

# Modélisation des signaux échantillonnés(cours 1)

## 1.1. Définitions

**1.1.1. Signal:** Il s'agit d'une grandeur physique générée par un appareil ou appliquée à un dispositif. Exemples : température, pression, courant, etc.

**1.1.2. Signal continu :** Lorsque le temps est continu et l'amplitude est continue, le signal est continu (analogique), tel que le signal brut délivré par un capteur physique. Un signal continu est caractérisé par une infinité d'amplitude.

**1.1.3. Signal discret :** Dans un système d'acquisition de données, les mesures sont réalisées à intervalles de temps réguliers. Le temps n'est plus traité comme une variable continue, mais est discrétisé (temps discret). En fait, un signal discret est obtenu par discrétisation d'un signal continu en utilisant un pas de discrétisation variable. Contrairement au signal continu, un signal discret est caractérisé par un nombre fini d'amplitude.

**1.1.4. Signal échantillonné :** Il s'agit d'un signal continu discrétisé par un pas de temps régulier. Ce pas est appelé période d'échantillonnage.

**1.1.5. Signal numérique :** Il s'agit d'un signal échantillonné quantifié en amplitude.

**1.1.6. Quantification :** L'opération de quantification consiste à attribuer un nombre binaire à toute valeur (amplitude) prélevée au signal lors de l'échantillonnage.

**1.1.7. Signal causal :** Un signal est dit causal s'il est nul pour toute valeur négative du temps. On note que nous ne considérerons dans la suite du polycopié que les signaux causals.

## 1.2. La transformation de Laplace

### 1.2.1. Définition de la transformation de Laplace

La transformation de Laplace est un moyen mathématique élégant de résoudre les équations différentielles linéaires ou linéarisées. En automatique, elle permet un développement simple des modèles entrées-sorties continus et une analyse qualitative directe de l'influence de variables externes sur un système (procédé).

On associe à la fonction  $f(t)$  une autre fonction  $F(p)$  de la variable complexe  $p$  appelée transformée de Laplace ainsi définie par :

$$F(p) = \mathcal{L}[f(t)] = \int_0^{+\infty} f(t)e^{-pt} dt \quad (1.1)$$

en supposant que le signal  $f(t)$  est nul pour  $t < 0$  (signal causal). On parle donc de transformée de Laplace monolatère (utilisée dans ce polycopié). Le symbole  $\mathcal{L}$  se lit 'l'opérateur de la transformée de Laplace'.

*Exemple 1.1 :* Calculons la transformée de Laplace de  $f(t) = e^{-at}$ , en utilisant l'équation (1.1), on obtient :

$$F(p) = \mathcal{L}[f(t)] = \mathcal{L}[e^{-at}] = \int_0^{+\infty} f(t)e^{-pt} dt = \int_0^{+\infty} e^{-at} e^{-pt} dt = \frac{-1}{p+a} [e^{-(a+p)t}]_0^{+\infty} = \frac{1}{p+a}$$

## 1.3. Echantillonnage d'un signal continu

### 1.3.1. Définition de l'échantillonnage

C'est une opération de conversion d'un signal continu  $f(t)$  en une série d'impulsions, dont les amplitudes sont déterminées par les valeurs du signal continu aux instants d'échantillonnage. L'échantillonnage produit donc, à partir d'un signal continu  $f(t)$ , la suite d'échantillons  $\{(kT)\}$ :

$$\{f(kT)\} = \{f(0), f(T), f(2T), f(3T), \dots, f(kT)\}$$

Que l'on note, en général :

$$f^*(t) = \{f(kT)\} = \{f_0, f_1, f_2, f_3, \dots, f_k\}$$

Ou encore :

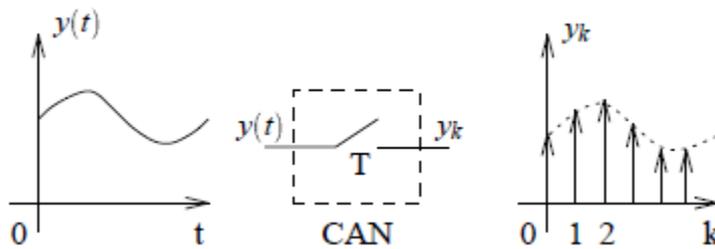
$$(k) = \{f(kT)\} = \{f_0, f_1, f_2, f_3, \dots, f_k\}$$

On définit :

$k$  : variable entière positive,  $k \in \mathbb{N}$  ;  $T$  : est la période d'échantillonnage (Par définition  $T > 0$ );  $kT$  : sont les instants d'échantillonnage;  $f(kT)$  et  $f_k$  : sont les amplitudes du signal continu  $f(t)$  aux instants d'échantillonnage  $kT$  ;  $f^*(t)$  ou  $f(k)$  : est le signal échantillonné du signal continu  $f(t)$ .

### 1.3.2. Principe et modélisation de l'échantillonnage

L'idée consiste à utiliser un interrupteur idéal que l'on ferme pendant une durée très courte, puis que l'on ouvre pendant une durée.. (Période d'échantillonnage). La figure suivante montre la modélisation de l'opération de l'échantillonnage par un interrupteur idéal.



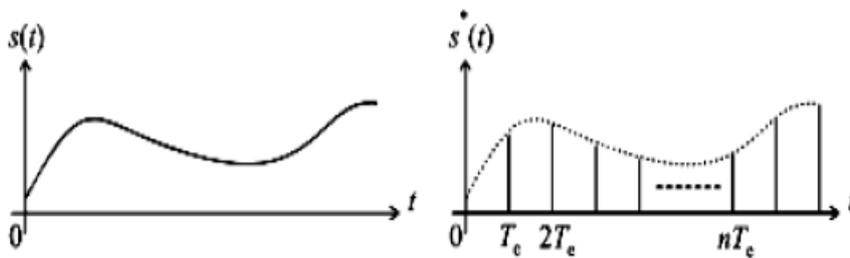
**Figure (1.3) :** Modélisation de l'échantillonnage par un interrupteur idéal : Échantillonnage d'un signal continu  $y(t)$

### 1.3.3. Opération de l'échantillonnage

L'opération de l'échantillonnage se traduit mathématiquement par la multiplication du signal continu  $s(t)$  par le peigne de Dirac **Figure (1.4)**. Pour  $t > 0$ , on a :

$$s^*(t) = \sum s(kT) \cdot \delta(t - kT) \quad (1.8)$$

Cette opération peut être schématisée comme suit :



**Figure (1.3) :** Schéma décrivant l'opération de l'échantillonnage d'un signal continu par un peigne de Dirac.

## 1.3 EXEMPLES DE SIGNAUX ÉCHANTILLONNÉS SIMPLES

### 1.3.1 Impulsion unité

On définit l'impulsion unité échantillonnée par le signal :

$$\delta^*(t) = \{1, 0, 0, \dots, 0\}$$

$$\begin{cases} \delta^*(nT_e) = 1 & \text{pour } n = 0 \\ \delta^*(nT_e) = 0 & \text{pour } n \neq 0 \text{ et } n \rightarrow \infty \end{cases}$$

Autrement dit :

La Figure 1.4 propose une représentation schématique de cette impulsion unité.

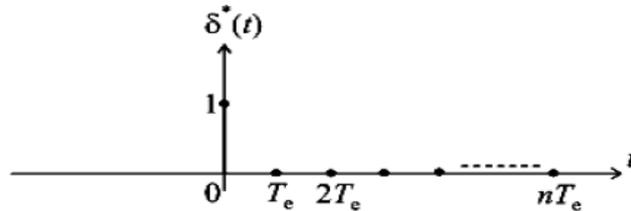


Figure 1.4

### 1.3.2 Echelon unité

On définit l'échelon unité échantillonné par le signal :

$$u^*(t) = \{1, 1, 1, \dots, 1\}$$

$$\begin{cases} u(k) = 1 & \forall k \geq 0 \\ u(k) = 0 & \forall k < 0 \end{cases}$$

Autrement dit :

La Figure 1.5 propose une représentation schématique de cet échelon unité.

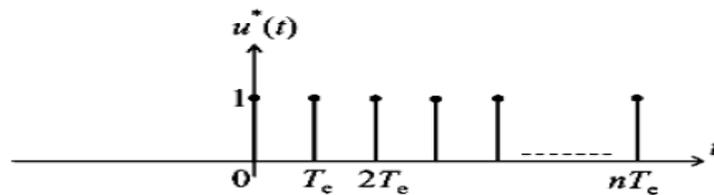


Figure 1.5

Cet échelon unité n'est rien d'autre que la somme d'impulsions unités décalées dans le temps :

Soit :

$$u^*(t) = \delta^*(t) + \delta^*(t - T_e) + \delta^*(t - 2T_e) + \dots$$

$$u^*(t) = \sum_{k=0}^{+\infty} \delta^*(t - kT_e)$$

### 1.3.2 Signal sinusoïdal

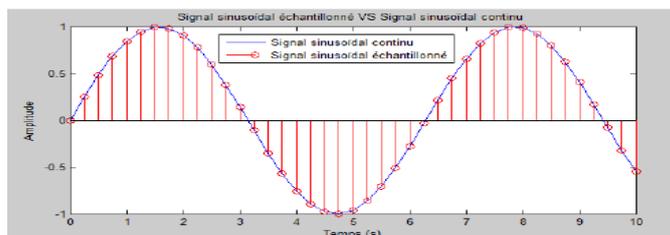
Soit le signal sinusoïdal continu (t) suivant :

$$s(t) = \sin(t), \forall t \geq 0$$

Sa version échantillonnée est obtenue comme suit Figure 1.6

:

$$f^*(t) = \sum f(kT) \cdot \delta(t - kT) = \sum \sin(kT) \cdot \delta(t - kT) \Leftrightarrow f(k) = \{\sin(kT), \forall k \geq 0, 0, \forall k < 0\}$$



**Figure 1.6**

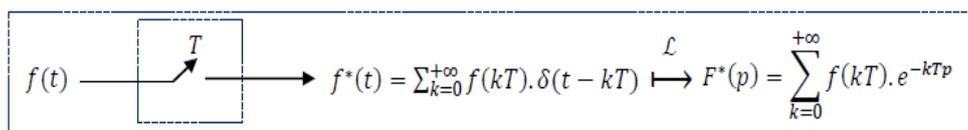
### 1.4.5. Transformée de Laplace d'un signal échantillonné

Soit  $f^*(t)$  le signal échantillonné de  $f(t)$ . Par définition (sans démonstration), la transformée de Laplace de  $f^*(t)$  est donnée comme suit :

$$F^*(p) = \mathcal{L}[f^*(t)] = \sum_{k=0}^{+\infty} f(kT) \cdot e^{-kTp}$$

avec  $F^*(p)$  est la transformée de Laplace de  $f^*(t)$ .

La figure suivante décrit un diagramme montrant le passage de  $(t)$  à  $F^*(p)$  :



### 1.4.6. Choix de la période d'échantillonnage

Parmi les points les plus importants dans la mise en place d'une commande numérique est le choix de la période d'échantillonnage  $T$ . Pour trouver la période d'échantillonnage convenable  $T$ , et par conséquent pour avoir un bon échantillonnage, on utilise le théorème de Shannon. Ce théorème de Shannon fournit les règles qui permettent de garantir un minimum de pertes d'informations dues à l'échantillonnage.

**Théorème de Shannon** : Soit  $f(t)$  un signal périodique de période  $T_f = \frac{2\pi}{\omega_f}$ .

Pour que l'on puisse reconstituer ce signal à partir de son signal échantillonné  $f^*(t)$  (c'est-à-dire pour qu'il n'y ait pas de perte d'information), il faut que la période d'échantillonnage vérifie

$$T < \frac{T_f}{2}.$$

**En fait, en pratique :**

- la période d'échantillonnage doit respecter le théorème de Shannon.
- une période d'échantillonnage trop petite aura pour inconvénient de réduire l'efficacité de la rétroaction face aux perturbations.
- une période d'échantillonnage trop élevée surcharge inutilement la mémoire de l'ordinateur.

Plusieurs auteurs proposent une formalisation de ce choix, le plus souvent quasi-empirique mais donnant de bons résultats du point de vue du compromis précision - vitesse de calcul nécessaire :

Sevely [3]

Yves choisit  $T_e$ , la période d'échantillonnage telle qu'elle soit

$$\frac{2\pi}{18\omega_0} \leq T_e \leq \frac{2\pi}{9\omega_0}$$

$$\frac{\tau}{9} \leq T_e \leq \frac{\tau}{4.5}$$

Bühler [4]

Hansruedi choisit  $T_e$ , la période d'échantillonnage telle qu'elle soit

- 5 fois plus petite que la constante de temps la plus rapide que l'on veut contrôler en boucle fermée

$$T_e \simeq \frac{\tau}{5}$$

- 8 fois plus petite que la pseudo-période, s'il s'agit de pôles complexes conjugués

$$T_e \simeq \frac{1}{8} \frac{2\pi}{\omega_0 \sqrt{1 - \xi^2}}$$

Critère fréquentiel [9]

On choisit  $F_e$ , la fréquence d'échantillonnage telle qu'elle soit 6 à 24 fois plus grande que la fréquence de coupure du système. Soit pour un système d'ordre 1 :

$$\tau \geq T_e \geq \frac{\tau}{4}$$

Pour un système d'ordre 2 :  $\frac{k}{p^2 + 2\xi\omega_0 p + \omega_0^2}$   
soit :

$$0.25 < \omega_0 T < 1\xi = 0.7$$

$$0.4 < \omega_0 T < 1.75\xi = 1$$

Le tableau suivant donne quelques indications valables pour un grand nombre de procédés sur la période d'échantillonnage recommandée :

<i>Signal à échantillonner</i>	<i>Période d'échantillonnage T recommandée</i>
<i>Courant dans les entrainements électriques</i>	$50 < T < 100 (\mu s)$
<i>Position en robotique</i>	$0.2 < T < 1 (ms)$
<i>Position en machine-outil</i>	$0.5 < T < 10 (ms)$
<i>Débit</i>	$1 < T < 3 (s)$
<i>Niveau</i>	$5 < T < 10 (s)$
<i>Pression</i>	$1 < T < 5 (s)$
<i>Température</i>	$10 < T < 45 (s)$

Tableau (1.1) : Période d'échantillonnage des procédés

## 1.5. Reconstitution d'un signal continu

L'opération inverse de l'échantillonnage, c'est-à-dire la transformation d'une suite d'échantillons en un signal continu acceptable technologiquement par les actionneurs, est appelée reconstitution, restitution ou encore extrapolation. Cette étape est indispensable en commande numérique ; en effet, à partir des nombres générés par l'ordinateur (généralement un calculateur

numérique), une grandeur de commande analogique doit être construite afin d'activer le système à commander. En fait, cette opération est réalisée à l'aide de filtres ou de bloqueurs.

### 1.5.1. Bloqueur d'ordre zéro (BOZ)

Le développement important de la théorie des systèmes échantillonnés est dû principalement au développement de la commande par ordinateur numérique. Dans ce cas, souvent, le signal fourni par le ordinateur numérique est un signal en escalier, c'est-à-dire variant par paliers. Un bloqueur d'ordre zéro (BOZ) est caractérisé par le fait que sa sortie entre les instants d'échantillonnage est constante et égale à  $u(k)$ . Le BOZ permet de reproduire exactement un signal constant échantillonné. Le bloqueur BOZ étant représenté sur la figure suivante :

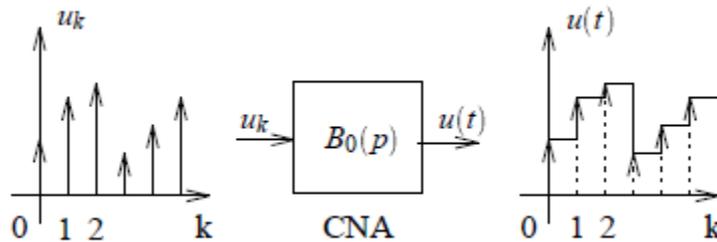
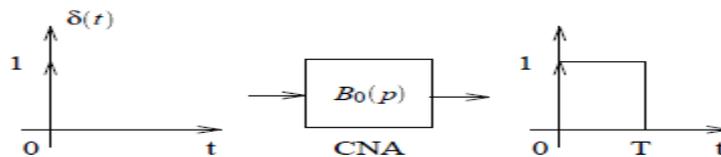


Figure (1.6) : Schéma fonctionnel d'un bloqueur d'ordre zéro (BOZ)

Sa réponse impulsionnelle est définie comme suit :



Alors, la réponse :

$$b_0(t) = (t) - (t - T), \forall 0 \leq t \leq T \quad (1.7)$$

La fonction de transfert continue du bloqueur BOZ s'écrit par :

$$B_0(p) = \frac{1 - e^{-Tp}}{p}$$

### 1.5.2. Bloqueur d'ordre un (BOU)

Le bloqueur d'ordre un (BOU) permet l'extrapolation entre les instants d'échantillonnage  $kT$  et  $(k + 1)T$  à partir de  $f(kT)$  et de  $f((k-1)T)$ . Il permet donc de reproduire un signal échantillonné. Le bloqueur BOU étant représenté par le schéma fonctionnel suivant :

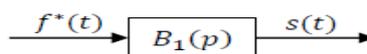


Figure (1.8): Schéma fonctionnel d'un bloqueur d'ordre un (BOU)

Sa fonction de transfert continue est donnée par l'équation suivante :

$$B_1(p) = (1 - e^{-Tp})^2 \left( \frac{1+Tp}{Tp^2} \right)$$

### 1.6. Transformée en 'z' des signaux échantillonnés

D'un point de vue mathématique, la transformation de Laplace est un moyen de traiter les signaux et les systèmes décrits en temps continu ; tandis que la transformation en 'z' est le moyen de traiter les signaux et les systèmes décrits en temps discret.

La transformée en 'z' d'un signal échantillonné  $f^*(t)$ , dénotée  $F(z)$  ou  $Z[f^*(t)]$ , est définie par la série de puissance négatives suivante, où 'z' est une variable complexe :

$$F(z) = Z[f^*(t)] = \sum_{k=0}^{\infty} f(kT) z^{-k}$$

Z représente l'opérateur de la transformée en 'z'.

