

COMMENT ORGANISER UNE FORMATION MATLAB™/SIMULINK

Julien SERRES

julien.serres@univ-amu.fr

Aix Marseille Univ, CNRS, ISM, Marseille, France

Institut des Sciences du Mouvement – Etienne-Jules Marey, Equipe Biorobotique

CP910 - 163, av. de Luminy

13288 MARSEILLE cedex 09

Résumé : Cet article présente un mini-projet de simulation Matlab™/Simulink durant 18 heures de formation destiné à des étudiants en première année d'école d'ingénieurs (niveau bac+3). L'idée est d'introduire l'outil logiciel et les contraintes inhérentes à la simulation telles que nous les rencontrons durant une activité de recherche et développement (R&D). Les mini-projets constituant alors des morceaux de simulation pouvant être contextualisé autour d'une problématique de R&D. L'article développera la pédagogie active mise en place, l'organisation de la formation, et les modalités d'évaluation, pour permettre aux étudiants de devenir autonome à l'issue de leur module ne représentant que 18 heures de formation sur le logiciel expert Matlab™/Simulink.

Mots clés : robotique, moteur à courant continu, contrôle moteur, pédagogie active.

I. ORGANISATION DE LA FORMATION

Depuis maintenant 10 ans, j'anime une formation de 18 heures au logiciel Matlab™/Simulink (The MathWorks, [1]) sous forme de mini-projet au sein d'une école d'ingénieur [2]. Le module étant placée au début du cycle de formation de première année, les étudiants ont peu ou pas de connaissance en programmation et le traitement des signaux numériques. J'ai alors du imaginer une pédagogie plus active pour augmenter l'autonomie et le niveau des étudiants par une formation magistrale réduite, puis complètement décentralisée.

Le module de 18 heures s'organise sous la forme de 6 séances de 3 heures. L'effectif de la promotion de première année est de 72 étudiants, répartis sous forme de 3 groupes de 24 étudiants encadrés par 2 enseignants. Dans le cadre de la virtualisation des logiciels dédiés à la formation et à l'autonomie informatique des étudiants, ces derniers travaillent sur leur propre ordinateur portable durant les séances. Les étudiants sont donc présents dès la première séance avec leur propre ordinateur comprenant tous les logiciels nécessaires à leur formation. Les 6 séances de travail s'organisent de la manière suivante :

Séance 1

- présentation des 3 mini-projets de simulation (0h20) comme étant des sous-ensembles d'un projet intégrateur,
- présentation de l'environnement de travail, des boîtes à outil associées du logiciel Matlab, et de la syntaxe du langage Matlab (1h30),
- présentation de la programmation graphique au moyen de l'outil Simulink (1h00),
- présentation des contraintes temporelles de la réalisation du mini-projet et des critères d'évaluation (0h10),

Séance 2

Programmation collective sous forme de TD d'une étape de chacun des mini-projets pour améliorer la compréhension de la finalité de chacun des mini-projets et traiter de problèmes de programmation appliqués à un contexte concret (celui de la R&D).

Séance 3 à 5

Programmation individuelle sous le logiciel Matlab™/Simulink des mini-projets. Les étudiants avancent à leur rythme, une interaction individuelle ou par petit groupe est privilégiée avec les enseignants sous forme de tutorat.

Séance 6

Achèvement de la programmation des mini-projets sous le logiciel Matlab™/Simulink. Présentation par binôme de leur solution de programmation à l'oral durant 7 minutes plus 5 minutes de questions. Suite à cet oral, une première note sur 10 points est établie, complétée par une note de rapport sur 10 points dont la taille maximale est fixée à 6 pages maximum sans les annexes.

I.1 Les 3 sujets de mini-projets

Pour éviter de tomber dans un mode d'enseignement collectif et séquentiel, il n'est apparu indispensable de proposer plusieurs sujets issus de mes travaux de recherche [3]. Pour se faire, je propose 3 sujets aux étudiants, ces derniers s'organisant pour choisir l'un des 3 sujets en fonction de leur centre d'intérêt par binôme. Dans chaque groupe, 8 étudiants travaillent sur le même sujet, et forme ainsi 4 équipes/binômes (à multiplier par 3 car il y a 3 sujets).

Les 3 sujets de mini-projets de simulation sont les suivants :

- **mini-projet 1 :** régulation de vitesse d'un propulseur sans capteur mécanique, ce sujet repose sur les références [4,5],

- **mini-projet 2** : programmation de l'interaction des photodiodes avec un environnement visuel 1D, ce sujet repose sur les références [3,6],
- **mini-projet 3** : capteur de mouvement bio-inspiré, ce sujet repose sur la référence [7] et exploite des signaux réels photorécepteurs issus d'une expérience de R&D.

I.2 L'évaluation des mini-projets

Deux items de notation permettent d'évaluer le mini-projet, chacun comptant pour une moitié :

- **Une soutenance orale lors de la dernière séance de projet** : 7 minutes par binôme suivie d'un entretien de 5 minutes permettant de poser quelques questions en lien avec les choix de programmation Matlab™/Simulink des étudiants.
- **La rédaction d'un rapport de 6 pages** (sans les éventuelles annexes) par binôme permettant d'évaluer la capacité des étudiants à produire des résultats de manière synthétique, notamment par la confection de figures à partir du logiciel Matlab™. Les étudiants élève-ingénieur ayant un niveau de compétences en langue anglaise proche du niveau européen B2, depuis l'année dernière, je demande à ce que leur rapport soit rédigé en anglais.



