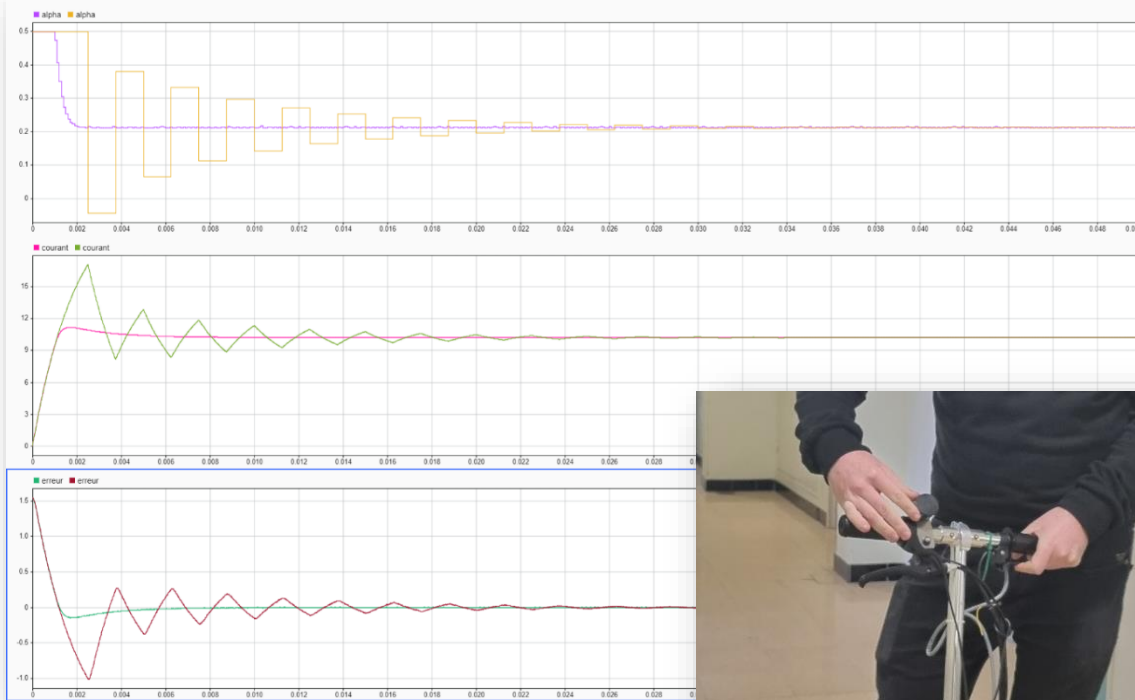
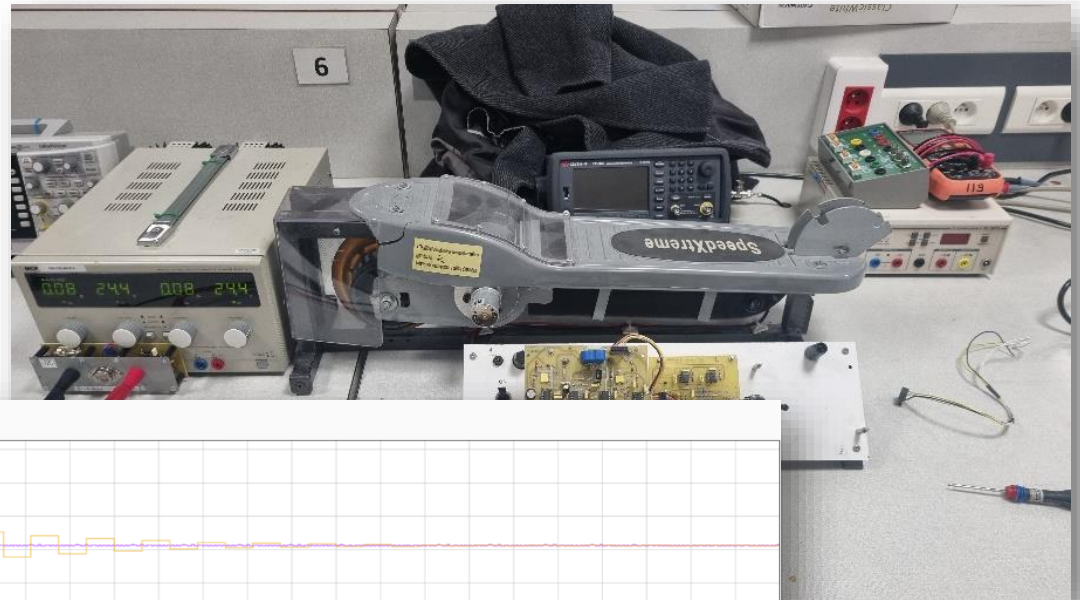


RAPPORT DE BE Trotinettes

(RELANDEAU ANTONIN, SOUAL KILIAN 4^{AE} SE TP1)

GitHub : https://github.com/kiso6/BE_Trott



INTRODUCTION :

Le bureau d'étude Trotтинette traite de la mise en place de différentes commandes de régulation sur une trottinette électrique. Le dispositif fonctionne sur une batterie 24V, et la commande sera faite de manière numérique sur un microcontrôleur STM32.

Ce bureau d'étude se décompose en plusieurs étapes, de l'analyse du cahier des charges jusqu'à la régulation en couple et en vitesse (qui n'a pas été réalisée dans notre cas). Le rapport est structuré par les différentes étapes de ce bureau d'étude.

De ce fait, dans une première partie, nous nous consacrerons à l'étude du cahier des charges et à la compréhension du système. Puis nous verrons quelle est la démarche de choix et de dimensionnement du correcteur pour la régulation de couple. Ensuite nous verrons son implémentation ainsi que les différents tests de caractérisation dans le monde réel. Notre dernière partie sera une brève explication de la commande en vitesse.

0. Réponse aux questions préliminaires :

1. Non, pas besoin de certification électrique. Il faut néanmoins faire gaffe aux potentielles explosions de composants électroniques à la mise sous tension.
2. Le moteur utilisé est un DC à balais, de tension nominale égale à 24V, de courant nominal égal à 10A et de puissance nominale 100W.
3. On va vouloir asservir le courant pour asservir le couple. En effet dans le cas des MCC le couple est donné par le produit $T = K.I$ avec K la cste de couple.
4. La consigne d'entrée est donnée par un potentiomètre/ une manette qui agit une tension entre 0 et 3.3V.
5. On doit mesurer le courant dans le moteur (en ampères). Ce courant est converti en tension pour pouvoir être géré par le μC .
6. Le système physique est commandé par une PWM +/- .
7. Le μC prend en compte des tensions. Le capteur de consigne est le potentiomètre d'entrée et le capteur de sortie (grandeur physique à asservir) est le capteur de courant.

10. Schéma bloc :

Construction du schéma bloc : Pour construire ce schéma, nous avons du analyser plusieurs éléments du système dont : le fonctionnement du moteur, de la carte électronique et du capteur de courant.

Ce sont donc les éléments que nous retrouvons ci-dessous :

- Le moteur est modélisé par la fonction de transfert d'un moteur à courant continu. Il s'agit d'un premier ordre avec un gain $K1 = 1/R$ et une constante de temps $\tau_1 = L/R$. Nous avons choisi de négliger la force électromotrice du moteur E .

