Mini-projet 1 : Régulation de vitesse d'un propulseur sans capteur mécanique

Motivation : Étudiant souhaitant s'investir dans le contrôle moteur, la robotique.

Mot-clés : moteur à courant continu, automatique, systèmes bouclés, systèmes embarqués

Objectif du projet :

Les mini-robots autonomes puisent généralement l'énergie électrique pour leur propulsion dans des batteries d'accumulateurs de type lithium polymère (Fig. 1a). Au cours de son utilisation, on observe une chute progressive de la tension de la batterie, soulevant ainsi un problème majeur pour le contrôle du mini-robot.

L'objectif du projet est de développer sous le logiciel Matlab™/Simulink un régulateur de vitesse sans capteur physique de vitesse de rotation (dit « *sensorless* » en anglais), par estimation de la vitesse de rotation de l'hélice du propulseur (Fig. 1b) à partir du courant d'induit et de la tension d'induit. Cela permettra au contrôle des propulseurs de s'affranchir de la chute progressive de la tension batterie, tout en évitant un régulateur de tension, ce qui alourdirait le robot.

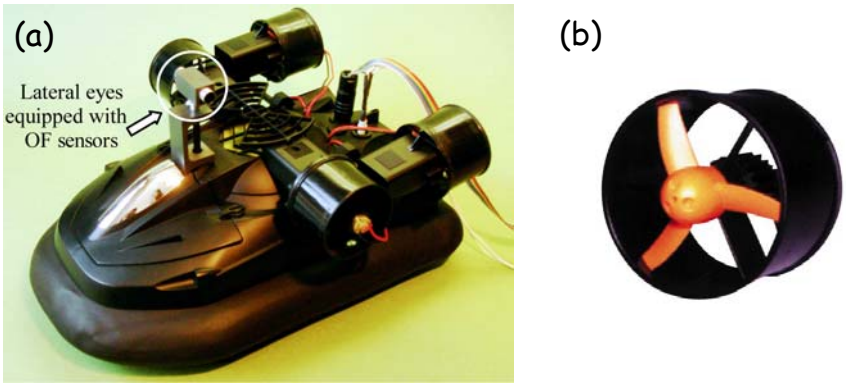

Figure 1. (a) Le mini-robot LORA (36x25x14 cm ; masse 850 g). (b) Propulseur GWS™ de type EDF50 équipé d'un micromoteur à courant continu de référence CN12-RXC. Video du robot LORA : https://www.dailymotion.com/video/xuggrs.

Observations expérimentales :

Lors de son fonctionnement, le mini-robot consomme un courant important faisant chuter la tension de la batterie de 8,2 V à 6,9 V en 15 minutes environ (5 secondes pour les tests en simulation) affectant ainsi la vitesse de rotation du propulseur.

Des mesures expérimentales sur le micromoteur à courant continu ont permis d'établir que :
- sa résistance d'induit R_m = 1 Ω,
- sa constante de temps électrique $\tau_e = L_m/R_m$ = 2 ms,
- sa constante de temps électromécanique du moteur $\tau_{em} = J_m \cdot R_m/K_m^2$ = 16 ms
- sa constante de couplage électromécanique $K_m = 2{,}70 \cdot 10^{-3}$ V/(rad/s)
- sous tension d'alimentation maximale 7,2 V, le micromoteur consomme 3,4 A.
- le couple de frottement sec est nul. Le couple de charge est modélisé par un frottement visqueux.
- le coefficient de frottement visqueux $f = 6{,}52 \cdot 10^{-6}$ SI.

Références à lire et à consulter :

[1] Julien Serres, Stéphane Viollet, Lubin Kerhuel, & Nicolas Franceschini. Régulation de vitesse d'un micromoteur à courant continu sans capteur au moyen d'un microcontrôleur dsPIC programmé par une passerelle Matlab™/Simulink. La Revue 3EI, SEE, 2009, pp.66-74. HAL Id : hal-01848804.
https://hal-amu.archives-ouvertes.fr/hal-01848804

[2] Stéphane Viollet, Lubin Kerhuel, & Nicolas Franceschini. A 1-gram dual sensorless speed governor for micro-air vehicles. In 2008 16th Mediterranean Conference on Control and Automation, IEEE, pp. 1270-1275, June, 2008.
https://www.kerhuel.eu/publi/Viollet_2007_IEEE_IROS.pdf

Les étapes du projet :

1) Déduire des données expérimentales les paramètres électriques et mécaniques du moteur, pour cela vous utiliserez un fichier Matlab que vous exécuterez avant la simulation. Modéliser par un schéma Simulink (à temps continu en utilisant les fonctions de transfert analogique en « s » ; la variable de Laplace étant « s » en anglais au lieu de « p » en français) le moteur à courant continu équipant le propulseur.

2) En utilisant le schéma Simulink (Fig. 2) : réglez, puis testez le bon fonctionnement du moteur à courant continu associé à son hacheur.