### 1.7. Propriétés de la transformée en 'z'

La transformée en 'z' possède les propriétés suivantes :

1. *Linéarité* :  $Z[a_1 f_1(t) + a_2 f_2(t)] = a_1 F_1(z) + a_2 F_2(z)$
2. *Translations temporelles*
  - *Cas du retard* :  $Z[f(t - mT)] = z^{-m} F(z)$
  - *Cas de l'avance* :  $Z[f(t + mT)] = z^m [F(z) - \sum_{k=0}^{m-1} f(kT) z^{-k}]$
3. *Translations complexes* :  $Z[F(p \mp a)] = F(e^{\mp aT} z)$
8. *Théorème de la valeur initiale* :  $\lim_{k \rightarrow 0} f(kT) = \lim_{z \rightarrow +\infty} F(z)$
9. *Théorème de la valeur finale* :  $\lim_{k \rightarrow +\infty} f(kT) = \lim_{z \rightarrow 1} \left[ \frac{z-1}{z} F(z) \right]$
10. *Théorème de convolution discrète* :  $Z[\sum_{k=0}^{\infty} f_1(kT) \cdot f_2((m-k)T)] = F_1(z) \cdot F_2(z)$

### 1.8. Table de transformée en 'z' de signaux élémentaires

Le tableau suivant montre les transformées en 'z' de certains signaux élémentaires

N°	$f(t)$	$F(z)$	$f(kT)$
1.	$\delta(t)$	1	$\delta(kT)$
2.	$u(t)$	$\frac{1}{1-z^{-1}}$	$u(kT)$
3.	$tu(t)$	$\frac{Tz^{-1}}{(1-z^{-1})^2}$	$kTu(kT)$
4.	$t^n u(t)$	$\lim_{a \rightarrow 0} (-1)^n \frac{d^n}{da^n} \left( \frac{1}{1 - e^{-aT} z^{-1}} \right)$	$(kT)^n u(kT)$
5.	$e^{-at} u(t)$	$\frac{1}{1 - e^{-aT} z^{-1}}$	$e^{-akT} u(kT)$
6.	$t^n e^{-at} u(t)$	$(-1)^n \frac{d^n}{da^n} \left( \frac{1}{1 - e^{-aT} z^{-1}} \right)$	$(kT)^n e^{-akT} u(kT)$
7.	$\sin(\omega t) u(t)$	$\frac{\sin(\omega T) z^{-1}}{1 - 2 \cos(\omega T) z^{-1} + z^{-2}}$	$\sin(\omega kT) u(kT)$
8.	$\cos(\omega t) u(t)$	$\frac{1 - \cos(\omega T) z^{-1}}{1 - 2 \cos(\omega T) z^{-1} + z^{-2}}$	$\cos(\omega kT) u(kT)$
9.	$e^{-at} \sin(\omega t) u(t)$	$\frac{e^{-aT} \sin(\omega T) z^{-1}}{1 - 2 e^{-aT} \cos(\omega T) z^{-1} + e^{-2aT} z^{-2}}$	$e^{-akT} \sin(\omega kT) u(kT)$
10.	$e^{-at} \cos(\omega t) u(t)$	$\frac{1 - e^{-aT} \cos(\omega T) z^{-1}}{1 - 2 e^{-aT} \cos(\omega T) z^{-1} + e^{-2aT} z^{-2}}$	$e^{-akT} \cos(\omega kT) u(kT)$

Tableau (1.2) : Table de transformée en 'z' de signaux élémentaires

## 1.9. Méthodes de calcul de la transformée en 'z'

Afin de calculer la transformée en 'z' d'un signal, deux méthodes de calcul peuvent être utilisées.

### 1.9.1. Première méthode : Passage de (t) à (z)

En fait, la transformée en 'z' d'un signal (t) correspond à la transformée de Laplace du signal échantillonné  $f^*(t)$  :

$$F(z) = Z[f(t)] = \mathcal{L}[f^*(t)]|_{z=e^{Tp}} = F^*(p)|_{z=e^{Tp}} = \sum_{k=0}^{\infty} f(kT) z^{-k}$$

### 1.9.2. Deuxième méthode (Méthode des résidus) :

Lorsqu'on sait la transformée de Laplace d'un signal continu, on peut calculer la transformée du signal en utilisant la méthode des résidus :

$$F(z) = \sum_{p_i} r_i = \sum_{p_i} \left[ \text{résidus de } \frac{F(p)}{1 - e^{Tp} z^{-1}} \right] \Big|_{p=p_i}$$

où :

$p_i$  sont les pôles de la fonction  $F(p)$ .

$r_i$  sont les résidus associés aux pôles  $p_i$ .

- **Cas 1 : Cas de pôles simples**

Lorsque la fonction  $F(p) = \frac{N(p)}{D(p)}$  a des pôles simples, le résidu correspondant à l'un de ces pôles simples a pour expression :

$$r_i = \text{résidu de } \frac{F(p)}{1 - e^{Tp}z^{-1}} \Big|_{p=p_i} = \frac{N(p_i)}{D'(p_i)} \cdot \frac{1}{1 - e^{Tp_i}z^{-1}}$$

avec  $D'(p_i) = \frac{dD(p)}{dp} \Big|_{p=p_i}$

- **Cas 2 : Cas de pôles multiples**

Dans le cas où  $F(p)$  possède des pôles multiples, le résidu correspondant à l'un de ces pôles multiples a pour expression :

$$r_i = \frac{1}{(n-1)!} \frac{d^{n-1}}{dp^{n-1}} \left[ (p - p_i)^n \cdot F(p) \cdot \frac{1}{1 - e^{Tp}z^{-1}} \right] \Big|_{p=p_i}$$

pour un pôle multiple  $p_i$  d'ordre  $n$  ( $n$  : est la multiplicité de pôle  $p_i$ ).

$$f^*(t) = \{f(kT)\} = Z^{-1}[F(z)]$$

**a. Méthode des résidus :** L'ensemble d'échantillons  $\{f(kT)\}$  est obtenu par l'expression suivante :

$$f(kT) = \sum_{p_i} r_i = \sum_{p_i} [\text{résidu de } z^{k-1} \cdot F(z)] \Big|_{z=p_i} \quad (1.26)$$

où  $p_i$  sont les pôles de la fonction  $F(z) = \frac{N(z)}{D(z)}$ .

On définit le résidu  $r_i$  à un pôle d'ordre  $n$  en  $z = p_i$  par :

$$r_i = \frac{1}{(n-1)!} \frac{d^{n-1}}{dz^{n-1}} [(z - p_i)^n \cdot z^{k-1} \cdot F(z)] \Big|_{z=p_i} \quad (1.27)$$

**b. Division polynomiale :** La fonction  $F(z)$  se présente fréquemment comme une fraction rationnelle en ' $z$ ' ou en ' $z^{-1}$ ':  $F(z) = \frac{N(z)}{D(z)} = \frac{N(z^{-1})}{D(z^{-1})}$ . Il s'agit de diviser le numérateur par le dénominateur pour obtenir une série en ' $z^{-1}$ '.

**Exemple 1.6**

Etant donné la fonction en 'z' suivante :  $F(z) = \frac{2z-3}{3z^2+2z-1}$ .

Calculons les premiers échantillons  $f_k = f(kT)$  par division polynomiale :

$$\begin{array}{r|l}
 2z - 3 & 3z^2 + 2z - 1 \\
 \hline
 -2z - \frac{4}{3} + \frac{2}{3}z^{-1} & 0 + \frac{2}{3}z^{-1} - \frac{13}{9}z^{-2} + \frac{32}{27}z^{-3} + \dots \\
 \hline
 -\frac{13}{3} + \frac{2}{3}z^{-1} & \\
 + \frac{13}{3} + \frac{26}{9}z^{-1} - \frac{13}{9}z^{-2} & \\
 \hline
 + \frac{32}{9}z^{-1} - \frac{13}{9}z^{-2} & 
 \end{array}$$

Les premiers échantillons  $f_k = f(kT)$  sont donc:

$$f_0 = 0 ; f_1 = \frac{2}{3} ; f_2 = -\frac{13}{9} ; f_3 = \frac{32}{27}$$

**c. Méthode de la décomposition en éléments simples :** Le principe de cette méthode est décrit comme suit :

- Former la fonction  $\frac{F(z)}{z}$  ;
- Décomposer la fonction  $\frac{F(z)}{z}$  en éléments simples ;
- Tirer l'expression de  $F(z)$  en fonction de 'z<sup>-1</sup>' ;
- Déterminer les échantillons  $f(kT)$  en utilisant la table de la transformée en 'z'.

**Exemple 1.7**

Soit la fonction en 'z' suivante :  $F(z) = \frac{1}{z^2-3z+2}$ .

Calculons les échantillons  $f_k = f(kT)$ :

Tout d'abord, formons  $\frac{F(z)}{z}$ :  $\frac{F(z)}{z} = \frac{1}{z(z^2-3z+2)}$  ;

Décomposons  $\frac{F(z)}{z}$  en éléments simples :  $\frac{F(z)}{z} = \frac{1}{z(z^2-3z+2)} = \frac{1}{z(z-1)(z-2)} = \frac{a}{z} + \frac{b}{z-1} + \frac{c}{z-2}$  ;

Les coefficients a, b et c sont calculés comme suit :

$$\begin{aligned}
 a &= z \frac{F(z)}{z} \Big|_{z=0} = z \frac{1}{z(z-1)(z-2)} \Big|_{z=0} = \frac{1}{(z-1)(z-2)} \Big|_{z=0} = \frac{1}{2} \\
 b &= (z-1) \frac{F(z)}{z} \Big|_{z=1} = (z-1) \frac{1}{z(z-1)(z-2)} \Big|_{z=1} = \frac{1}{z(z-2)} \Big|_{z=1} = -1 \\
 c &= (z-2) \frac{F(z)}{z} \Big|_{z=2} = (z-2) \frac{1}{z(z-1)(z-2)} \Big|_{z=2} = \frac{1}{z(z-1)} \Big|_{z=2} = \frac{1}{2}
 \end{aligned}$$

d'où

$$\frac{F(z)}{z} = \frac{1}{2} \cdot \frac{1}{z} - \frac{1}{z-1} + \frac{1}{2} \cdot \frac{1}{z-2}$$

Tirons l'expression de  $F(z)$  :  $F(z) = \frac{1}{2} - \frac{z}{z-1} + \frac{1}{2} \cdot \frac{z}{z-2}$  ;

Exprimons  $F(z)$  en fonction de ' $z^{-1}$ ' :  $F(z) = \frac{1}{2} - \frac{1}{1-z^{-1}} + \frac{1}{2} \cdot \frac{1}{1-2z^{-2}}$  ;

Introduisons l'opérateur  $Z^{-1}$  :

$$f_k = f(kT) = Z^{-1}[F(z)] = Z^{-1}\left[\frac{1}{2}\right] - Z^{-1}\left[\frac{1}{1-z^{-1}}\right] + \frac{1}{2} \cdot Z^{-1}\left[\frac{1}{1-2z^{-2}}\right]$$

$$f_k = Z^{-1}[1] - Z^{-1}\left[\frac{1}{1-z^{-1}}\right] + \frac{1}{2} \cdot Z^{-1}\left[\frac{1}{1-(2z^{-2})}\right]$$

Calculons les échantillons  $f_k = f(kT)$  en utilisant la table de la transformée en ' $z$ ' :

$$f_k = \frac{1}{2} \delta(k) + \frac{1}{2} 2^k - 1, \quad \forall k \geq 0$$

**d. Méthode de l'équation aux différences** : Cette méthode sera développée dans le chapitre suivant, nous en donnons seulement un exemple ici.

#### **Exemple 1.8**

Soit l'équation aux différences (équation de récurrence) suivante :

$$s(kT) = 0.5s((k-1)T) + e(kT)$$

En appliquant l'opérateur  $Z$ , il vient :

$$Z[s(kT)] = Z[0.5s((k-1)T)] + Z[e(kT)]$$

En se basant sur les propriétés de la transformée en ' $z$ ', on aura :

$$S(z) = 0.5z^{-1}S(z) + E(z)$$

Formons la fonction  $F(z) = \frac{S(z)}{E(z)}$  :

$$F(z) = \frac{1}{1-0.5z^{-1}} = \frac{z}{z-0.5}$$

## Modélisation des systèmes en BO et BF échantillonnés.

### 2.1. Définitions des systèmes

**2.1.1. Automatique :** Ensemble des techniques qui permettent d'assurer le contrôle d'un système dynamique (procédé physique) sans intervention humaine. En fait, l'automatique est une science qui traite de l'identification, de la modélisation et de la commande des systèmes dynamiques.

**2.1.2. Modélisation :** Le principe de la modélisation est de développer un modèle qui imite le plus fidèlement possible la dynamique d'un système réel.

**2.1.3. Commande :** L'objectif de la commande est d'agir sur les entrées d'un système afin de :

- maintenir la sortie du système constante : '*mode en régulation*' ;

- faire suivre à certaines sorties du système une consigne variable : '*mode en asservissement*'.

**2.1.4. Commande en boucle ouverte (BO) :** Dans une commande en boucle ouverte (BO), le signal de commande est indépendant du signal de sortie. Cette commande ne comporte pas de contre-réaction entre la sortie et l'entrée.

**2.1.5. Commande en boucle fermée (BF) :** Dans une commande en boucle fermée (BF), le signal de commande dépend d'une façon ou d'une autre du signal de sortie.

**2.1.6. Systèmes linéaires :** Un système est dit linéaire s'il répond au principe de superposition et de proportionnalité.

**2.1.7. Systèmes monovariables :** Un système est dit monovisible s'il possède un seul signal d'entrée et un seul signal de sortie.

**2.1.8. Systèmes continus :** Un système est dit continu lorsque les grandeurs le caractérisant délivrent une information à chaque instant continu ' $t$ ' (on parle de domaine du temps continu).

**2.1.9. Systèmes discrets :** Un système est dit discret lorsque les grandeurs le caractérisant délivrent une information à chaque pas de discrétisation ' $kiT$ ' (on parle de domaine du temps discret :  $t=kiT$ ).

**2.1.10. Systèmes discrets invariants dans le temps :** Un système discret est invariant dans le temps ou stationnaire si un décalage temporel de l'impulsion unité appliquée à son entrée provoque le même décalage temporel de la sortie.

**2.1.11. Systèmes linéaires invariants dans le temps () :** Un système est dit linéaire invariant dans le temps s'il est décrit par des équations différentielles (ou des équations récurrentes) linéaires à coefficients constants réels.

### 2.2. 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 2.1 montre le système physique commandé par le ordinateur, dans sa version simplifiée.

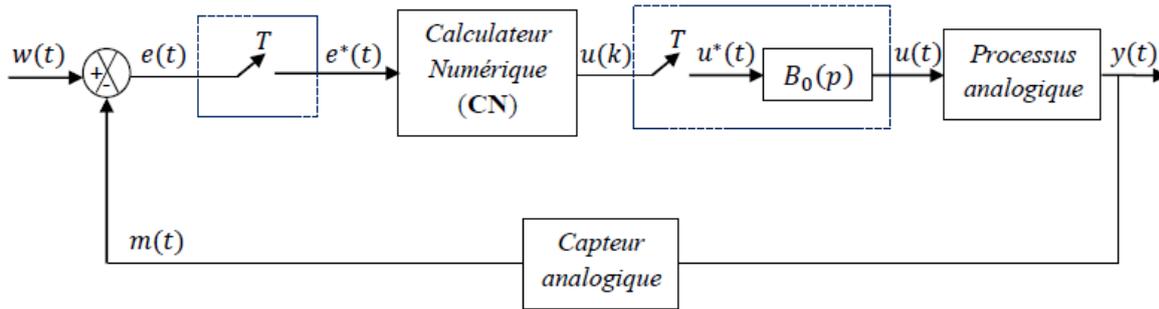


Figure (2.1) : Asservissement échantillonné d'un processus analogique

### 2.3. Discretisation des systèmes linéaires échantillonnés invariants dans le temps :

Considérons un système à temps continu modélisé par sa fonction de transfert  $G(p)$  (Figure 2.2). Nous possédons une bonne connaissance de ce type de modèles et il est tout à fait légitime de s'interroger sur l'existence d'un système échantillonné possédant les mêmes caractéristiques, c'est-à-dire le même comportement temporel et le même comportement fréquentiel.

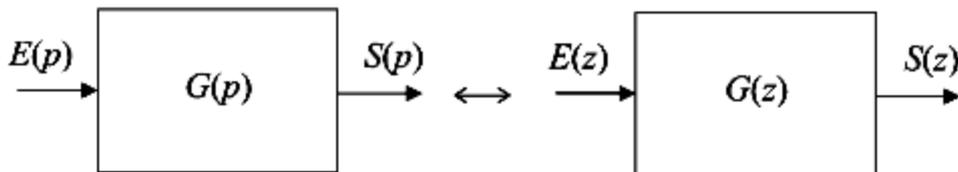


Figure.2.2 : Recherche d'une équivalence temps continu – temps discret.

Le système échantillonné  $G(z)$  sera réputé équivalent au système  $G(p)$  si, soumis à un signal d'entrée  $E(z)$  correspondant à l'échantillonnage du signal continu  $e(t)$  représenté par  $E(p)$ , il délivre à sa sortie un signal  $S(z)$  correspondant à l'échantillonnage du signal  $s(t)$  qui aurait été délivré par le système  $G(p)$ .

#### 2.3.1 Équivalence à la dérivation

a Définition :

Une fonction de transfert en temps continu est issue d'une équation différentielle linéaire à coefficients constants. Cette équation est formée de dérivées successives des signaux d'entrée et de sortie. Un des moyens les plus simples d'effectuer le lien entre une représentation en temps continu et en temps discret est de considérer que la variation  $dx/dt$  en temps continu correspond à la variation du signal entre deux instants d'échantillonnage :

$$\frac{dx}{dt} \approx \frac{x(k) - x(k-1)}{T_e}$$

Cette équivalence est d'autant plus vraie que la fréquence d'échantillonnage est grande. Or la transformée en  $z$  de l'expression de droite est :

$$Z\left(\frac{x(k) - x(k-1)}{T_e}\right) = \frac{1}{T_e} X(z)(1 - z^{-1})$$

De même, le terme  $dx/dt$  a pour transformée de Laplace :  $pX(p)$ . Par conséquent, l'équivalence naturelle entre une fonction de transfert continue en  $p$  et sa fonction de transfert échantillonnée en  $z$  est :

$$p \leftrightarrow \frac{1 - z^{-1}}{T_e}$$

Alors, Plusieurs méthodes de discrétisation peuvent être utilisées afin de numériser ce système LTI mono variable continu ( $p$ ), parmi elles on trouve : la discrétisation par un bloqueur d'ordre zéro (**BOZ**) ; la discrétisation par un bloqueur d'ordre un (**BOU**) ; la discrétisation par approximation de Tustin et la discrétisation par approximations d'Euler (arrière et avant).

