + N \cdot T_e} + \frac{N \cdot T_d}{T_d + N \cdot T_e} \right) \\ r_2 = K \cdot \left( \frac{T_d}{T_d + N \cdot T_e} \right) \\ s_1' = \frac{T_d}{T_d + N \cdot T_e} \end{cases} \quad (64)$$

Comme on le voit le correcteur obtenu comporte autant de degrés de liberté que le correcteur PID analogique. On reconnaît au dénominateur le terme  $(1 - q^{-1})$  en facteur qui assure l'effet d'intégration.

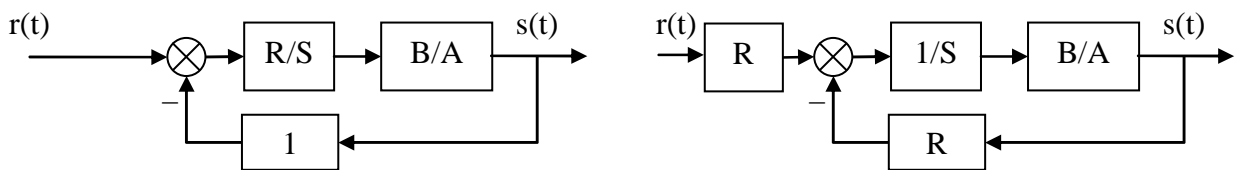


Figure 30 : PID sous forme RST

La fonction de transfert en boucle fermée obtenue est donnée par l'équation suivante :

$$H_{BF} = \frac{B \cdot R}{A \cdot S + B \cdot R} = \frac{B \cdot R}{P} \quad (65)$$

De manière générale le polynôme caractéristique  $P$  fixe les performances visées en boucle fermée. Le calcul de  $R$  et  $S$ ,  $P$  étant fixé et  $A$  et  $B$  étant connus, est réalisé en résolvant l'équation de Bézout  $P = A \cdot S + B \cdot R$ . (voir 9.1 Annexe 1 : Résolution de l'équation polynomiale de Bezout)

On voit que le numérateur  $B \cdot R$  est imposé une fois que  $R$  et  $S$  sont trouvés. Le régulateur introduit donc naturellement des zéros dans la boucle fermée.

Traisons l'exemple entièrement dans le cas où le système est modélisé comme un système du deuxième ordre avec faible retard. La fonction de transfert du procédé est donnée par :

$$H_{BO} = \frac{B}{A} = \frac{b_1 \cdot q^{-1} + b_2 \cdot q^{-2}}{1 + a_1 \cdot q^{-1} + a_2 \cdot q^{-2}} \quad (66)$$

Le correcteur PID rajoutant deux pôles, le polynôme caractéristique en boucle fermée spécifié doit posséder quatre pôles. Le polynôme caractéristique en boucle fermée doit donc être de la forme :

$$\begin{aligned} P &= 1 + p_1 \cdot q^{-1} + p_2 \cdot q^{-2} + p_2 \cdot q^{-3} + p_2 \cdot q^{-4} \\ P &= (1 + p_1' \cdot q^{-1} + p_2' \cdot q^{-2}) \cdot (1 + \alpha_1 \cdot q^{-1}) \cdot (1 + \alpha_2 \cdot q^{-1}) \end{aligned} \quad (67)$$

Et il doit être identifié au produit  $AS + BR$  avec  $R$  et  $S$  ayant les formes données par (64) :

$$\begin{aligned} &(1 + p_1' \cdot q^{-1} + p_2' \cdot q^{-2}) \cdot (1 + \alpha_1 \cdot q^{-1}) \cdot (1 + \alpha_2 \cdot q^{-1}) \\ &= (1 + a_1 \cdot q^{-1} + a_2 \cdot q^{-2}) \cdot (1 - q^{-1}) \cdot (1 + s_1' \cdot q^{-1}) \\ &+ (b_1 \cdot q^{-1} + b_2 \cdot q^{-2}) \cdot (r_0 + r_1 \cdot q^{-1} + r_2 \cdot q^{-2}) \end{aligned} \quad (68)$$

Les coefficients  $p_1'$  et  $p_2'$  définissent un modèle du second ordre aux caractéristiques (pulsation propre  $\omega_0$  et amortissement  $\sigma$  choisis et coefficients déduits conformément au Tableau 3 page 25). Les coefficients  $\alpha_1$  et  $\alpha_2$  correspondent à des pôles auxiliaires aperiodiques. Les choix typiques pour ces coefficients sont :

$$-0.05 < \alpha_1, \alpha_2 < -0.5 \quad (69)$$

On montre que la résolution de (68) est réalisée par une inversion matricielle simple :

$$M \cdot x = p \quad (70)$$

avec

$$\begin{cases} x^T = [1 & s_1' & r_0 & r_1 & r_2] \\ p^T = [1 & p_1 & p_2 & p_3 & p_4] \end{cases}$$

et

$$M = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ a_1' & 1 & b_1 & 0 & 0 \\ a_2' & a_1' & b_2 & b_1 & 0 \\ a_3' & a_2' & 0 & b_2 & b_1 \\ 0 & a_3' & 0 & 0 & b_2 \end{pmatrix} \quad (71)$$

On montre que M est inversible à la condition que A et B soient premiers entre eux, c'est-à-dire que A et B n'aient pas de pôles et de zéros en commun.

(Le problème peut aussi être résolu en utilisant la résolution de l'équation de Bézout).

### 7.3.2 Cas 2

Comme on l'a vu le correcteur PI numérique directement transposé du correcteur PID analogique décrit dans le chapitre 7.3.1 a l'inconvénient de ramener des zéros au dénominateur de la fonction de transfert en boucle fermée (équation 65). Il est possible, grâce à la flexibilité du régulateur numérique d'imposer au correcteur de ne pas ajouter de zéros en boucle fermée. La fonction de transfert désirée en boucle fermée sera donc de la forme :

$$H_{BF} = \frac{P(1)}{B(1)} \cdot \frac{B}{P} \quad (72)$$

Le facteur  $P(1)/B(1)$  garantit le gain statique unitaire du système en boucle fermée (voir page 12). Si le régulateur a la forme générale RST alors les polynômes R S et T doivent vérifier :

$$H_{BF} = \frac{B \cdot T}{A \cdot S + B \cdot R} = \frac{P(1)}{B(1)} \frac{B}{P} \quad (73)$$

S et R sont obtenus par la résolution de la même équation que précédemment (68). Le polynôme T est obtenu par simple identification :  $T = R(1)$ .

La structure, sous forme de schéma-bloc est donnée ci-dessous :

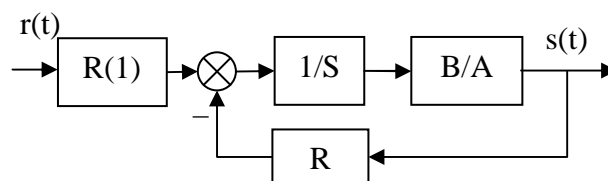


Figure 31 : PID2 sous forme RST

On montre que cette structure de PID, ramené en continu, correspond au PID canonique pour l'action intégrale. L'action proportionnelle et dérivée n'est appliquée que sur le signal y.

La structure du PID2 par rapport au PID1 a pour avantage de ne pas ramener de zéros à la boucle fermée. Cela se traduit par un dépassement moindre en transitoire lorsqu'on applique un échelon de commande. Par contre les deux systèmes réagissent de la même manière à un échelon de perturbation.



## 7.4 Placement de pôles

Les deux structures de PID décrites dans le chapitre précédent sont deux cas particulier d'une stratégie plus générale de placement de pôles.

La synthèse est plus générale que les PID présentés précédemment puisqu'il n'est pas nécessaire de faire d'hypothèses sur le degré de A et B, sur le retard, ni sur la présence ou non de zéros stables ou instables dans le procédé.

La structure du correcteur est la structure type RST telle que présentée dans la Figure 29. Pour la suite des calculs on posera pour la fonction de transfert les conventions suivantes :

$$H_{BO} = \frac{q^{-d} \cdot B(q^{-1})}{A(q^{-1})} \quad (74)$$

$$\begin{cases} A(q^{-1}) = 1 + a_1 \cdot q^{-1} + \dots + a_{n_A} \cdot q^{-n_A} \\ B(q^{-1}) = b_1 \cdot q^{-1} + \dots + b_{n_B} \cdot q^{-n_B} = q^{-1} \cdot B^*(q^{-1}) \end{cases}$$

Compte tenu de la forme RST du régulateur la fonction de transfert en boucle fermée vérifie :

$$H_{BF} = \frac{q^{-d} \cdot B(q^{-1}) \cdot T(q^{-1})}{P(q^{-1})} \quad (75)$$

$$P(q^{-1}) = A(q^{-1}) \cdot S(q^{-1}) + q^{-d} \cdot B(q^{-1}) \cdot R(q^{-1}) = 1 + p_1 \cdot q^{-1} + \dots + p_{n_P} \cdot q^{-n_P}$$

Le polynôme  $P(q^{-1})$  définit donc les pôles de la fonction de transfert en boucle fermée et traduit donc le comportement en *poursuite*.

De la même manière la fonction de transfert entre une perturbation et la sortie (fonction de sensibilité perturbation-sortie) est donnée par :

$$S_{yp} = \frac{A(q^{-1}) \cdot S(q^{-1})}{P(q^{-1})} \quad (76)$$

Ce même polynôme traduit donc aussi le comportement en *régulation*.

Les consignes en poursuite et en régulation peuvent être différentes comme cela est représenté par exemple sur la Figure 32.

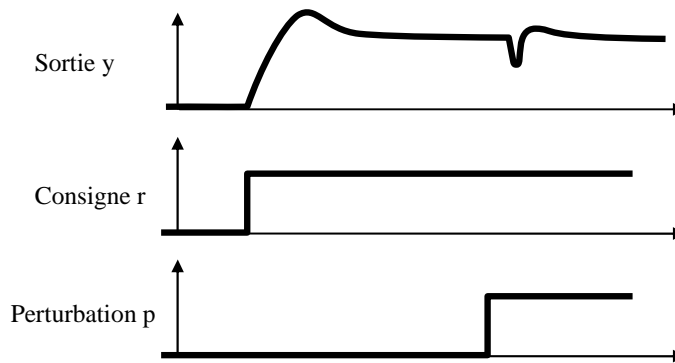


Figure 32 : Performances différentes en régulation et en poursuite

### 7.4.1 Choix des pôles en boucle fermée

De manière générale le comportement en boucle fermée est spécifié sous la forme d'une pulsation propre  $\omega_0$  et d'un amortissement  $\sigma$ , donc une fonction de transfert du second ordre. La spécification étant généralement donnée en continu il est nécessaire de traduire la valeur désirée de  $\omega_0$  et  $\sigma$  en un polynôme  $P_D(q^{-1})$  par discrétisation de la fonction de transfert continue (chapitre 4.5 page 29). Ce polynôme  $P_D$  traduit le comportement dominant du système bouclé.

Si l'on souhaite introduire un effet de filtrage supplémentaire dans certaines bandes de fréquence (réduction des bruits de mesure, adoucissement de la commande, robustesse) cela est décrit par un polynôme  $P_F$  comportant des pôles auxiliaires, plus rapides que les pôles de  $P_D$ .

Le polynôme caractéristique en boucle fermée est donc constitué de pôles dominants et de pôles auxiliaires :

$$P(q^{-1}) = P_D(q^{-1}) \cdot P_F(q^{-1}) \quad (77)$$

### 7.4.2 Régulation : calcul de R et S

Le polynôme  $P(q^{-1})$  étant fixé il s'agit de trouver R et S vérifiant :

$$P(q^{-1}) = A(q^{-1}) \cdot S(q^{-1}) + q^{-d} \cdot B(q^{-1}) \cdot R(q^{-1}) \quad (78)$$

Comme déjà vu au chapitre 7.3.1 on obtient une équation de Bézout qu'il s'agit de résoudre. (Voir aussi 9.1 Annexe 1 : Résolution de l'équation polynomiale de Bezout)

Il est parfois nécessaire de spécifier R et S avec des parties fixes  $H_R$  et  $H_S$ . Dans ce cas l'équation (78) est modifiée et on l'écrit sous la forme:

$$P(q^{-1}) = (A(q^{-1}) \cdot H_S(q^{-1})) \cdot S'(q^{-1}) + (q^{-d} \cdot B(q^{-1}) \cdot H_R(q^{-1})) \cdot R'(q^{-1}) \quad (79)$$

Les inconnues que l'on souhaite trouver sont  $S'$  et  $R'$ . Ceci est nécessaire dans les cas particuliers suivants :

- spécification d'erreur statique nulle,
- rejet d'une perturbation harmonique,
- blocage d'un signal,

Détaillons maintenant ces cas particuliers.

### Spécification d'une erreur statique nulle

On retrouve la nécessité d'avoir un terme intégrateur dans la boucle directe. La spécification se retrouve donc sur  $S$  :

$$S(q^{-1}) = (1 - q^{-1}) \cdot S'(q^{-1}) = H_S(q^{-1}) \cdot S'(q^{-1}) \quad (80)$$

### Rejet d'une perturbation harmonique

Si le système est perturbé par une perturbation harmonique à une fréquence fixe (par exemple 50Hz...) il est possible d'imposer à  $S$  d'avoir un module *nul* ou amorti à cette fréquence. Cela se traduit par la présence dans  $S$  d'une paire de zéros complexe à amortissement nul ou faible à cette fréquence :

$$S(q^{-1}) = (1 + \alpha \cdot q^{-1} + \beta \cdot q^{-2}) \cdot S'(q^{-1}) = H_S(q^{-1}) \cdot S'(q^{-1}) \quad (81)$$

### Blocage d'un signal

Pour certaines application on souhaite que le régulateur "laisse passer" certaines composantes du signal de sortie, c'est-à-dire qu'il ne réagisse pas à ces composantes. Cela se traduit par  $R=0$  à ces fréquences. On retrouve la même forme que précédemment :

$$R(q^{-1}) = (1 + \alpha \cdot q^{-1} + \beta \cdot q^{-2}) \cdot R'(q^{-1}) = H_R(q^{-1}) \cdot R'(q^{-1}) \quad (82)$$

### 7.4.3 Poursuite : calcul de T

Le schéma-bloc de l'asservissement est donné sur la Figure 33. Le terme  $T$  est décomposé en un terme  $T'$  et un terme  $B^*/A^*$ .

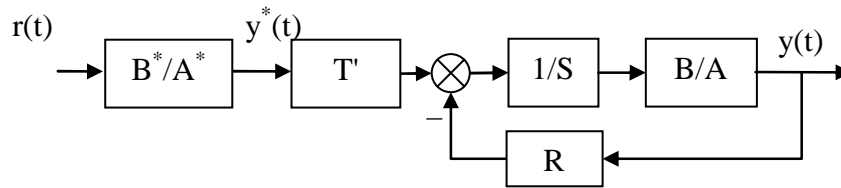


Figure 33 : placement de pôle

L'idée est de régler  $B^*$  et  $A^*$  de telle manière que le signal  $y^*$  ait la dynamique de poursuite désirée (par exemple un second ordre avec pulsation propre et amortissement donné) puis de régler  $T'$  de telle manière que le transfert entre  $y^*$  et  $y$  soit le plus "transparent" possible.

### Réglage de $T'$

Le transfert entre  $y^*$  et  $y$  est :

$$S_{yy^*}(q^{-1}) = T'(q^{-1}) \frac{q^{-d} \cdot B(q^{-1})}{P(q^{-1})} \quad (83)$$

Pour assurer le transfert le plus "transparent" possible il faut donc choisir

$$T'(q^{-1}) = \frac{P(q^{-1})}{B(1)} \quad (84)$$

Remarques :

- Si  $B(1)=0$  alors on remplace  $B(1)$  par 1 dans (84)
- De même dans (84)  $P(q^{-1})$  peut être remplacé par  $P_D$  correspondant à la dynamique dominante.

### Réglage de $B^*$ et $A^*$

$B^*$  et  $A^*$  sont choisis de telle manière que la fonction de transfert  $B^*/A^*$  corresponde à la dynamique de poursuite souhaitée.

## 7.5 Poursuite et régulation à objectifs indépendants

Cette méthode ressemble à la méthode de placement de pôles à ceci près que l'on va effectuer une simplification des zéros de  $B^*(q^{-1})$  : cela ne s'applique donc qu'aux systèmes avec des pôles stables. Plus précisément il faut qu'ils aient un amortissement suffisant ( $> 0,2$ ) de manière à ne pas être trop près du cercle unité.

La structure de l'asservissement est toujours celle donnée sur la Figure 33.

**Choix de P**

$P(q^{-1})$  est choisi de manière à simplifier  $B^*(q^{-1})$  (rappel : la forme de B est donnée dans l'équation 74). Comme indiqué dans le chapitre 7.4.2 page 60 il s'agit donc d'imposer les zéros de  $B(q^{-1})$  dans  $H_R(q^{-1})$  et  $H_S(q^{-1})$ .

**Choix de T**

Si l'on choisit  $T(q^{-1}) = P(q^{-1})$  alors le transfert entre  $y^*$  et  $y$  est :

$$S_{yy^*}(q^{-1}) = q^{-(d+1)} \quad (85)$$

**Réglage de  $B^*$  et  $A^*$** 

$B^*$  et  $A^*$  sont choisis de telle manière que la fonction de transfert  $B^*/A^*$  corresponde à la dynamique de poursuite souhaitée.



## 8. TECHNOLOGIE DE LA COMMANDE PAR ORDINATEUR

Du point de vue de l'automaticien le ordinateur est relié au système physique à piloter par un certain nombre d'intermédiaires plus ou moins gênants :

- les interfaces analogiques et numériques,
- le transport de l'information par bus et protocole,
- l'utilisation éventuelle de capteurs ou d'actionneurs « intelligents ».

Il ne s'agit pas ici de faire un exposé complet sur ces trois aspects mais de poser les problèmes qui se posent à l'automaticien lorsque le système qu'il doit commander est perturbé par ces éléments.

Enfin ce chapitre se termine sur les notions de « Hardware In the Loop » et de « prototypage rapide » qui sont partie intégrante des processus de réalisation de systèmes de contrôle.

### 8.1 Calculateurs et interfaces numériques / analogiques

Le premier problème à prendre en compte est la quantification de l'information. Du point de vue du ordinateur d'une part et du point de vue des convertisseurs d'autre part. Dans un premier temps, considérons les aspects performance de calcul lié à la nature du codage des nombres dans le ordinateur.

#### Les performances liées au types de variables

Prenons l'exemple où le concepteur de la loi de commande souhaite réaliser un correcteur de type proportionnel. Entre l'acquisition des mesures et l'écriture de la consigne il y aura la réalisation d'une de commande du type :

$$y = K \cdot (\text{consigne} - \text{mesure}) \quad (86)$$

En utilisant un langage de programmation évolué (par exemple le C) la ligne de code sera la transcription littérale de (86). Le programmeur devra décider de la définition et du type des variables, principalement du nombre d'octets utilisés pour stocker ses variables. Le Tableau 5 résume par exemple les types utilisés en langage C (norme ANSI)

Tableau 5 : Types ANSI C

Types entiers	char	8 bits
	short int	16 bits
	long int	32 bits
Types flottants (mantisse et exposant)	float	32 bits
	double	64 bits
	longdouble	128 bits

Une fois le code écrit en C l'exécution du programme sur un processeur cible ne peut se faire que par l'intermédiaire d'une compilation du programme. A chaque processeur est donc adapté une version de compilateur.

Les performances du programme (sa rapidité d'exécution) dépendront donc en partie de la qualité du compilateur. (Par exemple, une multiplication s'effectue t'elle bêtement par un décalage de tous les bits ou bien des solutions astucieuses sont-elles implémentées pour arrêter le décalage selon le contexte ?)

Le principal déterminant de la performance est cependant le processeur : son type (8 bits, 16 bits ou 64 bits) et son cadencement (en MHz ou GHz) sont les deux plus importants paramètres. Coder les variables sur 64 bits (double) dans le programme coûtera cher en temps de calcul si le processeur est un 8 bits. Le type de mémoire et leurs temps d'accès peuvent aussi être déterminants. Enfin, certains processeurs disposent d'un processeur spécifiquement dédié aux opérations mathématiques (coprocesseur mathématique).

Il n'est pas possible de dire exactement, pour tel type de compilateur associé à tel type de processeur, le temps que prendra par exemple une multiplication. En pratique on effectue une série de tests types (benchmark) et on affiche un temps moyen d'exécution.

En résumé, on peut être tenté, pour ne pas s'embêter, de coder toutes les variables en virgule flottante. Mais il faut bien savoir que cela a un coût en terme de temps de calcul, surtout si le processeur n'est pas adapté. Pour fixer les idées on peut dire qu'il y a un rapport 20 entre une addition réalisée en "short" et une addition réalisée en virgule flottante.

Enfin, l'utilisation de fonctions mathématiques (sinus, exponentielles, etc...) peut aussi avoir un coût très lourd si le travail n'est pas réalisé par un coprocesseur mathématique dédié. Le compilateur et sa librairie mathématique joue là encore un grand rôle.

### La quantification imposée par le processeur

Au-delà de l'aspect performance et temps de calcul, la quantification de l'information doit parfois aussi être prise en compte. Rappelons qu'un entier codé sur N bits peut prendre seulement  $2^N - 1$  valeurs différentes. Avant d'effectuer les calculs mathématiques dans le corps du programme il est donc judicieux d'appliquer un



facteur d'échelle de manière à utiliser l'amplitude maximale possible (entre 0 et  $2^N - 1$ ).

### **La quantification imposée par les conversions analogique et numérique**

Les convertisseurs analogique-numérique et numérique-analogique sont principalement caractérisés par leur amplitude de signal analogique d'entrée (généralement en volts, par exemple +/- 10V) et par leur résolution en bits (8 ou 10 bits pour les plus courants, jusqu'à 16 bits pour les convertisseurs haut de gamme).

Prenons l'exemple d'un capteur gyromètre transformant une vitesse de rotation en tension. Compte tenu de l'application le signal utile varie entre 2,4V et 2,6V. Si le signal mesuré est branché directement sur un convertisseur 0-10V 10bits alors un problème de conditionnement du signal se pose. En effet la quantification est de  $\frac{10V}{2^{10}-1} = 0,098V$ . Les 0,2V de signal utile seront donc discrétisés en  $\frac{0,2}{0,098} = 20$  intervalles seulement !