Figure 1. Robot LORA (taille : 0,36x0,21x0,17 m, masse : 878 g). Le robot est équipé de deux packs d'accumulateurs lithium polymère distincts : un pack faible puissance 7,2 V - 360 mAh ; taille 54x31x7 mm ; masse 20 g pour l'électronique, et un pack de forte puissance 7,2 V - 2200 mAh ; 21x33x96 mm, masse 112 g pour la propulsion procurant au robot une autonomie de 10 minutes [10,11]. Le robot est équipé de petits yeux relié à des capteurs de mouvement bio-inspirés, une partie du traitement de l'information visuelle pour extraire l'information de mouvement, appelé flux optique, est abordée dans les deux autres mini-projets.

II. OBJECTIFS D'UN DES 3 MINI-PROJETS

II.1 Description du projet « sensorless »

Les mini-robots autonomes (Fig. 1) puisent l'énergie électrique pour se propulser dans des batteries de type lithium polymère [8]. Au cours de leur utilisation et de leur vieillissement [9], on observe une chute progressive de la tension batterie, posant ainsi un problème majeur pour stabiliser le contrôle moteur de ces mini-robots. Une solution simple consisterait à mettre en place une carte de régulation pour stabiliser la tension d'alimentation du système de propulsion, néanmoins

cette solution a été écartée compte tenu des contraintes de masse et d'encombrement de la microrobotique, d'autant plus importante en microrobotique aérienne. ce mini-projet appelé « sensorless » propose de montrer aux étudiants que l'on peut supprimer cet étage de régulation de tension pour venir piloter la vitesse de rotation d'un micromoteur à courant continu sans avoir recours à une régulation de tension ni à la mise en place d'un capteur de rotation de l'arbre moteur.

L'objectif de ce mini-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. 2) à partir de la mesure conjointe du courant et de la tension d'induit. Cela permet ainsi de s'affranchir de la chute progressive de la tension batterie, tout en évitant un régulateur de tension, ce qui alourdirait le robot.



Figure 2. Propulseur GWS™ de type EDF50 équipé d'un micromoteur à courant continu de référence CN12-RXC. Ce propulseur est capable de développer une poussée de 72 g sous 7,2 V.

II.2 Observations expérimentales

Lors de son fonctionnement, le mini-robot (Fig. 1) consomme un courant important faisant chuter la tension de la batterie de 8,2 V à 6,9 V en ~10 minutes (durée que l'on placera à 5 secondes pour les tests en simulation) affectant ainsi la vitesse de rotation des 4 propulseurs équipant le robot (Fig. 2). Des mesures expérimentales sur le micromoteur à courant continu ont permis d'établir que :

- sa résistance d'induit : $R_m = 1 \Omega$,
- sa constante de temps électrique : $\tau_e = L_m/R_m = 2 \text{ ms}$,
- sa constante de temps électromécanique : $\tau_{em} = J_m \cdot R_m / (f \cdot R_m + K_m^2) = 23,5 \text{ ms}$ permettant d'obtenir un temps de montée proche des 52 ms comme observé lors des essais de mesure [5],
- sa constante de couplage électromécanique : $K_m = 2,70 \cdot 10^{-3} \text{ V}/(\text{rad/s})$
- sous tension d'alimentation maximale 7,2 V, le micromoteur consomme 3,4 A.
- le couple de frottement sec T_r est nul. Le couple de charge est modélisé par un frottement visqueux. Le coefficient de frottement visqueux est évalué à $f = 6,52 \cdot 10^{-6} \text{ SI}$.

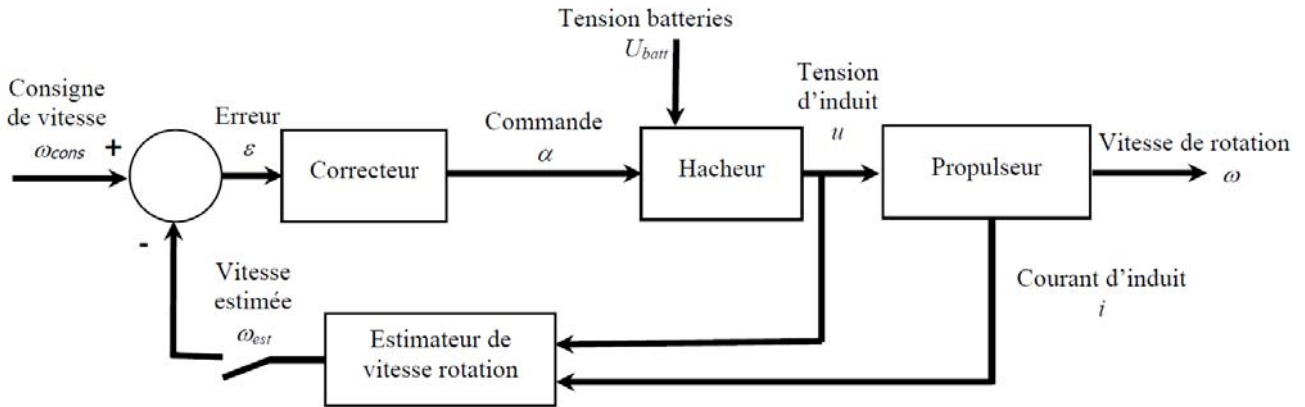


Figure 3. Schéma bloc de la simulation numérique pour réguler la vitesse de rotation d'une hélice sans capteur optique, ni électromécanique. Cette boucle de régulation est dite sans capteur ou « sensorless ». L'ouverture et la fermeture permet dans un premier temps de valider le fonctionnement du bloc « estimateur de vitesse de rotation », puis dans un second temps, en position fermée de régler le correcteur PI (Proportionnel-Intégral).