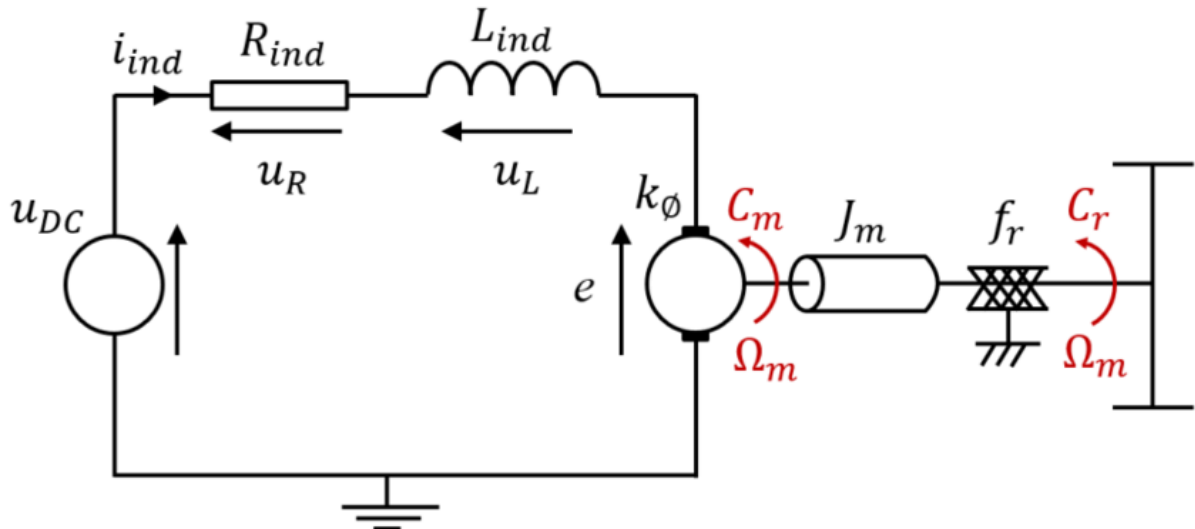
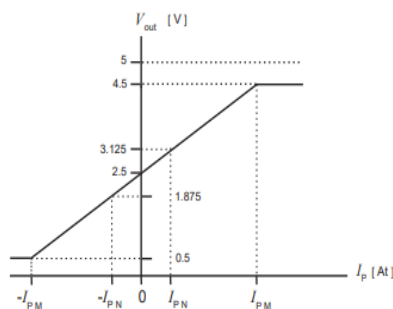


Schéma de principe du moteur CC

- Ensuite, nous avons modélisé le capteur de courant comme un simple gain de valeur 0.104. En effet, en regardant la datasheet de ce dernier, on voit que pour notre configuration sur la carte il se comporte de manière linéaire (équation de type $ax+b$).

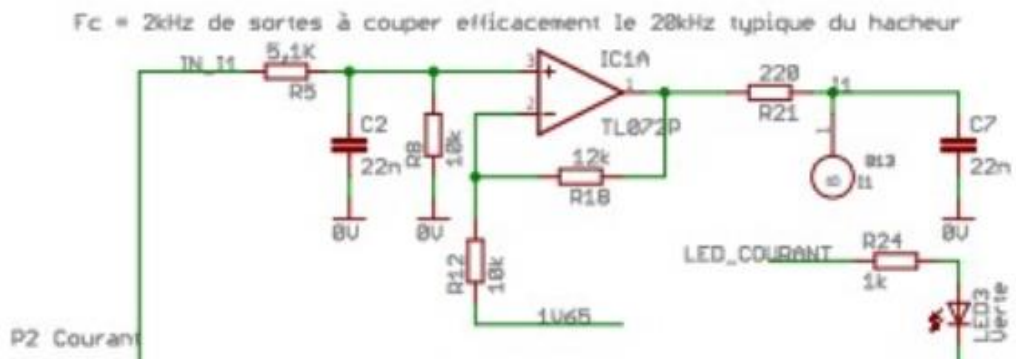
Output voltage - Primary current



| Number of primary turns | Primary nominal RMS current | Nominal output voltage V_{out} | Primary resistance R_p [mΩ] | Primary insertion inductance L_p [μH] | Recommended connections |
|-------------------------|-----------------------------|----------------------------------|-------------------------------|---|-------------------------|
| 1 | ±6 | 2.5 ± 0.625 | 0.18 | 0.013 | |
| 2 | ±3 | 2.5 ± 0.625 | 0.81 | 0.05 | |
| 3 | ±2 | 2.5 ± 0.625 | 1.62 | 0.12 | |

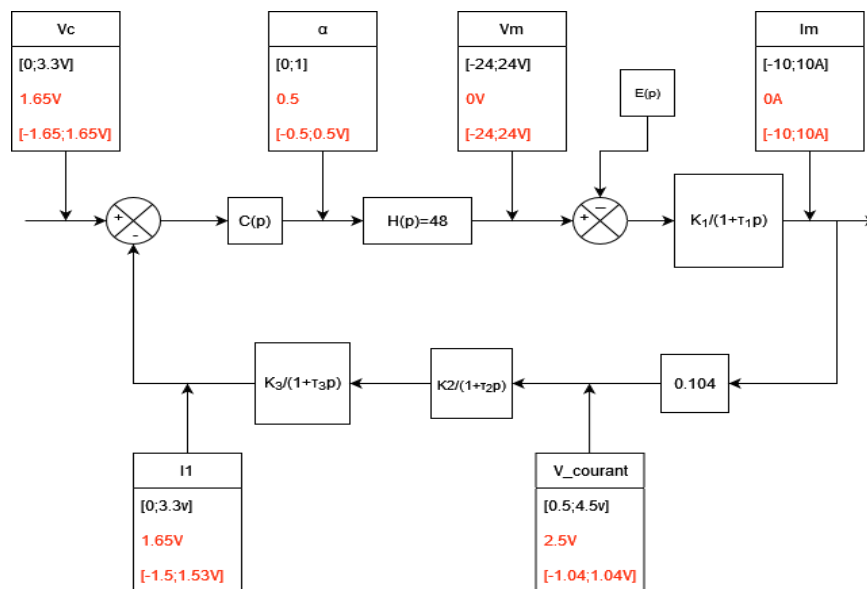
Caractéristiques du LTS-6 NP.

- Enfin, nous avons modélisé le filtre RC en sortie du capteur de courant (avant l'AOP) par la fonction de transfert d'un premier ordre également de gain K_3 et de constante de temps τ_3 . Pour ce faire nous avons calculé les impédances équivalentes puis appliqué les équations du RC. De même sorte nous avons calculé le filtre en sortie, en tenant compte du gain introduit par l'AO non inverseur (modélisé par K_2).



Capture de l'étage de sortie du capteur sur la carte

On en déduit alors le schéma bloc suivant :



| | | |
|----------------------------|------------------------------|-----------------------------|
| $K_1 = 1$ | $K_2 = 2.2$ | $K_3 = 0.662$ |
| $\tau_1 = 2\text{ms}$ | $\tau_2 = 5\mu\text{s}$ | $\tau_3 = 74\mu\text{s}$ |
| $f_{c1} = 79.57\text{ Hz}$ | $f_{c2} = 31.831\text{ kHz}$ | $f_{c3} = 2.141\text{ kHz}$ |

Point cours hacheur :

On entre avec $V=24V$ (tension batterie). Le hacheur $4Q$ est un convertisseur de puissance. Il adapte la quantité d'énergie, et le courant qui sera régulé par le μC .

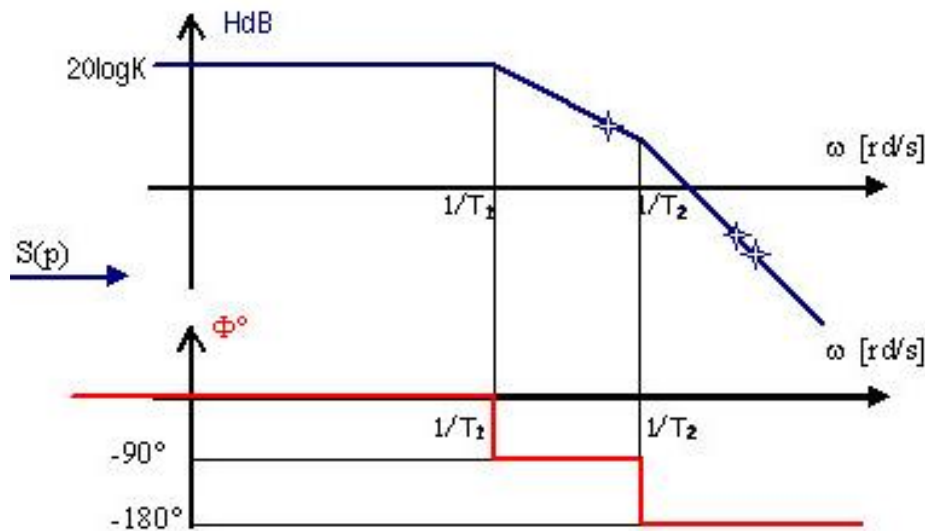
Le hacheur est basé sur la saturation d'un transistor à très haute fréquence. L'inertie moteur lui permet de ne pas voir le rapport cyclique.