Si la discrétisation du signal utile est trop grossière cela peut se traduire par une diminution des performances de l'asservissement, voire une instabilité. Le système n'est en effet plus linéaire et il n'existe pas d'outils théoriques permettant de prévoir les performances du système bouclé.

## **8.2 Transport de l'information : les bus et les protocoles**

Pas question de faire ni même d'ébaucher un cours sur les bus et les protocoles ici. Mais on va quand même donner quelques indications. L'automaticien a l'habitude de simplifier le problème qu'il se pose en dessinant un schéma simplifié de son problème d'asservissement (du type de la Figure 20 page 45) pour pouvoir appliquer les théories de la commande. Cependant il arrive bien souvent que l'information transite à l'intérieur du système, soit entre différents ordinateurs effectuant différentes parties du travail soit entre ordinateur et périphériques, par l'intermédiaire de bus informatique. A chaque type de bus correspond un ou plusieurs protocoles qui peuvent poser des problèmes pour l'automaticien :

- les données sont-elles discrétisées ? (le protocole peut par exemple ne permettre de transmettre que des paquets de données de 16 bits),
- le bus induit-t'il un retard ? (le composant mettant en forme les données du ordinateur pour les publier sur le bus a certaines performances qui peuvent induire un délai)
- le bus est-il déterministe ? (Certains bus ne garantissent pas que les données arriveront en un temps donné),
- le bus est-il sûr ? (Certains bus ne garantissent pas non plus que les données arriveront bien)

### **8.3 Capteurs et actionneurs "intelligents", "smart sensors"**

De plus en plus souvent les capteurs et les actionneurs sont associés à leur électronique de puissance, électronique de commande et de conditionnement de signal. L'interface avec le capteur ou l'actionneur se fait donc via un protocole de communication (par exemple via un bus RS232).

L'intérêt est de véhiculer non pas des signaux analogiques, tensions ou courants, mais des signaux numériques beaucoup moins sensibles aux éventuelles perturbations.

On retrouve donc, entre le capteur ou l'actionneur et le ordinateur, la présence d'un bus et d'un protocole de communication.

### **8.4 La philosophie "hardware in the loop"**

Les systèmes embarqués sont de plus en plus complexes et touchent des domaines de plus en plus variés : automobile, satellites, aéronautique, UAV, systèmes d'arme, sous-marins, systèmes, etc... Les systèmes à piloter ainsi que les systèmes de commande sont de plus en plus complexes.

Les techniques de simulation « Hardware In the Loop » (HIL) sont de plus en plus utilisées dans le développement et dans le test de ces systèmes embarqués. L'idée est de développer entièrement le système de contrôle sur une plateforme de test en utilisant le maximum de composants réels et en ne remplaçant que le minimum par la simulation. Par exemple en aéronautique l'« Iron Bird » regroupe dans un hangar tous les systèmes réels de l'avion en cours de développement. Seule la mécanique du vol est simulée : les sorties des actionneurs sont mesurées, rentrées dans un modèle de simulation numérique, puis les mesures effectuées normalement par les capteurs sont calculées et restituées par le modèle.

L'intérêt est de diminuer de manière importante les coûts et les délais en économisant le plus grand nombre possible de tests sur le système réel en situation. Lors des premiers tests réels les techniques « Hardware In the Loop » permettent d'avoir effectué la plus grande part du développement des lois de commande.

### **8.5 Le prototypage rapide en automatique**

Un peu dans le même ordre d'idée que le « Hardware In the Loop » le prototypage rapide permet de réaliser la conception des lois de commande en simulation puis de transporter rapidement (en un simple click !) le code du contrôleur simulé vers la cible temps réel implémentée sur le système réel.

La société Mathworks [9] investit massivement dans ce domaine en ayant enrichi son logiciel de calcul scientifique (Matlab) avec un logiciel de simulation graphique très évolué (Simulink) puis en développant des outils de prototypage rapide associés (Real Time Workshop, Embedded Target For...).

## 9. ANNEXE

### 9.1 Annexe 1 : Résolution de l'équation polynomiale de Bezout

Il s'agit de trouver R et S vérifiant :

$$P(q^{-1}) = A(q^{-1}).S(q^{-1}) + B(q^{-1}).R(q^{-1})$$

Où P, A et B sont connus.

Dans le cas où P, A, B, R et S sont des nombres entiers la solution de l'algorithme est donnée par l'algorithme de division Euclidienne étendu [8].

Dans le cas où P, A, B, R et S sont des polynômes, l'algorithme est donné ci-dessous, programmé en matlab ([1]).

```
function [Rp,Sp,nrp,nsp]=bezoutd(A,B,Hs,Hr,P)
%function [Rp,Sp,nrp,nsp]=bezoutd(A,B,Hs,Hr,P)
%solves AHsSp+BHrRp=P by coefficient comparison. Delay and discretization delay need to be integrated in B
%(zeros at the beginning).
%
%inputs:
%A=[a0 a1 ... aNa] ... vector of model denominator coefficients A=a0 + a1z^(-1) + a2z^(-2) +...+ aNaz^(-Na)
%B=[b0 b1 ... bNb] ... vector of model numerator coefficients B=b0 + b1z^(-1) + b2z^(-2) +...+ bNbz^(-Nb)
%Hs=[hs0 hs1 ... hsNhs] ... vector of controller denominator fixed part Hs=hs0 + hs1z^(-1) +...+ hsNhsz^(-Nhs)
%Hr=[hr0 hr1 ... hrNhr] ... vector of controller denominator fixed part Hr=hr0 + hr1z^(-1) +...+ hrNhrz^(-Nhr)
%P=[p0 p1 ... pNp] ... vector of desired polynomial coefficients P=p0 + p1z^(-1) + p2z^(-2) +...+ pNpz^(-Np)
%outputs:
%Rp=[rp0 rp1 rp2 ...] ... vector of coefficients for resulted controller numerator
%Sp=[sp0 sp1 sp2 ...] ... vector of coefficients for resulted controller denominator
%nrp ... order of Rp
%nsp ... order of Sp
%
%written by: J. Langer, I.D. Landau, H. Prochazka
%7th june 2002

PRECISION=1e-16;

D=size(A);
if D(1)>1, A=A'; end;
```

```

D=size(B);
if D(1)>1, B=B'; end;
D=size(Hs);
if D(1)>1, Hs=Hs'; end;
if D(1)==0, Hs=1; end;
D=size(Hr);
if D(1)>1, Hr=Hr'; end;
if D(1)==0, Hr=1;end;
D=size(P);
if D(1)>1, P=P'; end;

na=length(A)-1;
nb=length(B)-1;
np=length(P)-1;
nhs=length(Hs)-1;
nhr=length(Hr)-1;

if (nhs>0), Ah=real(convz(A,Hs)); else Ah=A*Hs; end; % Ah = A * Hs
nah=length(Ah) -1;
if (nhr>0), Bh=real(convz(B,Hr)); else Bh=B*Hr; end;
nbh=length(Bh) -1;
if (np>nah+nbh-1), disp('Bezout error: too many poles');end;

% increase size of P using zeros if necessary
if (np<nah+nbh-1),
%
% set remaining poles onto a circle with radius rmin
P=[P zeros(1,nah+nbh-1-np)];
rootsPdes=roots(P);
nextra=nah+nbh-1-np;
rmin=1e-16;
angle=[0:nextra-1]'/nextra*2*pi;
j=sqrt(-1);
rootsPextra=rmin*exp(j*angle);
P=poly([rootsPdes;rootsPextra]);
np=nah+nbh-1;

end;
P,

nsp=nbh-1;
nrp=nah-1;

%matrix is smaller than vector PD

```

```

if (np>nah+nbh-1),
    disp('Order of model denominator is too low! Add a polynom of higher order to Hs or Hr. ');
end;

% ns=nsp+nhs
% nr=nrp+nhr

M=[];
for j=1:nsp+1,
    V=[];
    if (j>1), V=[V ; zeros(j-1,1)]; end;% zeros in front of Ah
    V=[V ; Ah'];% Ah
    if (j<=nsp), V=[V ; zeros(nsp+1-j,1)]; end;% zeros behind Ah
    if (length(V)~=nah+nbh), disp('bezoutb: error V'); end;
    M=[M V]; % add one column to M
end;

for j=1:nrp+1,
    V=[];
    if (j>1), V=[V ; zeros(j-1,1)]; end;
    V=[V ; Bh'];
    if (j<=nrp), V=[V ; zeros(nrp+1-j,1)]; end;
    if (length(V)~=nah+nbh), disp('bezoutb: error V'); end;
    M=[M V];
end;

D=size(M);
if (D(1)~=nah+nbh), disp('bezoutb: error size M row'); end;
if (D(2)~=nah+nbh), disp('bezoutb: error size M column'); end;

% make P column vector
P=P';

global M1;
M1=M;

X= M\P;
% coefficients are real values
X=real(X);

% make X row vector
X=X';

```

## Commande des systèmes par ordinateur

```
Sp=X(1:nsp+1);  
Rp=X(nsp+2:nsp+nrp+2);
```

## 10. BIBLIOGRAPHIE

- [1] I. D. Landau, "*Commande des systèmes*", Edition Lavoisier 2002.
- [2] C. Nouals, "Théorie du Signal", Notes de cours ENSICA
- [3] F. Vincent, "*Traitement Numérique du Signal*", Notes de cours ENSICA.
- [4] D. Arzelier, "*Représentation et analyse des systèmes, notes de cours*", Edition ENSICA. Voir aussi <http://www.laas.fr/~arzelier/>
- [5] J. Bordeneuve-Guibé, "*Commande des systèmes linéaires*", Notes de cours ENSICA.
- [6] Y. Briere, "*Identification*", Notes de cours ENSICA.
- [7] L. Ljung, "*System Identification, Theory for the User*", Prentice Hall 1987.
- [8] Wikipedia , équation de Bezout : [http://en.wikipedia.org/wiki/Bézout's\\_identity](http://en.wikipedia.org/wiki/Bézout's_identity)
- [9] <http://www.mathworks.com>





# 11. INDEX

<b>A</b>	
actionneur intelligent .....	21
amortissement .....	15
ANSI C .....	60
Approximation numérique d'équation différentielle .....	28
ARMAX .....	35
<b>B</b>	
BJ35 .....	
Blocage d'un signal .....	56
bloqueurs d'ordre 0 .....	22
BOZ .....	22
bus.....	21, 61
<b>C</b>	
CAN.....	21
capteur intelligent .....	21
causal .....	6, 9
Causalité .....	10
CNA.....	21
compilateur .....	60
convertisseur analogique / numérique.....	21
convertisseur numérique / analogique.....	21
convertisseurs analogique-numérique .....	61
coprocesseur mathématique .....	60
Correspondances pôles - zéros.....	31
<b>D</b>	
Diagramme de Bode .....	18
Discrétisation de fonction de transfert continues .....	27
division polynomiale.....	6
<b>E</b>	
échelon unitaire.....	11
équation d'état .....	17
Erreur statique.....	42
Estimation de la réponse fréquentielle .....	39
Estimation de la réponse impulsionnelle.....	38
Euler .....	29
<b>F</b>	
facteur d'échelle .....	61
facteur d'oubli .....	36
<i>fenêtre de Hamming</i> .....	39
filtre anti-repliement .....	25
fonction de sensibilité .....	42
fonction de transfert discrète.....	9
<b>G</b>	
gain statique.....	11
gain statique unitaire.....	42
<b>I</b>	
Identification.....	32
Identification non paramétrique .....	38
intercorrélation.....	38
<b>L</b>	
lieu des racines.....	43
lieu d'Evans.....	43
LTI.....	8
<b>M</b>	
marge de gain.....	47
marge de module.....	46
marge de phase .....	47
méthode des moindres carrés récursifs .....	32
Méthode des trapèzes.....	29
modèle ARX .....	32
<b>O</b>	
OE.....	35
one step ahead prediction.....	35
opérateur de retard .....	9
<b>P</b>	
période d'échantillonnage (choix) .....	26
Placement de pôles .....	54
Pôles .....	11
Poursuite .....	56
<i>prédiction</i> .....	32
protocoles.....	61
Pseudo Random Binary Signal .....	34
pulsation .....	15
<b>Q</b>	
quantification .....	23, 24, 60
<b>R</b>	
R,S,T.....	41
rapidité.....	15
rayon de convergence .....	5, 9
Régulation.....	55

Rejet de perturbations .....	42
Rejet d'une perturbation harmonique .....	56
réponse à un échelon.....	11
réponse fréquentielle.....	18
Réponse impulsionnelle.....	10
Réponses indicielle .....	10
retard.....	23
root locus .....	43

### **S**

signal discret .....	5
smart sensors.....	62
stabilité.....	11
Systèmes du premier ordre.....	14
Systèmes du second ordre.....	15

### **T**

Théorème de Nyquist.....	46
théorème de Shannon.....	19, 24
théorème du retard .....	6
Transformée en z .....	5
transformée unilatérale .....	5
Tustin.....	29

### **V**

virgule flottante.....	60
------------------------	----

### **Z**

zero holder hold .....	23
zéros.....	11
zoh .....	23

# Commande des systèmes par ordinateur

Yves Briere  
ENSICA

Notes de cours  
2007



# TABLE DES MATIERES

Table des matières .....	1
1. Introduction .....	1
2. Rappels et prérequis .....	3
3. Signaux et systèmes discrets .....	5
3.1 Signaux discrets .....	5
3.1.1 Transformée en z .....	5
3.1.2 Signaux types .....	7
3.2 Systèmes discrets .....	9
3.2.1 Fonction de transfert discrète, définition .....	9
3.2.2 Notation "opérateur de retard $q^{-1}$ " .....	10
3.2.3 Réponse impulsionnelle .....	11
3.2.4 Causalité .....	11
3.2.5 Réponses indicielle .....	11
3.2.6 Régime stationnaire (gain statique) .....	12
3.2.7 Pôles, zéros, stabilité .....	12
3.2.8 Systèmes du premier ordre .....	15
3.2.9 Systèmes du second ordre .....	16
3.2.10 Calcul itératif de la réponse d'un système .....	18
3.2.11 Représentation par équation d'état .....	18
3.3 Réponse fréquentielle des systèmes échantillonnés .....	19
3.3.1 Diagramme de Bode d'un système échantillonné .....	19
3.3.2 Où l'on voit apparaître le théorème de Shannon .....	20
4. Systèmes continus échantillonnés .....	23
4.1 Structure des systèmes commandés par ordinateur .....	23
4.2 Échantillonnage des systèmes continus .....	24
4.3 Effets de l'échantillonnage .....	25
4.4 Où l'on revoit le théorème de Shannon .....	26
4.5 Discrétisation de fonction de transfert continues .....	29
4.5.1 Approximation numérique d'équation différentielle .....	30
4.5.2 Superposition des réponses indicielles .....	32
4.5.3 Correspondances pôles - zéros .....	33
4.5.4 Bilan .....	33
5. Identification des systèmes échantillonnés .....	35
5.1 Identification d'un modèle ARX par la méthode des moindres carrés récurifs .....	35
5.2 L'identification paramétrique en général et en pratique .....	37

5.2.1	Choix du signal d'excitation, traitement des mesures .....	37
5.2.2	Choix d'une structure de modèle.....	38
5.2.3	Choix d'un critère à minimiser.....	38
5.2.4	Choix d'un algorithme.....	38
5.2.5	Validation.....	39
5.2.6	Utilisation de logiciels .....	39
5.3	Identification "simple" .....	39
5.4	Identification non paramétrique .....	41
5.4.1	Estimation de la réponse impulsionnelle .....	41
5.4.2	Estimation de la réponse fréquentielle.....	42
6.	Systèmes bouclés .....	45
6.1	Fonction de transfert en boucle fermée .....	45
6.2	Erreur statique .....	46
6.3	Rejet de perturbations.....	46
6.4	Stabilité des systèmes à simple gain de rétroaction.....	47
6.5	Analyse des systèmes bouclés dans le domaine fréquentiel : stabilité.....	49
6.5.1	Théorème de Nyquist dans le cas où le système est stable en boucle ouverte (critère du revers) .....	50
6.5.2	Théorème de Nyquist dans le cas où le système est instable en boucle ouverte .....	50
6.5.3	Interprétations du lieu de Nyquist.....	50
7.	Méthodes de régulation des systèmes échantillonnés .....	53
7.1	Introduction .....	53
7.2	Structure d'un système commandé par ordinateur, hypothèses.....	54
7.2.1	Discretisation, choix de la période d'échantillonnage.....	54
7.2.2	Discretisation d'un correcteur continu .....	55
7.3	Régulateur PID numérique .....	55
7.3.1	Cas 1 .....	55
7.3.2	Cas 2 .....	58
7.4	Placement de pôles .....	59
7.4.1	Choix des pôles en boucle fermée .....	60
7.4.2	Régulation : calcul de R et S.....	60
7.4.3	Poursuite : calcul de T .....	61
7.5	Poursuite et régulation à objectifs indépendants .....	62
8.	Technologie de la commande par ordinateur.....	65
8.1	Calculateurs et interfaces numériques / analogiques.....	65
8.2	Transport de l'information : les bus et les protocoles .....	67
8.3	Capteurs et actionneurs "intelligents", "smart sensors" .....	68
8.4	La philosophie "hardware in the loop" .....	68
8.5	Le prototypage rapide en automatique .....	68
9.	Annexe .....	69
9.1	Annexe 1 : Résolution de l'équation polynomiale de Bezout.....	69
10.	Bibliographie.....	73
11.	Index.....	1







# 1. INTRODUCTION



Figure 1 : commande d'un systèmes de télémédecine par ordinateur, vu par Claude Serre



## 2. RAPPELS ET PREREQUIS

Ce polycopié ne comporte pas de rappels sur les systèmes continus. S'adressant aux étudiants en deuxième année de l'ENSICA il suppose connu les enseignements de première année et de seconde année en automatique et traitement du signal.

En première année le cours de théorie du signal [2] a rendu familier les notions de :

- transformée de Fourier, transformée de Laplace
- corrélation, convolution

Le cours de traitement numérique du signal [3] a abordé en plus :

- la discrétisation et la quantification des signaux.
- la transformée en z et la transformée de Fourier discrète.
- les filtres numériques RIF et RII.

En automatique [4] les bases de la représentation et de l'analyse des systèmes continus ont été posées. On ne reviendra pas dans ce cours sur les notions de :

- modèles mathématique des systèmes linéaires à temps continus à temps invariant (LTI) : fonction de transfert, équation d'état, pôles, zéros et dynamique.
- commandabilité et observabilité.
- stabilité des systèmes en boucle ouverte : pôles, critères de Routh-Hurwitz.
- stabilité des systèmes bouclés : critère de Nyquist, critère de Bode.
- lieu des racines.

En deuxième année les notions suivantes sont acquises [5] :

- synthèse des correcteurs par méthodes fréquentielles (avance et retard de phase, PID)
- commande par retour d'état (placement de pôles), commande Linéaire Quadratique (LQG)
- reconstruction de l'état par Observateur, observateur optimal de Kalman.



## 3. SIGNAUX ET SYSTEMES DISCRETS

### 3.1 Signaux discrets

#### 3.1.1 Transformée en z

Pour une approche mathématiquement rigoureuse de représentation des signaux discrets par la transformée en z consulter [3] et sa bibliographie. Dans ce cours on se cantonne à une approche utilitariste des outils et méthodes mathématiques tout en essayant de ne pas contredire fondamentalement la rigueur mathématique.

Un signal discret  $s(k)$  est une séquence de nombres réels appelés échantillons. L'indice  $k$  est généralement associé à un instant d'échantillonnage :  $t = k.T_e$ . La période d'échantillonnage  $T_e$  est supposée constante. Par définition la transformée en z du signal  $s(k)$  est la fonction de la variable  $z$ , notée abusivement  $s(z)$  :

$$s(z) = Z(s(k)) = \sum_{k=0}^{\infty} s(k) \cdot z^{-k} \quad (1)$$

La transformée définie ici est plus précisément appelée transformée unilatérale puisque les termes correspondant à  $k < 0$  ne sont pas pris en compte. Ceci n'a pas d'importance puisque les signaux que nous utilisons sont systématiquement causaux<sup>1</sup> :  $s(k) = 0$  pour  $k < 0$ .

Il y a bien évidemment un lien avec la transformée de Laplace et la transformée de Fourier (voir [3])

La fonction de la variable complexe  $s(z)$  est une série et ne converge pas forcément pour toute valeur de  $z$ . En toute rigueur il faudrait associer le rayon de convergence à chaque fonction en  $z$  :  $\{s(z) \text{ converge pour } |z| > \alpha\}$ .

Les automaticiens pour qui la transformée en z n'est bien souvent qu'une façon bien commode de regrouper des termes de récurrence utilisent souvent la pseudo-variable<sup>2</sup>  $q$ .

A partir d'une transformée en z  $s(z)$  il est possible de retrouver la séquence  $s(k)$  correspondante en appliquant la transformation inverse, dont on ne donnera pas la formule ici. A la place de faire appel à cette formule compliquée il est possible d'utiliser d'autres méthodes pour retrouver la séquence  $s(k)$  lorsque la fonction  $s(z)$  est donnée sous forme de fraction rationnelle :

<sup>1</sup> Pour une justification du terme « causal » voir page 11

<sup>2</sup> Voir page 10

- division polynomiale
- décomposition de la fraction rationnelle en éléments simples puis utilisation de transformations élémentaires telles que celles données dans le Tableau 2

La propriété fondamentale de la transformée en z est qu'elle est linéaire (la transformée en z d'une composition linéaire de signaux est la composition linéaire de la transformée en z de chaque signal). Les autres propriétés fondamentales sont données dans le Tableau 2.

