

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

0) Introduction :

Le minirobot LORA est propulsé par une hélice, équipé d'un micromoteur à courant continu lui-même alimenté par une batterie. C'est un système de contrôle embarqué. Le mini-projet 1 traite la régulation de vitesse du propulseur sans capteur de vitesse. Il s'agit d'établir avec MatLab® un modèle de simulation permettant à la fois de chiffrer les paramètres et de tester les limites du système de régulation du robot.

1) Paramètres électriques et mécaniques du moteur :

Le moteur du minirobot est un micromoteur à courant continu (CN12-RXC). C'est une machine électrique, ainsi qu'un convertisseur électromécanique qui convertit l'énergie électrique en énergie mécanique (en fonctionnement moteur) pour le propulseur qui est un système mécanique.

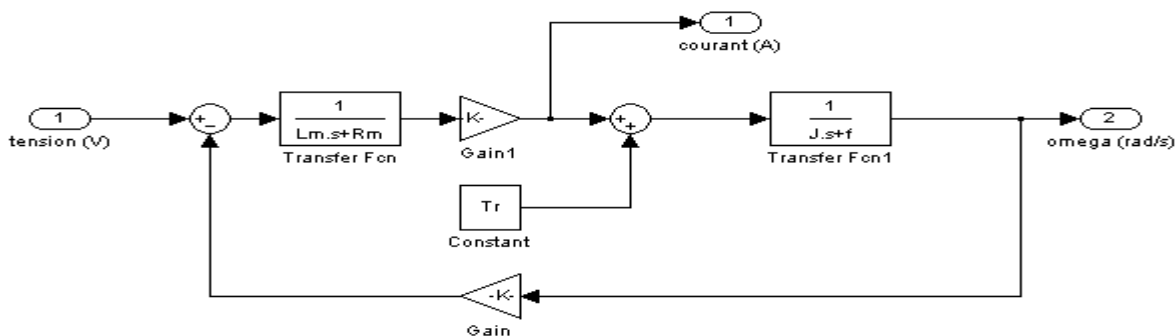
La machine à courant continu est réversible.

Un MCC est composé :

- D'une carcasse en fonte ou en acier renfermant le circuit magnétique du stator et le circuit inducteur bobiné ou à aimants permanents.
- D'un rotor constitué d'un circuit magnétique entièrement feuilleté.
- D'un ensemble de balais graphite ou en métal permettant de fournir le courant à l'induit.

En robotique, le rotor ne renferme plus de matériaux ferromagnétiques ce, afin de minimiser le moment d'inertie et l'inductance du rotor pour obtenir de fortes accélérations et des vitesses élevées.

Principe de fonctionnement : Lorsqu'un conducteur, qui est placé dans un champ magnétique, est parcouru par un courant, il est alors soumis à une force de Laplace. Il se déplace donc dans le champ magnétique et est le siège d'une force électromotrice (f.e.m.) induite par le déplacement. En fait cette f.e.m s'oppose à la tension d'alimentation en cherchant à réduire la courant, il s'agit donc d'une force contre électromotrice. On modélise le moteur sous Simulink® pour qu'il réponde aux équations qui suivent.



Les paramètres du moteur que nous allons utiliser ont été mesurées, elles sont donc les constantes du problème. Elles sont comme suit :

$R_m = 1\Omega$ Résistance d'induit, $L_m = 2\text{mH}$ Inductance pour avoir une constante de temps électrique $\zeta_e = 2\text{ms}$
 $K_m = 2.7\text{mVs/rad}$ constante de couplage électromécanique, $J_m = 116.64\text{pkg.m}^2$ constante d'inertie pour avoir une constante de temps électromécanique $\zeta_{em} = 16\text{ms}$
 $F = 6.52\mu$ coefficient de frottement visqueux

C'est aussi un système asservi, la grandeur de sortie (Omega) est comparée en permanence à la grandeur d'entrée, appelée grandeur de consigne en tension (V). La sortie est asservie à la grandeur d'entrée. C'est une régulation de vitesse : la grandeur de sortie s'aligne sur une consigne du fait même de la conception du moteur.

Le bloc moteur est composé d'une chaîne directe et d'une chaîne de retour.

La fonction de transfert de la chaîne directe est :
$$\frac{K}{(L_m.s + R_m).(J_m.s + f)} = \frac{K}{L_m.J_m.s^2 + (L_m.f + R_m.J_m).s + R_m.f}$$

Nota :

La variable de Laplace est notée s (en anglais au lieu de p en français)

La fonction de transfert de la chaîne de retour est : K, constante.

