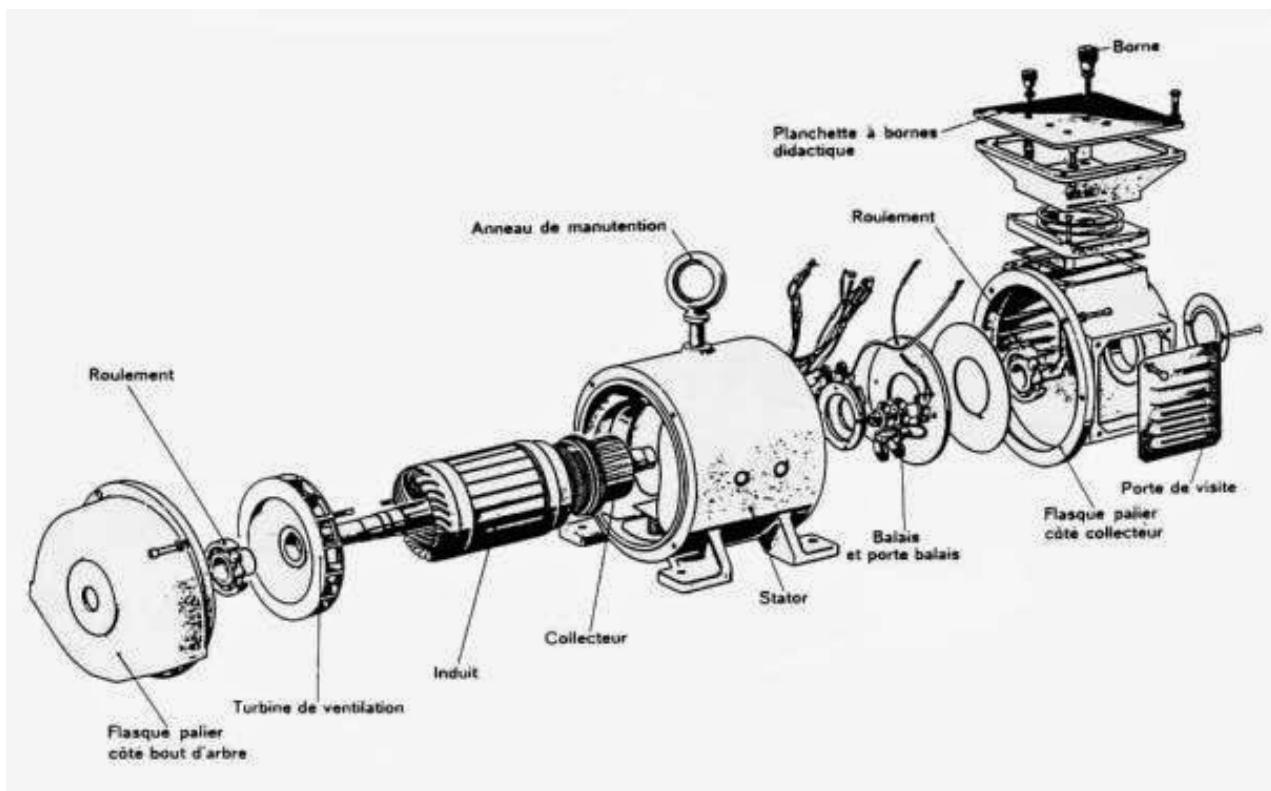


# Asservissement d'une machine à courant continu

PROJET GE1



**Emma BARETY, David MARTINEZ et Iona GUTH**

Matière du projet: Électrotechnique 3 / Automatique 3

Professeur: M.Flieller

Semestre d'étude: Semestre 7

Date du projet: 21/09/2022 - 16/01/2022



## Table des matières

<b>1</b>	<b>Introduction</b>	<b>4</b>
<b>2</b>	<b>Cahier des charges</b>	<b>5</b>
<b>3</b>	<b>Planification</b>	<b>8</b>
<b>4</b>	<b>Schéma de la MCC et son hacheur</b>	<b>9</b>
4.1.	Simulation et modélisation du moteur à vide	9
4.2.	Simulation et Modélisation du moteur avec pertes	12
4.3.	Modélisation du hacheur	15
4.4.	Simulation et Modélisation du Moteur avec pertes et génératrice	16
<b>5</b>	<b>L'asservissement du moteur</b>	<b>20</b>
5.1.	Rappel du cahier des charges pour l'asservissement	20
5.2.	Asservissement du courant traversant le moteur	20
5.3.	Asservissement en vitesse	29
5.4.	Écrêteur : comment limiter le courant dans la MCC	36
5.5.	Type d'asservissement	37
5.6.	Schéma final	39
<b>6</b>	<b>Conception des PCB</b>	<b>41</b>
6.1.	Choix et organisation de nos PCB	41
6.2.	Les cartes électroniques obtenues	43
<b>7</b>	<b>Tests et essais sur la MCC</b>	<b>48</b>
7.1.	Tests sur plaquette	48
7.2.	Tests sur la machine à courant continu (MCC)	50
<b>8</b>	<b>Sources</b>	<b>54</b>
<b>9</b>	<b>Mots clés</b>	<b>55</b>
<b>Table des figures</b>		<b>57</b>
<b>10</b>	<b>Liens des fichiers des figures</b>	<b>59</b>

# 1. Introduction

Dans le cadre de notre formation d'ingénieur en génie électrique, nous réalisons un projet qui allie électrotechnique et automatique. Ce projet nous permettra de mettre en œuvre nos connaissances afin de réaliser un système de commande décrit par un cahier des charges fixé. Il nous permettra aussi d'acquérir de nouvelles connaissances et une démarche pour les projets (gestion des étapes et du temps ...).

Le but de ce projet est de réaliser un asservissement d'une machine à courant continu (nommée MCC par la suite) alimentée par un hacheur. Plus particulièrement, nous réaliserons deux asservissements : un asservissement de vitesse et un asservissement de courant (boucle interne de courant). Un capteur de courant placé sur le hacheur nous permettra de mesurer le courant afin de réaliser cet asservissement. L'asservissement vitesse présente deux retours possibles : avec un capteur tachymétrique et avec un codeur incrémental.

Afin de mener à bien sa réalisation, nous allons partager le projet en différentes étapes qui seront détaillées dans les différents rapports :

- Modélisation et simulation de la MCC et du hacheur sur PSIM et Matlab-Simulink. Les deux simulations devront donner des résultats similaires durant tout le projet.
- Modélisation et simulation des asservissements en courant et vitesse sur PSIM et Matlab-Simulink. Des correcteurs et limiteurs nous aideront à le réaliser. Dans cette phase, nous simulons également le codeur incrémental et dimensionnons les composants.
- Conception et réalisation d'un circuit imprimé permettant d'asservir en courant et vitesse la MCC.
- Tests finaux, finalisation du projet.

## 2. Cahier des charges

Nous devons asservir en courant (limitation de courant) et en vitesse une machine à courant continu avec une charge. L'objectif est donc de réaliser une carte de commande permettant ces asservissements.

Pour la réalisation du projet, nous disposons d'un hacheur quatre quadrants comprenant le capteur de courant, un banc moteur comportant deux machines à courant continu couplées mécaniquement ensemble. L'une fonctionne en moteur et l'une fonctionne en génératrice (avec charge). La génératrice et la charge forment la charge du moteur.

De plus, une dynamo tachymétrique et un codeur incrémental sont mis à notre disposition.

Le cahier de charges 9 du projet est suivant :

- L'asservissement9 en vitesse se fera par deux moyens différents : soit par un capteur tachymétrique soit par un codeur incrémental via un switch qui permet de sélectionner le capteur choisi.
- La vitesse maximale indiquée de 4800 tr/min ne doit pas être dépassée.
- La boucle d'asservissement du courant se fera grâce au capteur prévu à cet effet dans le module du hacheur.
- Le courant de saturation ne devra pas être supérieur à 1,1 fois le courant nominal (soit 4,95A) .
- Le dépassement en vitesse et en courant est limité entre 10 et 20%.
- Le temps de réponse doit être le plus faible possible.
- Le budget est de 25 euros par groupe. Certains composants (bornes, connecteurs) seront fournis.

Le schéma de la structure de la commande sera le suivant :

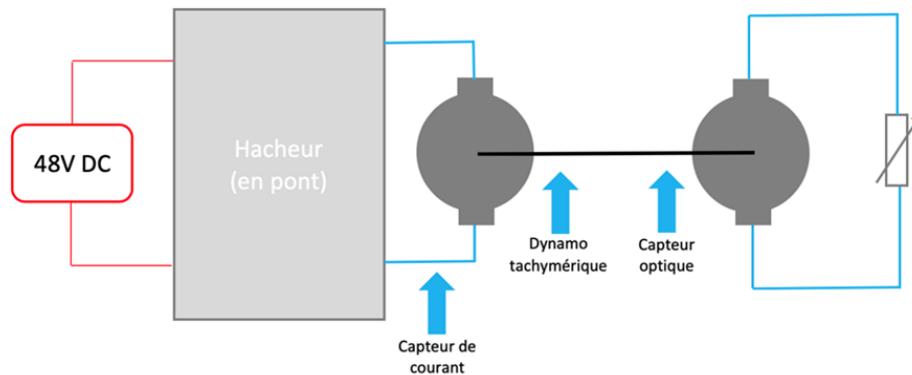


FIGURE 2.1 – Schéma de la MCC (Machine à courant continu) à asservir

Le moteur sera alimenté par le hacheur qui, avec le capteur de courant, nous permettra d'ajouter une commande en courant. Le rhéostat sera la charge de la génératrice et l'ensemble rhéostat + génératrice formera la charge du moteur. Sur l'arbre moteur entraînant la génératrice, la dynamo tachymétrique et le capteur optique sont déjà en place. Ces deux capteurs nous permettront de connaître la vitesse de rotation du moteur et donc de réaliser un asservissement en vitesse par deux moyens différents.

Les caractéristiques de la MCC 9 se retrouvent dans sa datasheet :

<i>Couple en rotation lente</i>	<b>0.54</b>	<i>N.m</i>	<i>M<sub>O</sub></i>
<i>Courant permanent rotation lente</i>	<b>4.5</b>	<i>A</i>	<i>I<sub>O</sub></i>
<i>Tension d'alimentation de définition</i>	<b>49</b>	<i>V</i>	<i>U</i>
<i>Vitesse de définition</i>	<b>3000</b>	<i>tr/mn</i>	<i>N</i>
<i>Tension maximale</i>	<b>65</b>	<i>V</i>	<i>Umax</i>
<i>Vitesse maximale</i>	<b>4800</b>	<i>tr/mn</i>	<i>Nmax</i>
<i>Courant impulsif</i>	<b>13</b>	<i>A</i>	<i>I<sub>max</sub></i>
<i>Fem par 1000 tr/mn (25°C)</i>	<b>13.3</b>	<i>V</i>	<i>Ke</i>
<i>Coefficient de couple électromagnétique</i>	<b>0.127</b>	<i>N.m/A</i>	<i>Kt</i>
<i>Couple de frottement sec</i>	<b>2.4</b>	<i>N.cm</i>	<i>Tf</i>
<i>Coefficient de viscosité par 1000tr/mn</i>	<b>0.53</b>	<i>N.cm</i>	<i>Kd</i>
<i>Résistance du bobinage (25°C)</i>	<b>1.52</b>	$\Omega$	<i>Rb</i>
<i>Inductance du bobinage</i>	<b>2.2</b>	<i>mH</i>	<i>L</i>
<i>Inertie rotor</i>	<b>0.000083</b>	<i>kg.m<sup>2</sup></i>	<i>J</i>
<i>Constante de temps thermique</i>	<b>7</b>	<i>min</i>	<i>T<sub>th</sub></i>
<i>Masse moteur</i>	<b>1.34</b>	<i>kg</i>	<i>M</i>

FIGURE 2.2 – Valeurs données par le constructeur

La tension nominale de notre moteur est de 49V et le courant nominal est de 4,5A. De plus, le moteur a une force électromotrice de 13,3V pour 1000 tr/min. Sa vitesse nominale est de 3000 tr/min.

Le coefficient de couple électromagnétique du moteur est de 0,127 N.m/A et le coefficient de viscosité est de 0,0053 N.m. La résistance de bobinage vaut 1,52  $\Omega$  et l'inductance de bobinage vaut 2,2mH. Enfin, l'inertie du rotor vaut 0,000083  $kg.m^2$ .

On prend les valeurs suivantes pour nos simulations et on utilise les notations suivantes :

TABLE 2.1 – Notations des variables pour le projet

Notations	Nom	Valeurs	Unités
$\Omega$	vitesse de rotation		rad/s
C	couple total		N.m
$I_{nom}$	courant nominal	4,5	A
$e_{nom}$	tension aux bornes de la machine	13,3	A
$C_m$	couple électromagnétique		N.m
$C_r$	couple résistant	0,012	N.cm
E	tension fournie par la machine	49	V
J	moment d'inertie dans la machine	$8,30 \cdot 10^{-5}$	N.m
$K_{phi}$	coefficient du couple électromagnétique	0,127	N.m/A
f	coefficient des pertes par frottements	$5,06 \cdot 10^{-5}$	N
L	inductance du bobinage	$2,20 \cdot 10^{-3}$	H
$C_0$	moment d'inertie du moteur	$2,40 \cdot 10^{-2}$	N.m
R	résistance du bobinage	1,52	$\Omega$
$R_{ch}$	résistance de charge	10	$\Omega$
N	vitesse de définition	3000	tr/min
$N_{max}$	vitesse de maximale	3000	tr/min

## 3. Planification

Afin de pouvoir nous organiser pour la réalisation du projet, nous avons réalisé un diagramme de Gantt.

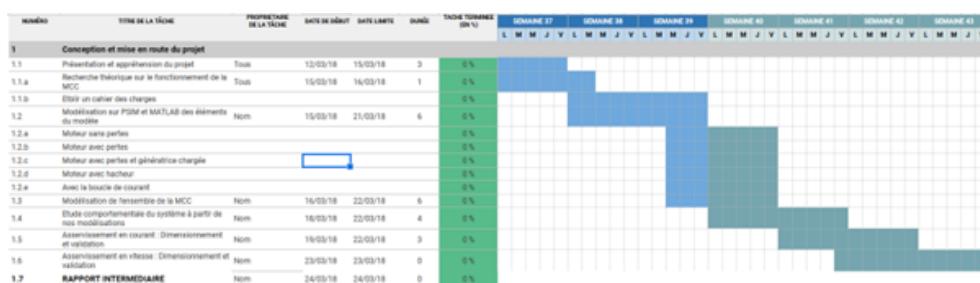


FIGURE 3.1 – Diagramme de Gantt prévisionnel 10

Cet outil prévisionnel nous a permis d'estimer le temps nécessaire à chaque étape, le travail qu'il nous restait à faire, de nous fixer des objectifs à court et long terme et de suivre l'avancement du projet. Il nous a aussi permis de nous répartir le travail de façon équitable, puisque nous avons, grâce à lui, identifié toutes les étapes à réaliser.

Nous l'avons adapté au fur et à mesure de notre avancement. Pour cette première partie du projet, nous avons réussi à tenir les délais et à réaliser tout ce qui était prévu (les différentes simulations) dans le temps imparti.

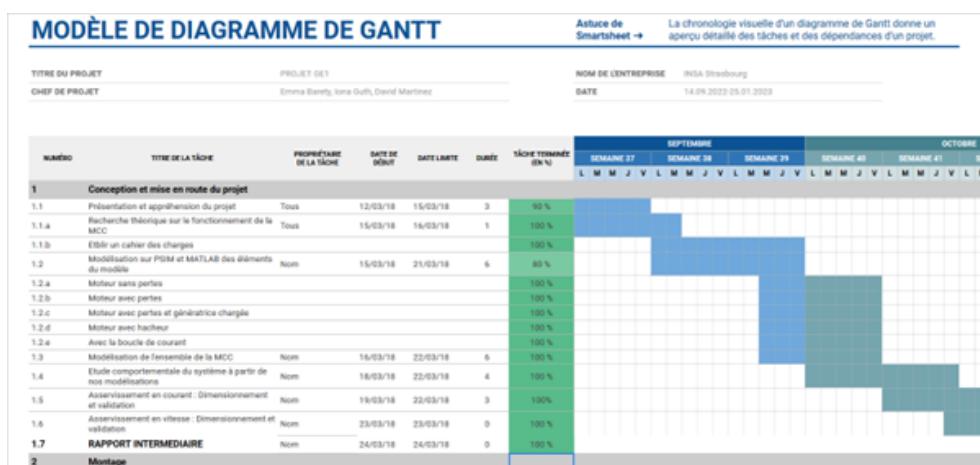


FIGURE 3.2 – Diagramme de Gantt réel10

## 4. Schéma de la MCC et son hacheur

### 4.1 Simulation et modélisation du moteur à vide

#### 4.1.1 Équations de la MCC

Une machine à courant continu (MCC) est une machine électrique. Il s'agit d'un convertisseur électromécanique permettant la conversion bidirectionnelle d'énergie entre une installation électrique parcourue par un courant continu et un dispositif mécanique ; selon la source d'énergie.

Formules d'une MCC sans pertes et sans charge

$$\begin{aligned} u(t) &= R \cdot i(t) + L \frac{di(t)}{dt} + e(t) \\ C_m(t) &= K_c \cdot i(t) \\ e(t) &= K_e \cdot \omega(t) \\ J \frac{d\omega}{dt} &= C_m(t) \end{aligned}$$

On note ici :  $K_e = K_c = K_\Phi$

Ce qui nous donne en Laplace :

$$\begin{aligned} I(s) &= \frac{U(s) - K_\Phi \Omega}{R + Ls} \\ C_m(s) &= K_\Phi \cdot I(s) \\ E(s) &= K_\Phi \cdot \Omega(s) \\ \Omega(s) &= \frac{K_\Phi \cdot I(s)}{Js} \end{aligned}$$

Après avoir établi les équations nécessaires à la modélisation, nous allons pouvoir représenter ce système dans Simulink, puis dans PSIM et comparer la simulation avec la modélisation.

#### 4.1.2 Simulation avec Simulink

Pour la simulation Simulink, nous avons essayé de représenter les équations suivantes sous forme de schéma bloc.

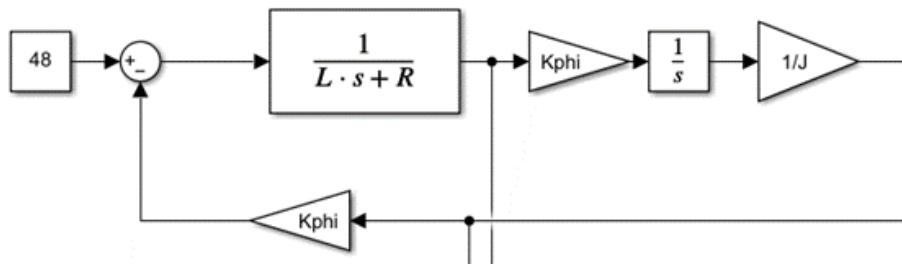


FIGURE 4.1 – Schéma bloc du moteur à vide 10

Les valeurs dans le schéma bloc sont définies en fonction du cahier des charges. Nous relevons la sortie du courant et de la vitesse dans un scope. Nous comparerons ces résultats aux résultats obtenus avec la modélisation PSIM.

### 4.1.3 Modélisation avec PSIM

Pour les simulations sur PSIM, nous allons utiliser une autre méthode d'approche.

En effet, en appliquant deux méthodes différentes, si on obtient les mêmes résultats, alors c'est que nos applications sont correctes. Nous allons donc directement prendre un moteur modélisé sur PSIM puis lui donner les caractéristiques de notre moteur réel.

