

Systèmes Asservis Échantillonnés

J.P. Chemla

Mars 2002

Jusqu'à présent, les systèmes asservis ont été étudiés dans leur état naturel : à temps continu. Nous les avons commandés à l'aide de correcteurs de même nature (avance et retard de phase, PID, retour d'état sont des correcteurs à temps continu).

L'avènement du numérique nous pousse à revoir l'automatique sous un autre angle : la commande par ordinateur. Plutôt que de limiter cette commande à une imitation de correcteurs analogiques, il est plus intéressant de tout revoir sous les aspects numériques et échantillonnés.

1 Rappels sur les systèmes numériques

1.1 représentation des signaux à temps discrets

1.1.1 Définition (simplifiée)

Un signal à temps discret est une suite $s_d(k), k \in Z$. Lorsque ce signal est l'échantillonnage d'un signal continu $s_c(t)$ à une période Δ , on a :

$$s_d(k) = s_c(k \cdot \Delta)$$

Remarque : nous ne considérerons dans la suite du cours que les signaux causaux, c'est à dire : $s_d(k) = 0 \quad \forall k < 0$.

1.2 Signaux numériques fondamentaux

Dirac centré Cette fonction est définie par :

$$\delta(k) = 0 \quad \forall k \neq 0; \quad \delta(0) = 1$$

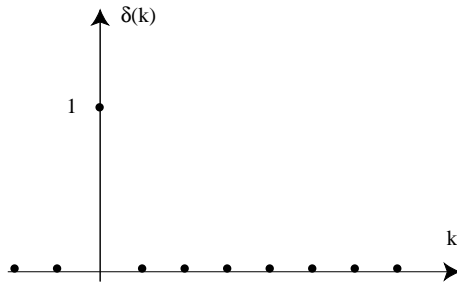


Figure 1: représentation d'un dirac

Echelon unitaire Cette fonction est définie par :

$$\Gamma(k) = 0 \quad \forall k < 0; \quad \Gamma(k) = 1 \quad \forall k \geq 0$$

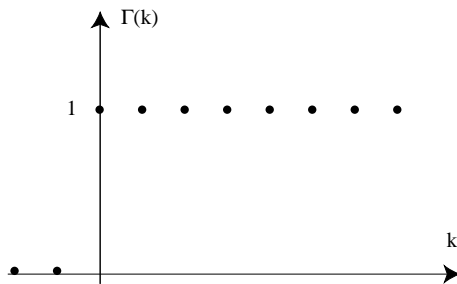


Figure 2: représentation d'un echelon

Rampe Cette fonction est définie par :

$$r(k) = 0 \quad \forall k \leq 0; \quad r(k) = k \quad \forall k > 0$$

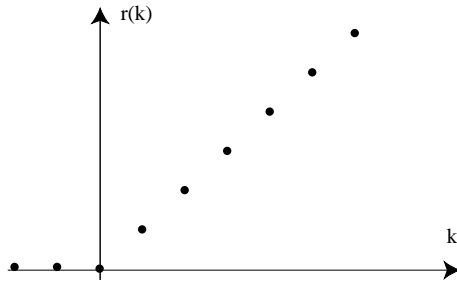


Figure 3: représentation d'une rampe

1.3 La transformée en z

1.3.1 Définition

Par définition, la transformée en z est :

$$S(z) = \sum_{k=0}^{\infty} s(k).z^{-k}$$

Exemple : si $S(z) = 1 - z^{-1} + 0.5z^{-2}$, on a

$$s(k) = 0 \quad \forall k < 0 \quad \text{ou} \quad \forall k > 2; \quad s(0) = 1; \quad s(1) = -1; \quad s(2) = 0.5$$

Autre exemple : pour $s(k) = a^k, \forall k \geq 0$,

$$S(z) = \sum_{k=0}^{\infty} \frac{a^k}{z} = \frac{1}{1 - \frac{a}{z}} = \frac{z}{z - a} \quad \text{si} \quad \left| \frac{a}{z} \right| < 1$$

Cet exemple est important à retenir car il peut permet d'avoir la transformée en z des réponses des systèmes continus du premier ordre en $s(t) = e^{-t/\tau}$. En échantillonné, cette réponse devient : $s(k) = e^{-\delta.k/\tau}$. En posant $a = e^{-\delta/\tau}$, on retrouve la transformée en z de cette réponse :

$$S(z) = \frac{z}{z - e^{-\frac{\delta}{\tau}}}$$

1.3.2 Propriétés

Soient f et g deux fonctions échantillonnées et F et G leur transformées en z . On notera $Tz()$ l'opérateur transformée en z .

Linéarité $Tz(a.f + b.g) = a.F + b.G$

Transforme la convolution en produit $Tz(f * g) = F.G$

Retard $Tz(f(k - k_0)) = z^{-k_0}.F(z)$

Avance Si $y(k) = x(k + k_0)$; on a

$$Y(z) = z^{k_0}[X(z) - x(0).z^0 - x(1).z^1 - \dots x(k_0 - 1).z^{-k_0+1}]$$

Multiplication par a^k $Tz(a^k.f(k)) = F\left(\frac{z}{a}\right)$

1.3.3 Transformée en z de fonctions classiques

$$\begin{aligned}Tz(\delta(k)) &= 1 \\Tz(\Gamma(k)) &= \frac{z}{z-1} \\Tz(r(k)) &= \frac{z}{(z-1)^2} \\Tz(a^k) &= \frac{z}{z-a}\end{aligned}$$

Pour plus de transformées, voir table des transformées usuelles sur feuille à part.