Tableau 1 : propriétés de la transformée en z

1	$s(z) = Z(s(k)) = \sum_{k=0}^{\infty} s(k) \cdot z^{-k}$
2	$Z(\alpha \cdot s_1(k) + \beta \cdot s_2(k)) = \alpha \cdot Z(s_1(k)) + \beta \cdot Z(s_2(k))$
3	$Z(k \cdot s(k)) = -z \cdot \frac{ds(z)}{dz}$
4	$Z(c^k \cdot s(k)) = s\left(\frac{z}{c}\right)$
5	$Z(s(k-1)) = s(k=-1) + z^{-1} \cdot s(z)$
6	$Z(s(k+1)) = z \cdot s(z) - z \cdot s(k=0)$
7	$s(k=0) = \lim_{z \rightarrow \infty} (s(z))$
8	$s(k \rightarrow \infty) = \lim_{z \rightarrow 1} \left( \frac{z-1}{z} \cdot s(z) \right)$

Remarque 1 : la propriété 5 ne prend pas en compte le fait que le signal est causal : en pratique  $s(k=-1) = 0$ .

Remarque 2 : la propriété 5 justifie le fait que le terme  $z^{-1}$  soit appelé "opérateur de retard unitaire", cette propriété est aussi appelée "théorème du retard".

Remarque 3 : les propriétés 7 et 8 sont données sous réserve de convergence.

Transformées inverses

### Exemple 1 : transformée inverse par division polynomiale

On considère un signal  $s(k)$  dont la transformée en z est donnée par la fraction rationnelle :

$$S(z) = \frac{0.1 \cdot z^2 + 0.1 \cdot z}{z^3 - 2.6 \cdot z^2 + 2.2 \cdot z - 0.6} \quad (3.1)$$

L'objectif est d'obtenir  $S(z)$  sous forme polynomiale :

$$S(z) = x_0 + x_1 \cdot z^{-1} + x_2 \cdot z^{-2} + \dots \text{etc.} \tag{3.2}$$

Cette suite est généralement infinie, la réponse temporelle est donnée par la succession des termes  $\{x_i\}$ . En traitant l'équation (3.1) comme une simple division l'algorithme de division classique (celui appris en CM2) permet d'obtenir par récurrence tous les termes de la suite (3.2) :

$0.1 \cdot z^2$	$+ 0.1 \cdot z$				$z^3 - 2.6 \cdot z^2 + 2.2 \cdot z - 0.6$ $0.1 \cdot z^{-1} + 0.36 \cdot z^{-2} + 0.716 \cdot z^{-3}$	
$0.1 \cdot z^2$	$- 0.26 \cdot z$	$+ 0.22$	$- 0.06 \cdot z^{-1}$			
0	$0.36 \cdot z$	$- 0.22$	$0.06 \cdot z^{-1}$			
	$0.36 \cdot z$	$- 0.936$	$0.792 \cdot z^{-1}$	$- 0.216 \cdot z^{-2}$		
0	$0.716$	$- 0.732 \cdot z^{-1}$	$\dots$	$\dots$		

Les éléments successifs de la suite  $s(k)$  sont donc  $s(k) = \{0, 0.1, 0.36, 0.716, \dots\}$ . Pour obtenir les termes suivants il faudrait continuer la division.

**Exemple 2 : transformée inverse par décomposition en éléments simples**

La deuxième méthode présentée consiste à décomposer la fraction rationnelle (3.1) en éléments simples puis d'utiliser les tables de transformées en  $z$  (Tableau 2). Pour notre exemple cela donne :

$$S(z) = \frac{0.1 \cdot z^2 + 0.1 \cdot z}{z^3 - 2.6 \cdot z^2 + 2.2 \cdot z - 0.6} = \frac{-z}{z-1} + \frac{0.5 \cdot z}{(z-1)^2} + \frac{z}{z-0.6}$$

La réponse temporelle est donc composée des trois éléments :

$$s(k) = -(1 \ 1 \ 1 \ \dots) + 0.5 \cdot (1 \ 2 \ 3 \ \dots) + (1 \ 0.6 \ 0.6^2 \ \dots)$$

$$\text{soit : } s(k) = \begin{cases} s(0) = -1 + 0.5 + 1 \\ s(2) = -1 + 0.5 \cdot 2 + 0.6 \cdot 0.5 \\ s(3) = -1 + 0.5 \cdot 3 + 0.6^2 \\ s(4) = \dots \end{cases}$$

**3.1.2 Signaux types**

Le Tableau 2 donne les transformée en  $z$  de certains signaux fondamentaux. La Figure 2 représente l'allure de quelques signaux types.

Tableau 2 : transformée en z élémentaires

$s(k)$	$S(z)$
signal temporel	Transformée en z
$\delta(k)$ : impulsion unité	1
$u(k)$ : échelon unité	$\frac{z}{z-1}$
$k \cdot u(k)$	$\frac{z}{(z-1)^2}$
$c^k \cdot u(k)$	$\frac{z}{z-c}$
$k \cdot a^k \cdot u(k)$	$\frac{a \cdot z}{(z-a)^2}$
$\sin(\omega \cdot k) \cdot u(k)$	$\frac{z \cdot \sin(\omega)}{z^2 - 2 \cdot z \cdot \sin(\omega) + 1}$
$\cos(\omega \cdot k) \cdot u(k)$	$\frac{z \cdot (z - \cos(\omega))}{z^2 - 2 \cdot z \cdot \sin(\omega) + 1}$
$a^k \cdot \sin(\omega \cdot k) \cdot u(k)$	$\frac{a \cdot z \cdot \sin(\omega)}{z^2 - 2 \cdot a \cdot \cos(\omega) \cdot z + a^2}$
$a^k \cdot \cos(\omega \cdot k) \cdot u(k)$	$\frac{z \cdot (z - a \cdot \cos(\omega))}{z^2 - 2 \cdot a \cdot \cos(\omega) \cdot z + a^2}$



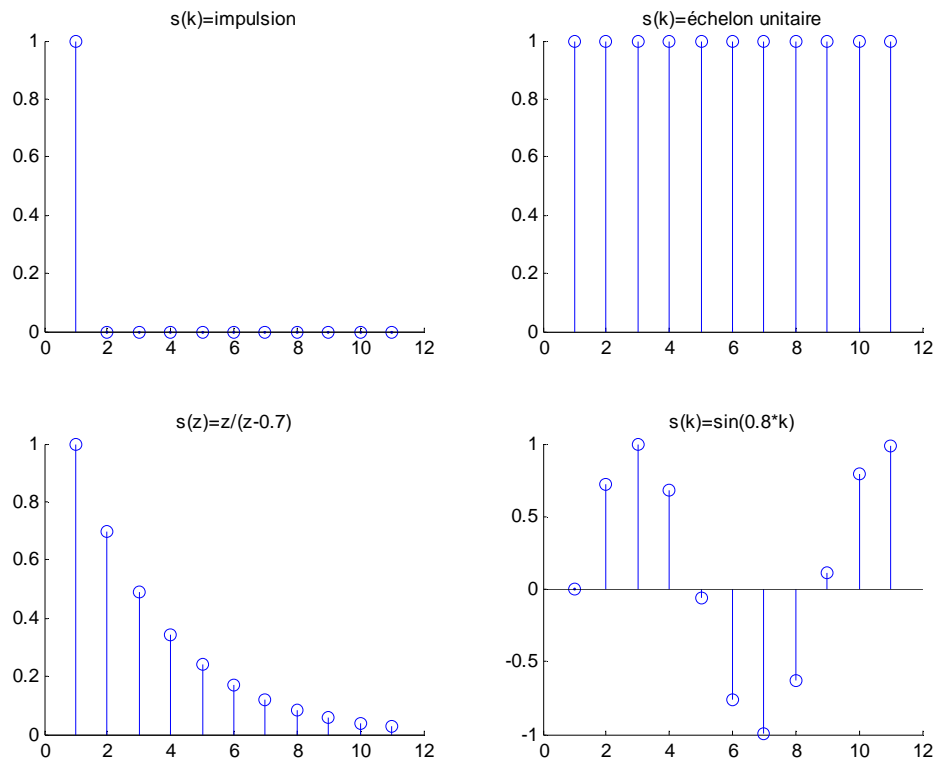


Figure 2 : réponses de quelques signaux types

### 3.2 Systèmes discrets

#### 3.2.1 Fonction de transfert discrète, définition

On considère maintenant un système discret linéaire et invariant dans le temps (LTI) représenté sur la Figure 3.

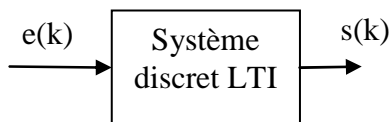


Figure 3 : système discret, notations

La relation entre la séquence d'entrée et la séquence de sortie est linéaire et définie par un ensemble de coefficients constants  $a_{1:n}$  et  $b_{1:m}$  :

$$\begin{aligned} s(k) + a_1 \cdot s(k-1) + a_2 \cdot s(k-2) + \dots + a_n \cdot s(k-n) = \\ b_0 \cdot e(k) + b_1 \cdot e(k-1) + b_2 \cdot e(k-2) + \dots + b_m \cdot e(k-m) \end{aligned} \quad (2)$$

On peut appliquer la transformée en  $z$  à chaque membre de l'équation puis en appliquant en récurrence la propriété 5 (théorème du retard) on obtient la fonction de transfert discrète du système :

$$\frac{s(z)}{e(z)} = F(z) = \frac{b_0 + b_1 \cdot z^{-1} + b_2 \cdot z^{-2} + \dots + b_m \cdot z^{-m}}{1 + a_1 \cdot z^{-1} + a_2 \cdot z^{-2} + \dots + a_n \cdot z^{-n}} \quad (3)$$

Remarque 1 : la fonction de transfert précédente est normalisée en prenant  $a_0 = 1$ .

Remarque 2 : on remarquera que la fonction de transfert est donnée en  $z^{-1}$  plutôt qu'en  $z$ . Cela est justifié par le fait que celle-ci traduit l'équation d'évolution du système physique causal (2).

Le script Matlab permettant de créer un objet Matlab correspondant à une fonction de transfert discrète est donné ci-dessous.

#### Matlab

```
% Création d'un système linéaire discret :
Te=0.1;
sysd = tf([b0 b1 b2],[1 a0 a2 a3],Te,'variable','z^-1');
```

On notera l'utilisation de l'argument 'variable', 'z^-1'.

### 3.2.2 Notation "opérateur de retard $q^{-1}$ "

La fonction de transfert est une fonction de la variable complexe  $z$ . Il faudrait donc en toute rigueur l'associer à son rayon de convergence. En pratique on ne s'embarrasse jamais de préciser cette zone de convergence. C'est pourquoi on trouve souvent la notation utilisant l'opérateur  $q^{-1}$  de décalage à la place de la variable complexe  $z^{-1}$  (4).  $F(q)$  n'est alors plus à proprement parler une fonction mais on lui appliquera les mêmes manipulations algébriques qu'à  $F(z)$ .

$$F(q) = \frac{b_0 + b_1 \cdot q^{-1} + b_2 \cdot q^{-2} + \dots + b_m \cdot q^{-m}}{1 + a_1 \cdot q^{-1} + a_2 \cdot q^{-2} + \dots + a_n \cdot q^{-n}} \quad (4)$$

Attention : selon les auteurs l'opérateur qui remplace  $z^{-1}$  est noté  $q$  (Matlab) ou  $q^{-1}$  (Landau [1]). Dans la suite du document c'est cette dernière convention qui sera appliquée.

L'utilisation de l'opérateur de retard  $q^{-1}$  à la place de la variable complexe  $z^{-1}$  permet de traiter du cas des fonctions de transfert à coefficients non constant, cas non prévu par la transformation en  $z$ .

### 3.2.3 Réponse impulsionnelle

Essayons d'appliquer au système un signal d'entrée particulier : une impulsion unitaire. La transformée en  $z$  du système est obtenue en multipliant la fonction de transfert par la transformée en  $z$  de l'impulsion unitaire : 1. On obtient :

$$s(z) = \frac{b_0 + b_1 \cdot z^{-1} + b_2 \cdot z^{-2} + \dots + b_m \cdot z^{-m}}{1 + a_1 \cdot z^{-1} + a_2 \cdot z^{-2} + \dots + a_n \cdot z^{-n}} \times 1 \quad (5)$$

Cela justifie le fait que la fonction de transfert du système peut aussi s'appeler "réponse impulsionnelle".

#### Matlab

```
% Réponse impulsionnelle d'un système :
[y,t] =impulse(sysd);
```

### 3.2.4 Causalité

Un système est dit causal si les sorties ne dépendent que du passé et non pas du futur. La conséquence est que la réponse impulsionnelle de  $F(z)$  ne peut pas comporter de terme non nuls pour  $k < 0$ . Cela justifie la confusion sémantique entre "système causal" et "signal causal", c'est-à-dire ne comportant pas de termes non nuls pour  $k < 0$ .

Attention : à la différence des systèmes continus un système physique (causal) discret peut très bien être modélisé avec un degré  $m$  supérieur au degré  $n$  : le système peut avoir plus de zéros que de pôles.

La transformée en  $z$  représentant des signaux échantillonnés elle peut tout aussi bien représenter les mesures prises à différents endroits d'une poutre. Le temps étant remplacé ici par une abscisse la notion de causalité ne s'applique pas ! Attention : beaucoup des propriétés de systèmes échantillonnés que l'on étudie sont liés à la causalité, prudence donc si l'on applique ces méthodes à d'autres domaines !

### 3.2.5 Réponses indicielle

Pour caractériser des systèmes on s'intéresse souvent à leur réponse impulsionnelle ou, plus souvent, à leur réponse indicielle. Pour calculer la réponse

à un échelon d'un système il faut donc multiplier la fonction de transfert par la transformée en z d'un échelon unitaire :

$$s(z) = \frac{b_0 + b_1 \cdot z^{-1} + b_2 \cdot z^{-2} + \dots + b_m \cdot z^{-m}}{1 + a_1 \cdot z^{-1} + a_2 \cdot z^{-2} + \dots + a_n \cdot z^{-n}} \times \frac{1}{1 - z^{-1}} \quad (6)$$

Il faudrait ensuite calculer la transformée en z inverse de s(z). En pratique, si l'on doit faire le calcul à la main on décompose en éléments simples puis on utilise des tables. Avec Matlab c'est plus simple. Voici par exemple le script pour obtenir les 20 premiers échantillons de la réponse indicielle d'un système :

**Matlab**

```
% Réponse indicielle d'un système :
[y,t] =step(sysd,20);
```

### 3.2.6 Régime stationnaire (gain statique)

Le gain statique est obtenu en appliquant une entrée indicielle et en calculant la valeur du régime permanent. En appliquant le théorème de la valeur finale il vient :

$$F_0 = \lim_{z \rightarrow 1} \left( \frac{z-1}{z} \cdot \frac{b_0 + b_1 \cdot z^{-1} + b_2 \cdot z^{-2} + \dots + b_m \cdot z^{-m}}{1 + a_1 \cdot z^{-1} + a_2 \cdot z^{-2} + \dots + a_n \cdot z^{-n}} \times \frac{1}{1 - z^{-1}} \right) \quad (7)$$

Soit :

$$F_0 = \frac{\sum_{i=0}^{i=m} b_i}{1 + \sum_{i=1}^{i=n} a_i} \quad (8)$$

Le gain statique est obtenu donc simplement en faisant le quotient de la somme des coefficients.

### 3.2.7 Pôles, zéros, stabilité

Comme pour les systèmes continus la dynamique et la stabilité des systèmes dépend des propriétés du dénominateur de la fonction de transfert.

$$F(z) = \frac{e_0 \cdot (1 + e_1 \cdot z^{-1}) \cdot (1 + e_2 \cdot z^{-1}) \cdots (1 + e_m \cdot z^{-1})}{(1 + c_1 \cdot z^{-1}) \cdot (1 + c_2 \cdot z^{-1}) \cdots (1 + c_n \cdot z^{-1})} \quad (9)$$

Les coefficients  $c_i$  sont les pôles de la fonction de transfert, les coefficients  $e_i$  sont les zéros. La fonction de transfert étant au départ composée de coefficients purement réels les coefficients  $c_i$  sont soit réels soit imaginaires par paires conjuguées.

Une fonction de transfert est donc aussi bien définie par la donnée de ses pôles et de ses zéros. Le script Matlab pour créer un système ainsi est donné ci-dessous :

#### Matlab

```
% Système discret défini par ses pôles et zéros :
Te=0.1;
sysd=zpk([c1 c2 c3],[e1 e2],e0,Te);
```

Si l'on décompose la fonction de transfert en éléments simples on obtient une somme d'éléments du type :

$$F(z) = \frac{d_1}{1 + c_1 \cdot z^{-1}} + \frac{d_2}{1 + c_2 \cdot z^{-1}} + \cdots + \frac{d_n}{1 + c_n \cdot z^{-1}} \quad (10)$$

Un système linéaire temps invariant est dit stable si sa réponse impulsionnelle tend asymptotiquement vers 0. Compte tenu de (10) cela nécessite que chaque pôle  $c_i$  soit de module inférieur à 1.

**Propriété** : un système discret LTI est stable si et seulement si tous ses pôles ont leur modules inférieur ou égal à 1.

#### Critère algébrique attestant de la stabilité : critère de Routh

On rappelle que le critère de Routh permet de prédire si les racines d'un polynôme sont toutes à partie réelle négative, donc stables. Énonçons le théorème de Routh dans le cas d'un polynôme  $P'(w)$  :

$$P'(w) = a_0 + a_1 \cdot w + a_2 \cdot w^2 + \cdots + a_n \cdot w^n \quad (11)$$

Dans un premier temps on construit le tableau :

$$\begin{array}{cccccc|c}
 \beta_{n,0} & \beta_{n,1} & \beta_{n,2} & \beta_{n,q-1} & \beta_{n,q} & 0 & \\
 \beta_{n-1,0} & \beta_{n-1,1} & \beta_{n-1,2} & \beta_{n-1,q-1} & \beta_{n-1,q-1} & 0 & \\
 \beta_{n-2,0} & \beta_{n-2,1} & \beta_{n-2,2} & \beta_{n-2,q-1} & \beta_{n-2,q-1} & & \\
 & & & & & & \\
 \beta_{2,0} & \beta_{2,1} & 0 & & & & \\
 \beta_{1,0} & 0 & & & & & \\
 \beta_{0,0} & 0 & & & & & 
 \end{array} \quad (12)$$

Puis on vérifie que tous les premiers éléments de la première colonne ont le même signe : cela garantit que tous les pôles de  $P'(w)$  ont leur partie réelle négative.

Les éléments du tableau sont construits par récurrence ligne par ligne en appliquant les formules suivantes :

Pour  $i = n$  et  $i = n - 1$  :

$$\begin{cases} \beta_{n,j} = a_{n-2j} \\ \beta_{n-1,j} = a_{n-1-2j} \end{cases} \quad (13)$$

Et pour  $i < n-1$  :

$$\beta_{i,j} = \frac{\beta_{i+1,0} \cdot \beta_{i+2,j+1} - \beta_{i+1,j+1} \cdot \beta_{i+2,0}}{\beta_{i+1,0}} \quad (14)$$

Dans le cas qui nous intéresse, un système échantillonné, on cherche plutôt à vérifier que les pôles du polynôme  $P(z)$  sont de module inférieur à un. L'idée est d'effectuer un changement de variable  $z \rightarrow w$  pour pouvoir appliquer le critère de Routh.

$$P'(w) = (1-w)^n \cdot P\left(\frac{1+w}{1-w}\right) \quad (15)$$

Si l'on effectue la transformation (15) alors le fait que  $P'(w)$  ait toutes ses racines à partie réelle négative garantit que  $P(z)$  a toutes ses racines dans le cercle unité. Pour vérifier la stabilité de  $P(z)$  il suffit donc d'effectuer le changement de variable (15) puis d'appliquer le critère de Routh conventionnel (12 13 14).

### Critère algébrique attestant de la stabilité : critère de Jury

Le critère de Jury s'apparente au critère de Routh. Nous le donnons pour un polynôme de degré 2, 3 et 4 du type :

$$P(z) = a_0 + a_1 \cdot z + a_2 \cdot z^2 + \dots + a_n \cdot z^n \quad (16)$$

Les conditions de stabilité sont données ci-dessous (ces conditions sont données dans le cas  $a_n > 0$ , il suffit de multiplier tous les termes par  $-1$  si cette condition n'est pas remplie par  $P(z)$ ) :

$$\begin{aligned}
 n = 2 & \begin{cases} a_0 + a_1 + a_2 > 0 \\ a_0 - a_1 + a_2 > 0 \\ -a_0 + a_2 > 0 \end{cases} \\
 n = 3 & \begin{cases} a_0 + a_1 + a_2 + a_3 > 0 \\ -a_0 + a_1 - a_2 + a_3 > 0 \\ -|a_0| + a_3 > 0 \\ a_0 a_2 - a_1 a_3 - a_0^2 + a_3^2 > 0 \end{cases} \\
 n = 4 & \begin{cases} a_0 + a_1 + a_2 + a_3 + a_4 > 0 \\ a_0 - a_1 + a_2 - a_3 + a_4 > 0 \\ a_4^2 - a_0^2 - |a_0 a_3 - a_1 a_4| > 0 \\ (a_0 - a_4)^2 (a_0 - a_2 + a_4) + (a_1 - a_3)(a_0 a_3 - a_1 a_4) > 0 \end{cases}
 \end{aligned} \quad (17)$$

### 3.2.8 Systèmes du premier ordre

On donne dans la Figure 4 les réponses indicielles de systèmes du premier ordre ainsi que la localisation des pôles dans le plan complexe. Le cercle de rayon unité représente la zone dans laquelle les pôles doivent se situer pour assurer la stabilité. On remarque sur les deux premières figures l'influence de la place du zéro par rapport au point  $(1,0)$  pour la rapidité.

