École Centrale des Arts et Métiers

Rapport de laboratoire

Réglages échantillonnés

Thomas Anizet	14164
Hadrien Hachez	15306
Armen Hagopian	14040
Amaury Lekens	14027
Alexis Nootens	16139
Benoît Wéry	14256

Enseignant : B. Arnould

2 juin 2018

Séance 1

Étude d'un premier ordre par les 4 méthodes de discrétisation.

1.1 Introduction

Durant la première séance de laboratoire, il nous a été demandé du calculer l'expression récurrente d'un premier ordre par les 4 méthodes de discrétisation, à savoir :

— La méthode de Euler 1 ;
— La méthode de Euler 2 ;
— La méthode bilinéaire ;
— La méthode équivalent échantillonné bloqué.

Une fois les 4 expressions calculées, l'objectif était d'utiliser le logiciel MATLAB afin de représenter les réponses temporelles et fréquentielles de ces 4 expressions. Ceci, dans la but d'analyser les différents paramètres de ces 4 méthodes de discrétisation.

1.2 Rappels théoriques

Actuellement, les méthodes de traitements numériques pour l'analyse des signaux ont largement pris le dessus comparé aux méthodes analogiques ancestrales. C'est pourquoi nous abordons les notions de procédés de discrétisation. En effet, en traitement numérique, un signal analogique est tout d'abord échantillonné (= discrétisé) avant d'être quantifié et finalement codé. La discrétisation est le procédé par lequel un signal continu est transformé un signal discret. Autrement dit, la discrétisation d'un signal continu f(t) revient à garder un certain nombre de valeurs discrètes (..., f(t0), f(t1), f(t2), ...) correspondant aux valeurs (..., t0, t1, t2, ...) de la variable t : On parle également d'échantillonnage pour les signaux. les différentes valeurs discrètes (..., f(t0), f(t1), f(t2), ...) varient en fonction de la période d'échantillonnage. La figure 1.1 ci-dessous présente le principe d'échantillonnage.

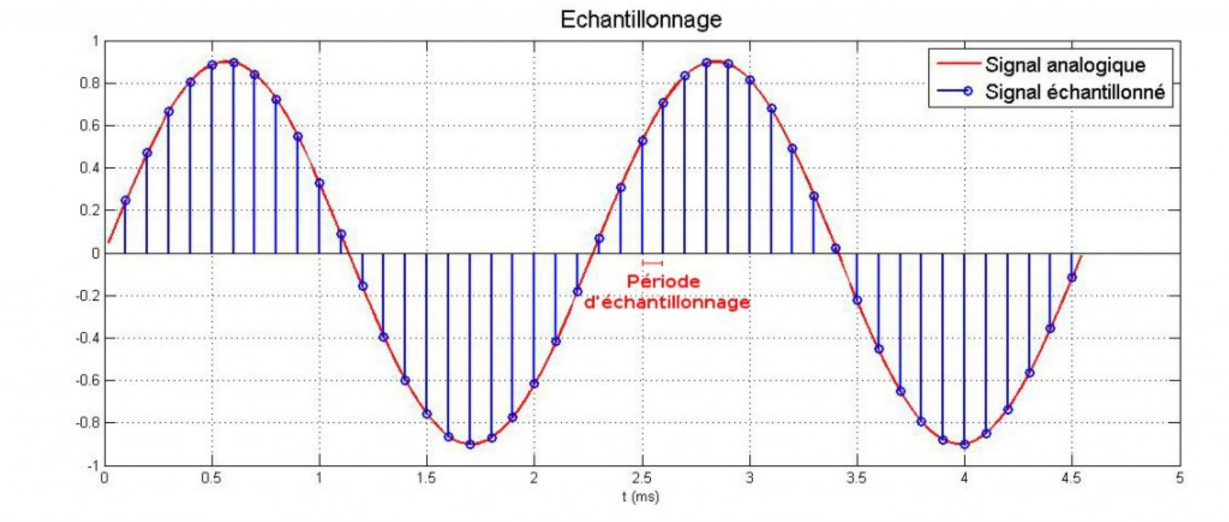

FIGURE 1.1 – Échantillonnage d'un signal continu sinusoïdal.