1.3.4 Théorèmes des valeurs initiales et finales

- $s(0) = \lim_{z \rightarrow +\infty} S(z)$
- $\lim_{k \rightarrow +\infty} s(k) = \lim_{z \rightarrow 1} (z-1).S(z)$

1.3.5 Recherche de l'original

Le calcul de la transformée inverse $s(k)$ d'une fonction en $S(z)$ peut se faire selon deux méthodes :

1. Par décomposition en éléments simples de $\frac{S(z)}{z}$ puis en utilisant la table de transformées. Exemple :

$$\begin{aligned}S(z) &= \frac{2z}{(z-1)(z-0,5)} \\ \frac{S(z)}{z} &= \frac{4}{z-1} - \frac{4}{z-0,5} \\ s(k) &= 4 - 4.(0,5)^k\end{aligned}$$

2. On peut aussi calculer les premières valeurs de $s(k)$ en effectuant une division de polynômes :

$$S(z) = \frac{z}{z-2} \quad (\text{on sait que } s(k) = 2^k)$$

La division polynomiale donne :

$$\begin{array}{r|l} z & z-2 \\ -z & +2 \\ \hline & 2 \\ & -2 & +4z^{-1} \\ \hline & & 4z^{-1} \end{array}$$

On retrouve les premières valeurs de $s(k)$: $s(0) = 1$; $s(1) = 2$; $s(2) = 4$.

2 Fonction de transfert en z d'un système numérique

2.1 Définition, ordre et réponse d'un système

Un système numérique est défini par une équation de récurrence entre son entrée $e(k)$ et sa sortie $s(k)$. Cette équation est de la forme :

$$s(k+n) + a_{n-1}s(k+n-1) + \dots + a_1s(k+1) + a_0s(k) = b_0e(k) + b_1e(k+1) + \dots + b_me(k+m)$$

$$s(k+n) + \sum_{i=0}^n a_i s(k+i) = \sum_{j=0}^m b_j e(k+j)$$

Comme le système est causal, la sortie à l'instant $k+n$ ne peut pas dépendre de l'entrée aux instants d'après. On en déduit que $m \leq n$.

On remarque que l'on peut connaître de proche en proche toutes les valeurs de la sortie $s(k)$ si l'on connaît l'entrée $e(k)$ et les conditions initiales.

On appelle fonction de transfert $G(z)$ du système le rapport entre la transformée en z , $S(z)$, de la sortie et $E(z)$, celle de l'entrée.

On obtient :

$$G(z) = \frac{S(z)}{E(z)} = \frac{b_0 + b_1.z + \dots + b_m.z^m}{z^n + a_{n-1}.z^{n-1} + \dots + a_1.z + a_0}$$

On appelle l'ordre d'un système l'ordre de l'équation de récurrence. C'est la différence maximale d'indice dans les termes de sortie. C'est aussi le degré du dénominateur de la fonction de transfert.

Comme pour les fonctions de transfert continues, cette fonction de transfert est une fonction rationnelle dont le degré du dénominateur est supérieur au degré du numérateur.

On peut connaître la réponse d'un système à une entrée donnée en formant la transformée en z inverse de $T(z).E(z)$. Rappel : la transformée inverse d'une fonction de transfert seule n'a aucune signification.

Exemple : Soit un système défini par son équation de récurrence :

$$s(k+1) - 0,5.s(k) = e(k)$$

On connaît les conditions initiales : $s(0) = 0$. L'entrée est un échelon unitaire. Pour calculer la sortie $s(k) \forall k$, on peut utiliser l'équation de récurrence de proche en proche ou utiliser la transformée en z .

De proche en proche : $s(1) = 0,5.s(0) + e(0) = 1$

$$s(2) = 0,5.s(1) + e(1) = 1,5 \quad s(3) = 0,5.s(2) + e(2) = 1,75 \dots$$

Avec la transformée en z : La fonction de transfert est :

$$G(z) = \frac{1}{z - 0,5} \Rightarrow S(z) = \frac{z}{(z-1)(z-0,5)}$$

En décomposant en éléments simples $\frac{S(z)}{z}$, on trouve

$$S(z) = \frac{2.z}{z-1} - \frac{2.z}{z-0,5} \Rightarrow s(k) = 2 - 2.(0,5)^k$$

2.2 Retards purs

Les systèmes peuvent présenter un retard pur. Ils se manifestent dans l'équation de récurrence par l'apparition de termes en $e(k-r)$ où r est le retard en nombre de pas d'échantillonnage.

Exemple 1 Un système définit par :

$$s(k+2) + a_1.s(k+1) + a_2.s(k) = b.e(k-5)$$

avec des conditions initiales nulles ne donnerait un résultat ($s(k) > 0$) que pour $k \geq 5$.

Exemple 2 Un système définit par :

$$s(k+3) + 3.s(k+2) + 2.s(k+1) + s(k) = e(k+1) + 2.e(k) + e(k-1) + e(k-2)$$

est d'ordre 3 et présente un retard de 2 pas d'échantillonnage (par la présence du terme $e(k-2)$).

Dans la fonction de transfert, le retard se manifeste par la présence d'un terme en z^{-r} en facteur. Dans l'exemple précédent, la fonction de transfert est :

$$G(z) = z^{-2} \cdot \frac{z^3 + 2z^2 + z + 1}{z^3 + 3z^2 + 2z + 1}$$

Remarque Un zéro au numérateur ($z - z_0$) est une avance partielle. Si $z_0 = 0$ (zéro nul), il y a une avance d'un pas entier.

2.3 Intégration numérique

En numérique, l'intégration est une sommation. Son équation de récurrence est donc:

$$s(k+1) = s(k) + e(k) \quad \Rightarrow \quad I(z) = \frac{1}{z-1}$$

Nous verrons dans le chapitre suivant que cette fonction de transfert correspond à un facteur près à l'échantillonnage d'un intégrateur continu.

