



SPÉCIALITÉ GÉNIE ELECTRIQUE, QUATRIÈME ANNÉE - GE4
ANNÉE 2025-2026, SEMESTRE 7

STM-GE-08-PROJET GES7

Rapport final

Asservissement de machine à courant continu

ARCHIS - IDMONT - RAMDANI - SCHMITT STATUCKI

Strasbourg - FR, 2025-2026

Table des matières

Introduction	5
1 Présentation des machines à courant continu utilisées	5
1.i Fonctionnement général d'une machine à courant continu	5
1.ii Caractéristiques des machines	6
1.iii Équations de fonctionnement	6
2 Structure du système complet	7
3 Cahier des charges	8
4 Temporalité du projet	8
5 Modélisation et simulation du système en boucle ouverte	9
5.i Modèle du moteur seul sans frottements ni charge	9
5.i.a Sous Matlab/Simulink	9
5.i.b Sous PSIM	10
5.i.c Comparaison des deux modèles	12
5.ii Prise en compte des frottements et de la charge	13
5.ii.a Notions théoriques	13
5.ii.b Implémentation sous PSIM	14
5.ii.c Implémentation sous Matlab/Simulink	14
5.ii.d Comparaison des deux modèles	15
5.iii Etude du régime permanent	17
5.iv Modélisation du hacheur	18
5.iv.a Notions théoriques	18
5.iv.b Implémentation sous PSIM	19
5.iv.c Implémentation sous Matlab/Simulink	19
5.v Prise en compte de la commande	19
5.v.a Notions théoriques	19
5.v.b Implémentation sous PSIM	20
5.v.c Implémentation sous Matlab/Simulink	20
5.vi Etude du régime permanent considérant la commande	21
5.vi.a Sous Matlab/Simulink	21
5.vi.b Sous PSIM	21
6 Dimensionnement et simulation des asservissements	23
6.i Préliminaire à l'asservissement	23
6.ii Asservissement en courant	24
6.ii.a Calcul du correcteur	24
6.ii.b Vérification du correcteur	27
6.iii Asservissement de vitesse	28
6.iii.a Démarche globale	28
6.iii.a.1 Calcul du correcteur	28
6.iii.a.2 Vérification du correcteur	32
6.iii.b Prise en compte des capteurs	33
6.iv Lien entre valeur de référence et valeur moyenne	35
7 Conception et dimensionnement des circuits électroniques	36
7.i Asservissement	36
7.i.a Compartateur	36
7.i.b Correcteur de vitesse	37
7.i.c Correcteur de courant	38
7.ii Circuit de mise en forme du signal tachymétrique	38
7.iii Circuit de mise en forme du signal du codeur incrémental	38
7.iv Limiteur de courant	43
7.v Limiteur de la tension de commande	44
7.vi Indicateur de sens de rotation	44

7.vii Récapitulatif	44
8 Simulation du système réel	45
9 Réalisation pratique	47
9.i Implémentation dans Proteus	47
9.ii Création du PCB	53
9.ii.a Règles de conception	53
9.ii.b Modélisations finies	54
9.iii Soudures et Tests de la carte	57
9.iv Tests préliminaires	58
Arborescences des fichiers par figure	59

Liste des tableaux

1	Caractéristiques des MCC	6
2	Tableau récapitulatif des composants utilisés	44

Table des figures

1	Schéma équivalent d'un moteur à courant continu	6
2	Structure globale du système	7
3	Diagramme de Gant prédictif	8
4	Fichier Matlab de déclaration des constantes : Moteur seul sans frottements ni charge *	9
5	Schéma bloc Simulink du moteur seul sans frottements ni charge *	10
6	Essai pour la détermination de la MCC équivalente sur PSIM *	10
7	Moteur seul sans frottements ni charge sur PSIM *	11
8	Comparaison Matlab/PSIM moteur seul : courant d'induit *	12
9	Comparaison Matlab/PSIM moteur seul : vitesse *	12
10	Schéma équivalent du système avec une charge	13
11	Montage du système muni des éléments de frottement et de la charge simulés sur PSIM *	14
12	Montage du système muni des éléments de frottement et de la charge simulé sur Matlab *	14
13	Comparaison des courants induit simulés sur Matlab en bleu et sur PSIM en rouge *	15
14	Zoom sur le pic de courant *	15
15	Comparaison des vitesses simulés sur Matlab en bleu et sur PSIM en rouge *	16
16	Zoom sur le régime permanent *	16
17	Schéma de principe d'un hacheur en pont	18
18	Signaux de commande MLI du hacheur	18
19	Implémentation du hacheur sur PSIM *	19
20	Signal d'entrée de la MCC	19
21	Modélisation de la commande sous PSIM *	20
22	Modélisation de la commande sous Matlab *	20
23	Reproduction de la formule $< U_{AB} >$ sous PSIM *	21
24	Courant d'induit moteur avec hacheur (vert) et courant d'induit moteur avec modélisation du hacheur (rouge) *	22
25	Vitesse moteur avec hacheur (vert) et vitesse moteur avec modélisation du hacheur (rouge) *	22
26	Zoom sur l'ondulation de vitesse *	23
27	Schéma-bloc du système global	24
28	Schéma-bloc simplifié du système global	24
29	Schéma-bloc Simulink *	25
30	Comparaison du courant d'induit issu du schéma global et du modèle *	25
31	Boucle fermée de courant *	26
32	Réponse asservie du courant d'induit *	26
33	Schéma de vérification de l'asservissement de courant sur PSIM *	27
34	Courbes de vérification de l'asservissement de courant sur PSIM *	27
35	Fonction du huitième ordre *	28
36	Comparaison des 3 modèles *	29
37	Schéma-bloc de l'asservissement de vitesse final	30
38	Asservissement simplifié de la vitesse	31
39	Graphe de la vitesse asservie *	31
40	Vérification de l'asservissement de vitesse : Schéma PSIM *	32
41	Courbes de vérification de l'asservissement de vitesse sur PSIM *	32
42	Modifications prenant en compte les capteurs *	33
43	Schéma-bloc de l'asservissement avec capteurs *	34
44	Vérification de l'asservissement avec capteur (vitesse) *	34
45	Vérification de l'asservissement avec capteur (courant) *	35
46	Signaux liés au codeur incrémental	35
47	Circuit comparateur	36
48	Circuit correcteur PI	37
49	Principe de fonctionnement du codeur incrémental	38
50	Signaux liés au codeur incrémental	39

51	NE555 monté en astable	40
52	Circuit déivateur	40
53	Schéma de simulation de la méthode 1 sous PSIM	41
54	Résultat méthode 1	42
55	Multiplexeur analogique	42
56	Schéma de simulation de la méthode 2 sous PSIM	43
57	Circuit limiteur de courant	43
58	Circuit indicateur de sens de rotation	44
59	Schéma de simulation du système réel sous PSIM - Boucle d'asservissement	45
60	Schéma de simulation du système réel sous PSIM - Partie capteurs	45
61	Résultats pour une consigne positive avec le capteur tachymétrique	46
62	Résultats pour une consigne négative avec le capteur tachymétrique	46
63	Résultats pour une consigne positive avec le codeur incrémental	46
64	Résultats pour une consigne négative avec le codeur incrémental	47
65	Entrées de la carte	48
66	Première chaîne	48
67	Seconde partie de la chaîne principale	49
68	Chaîne complète d'asservissement	49
69	Comparateur de tension	50
70	Pont diviseur de tension par deux	50
71	Déivateur + filtre passe bas + Gain variable	51
72	Circuit à bascule D	51
73	Circuit de Multiplexage à AOP	52
74	Chaîne complète d'acquisition du codeur	52
75	Diviseur de tension pour le signal issu de la dynamo tachymétrique.	53
76	Design du PCB avant impression	54
77	Modélisation du PCB accueillant le composant CMS	55
78	PCB réel	55
79	Modélisation du PCB accueillant le connecteur VGA	56
80	PCB assurant le lien avec le connecteur VGA	56
81	Schéma de câblage du connecteur VGA	56
82	Face CMS	57
83	Face traversants	57
84	Schéma de test préliminaire sur PSIM	58

Introduction

Le projet GE-S7 mêle les domaines de l'automatique et de l'électrotechnique dans le contexte de l'asservissement de machines à courant continu. En particulier, nous réaliserons un asservissement de courant et de vitesse en accord avec un cahier des charges établi au préalable. Ce projet va se dérouler en plusieurs étapes majeures que nous pouvons résumer de la manière suivante :

- Étude du système à asservir
- Modélisation et simulation du système en boucle ouverte
- Dimensionnement et simulation des asservissements
- Conception et réalisation du circuit imprimé
- Tests réels et comparaison avec les simulations

Le projet a pour objectifs de réaliser l'asservissement d'une machine et d'en comprendre les enjeux à travers les différentes étapes décrites ci-dessus. Il nous permet de nous mettre dans un cadre concret pour développer des compétences qui seront importantes dans notre futur métier d'ingénieur et en particulier, la démarche de projet et le travail en équipe.

Nous serons amenés dans ce rapport à fournir de nombreuses captures d'écran. Pour éviter d'alourdir le document et faciliter la lecture, nous indiquerons l'origine de chaque capture d'écran dans la section 9.iv où le fichier d'origine de chaque figure sera indiquée. Il sera possible de s'y rendre en cliquant sur l'astérisque "" présente à la fin de chaque titre de figure, et de revenir à la figure en cliquant sur le numéro de cette dernière dans l'arborescence.*

1 Présentation des machines à courant continu utilisées

1.i Fonctionnement général d'une machine à courant continu

Une machine à courant continu est composée comme toute machine électrique tournante d'une partie fixe : le stator, et d'une partie rotative : le rotor. Le stator (*ou inducteur*) consiste soit en un aimant permanent, soit en un électroaimant. Le rotor (*ou induit*), lui, est bobiné et va être traversé par un courant continu.

L'interaction entre le courant au rotor et le champ magnétique du stator va générer une force sur les conducteurs rotoriques permettant la rotation de la machine. De plus, pour permettre à la machine de tourner et de ne pas se bloquer du fait de la polarité du champ magnétique, une pièce appelée "collecteur" permet d'inverser la polarité du courant traversant les enroulements lors de la rotation, permettant ainsi un mouvement continu de la machine.

De ce fait, si nous alimentons l'induit de notre MCC avec un courant continu (et éventuellement l'inducteur si elle n'est pas à aimant permanent), nous fonctionnerons en mode **moteur**, et à l'inverse, en appliquant un mouvement de rotation sur l'arbre de notre MCC, nous fonctionnerons en **génératrice**.

Dans ce rapport, nous abrégerons souvent "machine à courant continu" par MCC.

1.ii Caractéristiques des machines

Nous utiliserons donc dans ce projet deux machines **identiques**. Une fonctionnera en moteur, l'autre en génératrice. Le type de machine utilisé est à **aimant permanent**, il ne sera donc pas nécessaire d'alimenter l'inducteur. Les machines sont à deux paires de pôles et leurs caractéristiques sont listées dans le [tableau 1](#).

TABLE 1 – Caractéristiques des MCC

Caractéristique	Valeur	Unité	Symbol
Couple en rotation lente	0,54	N.m	M_O
Courant nominal	4,5	A	I
Tension nominale d'alimentation	49	V	U
Vitesse nominale	3000	tr/min	N
Tension maximal	65	V	U_{max}
Vitesse maximale	4800	tr/min	N_{max}
Courant impulsional	13	A	I_{max}
Fem par 1000 tr/min à 25°	13,3	$\frac{V}{10^3 \cdot tr/min}$	$K_e(tr/min)$
Coefficient de couple électromagnétique	0,127	N.m/A	K_ϕ
Couple de frottement sec	2,4	N.cm	T_f
Coefficient de viscosité par 1000 tr/min	0,53	N.cm	K_d
Résistance de bobinage à 25°	1,52	Ω	R
Inductance de bobinage	2,2	mH	L
Inertie du rotor	0,000083	kg.m²	J
Constante de temps thermique	7	min	T_{th}
Masse	1,34	kg	M

1.iii Équations de fonctionnement

Pour les besoins du projet, nous aurons besoin des équations de fonctionnement de la MCC en mode moteur. À partir du schéma équivalent d'un moteur à courant continu illustré à la [figure 1](#) et des équations de la mécanique, nous obtenons les équations de fonctionnement.

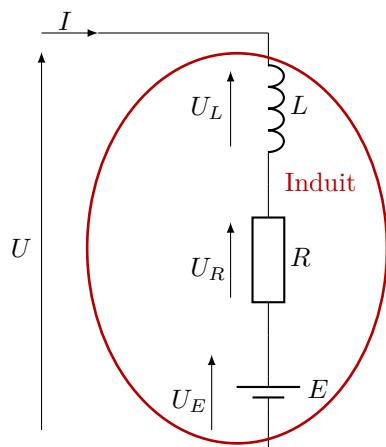


FIGURE 1 – Schéma équivalent d'un moteur à courant continu

Équations de fonctionnement :

- $u(t) = e(t) + Ri(t) + L \frac{di(t)}{dt}$
 - $J \frac{d\omega}{dt} = C_m(t) - C_r(t)$
 - $e(t) = K_e \omega(t)$
 - $C_m = K_c i(t)$
- (1)

Avec :

- J : l'inertie du rotor
- C_m : le couple magnétique
- C_r : le couple résistant ; comportant le couple de frottement et le couple de charge.
- K_e : la constante de force électromotrice
- K_c : la constante de couple
- ω : la vitesse de rotation du moteur
- R : la résistance d'induit (ou de bobinage)
- L : l'inductance d'induit (ou de bobinage)
- u : la tension aux bornes de l'induit
- e : la force électromotrice
- i : le courant d'induit

À noter qu'en fonctionnement moteur, nous sommes en convention récepteur. Pour un fonctionnement en génératrice, on passe en convention générateur et donc le sens du courant est inversé. Ainsi, on garde les mêmes équations à des signes près. De même, le signe des couples changera lui aussi. Il est ainsi aisément de déduire les équations en mode génératrice au besoin.

2 Structure du système complet

La structure globale du système est présentée en [figure 2](#) :

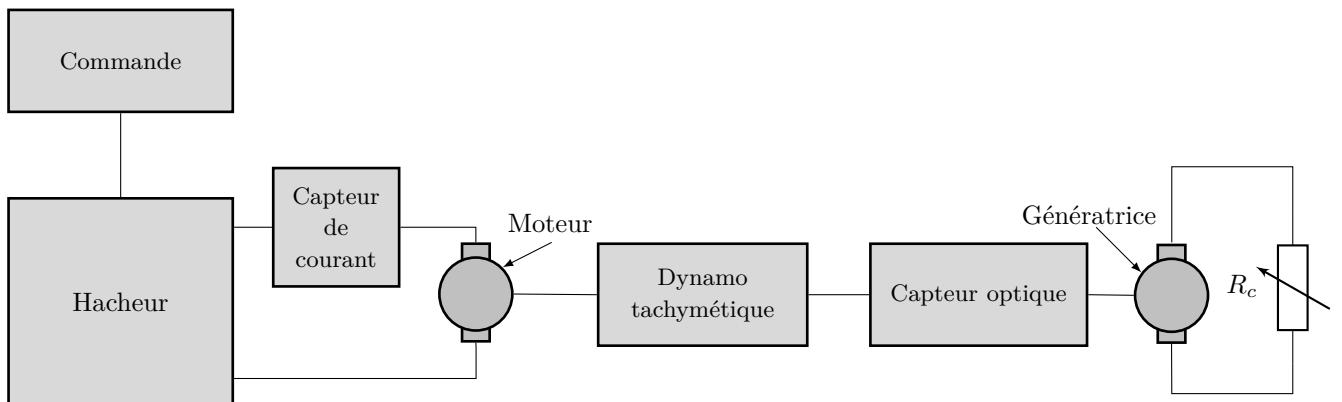


FIGURE 2 – Structure globale du système

Nous avons donc un hacheur en pont commandable alimenté en 48V-DC qui alimente notre moteur. À l'intérieur du hacheur se trouve un capteur de courant permettant de mesurer le courant en sortie de ce dernier. Le moteur et la génératrice se trouvent sur un même banc et sont mécaniquement couplés. On retrouve aussi sur ce banc une dynamo tachymétrique et un capteur optique permettant la mesure de la vitesse de rotation de l'arbre des deux machines. Le choix du capteur sera possible via un commutateur électrique. La génératrice alimentera un rhéostat R_c . Le moteur aura donc pour charge la génératrice alimentant le rhéostat.

Notre objectif sera d'ajouter à ce système une carte d'asservissement permettant de satisfaire les demandes du cahier des charges. Cette carte utilisera les données des différents capteurs à notre disposition et viendra agir sur les variables du système.

3 Cahier des charges

Nous avons pour ce projet un cahier des charges à respecter. Le cœur de ce dernier est comme nous l'avons dit l'asservissement en vitesse et en courant de notre MCC, c'est donc sur ces deux aspects que va se focaliser le cahier des charges.

Les contraintes que nous avons sont les suivantes :

- Le courant maximal ne devra pas être supérieur à 1,1 fois le courant nominal.
- La vitesse ne devra pas dépasser la vitesse maximale imposée par le constructeur.
- Les machines devront tourner à la vitesse de référence et le courant devra s'adapter en conséquence.

Ainsi, pour respecter les contraintes précédentes et avoir un système performant nous tâcherons de respecter le cahier des charges suivant :

- Les erreurs en régime permanent des deux grandeurs devront être nulles.
- Les dépassements en vitesse et en courant devront se situer entre 10% et 20% au maximum.
- Le temps de réponse de l'asservissement en courant devra être équivalent à dix fois la période du signal MLI du hacheur.
- Le temps de réponse de l'asservissement en vitesse doit être trois fois plus rapide que le temps de réponse du système en boucle ouverte.

Pour résumer, le cahier des charges impose une vitesse suivant une valeur de référence et un courant qui s'y adapte, cela avec des valeurs de dépassement comprises entre 10% et 20% au maximum et un temps de réponse fixé en fonction de la structure du système.

4 Temporalité du projet

Dans un soucis d'organisation, il nous a été suggéré de créer un diagramme de Gant. Cette représentation sert à organiser la temporalité de chaque activité en lien avec le projet. Pour chaque élément, des quantifications en semaines sont inscrites pour identifier le début et la durée totale de chacune d'entre elles. Voici donc notre diagramme de Gant prédictif :



FIGURE 3 – Diagramme de Gant prédictif

5 Modélisation et simulation du système en boucle ouverte

Nous allons donc commencer par la modélisation et la simulation de notre système en boucle ouverte afin de pouvoir par la suite dimensionner nos correcteurs sur un modèle étant le plus fidèle possible par rapport à la réalité.

Dans tout ce rapport on considère que la puissance électromagnétique est la même que cela soit d'un point de vue électrique ou mécanique, ainsi on a dans l'[équation 1](#) : $K_c = K_e = K_\phi$.

5.i Modèle du moteur seul sans frottements ni charge

Dans un premier temps, nous ne prendrons pas en compte les frottements et le moteur fonctionnera seul, sans charge. Ainsi, on aura $C_r = 0$ (*cf. équation 1*).

5.i.a Sous Matlab/Simulink

Nous voulons étudier le courant et la vitesse de notre système, ainsi, nous allons faire en sorte que dans notre modèle nous puissions accéder à ces grandeurs pour les analyser.

A partir des équations (1), l'induit de la MCC utilisé en moteur peut être modélisé par l'équation suivante :

$$u(t) = R \cdot i_M(t) + L \cdot \frac{di_M(t)}{dt} + K_\phi \omega(t)$$

En utilisant la transformée de Laplace, on a :

$$U - K_\phi \Omega = I_M(R + Lp)$$

On peut en déduire la fonction de transfert donnant accès au courant suivante :

$$H_1(p) = \frac{I}{U - K_\phi \Omega} = \frac{1}{R + L \cdot p}$$

De plus, toujours grâce aux équations (1) et sachant que $C_r = 0$, on a :

$$J \cdot \frac{d\Omega}{dt} = K_\phi i_M$$

De la même façon, on obtient la fonction de transfert donnant accès à la vitesse suivante :

$$H_2(s) = \frac{\Omega}{I_M} = \frac{K_\phi}{J \cdot p}$$

Les deux fonctions de transfert nous permettent bien d'obtenir le courant de l'induit moteur ainsi que la vitesse du moteur dans une même simulation.

Dans Matlab, on utilise l'outil Simulink pour modéliser notre système puis on implémente les deux fonctions de transfert. On identifie le courant moteur et la vitesse afin d'utiliser les valeurs de leurs courbes pour s'assurer de l'équivalence avec la simulation PSIM que nous effectuerons plus tard. Le schéma est montré en [figure 5](#).

Tout au long du projet, nous utiliserons un fichier Matlab qui nous permettra de définir et stocker toutes nos constantes. Voici ce dernier en [figure 4](#) incluant les constantes nécessaires à la simulation de cette partie. Nous le compléterons au fil des simulations avec les grandeurs nécessaires mais sa structure restera la même :

```
% données pour 1 moteur
R=1.52; % résistance de l'induit
Rch=10; % Résistance de charge
C0=2.2e-2; % Couple de frottement sec pour une machine
f=0.53e-2/1000*(30/pi); % coefficient de frottement visqueux pour une machine
L=2.2e-3; % inductance propre de l'induit
J=8.3e-5; % moment d'inertie pour 1 machine
U=48; % tension d'alimentation
KPhi=0.127; % constante de couple, de fem
UM=15; % tension de commande
u_commande=11.25;
```

FIGURE 4 – Fichier Matlab de déclaration des constantes : Moteur seul sans frottements ni charge *

On réalise ainsi le schéma bloc suivant :

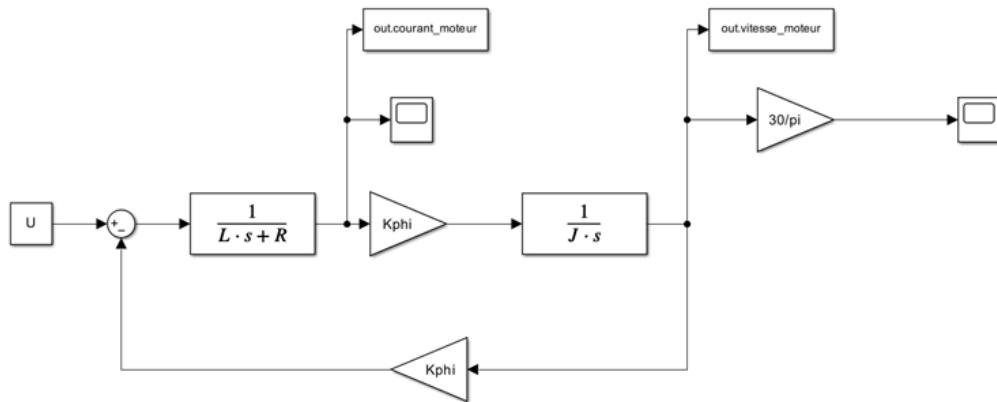


FIGURE 5 – Schéma bloc Simulink du moteur seul sans frottements ni charge *

Les blocs "out." permettent d'acquérir et de rendre accessible les valeurs du courant d'induit et de la vitesse de rotation du moteur dans les scripts Matlab. Le facteur $\frac{30}{\pi}$ permet de fournir une réponse en tr/min et non en rad/s dans l'oscilloscope.

5.i.b Sous PSIM

Parallèlement à cela, nous réalisons une simulation PSIM de notre moteur. Il y a un point important à aborder avant de continuer. Notre moteur réel est une MCC à aimants permanents, mais sur le logiciel PSIM, nous ne disposons pas du modèle de ce moteur. Nous avons néanmoins accès à un modèle de machine à excitation bobinée, qui possède les mêmes caractéristiques générales (résistance d'induit, moment d'inertie, etc). Nous devrons donc dans un premier temps faire des essais avec ce modèle afin de trouver un coefficient à ajouter à la tension d'inducteur permettant de réaliser une équivalence. En effet, une MCC à excitation bobinée est définie par ces caractéristiques internes mais aussi par sa tension d'excitation. C'est sur cette dernière qu'il faudra jouer pour obtenir l'équivalence.

Pour ce faire, nous commençons par nous placer en charge avec une seconde machine en génératrice, dont les caractéristiques seront les mêmes que pour notre MCC réelle et avec une tension d'excitation arbitraire mais fixée (on choisira 10 V). La disposition de cette essai est présentée en [figure 6](#).

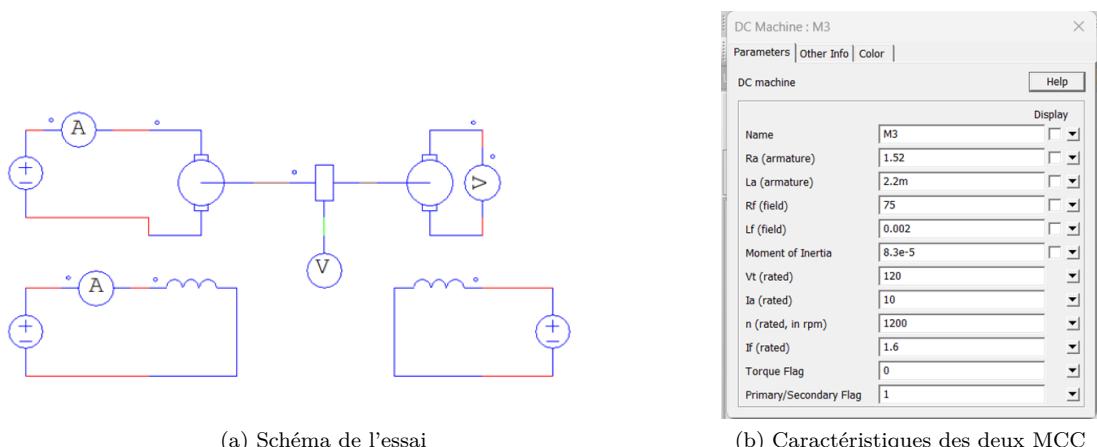


FIGURE 6 – Essai pour la détermination de la MCC équivalente sur PSIM *

L'idée est ici la suivante. Nos deux types de moteurs, à aimant permanent et à excitation bobinée, ont un coefficient de force électromotrice qui leur est propre. Nous allons donc chercher ce coefficient pour le moteur sur PSIM afin de pouvoir calculer le rapport des coefficient des deux machines, la réelle et celle sur PSIM avec la tension d'excitation précédemment mentionnée. Rappelons la formule du coefficient de force électromotrice :

$$K_\phi = \frac{E_{gen}}{\Omega}$$

Où E_{gen} est la tension induite au générateur et Ω la vitesse de rotation de l'ensemble.

On mesure donc la vitesse qui est donnée ici en tr/min , puis on la convertit en rad/s . Il suffit pour cela de la multiplier par $\frac{2\pi}{60}$. On obtient ainsi une vitesse angulaire de $690,67 rad/s$. La force électromotrice E_{gen} est en régime permanent, la tension que l'on a imposé à l'induit, soit $48V$, on calcule alors K_ϕ :

$$K_\phi = \frac{48}{690,67} = 0,0694976577$$

On peut à présent faire le rapport des coefficients de nos deux modèles, on calcule :

$$\frac{0,127}{0,0694976577} = 1,827480936166667$$

Ainsi, comme la relation entre la tension d'excitation et le coefficient de force électromotrice est proportionnelle (c'est à dire qu'affecter à la tension d'excitation un certain coefficient revient à affecter ce même coefficient au coefficient force électromotrice tout en gardant la tension constante), nous devons multiplier la tension d'inducteur par le coefficient précédent. Dès lors, dans toutes nos simulations, nous devrons ajouter ce coefficient à nos tensions d'inducteurs pour chaque machine. C'est bien à l'inducteur et non à l'induit que ce coefficient apparaît, car c'est l'inducteur qui diffère entre les deux machines.