La fonction de transfert en boucle fermée est :

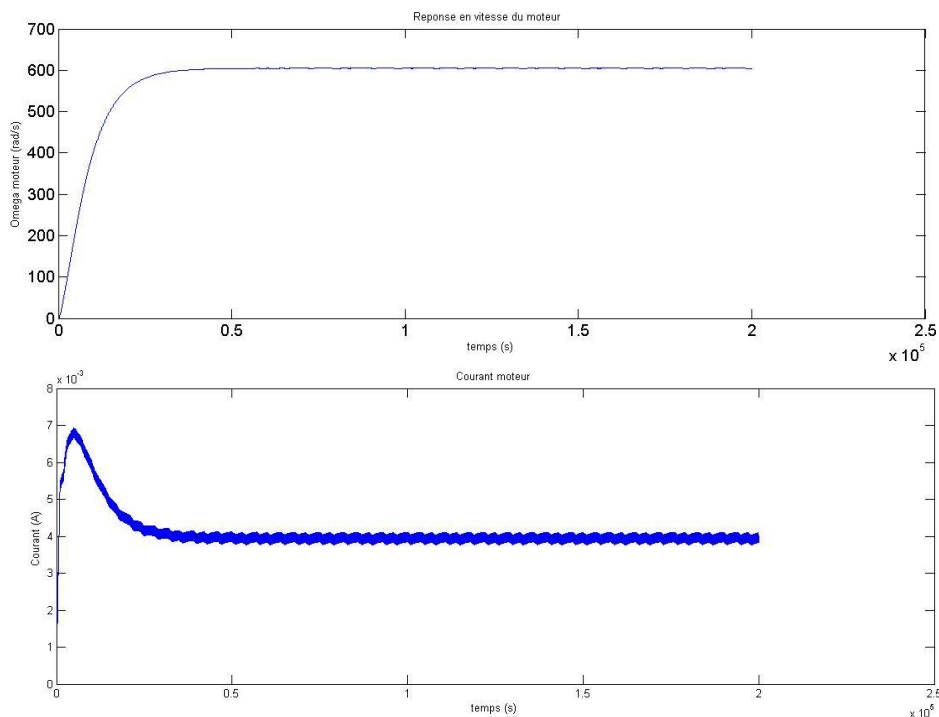
$$\frac{\frac{K}{Lm.Jm.s^2 + (Lm.f + Rm.Jm).s + Rm.f}}{1 + \frac{K^2}{Lm.Jm.s^2 + (Lm.f + Rm.Jm).s + Rm.f}} = \frac{K}{K^2 + Lm.Jm.s^2 + (Lm.f + Rm.Jm).s + Rm.f}$$

$$\frac{\frac{K}{K^2 + Rm.f}}{1 + \frac{(Lm.f + Rm.Jm)}{K^2 + Rm.f}s + \frac{Lm.Jm}{K^2 + Rm.f}s^2} = \frac{Ko}{1 + 2.j.z.\frac{\omega}{\omega o} + \left(\frac{j.\omega}{\omega o}\right)^2}$$

Avec $\omega o = \frac{K^2 + Rm.f}{Lm.Jm}$ la pulsation propre au bloc moteur

Et $z = \frac{\omega o}{2} \cdot \frac{(Lm.f + Rm.Jm)}{K^2 + Rm.f}$ le coefficient d'amortissement.

On notera que pour ce qui est de notre modèle, le couple résistant, Tr est supposé nul. Nous montrerons que dans le cas où cette hypothèse est fausse, notre système de régulation n'est pas précis. Nous pouvons du fait de la simulation sous Simulink avoir des données chiffrées.



Cette simulation est faite avec une valeur de commande à 35% de 8,2 V. On a donc une réponse légèrement inférieure au 800 rad/s normalement attendu pour une commande à 50%.

On observe :

Pour ce qui est de la simulation du fonctionnement du bloc moteur, on établit le schéma présenté plus haut dans le logiciel de simulation. On obtient les résultats en entré suivants.

Au démarrage, le courant induit n'est limité que par la résistance de l'induit.

On ne peut donc pas appliquer au démarrage la tension nominale du moteur car le courant de démarrage serait très supérieur au courant nominal.

Il faut limiter le courant de démarrage à une valeur acceptable par la machine (généralement 2.In). Ce qui est quasiment le cas ici.

Pendant la phase d'accélération, le courant induit augmente très fortement dû aux frottements et à l'inertie des pièces.

Ensuite une fois que la vitesse de rotation a atteint le régime permanent, le courant se stabilise, à une valeur inférieure à celle pendant la phase d'accélération, car le couple en régime établi est plus faible que durant la phase d'accélération.

Le courant est proportionnel non pas à la vitesse de rotation mais au couple.