1. ÉTUDE DE LA STABILITÉ DE LA FTBO ET CHOIX DE CORRECTEUR :

1.1 Simplification de $G(p)$:

Notre fonction de transfert $G(p)$ est celle d'un troisième ordre. Elle possède trois pôles caractérisés par trois constantes de temps (équivalent fréquences de coupures) différentes. Lorsque l'on passe en fréquences ces trois constantes de temps, on remarque que l'on aura une fréquence de coupure à 80Hz, une fréquence de coupure aux alentours de 2kHz et l'autre à environ 30kHz.

On remarque rapidement que la dynamique du pôle qui à pour fréquence de coupure 30kHz n'aura pas une grande influence sur le comportement de notre système, elle est donc négligeable. Nous pouvons alors l'approximer par un second ordre qui s'écrit : $G(p) = \frac{K}{(1+\tau_1 p)(1+\tau_2 p)}$. Nous pouvons alors tracer qualitativement le diagramme de BODE de notre système approximé :



Diagrammes de Bode illustratifs d'un second ordre

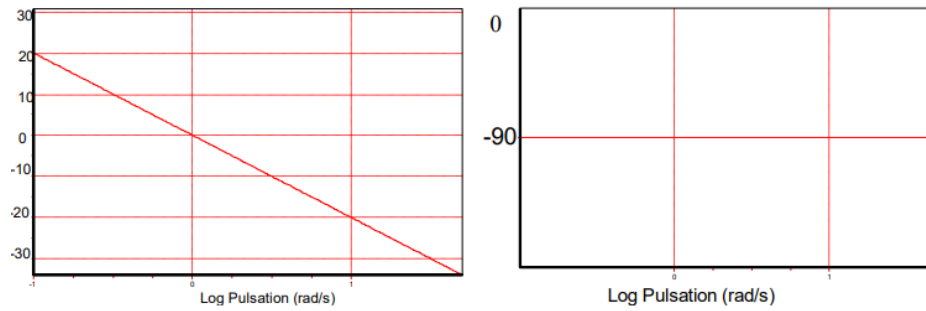
L'objectif de notre correcteur sera donc de placer notre fréquence de transition en BO de sorte à ce que la condition sur la marge de phase soit satisfaite, c'est-à-dire que $\varphi > 45^\circ$ à f_T . Pour cela, nous allons positionner f_T à 400Hz.

1.2 Idée du correcteur proportionnel :

En premier lieu, nous pouvons penser à mettre en place un simple correcteur proportionnel avec un gain K choisi de sorte à faire bouger la fréquence de transition pour atteindre la marge de phase souhaitée. Néanmoins, cette solution présente de forts risques d'instabilité en fonction de la valeur de K , malgré le fait qu'il rende notre système plus rapide.

1.3 Idée du correcteur intégral :

Dans un second temps, on peut penser à mettre en place un correcteur intégral, qui permettrait de supprimer l'erreur en régime permanent (intégrateur pur) mais aussi de déplacer notre fréquence de transition. Encore une fois, nous ne choisirons pas cette solution, car seul, il faudrait que nous placions la fréquence de transition à 80Hz pour satisfaire les conditions de marge de phase (le correcteur intégral modifiant la phase du système de -90°).



Diagrammes de Bode illustratifs d'un correcteur proportionnel

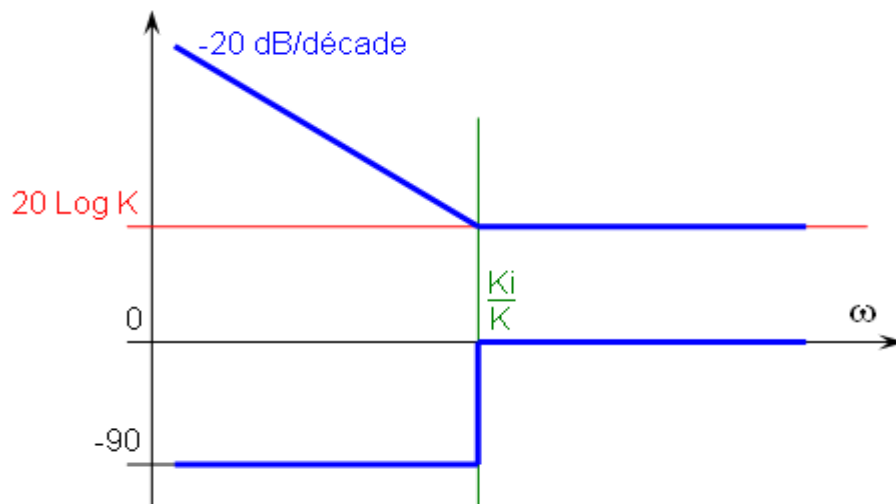
1. 3 Idée du correcteur PI :

Enfin, nous avons aboutit au correcteur PI (Proportionnel Intégral) de la forme $PI = K + \frac{Ki}{\tau p}$ pour mettre en place une compensation de pôle dominant.

Le but de la compensation de pôle, est de placer la fréquence de coupure du correcteur sur celle du pôle à compenser, de sorte à rajouter un zéro en BF qui va supprimer ce pôle (ou du moins atténuer très fortement son effet sur le système réel).

Si on étudie rapidement le correcteur PI :

- en basses fréquences, i.e. $j\omega \rightarrow 0$, $PI(j\omega) \approx \frac{K}{\tau j\omega}$, il se comporte donc comme un intégrateur pur de fréquence de coupure $\frac{1}{2\pi\tau}$ et sa phase vaut -90° .
- en hautes fréquence, $PI(j\omega) \approx K$, donc un proportionnel pur de phase nulle.



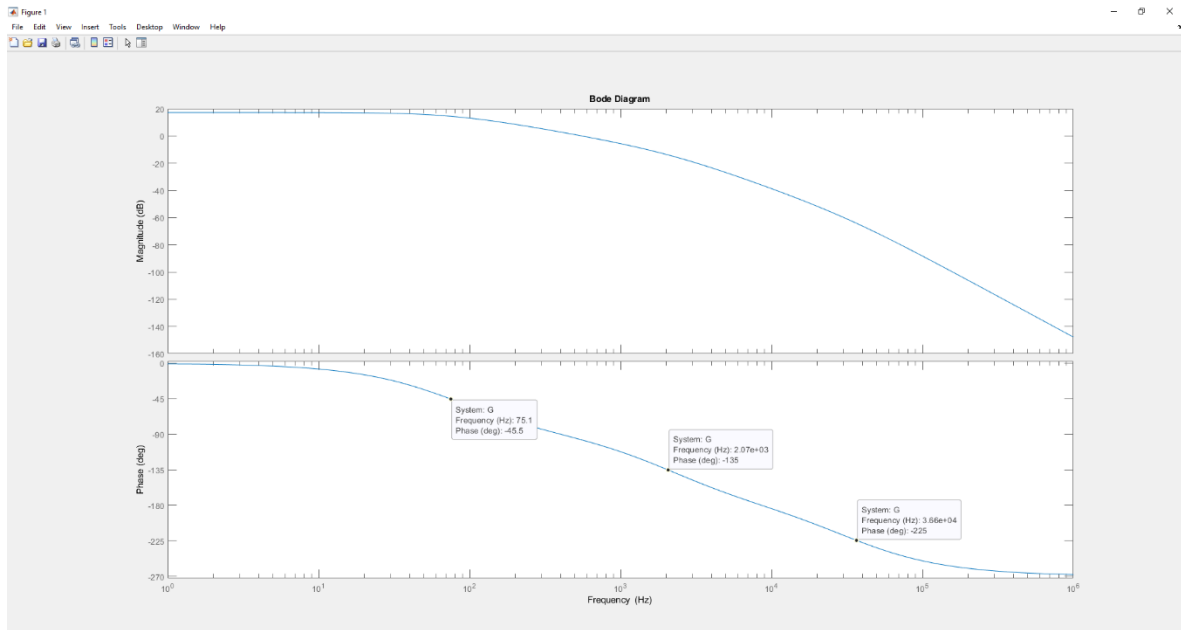
Diagrammes de Bode illustratifs d'un correcteur PI

En somme, on obtient que la phase est croissante (de -90° vers 0°) elle va donc permettre de baisser la phase de notre système en basse fréquence et donc d'augmenter la marge de phase. L'avantage de ce type de correcteur est qu'il va nous permettre d'atteindre l'erreur nulle en régime permanent de par le caractère intégrateur du PI.