### 2.3.2. Discrétisation par un bloqueur d'ordre zéro (BOZ)

La fonction de transfert discrète du système peut être obtenue par la discrétisation de la fonction de transfert continue en utilisant un bloqueur d'ordre zéro (BOZ), comme le montre les figures suivantes :

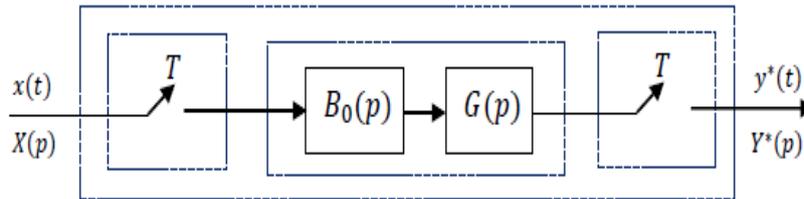


Figure (2.3) : Schéma fonctionnel figurant la discrétisation du système ( $p$ ) continu par un **BOZ**

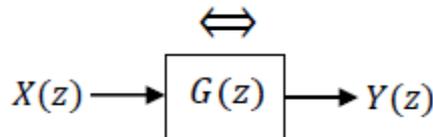


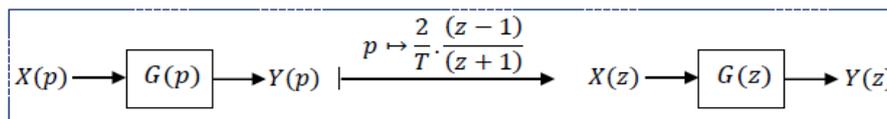
Figure (2.4) : Schéma fonctionnel d'un système LTI discret équivalent.

La fonction de transfert discrète obtenue par le **BOZ** est donnée par :

$$G(z) = (1 - z^{-1})Z \left[ \frac{G(p)}{p} \right]$$

### 2.3.3. Discrétisation par approximation de Tustin

Il est également possible d'obtenir la version discrète du système en utilisant l'approximation de 'Tustin' (approximation bilinéaire) comme le montre la figure suivante :



Il s'agit de remplacer la variable complexe de Laplace  $p$  par l'expression, par conséquent, dans ce cas la fonction de transfert discrète s'obtient comme suit :

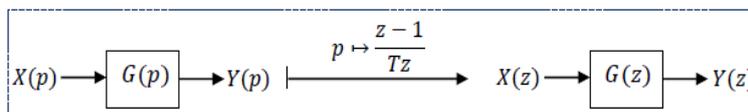
$$G(z) = G(p) \Big|_{p = \frac{2}{T} \cdot \frac{(z-1)}{(z+1)}}$$

### 2.3.4. Discrétisation par approximation d'Euler

Cette approche consiste à approximer la dérivée continue entre deux instants d'échantillonnage (principe d'Euler). En fait, deux approximations sont considérées : discrétisation arrière et discrétisation avant.

#### a. Discrétisation arrière : $p \mapsto \frac{z-1}{Tz}$

Cette discrétisation est représentée par la figure suivante :

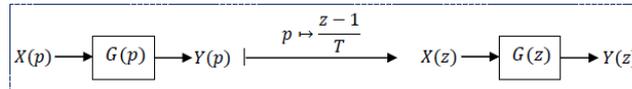


Dans ce cas, la fonction de transfert discrète est obtenue comme suit :

$$G(z) = G(p) \Big|_{p = \frac{z-1}{T}}$$

b. Discrétisation avant :  $p \mapsto z-1/T$

Cette discrétisation est représentée par la figure suivante :



Dans ce cas, la fonction de transfert discrète est obtenue comme suit :

$$G(z) = G(p) \Big|_{p = \frac{z-1}{T}}$$

Applications :

On considère un système continu de premier ordre de gain statique 1 et de constante de temps

$$10 : G_c(p) = \frac{1}{1+10p}$$

La version discrète de  $G_c(p)$  est obtenue comme suit :

- discrétisation par un bloqueur d'ordre zéro (BOZ) :

$$G_d(z) = (1 - z^{-1}) Z \left[ \frac{G_c(p)}{p} \right]$$

Formons  $\frac{G_c(p)}{p}$  :

$$\frac{G_c(p)}{p} = \frac{1}{p(1+10p)} = \frac{1/10}{p(p+1/10)}$$

Décomposons  $\frac{G_c(p)}{p}$  en éléments simples :  $\frac{G_c(p)}{p} = \frac{1/10}{p(p+1/10)} = \frac{a}{p} + \frac{b}{p+1/10}$ .

Calculons les coefficients  $a$  et  $b$  :

$$a = p \frac{1/10}{p(p+1/10)} \Big|_{p=0} = 1$$

$$b = (p + 1/10) \frac{1/10}{p(p+1/10)} \Big|_{z=-1/10} = -1$$

Calculons la transformée en 'z' :

$$Z \left[ \frac{G_c(p)}{p} \right] = Z \left[ \frac{1}{p} - \frac{1}{p+1/10} \right] = Z \left[ \frac{1}{p} \right] - Z \left[ \frac{1}{p+1/10} \right]$$

$$Z \left[ \frac{G_c(p)}{p} \right] = \frac{1}{1-z^{-1}} - \frac{1}{1-e^{-7/10}z^{-1}}$$

Prenons  $T = 10$  (s), on aura :

$$Z \left[ \frac{G_c(p)}{p} \right] = \frac{1}{1-z^{-1}} - \frac{1}{1-e^{-1}z^{-1}}$$

Calculons la version discrète :

$$G_d(z) = (1 - z^{-1}) Z \left[ \frac{G_c(p)}{p} \right]$$

$$G_d(z) = (1 - z^{-1}) \left( \frac{1}{1-z^{-1}} - \frac{1}{1-e^{-1}z^{-1}} \right)$$

il vient :

$$G_d(z) = \frac{(1-e^{-1})z^{-1}}{1-e^{-1}z^{-1}} = \frac{0.6321z^{-1}}{1-0.3679z^{-1}} = \frac{0.6321}{z-0.3679}$$

- discrétisation par approximation de Tustin:

$$\begin{aligned}
 G_d(z) &= G_c(p) \Big|_{p=\frac{2}{T} \cdot \frac{(z-1)}{(z+1)}} \\
 &= \left[ \frac{1}{1+10p} \right] \Big|_{p=\frac{2}{T} \cdot \frac{(z-1)}{(z+1)}} \\
 &= \frac{1}{1+10 \left[ \frac{2}{T} \cdot \frac{(z-1)}{(z+1)} \right]} = \frac{Tz+T}{(T+20)z+(T-20)}
 \end{aligned}$$

d'où

$$G_d(z) = \frac{Tz+T}{(T+20)z+(T-20)}$$

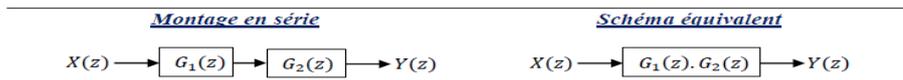
Prenons  $T = 10$  (s), il vient :

$$G_d(z) = \frac{z+1}{3z-1} = \frac{0,3333z+0,3333}{z-0,3333} = \frac{0,3333+0,3333z^{-1}}{1-0,3333z^{-1}}$$

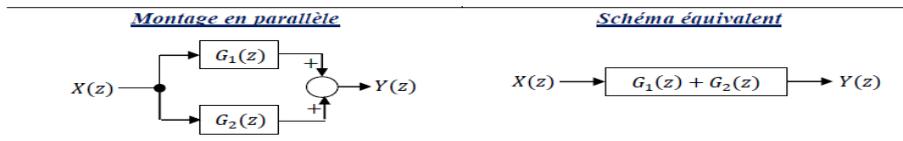
## 2.4. Association des systèmes discrets

Comme dans le cas continu, on peut calculer la fonction de transfert équivalente de deux systèmes LTI à temps discret mis en série, en parallèle ou en rétroaction.

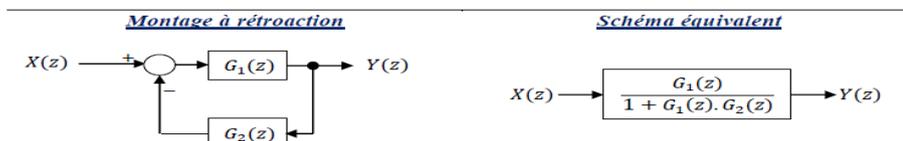
a. Deux systèmes montés en série



b. Deux systèmes montés en parallèle



c. Deux systèmes montés en rétroaction

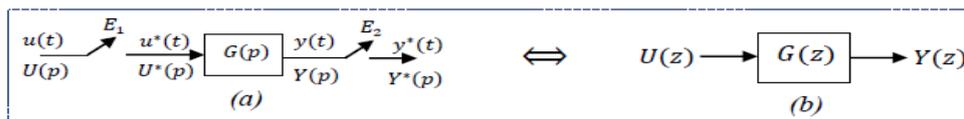


## 2.5. Fonctions de transfert échantillonnées des systèmes complexes

Il s'agit de calculer la fonction de transfert échantillonnée équivalente d'un système résultant de l'association de systèmes élémentaires regroupés pour former, soit des commandes en boucle ouverte, soit en boucle fermée. Généralement, la fonction de transfert échantillonnée équivalente dépend essentiellement au nombre d'échantillonneurs et leurs positions dans la boucle.

### 2.5.1. Systèmes en boucle ouverte (BO)

- Cas 1 : Cas d'un système encadré par deux échantillonneurs



avec

$$G(z) = \mathcal{Z}[G(p)] = \frac{\mathcal{Z}[Y(p)]}{\mathcal{Z}[U(p)]} = \frac{Y(z)}{U(z)}$$

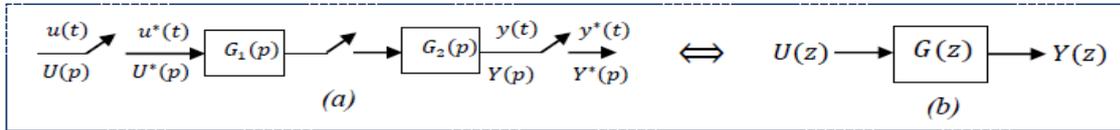
d'où :

$$\begin{aligned}
 Y^*(p) &= [Y(p)]^* = [U^*(p) \cdot G(p)]^* \\
 &= U^*(p) \cdot [G(p)]^* \\
 &= U^*(p) \cdot G^*(p) \xrightarrow{z = e^{Tp}} Y(z) = U(z) \cdot G(z)
 \end{aligned}$$

Par conséquent :

$$Y(z) = U(z) \cdot G(z) \Leftrightarrow G(z) = \frac{z[Y(p)]}{z[U(p)]} = \frac{Y(z)}{U(z)} = Z[G(p)]$$

- Cas 2 : Cas d'un échantillonneur intermédiaire dans la chaîne:

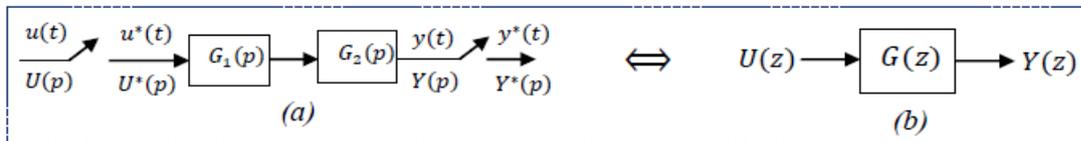


avec

$$G(z) = \frac{z[Y(p)]}{z[U(p)]} = Z[G_1^*(p)] \cdot Z[G_2^*(p)]$$

$$G(z) = G_1(z) \cdot G_2(z)$$

- Cas 3 : Cas sans échantillonneur intermédiaire dans la chaîne



avec

$$G(z) = \frac{z[Y(p)]}{z[U(p)]} = Z[G_1(p) \cdot G_2(p)]$$

$$G(z) = G_1 G_2(z)$$

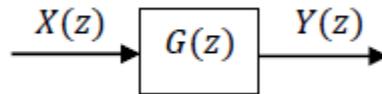
### 2.5.2. Systèmes en boucle fermée : systèmes bouclés (BF)

Le tableau suivant résume les cinq configurations typiques de systèmes de commande discrets en boucle fermée :

Schéma fonctionnel	Fonction de transfert discrète de la sortie
	$Y(z) = \frac{G(z)U(z)}{1 + GH(z)}$
	$Y(z) = \frac{G(z)U(z)}{1 + G(z)H(z)}$
	$Y(z) = \frac{G_1(z)G_2(z)U(z)}{1 + G_1(z)G_2H(z)}$
	$Y(z) = \frac{G_2(z)G_1U(z)}{1 + G_1G_2H(z)}$
	$Y(z) = \frac{GU(z)}{1 + GH(z)}$

## 2.6. Représentation des systèmes linéaires échantillonnés

Les systèmes LTI discrets mono variables peuvent être représentés par le schéma de la figure suivant ,qui est appelé un diagramme.



Trois façons peuvent être utilisées pour représenter les systèmes linéaires échantillonnés invariants dans le temps, à savoir :

- fonctions de transfert discrètes ;
- équations aux différences (équations récurrentes) ;
- représentation par schéma-bloc.

### 2.6.1 Fonctions de transfert discrètes

D'un point de vue analytique, une fonction de transfert discrète d'un système linéaire invariant dans le temps 'LTI' discret peut être décrite par :

1/

$$G(z) = \frac{N(z)}{D(z)} = \frac{b_0 z^m + b_1 z^{m-1} + \dots + b_m}{z^n + a_1 z^{n-1} + \dots + a_n}$$

2/

$$G(z) = \frac{N(z)}{D(z)} = z^{-d} \frac{b_0 + b_1 z^{-1} + \dots + b_m z^{-m}}{1 + a_1 z^{-1} + \dots + a_n z^{-n}}$$

Ou

$$G(z) = b_0 \frac{(z-z_1)(z-z_2)\dots(z-z_m)}{(z-p_1)(z-p_2)\dots(z-p_n)} = b_0 \frac{\prod_{j=1}^m (z-z_j)}{\prod_{i=1}^n (z-p_i)}$$

### 2.6.2. Equations aux différences (équations récurrentes)

L'équation récurrente joue un rôle équivalent, dans l'étude des systèmes à temps discret, à celui que joue l'équation différentielle dans l'étude des systèmes à temps continu.

De façon formelle, l'équation récurrente d'un système discret, décrit par l'équation :

$$y(k) = \sum_{j=0}^m b_j x(k-d-j) - \sum_{i=1}^n a_i y(k-i)$$

Donc une équation récurrente permet d'exprimer la sortie ( $k$ ) à l'instant ' $k$ ' en fonction des échantillons passés de la sortie et de l'entrée.

Soit la fonction de transfert discrète :  $F(z) = \frac{0.2}{z-0.8}$ .

Déterminons l'équation récurrente correspondante :

L'équation récurrente correspondante est obtenue:

$$y(k) = 0.8y(k-1) + 0.2x(k-1)$$

calculons les premiers échantillons de sortie  $y(k)$  (tableau suivant) :

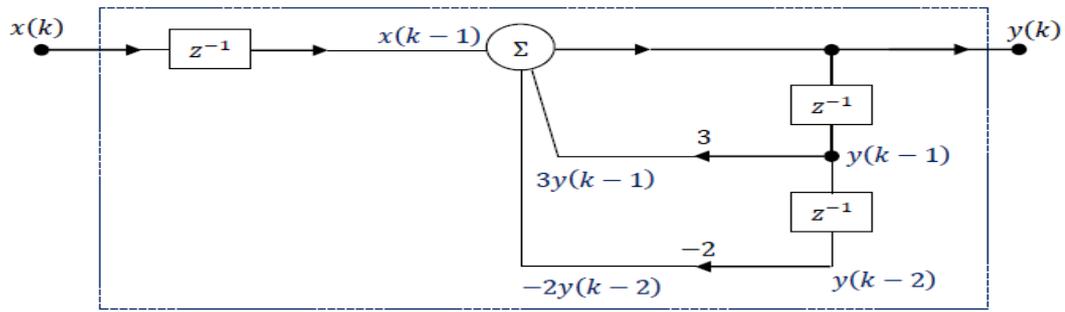
$k$	0	1	2	3	4	5
$x(k)$	1	1	1	1	1	1
$y(k)$	0	0.2	0.36	0.488	0.590	0.672

### 2.6.3. Représentation des systèmes LTI discrets par schéma-blocs

Afin de représenter un système LTI discret par un schéma-blocs, trois opérateurs élémentaires peuvent être utilisés, à savoir : multiplication par un scalaire, sommation algébrique et décalage temporel (retard).

Exemple :

Soit un système LTI discret défini par le schéma-blocs suivant :



A partir du schéma-blocs de la figure, on peut tirer facilement l'équation récurrente correspondante :

$$y(k) = x(k - 1) + 3y(k - 1) - 2y(k - 2)$$

En utilisant les propriétés de la transformée en 'Z', on aura :

$$Y(z) = z^{-1}X(z) + 3z^{-1}Y(z) - 2z^{-2}Y(z)$$

Formons l'expression de la fonction de transfert discrète :

$$F(z) = \frac{Y(z)}{X(z)} = \frac{z}{z^2 - 3z + 2}$$

# Analyse des systèmes

Avant de faire une correction quelconque il faut analyser le système. Dans ce chapitre nous aborderons le lien entre les pôles en  $p$  et les pôles en  $z$  afin de comprendre comment le système réagit à une entrée de consigne et dans quelle mesure il est possible de transformer cette réaction. **Deux points** sont fondamentaux, **la stabilité du système** et **sa précision**. En et on cherche toujours à améliorer ces deux points lorsque l'on asservit un système.