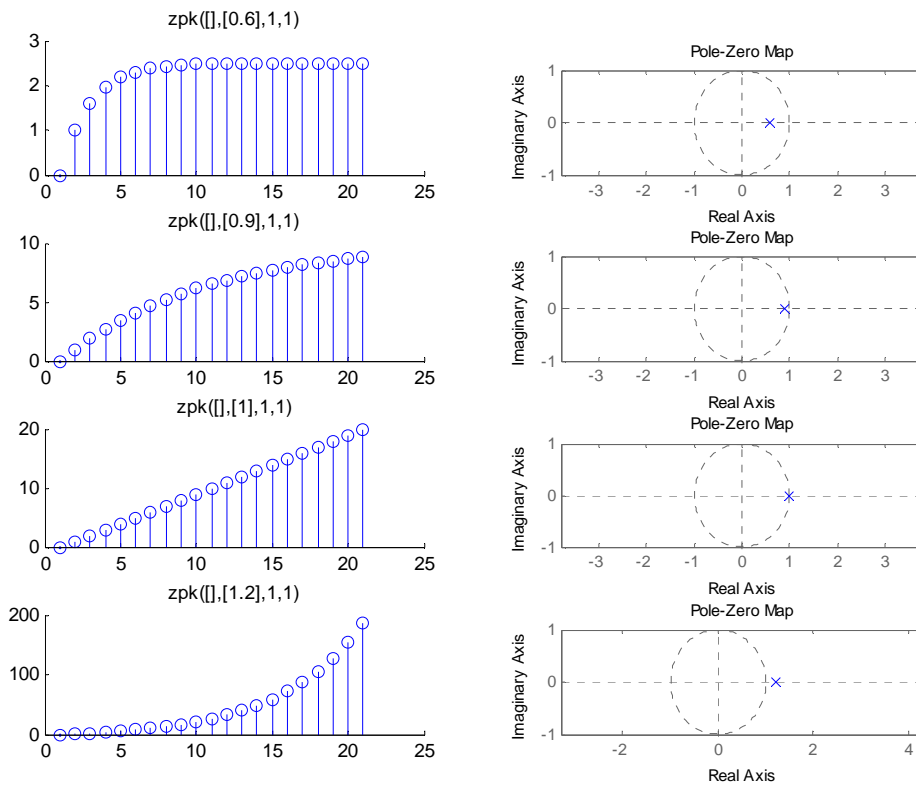


Figure 4 réponses indicielles de systèmes du premier ordre

### 3.2.9 Systèmes du second ordre

On donne dans la Figure 6 les réponses indicielles de systèmes du second ordre ainsi que la localisation des pôles dans le plan complexe. La rapidité et l'amortissement dépend de la position des pôles dans le cercle unité. La Figure 5 représente les courbes iso-amortissement et iso-pulsation des systèmes du second ordre.



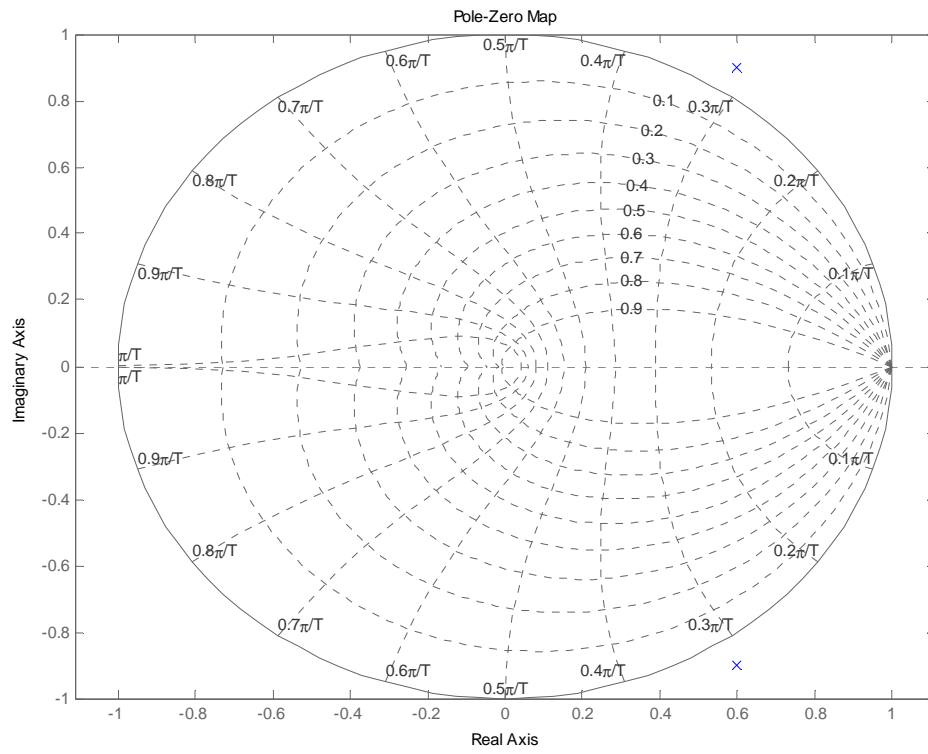


Figure 5 : courbes iso-amortissement et iso-pulsation des systèmes discrets du second ordre

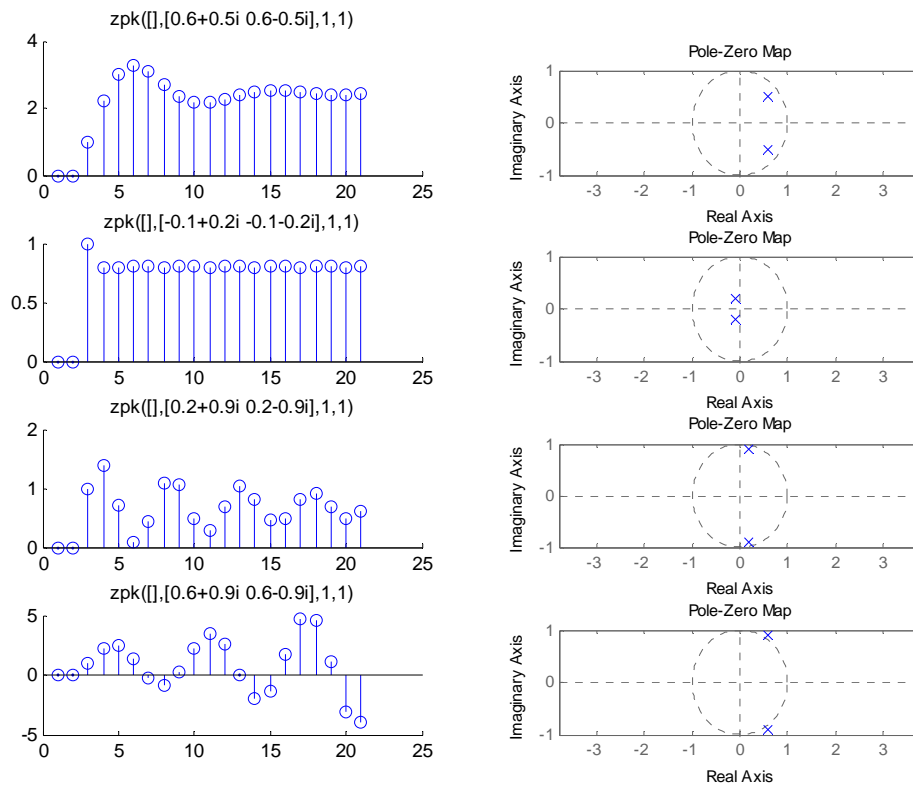


Figure 6 : réponses de systèmes du second ordre

### 3.2.10 Calcul itératif de la réponse d'un système

Si l'on dispose de la fonction de transfert d'un système et que l'on connaît les valeurs successives  $e(k)$  du signal d'entrée il est facile de calculer itérativement la sortie  $y(k)$  du système par l'équation de récurrence (2). Il faut initialiser par les  $n$  valeurs initiales de  $s(k)$ , prises généralement égales à 0.

### 3.2.11 Représentation par équation d'état

Un système discret peut aussi être représenté par un ensemble d'équations d'état comme dans le domaine continu (52). Cette représentation est naturellement adaptée aux systèmes multi-entrées multi-sorties. On définit alors un état interne  $x(k)$  ainsi qu'une fonction d'évolution  $f$  et une équation de mesure  $h$ .

$$\begin{cases} x(k+1) = f(x(k), u(k)) \\ s(k) = h(x(k), u(k)) \end{cases} \quad (18)$$

### 3.3 Réponse fréquentielle des systèmes échantillonnés

#### 3.3.1 Diagramme de Bode d'un système échantillonné

Lorsqu'on étudie la réponse fréquentielle des systèmes continus on considère des entrées particulières sinusoïdales :

$$\begin{cases} e(t) = \cos(2 \cdot \pi \cdot f \cdot t) + j \cdot \sin(2 \cdot \pi \cdot f \cdot t) = e^{j \cdot 2 \cdot \pi \cdot f \cdot t} \\ f \in [0 \quad \infty] \end{cases} \quad (19)$$

De même, pour des systèmes échantillonnés on considèrera des entrées sinusoïdales échantillonnées :

$$\begin{cases} e(k) = \cos(2 \cdot \pi \cdot f \cdot k) + j \cdot \sin(2 \cdot \pi \cdot f \cdot k) = e^{j \cdot 2 \cdot \pi \cdot f \cdot k} \\ f \in [0 \quad 1] \end{cases} \quad (20)$$

La sortie est aussi un signal échantillonné sinusoïdal de gain  $A(f)$  et de phase  $\Phi(f)$  variables et dépendant de  $f$ ; la fréquence réduite.

$$\begin{cases} s(k) = A(f) \cdot e^{j \cdot (2 \cdot \pi \cdot f \cdot k + \Phi(f))} \\ f \in [0 \quad 1] \end{cases} \quad (21)$$

Le nombre complexe, noté abusivement  $F(f)$ , d'amplitude  $A(f)$  et de phase  $\Phi(f)$  est obtenu à partir de  $F(z)$  en remplaçant  $z$  par  $e^{j \cdot 2 \cdot \pi \cdot f}$  :

$$\begin{cases} F(f) = F(z = e^{j \cdot 2 \cdot \pi \cdot f}) \\ f \in [0 \quad 1] \end{cases} \quad (22)$$

Le diagramme représentant le gain et la phase en fonction de la fréquence réduite s'appelle aussi un diagramme de Bode. L'échelle des fréquences est généralement linéaire de 0 à 0.5 (puisque le diagramme est nécessairement symétrique). La Figure 7 représente par exemple le diagramme de Bode d'un système du second ordre.

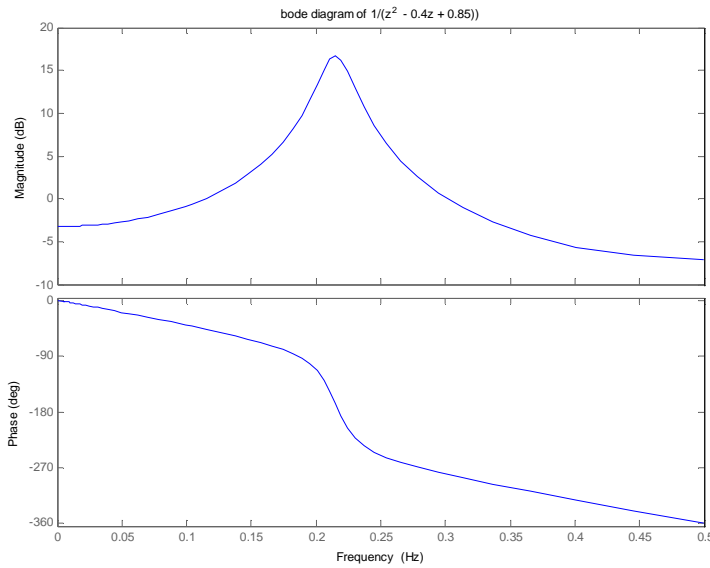


Figure 7 : exemple du diagramme de Bode d'un système discret du second ordre

### 3.3.2 Où l'on voit apparaître le théorème de Shannon

Le changement de variable  $z = e^{j2\pi f}$  n'est pas bijectif puisque  $e^{j2\pi f} = e^{j2\pi(f+k)}$  avec  $k$  entier. Si l'on fait le lien entre domaine continu et domaine fréquentiel on peut remarquer que le changement de variable  $z = e^{j2\pi f}$  fait correspondre une zone du plan complexe continu à une zone du plan complexe discret. La transformation est bijective si l'on se contente de la zone du plan complexe continu contenu entre les droites d'ordonnée 0 et 1.

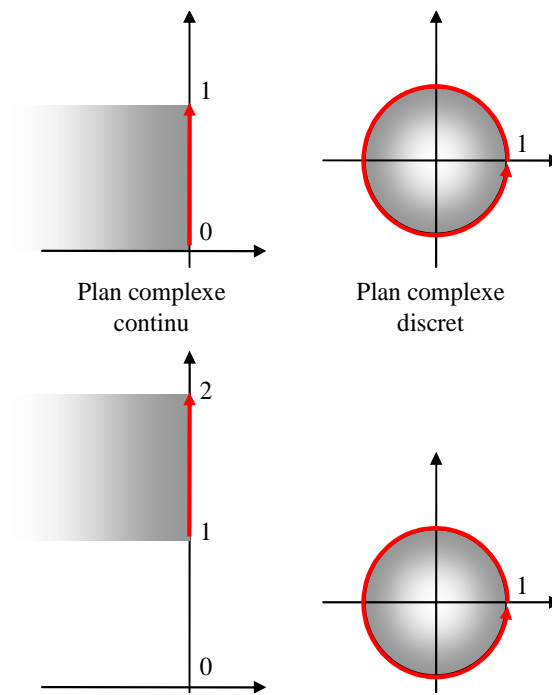


Figure 8 : correspondance entre plan complexe continu et plan complexe discret.



## 4. SYSTEMES CONTINUS ECHANTILLONNES

### 4.1 Structure des systèmes commandés par ordinateur

Les systèmes commandés par ordinateurs sont presque toujours des systèmes physiques continus par nature. Le ordinateur traite par contre les entrées et les sorties de manière discrète avec une période d'échantillonnage  $T_e$ . Les éléments importants, du point de vue de l'automaticien sont donc :

- le convertisseur analogique / numérique CAN ou du moins le système qui fournit séquentiellement les mesures au ordinateur à la période  $T_e$ .
- le ordinateur lui-même. Il doit effectuer l'acquisition des mesures, le traitement des informations, l'envoi des consignes dans un temps inférieur à  $T_e$ .
- le convertisseur numérique / analogique CNA.
- le système physique.

On suppose, pour faire simple et pour pouvoir utiliser la panoplie des outils mathématiques, que les systèmes sont échantillonnés à la même période  $T_e$  et au même moment.

La Figure 9 montre le système physique commandé par le ordinateur, dans sa version simpliste.

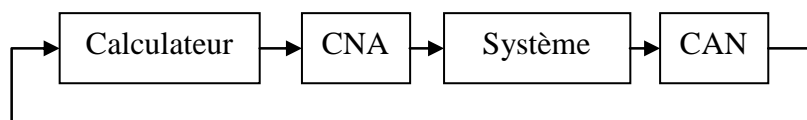


Figure 9 : système commandé par ordinateur

Au XXIème siècle les choses sont quand même parfois un petit peu plus compliquées :

- le ordinateur fournit bien souvent un signal de commande numérique (transporté par bus informatique USB, RS232, CAN, Ethernet, etc...) directement interprété par l'actionneur. On parle "d'actionneur intelligent".
- de même le "capteur intelligent" fournit aussi bien souvent la mesure sous forme d'un signal numérique.
- Les signaux sont transportés par des bus informatiques qui n'ont pas que ça à faire : il peut y avoir du retard.
- de même les capteurs et actionneurs sont rarement tous seuls, ce qui se traduit par des périodes d'échantillonnage différentes, et parfois non synchrones...

- enfin le ordinateur peut aussi avoir d'autres choses à faire...

Ceci étant dit nous allons pour le moment nous restreindre au cas représenté sur la Figure 9 avec des instants d'échantillonnage synchrones et régulièrement espacés d'un temps  $T_e$ .

## 4.2 Echantillonnage des systèmes continus

On dispose généralement de la fonction de transfert continue (notée en utilisant la variable complexe  $s$  ou  $p$ )  $F(s)$  du système continu. Le problème est d'obtenir la fonction de transfert discrète du système échantillonné de manière à pouvoir synthétiser un correcteur. Le système continu est associé au convertisseur numérique – analogique qu'il faut prendre en compte.

Les convertisseurs numérique analogiques peuvent généralement se représenter par des "échantillonneurs – bloqueurs d'ordre 0" ou BOZ. Cela signifie que la commande est maintenue à une valeur constante pendant toute la durée du temps d'échantillonnage.

Si l'on applique une commande discrète impulsionnelle à un BOZ il va maintenir la commande 1 pendant la durée  $T_e$  puis revenir à zéro (Figure 10).

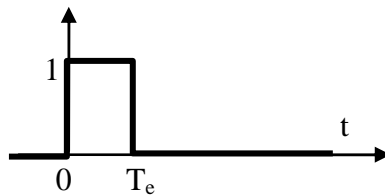


Figure 10 : réponse impulsionnelle d'un bloqueur d'ordre 0

La réponse impulsionnelle  $BOZ(t)$  est donc composée d'un échelon unitaire suivi d'un échelon unitaire de signe opposé et décalé d'un temps  $T_e$ . La transformée de Laplace de la réponse impulsionnelle du bloqueur est donnée par la formule (23) :

$$BOZ(s) = \frac{1}{s} - \frac{e^{-jT_e}}{s} \quad (23)$$

Du point de vue du ordinateur il faut donc prendre en compte le système continu de fonction de transfert  $F_{BOZ}(s)$  que l'on devra échantillonner :

$$F_{BOZ}(s) = \frac{1 - e^{-jT_e}}{s} \cdot F(s) \quad (24)$$



$$F_{\text{BOZ}}(z) = Z^{-1} \left( \frac{1 - e^{-jT_e}}{s} \cdot F(s) \right) \quad (25)$$

En pratique l'obtention de la transformée en z d'un système échantillonné par bloqueur d'ordre 0 se fait à l'aide de tables. On donne quelques exemples fondamentaux dans le Tableau 3 ci-dessous.

Tableau 3 : transformée en z de systèmes continus échantillonnés

F(s)	F <sub>BOZ</sub> (z)
1	1
$\frac{1}{s}$	$\frac{T_e \cdot z^{-1}}{1 - z^{-1}}$
$\frac{1}{1 + T \cdot s}$	$\begin{cases} \frac{b_1 \cdot z^{-1}}{1 + a_1 \cdot z^{-1}} \\ b_1 = 1 - e^{-T_e/T} \quad a_1 = -e^{-T_e/T} \end{cases}$
$\frac{e^{-Ls}}{1 + T \cdot s}$	$\begin{cases} \frac{b_1 \cdot z^{-1} + b_2 \cdot z^{-2}}{1 + a_1 \cdot z^{-1}} \quad a_1 = -e^{-T_e/T} \\ b_1 = 1 - e^{-(L-T_e)/T} \quad b_2 = e^{-T_e/T} (e^{L/T} - 1) \end{cases}$
$L < T_e$	

Si l'on dispose de Matlab c'est bien sûr encore plus simple. On utilise la fonction c2d avec l'argument 'zoh' pour "zero holder hold" (bloqueur d'ordre zéro).

#### Matlab

```
% Définition d'un système continu :
sysc=tf(1,[1 1]);
% Obtention de la transformée en z du système échantillonné à Te
Te=0.1;
sysd=c2d(sysc,Te,'zoh');
% Notation en z^-1
sysd.variable='z'
```

### 4.3 Effets de l'échantillonnage

Le convertisseur numérique a deux effets (néfastes) importants : l'effet de retard et l'effet de quantification.

L'effet de retard se voit dans (24) : on peut dire en première approximation que l'échantillonnage se traduit par un retard pur de  $T_e/2$ . Cela a donc, si ce temps

d'échantillonnage est non négligeable par rapport aux constantes de temps du système, un effet déstabilisant.

Pour illustrer l'effet de quantification supposons par exemple que l'on utilise un convertisseur analogique/numérique 8 bits 0-10V. Le plus petit incrément que le convertisseur peut distinguer est de  $10V/2^8 = 40mV$ . Si le signal utile varie physiquement entre 5V et 5,4V on n'aura que dix incréments utilisables... Ce phénomène n'entre pas dans la panoplie des phénomènes modélisables par la transformée en z. En pratique il est donc souvent nécessaire d'étudier le système et de synthétiser la commande numérique en négligeant ce phénomène puis de vérifier a posteriori qu'il n'a pas un effet trop néfaste.

#### 4.4 Où l'on revoit le théorème de Shannon

Le théorème de Shannon est bien connu en ce qui concerne l'échantillonnage de signaux discrets : on sait que le signal ne doit pas comporter de composante de fréquence supérieure à  $f_e/2$  si l'on ne veut pas observer de repliement. On doit aussi prendre des précautions si l'on veut travailler sur la fonction de transfert discrète correspondant à un système continu échantillonné.

On va traiter un exemple concret pour mettre en évidence le problème de repliement dans le cas de la commande des systèmes continus. On s'intéresse à deux systèmes continus du second ordre ayant des pôles de même partie réelle mais de parties imaginaires séparées de  $2\pi T_e$ . Le script Matlab ci-dessous montre la création des deux systèmes :

##### Matlab

```
% Définition de deux systèmes continus :
sys1=zpk([],[-1-2i -1+2i],2^2+1);
sys2=zpk([],[-1-(2+2*pi)*i -1+(2+2*pi)*i],1+(2+2*pi)^2);
```

On voit sur la Figure 11 que la réponse indicielle des deux systèmes est bien différente.

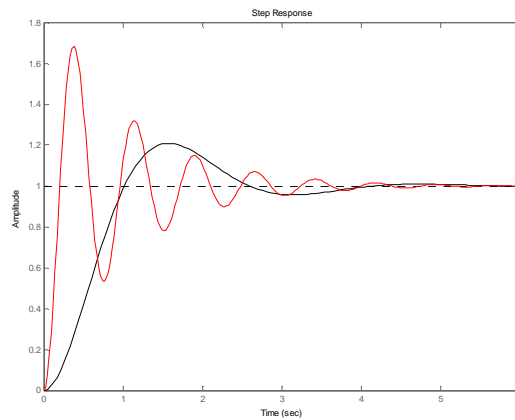


Figure 11 : réponse indicielle de deux systèmes du second ordre

Si l'on calcule la fonction de transfert échantillonnée les deux systèmes à la période de 1 seconde on obtient deux réponses indicielles très proches (représentés par des  $\circ$  sur la Figure 12). Conclusion : la période d'échantillonnage était mal choisie

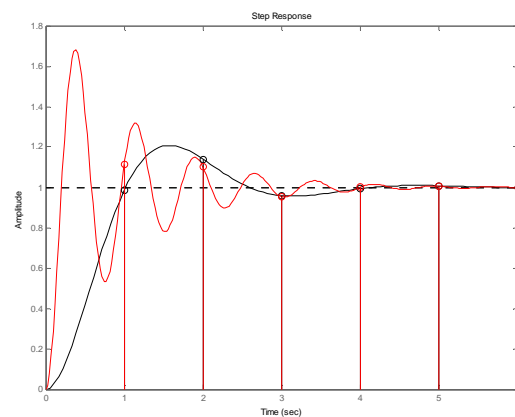


Figure 12 : réponse indicielle de deux systèmes du second ordre échantillonnés à 1s

En pratique la commande échantillonnée de systèmes continus doit être associée à certaines précautions :

- le système ne doit pas comporter de dynamiques propres situées dans les fréquences supérieures à  $f_c/2$  (exemple illustré dans ce chapitre).
- les signaux échantillonnés doivent être "nettoyés" des fréquences supérieures à  $f_c/2$  avec un filtre anti-repliement. (A moins que le signal soit suffisamment propre et ne comporte pas de composantes significatives au-delà de ces fréquences).