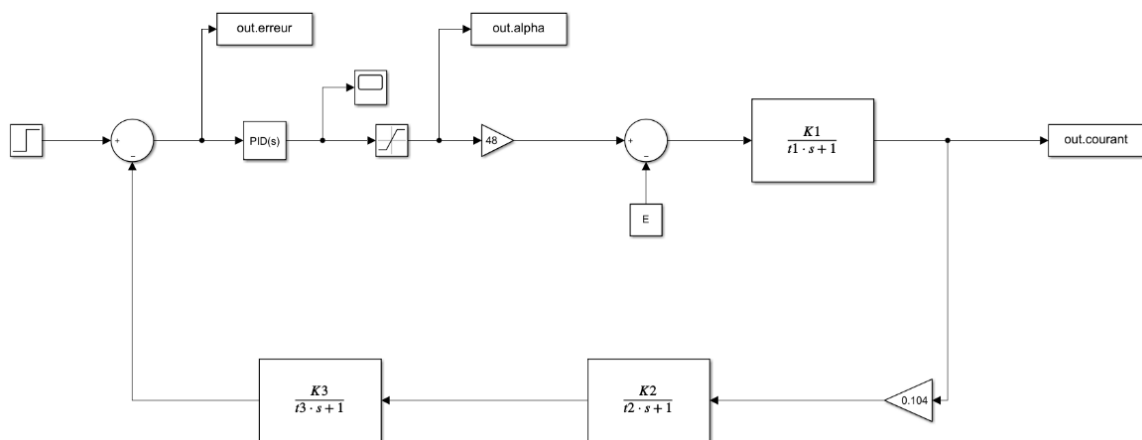
2. ETUDE DU CORRECTEUR C SOUS MATLAB :

2. 1 Squelette du script MATLAB:

Dans un premier temps, nous avons écrit un script MATLAB (*cf. GitHub/MATLAB*) dans lequel nous avons fait figurer tous les éléments présents dans la fonction de transfert du système G , c'est-à-dire les gains K_1, K_2, K_3 ainsi que celui du capteur de courant. Mais également, les constantes de temps/fréquences de coupures de chaque pôles. L'idée a été ensuite de modéliser tous les sous-systèmes par trois fonctions de transfert, $G_{1,2,3}$ pour obtenir notre système non corrigé et le tracer en boucle ouverte pour vérifier l'étude faite précédemment. Nous avons tracé donc sur MATLAB notre système en boucle ouverte:



On peut donc confirmer que notre marge de phase est bonne, car à f_T on a environ 80° de marge de phase ce qui est supérieur à 45° . Nous avons alors construit un diagramme Simulink avec saturateur sur α pour tester le comportement de $C(p)$ en continu :



2. 2 Correcteur $C(p)$ dans le domaine continu:

Dans un premier temps, nous avons implémenté le correcteur en continu (domaine de Laplace). Nous sommes partis sur un correcteur PI, avec une fréquence de coupure placée à 80Hz pour compenser ce pôle dans le système en BF. Le but était de placer la fréquence de transition à 400Hz.

Pour cela, nous avons déterminé la relation entre K_p et K_i (cf. *GitHub/MATLAB*) de sorte à compenser le pôle, puis nous avons calculé K_i de sorte à régler la marge de phase souhaitée. On remarque ici un léger overshoot dû à la saturation (problème de Simulink qui ne repère pas le saturateur au départ).

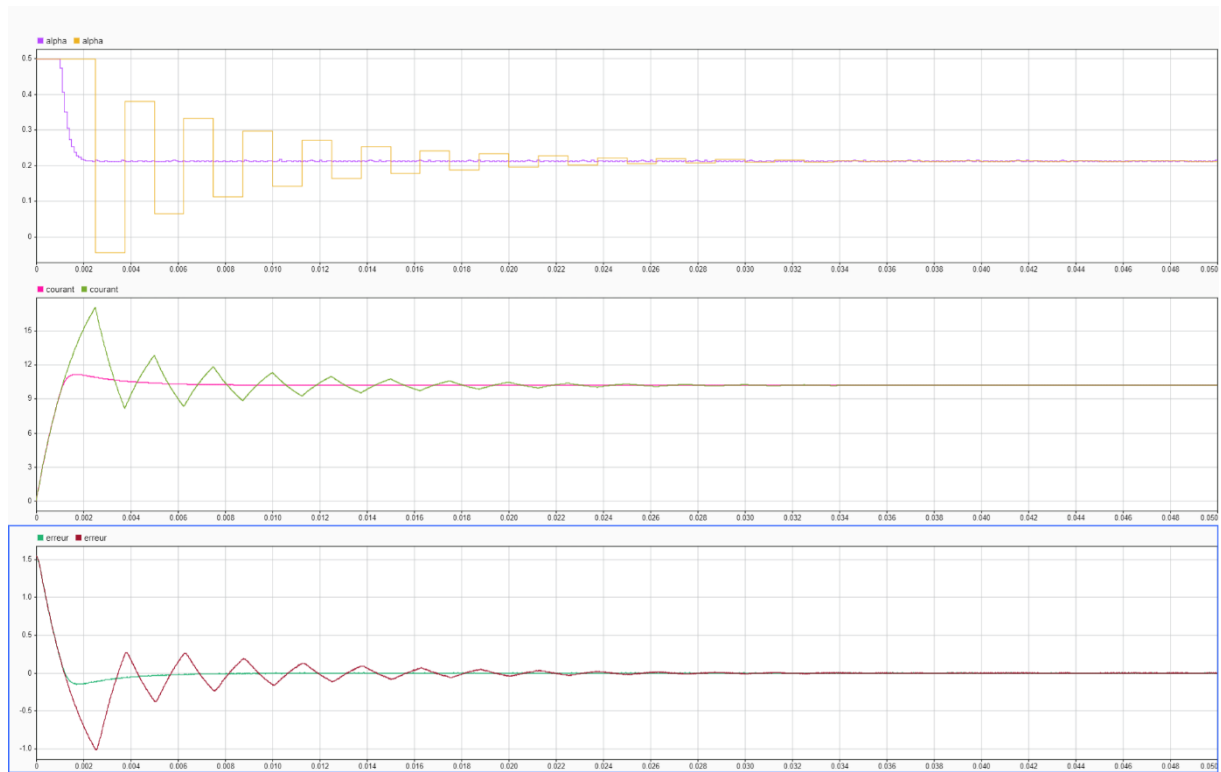


Résultat à un échelon unitaire du système corrigé pour le courant moteur I , α et l'erreur.

2. 3 Correcteur $C(z)$ dans le domaine discret:

Nous avons ensuite passé notre PID dans le Simulink en mode discret, pour observer que notre correcteur satisfaisait les spécifications du cahier des charges avant de l'implémenter sur microcontrôleur. En première approche, on a réglé la fréquence d'échantillonnage à 800Hz de sorte à satisfaire strictement le critère de Shannon (la bande passante du système vaut environ $f_T=400\text{Hz}$).

