

Séance 1 Bloc 2

L'objectif de la séance sera de simuler le fonctionnement du moteur avec encodeur tout en prenant **en main matlab et simulink** dans l'optique de la simulation. Pour pouvoir simuler le fonctionnement du moteur, nous devons déterminer sa fonction de transfert. Nous ne disposons pas des paramètres internes du moteur, donc nous passerons par la réponse indicielle pour trouver cette fonction de transfert. L'étape de simulation permet d'implémenter plus rapidement la correction du fonctionnement du moteur avec encodeur asservi.

Attention, les simulations aussi seront les livrables du bloc 2. Donc conservez vos codes, simulations, résultats etc...

Travail attendu pour cette séance :

1-Faire le montage permettant de faire tourner le moteur avec encodeur . Le logiciel oneshot vous permettra d'afficher sur le moniteur série la vitesse en rotations par minute du moteur (RPM).

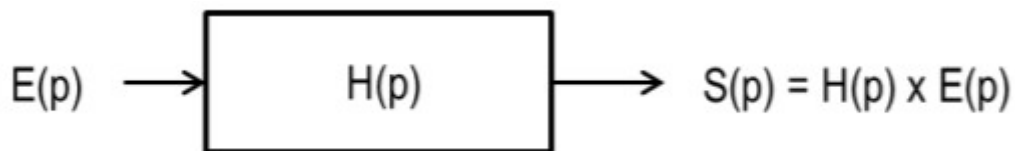
Les impulsions du codeur sont comptées sur Arduino en utilisant des interruptions

2-Donner l'algorithme du programme et expliquer la méthode d'obtention de la vitesse ; voir les liens dans « aide à la programmation ».

3- A partir des données obtenues en 1, tracez la réponse indicielle du moteur.

4- Le moteur est un système de second ordre, trouvez sa fonction de transfert. Trouver la fonction de transfert revient à trouver les polynômes de son numérateur et de son dénominateur.

5- Utilisez les schémas blocs sur Simulink pour obtenir la réponse indicielle du moteur en boucle ouverte. L'échelon, le signal d'entrée, est donné par STEP et l'observation des signaux se fait avec SCOPE. Vous mettrez dans le bloc transfer function l'équation obtenue grâce à TFEST.



6- A partir de l'observation de la réponse indicielle en boucle ouverte, justifiez l'utilisation de la boucle fermée . Puis, celle de facteurs de correction.

7- Réalisez le branchement du système en boucle fermée à retour unitaire. A partir de l'observation de la réponse indicielle en boucle fermée, justifiez celle de facteurs de correction.

Procédure :

A- La réponse indicielle (RI)

Le montage à réaliser comprend le moteur + le pont en H + l'étage de puissance le tout correctement connecté avec la carte Arduino. Il y a deux méthodes pour la trouver :

1-Le code dans le fichier : RepIndicielleMot-FIT0520_one_shot est à téléverser dans Arduino. Donc la pin 2 Arduino doit recevoir le fil jaune (ou le fil vert). Ces deux fils donnent la valeur de la vitesse réelle du moteur. Ce programme affiche sur le moniteur série la réponse indicielle (RI) réelle du moteur.

Faire la moyenne de 3 mesures différentes de cette RI. Il faudra récupérer ces valeurs et les mettre dans un fichier EXCEL.

Ce n'est qu'après avoir trouvé la RI que la fonction de transfert pourra être trouvée. La fonction de transfert est une modélisation mathématique du moteur. En la trouvant, on pourra trouver les coefficients correcteurs proportionnel, intégral et dérivé respectivement K_p , K_i et K_d nécessaires à l'asservissement du moteur réel. En ajoutant ces coefficients au programme, on s'assure que la réponse du moteur à une consigne de vitesse sera rapide, précise et stable.

2- Reproduire la méthode donnée dans le fichier 4a-acquisition_matlabarduino

Données supplémentaires :

Le jeu de 50 RI contenu dans le fichier copie de réponse indicielle, doit permettre aux personnes n'ayant pas le matériel à leur disposition de travailler aussi et aider leur équipe. Ce sont des réponses de moteurs semblables à celui de chacune des équipes mais qui pourrait avoir une réponse différente.

Le jeu des RI permettra l'obtention de la fonction de transfert et des coefficients qui devront quand même être adaptées plus tard aux valeurs du vrai moteur de chaque équipe.

B- La fonction de transfert

Avant tout la réponse indicielle vous permet de trouver la fonction de transfert. La fonction de transfert est une modélisation du moteur. Vous utiliserez matlab pour la trouver.

Il y a plusieurs méthodes pour trouver la fonction de transfert qui est le rapport de la sortie sur l'entrée en fonction de p (soit s en notation anglo-saxonne pour matlab), parmi elles :

1-vous utilisez deux fichiers matlab : l'un contenant les entrées qui ont donné la réponse indicielle donc **que des 5Volts** et l'autre la vitesse en rotations par minute (le jeu des 50). Les deux fichiers doivent être de même taille.

a-En utilisant iddata puis TFest , vous obtenez directement la fonction de transfert.

<https://fr.mathworks.com/help/ident/ref/iddata.html>

b-Toujours avec les même deux fichiers (entrée échelon et réponse en RPM) avec la toolbox identification de systèmes

<https://fr.mathworks.com/help/ident/getting-started-1.html>

<https://fr.mathworks.com/videos/introduction-to-system-identification-toolbox-68901.html>

2-En travaillant sur la courbe de la vitesse en fonction du temps en utilisant l'instruction curvefitting <https://fr.mathworks.com/help/curvefit/curve-fitting.html>

De la sortie en fonction du temps, en fonction de la table de Laplace vous déduisez celle en fonction de p (donc de s sous matlab). Puis sachant que l'entrée est un échelon vous déduisez la fonction de transfert.

3-En suivant la méthode donnée dans le fichier 4a-acquisition_matlabarduino.

C- La correction PID (pour ceux ayant fini le travail de cette séance)

Cette étape se fait par simulation sous Simulink , après avoir

- modélisé le moteur grâce au bloc fonction de transfert
- placé l'entrée avec le bloc échelon (step)
- Et le bloc pour visualiser (avec le bloc scope)

La correction PID se fait en changeant les paramètres du bloc PID jusqu'à avoir une sortie qui

respecte la consigne le mieux possible : rapidement, avec précision et de façon stable.