### Choix de la période d'échantillonnage

Le choix de la période d'échantillonnage est très important et dépend de la dynamique des signaux. Attention, les systèmes commandés en boucle fermée ont souvent une dynamique plus rapide que le système en boucle ouverte : il faut donc prendre en compte la dynamique en boucle fermée pour le choix de la fréquence d'échantillonnage.

En pratique on préconise de prendre une fréquence 5 à 25 fois supérieure à la bande passante souhaitée en boucle fermée.

On peut être tenté de prendre une fréquence d'échantillonnage très élevée si la technologie le permet (ordinateur et convertisseurs suffisamment rapide). Ceci peut amener rapidement à des problèmes numériques. Imaginons par exemple le cas d'un système ayant une dynamique très lente (quelques secondes) et un système de mesure codant les mesures sur 10 bits (cas classique). La Figure 13 illustre cet exemple. Le signal varie peu entre deux instants d'échantillonnage et les variations du signal sont faibles par rapport à la valeur de l'incrément numérique (figure du haut). Le calcul de la simple dérivée numérique se traduit donc par une succession de  $-1$   $0$  et  $+1$ .

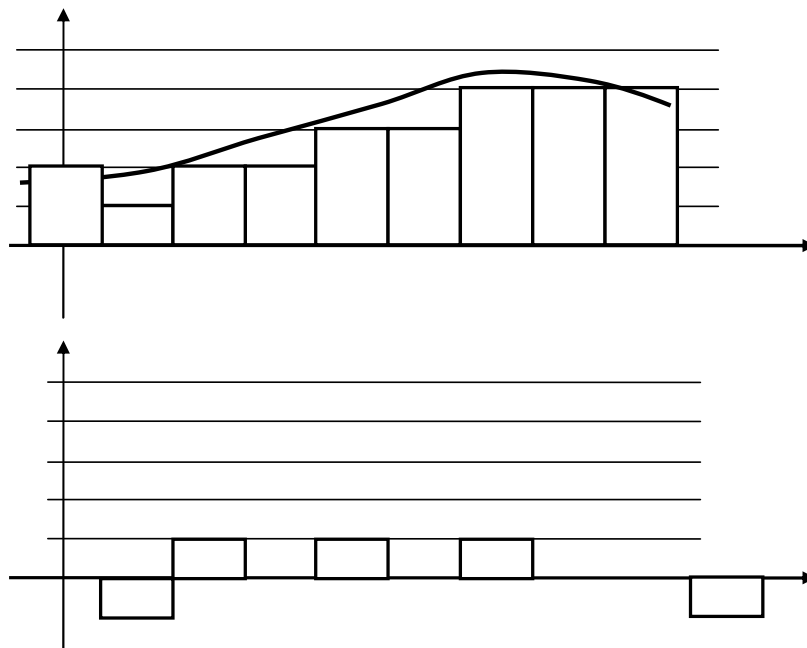


Figure 13 : période d'échantillonnage trop faible. En haut : signal échantillonné et discrétisé ; en bas : calcul de la dérivée numérique.

### Filtre anti-repliement

En toute rigueur chaque fois qu'un signal est échantillonné il faudrait faire précéder le convertisseur numérique d'un filtre analogique anti-repliement éliminant physiquement toutes les composantes du signal de fréquence supérieure à  $f_e/2$ .

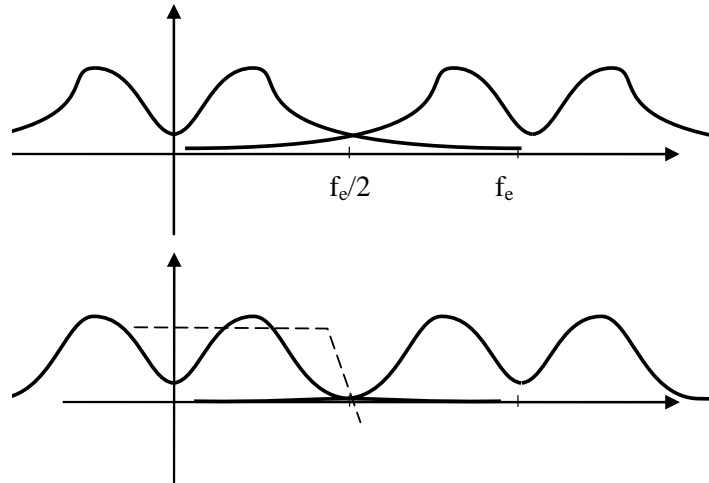


Figure 14 : Effet du filtre anti-repliement

Bien souvent la période d'échantillonnage de l'asservissement est relativement lente (quelques secondes) et il est alors impossible de réaliser un filtre de constante de temps aussi faible. On procède alors à un sur-échantillonnage : le signal est filtré, échantillonné à haute fréquence  $T'_e$  puis filtré numériquement et sous-échantillonné à la période  $T_e$  beaucoup plus grande que  $T'_e$  comme illustré sur la

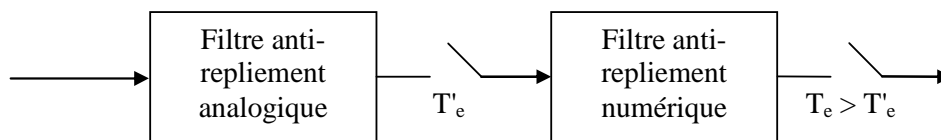


Figure 15 : Sur échantillonnage

Un simple filtre numérique de type moyenne mobile est souvent suffisant.

### 4.5 Discrétisation de fonction de transfert continues

Une manière classique pour réaliser un contrôle par ordinateur est de commencer par synthétiser un correcteur continu avec les outils de l'automatique classique puis de trouver un correcteur discret qui "approche" convenablement le correcteur continu.

#### 4.5.1 Approximation numérique d'équation différentielle

Supposons que le correcteur continu que l'on a synthétisé soit du type PD :

$$\text{PID}(s) = \frac{e(s)}{\varepsilon(s)} = k_p + k_D \cdot s \quad (26)$$

Il s'agit de calculer les commandes  $e(k)$  successives issues du correcteur en fonction des erreurs  $\varepsilon(k)$ . Une approche classique et intuitive est d'approximer la dérivée par une différence :

$$e(k) = k_p \cdot \varepsilon(k) + k_D \cdot \frac{\varepsilon(k) - \varepsilon(k-1)}{T_e} \quad (27)$$

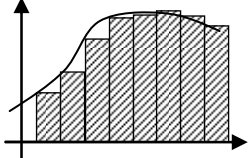
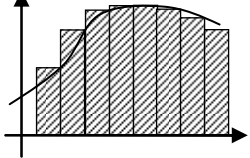
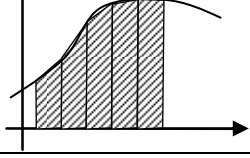
Si on calcule la transformée en  $z$  de  $e(k)$  on obtient :

$$e(z) = k_p \cdot \varepsilon(z) + k_D \cdot \frac{\varepsilon(z) - z^{-1}\varepsilon(z)}{T_e} = \left( k_p + k_D \cdot \frac{1 - z^{-1}}{T_e} \right) \cdot \varepsilon(z) \quad (28)$$

On retrouve la fonction de transfert PD(s) dans laquelle on a effectué le changement de variable  $s = \frac{1-z^{-1}}{T_e}$

De manière plus générale, l'approximation de fonction de transfert revient à effectuer l'approximation numérique d'intégration. Selon la méthode d'intégration choisie on effectue des changements de variable différents.

Tableau 4 : méthodes d'intégration

Méthode	Changement de variable	
Euler progressif (Euler's forward method)	$s = \frac{1 - z^{-1}}{z^{-1} \cdot T_e}$	
Euler retrograde (Euler's backward method)	$s = \frac{1 - z^{-1}}{T_e}$	
Méthode des trapèzes, ou bilinéaire, ou de Tustin	$s = \frac{2}{T_e} \cdot \frac{1 - z^{-1}}{1 + z^{-1}}$	

En pratique le changement de variable  $s = \frac{1 - z^{-1}}{z^{-1} \cdot T_e}$  n'est pas sans risque. On donne sur la Figure 16 la correspondance entre la zone stable du plan complexe en  $s$  et la zone correspondante après la transformation (Euler progressif, Euler rétrograde, bilinéaire). On voit notamment qu'il est possible, partant d'un système continu stable, d'obtenir des pôles dans le domaine discret en dehors du cercle de stabilité si la transformation appliquée est celle d'Euler progressif.

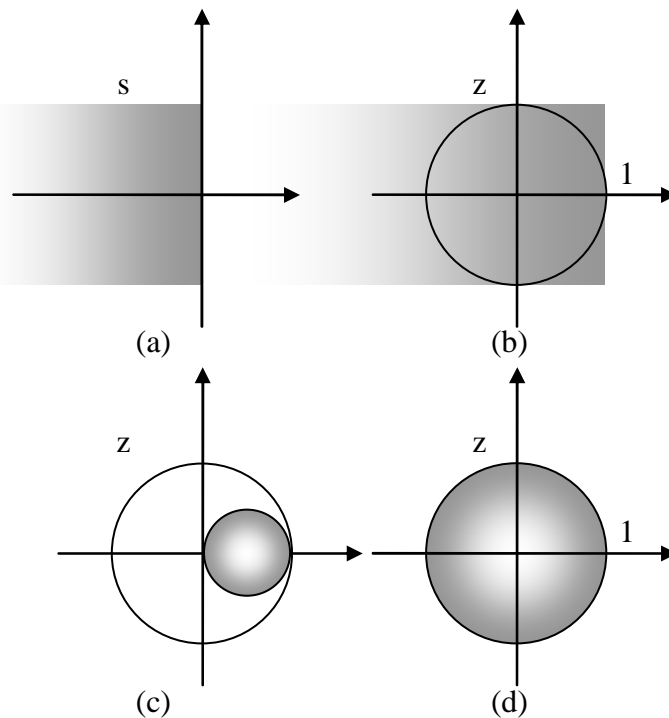


Figure 16 : correspondance des plans complexes. (a) : zone de stabilité dans le plan complexe continu en  $z$  ; (b) zone correspondante avec l'approximation d'Euler progressif ; (c) zone correspondante avec l'approximation d'Euler rétrograde ; (d) zone correspondante avec l'approximation bilinéaire.

La fonction `c2d` de Matlab permet d'obtenir l'équivalent discret d'une fonction de transfert continue avec la méthode d'intégration comme argument :

```

Matlab
sysc=tf([1 0],[1 2 5]);
Te=0.1;
% Méthode d'euler :
sysd1=c2d(sysc,Te,'zoh');
% Méthode des trapèzes :
sysd2=c2d(sysc,Te,'foh');
% Méthode bilinéaire :
sysd3=c2d(sysc,Te,'tustin');
    
```

#### 4.5.2 Superposition des réponses indicielles

Une autre méthode pour approximer une fonction de transfert discrète est de s'assurer que la réponse indicielle de sa version discrétisée correspond exactement lors des temps d'échantillonnage.



### 4.5.3 Correspondances pôles - zéros

La dernière méthode est de partir d'une décomposition de la fonction de transfert continue en pôles et zéros et d'effectuer la conversion pôles à pôles et zéros par zéros. La correspondance, selon que le pôle est réel ou complexe est donnée ci-dessous :

$$\begin{aligned} (s + a) &\rightarrow (z - e^{-a \cdot T_c}) \\ (s + a)^2 + b^2 &\rightarrow z^2 - 2 \cdot e^{-a \cdot T_c} \cdot \cos(b \cdot T_c) \cdot z + e^{-2 \cdot a \cdot T_c} \end{aligned} \quad (29)$$

### 4.5.4 Bilan

Aucune méthode n'est parfaite et on n'aura jamais en même temps :

- même réponse indicielle
- même réponse impulsionnelle
- approximation (Euler, Tustin, etc...) de l'équation différentielle
- correspondance des pôles et des zéros
- correspondance des réponses fréquentielles

Ceci étant dit le choix de la période d'échantillonnage est primordial. Si la fréquence d'échantillonnage est beaucoup plus faible que la dynamique de la fonction de transfert à approximer la différence entre chaque méthode sera minimale.



## 5. IDENTIFICATION DES SYSTEMES ECHANTILLONES

Dans le présent chapitre on donne les bases des méthodes d'identification. Pour une approche complète, voir le cours de troisième année "Identification avion" [6] et les ouvrages [1] et [7].

Même si l'on dispose d'un modèle théorique du processus que l'on souhaite commander il est presque toujours nécessaire de l'affiner en procédant à des essais sur le système réel. Bien souvent le seul modèle dont l'on dispose est ce modèle issu de l'expérimentation. Différentes méthodes s'appliquent selon l'objectif recherché et le niveau de précision exigé. On distinguera :

- les modèles paramétriques
- les modèles simples
- les modèles non paramétriques

### 5.1 Identification d'un modèle ARX par la méthode des moindres carrés récursifs

A partir de cet exemple, le plus simple qui soit, on pose les bases de toutes les méthodes d'identification.

On considère un système d'entrée  $u(t)$  et de sortie  $y(t)$  représenté par l'équation de récurrence suivante :

$$y(t) + a_1 \cdot y(t-1) + a_2 \cdot y(t-2) + \dots + a_n \cdot y(t-n) = b_0 \cdot u(t) + b_1 \cdot u(t-1) + b_2 \cdot u(t-2) + \dots + b_m \cdot u(t-m) \quad (30)$$

L'objectif est, à partir d'essais et de mesures, d'identifier les meilleures valeurs possibles pour les coefficients  $a_i$  et  $b_i$ .

L'équation (30) peut être réécrite pour mettre en évidence la *prédiction* de la sortie  $y(t)$  à partir des mesures passées :

$$y(t) = -a_1 \cdot y(t-1) - a_2 \cdot y(t-2) - \dots - a_n \cdot y(t-n) + b_0 \cdot u(t) + b_1 \cdot u(t-1) + b_2 \cdot u(t-2) + \dots + b_m \cdot u(t-m) \quad (31)$$

Soit :

$$y(t) = \varphi^T(t) \cdot \theta(t)$$

avec :

$$\begin{cases} \varphi^T(t) = [-y(t-1) \ \dots \ -y(t-n) \ , \ u(t-1) \ \dots \ u(t-m)] \\ \theta(t) = [a_1 \ \dots \ a_n \ , \ b_1 \ \dots \ b_m] \end{cases} \quad (32)$$

Le but est de trouver le meilleur ensemble de paramètres  $\theta(t)$  : celui qui, à partir d'un ensemble de mesures  $\varphi(t)$  permet de prédire avec la meilleure précision la sortie effectivement mesurée.

Supposons qu'à la suite d'une expérience on dispose d'un ensemble de mesures  $\{y(t), u(t)\}_{t=1:N}$  et que l'ensemble des paramètres  $\theta$  n'est pas bien connu. On construit la séquence des *sorties prédites* :

$$\hat{y}(t) = \varphi^T(t) \cdot \theta(t) \quad (33)$$

Le paramètre  $\theta$  est celui qui minimise une norme quadratique :

$$\theta = \arg \max \left( \frac{1}{N} \sum_{t=1}^N (y(t) - \hat{y}(t))^2 \right) = \arg \max \left( \frac{1}{N} \sum_{t=1}^N (y(t) - \varphi^T(t) \cdot \theta(t))^2 \right) \quad (34)$$

Compte tenu du fait que l'équation est linéaire au départ et que le terme à minimiser est du type quadratique la solution est simplement obtenue par une pseudo-inverse :

$$\theta = \left[ \sum_{t=1}^N (\varphi(t) \cdot \varphi^T(t)) \right]^{-1} \sum_{t=1}^N (\varphi(t) \cdot y(t)) \quad (35)$$

On obtient donc la solution classique des moindres carrés. La formule (35) est le plus souvent utilisée sous sa forme récursive :

$$\theta(t) = \theta(t-1) + F(t) \cdot \varphi(t) \cdot (y(t) - \theta^T(t-1) \cdot \varphi(t))$$

avec

$$F(t)^{-1} = F(t-1)^{-1} + \varphi(t) \cdot \varphi(t)^T \quad (36)$$

ou

$$F(t) = F(t-1) - \frac{F(t-1) \cdot \varphi(t) \cdot \varphi(t)^T \cdot F(t-1)}{1 + \varphi(t)^T \cdot F(t-1) \cdot \varphi(t)}$$

(On laisse au lecteur le plaisir d'effectuer lui-même les démonstrations, qu'il peut du reste trouver dans [1] et [7]).

## 5.2 L'identification paramétrique en général et en pratique

L'exemple précédent a permis de mettre en évidence tous les aspects de la méthodologie classique de l'identification :

- il a fallu choisir un ensemble de mesures  $\{y(t), u(t)\}_{t=1:N}$ . Le choix du signal d'excitation  $u(t)$  est en pratique primordial,
- il a fallu choisir une structure pour le modèle à identifier (ici les ordres  $n$  et  $m$ ),
- il a fallu choisir un critère (ici un critère quadratique sur l'erreur entre la sortie et la sortie prédite),
- il a fallu choisir un algorithme et le paramétrer (ici l'algorithme des moindres carrés récursifs qu'il a fallu initialiser)

Enfin un point non traité dans le chapitre précédent concerne la validation :

- il aurait fallu déterminer une méthode pour comparer différents modèles et choisir le meilleur

### 5.2.1 Choix du signal d'excitation, traitement des mesures

De manière générale le système que l'on doit identifier doit être excité dans tout le domaine fréquentiel qui nous intéresse. Le signal classiquement utilisé est une séquence pseudo binaire aléatoire (Pseudo Random Binary Signal) tel celui représenté sur la Figure 17.

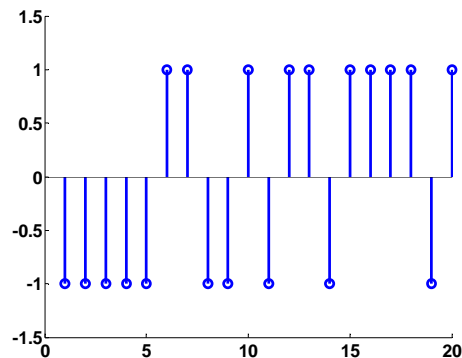


Figure 17 : exemple d'une séquence PRBS

Le signal d'excitation doit se superposer au signal de commande correspondant au point de fonctionnement étudié. Son amplitude doit être faible par rapport à l'amplitude des commandes du système (quelques %).

Une fois le système excité et les mesures effectuées il faut supprimer les valeurs moyennes des mesures.

### 5.2.2 Choix d'une structure de modèle

Le modèle étudié dans le chapitre précédent est de type ARX. On peut le noter sous forme polynomiale :

$$A \cdot y = B \cdot u \quad (37)$$

Ce modèle correspond en fait à un parti-pris sur la manière dont on traite les perturbations. En réalité on prend en compte un signal d'erreurs possibles  $e$  (de moyenne nulle) tels que :

$$A \cdot y = B \cdot u + e \quad (38)$$

Si l'on fait cette hypothèse de modélisation alors la méthode optimale est celle des moindres carrés. Malheureusement cette manière de prendre en compte les perturbations est bien restrictive. D'autres modélisations sont possibles, adaptées à d'autres algorithmes, par exemple le modèle OE :

$$y = \frac{B}{F} \cdot u + e \quad (39)$$

Il est aussi possible de "structurer" le modèle de bruit comme par exemple dans les modèles ARMAX ou BJ :

$$\begin{aligned} A \cdot y &= B \cdot u + C \cdot e \\ y &= \frac{B}{F} \cdot u + \frac{C}{D} \cdot e \end{aligned} \quad (40)$$

La structure du modèle étant choisie il faut choisir l'ordre de ses différents polynômes.

### 5.2.3 Choix d'un critère à minimiser

Dans la plupart des cas le critère à minimiser est l'erreur quadratique entre la sortie et la sortie prédite à un coup (one step ahead prediction). Ceci n'est bien évidemment pas restrictif et peut être généralisé à la prédiction à  $N$  coups.

### 5.2.4 Choix d'un algorithme

Même si chaque structure de modèle est généralement associée à un algorithme "optimal" celui-ci doit généralement se paramétrer.

Prenons par exemple l'algorithme des moindres carrés récursifs tel que présenté dans (36). Cette équation peut être modifiée en intégrant un "facteur d'oubli"  $\lambda$  :

$$\begin{aligned} \theta(t) &= \theta(t-1) + F(t) \cdot \varphi(t) \cdot (y(t) - \theta^T(t-1) \cdot \varphi(t)) \\ \text{avec} & \\ F(t)^{-1} &= \lambda \cdot F(t-1)^{-1} + \varphi(t) \cdot \varphi(t)^T \end{aligned} \tag{41}$$

Ce paramètre (typiquement entre 0.95 et 0.99) permet, comme son nom l'indique, "d'oublier" les anciennes mesures et de se concentrer sur les mesures les plus récentes.

### 5.2.5 Validation

Une fois l'algorithme d'identification appliqué et le modèle obtenu (ce qui prend généralement quelques secondes avec un ordinateur moderne) il reste à expertiser celui-ci. Selon la "qualité" du modèle obtenu on remet en cause l'une ou l'autre des étapes (signal d'excitation, structure du modèle, algorithme) et on réitère jusqu'à obtenir un modèle satisfaisant.

En pratique on essaie un certain nombre de modèles avant de choisir le "meilleur".