Or, la transformation bilinéaire introduit dans le système un retard pur en exponentielle, qui dépend de T_e ($\exp[-T_e \cdot p/2]$). Ce retard génère des oscillations dans le système, car il engendre une perte de marge de phase. Nous avons donc décidé de passer f_e à 10kHz, ce qui nous a donné des résultats plus que satisfaisants, car on colle quasiment au comportement en continu (10kHz étant acceptable car les CAN du STM32 fonctionnent à des vitesses largement supérieures et que la PWM est à 20kHz). Les résultats de nos expérimentations sont les suivants :



Résultats à un échelon pour $C(z)$ pour différentes valeurs de T_e

En jaune $f_e = 800\text{Hz}$, on satisfait purement Shannon, présence d'un overshoot de 70% de la valeur finale du courant. On remarque aussi un comportement oscillatoire. En mauve $f_e = 10\text{kHz}$, on remarque un léger overshoot. En effet, on a une valeur de courant qui monte à 10.5 A, soit un dépassement de 5% qui est lié au retard évoqué plus haut. On peut alors valider notre correcteur en z , et on peut passer à l'implémentation sur Keil.

3. IMPLEMENTATION SUR KEIL :

3. 1 Calcul de l'équation de récurrence:

On sait que $C(p) = \frac{\text{Sortie}}{\text{Entrée}} = \frac{\alpha}{\varepsilon} = Kp + \frac{Ki}{\tau p}$. En appliquant la transformée bilinéaire en z^{-1} , on obtient que : $C(z^{-1}) = K_p + \frac{Te \times Ki}{2} \frac{1-z^{-1}}{1+z^{-1}}$ et en développant on obtient finalement l'équation de récurrence à implémenter dans Keil :

$$\alpha = \varepsilon z^{-1} \left(K_p - \frac{Te}{2} K_i \right) + \varepsilon \left(K_p + \frac{Te}{2} K_i \right) - \alpha z^{-1}$$

Où les termes en z^{-1} représentent les valeurs précédentes des variables qu'ils suivent. On voit donc apparaître une structure de tableau pour les variables α et ε . Dans notre implémentation nous avons choisi un tableau de taille deux avec l'indice 0 pour la valeur actuelle et l'indice 1 pour la valeur précédente.

3. 2 Implémentation dans Keil:

D'abord nous définissons en `#define` toutes les valeurs constantes de notre système (Gains, fréquences d'échantillonnage, valeurs de résistances,...) de sorte à avoir un code qui s'adapte au aléas du monde réel comme des défauts sur une inductance/résistance par exemple. Les bibliothèques de la carte de puissance nous étant fournies, nous n'avons qu'à coder dans l'interruption, qui surviendra à la fréquence T_e , notre équation de récurrence ainsi que la modification des rapports cycliques de la PWM (via la fonction `R_Cycle(...)`).

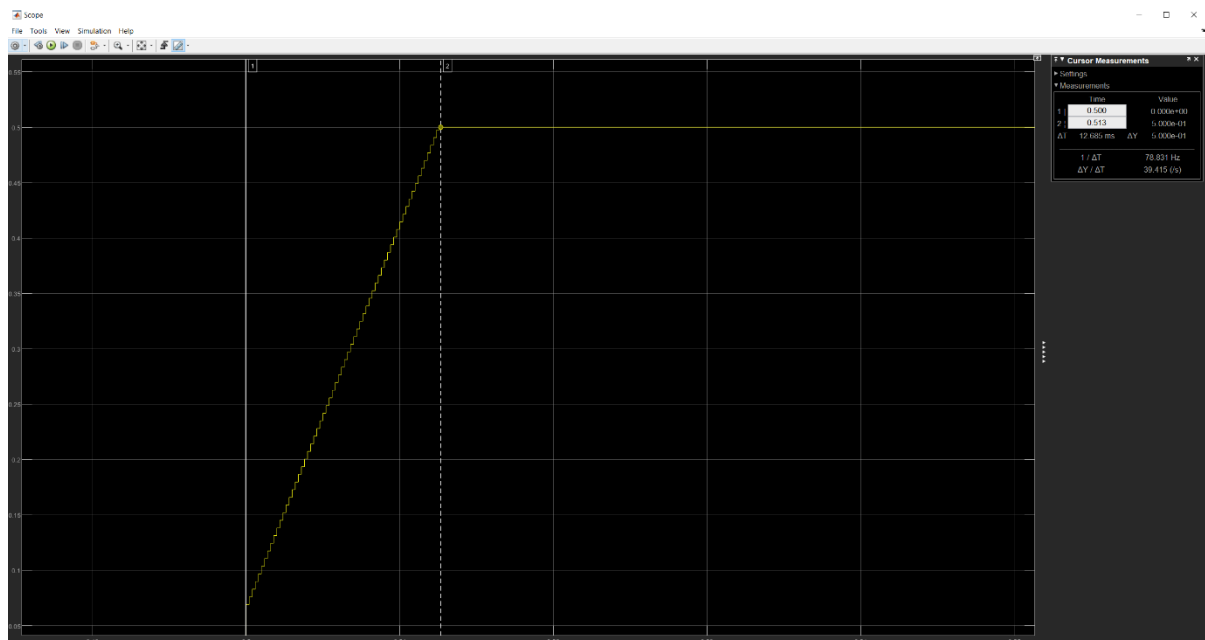
Les drivers sont déjà donnés, nous avons donc quand même vérifié que la polarité de la PWM était inversée, pour savoir quoi rentrer dans la fonction `R_Cycle`.

Lorsque nous codons dans Keil, toutes les valeurs analogiques (tension image du courant par le capteur, rapport cyclique α) doivent être converties en valeur numérique entre 0 et 4095 pour être interprétables par le STM32.

Pour récupérer les valeurs précédentes dans le calcul de notre correction, nous avons mis en place un système de buffer qui sont en réalité des tableaux de deux valeurs (car nous n'avons besoin que de la $n-1$ ème valeur).

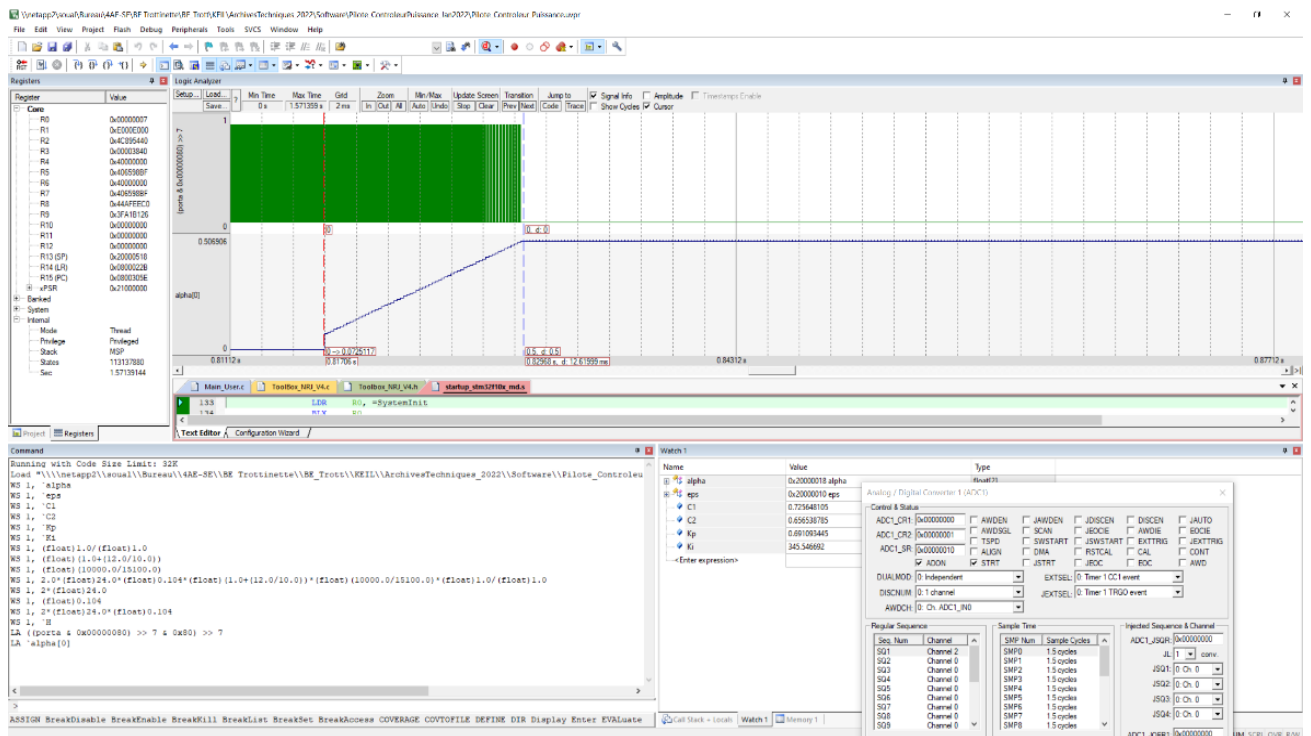
3. 3 Simulation sous Keil:

