

Rapport BE Trottinette

Mailis Dy, Cyprien Heusse

4AE-SE-TP3

Introduction

Dans ce bureau d'étude, nous allons analyser une trottinette électrique dans l'objectif de mettre en place la commande de couple puis de vitesse par l'utilisateur.

Pour cela, nous disposons de plusieurs capteurs (tachymètre, capteur de courant, potentiomètre) et nous pouvons contrôler le courant envoyé au moteur via une PWM.

Dans un premier temps, nous analyserons le fonctionnement du système dans le but de trouver un modèle fidèle pour celui-ci. Ce modèle nous permettra par la suite de mettre en place les différentes commandes du système.

Enfin, nous mettrons en place plusieurs améliorations permettant de rendre la trottinette plus précise et facile à manier par l'utilisateur.

Questionnaire

II- Mission n°1 : l'asservissement de couple

a. Première approche

1) Quelle est la tension de batterie requise pour le fonctionnement de la trottinette ? Ce niveau de tension est-il dangereux ?

La tension de batterie requise pour le fonctionnement de la trottinette est de 24V. Ce n'est pas une tension dangereuse.

2) Quelle sont la tension, le courant, la puissance nominales du moteur ? De quel type s'agit-il ?

Il s'agit d'un moteur ayant une tension nominale de 24V, un courant maximal de 10A et une puissance nominale d'environ 100W. C'est un moteur à courant continu (CC).

3) Qu'entend-t-on par "asservissement de couple" ? On pourra se reporter au cours sur la MCC (modélisation_MCC.pdf) pour répondre à cette question.

Un asservissement de couple consiste à imposer un courant à la MCC qui pilotera directement le couple du moteur laissant la vitesse de celui-ci en fonction de la charge mécanique appliquée à l'arbre.

4) Quelle est la consigne du système ? Quelle est sa dimension (son unité) ?

La commande du système est l'angle en radians.

5) Quelle est la grandeur captée nécessaire pour opérer l'asservissement de couple ? Quelle est sa dimension (son unité) ?

Il s'agit du courant qui est capté grâce à un capteur à effet HALL, son unité est l'Ampère. Le courant est une représentation du couple électromagnétique qui correspond au couple mécanique + les frottements (pertes). En considérant ces pertes comme faibles, on peut estimer le couple mécanique directement.

6) Quelle est la commande du système physique (Système physique = hacheur + moteur) ?

PWM représentant l'angle de commande en entrée du hacheur.

7) L'asservissement de couple (comme de vitesse) se fait par la comparaison entre la grandeur de consigne et la grandeur physique que l'on veut asservir. Ces deux grandeurs, à l'entrée du μC doivent donc avoir la même dimension pour pouvoir être comparées. Laquelle ? Quel est le capteur de consigne ? Quel est le capteur de la grandeur physique à asservir ?

Le potentiomètre permet d'obtenir une tension qui dépend de l'angle donné par l'utilisateur. Il s'agit du capteur de consigne.

Le capteur de courant est quant à lui le capteur de la grandeur physique à asservir. Il donne aussi une tension représentant le courant utilisé par le moteur.

8) Dessiner un premier schéma, une première boucle de régulation en nommant chaque bloc.

Avoir une petite période d'échantillonnage permet une transformée bilinéaire très précise. Cela nous permet donc d'obtenir une très bonne approximation du linéaire, mais demande également beaucoup de calculs ce qui peut poser un problème avec la vitesse du CPU.

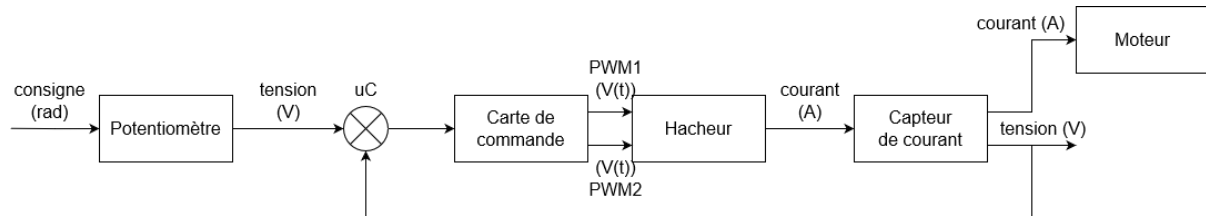
En choisissant un T_E plus grand, on diminue donc l'intensité de calculs donc cela permet au CPU de faire d'autres actions en parallèle, mais notre signal sera grandement approximé.