Le premier critère de qualité est la capacité du modèle à prédire effectivement la sortie (sortie prédite à un coup, sortie prédite à N coups). Il faut pour cela utiliser un jeu de mesures non utilisé pour l'identification. (En pratique le jeu de mesures initiales est coupé en deux : une partie pour l'identification et une partie pour la validation).

Le second critère de qualité est l'analyse du résidu : la composition spectrale de l'erreur de prédiction.

### 5.2.6 Utilisation de logiciels

En pratique il est commode d'utiliser un célèbre outil de calcul scientifique et sa toolboxe d'identification.

Comme les calculs proprement dit ne prennent que quelques secondes il est facile de réaliser un grand nombre d'identifications et de stocker les différents modèles pour les comparer ensuite. On est rapidement noyé sous le grand nombre de modèles possibles...

## 5.3 Identification "simple"

Il est souvent suffisant d'obtenir un modèle "simple" de type premier ordre et un simple essai indiciel convient souvent. Le système est excité avec un échelon unité et on cherche le meilleur modèle du premier ordre correspondant.

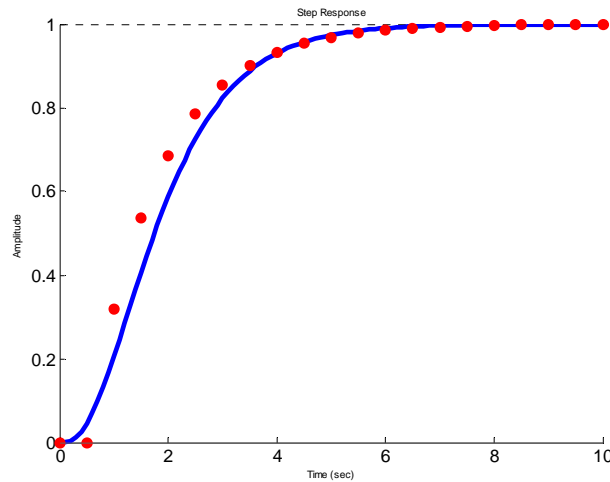


Figure 18 : identification sur un essai indiciel

Dans l'exemple de la Figure 18 on superpose le résultat d'une mesure (en bleu) avec la simulation d'un modèle discret avec une période de 0.5s. Le modèle cherché est du premier ordre avec un retard pur  $d$  :

$$y(z) = \frac{(1 + c_1) \cdot z^{-d}}{1 + c_1 \cdot z^{-1}} \cdot u(z) \quad (42)$$

Il s'agit de trouver l'entier  $d$  et le nombre réel  $c_1$  tels que la réponse indicielle du modèle corresponde pour le mieux aux mesures issues de l'essai. Ceci peut se formaliser par exemple de la manière suivante :

$$(c_1 \quad d_1) = \arg \min \sum_{t=1}^N (y_{\text{modèle}}(t) - y_{\text{mesure}}(t))^2 \quad (43)$$

Cette optimisation peut se faire à la main ou grâce à une procédure automatisée (avec le célèbre outil de calcul scientifique).

Dans le cas où le système comprend naturellement une intégration (par exemple la commande d'un moteur à courant continu par une tension et la mesure de l'angle de rotation) il est utile de prendre en compte explicitement cette intégration dans le modèle :

$$y(z) = \frac{1}{1 - z^{-1}} \cdot \frac{(1 + c_1) \cdot z^{-d}}{1 + c_1 \cdot z^{-1}} \cdot u(z) \quad (44)$$



La procédure est la même mais au lieu de prendre en compte la sortie  $y(t)$  on prend en compte la dérivée  $y'(t)$  :

$$y'(t) = y(t) - y(t-1) \quad (45)$$

## 5.4 Identification non paramétrique

Pour terminer ce panorama rapide sur les méthodes d'identification il faut parler des méthodes non paramétriques : celles qui permettent de tracer la réponse impulsionnelle ou la réponse fréquentielle du système.

### 5.4.1 Estimation de la réponse impulsionnelle

La réponse impulsionnelle d'un système peut être obtenue simplement en appliquant une entrée de type "impulsion" au système et d'en mesurer les conséquences.

La seconde méthode consiste à utiliser la méthode des moindres carrés (ou une méthode dérivée) comme vue au chapitre 5.1. La réponse impulsionnelle d'un système peut en effet être vue comme la fonction de transfert particulière où le dénominateur est de degré nul et le numérateur comporte un nombre infini (en pratique : grand) d'éléments ( $n = 0$  et  $m$  grand) :

$$y(t) = b_0 \cdot u(t) + b_1 \cdot u(t-1) + b_2 \cdot u(t-2) + \dots + b_m \cdot u(t-m) \quad (46)$$

La dernière méthode, qui sera développée ici est l'analyse d'intercorrélation.

Par définition l'intercorrélation  $R_{yu}$  entre deux signaux est :

$$R_{yu}(\tau) = E[(y(t) - E(y(t))) \cdot (u(t-\tau) - E(u(t-\tau)))] \quad (47)$$

Dans cette formule l'espérance mathématique peut être approchée grâce à la loi des grands nombres :

$$R_{yu}(\tau) \approx R_{yu}^N(\tau) = \lim_{N \rightarrow \infty} \frac{1}{N} \sum_{t=1}^N [(y(t) - m_y) \cdot (u(t-\tau) - m_u)] \quad (48)$$

On s'intéresse maintenant à un système tel que décrit par l'équation impulsionnelle (46) et on suppose que le signal d'entrée est un bruit blanc. Le signal d'entrée vérifie donc :

$$\begin{cases} \tau \neq 0 & R_{uu}(\tau) = 0 \\ \tau = 0 & R_{uu}(\tau) = R \end{cases} \quad (49)$$

Il vient alors naturellement :

$$b_k \approx \frac{1}{R} R_{yu}(k) \quad (50)$$

#### 5.4.2 Estimation de la réponse fréquentielle

On utilise ici le spectre d'intercorrélation. Par définition le spectre d'intercorrélation entre deux signaux est :

$$\Phi_{yu}(\omega) = \sum_{\tau=-\infty}^{\infty} R_{yu}(\tau) \cdot e^{-i\omega\tau} \quad (51)$$

On suppose que le signal d'entrée  $u(t)$  et le signal de sortie  $y(t)$  sont reliés par une fonction de transfert linéaire du type :

$$y(t) = F \cdot u(t) + v(t) \quad (52)$$

où  $v(t)$  est une perturbation indépendante de  $u(t)$ . On a alors les relations :

$$\Phi_{yu}(\omega) = F(j\omega) \cdot \Phi_{uu}(\omega) \quad (53)$$

Il reste à estimer les spectres d'intercorrélation pour remonter à la réponse fréquentielle  $F(j\omega)$  recherchée. Une idée naturelle est d'utiliser la relation (48) mais cela marche très mal dans presque tous les cas, même en choisissant  $N$  assez grand. L'amélioration est d'utiliser une "fenêtre" pour favoriser les valeurs faibles de  $\tau$  par rapport aux valeurs élevées. La solution classique utilisant la *fenêtre de Hamming* donne :

$$\Phi_{yu}^N(\omega) = \sum_{\tau=-\gamma}^{\gamma} R_{yu}^N(\tau) \cdot w_{\gamma}(\tau) \cdot e^{-i\omega\tau} \quad (54)$$

avec

$$\begin{cases} w_{\gamma}(\tau) = \frac{1}{2} \left( 1 + \cos\left(\frac{\pi \cdot \tau}{\gamma}\right) \right) & |\tau| < \gamma \\ w_{\gamma}(\tau) = 0 & |\tau| > \gamma \end{cases}$$

On utilise ensuite la relation (51) pour estimer la réponse fréquentielle :

$$F(j \cdot \omega) = \frac{\Phi_u^N(\omega)}{\Phi_{yu}^N(\omega)} \quad (55)$$



## 6. SYSTEMES BOUCLES

### 6.1 Fonction de transfert en boucle fermée

Un système et sa commande par ordinateur peut être représenté par un ensemble de fonctions de transfert formant un schéma-bloc plus ou moins compliqué (Figure 19).

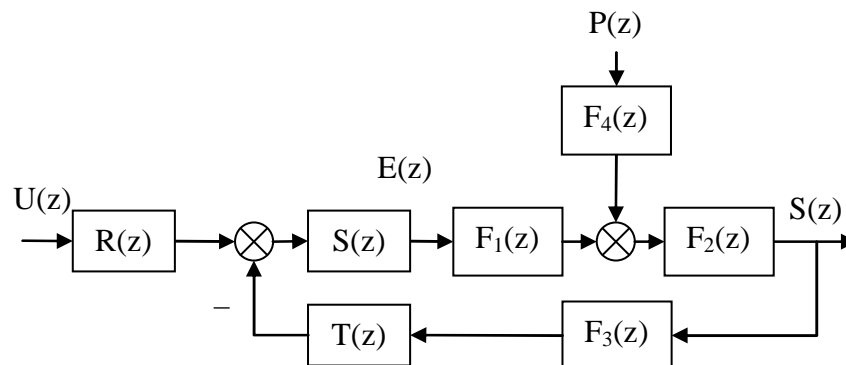


Figure 19 : système discret commandé par ordinateur. Le correcteur est représenté par les fonctions R,S,T, le système par  $F_1$  et  $F_2$ , le modèle de perturbations par  $F_4$  et le modèle des actionneurs par  $F_4$ .

Quelle que soit la structure d'un système asservi il est possible de se ramener à une forme canonique comprenant une fonction de transfert en boucle ouverte en combinant judicieusement les fonctions de transfert du procédé, du régulateur et des perturbations. La Figure 20 représente par exemple le système de la Figure 19 où  $F$  est une combinaison des fonctions de transfert du système, du régulateur, des perturbations et des capteurs.

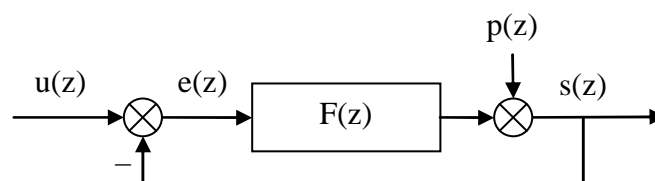


Figure 20 : système discret, notations.

Par défaut on considère une contre-réaction négative et une perturbation additive sur la sortie. La fonction de transfert  $F$  est appelée *fonction de transfert en boucle ouverte*. Comme pour les systèmes continus la *fonction de transfert en boucle fermée* est obtenue simplement à partir de  $F$  (58).

$$F(z) = \frac{s(z)}{e(z)} = \frac{B(z^{-1})}{A(z^{-1})} \quad (56)$$

$$G(z) = \frac{s(z)}{u(z)} = \frac{B(z^{-1})}{A(z^{-1}) + B(z^{-1})}$$

## 6.2 Erreur statique

L'erreur statique du système en boucle fermée, comme pour le système en boucle ouverte, est simplement le rapport des coefficients.

$$G_0 = \frac{\sum_{i=0}^{i=m} b_i}{1 + \sum_{i=0}^{i=m} b_i + \sum_{i=1}^{i=n} a_i} \quad (57)$$

Si l'on souhaite que l'asservissement confère un gain statique unitaire au système en boucle fermée (c'est-à-dire que l'on force la sortie  $s(k)$  à tendre vers la valeur de consigne  $u(k)$ ) il faut que ce rapport de coefficients soit égal à un. On montre facilement que cela est obtenu si la fonction de transfert  $F(z)$  comporte un intégrateur :

$$F(z) = \frac{s(z)}{e(z)} = \frac{1}{1 - z^{-1}} \cdot \frac{B(z)}{A'(z)} \quad (58)$$

## 6.3 Rejet de perturbations

La fonction de transfert entre la perturbation  $p(z)$  et la sortie  $y(z)$  est appelée fonction de sensibilité perturbation-sortie, notée  $S_{yp}$ .

$$S_{yp}(z) = \frac{s(z)}{p(z)} = \frac{A(z^{-1})}{A(z^{-1}) + B(z^{-1})} \quad (59)$$

Si l'on souhaite qu'une perturbation de type échelon soit parfaitement rejetée il faut que  $A(1) = 0$ , on retrouve donc la condition (58) à savoir que le système doit comporter une intégration dans la boucle directe. On peut être en outre être amené à spécifier que les perturbations ne soient pas amplifiées dans une certaine bande de fréquence. On rajoute donc une contrainte du type  $|S_{yp}(e^{j2\pi f})| < S_{\max}$  pour  $f$  appartenant à un certain domaine de  $[0 \ 0.5]$ .

## 6.4 Stabilité des systèmes à simple gain de rétroaction

Les critères de Routh et de Jury (donnés dans le chapitre 3.2.7 p13) ou le calcul numérique des pôles permettent de déterminer la stabilité du système en boucle fermée si l'on connaît sa fonction de transfert.

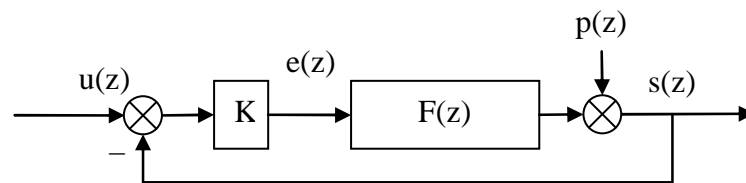


Figure 21 : système bouclé par un simple gain.

On s'intéresse ici au cas particulier où l'asservissement est réalisé par un simple gain de rétroaction  $K$  (Figure 21). On cherche la valeur de  $K$  qui amène le système en limite de stabilité. L'outil adapté ici est le lieu d'Evans ou lieu des racines (root locus) qui permet de visualiser l'évolution des pôles du système bouclé lorsque le gain  $K$  varie.

Prenons l'exemple d'un système comportant trois pôles et deux zéros. L'exemple est donné sous forme de script Matlab :

### Matlab

```
% Définition d'un système discret :
sys=zpk([-2.037 -0.1406],[0.6065 0.3679 0.3679],1,1)
%Tracé du lieu des racines
rlocus(sys);
```

Le script précédent produit le tracé de la Figure 22. On pourrait y lire que la limite de stabilité est obtenue pour  $K=0.2$  environ (détail de la Figure 23).

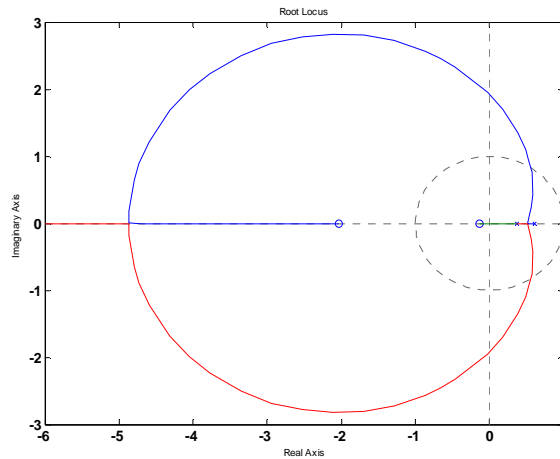


Figure 22 : lieu des racines

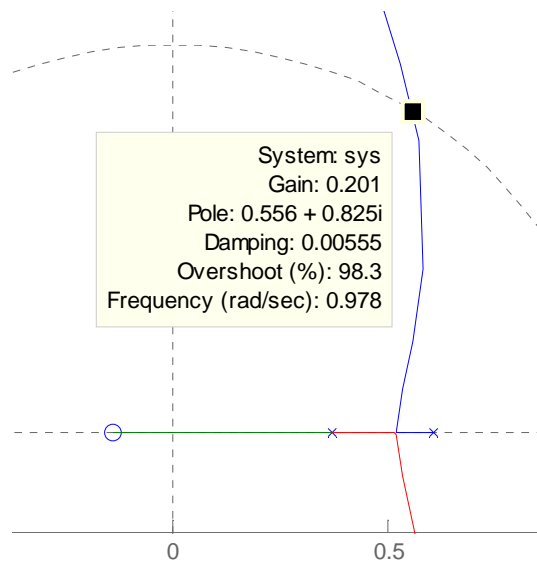


Figure 23 : lieu des racines : détail de la limite de stabilité

On voit sur la Figure 22 que le lieu des racines d'un système échantillonné ressemble à celui d'un système continu :

- on commence par tracer les pôles ( $\times$ ) et les zéros ( $o$ ) du système en boucle ouverte.
- les pôles du système en boucle fermée décrivent un lieu qui part des pôles (pour  $K = 0$ ) et arrive aux zéros (pour  $K = \infty$ )
- lorsque il y a  $n-m$  plus de pôles que de zéros le système comporte  $n-m$  branches asymptotiques.

On ne décrira pas ici les diverses propriétés du lieu des racines qui permettent de le tracer à la main puisque il est facile à obtenir avec un ordinateur.



Il peut aussi être utilisé pour régler "à la main" le gain permettant d'obtenir un ou plusieurs modes dominants ayant certaines propriétés de fréquence et d'amortissement. On voit sur la Figure 24 que le gain 0.017 permet d'obtenir un pôle double d'amortissement 0.706 et de fréquence 0.65 rad/s et un pôle simple de fréquence 1.63 rad/s.

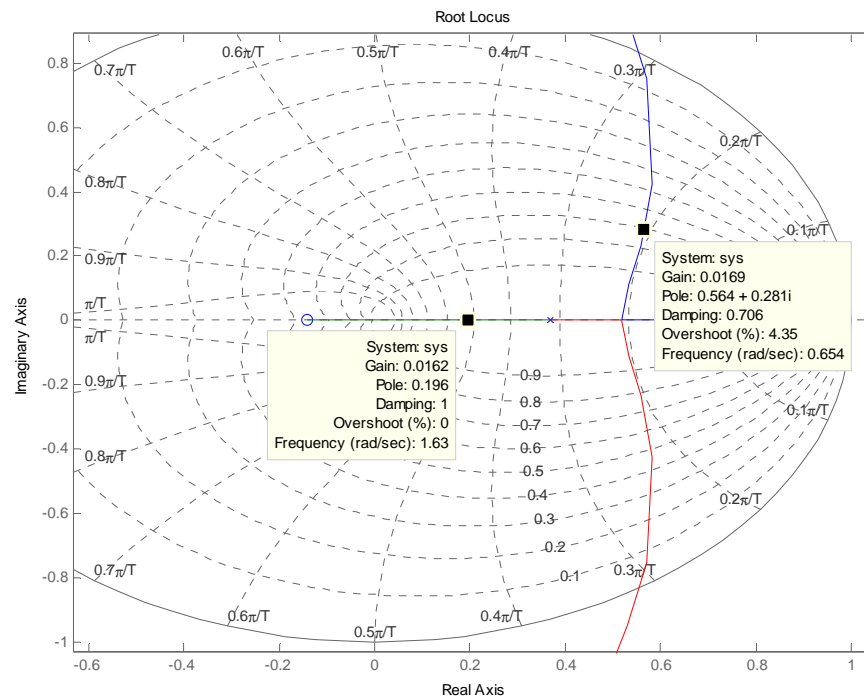


Figure 24 : utilisation du lieu des racines pour régler le gain de la boucle fermée

L'outil rltool de Matlab permet de nombreuses manipulations graphiques de ce type.

## 6.5 Analyse des systèmes bouclés dans le domaine fréquentiel : stabilité

Comme dans le cas continu il est possible de prévoir des propriétés, en particulier la stabilité, d'un système en boucle fermée à partir de l'examen de la fonction de transfert en boucle ouverte.

On ne redémontrera pas ici le théorème de Nyquist mais on se contente de rappeler l'énoncé.

On rappelle que le *lieu de Nyquist* est le lieu de  $F(j2\pi f)$  dans le plan complexe pour  $f$  variant de 0 à  $+\infty$ . A noter que le lieu pour la zone  $[0, 0.5]$  est symétrique de celui pour la zone  $[0.5, 1]$ . Pour cette raison on ne représente généralement que le lieu pour  $f$  appartenant à  $[0, 0.5]$ .

### 6.5.1 Théorème de Nyquist dans le cas où le système est stable en boucle ouverte (critère du revers)

Le système est stable en boucle fermée si le lieu de Nyquist de la fonction de transfert en boucle ouverte parcouru dans le sens des fréquences croissantes de 0 à 0.5 laisse le point critique  $(-1,0)$  à gauche.

La Figure 25 représente par exemple le lieu de Nyquist d'une fonction de transfert en boucle ouverte qui sera stable en boucle fermée.

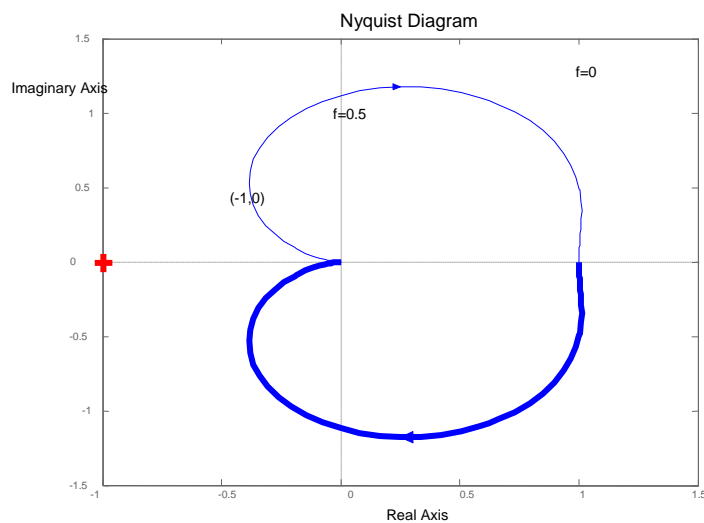


Figure 25 : lieu de Nyquist d'un système stable

### 6.5.2 Théorème de Nyquist dans le cas où le système est instable en boucle ouverte

Si le système en boucle ouverte comporte  $N$  pôles instables il sera stable en boucle fermée si le lieu de Nyquist de la fonction de transfert en boucle ouverte parcouru dans le sens des fréquences croissantes de  $-0.5$  à  $0.5$  entoure le point critique  $(-1,0)$   $N$  fois dans le sens des aiguilles d'une montre..