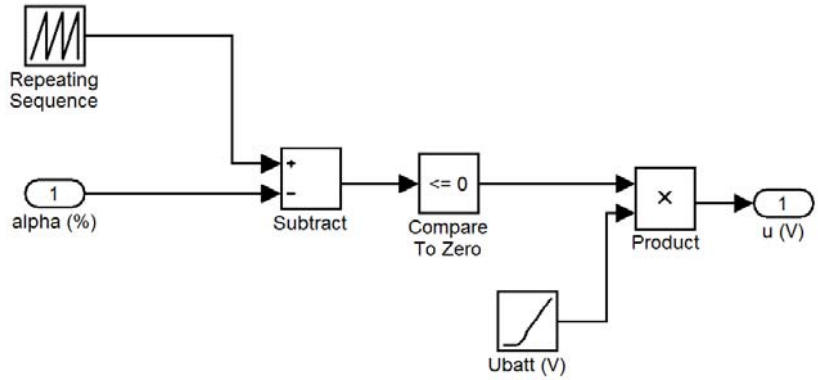

Figure 2. Schéma de simulation Simulink permettant de représenter la fonction du hacheur.

Vous compléterez les blocs de simulation (Fig. 2) dans le but d'obtenir une fréquence de fonctionnement du hacheur de **20 kHz** et une précision de calculs sur la variable de simulation α de 1%, sachant que $\alpha \in [1\%;100\%]$. Vous devez alors pré-calculer le pas de calcul de votre simulation (T_e) permettant d'atteindre cet objectif de précision numérique (à régler dans Simulation -> Configuration Parameters). Vous devrez ensuite paramétrer la rampe de décroissance de la tension batterie en fonction du cahier des charges.

3) Développer une boucle de régulation de vitesse par estimation de la vitesse de rotation (Fig. 3) en complétant votre schéma Simulink construit aux 1) et 2). Vous pourrez vous aider du document d'accompagnement publié dans la revue 3EI [Refs. 1-2]. Une partie de la simulation, i.e., la boucle de régulation, fonctionnera à 1 kHz. Vous devrez alors utiliserez des blocs « Rate Transition » ajustés à 1 kHz pour imposer cette fréquence de calcul aux blocs « Estimateur de vitesse de rotation » et « Correcteur ».

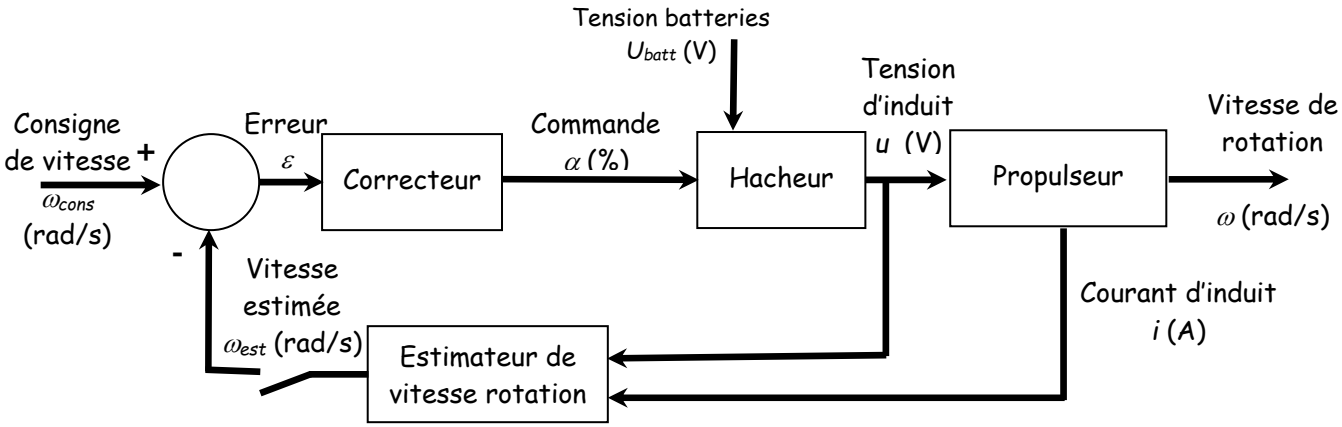

Figure 3. Boucle de régulation de vitesse par estimation de la vitesse de rotation sans capteur (ni optique, ni magnétique, ni électromécanique).

Dans cette étape, vous programmerez dans un premier temps la boucle de régulation en boucle ouverte pour valider le bloc « Estimation de vitesse de rotation » sur la figure 3.

La tension d'alimentation du microcontrôleur dsPIC étant de 3,3 V, alors que la tension d'alimentation U_{batt} est au maximum de 8,2 V, les deux tensions représentant la tension d'induit et le courant d'induit sont divisées par un facteur 3 au moyen d'un diviseur de tension avant numérisation par les entrées CAN[1] du microcontrôleur (Fig. 4).

