

AuroFC2U2 - correction numérique, méthode de Zdan

Introduction

Le but de ce sujet est d'effectuer la synthèse de correcteurs numériques par la méthode des pôles dominants, aussi appelée méthode de ZDAN. C'est une méthode de pilotage de procédés aux instants d'échantillonnage. Son principe consiste à compenser les pôles et zéros stables de la transmittance bloquée du système à corriger, puis à ajouter de nouveaux pôles par l'intermédiaire d'un correcteur afin d'obtenir un modèle d'ordre deux défini par un cahier des charges.

Les manipulations seront effectuées sous Matlab Simulink à l'aide d'une carte de contrôle commande permettant de piloter en vitesse une maquette constituée d'un moteur à courant continu. Des capteurs de vitesse et de position fournissent une image de la sortie du système directement dans Matlab Simulink.

1 Asservissement de vitesse d'un moteur à courant continu

On considère que le moteur est modélisé par un système d'ordre un de fonction de transfert :

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

1.1 Identification du système

1. Choisir un période d'échantillonnage faible ($T_e = 25ms$) et faire un essai en boucle ouverte à une rampe, dans l'intervalle $[0, 10V]$ pour déterminer la zone de linéarité du système¹.
2. Réaliser un essai à un échelon dans la zone de linéarité du système, et en déduire les paramètres du modèle d'ordre un du moteur.
3. vérifier vos calculs en réalisant le même essai avec le modèle estimé précédemment. Pour obtenir un modèle plus conforme à la réalité, on pourra ajouter un offset et un étage de saturation. Par la suite, le modèle du moteur pourra être intégré sous la forme d'un macro-bloc. Pour cela, il suffit de sélectionner les blocs à intégrer dans le macro-bloc, puis avec le bouton droit de la souris, sélectionner le menu *create subsystem*
4. Choisir une période d'échantillonnage par la méthode de Büllher, puis calculer la transmittance bloquée du système à cette période d'échantillonnage.
5. vérifier vos calculs en réalisant un essai à un échelon sur la transmittance bloquée du système obtenu. Pour obtenir un modèle plus conforme à la réalité, on pourra ajouter un offset et un étage de saturation au modèle. Un macro-bloc du modèle équivalent aux instants d'échantillonnage sera généré et utilisé par la suite.

1. Sur matlab 6.5, La voie de sortie est pilotée en pleine échelle $[\pm 10V]$ sur l'intervalle de commande $[-1, 1]$

1.2 Synthèse d'un correcteur par la méthode des pôles dominants

On cherche maintenant à corriger le système par la méthode des pôles dominants. Le cahier des charges impose le comportement suivant :

- une erreur statique d'ordre 1 nulle,
 - un dépassement inférieur à 5%,
 - un temps de pic : $t_{pic} = 5T_e$.
1. Calculer le modèle à obtenir pour répondre aux exigences temporelles du cahier des charges. Vérifier les calculs par un essai indiciel du modèle sous Matlab Simulink
 2. Donner la structure du correcteur
 3. Calculer les paramètres du correcteur
 4. Vérifier les performances du correcteur obtenu à la fois sur le système réel et le système simulé². Un signal additif de type créneau retardé sera placé en sortie du modèle numérique pour simuler une perturbation. Vérifier les performances du système en appliquant une consigne de type créneau entre 2V et 2.5V ; puis une consigne de type créneau entre 3V et 5V. Conclure³

2 Asservissement en position d'un moteur à courant continu

Pour mesurer la position de l'axe du moteur, on ajoute un intégrateur au système :

$$G(p) = \frac{K_v}{p(1 + \tau p)}$$

2.1 Identification du système

La fonction de transfert du système comportant un intégrateur, son identification s'effectue en boucle fermée.

1. Montrer que la fonction de transfert bouclée de $G(p)$ peut s'écrire sous la forme d'un système d'ordre deux classique et extraire les relations (à partir de la réponse indicelle) qui conduisent à la détermination du gain statique et de la constante de temps du système en fonction du coefficient d'amortissement et de la pulsation propre du système bouclé.
2. Sous Matlab Simulink, réaliser l'essai permettant d'identifier le système. On pourra appliquer un gain proportionnel (amplification ou atténuation) pour améliorer la visibilité de la réponse indicelle et permettre des mesures graphiques plus précises des paramètres du système. Vérifiez vos calculs sous Matlab Simulink.
3. Choisir une période d'échantillonnage par la méthode de Bülher, puis calculer la transmittance bloquée du système à cette période d'échantillonnage.
4. vérifier vos calculs en réalisant un essai à un échelon sur la transmittance bloquée du système obtenu (en boucle fermée).

² On pourra à la fois observer le comportement du système aux instants d'échantillonnage (transmittance discrète) et entre les instants d'échantillonnage (utilisation d'un bloqueur couplé à un modèle continu)

³ l'observation d'un système corrigé passe obligatoirement par la visualisation de quatre signaux : la consigne, l'erreur, la commande et la sortie

2.2 Etude du système

1. On souhaite connaître l'erreur statique d'ordre un du système bouclé. Calculer, à partir de la transmittance du système, la valeur de cette erreur statique. Valider ce calcul par un essai à un échelon à la fois sur le système réel et le système simulé.

2.3 Synthèse d'un correcteur par la méthode des pôles dominants

On cherche maintenant à corriger le système par la méthode des pôles dominants. Le cahier des charges impose le comportement suivant :

- une erreur statique d'ordre 1 nulle,
 - un dépassement inférieur à 20%
 - un temps de pic : $t_{pic} = 5T_e$
1. Calculer le modèle à obtenir pour répondre aux exigences temporelles du cahier des charges.
 2. Donner la structure du correcteur
 3. Calculer les paramètres du correcteur
 4. Vérifier les performances du correcteur obtenu à la fois sur le système réel et le système simulé. Un signal additif de type créneau retardé sera placée en sortie du modèle numérique pour simuler une perturbation⁴.
 5. Observer le comportement du système simulé lorsque les perturbations sont placées avant le système.

4. l'observation d'un système corrigé passe obligatoirement par la visualisation de quatre signaux : la consigne, l'erreur, la commande et la sortie