2) Etude du hacheur :

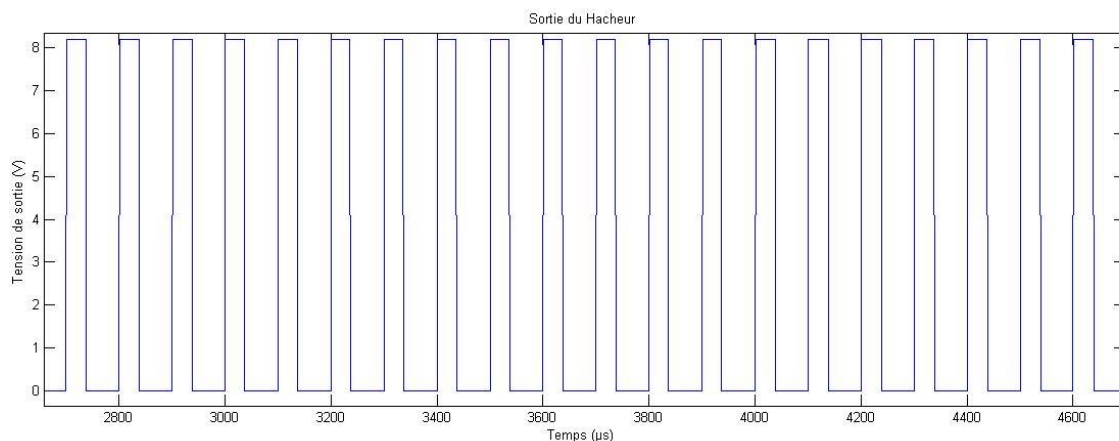
Principe du hacheur :

C'est un convertisseur qui transforme un signal constant, ici la tension issue de la batterie, en un signal rectangulaire de rapport cyclique variable.

Les hacheurs sont très utilisés pour la variation de vitesse de moteurs à courant continu et sont des éléments essentiels des alimentations à découpage. Ils ont un rendement important.

Certains hacheurs sont également réversibles : ils peuvent alors fournir de l'énergie à la charge, généralement une machine à courant continu dans ce type d'application, ou bien en prélever ce qui permet de freiner la machine.

Dans notre cas, la tension délivrée en sortie est inférieure à la tension appliquée en entrée, le hacheur est dit dévolteur (ou abaisseur ou Buck). $U_{Hachée} = \alpha \cdot U_{Batterie}$ Avec α le rapport cyclique. On présente ci-dessous la tension en sortie du hacheur pour un alpha de 35.



Du aux constantes de temps importantes du bloc moteur, celui-ci ne voit que la valeur moyenne de l'alimentation $U_{Hachée}$, on agit donc sur le rapport cyclique du signal haché pour modifier la vitesse de rotation.

Les constantes de temps électriques sont de l'ordre de la milli voire micro seconde alors que les constantes de temps mécaniques sont de l'ordre de la seconde voire la minute.

On simule donc le fonctionnement d'un hacheur à l'aide d'un signal d'entrée en triangle, qui produit donc une valeur que l'on vient ensuite comparer à alpha. Ce signal périodique comparé à une constante produit donc un signal de rapport cyclique comparable à alpha. On compare ensuite la sortie à 0 de sorte à obtenir un signal de rapport cyclique alpha et de valeur 1. On fait ensuite le produit de ce signal, soit zéro soit un avec la tension de la batterie. On obtient ainsi un signal de fréquence celle d'entrée et de valeur moyenne : $U_{bat} \cdot \alpha$.

En pratique U_{bat} varie entre 8.2V à 6.9V en effet la tension aux bornes de la batterie chute au cours du temps, cette chute dure environ 15 minutes pour cette simulation on ne tiendra pas compte de cette chute de tension. Pour la simulation $U_{bat} \approx 8.2$ V. En revanche, nous avons réalisé une simulation où la tension de batterie décroît fortement, nous en parlerons plus loin dans ce rapport.

Pour ce qui est de α , cette valeur varie entre 0% et 100%. Pour la simulation on se place en régime permanent à 35%

On veut une fréquence de fonctionnement du hacheur de 10kHz et une précision de calculs de α de 1%.

Pour le bloc « repeating sequence » : (délivrant le signal périodique)

Time values : 0 à $1/f$ avec $f=10\text{kHz}$ donc $t=1/f=0.1\text{ms}$

Output values : Plage de valeur de α , de 0 à 100.

Pour la précision de α :

On fixe le pas de temps (time step) à $1\mu\text{s}$ car $0.1\text{ms}/1\mu\text{s} = 100$ points d'échantillonnages soit une précision de calcul de α de 1%.

Le minirobot puise l'énergie électrique pour sa propulsion dans une batterie d'accumulateur de type lithium polymère (qui ne doit pas être déchargé en dessous de 3V de sorte à ne pas endommager le microcontrôleur).

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 minirobot. On notera que le rapport cyclique doit augmenter en même temps que la tension d'alimentation augmente de sorte à maintenir la vitesse de rotation souhaitée.

A l'aide du graphique proposé plus haut, on valide le bloc hacheur. Il est alors placé dans un sub-système et inséré dans le fichier global de simulation.