Le moteur modélisé sur PSIM est une MCC standard avec un stator sous forme d'un circuit d'excitation et se présente ainsi :

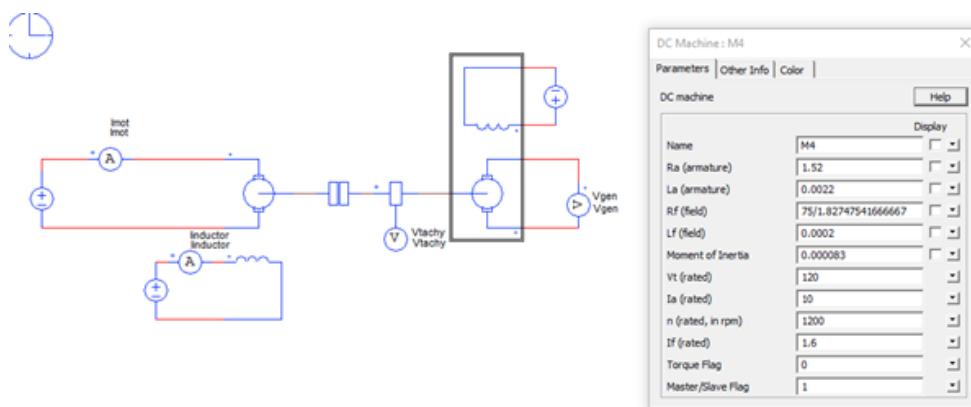


FIGURE 4.2 – Modélisation de la MCC à vide10

La modélisation sur PSIM nous permet de placer directement les composants pour lesquels on définit ensuite les propriétés. Elles se rapprochent donc plus du fonctionnement direct du système et ne nécessitent pas autant de calculs que la simulation sous Matlab. Donnons ici un exemple : pour le moteur, nous plaçons un « bloc » moteur et nous entrons ensuite les caractéristiques correspondant à notre moteur.

Toutefois, le logiciel ne bénéficie que d'un modèle de moteur à excitation séparée alors que notre moteur est un moteur à aimants permanents. Il faut ainsi dimensionner l'excitation de telle sorte à ce qu'elle corresponde au fonctionnement de notre machine à aimant permanent.

Méthode pour trouver l'équivalent entre  $U_f$ ,  $L_f$ ,  $R_f$  et  $K_\Phi$  :

Dans un premier temps, on souhaite que  $f$  soit très petit devant les autres constantes de temps du système. En effet, avec les aimants permanents, le flux apparaît directement. Ainsi,  $\tau_f \ll \tau_e$  et  $\tau_f \ll \tau_m$ . Cependant  $\tau_m < \tau_e$  donc on s'occupera pour les calculs, que de la première condition.

On a d'autre part :

$$\begin{aligned} - \tau_m &= \frac{J}{C_f} = \frac{8,3 \cdot 10^{-5}}{5,06 \cdot 10^{-5}} = 1,64s \text{ pour la constante de temps mécanique.} \\ - \tau_e &= \frac{R}{L} = \frac{1,52}{2,2 \cdot 10^{-3}} = 1,44s \text{ pour la constante de temps électrique.} \end{aligned}$$

On remarque donc bien que la constante de temps électrique est plus petite que celle mécanique.

Notre moteur fonctionnant avec des aimants permanents, nous devons fixer des caractéristiques au stator de notre moteur en simulation telles que celui-ci fonctionne de la même façon que notre moteur réel.  $L_f$ , inductance de la partie excitatrice a une influence sur le temps que met le flux du stator à se mettre en place. Nous allons donc le fixer assez petit pour nous approcher du flux constant réel fourni par nos aimants permanents.  $R_f$  et  $I_f$  influencent la vitesse de rotation et nous allons donc fixer  $I_f$  puis chercher  $R_f$  sorte à ce que le moteur ait le comportement souhaité. Pour cela, on va installer la même MCC en génératrice en sortie de notre moteur pour pouvoir changer  $R_f$  sans que la vitesse de rotation ne soit affectée. Nous allons mesurer la tension aux bornes de la génératrice et connaissant la vitesse de rotation et la Fem par 1000tr/min de notre moteur (13.3V), nous allons pouvoir retrouver la valeur de  $R_f$  souhaitée.

On trouve alors :

$$R_f = 78,2 \text{ et } L_f = 0,002H$$

On trouve bien une tension de 48V. En effet, nous avons deux machines identiques sans couple résistant. On a donc un couple moteur nul et une  $fem = 48V$ , car il n'y a pas de chute de tension. En effet les machines sont identiques, donc il n'y a pas de couple, ce qui implique qu'il n'y a pas de courant.

#### 4.1.4 Comparaison des résultats

Nous comparons les résultats avec un programme Matlab. Pour cela, nous avons besoin de récupérer les valeurs de notre modélisation sous forme de tableau :

$$X = [temps, courant, vitesse]$$

Nous comparerons donc les points du courant et de la vitesse avec notre courbe de courant sur Matlab. Il faut faire attention à prendre assez de points. On remarque que la théorie est une réussite : les 2 courbes se superposent.

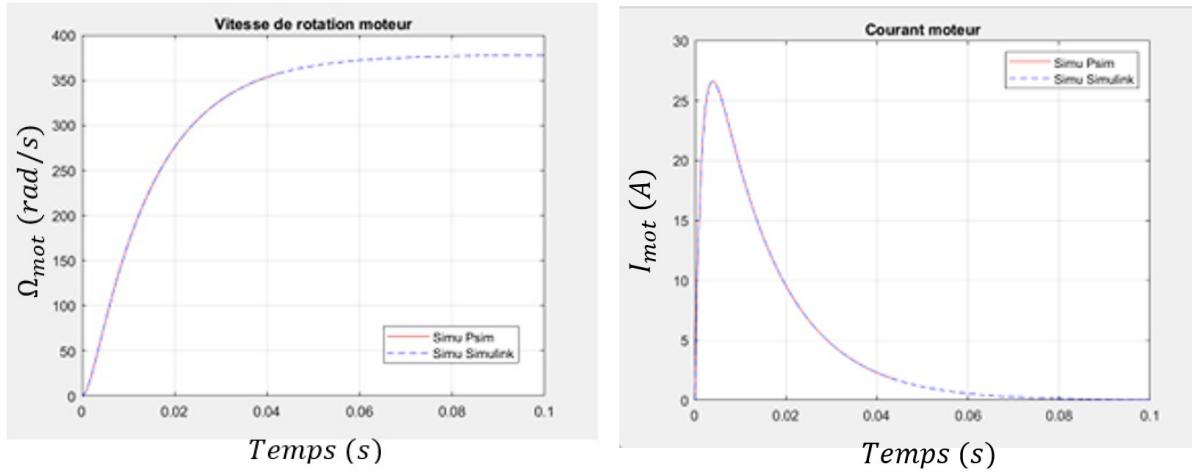


FIGURE 4.3 – Comparaison du modèle PSIM et de la simulation Matlab

10

Nous pouvons passer aux simulations suivantes : le moteur avec des pertes.

## 4.2 Simulation et Modélisation du moteur avec pertes

Dans cette partie, nous allons effectuer la même logique qu'avec le moteur à vide. Cette fois-ci le moteur a des pertes, appelés frottements, et le couple-résistant du moteur est pris en compte.

On note :

$$f = C_f \cdot \Omega = \frac{30}{\pi} \cdot \frac{1}{1000} \cdot 0,53 \cdot 10^{-2} = 5,06 \cdot 10^{-5} N.m/rad.s$$

$$C_r = 2 \cdot C_0 = 0,024 N.m$$

### 4.2.1 Équations de la MCC avec pertes

Les équations de notre système vont donc changer. En effet en ajoutant les frottements, le système devient :

Formules d'une MCC avec pertes

$$u(t) = R \cdot i(t) + L \frac{di(t)}{dt} + e(t)$$

$$C_m(t) = K_c \cdot i(t)$$

$$e(t) = K_e \cdot \omega(t)$$

$$J \frac{d\omega}{dt} = C_m(t) - f \cdot \omega(t) - C_r(t)$$

Ce qui donne dans le domaine de Laplace :

$$I(s) = \frac{U(s) - K_\Phi \Omega}{R + Ls}$$

$$C_m(s) = K_\Phi \cdot I(s)$$

$$E(s) = K_\Phi \cdot \Omega(s)$$

$$\Omega(s) = \frac{C_m + C_r}{Js + f}$$

Une fois les équations trouvées, nous allons pouvoir déterminer le schéma bloc et effectuer les mesures.

#### 4.2.2 Simulation Simulink

Le schéma bloc n'est plus le même. En effet, celui-ci devient plus complexe, puisque qu'on y ajoute le couple résistant  $C_r$  et les frottements f.

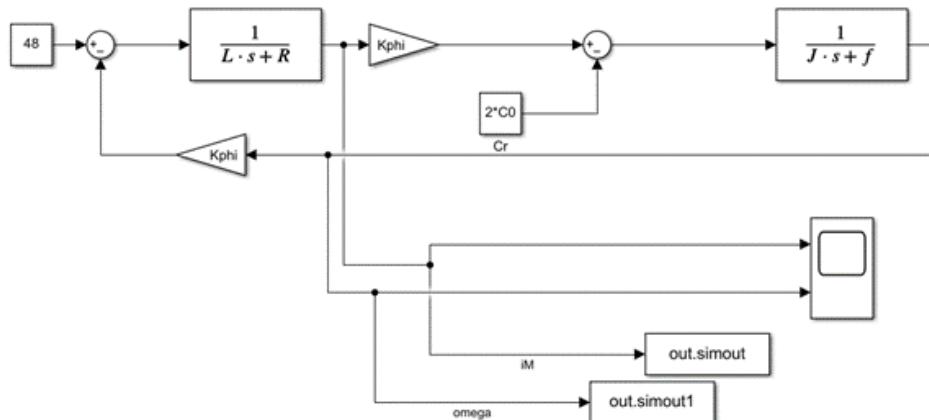


FIGURE 4.4 – Schéma bloc du moteur avec des pertes

10

Maintenant que nous avons le schéma bloc avec frottements, nous pouvons simuler les réponses en vitesse et en courant sur PSIM et Simulink, toujours dans l'optique de valider notre modèle.

### 4.2.3 Modélisation PSIM

La modélisation PSIM avec frottements est la suivante :

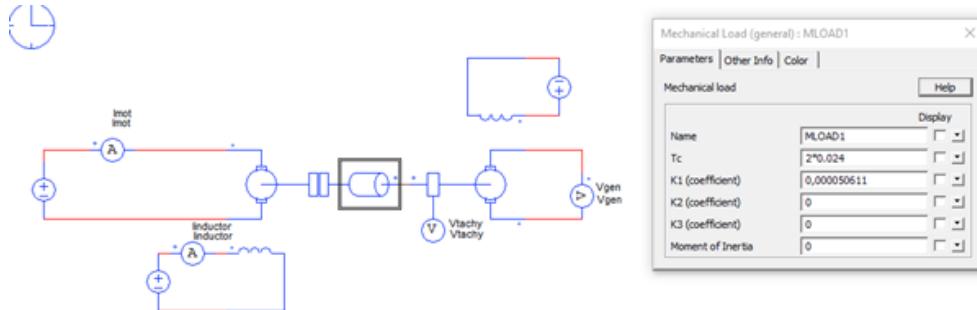


FIGURE 4.5 – Modélisation de la MCC avec des frottements

10

### 4.2.4 Comparaison des résultats

Comme avec la vitesse, les courbes se superposent parfaitement. On peut aussi voir qu'avec l'ajout des frottements, le courant d'induit ne s'annule plus, comme si la machine n'était plus à vide. De plus, cette valeur augmente avec l'augmentation des couples de frottements. Aux vues des simulations effectuées et des comparaisons entre PSIM et Simulink, nous pouvons valider notre modèle avec l'ajout des frottements. Durant ces deux tests, on a laissé la MCC à vide. Il nous reste donc un dernier paramètre à étudier, l'ajout d'une charge sur la génératrice.

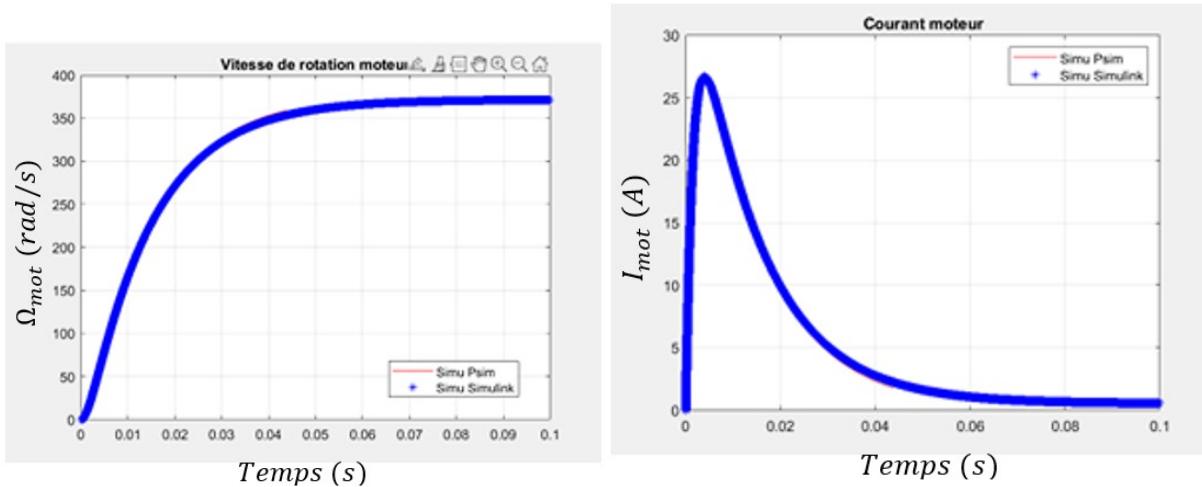


FIGURE 4.6 – Comparaison du modèle PSIM et de la simulation Matlab

10

## 4.3 Modélisation du hacheur

Le hacheur, ou convertisseur continu - continu, est un dispositif d'électronique de puissance qui met en œuvre un ou plusieurs interrupteurs électroniques commandés pour permettre de modifier la valeur de la tension (moyenne) d'une source de tension continue avec un rendement élevé.

On modélise le hacheur sur PSIM par un pont en H composé de 4 transistors MOSFET, de la façon suivante :

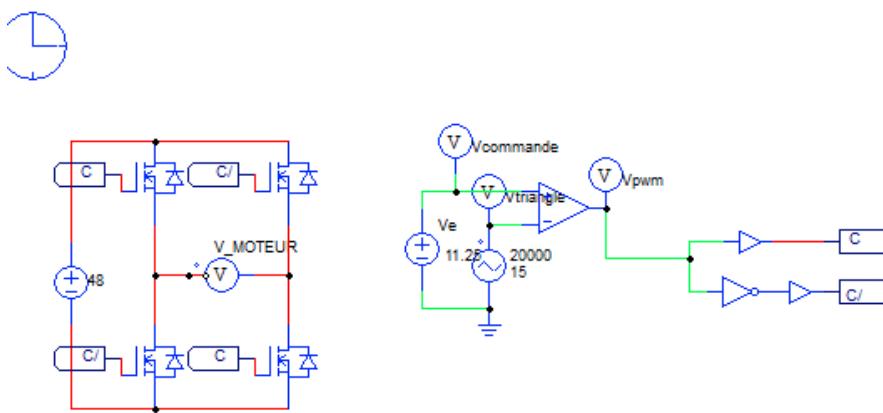


FIGURE 4.7 – Modélisation du hacheur sur PSIM

10

Ce hacheur fournit une tension réglable qui peut être positive comme négative, il fonctionne donc en mode bipolaire. Sa tension de sortie est fonction de sa tension de commande  $V_{pwm}$ , de sa tension d'alimentation fixe (ici 48V) et du rapport cyclique (duty cycle)  $\alpha$  du signal de commande de nos transistors.

Ainsi, la commutation des transistors 1 et 4 est identique, opposée à 2 et 3 permet de piloter le hacheur. Le rapport cyclique  $\alpha$  correspond au rapport du temps de fermeture de l'interrupteur sur la période du hacheur. Par exemple, pour un  $\alpha = 0$ , on aura une tension de sortie de  $V_S = V_{moteur} = -48V$  et pour  $\alpha = 1$ , on a  $V_S = V_{moteur} = 48V$ . Entre ces deux valeurs de  $\alpha$ , la tension de sortie variera de façon linéaire. Si le rapport cyclique  $\alpha$  augmente, alors  $V_S$  augmentera. Si  $\alpha$  diminue, alors  $V_S$  diminuera.

Le hacheur est également commandé par une tension continue proportionnelle à un rapport cyclique :  $V_{pwm}$ , en vert sur les courbes, qui contrôle le mode de conduction des transistors. Sur la modélisation, nous obtenons la commande du hacheur  $V_{pwm}$  en comparant un signal triangle ( $V_{triangle}$ ) allant de 0 à 15V à la tension continue  $V_e = V_{commande}$  à l'aide d'un AOP.

Ce qui nous donne les courbes suivantes : Enfin, nous avons placé en sortie du comparateur des suiveurs, qui permettent d'effectuer une adaptation d'impédance. Comme les courants d'en-

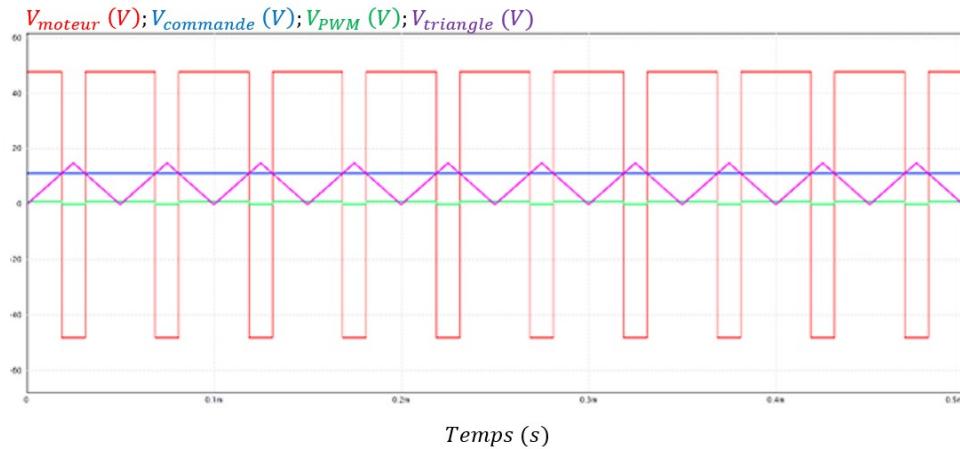


FIGURE 4.8 – Courbes représentant les différentes tensions étudiées lors de la simulation du hacheur

10

trées d'un AOP sont quasiment nuls, cela nous permet de ne pas tirer de courant sur les étages précédents et ainsi d'isoler les étages les uns par rapport aux autres.

Nous choisissons une fréquence de fonctionnement de 20 kHz. Il s'agit de la fréquence de fonctionnement standard pour ce type de hacheur, mais elle permet également de rendre inaudible le bruit émis par le hacheur.

## 4.4 Simulation et Modélisation du Moteur avec pertes et génératrice

### 4.4.1 Equations de la MCC

Formules d'une MCC avec pertes et génératrice

$$\begin{aligned}
 u(t) &= R \cdot i(t) + L \frac{di(t)}{dt} + R_{ch} \cdot i(t) = K_\Phi \cdot \omega(t) \\
 C_m(t) &= K_c \cdot i(t) \\
 e(t) &= K_e \cdot \omega(t) \\
 J \frac{d\omega}{dt} &= C_m(t) - C_r(t) - f \cdot \omega(t)
 \end{aligned}$$

Ce qui donne en Laplace :

$$I(s) = \frac{U(s) - K_\Phi \Omega(s)}{(R + R_{ch} + Ls)}$$

$$K_\Phi \cdot I(s) = \frac{K_\Phi^2 \cdot \Omega(s)}{(R + R_{ch} + Ls)}$$

$$C_m(s) = K_\Phi \cdot I(s)$$

$$E(s) = K_\Phi \cdot \Omega(s)$$

$$\Omega(s) = \frac{C_m + C_r}{Js + f}$$

$$\frac{C_m(t)}{\Omega(s)} = \frac{K_\Phi^2}{(R + R_{ch} + Ls)}$$

Grâce à ces équations, nous pouvons trouver le schéma bloc du système et commencer nos simulations. Avec ces équations, nous pouvons trouver les caractéristiques du moteur et de sa génératrice.

Nous avons les grandeurs suivantes :

- **Tension aux bornes de la génératrice :**  $U = e_m \cdot i(t) \implies U = 46,76V$
- **Courant qui traverse la génératrice :**  $I_g = \frac{K_\Phi - 2f}{K_\Phi} = 3.87A$
- **Résistance de charge :**  $R_{ch} = \frac{3U}{I_g} = 8,78\Omega$

Nous prendrons donc  $R_{ch} = 10\Omega$  pour avoir une valeur normalisée proche de la valeur théorique.