Notre système approximé au premier ordre à un τ de $400\mu s$, on propose donc de prendre un T_E 20 fois plus petit que cette durée pour obtenir un $T_E = 20\mu s$.

I- Modélisation du système

La première partie de ce bureau d'étude consiste à modéliser le système trottinette afin de pouvoir par la suite le commander.

La première étape est de déterminer le lien entre les différents composants :



Sur ce schéma-bloc, on a mis les différents composants ainsi que le type d'unité physique qu'ils prennent en entrée et en sortie. Tout le système est déjà construit et nous ne pouvons intervenir que sur le code microcontrôleur et donc la loi de commande. Pour pouvoir déterminer une loi de commande qui réponde aux spécifications du cahier des charges, il nous faut connaître le comportement de tous les composants présents sur ce schéma.

a. Potentiomètre

Le potentiomètre se content de prendre une consigne qui correspond à un angle et de le convertir en une tension directement proportionnelle à l'angle grâce à une résistance variable. Ce bloc peut donc être modélisé par un gain qu'on appellera K_P .

b. Capteur de courant

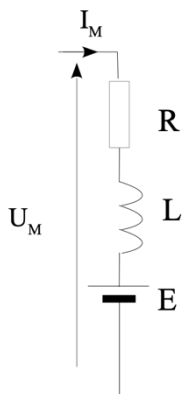
Le capteur de courant convertit quant à lui un courant (le courant qui traverse le moteur) en une tension que l'on pourra lire à l'aide d'un CAN sur le microcontrôleur. Il convertit le courant de manière proportionnelle et peut donc être représenté par un gain K_C .

Ce gain a été trouvé directement grâce à la datasheet du capteur utilisé (LTS-6NP), qui nous indique donc une pente de conversion de $K_C = \frac{0.625}{6} = 0.1042$

| Number of primary turns | Primary nominal r.m.s. current I_{PN} [A] | Nominal output voltage V_{OUT} [V] | Primary resistance R_P [mΩ] | Primary insertion inductance L_P [μH] | Recommended connections |
|-------------------------|---|--------------------------------------|-------------------------------|---|--|
| 1 | ± 6 | 2.5 ± 0.625 | 0.18 | 0.013 | <div style="display: flex; align-items: center;"> <div style="margin-right: 10px;"> 6 ○ 1 </div> <div style="margin-right: 10px;"> 5 ○ 2 </div> <div style="margin-right: 10px;"> 4 ○ 3 </div> <div style="text-align: center;"> OUT </div> </div> |

c. Moteur

On a vu qu'un moteur à courant continu comme celui qu'on utilise peut être modélisé de la façon suivante :



Les valeurs de R et de L sont données dans la spécification du moteur en question. Cependant, ces valeurs ne sont pas exactes et varient d'un moteur à l'autre. Notre modélisation du système ne sera donc pas parfaite mais on améliorera cela dans les parties suivantes.

Ici, les valeurs de la résistance et de la bobine sont $R = 1\Omega$ et $L = 2mH$. Cela nous donne donc un système du premier ordre avec un $\tau = \frac{L}{R} = 2 * 10^{-3}s$. Le système complet est

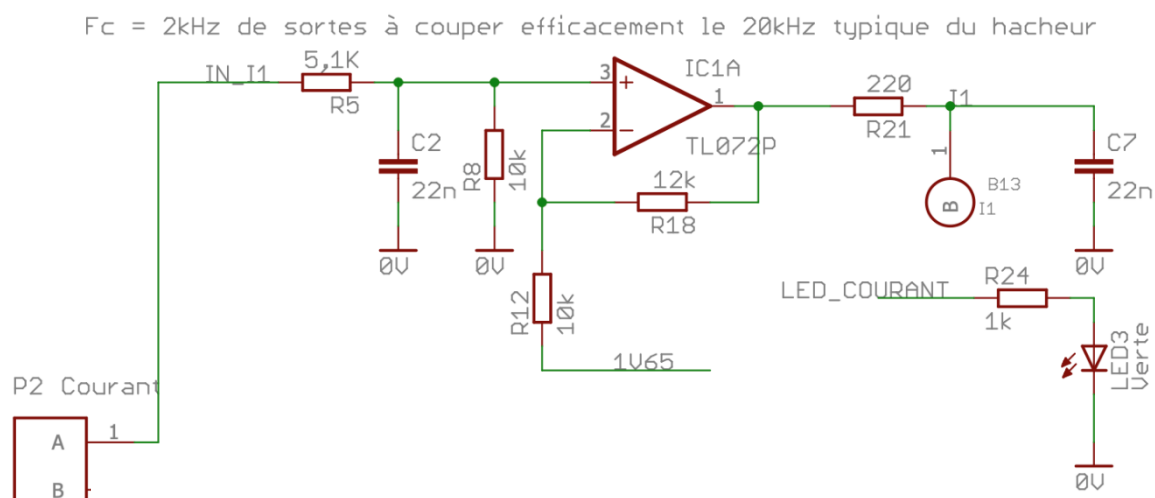
$$\frac{\frac{1}{R}}{1 + \frac{L}{R} * p}$$