Notez que ces analyses sont totalement indépendantes du fait que l'on parle d'un système en boucle ouverte ou en boucle fermée. Hormis l'analyse de la stabilité par le critère de Nyquist (non abordé dans ce cours) qui prédit la stabilité d'un système boucle par un retour unitaire, l'ensemble des autres points s'appliquent sur une fonction de transfert, celle-ci représentant soit un système en boucle ouverte soit en boucle fermée.

## 3.1 Stabilité

**Définition 1** : Un système est dit stable si écarté de sa position de repos, celui-ci revient à cette position lorsque la cause qui l'en a écarté cesse.

**Définition 2** : Un système est dit stable si sa réponse à toute entrée bornée est bornée.

### 3.1.1 Conditions de stabilité :

Appliquons la définition 1 sur un système quelconque.

Soit un système donné par sa transformée en  $z$ ,  $G(z)$

$$G(z) = \frac{S(z)}{E(z)} = \frac{b_0 + b_1z + b_2z^2 + \dots + b_mz^m}{a_0 + a_1z + a_2z^2 + \dots + a_nz^n} \text{ avec } m < n$$

ainsi la sortie  $S(z)$  est donnée par :

$$S(z) = \frac{b_0 + b_1z + b_2z^2 + \dots + b_mz^m}{a_0 + a_1z + a_2z^2 + \dots + a_nz^n} E(z)$$

Le système étant initialement en équilibre, soit en 0, si on applique une petite perturbation sur l'entrée, le système doit revenir à sa position d'équilibre, soit 0. En prenant comme perturbation  $e(t) = \delta(t)$ , soit  $E(z) = 1$ , la sortie  $S(z)$  devient :

$$S(z) = \frac{b_0 + b_1z + b_2z^2 + \dots + b_mz^m}{a_0 + a_1z + a_2z^2 + \dots + a_nz^n}$$

et par conséquent,  $s(kT_e)$  est de la forme :

$$s(kT_e) = C_1z_1^k + C_2z_2^k + C_3z_3^k + \dots + C_nz_n^k$$

les  $C_i$  et les  $z_i$  étant complexes. En utilisant la forme  $z_i = \rho_i e^{j\theta_i}$

$$s(kT_e) = C_1(\rho_1 e^{j\theta_1})^k + C_2(\rho_2 e^{j\theta_2})^k + C_3(\rho_3 e^{j\theta_3})^k + \dots + C_n(\rho_n e^{j\theta_n})^k$$

$$s(kT_e) = C_1\rho_1^k e^{jk\theta_1} + C_2\rho_2^k e^{jk\theta_2} + C_3\rho_3^k e^{jk\theta_3} + \dots + C_n\rho_n^k e^{jk\theta_n}$$

Pour que le système soit stable, il faut alors que :

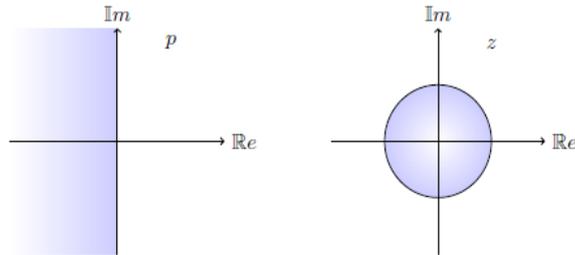
$$\lim_{k \rightarrow \infty} s(kT_e) = 0$$

donc que tous les  $\rho_i$  soit inférieurs à 1, c'est à dire :

$$|z_i| < 1$$

En d'autres termes, pour qu'un système soit stable, il faut et il suffit que les pôles de la fonction de transfert soient tous de module inférieur à 1.

Un système linéaire discret décrit par une fonction de transfert rationnelle  $G(z)$  est stable si seulement si tous ses pôles sont à l'intérieur du cercle unité (de rayon  $R=1$  et de centre  $(0,0)$  dans le plan complexe ' $Z$ '), comme le montre la figure suivante



### 3.1.2. Théorème de stabilité

Soit le système discret décrit par la fonction de transfert :

$$G(z) = \frac{Y(z)}{U(z)} = \frac{b_0 z^m + b_1 z^{m-1} + \dots + b_m}{a_0 z^n + a_1 z^{n-1} + \dots + a_n} = \frac{N(z)}{D(z)}$$

On définit les pôles  $p_i$  du système  $G(z)$  qui sont les racines de l'équation caractéristique :

$$D(z) = a_0 z^n + a_1 z^{n-1} + \dots + a_n = 0$$

alors :

$$\text{Si } |p_i| < 1 \Leftrightarrow \text{Le système } G(z) \text{ est stable}$$

### 3.1.2 Critère de Jury

Soit  $H(z) = \frac{N(z)}{D(z)}$  la fonction de transfert d'un système échantillonné.

$$D(z) = a_0^0 + a_1^0 z + a_2^0 z^2 + \dots + a_n^0 z^n$$

On construit la matrice de dimensions  $(n-1) \times (n+1)$  suivante :

$$\begin{bmatrix} a_0^0 & a_1^0 & a_2^0 & \dots & a_{n-1}^0 & a_n^0 \\ a_0^1 & a_1^1 & a_2^1 & \dots & a_{n-1}^1 & 0 \\ a_0^2 & a_1^2 & a_2^2 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & & & \vdots \\ a_0^{n-2} & a_1^{n-2} & a_2^{n-2} & 0 & \dots & 0 \end{bmatrix}$$

dont les éléments sont définis comme suit :

$$a_k^{j+1} = \begin{cases} \begin{vmatrix} a_0^j & a_{n-j-k}^j \\ a_{n-j}^j & a_k^j \end{vmatrix} & \text{pour } 0 \leq k \leq n-j-1 \\ 0 & \text{pour } k > n-j-1 \end{cases}$$

Le polynôme  $D(z)$  n'a aucun zéro de module supérieur à 1 si les  $n+1$  conditions suivantes sont respectées :

1.  $\sum_{i=0}^n a_i^0 = D(1) > 0$
2.  $(-1)^n \sum_{i=0}^n (-1)^i a_i^0 = (-1)^n D(-1) > 0$
3.  $|a_0^0| - a_n^0 < 0$
4.  $|a_0^j| - |a_{n-j}^j| > 0$  pour  $j = 1, \dots, n-2$

Note : Si  $D(1) = 0$ , cela veut dire que 1 est racine de  $D(z)$  donc qu'il y a un pôle en 1, le système est au mieux *marginalelement stable*. On factorise alors  $D(z)$  sous la forme  $D(z) = (z-1)D'(z)$  et étudie la stabilité de  $D'(z)$ .

Exemple à l'ordre 3 ( $n = 3$ ) :

$$D(z) = a_0^0 + a_1^0 z + a_2^0 z^2 + a_3^0 z^3$$

$$\begin{bmatrix} a_0^0 & a_1^0 & a_2^0 & a_3^0 \\ a_0^1 & a_1^1 & a_2^1 & 0 \end{bmatrix}$$

1.  $\sum_{i=0}^3 a_i^0 = D(1) > 0$
2.  $(-1)^3 \sum_{i=0}^3 (-1)^i a_i^0 = (-1)^3 D(-1) > 0$  soit  $D(-1) < 0$
3.  $|a_0^0| - a_3^0 < 0$
4.  $|a_0^1| - |a_2^1| > 0$  soit  $(a_0^0)^2 - (a_3^0)^2 < a_0^0 a_2^0 - a_1^0 a_3^0$

### 3.1.3 Critère de Routh-Hurwitz appliqué sur la transformée en $w$

**Transformée en  $w$**  La transformée en  $w$  est une transformation homographique qui fait correspondre exactement l'intérieur du cercle unité au demi plan gauche du plan complexe.

$$w = \frac{z-1}{z+1} \quad z = \frac{1+w}{1-w}$$

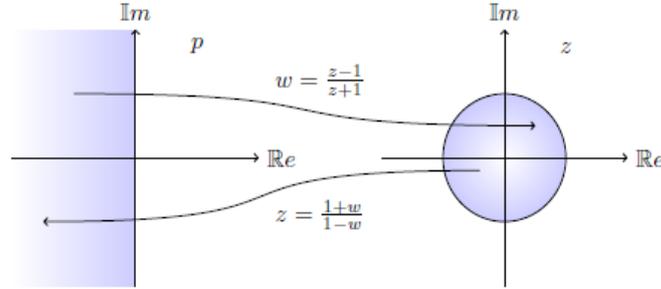


FIGURE 3.3 – Illustration de la transformée homographique  $w$ .

En appliquant la transformée en  $w$  au système, les racines en  $z$  de module inférieur à 1 sont transformées en des racines à partie réelle négative, il suffit alors d'appliquer le critère de Routh-Hurwitz sur la transformée en  $w$  pour connaître le signe des racines du polynôme considéré et donc la stabilité du système.

### 3.2 Correspondance des plans $z$ et $p$

L'objectif est de déterminer le type de comportement du système à la vue des pôles du système tracés dans le plan complexe.

Étudions le lien entre un pôle simple en  $p$  et son transformé par la transformation en  $z$

**Pôles simples**

$$\frac{1}{p+a} \longrightarrow \frac{z}{z - e^{-aT_e}}$$

Le pôle en  $-a$  est transformé en un pôle en  $e^{-aT_e}$

L'axe des réels en  $p$  est transformé en l'axe des réels positifs en  $z$ . L'axe des réels négatifs en  $z$  n'a pas de correspondant en  $p$ .

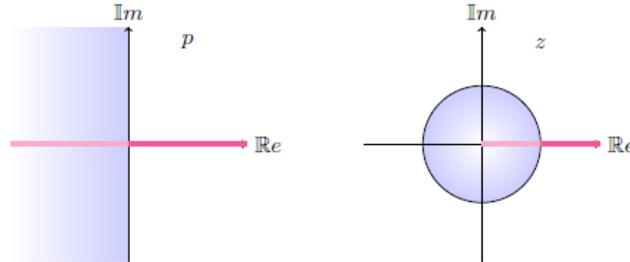


FIGURE 3.4 – Transformation des pôles simples de  $p$  vers  $z$ .

**Pôles complexes conjugués**

$$G(p) = \frac{\omega_n^2}{p^2 + 2\xi\omega_n p + \omega_n^2} \longrightarrow B_0 G(z) = \frac{b_1 z + b_0}{z^2 - 2(e^{-\xi\omega_k T_e}) \cos(\omega_d T_e) z + e^{-2\xi\omega_k T_e}}$$

$$\begin{aligned} p_1, \bar{p}_1 &= -\xi\omega_n \pm i\omega_d \longrightarrow z_1, \bar{z}_1 = e^{-\xi\omega_k T_e} \cos(\omega_d T_e) \pm \sqrt{e^{-2\xi\omega_k T_e} \cos^2(\omega_d T_e) - e^{-2\xi\omega_k T_e}} \\ \text{où } \omega_d &= \omega_n \sqrt{1 - \xi^2} \\ &= e^{-\xi\omega_k T_e} (\cos(\omega_d T_e) \pm i \sin(\omega_d T_e)) \\ &= e^{-\xi\omega_k T_e} e^{\pm i\omega_d T_e} \end{aligned} \tag{3.1}$$

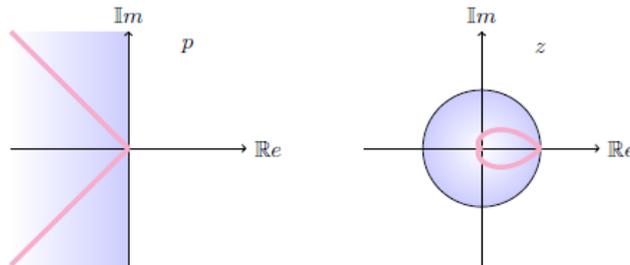


FIGURE 3.5 – Transformation des pôles complexes conjugués de  $p$  vers  $z$ .

**Généralisation** Dans les deux cas nous avons bien la relation :

$$\begin{aligned} \text{Pôle du système continu} &\longrightarrow \text{Pôle du système échantillonné} \\ p_i &\longrightarrow z_i = e^{T_e p_i} \end{aligned} \tag{3.2}$$

Les figures 3.6 et 3.7 montre cette relation.

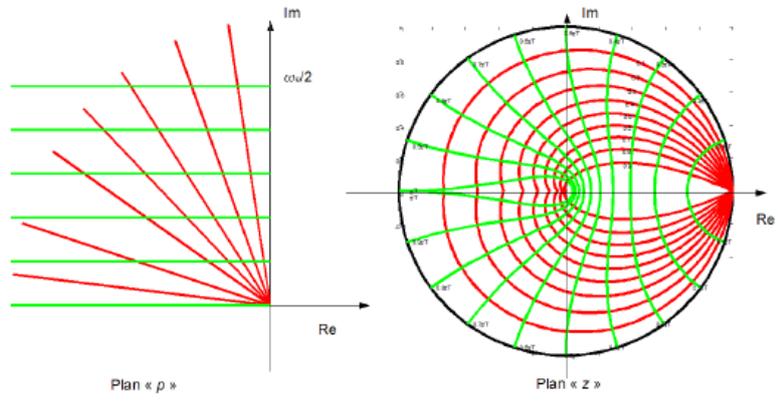


FIGURE 3.6 – Lieu des pôles iso-amortissement.

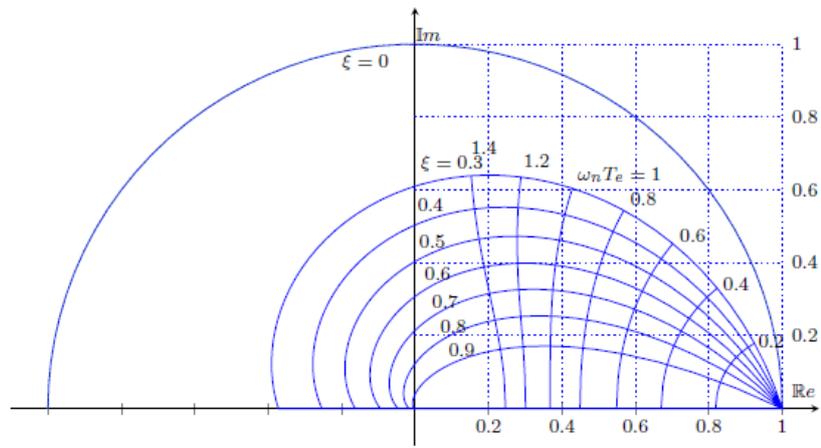


FIGURE 3.7 – Lieu des pôles iso-amortissement.

Dans les pages suivantes, on présente une pratique de l'exploitation du critère pour les cas particuliers de polynômes de deuxième, troisième et quatrième degré:

Cas 1 : polynômes de second degré

$$d(z) = a_2 z^2 + a_1 z^1 + a_0 \text{ avec } a_2 > 0$$

Le critère de Jury impose les conditions suivantes :

$$\text{Condition1 : } |a_0| < a_2$$

$$\text{Condition2 : } d(+1) = a_2 + a_1 + a_0 > 0 \text{ et } d(-1) = a_2 - a_1 + a_0 > 0$$

Ces conditions imposent que les racines soient comprises entre -1 et +1 .

Cas 2 : polynômes de troisième degré

$$d(z) = a_3 z^3 + a_2 z^2 + a_1 z + a_0 \text{ avec } a_3 > 0$$

$$\text{Condition1 : } |a_0| < a_3$$

$$\text{Condition2 : } a_0^2 - a_3^2 < a_0 a_2 - a_1 a_3$$

Condition 3 :  $d(+1) = a_3 + a_2 + a_1 + a_0 > 0$  et  $d(-1) = -a_3 + a_2 - a_1 + a_0 > 0$

Cas 3 : polynômes de quatrième degré

$$d(z) = a_4 z^4 + a_3 z^3 + a_2 z^2 + a_1 z + a_0 \text{ avec } a_4 > 0$$

Condition 1:  $a_4^2 - a_0^2 < |a_0 a_3 - a_1 a_4|$

Condition 2:  $(a_0 - a_4)^2(a_0 - a_2 + a_4) + (a_1 - a_3)(a_0 a_3 - a_1 a_4) > 0$

Condition :  $d(+1) = a_4 + a_3 + a_2 + a_1 + a_0 > 0$

et  $d(-1) = a_4 - a_3 + a_2 - a_1 + a_0 > 0$

### 3.4 Précision des systèmes échantillonnés

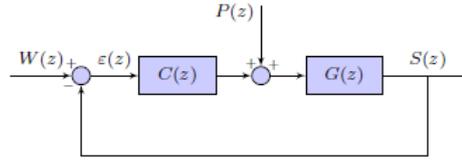


FIGURE 3.11 – Schéma général pour l'étude de la précision des systèmes.

On considère le schéma présenté en figure 3.11. Calculons l'erreur statique du système pour une entrée  $W(z)$  en échelon et une perturbation  $P(z)$  nulle.