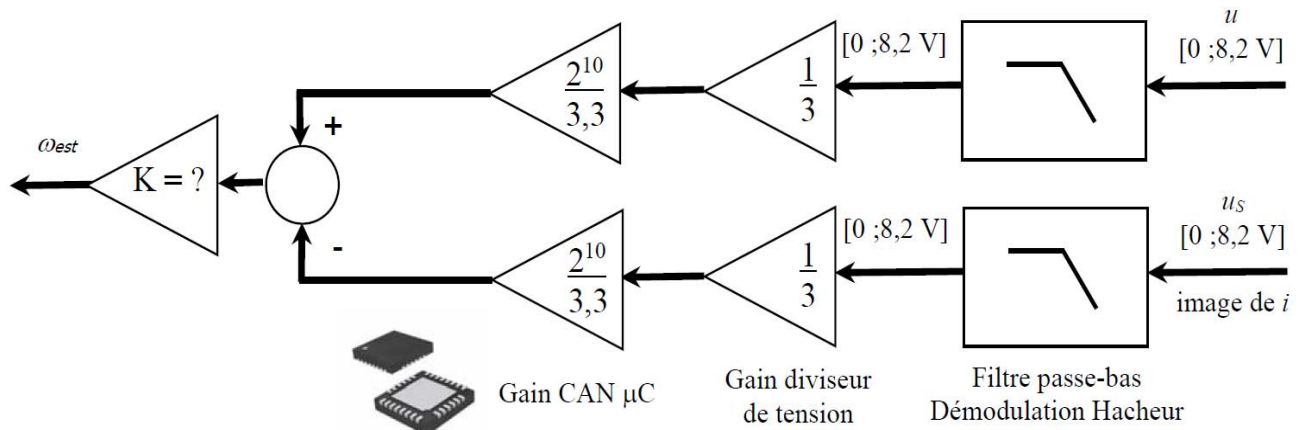


Figure 4. Chaîne de traitement de l'estimateur de vitesse de rotation à partir de la mesure des tensions u et u_s via trois potentiels électriques issus des entrées CAN du microcontrôleur de type dsPIC. La référence du dsPIC est un microcontrôleur dsPIC30F2010 de Microchip® au format QFN de 28 pins (6x6 mm ; 20 MHz ; 3,3 V). La tension u_s représente la tension shunt aux bornes d'une résistance R_s montée en série avec le moteur, la tension u_s est donc directement une tension image du courant d'induit i . Dans [4,5], on montre que l'estimateur fonctionne de manière optimale lorsque R_s est égale à la résistance d'induit R_m . Le gain K est un gain à faire régler par les étudiants ainsi que la fréquence du coupure du filtre passe-bas du premier ordre incarné physiquement par un circuit RC [4,5].

II.3 Cinq étapes pour réaliser son projet

Chaque mini-projet est décomposé sous la forme de 5 grandes étapes pour permettre à l'étudiant de mener à bout sa simulation. Chaque étape nécessitant des efforts d'abstraction pour bien régler tous les paramètres inhérents à une simulation numérique.

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 le moteur à courant continu équipant les propulseurs du robot en utilisant les fonctions de transfert analogique en « s » ; la variable de Laplace étant « s » en anglais au lieu de « p » en français).

2) En utilisant le schéma Simulink (Fig. 3) : réglez, puis testez le bon fonctionnement du moteur à courant continu associé à son hacheur. Vous complèterez les

blocs de simulation (Fig. 3) dans le but d'obtenir une fréquence de fonctionnement du hacheur de 10 kHz (à ajuster dans l'onglet Simulation → Configuration Parameters) et une précision de calculs sur la variable représentant le rapport cyclique α à 1% pilotant le micromoteur, sachant que $\alpha \in [1\% ; 100\%]$. La rampe de décroissance de la tension batterie est à paramétrer en fonction du cahier des charges. 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 Fig. 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 et le courant d'induit sont divisées par un facteur 3 au moyen d'un diviseur de tension avant numérisation par les CAN¹ du microcontrôleur (Fig. 4).