Nous pouvons maintenant enfin passer à la simulation en tant que telle. Rappelons que nous cherchons tout d'abord à simuler le moteur à vide, plaçons alors un moteur sans aucune charge. Le schéma est présenté en figure 7.

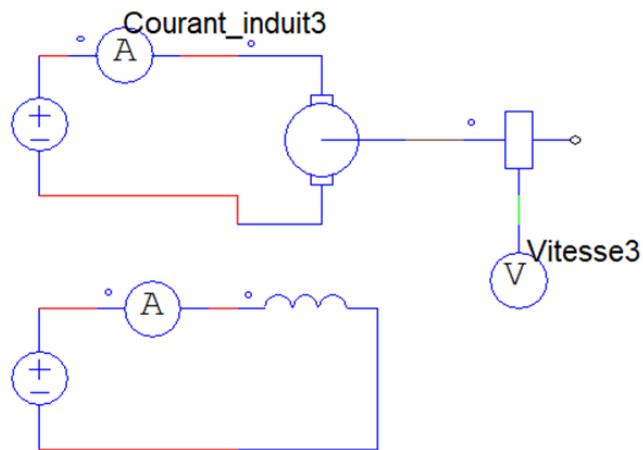


FIGURE 7 – Moteur seul sans frottements ni charge sur PSIM *

5.i.c Comparaison des deux modèles

Afin de nous assurer que notre modèle est le bon, superposons la courbe du courant d'induit du modèle Matlab avec celui du modèle PSIM. Pour ce faire, nous pouvons copier tous les points de la courbe PSIM et les reporter dans un fichier Matlab. Nous allons ensuite la tracer à l'aide de la fonction plot, et la superposer à celle que nous avons obtenu via Simulink. Nous obtenons le graphe suivant en [figure 8](#) :

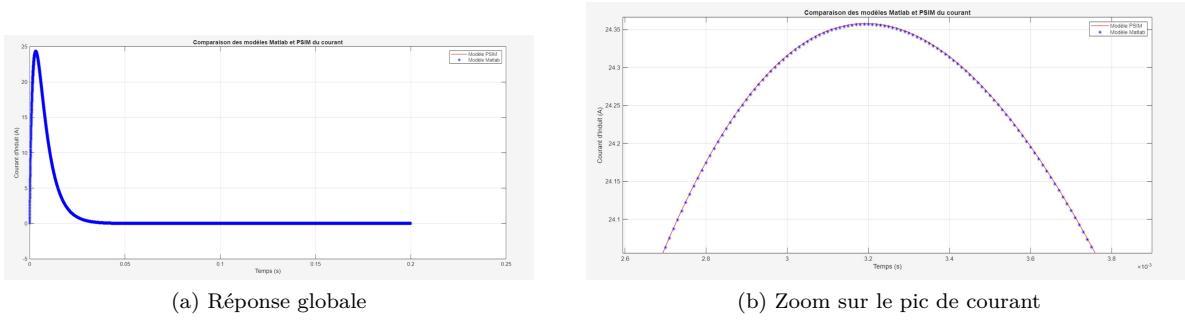


FIGURE 8 – Comparaison Matlab/PSIM moteur seul : courant d'induit *

On a en abscisse le temps en seconde et en ordonnée le courant en ampère.

Faisons de même pour la vitesse :

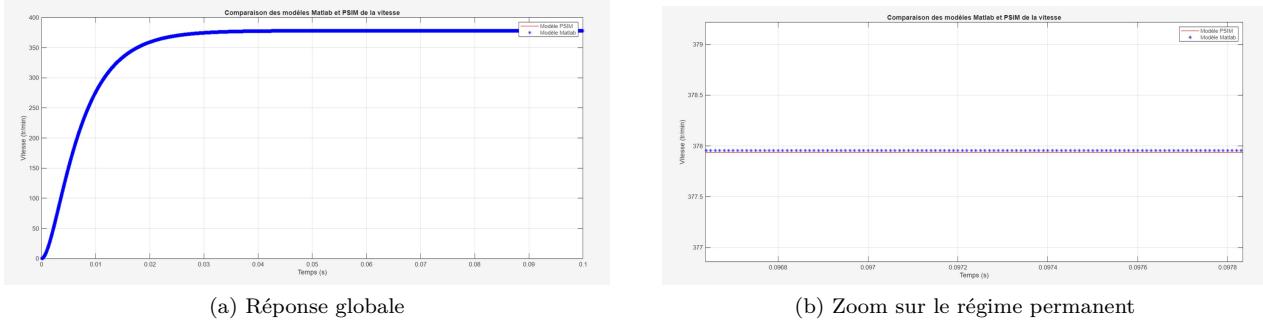


FIGURE 9 – Comparaison Matlab/PSIM moteur seul : vitesse *

On a en abscisse le temps en seconde et en ordonnée la vitesse en $10^1 \cdot \text{tr/min}$.

Nous pouvons observer une très bonne corrélation entre les deux simulations, la différence observée étant vraiment minime par rapport aux grandeurs en jeux. Ceci confirme donc que notre modèle est bon, car la simulation sur deux logiciels distincts nous donne le même résultat.

5.ii Prise en compte des frottements et de la charge

5.ii.a Notions théoriques

Prenons maintenant en compte la charge et les frottements . On établit les relations en fonction des frottements et de la charge :

- **Frottements :**

A partir des données de la machine, on connaît les valeurs de frottement sec et de coefficient de viscosité qui nous permettront de simuler les frottements. Les valeurs de frottement doivent être doublées car il faut compter les frottements des deux machines accouplées. On a donc l'équation suivante :

$$C_f = 2 \cdot C_0 + 2 \cdot f\omega$$

Avec C_f , le couple de frottement, C_0 , le couple de frottement sec (cf. " T_f ", tableau 1) et $f = \frac{K_d \cdot 30}{1000\pi}$ car le coefficient de viscosité est donné pour 1000 tr/min (cf. tableau 1).

- **Charges :**

En prenant en compte la génératrice qu'entraîne le moteur et la charge sur laquelle cette génératrice débite, on obtient le schéma suivant :

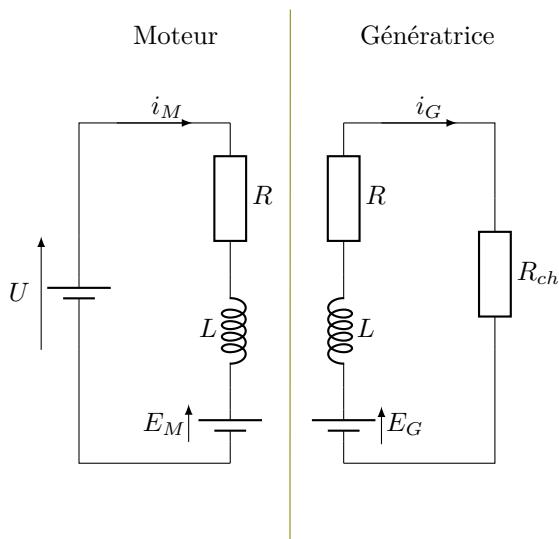


FIGURE 10 – Schéma équivalent du système avec une charge

À partir de la figure 10, de ce qui a été montré pour les frottements, et des équations (1) adaptées à une MCC génératrice, on obtient les relations suivantes :

$$K_\phi \Omega = (R_{ch} + R)i_G + L \frac{di_G}{dt} \quad (2)$$

$$2J \cdot \frac{d\Omega}{dt} = K_\phi(i_M - i_G) - 2 \cdot C_0 - 2 \cdot \Omega f \quad (3)$$

L'équation 2 nous permet d'établir la fonction de transfert reliant le courant du générateur à la vitesse. L'équation 3 nous permet de réaliser la "boucle de rétroaction" des frottements et de la charge. Ces dernières nous seront utiles pour implémenter ces éléments à la simulation sur Matlab.

5.ii.b Implémentation sous PSIM

On ajoute d'abord les frottements sur PSIM. Le bloc "Mechanical load" (charge mécanique) nous permet de simuler les frottements entre le moteur et le générateur. On y rentre les valeurs de frottement sec et de coefficient de viscosité.

Pour la charge, on ajoute simplement une résistance choisie de manière arbitraire à 10Ω dans le circuit induit du générateur comme suit en [figure 11](#) :

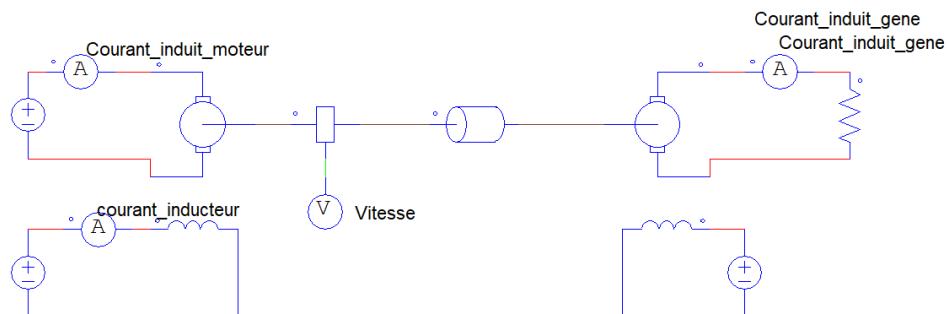


FIGURE 11 – Montage du système muni des éléments de frottement et de la charge simulés sur PSIM *

5.ii.c Implémentation sous Matlab/Simulink

On peut ensuite faire l'équivalent sur Matlab. A l'aide des équations précédemment établies ([2](#) et [3](#)), on ajoute les différents éléments au montage Simulink et leurs valeurs au fichier de données.

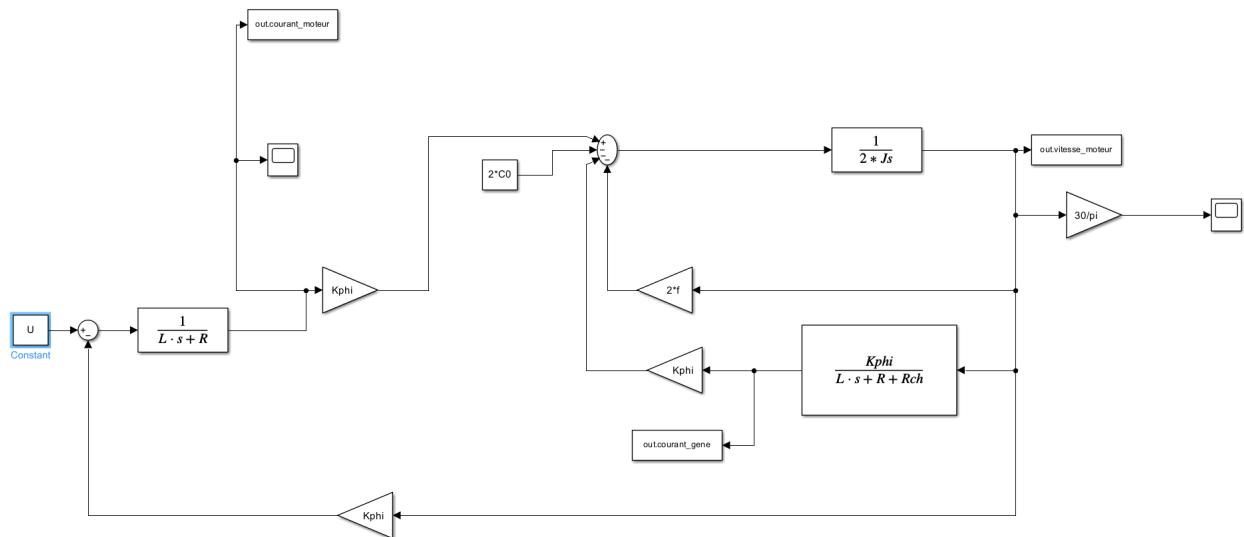


FIGURE 12 – Montage du système muni des éléments de frottement et de la charge simulé sur Matlab *

5.ii.d Comparaison des deux modèles

On relève les valeurs de simulation de PSIM pour les comparer à celles obtenues avec Matlab pour comparer les montages et s'assurer de la validité de nos simulations.

On applique la même méthode que dans la partie précédente.

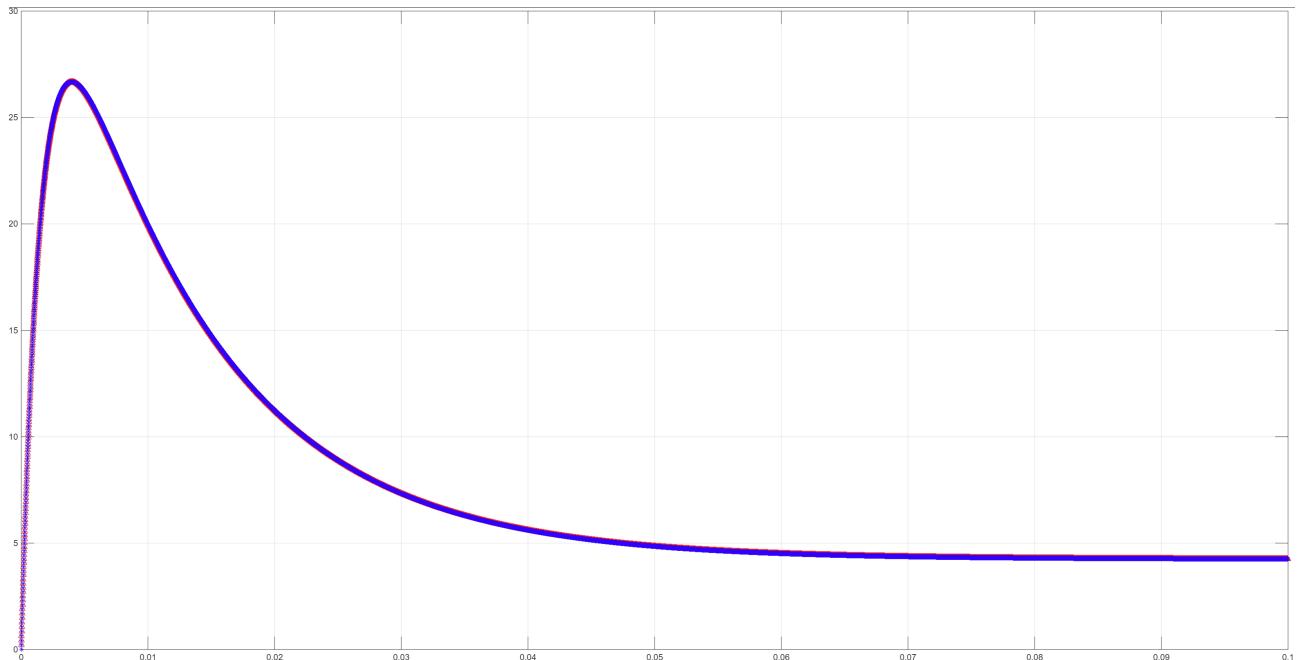


FIGURE 13 – Comparaison des courants induit simulés sur Matlab en bleu et sur PSIM en rouge *

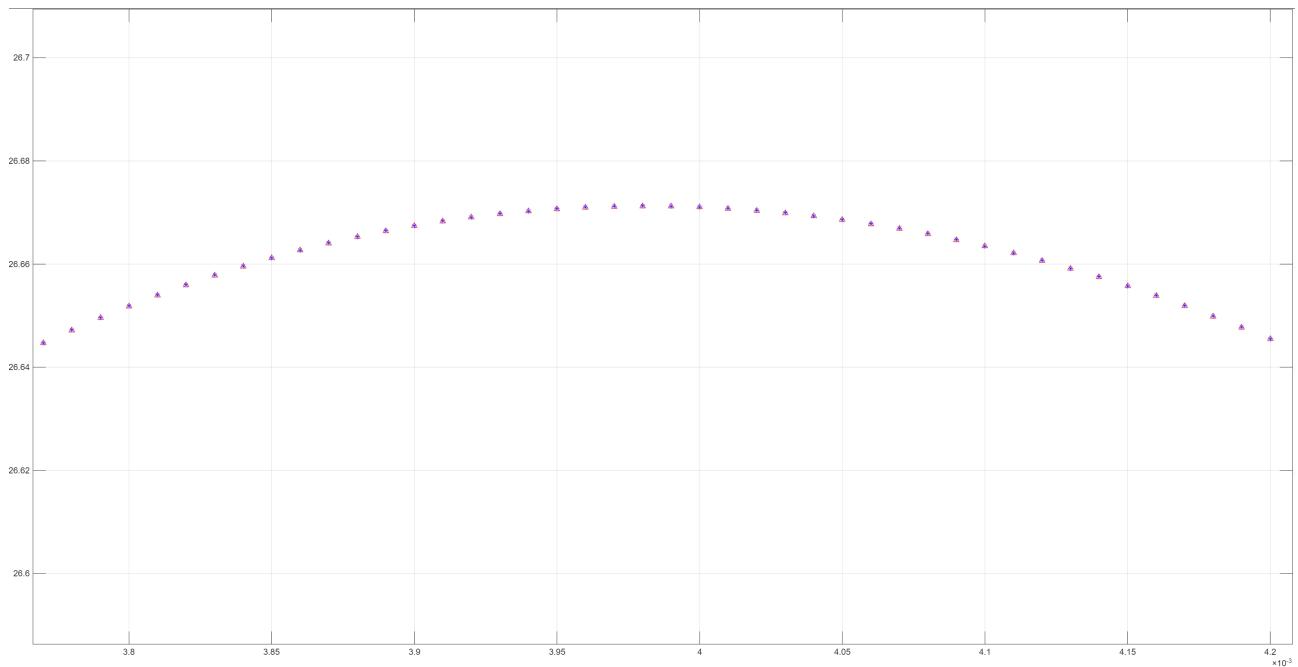


FIGURE 14 – Zoom sur le pic de courant *

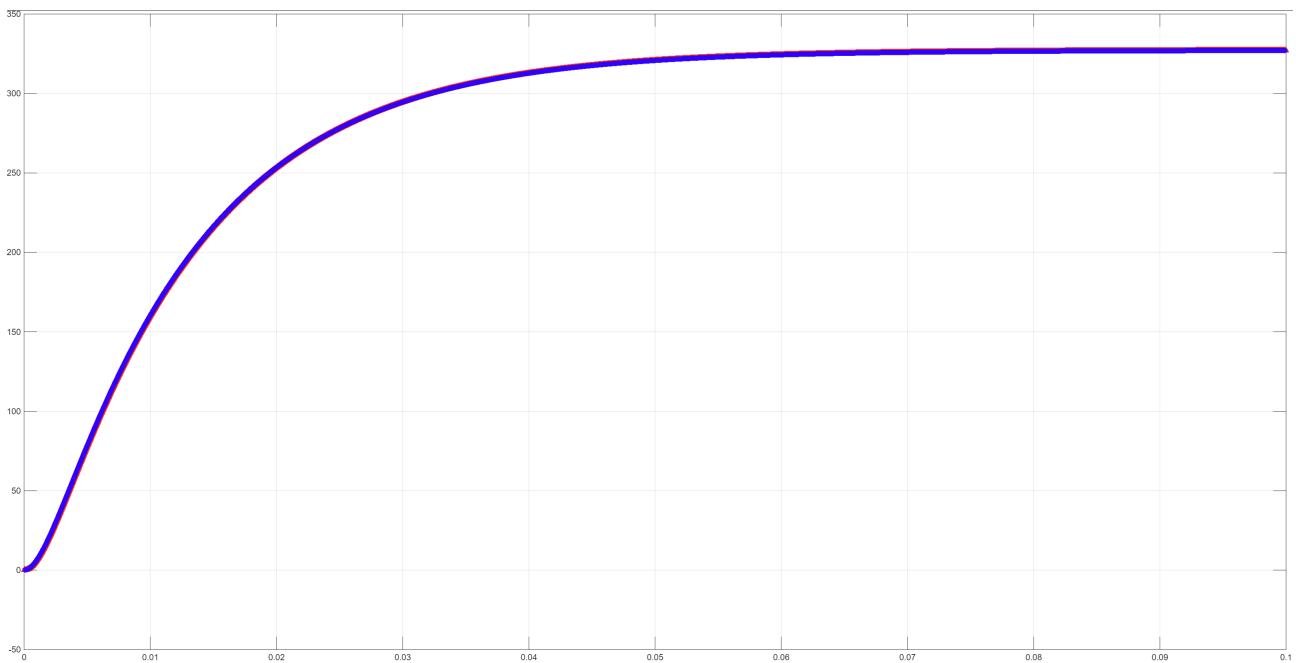


FIGURE 15 – Comparaison des vitesses simulées sur Matlab en bleu et sur PSIM en rouge *

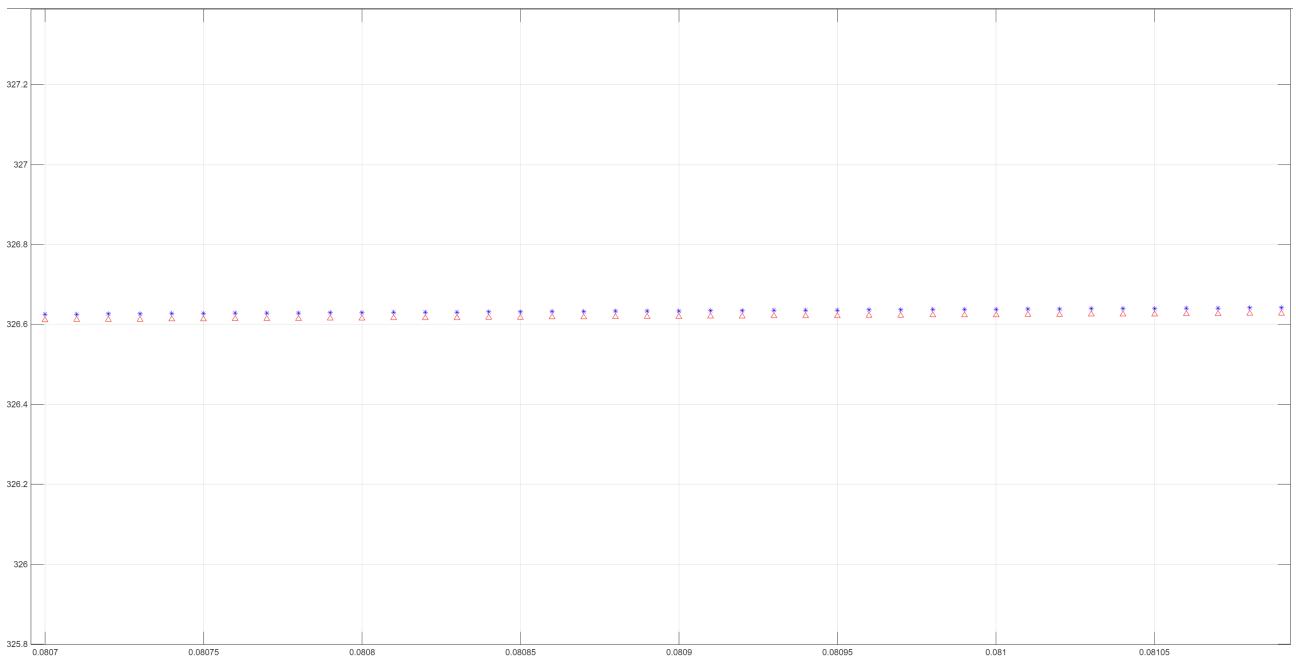


FIGURE 16 – Zoom sur le régime permanent *

Nous pouvons observer encore une fois une très bonne corrélation entre les deux simulations, la différence observée étant vraiment minime par rapport aux grandeurs en jeu. Ceci confirme donc que notre modèle est correct avec l'ajout de ces nouveaux éléments.

5.iii Etude du régime permanent

En amont de la réalisation de l'asservissement de notre système, nous allons réaliser l'étude du régime permanent de celui-ci. Pour cela, nous allons utiliser un outil de modélisation sous forme de systèmes d'états. On définit le vecteur d'état X composé de 3 variables d'état : le courant d'induit moteur i_m , le courant d'induit génératrice i_g et la vitesse du moteur Ω . Le vecteur peut ainsi se noter ainsi :

$$X = \begin{pmatrix} i_m \\ i_g \\ \Omega \end{pmatrix}$$

L'étude du régime permanent est donc équivalente à l'étude de l'évolution de ce vecteur lorsque $t \rightarrow \infty$. En pratique, le régime permanent est atteint lorsque t est largement plus élevé que les constantes de temps du système. A partir des équations de moteur chargé par la génératrice, on peut écrire les relations entre les différentes variables d'états :

$$\begin{cases} U = Ri_m + L \frac{di_m}{dt} + K_\phi \Omega \\ K_\phi \Omega = (R_{ch} + R)i_g + L \frac{di_g}{dt} \\ 2J \frac{d\Omega}{dt} = K_\phi(i_m - i_g) - 2C_0 - 2f\Omega \end{cases} \Leftrightarrow \begin{cases} \frac{di_m}{dt} = \frac{U}{L} - \frac{R}{L}i_m - \frac{K_\phi}{L}\Omega \\ \frac{di_g}{dt} = \frac{K_\phi}{L}\Omega - \frac{(R+R_{ch})}{L}i_g \\ \frac{d\Omega}{dt} = \frac{K_\phi}{2J}i_m - \frac{K_\phi}{2J}i_g - \frac{C_0}{J} - \frac{f}{J} \end{cases}$$

Soit sous forme matricielle :

$$\frac{dX}{dt} = \begin{pmatrix} -\frac{R}{L} & 0 & -\frac{K_\phi}{L} \\ 0 & -\frac{(R+R_{ch})}{L} & \frac{K_\phi}{L} \\ \frac{K_\phi}{2J} & -\frac{K_\phi}{2J} & -\frac{f}{J} \end{pmatrix} \begin{pmatrix} i_m \\ i_g \\ \Omega \end{pmatrix} + \begin{pmatrix} \frac{U}{L} \\ 0 \\ -\frac{C_0}{J} \end{pmatrix} = AX + BU$$

Les constantes de temps du système τ_i sont liées aux valeurs propres λ_i de la matrice A d'après la formule : $\tau_i = \frac{-1}{\lambda_i}$. Pour calculer ces dernières, on utilisera Matlab. On entre la matrice A sur le logiciel et on calcule les constantes de temps avec la commande : `cstesdetemps=-1./eig(A)`. On obtient alors les valeurs suivantes :

$$\begin{cases} \tau_1 = 0,0016 \text{ s} \\ \tau_2 = 0,0122 \text{ s} \\ \tau_3 = 0,0002 \text{ s} \end{cases}$$

Ainsi, dans nos simulations, si nous voulons observer avec certitude le régime permanent nous tacherons de prendre ce qui vient d'être énoncer en compte.