#### 4.4.2 Simulation Simulink

Là encore, nous avons représenté toutes ces équations dans Matlab/Simulink.

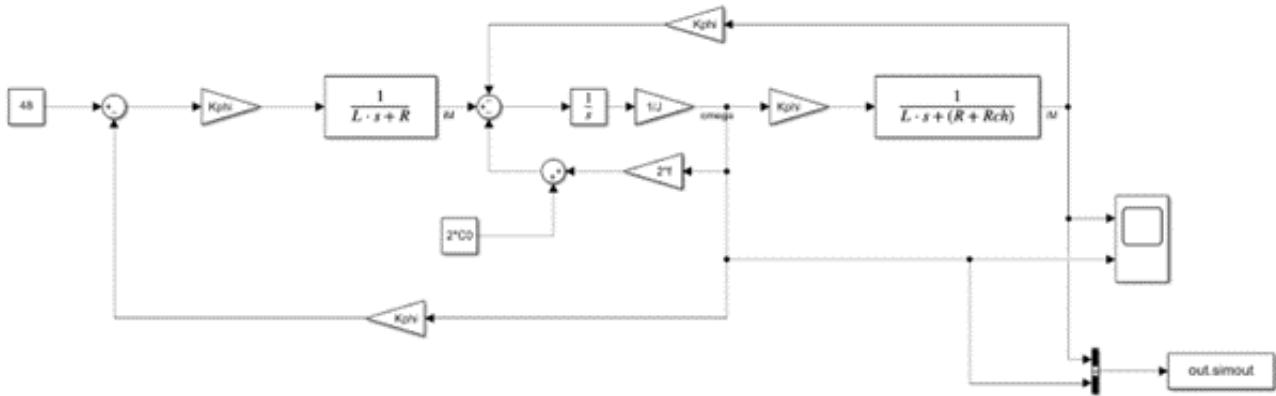


FIGURE 4.9 – Schéma bloc sous Simulink du moteur avec charge et frottements

Ce système doit également être reporté dans PSIM afin d'avoir un modèle vraiment convaincant.

#### 4.4.3 Modélisation PSIM

On modélise la MCC avec frottements et hacheur de la façon suivante :

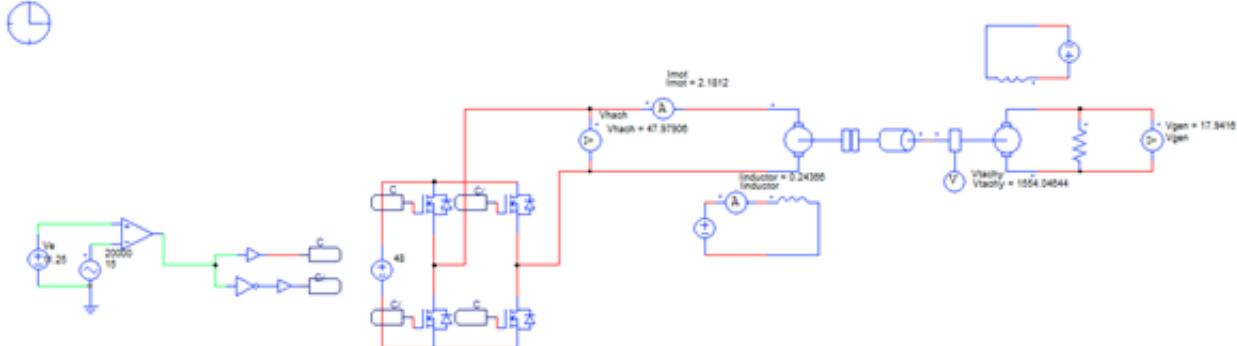


FIGURE 4.10 – Modélisation du moteur avec frottements et sa charge 10

On remarque que le schéma devient de plus en plus complexe. C'est le but de nos simulations : on complexifie le schéma petit à petit jusqu'à arriver à un schéma complet et prêt à être mis en place dans la vraie vie.

#### 4.4.4 Comparaison des résultats

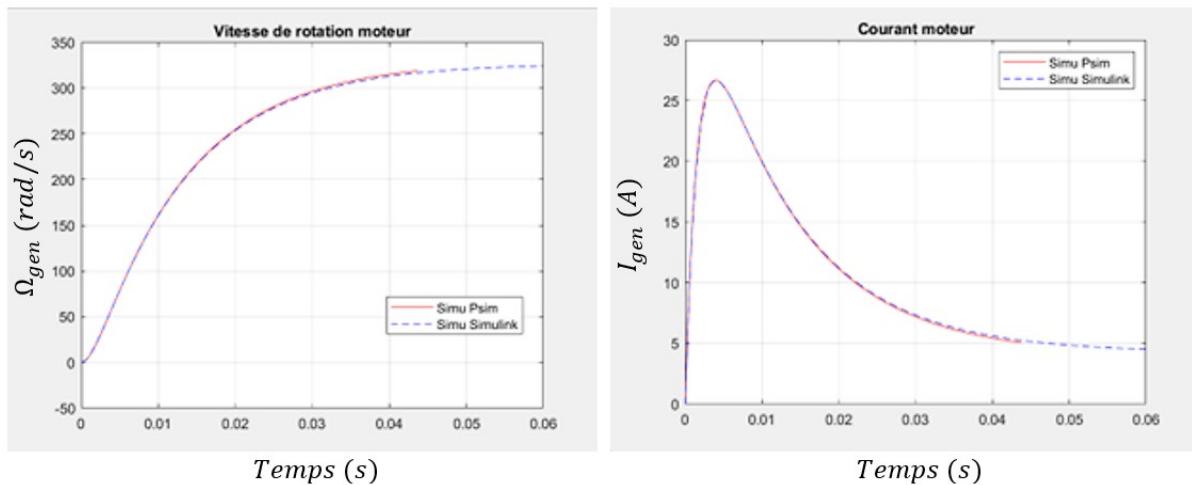


FIGURE 4.11 – Comparaison du modèle PSIM et de la simulation Matlab

Les courbes se superposent toujours. De plus, on remarque que le pic de courant n'a pas changé malgré l'ajout de la charge, donc cela valide bien le fait que le couple moteur reste constant et que c'est la vitesse qui va diminuer avec l'ajout de la charge. On peut donc valider notre modèle final de simulation de la MCC.

Nous pouvons maintenant étudier les autres éléments du montage dont nous devons comprendre le fonctionnement et être capable de modéliser sur PSIM et Simulink pour pouvoir continuer à simuler et observer les résultats.

Le modèle et la simulation peuvent désormais être asservis.

## 5. L'asservissement du moteur

### 5.1 Rappel du cahier des charges pour l'asservissement

On a le cahier des charges suivant :

- Dépassemement en vitesse et en courant : compris entre 10% et 20%.
- Temps de réponse pour le courant : 10 périodes du hacheur qui a une fréquence de fonctionnement de 20 kHz, soit  $0,5ms$ .
- Temps de réponse en courant en boucle fermée (BF) : 3 fois plus rapide qu'en boucle ouverte (BO).

On a calculé avec Matlab, un temps de réponse en BO de  $\tau_{BO} = 52ms$  on veut donc un temps de réponse en BF d'environ  $\tau_{BO} = 17ms$ .

### 5.2 Asservissement du courant traversant le moteur

#### 5.2.1 Simulation sous Matlab

Simplification du système

Notre système après les simulations du moteur et du hacheur est constitué de plusieurs blocs.

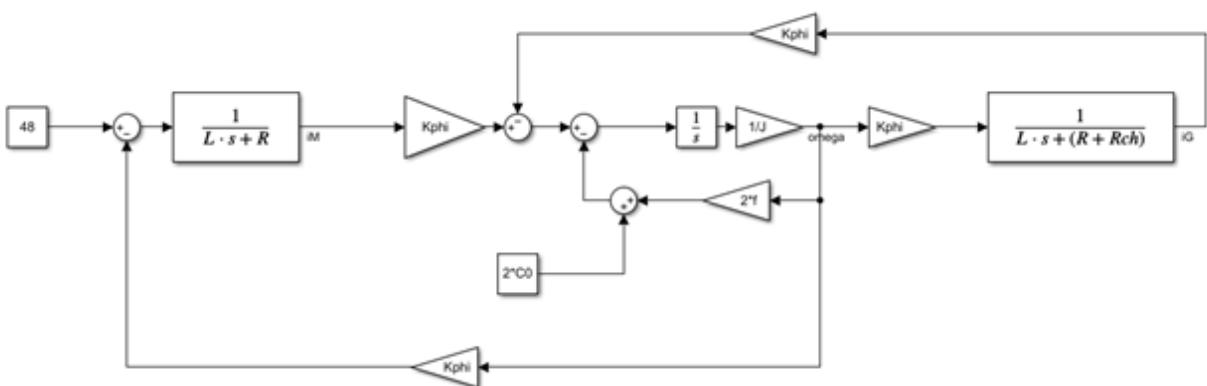


FIGURE 5.1 – Schéma bloc sous Simulink9 du système à asservir en courant

Pour pouvoir simplifier notre système et qu'il puisse être utilisable dans PID tuner, nous avons regroupé chaque bloc afin d'avoir une seule et même fonction de transfert et ainsi grâce à Matlab, nous avons pu les calculer.

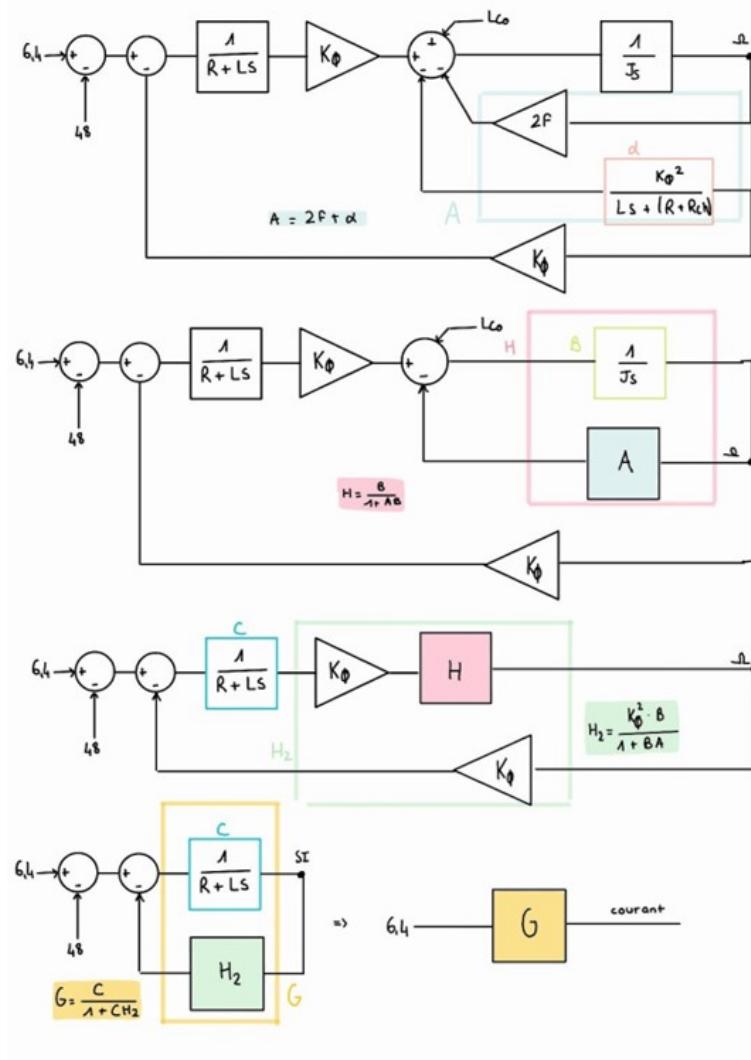


FIGURE 5.2 – Description de la simplification du système

Nous avons ainsi réussi à obtenir un système simplifié grâce à la méthode 5.2 et au programme ?? qui nous donne  $G$ , la fonction de transfert de courant qui traverse le moteur en fonction du signal d'entrée (forcément) mais aussi en fonction de la charge de la génératrice.

```

1 % CALCUL DU SYSTEME SIMPLIFIE
2
3 alpha= tf(Kphi^2,[L (R+Rch)]);
4 A= 2*f + alpha;
5 B= tf(1,[J 0]);
6 C= tf(1,[L R]);
7 H= B/(1+B*A);
8 H2= (Kphi^2)*H;
9 G= C/(1+C*H2);
10

```

FIGURE 5.3 – Programme que l'on a utilisé dans Matlab

10

```

G =

```

$$\frac{3.334e-14 s^4 + 1.873e-10 s^3 + 1.166e-07 s^2 + 2.173e-06 s}{7.335e-17 s^5 + 4.627e-13 s^4 + 5.477e-10 s^3 + 2.184e-07 s^2 + 2.535e-05 s}$$

FIGURE 5.4 – Fonction de transfert obtenue dans Matlab

10

### Étude de la stabilité et du système équivalent

Nous relevons la marge de phase et la marge de gain :

$$\Delta\phi = |-180 - (-81,5)| = 98,5$$

$$\Delta G = |-20,3 - (-3,69)| = 16,61dB$$

D'un point de vue pratique, dans l'industrie, il faut que la marge de phase soit supérieure ou égale à 45 et que la marge de gain, soit comprise entre 10 dB et 20 dB. Notre marge de phase étant très grande, le système sera certainement assez nerveux. Mais la stabilité ne sera pas affectée par l'asservissement.

De plus, pour des systèmes de cette ampleur, une marge de phase de 45 donnerait un système beaucoup trop nerveux! Idéalement, on pourrait choisir une marge de 70, mais comme les résultats de l'asservissement se sont avérés rentrer dans le cahier des charges sans baisser la marge de phase, nous avons décidé de garder celle-ci.

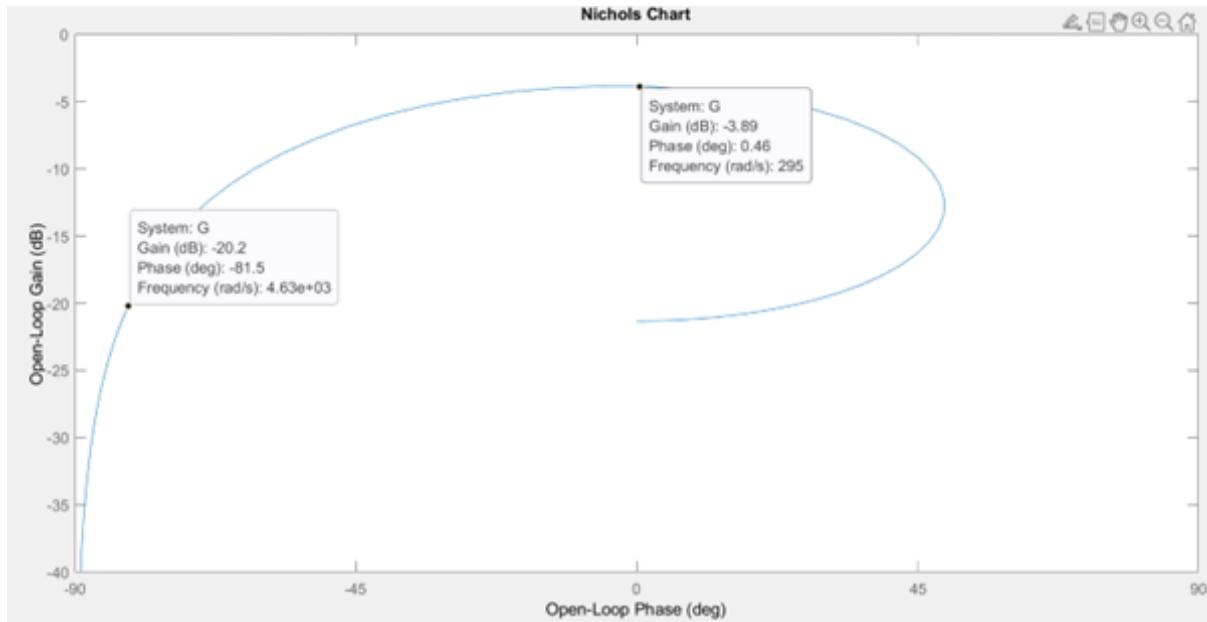


FIGURE 5.5 – Diagramme de Black (Nichols) de la fonction de transfert G

10

Grâce à ce programme, nous obtenons les coefficients du numérateur et du dénominateur de la fonction du transfert équivalente. Avec ceux-ci, nous pourrons facilement simuler le système simplifié.

### Simulation du système simplifié

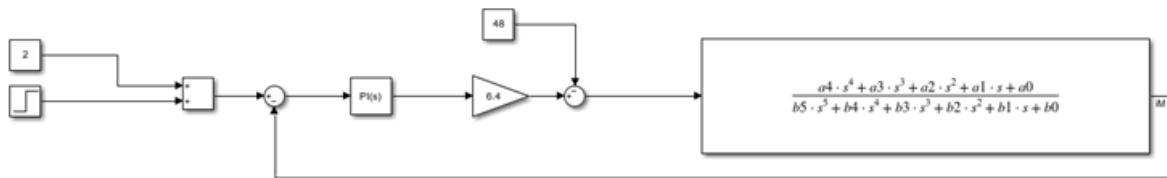


FIGURE 5.6 – Schéma de la boucle de courant sur Matlab/Simulink

10

Avec ce rebouclage, on peut commencer à positionner un correcteur PI (Proportionnel Intégral) et configurer ses paramètres  $K_p$  et  $K_i$  grâce à l'application intégrée à Matlab : PID tuner.

Ce qui nous donne :

Coefficient du correcteur PI pour l'asservissement de courant

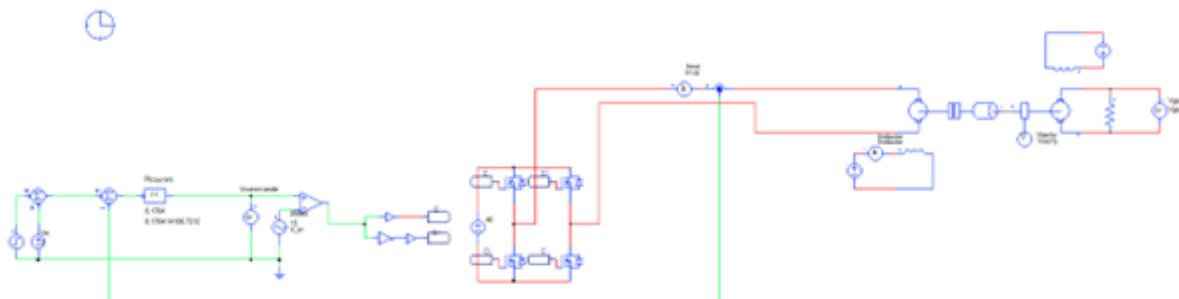
$$K_p = 8,1784$$

$$K_i = 14106,7212$$

Ces valeurs nous permettent d'avoir un système avec temps de réponse à 10 fois la période de MLI du moteur et un dépassement de moins de 20%.

### 5.2.2 Modélisation PSIM

Pour effectuer l'asservissement en courant sous Psim 9, nous reprenons le modèle de la MCC avec frottements et hacheur, auquel nous ajoutons une boucle de retour, des correcteurs, selon le schéma suivant :



10

FIGURE 5.7 – Modélisation de l'asservissement en courant de notre moteur

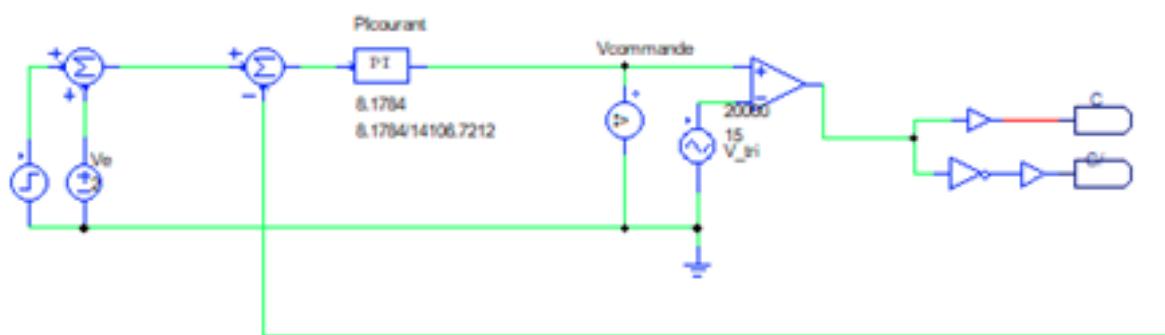


FIGURE 5.8 – Zoom sur les correcteurs de l'asservissement

10

Nous reprenons la valeur des correcteurs dimensionnés précédemment sur Matlab avec le PID Tuner, qui satisfont le cahier des charges et que nous adaptons à Psim.