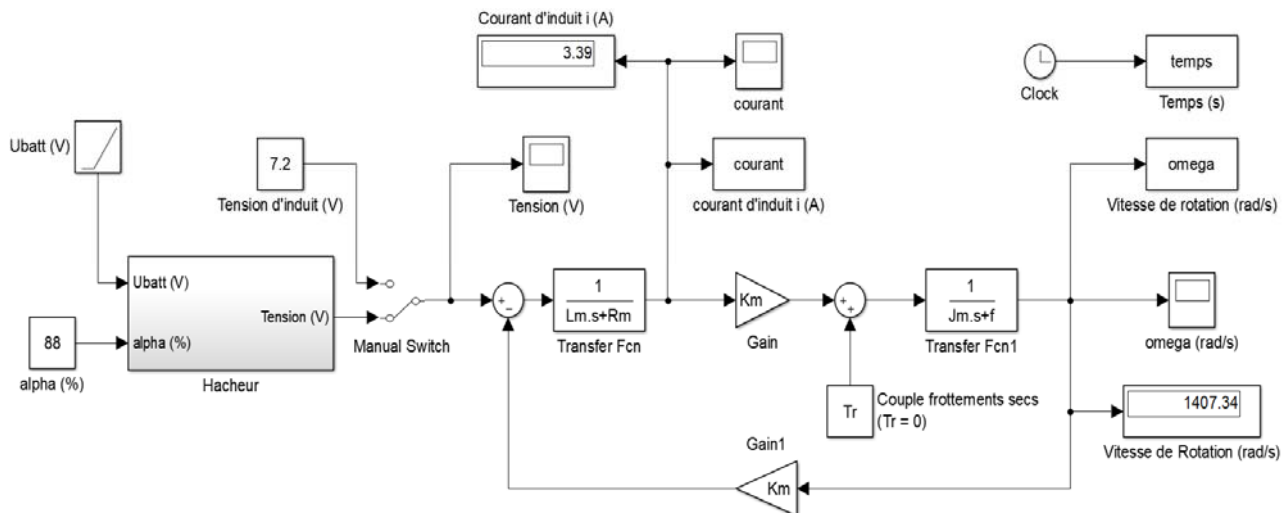


Figure 5. Schéma Simulink du bloc Propulseur (Fig. 1) couplé au bloc Hacheur détaillé sur la Fig. 6. Les paramètres du modèle dynamique du moteur à courant continu sont : $L_m = 2\text{mH}$; $R_m = 1\Omega$; $J_m = 3,25 \cdot 10^{-7} \text{ kg}\cdot\text{m}^2$; $f = 6,52 \cdot 10^{-6} \text{ SI}$; $K_m = 2,70 \cdot 10^{-3} \text{ V}/(\text{rad/s})$.

3) Développer une boucle de régulation de vitesse par estimation de la vitesse de rotation (Fig. 3) en complétant votre schéma Simulink construit lors des étapes 1) et 2). Pour simuler le système Hacheur+Propulseur, nous imposerons une fréquence d'échantillonnage de 10 kHz, donc cette fréquence est à configurer dans les paramètres de Simulink. La boucle de régulation, quant à elle, fonctionnera à 1 kHz. Pour cela, vous utiliserez des blocs « Rate Transition » et ajusterez manuellement à 1 kHz les blocs de la boucle travaillant à cette fréquence.

Pour déterminer la valeur du gain K de l'estimateur de vitesse :

- on placera le système en boucle ouverte (interrupteur ouvert sur Fig. 3)
- on commandera le hacheur avec $\alpha = 50\%$ 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 du gain 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 ») compatible avec la programmation d'un microcontrôleur de type dsPIC de Microchip® dont les caractéristiques utiles sont les suivantes :

- Tension d'alimentation 3,3 V,
- CAN¹ de résolution 10 bits,
- fréquence d'échantillonnage des CAN 1 kHz,
- Calculs effectués en nombres entiers 16 bits,
- périphérique de sortie contrôle moteur MLI² (« output compare ») : 10 kHz

Pour cela, il faut boucler la chaîne de retour et maintenir la chaîne de commande du hacheur ouverte en commandant le hacheur toujours avec $\alpha = 50\%$ pour chercher à faire tendre vers 0 la sortie du correcteur PI dont la plage de valeur en sortie sera calibrée 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 car la commande du hacheur ne peut pas être physiquement supérieure à 100%.

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 valider votre simulation et réduire le temps de simulation, on fera chuter linéairement la tension batterie en 5 s au lieu de 15 min. Il vous appartiendra de représenter vos résultats de simulation en utilisant les fonctions graphiques du logiciel Matlab (plot, subplot...).

III. RESULTATS ATTENDUS PAR ETAPE

III.1 L'ETAGE PROPULSEUR

Le modèle dynamique du moteur à courant continu est bien décrit dans [5,12]. Toutes les équations électriques et mécaniques sont représentées sous forme de fonction de transfert dans le domaine de Laplace (Fig. 5). Lorsque le moteur est alimenté sous tension maximale 7,2 V (ou bien avec un rapport cyclique $\alpha = 88\%$ via le hacheur), le micromoteur consomme bien 3,4 A avec un temps de montée t_m de 52 ms (i.e., $t_m = 2,2 \cdot \tau_{em}$).

III.2 L'ETAGE HACHEUR

La manière de plus simple de programmer un hacheur sous Simulink est représenté sur la Fig. 6. Le bloc « Repeating Sequence » permet de programmer la modulante à $f_H = 10 \text{ kHz}$, pour se faire il faut régler le paramètre « Times values » à $[0 \text{ e-}4]$ correspondant à

¹ CAN = Convertisseur Analogique / Numérique

² MLI = Modulation en Largeur d'Impulsions

la période souhaitée du signal triangulaire exprimée en secondes, et le paramètre « Output values » à [1 100], correspondant à une gamme de 100 valeurs de rapport cyclique α . Ensuite, on souhaite obtenir une précision de 1% sur le calcul du rapport cyclique. Il faut alors considérer 100 points de calcul par période pour positionner le rapport cyclique α à 1% près. Cela revient donc à diviser par 100 la fréquence de la modulante f_H . Pour se faire, il faut ajuster le Solver de Simulink (onglet Simulation → Configuration Parameters) :

- Type : fixed-step, i.e., simulation à pas fixe,
- Solver : ode1 (Euler), i.e., déclare la méthode d'Euler pour numériser les équations déclarées à temps continu, notamment représentées par les fonctions de transfert représentées dans le domaine de Laplace pour le propulseur,
- Fixed-step size : 1e-6, i.e., 1 μ s pour le pas de calcul fondamental qui sera le plus petit pas de calcul envisageable dans cette simulation numérique.