On cherche finalement à trouver les valeurs finales des 3 variables d'état, c'est-à-dire leur valeur en régime permanent. On note X_∞ le vecteur contenant ces valeurs finales. Cela revient à résoudre :

$$\frac{dX}{dt} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} \Leftrightarrow MX_\infty + N = 0 \Leftrightarrow X_\infty = -(M)^{-1}N$$

En utilisant Matlab à nouveau, on obtient :

$$X_\infty = \begin{pmatrix} 4,2452 \text{ A} \\ 3,6065 \text{ A} \\ 327,144515 \text{ rad/s} \approx 3124 \text{ tr/min} \end{pmatrix}$$

5.iv Modélisation du hacheur

5.iv.a Notions théoriques

Jusqu'à présent, nous avons utilisé dans nos modèles une source de tension continue pour alimenter notre MCC. Hors, dans les faits, c'est bien un hacheur qui alimentera notre moteur. Ainsi, nous allons modéliser plus finement notre système en modélisant le comportement du hacheur. Nous effectuerons cela sur PSIM.

Nous savons que le hacheur utilisé est un hacheur en pont fonctionnant en mode bipolaire dont le schéma de principe est donné à la [figure 17](#).

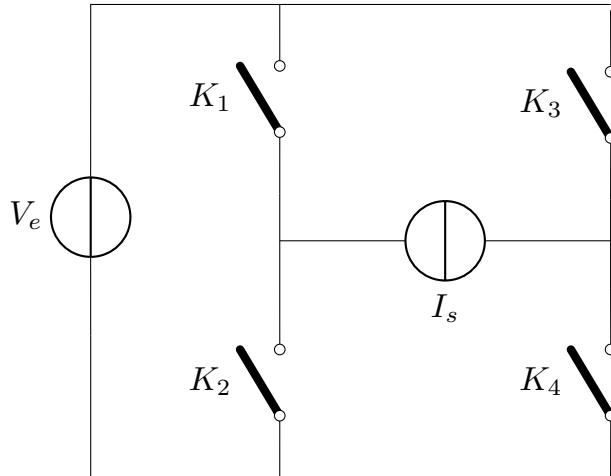


FIGURE 17 – Schéma de principe d'un hacheur en pont

V_e sera la tension en entrée de notre hacheur, c'est-à-dire 48V, la source de courant sera notre moteur et les interrupteurs seront des transistors MOSFET. Pour obtenir le comportement voulu, les transistors sont commandés par un signal MLI (modulation largeur d'impulsion) ayant la forme présentée en [figure 18](#) :

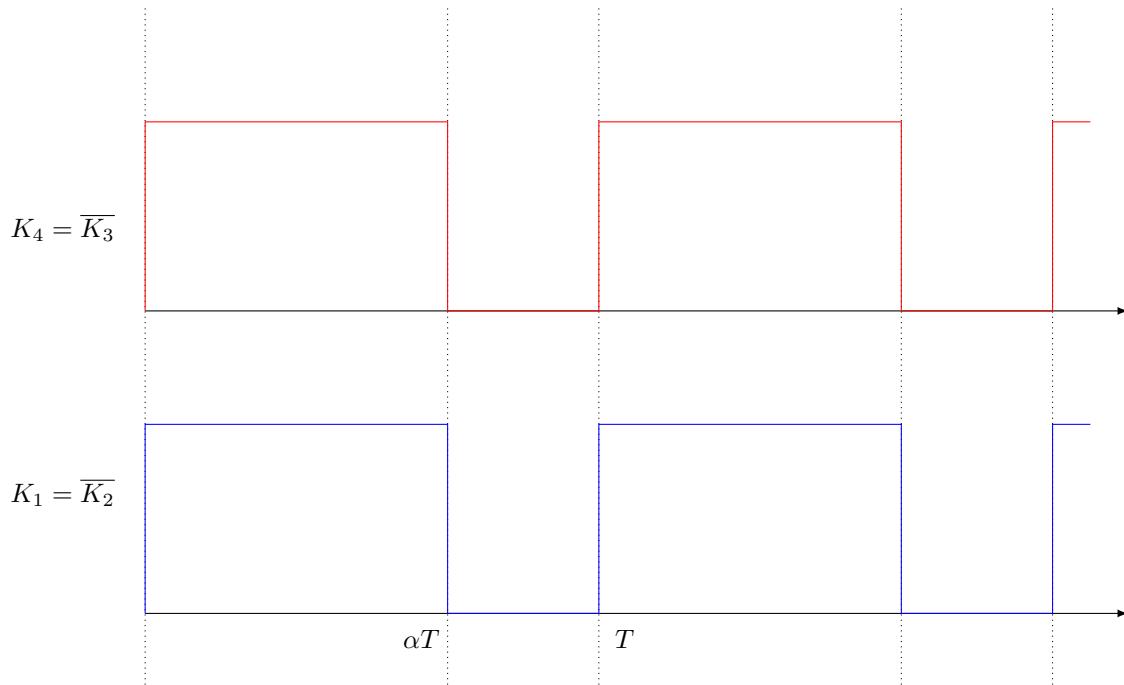


FIGURE 18 – Signaux de commande MLI du hacheur

5.iv.b Implémentation sous PSIM

On réalise donc le schéma du hacheur ayant la structure précédemment mentionnée et permettant d'obtenir les signaux de commande souhaités sur PSIM. Ce dernier est présenté en [figure 19](#) :

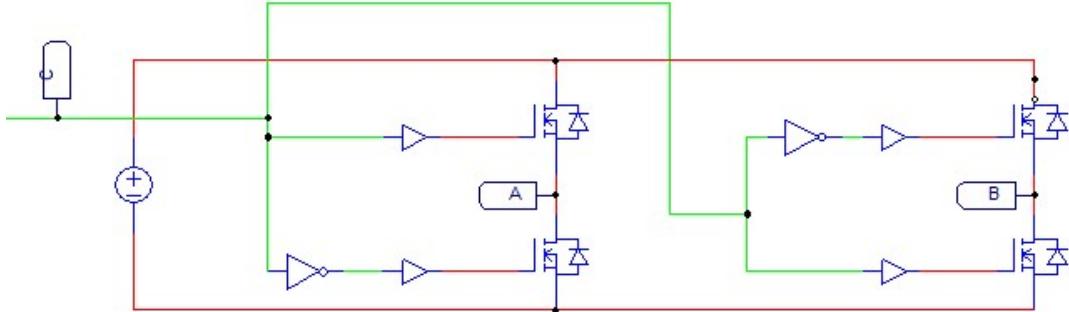


FIGURE 19 – Implémentation du hacheur sur PSIM *

Pour l'instant, le signal de commande des MOSFETS 'C' est un signal carré alternant entre 0 et 10 V avec un rapport cyclique de 75%. Nous prendrons en compte la commande du système dans la partie suivante.

5.iv.c Implémentation sous Matlab/Simulink

Pour la modélisation sur Matlab, nous allons simplement utiliser la valeur moyenne de la tension en sortie du hacheur, ce que nous avons déjà fait depuis le début des simulations.

5.v Prise en compte de la commande

5.v.a Notions théoriques

Comme évoqué précédemment, on a cherché à prendre en compte l'alimentation U_{AB} du moteur issue du hacheur. Pour sa modélisation sur Matlab, nous avons utilisé sa valeur moyenne. En pratique, comme nous l'avons vu, la tension issue du hacheur est un signal carré compris entre -48V et 48V et dont la valeur moyenne va ajuster la vitesse de rotation du moteur. Le hacheur est piloté par un signal PMW (Pulse Width Modulation) ou en français, MLI (Modulation de Largeur d'Impulsion) dont on va contrôler le rapport cyclique qui permettra donc d'ajuster la valeur moyenne en sortie. Pour créer un signal MLI , on utilisera un comparateur : celui-ci va comparer une tension de commande DC $u_{commande}$ et un signal triangulaire. Ainsi, lorsque la tension de commande est supérieure au signal triangle, la tension de sortie du comparateur prend une valeur haute sinon elle vaut 0. Visuellement, on a la forme d'ondes suivantes :

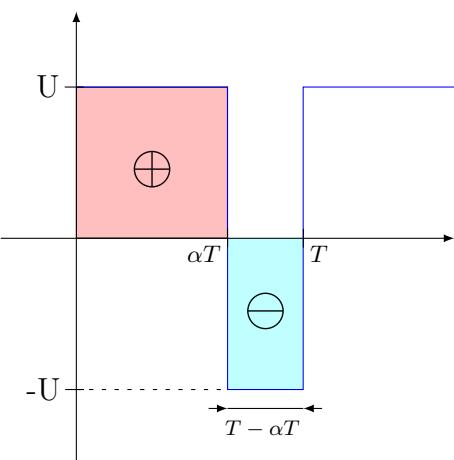


FIGURE 20 – Signal d'entrée de la MCC

Grâce à la méthode des aires, on peut établir la relation entre la tension d'alimentation du hacheur (ici 48 V) et sa tension moyenne de sortie, on a donc :

$$\langle U_{AB} \rangle = \frac{\text{aire} \oplus -\text{aire} \ominus}{T} = \frac{U\alpha T - U(T - \alpha T)}{T} = (2\alpha - 1)U = \left(2\frac{u_{commande}}{U_M} - 1\right)U \quad (4)$$

Nous avons ainsi modélisé le signal fourni par le hacheur par le calcul de sa valeur moyenne. Cela sera particulièrement utile pour les simulations sur Simulink qui ne contient pas de modèle de hacheur intégré.

5.v.b Implémentation sous PSIM

A partir de ce qui a été dit précédemment, on implémente la commande de la manière suivante sur PSIM :

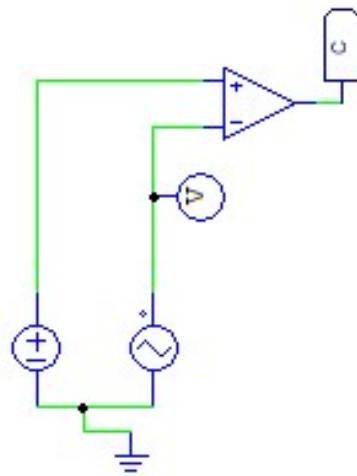


FIGURE 21 – Modélisation de la commande sous PSIM *

5.v.c Implémentation sous Matlab/Simulink

De même, à partir de l'[équation 4](#), on implémente la commande sur Simulink en changeant la constante U précédente par ce qui est montré en [figure 22](#) :

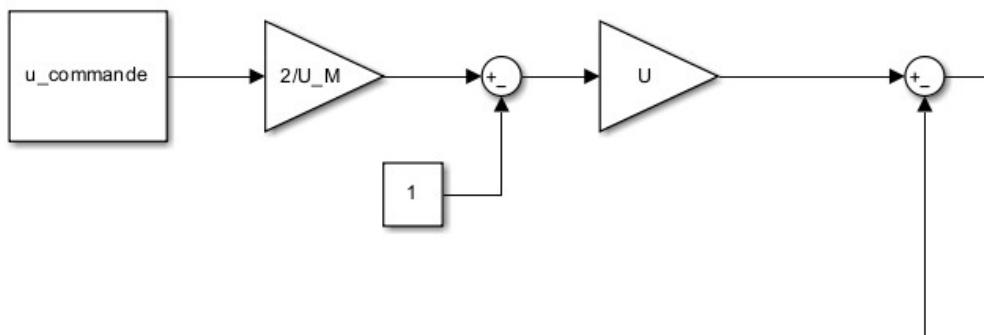


FIGURE 22 – Modélisation de la commande sous Matlab *

5.vi Etude du régime permanent considérant la commande

Nous pouvons à présent vérifier la conformité de la modélisation de la commande sur Simulink puis sur PSIM. Pour cela, nous prendrons pour l'exemple un rapport cyclique $\alpha = 0,75$ avec $u_{commande} = 11,25V$ et $U_M = 15V$.

5.vi.a Sous Matlab/Simulink

On peut recalculer sur Matlab, les valeurs finales de nos variables d'états en remplacement U par la valeur de $< U_{AB} >$ établie en (4) et grâce aux valeurs de l'exemple évoquées précédemment. On obtient alors :

$$X'_{\infty} = \begin{pmatrix} 2,2882A \\ 1,78142A \\ 1543tr/min \end{pmatrix}$$

5.vi.b Sous PSIM

Testons à présent notre modélisation sur PSIM. Pour cela, nous allons reproduire la formule de $< U_{AB} >$ à l'aide de blocs de la manière suivante :

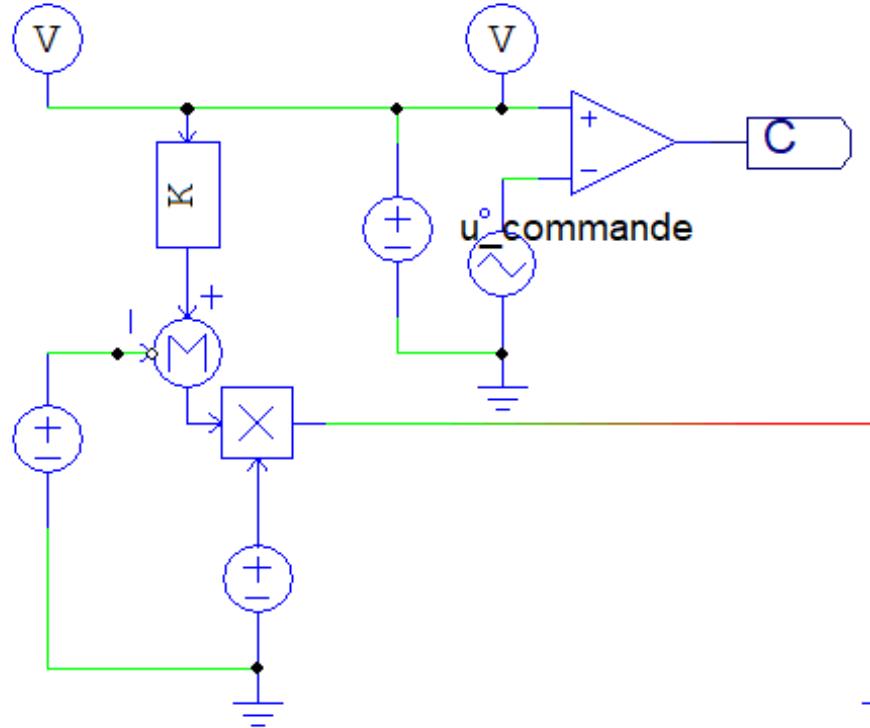


FIGURE 23 – Reproduction de la formule $< U_{AB} >$ sous PSIM *

On observe ici la création de 2 signaux : le premier (label C) est le signal MLI évoqué plus tôt qui va piloter le hacheur et en second, on crée le signal $< U_{AB} > = (2 \frac{u_{commande}}{U_M} - 1)U$ à partir de $u_{commande}$, ici une source de tension continue. Nous pouvons ainsi comparer l'évolution de nos variables d'états avec 2 entrées différentes : l'une est le signal directement fourni par le hacheur et l'autre est le signal de sortie modélisé.

En [figure 24](#), [figure 25](#) et [figure 26](#) sont présentées, l'allure du courant d'induit et de la vitesse avec ces deux signaux :

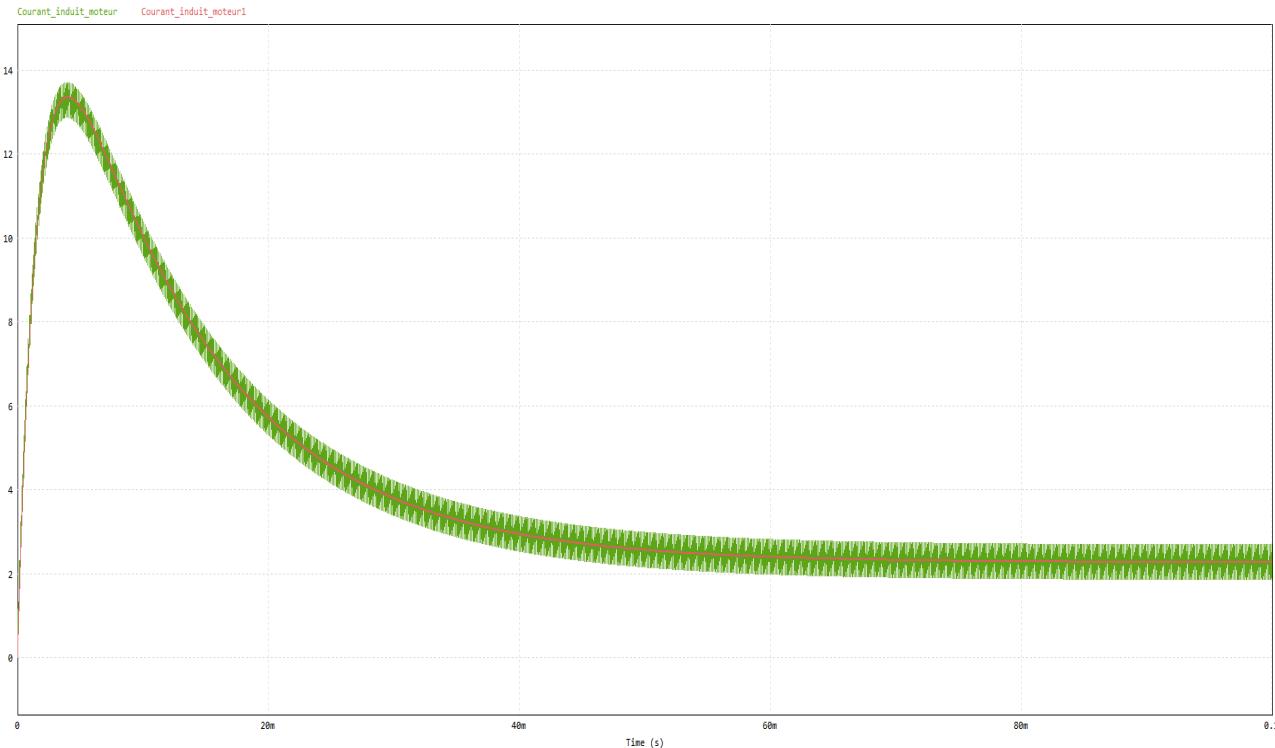


FIGURE 24 – Courant d'induit moteur avec hacheur (vert) et courant d'induit moteur avec modélisation du hacheur (rouge) *

On a en abscisse le temps en seconde et en ordonnée le courant en ampère.

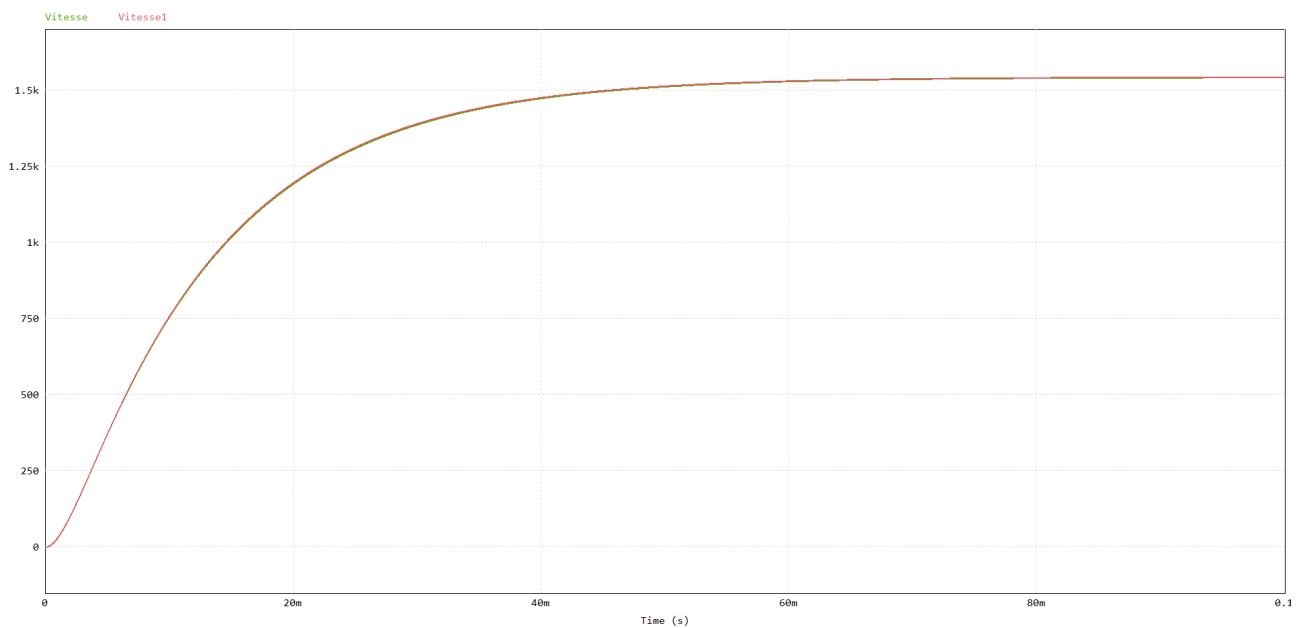


FIGURE 25 – Vitesse moteur avec hacheur (vert) et vitesse moteur avec modélisation du hacheur (rouge) *

On a en abscisse le temps en seconde et en ordonnée la vitesse en tr/min.

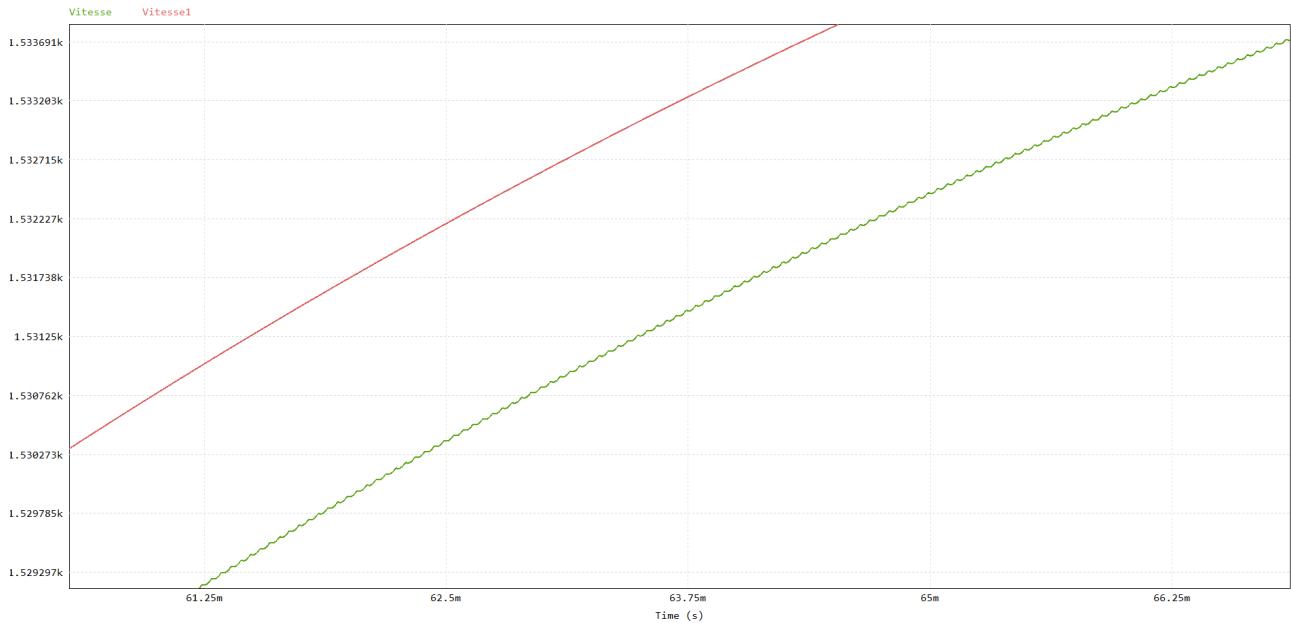


FIGURE 26 – Zoom sur l'ondulation de vitesse *

On a en abscisse le temps en seconde et en ordonnée la vitesse en tr/min.

On constate plusieurs choses :

- Tout d'abord avec le signal directement fourni par le hacheur, on constate une ondulation sur les deux courbes. Celle-ci est dû au signal du hacheur qui n'est pas continu mais carré.
- Malgré cette ondulation, on constate que l'évolution des grandeurs est la même que celle avec la modélisation du hacheur, ce qui confirme la bonne conception de notre hacheur sur PSIM. De plus, la courbe avec modélisation du hacheur est pratiquement la même que celle que nous obtenons sur Matlab avec la même modélisation de la commande. Cela conforte encore notre modèle.
- Enfin, les valeurs des grandeurs en régime permanent obtenus avec ces courbes sont cohérentes avec celles calculées via Matlab précédemment.

On voit que PSIM nous permet donc de simuler plus finement le système en prenant en compte les ondulations réelles du système.

6 Dimensionnement et simulation des asservissements

6.i Préliminaire à l'asservissement

Avant de commencer le dimensionnement de nos asservissements, nous devons légèrement modifier notre modèle. En effet, pour réaliser nos asservissements, on va considérer que l'on se situe autour d'un point d'équilibre et que nous allons linéariser le système autour de ce point d'équilibre. Ainsi, nous ne prendrons pas en compte les éléments invariants à une perturbation δ sur notre système. De ce fait, nous négligerons dans notre asservissement les frottements secs $2C_0$ et prendrons en compte uniquement le gain du hacheur à une perturbation, soit $\frac{2U}{U_M} = 6,4$. On ajustera donc ces éléments dans nos modèles PSIM et Matlab. On utilisera aussi un limiteur de tension pour limiter la tension de commande afin qu'elle ne sorte pas de sa plage de fonctionnement normale.