3) Boucle de régulation :

Pour ce qui est de la régulation de vitesse, on fera une estimation de la vitesse de rotation de rotation de l'hélice du propulseur à partir d'une image en tension du courant d'induit et de la tension délivrée par le hacheur, afin de s'affranchir d'un capteur de vitesse et de la chute progressive de la tension batterie, tout en évitant un régulateur de tension qui alourdirait le robot. Cette fonction de calcul est normalement assurée par le microcontrôleur.

Toutes les grandeurs physiques sont des grandeurs analogiques cependant les systèmes numériques (comme le microcontrôleur du minirobot) mémorisent les données sous forme de nombre finis, perdant ainsi un peu de précision mais étant moins sensibles aux perturbations, cependant ils ne traitent que les informations numériques et ceci doit être pris en compte dans la modélisation de notre problème.

Il est donc essentiel d'avoir un convertisseur analogique/numérique en amont du microcontrôleur et un convertisseur numérique/analogique en aval, ils peuvent être compris dans le composant ou doivent être implantés physiquement sur la carte.

Les CAN réalisent la transformation d'un signal analogique (la tension représentative du courant induit) en un signal numérique de N bits. Ce nombre de N bits est une image du courant induit.

La résolution d'un CAN (appelé aussi Quantum q) est la petite variation du signal analogique d'entrée qui provoque un changement d'une unité sur le signal numérique de sortie.

$$q = \frac{\text{Valeur pleine échelle}}{2^N}$$

Les CNA réalisent la transformation d'un signal numérique issue du microcontrôleur du minirobot en un signal analogique.

On veut déterminer la valeur numérique du gain à placer en sortie de ce convertisseur pour avoir $\Omega_{\text{estimé}} = \Omega_{\text{en régime permanent}}$.

Pour ce faire, on modélise le système d'estimation avec en premier lieu un filtre passe-bas de fréquence de coupure 1kHz en effet, il s'agit de gommer les fréquences induites par le hacheur qui en fournissant une tension périodique à 10kHz entraîne un grand nombre d'harmoniques dans le signal or seul la valeur

moyenne nous intéresse car c'est elle qui est ressentie par le système. Cette fonction peut-être réalisé par un simple filtre RC. Nous le modélisons par une fonction de transfert du premier ordre de fréquence de coupure 1kHz.

L'étape suivante est d'adapter la tension fournie par le hacheur. Pour réaliser cela c'est un pont diviseur de tension qui vient diminuer l'amplitude de la tension induite filtrée afin d'être traité par le microcontrôleur cette tension ne doit pas être supérieure à 3,3 V. On divise la tension par un pont diviseur de tension ($R+2R$), en effet avec une tension d'alimentation de 8,2V on a 2.73V en entrée du microcontrôleur qui est inférieur à la tension d'alimentation du microcontrôleur qui est de 3,3V. Cette fonction est réalisée par un simple gain d'un tiers.

Il convient ensuite de transformer le signal analogique en signal numérique on place donc un CAN de 10bits avec une tension pleine échelle de 3,3V.

Où : $2^{10}=1024$

Donc on met un gain de $1024/3,3$ soit 310.3030, c'est une constante de notre problème.

Si la tension d'entrée de 0V, on aura 00 0000 0000 % en sortie, si la tension d'entrée est de 3,3V on aura 11 1111 1111% en sortie.

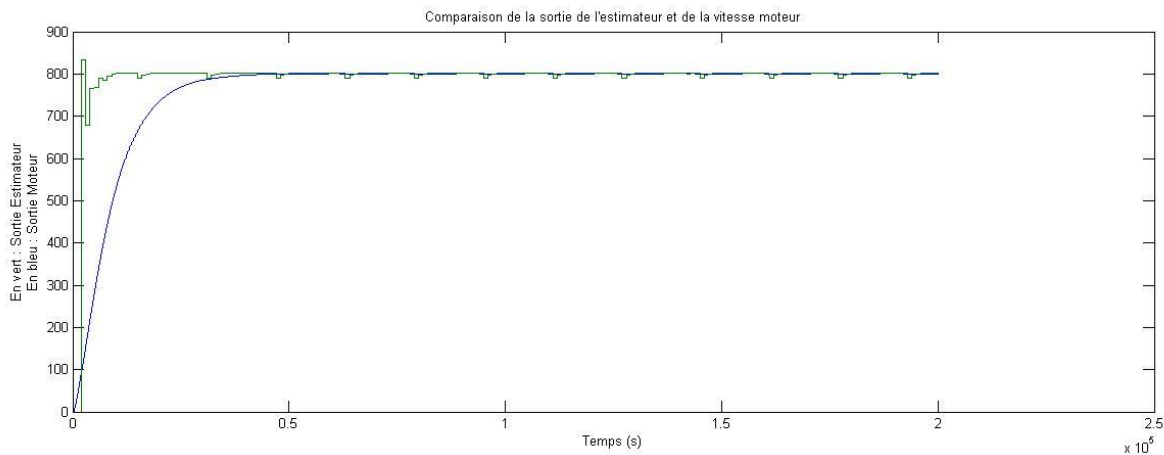
% signifie une valeur binaire.

$$q = \frac{3,3}{2^{10}} = 3,22mV$$

On notera que pour permettre à la simulation de bien prendre en compte ce passage du monde analogique au monde numérique on insert un bloc d'échantillonnage du signal, on passe aussi d'un type de donnée : double, plus grande précision possible à un mode unsignedint 16.