Enfin la batterie est représentée par une fonction « Ramp » de coefficient directeur -0,26 V/s et d'ordonnée à l'origine de 8,2 V pour répondre au modèle de décroissance de la batterie durant les 5 secondes de simulation.

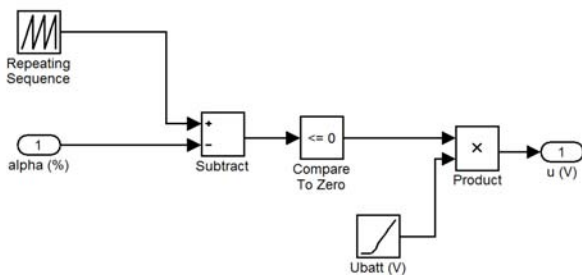


Figure 6. Schéma Simulink modélisant le hacheur.

III.3 L'ETAGE ESTIMATEUR

L'estimateur de vitesse de rotation dit « sensorless » ou en « U - R-I » comprend tout d'abord un filtre passe-bas analogique de Butterworth du 1^{er} ordre pour filtrer les harmoniques générées par le hacheur afin de ne conserver que la valeur moyenne des potentiels électriques liés à u et u_S (Fig. 4, Fig. 7a). En choisissant une fréquence de coupure $f_C = 100$ Hz pour éliminer la modulante du hacheur à $f_H = 10$ kHz, nous obtenons une estimation peu bruitée et faiblement déphasée par rapport à la vitesse de rotation réelle de l'hélice (Fig. 7b). Il s'en suit un gain diviseur de tension d'une valeur de 1/3, car le microcontrôleur dsPIC possède une plage de mesure de tension allant jusqu'à 3,3 V correspondant à sa tension d'alimentation, alors que les tensions d'induit u et de shunt u_S varient entre 0 V et 8,2 V. Enfin on place un gain modélisant le CAN 10 bits d'une valeur $2^{10}/3,3$; suivi d'un bloc « Rate Transition » permettant d'échantillonner les signaux analogiques (en noir sur la Fig. 7a) à une fréquence de 1 kHz correspondant à la fréquence de travail du microcontrôleur (en vert sur la Fig. 7a). On peut observer l'effet de l'échantillonnage et de la quantification sur l'estimée de la vitesse de rotation (Fig. 7b). En boucle ouverte, le gain K (Fig. 4, Fig. 7a)

en sortie du soustracteur est ajusté de telle sorte que l'on obtienne une vitesse de rotation estimée en régime permanent identique à celle obtenue en sortie du bloc « Propulseur » (i.e., vitesse de rotation réelle).

Pour $\alpha = 50\%$ (car l'estimateur de vitesse de rotation n'est pas parfaitement linéaire, certains étudiants proposent de mettre en œuvre une table de correspondance entre la valeur de α et le gain K pour rendre linéaire l'estimateur), le meilleur réglage est $K=3,7$ (Fig. 7b). Le profil de l'hélice (Fig. 2) étant asymétrique, l'estimateur ne fonctionne que pour des vitesses positives. Le gain K est ajusté pour la valeur numérique de l'entier positif représentant la vitesse de rotation estimée corresponde à la vitesse réelle exprimée en rad/s.

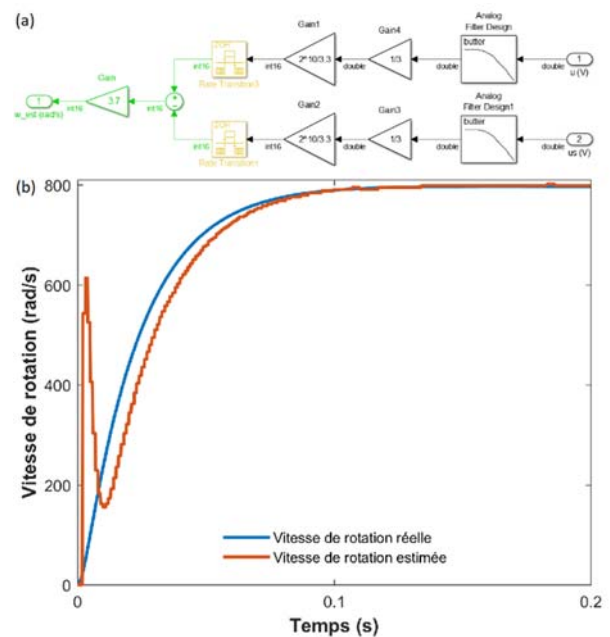


Figure 7. (a) Schéma Simulink de l'estimateur. (b) Comparaison entre les vitesses de rotation réelle et estimée permettant de régler le gain de l'estimateur K sur la Fig. 4.

III.4 L'ETAGE CORRECTEUR

Un correcteur numérique PI (Proportionnel-Intégral) mixte est introduit dans la boucle de par sa simplicité de réglage (Fig. 8a), uniquement un gain proportionnel K_P sera réglé pour calculer le rapport cyclique commandant le hacheur. Le critère retenu pour le réglage du correcteur est libre, mais il est proposé de régler pour obtenir un amortissement de 0,5 correspondant à une seule oscillation de la vitesse de rotation. Cela revient à régler le premier dépassement de la vitesse de rotation à 15% de la consigne de vitesse. Pour une consigne de vitesse de 1000 rad/s (Fig. 8a), on règle le gain K_P dans le gain représentant la fonction de transfert du correcteur PI pour atteindre l'objectif de réglage. La valeur de 0,0021 pour K_P permet d'obtenir un dépassement de la vitesse de rotation limité à 15 % de la valeur finale (soit 1150 rad/s, Fig. 8b).

