

## Contrôle commande

### TP : Etude d'un asservissement d'un moteur à courant continu

*Le TP comporte à la fois des parties à préparer avant de venir en séance (elles sont repérées par un symbole ✍) et d'autres qui concernent des manipulations à effectuer pendant la séance, soit sur Matlab Simulink (repérées par le symbole ⚙), soit sur le microcontrôleur et le montage analogique associé (elles sont repérées par un symbole 🛠).*

### Intro

Le but de ce TP est de réaliser un asservissement numérique de vitesse d'un moteur à courant continu (MCC), qui est présent sur les maquettes MCC vues en ET3 ESR (module Automatique). Le correcteur numérique sera réalisé à l'aide d'un microcontrôleur STM32, via la carte de développement Nucléo F446. L'annexe en fin d'énoncé fournit des informations sur cette carte de développement et le brochage associé.

Chaque étude se décomposera en trois phases : le calcul théorique, la simulation (sous Matlab Simulink) et le test sur maquette (via la programmation du microcontrôleur sous STM32IDE).

Afin de préparer l'étude du moteur, on propose d'étudier dans un premier temps un système électronique « simple » émulant le fonctionnement du moteur : un circuit électronique du second ordre.

# I. Etude d'un second ordre

## A. Préparation

On considère une fonction de transfert  $H$  du second ordre, mis sous la forme suivante :

$$H(p) = \frac{H_0}{(1 + \tau p)^2}$$

Avec :

$$H_0 = 1$$

$$\tau = 4.7 \text{ ms}$$

✎ Prévoir un circuit à base d'amplificateurs opérationnels permettant de réaliser cette fonction de transfert.

On considère un asservissement à retour unitaire de cette fonction de transfert. On la soumet à un échelon de tension d'amplitude  $E_0$ .

✎ Rappeler l'expression de l'erreur statique et faire l'application numérique pour une valeur  $E_0 = 2 \text{ V}$ .

On souhaite corriger le système à l'aide d'un correcteur PI numérique, dont la fonction de transfert est rappelée :

$$C(p) = K \frac{(1 + \tau_i p)}{\tau_i p}$$

On adoptera une méthode de correction par compensation du pôle ayant la plus grande valeur de constante de temps (pôle dit dominant).

- ✎ Ecrire l'expression de la nouvelle fonction de transfert (analogique) en boucle ouverte dans ces conditions.
- ✎ Représenter le diagramme de Bode asymptotique de cette fonction de transfert (en boucle ouverte).
- ✎ Rappeler les deux équations définissant une marge de phase de  $45^\circ$ . On notera  $\omega_0$  la pulsation liée à cette marge.
- ✎ Donner l'expression de  $\omega_0$ .
- ✎ Déterminer l'expression de  $K$ , notée  $K_a$ , qu'il faut imposer pour obtenir cette marge.
- ✎ Ecrire la fonction de transfert (analogique) en boucle fermée correspondante.
- ✎ En adoptant la méthode d'Euler, déterminer l'expression du correcteur numérique équivalent.
- ✎ En déduire l'équation aux différences correspondant à ce correcteur, en fonction des constantes du correcteur PI analogique.
- ✎ Que devient l'étude précédente si l'on intègre une chaîne CAN-CNA en série avec celle-ci, caractérisée par une fréquence d'échantillonnage  $F_e$  ?

## B. Etude du système corrigé idéal

Pour chaque étape, vous ferez plusieurs simulations : dans le domaine continu (qui est la référence), en prenant un correcteur dans le domaine discret, puis en écrivant l'équation aux différences à la place du correcteur.

Vous ferez ensuite le câblage associé, puis le codage du correcteur, avant de passer à l'étape suivante. On négligera dans un premier temps l'influence du CAN et du CNA dans les calculs et dans la simulation.

La mise en place se fera progressivement : on testera d'abord la boucle ouverte seule (sans correcteur), puis la boucle fermée non corrigée à retour unitaire, puis une correction proportionnelle avant de mettre en place la correction PI.