### 6.5.3 Interprétations du lieu de Nyquist

Comme dans le cas continu, la distance à laquelle on passe du point critique renseigne sur la robustesse du système bouclé. La distance sera exprimée de différente manière (Figure 26) :

- marge de module  $\Delta M$  : plus petite distance entre le lieu des racines et le point critique,

- marge de gain  $\Delta G$
- marge de phase  $\Delta P$

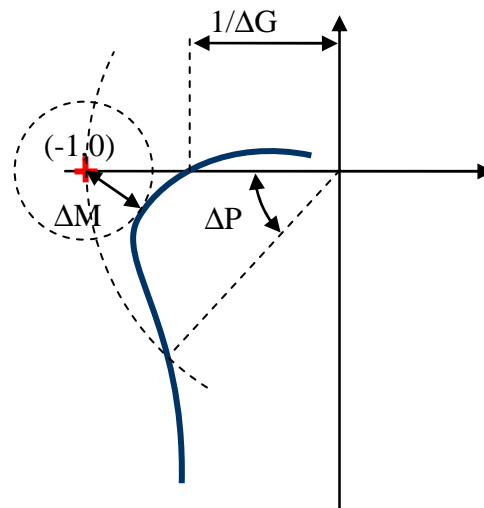


Figure 26 : Marges de stabilité

Les valeurs typiques de ces marges sont données ci-après :

- $\Delta G \geq 6\text{dB}$
- $\Delta M \geq -6\text{dB}$
- $30^\circ \leq \Delta P \leq 60^\circ$

La distance entre le point critique et le lieu des racines est la norme de  $1+F(j2\pi f)$ . On peut donc exprimer la marge de module en fonction de la fonction de sensibilité :

$$\begin{aligned} \Delta M &= \min |1 + F(j \cdot \omega)|_{\omega \in [0, \pi]} \\ \Delta M &= \min \left| 1 + \frac{B(j \cdot \omega)}{A(j \cdot \omega)} \right|_{\omega \in [0, \pi]} = \max \left| \frac{A(j \cdot \omega)}{A(j \cdot \omega) + B(j \cdot \omega)} \right|_{\omega \in [0, \pi]} \quad (60) \\ \Delta M &= \max |S_{yp}(j \cdot \omega)|_{\omega \in [0, \pi]}^{-1} \\ \Delta M_{\text{dB}} &= -\max |S_{yp}(j \cdot \omega)|_{\text{dB}} \end{aligned}$$

Si l'on trace l'évolution de  $S_{yp}$  en fonction de  $f$  on obtient donc la valeur de la marge de module.



# 7. METHODES DE REGULATION DES SYSTEMES ECHANTILLONES

## 7.1 Introduction

Les bases théoriques permettant de concevoir le correcteur pour un système échantillonné ont été posées dans les chapitres précédents. Ce chapitre fait le bilan des connaissances acquises et donne quelques solutions pour calculer des correcteurs à partir d'un cahier des charges précis.

Les systèmes que l'on cherche à commander sont généralement, par nature, continus. La réalisation d'un correcteur par un ordinateur implique que l'on génère une équation de commande discrète (généralement une équation récurrente). L'obtention de cette équation de commande peut se faire de deux manières :

- la première méthode consiste à d'abord synthétiser un correcteur continu (en utilisant les outils classiques de l'automatique et la simulation) et discrétiser ensuite ce correcteur. C'est l'objet du chapitre 7.2.2.
- la seconde méthode consiste à tenir la représentation discrète du système continu associé à son interface numérique (convertisseurs analogique-numérique et numérique-analogique) comme expliqué dans le chapitre 4.5 puis synthétiser directement le correcteur discret. Cela sera expliqué dans le chapitre 7.3 et suivants.

Ce choix peut être résumé par la figure suivante :

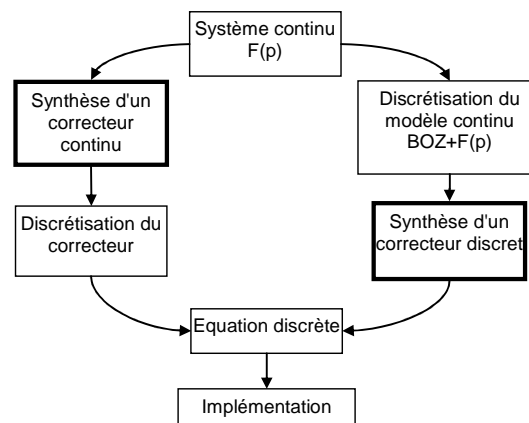


Figure 27 : Réalisation d'un correcteur discret.

Au-delà des affaires de goût, le choix de la méthode sera souvent dicté par l'application. En effet, la première méthode est d'autant meilleure que le correcteur discrétisé approche correctement le correcteur continu. Cela implique que le temps d'échantillonnage soit très faible par rapport aux constantes de temps du système

bouclé. Dans le cas où les contraintes temps réel sont fortes (capacités de calcul partagées par exemple) il sera préférable de synthétiser un correcteur discret.

## 7.2 Structure d'un système commandé par ordinateur, hypothèses

Par la suite on se ramènera toujours à un système tel que celui représenté sur la Figure 28 en cherchant un correcteur  $K(z)$  linéaire. On montre que quel que soit le correcteur linéaire on peut se ramener à la forme représenté sur la Figure 29 (forme RST).

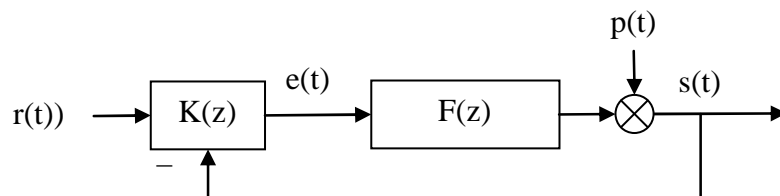


Figure 28 : système bouclé, forme générale.

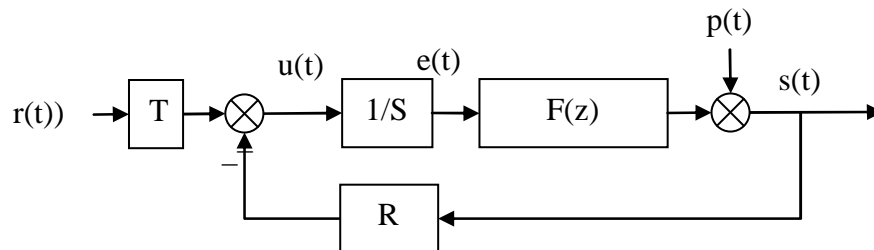


Figure 29 : système bouclé, forme canonique.

### 7.2.1 Discrétisation, choix de la période d'échantillonnage

Du point de vue du ordinateur, l'extérieur (le procédé) est représenté par la fonction de transfert  $F(z)$ . L'hypothèse importante est que la commande (sortie du ordinateur) et la mesure (entrée du ordinateur) sont échantillonnées de manière synchrone à la même période  $T_e$ .

Le choix judicieux de la fréquence d'échantillonnage est très important. En pratique on choisit cette fréquence en fonction de la bande passante du système en **boucle fermée** :

$$f_e = (6 \text{ à } 25) f^{BF} \quad (61)$$

### 7.2.2 Discrétisation d'un correcteur continu

La première méthode pour obtenir les équations d'un correcteur numérique est de faire sa synthèse avec les outils de l'automatique continue puis de discrétiser la fonction de transfert du correcteur continu obtenu (voir pour cela le chapitre 4.2 *Echantillonnage des systèmes continus*).

On va voir par la suite que la synthèse directe d'un correcteur discret offre plus de degré de liberté et permet en général de remplir un cahier des charges plus complet.

## 7.3 Régulateur PID numérique

Le régulateur le plus populaire est le régulateur PID. Nous allons voir dans un premier temps comment on génère un régulateur PID à partir des équations bien connues du régulateur PID continu et comment ces équations peuvent être généralisées pour obtenir plus de degrés de liberté pour le réglage.

### 7.3.1 Cas 1

Un régulateur PID continu a une fonction de transfert du type :

$$K_{\text{PID}}(s) = K \cdot \left( 1 + \frac{1}{T_i \cdot s} + \frac{T_d \cdot s}{1 + \frac{T_d}{N} \cdot s} \right) \quad (62)$$

La discrétisation de ce correcteur continu par la méthode d'Euler (Tableau 4 page 31) se fait en remplaçant la variable de Laplace  $s$  par le terme  $\frac{1-q^{-1}}{q^{-1} \cdot T_e}$ . On obtient :

$$K_{\text{PID}}(q) = K \cdot \left( 1 + \frac{1}{T_i \cdot \frac{1-q^{-1}}{T_e}} + \frac{T_d \cdot \frac{1-q^{-1}}{T_e}}{1 + \frac{T_d}{N} \cdot \frac{1-q^{-1}}{T_e}} \right) \quad (63)$$

En réarrangeant on obtient la forme générale RST (Figure 30):

$$K_{PID}(q) = K \cdot \frac{r_0 + r_1 \cdot q^{-1} + r_2 \cdot q^{-2}}{(1 - q^{-1}) \cdot (1 + s_1' \cdot q^{-1})}$$

$$\begin{cases} r_0 = K \cdot \left( 1 + \frac{T_e}{T_i} + \frac{N \cdot T_d}{T_d + N \cdot T_e} \right) \\ r_1 = -K \cdot \left( 1 + \frac{T_d}{T_d + N \cdot T_e} + \frac{T_e}{T_i} \cdot \frac{T_d}{T_d + N \cdot T_e} + \frac{N \cdot T_d}{T_d + N \cdot T_e} \right) \\ r_2 = K \cdot \left( \frac{T_d}{T_d + N \cdot T_e} \right) \\ s_1' = \frac{T_d}{T_d + N \cdot T_e} \end{cases} \quad (64)$$

Comme on le voit le correcteur obtenu comporte autant de degrés de liberté que le correcteur PID analogique. On reconnaît au dénominateur le terme  $(1 - q^{-1})$  en facteur qui assure l'effet d'intégration.

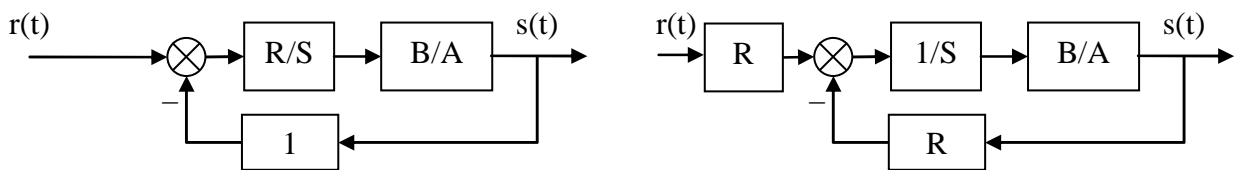


Figure 30 : PID sous forme RST

La fonction de transfert en boucle fermée obtenue est donnée par l'équation suivante :

$$H_{BF} = \frac{B \cdot R}{A \cdot S + B \cdot R} = \frac{B \cdot R}{P} \quad (65)$$

De manière générale le polynôme caractéristique  $P$  fixe les performances visées en boucle fermée. Le calcul de  $R$  et  $S$ ,  $P$  étant fixé et  $A$  et  $B$  étant connus, est réalisé en résolvant l'équation de Bézout  $P = A \cdot S + B \cdot R$ . (voir 9.1 Annexe 1 : Résolution de l'équation polynomiale de Bezout)

On voit que le numérateur  $B \cdot R$  est imposé une fois que  $R$  et  $S$  sont trouvés. Le régulateur introduit donc naturellement des zéros dans la boucle fermée.

Traisons l'exemple entièrement dans le cas où le système est modélisé comme un système du deuxième ordre avec faible retard. La fonction de transfert du procédé est donnée par :



$$H_{BO} = \frac{B}{A} = \frac{b_1 \cdot q^{-1} + b_2 \cdot q^{-2}}{1 + a_1 \cdot q^{-1} + a_2 \cdot q^{-2}} \quad (66)$$

Le correcteur PID rajoutant deux pôles, le polynôme caractéristique en boucle fermée spécifié doit posséder quatre pôles. Le polynôme caractéristique en boucle fermée doit donc être de la forme :

$$\begin{aligned} P &= 1 + p_1 \cdot q^{-1} + p_2 \cdot q^{-2} + p_2 \cdot q^{-3} + p_2 \cdot q^{-4} \\ P &= (1 + p_1' \cdot q^{-1} + p_2' \cdot q^{-2}) \cdot (1 + \alpha_1 \cdot q^{-1}) \cdot (1 + \alpha_2 \cdot q^{-1}) \end{aligned} \quad (67)$$

Et il doit être identifié au produit  $AS + BR$  avec  $R$  et  $S$  ayant les formes données par (64) :

$$\begin{aligned} &(1 + p_1' \cdot q^{-1} + p_2' \cdot q^{-2}) \cdot (1 + \alpha_1 \cdot q^{-1}) \cdot (1 + \alpha_2 \cdot q^{-1}) \\ &= (1 + a_1 \cdot q^{-1} + a_2 \cdot q^{-2}) \cdot (1 - q^{-1}) \cdot (1 + s_1' \cdot q^{-1}) \\ &+ (b_1 \cdot q^{-1} + b_2 \cdot q^{-2}) \cdot (r_0 + r_1 \cdot q^{-1} + r_2 \cdot q^{-2}) \end{aligned} \quad (68)$$

Les coefficients  $p_1'$  et  $p_2'$  définissent un modèle du second ordre aux caractéristiques (pulsation propre  $\omega_0$  et amortissement  $\sigma$  choisis et coefficients déduits conformément au Tableau 3 page 25). Les coefficients  $\alpha_1$  et  $\alpha_2$  correspondent à des pôles auxiliaires aperiodes. Les choix typiques pour ces coefficients sont :

$$-0.05 < \alpha_1, \alpha_2 < -0.5 \quad (69)$$

On montre que la résolution de (68) est réalisée par une inversion matricielle simple :

$$M \cdot x = p \quad (70)$$

avec

$$\begin{cases} x^T = [1 & s_1' & r_0 & r_1 & r_2] \\ p^T = [1 & p_1 & p_2 & p_3 & p_4] \end{cases}$$

et

$$M = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ a_1' & 1 & b_1 & 0 & 0 \\ a_2' & a_1' & b_2 & b_1 & 0 \\ a_3' & a_2' & 0 & b_2 & b_1 \\ 0 & a_3' & 0 & 0 & b_2 \end{pmatrix} \quad (71)$$

On montre que M est inversible à la condition que A et B soient premiers entre eux, c'est-à-dire que A et B n'aient pas de pôles et de zéros en commun.

(Le problème peut aussi être résolu en utilisant la résolution de l'équation de Bézout).

### 7.3.2 Cas 2

Comme on l'a vu le correcteur PI numérique directement transposé du correcteur PID analogique décrit dans le chapitre 7.3.1 a l'inconvénient de ramener des zéros au dénominateur de la fonction de transfert en boucle fermée (équation 65). Il est possible, grâce à la flexibilité du régulateur numérique d'imposer au correcteur de ne pas ajouter de zéros en boucle fermée. La fonction de transfert désirée en boucle fermée sera donc de la forme :

$$H_{BF} = \frac{P(1)}{B(1)} \cdot \frac{B}{P} \quad (72)$$

Le facteur  $P(1)/B(1)$  garantit le gain statique unitaire du système en boucle fermée (voir page 12). Si le régulateur a la forme générale RST alors les polynômes R S et T doivent vérifier :

$$H_{BF} = \frac{B \cdot T}{A \cdot S + B \cdot R} = \frac{P(1) B}{B(1) P} \quad (73)$$

S et R sont obtenus par la résolution de la même équation que précédemment (68). Le polynôme T est obtenu par simple identification :  $T = R(1)$ .

La structure, sous forme de schéma-bloc est donnée ci-dessous :

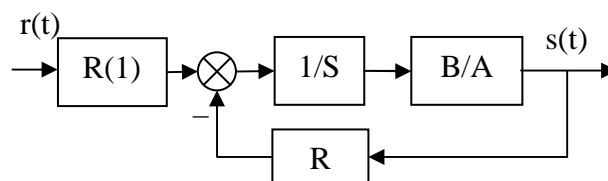


Figure 31 : PID2 sous forme RST

On montre que cette structure de PID, ramené en continu, correspond au PID canonique pour l'action intégrale. L'action proportionnelle et dérivée n'est appliquée que sur le signal  $y$ .

La structure du PID2 par rapport au PID1 a pour avantage de ne pas ramener de zéros à la boucle fermée. Cela se traduit par un dépassement moindre en transitoire lorsqu'on applique un échelon de commande. Par contre les deux systèmes réagissent de la même manière à un échelon de perturbation.

## 7.4 Placement de pôles

Les deux structures de PID décrites dans le chapitre précédent sont deux cas particulier d'une stratégie plus générale de placement de pôles.

La synthèse est plus générale que les PID présentés précédemment puisqu'il n'est pas nécessaire de faire d'hypothèses sur le degré de A et B, sur le retard, ni sur la présence ou non de zéros stables ou instables dans le procédé.

La structure du correcteur est la structure type RST telle que présentée dans la Figure 29. Pour la suite des calculs on posera pour la fonction de transfert les conventions suivantes :

$$H_{BO} = \frac{q^{-d} \cdot B(q^{-1})}{A(q^{-1})} \quad (74)$$

$$\begin{cases} A(q^{-1}) = 1 + a_1 \cdot q^{-1} + \dots + a_{n_A} \cdot q^{-n_A} \\ B(q^{-1}) = b_1 \cdot q^{-1} + \dots + b_{n_B} \cdot q^{-n_B} = q^{-1} \cdot B^*(q^{-1}) \end{cases}$$

Compte tenu de la forme RST du régulateur la fonction de transfert en boucle fermée vérifie :

$$H_{BF} = \frac{q^{-d} \cdot B(q^{-1}) \cdot T(q^{-1})}{P(q^{-1})} \quad (75)$$

$$P(q^{-1}) = A(q^{-1}) \cdot S(q^{-1}) + q^{-d} \cdot B(q^{-1}) \cdot R(q^{-1}) = 1 + p_1 \cdot q^{-1} + \dots + p_{n_P} \cdot q^{-n_P}$$

Le polynôme  $P(q^{-1})$  définit donc les pôles de la fonction de transfert en boucle fermée et traduit donc le comportement en *poursuite*.

De la même manière la fonction de transfert entre une perturbation et la sortie (fonction de sensibilité perturbation-sortie) est donnée par :

$$S_{yp} = \frac{A(q^{-1}) \cdot S(q^{-1})}{P(q^{-1})} \quad (76)$$

Ce même polynôme traduit donc aussi le comportement en *régulation*.

Les consignes en poursuite et en régulation peuvent être différentes comme cela est représenté par exemple sur la Figure 32.

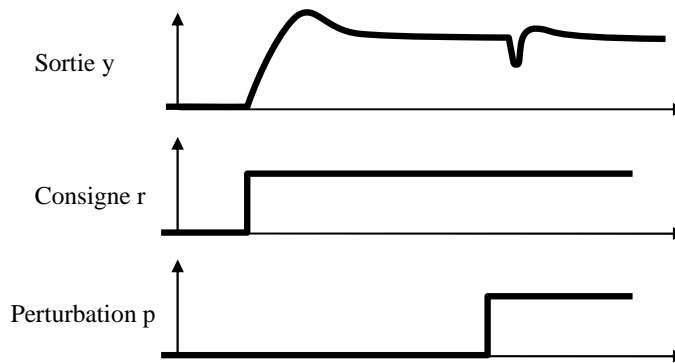


Figure 32 : Performances différentes en régulation et en poursuite

### 7.4.1 Choix des pôles en boucle fermée

De manière générale le comportement en boucle fermée est spécifié sous la forme d'une pulsation propre  $\omega_0$  et d'un amortissement  $\sigma$ , donc une fonction de transfert du second ordre. La spécification étant généralement donnée en continu il est nécessaire de traduire la valeur désirée de  $\omega_0$  et  $\sigma$  en un polynôme  $P_D(q^{-1})$  par discrétisation de la fonction de transfert continue (chapitre 4.5 page 29). Ce polynôme  $P_D$  traduit le comportement dominant du système bouclé.

Si l'on souhaite introduire un effet de filtrage supplémentaire dans certaines bandes de fréquence (réduction des bruits de mesure, adoucissement de la commande, robustesse) cela est décrit par un polynôme  $P_F$  comportant des pôles auxiliaires, plus rapides que les pôles de  $P_D$ .

Le polynôme caractéristique en boucle fermée est donc constitué de pôles dominants et de pôles auxiliaires :

$$P(q^{-1}) = P_D(q^{-1}) \cdot P_F(q^{-1}) \quad (77)$$

### 7.4.2 Régulation : calcul de R et S

Le polynôme  $P(q^{-1})$  étant fixé il s'agit de trouver R et S vérifiant :

$$P(q^{-1}) = A(q^{-1}) \cdot S(q^{-1}) + q^{-d} \cdot B(q^{-1}) \cdot R(q^{-1}) \quad (78)$$

Comme déjà vu au chapitre 7.3.1 on obtient une équation de Bézout qu'il s'agit de résoudre. (Voir aussi 9.1 Annexe 1 : Résolution de l'équation polynomiale de Bezout)

Il est parfois nécessaire de spécifier R et S avec des parties fixes  $H_R$  et  $H_S$ . Dans ce cas l'équation (78) est modifiée et on l'écrit sous la forme:

$$P(q^{-1}) = (A(q^{-1}) \cdot H_S(q^{-1})) \cdot S'(q^{-1}) + (q^{-d} \cdot B(q^{-1}) \cdot H_R(q^{-1})) \cdot R'(q^{-1}) \quad (79)$$

Les inconnues que l'on souhaite trouver sont  $S'$  et  $R'$ . Ceci est nécessaire dans les cas particuliers suivants :

- spécification d'erreur statique nulle,
- rejet d'une perturbation harmonique,
- blocage d'un signal,

Détaillons maintenant ces cas particuliers.