6.ii Asservissement en courant

6.ii.a Calcul du correcteur

Pour simplifier la réalisation de l'asservissement, nous allons regrouper les différentes parties qui constituent notre schéma bloc global et établir la fonction de transfert en boucle ouverte du courant. On a le schéma bloc suivant :

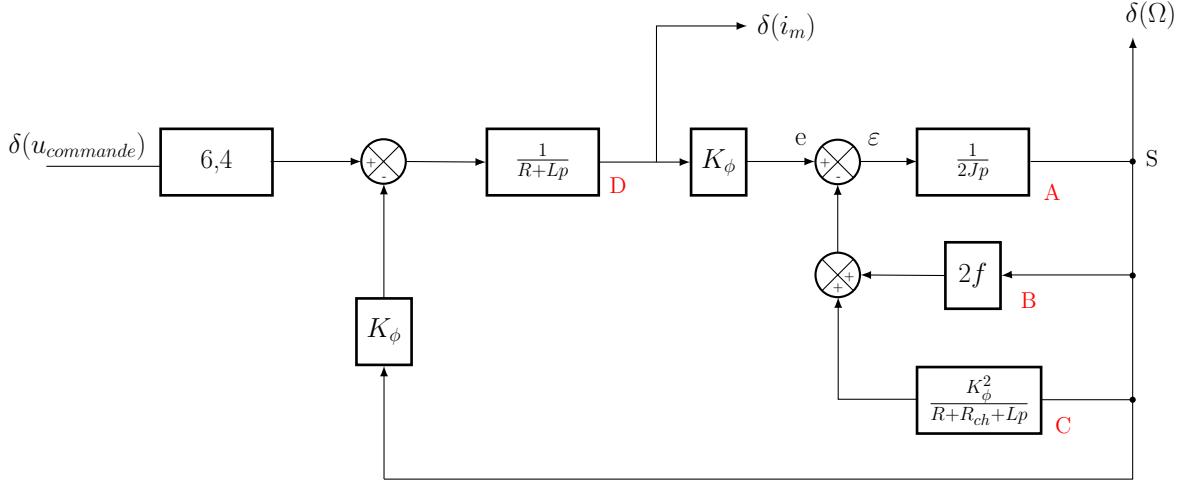


FIGURE 27 – Schéma-bloc du système global

A partir de cette structure, nous pouvons établir la fonction de transfert souhaitée. Nous allons dans un premier temps établir une fonction de transfert H qui sera le rapport entre la vitesse Ω et le courant i_m . Nous pouvons écrire les équations suivantes :

$$\begin{cases} \varepsilon = e - B \cdot S - C \cdot S \\ S = A \cdot \varepsilon = A(e - (B + C)S) \end{cases} \quad (5)$$

On en tire l'expression suivante :

$$S = \frac{A \cdot e}{1 + A(B + C)}$$

En nommant $H = K_\phi \cdot \frac{A}{1 + A(B + C)}$, on a finalement :

$$H = \frac{\Omega(p)}{i_m(p)} = K_\phi \cdot \frac{A}{1 + A(B + C)}$$

Nous pouvons alors simplifier le schéma bloc de la manière suivante :

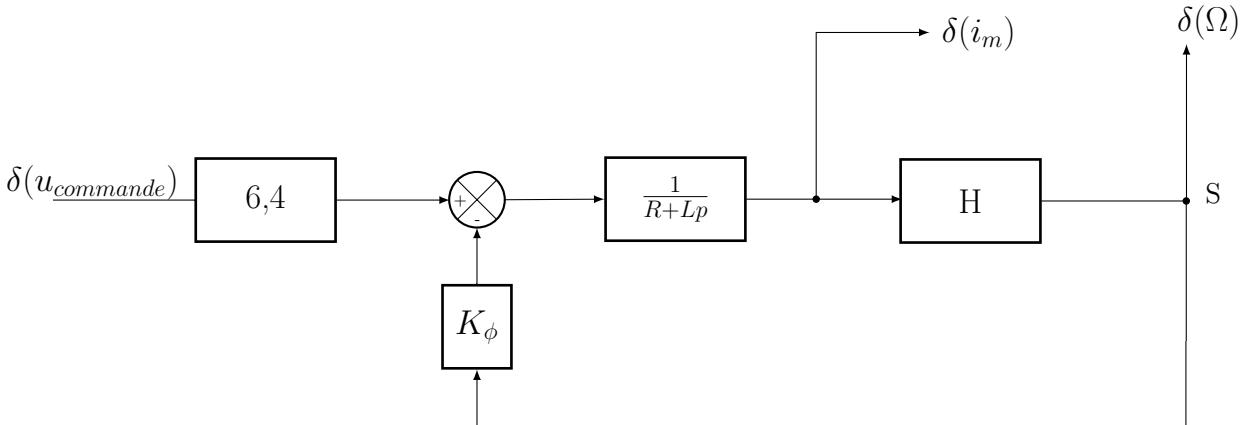


FIGURE 28 – Schéma-bloc simplifié du système global

Nous pouvons finalement établir l'expression du courant d'induit moteur en fonction de la tension de commande :

$$\delta(i_m) = 6,4 \cdot \frac{1}{\frac{R + Lp}{K_\phi^2 \cdot H} + \frac{1}{R + Lp}} \cdot \delta(u_{commande}) \quad (6)$$

A présent, il convient de comparer l'exactitude de notre modélisation du courant sur Simulink. Nous avons le schéma-bloc suivant sur Simulink :

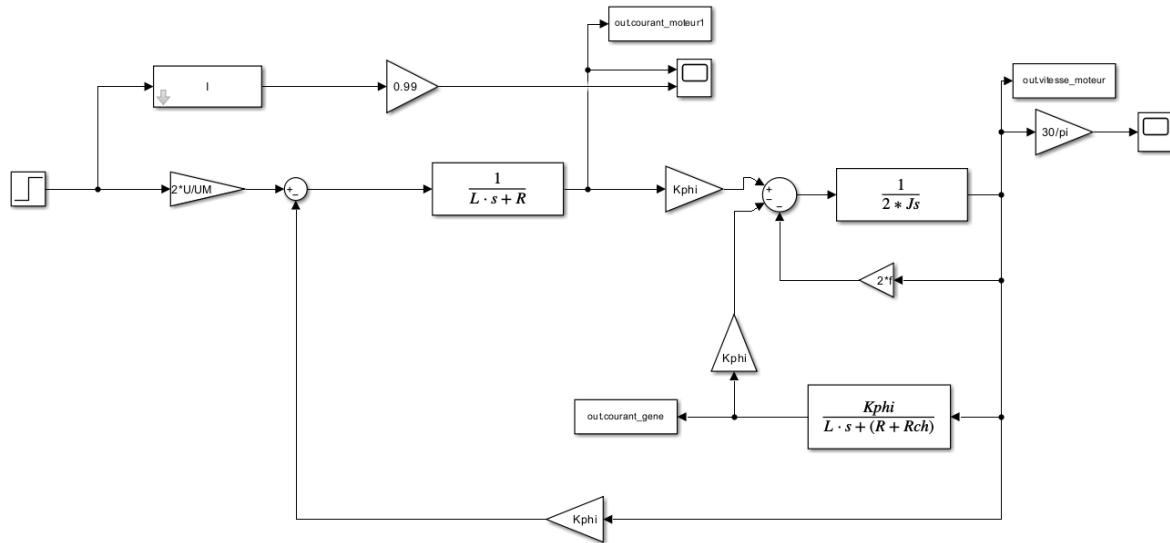


FIGURE 29 – Schéma-bloc Simulink *

La partie basse correspond au schéma-bloc complet du système et au-dessus, le bloc I correspond à la fonction de transfert établie plus haut sur l'équation 6. Le facteur 0,99 permettra de mieux visualiser les deux courbes sur le scope. On réalise un essai indiciel :

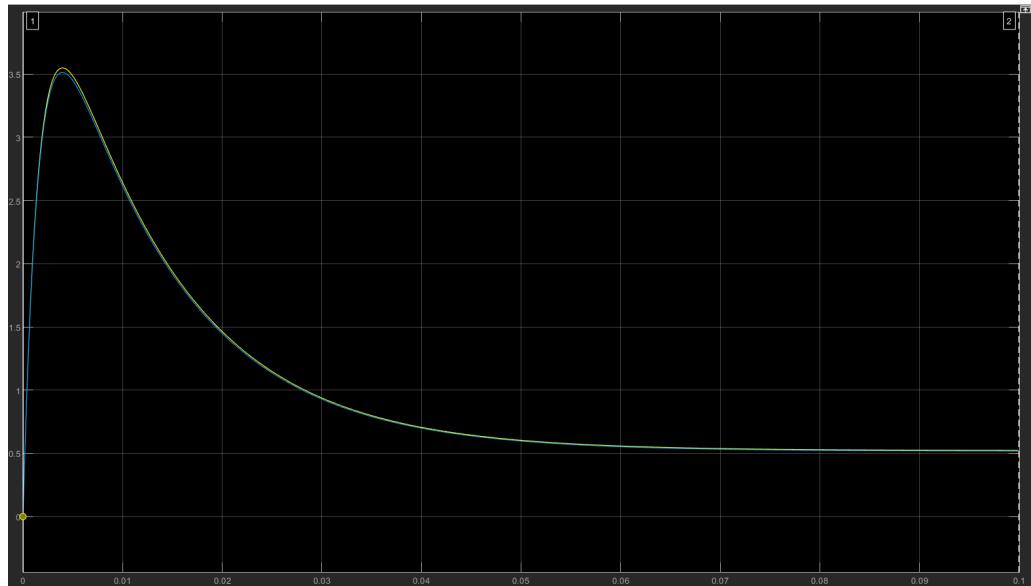


FIGURE 30 – Comparaison du courant d'induit issu du schéma global et du modèle *

On observe une correspondance exacte entre les deux courbes ce qui témoigne de la bonne modélisation du courant d'induit à partir de la fonction de transfert en boucle ouverte.

Le modèle maintenant établi et validé, nous pouvons procéder à son asservissement. Nous avons le schéma bloc suivant :

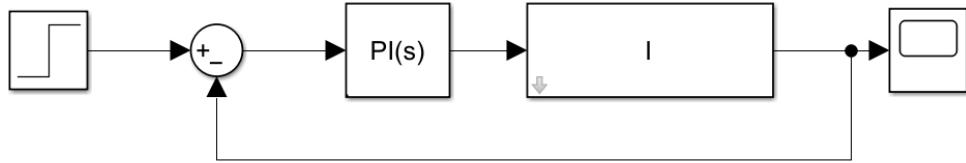


FIGURE 31 – Boucle fermée de courant *

Grâce à la modélisation effectuée précédemment, ce schéma est très simplifié : le correcteur va nous permettre d'ajuster la réponse du courant selon notre cahier des charges. Notre fonction de transfert est en pratique trop complexe pour être étudiée à la main, nous allons donc utiliser l'outil *PID Tuner* de Simulink qui va calculer automatiquement un correcteur selon nos préférences. On rappelle que l'on souhaite un temps de réponse à 5 % égal à au moins 10 fois la période du signal MLI qui pilote le hacheur. La fréquence de ce signal est fixée à $f_{MLI} = 20kHz$ d'où $T_{MLI} = \frac{1}{f_{MLI}} = 5 \times 10^{-5}s$. Nous devons donc fixer $t_{r5\%} = 10 \times T_{MLI} = 5 \times 10^{-4}s$. On rappelle que l'on devra limiter le dépassement entre 10 % et 20 % de la valeur finale.

Ainsi grâce au *PID Tuner*, on règle la réponse en boucle fermée pour qu'elle soit robuste, c'est-à-dire peu sensible aux perturbations et assez rapide pour correspondre à notre cahier des charges. Nous obtenons les valeurs suivantes :

$$PI(p) = P + I \frac{1}{p}$$

$$P = 6,2186$$

$$I = 41446 s^{-1}$$

La réponse à un essai indiciel est la suivante :

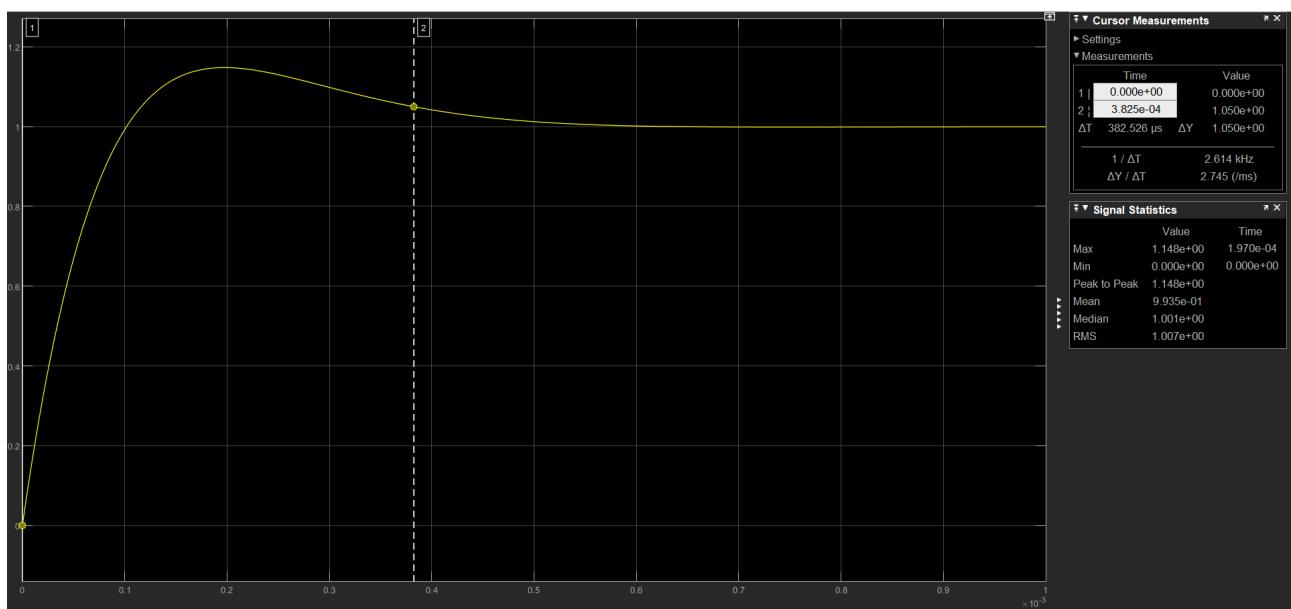


FIGURE 32 – Réponse asservie du courant d'induit *

La réponse correspond bien à nos attentes avec un dépassement maximum de 14,8 % et un temps de réponse effectif de 383 μs , inférieur au temps de réponse maximum imposé de 500 μs .

6.ii.b Vérification du correcteur

Nous allons maintenant vérifier sur PSIM que notre correcteur a bien été dimensionné. On réalise donc l'asservissement sur PSIM avec les mêmes valeurs de correcteur que trouvées précédemment à l'aide du schéma suivant :

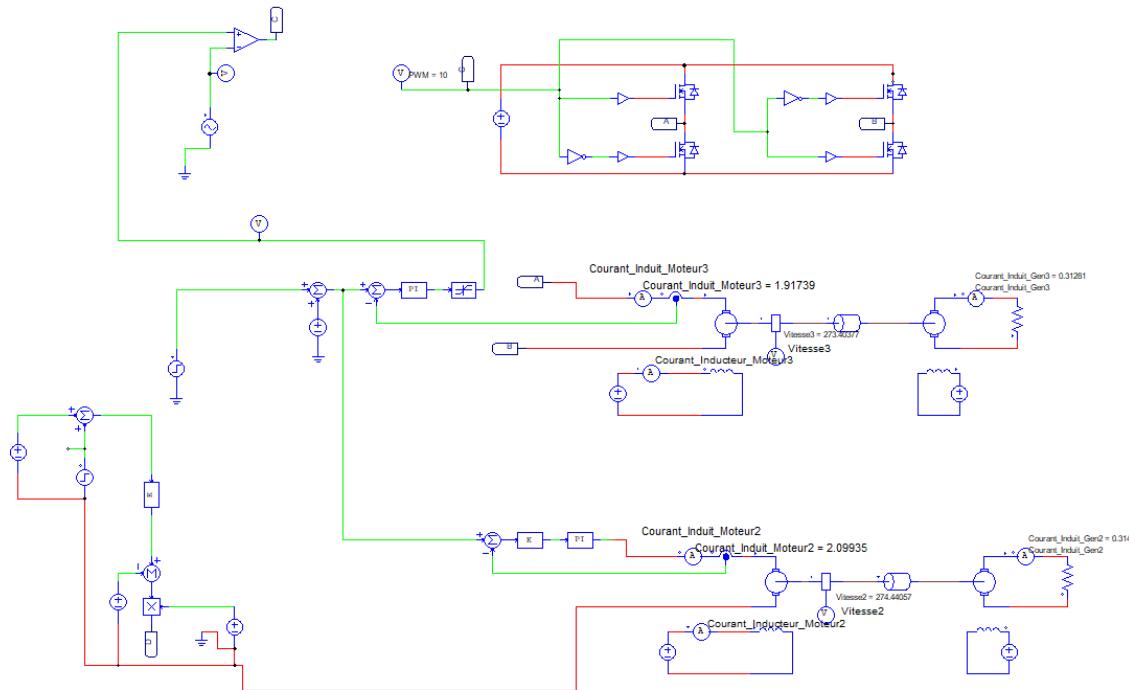


FIGURE 33 – Schéma de vérification de l'asservissement de courant sur PSIM *

Nous avons la partie du bas qui simule le correcteur avec le hacheur modélisé et la partie juste au dessus qui utilise le hacheur directement. L'objectif est de voir ici si la réponse à une perturbation avec le hacheur et le correcteur suit la même évolution que la modélisation avec le même correcteur.

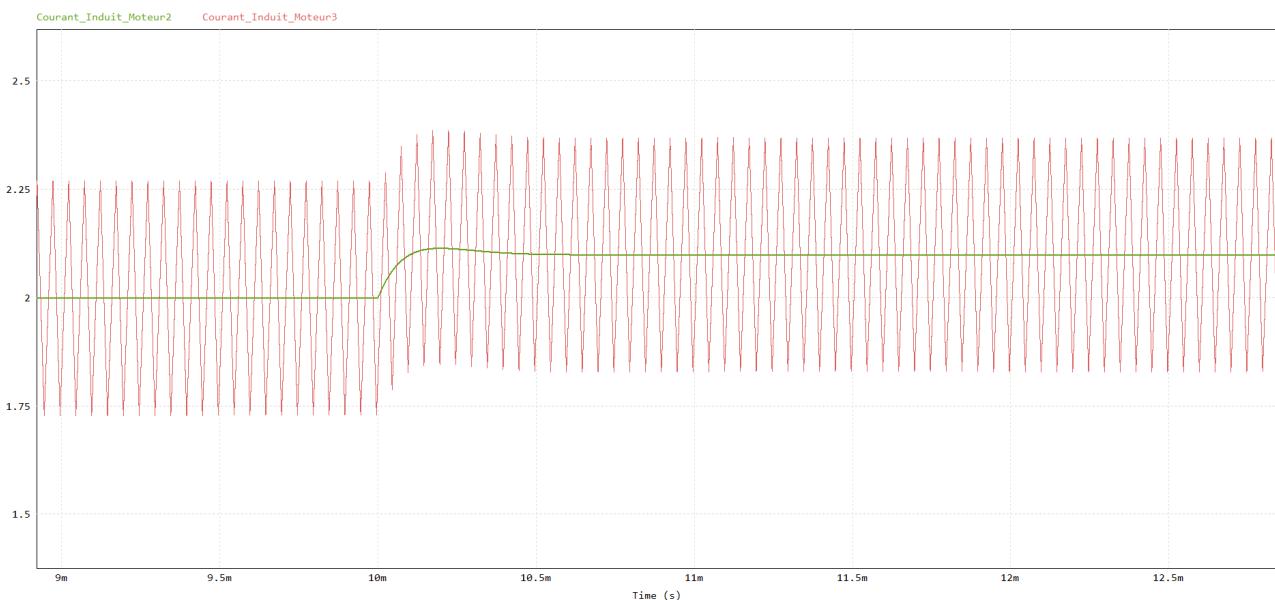


FIGURE 34 – Courbes de vérification de l'asservissement de courant sur PSIM *

En vert, on a la courbe avec le hacheur modélisé et en rouge, la courbe avec le hacheur. Comme nous avons considéré être autour d'un point d'équilibre, il est impossible de comparer les réponses à la mise en route du système, il faut donc comparer les réponse à une perturbation en régime établi comme sur la [figure 34](#). On constate que les réponses suivent la même évolution, ce qui valide le calcul de notre correcteur.

On a bien pris soin de s'assurer que l'asservissement avec la modélisation du hacheur sur PSIM était équivalente à celle sur Matlab (temps de réponse, gain statique, dépassement) avant d'effectuer les comparaisons précédentes.

6.iii Asservissement de vitesse

Pour l'asservissement de vitesse, étant donné que nous utiliserons deux types de capteur qui seront susceptibles de ne pas avoir les mêmes gains, nous expliciterons dans un premier temps la démarche et la réflexion en faisant l'hypothèse que nous avons un gain de capteur unitaire. Par la suite, il nous suffira simplement de prendre en compte les gains (*ou le gain si les capteurs possèdent le même gain*) des capteurs et d'effectuer la même démarche. Ainsi, lors de la prise en compte des capteurs, nous expliciterons uniquement la validité des résultats et non plus la démarche qui sera donc la même que celle présentée en amont.

6.iii.a Démarche globale

6.iii.a.1 Calcul du correcteur

De manière similaire à l'asservissement de courant, nous pouvons déterminer la fonction de transfert en boucle fermée équivalente globale pour la vitesse. Celle-ci se calcule grâce à la formule de Black, non plus au niveau de la sortie de courant mais bien celle de vitesse.

Si nous reprenons les écritures précédentes, elle peut s'écrire :

$$\frac{K_\phi \cdot H \cdot D}{1 + K_\phi \cdot H \cdot D}$$

A l'aide de Matlab, nous pouvons rapidement calculer cette fonction. Il s'agit d'une fonction de transfert du huitième ordre. Lorsque nous simulons notre système en remplaçant notre schéma bloc par cette seule fonction, voici ce que nous obtenons.



FIGURE 35 – Fonction du huitième ordre *

Nous avons rajouté un facteur 0,99 à la fonction de transfert nouvellement calculée afin de pouvoir discerner nos deux courbes. On remarque effectivement que cette fonction est la bonne modélisation de notre système.

Seulement, cette fonction a été calculée à partir de H , une fonction de transfert du troisième ordre : cela peut rendre sa manipulation fastidieuse, particulièrement lors du calcul du correcteur. Pour résoudre ce problème, nous pouvons réaliser une simplification d'ordre physique. La constante de temps de l'inductance de la génératrice étant très rapide, nous pouvons simplement la négliger devant le système complet. Nous pouvons donc simplifier les expressions de certaines fonctions en supposant L nulle.

On peut donc dans la modélisation de la génératrice modifier la fonction que nous avions appelée " C ", celle-ci devient :

$$C = \frac{K_\phi^2}{R + R_{ch} + L\omega} = \frac{K_\phi^2}{R + R_{ch}}$$

On remarque que C est à présent une constante, nous pouvons alors l'ajouter à notre autre constante $B = 2f$. Nous obtenons ainsi une nouvelle constante que nous noterons $f_{eq} = \frac{K_\phi^2}{R + R_{ch}} + 2f$.

Ensuite, nous pouvons calculer la nouvelle fonction de transfert du point de vue de la vitesse. Celle-ci se calcule aisément par la formule de Black suivante :

$$G = \frac{A}{1 + A \cdot f_{eq}} = \frac{\frac{1}{2J_p}}{1 + \frac{1}{2J_p} f_{eq}} = \frac{1}{2J_p + f_{eq}}$$

On se retrouve maintenant avec une fonction de transfert du premier ordre, bien plus simple à manipuler que H . Néanmoins, cette approximation pourrait être trop "simplificatrice", comparons donc nos 3 courbes jusqu'ici obtenues afin de valider notre modèle.

Pour ce faire, calculons la nouvelle fonction de transfert en boucle fermée. En effet, G est la nouvelle expression de la fonction que nous avions appelée H . On reprend alors l'expression que nous avions en remplaçant H par G :

$$\frac{K_\phi \cdot G \cdot D}{1 + K_\phi \cdot G \cdot D}$$

Nous obtenons une fonction de transfert en boucle fermée du quatrième ordre.

Comparons maintenant nos 3 courbes, celle sans avoir simplifié le schéma bloc, celle en ayant simplifié mais en ayant gardé L non nul, et celle avec le L de la génératrice nul. Notons qu'un facteur 0.99 et 1.01 ont été ajoutés aux courbes simplifiées pour une meilleure visibilité.

On obtient :

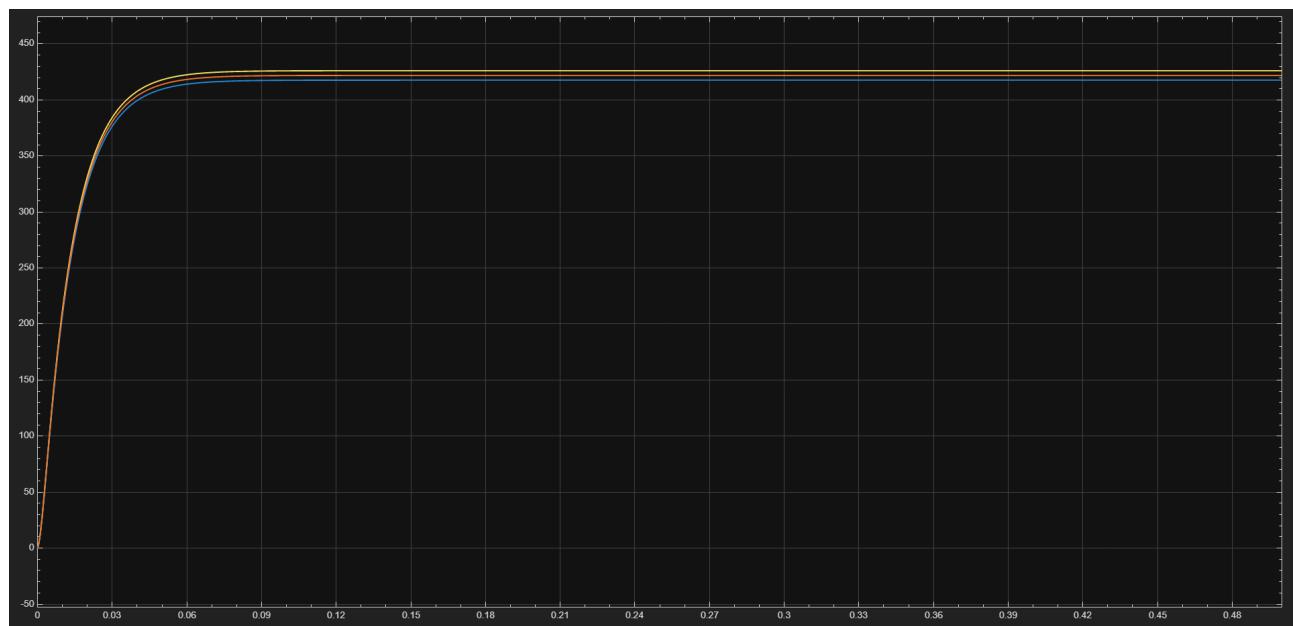


FIGURE 36 – Comparaison des 3 modèles *

Au vue de ces courbes, il va sans dire que notre modèle avec une fonction du premier ordre plutôt que du troisième ordre est parfaitement justifié.

Avant de calculer l'asservissement, il est possible d'effectuer une dernière simplification dans le schéma bloc qui va alléger nos calculs. Nous savons que l'asservissement de vitesse ne peut se faire sans celui du courant, il faudrait alors prendre toute la partie précédente en compte ici pour calculer notre correcteur et nos fonctions de transfert équivalentes.

Cependant, rappelons nous que le temps de réponse de la boucle de courant est de l'ordre de la centaine de µs tandis que celui la vitesse est de l'ordre du dixième de seconde. Ainsi, comme $\tau_{courant} \ll \tau_{vitesse}$, nous pouvons assimiler l'entièreté de la boucle de courant à un gain unitaire, cette dernière n'ayant quasiment aucun impact sur le réponse en vitesse de notre système.

Maintenant que ce modèle a été validé, nous pouvons passer à l'asservissement. L'avantage que nous avons à présent, est que nous pouvons calculer facilement à la main les coefficients de notre correcteur proportionnel-intégral. On a donc le schéma bloc suivant :

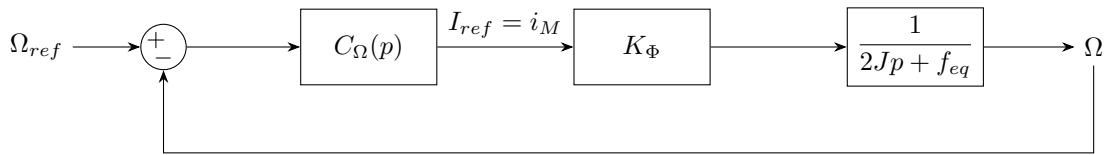


FIGURE 37 – Schéma-bloc de l'asservissement de vitesse final

Rappelons les calculs :

$$F(p) = \frac{C(p)G(p)}{1 + C(p)G(p)} = \frac{K_i \frac{1 + \tau p}{\tau p} \frac{A}{1 + \tau p}}{1 + K_i \frac{1 + \tau p}{\tau p} \frac{A}{1 + \tau p}} = \frac{\frac{K_i A}{\tau p}}{1 + \frac{K_i A}{\tau p}} = \frac{1}{1 + \frac{\tau}{K_i A} p} = \frac{K_{BF}}{1 + \tau_{BF} p}$$

Où $C(p)$ est le correcteur, $T(p)$ est la fonction de transfert de notre système en boucle ouverte et $H(p)$ est la fonction de transfert de notre système asservi en boucle fermée.