2.4 Forme standard

De même qu'en continu, on peut écrire une fonction de transfert numérique en mettant en évidence les points suivants :

- les intégrateurs,
- le gain
- l'éventuel retard

Cette forme est la suivante :

$$G(z) = \frac{K}{(z-1)^m} \cdot z^{-r} \cdot \frac{N(z)}{D(z)}$$

où le gain K est :

$$K = \lim_{z \rightarrow 1} (z-1)^m \cdot G(z) \quad \Rightarrow \quad \lim_{z \rightarrow 1} \frac{N(z)}{D(z)} = 1$$

Exemple : Mettre sous forme standard la fonction de transfert du système dont l'équation de récurrence est :

$$2.s(k) = s(k-1) + 4.s(k-2) - 3.s(k-3) + 3.e(k) + 2.e(k-1) + 5.e(k-2)$$

Réponse : gain : 2; ordre 3; type : 2; avance d'un pas

$$G(z) = \frac{2}{(1-z)^2} \cdot z \cdot \frac{3z^2 + 2z + 5}{4z + 6}$$

2.5 Pôles et stabilité

Comme en continu, les pôles de la fonction de transfert caractérisent la dynamique du système.

2.5.1 Condition de stabilité

Un système numérique est stable si et seulement si tous les pôles de ce système sont de module inférieur à 1 ; c'est à dire si tous les pôles sont à l'intérieur du cercle de rayon 1 et de centre $(0,0)$ dans le plan complexe.

Dans la figure 4, on trouvera quelques exemples de comportements de systèmes du premier ordre en fonction du lieu du pôle. Un fichier *polenz.sg* téléchargeable à l'adresse : <http://homepage.mac.com/jpchemla>, dans le répertoire *Auto echantillonnee* vous permet d'expérimenter la réponse d'un système d'ordre 1 ou 2 en fonction du placement du ou des pôles. Ce fichier s'ouvre avec l'application *Sysquake Le* téléchargeable à l'adresse : <http://www.calerga.com/>.

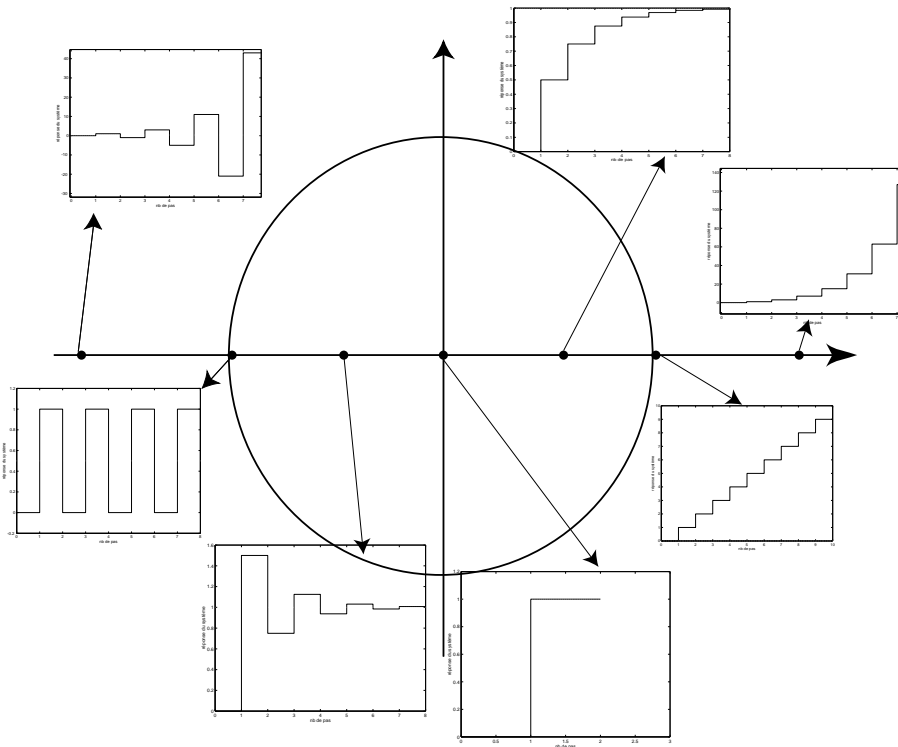


Figure 4: sortie d'un système de premier ordre en fonction de son pôle en z

2.5.2 Critère de Jury

Il n'est pas toujours facile de calculer les pôles d'une fonction de transfert surtout quand celle-ci se présente sous forme développée. Un critère permet de déterminer la stabilité d'un système à partir des coefficients du dénominateur.

Soit une fonction de transfert dont le dénominateur s'écrit :

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

Le critère nécessite de former le tableau 1.

	z^0	z^1	z^n
1	a_0	a_1	a_n
2	a_n	a_{n-1}	a_0
3	b_0	b_1	b_{n-1}	0
4	b_{n-1}	b_{n-2}	b_0	0
5	c_0	c_1	...	c_{n-2}	0	0
6	c_{n-2}	c_{n-3}	...	c_0	0	0

Table 1: tableau de Jury

$$\text{où } b_j = a_0 \cdot a_j - a_n \cdot a_{n-j}; \quad c_j = b_0 \cdot b_j - b_{n-1} \cdot b_{n-j-1}.$$

Le critère de Jury est : Le système est stable si $D(z)$ vérifie les conditions suivantes :

1. $D(1) > 0$,
2. $D(-1) > 0$ pour n pair, $D(-1) < 0$ sinon,
3. $|a_0| < a_n$; $|b_0| > |b_{n-1}|$; $|c_0| > |c_{n-2}|$; ...