On remarque aussi la tension E qu'il faudra ajouter comme une constante dans le schéma-bloc. Cependant, la modélisation deviendrait inutilement très complexe et nous avons donc décidé de l'ignorer - dans la suite en prenant $E = 0V$. - en négligeant la valeur de E dans nos calculs de commande. Cette approximation restera valable puisqu'on a mentionné plus haut que $\tau = 400\mu s$, et donc le système est assez rapide pour approximer la valeur de E comme étant statique, et les calculs de lois de commandes s'appuient sur des variations de valeurs.

d. Hacheur

Le hacheur peut être modélisé comme un gain également qui prend la valeur de la PWM en pourcentage et la traduit en une tension qui varie entre $-V_{bat}$ et $+V_{bat}$ ou alors entre 0 et $2 * V_{bat}$ ce qui revient à un gain $K_H = 2 * V_{bat} = 48$.

Un autre élément à prendre en compte est le circuit que l'on trouve entre le capteur de courant et l'entrée du CAN. Ce circuit est le suivant :



Ce circuit a pour objectif d'adapter la tension qui sort du capteur de courant pour qu'il occupe le plus efficacement possible la tension de pleine échelle du CAN.

Sa fonction de transfert est la suivante :

$$\frac{\frac{R_{12} + R_{18}}{R_8 + R_5}}{\left(1 + \frac{R_5 * R_8 * C_2}{R_8 + R_5} * p\right) (1 + R_{21} * C_7 * p)} = \frac{K}{(1 + \tau_1 * p)(1 + \tau_2 * p)}$$

Avec $K = 1,457$, $\tau_1 = 7,43 * 10^{-5}s$ et $\tau_2 = 4,84 * 10^{-6}s$ ce qui nous donne $f_{c1} = 2142Hz$ et $f_{c2} = 32883Hz$. f_{c2} étant très supérieur à f_{c1} on peut décider de l'ignorer dans notre modélisation. Ce montage agit aussi comme un filtre permettant de couper les fréquences parasites du hacheur qui sont aux alentours de $20KHz$, fréquence de la PWM.

Une fois tout le système modélisé convenablement, on peut s'interroger sur la façon de commander ce système.

II- Asservissement du couple

Notre objectif est de mettre en place une commande de couple. C'est-à-dire qu'on veut que le potentiomètre en entrée contrôle directement le couple fourni par le moteur. Le cahier des charges précise qu'on veut un asservissement continu ayant les caractéristiques suivantes :

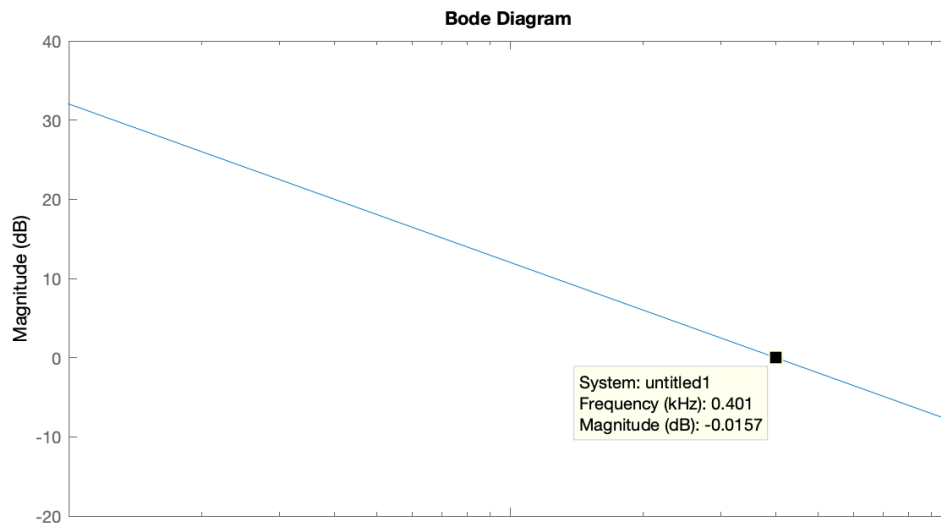
- Une marge de phase supérieure à 45°
- Une fréquence de transition en boucle ouverte qui se situe entre 300Hz et 500Hz
- Une erreur statique nulle en boucle fermée.