### 1. Mise en place de la boucle ouverte

- ☞ Faire la simulation de la boucle ouverte (analogique) sous Matlab Simulink (l'entrée sera un échelon de tension).
- ☞ Câbler le montage analogique. Vérifier-le indépendamment de la partie numérique, sur la plaque labdec.
- ☞ Ouvrir le projet donné et analyser le code fourni.
- ☞ Implémenter un gain numérique unitaire et câbler l'ensemble de la boucle ouverte. Tester-la.

### 2. Etude de la boucle fermée

- ☞ Faire la simulation de la boucle fermée (analogique) sous Matlab. Mesurer le temps de réponse à 5%.
- ☞ Faire la simulation du diagramme de Bode de la boucle ouverte. Déterminer la marge de phase et conclure.
- ☞ Modifier le câblage pour fermer la boucle.
- ☞ Implémenter un gain numérique unitaire et tester la boucle fermée. Comparer à la simulation.

### 3. Correction proportionnelle

- ☞ A l'aide de la simulation sous Matlab, dans le domaine analogique, prévoir le gain qu'il faut imposer pour avoir une marge de phase de  $45^\circ$ .
- ☞ Faire les simulations associées à l'aide d'un correcteur P (inclus dans le PID contrôler).
- ☞ Coder un gain proportionnel.
- ☞ Tester alors la boucle fermée corrigée. Comparer à la simulation.

## 4. Correction PI

On souhaite mettre en place le correcteur PI dimensionné en préparation.

- ☞ Réaliser sous Matlab Simulink le correcteur PI analogique. Fixer la constante de l'intégrateur identique à la constante de temps du système et imposer un gain  $K$  unitaire.
- ☞ Vérifier sous Matlab Simulink le gain  $K$  qu'il faut imposer pour avoir une marge de phase de  $45^\circ$  et le comparer à la préparation théorique ( $K_a$ ).
- ☞ Faire la simulation en temporel et noter les grandeurs.
- ☞ Refaire un schéma en remplaçant le correcteur PI analogique par un correcteur PI numérique, en utilisant la méthode d'Euler. Vous prendrez une période d'échantillonnage  $T_e = 10 \mu s$ .
- ☞ Vérifier le bon fonctionnement de la simulation (comparer la correction analogique et numérique, d'un point de vue temporel et vis-à-vis des marges).
- ☞ Refaire un schéma en remplaçant le correcteur PI numérique par des blocs permettant de coder l'équation aux différences. Coder le correcteur PI précédant. Vérifier le bon fonctionnement de ce nouveau bloc.
- ☞ Implémenter le code sur le microcontrôleur et tester le système corrigé, et le comparer aux simulations.
- ☞ En utilisant le PID tuner de Matlab Simulink, simuler puis coder la mise en place d'un correcteur PI. Vous paramétrez le tuner afin d'avoir une réponse la plus rapide possible sans dépassements.
- ☞ Tester cette solution sur le microcontrôleur

## C. Etude de l'influence du bloqueur

On change la période d'échantillonnage :  $T_e = 2 \text{ ms}$ .

- ☞ Observer les changements en temporel entre le système analogique et le système numérique sous Matlab Simulink.
- ☞ Observer la marge de phase obtenue avec le correcteur en  $z$ , ainsi que les réponses temporelles en simulation. Pourquoi cela a-t-il changé ?
- ☞ Dupliquer la boucle comportant le système analogique et rajouter la fonction de transfert analogique du bloqueur linéarisé.
- ☞ Trouver la valeur de  $K$ , notée  $K_n$ , pour avoir une marge de phase de  $45^\circ$ . Vérifier que la pulsation de la marge de phase est compatible avec la forme linéarisée du bloqueur.
- ☞ Dupliquer la boucle comportant le système numérique corrigé en  $z$  et y placer les nouvelles valeurs du correcteur. Faire les simulations et comparer le système analogique initial, le système corrigé avec et sans prise en compte du bloqueur.
- ☞ Modifier dans le code la période d'échantillonnage et les coefficients.
- ☞ Tester le correcteur et conclure.

On diminue encore la fréquence d'échantillonnage :  $T_e=10$  ms.