Exemple : Donner la stabilité des systèmes définis par leur fonction de transfert :

$$G_1(z) = \frac{z - 0,5}{z^2 - 0,9z + 0,8}; \quad G_2(z) = \frac{2 \cdot (z + 2)}{z^2 + 3,5z - 2}$$

2.6 Précision

Comme en continu, la précision du système en boucle fermée dépend du gain et du type (nombre d'intégrateurs) du système en boucle ouverte (voir figure 5).

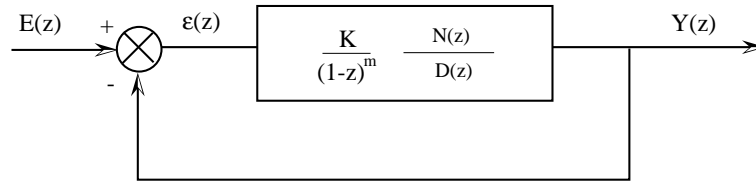


Figure 5: Asservissement numérique

Le calcul de l'erreur donne :

$$\epsilon(z) = \frac{E(z)}{1 + K \cdot \frac{1}{(z-1)^m} \cdot \frac{N(z)}{D(z)}}$$

En étudiant $\lim_{k \rightarrow \infty} (\epsilon(k)) = \lim_{z \rightarrow 1} (z-1)\epsilon(z)$, on trouve les résultats suivants :

Entrée	Echelon $E_0\Gamma(k)$	Rampe ak
m=0	$\frac{E_0}{1+K}$	∞
m=1	0	$\frac{a}{k}$
m=2	0	0

3 Fonction de transfert d'un système continu échantillonné

Les systèmes numériques n'existent que sous forme de modèle dans un ordinateur. Notre objectif est de faire de la commande, par un processeur (numérique), d'un système réel continu par nature. Comme nous l'avons dit au début du cours, on peut soit voir le correcteur numérique comme imitation d'un correcteur continu, soit voir le système réel continu comme un système numérique, c'est à dire tel qu'il sera vu par le calculateur.

3.1 Principe général

La figure 6 donne le schéma de principe de la correction numérique.

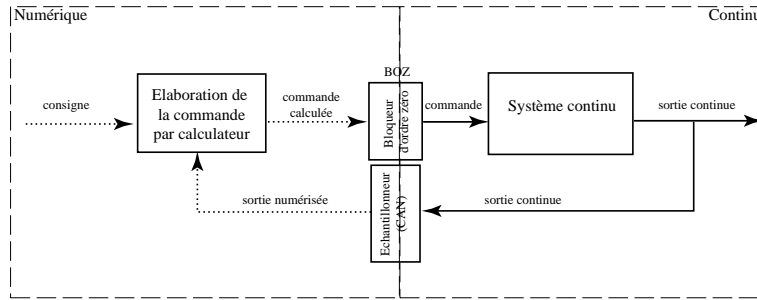


Figure 6: Asservissement numérique

Vu du calculateur, le système continu précédé du Bloqueur d'Ordre Zéro (BOZ) et suivi de l'échantillonneur est considéré comme un système numérique. Ce système continu, défini par sa fonction de transfert en Laplace (en p) est donc vu par le calculateur comme un système numérique ayant une fonction de transfert en z . L'équivalence est montrée dans la figure 7.

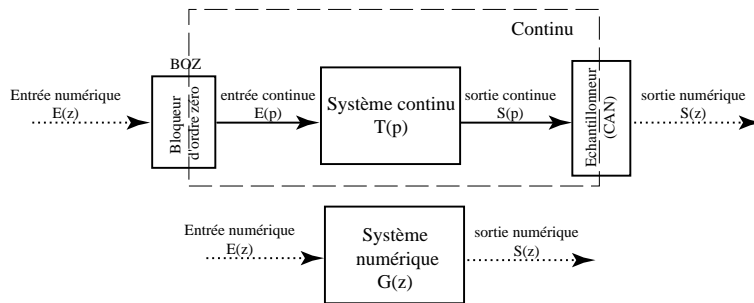


Figure 7: Equivalent numérique d'un système continu muni d'un BOZ et échantillonné

On attire l'attention que cette équivalence n'est pas entre $T(p)$ et $G(z)$ qui sont de nature différente mais entre l'ensemble $T(p)$, bloqueur et échantillonneur, et $G(z)$. On peut montrer que :

$$G(z) = Tz[B(p).T(p)] = (1 - z^{-1}).Tz \left[\frac{T(p)}{p} \right]$$

où $B(p)$ est la fonction de transfert du BOZ et $Tz[F(p)]$ désigne la transformée en z de la transformée de Laplace inverse de $F(p)$ échantillonnée. Un tableau des fonctions classiques fourni en annexe permet de passer rapidement de $F(p)$ à $Tz[F(p)]$.

Exemple : Soit un système continu du premier ordre défini par sa fonction de transfert

$$T(p) = \frac{K}{1 + \tau p}$$

Ce système, muni d'un BOZ et d'un échantillonneur sera considéré comme un système numérique dont la fonction de transfert en z est :

$$G(z) = (1 - z^{-1}) \cdot Tz \left[\frac{K}{p(1 + \tau p)} \right]$$

Pour calculer la $Tz[\dots]$, nous pouvons commencer par calculer sa transformée de Laplace inverse :

$$s(t) = TL^{-1} \left[\frac{K}{p(1 + \tau p)} \right] = K \cdot (1 - e^{-\frac{t}{\tau}})$$

Une fois échantillonné à la période Δ , nous avons :

$$s(\Delta k) = K \cdot (1 - e^{-\frac{k \cdot \Delta}{\tau}})$$

En posant $a = e^{-\frac{\Delta}{\tau}}$, on a la transformée en z cherchée :

$$Tz \left[\frac{K}{p(1 + \tau p)} \right] = S(z) = K \cdot \frac{z(1 - a)}{(z - 1)(z - a)}$$

On en déduit la fonction de transfert en z d'un système du premier ordre continu muni d'un BOZ et d'un échantillonneur :

$$G(z) = \frac{K(1 - a)}{z - a} \quad \text{avec} \quad a = e^{-\frac{\Delta}{\tau}}$$

Remarque importante : La fonction de transfert d'un système continu échantillonné dépend de la période d'échantillonnage Δ .