On notera C la commande, G la fonction de transfert du système et F la fonction de transfert du retour.

Notre système en boucle ouverte est donc $C * G * F$.

a. Asservissement continu

On voudrait que notre système complet ait un diagramme de Bode comme ceci :



Or pour l'instant, notre système à 3 fréquences de coupures : celle du moteur et les deux du circuit du capteur de courant. On a déjà dit que l'on peut ignorer celle à 32kHz mais on peut aussi ignorer (pour le moment) celle qui est à 2kHz comme elle se situe après la fréquence de transition que l'on veut. En revanche, il nous faudra nous occuper de celle du moteur qui se situe à 80Hz , c'est-à-dire en plein dans la plage sur laquelle on veut agir.

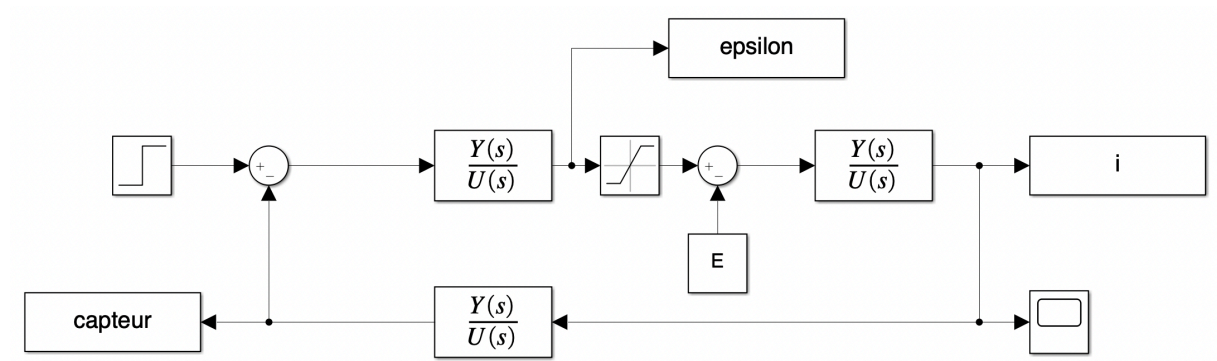
Nous choisissons de mettre en place une commande de type Proportionnel Intégrateur (P.I.) car elle permet d'obtenir une erreur en boucle fermée nulle pour un échelon en entrée et permet aussi d'avoir une granularité sur deux paramètres K_I et K_P . Elle s'écrit de la façon suivante : $K_P + \frac{K_I}{p}$ mais peut se réécrire comme ceci : $\frac{K_P * p + K_I}{p} = \frac{1 + \tau * p}{\tau_i * p}$. L'avantage de cette réécriture est que l'on voit tout de suite que l'on pourra effectivement annuler la composante du moteur qui a une fréquence de coupure à 80Hz avec la partie numérateur de notre correcteur. La partie dénominateur assurera la pente de $20\text{dB}/\text{decade}$ que l'on veut et on pourra la modifier pour atteindre les 400Hz en changeant la valeur de τ_i . -- On choisit donc $\tau = \frac{L}{R}$. -- On a $\tau = \frac{L}{R} = 2\text{ms}$, correspondant à la caractéristique du moteur.