On limite ensuite le rapport cyclique α en sortie du correcteur PI à l'aide d'un bloc « Saturation » pour garantir une commande de hacheur dans l'intervalle [1% ; 100%], puis on place un dernier bloc « Rate Transition » pour transformer le signal α cadencé à 1 kHz (fréquence de calcul du microcontrôleur) en un signal analogique cadencé à 1 MHz (fréquence de calcul du système dynamique micromoteur à courant continu).

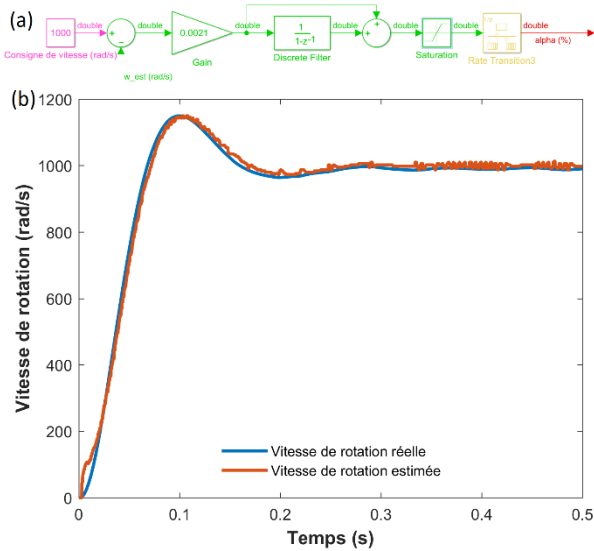


Figure 8. (a) Schéma Simulink du correcteur PI. (b) Comparaison entre les vitesses de rotation réelle et estimée pour un réglage du gain du correcteur correspondant à un dépassement de 15% de la vitesse de rotation. Le temps de montée en boucle fermée est mesuré ici à 45 ms, contre 52 ms en boucle ouverte.

III.5 FONCTIONNEMENT GLOBAL

La figure 9 représente le schéma Simulink global de la simulation en boucle fermée. La figure 10 représente les résultats pour une consigne de vitesse de rotation de 1000 rad/s, on observe sur la Fig. 10a une oscillation avec un dépassement de 15% vis-à-vis d'une consigne fixée à 1000 rad/s en dépit de la chute de tension batterie observée sur la Fig. 10c. Le pilotage dit « sensorless » du micromoteur à courant d'induit permet de maintenir une tension d'induit moyenne constante en dépit de la chute de tension batterie (Fig. 10c). En effet, on observe que lorsque la tension batterie chute au cours du temps, le rapport cyclique α augmente pour compenser la chute de valeur moyenne de tension d'induit. Ainsi, en assurant une tension d'induit moyenne constante aux bornes du micromoteur, on peut mieux exploiter l'énergie électrique disponible dans la batterie tout en rendant invariant le pilotage du propulseur vis-à-vis de la chute de tension batterie sans pour autant introduire un régulateur de tension à bord du microrobot.