3.2 PID numérique

L'intégration numérique est une sommation, la dérivation numérique est une soustraction. Dans ce paragraphe, nous allons en fait calculer l'équivalent numérique des correcteurs connus en analogique : l'intégrateur, le dérivateur et pour finir le PID.

INTÉGRATEUR La fonction de transfert en continu de l'intégrateur est :

$$\frac{S(p)}{E(p)} = I(p) = \frac{1}{T_i p}$$

Si on numérise ce système (à l'aide d'un BOZ et d'un échantillonneur), on obtient la fonction de transfert en z suivante :

$$\begin{aligned}\frac{S(z)}{E(z)} = I(z) &= (1 - z^{-1})Tz \left[\frac{1}{T_i p^2} \right] \\ &= \frac{\Delta}{T_i} \cdot \frac{1}{z - 1} \\ \Rightarrow s(k + 1) &= s(k) + \frac{\Delta}{T_i} e(k)\end{aligned}$$

Pour créer un intégrateur numérique ayant le même comportement qu'un intégrateur analogique de constante T_i , il suffit d'accumuler les valeurs prises par l'entrée ($e(k)$) pondérées par un facteur Δ/T_i .

DÉRIVATEUR La fonction de transfert en continu d'un dérivateur parfait est :

$$\frac{S(p)}{E(p)} = D(p) = T_d p$$

Ce dérivateur analogique n'existe pas (il n'est pas causal) puisque que sa fonction de transfert a un numérateur de degré plus haut que le dénominateur (ici constant). On peut néanmoins trouver le dérivateur numérique équivalent.

$$\begin{aligned}s(k) &= \frac{T_d}{\Delta} (e(k) - e(k - 1)) \\ \Rightarrow D(z) &= \frac{T_d}{\Delta} \cdot \frac{z - 1}{z}\end{aligned}$$

DÉRIVATEUR FILTRÉ Si l'on veut l'équivalent numérique d'un dérivateur réalisable en analogique, on part du dérivateur filtré dont la fonction de transfert est :

$$D(p) = \frac{T_d p}{1 + \frac{T_d}{N} p}$$

Si on numérise ce système (à l'aide d'un BOZ et d'un échantillonneur), on obtiendrait la fonction de transfert en z suivante :

$$\begin{aligned}\frac{S(z)}{E(z)} = D(z) &= (1 - z^{-1})Tz \left[\frac{T_d}{1 + \frac{T_d}{N} p} \right] \\ &= N \cdot \frac{z - 1}{z - z_0} \quad \text{avec : } z_0 = e^{-\frac{N\Delta}{T_d}} \\ \Rightarrow s(k + 1) &= z_0 s(k) + N \cdot (e(k + 1) - e(k))\end{aligned}$$

Sur la figure 8, on peut comparer les sorties d'un dérivateur et d'un dérivateur filtré en réponse à un échelon.

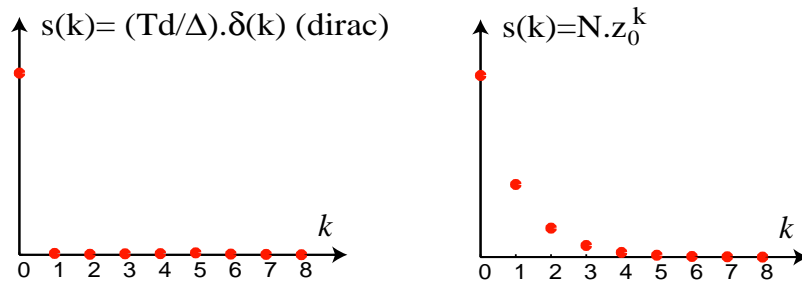


Figure 8: sorties comparées d'un dérivateur (à gauche) et d'un dérivateur filtré (à droite) en réponse à un échelon

PID NUMÉRIQUE Le correcteur le plus courant, en analogique est le correcteur PID. Ce dernier est souvent utilisé dans l'industrie et mis en oeuvre par des PID numérique ou des cartes à contrôleur ou des automates. Toutes ces solutions sont en fait des calculateurs numériques qui vont simuler un correcteur analogique. Comment sont réalisés ces PID numériques ? c'est ce que nous décrivons dans ce paragraphe.

La fonction de transfert d'un PID analogique (avec dérivateur filtré) est :

$$C(p) = K \left(1 + \frac{1}{T_i p} + \frac{T_d p}{1 + \frac{T_d}{N} p} \right)$$

Son équivalent en numérique est (d'après les résultats précédents) :

$$C(z) = K \left(1 + \frac{\Delta}{T_i} \cdot \frac{1}{z-1} + N \cdot \frac{z-1}{z-z_0} \right)$$

Les paramètres pratiques de ce correcteur sont :

$z_0 = e^{-\frac{N\Delta}{T_d}}$ est un pôle adoucissant de l'ordre de 0,2 à 0,4 qui s'amortit en quelques coups.

N règle l'effet dérivé

T_i règle l'effet intégral

K règle l'amplitude de la correction (gain)

4 Commande par ordinateur

4.1 Principe et intérêt

Dans ce chapitre, le ordinateur va calculer une commande en fonction de la consigne et de la sortie numérisée du système. Le principe est de presque

tout faire en numérique. Le système continu sera vu, lui aussi, comme un système numérique.