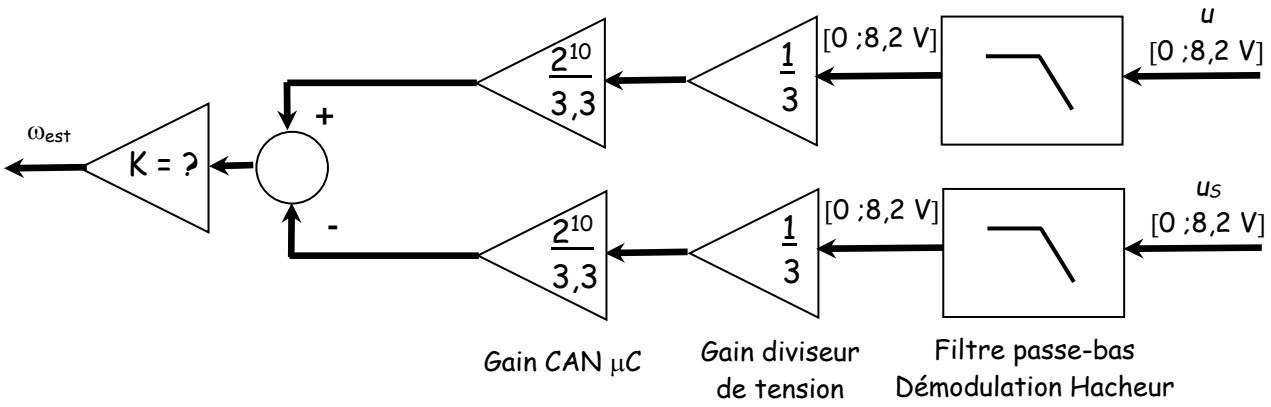

Figure 4. Schéma bloc de principe de l'estimateur de vitesse de rotation à partir de la mesure des tensions u et u_S via deux CAN[1] du microcontrôleur dsPIC.

Pour déterminer la valeur du gain K de l'estimateur de vitesse :
- on placera le système bouclé en boucle ouverte (interrupteur ouvert sur Fig. 3)
- on commandera le hacheur avec un rapport cyclique α = 50% pour se placer au milieu de la gamme de commande, car l'estimateur de vitesse n'est pas parfaitement linéaire.
- on ajustera ensuite la valeur de K manuellement pour rechercher $\omega = \omega_{est}$ en régime permanent.

4) Ajuster les paramètres de votre régulateur numérique PI (à temps discret, donc en « z ») pour obtenir une réponse indicielle en boucle fermée de la boucle de régulation « sensorless » présentant une oscillation avec un dépassement de 15%. Le format des données calculées dans la boucle devra être compatible avec la programmation d'un microcontrôleur de type dsPIC de Microchip© (Fig. 5) dont les caractéristiques utiles sont les suivantes :

- Tension d'alimentation 3,3 V,
- CAN[1] de résolution 10 bits,
- fréquence d'échantillonnage des entrées CAN[1] 1 kHz,
- périphérique de sortie contrôle moteur MLI[2] (« *output compare* ») : 20 kHz représentant la fréquence de hachage.

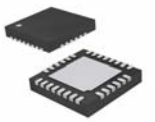

Figure 5. Microcontrôleur dsPIC30F2010 de Microchip© au format QFN de 28 pins (6x6 mm ; 20 MHz ; 3,3 V)

Pour cela, il faut boucler la chaîne de retour et maintenir la chaîne de commande du hacheur ouverte en commandant le hacheur toujours avec α = 50% pour chercher à faire tendre vers 0 la sortie du correcteur PI dont la plage de valeur en sortie sera entre 0% et 100% pour être compatible avec la commande du hacheur avec une précision de 1%, puis de la saturer au moyen d'un bloc « *Saturation* »

[1] CAN = Convertisseur Analogique Numérique
[2] MLI = Modulation en Largeur d'Impulsions

car la commande du hacheur ne peut pas être physiquement supérieure à 100%. Pour cela, il vous faudra ajuster le gain de votre correcteur numérique PI.

5) Tester, en simulation, la robustesse de la boucle de régulation numérique « sensorless » lorsque la tension de la batterie chute au cours du temps pour 3 vitesses de rotation (e.g., 500, 1000, et 1500 rad/s). Pour les besoins de la simulation, on fera chuter linéairement la tension batterie en 5 secondes au lieu de 15 minutes.

⇒ Utilisez la commande Matlab « *sim* » pour lancer votre simulation Simulink à partir de votre script Matlab : `sim('Model_Name',Simulation_Time).`

⇒ Représentez vos résultats de simulation en utilisant les fonctions Matlab : *plot*, *subplot*… Cette partie est importante pour mettre en valeur vos résultats dans votre rapport.

****** FIN ******