Pour trouver τ_i , on utilise le fait que notre système ait un gain statique nul. La FTBO peut s'approximer par $\left| \frac{K}{\tau_i p} \right|$ et donc cela se traduit par : $\left| \frac{K}{\tau_i p} \right| = 1$, à 400Hz avec $K = K * K_c * K_h$

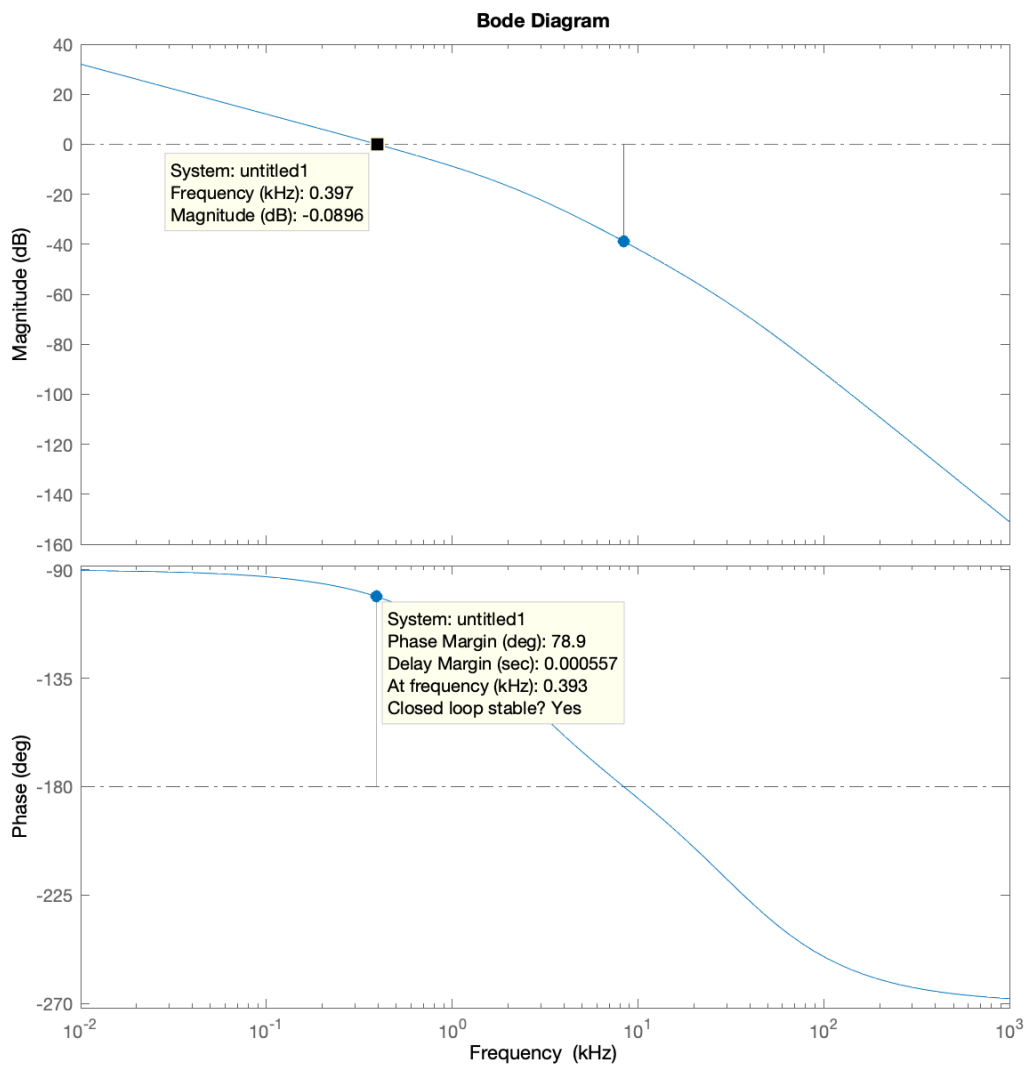
On obtient ainsi $\tau_i = 2.8ms$

On ajoute une saturation sur la commande pour éviter les effet windup.

On simule finalement le système sur MatLab en utilisant Simulink. Le système est le suivant :



Le diagramme de Bode de ce système en boucle ouverte est le suivant :



On voit bien que les caractéristiques exigées par le cahier des charges sont respectées : on a une marge de phase bien supérieure à 45° et une fréquence de coupure précisément à 400Hz.