Ayant introduit la notion de discrétisation, il est intéressant de rappeler le principe des 4 méthodes de discrétisation étudiées en laboratoire :

La méthode de Euler 1 :

La méthode de Euler 2 :

La méthode bilinéaire :

La méthode équivalent échantillonné bloqué :

1.3 Analyse

1.4 Conclusion

Séance 2

Synthèse d'un régulateur continu

2.1 Introduction

Durant cette seconde séance de laboratoire, il a été demandé aux étudiants de synthétiser sur le logiciel MATLAB un régulateur continu en boucle fermée. Une fois cette étape réalisée, la discrétisation de ce régulateur selon 3 périodes d'échantillonnage a été effectuée afin d'obtenir 3 régulateurs discrets. Le but final était alors de comparer, grâce à Simulink, les performances du régulateur continu avec les 3 régulateurs discrets obtenus.

2.2 Notions théoriques

2.3 Analyse

Nous avons construit un régulateur continu par cancelation des poles et zeros en gardant l'intégrateur. Dans la chaine direct il reste un simple intégrateur avec un simple gain qu'il faut determiner. Quand on reboucle un intégrateur avec un simple gain on a un filtre du premier ordre donc ça permet de placer la constante de temps.

On synthétise le régulateur continu et puis on discretise. L'inconvenient de cette procedure est qu'on n'a pas pris en compte le bloqueur d'ordre 0. Ce bloqueur en moyenne entraine un retard d'une demi période. Ça veut dire que ça fera tomber la courbe des phases et qu'on pourra partir en instabilité à un moment. Plus la période d'échantillonnage est grande plus ça se degrade. On fait le passage continu discret du régulateur par les differences finies à gauche et on sait qu'avec les pôles dans le plan gauche, on peut avoir une projection a l'extérieur du cercle. Donc des pôles stables continus peuvent devenir instables en discret. Ce sont les inconvénients de la méthode.

La deuxième étape était de discrétiser le système par l'équivalent échantillonné bloqué. Cela veut dire qu'on passe du domaine S vers Z en tenant compte du bloqueur et qu'on réalise une vraie synthèse discrète de la même façon en prenant les pôles du système pour les mettre aux zeros du régulateur et inversement. On maintient l'intégrateur discret avec le dénominateur. Et on reboucle pour aller placer les poles discrets à l'endroit voulu c'est à dire les poles continus projetés en Z avec $Z = e^{sh}$.

La deuxième réponse : la réponse du système souhaité et celle qu'on obtient sont quasiment les mêmes car la période d'échantillonnage est petite et que le temps mort supplémentaire ne nous embête pas trop.

La vraie synthèse discrete est lorsque le système est discrétisé avec c2d et là on a que la réponse est exactement celle qu'on souhaite. Si on augmente la période d'échantillonnage, constante de temps au dénominateur est 3 secondes, c'est pas mal une période d'échantillonnage de 1 seconde.

Avec la discrétisation du régulateur, on commence à avoir des problèmes car on ne tient pas compte du bloqueur.

Dans le dernier cas, aux instants d'échantillonnage, on est sur la valeur souhaitée. En dehors de ces instants, on ne maitrise pas car on a imposé les pôles d'un système discret pour qu'aux instants d'échantillonnage on soit sur la réponse du filtre du premier ordre. La réponse souhaitée est la réponse avec un régulateur synthétisé en continu et discrétisé et donc on n'a pas tenu compte dans la synthèse du régulateur du bloqueur d'ordre 0. Mettre sur les scopes : ce qu'on souhaite et ce qu'on obtient et puis on voit ce qu'on souhaite On voit l'influence de l'intégrateur, le système part chercher une asymptote a l'infini, à la période d'échantillonnage qui suit on est sur le modele de reference souhaité. On est sur la valeur.

Le régulateur discrétisé on a des dépassements importants et avec le vrai régulateur discret malgré que la période d'échantillonnage est importante, aux instants d'échantillonnage on est à nouveau sur la trajectoire. Donc c'était le but de cet exercice, il faut se méfier quand on fait synthèses continues et qu'on discrétise.

On a pris en compte le bloqueur en faisant le passage S vers Z avec c2d Quand on fait c2d on fait H(s)*B0(s) et c'est le produit de ces deux là qu'on passe dans le plan Z. On ajoute donc à H(s) le bloqueur d'ordre 0, c'est pas correct point de vue mathématique que la transmittance en S, il bloque pendant une période donc si on applique une impulsion discrete a l'entrée du bloqueur pendant la période d'échantillonnage h il doit faire ça (voir notes). Impulsion discrete c'est celle calculée dans le régulateur par la routine qu'on a établi. Le signal en bleu, c'est un echelon auquel on soustraie un echelon décalé d'une période. C'est ce signal qu'on obtient pour une impulsion discrete en entree.