En effet, sur Matlab, le correcteur est de la forme  $C(p) = K + Ip$  avec  $K_p = 8,1784$  et  $K_i = 14106,7212$ .

Sur Psim, il est de la forme  $C(p) = K_{psim} \cdot \frac{1+\tau p}{\tau p} = K_{psim} + \frac{K_{psim}}{\tau p}$ .

Ainsi, par identification,  $K_p = K_{psim} = 8,1784$  et  $K_i = 14106.7212$  avec  $\tau = \frac{K_p}{K_i}$

Nous entrons donc ces paramètres dans notre modèle. Nous sommes bien en boucle fermée, avec  $V_e = 2V$  et un échelon de 0,5 V qui se déclenche à 0,4 s.

Le courant de l'induit est réinjecté dans le comparateur, permettant un asservissement.

Nous obtenons donc les courbes suivantes pour le moteur :

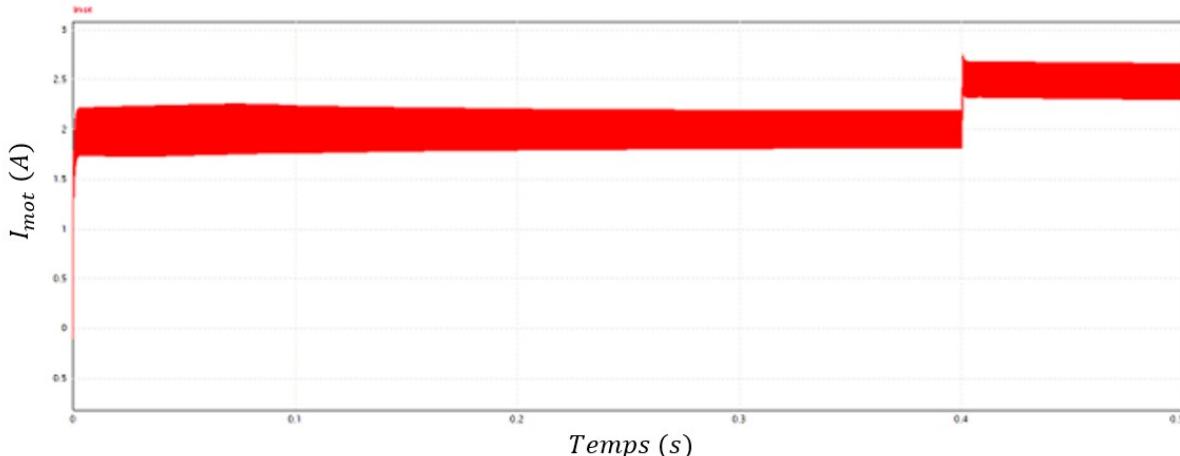


FIGURE 5.9 – Courbe représentative du courant  $i_{moteur}$  après asservissement

10

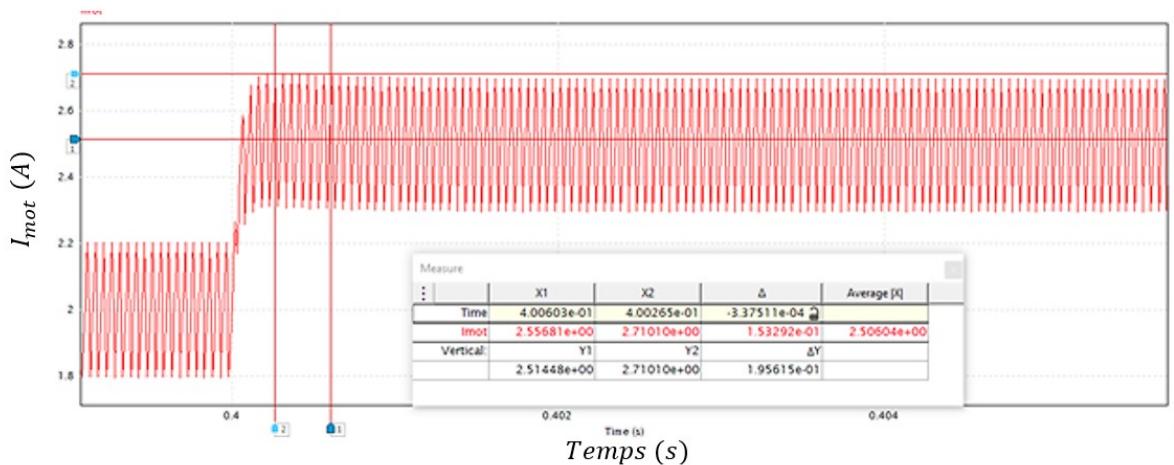


FIGURE 5.10 – Mesure du dépassement sur le courant  $i_{moteur}$  après échelon à 0,4 s

10

D'après le sujet, on veut un courant qui ne dépasse pas 4,95 A : on est ici à 2,7 A, avec un dépassement de 14%, ce qui est inférieur à 20% donc on respecte le cahier des charges.

De plus, nous avons un temps de réponse à 5% qui vaut  $T = \frac{1}{20000} \cdot 10 = 0,5ms$ .

Comme cette simulation se superpose à Matlab, nous avons remplacé le correcteur PI par des composants réels.

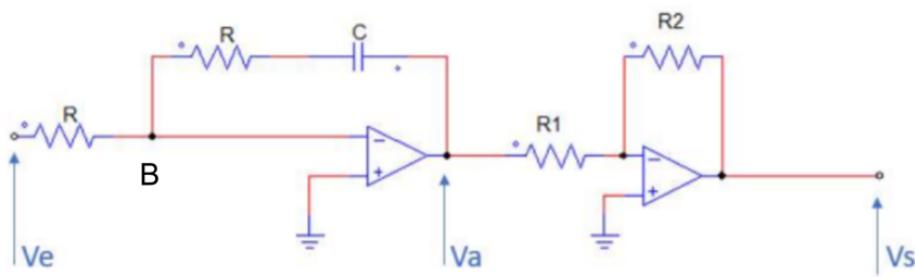


FIGURE 5.11 – Schéma d'un PI avec des AOP

10

### Dimensionnement des correcteurs réels

On calcule la fonction de transfert :

$$\frac{V_e(s)}{V_s(s)} = \frac{V_e(s)}{V_a(s)} \cdot \frac{V_a(s)}{V_s(s)} = H_1(s) \cdot H_2(s)$$

On considère que les AOP sont idéaux, donc  $\epsilon = 0 \implies V^+ = V^-$

On va calculer les correcteurs en séparant les deux fonctions de transfert :

– Pour  $H_1(s)$  :

$$\text{On note } R_{eq} = R + \frac{1}{sC}$$

$$\text{On utilise } V^- = \frac{V_a \cdot R + V_e \cdot Z_{eq}}{R + Z_{eq}}$$

$$\text{Donc } V_a \cdot R = -V_e \cdot (R + \frac{1}{sC}) = -V_e \cdot \frac{sRC + 1}{sC}$$

$$\text{Ainsi, } \frac{V_e(s)}{V_a(s)} = \frac{RCs + 1}{sC}$$

– Pour  $H_2(s)$  :

On applique le théorème de Millman au point B.

$$V^- = \frac{\frac{V_a}{R_1} + \frac{V_s}{R_2}}{\frac{1}{R_1} + \frac{1}{R_2}} \implies \frac{V_s \cdot R_1 + V_a \cdot R_2}{R_1 + R_2}$$

$$\text{Ainsi } H_2(s) = \frac{V_s}{V_a} = -\frac{R_2}{R_1}$$

Donc par identification :

$$K_{psim} = -\frac{R_2}{R_1}$$

$$\tau = RC$$

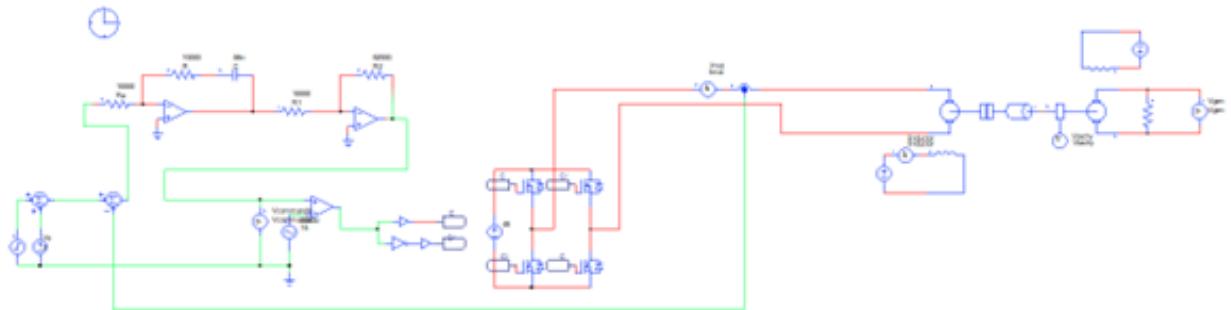
On a donc choisi :

$$R = R_a = R1 = 10k\Omega$$

$$R2 = 82k\Omega$$

$$C = 58nF$$

On place ces valeurs de composants dans l'asservissement et on obtient le schéma suivant :



10

FIGURE 5.12 – Schéma de l'asservissement avec des composants réels

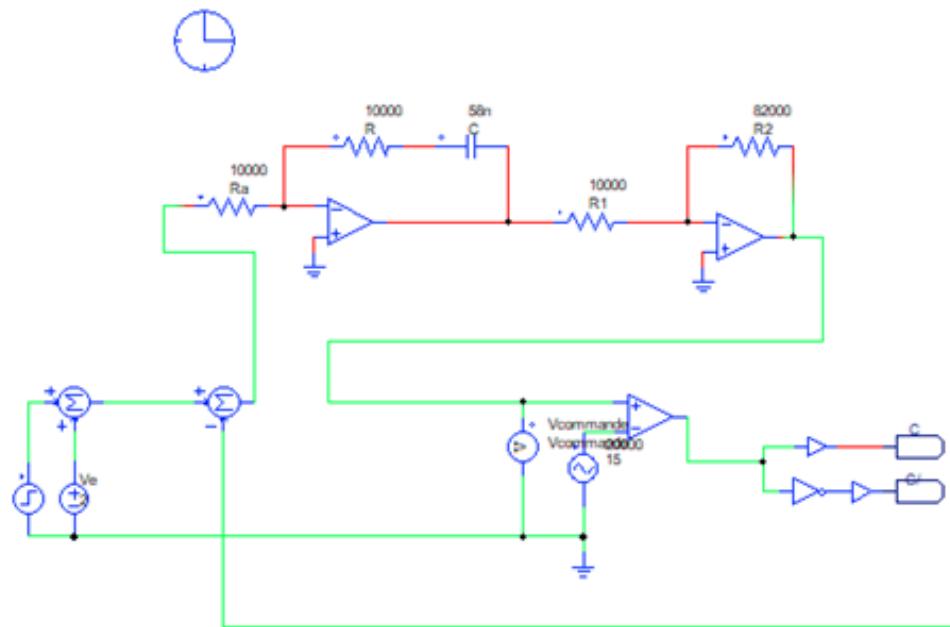


FIGURE 5.13 – Zoom sur les correcteurs réels

10

On peut alors mesurer sur la simulation le courant du moteur  $I_{moteur}$  :

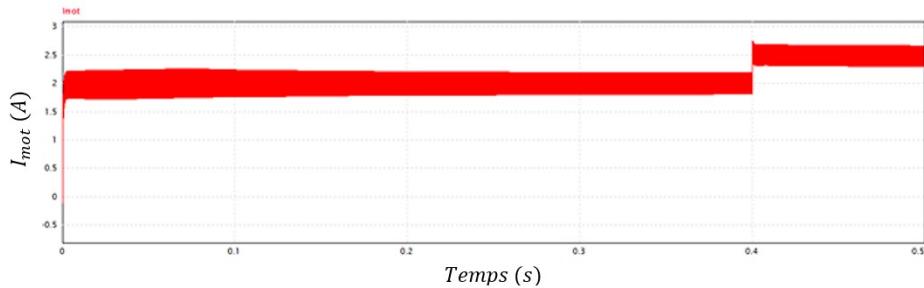


FIGURE 5.14 – Courbe représentative du courant  $I_{moteur}$  après asservissement avec des composants réels

10

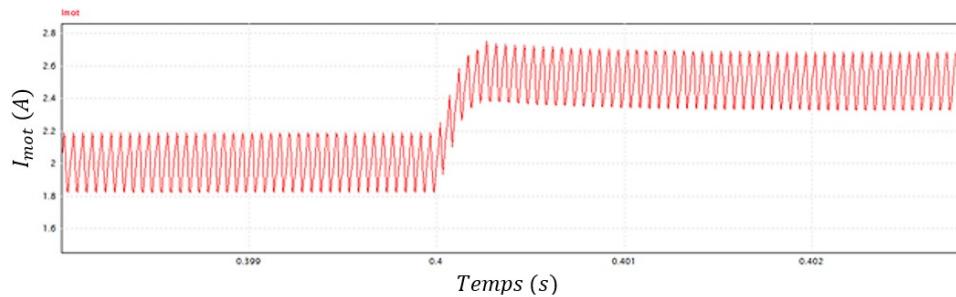


FIGURE 5.15 – Mesure du dépassement sur le courant  $i_{moteur}$  après échelon à 0,4 s avec des composants réels

10

Ce qui nous donne la même courbe que précédemment, qui répond donc au cahier des charges. Il faut alors maintenant comparer notre modèle à notre simulation.

### 5.2.3 Comparaison entre PSIM et Matlab

Maintenant que les simulations sous PSIM et Matlab sont faites, on peut les comparer et savoir si notre système est juste.

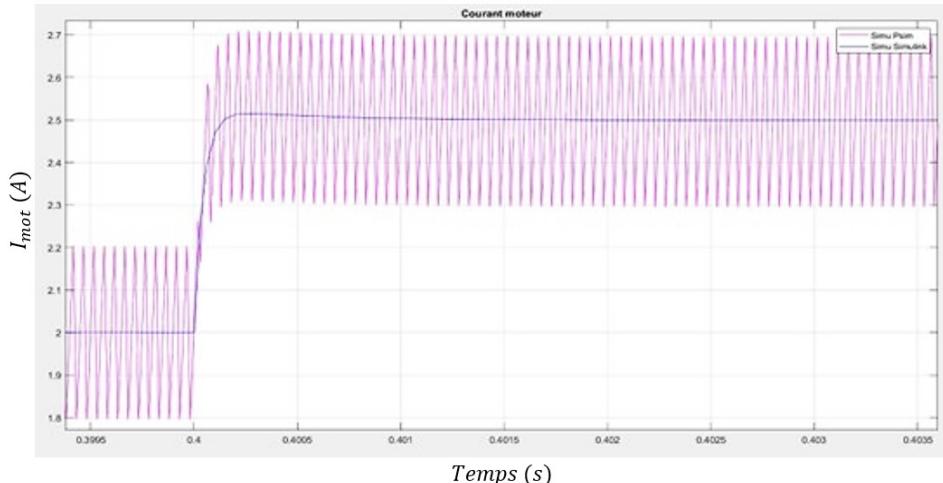


FIGURE 5.16 – Courbes du courant dans le moteur sous PSIM (en magenta) et MATLAB (en bleu)

10

Nous remarquons que les deux simulations sont bien en accord, cela signifie donc que nos systèmes sont corrects. Nous relevons également un dépassement de 19% ce qui respecte le cahier des charges.

## 5.3 Asservissement en vitesse

### 5.3.1 Simulation Matlab

Après avoir créé la boucle de courant qui nous permet de protéger notre moteur de trop grands pics d'intensité, nous devons piloter sa vitesse de rotation à l'aide du hacheur qui va faire varier sa tension à ses bornes ainsi que de la dynamo tachymétrique 9 qui va nous permettre de nous rendre une tension image de la vitesse de rotation du moteur.

De cette façon, le schéma bloc de l'asservissement du moteur sous Matlab ressemble à cela :

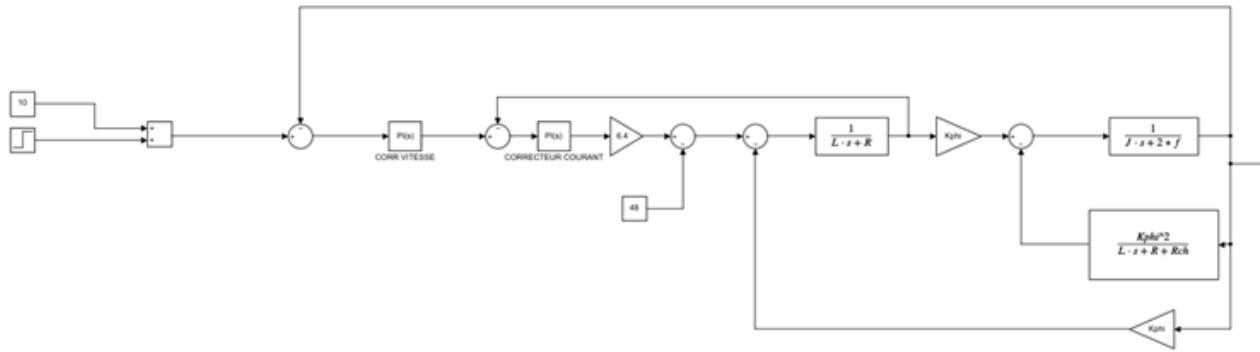


FIGURE 5.17 – Schéma bloc de notre simulation matlab pour l'asservissement de vitesse

10

Nous savons que l'on a une vitesse de réponse  $T_0 = 265$  ms en BO grâce à la fonction damp sur matlab, de plus nous voulons que le système soit 3x plus rapide qu'en boucle fermée et un dépassement entre 10% et 20%.

Nous rentrons donc nos attentes dans PID tuner et obtenons grâce à cela les paramètres  $K_p$  et  $K_i$  du correcteur proportionnel et intégral (PI). C'est ainsi que nous obtenons :

$$K_p = 2,1 \text{ et } K_i = 52,1$$

Ces valeurs nous semblent grandes, mais il ne faut pas oublier le gain de la dynamo tachymétrique. En effet, le gain tachymétrique nous impose un facteur 0.006 dans notre asservissement. S'il est utile dans Matlab, il ne faudra pas oublier de le prendre en compte dans nos correcteurs.

Nous rentrons donc nos valeurs de  $K_p$  et  $K_i$  afin d'asservir la vitesse du moteur pour ensuite passer à la simulation sous PSIM.

### 5.3.2 Modélisation PSIM

Nous réalisons la simulation 9 sous PSIM de l'asservissement en vitesse. Pour cela, nous avons repris le modèle précédent avec l'asservissement de courant. L'image de la vitesse obtenue par le capteur au niveau de la dynamo tachymétrique est réinjectée dans un comparateur en entrée du système.

Les correcteurs utilisés sont ceux dimensionnés sur Matlab et adaptés à PSIM comme nous l'avions fait pour le courant. Ainsi, nous avons :

$$K_{PSIM} = K_p = 2,1$$

$$\tau = \frac{K_p}{K_i} = 0,040$$

Il avait été démontré en cours que l'asservissement de courant est négligeable dans ce cas, nous l'avons donc laissé par souci de précision, mais ne nous en préoccupons pas.

Nous avons également ajouté un correcteur  $K=\pi/30$ , qui nous permet d'observer la vitesse dans les mêmes unités que Matlab afin de pouvoir par la suite superposer les deux courbes et les comparer. Plus tard, il sera intégré directement dans notre moteur. Nous l'avons laissé juste pour le début des simulations.

Nous avons pris  $V_e = 10V$ ,  $V_{step} = 0,5V$  et  $t_{step} = 0,4s$ .