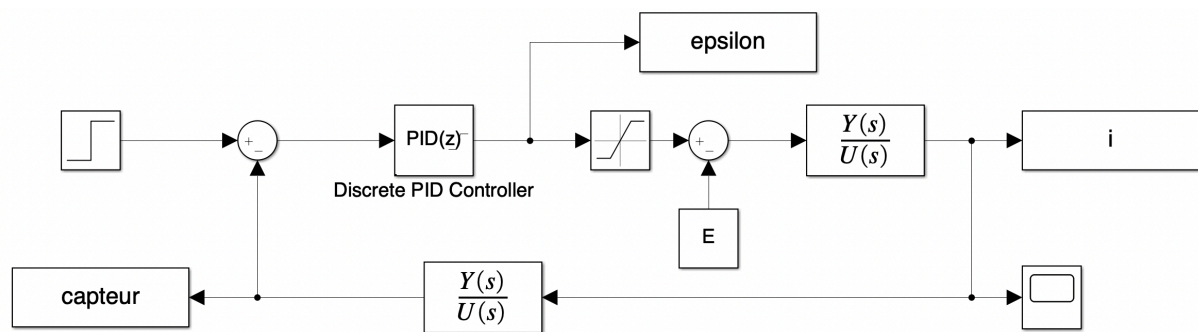
b. Asservissement discret

Une fois la commande en continu trouvée, il faut passer dans le domaine discret car on implémentera la loi de commande sur un microcontrôleur, intrinsèquement discret. Pour cela nous utilisons l'approximation bilinéaire de la variable continue p . C'est-à-dire que l'on considère que $p = \frac{2}{T_E} * \frac{z-1}{z+1}$. Il faut pour cela choisir une période d'échantillonnage pour notre système discret.

Nous avons évoqué plus haut les différents critères nous poussant à choisir une période d'échantillonnage assez faible. Nous avons décidé finalement $T_E = 20 \mu s$, mais cette période est trop petite pour que notre système fonctionne correctement... Nous décidons donc de passer à une période d'échantillonnage de $T_E = 100 \mu s$.

Nous calculons les coefficients le plus possible en gardant les lettres de façon à pouvoir implémenter des constantes facilement modifiables dans notre code permettant de facilement changer les paramètres pour s'adapter au système physique. On pourra changer la période d'échantillonnage en une seule ligne par exemple, sans avoir à recalculer tous les coefficients qui en dépende.

On simule ensuite notre système sur Matlab en réutilisant le fichier Simulink précédent, utilisé pour la partie discrète mais en changeant le bloc de fonction de transfert continu par un bloc discret. On vérifie ainsi la cohérence et la justesse de nos résultats.



III- Implémentation

a. Keil

Tout comme dans les fichiers Matlab, nous nous arrangeons pour définir les variables de manière générale et de les calculer quand c'est possible de manière littérale afin de pouvoir faire varier assez facilement nos variables sans avoir à changer tout le code à chaque fois. Nous avons choisi de convertir toutes nos valeurs en tension afin que cela nous soit plus clair.

Dans un premier temps, nous partons du principe que nos constantes de temps sont bonnes et nous allons juste construire le programme de commande de base.

A chaque interruption du Systick, on lit la valeur à la sortie du potentiomètre et la valeur du courant à la sortie du capteur. On calcule ensuite *epsilon*, la différence entre ces deux valeurs pour connaître la valeur de correction à apporter selon la fonction de transfert que l'on avait choisi :

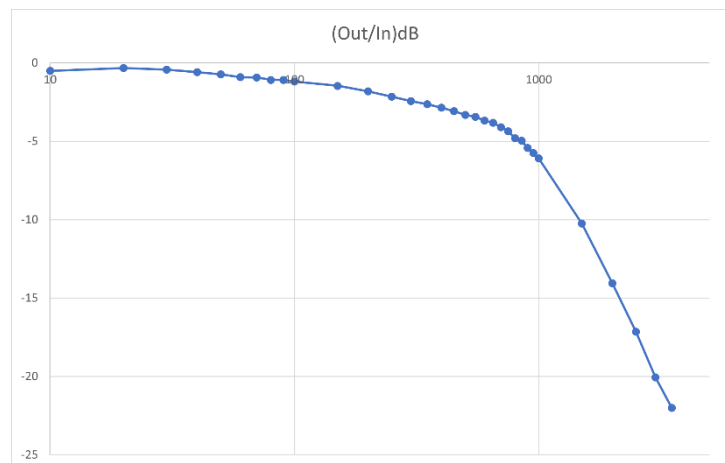
$$y_k = a_1 * u_k + a_0 * u_{k-1} + y_{k-1}$$

Avec $a_0 = \frac{T_E}{2*\tau_i} - \frac{\tau_1}{\tau_i}$ et $a_1 = \frac{T_E}{2*\tau_i} + \frac{\tau_1}{\tau_i}$ et donc dans notre cas : $\epsilon = y_k$

On ajoute aussi la saturation de la sortie par un *if*. Il est important à la fin de la boucle de mettre y_{k-1} égal à la précédente valeur saturée, et non pas la valeur avant saturation.