On peut réécrire l'expression de G :

$$\frac{1}{2Jp + f_{eq}} = \frac{\frac{1}{f_{eq}}}{1 + \frac{2J}{f_{eq}} p}$$

On en déduit

$$\tau_{BO} = \frac{2J}{f_{eq}}$$

De même on trouve $A = \frac{K_{Phi}}{f_{eq}}$. Il ne faut pas oublier le facteur K_{Phi} dans la formule de Black.

De ça, on calcule ensuite $\tau_{BF} = \frac{\tau_{BO}}{K_i A} = \frac{\frac{2J}{f_{eq}}}{K_i \cdot \frac{1}{f_{eq}} \cdot K_{Phi}}$

Or, le cahier des charges stipule un temps de réponse en boucle fermée 3 fois inférieur à celui en boucle ouverte, on en déduit alors que $K_i \cdot \frac{1}{f_{eq}} \cdot K_{Phi} = 3 \rightarrow K_i = \frac{3f_{eq}}{K_{Phi}} = 0,0354$.

Précisons que lors des calculs, nous avons volontairement choisi $T_i = \tau_{BO}$ afin de simplifier l'expression. Nous savons que cette méthode n'est pas la plus optimale, mais elle fournit un résultat satisfaisant pour ce que nous cherchons. Nous comparerons nos calculs aux résultats trouvés avec le PID Tuner pour s'en assurer.

Nos coefficients de correcteur maintenant calculés, testons notre asservissement sur Matlab. Pour ce faire, de façon analogue à l'asservissement de courant, nous venons créer une simple boucle de rétroaction munie d'un correcteur PI en essai indiciel.

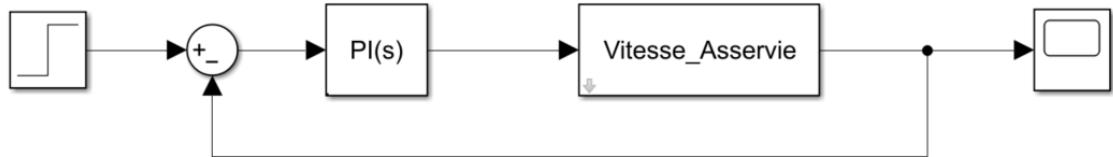


FIGURE 38 – Asservissement simplifié de la vitesse

Ici, dans le PI, nous remplissons $P = K_i = 0,0354$ et $I = \frac{1}{\tau_{BO}} = 27,1 \text{ s}^{-1}$.

Lorsque nous lançons la simulation avec un échelon de 1, nous obtenons la courbe suivante :

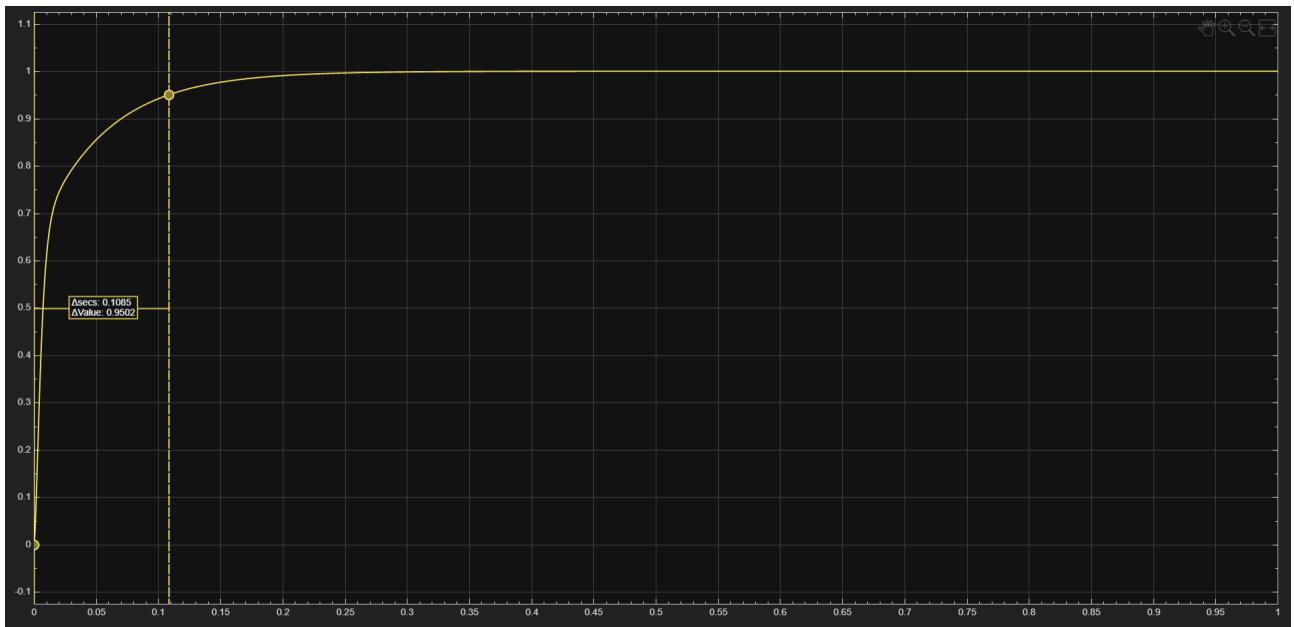


FIGURE 39 – Graphe de la vitesse asservie *

A l'aide des curseurs, nous pouvons déterminer le temps de réponse à 5%, ici nous le mesurons à 0.1085s.

$$\text{Rappelons } tr_{5\%BF} = 3\tau_{BF} = 3 \times \frac{\frac{2J}{f_{eq}}}{K_i \times \frac{1}{f_{eq}} \times K_{Phi}} = \frac{2J}{f_{eq}} = 0.1107s$$

On trouve ce que l'on cherchait, le correcteur PI est donc bien dimensionné.

Avec le PID Tuner, nous trouvons :

$$K_i = 0,0110$$

$$T_i = 0,0142s$$

6.iii.a.2 Vérification du correcteur

Comme précédemment pour le courant, nous allons vérifier notre asservissement sur PSIM. Étant donné que pour calculer notre correcteur de vitesse, nous avons fait l'hypothèse que le courant s'établissait instantanément, il ne faut donc pas oublier de tester l'asservissement de vitesse **avec** l'asservissement de courant. Nous aurons donc ici notre système complet asservi. Voici le schéma de ce dernier sur la figure suivante :

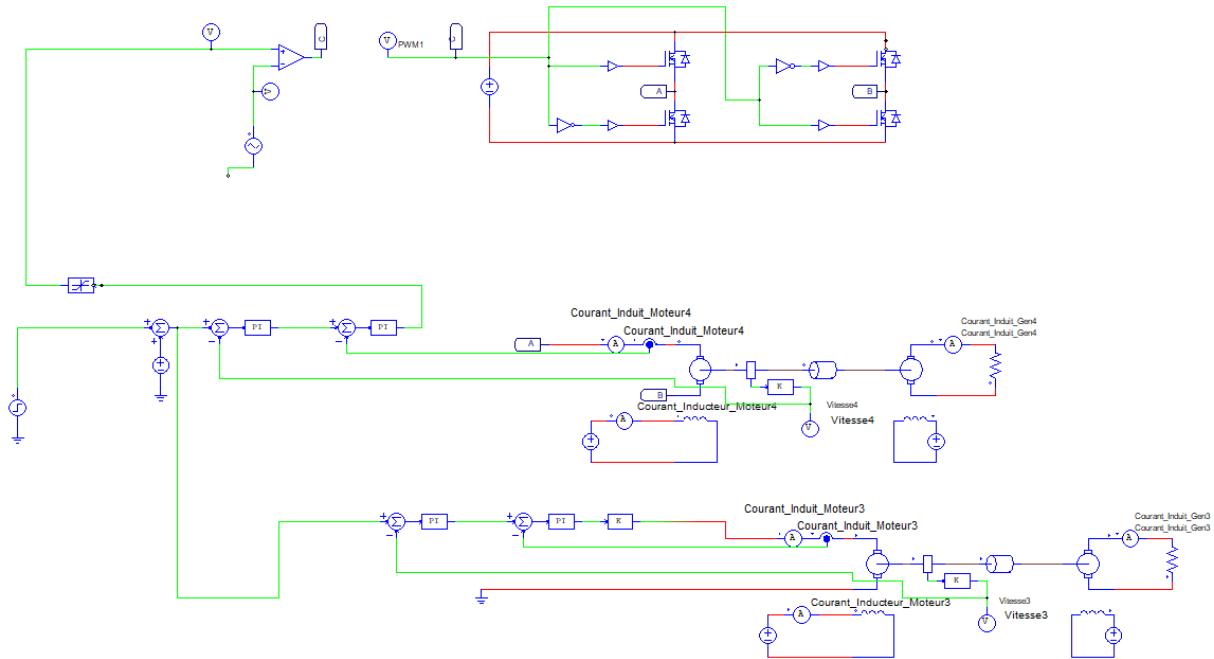


FIGURE 40 – Vérification de l'asservissement de vitesse : Schéma PSIM *

Nous avons la partie du bas qui simule le correcteur avec le hacheur modélisé et la partie juste au dessus qui utilise le hacheur directement. Le premier correcteur à gauche est celui de vitesse et le deuxième à droite, celui de courant. Comme précédemment, l'objectif est de voir ici si la réponse à une perturbation avec le hacheur et le correcteur suit la même évolution que la modélisation avec le même correcteur.

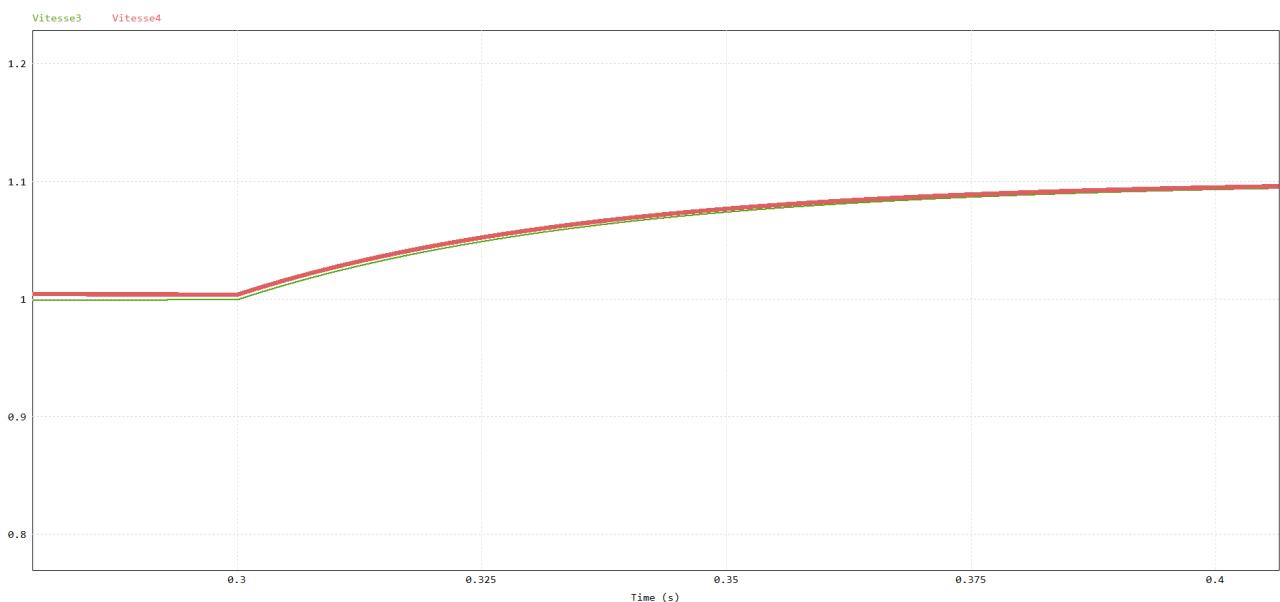


FIGURE 41 – Courbes de vérification de l'asservissement de vitesse sur PSIM *

En vert, on a la courbe avec le hacheur modélisé et en rouge, la courbe avec le hacheur. Comme nous avons considéré être autour d'un point d'équilibre, il est impossible de comparer les réponses à la mise en route du système, il faut donc comparer les réponse à une perturbation en régime établi comme sur la [figure 41](#). On constate que les réponses suivent la même évolution, ce qui valide le calcul de notre correcteur.

On a bien pris soin de s'assurer que l'asservissement avec la modélisation du hacheur sur PSIM était équivalente à celle sur Matlab (temps de réponse, gain statique, dépassement) avant d'effectuer les comparaisons précédentes.

6.iii.b Prise en compte des capteurs

Comme mentionné précédemment, notre système réel comporte des capteurs pour mesurer la vitesse de rotation de la MCC. En particulier, nous disposons de deux types de capteurs de vitesse différents : une dynamo-tachymétrique et un codeur incrémental. Nous considérons que leurs constantes de temps sont largement plus rapides que celles de notre système et les modéliserons donc seulement par des gains. Pour la dynamo-tachymétrique, son gain est spécifié comme étant de 6 V pour 1000 tr/min. Nous veillerons à diviser ce gain par deux pour faire en sorte que la tension de commande reste inférieur à 15 V.

Finalement, dans notre modèle Matlab, la vitesse de sortie est en *rad/s*, on multiplie donc cette vitesse par $\frac{30}{\pi}$ pour la convertir en *tr/min*. Le gain final à appliquer sur Matlab pour la dynamo-tachymétrique est donc :

$$k = \frac{1}{2} \cdot \frac{6}{1000} \cdot \frac{30}{\pi} \approx 28 \cdot 10^{-3}$$

Sur PSIM, le capteur nous retourne déjà une vitesse en *tr/min*, il suffira donc d'appliquer le gain de $\frac{3}{1000}$. Les modifications apportées sont présentées ci-dessous :

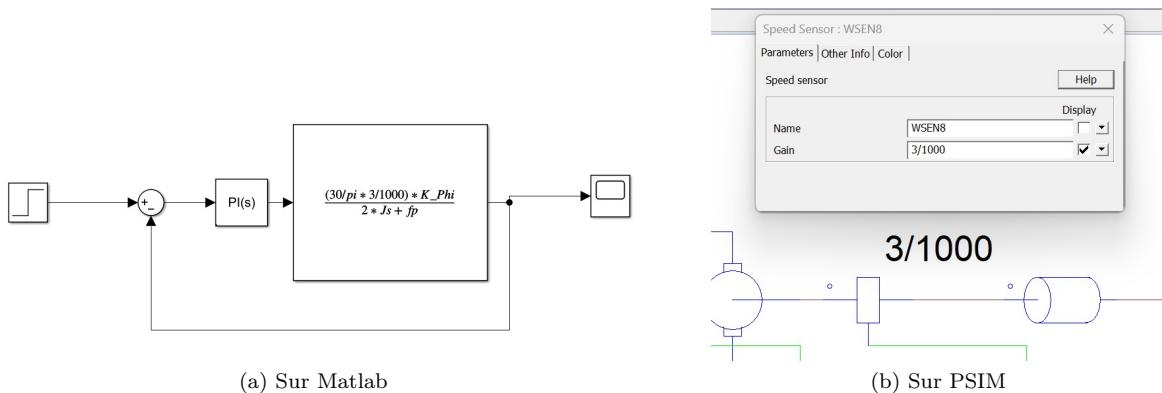


FIGURE 42 – Modifications prenant en compte les capteurs *

Pour le codeur incrémental, nous ferons en sorte qu'il ait un gain identique à la dynamo-tachymétrique pour n'avoir qu'à dimensionner qu'un seul correcteur.

Après calcul des nouveaux correcteurs grâce au *PID Tuner*, nous obtenons les valeurs suivantes :

$$C(p) = P \left(1 + I \frac{1}{p} \right) \text{ avec } P = 2.27 \text{ et } I = 22.8 \text{ s}^{-1}$$

Comme précédemment, nous avons vérifié ce correcteur et obtenu des résultats concluants. Cependant, nous allons cette fois-ci mettre en avant une comparaison complète de notre système. Pour cela nous simulerons le système complet et corrigé (courant et vitesse) sur PSIM, ainsi que le système complet et corrigé **sans les hypothèses simplificatrices** sur Matlab. C'est-à-dire que nous garderons tous les éléments de notre schéma-bloc hormis les éléments évoqués dans la partie préliminaire à l'asservissement.

Nous avons ainsi le schéma Simulink suivant :

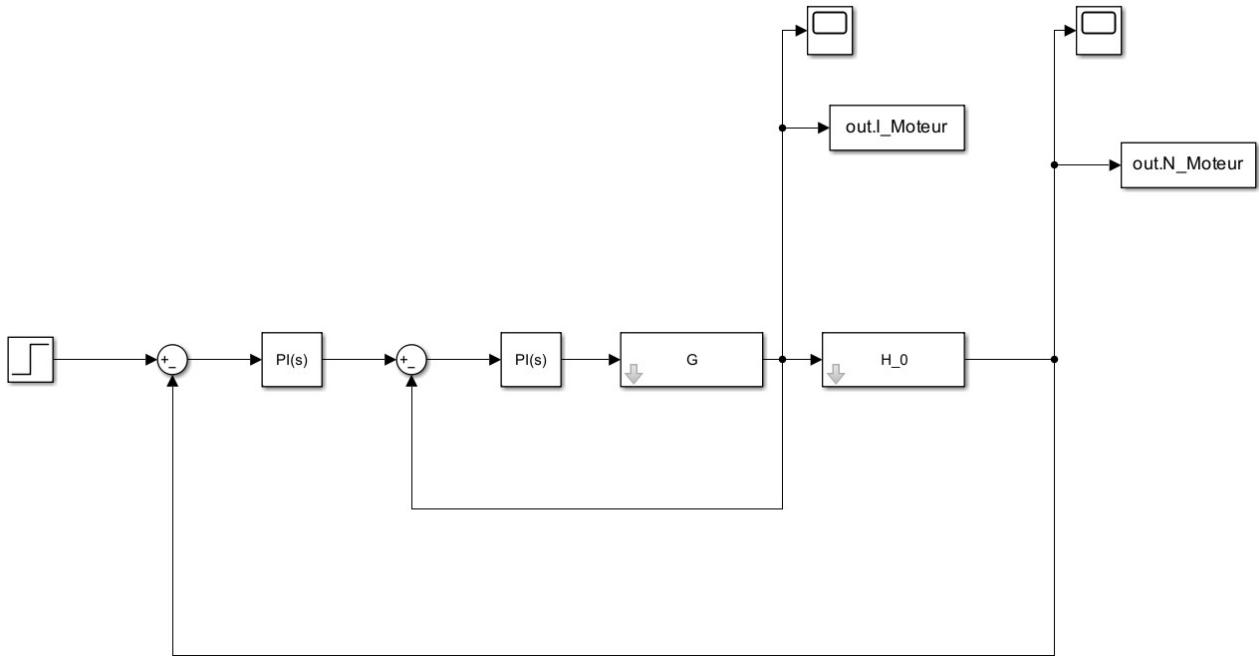


FIGURE 43 – Schéma-bloc de l'asservissement avec capteurs *

Où G et H_0 sont les fonctions de transfert permettant d'obtenir respectivement le courant (ou sa tension image considérant un gain unitaire) à partir de la commande issue des correcteurs et la tension image de la vitesse à partir du courant moteur. Le schéma utilisé pour PSIM est le même que celui utilisé pour la vérification de l'asservissement de vitesse, ce dernier faisant déjà intervenir tous les éléments.

Nous retrouverons sur les figures suivantes le modèle avec le hacheur de PSIM en rouge et le modèle Matlab en bleu.

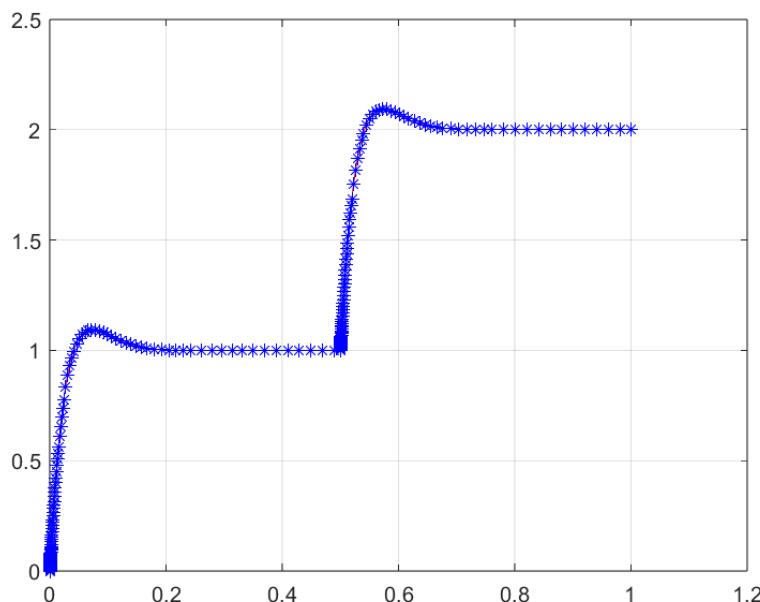


FIGURE 44 – Vérification de l'asservissement avec capteur (vitesse) *

On a en abscisse le temps en seconde et en ordonnée l'image de la vitesse en volt.

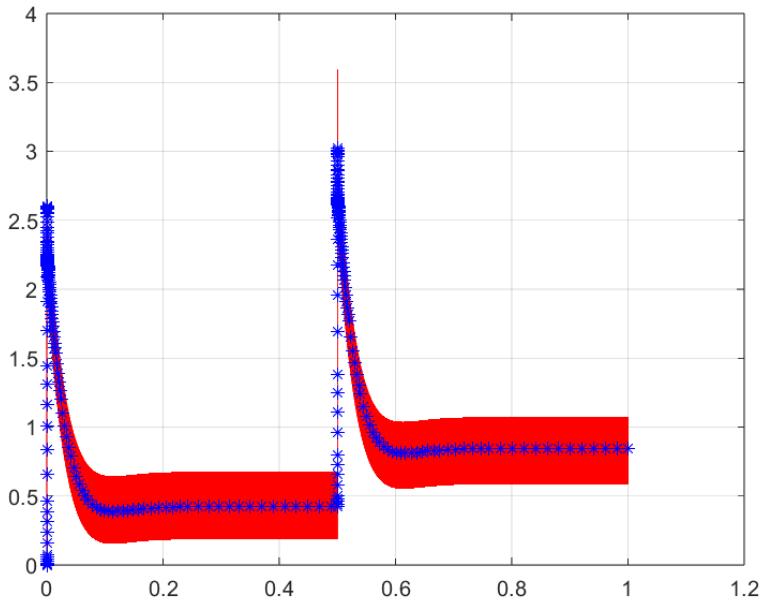


FIGURE 45 – Vérification de l’asservissement avec capteur (courant) *

On a en abscisse le temps en seconde et en ordonnée le courant en ampère (ou son image en volt).

On observe une concordance parfaite qui valide complètement notre cos correcteurs finaux.

6.iv Lien entre valeur de référence et valeur moyenne

Le but d’un asservissement est de faire tendre une grandeur caractéristique d’un système vers une valeur de référence (une consigne). Cependant, nous avons vu dans les études précédentes que les grandeurs que nous avons à asservir présentent des ondulations plus ou moins importantes. Il est donc légitime de se questionner sur le sens physique de la valeur de référence, étant donné que, la sortie n’étant pas constante, il est théoriquement impossible pour le système d’atteindre cette valeur. Dans notre cas, en considérant le fait que nous avons un intégrateur dans nos correcteurs, nous allons montrer que les valeurs de références Ω_{ref} et I_{Mref} sont égales aux valeurs moyennes $\langle \Omega \rangle$ et $\langle i_M \rangle$. Cela implique que c’est la valeur moyenne en sortie de notre système que nous allons asservir. La valeur de référence fixera la valeur moyenne de sortie du système. Nous allons démontrer cela pour le courant, et le cas de la vitesse sera purement analogue.

Soit y la variable d’état supplémentaire dûe à la présence d’un intégrateur dans notre système. On sait que :

$$y(t) = y(t_0) + \int_{t_0}^t (I_{Mref} - i_M) dt$$

Où t_0 est l’instant initial, c’est-à-dire le moment auquel nous avons mis en route le système.

Si l’on se place en régime permanent, on peut représenter la variation de courant comme suit :

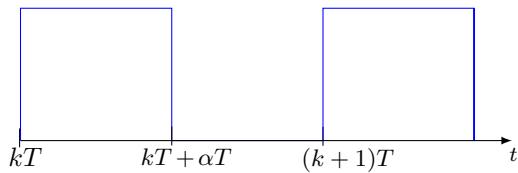


FIGURE 46 – Signaux liés au codeur incrémental

Où T représente une période de MLI du hacheur et α son rapport cyclique.

On peut en déduire que $i_M((k+1)T) = i_M(kT)$. De plus, on a :

$$\begin{cases} y((k+1)T) = y(t_0) + \int_{t_0}^{(k+1)T} (I_{Mref} - i_M) dt \\ y(kT) = y(t_0) + \int_{t_0}^{kT} (I_{Mref} - i_M) dt \end{cases}$$

Ainsi :

$$y((k+1)T) - y(kT) = \int_{kT}^{(k+1)T} (I_{Mref} - i_M) dt = T \cdot I_{Mref} - \int_{kT}^{(k+1)T} i_M dt = 0$$

Il est donc enfin possible d'obtenir le résultat final :

$$T \cdot I_{Mref} - \int_{kT}^{(k+1)T} i_M dt = 0 \Leftrightarrow I_{Mref} = \frac{1}{T} \int_{kT}^{(k+1)T} i_M dt = \langle i_M \rangle$$

7 Conception et dimensionnement des circuits électroniques

Maintenant que nous avons correctement dimensionné nos correcteurs, il est temps de concevoir le système électronique qui permettra de réaliser l'asservissement de manière effective. Pour cela, il faudra dimensionner les circuits relatifs à l'asservissement ainsi que certains circuits annexes primordiaux au bon fonctionnement de l'ensemble du système et au respect du cahier des charges. Il sera important de prendre en compte la tension d'alimentation de la carte qui sera une alimentation **symétrique +15 -15 V**.

7.i Asservissement

Dans cette partie, nous nous concentrerons sur les circuits propres à l'asservissement.