#### 3.4.1 Erreur vis-à-vis de la consigne

L'erreur est :

$$\begin{aligned}\varepsilon(z) &= W(z) - S(z) \\ &= W(z) - \varepsilon(z)C(z)G(z) \\ \varepsilon(z)[1 + C(z)G(z)] &= W(z)\end{aligned}$$

$$\varepsilon(z) = \frac{W(z)}{1 + C(z)G(z)}$$

Calculons maintenant la limite de  $\varepsilon(kT_e)$  lorsque  $k \rightarrow \infty$

$$\lim_{t \rightarrow \infty} \varepsilon^*(t) = \lim_{k \rightarrow \infty} \varepsilon(kT_e) = \lim_{z \rightarrow 1} (1 - z^{-1})\varepsilon(z) = \lim_{z \rightarrow 1} (1 - z^{-1}) \frac{W(z)}{1 + C(z)G(z)}$$

Avec une entrée en échelon unité

$$W(z) = \frac{z}{z-1}$$

on obtient :

$$\lim_{k \rightarrow \infty} \varepsilon(kT_e) = \lim_{z \rightarrow 1} (1 - z^{-1}) \frac{z}{z-1} \frac{1}{1 + C(z)G(z)} = \lim_{z \rightarrow 1} \frac{1}{1 + C(z)G(z)}$$

L'objectif étant d'avoir une erreur statique nulle, il faut que

$$\lim_{k \rightarrow \infty} \varepsilon(kT_e) = 0$$

et par conséquent que :

$$\lim_{z \rightarrow 1} \frac{1}{1 + C(z)G(z)} = 0$$

Supposons maintenant que  $C(z)$  ou  $G(z)$  possède un intégrateur pur. La transmittance en boucle ouverte peut s'écrire :

$$C(z)G(z) = \frac{N(z)}{D(z)} = \frac{N(z)}{(z-1)D'(z)}$$

donc

$$\lim_{z \rightarrow 1} \frac{1}{1 + C(z)G(z)} = \lim_{z \rightarrow 1} \frac{1}{1 + \frac{N(z)}{(z-1)D'(z)}} = \lim_{z \rightarrow 1} \frac{(z-1)D'(z)}{(z-1)D'(z) + N(z)} = 0$$

**Conclusion :** pour qu'un système présente une **erreur statique nulle** pour une entrée en échelon, il faut que la transmittance en boucle ouverte présente **au moins un intégrateur pur**.

### 3.4.2 Erreur vis-à-vis de la perturbation

Supposons cette fois que l'entrée  $W(z)$  est nulle et que la perturbation  $P(z)$  est un échelon unité. L'erreur est :

$$\begin{aligned}\varepsilon(z) &= W(z) - S(z) \\ &= 0 - (P(z)G(z) + \varepsilon(z)C(z)G(z)) \\ \varepsilon(z)(1 + C(z)G(z)) &= -P(z)G(z)\end{aligned}$$

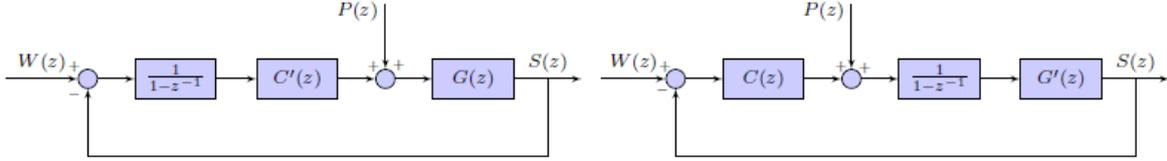
$$\lim_{k \rightarrow \infty} \varepsilon(kT_e) = \lim_{z \rightarrow 1} (1 - z^{-1}) \frac{z}{z-1} \frac{-G(z)}{1 + C(z)G(z)} = \lim_{z \rightarrow 1} \frac{-G(z)}{1 + C(z)G(z)}$$

Supposons que le système en boucle ouverte possède un seul intégrateur pur et étudions les deux cas suivants

- Cas 1 : l'intégrateur pur est dans  $C(z) = \frac{N_C(z)}{D_C(z)}$ .
- Cas 2 : l'intégrateur pur est dans  $G(z) = \frac{N_G(z)}{D_G(z)}$ .

Dans les deux cas :

$$\lim_{k \rightarrow \infty} \varepsilon(kT_e) = \lim_{z \rightarrow 1} \frac{-\frac{N_G(z)}{D_G(z)}}{1 + \frac{N_C(z)}{D_C(z)} \frac{N_G(z)}{D_G(z)}} = \lim_{z \rightarrow 1} \frac{-N_G(z)D_C(z)}{D_C(z)D_G(z) + N_C(z)N_G(z)}$$



Cas 1 : l'intégrateur pur est dans  $C(z)$ .

$$C(z) = \frac{1}{1 - z^{-1}} C'(z) = \frac{-N_C(z)}{(z-1)D'_C(z)}$$

$$\begin{aligned}\lim_{k \rightarrow \infty} \varepsilon(kT_e) &= \\ \lim_{z \rightarrow 1} \frac{-N_G(z)(z-1)D'_C(z)}{(z-1)D'_C(z)D_G(z) + N_C(z)N_G(z)} &= \\ \lim_{k \rightarrow \infty} \varepsilon(kT_e) &= 0\end{aligned}$$

Cas 2 : l'intégrateur pur est dans  $G(z)$ .

$$G(z) = \frac{1}{1 - z^{-1}} G'(z) = \frac{-N_G(z)}{(z-1)D'_G(z)}$$

$$\begin{aligned}\lim_{k \rightarrow \infty} \varepsilon(kT_e) &= \\ \lim_{z \rightarrow 1} \frac{-N_G(z)D_C(z)}{D_C(z)(z-1)D'_G(z) + N_C(z)N_G(z)} &= \\ \lim_{k \rightarrow \infty} \varepsilon(kT_e) &= \lim_{z \rightarrow 1} \frac{-D_C(z)}{N_C(z)} \neq 0\end{aligned}$$

**Conclusion :** pour qu'un système présente une **erreur statique nulle** pour une perturbation en échelon, il faut au moins **un intégrateur pur en amont de la perturbation**.

### 3.4.3 Extension du raisonnement à tous types d'entrées

Calcul de l'erreur vis-à-vis de la consigne, le système étant soumis à une entrée canonique quelconque de la forme  $w(t) = t^m$  donc :

$$W(z) = \frac{A(z)}{(z-1)^{m+1}}$$

où  $A(z)$  est un polynôme en  $z$  n'ayant pas  $(z-1)$  en facteur.

La transmittance en boucle ouverte peut s'écrire :

$$CG(z) = \frac{N(z)}{(z-1)^n D(z)}$$

L'expression de l'erreur est alors :

$$\lim_{k \rightarrow \infty} \varepsilon(kT_e) = \lim_{z \rightarrow 1} (1 - z^{-1}) \frac{A(z)}{(z-1)^{m+1}} \frac{1}{1 + CG(z)} = \lim_{z \rightarrow 1} (1 - z^{-1}) \frac{A(z)}{(z-1)^{m+1}} \frac{1}{1 + \frac{N(z)}{(z-1)^n D(z)}}$$

soit :

$$\lim_{k \rightarrow \infty} \varepsilon(kT_e) = \lim_{z \rightarrow 1} \frac{z-1}{z} \frac{A(z)}{(z-1)^{m+1}} \frac{(z-1)^n D(z)}{(z-1)^n D(z) + N(z)} = \lim_{z \rightarrow 1} \frac{1}{z} \frac{(z-1)^n}{(z-1)^m} \frac{A(z)D(z)}{(z-1)^n D(z) + N(z)}$$

Cas :  $n > m$

$$\lim_{k \rightarrow \infty} \varepsilon(kT_e) = 0$$

Cas :  $n = m = 0$

$$\lim_{k \rightarrow \infty} \varepsilon(kT_e) = \lim_{z \rightarrow 1} \frac{A(z)D(z)}{D(z) + N(z)} = \lim_{z \rightarrow 1} \frac{A(1)D(1)}{D(1) + N(1)} = \frac{A(1)}{1 + \frac{N(1)}{D(1)}} = \frac{1}{1 + K}$$

Note 1 :  $K$  est le gain en boucle ouverte (système + correcteur),  $K = \frac{N(1)}{D(1)}$ .

Note 2 : si  $n = m = 0$  alors l'entrée est un échelon unité donc  $A(z) = z$ .

Cas :  $n = m \geq 1$

$$\lim_{k \rightarrow \infty} \varepsilon(kT_e) = \lim_{z \rightarrow 1} \frac{A(z)D(z)}{(z-1)^n D(z) + N(z)} = \lim_{z \rightarrow 1} \frac{A(z)D(z)}{N(z)} = \frac{A(1)D(1)}{N(1)} = \frac{A(1)}{K} = \frac{T_e}{K}$$

Note 1 :  $K$  est le gain en boucle ouverte (système + correcteur),  $K = \frac{N(1)}{D(1)}$ .

Note 2 :  $A(1) = m!T_e^m$  (voir tables A.1 page 96).

Cas :  $n < m$

$$\lim_{k \rightarrow \infty} \varepsilon(kT_e) = \infty$$

Les valeurs de  $\lim_{k \rightarrow \infty} \varepsilon(kT_e)$  sont résumées dans le tableau 3.6.

TABLE 3.6 – Erreur permanente en fonction de l'entrée et de la classe du système en l'absence d'entrée de perturbation. <sup>4</sup>

classe du système	échelon	rampe	parabole	...
=				...
nb d'intégrateurs purs	$m = 0$	$m = 1$	$m = 2$	...
0	$\frac{1}{1+K}$	$\infty$	$\infty$	...
1	0	$\frac{T_e}{K}$	$\infty$	...
2	0	0	$\frac{2T_e^2}{K}$	...
3	0	0	0	...
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$

**Exercice 1.1 :**

1. Trouver la transformée de Laplace de  $f(t) = 2e^{-t} \cos(10t) - t^4$  pour  $t \geq 0$ .
2. Déterminer la transformée  $Y(p)$  du signal de sortie pour l'équation différentielle  $\frac{d^2y}{dt^2} + 3\frac{dy}{dt} - \frac{dy}{dt} + 6y = 1$ . On donne les conditions initiales suivantes:

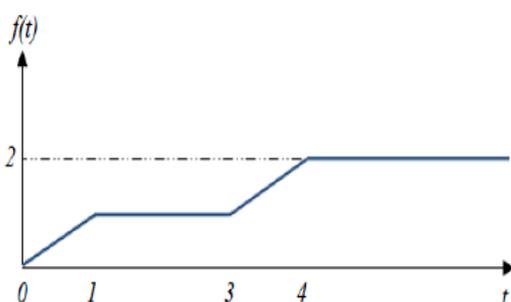
$$y(0^+) = 0, \left. \frac{dy}{dt} \right|_{t=0^+} = 0 \text{ et } \left. \frac{d^2y}{dt^2} \right|_{t=0^+} = 1.$$

3. Déterminer la fonction  $f(t)$  dans les cas suivants :

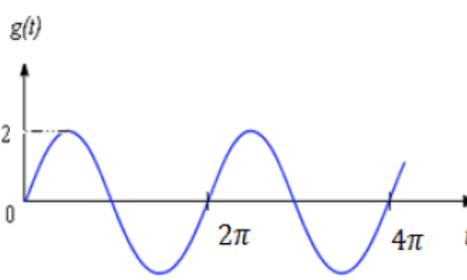
$$F(p) = \frac{1}{p^2 - 2p + 1}, F(p) = \frac{p-1}{p^2 - 2p + 2}, F(p) = \frac{p}{(p+1)(p^2+1)}$$

**Exercice 1.2 :**

1. Echantillonner graphiquement les signaux suivants :



(a). Avec  $T = 0.5$  (s)



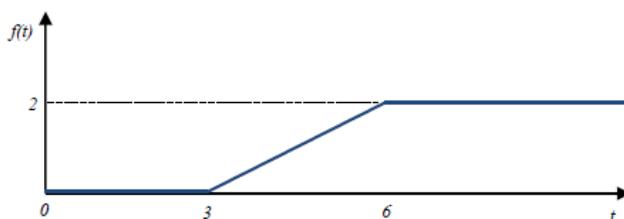
(b). Avec  $T = \pi/6$  (s)

2. Tracer les signaux suivants :

$$f_1^*(t) = \sum_{k=0}^5 (-1)^k T \cdot \delta(t - kT), f_2^*(t) = \sum_{k=0}^4 2^k T \cdot \delta(t - kT), \text{ avec } T = 1 \text{ (s)}.$$

**Exercice 1.3 :**

1. Calculer la transformée en 'z' de la fonction  $f(t)$  définie par la courbe suivante :



On donne  $T = 1$  (s).

2. Calculer la transformée en 'z' de  $f(t) = e^{-4t} + te^{-2t}$ ,  $\forall t \geq 0$  et  $\forall T > 0$ .
3. Calculer la transformée en 'z' de  $f(t)$  telle que :  $3\frac{df(t)}{dt} + 2f(t) = u(t)$ ,  $\forall t \geq 0$  et  $\forall T > 0$ . Avec  $u(t)$  : représente l'échelon unité.

4. Calculer la transformée en 'z' de la fonction  $f$  définie par le tableau suivant :

$kT$	0	1	2	3	4	5	6	7	8	...	$\infty$
$f(kT)$	1	4	6	4	1	0	0	0	0	0	0

**Exercice 1.5 :**

1. En utilisant la méthode des résidus, calculer la transformée en 'z' des fonctions suivantes :

$$F_1(p) = \frac{p+3}{(p+1)(p+2)}, F_2(p) = \frac{2p+5}{p^2-3p+2}, F_3(p) = \frac{p}{(p+1)(p^2-1)}, F_4(p) = \frac{1}{p^2}.$$

2. En utilisant la méthode de décomposition par fractions rationnelles, calculer la transformée en 'z' des fonctions suivantes :

$$F_1(p) = \frac{p+3}{(p+1)(p+2)}, F_2(p) = \frac{1}{(p+1)(p+2)(p+3)}, F_3(p) = \frac{1}{p^2(p-1)}.$$

**Exercice 1.6 :**

1. Déterminer les premiers éléments de la suite d'échantillons  $\{s(k)\}$  dans les cas suivants :

$$S(z) = \frac{z}{z-1}, S(z) = \frac{z}{z^2+1}, S(z) = \frac{1}{z^2-2z}$$

2. Calculer les signaux échantillonnés dans les cas suivants :

a.  $F_1(z) = \frac{z+1}{z+2}, F_2(z) = \frac{1}{z^2+1}, F_3(z) = \frac{(1-\alpha)z^{-1}}{(1-z^{-1})(1-\alpha z^{-1})}$  avec  $\alpha = e^{-aT}$  et  $T$  est la période d'échantillonnage, en utilisant la méthode des résidus.

b.  $G_1(z) = \frac{1}{z-0.5}, G_2(z) = \frac{z-0.5}{z^2-9z+8}, G_3(z) = \frac{2(z+2)}{z^2+3.5z-2}$ , en utilisant la méthode de décomposition en éléments simples.

c.  $H_1(z) = \frac{z}{z-1}, H_2(z) = \frac{z}{z^2+1}, H_3(z) = \frac{z^2}{z^3+1}$ , en utilisant la méthode de division suivant les puissances croissantes de ' $z^{-1}$ '.

**Exercice 2.1 :**

Trouver la version échantillonnée de la fonction de transfert continue  $G(p) = \frac{1}{1+10p}$  en utilisant les méthodes de discrétisation suivantes :

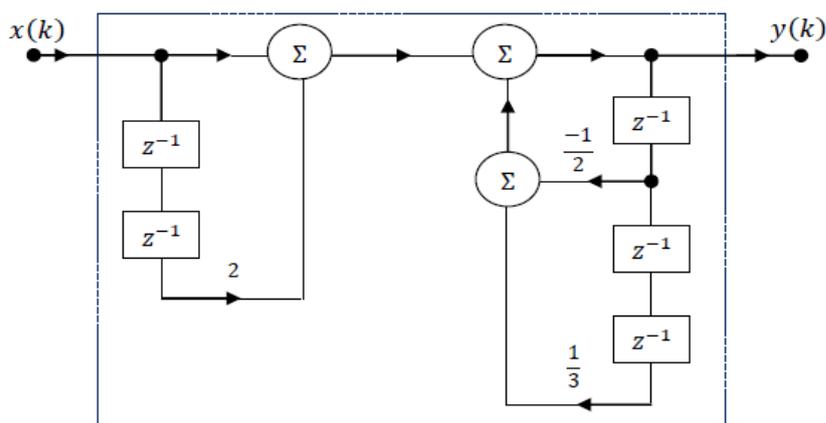
- a. Discrétisation par un bloqueur d'ordre zéro (BOZ).
- b. Approximation de Tustin.
- c. Approximation d'Euler arrière.
- d. Approximation d'Euler avant.

**Exercice 2.2 :**

1. Soit la fonction de transfert échantillonnée :

$$F(z) = \frac{a_1 z^2 + a_2}{b_1 z^2 + b_2 z + b_3}$$

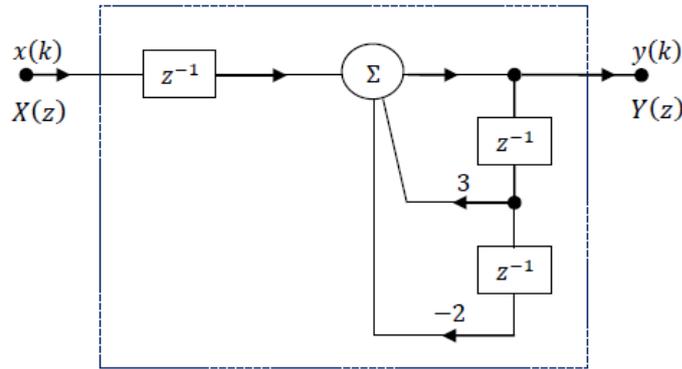
- a. Trouver l'équation récurrente correspondante.
  - b. Tracer le schéma-blocs correspondant.
2. Tracer le schéma-blocs d'un système LTI discret décrit par l'équation récurrente suivante :
- $$y(k) + \frac{1}{4}y(k - 1) + \frac{1}{8}y(k - 2) = x(k) + x(k - 1)$$
3. Soit un système LTI discret défini par le schéma-blocs suivant :



- a. Etablir l'équation récurrente correspondante.
- b. Déduire la fonction de transfert échantillonnée  $F(z) = \frac{Y(z)}{X(z)}$ .
- c. Identifier l'ordre, les zéros, les pôles et le gain du système.
- d. Simplifier le schéma-blocs précédent.

**Exercice 2.3 :**

Soit un système LTI discret défini par le schéma-blocs suivant :



1. Etablir la fonction de transfert échantillonnée  $F(z) = \frac{Y(z)}{X(z)}$ .
2. Quel est l'ordre du système.
3. Identifier les zéros, les pôles et le gain du système.
4. Dédire l'équation récurrente correspondante.
5. Calculer la réponse impulsionnelle du système considéré en boucle ouverte.