IV. CONCLUSIONS

IV.1 FACILITE LA SYNTHÈSE DE CORRECTEUR

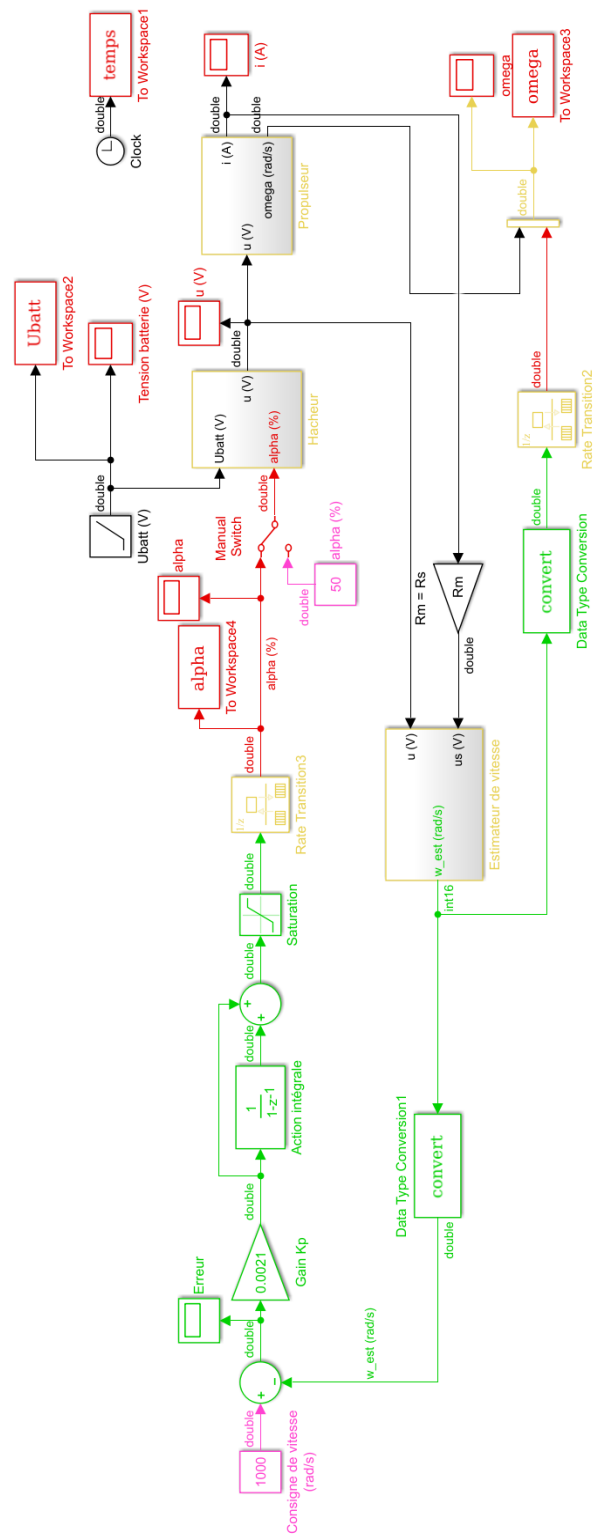


Figure 9. Schéma Simulink complet de la boucle d'asservissement « sensorless » dont la consigne de vitesse de rotation est ici fixée à 1000 rad/s. Les flux de données en vert représentent des calculs cadencés à 1 kHz comme embarqué ultérieurement sur la cible matérielle. Les flux de données en rouge ou en noir représentent des calculs à 1 MHz correspondant aux blocs analogiques de la simulation numérique, cette fréquence est fixée par la précision du calcul du rapport cyclique α fixée à 1%.

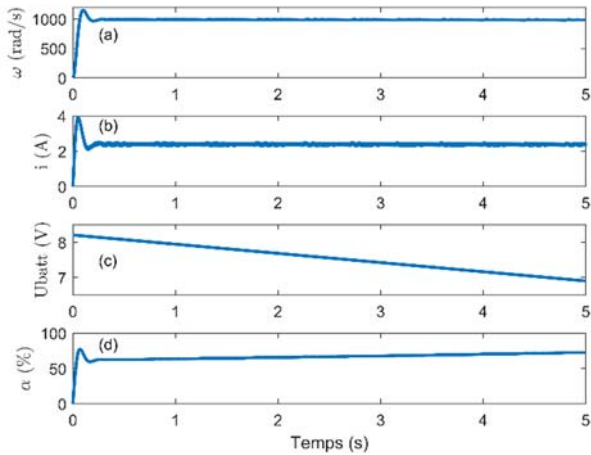


Figure 10. (a) Vitesse de rotation ω en fonction du temps. (b) Courant d'induit i en fonction du temps. (c) Tension batterie U_{batt} en fonction du temps qui passe de 8,2 V à 6,9 V en 5 secondes. (d) Rapport cyclique α commandant le hacheur.

L'utilisation de logiciel MatlabTM/Simulink permet de simuler tout procédé de commande par ordinateur en respectant les contraintes inhérentes à l'échantillonnage (cadence des calculs) et à la quantification (format des données). Simuler de telles contraintes permet d'ajuster des correcteurs numériques sans pour autant faire appel à la synthèse de filtres numériques dont la démarche est parfois bien trop théorique en début de formation d'un cycle ingénieur ou premier cycle universitaire. La simulation MatlabTM/Simulink est alors utile pour régler des correcteurs. Dans bien des situations d'apprentissage, les étudiants doivent retranscrire le correcteur optimisé sous une autre forme (analogique, numérique, ou bien d'autres langages de programmation), ce qui pénalise l'intérêt de la simulation MatlabTM/Simulink, alors même que son intérêt réside dans la diminution du temps de mise en œuvre de la loi de commande sur le ordinateur.

IV.2 PASSERELLE MATLABTM/SIMULINK – CIBLE MATERIELLE

Une boîte à outils « Simulink – Embedded Target for PIC » [13] permet alors de programmer directement une cible matérielle de Microchip© à partir d'un schéma Simulink [5] lorsqu'on souhaite travailler sur un système embarqué. La simulation MatlabTM/Simulink peut aussi être directement coupler une carte E/S dSPACE en utilisant la boîte à outils « Real-Time Interface » [14]. Les correcteurs numériques optimisés sous MatlabTM/Simulink peuvent aussi être mis en œuvre à l'aide du logiciel LabVIEW et d'une carte E/S NI-DAQ USB 6009 [15], ce qui revêt une solution bien moins onéreuse qu'une carte E/S dSPACE couplé au logiciel ControlDesk©. Aujourd'hui, de multiples solutions existent pour programmer une cible matérielle directement à partir d'une simulation MatlabTM/Simulink.

IV.3 INTERET PEDAGOGIQUE

En termes de formation logicielle, il m'apparaît aujourd'hui essentiel de former les étudiants à des

logiciels experts dans le cadre de projets où le logiciel expert est utilisé comme un outil permettant de mener à bien un projet, et non une juxtaposition d'exercices de programmation. Sa prise en main devant se résumer qu'à quelques heures de formation pour comprendre l'environnement logiciel et la syntaxe. En effet, la disponibilité massive d'information sur un logiciel expert (Internet, Wiki, Blogs, Documentation en ligne) rend aujourd'hui complètement caduc l'enseignement de tels contenus. Cette pédagogie active fixe des objectifs en termes de maîtrise logicielle à travers les différentes étapes de développement du projet de simulation, ce qui implique davantage les étudiants dans leur projet et libère leur créativité. La production étudiante est alors plus importante hors temps scolaire, ce qui permet pendant le temps de formation d'avoir un rôle de tuteur pour guider les choix des étudiants. Une restitution sous la forme d'un court rapport de 6 pages couplé à un court oral de 12 minutes durant la dernière séance, s'avère être très pertinent pour valider le niveau de maîtrise logicielle des étudiants lors de la démonstration de leur simulation.