7.i.a Comparateur

Dans notre montage, nous aurons à plusieurs reprises besoin d'un comparateur. Ce composant a pour rôle, comme son nom l'indique, de comparer deux signaux d'entrées. La plupart du temps, un comparateur est un soustracteur, bien qu'il puisse également être un additionneur. Dans notre cas, nous utiliserons un soustracteur, dont le schéma est le suivant :

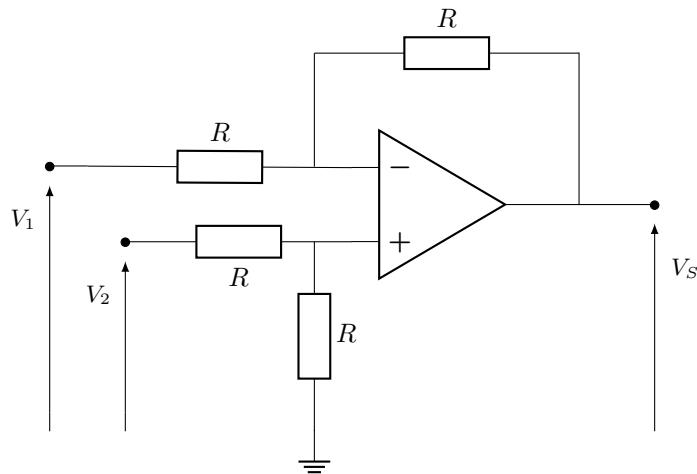


FIGURE 47 – Circuit comparateur

De ce schéma, nous pouvons ensuite calculer la fonction de transfert suivante en supposant l'AOP parfait :

$$V_- = \frac{\frac{V_1}{R} + \frac{V_s}{R}}{\frac{1}{R} + \frac{1}{R}} = \frac{V_1 + V_s}{2}$$

$$V_+ = \frac{1}{2} V_2$$

$$V_+ = V_- \Leftrightarrow V_2 = V_1 + V_s \Leftrightarrow V_s = V_2 - V_1$$

Nous obtenons bien l'équation de comparaison souhaitée. Pour réaliser ce circuit, il suffit donc d'avoir un AOP et 4 résistances de même valeur.

7.i.b Correcteur de vitesse

Nous devons asservir deux composantes de notre système : la vitesse de la MCC et le courant d'induit. Pour cela, nous utiliserons un correcteur de type PI (Proportionnel Intégral) qui permettra de contrôler le régime transitoire mais aussi d'annuler l'erreur statique. Ce correcteur se construit à partir d'un circuit intégrateur en série avec un amplificateur.

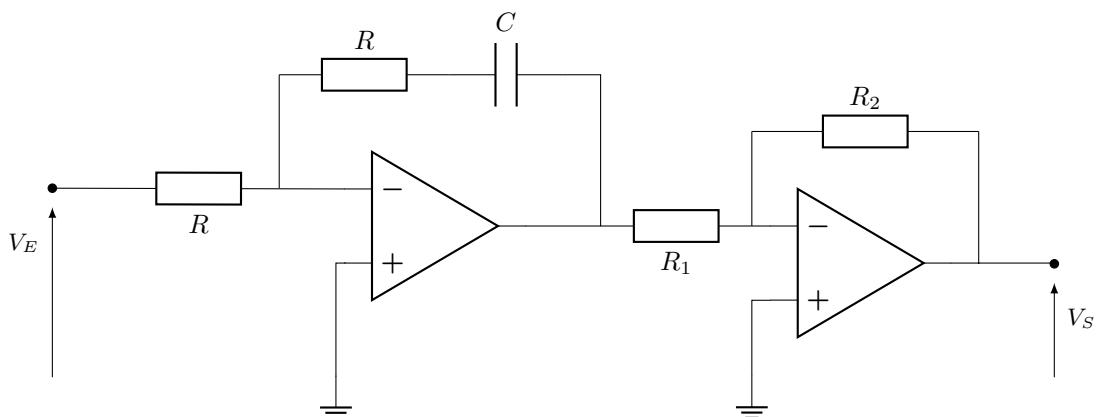


FIGURE 48 – Circuit correcteur PI

De ce schéma, on déduit la fonction de transfert suivante :

$$\begin{aligned} V &= -V_e \cdot \frac{R + \frac{1}{jC\omega}}{R} = -V_e \cdot \frac{1 + jRC\omega}{jRC\omega} \\ V_s &= -V \cdot \frac{R_2}{R_1} = V_e \cdot \frac{R_2}{R_1} \frac{1 + jRC\omega}{jRC\omega} \\ \text{si } p = j\omega \quad \Rightarrow \quad C(p) &= \frac{V_s}{V_e} = \frac{R_2}{R_1} \frac{1 + RCp}{RCp} = P(1 + \frac{1}{T_i p}) \end{aligned} \tag{7}$$

avec $\begin{cases} T_i = RC \\ K = \frac{R_2}{R_1} \end{cases}$

Il faut à présent choisir les valeurs de composants pour obtenir les coefficients K et T_i calculés précédemment. Pour rappel, pour l'asservissement de vitesse, nous avions :

$$P = 2,27$$

$$T_i = 0,044$$

Grâce aux relations décrites plus haut, nous pouvons calculer les valeurs de résistances et condensateurs nécessaires.

Les calculs effectués sont les suivants :

$$K_{vi} = 2,27 \implies R_1 = 1200 \Omega \text{ et } R_2 = 2700 \Omega \implies K_{vi_{réel}} = \frac{2700}{1200} = 2,25$$

$$T_{iv} = 0,044 \text{ s} \implies R = 43 \text{ k}\Omega \text{ et } C = 1 \cdot 10^{-6} \text{ F} \implies T_{vi_{réel}} = 43000 \times 1,10^{-6} \approx 0,043 \text{ s}$$

7.i.c Correcteur de courant

Le circuit de correction du courant est le même que pour l'asservissement de vitesse, la seule différence est la valeur des composants. On rappelle que pour l'asservissement de courant, on a les valeurs suivantes :

$$P = 6,219$$

$$T_i = 0,00015$$

Les valeurs des correcteurs sont donc calculées ainsi :

$$K_{ci} = 6,219 < 1 \implies R_1 = 1100 \Omega \text{ et } R_2 = 6800 \Omega \implies K_{ci_{réel}} = \frac{6800}{1100} = 6,182$$

$$T_{ci} = 0,00015 \text{ s} \implies R_8 = 1500 \Omega \text{ et } C = 100 \cdot 10^{-9} \text{ F} \implies T_{ci_{réel}} = 1500 \times 100 \cdot 10^{-9} \approx 0,00015 \text{ s}$$

7.ii Circuit de mise en forme du signal tachymétrique

La dynamo tachymétrique ayant un gain de 6 V pour 1000 tr/min, si la consigne appliquée était trop élevée, nous dépasserions la limite maximale supportée par le hacheur qui est de 15 V. Pour remédier simplement à ça, nous avons rajouter un pont diviseur de tension par 2 avec 2 résistances identiques suivi par un AOP en suiveur en sortie de la dynamo tachymétrique.

7.iii Circuit de mise en forme du signal du codeur incrémental

Avant de discuter de la solution finale adoptée pour la mise en forme du signal du codeur incrémental, étudions d'abord son fonctionnement.

Le codeur incrémental est un disque comportant 2 pistes : une piste extérieure et une piste intérieure. La piste extérieure est une série de fenêtres réparties équitablement autour du disque : ici on a un codeur incrémental avec 1800 fenêtres donc elles sont chacune décalées d'un angle de $\frac{2\pi}{1800}$. En pratique, une LED vient éclairer le disque en continu d'un côté et de l'autre, un photo-transistor va être éclairé à intervalle régulier. Une illustration du principe est présentée ci-dessous :

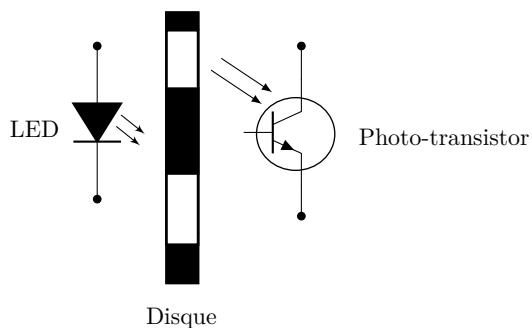


FIGURE 49 – Principe de fonctionnement du codeur incrémental

La fréquence à laquelle le transistor va être éclairé va être directement proportionnelle à la vitesse du moteur. Le photo-transistor va être alternativement saturé et bloqué et va donc fournir un signal carré à sa sortie, nommé *A*.

La période de ce signal est ainsi :

$$T = \frac{1}{1800} \cdot \frac{2\pi}{\Omega}$$

La piste intérieure est la même que la piste extérieure mais décalée d'un quart de l'angle entre deux fenêtres c'est-à-dire $\frac{1}{4} \cdot \frac{2\pi}{1800}$. Le signal représentant la variation de lumière à travers cette piste sera nommé B .

Pour mettre en forme ces signaux et récupérer la vitesse du moteur, deux méthodes sont possibles. Ces deux méthodes exploitent la génération d'un signal particulier à partir du signal du codeur pour récupérer l'information de la vitesse. Sur la figure suivante sont présentés les formes d'onde du codeur ainsi que les signaux des deux méthodes permettant la récupération de l'information utile.

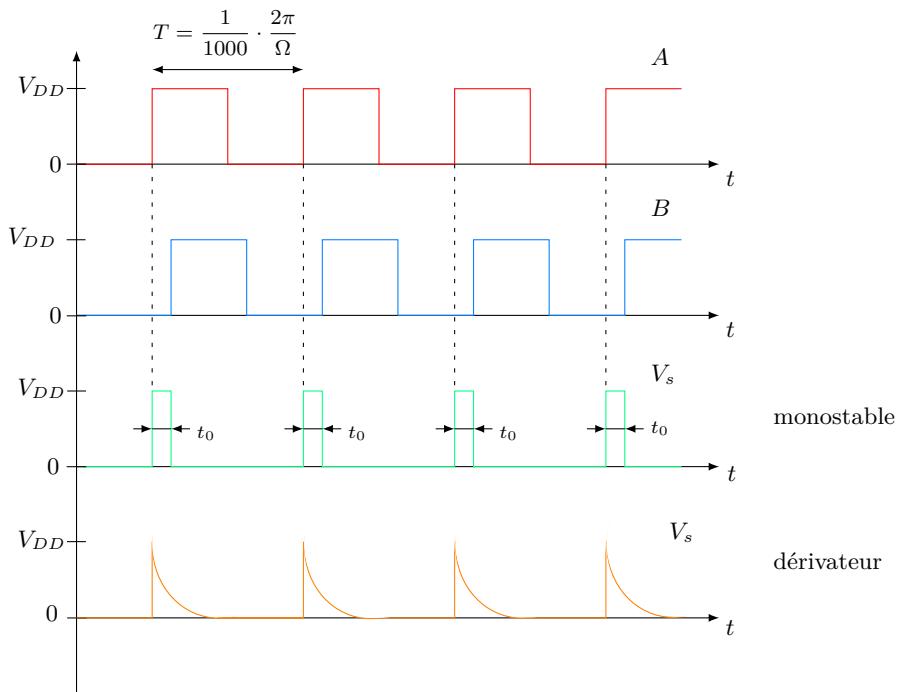


FIGURE 50 – Signaux liés au codeur incrémental

Pour s'assurer de la bonne qualité des signaux A et B à l'entrée de la carte, nous utiliserons deux comparateurs qui compareront les signaux du codeur avec $\frac{V_{DD}}{2}$. Nous obtiendrons ainsi en sortie des signaux carré de bonne qualité pour leur traitement.

- **Méthode 1 : Circuit monostable**

La première méthode consiste à concevoir un monostable qui produirait une impulsion de largeur t_0 fixée à chaque front montant du signal du codeur. Ainsi, on a la relation suivante :

$$\langle V_s \rangle = V_{DD} \cdot \frac{t_0}{T} = \frac{1800 \cdot \Omega}{2\pi} \cdot V_{DD} \cdot t_0 = \frac{1800 \cdot V_{DD} \cdot t_0}{2\pi} \cdot \Omega$$

On a bien ici une tension directement proportionnelle à la vitesse, il est donc possible de récupérer cette information. Sachant que nous devons obtenir un gain identique à celui de la dynamo tachymétrique, il faut multiplier le signal par une constante k' tel que si on note k le gain de la dynamo tachymétrique (*par rapport à une vitesse en rad/s*), on ait :

$$k' \cdot \frac{1800 \cdot V_{DD} \cdot t_0}{2\pi} = k \Leftrightarrow k' = k \cdot \frac{2\pi}{1800 \cdot V_{DD} \cdot t_0} = \left(\frac{1}{2} \cdot \frac{6}{1000} \cdot \frac{30}{\pi} \right) \cdot \frac{2\pi}{1800 \cdot V_{DD} \cdot t_0}$$

Il y a une contrainte à respecter avec cette méthode, c'est le choix de t_0 . En effet, ce dernier doit être plus petit que la période du codeur pour être certain qu'aucun front montant ne soit raté et donc ne fausse le résultat.

Dans la pratique, le monostable se construirait avec un **NE555** monté en monostable comme dans le schéma suivant :

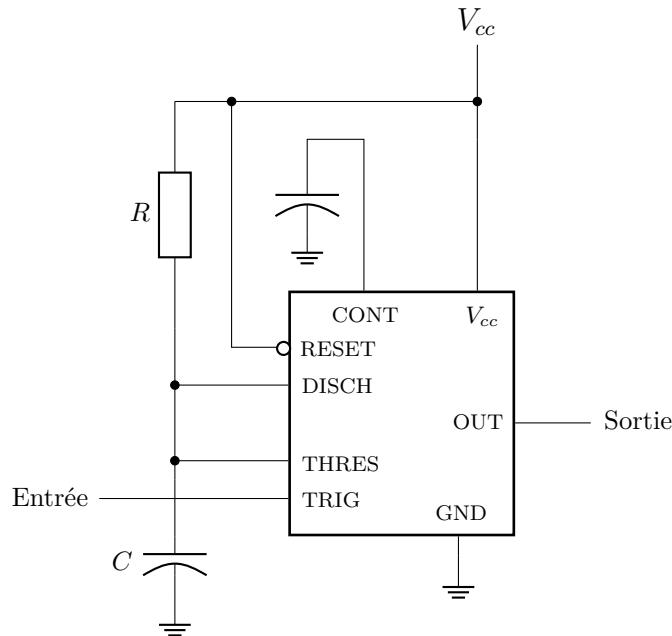


FIGURE 51 – NE555 monté en astable

On aurait alors :

$$t_0 = \ln(3) \cdot R \cdot C$$

- **Méthode 2 : Circuit déivateur**

L'autre méthode exploite le signal de sortie d'un circuit déivateur. Le schéma de ce circuit est présenté ci-dessous :

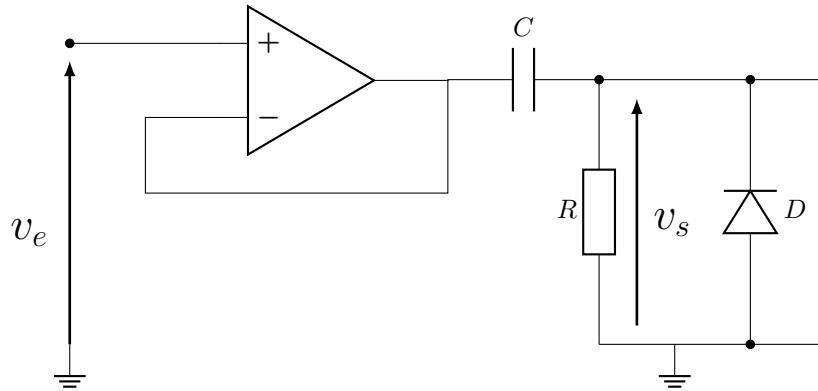


FIGURE 52 – Circuit déivateur

La diode permet de ne garder qu'uniquement les alternances positives.

En réponse à l'échelon de tension V_{DD} imposé par le codeur et en notant $\tau = RC$, on sait qu'on a en sortie un signal régit par l'équation suivante :

$$v_s(t) = V_{DD} \cdot \exp\left(-\frac{t}{\tau}\right)$$

On en déduit la valeur moyenne de ce signal :

$$\langle v_s(t) \rangle = \frac{1}{T} \int_0^T V_{DD} \cdot \exp\left(-\frac{t}{\tau}\right) dt = \frac{\tau}{T} \cdot V_{DD}$$

On constate qu'on obtient le même résultat qu'avec la méthode 1, le gain à appliquer suit donc la même formule.

La contrainte à respecter pour τ est qu'il doit être négligeable devant la période du codeur. En effet, il faut que le signal de sortie du déivateur ait le temps d'atteindre son régime permanent avant l'échelon suivant. La justification est la même que pour la méthode 1 à la différence qu'ici τ ne doit pas seulement être inférieur à T mais très inférieur.

Avec ces deux méthodes, nous obtiendrons des signaux contenant des harmoniques en plus de la valeur moyenne qui contient l'information utile : cela pourrait venir perturber le fonctionnement du système. De ce fait, pour obtenir un signal moins oscillant autour de la valeur moyenne, nous procéderons à un filtrage passe-bas. Il y a deux contraintes à respecter sur la fréquence de coupure :

- Elle doit être assez basse pour atténuer suffisamment les harmoniques.
- Elle ne doit pas être trop basse afin de ne pas introduire un pôle non-négligeable qui pourrait venir ralentir la réponse de notre système.

Après de premières estimations en tenant compte de la vitesse nominale du moteur et du temps réponse souhaité pour le système, nous avons procédé à des tests empiriques par simulation sur PSIM. Nous sommes arrivés à une valeur concluante de fréquence de coupure d'environ 1 kHz.

Finalement, compte-tenu de l'incertitude sur les composants que nous utiliserons, il est fort probable que les gains (*cf. méthodes 1 & 2*) calculés et mise en œuvre ne soit pas exactement les bons. C'est pour cela que nous mettrons en œuvre un gain réglable que nous ajusterons lors des tests pour obtenir une réponse la plus proche possible de nos attentes. Il est important de d'abord filtrer puis amplifier pour éviter de saturer les AOPs.

Nous avons dans un premier temps simulé la méthode 1 en tenant compte de toute les remarques précédentes sur PSIM. Le schéma est présenté sur la figure ci-dessous :

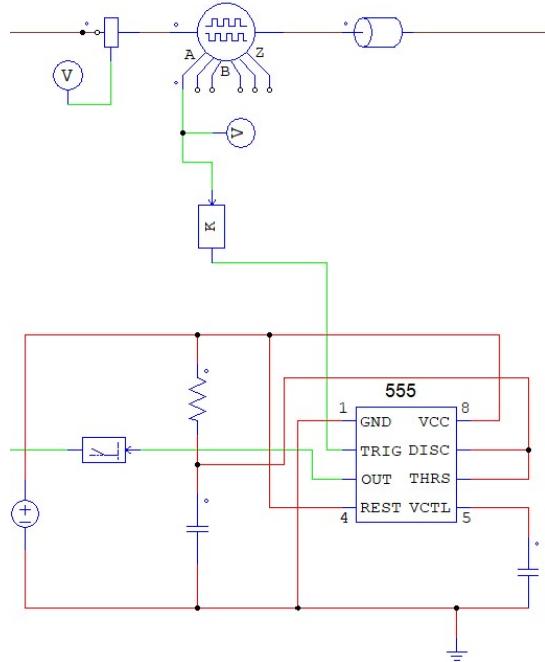


FIGURE 53 – Schéma de simulation de la méthode 1 sous PSIM

Il est important de noter que lorsque nous avons réalisé ces simulations, nous n'avions pas encore tous les éléments concernant le système (tension de sortie du capteur à l'état haut...).

De plus, ce n'est pas la méthode finale que nous avons décidé d'adopter, et le but de ces simulations était de nous assurer du fonctionnement de la méthode. Ainsi, certains éléments pourraient manquer de cohérence, mais cela ne retire pas les conclusions quant à son bon fonctionnement. Si nous avions choisi cette méthode, nous aurions, comme nous le verrons pour l'autre solution, approfondi la simulation avec les composants réels.

Le gain se situe dans le même bloc que le filtre. Sur la figure suivante, on retrouve la réponse du système asservie avec le retour vitesse provenant du codeur. En rouge, on retrouve la réponse en sortie du circuit de mise en forme et en vert la réponse en sortie du capteur tachymétrique.

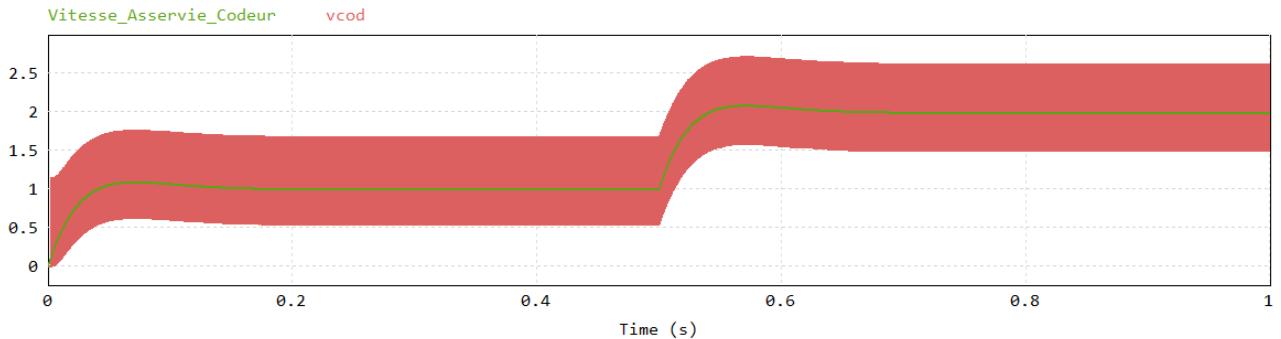


FIGURE 54 – Résultat méthode 1

On a en abscisse le temps en seconde et en ordonnée la tension image de la vitesse en volt.

On obtient bien les caractéristiques attendues et on conclut que la méthode fonctionne.

Nous nous sommes assurés de la même façon avec succès du bon fonctionnement de la deuxième méthode. Au vu de sa simplicité, nous avons opté pour cette solution. Avant d'évoquer sa mise en œuvre, un dernier élément est à préciser. Le capteur tachymétrique renvoie non seulement une information sur la vitesse de rotation, mais aussi sur le sens grâce au signe de la valeur renvoyée. Cela permet d'asservir le moteur dans les deux sens. Ce n'est pour l'instant pas le cas avec notre circuit du codeur. Pour corriger cela, nous utiliserons un multiplexeur analogique dont le schéma est présenté ci-dessous :

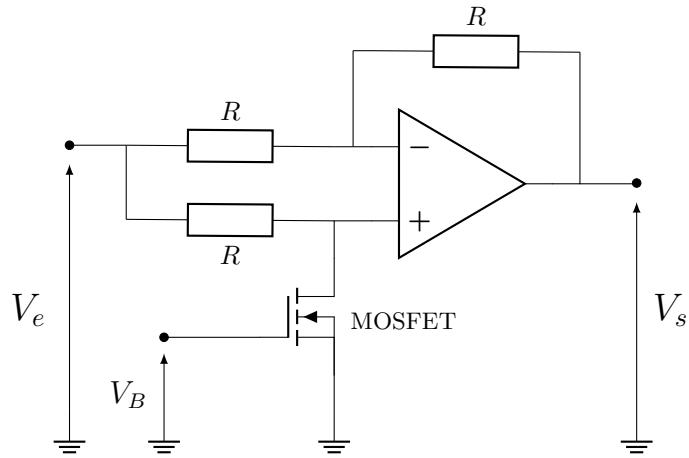


FIGURE 55 – Multiplexeur analogique

Ce montage permute entre le signal inversé ou non-inversé en fonction du signal appliqué à la base du MOSFET. Ce dernier proviendra du circuit **indicateur de sens de rotation** que nous verrons plus tard.

Nous pouvons maintenant passer à la mise en œuvre complète du circuit de mise en forme.

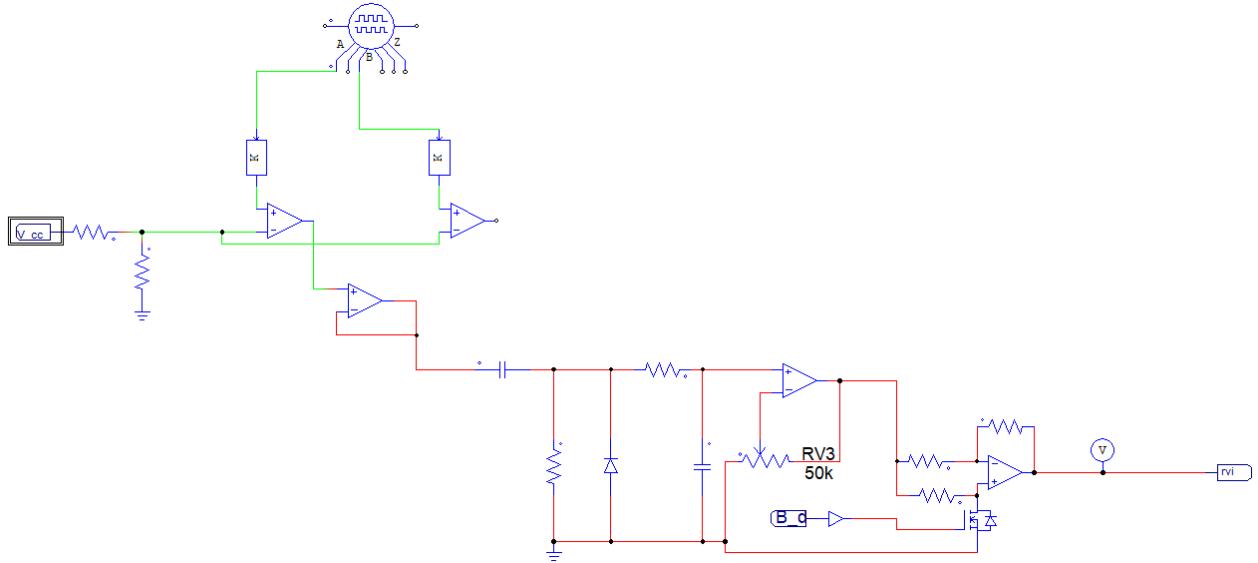


FIGURE 56 – Schéma de simulation de la méthode 2 sous PSIM

On note l'ajout d'un suiveur avant le circuit déivateur pour ne pas que les étages suivants n'ait d'influence sur la sortie du comparateur. Le filtre sera simplement un circuit RC. Nous exposerons les résultats de simulation du système complet dans le chapitre suivant.

7.iv Limiteur de courant

Afin de respecter le cahier des charges et limiter le courant d'induit à $1,1 \cdot I_{nom} = 5 A$, nous utiliserons un circuit limiteur de courant. Pour réaliser cela, nous avons pu mettre en place le circuit suivant à l'aide d'un AOP :

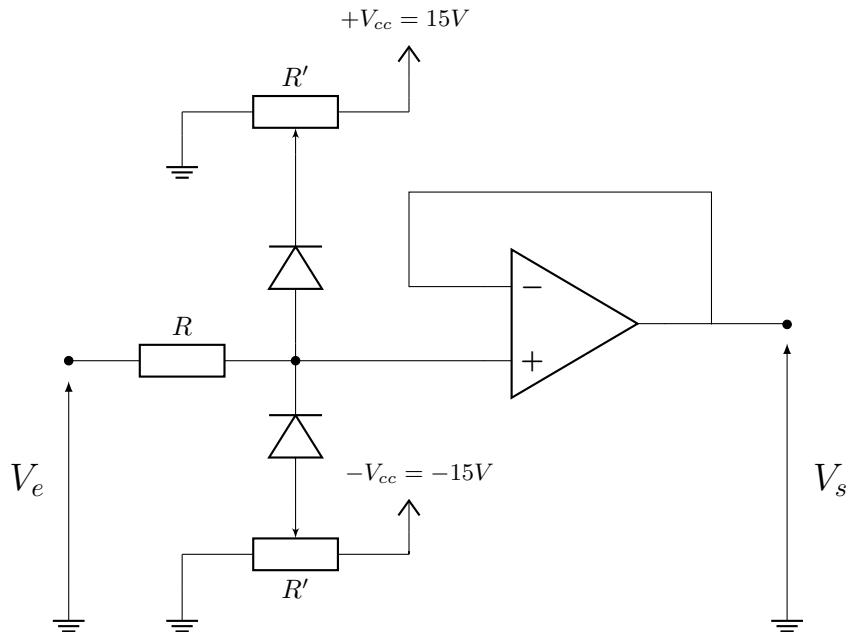


FIGURE 57 – Circuit limiteur de courant

On réglera R' pour obtenir la tension de seuil désirée.