Nous obtenons donc le modèle suivant :

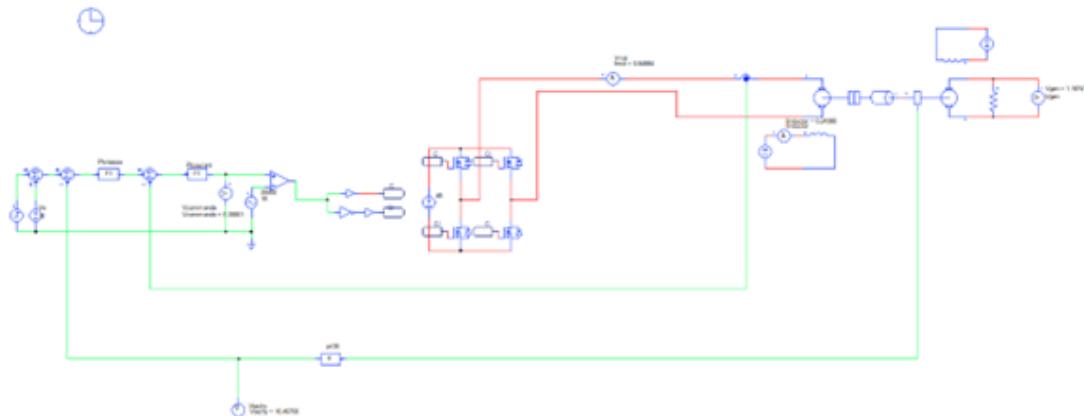


FIGURE 5.18 – Schéma bloc de notre modélisation PSIM pour l'asservissement de vitesse

10

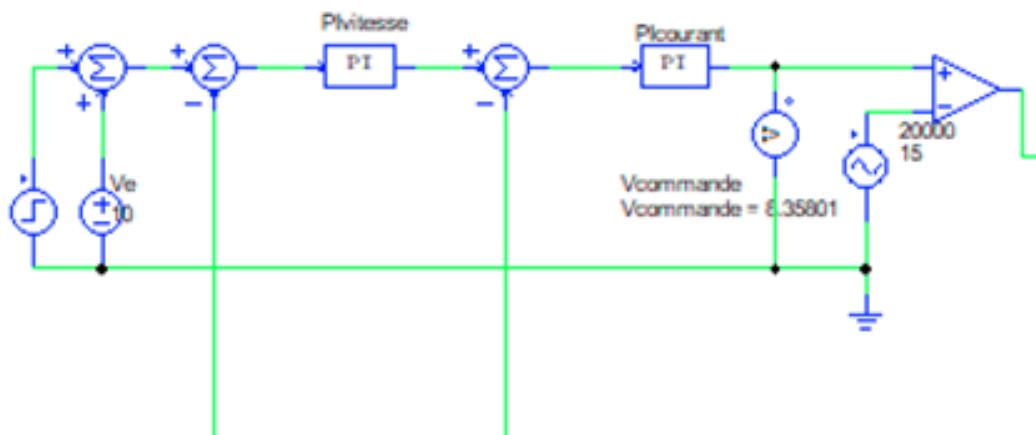


FIGURE 5.19 – Zoom sur les correcteurs de l'asservissement en vitesse

10

Nous avons ajouté un correcteur  $K=\pi/30$  dans la boucle de retour. Il nous permet d'adapter les unités afin qu'elles soient identiques à Matlab et que nous puissions directement comparer les deux courbes en les superposant.

Nous obtenons la courbe suivante nous permettant d'observer la vitesse de ce modèle :

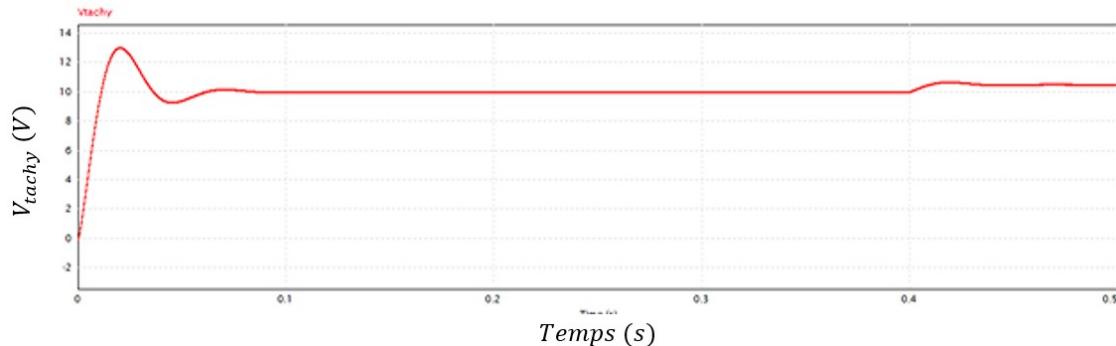


FIGURE 5.20 – Vtachy en fonction du temps - issu de la simulation PSIM de la figure précédente

10

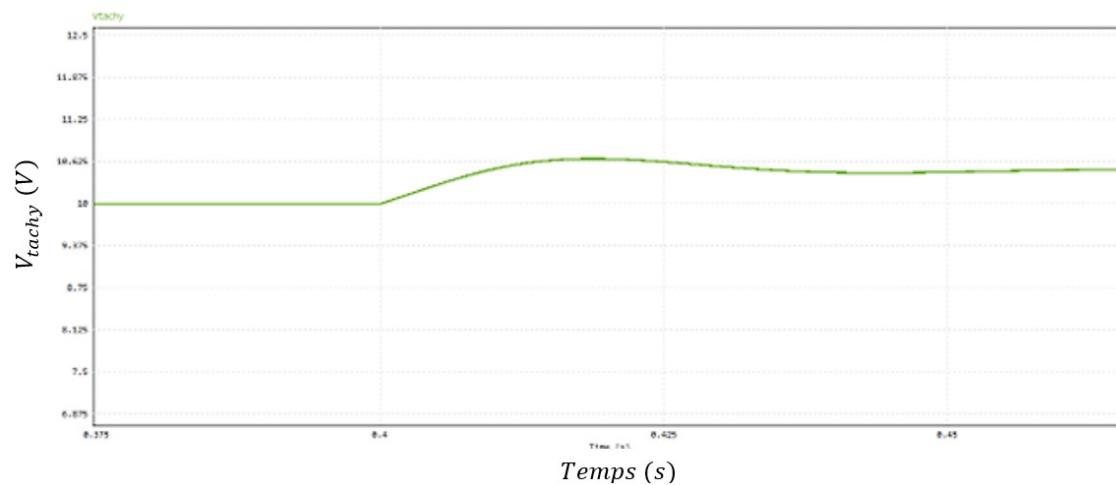


FIGURE 5.21 – Zoom sur Vtachy en fonction du temps au niveau de l'échelon à 0,4s - issu de la simulation PSIM de la figure précédente

10

Le premier dépassement observé est lié au démarrage, notre échelon est seulement à 0,4s, nous nous concentrerons donc sur ce dépassement.

Nous avons un dépassement <20%, ce qui satisfait le cahier des charges.

### 5.3.3 Asservissement avec des composants réels

Comme la partie précédente nous a démontré que nos correcteurs satisfont le cahier des charges, on cherche désormais à les remplacer par des composants réels.

Pour cela, nous reproduisons le correcteur PI grâce à un montage avec des AOP 5.11 comme pour l'asservissement de courant.

#### Dimensionnement des correcteurs réels

On calcule la fonction de transfert :

$$\frac{V_e(s)}{V_s(s)} = \frac{V_e(s)}{V_a(s)} \cdot \frac{V_a(s)}{V_s(s)} = H_1(s) \cdot H_2(s)$$

On considère que les AOP sont idéaux, donc  $\epsilon = 0 \implies V^+ = V^-$

On va calculer les correcteurs en séparant les deux fonctions de transfert :

– Pour  $H_1(s)$  :

$$\text{On note } R_{eq} = R + \frac{1}{sC}$$

$$\text{On utilise } V^- = \frac{V_a \cdot R + V_e \cdot Z_{eq}}{R + Z_{eq}}$$

$$\text{Donc } V_a \cdot R = -V_e \cdot (R + \frac{1}{sC}) = -V_e \cdot \frac{sRC + 1}{sC}$$

$$\text{Ainsi, } \frac{V_e(s)}{V_a(s)} = \frac{RCs + 1}{sC}$$

– Pour  $H_2(s)$  :

On applique le théorème de Millman au point B.

$$V^- = \frac{\frac{V_a}{R1} + \frac{V_s}{R2}}{\frac{1}{R1} + \frac{1}{R2}} \implies \frac{V_s \cdot R1 + V_a \cdot R2}{R1 + R2}$$

$$\text{Ainsi } H_2(s) = \frac{V_s}{V_a} = -\frac{R2}{R1}$$

Donc par identification :

$$K_{psim} = -\frac{R2}{R1}$$

$$\tau = RC$$

On a donc choisi :

$$R = R_a = R1 = 10k\Omega$$

$$R2 = 1,2k\Omega$$

$$C = 39nF$$

$$R = 200k\Omega$$

Pour les résistances, on utilisera des potentiomètres, afin d'être plus variables et de pouvoir

corriger l'asservissement si besoin.

Nous vérifions ainsi si notre courbe ressemble à la courbe avec les correcteurs théoriques 5.20 :

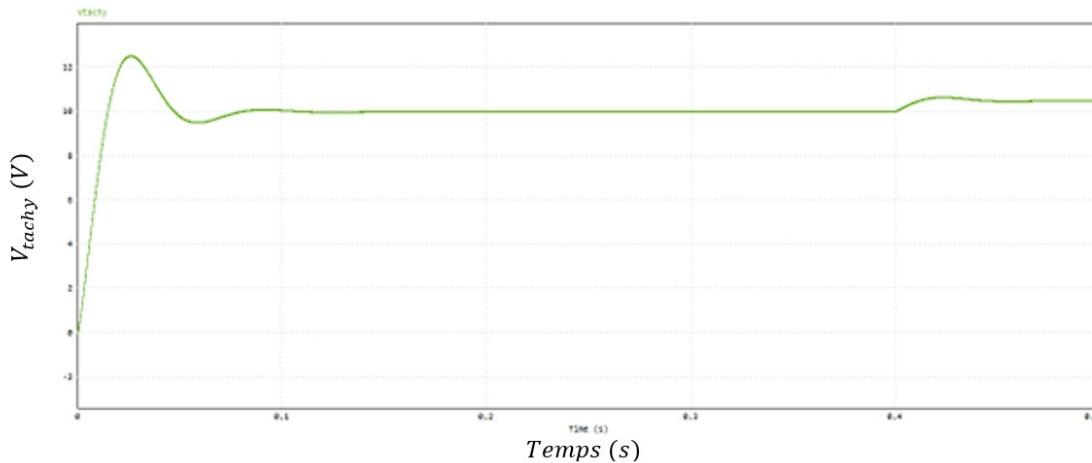


FIGURE 5.22 – Vtachy en fonction du temps - issu de la simulation PSIM de la figure précédente 10

Nous obtenons la même courbe que précédemment. Même si les résultats sont similaires, nous obtenons un écrêtage à 12V. Cela ne devrait pas avoir lieu. Nous devons donc trouver une piste d'amélioration. Un limiteur de tension pourrait être une bonne solution. Un écrêteur de courant pourrait également parer à ce phénomène.

### 5.3.4 Comparaison PSIM et Matlab

Maintenant que les deux simulations sont faites, nous pouvons lancer le programme de comparaison et vérifier si lesdites simulations sont correctes.

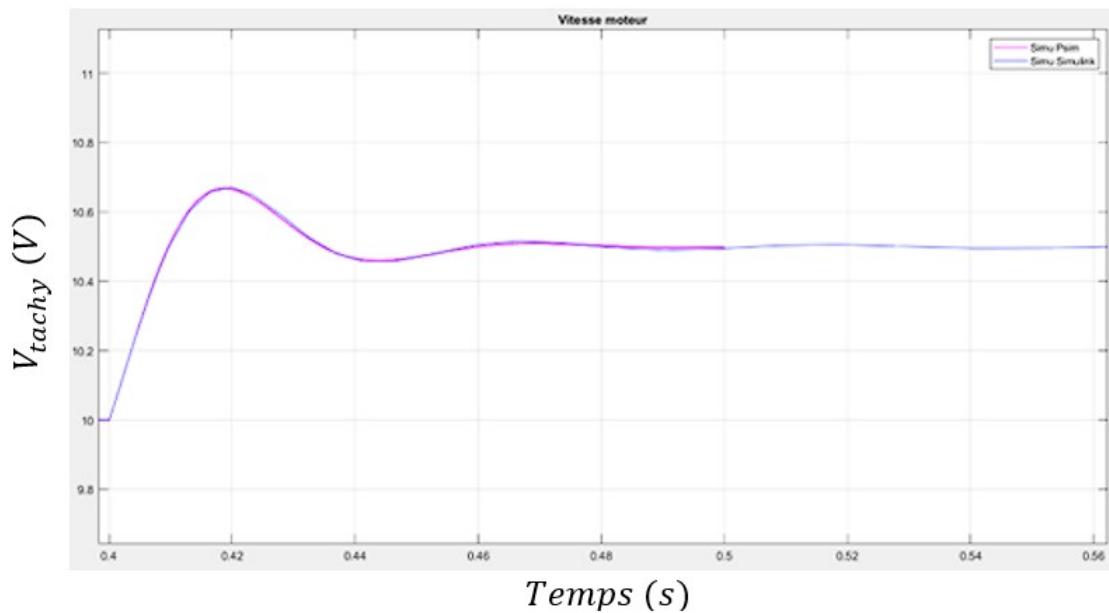


FIGURE 5.23 – Simulation de l’asservissement de vitesse entre PSIM (en magenta) et MATLAB (en bleu)

10

On remarque qu’effectivement nos courbes correspondent bien dans les systèmes sont corrects. Nous avons asservi en vitesse notre système. Un problème saute aux yeux : le dépassement est supérieur à 20%. Nous avons donc changé nos valeurs de correcteur et recommencé jusqu’à obtenir le même résultat avec d’autres correcteurs.

En fin de compte, nous étions très proches des bons correcteurs. Il nous fallait ajuster un tout petit peu notre correcteur P.

$$K_{PSIM} = K_p = 2,8$$

$$\tau = \frac{K_p}{K_i} = 0,036$$

Ce qui nous a fait corriger les valeurs des composants choisis :

$$R = R_a = R1 = 8,2k\Omega$$

$$R2 = 22k\Omega$$

$$C = 1,8F$$

$$R = 20k\Omega$$

Pour ce système, le dépassement était à 13% et le temps de réponse à 88 ms.

À noter cependant, nous devrons faire les simulations PSIM en remplaçant les comparateurs

par des systèmes avec AOP et nous devrons potentiellement ajouter un limiteur, car l'asservissement en vitesse est susceptible de générer des pics de courants non souhaités.

## 5.4 Écrêteur : comment limiter le courant dans la MCC

Nous avons donc ajouté un circuit écrêteur de tension à notre schéma. En effet, en limitant la tension d'entrée, le courant sera ainsi limité. C'est le principe de la loi d'Ohm.

### 5.4.1 Fonctionnement du circuit d'écrêtage

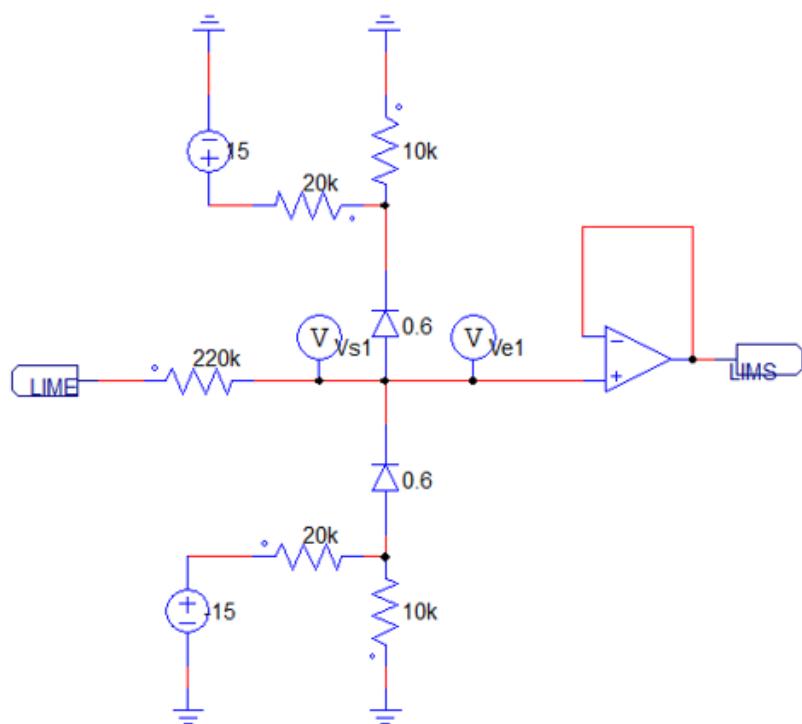


FIGURE 5.24 – Schéma du schéma écrêteur utilisé

10

La partie supérieure de ce montage sert à effectuer un écrêtage négatif.

Ce montage est un redresseur avec une simple alternance. Pendant l'alternance positive du signal d'entrée, la diode D est conductrice.

Le courant qui traverse la résistance R ( $20k + 10k$ ) produit dans celle-ci une chute de tension, ayant la forme de l'alternance positive du signal d'entrée.

Pendant l'alternance négative, la diode D polarisée dans le sens inverse de sa conduction, présente une résistance interne très élevée. Cela implique donc un blocage du courant.

Le courant négatif qui traverse la résistance R est donc très faible et par conséquent la chute de tension à ses bornes, est négligeable.

Le même principe est fait sur la partie inférieure du schéma. Il permet d'écrêter le signal positif. Il suffira juste d'inverser le sens de la diode.

### 5.4.2 Courbes caractéristiques du schéma écrêteur

Pour vérifier le fonctionnement de notre limiteur, nous avons imposé un circuit alternatif [-10V;10V] à l'entrée et relevé le signal de sortie :

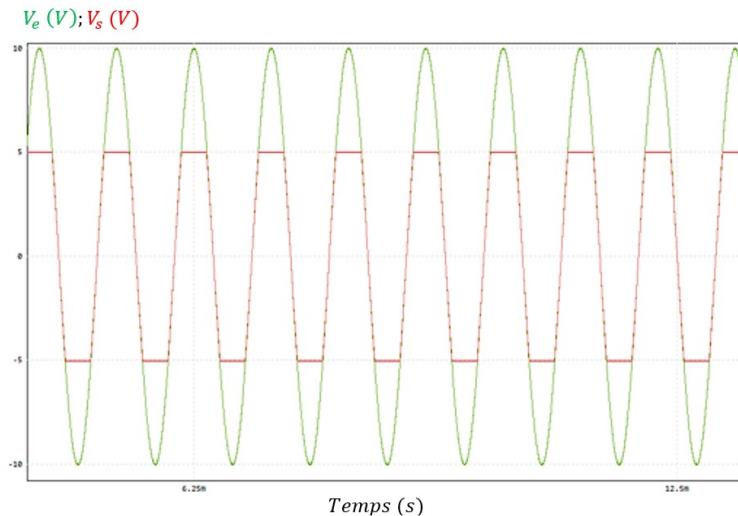


FIGURE 5.25 – Caractéristique de l'écrêtage avec une tension d'entrée et la tension de sortie associée

10

On voit bien que l'écrêtage a lieu à 5V et -5V. Cela sera donc équivalent à un écrêtage de 5A et -5A.

Grâce à ce montage, notre asservissement respectera le cahier des charges. Nous regardons la courbe que nous donne notre modélisation PSIM, sans imposer une tension d'entrée.

## 5.5 Type d'asservissement

### 5.5.1 Asservissement par retour dynamo tachymétrique

Afin de pouvoir assurer un asservissement, la dynamo tachymétrique va donner une tension qui est fonction de la vitesse de rotation du moteur à courant continu (MCC). Ainsi, nous n'avons qu'à récupérer ce signal, l'adapter pour qu'il soit sur les mêmes échelles que le signal de consigne et le réinjecter dans le système afin d'asservir le moteur.