Figure 11. Quadrirotor Crazyflie 2.0 équipé de quatre micromoteurs à courant continu [16]. Dans sa version commerciale, la chute de tension batterie n'est pas compensée au cours du vol, limitant ainsi son vol à seulement 7 minutes.

IV.3 PERSPECTIVES

En termes de continuité pédagogique, les micromoteurs à courant continu sont aujourd'hui omniprésents en microrobotique aérienne (Fig. 11), on peut citer par exemple le micro quadrirotor Crazyflie 2.0 pesant seulement 27 grammes pour une autonomie de vol de 7 minutes [16]. Dans sa version commerciale, la chute de tension batterie n'est pas compensée au cours de son vol, limitant ainsi son autonomie de vol, le but de ce nouveau projet serait donc d'intégrer quatre contrôleurs « sensorless » à bord du Crazyflie 2.0 (Fig. 10) pour venir accroître son autonomie de vol. A cet effet, il a été démontré qu'une paire de contrôleur « sensorless » pèse seulement 1 g pour une taille de 13x15 mm, et peut fournir un courant maximal de 2 A sous 8 V [4,5], ce qui est compatible avec les micromoteurs à bord du Crazyflie 2.0, qui consomme unitairement 1 A sous 4,2 V. En ajoutant deux paires de

contrôleur « sensorless » à bord du Crazyflie 2.0, cela représenterait alors une masse supplémentaire de seulement 2 g, alors que le Crazyflie 2.0 peut porter une charge utile allant jusqu'à 15 g. La mise en œuvre de ce

type de projet permettrait alors de démontrer toute la potentialité du prototype rapide sous le logiciel Matlab™/Simulink.

REFERENCES BIBLIOGRAPHIQUES

- [1] <https://fr.mathworks.com/>
- [2] <http://www.mines-st-etienne.fr/fr/ISMIN>
- [3] Serres J., "De l'abeille au robot : la régulation du flux optique - Contrôle conjoint de vitesse et d'évitements » d'obstacles latéraux pour véhicules totalement actionnés", Thèse de doctorat, Université de Montpellier II - Sciences et Techniques du Languedoc, 2008. Ref. HAL : tel-00645272
- [4] Viollet S., Kerhuel L., Franceschini N., "A 1-gram dual sensorless speed governor for micro-air-vehicles", Proc. of the 16th IEEE Mediterranean Conference on Control and Automation (MED), Ajaccio, France, June 25-27, 2008.
- [5] Serres J., Viollet S., Kerhuel L., Franceschini N., "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, No 56, pp. 66-74, Mars 2009. Ref. HAL : hal-01848804.
- [6] Ruffier, F., "Pilote automatique biomimétique - Système générique inspiré du contrôle visuomoteur des insectes pour : le décollage, le suivi de terrain, la réaction au vent et l'atterrissage automatiques d'un micro-aéronef", Thèse de doctorat, Institut National Polytechnique de Grenoble - INPG, 2004. Ref. HAL: tel-00696454.
- [7] Expert F., Viollet S., Ruffier F., "Outdoor Field Performances of Insect-Based Visual Motion Sensors", Journal of Field Robotics, Vol 29, No 4, pp. 529-541, 2011. Ref. HAL : hal-00712699.
- [8] Broussely M., "Les accumulateurs lithium-ion, état de l'art", La Revue 3EI, No 48, pp. 36-45, Mars 2007.
- [9] Pelissier S., "Introduction au vieillissement des batteries lithium-ion et aux protocoles expérimentaux de test de durée de vie", La Revue 3EI, No 68, pp. 17-22, Avril 2012.
- [10] Roubieu F. L., Serres J., Franceschini N., Ruffier F., Viollet S., "A fully-autonomous hovercraft inspired by bees: wall following and speed control in straight and tapered corridors", Proc. of the IEEE International Conference on Robotics and Biomimetics (ROBIO) 2012, Guangzhou, China, pp. 1311-1318. Ref. HAL : hal-00743129.
- [11] Roubieu F. L., Serres J. R., Colonnier F., N Franceschini N., Viollet S., Ruffier F., "A biomimetic vision-based hovercraft accounts for bees' complex behaviour in various corridors", Bioinspiration & Biomimetics, Vol 9, No 3, 036003. Ref. HAL : hal-01446797.
- [12] Roussel J-M., Rebeix P., Emile B., "Régulation de vitesse d'un moteur à courant continu alimenté par un variateur industriel DMV2342", La Revue 3EI, No 68, pp. 46-50, Avril 2012.
- [13] Simulink toolbox - Embedded Target for PIC : [http://www.kerhuel.eu/wiki/Simulink - Embedded Target for PIC](http://www.kerhuel.eu/wiki/Simulink_-_Embedded_Target_for_PIC)
- [14] Roussel J-M., Emile B., Perronnin E., "Identification et régulation d'une alimentation à découpage forward dans l'environnement Matlab/Simulink/dSPACE", La Revue 3EI, No 72, pp. 59-64, Avril 2013.
- [15] Ilary J-P., Le Gall F., "Séance d'essais systèmes en BTS électrotechnique sur le thème « Asservissement & Labview d'une MCC » ", La Revue 3EI, No 80, pp. 43-46, Avril 2015.
- [16] Micro quadrirotor Crazyflie 2.0, coût unitaire 210€ <https://store.bitcraze.io/collections/kits/products/crazyflie-2-0>