7.v Limiteur de la tension de commande

Ce limiteur est intégré au hacheur et permet de ne pas dépasser la tension de commande maximale.

7.vi Indicateur de sens de rotation

Pour obtenir l'information sur le sens du rotation du moteur, nous allons utiliser le circuit suivant :

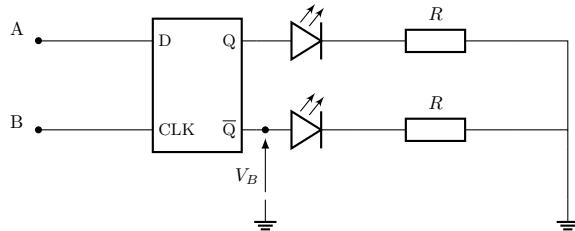


FIGURE 58 – Circuit indicateur de sens de rotation

Où **A** et **B** sont les signaux issus du codeur après passage dans les comparateurs de remise en forme. Avec ce circuit, en fonction du sens de rotation, Q ou \bar{Q} sera à l'état haut et éclairera une LED d'indication. En effet, dans un sens le signal A sera en avance sur B et en retard dans l'autre. Ainsi, au front montant de B, on récupère l'information sur le sens de rotation grâce à la valeur de A. On utilise aussi la sortie \bar{Q} pour piloter le transistor du multiplexeur analogique.

7.vii Récapitulatif

Après avoir détaillé l'ensemble des circuits qui composent notre système, nous récapitulons ici l'ensemble des composants utilisés pour chaque montage et leur valeur réelle. Les composants ont été commandés sur le site marchand de Farnell. Dans notre cas, nous avons choisi d'utiliser principalement des éléments CMS à l'exception des potentiomètres et de la bascule D.

Fonction	Composants et valeurs
Correcteur de vitesse	$R = 43 \text{ k}\Omega$, $R_1 = 1,2 \text{ k}\Omega$, $R_2 = 2,7 \text{ k}\Omega$, $C = 1 \mu\text{F}$, AOP : TL084 ($\times 2$)
Correcteur de courant	$R = 1 \text{ k}\Omega$, $R_1 = 1,1 \text{ k}\Omega$, $R_2 = 6,8 \text{ k}\Omega$, $C = 150 \text{ nF}$, AOP : TL084 ($\times 2$)
Comparateur courant	$R = 1,2 \text{ k}\Omega$ ($\times 4$), AOP : TL084
Comparateur vitesse	$R = 1,2 \text{ k}\Omega$ ($\times 4$), AOP : TL084
Limiteur de courant	$R = 1,2 \text{ k}\Omega$ ($\times 4$), $R' = 10 \text{ k}\Omega$ ($\times 2$), AOP : TL084, Diodes : 1N4148WS ($\times 2$)
Indicateur de sens de rotation	Bascule D : CD4013, LED rouge + LED jaune, $R = 1 \text{ k}\Omega$ ($\times 2$)
Circuit dérivateur	$R = 1 \text{ k}\Omega$, $C = 1 \text{ nF}$, Diode : 1N4148WS, AOP : TL084
Filtre RC	$R = 1,1 \text{ k}\Omega$, $C = 150 \text{ nF}$
Multiplexeur analogique	$R = 1,1 \text{ k}\Omega$ ($\times 4$), MOSFET : BSS138, AOP : TL084
Comparateurs signaux codeur	$R = 1,1 \text{ k}\Omega$ ($\times 2$), Comparateurs : LM2903 ($\times 2$)
Amplificateur réglable pour codeur	$R' = 10 \text{ k}\Omega$, AOP : TL084
Diviseur du signal tachymétrique	$R = 1,2 \text{ k}\Omega$ ($\times 2$)

TABLE 2 – Tableau récapitulatif des composants utilisés

Les notations des composants sont les mêmes que sur les circuits détaillés plus haut. La notation R' correspond aux potentiomètres. En pratique, nous avons utilisé des TL084 qui inclut 4 AOPs, la notation ($\times 2$) dans le tableau signifie donc que nous avons utilisé 2 AOPs d'un seul composant TL084 pour le circuit.

8 Simulation du système réel

Maintenant que notre système réel est complètement dimensionné, nous pouvons passer à sa simulation complète sur PSIM avant de passer à sa réalisation effective. Voici le schéma de notre système réel sous PSIM :

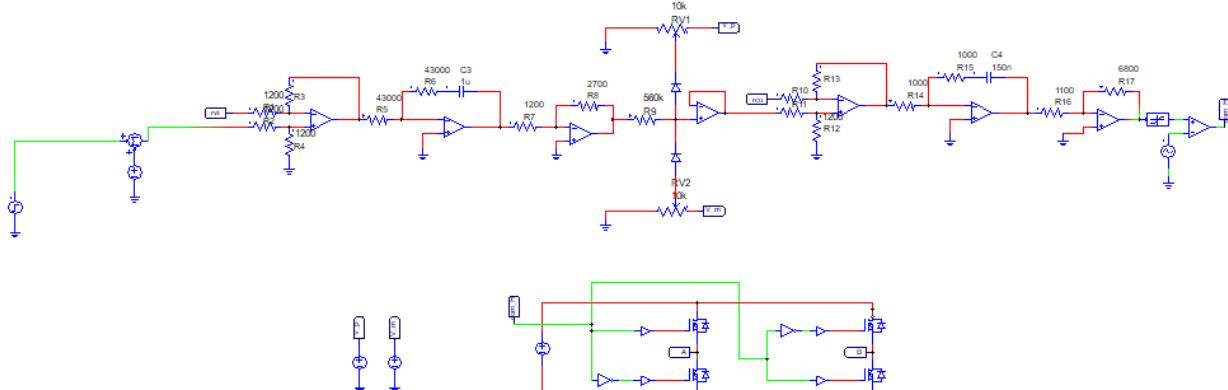


FIGURE 59 – Schéma de simulation du système réel sous PSIM - Boucle d'asservissement

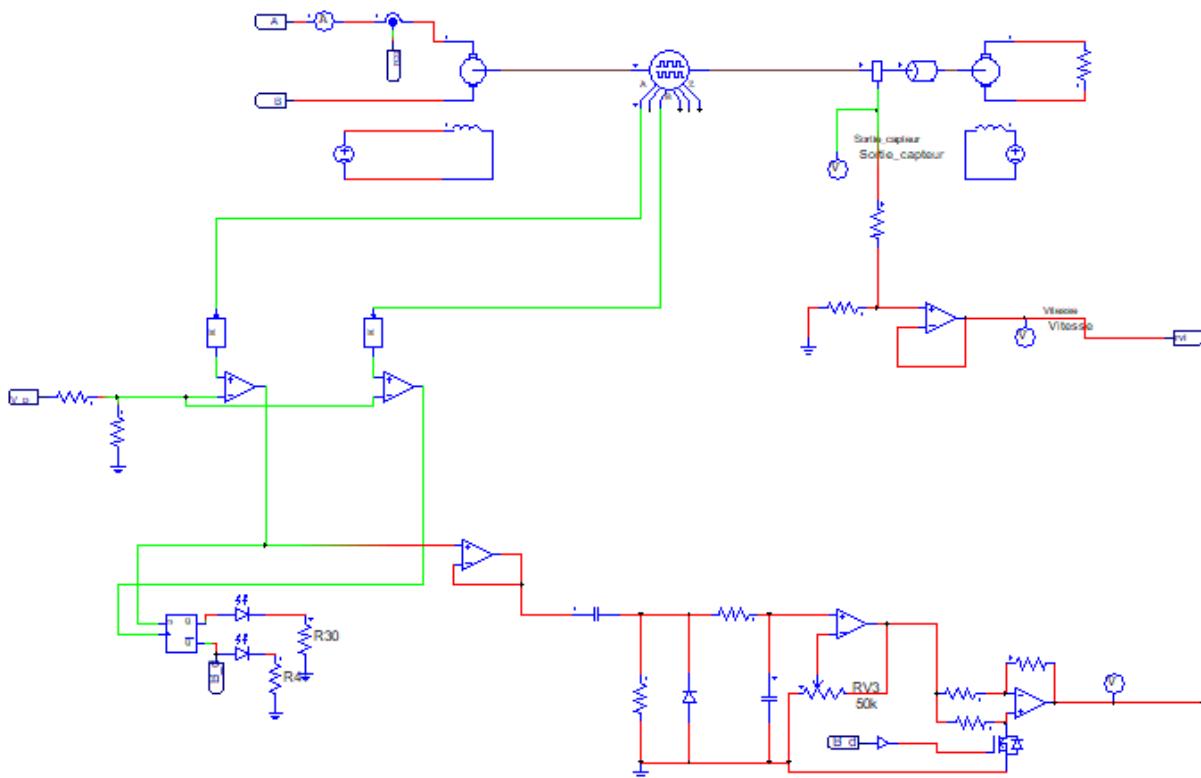


FIGURE 60 – Schéma de simulation du système réel sous PSIM - Partie capteurs

Nous avons simulé grâce à ce schéma le bon fonctionnement du système complet avec **les deux capteurs** pour des consignes **positives** et **négatives**.

Voici les résultats, on retrouvera en rouge les courbes théoriques attendues avec les composants blocs "parfaits" intégrés à PSIM et en vert les courbes réelles en sortie des capteurs correspondants :

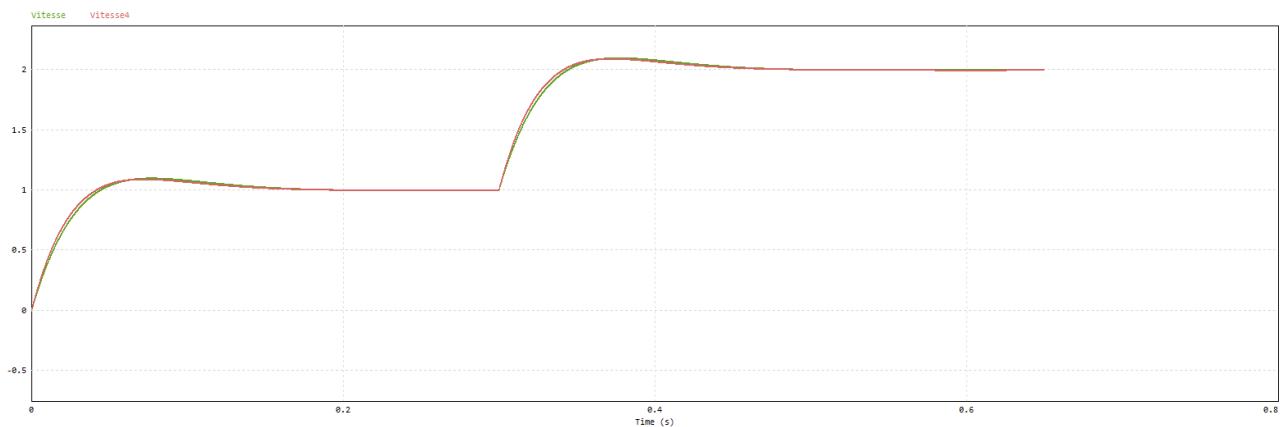


FIGURE 61 – Résultats pour une consigne positive avec le capteur tachymétrique

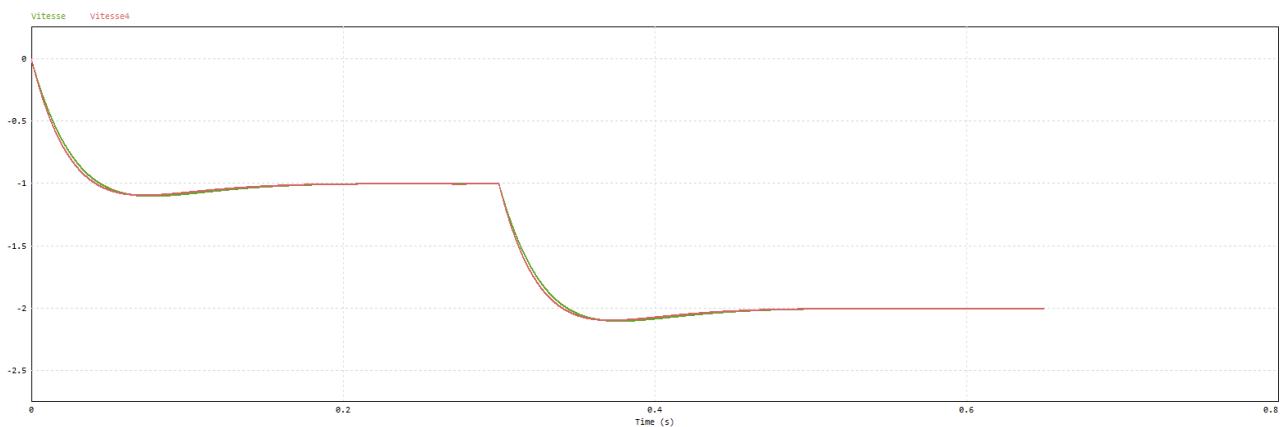


FIGURE 62 – Résultats pour une consigne négative avec le capteur tachymétrique

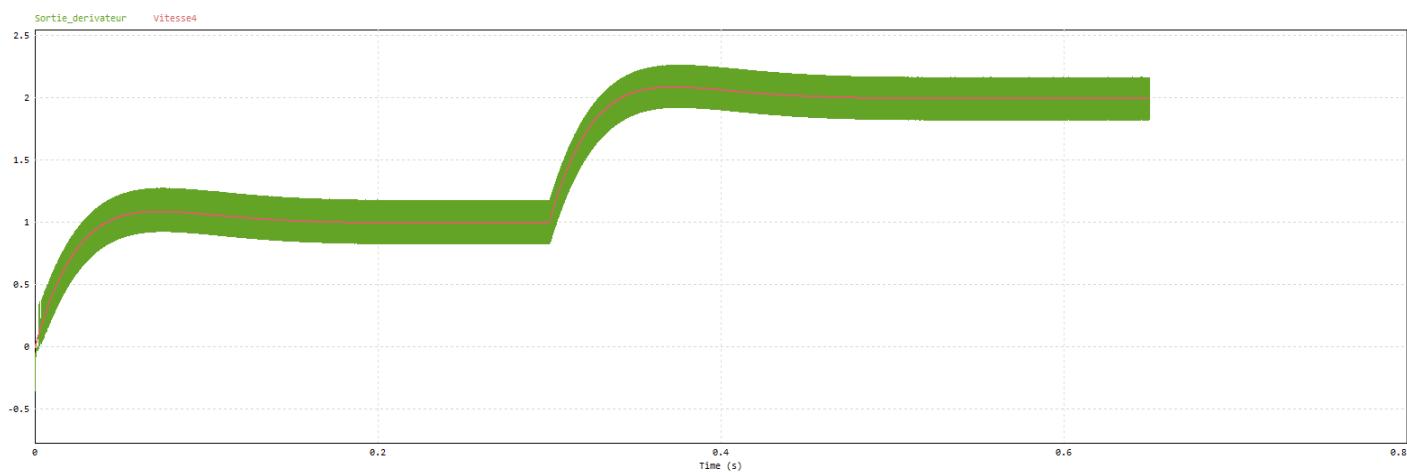


FIGURE 63 – Résultats pour une consigne positive avec le codeur incrémental

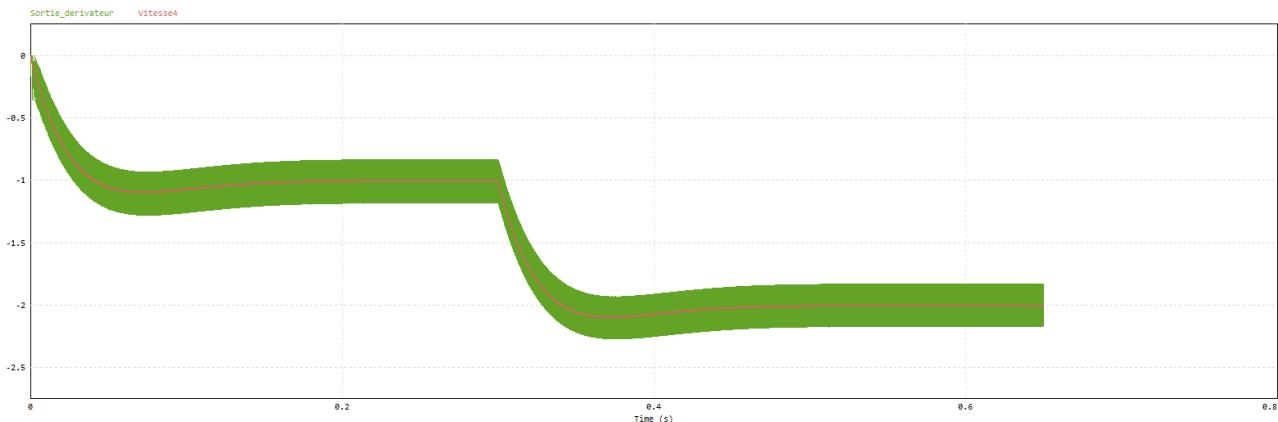


FIGURE 64 – Résultats pour une consigne négative avec le codeur incrémental

On a en abscisse le temps en seconde et en ordonnée l'image de la vitesse en volt.

On constate une très bonne correspondance entre théorie et système réel simulé ce qui valide donc notre dimensionnement. On peut à présent passer à la création de la carte électronique.

9 Réalisation pratique

9.i Implémentation dans Proteus

Maintenant que nous avons définis et dimensionnés nos éléments, nous pouvons commencer à les implémenter dans Proteus.

Premièrement, mettons en place les entrées et sorties des signaux utiles. Nous avons opté, pour ce projet, de séparer les bornes d'entrées de la carte afin d'avoir plus de flexibilité. Ainsi, l'entrée sur la carte se fera via des borniers à 2, 3 ou 4 entrées. La masse de notre circuit de commande sera faite en reliant le "-" de l'alimentation positive et le "+" de l'alimentation négative. Ceci sera fait directement sur l'alimentation. De plus, nous allons placer des condensateurs de découplage sur les alimentations, afin de réduire au maximum le bruit. Ils auront une valeur de 1 μ F.

On définit donc tout cela sur Proteus comme suit :

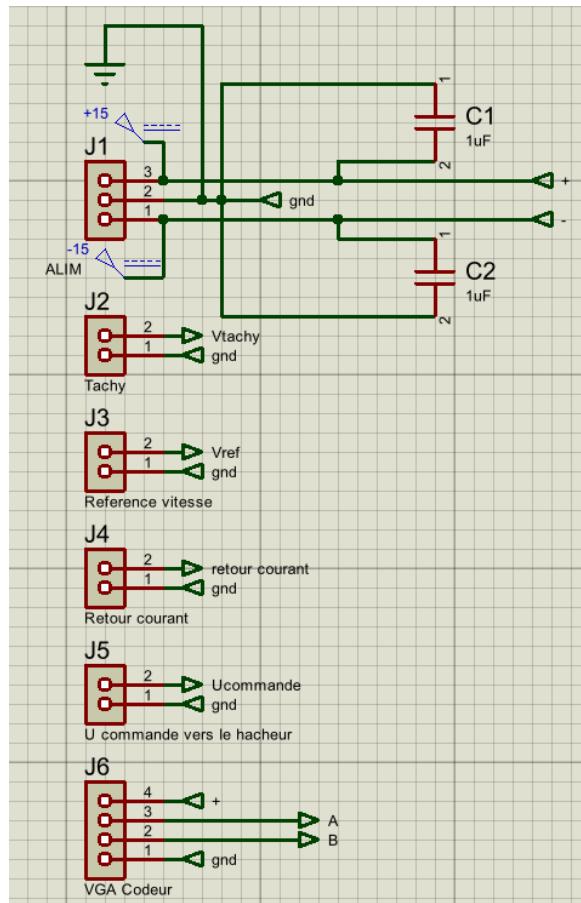


FIGURE 65 – Entrées de la carte

Il est important de bien rajouter la masse sur les borniers afin d'avoir la même référence de tension sur les capteurs, le hacheur et le circuit de commande.

Ensuite, nous allons réaliser la boucle principale de notre asservissement, c'est à dire le comparateur suivi du correcteur de vitesse, puis du limiteur de courant, du second comparateur et du correcteur de courant.

Pour ce faire, nous avons décidé d'utiliser des TL084, car ces boîtiers comprennent 4 AOP, ce qui permettra par la suite de simplifier les routages. Avec un premier TL084, on peut donc réaliser la chaîne suivante :

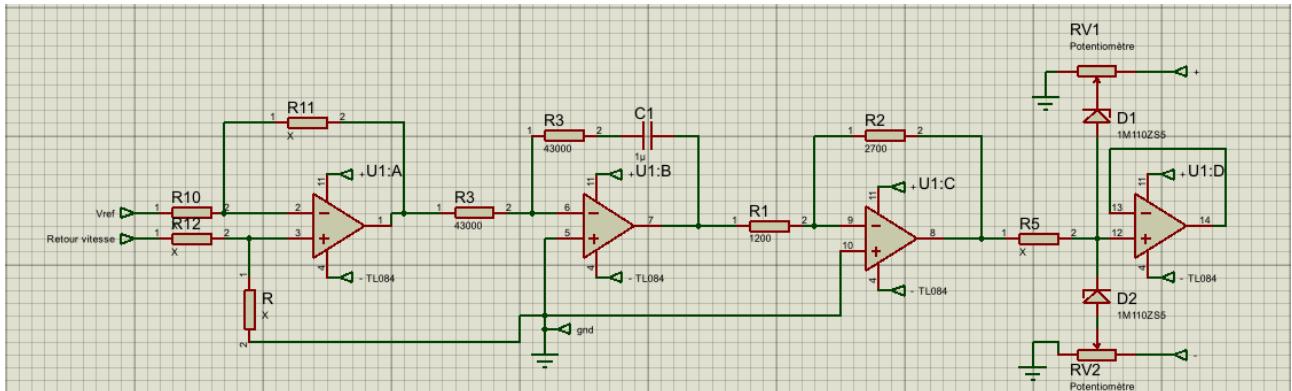


FIGURE 66 – Première chaîne

On y retrouve, comme sur le schéma bloc, de gauche à droite :

- Un soustracteur, la valeur des résistances importent peu du moment qu'elles sont toutes égales. Il s'agit du comparateur de vitesse.
- Le correcteur PI de la vitesse, composé d'abord de la partie intégrale suivie de la partie proportionnelle.
- Un limiteur de courant composé de diodes et de potentiomètres.

Avec le second TL084, on termine la chaîne principale de l'asservissement, comme suit :

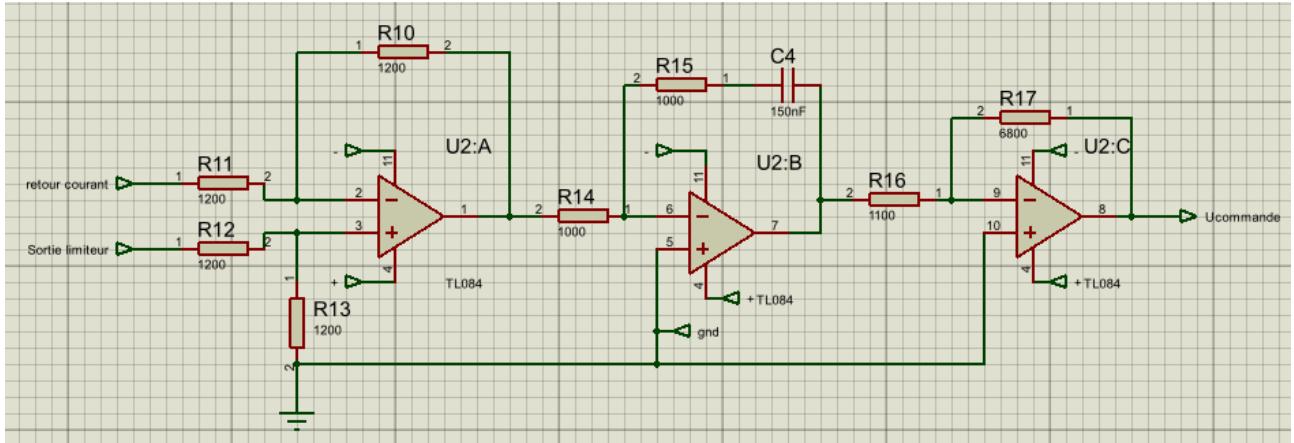


FIGURE 67 – Seconde partie de la chaîne principale

On retrouve à nouveau de gauche à droite,似ilairement au schéma bloc :

- Un nouveau soustracteur, qui joue le rôle de comparateur de courant.
- Le correcteur PI du courant, composé d'abord de la partie intégrale suivie de la partie proportionnelle.

Au final, on se retrouve avec le montage complet suivant :

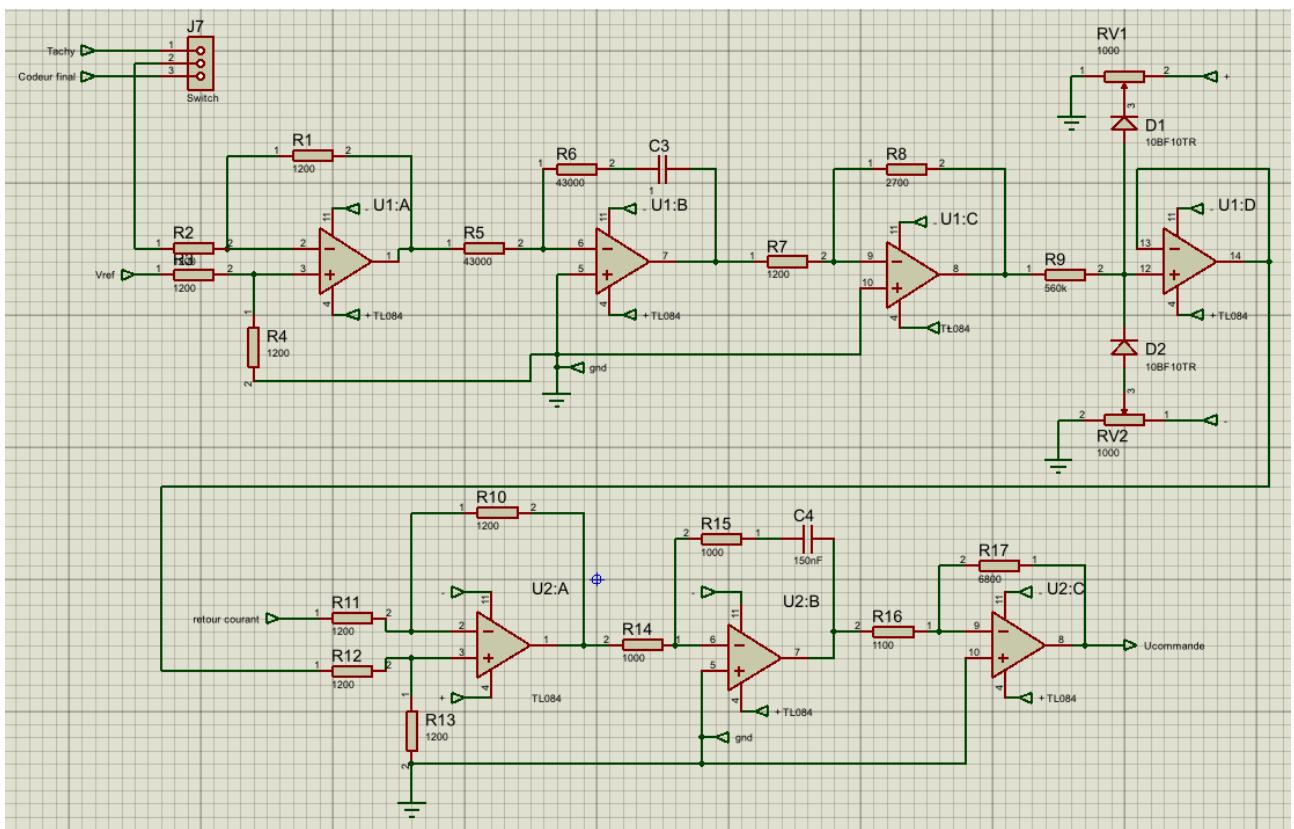


FIGURE 68 – Chaîne complète d'asservissement

En plus de la chaîne d'asservissement principale, nous allons avoir besoin de certains circuits auxiliaires, notamment pour l'acquisition des signaux d'entrée.