2.4 Conclusion

Séance 3

Synthèse d'un régulateur discret

3.1 Introduction

Durant la troisième et dernière séance de laboratoire, il a été demandé aux élèves de synthétiser sur le logiciel MATLAB® un régulateur discret de compensation en boucle fermée et d'en analyser la réponse. Pour ce faire, il a tout d'abord fallu calculer l'équivalent échantillonné bloqué de ce régulateur (discrétisation). Et ensuite, imposer un intégrateur dans le régulateur discret de compensation.

3.2 Hypothèses

Soit le système défini par l'équation :

$$H_c(s) = \frac{1}{s\,(s+0{,}5)} = \frac{2}{s\,(2s+1)} \tag{3.1}$$

Nous cherchons à effectuer la synthèse d'un régulateur discret compensant le système (3.1) en boucle fermée, et respectant les spécifications suivantes :

1. Dépassement de 5% ($\xi = 0{,}69$)
2. Temps de réponse à 95% ($3T$) de 1
3. Erreur de vitesse nulle
4. $h = 0{,}1$

Notons avant tout que la spécification sur l'erreur de vitesse nulle implique un double intégrateur dans la chaîne directe. Ceci nous indique d'avance que le régulateur suivra la forme :

$$R(Z) = \frac{\dots}{(Z-\alpha)(Z-\beta)} \tag{3.2}$$

3.3 Analyse

La première étape est toujours la même, nous devons traduire la fonction de transfert dans le domaine discret. Pour cela nous utilisons la commande `c2d()` de MATLAB® qui équivaut à convoluer la fonction par un bloqueur d'ordre 1. C.-à-d. effectuer l'opération :

$$H_d(Z) = \underset{s \to Z}{\mathcal{L}} H_c(s) \tag{3.3}$$

$$= \frac{k}{(Z-1)(Z-p)} \tag{3.4}$$

Dans cette forme, les points sont distants de la période d'échantillonnage. Cela entraine que l'influence de l'entrée sur la sortie est perçue après cette même période. Ceci implique également que la différence entre l'ordre du dénominateur et du numérateur doit être égale à 1. Pour imposer cet ordre, nous rajoutons un zéro au numérateur de (3.4). Ce qui donne :

$$H_d(Z) = \frac{k\,(Z-z)}{(Z-1)(Z-p)} \tag{3.5}$$

De cette forme discrétisée, nous désirons extraire les pôles et les zéros. MATLAB® propose une fonction permettant de les extraire en un appel, `zpkdata()`. Nous employons par la suite ces coefficients obtenus dans le régulateur $R(Z)$ pour compenser $H_d(Z)$. $R(Z)$ est donc une régulateur de compensation.

$$R(Z) = k\,\frac{(Z-p)}{(Z-z)(Z-1)} \tag{3.6}$$

On a un décalage car l'ordre du dénominateur est plus grand que l'ordre du numérateur. Quand le régulateur aperçoit un écart de réglage à sa consigne, il doit réagir immédiatement. Il « mouline » dans un temps qui est supposé très petit et négligeable sur la période puis il envoie. On rajoute un zéro au numérateur pour faire en sorte qu'il n'y ait pas de retard dans le régulateur.

$$R(Z) = k\,\frac{(Z-p)(Z-n)}{(Z-z)(Z-1)} \tag{3.7}$$

Ensuite nous faisons le produit en boucle ouverte du régulateur avec le système.

$$R(Z)H_d(Z) = B_0(Z) = \frac{k\,(Z-n)}{(Z-1)^2} \tag{3.8}$$

On constate que dans le chaîne directe, il y a bien un retard d'une période.

3.4 Synthèse

Dans le plan complexe les pôles continus sont sur une droite d'amortissement inclinée d'un certain angle Θ, tel que dans notre cas $sin(\Theta) = 0{,}69$. C'est le sin de l'angle d'inclinaison de la droite d'amortissement sur laquelle se trouvent les pôles. Le $sin(\Theta)$ détermine ainsi le dépassement de 5% cité dans l'énoncé du laboratoire et qui est une des spécifications du régulateur.

Ensuite, nous avons le temps de réponse à 95% de 1. Si nous traduisons cela sous forme de pôles et de zéro dans le cas d'un filtre du 1er ordre, nous trouvons que le lien entre un pôle et une constante de temps est l'inverse l'un de l'autre. La constante de temps vaut T=1/3. La partie réelle du pôle va donc se trouver à l'inverse (et signe opposé) de cette