## 5.5.2 Asservissement par codeur incrémental

Nous avons vu que nous pouvions asservir la MCC avec l'aide d'une dynamo tachymétrique. Cependant, il existe plusieurs moyens de convertir une vitesse en tension. L'autre procédé que nous allons utiliser ici, s'appelle le codeur incrémental. Le principe est le suivant. Ce système possède un disque, fixé à l'arbre du moteur, dans lequel se trouve des trous et surfaces pleines. Ainsi, nous faisons passer un laser à travers le disque qui sera réceptionné par une photodiode. Ainsi, si le signal de sortie est de 5V, la lumière du laser passe à travers le disque, ce qui signifie que le laser passe à travers un trou. À l'inverse, si le signal en sortie est de 0V, alors le laser ne traverse pas le disque, ce qui signifie qu'il est contre une partie opaque. Grâce à cela, nous obtenons, en sortie, un signal carré qui a une fréquence en fonction de la vitesse de rotation du moteur (une vitesse de rotation de 3000 tr/min équivaut à une fréquence en sortie de 90 kHz).

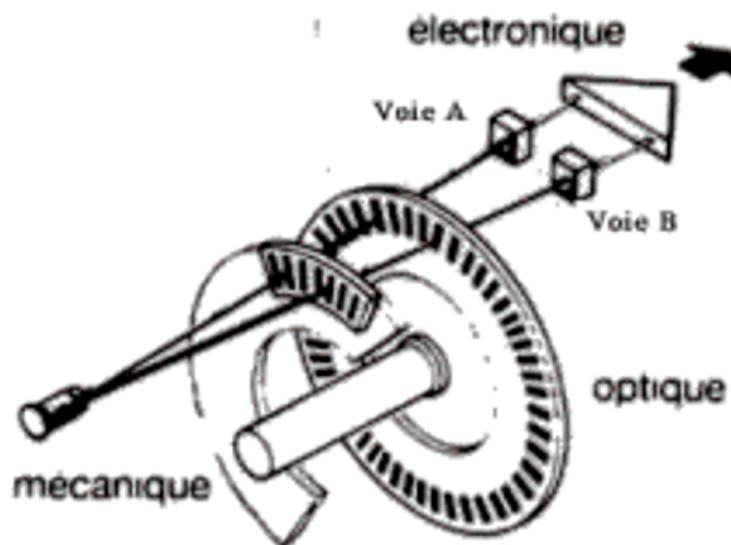


FIGURE 5.26 – Fonctionnement codeur incrémental

Toutefois, l'information de la vitesse de rotation du moteur se trouve dans la fréquence du signal. Or, pour pouvoir asservir le système, nous devons avoir un signal qui a un niveau de tension en fonction de la vitesse de rotation. Il faut donc adapter le signal reçu.

Pour cela, nous allons tout d'abord utiliser un filtre moyenneur qui permettra d'obtenir la moyenne du signal obtenu et ainsi avoir un niveau de tension image de la vitesse de rotation.

Pour finir, la dernière chose à faire est d'adapter le signal afin qu'il puisse être utilisable par notre correcteur PI.

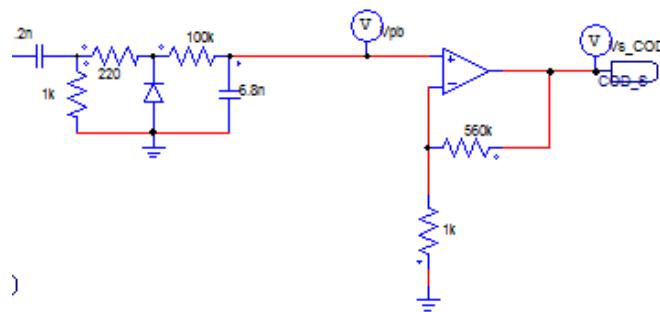


FIGURE 5.27 – Montage du codeur incrémental

10

## 5.6 Schéma final

Nous avons rassemblé les différentes parties explicitées ci-dessus dans une seule et même simulation Proteus.

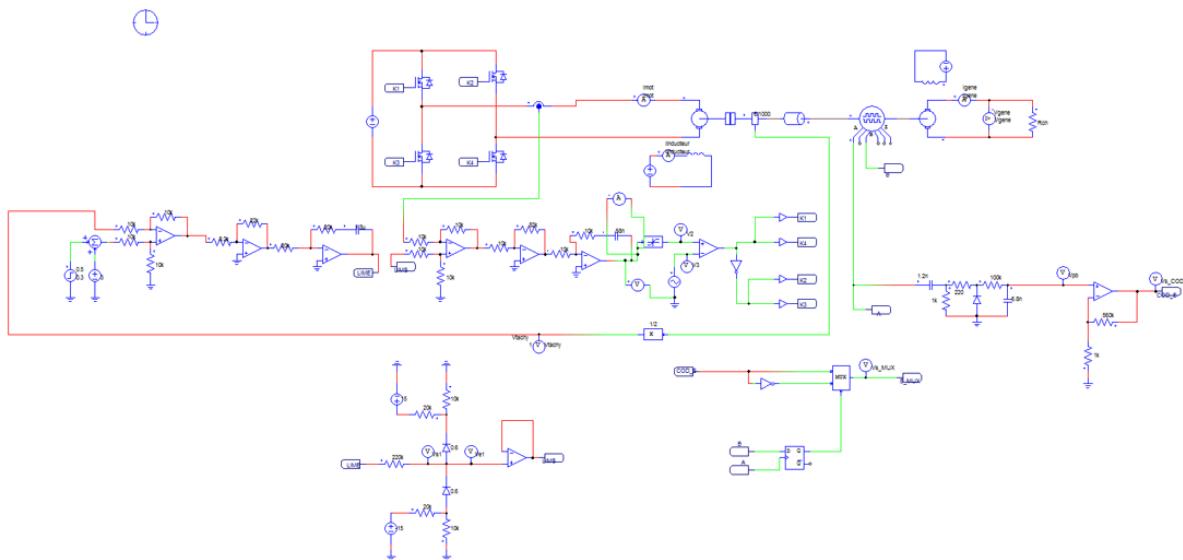


FIGURE 5.28 – Schéma final de l'asservissement de la MCC

10

Nous y retrouvons notre MCC avec frottements et génératrice et le hacheur. Nous y avons ajouté l'asservissement en vitesse, en courant, l'écrêteur et le codeur incrémental. L'utilisateur pourra choisir d'utiliser le codeur incrémental ou la dynamo tachymétrique pour l'asservissement

en vitesse grâce à un switch (il sera visible sur Proteus, nous déplaçons simplement les labels sur Psim).

$V_{tachy}$  (V);  $I_{gene}$  (A);  $I_{inducteur}$  (A);  $I_{mot}$  (A)

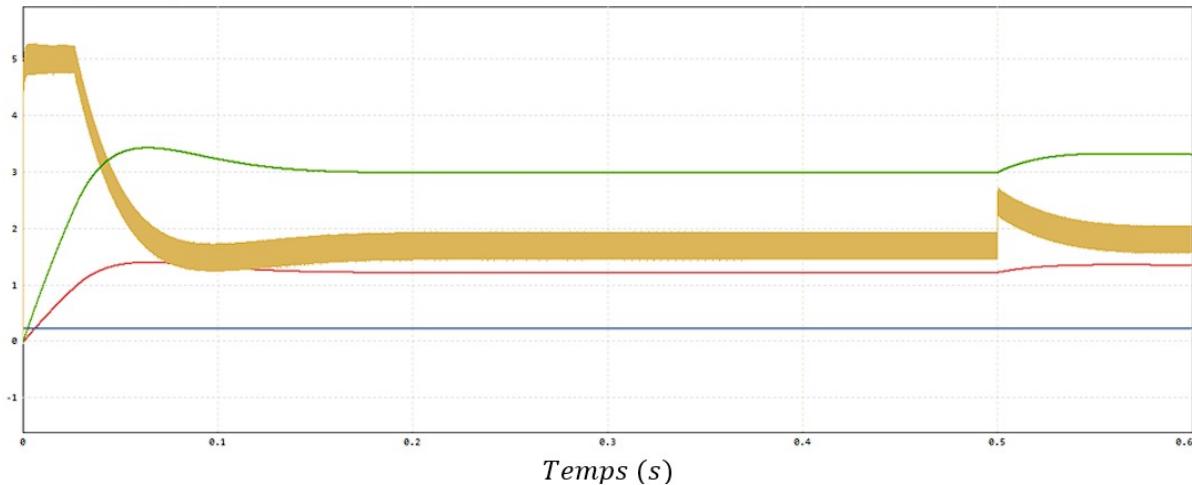


FIGURE 5.29 – Analyse des différents courants de notre MCC

10

En lançant la simulation, nous obtenons les résultats suivants :

- pour la vitesse ( $V_{tachy}$  en vert) : la vitesse maximale est de 3500 tours/min, ce qui est bien inférieur aux 4800 tours/min spécifiés. La vitesse nominale est de 3000 tours/min (puisque la tension image de la vitesse est de 3V), c'est ce que nous souhaitons. Le dépassement est de 13.3 %, ce qui est bien entre 10 et 20%
- pour le courant : le courant de saturation est inférieur à 5A, grâce à notre écrêteur. Le dépassement est de 19% ce qui est bien entre 10 et 20%

On remarque alors que le courant dans le moteur " $I_{mot}$ " est bien écrété à 5A, ce qui signifie que notre montage limiteur est bien fonctionnel.

Comme nos résultats sont satisfaisants, nous pouvons passer à la réalisation des PCB.

# 6. Conception des PCB

## 6.1 Choix et organisation de nos PCB

Une fois les simulations réalisées sur Matlab 9 et Psim 9, nous avons reproduit la simulation Psim sur Proteus, puisque ce logiciel permet de réaliser des PCB. Pour simplifier les soudures, nous avons séparé notre schéma de simulations en 3 parties : une pour l'asservissement avec la dynamo tachymétrique et les correcteurs, une dédiée à la prise VGA et une pour le codeur incrémental 9.

Pour éviter de surcharger nos schémas avec des fils, nous avons utilisé des labels ( par exemple : -15V, +15V l'alimentation, sortie, codeur... ).

Nous avons placé des T-block au lieu des connecteurs coaxiaux pour les rendre plus accessibles.

Une fois les schémas terminés, nous avons réalisé les PCB9 : nous avons d'abord dimensionné nos cartes, puis placé les composants. Nous avons commandé les différents composants en respectant le budget imposé par le cahier des charges et en prenant garde de prendre les composants réels (et non les modèles des premières simulations) et traversants pour faciliter les soudures.

Lorsque les composants n'étaient pas disponibles sur Proteus, nous avons réalisé les packages ou pris des composants équivalents sur Proteus, en veillant à ce que l'encombrement et l'espace-ment entre les pins soient identiques.

Nous avons également ajouté des capacités de découplage sur nos schémas, au niveau de chaque AOP et de l'alimentation.

Enfin, les AOP étant généralement fragiles, nous avons privilégié l'utilisation de supports à AOP.

Nous avons récupéré une boîte comprenant déjà des connecteurs, que nous souhaitons réutiliser pour notre projet.

Nous avons donc dimensionné nos PCB pour qu'ils entrent dans la boîte, dont les dimensions sont les suivantes :

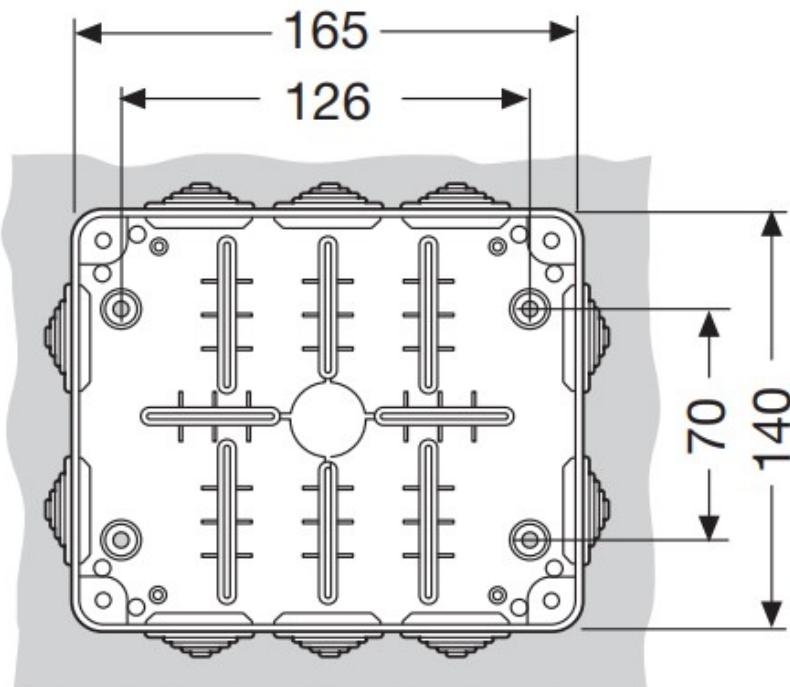


FIGURE 6.1 – Photo de la boîte

Nous avons prévu des trous de fixations sur nos cartes pour éviter qu'elles ne bougent dans la boîte. Nous avons également essayé de superposer les connecteurs et de les placer vers l'extérieur afin qu'ils soient facilement accessibles et au plus proche de ceux de la boîte.

Nous avons ensuite routé notre carte, en minimisant au maximum le nombre de vias et en choisissant des pistes adéquates pour les soudures. Pour les minimiser, nous avons utilisé un plan de masse.

En soudant nos cartes, nous avons régulièrement vérifié que nous n'avons pas fait de courts-circuits.



**1SL0824A00 - 1SL0830A00  
 1SL0874A00 - 1SL0854A00  
 1SL0860A00 - 1SL0880A00  
 1SL0924A00 - 1SL0930A00  
 1SL0974A00 - 1SL0954A00  
 1SL0960A00 - 1SL0980A00**

FIGURE 6.2 – Dimensions du boîtier

## 6.2 Les cartes électroniques obtenues

Nous obtenons donc les simulations, cartes et PCB suivants.

### 6.2.1 Pour la prise VGA :

Elle sera directement reliée au codeur incrémental situé à l'extérieur de la boîte. Nous l'avons relié à un T4-Block pour pouvoir la relier plus facilement au reste du projet, qui sera dans le fond de la boîte.