### Spécification d'une erreur statique nulle

On retrouve la nécessité d'avoir un terme intégrateur dans la boucle directe. La spécification se retrouve donc sur  $S$  :

$$S(q^{-1}) = (1 - q^{-1}) \cdot S'(q^{-1}) = H_S(q^{-1}) \cdot S'(q^{-1}) \quad (80)$$

### Rejet d'une perturbation harmonique

Si le système est perturbé par une perturbation harmonique à une fréquence fixe (par exemple 50Hz...) il est possible d'imposer à  $S$  d'avoir un module *nul* ou amorti à cette fréquence. Cela se traduit par la présence dans  $S$  d'une paire de zéros complexe à amortissement nul ou faible à cette fréquence :

$$S(q^{-1}) = (1 + \alpha \cdot q^{-1} + \beta \cdot q^{-2}) \cdot S'(q^{-1}) = H_S(q^{-1}) \cdot S'(q^{-1}) \quad (81)$$

### Blocage d'un signal

Pour certaines applications on souhaite que le régulateur "laisse passer" certaines composantes du signal de sortie, c'est-à-dire qu'il ne réagisse pas à ces composantes. Cela se traduit par  $R=0$  à ces fréquences. On retrouve la même forme que précédemment :

$$R(q^{-1}) = (1 + \alpha \cdot q^{-1} + \beta \cdot q^{-2}) \cdot R'(q^{-1}) = H_R(q^{-1}) \cdot R'(q^{-1}) \quad (82)$$

### 7.4.3 Poursuite : calcul de T

Le schéma-bloc de l'asservissement est donné sur la Figure 33. Le terme  $T$  est décomposé en un terme  $T'$  et un terme  $B^*/A^*$ .

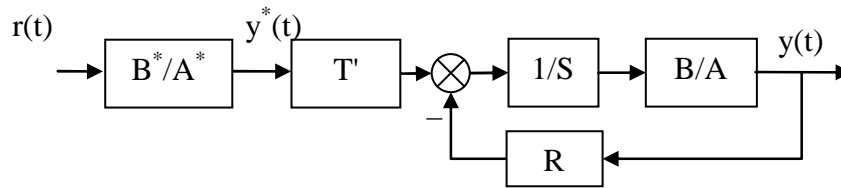


Figure 33 : placement de pôle

L'idée est de régler  $B^*$  et  $A^*$  de telle manière que le signal  $y^*$  ait la dynamique de poursuite désirée (par exemple un second ordre avec pulsation propre et amortissement donné) puis de régler  $T'$  de telle manière que le transfert entre  $y^*$  et  $y$  soit le plus "transparent" possible.

### Réglage de $T'$

Le transfert entre  $y^*$  et  $y$  est :

$$S_{y^*y}(q^{-1}) = T'(q^{-1}) \frac{q^{-d} \cdot B(q^{-1})}{P(q^{-1})} \quad (83)$$

Pour assurer le transfert le plus "transparent" possible il faut donc choisir

$$T'(q^{-1}) = \frac{P(q^{-1})}{B(1)} \quad (84)$$

Remarques :

- Si  $B(1)=0$  alors on remplace  $B(1)$  par 1 dans (84)
- De même dans (84)  $P(q^{-1})$  peut être remplacé par  $P_D$  correspondant à la dynamique dominante.

### Réglage de $B^*$ et $A^*$

$B^*$  et  $A^*$  sont choisis de telle manière que la fonction de transfert  $B^*/A^*$  corresponde à la dynamique de poursuite souhaitée.

## 7.5 Poursuite et régulation à objectifs indépendants

Cette méthode ressemble à la méthode de placement de pôles à ceci près que l'on va effectuer une simplification des zéros de  $B^*(q^{-1})$  : cela ne s'applique donc qu'aux systèmes avec des pôles stables. Plus précisément il faut qu'ils aient un amortissement suffisant ( $> 0,2$ ) de manière à ne pas être trop près du cercle unité.

La structure de l'asservissement est toujours celle donnée sur la Figure 33.

**Choix de P**

$P(q^{-1})$  est choisi de manière à simplifier  $B^*(q^{-1})$  (rappel : la forme de B est donnée dans l'équation 74). Comme indiqué dans le chapitre 7.4.2 page 60 il s'agit donc d'imposer les zéros de  $B(q^{-1})$  dans  $H_R(q^{-1})$  et  $H_S(q^{-1})$ .

**Choix de T**

Si l'on choisit  $T(q^{-1}) = P(q^{-1})$  alors le transfert entre  $y^*$  et  $y$  est :

$$S_{yy^*}(q^{-1}) = q^{-(d+1)} \quad (85)$$

**Réglage de  $B^*$  et  $A^*$** 

$B^*$  et  $A^*$  sont choisis de telle manière que la fonction de transfert  $B^*/A^*$  corresponde à la dynamique de poursuite souhaitée.





## 8. TECHNOLOGIE DE LA COMMANDE PAR ORDINATEUR

Du point de vue de l'automaticien le ordinateur est relié au système physique à piloter par un certain nombre d'intermédiaires plus ou moins gênants :

- les interfaces analogiques et numériques,
- le transport de l'information par bus et protocole,
- l'utilisation éventuelle de capteurs ou d'actionneurs « intelligents ».

Il ne s'agit pas ici de faire un exposé complet sur ces trois aspects mais de poser les problèmes qui se posent à l'automaticien lorsque le système qu'il doit commander est perturbé par ces éléments.

Enfin ce chapitre se termine sur les notions de « Hardware In the Loop » et de « prototypage rapide » qui sont partie intégrante des processus de réalisation de systèmes de contrôle.

### 8.1 Calculateurs et interfaces numériques / analogiques

Le premier problème à prendre en compte est la quantification de l'information. Du point de vue du ordinateur d'une part et du point de vue des convertisseurs d'autre part. Dans un premier temps, considérons les aspects performance de calcul lié à la nature du codage des nombres dans le ordinateur.

#### Les performances liées au types de variables

Prenons l'exemple où le concepteur de la loi de commande souhaite réaliser un correcteur de type proportionnel. Entre l'acquisition des mesures et l'écriture de la consigne il y aura la réalisation d'une de commande du type :

$$y = K \cdot (\text{consigne} - \text{mesure}) \quad (86)$$

En utilisant un langage de programmation évolué (par exemple le C) la ligne de code sera la transcription littérale de (86). Le programmeur devra décider de la définition et du type des variables, principalement du nombre d'octets utilisés pour stocker ses variables. Le Tableau 5 résume par exemple les types utilisés en langage C (norme ANSI)

Tableau 5 : Types ANSI C

Types entiers	char	8 bits
	short int	16 bits
	long int	32 bits
Types flottants (mantisse et exposant)	float	32 bits
	double	64 bits
	longdouble	128 bits

Une fois le code écrit en C l'exécution du programme sur un processeur cible ne peut se faire que par l'intermédiaire d'une compilation du programme. A chaque processeur est donc adapté une version de compilateur.

Les performances du programme (sa rapidité d'exécution) dépendront donc en partie de la qualité du compilateur. (Par exemple, une multiplication s'effectue t'elle bêtement par un décalage de tous les bits ou bien des solutions astucieuses sont-elles implémentées pour arrêter le décalage selon le contexte ?)

Le principal déterminant de la performance est cependant le processeur : son type (8 bits, 16 bits ou 64 bits) et son cadencement (en MHz ou GHz) sont les deux plus importants paramètres. Coder les variables sur 64 bits (double) dans le programme coûtera cher en temps de calcul si le processeur est un 8 bits. Le type de mémoire et leurs temps d'accès peuvent aussi être déterminants. Enfin, certains processeurs disposent d'un processeur spécifiquement dédié aux opérations mathématiques (coprocesseur mathématique).

Il n'est pas possible de dire exactement, pour tel type de compilateur associé à tel type de processeur, le temps que prendra par exemple une multiplication. En pratique on effectue une série de tests types (benchmark) et on affiche un temps moyen d'exécution.

En résumé, on peut être tenté, pour ne pas s'embêter, de coder toutes les variables en virgule flottante. Mais il faut bien savoir que cela a un coût en terme de temps de calcul, surtout si le processeur n'est pas adapté. Pour fixer les idées on peut dire qu'il y a un rapport 20 entre une addition réalisée en "short" et une addition réalisée en virgule flottante.

Enfin, l'utilisation de fonctions mathématiques (sinus, exponentielles, etc...) peut aussi avoir un coût très lourd si le travail n'est pas réalisé par un coprocesseur mathématique dédié. Le compilateur et sa librairie mathématique joue là encore un grand rôle.

### La quantification imposée par le processeur

Au-delà de l'aspect performance et temps de calcul, la quantification de l'information doit parfois aussi être prise en compte. Rappelons qu'un entier codé sur N bits peut prendre seulement  $2^N - 1$  valeurs différentes. Avant d'effectuer les calculs mathématiques dans le corps du programme il est donc judicieux d'appliquer un

facteur d'échelle de manière à utiliser l'amplitude maximale possible (entre 0 et  $2^N - 1$ ).

### **La quantification imposée par les conversions analogique et numérique**

Les convertisseurs analogique-numérique et numérique-analogique sont principalement caractérisés par leur amplitude de signal analogique d'entrée (généralement en volts, par exemple +/- 10V) et par leur résolution en bits (8 ou 10 bits pour les plus courants, jusqu'à 16 bits pour les convertisseurs haut de gamme).

Prenons l'exemple d'un capteur gyromètre transformant une vitesse de rotation en tension. Compte tenu de l'application le signal utile varie entre 2,4V et 2,6V. Si le signal mesuré est branché directement sur un convertisseur 0-10V 10bits alors un problème de conditionnement du signal se pose. En effet la quantification est de  $\frac{10V}{2^{10}-1} = 0,098V$ . Les 0,2V de signal utile seront donc discrétisés en  $\frac{0,2}{0,098} = 20$  intervalles seulement !

Si la discrétisation du signal utile est trop grossière cela peut se traduire par une diminution des performances de l'asservissement, voire une instabilité. Le système n'est en effet plus linéaire et il n'existe pas d'outils théoriques permettant de prévoir les performances du système bouclé.

## **8.2 Transport de l'information : les bus et les protocoles**

Pas question de faire ni même d'ébaucher un cours sur les bus et les protocoles ici. Mais on va quand même donner quelques indications. L'automaticien a l'habitude de simplifier le problème qu'il se pose en dessinant un schéma simplifié de son problème d'asservissement (du type de la Figure 20 page 45) pour pouvoir appliquer les théories de la commande. Cependant il arrive bien souvent que l'information transite à l'intérieur du système, soit entre différents calculateurs effectuant différentes parties du travail soit entre ordinateur et périphériques, par l'intermédiaire de bus informatique. A chaque type de bus correspond un ou plusieurs protocoles qui peuvent poser des problèmes pour l'automaticien :

- les données sont-elles discrétisées ? (le protocole peut par exemple ne permettre de transmettre que des paquets de données de 16 bits),
- le bus induit-t'il un retard ? (le composant mettant en forme les données du ordinateur pour les publier sur le bus a certaines performances qui peuvent induire un délai)
- le bus est-il déterministe ? (Certains bus ne garantissent pas que les données arriveront en un temps donné),
- le bus est-il sûr ? (Certains bus ne garantissent pas non plus que les données arriveront bien)

### **8.3 Capteurs et actionneurs "intelligents", "smart sensors"**

De plus en plus souvent les capteurs et les actionneurs sont associés à leur électronique de puissance, électronique de commande et de conditionnement de signal. L'interface avec le capteur ou l'actionneur se fait donc via un protocole de communication (par exemple via un bus RS232).

L'intérêt est de véhiculer non pas des signaux analogiques, tensions ou courants, mais des signaux numériques beaucoup moins sensibles aux éventuelles perturbations.

On retrouve donc, entre le capteur ou l'actionneur et le ordinateur, la présence d'un bus et d'un protocole de communication.

### **8.4 La philosophie "hardware in the loop"**

Les systèmes embarqués sont de plus en plus complexes et touchent des domaines de plus en plus variés : automobile, satellites, aéronautique, UAV, systèmes d'arme, sous-marins, systèmes, etc... Les systèmes à piloter ainsi que les systèmes de commande sont de plus en plus complexes.

Les techniques de simulation « Hardware In the Loop » (HIL) sont de plus en plus utilisées dans le développement et dans le test de ces systèmes embarqués. L'idée est de développer entièrement le système de contrôle sur une plateforme de test en utilisant le maximum de composants réels et en ne remplaçant que le minimum par la simulation. Par exemple en aéronautique l'« Iron Bird » regroupe dans un hangar tous les systèmes réels de l'avion en cours de développement. Seule la mécanique du vol est simulée : les sorties des actionneurs sont mesurées, rentrées dans un modèle de simulation numérique, puis les mesures effectuées normalement par les capteurs sont calculées et restituées par le modèle.

L'intérêt est de diminuer de manière importante les coûts et les délais en économisant le plus grand nombre possible de tests sur le système réel en situation. Lors des premiers tests réels les techniques « Hardware In the Loop » permettent d'avoir effectué la plus grande part du développement des lois de commande.

### **8.5 Le prototypage rapide en automatique**

Un peu dans le même ordre d'idée que le « Hardware In the Loop » le prototypage rapide permet de réaliser la conception des lois de commande en simulation puis de transporter rapidement (en un simple click !) le code du contrôleur simulé vers la cible temps réel implémentée sur le système réel.

La société Mathworks [9] investit massivement dans ce domaine en ayant enrichi son logiciel de calcul scientifique (Matlab) avec un logiciel de simulation graphique très évolué (Simulink) puis en développant des outils de prototypage rapide associés (Real Time Workshop, Embedded Target For...).

## 9. ANNEXE

### 9.1 Annexe 1 : Résolution de l'équation polynomiale de Bezout

Il s'agit de trouver R et S vérifiant :

$$P(q^{-1}) = A(q^{-1}).S(q^{-1}) + B(q^{-1}).R(q^{-1})$$

Où P, A et B sont connus.

Dans le cas où P, A, B, R et S sont des nombres entiers la solution de l'algorithme est donnée par l'algorithme de division Euclidienne étendu [8].

Dans le cas où P, A, B, R et S sont des polynômes, l'algorithme est donné ci-dessous, programmé en matlab ([1]).

```
function [Rp,Sp,nrp,nsp]=bezoutd(A,B,Hs,Hr,P)
%function [Rp,Sp,nrp,nsp]=bezoutd(A,B,Hs,Hr,P)
%solves AHsSp+BHrRp=P by coefficient comparison. Delay and discretization delay need to be integrated in B
%(zeros at the beginning).
%
%inputs:
%A=[a0 a1 ... aNa] ... vector of model denominator coefficients A=a0 + a1z^(-1) + a2z^(-2) +...+ aNaz^(-Na)
%B=[b0 b1 ... bNb] ... vector of model numerator coefficients B=b0 + b1z^(-1) + b2z^(-2) +...+ bNbz^(-Nb)
%Hs=[hs0 hs1 ... hsNhs] ... vector of controller denominator fixed part Hs=hs0 + hs1z^(-1) +...+ hsNhsz^(-Nhs)
%Hr=[hr0 hr1 ... hrNhr] ... vector of controller denominator fixed part Hr=hr0 + hr1z^(-1) +...+ hrNhrz^(-Nhr)
%P=[p0 p1 ... pNp] ... vector of desired polynomial coefficients P=p0 + p1z^(-1) + p2z^(-2) +...+ pNpz^(-Np)
%outputs:
%Rp=[rp0 rp1 rp2 ...] ... vector of coefficients for resulted controller numerator
%Sp=[sp0 sp1 sp2 ...] ... vector of coefficients for resulted controller denominator
%nrp ... order of Rp
%nsp ... order of Sp
%
%written by: J. Langer, I.D. Landau, H. Prochazka
%7th june 2002

PRECISION=1e-16;

D=size(A);
if D(1)>1, A=A'; end;
```

```

D=size(B);
if D(1)>1, B=B'; end;
D=size(Hs);
if D(1)>1, Hs=Hs'; end;
if D(1)==0, Hs=1; end;
D=size(Hr);
if D(1)>1, Hr=Hr'; end;
if D(1)==0, Hr=1;end;
D=size(P);
if D(1)>1, P=P'; end;

na=length(A)-1;
nb=length(B)-1;
np=length(P)-1;
nhs=length(Hs)-1;
nhr=length(Hr)-1;

if (nhs>0), Ah=real(convz(A,Hs)); else Ah=A*Hs; end; % Ah = A * Hs
nah=length(Ah) -1;
if (nhr>0), Bh=real(convz(B,Hr)); else Bh=B*Hr; end;
nbh=length(Bh) -1;
if (np>nah+nbh-1), disp('Bezout error: too many poles');end;

% increase size of P using zeros if necessary
if (np<nah+nbh-1),
%
% set remaining poles onto a circle with radius rmin
P=[P zeros(1,nah+nbh-1-np)];
rootsPdes=roots(P);
nextra=nah+nbh-1-np;
rmin=1e-16;
angle=[0:nextra-1]'/nextra*2*pi;
j=sqrt(-1);
rootsPextra=rmin*exp(j*angle);
P=poly([rootsPdes;rootsPextra]);
np=nah+nbh-1;

end;
P,

nsp=nbh-1;
nrp=nah-1;

%matrix is smaller than vector PD

```

```

if (np>nah+nbh-1),
    disp('Order of model denominator is too low! Add a polynom of higher order to Hs or Hr. ');
end;

% ns=nsp+nhs
% nr=nrp+nhr

M=[];
for j=1:nsp+1,
    V=[];
    if (j>1), V=[V ; zeros(j-1,1)]; end;% zeros in front of Ah
    V=[V ; Ah'];% Ah
    if (j<=nsp), V=[V ; zeros(nsp+1-j,1)]; end;% zeros behind Ah
    if (length(V)~=nah+nbh), disp('bezoutb: error V'); end;
    M=[M V]; % add one column to M
end;

for j=1:nrp+1,
    V=[];
    if (j>1), V=[V ; zeros(j-1,1)]; end;
    V=[V ; Bh'];
    if (j<=nrp), V=[V ; zeros(nrp+1-j,1)]; end;
    if (length(V)~=nah+nbh), disp('bezoutb: error V'); end;
    M=[M V];
end;

D=size(M);
if (D(1)~=nah+nbh), disp('bezoutb: error size M row'); end;
if (D(2)~=nah+nbh), disp('bezoutb: error size M column'); end;

% make P column vector
P=P';

global M1;
M1=M;

X= M\P;
% coefficients are real values
X=real(X);

% make X row vector
X=X';

```

## Commande des systèmes par ordinateur

```
Sp=X(1:nsp+1);  
Rp=X(nsp+2:nsp+nrp+2);
```



## 10. BIBLIOGRAPHIE

- [1] I. D. Landau, "*Commande des systèmes*", Edition Lavoisier 2002.
- [2] C. Nouals, "Théorie du Signal", Notes de cours ENSICA
- [3] F. Vincent, "*Traitement Numérique du Signal*", Notes de cours ENSICA.
- [4] D. Arzelier, "*Représentation et analyse des systèmes, notes de cours*", Edition ENSICA. Voir aussi <http://www.laas.fr/~arzelier/>
- [5] J. Bordeneuve-Guibé, "*Commande des systèmes linéaires*", Notes de cours ENSICA.
- [6] Y. Briere, "*Identification*", Notes de cours ENSICA.
- [7] L. Ljung, "*System Identification, Theory for the User*", Prentice Hall 1987.
- [8] Wikipedia , équation de Bezout : [http://en.wikipedia.org/wiki/Bézout's\\_identity](http://en.wikipedia.org/wiki/Bézout's_identity)
- [9] <http://www.mathworks.com>



# 11. INDEX

<b>A</b>	
actionneur intelligent .....	21
amortissement .....	15
ANSI C .....	60
Approximation numérique d'équation différentielle .....	28
ARMAX .....	35
<b>B</b>	
BJ35 .....	
Blocage d'un signal .....	56
bloqueurs d'ordre 0 .....	22
BOZ .....	22
bus.....	21, 61
<b>C</b>	
CAN.....	21
capteur intelligent .....	21
causal .....	6, 9
Causalité .....	10
CNA.....	21
compilateur .....	60
convertisseur analogique / numérique.....	21
convertisseur numérique / analogique.....	21
convertisseurs analogique-numérique .....	61
coprocesseur mathématique .....	60
Correspondances pôles - zéros.....	31
<b>D</b>	
Diagramme de Bode .....	18
Discrétisation de fonction de transfert continues .....	27
division polynomiale.....	6
<b>E</b>	
échelon unitaire.....	11
équation d'état .....	17
Erreur statique.....	42
Estimation de la réponse fréquentielle .....	39
Estimation de la réponse impulsionnelle.....	38
Euler .....	29
<b>F</b>	
facteur d'échelle .....	61
facteur d'oubli .....	36
<i>fenêtre de Hamming</i> .....	39
filtre anti-repliement .....	25
fonction de sensibilité .....	42
fonction de transfert discrète.....	9
<b>G</b>	
gain statique.....	11
gain statique unitaire.....	42
<b>I</b>	
Identification.....	32
Identification non paramétrique .....	38
intercorrélation.....	38
<b>L</b>	
lieu des racines.....	43
lieu d'Evans.....	43
LTI.....	8
<b>M</b>	
marge de gain.....	47
marge de module.....	46
marge de phase .....	47
méthode des moindres carrés récursifs .....	32
Méthode des trapèzes.....	29
modèle ARX .....	32
<b>O</b>	
OE.....	35
one step ahead prediction.....	35
opérateur de retard .....	9
<b>P</b>	
période d'échantillonnage (choix) .....	26
Placement de pôles .....	54
Pôles .....	11
Poursuite .....	56
<i>prédiction</i> .....	32
protocoles.....	61
Pseudo Random Binary Signal .....	34
pulsation .....	15
<b>Q</b>	
quantification .....	23, 24, 60
<b>R</b>	
R,S,T.....	41
rapidité.....	15
rayon de convergence .....	5, 9
Régulation.....	55

Rejet de perturbations .....	42
Rejet d'une perturbation harmonique .....	56
réponse à un échelon.....	11
réponse fréquentielle.....	18
Réponse impulsionnelle.....	10
Réponses indicielle .....	10
retard.....	23
root locus .....	43

**S**

signal discret .....	5
smart sensors.....	62
stabilité.....	11
Systèmes du premier ordre.....	14
Systèmes du second ordre.....	15

**T**

Théorème de Nyquist.....	46
théorème de Shannon.....	19, 24
théorème du retard .....	6
Transformée en z .....	5
transformée unilatérale .....	5
Tustin.....	29

**V**

virgule flottante.....	60
------------------------	----

**Z**

zero holder hold .....	23
zéros.....	11
zoh .....	23