**Exercice 2.4:**

Résoudre les équations récurrentes suivantes :

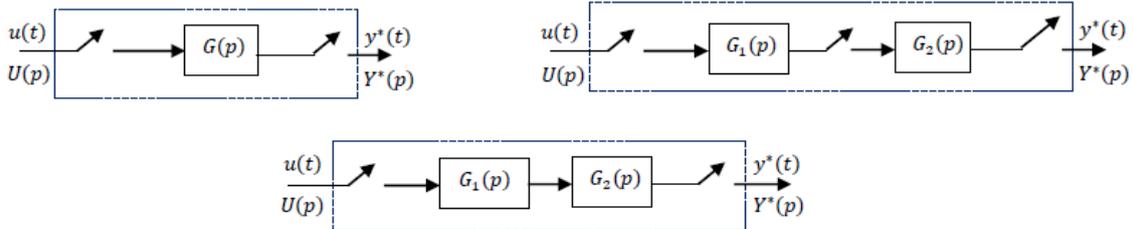
a.  $y(k) - 2y(k - 1) + y(k - 2) = 2u(k - 2)$ .

On donne :  $y(0) = y(1) = 1$  et  $u(k) = 1, \forall k \geq 0$ .

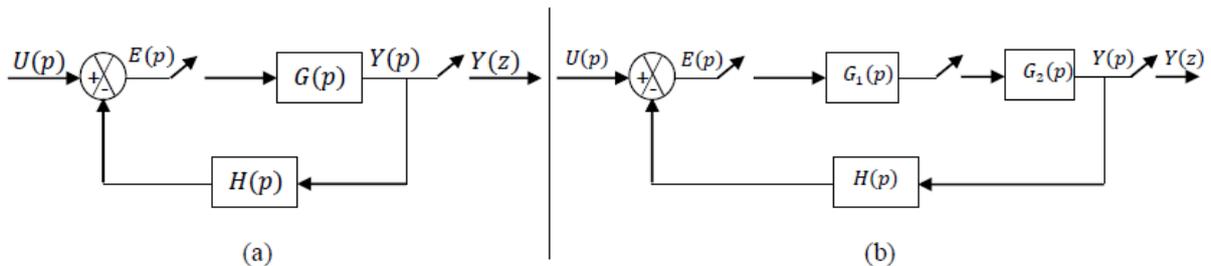
b.  $y(k) = 2r(k - 1) - r(k - 2)$ . On donne :  $y(0) = y(1) = 0$  et  $r(k) = k, \forall k \geq 0$ .

**Exercice 2.5 :**

1. Trouver la fonction de transfert échantillonnée  $G(z) = \frac{Y(z)}{U(z)}$  dans les cas suivants :

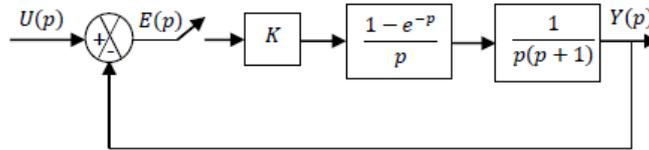


2. Etablir l'expression de la sortie échantillonnée  $Y(z)$  relative aux systèmes asservis échantillonnés (a) et (b) suivants :



**Exercice 3.1 :**

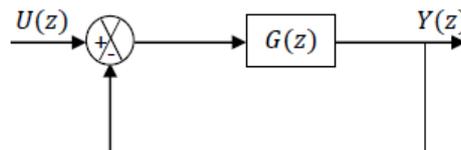
On considère un système de contrôle en boucle fermée illustré par le schéma fonctionnel suivant :



1. En utilisant la définition de la stabilité, étudier la stabilité lorsque le paramètre  $K = 1$  dans les cas suivants :
  - Système en boucle ouverte (sans comparateur).
  - Système en boucle fermée.
2. En utilisant le critère de Jury, évaluer la stabilité du système en boucle fermée en fonction du paramètre  $K$ .

**Exercice 3.2 :**

On a relevé la réponse impulsionnelle  $y(k)$  d'un système asservi numérique bouclé comme le montre le schéma fonctionnel suivant :



Les échantillons de la réponse impulsionnelle obtenus sont donnés dans le tableau suivant :

$k$	0	1	2	3	4	5	6	7	...	$\infty$
$y(k)$	0	0,9	0,1	0	0	0	0	0	...	0

1. Dédurre l'expression de la réponse impulsionnelle du système.
2. Tracer la réponse impulsionnelle du système.
3. Dédurre la fonction de transfert  $G(z)$ .
4. En utilisant le critère de Jury, évaluer la stabilité :
  - du système  $G(z)$  en boucle ouverte ;
  - du système bouclé précédent.

**Exercice 3.3 :**

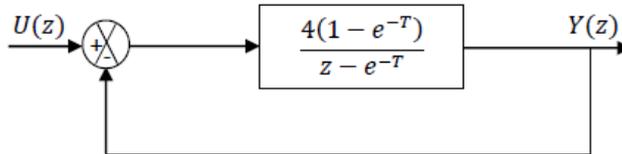
On considère un système discret de fonction de transfert en boucle fermée :

$$F(z) = \frac{K}{z^2 + z + K}$$

Etudier la stabilité de ce système en fonction du paramètre  $K$  à l'aide du critère de Jury.

**Exercice 3.4 :**

Soit le système asservi discret suivant :



Etudier la précision du système bouclé dans les cas suivants :

- $U(z)$  : est un échelon de position.
- $U(z)$  : est un échelon de vitesse.
- $U(z)$  : est un échelon d'accélération.

On donne :  $T = 0.2$  (s).

**Exercice 3.5 :**

Un système échantillonné de fonction de transfert  $G(z)$  est placé dans une boucle de régulation à retour unitaire :

$$G(z) = \frac{0.16K}{(z-0.8)^2}, \text{ avec } K > 0.$$

La période d'échantillonnage est :  $T = 0.1$  (s).

1. Calculer la fonction de transfert en boucle fermée et déterminer la condition de stabilité du système en boucle fermée.
2. Le gain étant réglé sur  $K = 1$ , déterminer, en boucle fermée : l'erreur de position, l'équation de récurrence, et calculer, puis tracer, les premiers éléments de la suite des échantillons de sortie lorsque l'entrée est un échelon unitaire. En déduire la valeur du premier dépassement en boucle fermée.
3. Répondre aux mêmes questions en réglant le gain sur  $K = 2$  puis 5. Conclure.

**Exercice 3.6 :**

Un système à temps continu de fonction de transfert  $G(p) = \frac{K}{p+10}$  est placé dans une boucle de régulation à temps discret à retour unitaire et commandé numériquement. La période d'échantillonnage  $T$  est réglable.

1. Déterminer, en fonction de  $K$  les conditions de stabilité du système échantillonné en boucle fermée. Comparer les conditions de stabilité du système pour  $T = 1$  (s),  $T = 0.5$  (s) et  $T = 0.01$ (s).
2. La valeur du gain étant réglée sur  $K = 10$ , déterminer la condition sur  $T$  pour que le système soit stable.

## 1-Introduction

Ce TP a pour but de présenter et d'initier à la modélisation des systèmes échantillonnés en utilisant le Matlab/Simulink en partant de l'acquisition (procédure d'échantillonnage) et en terminant par la reconstruction (conversion numérique /analogique) en utilisant d'Arduino par l'exploitation de l'application **Ident** du Matlab/simulink afin de valider les résultats de simulation.

Les cartes Arduino sont conçues pour réaliser des prototypes et des maquettes de cartes électroniques pour l'informatique embarquée. Ces cartes permettent un accès simple et peu coûteux à l'informatique embarquée. Le langage Arduino se distingue des langages utilisés dans l'industrie de l'informatique embarquée de par sa simplicité. Cela en fait un langage parfait pour réaliser des prototypes ou des petites applications.

Les possibilités des cartes Arduino sont énormes, un grand nombre d'applications ont déjà été réalisées et testées. D'autres exemples d'applications sont disponibles sur le web.

## 2- Le matériel : Arduino UNO

C'est un circuit imprimé comportant tous les composants électroniques nécessaires pour faire fonctionner un microcontrôleur (Atmega 328) associé à une interface USB lui permettant de communiquer avec un ordinateur.



Figure 1 Description de la Carte Arduino "Uno"

- Microcontroller : ATmega328
- Operating Voltage : 5v
- Input Voltage (recommended) : 7-12 v
- Input Voltage (limits) : 6-20 v
- DC Current per I/O Pin : 40mA
- DC Current for 3.3V Pin :50mA
- Flash Memory :32 KB
- Clock Speed : 16MHz
- Pins assignments :
- Analog read(A0-A5)
- Analog write[PWM] (3,5,6,9,10,11)
- Digital read(2-19)
- Digital write(2-19)

### 2.1 Le logiciel Arduino

Arduino IDE ( Integrated Development Environment ). Le logiciel est gratuit et open source dont la simplicité d'utilisation est remarquable. Ce logiciel va nous permettre de programmer la carte Arduino pour :

- Réaliser l'interfaçage avec Matlab/simulink
- Implémenter la commande directement sur la carte.

### 2.2 L'interfaçage Arduino/Matlab/Simulink

Il existe trois possibilités d'interfacer la carte Arduino avec Matlab/Simulink, à savoir :

1. Programmation de la carte Arduino Uno comme une carte d'interface.

2. Utilisation du package ArduinoIO.
3. Utilisation du package Arduino Target.

### 2.3 Programmation de la carte Arduino Uno comme une carte d'interface

Cette solution consiste d'une part à utiliser les fonctions offert par le langage Arduino qui permet d'envoyer et d'acquérir des données binaires via le port série (USB) et d'autre part à développer sous Simulink un programme pour traiter ou visualiser ces données.

### 2.4 Installation du package ArduinoIO

1. Lancer Matlab2013
2. Exécuter la commande : install-arduino
3. Fermer et relancer Matlab puis Simulink
4. Dans les bibliothèques se trouvent maintenant les blocs dans Arduino IO library.

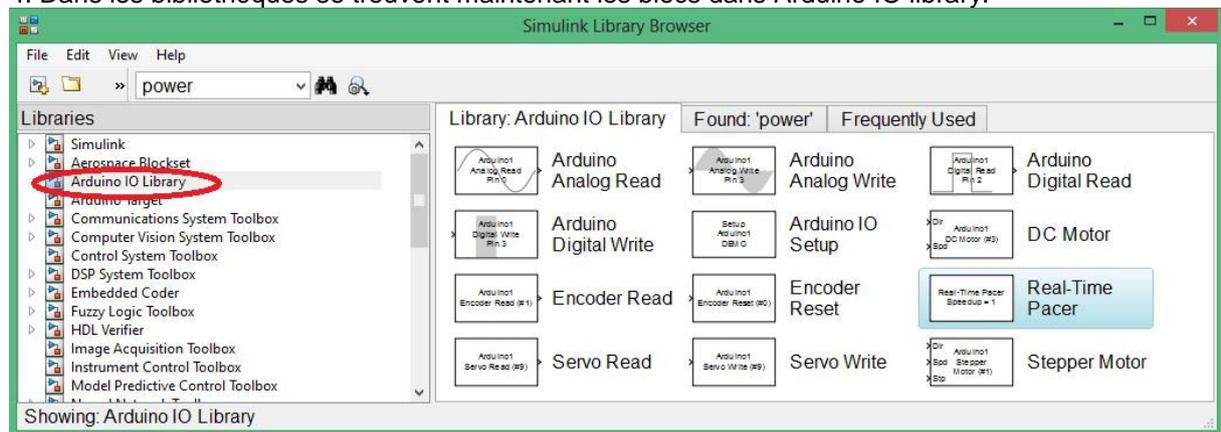


Figure 1 – ArduinoIO Library

### 2.2.3 Exploitation de la bibliothèque ArduinoIO sous Simulink

Les blocs nécessaires pour notre objectif d'asservissement sont les suivants :

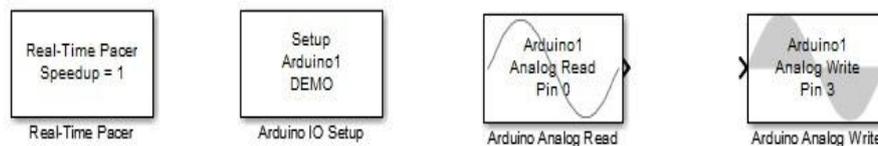


Figure 2 – Les Blocs d'ArduinoIO nécessaires pour l'identification et la commande.

- **Real-Time Pacer** : Ce bloc permet de ralentir le temps de simulation de sorte qu'il synchronise avec le temps réel écoulé. Le coefficient de ralentissement est contrôlable par l'intermédiaire du paramètre *Speedup*.
- **Arduino IO Setup** : Pour configurer sur quel port la carte Arduino UNO est connectée. Pour cela il suffit de voir dans Gestionnaire des périphériques. voir Figure 4.
- **Arduino Analog Read** : Pour configurer à partir de quel pin [0,1,2,3,4,5] on va acquérir les données du capteur.
- **Arduino Analog Write** : Pour configurer à partir de quel pin [3,5,6,9,10,11] on va envoyer la commande en PWM vers l'actionneur.

### 2.2.4 Exploitation du package ArduinoIO sous Matlab

Le package ArduinoIO offre une panoplie de commandes permettant d'écrire un programme sous Matlab (M-file). Pour accéder à ces commandes il faut créer un objet arduino dans l'espace de travail et spécifier le port sur lequel la carte arduino est connecté avec la commande :

```
>> a = arduino( );
```

### 2.3 Arduino Target

Embedded Coder Support Package for Arduino permet de créer des applications Simulink qui vont fonctionner de façon autonome sur la carte Arduino. on dit que la carte Arduino est devenue une cible (Target) et elle peut fonctionner d'une façon autonome (sans avoir recours à Matlab/Simulink).

Dans la suite, on utilisera les blocs Simulink offert par le package *ArduinoIO Library* et la librairie *Instrument Control Toolbox* pour l'acquisition et l'envoi des données.

## 3 Acquisition des données

### 3.1 Présentation du A/DC

La carte Arduino Uno dispose de 6 entrées analogiques notées A0, A1,...A5 mais d'un seul convertisseur analogique/numérique, la durée d'une conversion est de l'ordre de 100µs. Il a une résolution de 10 bits. La donnée numérique qu'il fournit après conversion est donc comprise entre 0 et 1024.

### 4 Envoi des données

#### 4.1 Présentation des sorties analogiques (mode PWM)

La carte Arduino Uno dispose de 6 sorties (3,5,6,9,10 et 11) qui peuvent être utilisées en mode PWM, c'est-à-dire en modulation de largeur d'impulsion. Ce sont des signaux logiques binaires de fréquence constante (500Hz) mais de rapport cyclique variable.



Figure 3 – Description du signal PWM

Lorsqu'un moteur ou une lampe est alimenté par ce type de tension, tout se passe comme si il était alimenté par une tension continue ajustable entre 0V (rapport cyclique= 0) et 5V (rapport cyclique=255). Ces sorties doivent être initialisées comme des sorties digitales.

$$V_{out} = V_s \times \frac{\tau_o}{\tau_c}; \quad \text{avec : } \tau_c = 2ms$$

La syntaxe de l'instruction permettant de générer le signal PWM est la suivante :

**analogWrite(pin, valeur) ;**

- pin : la pin sur laquelle on souhaite envoyer le signal (3,5,6,9,10 ou 11).
- valeur : le rapport cyclique entre 0 et 255.

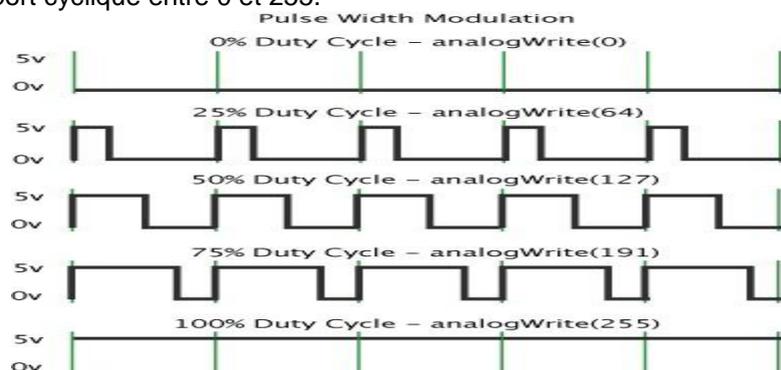
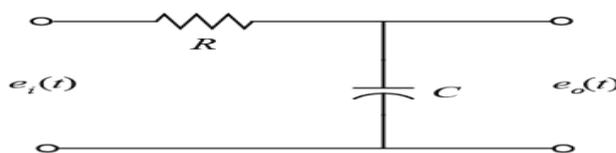


Figure 4 – Exemples de variation du rapport cyclique

### 5-Travail à réaliser :

**Faire le montage suivant :R=200K et C=04.7microH.**



#### a)Acquisition (echantillonnage du signal d'entrée) :

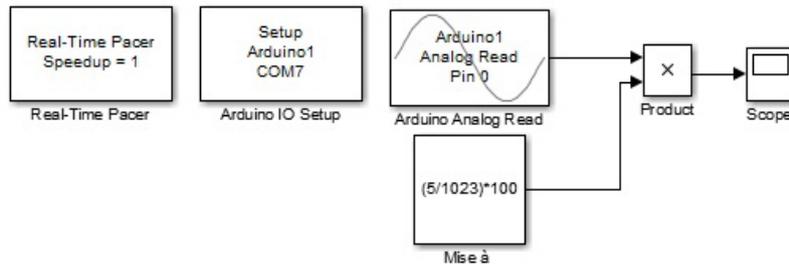


Figure 5 – Acquisition de la tension sur la capacité sous ArduinoIO Library.