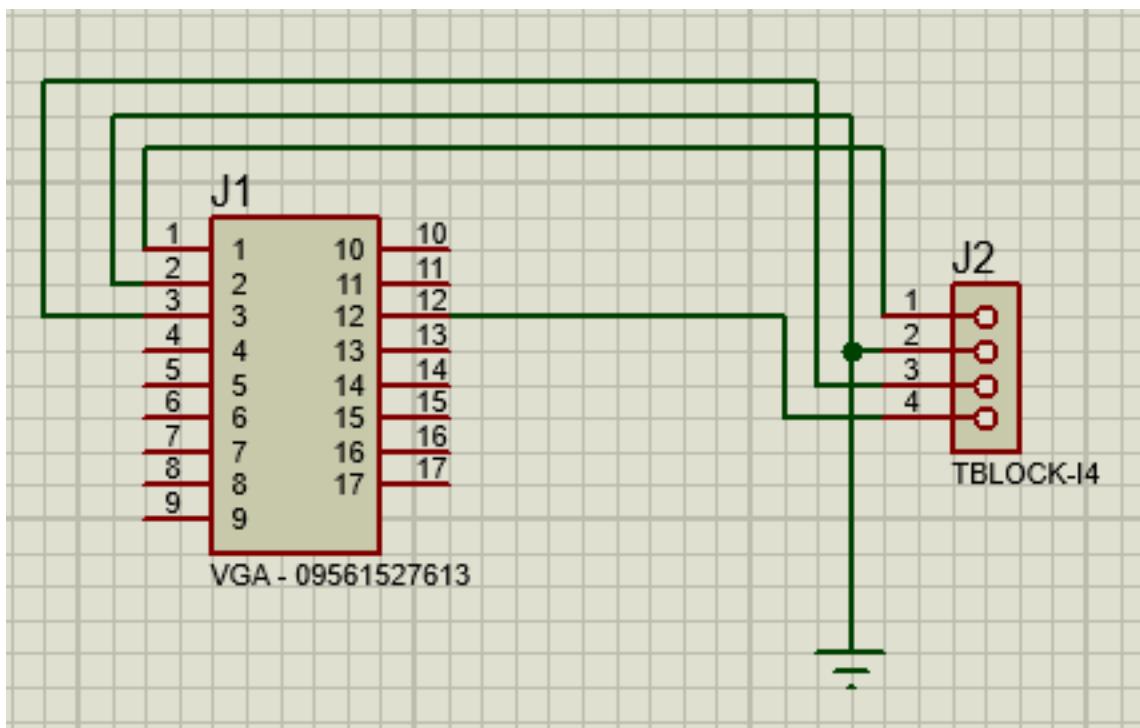


FIGURE 6.3 – Schéma de la carte avec VGA

10

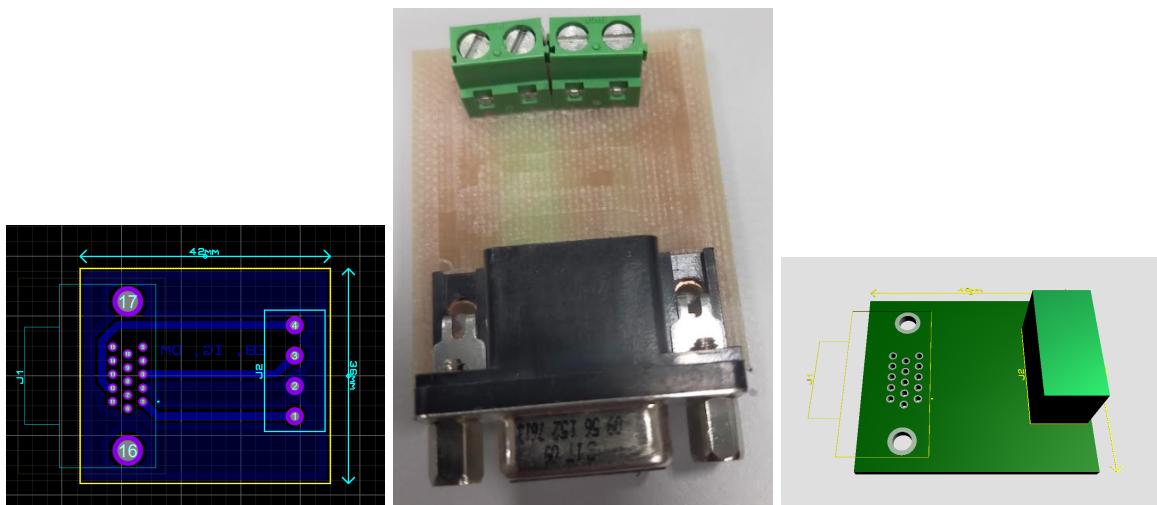


FIGURE 6.4 – PCB, vue 3D et soudures de la carte VGA

10

## 6.2.2 Pour le codeur incrémental

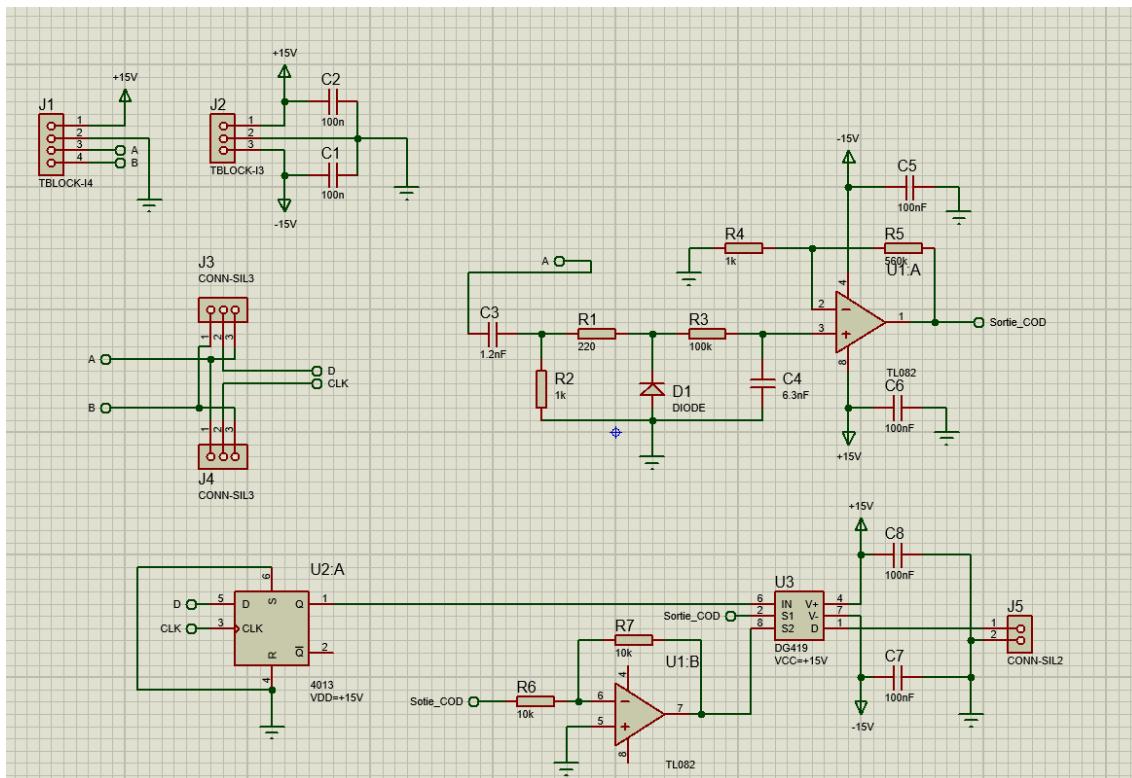


FIGURE 6.5 – Schéma de la carte du codeur incrémental

10

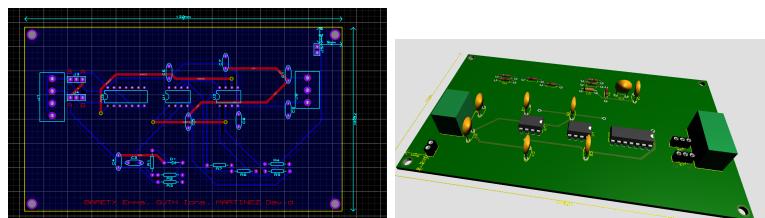


FIGURE 6.6 – PCB et vue 3D du codeur incrémental

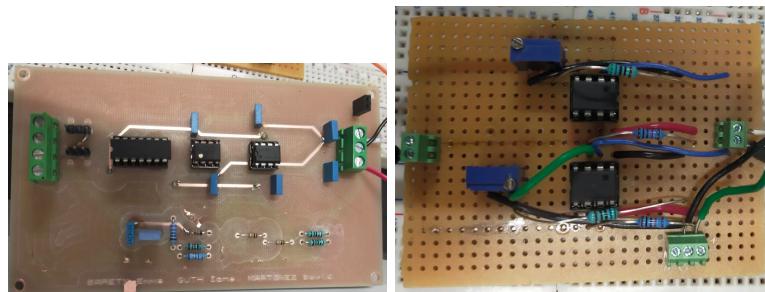


FIGURE 6.7 – PCB réalisés

Nous nous sommes rendus compte, que bien que sur Psim notre schéma était parfaitement fonctionnel, en réalité, nous avions besoin d'ajouter des comparateurs pour que le signal en sortie de notre PCB soit exploitable par le reste de notre système avec les asservissements. Nous avons donc conçu une autre carte électronique sur une plaque Epoxy avec ceux-ci (nous n'étions pas sûrs que le PCB soit prêt à temps). Dans l'idéal, nous devrions refaire un PCB avec.

### 6.2.3 Pour l'asservissement avec la dynamo-tachymétrique

Cette plaquette reprend les correcteurs en vitesse, en courant, l'écrêteur. Grâce à un switch, l'utilisateur pourra choisir s'il veut utiliser le codeur incrémental ou la dynamo-tachymétrique pour l'asservissement en vitesse. Nous avons remplacé le package de l'interrupteur par un t-block auquel il sera relié afin qu'il puisse être accessible sans ouvrir la boîte. Comme les valeurs réelles des composants diffèrent souvent de leur valeur théorique, nous avons choisi d'utiliser des potentiomètres, afin de pouvoir ajuster les correcteurs pour qu'ils correspondent à nos simulations.

La sortie de la plaquette sera à brancher sur le hacheur.

Nous avons aussi utilisé des soustracteurs réalisés avec des composants réels (AOP, résistances...).

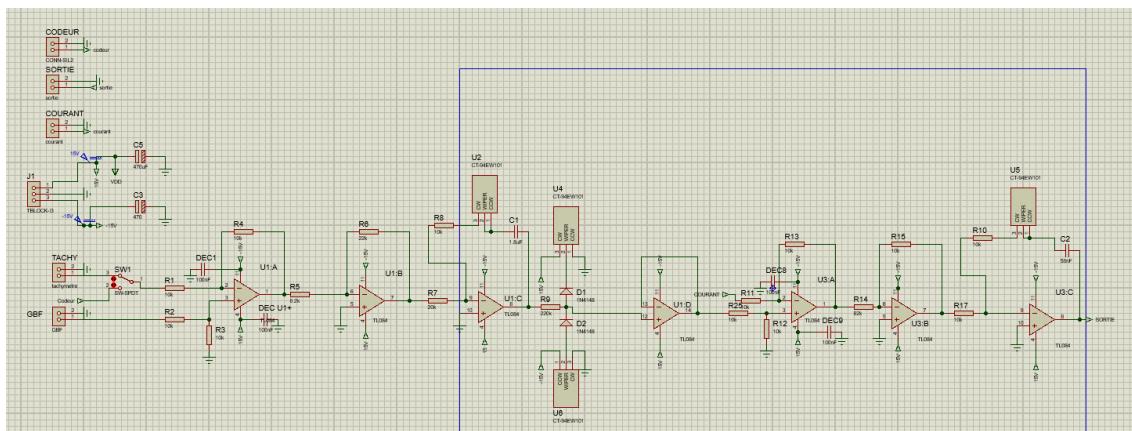


FIGURE 6.8 – Schéma de l'asservissement et de la dynamo tachymétrique

10

Il peut y avoir quelques différences entre nos PCB réels et nos schémas Proteus. Par exemple : nous avons mis une résistance 20k, comme sur Psim mais en réalité, cela n'existe pas, nous avons dû le remplacer par deux résistances 10k.

L'espacement entre le T4-block de la carte VGA ne correspondaient pas à ceux de Proteus, nous les avons remplacés par deux T2-blocks.

Nous pouvons désormais passer au débogage et aux tests.

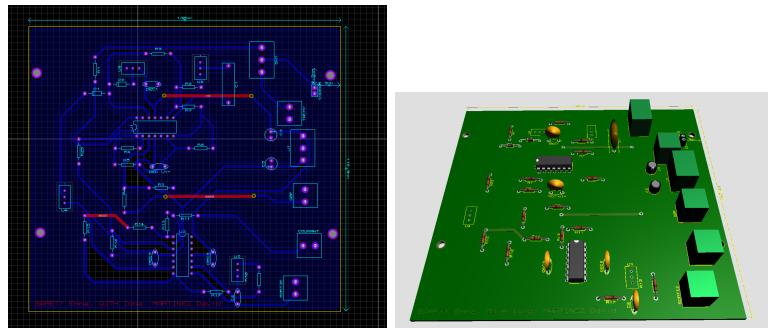


FIGURE 6.9 – PCB vue 3D de l'asservissement et de la dynamo tachymétrique

10

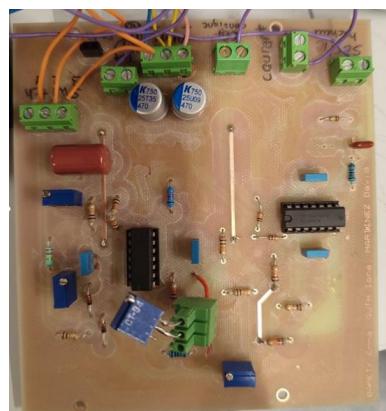


FIGURE 6.10 – PCB réalisé des asservissements avec la dynamo tachymétrique

# 7. Tests et essais sur la MCC

## 7.1 Tests sur plaquette

### 7.1.1 Quelques problèmes

Durant nos tests réalisés sur la plaquette, nous avons eu quelques déboires. Cela pouvait venir d'une soudure pas bien réalisée, mais nos plus gros problèmes venaient tous d'abord d'une erreur sur le PCB qui fait que deux résistances étaient inversées, ce qui faisait que le correcteur Proportionnel-Integral (PI) du limiteur de courant atténuait le signal au lieu de l'amplifier ce qui rendait un faux signal en sortie.

De plus, nous avons dû remplacer une résistance de notre correcteur PI pour l'asservissement de vitesse par un potentiomètre afin de pouvoir le modifier à la fin puisque nous avons remarqué que notre correcteur ne réagissait pas comme nous le voulions.

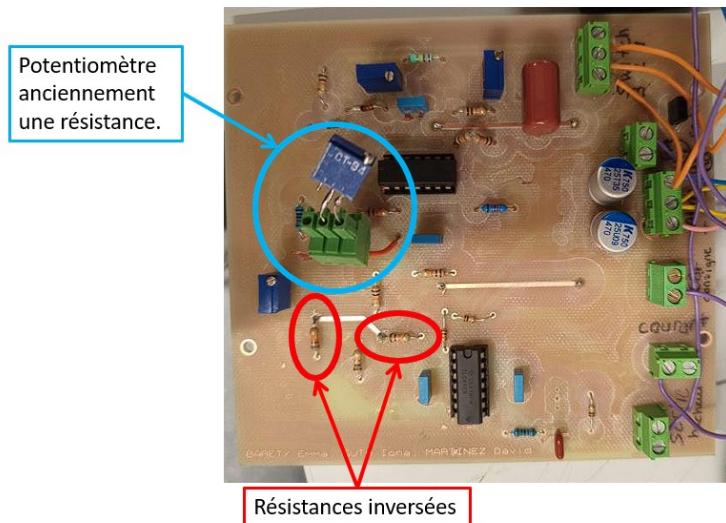


FIGURE 7.1 – Localisation des problèmes sur le PCB

### 7.1.2 Résultats des tests sur plaquette

Malgré ces quelques complications, nous avons réussi à obtenir des résultats concluant. Que ce soit au niveau de l'écrêteur, puisqu'il écrête bien à +5V et -5V.

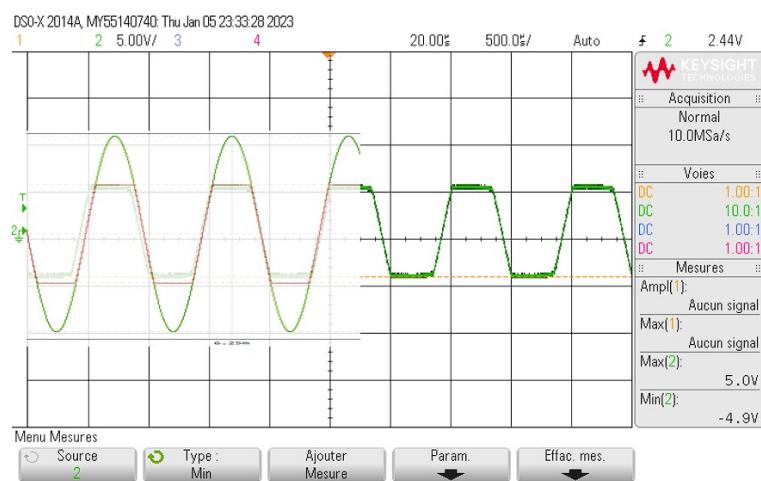


FIGURE 7.2 – Comparaison entre notre écrêteur en simulation et celui en réel 10

En plus de cela, nous avons aussi testé notre correcteur de courant et avons réussi à obtenir un gain de 8,1 sachant que l'on souhaitait un gain de 8,2, notre montage est plus que correct.

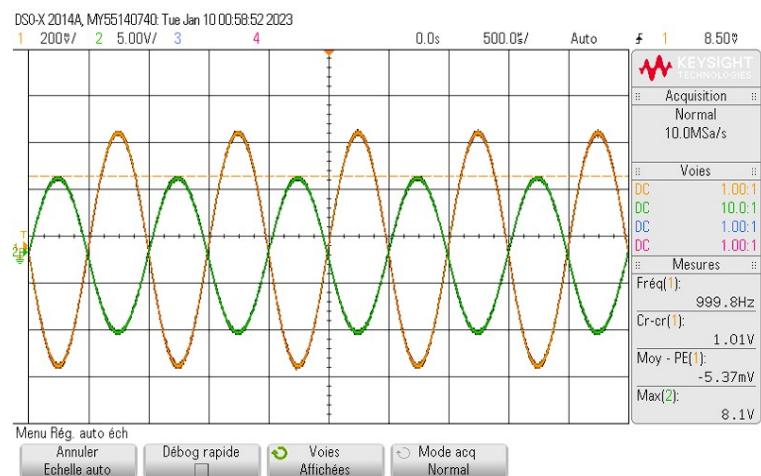


FIGURE 7.3 – Mesure du gain du correcteur P du correcteur de courant 10

De plus, nous avons testé le codeur incrémental sur plaque de test et avons obtenu les résultats suivants.

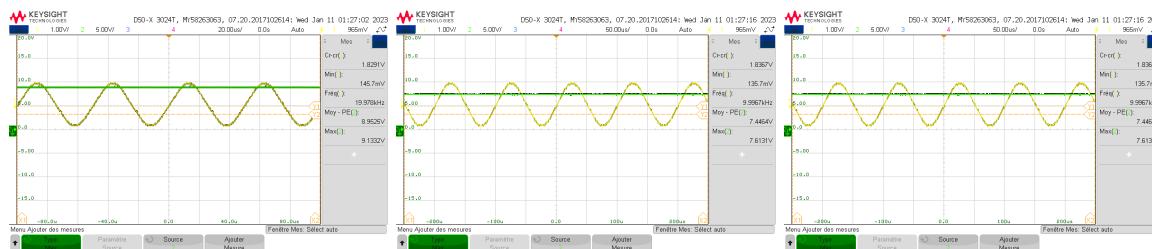


FIGURE 7.4 – Mesure du signal de sortie du codeur incrémental en fonction de la fréquence du signal d'entrée 10

Ainsi, nous pouvons remarquer que plus la fréquence du signal d'entrée est grande, plus la tension du signal de sortie est grande et vice versa.

## 7.2 Tests sur la machine à courant continu (MCC)

### 7.2.1 Essai de l'asservissement par la dynamo tachymétrique

Une fois nos tests sur plaquette réalisée, nous nous sommes attelés aux tests sur la MCC. Nous avons donc branché notre système et avons envoyé une consigne grâce à un générateur de signaux basse tension.

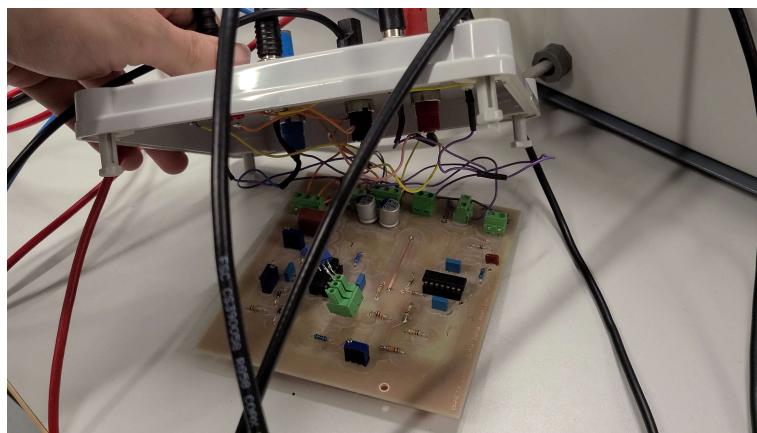


FIGURE 7.5 – Photo du système branché pour les tests sur la MCC

Ainsi, avec notre montage, nous avons réussi à asservir la MCC en récupérant le signal en sortie de la dynamo tachymétrique.

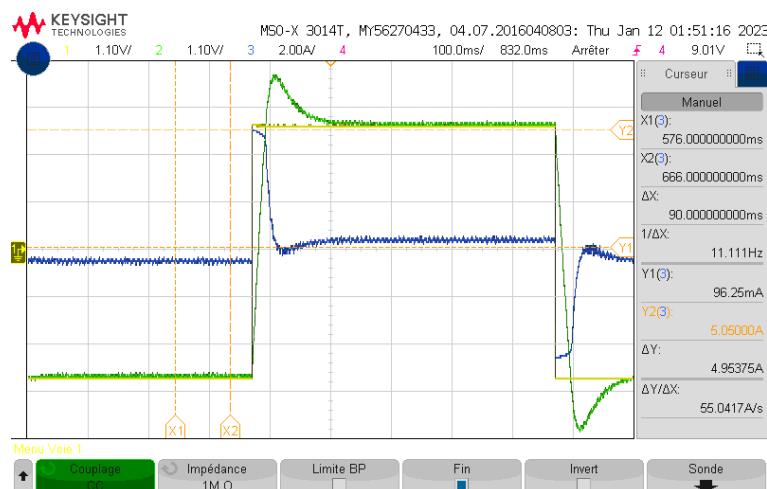


FIGURE 7.6 – Courbe de la vitesse du moteur (vert) et du courant dans le moteur (bleu) en fonction du signal de consigne (jaune) 10

Ainsi, nous pouvons voir que le dépassement est conforme au cahier des charges fonctionnelles puisqu'il est inférieur à 20% (18.87%).

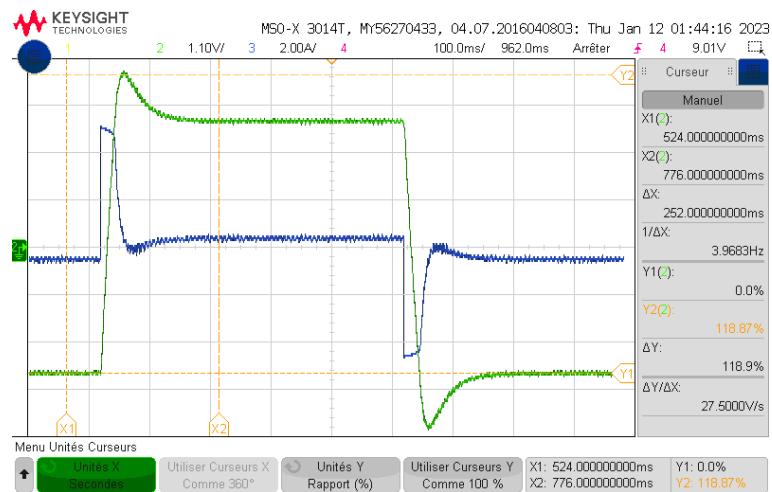


FIGURE 7.7 – Mesure du dépassement 10

De plus, nous avons mesuré le temps de réponse à 5%, et il est de 82 ms. Ce qui est très bon. Et se rapproche fortement de notre modélisation qui était de 88 ms.

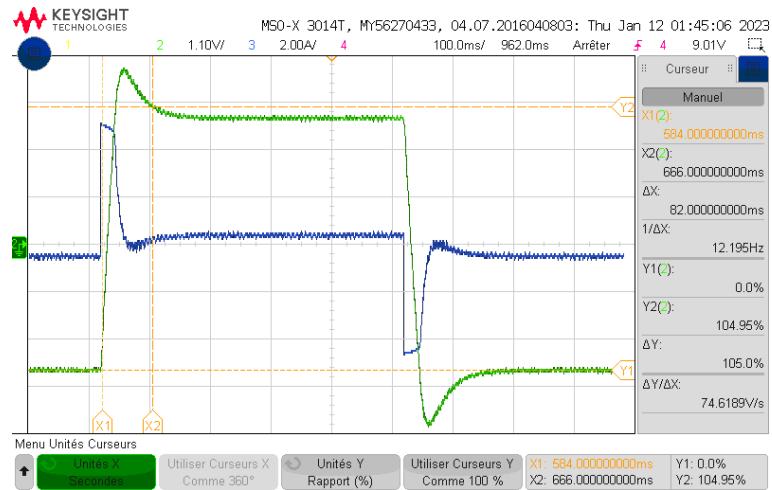


FIGURE 7.8 – Mesure du temps de réponse à 5%

## 7.2.2 Essai de l'asservissement par le codeur incrémental

Malheureusement, par manque de temps, nous n'avons pas pu tester notre asservissement avec le codeur incrémental. En effet, le codeur incrémental étant plus complexe à brancher. Il ne faut pas se tromper dans les branchements, c'est-à-dire, ne pas connecter le potentiel « plus » à la masse ni inverser les 2 pistes d'entrée du codeur qui vont permettre de différentier le sens du moteur.