Pour pouvoir flasher notre code sur la carte et tester sur la maquette, nous devons être sûrs que le comportement de notre code est conforme à ce que nous avons simulé sous MATLAB. Pour cela, nous avons isolé le correcteur sous MATLAB et nous lui avons envoyé échelon d'amplitude 0.1V. Le comportement sera conforme si le comportement en forme de rampe au début de l'échelon est de la même durée sur les deux simulations.



Simulation du correcteur seul sous Simulink

On voit que la durée de la rampe est d'environ 12.685ms. Sous Keil, nous avons mis 0.1V en entrée sur l'ADC pour simuler la commande de la trottinette, on remarque aussi que notre correcteur seul possède la même caractéristique temporelle :

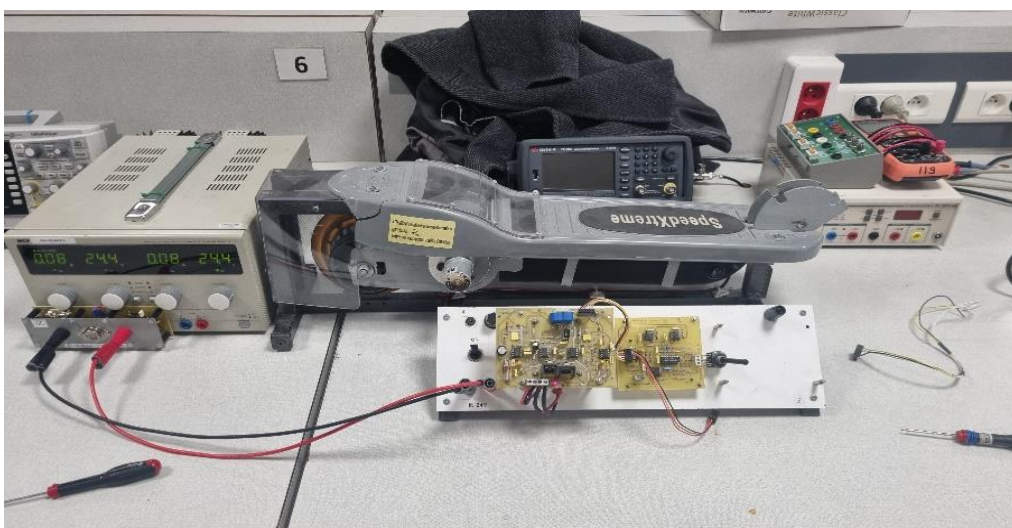


Simulation du correcteur sous Keil

On peut donc valider le comportement du correcteur, et lancer les tests sur la cible réelle.

4. SYNTHÈSE SUR LE MODÈLE REEL :

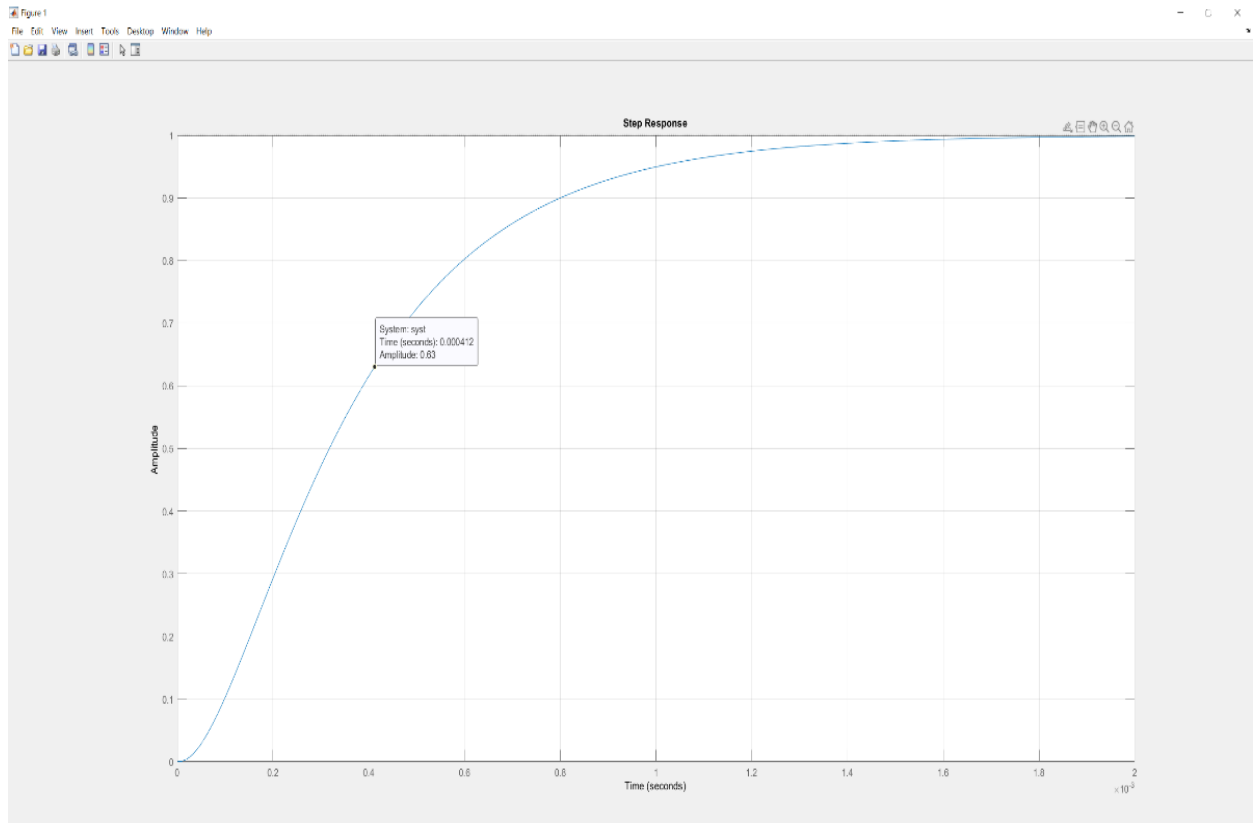
Pour tester notre le système en conditions réelles, nous allons réaliser deux manipulations classiques de test en automatique, et les comparer a un résultat de simulation donné par MATLAB. Nous réaliserons donc dans un premier temps un test fréquentiel (tracé du diagramme de Bode) et la réponse à un échelon.



Maquette des tests

4. 1 Echelon unitaire:

On attaque la carte via le GBF avec un signal carré d'amplitude 100mVpp, un offset de 1.625 V (représente le zéro) et une fréquence relativement basse de 1Hz. Le but est d'observer la sortie de notre système et de calculer la constante de temps et le temps de montée pour le comparer aux simulations du modèle sous MATLAB.