On convertit ensuite la valeur obtenue en rapport cyclique pour la PWM.

b. Bode du système asservi réel



On réalise ensuite à la main le Bode du système asservi pour s'assurer que son allure correspond à nos attentes. On fait donc de nombreuses mesures en faisant varier la fréquence d'entrée sur le GBF et en observant l'amplitude de sortie sur l'oscilloscope. Voici le résultat obtenu :

On a donc la bonne allure et la bonne fréquence de coupure.

c. Autoidentification de τ

Une fois que nous avons vérifié le bon fonctionnement de notre système, nous remarquons quand même que la fréquence de coupure n'est pas ce que nous avions prévu et se trouve plus proche des $1kHz$ que des $400Hz$ prévus. Ceci peut s'expliquer par l'inexactitude des paramètres L et R du moteur à courant continu dont nous avons parlé dans les parties précédentes. Pour remédier à cela, nous proposons la mise en place d'une partie logicielle permettant l'identification du vrai τ du moteur au démarrage du système. Pour cela, nous mettons en place un code qui lance le système avec une PWM de 40% au démarrage, attend un peu que le système stabilise et passe soudainement la PWM à 60% de façon à créer un échelon artificiel. Le logiciel échantillonne les différentes valeurs prise par le courant (représentation du couple du moteur) et identifie quand la valeur finale est atteinte. Une fois cette valeur identifiée, il suffit de trouver au bout de combien d'échantillon nous avons atteint 63% de cette valeur pour trouver notre τ . Nous recalculons ensuite la valeur des coefficients de la loi de commande en fonction de cette nouvelle vraie valeur et lançon le correcteur normalement.

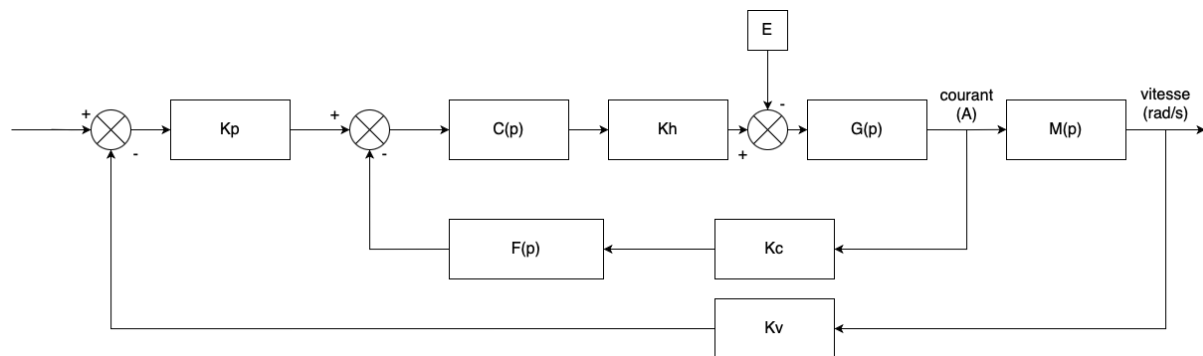
Nous avons dû faire face à des problèmes physiques quand nous avons essayé d'implémenter cette approche. Nous utilisions initialement les valeurs 0% et 100% mais trouvions une réponse à un échelon qui n'était pas cohérente avec le système. C'était dû au hacheur qui « saturait » pour des valeurs de PWM trop élevée.

Cette approche n'est pas parfaite, il faudrait mettre en place une approche qui soit plus précise en diminuant par exemple le temps d'échantillonnage au démarrage avant de le remettre à sa valeur de correcteur.

IV- Asservissement de vitesse

On souhaite à présent mettre en place un asservissement de vitesse. Celui-ci permet de fournir plus de couple dans des situations « extrêmes » comme en montée en restant à vitesse constante. Cette commande est plus pratique pour l'utilisateur et correspond ainsi davantage au vrai monde.