## Résumé

Au cours de ce projet, nous avons réalisé l'asservissement en courant et en vitesse d'une machine à courant continu. Nous avons commencé par réaliser la modélisation de notre système sur Psim et Matlab, à vide, puis avec des frottements, une génératrice, et enfin en incluant le hacheur. Nous avons ensuite conçu et simulé les différents éléments demandés dans le cahier des charges (asservissements, écrêteur...). Cela nous a permis d'acquérir de nombreuses connaissances sur les MCC, les asservissements, les logiciels de simulation tels que Matlab/Simulink et Psim. Pour obtenir un résultat au plus proche de la réalité, nous prenions garde à ce que les résultats des deux logiciels se superposent toujours.

Puis, nous avons dimensionné nos composants en utilisant uniquement des valeurs normalisées et les avons commandés en respectant le budget de 25 euros/groupe.

Par la suite, nous avons réalisé nos PCB grâce à Proteus. Nous avons conçu trois cartes pour simplifier les soudures et les débugages. Enfin, nous avons soudé les composants, ajusté si besoin nos PCB et nous les avons débuggés. Une fois les tests sur plaquette d'essai concluants, nous avons réalisés les tests finaux sur le moteur.

Les résultats sont les suivants: l'écrêteur, l'asservissement en vitesse et en courant avec la dynamo tachymétrique sont fonctionnels. Le codeur incrémental l'est aussi sur plaquette d'essai (nous n'avons pas pu le tester sur moteur suite à des soucis de connectique, que nous espérons résoudre très prochainement).

Un des axes d'amélioration possible concerne les PCB : nous pourrions remplacer les potentiomètres par des résistance pour plus de précision et refaire celui du codeur incrémental sur un seul PCB.

En plus des connaissances techniques, nous avons appris à utiliser Latex pour nos rapports, à organiser notre travail, à travailler dans un groupe de trois de façon équitable. Nous nous sommes tous pleinement investis dans ce projet. C'était également la première fois que nous travaillons ensemble et nous avons su utiliser nos forces respectives pour le mener à bien. En conclusion, nous avons su développer nos connaissances et nos compétences pour pouvoir vous présenter un projet fonctionnel, qui satisfait le cahier des charges.

**Mots clés:**

MCC, asservissement, automatique, électrotechnique, cahier des charges, codeur incrémental, dynamo tachymétrique, génératrice, hacheur, PCB, Simulation, Simulink, Proteus, Psim

## 8. Sources

1. Cours d'automatique GE3 et GE4 : Mme Ngo et M.Durant
2. Cours d'électrotechnique GE4 : M.Flieller
3. Mémoire de fin d'études en vue de l'obtention du Diplôme de Master, Bougandoura Abdewahab  
Thème Etude sur la Commande du Hacheur Quatre Quadrants, 22 juin 2013, chapitre 2 : Etude d'un hacheur 4 quadrants <http://bib.univ-oeb.dz:8080/jspui/bitstream/123456789/7499/1/MGE-00008.pdf>  
Consulté le 10/10/22
4. Ecrêtage par Diode : [https://www.electronique-et-informatique.fr/Electronique-et-Informatique/Ecreteurs\\_negatifs\\_positifs\\_mixtes.php](https://www.electronique-et-informatique.fr/Electronique-et-Informatique/Ecreteurs_negatifs_positifs_mixtes.php)

## 9. Mots clés

- **Automatique** : L'automatique est une science qui traite la modélisation, l'analyse, l'identification et la commande des systèmes dynamiques.
- **Asservissement** : En automatique, un asservissement est un système dont l'objet principal est d'atteindre le plus rapidement possible sa valeur de consigne et de la maintenir, quelles que soient les perturbations externes. Le principe général est de comparer la consigne et l'état du système de manière à le corriger efficacement.
- **Cahier des charges** : Document détaillant les contraintes à respecter et les attentes d'un projet.
- **Codeur incrémental** : Les codeurs incrémentaux sont des systèmes qui génèrent des informations sur la position, l'angle et le nombre de tours. Ces informations sont définies à l'aide du nombre d'impulsions par rotation transmises par le codeur à la commande pour chaque rotation.
- **Dynamo tachymétrique** : capteur de vitesse constitué d'une dynamo à courant continu (bobinage avec collecteur qui tourne autour d'un aimant)
- **Électrotechnique** : science mêlant électricité et technique, qui se rapporte aux applications de l'énergie électrique.
- **Génératrice** : Machine produisant du courant électrique, généralement à partir d'énergie mécanique. C'est l'inverse d'une machine en mode moteur, qui elle produit de l'énergie mécanique à partir d'une énergie mécanique.
- **Hacheur** : Le hacheur, ou convertisseur continu - continu, est un dispositif d'électronique de puissance qui met en œuvre un ou plusieurs interrupteurs électroniques commandés pour permettre de modifier la valeur de la tension d'une source de tension continue avec un rendement élevé.
- **MCC** : Machine à courant-continu : la machine à courant continu est une machine électrique qui permet la conversion bidirectionnelle de l'énergie d'une installation électrique parcourue par un courant continu en énergie mécanique  
Toute machine à courant continu comporte deux circuits magnétiques, appelés stator (partie fixe) et rotor (partie mobile). Dans le cas du moteur à courant continu, le stator, aussi appelé inducteur, crée un champ magnétique B. Le rotor, aussi appelé induit, est alimenté en courant continu.
- **PCB (ou circuit imprimé ou carte électronique)** : Il s'agit d'un support, en général une plaque, permettant de maintenir et de relier électriquement un ensemble de composants électroniques entre eux
- **Simulation** : Production d'un modèle informatique d'un système, à des fins d'étude.

- **Simulink** : environnement de programmation graphique basé sur MATLAB pour la modélisation, la simulation et l'analyse de systèmes dynamiques multidomaines. Son interface principale est un outil graphique comprenant des diagrammes de blocs personnalisables.
- **Proteus** : logiciel de simulation permettant la réalisation de PCB
- **Psim** : Logiciel de simulation, conçu spécifiquement pour être utilisé dans les simulations d'électronique de puissance et d'entraînement de moteurs, mais il peut être utilisé pour simuler tout circuit électronique.

## 9. Table des figures

2.1	Schéma de la MCC (Machine à courant continu) à asservir . . . . .	6
2.2	Valeurs données par le constructeur . . . . .	6
3.1	Diagramme de Gantt prévisionnel 10 . . . . .	8
3.2	Diagramme de Gantt réel10 . . . . .	8
4.1	Schéma bloc du moteur à vide 10 . . . . .	10
4.2	Modélisation de la MCC à vide10 . . . . .	10
4.3	Comparaison du modèle PSIM et de la simulation Matlab . . . . .	12
4.4	Schéma bloc du moteur avec des pertes . . . . .	13
4.5	Modélisation de la MCC avec des frottements . . . . .	14
4.6	Comparaison du modèle PSIM et de la simulation Matlab . . . . .	14
4.7	Modélisation du hacheur sur PSIM . . . . .	15
4.8	Courbes représentant les différentes tensions étudiées lors de la simulation du hacheur . . . . .	16
4.9	Schéma bloc sous Simulink du moteur avec charge et frottements . . . . .	17
4.10	Modélisation du moteur avec frottements et sa charge 10 . . . . .	18
4.11	Comparaison du modèle PSIM et de la simulation Matlab . . . . .	18
5.1	Schéma bloc sous Simulink9 du système à asservir en courant . . . . .	20
5.2	Description de la simplification du système . . . . .	21
5.3	Programme que l'on a utilisé dans Matlab . . . . .	22
5.4	Fonction de transfert obtenue dans Matlab . . . . .	22
5.5	Diagramme de Black (Nichols) de la fonction de transfert G . . . . .	23
5.6	Schéma de la boucle de courant sur Matlab/Simulink . . . . .	23
5.7	Modélisation de l'asservissement en courant de notre moteur . . . . .	24
5.8	Zoom sur les correcteurs de l'asservissement . . . . .	24
5.9	Courbe représentative du courant $i_{moteur}$ après asservissement . . . . .	25
5.10	Mesure du dépassement sur le courant $i_{moteur}$ après échelon à 0,4 s . . . . .	25
5.11	Schéma d'un PI avec des AOP . . . . .	26
5.12	Schéma de l'asservissement avec des composants réels . . . . .	27
5.13	Zoom sur les correcteurs réels . . . . .	27
5.14	Courbe représentative du courant $I_{moteur}$ après asservissement avec des composants réels . . . . .	28
5.15	Mesure du dépassement sur le courant $i_{moteur}$ après échelon à 0,4 s avec des composants réels . . . . .	28
5.16	Courbes du courant dans le moteur sous PSIM (en magenta) et MATLAB (en bleu) . . . . .	29
5.17	Schéma bloc de notre simulation matlab pour l'asservissement de vitesse . . . . .	30
5.18	Schéma bloc de notre modélisation PSIM pour l'asservissement de vitesse . . . . .	31

5.19 Zoom sur les correcteurs de l'asservissement en vitesse . . . . .	31
5.20 Vtachy en fonction du temps - issu de la simulation PSIM de la figure précédente . . . . .	32
5.21 Zoom sur Vtachy en fonction du temps au niveau de l'échelon à 0,4s - issu de la simulation PSIM de la figure précédente . . . . .	32
5.22 Vtachy en fonction du temps - issu de la simulation PSIM de la figure précédente 10 . . . . .	34
5.23 Simulation de l'asservissement de vitesse entre PSIM (en magenta) et MATLAB (en bleu) . . . . .	35
5.24 Schéma du schéma écrêteur utilisé . . . . .	36
5.25 Caractéristique de l'écrêteage avec une tension d'entrée et la tension de sortie associée . . . . .	37
5.26 Fonctionnement codeur incrémental . . . . .	38
5.27 Montage du codeur incrémental . . . . .	39
5.28 Schéma final de l'asservissement de la MCC . . . . .	39
5.29 Analyse des différents courants de notre MCC . . . . .	40
<hr/>	
6.1 Photo de la boîte . . . . .	42
6.2 Dimensions du boîtier . . . . .	43
6.3 Schéma de la carte avec VGA . . . . .	44
6.4 PCB, vue 3D et soudures de la carte VGA . . . . .	44
6.5 Schéma de la carte du codeur incrémental . . . . .	45
6.6 PCB et vue 3D du codeur incrémental . . . . .	45
6.7 PCB réalisés . . . . .	45
6.8 Schéma de l'asservissement et de la dynamo tachymétrique . . . . .	46
6.9 PCB vue 3D de l'asservissement et de la dynamo tachymétrique . . . . .	47
6.10 PCB réalisé des asservissements avec la dynamo tachymétrique . . . . .	47
<hr/>	
7.1 Localisation des problèmes sur le PCB . . . . .	48
7.2 Comparaison entre notre écrêteur en simulation et celui en réel 10 . . . . .	49
7.3 Mesure du gain du correcteur P du correcteur de courant 10 . . . . .	49
7.4 Mesure du signal de sortie du codeur incrémental en fonction de la fréquence du signal d'entrée 10 . . . . .	49
7.5 Photo du système branché pour les tests sur la MCC . . . . .	50
7.6 Courbe de la vitesse du moteur (vert) et du courant dans le moteur (bleu) en fonction du signal de consigne (jaune) 10 . . . . .	50
7.7 Mesure du dépassement 10 . . . . .	51
7.8 Mesure du temps de réponse à 5% . . . . .	51

# 10. Liens des fichiers des figures

- Figure 3.1 : Diagramme de Gantt prévisionnel :  
*PROJETGE1/PrevisionnelDiagrammeGantt*
- Figure 3.2 : Diagramme de Gantt réel :  
*DiagrammedeGranttre : PROJETGE1/DiagrammedeGantt*
- Figure 4.1 : schéma bloc du moteur à vide :  
*PROJETGE1//septembre//21septembre//simulation – matlab – a – vide*  
 Figure 4.2 : Modélisation de la MCC à vide  
*PROJET – GE1//septembre//21septembre//Moteur – seul – psim*
- Figure 4.3 : Comparaison du modèle PSIM et de la simulation matlab  
*PROJET – GE1//septembre//21septembre//simulink – X – psim – ImoteurVtachy*
- Figure 4.4 : Schéma bloc du moteur avec des pertes  
*PROJET – GE1//septembre//21septembre//simulation – matlab – moteurpertes*
- Figure 4.5 : Modélisation de la MCC avec des frottements  
*PROJETGE1//septembre//21septembre//moteurfrottementsseulpsim*
- Figure 4.6 : Comparaison de la modélisation PSIM et de la simulation simulink  
*PROJET – GE1//septembre//21septembre//Simulink – X – psim – ImoteurVtachy*
- Figure 4.7 : Modélisation du hacheur sur PSIM  
*PROJET – GE1//octobre//13octobre//hacheur – seul – psim*
- Figure 4.8 : Courbe représentant les différentes tensions étudiées lors de la simulation PSIM du hacheur  
*PROJET – GE1//octobre//13octobre//hacheur – seul – psim*
- Figure 4.9 : Schéma bloc du moteur avec charge et frottements  
*PROJET – GE1//septembre//21septembre//simulation – matlab – moteurpertes – generatrice*
- Figure 4.10 : Modélisation du moteur avec frottements et sa charge  
*PROJET – GE1//septembre//septembre//21septembre//simulation – matlab – moteurpertes – generatrice*
- Figure 4.11 : Comparaison du modèle Psim et de la simulation Matlab  
*PROJET – GE1//septembre//28septembre//simulink – X – psim – ImoteurVtachy – hacheur*
- Figure 5.1 : Schéma bloc sous Simulink du système à asservir en courant  
*PROJET – GE1//octobre//19octobre//simulation – matlab – asser – courant*
- Figure 5.3 : Extrait du programme Matlab  
*PROJET – GE1//octobre//19octobre//calcul – simp – courant*
- Figure 5.4 : Fonction de transfert obtenue dans Matlab  
*PROJET – GE1//octobre//19octobre//calcul – simp – courant*
- Figure 5.5 : Diagramme de Black (Nichols) de la fonction de transfert G

*PROJET – GE1//octobre//19octobre//calcul – simp – courant*

- Figure 5.6 : Schéma de la boucle de courant sur Matlab/Simulink

*PROJET – GE1//octobre//19octobre//matlab – system – simplifie – courant*

- Figure 5.7 : Modélisation de l'asservissement en courant de notre moteur

*PROJET – GE1//octobre//18octobre//asservissement – courant – psim*

- Figure 5.8 : Zoom sur les correcteurs de l'asservissement

*PROJET – GE1//octobre//18octobre//asservissement – courant – psim*

- Figure 5.9 : Courbe représentative du courant  $i_{mot}$

*PROJET – GE1//octobre//18octobre//asservissement – courant – psim*

- Figure 5.10 : Mesure du dépassement sur le courant  $i – moteuraprschelonde0,4s$

*PROJET – GE1//octobre//18octobre//asservissement – courant – psim*

- Figure 5.11 : Schéma d'un PI avec des AOP

*PROJET – GE1//octobre//18octobre//aop – asservissement – courant – psim*

- Figure 5.12 : Schéma de l'asservissement avec des composants réels

*PROJET – GE1//octobre//18octobre//aop – asservissement – courant – psim*

- Figure 5.13 : Zoom sur les composants réels

*PROJET – GE1//octobre//18octobre//aop – asservissement – courant – psim*

- Figure 5.14 : Courbe représentative du courant  $I_{mot}$  après asservissement avec des composants réels

*PROJET – GE1//octobre//18octobre//aop – asservissement – courant – psim*

- Figure 5.15 : Mesure du dépassement avec le courant  $i_{moteur}$  après échelon à 0,4s avec des composants réels

*PROJET – GE1//octobre//18octobre//aop – asservissement – courant – psim*

Figure 5.16 : Courbes du courant dans le moteur Psim (en magenta et Matlab (en bleu)

*PROJET – GE1//octobre//19octobre//comparaison – PSIMxMATLAB – asser – courant*

- Figure 5.17 : Schéma bloc de notre simulation Matlab pour l'asservissement de vitesse

*PROJET – GE1//novembre//16novembre//comparaison – PSIMxMATLAB – vitesse*

- Figure 5.18 : Schéma de notre modélisation PSIM pour l'asservissement de vitesse

*PROJET – GE1//novembre//9novembre//asservissement – vitesse – psim*

- Figure 5.19 : Zoom sur les correcteurs de l'asservissement en vitesse

*PROJET – GE1//novembre//9novembre//asservissement – vitesse – psim*

- Figure 5.20 : Vtachy en fonction du temps-issu de la simulation PSIM de la figure précédente

*PROJET – GE1//novembre//9novembre//asservissement – vitesse – psim*

- Figure 5.21 : Zoom sur Vtachy en fonction du temps au niveau de l'échelon à 0.4s - issu de la simulation de la figure précédente

*PROJET – GE1//novembre//9novembre//asservissement – vitesse – psim*

- Figure 5.22 : Vtachy en fonction du temps-issu de la simulation PSIM de la figure précédente

*PROJET – GE1//novembre//13novembre//asservissement – vitesse – psim*

- Figure 5.24 : Schéma de l'écrêteur utilisé

*PROJET – GE1//novembre//22novembre//PSIM – codeur – incremntal – ok(2sens)*

- Figure 5.25 : Caractéristique de l'écrêtage avec une tension d'entrée et une tension de sortie associée

*PROJET – GE1//novembre//22novembre//PSIM – codeur – incremntal – ok(2sens)*

- Figure 5.27 : Montage du codeur incrémental

*PROJET – GE1//novembre//22novembre//PSIM – codeur – incremntal – ok(2sens)*

- Figure 5.28 : Schéma final de l'asservissement de la MCC

*PROJET – GE1//novembre//22novembre//PSIM – codeur – incremntal – ok(2sens)*

- Figure 5.29 : Analyse des différents courants de notre MCC

*PROJET – GE1//novembre//22novembre//PSIM – codeur – incremntal – ok(2sens)*

- Figure 6.3 : Schéma de la carte avec VGA

- Figure 6.4 : PCB vue 3D et soudures de la carte VGA

*PROJET – GE1//PCB//Prise – VGA*

- Figure 6.5 : Schéma de la carte du codeur incrémental

*PROJET – GE1//dembre//08dembre//PCB – codeur – incremental – 8.14*

- Figure 6.6 : PCB et vue 3D du codeur incrémental

*PROJET – GE1//dembre//08dembre//PCB – codeur – incremental – 8.14*

- Figure 6.8 : Schéma de l'asservissement et de la dynamo tachymétrique

*PROJET – GE1//PCB//Projet – ge1 – barety – guth – martinez*

- Figure 6.9 : PCB vue 3D de l'asservissement et de la dynamo tachymétrique

*PROJET – GE1//PCB//Projet – ge1 – barety – guth – martinez*

- Figure 7.2 : Comparaison entre notre écrêteur en simulation et celui en réel

*Test : PROJET – GE1//scopes//test – ecreteur // Simulation :*

*PROJET – GE1//novembre//22novembre//PSIM – codeur – incremntal – ok(2sens) //*

- Figure 7.3 : Mesure du gain du correcteur P du correcteur de courant

*PROJET – GE1//scopes//Mesure – du – gain – du – correcteur – P – du – correcteur – de – courant*

- Figure 7.4 : Mesure du signal de sortie du codeur incrémental en fonction de la fréquence du signal d'entrée

*PROJET – GE1//scopes//Mes – cod – 1//*

*PROJET – GE1//scopes//Mes – cod – 2//*

*PROJET – GE1//scopes//Mes – cod – 3//*

- Figure 7.6 : Courbe de la vitesse du moteur (vert) et du courant dans le moteur (bleu) en fonction du signal de consigne (jaune)

*PROJET – GE1//scopes//scope – asser – vitesse – tachy*

- Figure 7.7 : Mesure du dépassement

*PROJET – GE1//scopes//Mesure – Depassement*

- Figure 7.8 : Mesure du temps de réponse à 5%

*PROJET – GE1//scopes//Mesure – tps – rep*