Sortie MATLAB test unitaire

On réalise d'abord un premier test avec moyennage. Sur la sortie oscillo, on va mesurer la constante de temps en utilisant les fonctions de mesure. On cherche la constante de temps, notre échelon à une valeur finale de 800mV, d'où $\tau = 0.63 \cdot 800\text{mV} = 504\text{mV}$. En mesurant, on observe que $\tau = 580\mu\text{s} = 0.580\text{ms}$. On observe un écart avec le résultat de MATLAB, on a $\Delta = \frac{412-580}{580} = 28\%$. Cet écart peut être expliqué par plusieurs facteurs :

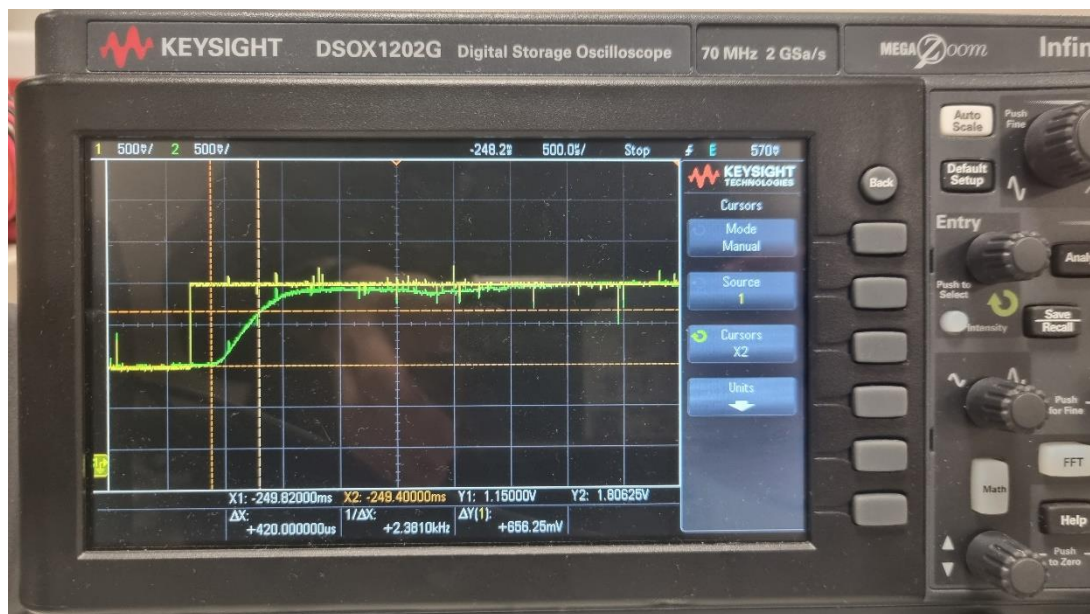
- Sur le système réel, on voit que le système ne réagit pas au même moment que le signal d'entrée. Si l'on mesure dès le moment où le système réagit, on obtient une constante de temps qui vaut $\tau = 512\text{ms}$ et donc un écart réduit à 18%.
- On peut aussi considérer que le système réel n'est pas parfait comme le système simulé, il y a des erreurs de valeurs sur les composants (tolérances de résistances,...) qui peuvent impacter l'erreur.

- On peut considérer que le moyennage affecte le comportement.



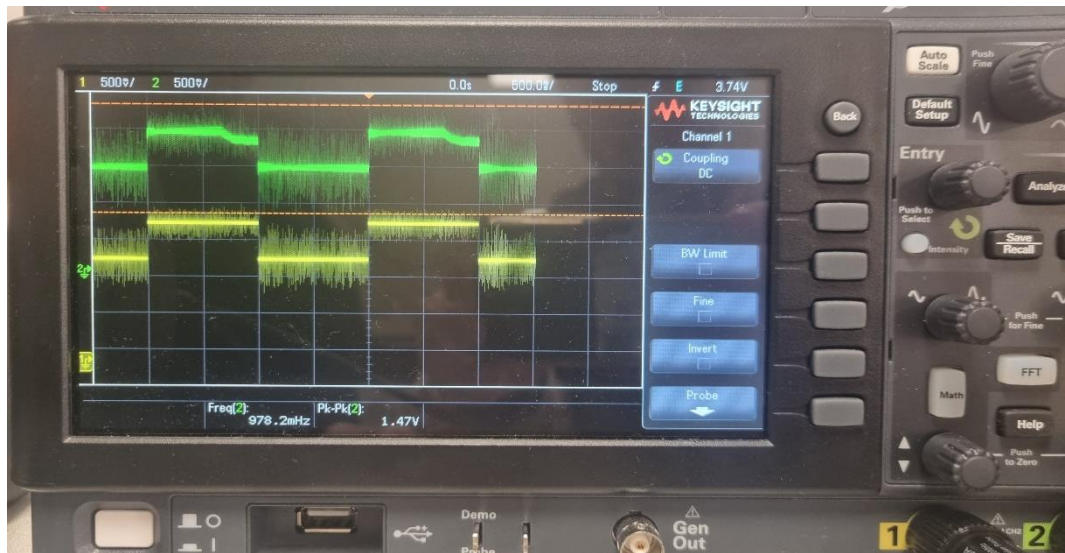
Sortie de l'oscilloscope moyenné

On a ensuite testé le système réel sans moyennage. On remarque sur la sortie oscillo que notre constante de temps finale vaut $\tau = 420\text{ms}$, ce qui constitue une erreur de $\Delta = \frac{412-420}{420} = 2\%$. On peut alors dire que ce résultat est satisfaisant, les écarts s'expliquant par les incertitudes du modèle physique, et aussi de certains bruits parasites.



Sortie de l'oscilloscope sans moyennage

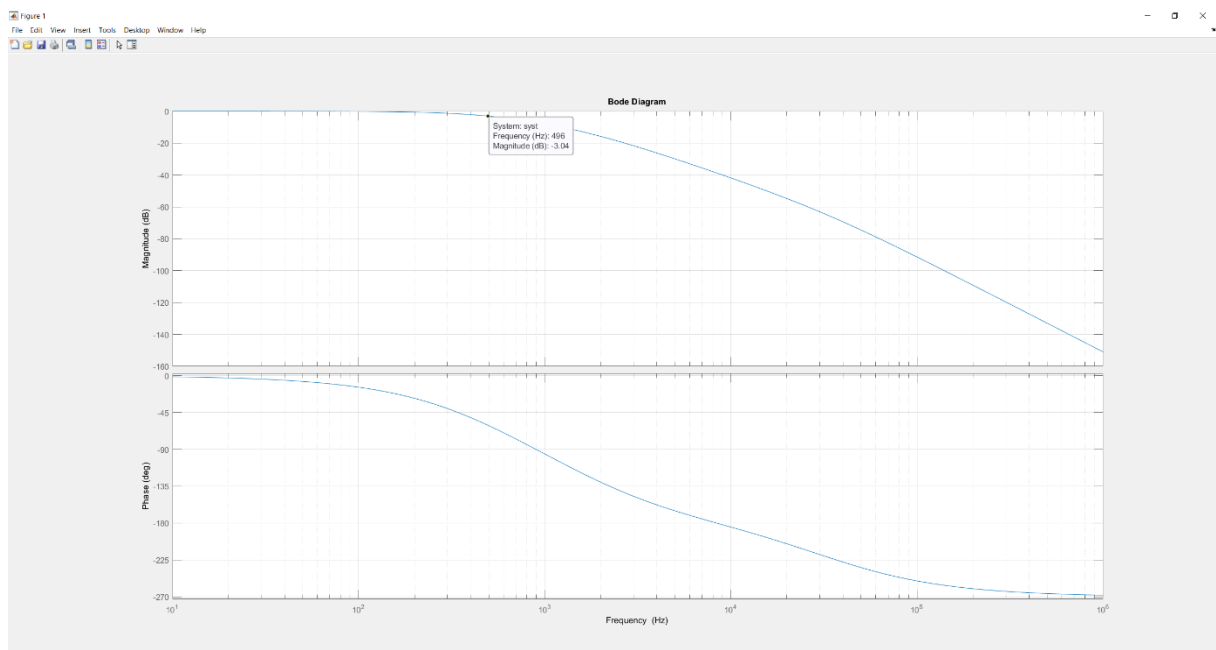
Note : Quand la fréquence du signal carré est basse, on remarque une saturation de notre sortie. En fait, pour réguler le courant, le système demande au hacheur une tension de plus de 24V qu'il n'est pas capable de fournir et le courant est alors saturé (car la tension induite dans le moteur E vient parasiter le système). Dans la vraie vie, la tension délivrée par la batterie sera fortement dépendante de la topologie du terrain (pente, frottements,...) ce qui fait que cette situation ne se produira donc pas à basses fréquences, car le courant requis pourra être délivré.