- ☞ Changer la fréquence d'échantillonnage du code et observer ce qui se passe sur le circuit analogique. Proposer alors une solution pour que le fonctionnement du circuit soit assuré, basée sur la méthode vue précédemment.

## II. Application à l'asservissement d'un moteur MCC

Vous aller appliquer les méthodes vues sur le circuit analogique précédent à l'asservissement d'une MCC.

### A. Rappels sur la maquette MCC

La MCC est alimentée sous 12V maximum, via un amplificateur de puissance, présent sur la maquette. Elle est accouplée à une génératrice qui peut être chargée ou freinée, ainsi qu'à une autre génératrice tachymétrique qui donne une tension proportionnelle à la vitesse de la machine MCC.

Attention : la sortie de la génératrice tachymétrique que vous utiliserez est située sur le côté de la maquette. Il ne faut pas utiliser la sortie sur la face avant.

L'ensemble {amplificateur, MCC, génératrice tachymétrique} est équivalent à un système passe-bas du second ordre, caractérisé par deux constantes de temps : l'une  $\tau_e$  liée à la partie électrique de la MCC, et l'autre  $\tau_m$  liée à la partie mécanique de la MCC. On notera  $H_0$  le gain statique.

La fonction de transfert se met donc sous la forme suivante :

$$H_m(p) = \frac{H_0}{(1 + \tau_e p)(1 + \tau_m p)}$$

Avec :

$$\tau_m = 47 \text{ ms}, \tau_e = 0.5 \text{ ms et } H_0 \approx 1.4$$

Un filtre passe-bas a été rajouté après la génératrice tachymétrique, afin d'éliminer les bruits parasites du signal. Il est réalisé à l'aide d'un simple circuit RC dont les valeurs sont les suivantes :

$$\begin{cases} R = 47 \text{ k} \\ C = 940 \text{ nF} \end{cases}$$

La sortie du filtre passe-bas se fait sur le côté de la maquette, que vous utiliserez comme sortie de la boucle ouverte.

- ✎ Déterminer la constante de temps  $\tau$  liée à ce filtre, et la forme de la fonction de transfert globale notée  $H$ .

Au regard des ordres de grandeur, ce système peut être assimilé à un second ordre : la constante de temps électrique est en effet négligeable devant les autres, ce qui permet de négliger son influence.

## **B. Mise en place de l'asservissement**

Ne disposant que de 7 maquettes MCC, vous allez dans un premier temps mettre en place l'asservissement numérique avec la simulation.

A l'aide de Matlab Simulink (sous lequel vous ferez un nouveau projet), de la méthodologie et des résultats de la première partie, mettre en place un asservissement de la MCC.

Il est recommandé de suivre une méthode permettant de valider des étapes intermédiaires, et de ne pas faire directement l'asservissement final.

## ANNEXE

Le microcontrôleur utilisé est un STM32 de chez ST (architecture 32 bits). Ce microcontrôleur est utilisé via la carte Nucleo F446, qui permet de le programmer et d'avoir des ports d'entrée/sortie sur des connecteurs. Cette carte de développement est connectée à une carte d'interface (qui est en dessous), permettant d'avoir des connecteurs adaptés aux maquettes pédagogiques de Polytech, et permettant de faire une remise en forme des signaux (tous les signaux analogiques sont centrés sur 0, et peuvent aller de -12 V à +12 V).

<b>Signal</b>	<b>Broches du STM32</b>	<b>Connecteurs de la maquette analogique</b>
ADC1Channel 9		Vin1
ADC2 Channel 12		Vin2
DAC 1		Vout2
DAC 2		Vout1
IN/OUT	PA6	Debug
IN/OUT	PA7	Debug
IN/OUT	PD14	Debug
IN/OUT	PD15	Debug
IN/OUT	PF12	Debug

Vous utiliserez :

- Le CAN « ADC1 » pour la consigne
- Le CAN « ADC2 » pour la valeur de retour de l'asservissement
- Le CNA « DAC1 » pour fournir le signal en entrée de votre système analogique à asservir (circuit analogique ou MCC)