On distingue 4 étapes dans la mise en place d'un asservissement numérique.

1. Le choix du pas d'échantillonnage Δ sur lequel seront réglés : le calculateur, le BOZ et l'échantillonneur. Une fois choisie, on peut calculer la fonction de transfert du processus numérisé.
2. Le choix du modèle numérique (fonction de transfert en z) à atteindre en boucle fermée après correction.
3. On en déduit le correcteur nécessaire.
4. On programme le calculateur pour qu'il réalise la correction calculée précédemment afin qu'il élabore la commande.

Si, par nature, un correcteur numérique est plus lent (car cadencé par la période d'échantillonnage) qu'un correcteur analogique, ses avantages sont les suivants :

- certaines corrections numériques sont impossible à réaliser en analogique,
- par sa capacité à mémoriser des signaux, la correction de systèmes à retard est plus aisée,
- la flexibilité de la programmation permet de réaliser des corrections très fines, facilement réglables voire auto-ajustables (réglage automatique du correcteur).

4.2 Choix du pas d'échantillonnage

Le processus n'est observé et la commande ne peut changer qu'aux instants $\Delta.k$. Le choix du pas Δ est important car

- Si Δ est trop petit, le calculateur corrigera sans arrêt à tout petits coups;
- Si Δ est trop grand, le calculateur risque de perdre des informations importantes mais trop rapides ou même ne plus pouvoir commander car les erreurs (sortie-consigne) seront trop importantes.

En pratique, on choisira une fréquence d'échantillonnage f_e en fonction de la fréquence la plus haute que l'on souhaite observer f_h (ce qui est directement

lié à la fréquence de coupure du système que l'on souhaite commander. On choisit :

$$5f_h \frac{1}{\Delta} = f_e < 25f_h$$

Pour un premier ordre : $\frac{\tau}{4} < \Delta < \tau$ Pour un second ordre : $\frac{0,25}{\omega_n} < \Delta < \frac{1,25}{\omega_n}$

Pour visualiser l'influence de la période d'échantillonnage sur un asservissement, vous pouvez télécharger le fichier *correch.sq* sur <http://homepage.mac.com/jpchemla/>. (Ce fichier s'ouvre avec *Sysquake LE* téléchargeable sur <http://www.calerga.com>). Ce fichier visualise la sortie en BO et BF en continu (courbes bleues) et en échantillonné (courbes en échelles) d'un système du premier ordre. Le gain, la période d'échantillonnage et la constante de temps du système sont modifiables.

4.3 Structures possibles

4.3.1 Correcteurs RST

La commande $u(k)$ tient compte

- des valeurs précédentes $u(k-1), u(k-2), \dots$ de la commande envoyée afin de doser progressivement les efforts.
- des résultats obtenus en sortie : $y(k), y(k-1), \dots$
- de la trajectoire de consigne : $y_c(k), y_c(k-1), \dots$

La commande peut donc s'écrire, dans le cas général :

$$u(k) = \sum_i r_i u(k-i) + \sum_j t_j y_c(k-j) + \sum_l s_l y(k-l)$$

Cette commande, souvent appelée RST peut se mettre concrètement en place suivant la figure 9.

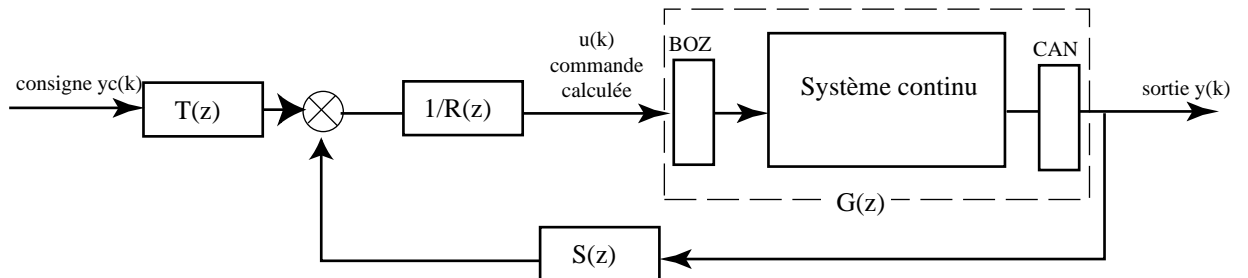


Figure 9: Structure RST d'asservissement numérique

Déterminer le correcteur, c'est déterminer les polynômes $T(z)$, $R(z)$ et $S(z)$.

4.3.2 structure classique

Le plus souvent, on a $T(z) = S(z)$ et la structure du correcteur devient plus classique (voir figure 10). Le correcteur $C(z)$ est introduit entre le calcul de l'erreur ($y(k) - y_c(k)$) et la commande $u(k)$. Par rapport à la structure précédente, on a $C(z) = \frac{S(z)}{R(z)}$.