On fait enfin la différence entre ces grandeurs de sorte à obtenir une image de la vitesse du moteur. On vient ensuite lire la valeur obtenue, on la compare avec celle lu au niveau du moteur et on entre le résultat de la division dans le bloc. On a en boucle ouverte une estimation numérique de la vitesse du moteur.

La simulation donne pour une entrée alpha à 50% : (on a représenté les courbes sur le même graphique pour le rendre plus visuel, on voit que l'estimation correspond bien à la vitesse)



4) Régulation de la vitesse de rotation du moteur :

Notion de stabilité :

Un système est stable si la réponse libre du système tend vers 0 quand t tend vers l'infini et si une entrée finie implique une sortie finie. Pour cela le régime libre doit être amorti. Cette condition est réalisée si tous les pôles ont leur partie réelle négative. Un pôle est une racine du dénominateur de la fonction de transfert total du système.

Notion de précision :

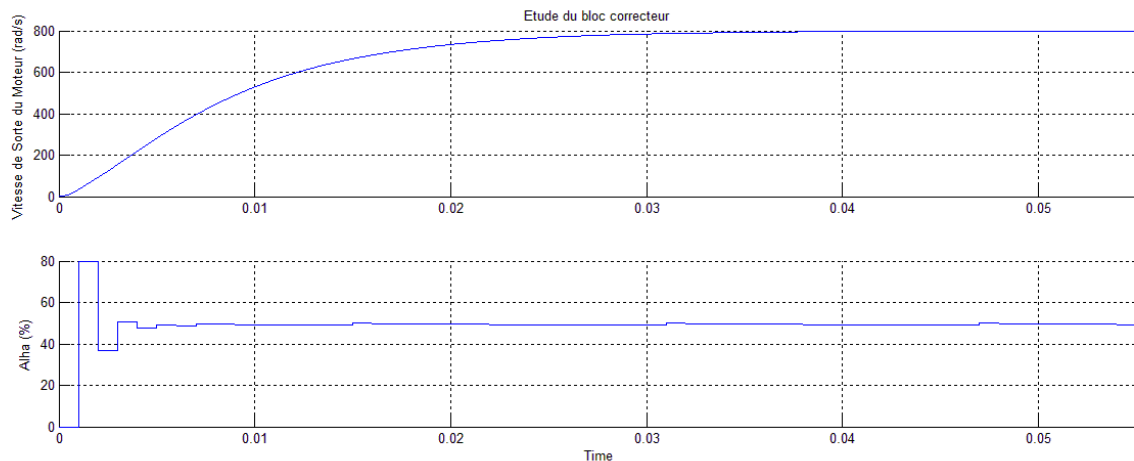
C'est l'écart entre la consigne et le signal fourni par la chaîne de retour en régime établi. Un système est d'autant plus précis que l'erreur en régime établi est faible. La précision d'un système dépend du type du signal d'entrée (échelon, rampe,...) et des perturbations appliquées au système. Plus le système est insensible aux perturbations plus il est dit robuste.

On utilise un correcteur PI (Proportionnelle/Intégrale) pour améliorer les performances du système. L'action proportionnelle augmente la rapidité et la précision du système mais diminue sa stabilité. L'action intégrale augmente la précision du système. L'intérêt d'un correcteur PI est d'augmenter la précision sans pour autant diminuer la stabilité. Le correcteur utilisé ici est un régulateur PID avec action dérivée nulle.

On place le correcteur en sortie de la différence qui est opérée entre la consigne et l'image de la vitesse moteur.

Le correcteur que nous mettons en place dans la simulation consiste en un gain pur en entrée, ce afin d'adapter la vitesse nominale 800 radians par seconde sur 50 % pour alpha. On place ensuite un intégrateur et l'on fait la somme de la sortie de l'intégrateur et du signal en entrée de cet intégrateur. On obtient ainsi une valeur pour alpha qui est à même de faire tendre la vitesse de rotation du moteur vers celle de la consigne.

On a ainsi sous une entrée nominale les résultats de simulation suivant :



On observe que la valeur d'alpha se stabilise autour de 50%, on voit aussi que lors du démarrage, on a accordé plus de puissance au moteur de sorte à accélérer la phase de mise en mouvement.