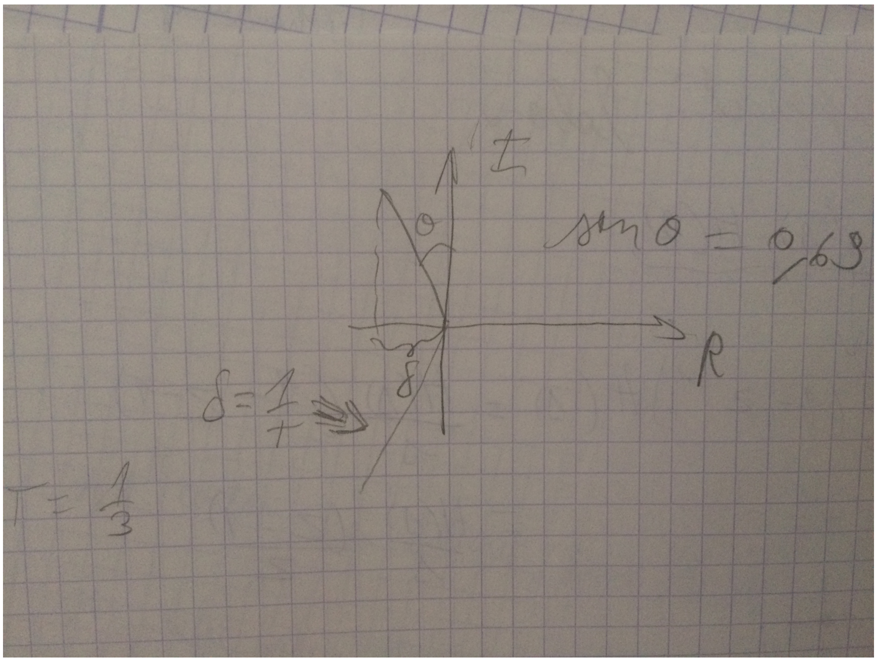

FIGURE 3.1 – Temps de réponse.

constante de temps. On peut le voir sur la Figure 3.1 où, comme on l'a dit, $\delta = T^{-1}$ et donc $\delta = 3$. On trouve ainsi la paire de pôles complexes qui donne en boucle fermée une réponse pour que les deux premières conditions de l'énoncé soient respectées.

Nous pouvons également faire la synthèse du régulateur. Nous savons que $H_s(Z)$ est sous la forme :

$$H_s(Z) = \frac{\dots}{(Z - P_{d1})(Z - P_{d2})} \tag{3.9}$$

Nous cherchons à trouver les valeurs de P_{d1} et P_{d2}. Nous allons les trouver par équivalences avec $F(Z)$, la transmittance en boucle fermée.

$$F(Z) = \frac{k\,(Z-n)}{(Z-1)^2 + k\,(Z-n)} \tag{3.10}$$

$$\begin{cases} k = 2 - P_{d1} - P_{d2} \\ n = \frac{(1 - P_{d1} P_{d2})}{k} \end{cases} \tag{3.11}$$

En utilisant la fonction `dtr2ord2o.m` fournie, nous avons pu générer la transmisttance d'un système continu du second ordre répondant aux spécifications. Nous avons pu extraire les valeurs P_{d1} et P_{d2} de cette transmittance, et par résolution du système 3.11 nous avons obtenu la transmittance du système à synthétiser en boucle fermée.

Cette séquence d'opération a été réalisé sous MATLAB à l'aide du script suivant :

```
h = 0.1;
nH = 1;
dH = conv([1 0], [1,0.5]);
Hc = tf(nH, dH);
Hd = c2d(Hc, h, 'zoh');

% R(z) = [K(Z-p)(Z-n)]/[(Z-z)(Z-1)]
[z,p,~] = zpkdata(Hd, 'v');
p = p(2);

[pc] = dtr2ord2o(1, 5, 5); % trouve les poles pd1 et pd2
pd = exp(pc*h);            % discretise les poles
K = 2-pd(1)-pd(2);         % calcul le gain d'Evans unite
n = (1-pd(1)*pd(2))/K;     % calcul du zero additionnel

Z = tf('z', h);
F = K*(Z-n)/((Z-1)^2+K*(Z-n));    % boucle fermee
R = K*(Z-p)*(Z-n)/((Z-z)*(Z-1));  % regulateur

% autre maniere de fermer la boucle
Bo = R*Hd;
Bf = feedback(Bo,1);
Bf = minreal(Bf);
```

3.5 Conclusion

Une fois les valeurs de k et n obtenues, la fonction de transfert du régulateur $R(Z)$ et la transmittance en boucle fermée $F(Z)$. Nous pouvons finalement réaliser un modèle simulink pour simuler l'exactitude de nos calculs. Nous avons tenté de reproduire au mieux l'exemple présenté au laboratoire, notre version est présenté à la figure 3.2.

La partie supérieur du graphique représente l'action correctrice, on note ses oscillations du à la présence d'un zéro dans la partie gauche du cercle des pôles et des zéros.