**b)Reconstruction du signal analogique(N/A) (Alimentations ou génération de l'entrée) :**

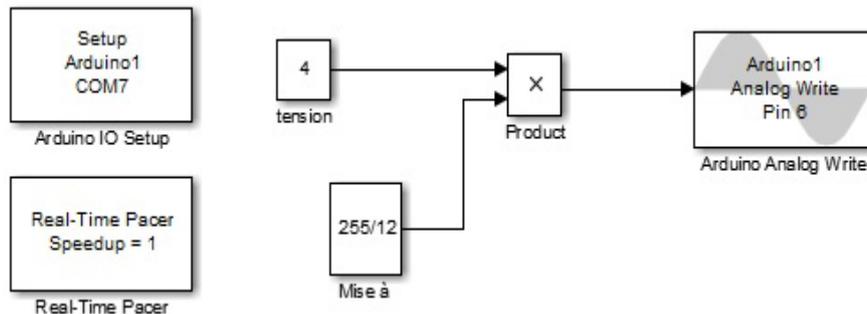


Figure 6 – Envoi de la commande PWM(N/A) sous ArduinIO Library

**c) Modélisation du procédé (Circuit RC) :**

Le but de cette partie est de déterminer la fonction de transfert échantillonnée de notre procédé électrique en boucle ouvert notée  $G(z)$ . L'entrée du système est la tension  $E_i(z)$  en volts et la sortie est la tension au borne de la capacité  $E_o(z)$ .

$$G(s) = \frac{E_o(s)}{E_i(s)} = \frac{\frac{1}{Cs}}{R + \frac{1}{Cs}}$$

**c.1 Présentation de l'étape d'identification avec Matlab**

Cette étape est constituée de deux parties. La première est assurée par l'environnement Simulink et le package ArduinoIO pour l'envoi et l'acquisition des données. La deuxième partie est assurée par l'outil **System identification** sous Matlab.

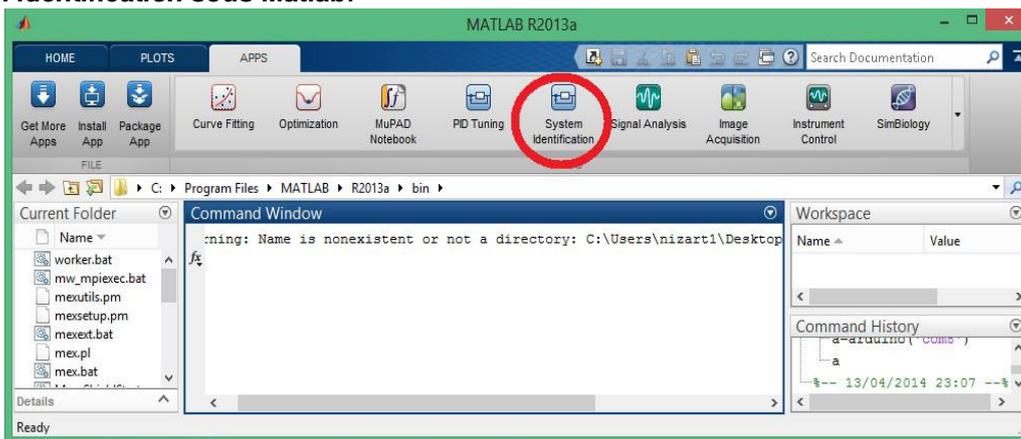


Figure 7 – L'utilisation de l'outil System Identification

**c.2 Acquisition de la réponse indicielle du système**

Plusieurs méthodes sont utilisées pour la modélisation d'un système comme la détermination des équations physiques du système, l'étude de la réponse d'un système à une entrée...etc. Dans notre cas on va identifier notre système en étudiant la réponse de notre système à échelon de tension, ou à un signal carré ou sinusoïdale (exploitation fréquentielle-Bode).

Le modèle Simulink permettant de réaliser l'acquisition de la réponse du système à un échelon de tension est le suivant :

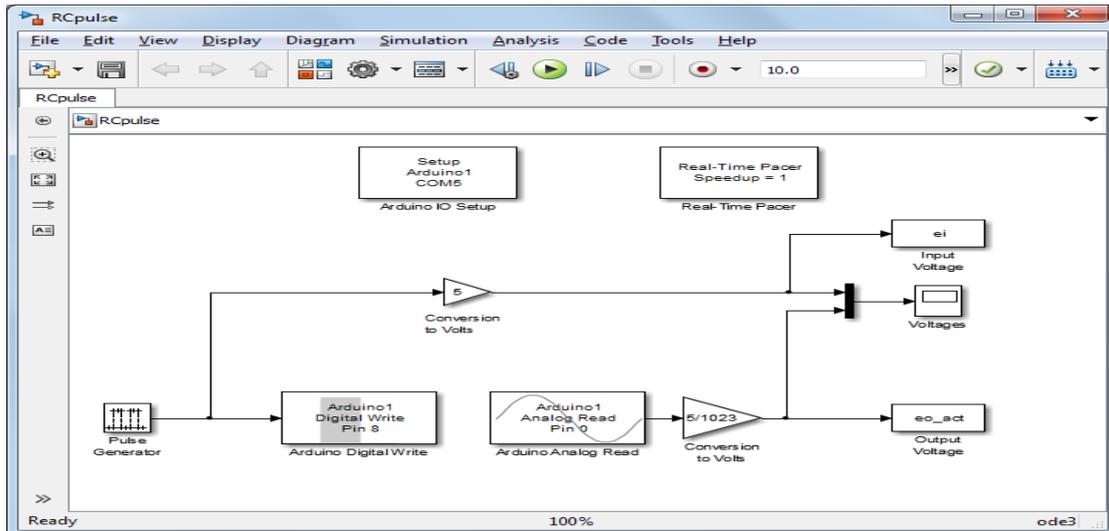


Figure 7 – Modèle Simulink pour la détermination de la réponse indicielle ou periodique.

### c.3 Détermination de la fonction de transfert $G(z)$ [tp 2 cours] :

Après avoir déterminé la réponse du système, on passe à la détermination de la fonction de transfert  $G(z)$ .

#### 1. Ouvrir l'outil *System identification Tool*

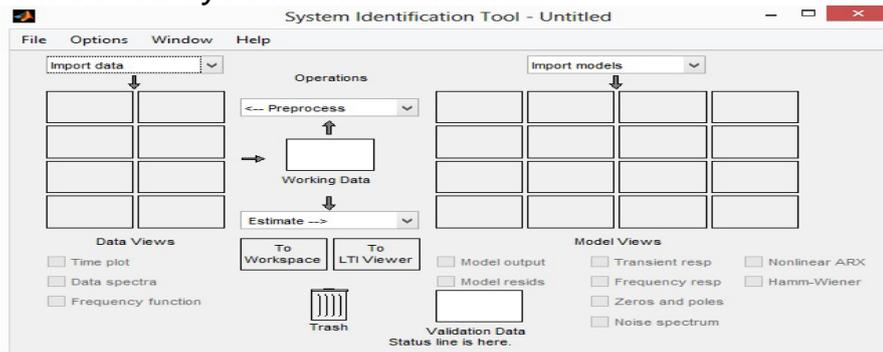


Figure 8 – L'interface de l'outil System identification

#### 2. Cliquer sur *import data* et choisir *Time domain data*.

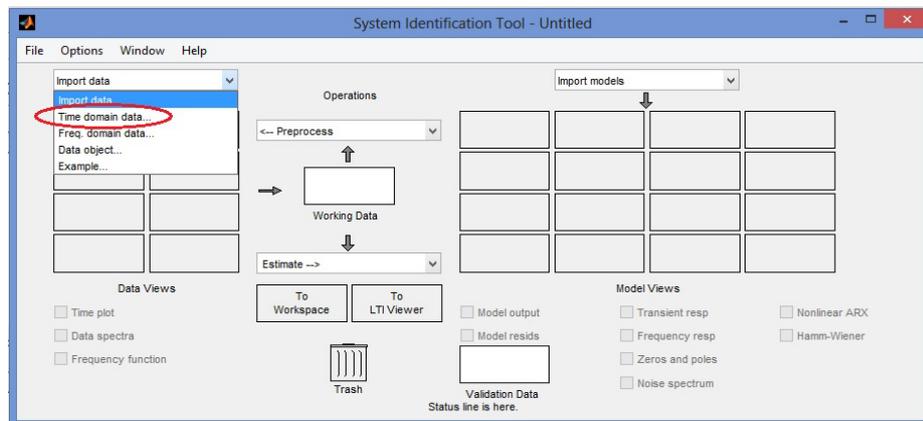
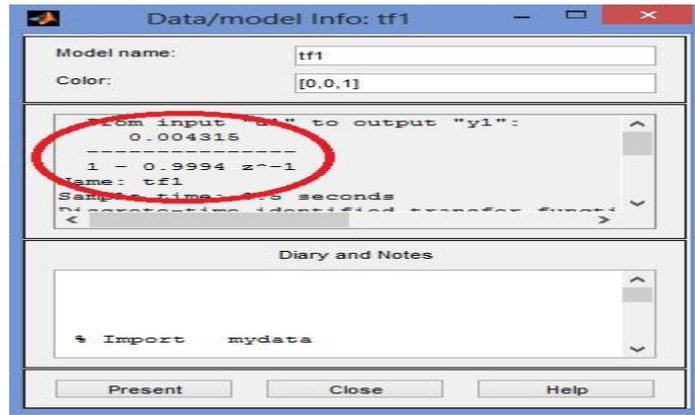


Figure 28 – Choix des types des données "Time Domain Data"

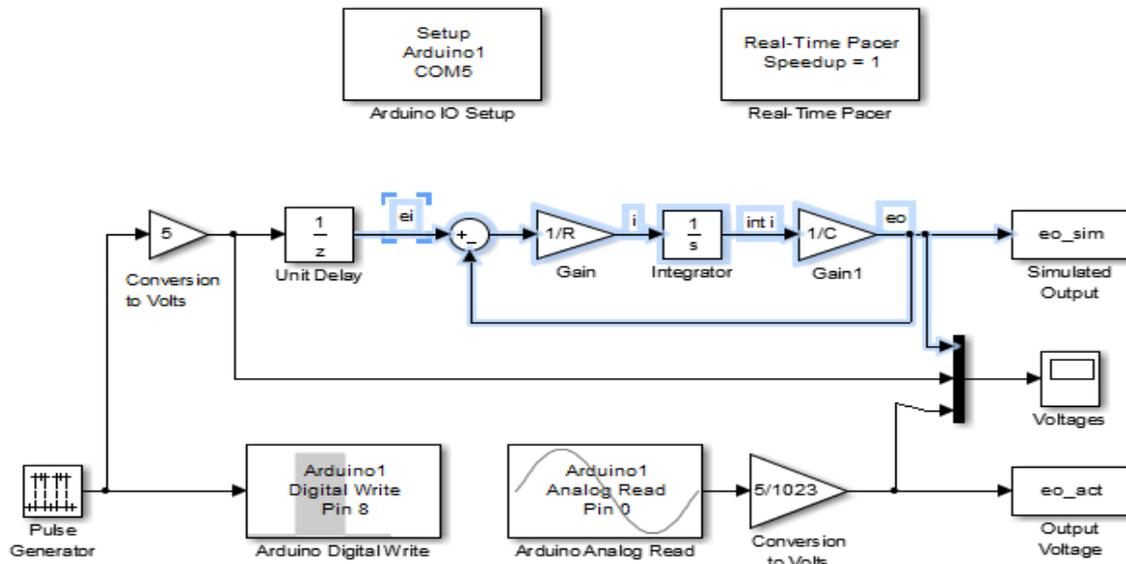
#### 7. Une fenêtre apparaît dans laquelle vous trouvez $G(z)$ .

Figure 9 – Récupération de la fonction de transfert estimée



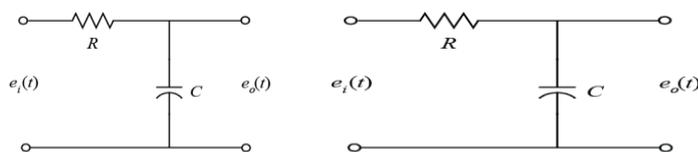
**7-Validation du model (résultats pratiques et simulés) :**

Faire le montage de simulation sous Matlab/simulink ci-dessous :



Commenter sur les résultats obtenus de simulations et réels et tirer des conclusions.

**Faire le meme travail que 5) pour le système de deuxieme ordre suivant :R1=200K et C1=04.7microH et R2=100K et C2=04.7microH.**



Régulation numérique (P,PI et PID )sous Matlab et simulink sur la carte Arduino

**I- Le but de la manipulation :**

- a- D'étudier les systèmes échantillonnés donnés ci-dessous en recherchant les **modèles continus et discrets des systèmes du premier ordre et du second ordre en BO** et de choisir convenablement la période d'échantillonnage et les paramètres des régulateurs en utilisant l'application **PID du Simulink** .Enfin ,on doit en enregistrer les réponses temporelles et d'étudier la stabilité des systèmes échantillonnés.
- b- Apprendre à programmer les systèmes de commande par l'Arduino en utilisant le langage Matlab et Simulink.

**II-Travail demandé**

**2-Faire le programme sous Sumilink :**

**2.1 Commande du système (Circuit RC ,RC-RC en cascade) en boucle fermée avec l'entrée**

**a)r(t)=5U(t) et b)r(t)= 2.5 +2.5sin(wt)) :**

L'étape de la commande du procédé est constituée de deux parties.

La première partie consiste à utiliser l'outil Matlab *PID Tuning* pour déterminer les différents paramètres de notre régulateur PID à savoir  $K_p, K_i$  et  $K_D$  en fonction de notre objectif de commande.

La deuxième partie consiste à implémenter sur Simulink puis sur la carte Arduino le correcteur *PID(z) optimal trouvé*.

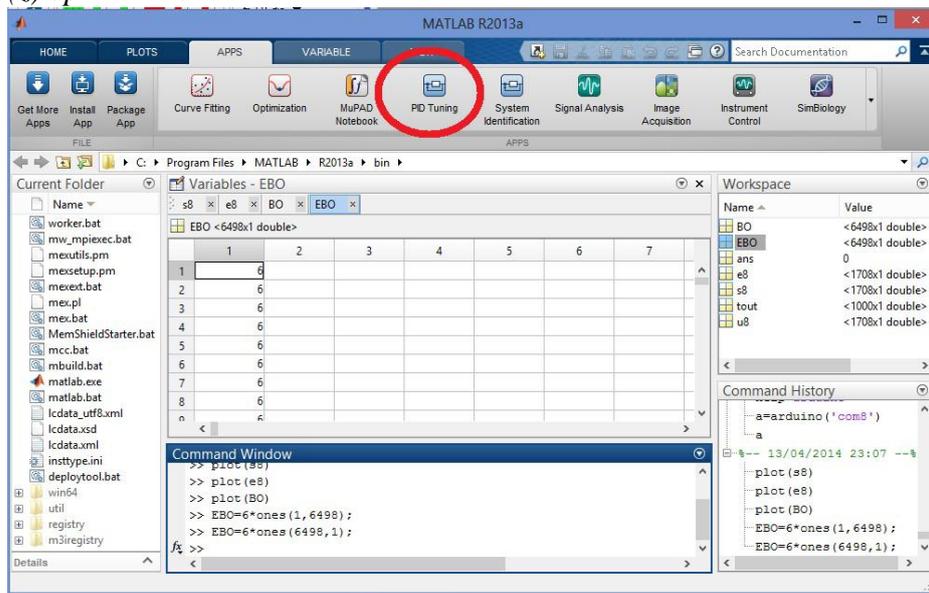


Figure 1 – Emplacement de l'outil PID tuning

**2.1 Synthèse du régulateur numérique**

1. Ouvrir l'outil *PID Tuner*

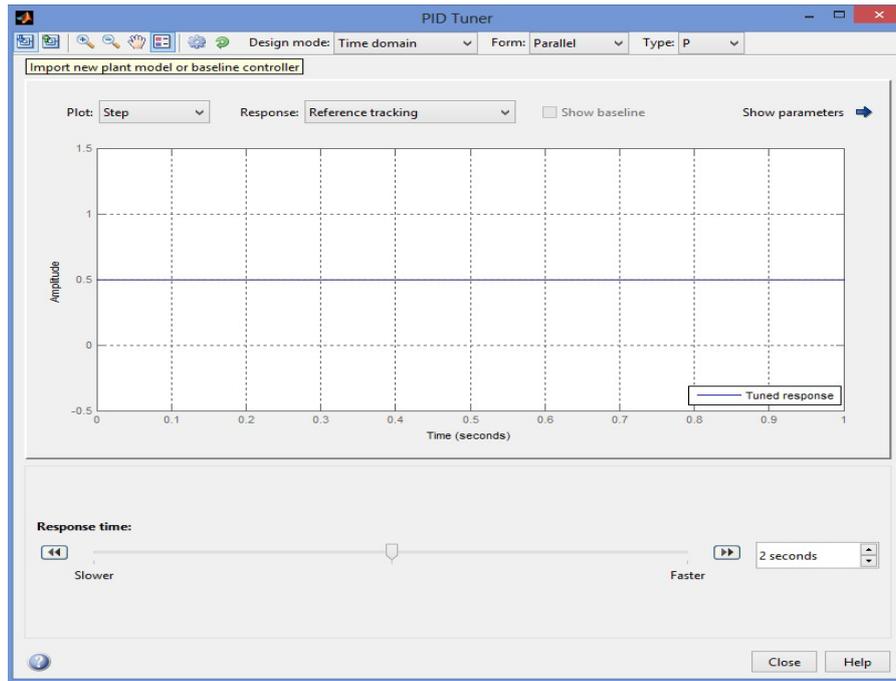


Figure 2 – Interface de l’outil ”PID tuning”

2. Cliquez sur *Import new plant*, une nouvelle fenêtre apparaît.

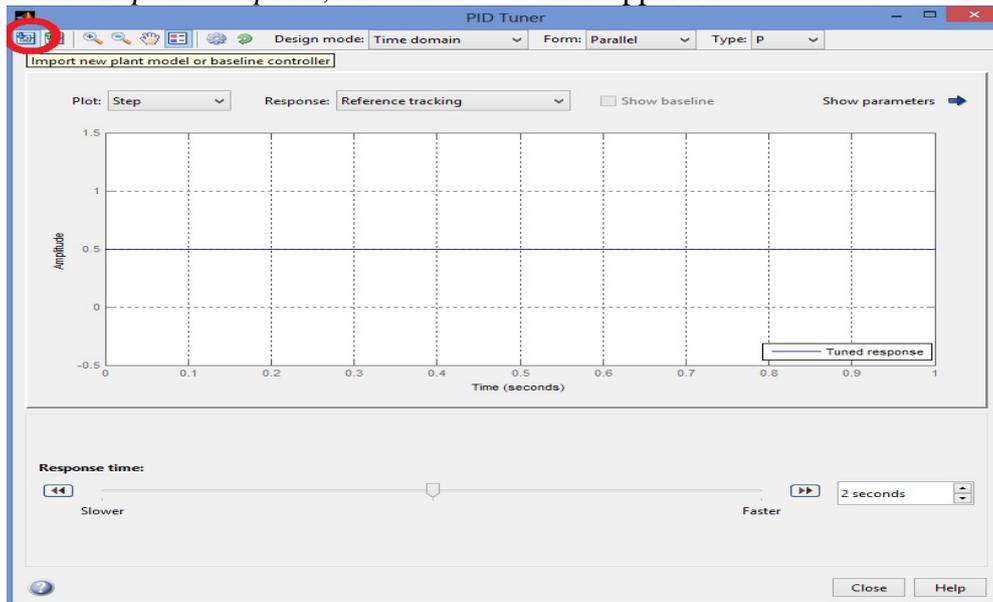


Figure 3– Importation du modèle estimé

3. Une nouvelle fenêtre apparaît dans laquelle vous allez sélectionner **tf1** ensuite cliquer sur *import* puis *close*.