Résultats et Fiabilisation du modèle :

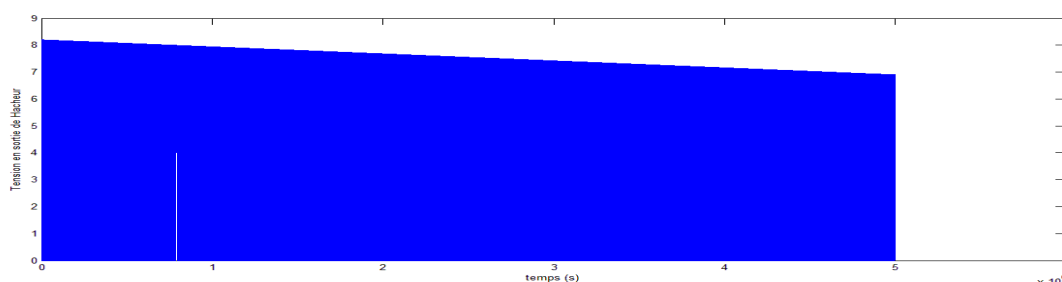
Nous avons procédé à un certain nombre d'autres simulations pour tester la commande que nous avons mis en place.

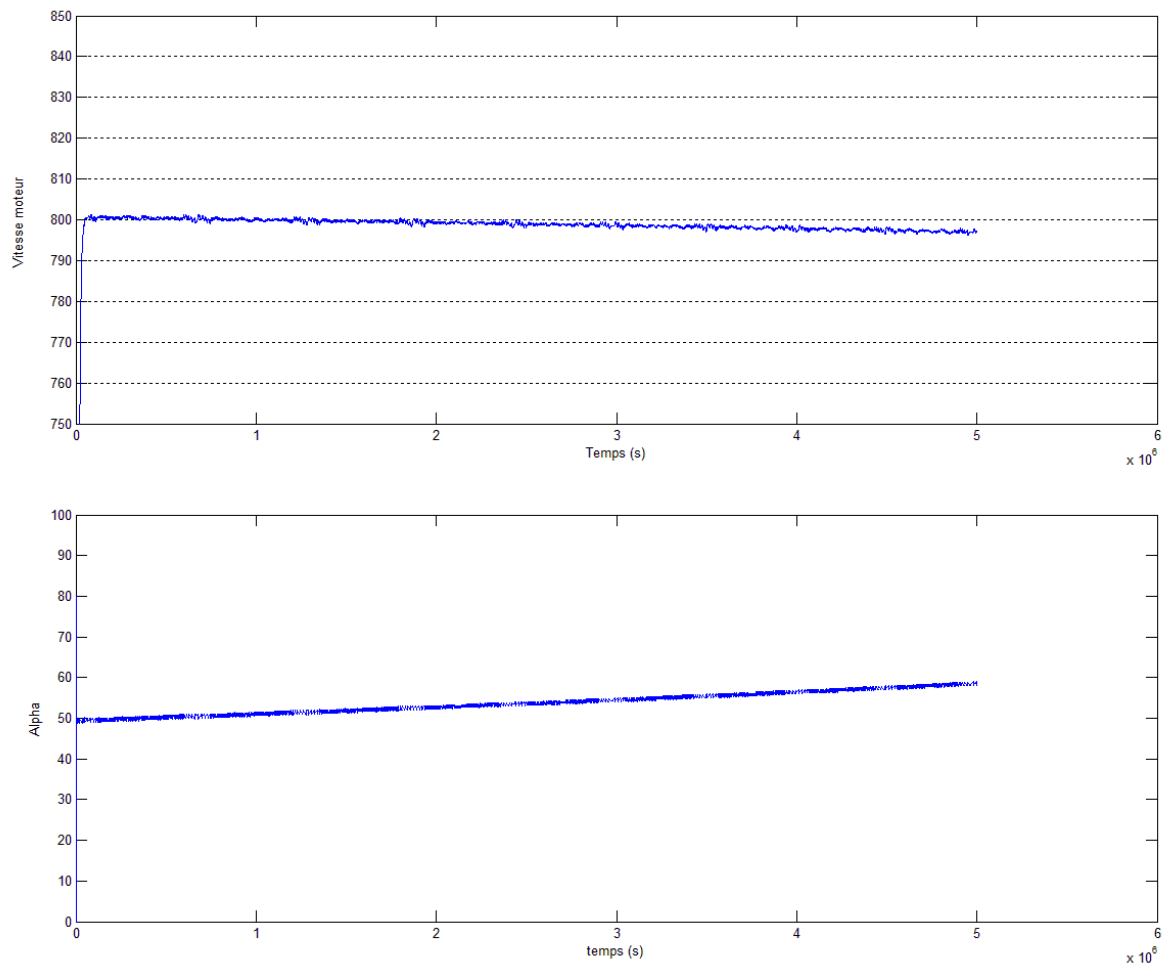
Dans un premier temps nous avons étudié les constantes de temps de notre système. Nous relevons un temps de réponse à 5% de 0.0234 secondes sur notre simulation.