Pour mettre en place cet asservissement, il nous faut être capable de retrouver une information de vitesse. Pour cela, un tachymètre est prévu dans le système. Il retranscrit directement la vitesse à un gain K_V prêt. Ainsi, le système complet ressemble à ceci :



L'objectif est de mettre le point de fonctionnement de notre asservissement autour de 10Hz . Ceci nous permet de considérer que tout ce système se comporte comme un intégrateur $\frac{K}{p}$. Nous pouvons identifier K en utilisant les contraintes fournies, comme pour le précédent calcul de fonction de transfert.

Les contraintes pour le système asservi sont :

- Une marge de phase $M_\phi > 45^\circ$ nous choisirons 50°
- Une fréquence de transition $f_T = 10\text{Hz}$ donc $\omega_T = 20\pi$

Nous choisissons d'asservir le système par un autre correcteur P.I. Pour rappel, sa forme est :

$$\frac{1 + \tau p}{\tau_i p}$$

Il faut donc trouver τ et τ_i .

Identification de τ :

Le système complet est donc de la forme $FTBO = \frac{K(1+\tau p)}{\tau_i p} = \frac{-K-K\tau j\omega}{\tau_i \omega^2}$

On veut :

$$\begin{aligned} M_\phi = 50^\circ &\Leftrightarrow \arctan(\tau\omega_T) + 180 = 50 \Leftrightarrow \arctan(\tau\omega_T) = -130 \\ &\Leftrightarrow \tau\omega_T = 1,191753 \\ &\Leftrightarrow \tau = 18,967\text{ms} \end{aligned}$$

Identification de τ_i :

Il nous faut maintenant fixer la fréquence de transition de sorte que à cette fréquence, le système en boucle ouverte ait pour valeur de module 1.

Comme précédemment, on utilise le fait que le gain de la *FTBO* vaille 1 à la fréquence de transition à 10Hz pour trouver τ_i .

On a toujours $FTBO = \frac{-K - K\tau j\omega}{\tau_i \omega^2}$ donc

$$\begin{aligned} |FTBO| &= \left| \frac{-K - K\tau j\omega}{\tau_i \omega^2} \right| = \sqrt{\frac{K^2 + (K\tau j\omega)^2}{(\tau_i \omega^2)^2}} = 1 \\ \Leftrightarrow \tau_i \omega_T^2 &= \sqrt{K^2 + (K\tau j\omega_T)^2} \\ \Leftrightarrow \tau_i &= \frac{\sqrt{K^2 + (K\tau j\omega_T)^2}}{\omega_T^2} = 23,25ms \end{aligned}$$

Dans le fichier Keil on utilise une période plus grande pour la correction de vitesse pour qu'elle se fasse moins souvent que l'échantillonnage.

On réalise par la suite que nous n'avons pas pris en compte le gain de conversion du CAN : $K' = \frac{3.3}{20}$, et on recalcule donc les constantes de temps associées pour parfaire notre système au maximum.

Améliorations finales :

- Nous avons remplacé l'utilisation du potentiomètre par l'accélérateur sur la poignée augmenter la praticité d'utilisation de la trottinette.
 - Nous avons également changé la plage de vitesses fournies par la poignée : nous avons supprimé la marche arrière qui était présente sur la configuration précédente.
- Pour cela nous avons mesuré la tension résultante au repos et au maximum de la poignée, et nous avons convertit dans Keil. Pour trouver la valeur exacte de la conversion, nous avons procédé par tâtonnement, mais il aurait été possible de le faire de manière plus rigoureuse.

Conclusion

Nous avons réussi à implémenter la commande de vitesse qui est la commande la plus pratique et naturelle pour les utilisateurs. Cette commande est efficace car elle permettra une vitesse stable peu importe l'effort à fournir par le moteur (couple). En effet, cette commande se base sur la commande de couple que nous avons mis en place dans un premier temps.

Les autres ajouts que nous avons effectués vont aussi dans le sens de rendre cette trottinette « user-friendly ». Pour aller davantage dans ce sens, nous aurions pu ajouter d'autres caractéristiques comme l'auto-détection de la plage de vitesse possible avec la poignée de vitesse afin d'avoir un arrêt propre quand on lâche la poignée et la plus grande vitesse possible quand on la pousse au maximum. Nous aurions pu l'implémenter de manière comparable à l'auto-identification de τ .