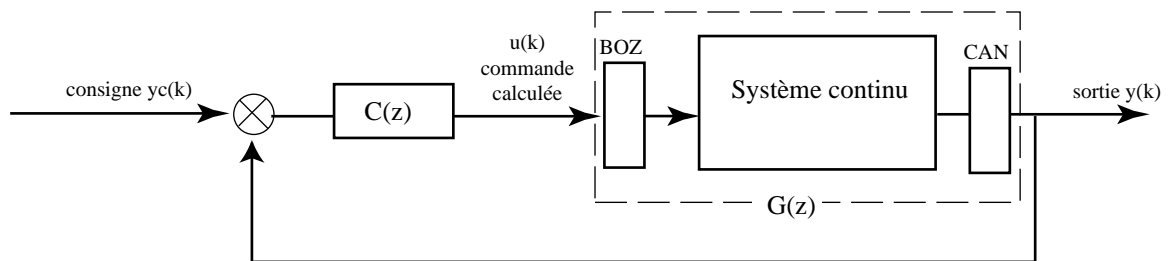


Figure 10: Structure classique d'asservissement numérique

4.3.3 Démarche en boucle ouverte

La démarche est d'obtenir une fonction de transfert en boucle ouverte (produit de $G(z)$ et de $C(z)$) qui ait les propriétés requises pour avoir un bon asservissement en boucle fermée. Ces propriétés sont :

- $C(z).G(z)$ doit avoir les intégrateurs nécessaires (1 si on veut une erreur en position nulle, 2 pour avoir une erreur de vitesse nulle, ...)
- le gain est réglé pour contrôler l'amortissement
- $C(z)$ doit avoir les zéros nécessaires pour réduire les temps de réponse.

4.3.4 Démarche en boucle fermée

On se fixe la fonction de transfert à obtenir en boucle fermée ($H_m(z) = \frac{B_m}{A_m}$). On en déduit le correcteur à mettre pour y parvenir.

Le plus souvent, on choisit, comme fonction de transfert en BF on choisit l'un des deux modèles suivant :

1. une fonction de transfert de gain unité qui permet d'annuler l'erreur en quelques (n) coups :

$$H(z) = \frac{1}{1 + a_{n-1}z^{-1} + \dots + a_0z^{-n}} [1 + a_{n-1}z^{-1} + \dots + a_0z^{-n}]$$

2. un second ordre d'amortissement réglable et de gain unité.

4.3.5 Contraintes sur le correcteur

- Le correcteur doit être réalisable. En particulier, le degré de son dénominateur doit être supérieur au degré de son numérateur,
- le système doit être stable en boucle fermée,
- le correcteur ne doit pas compenser un zéro du système à corriger de module supérieur à 1 par un pôle instable. Voir le fichier textitcorrP1.sq en plaçant le zéro du système à corriger en dehors du cercle unité. Ce fichier est sur le site habituel (<http://homepage.mac.com/jpchemla/> et s'ouvre avec le logiciel Sysquake LE téléchargeable sur www.calerga.com).

4.3.6 Les trois types de processus à corriger

Processus de type P1 ce sont les plus faciles à corriger car ils ne présentent pas de zéro de module > 1 (tous les zéros sont stables) et ils vérifient :

$$\deg(\text{denominateur}) - 1 \leq \deg(\text{numérateur}) \leq \deg(\text{denominateur})$$

Processus de type P2 ils n'ont que des zéros stables et vérifient

$$\deg(\text{denominateur}) - \deg(\text{numérateur}) \geq 2$$

Processus de type P3 ce sont les processus qui ont des zéros instables (de module > 1)

4.4 Régulation de processus de type P1

4.4.1 La consigne est en échelon

Soit $G(z)$ la fonction de transfert du système à commander. On adoptera, par exemple, une démarche en boucle ouverte. Comme la consigne est en

échelon, la fonction de transfert en BO doit avoir un intégrateur pour obtenir une erreur statique nulle. On choisit le correcteur $C(z)$ tel que :

$$G(z).C(z) = \frac{K}{z-1} \quad \Rightarrow \quad C(z) = \frac{K}{z-1} \cdot \frac{1}{G(z)}$$

La fonction de transfert en boucle fermée devient :

$$H(z) = \frac{K}{z - (1 - K)}$$

Le gain est de 1 (pas d'erreur statique), le système est stable pour $0 < K < 2$.

Pour $K = 1$, la fonction de transfert en BF est $H(z) = z^{-1}$ c'est à dire que la sortie $y(k)$ du système vérifie $y(k) = e(k-1)$. La sortie du système est l'entrée retardée d'un pas d'échantillonnage ! On peut vérifier l'effet de K sur le correcteur (commande et sortie) en utilisant le fichier *corrP1* sur le serveur habituel.

4.4.2 La consigne est en rampe

Pour avoir une erreur statique nulle, il faut 2 intégrateurs dans la boucle ouverte. La première idée serait de choisir le correcteur $C(z)$ tel que :

$$C(z).G(z) = \frac{K}{(z-1)^2} \Rightarrow H(z) = \frac{K}{z^2 - 2z + K + 1}$$

Où $H(z)$ serait la fonction de transfert en BF. Un simple calcul des pôles montre que ce système serait instable. Pour améliorer la stabilité, on choisit d'introduire un zéro au numérateur, ce qui est équivalent à introduire une avance partielle.

$$C(z).G(z) = \frac{K(z-z_0)}{(z-1)^2}$$

Le calcul de la fonction de transfert en BF donne cette fois :

$$H(z) = \frac{K(z-z_0)}{z^2 + (K-2)z + 1 - K.z_0} \quad \text{stable pour} \quad 0 < z_0 < 1 \quad 0 < K < \frac{4}{1+z_0}$$

Un cas particulier se présente pour : $K = 2$ et $z_0 = 0.5$. Pour ces valeurs, on a

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

C'est aussi un cas de *réponse pile* puisque par la forme de $H(z)$, on a forcément une réponse finie en 2 pas d'échantillonnage. Pour s'en convaincre, on peut calculer la sortie à une entrée en rampe unité : $e(k) = k$. La transformée inverse de $S(z) = H(z).E(z)$ donne :

$$s(k) = 2.e(k - 1) - e(k - 2)$$

Ce qui donne pour $s(k)$:

$$s(0) = 0; \quad s(1) = 0; \quad s(k) = 2.(k - 1) - (k - 2) = k \quad \forall k \geq 2$$

la sortie rejoint l'entrée pour $k = 2$.