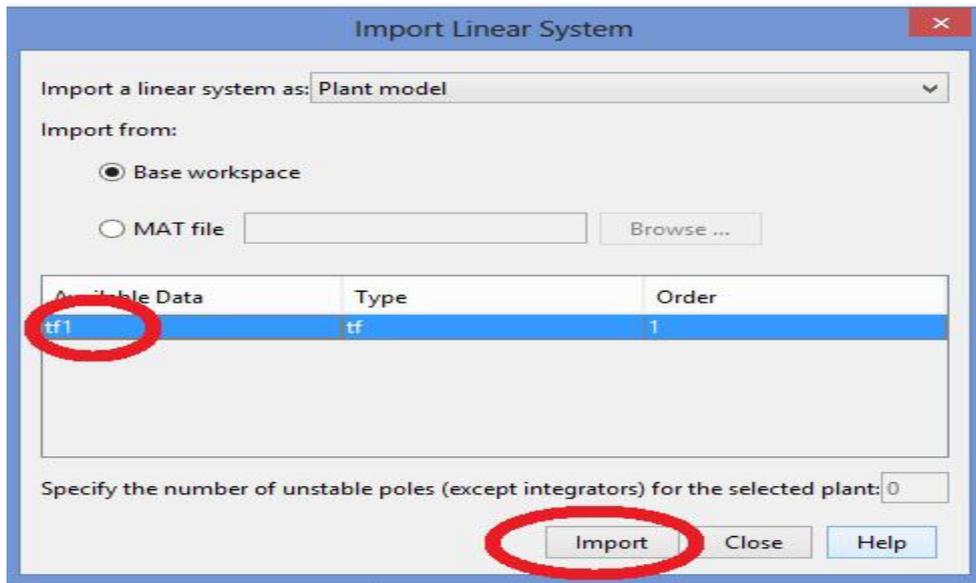


Figure 4 – Interface de l’outil ”Import Linear System”

4. Revenir à la fenêtre *PID Tuner*, vous pouvez choisir le type de régulateur à implémenter et les objectifs de la commande en boucle fermée et voir la réponse de la sortie du système.

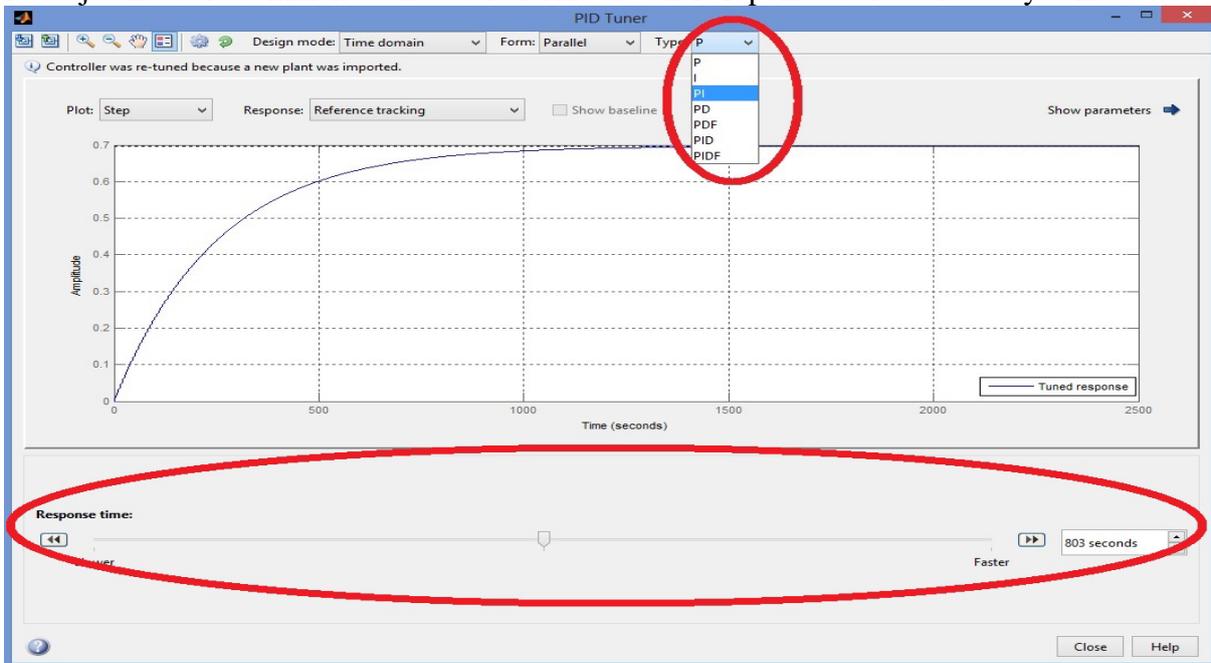


Figure 5– Choix du régulateur à implémenter

5. Cliquer sur la flèche de *show parameter* pour voir les paramètres utilisés de votre régulateur ainsi que les performances du système en boucle fermée.

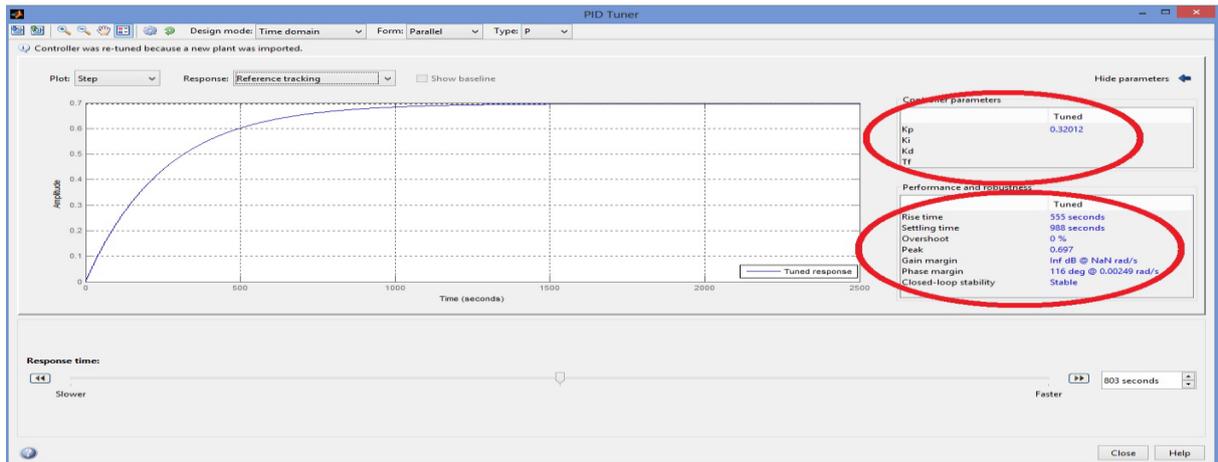


Figure 6 – Récupération des paramètres du régulateur

## 2.2 Implémentation de la commande sous Simulink

La boucle d'asservissement à implémenter sur Simulink se traduit par le schéma suivant :

Figure 7 – Synoptique de la boucle d'asservissement à implémenter

L'asservissement de notre procédé est assuré par le schéma Simulink ci-dessous qui regroupe la consigne, le comparateur, le correcteur  $PID(z)$ , le traitement de la tension issue du capteur et l'envoi de la commande PWM.

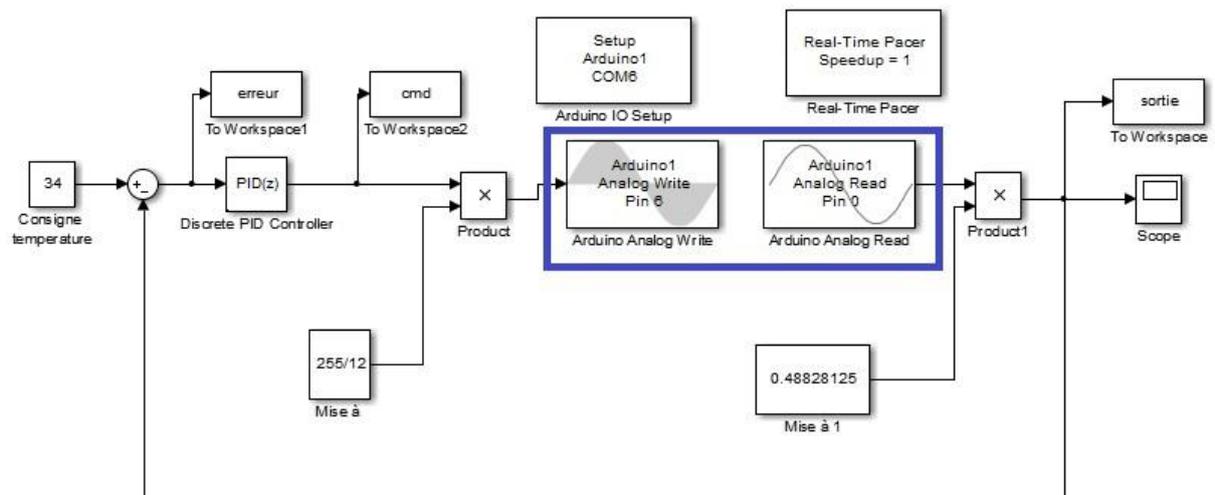


Figure 8 – Modèle Simulink d'asservissement de la tension à la borne de la capacité.

L'appui deux fois sur le bloc  $PID(z)$  permet d'introduire les paramètres  $K_p$   $K_i$   $K_d$  et de configurer le régulateur selon notre objectif de commande

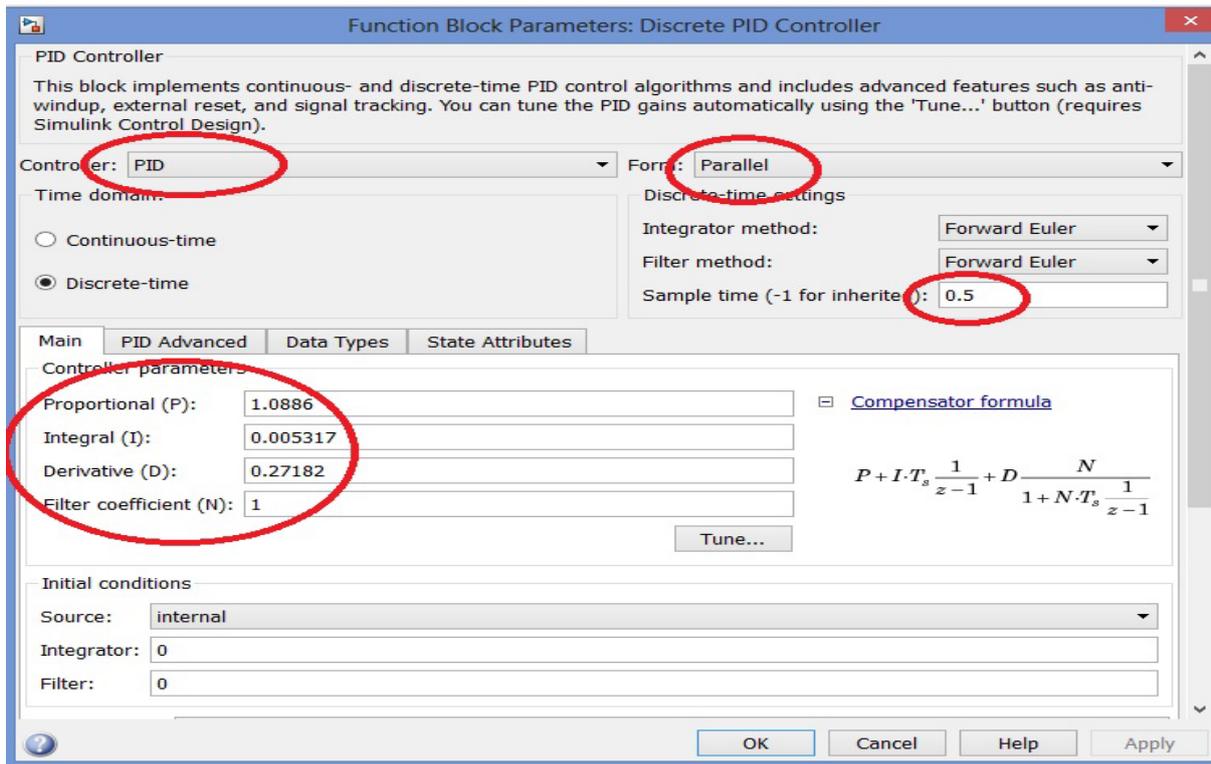


Figure 8 – Saisie des paramètres du régulateur

## 2.3 Implémentation de la commande sur la carte Arduino

Dans cette partie on va utiliser les fonctions offertes par Arduino pour envoyer (la commande) et acquérir la tension instantanée. L'implémentation du régulateur se fera directement sur la carte Arduino.

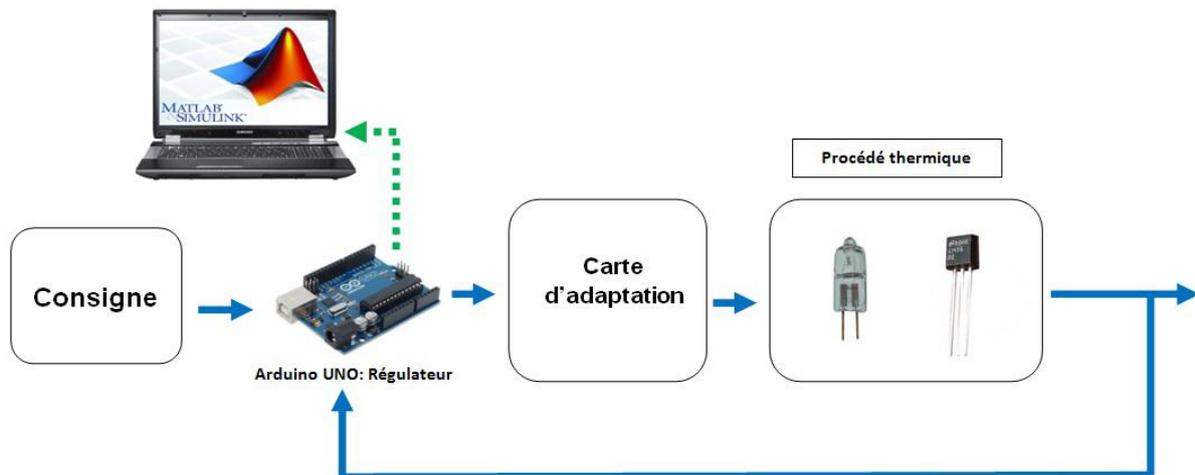


Figure 9 – Schéma synoptique de l'asservissement à implémenter

## 3-Faire le programme sous Matlab :

3-1 Ecrire le programme sous Matlab

```
clear all;clc;
clear board;
```

```
board = arduino('COM7');
pinMode(board,9,'output')
pinMode(board,8,'input')
```

```
t=0;
i=0,
tic;
```

```

while (toc<10)

    i=i+1;
    t(i)=toc;
    u=3*stepfun(toc,2);
    uu(i)=u;
h=toc/i;%%Pas d'echantillonnage...Ts
%     u=u*255/5; pour analogique sortie
    digitalWrite(board,9,u); % port PWM 9 Arduino Uno.
%     analogWrite(board,9,u);
c=analogRead(board,8)*5/1023; % port d'entrée analogique 8
%f=tf(7.025,[1 7.023]);fonction obtenue par matlab identification
%ff =f*ve;

cc(i)=c;

figure(1)
plot(t,uu,t,cc)
%pause (0.01);

end

```

```
disp('close Arduino board');
```

```

////////////////////////////////////
////////////////////////////////////

```

3.2-Determiner la FT GH(p) et GH(z) pour différentes Te=0.1 ;0.01 et 0.001.

3.3-Faire le programme de commande du système(RC et RC-RC) en BF par les différents régulateurs numériques P,PI et PID:

3.3.1 régulateurs numériques P :

```
clear all;clc;
clear board;
```

```
board = arduino('COM7');
```

```
t=0;i=0,tic;Kp=1;Ki=22;h=0.001;e0=0;tf=50
```

```
while (toc<tf)
```

```
    i=i+1;
```

```
h=toc/i; %%%Pas d'echantillonnage...Ts
```

```
    t(i)=toc;
```

```
    %r=3;
```

```
    r=1+3*stepfun(toc,20);
```

```
    c=analogRead(board,8)*5/1023;
```

```
    e=r-c;
```

```
    u1=e*Kp;
```

```
    ei=ei+h*e;%% integration de l'erreur
```

```
    u2= ei*Ki;
```

```
    u=(u1+u2)*255/5;
```

```
%%%%% u=Kp*e+Ki*(e-e0)/h;
```

```
if u> 255 %%% Limitation de la commande
```

```
    u=255
```

```
else u<0
```

```
    u=0
```

```
end
```

```
analogWrite(board,9,round(u));
```

```
rr(i)=r;  
cc(i)=c;  
e0=e;
```

```
figure(1)  
plot(t,rr,t,cc)  
pause (0.001);
```

```
end
```

```
disp('close Arduino board');
```

```
.....
```

**3.3.2-Refaire le programme en BF pour les régulateurs suivants: PI et PID.**

**Commenter sur les résultats trouvés pratiques et de simulations**