Dans la partie inférieur du graphique, nous observons tout d'abord l'échelon unitaire en jaune s'exprimant à 0.5 unité de temps. La courbe en bleu est l'équivalent en S de la fonction $F(Z)$, on peut y noter sa réponse stable avec un très faible dépassement. Et pour finir en orange se trouve la sortie du système continue $H_c(S)$ qui est placé en cascade à la suite du régulateur $R(Z)$. On désirait un dépassement de 5% mais nous obtenons à la place un dépassement monstrueux, cela est dû au zéro que nous avons du ajouter qui entraine également les oscillations de la réponse entre chaque instant d'échantillonnage. Afficher la sortie du système continue au lieu de $F(Z)$ permet d'observer ces oscillations qui seraient cachées dans l'équivalent échantillonné.

On peut noter que le régulateur effectue correctement sa tâche puisque après une phase de stabilisation et à chaque instant d'échantillonnage, toutes les 0.1 unités de temps, la différence entre la courbe et la valeur attendue est 0 !

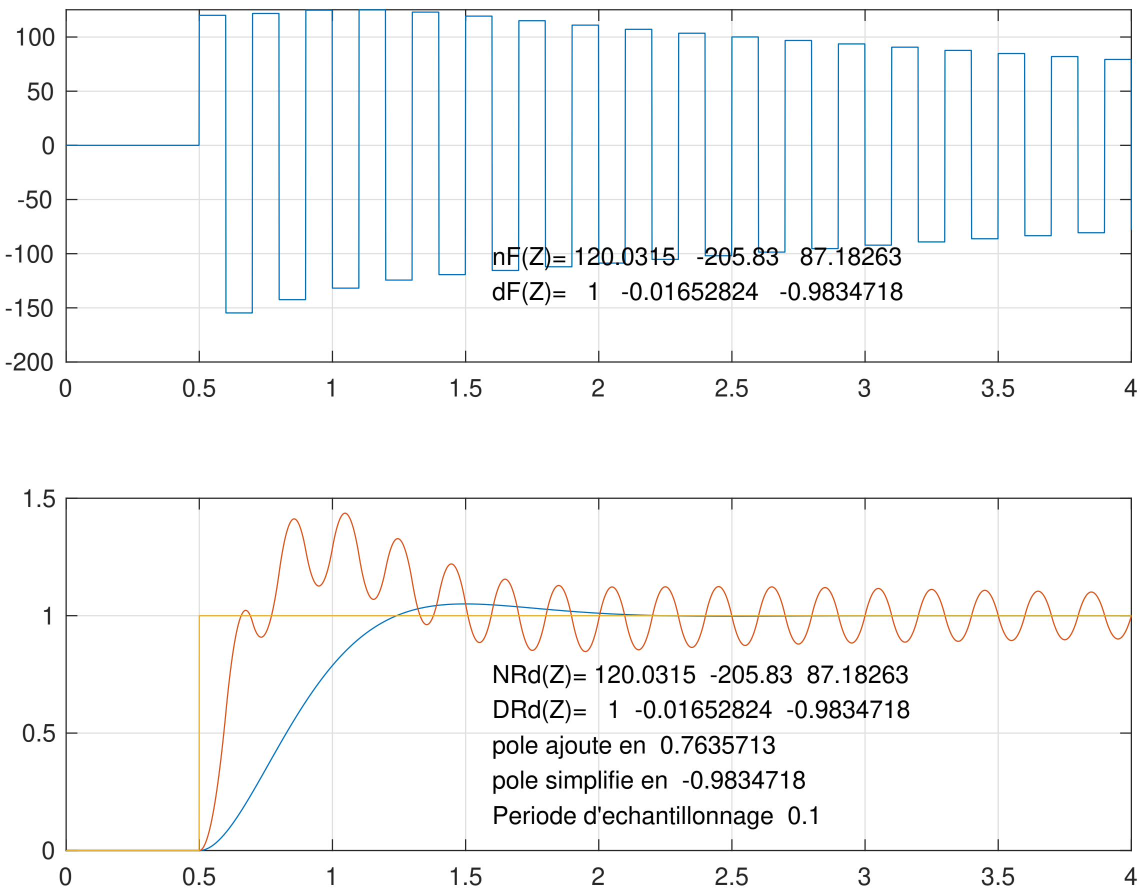

FIGURE 3.2 – Simulation de la transmittance en boucle fermée échantillonnée. On peut apercevoir sur la courbe orange les oscillations de la sortie entre chaque période d'échantillonnage qui dure 100 ms.