4.5 Régulation de processus de type P2

Le principe reste identique à la régulation précédente mais la fonction de transfert en BF à obtenir doit comporter les retards nécessaires à la faisabilité du correcteur.

Exemple

$$G(z) = \frac{1}{z^3 - 0,6z^2 - 0,15z + 0,01}$$

Si on désire obtenir une fonction de transfert $H(z)$ du second ordre en BF, celle ci doit comporter un retard d'un pas pour que le correcteur soit réalisable.

$$H(z) = \frac{1 + a_1 + a_0}{z^2 + a_1z + a_0} . z^{-1}$$

cette fonction de transfert est de second ordre, a un retard 1 et est de gain 1 (pas d'erreur statique). D'où le correcteur :

$$C(z) = \frac{1}{G(z)} \cdot \frac{H(z)}{1 - H(z)}$$

est tel que le degré de son numérateur est égal à celui de son dénominateur.

4.6 Régulation de processus de type P1 ou P2 avec retard

Certains systèmes présentent des retards intrinsèques entre l'entrée et la sortie. Leur fonction de transfert peut s'écrire :

$$G(z) = F(z).z^{-n}$$

où $F(z)$ est une fonction de transfert sans retard et n est le nombre de pas d'échantillonnage qui forme le retard. (on suppose ici que le retard est un nombre entier de pas d'échantillonnage). La méthode dans ce cas est de calculer le correcteur nécessaire sur le système sans retard, puis on en déduit le correcteur à appliquer réellement.

On adopte par exemple une démarche en BF. Soit $H(z)$ la fonction de transfert que l'on souhaite obtenir après correction. $H(z)$ présente naturellement le même retard que le système en BO : $H(z) = H'(z).z^{-n}$. On calcule le correcteur $C'(z)$ qui permettrait d'obtenir $H'(z)$ en ne considérant que $F(z)$, cad :

$$\frac{C'(z).F(z)}{1 + C'(z).F(z)} = H'(z)$$

On peut montrer que pour obtenir $H(z)$ à partir de $G(z)$, il faut le correcteur $C(z)$:

$$C(z) = \frac{C'(z)}{1 + C'(z).F(z).(1 - z^{-n})}$$

4.7 Régulateurs RST

Ces correcteurs permettent de réaliser un asservissement d'un processus de type P3. D'une structure plus complexe que la structure classique, ces régulateurs sont aussi plus souples. Le calcul de ces correcteurs peut se faire automatiquement par un logiciel.

On présente ici une méthode (un peu fastidieuse car aucune hypothèse n'est faite sur le système) de calcul de ce type de correcteur.

On adopte une démarche en boucle fermée c'est à dire que l'on se fixe la fonction de transfert à atteindre en BF. Notons ce modèle :

$$H(z) = \frac{B_m(z)}{A_m(z)}$$

D'après la structure RST de ce régulateur, la fonction de transfert en BF vérifie aussi (voir figure 9) :

$$H(z) = \frac{T(z).B(z)}{A(z).R(z) + B(z).S(z)} \quad \text{avec} \quad G(z) = \frac{B(z)}{A(z)}$$

Calculer le correcteur c'est établir les polynômes $R(z)$, $S(z)$ et $T(z)$ en fonction des polynômes $A(z)$, $B(z)$, $A_m(z)$ et $B_m(z)$ qui sont respectivement les dénominateurs et numérateurs du système à asservir et du modèle à atteindre en BF.

4.7.1 Décomposition de $G(z)$

$B(z)$ contient des zéros stables (de module < 1) et des zéros instables. Ce polynôme peut donc se scinder en deux parties, l'une contenant les zéros stables :

$$B_s(z) = z^m + a_{m-1}z^{m-1} + \dots$$

et l'autre contenant les zéros instables :

$$B_i(z) = K(z - z_0)(z - z_1) \dots$$

On peut aussi inclure dans cette dernière partie les zéros dont la norme est trop proche de 1

$$\Rightarrow B(z) = B_i(z).B_s(z)$$

4.7.2 Le choix du modèle à obtenir en BF $H(z)$

Comme aucun correcteur ne peut supprimer des retards intrinsèques au système, ces derniers doivent se retrouver dans le modèle $H(z)$. Cela se traduit par :

$$\deg(A_m) - \deg(B_m) \geq \deg(A) - \deg(B)$$

Puisqu'on sait que l'on ne doit pas compenser les zéros instables du système, ces derniers se retrouveront forcément dans la fonction de transfert en BF.

$$B_m(z) = B_i(z).B'_m(z)$$

4.7.3 Propriétés de S , R et T

$R(z)$ c'est le dénominateur du correcteur. Il doit être de degré suffisant pour assurer la causalité, il compense la partie stable du numérateur du système et il assure l'erreur statique nulle (pour un échelon ou une rampe selon les conditions demandées) par la présence d'intégrateurs.

$$R(z) = (z - 1)^i . B_{s(z)} . R'(z)$$

$T(z)$ assure l'obtention du numérateur voulu ($B_m(z)$) :

$$T(z) = B'_m(z).A_0(z) \quad \text{avec} \quad \deg(A_0) \geq 2.\deg(A) - \deg(A_m) - \deg(B_s) + i - 1$$

$S(z)$ doit vérifier

$$\deg(S) < \deg(A) + i \quad \text{et} \quad (z-1)^i . A(z) . R'(z) + B_i(z) . S(z) = A_0(z) . A_m(z)$$

En se fixant A_0 et le nombre d'intégrateurs nécessaires, on peut calculer R , S et T .

5 Pour aller plus loin...