Sortie de l'oscilloscope quand la fréquence est basse

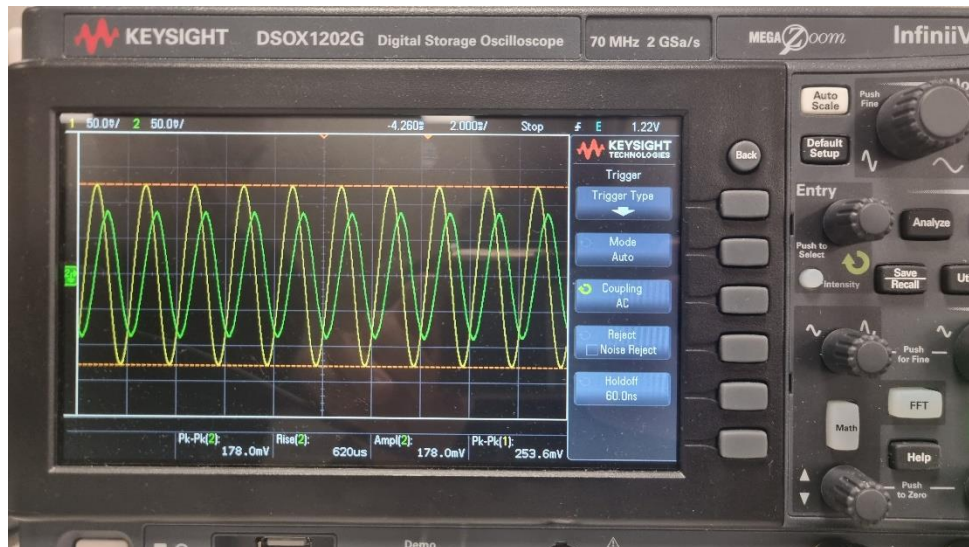
4. 2 Test fréquentiel:

Pour ce test, nous envoyons dans la carte, via l'entrée GBF, un sinus d'amplitude 250mVpp avec un offset de 1.65V (pour caler le 0). Le but de ce test est de vérifier les fréquences de coupure et de transition.

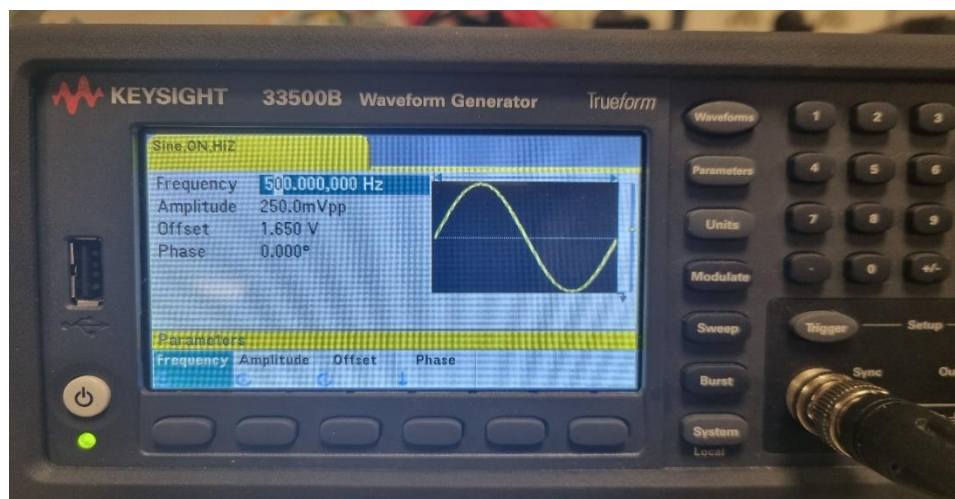


Bode du système

Dans un monde idéal, le système corrigé possède une fréquence de coupure à 496Hz qui est placée par le correcteur ainsi qu'une autre à 2kHz (qui ne sera probablement pas visualisable). Pour cela, on va modifier la fréquence du signal d'entrée sur le GBF de sorte à obtenir une division d'amplitude par $\sqrt{2}$ ce qui est représentatif d'une fréquence de coupure. Dans notre cas, on stoppe le test quand le signal de sortie est atténué jusqu'à atteindre $\approx 177\text{mV}$.



Sortie de l'oscilloscope pour l'essai en fréquence



Réglage GBF pour attendre la fréquence de coupure

On voit alors que notre fréquence de coupure est atteinte pour $f=500\text{Hz}$, ce qui est presque exactement égal à la valeur théorique espérée, l'écart est inférieur à 1%. On peut donc valider le comportement en fréquences de notre correcteur.

Tableau de synthèse des écarts et des tests :

| Test | Amplitude (mVpp) | Offset (V) | Valeur théorique | Valeur mesurée | ε (%) |
|-----------------------|------------------|------------|---------------------|---------------------|-------------------|
| Réponse indicielle | 100 | 1.65 | $\tau=412\text{ms}$ | $\tau=420\text{ms}$ | 2 |
| Réponse fréquentielle | 250 | 1.65 | $F_c=496\text{Hz}$ | $F_c=500\text{Hz}$ | 0.8 |

Alors, au vu des résultats de ces deux tests, nous pouvons dire que notre correcteur satisfait les exigences du cahier des charge et nous pouvons donc valider notre solution.

5. UN MOT SUR LA REGULATION DE VITESSE :

Lors de toute la mise en place de la régulation de couple, nous avons contrôlé le courant envoyé dans le moteur. Cette analyse/mise en place s'est fait dans une approche dite de « boîte blanche », c'est-à-dire que l'ensemble du modèle était détaillé et connu.

Pour la mise en place de la régulation en vitesse, non traitée, la philosophie aurait été de placer un correcteur en amont de la régulation de courant, qui aurait traité une erreur entre la vitesse de consigne et le retour de vitesse mesuré par le tachymètre. Le correcteur aurait alors lui-même fourni le courant qui aurait été corrigé pour asservir le couple ensuite.

Dans le but de le réaliser, nous aurions adopté une approche « boîte noire », qui aurait consisté à caractériser le modèle en faisant des tests de consigne d'entrée et en mesurant la sortie sur le tachymètre. On aurait alors pu en tirer une quantité non négligeables d'informations (ordre du système, tracer un Bode, peut-être des pôles,...) et on aurait pu mettre en place une correction sur le STM32 à la manière de ce qui à été réalisé précédemment.

CONCLUSION :

Une fois notre solution validée, nous avons pu la tester sur le modèle réel, c'est-à-dire la trottinette sur batterie, et nous avons vu que cela fonctionnait bien. On a également pu ressentir le léger dépassement de consigne lorsque la commande est trop forte , mais également la phase de récupération d'énergie de la batterie lors du freinage !

Pour résumer les compétences mobilisées et acquises pendant ce bureau d'étude, on peut dire que nous avons utilisé :

- Une grande partie de l'enseignement de systèmes bouclés, notamment sur l'analyse de la FTBO dans le domaine de Laplace ainsi que pour la mise en place et l'étude des correcteurs (Proportionnel, Intégral, PI,...)
- Des connaissances en commande numérique, en particulier sur le passage du monde continu au monde discret (Transformée en Z)
- Quelques connaissances sur les périphériques du STM32 pour être capables de simuler avec Keil, mais aussi de comprendre la librairie qui nous était fournie.
- Des notions d'analyse de circuits électroniques pour interpréter la carte, et calculer les fonctions de transfert des filtres à l'interface du capteur de courant.