Dans un second temps nous avons procédé à une fiabilisation du système avec une batterie en perte de puissance on observe que le comportement en vitesse ne change pas, la valeur de commande est toujours respectée. Ce qui change en revanche c'est la valeur d'alpha qui croît au fur et à mesure qu'Ubat diminue.

Dans cette première simulation on fait chuter la tension de la batterie en 5 secondes, avec une batterie pleine à 8,2 V.

On observe les résultats suivants :

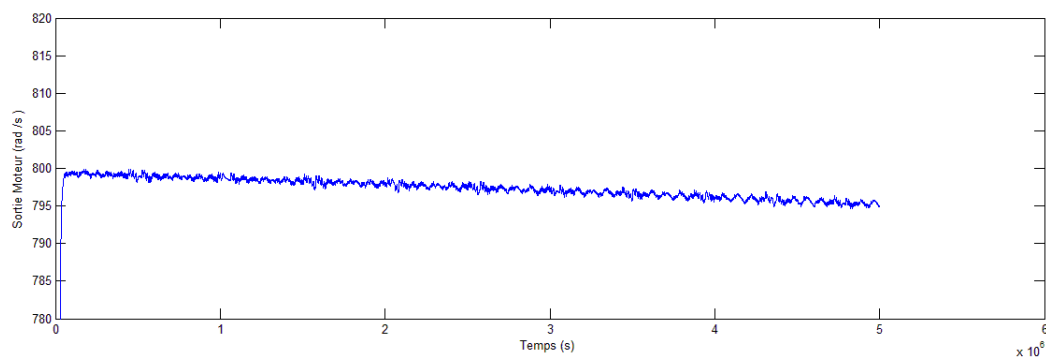




La vitesse de sortie reste bien relativement proche de 800 radians par seconde. En revanche, le rapport cyclique lui augmente fortement tandis que la tension délivrée par la batterie diminue.

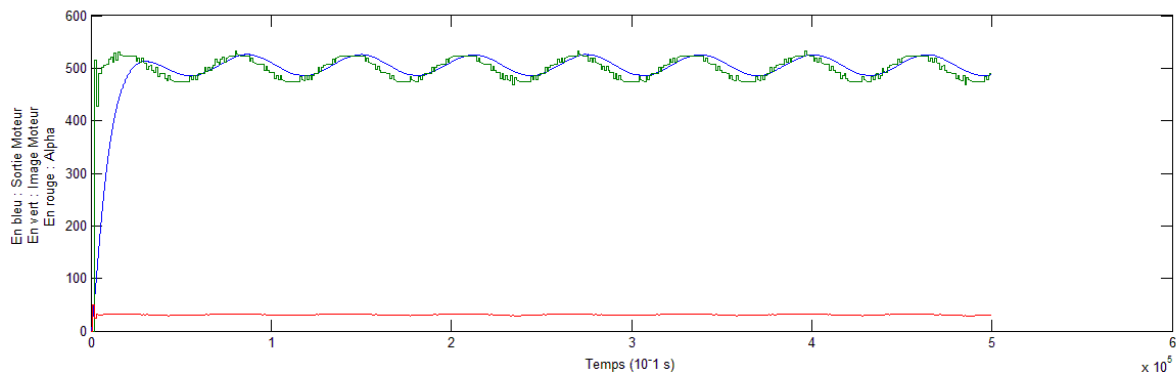
Dans une seconde simulation on fait chuter la tension de la batterie en 5 secondes avec une batterie déjà déchargé à 7,7 V.

On observe les résultats suivants :

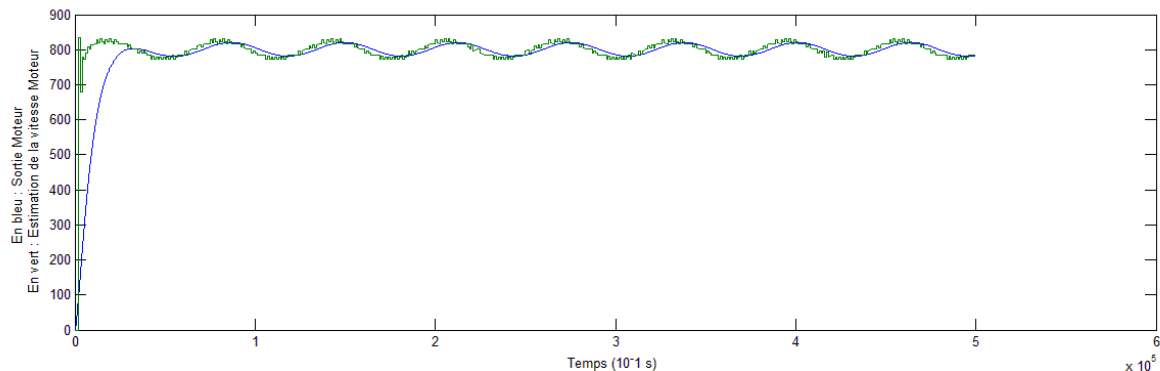


On voit bien qu'il n'y a toujours pas de décrochement net dans notre modèle.

On effectue ensuite une simulation avec une entrée de commande non plus en échelon mais sinusoïdale, afin de tester les performances de notre système de commande. On trouve les résultats suivants :



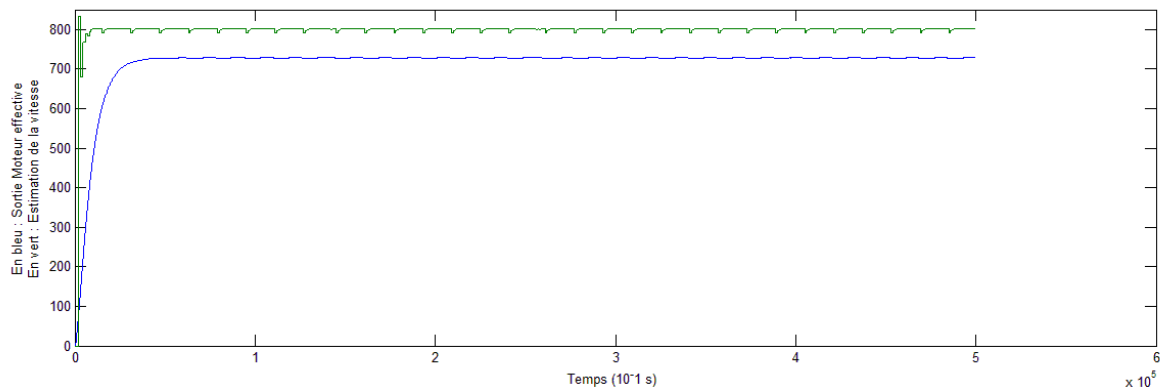
On observe un léger retard dans entre la commande et la sortie effective du système. L'ordre de grandeur de ce retard est d'environ 10^{-2} s. On observe que le retard est moins important si l'on se place sur la vitesse nominale de 800 rad / s comme constante dans notre entrée :



On observe en outre que la commande Alpha ne décrit pas de grandes variations sur une entrée sinusoïdale.

Enfin, on va tester la robustesse de notre modèle vis-à-vis des perturbations externes, on va par insérer dans le bloc moteur un effort résistant de 10^{-3} Newtons, on soumet le système à son entrée nominale on observe que tout de suite, l'image de l'on crée de la vitesse moteur n'est pas bonne.

On observe :



Les résultats ne sont pas très concluants, la vitesse estimée et la sortie effective sont éloignés d'environ 14%.

En effet, pour calculer la vitesse du moteur, on vient prendre des informations qui ne tiennent pas compte de cette perturbation. Notre modèle atteint donc là ces limites.

La mise en place du modèle MatLab® :

Comme nous en avons parlé plus haut, nous avons essayé de coller au mieux à la réalité sur cette simulation. En particulier, nous avons adapté les types de donnée que nous faisons passer entre les blocs. Il y a donc eut des adaptations à faire en termes d'affichage des données. Avant d'afficher des données, il faut revenir à des doubles on converti donc dans le CAN en uint16 pour repasser en double pour afficher les données. Cette opération est cependant essentielle car les blocs préprogrammés que propose Simulink® en général qu'un certain type de donnée. L'apprentissage de la gestion de ces types de donnée est donc primordial dans l'outil de simulation. En outre, lors de l'affichage, il faut être attentif à la fréquence

d'échantillonnage des données, il convient donc d'insérer un bloc de transition pour assurer la même fréquence.

Pour ce qui est de la détermination des coefficients, pour mettre au point les valeurs numériques, nous nous plaçons dans un cas simple, où la valeur peut être simplement extrapolée. On rend alors cette valeur unitaire puis on compare la valeur trouvée avec celle attendue. On remplace alors la valeur par celle calculée à l'aide d'une simple règle de 3 et on vérifie avec une nouvelle simulation que la valeur entrée donne bien le résultat attendu.

Enfin, nous avons inséré un bloc de saturation dans le correcteur afin de garantir qu' α ne passe soit pas négatif ou supérieur à 100 %.

Conclusion :

Nous avons réalisé lors de ce projet une fonction que nous avons déjà très longuement étudiée en cours mais l'apport sur la maîtrise du logiciel n'en a été que bénéfique. Nous avons apprécié développer et mettre en pratique les concepts théoriques sur une simulation. En somme, MatLab® et surtout Simulink® sont des outils que nous avons appris à comprendre et maîtriser au cours de ce projet. Pour terminer, il reste cependant un détail sur lequel nous restons sur notre faim, lors de la détermination numérique des coefficients nous n'avons aucune certitude que les valeurs prises soient optimales. En somme, cela nous a permis de voir à la fois certaines limites du logiciel et de notre modèle.