Comme nous l'avons expliqué en première partie, les signaux issus de codeur incrémental doivent d'abord être comparés à une tension de référence 7,5 V, puis dérivés et enfin filtrés. La simulation PSIM a démontré l'importance du filtrage dans cette application.

Sur Proteus, cela équivaut aux schémas suivants :

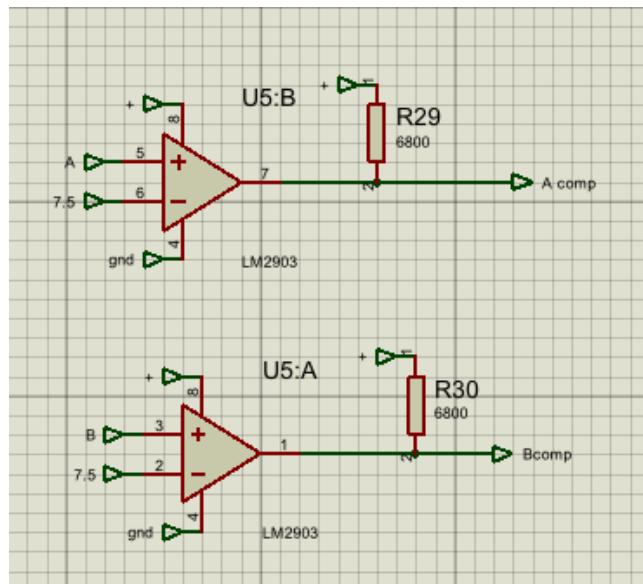


FIGURE 69 – Comparateur de tension

Pour le comparateur, nous avons opté pour le LM2903, car ce dernier comporte deux comparateurs dans un unique boîtier, simplifiant à nouveau le routage par la suite.

Nous avons également rajouté des résistances de pull-up à la sortie du comparateur car les sorties du codeur sont à collecteur ouvert. Nous nous sommes rendus compte par la suite que nous aurions dû les mettre en amont du comparateur et non en aval, mais nous avons su résoudre ce problème.

Pour réaliser la tension de référence 7.5 V, nous avons simplement implémenter un circuit pont diviseur de tension avec un suiveur. Sous Proteus, cela donne :

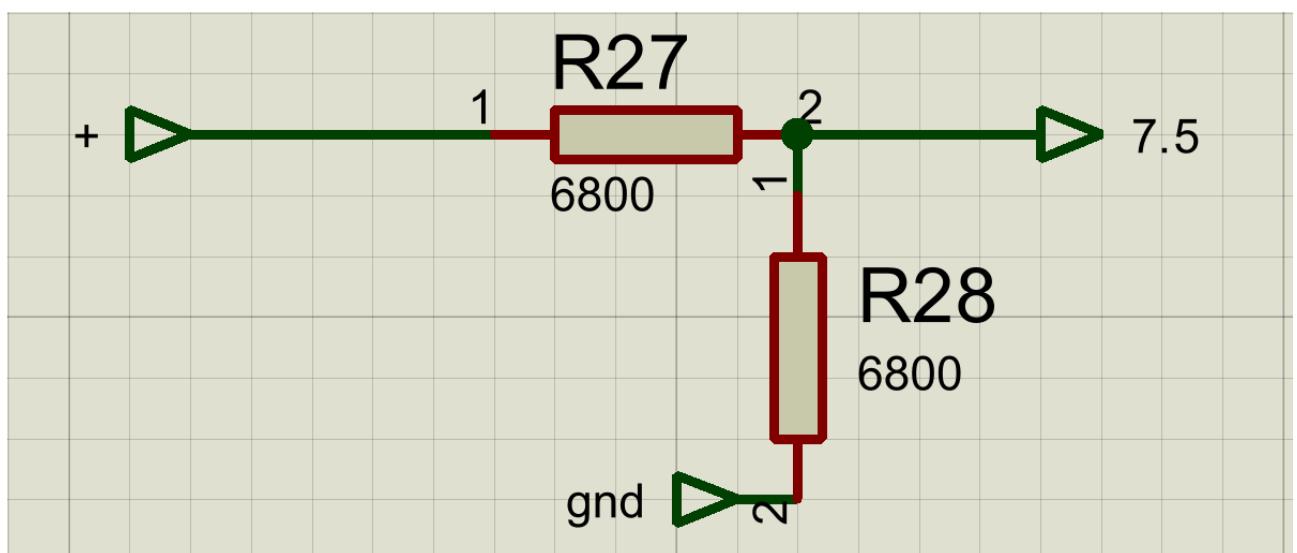


FIGURE 70 – Pont diviseur de tension par deux

Il s'agit d'un montage très basique mais fonctionnel. L'alimentation n'étant pas forcément stable, nous aurions pu également rajouter des condensateurs tampons pour palier aux éventuelles chutes de tension, ou directement utiliser un circuit régulateur de tension comme le LM7905. Cependant, nous avons jugé que pour cette application, un pont diviseur suffirait.

Ensuite, nous avons besoin d'un déivateur suivi immédiatement d'un filtre passe-bas puis d'un gain variable. Ce schéma est le suivant :

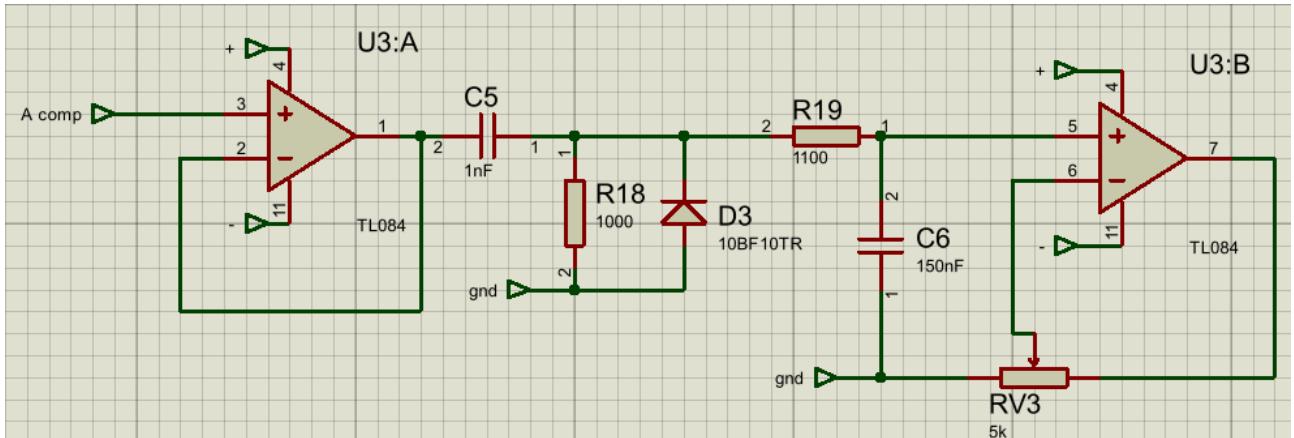


FIGURE 71 – Déivateur + filtre passe bas + Gain variable

Parallèlement à cela, on réalise, à l'aide d'une bascule D, le circuit permettant de déterminer le sens de rotation du codeur.

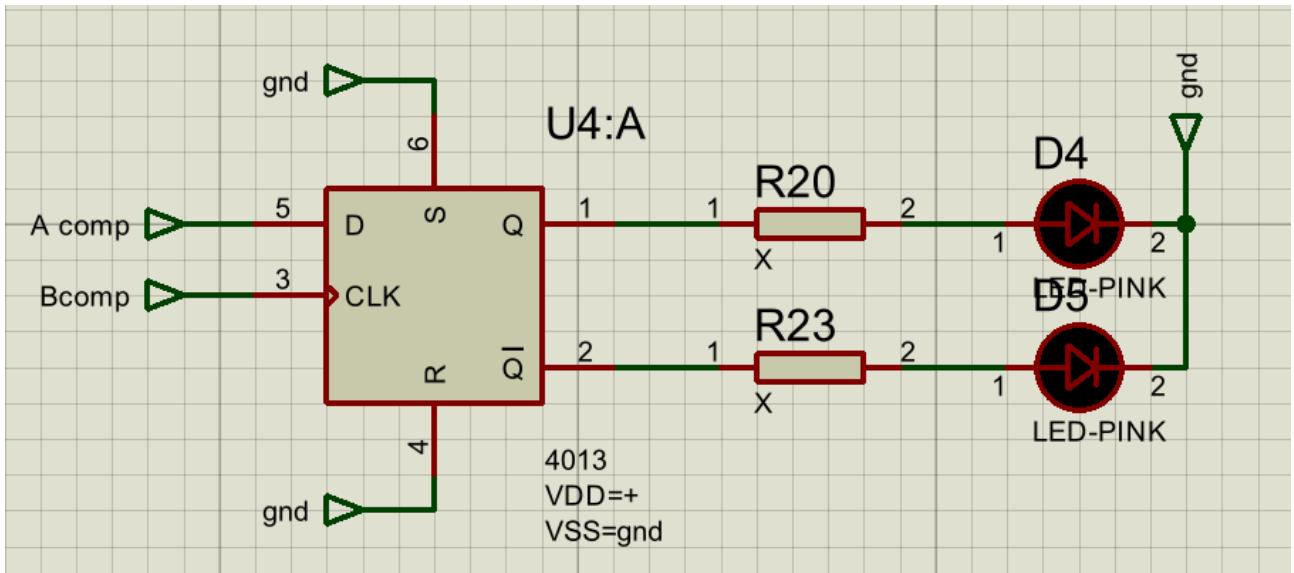


FIGURE 72 – Circuit à bascule D

Nous avons rajouté des LEDs en tant qu'indicateur visuel du sens de rotation.

Enfin, pour terminer l'acquisition du signal provenant du codeur, nous devons multiplexer le signal en sortie du filtre. Pour ce faire, on réalise le montage suivant :

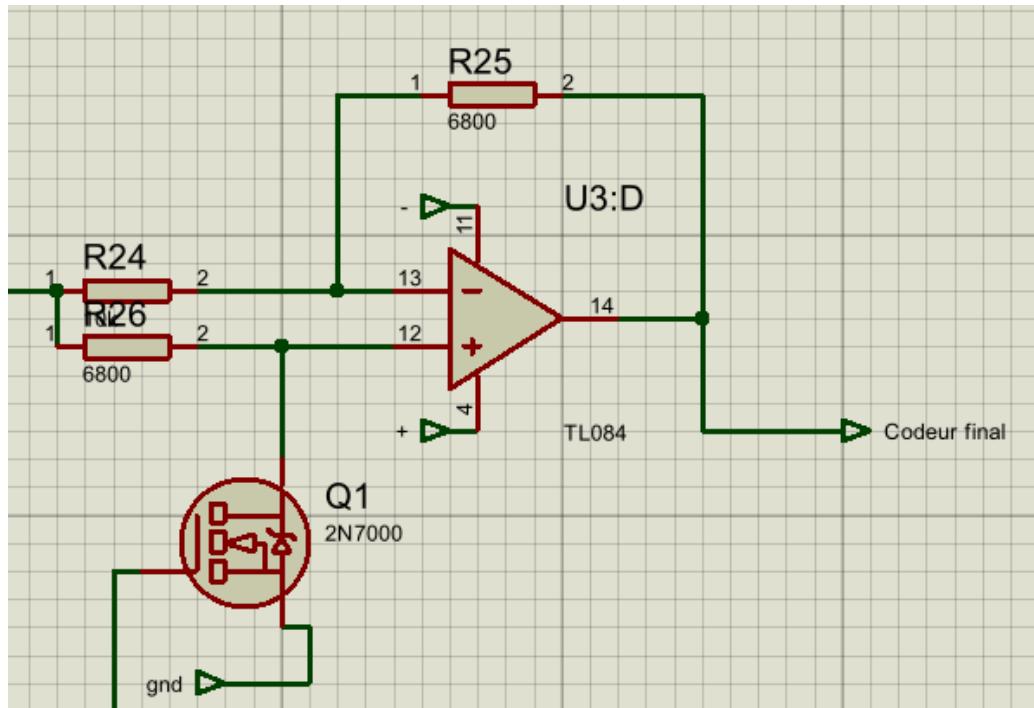


FIGURE 73 – Circuit de Multiplexage à AOP

Où le signal de grille du MOS est le signal \bar{Q} de la bascule et celui en entrée de R_{24} et R_{26} est le signal en sortie du filtre passe bas.

Ainsi, la chaîne complète d'acquisition du signal du codeur incrémental est donnée ci-dessous :

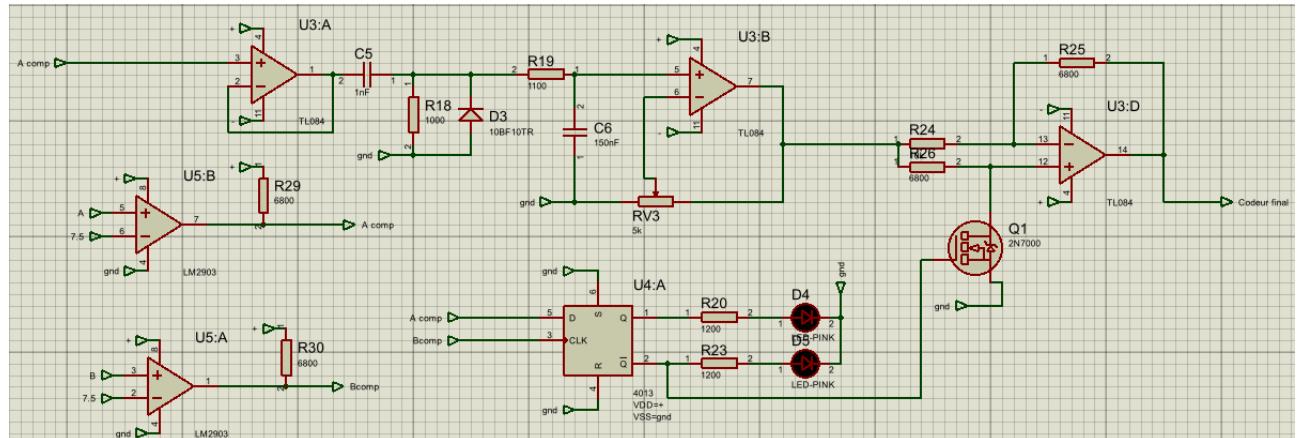


FIGURE 74 – Chaîne complète d'acquisition du codeur

Enfin, pour finaliser la commande, il ne nous reste plus qu'à implémenter un simple circuit diviseur de tension par deux pour l'acquisition du signal de la dynamo tachymétrique. Ceci donne :

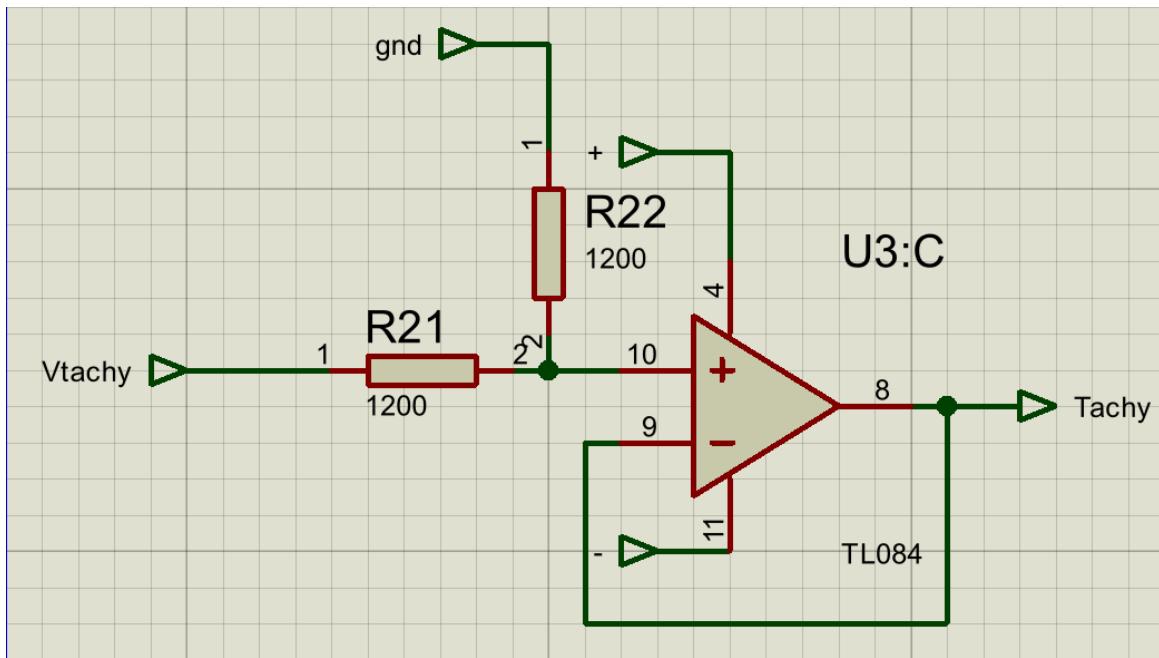


FIGURE 75 – Diviseur de tension pour le signal issu de la dynamo tachymétrique.

Avec ceci nous concluons la partie schématique de l'implémentation sur Proteus. Après avoir vérifié les liens entre toutes les sources et les circuits, nous pouvons passer au routage de la carte.

9.ii Création du PCB

Nous pouvons maintenant nous concentrer sur la réalisation du PCB. Cette partie du projet est importante car c'est avec elle que le projet s'ancre dans le réel. Nous devrons maintenant nous projeter afin de concevoir une carte optimisée et physiquement réalisable.

Pour les composants, nous avons choisi de partir sur une carte quasi entièrement composée de composants CMS. Cela nous permettra d'une part de simplifier les soudures car il suffira de placer les composants et de les mettre au four, mais également de gagner de la place car les composants CMS sont très petits. Des inconvénients existent néanmoins : les composants CMS seront plus compliqués à ressouder par la suite, et nous ne pourrons pas utiliser leurs broches comme des vias.

Avant de commencer le routage, nous avons dû commander les composants en portant une grande attention à leurs dimensions. En effet, il fallait que leur taille ne soit pas trop petite afin de pouvoir les manipuler. Ensuite, une fois les composants choisis, nous avons pu en extraire les packages. La plupart du temps, les boîtiers des composants ne correspondaient pas à ce que nous avions sur Proteus, il a donc fallu réaliser les packages manuellement en s'aidant des datasheets des composants. En plus de cela, il fallait porter soin à l'ordre des broches des composants plus complexes comme le comparateur ou le transistor pour assurer de leur bon fonctionnement.

9.ii.a Règles de conception

Maintenant, parlons des règles de design que nous avons respecté pour s'assurer une carte réussie.

Tout d'abord, nous avons appris lors des projets précédents qu'il est primordial de laisser un espace entre chaque piste pour éviter les courts-circuits. Dans l'onglet "règle de conception", nous avons alors stipulé un écart minimum de 25th entre toutes les zones cuivrées ; pistes, vias, broches, plan de masse...

Des exceptions à cette règle ont néanmoins été faites au niveau des broches du comparateur en raison de sa très petite taille.

La prochaine étape a consisté à rajouter un plan de masse sur chaque face du PCB. Cela présente plusieurs avantages. D'une part, cela accorde un accès facile à la masse sur toute une face, simplifiant grandement les routages. D'autre part, cela évite de retirer inutilement une grande quantité de cuivre et permet alors de gagner du temps et d'économiser de la matière lors de la conception physique de la carte.

Au niveau des packages de composants, servant à modéliser ces derniers sur le logiciel, nous avons, pour ceux réalisés par nous-mêmes, volontairement agrandi les pastilles cuivrées afin d'avoir une marge de manœuvre lors des soudures. Cela permet également une meilleure répartition de l'étain sur le cuivre.

Enfin, nous avons placé tous les composants CMS sur une face, et tous les composants traversants sur l'autre. Ainsi, nous savons que toutes les soudures se feront sur la même face (à l'exception des vias et de certaines broches des TL084 qui servent de vias).

La carte étant un circuit de commande, nous n'avions pas de contraintes de puissances ou de perturbations électromagnétiques majeures, la taille des pistes importait donc peu.

Certains de nos composants ne seront pas directement placés sur la carte mais reliés par des fils, comme les LEDs et le switch. Pour les réaliser, nous avons donc pris des packages ayant le même nombre de broches sans s'attarder sur les dimensions.

9.ii.b Modélisations finies

Maintenant que nous avons les règles de conception en tête, nous pouvons commencer le design du PCB.

Le PCB final est donné ci-dessous :

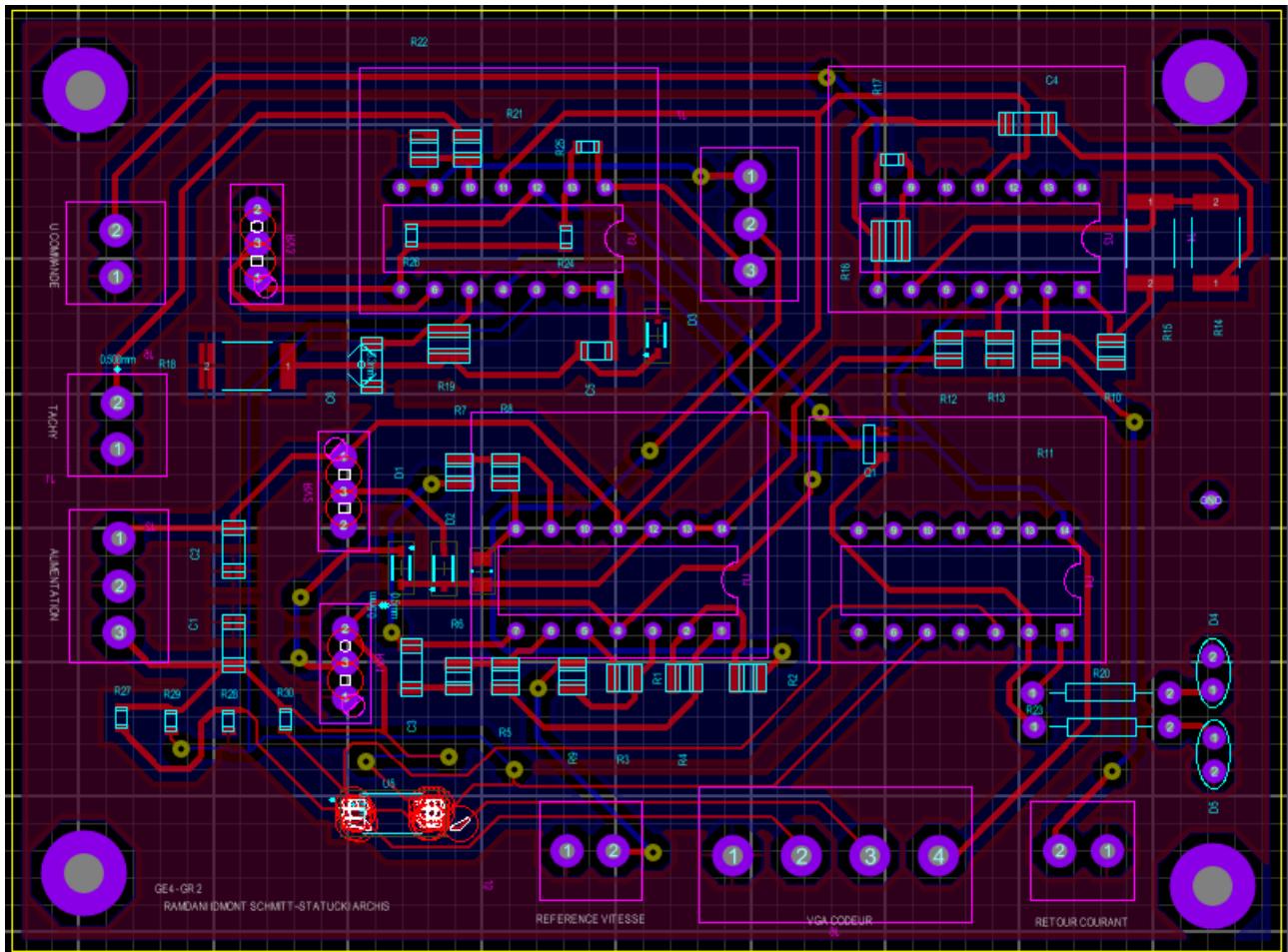


FIGURE 76 – Design du PCB avant impression

Remarquons que les règles de conception ont bien été respectées avec un écart précis entre chaque piste. Les seules exceptions aux règles se situent au niveau du comparateur et des potentiomètres, car les broches des composants elles-mêmes ne les respectent pas.

Concernant les TL084, nous avions d'abord opté pour des composants traversants, afin de pouvoir utiliser les broches de ces derniers comme des vias et ainsi limiter les perçages sur la carte. Cependant, au moment de la commande, il ne restait plus que leur version CMS sur Farnell. Ainsi, nous avons dû "transformer" ces composants CMS en composants traversants.

Lorsque l'on utilise des puces CMS comme nous l'avons fait pour les AOP, il est d'usage pour des projets de ce type de réaliser une carte de support à CMS pour simplifier la mise en place de ces derniers. C'est ce que nous avons fait :

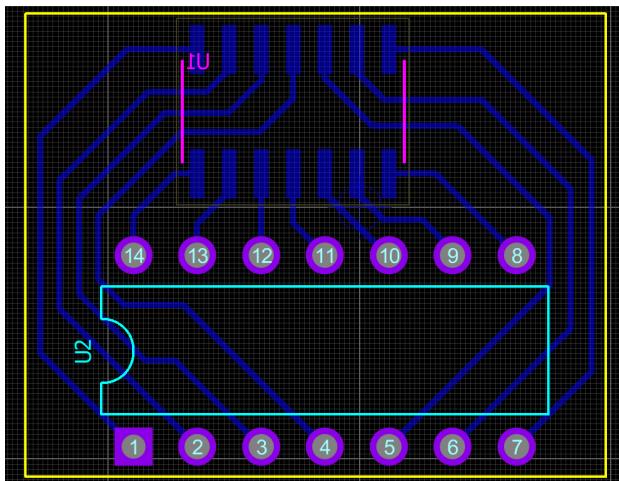


FIGURE 77 – Modélisation du PCB accueillant le composant CMS

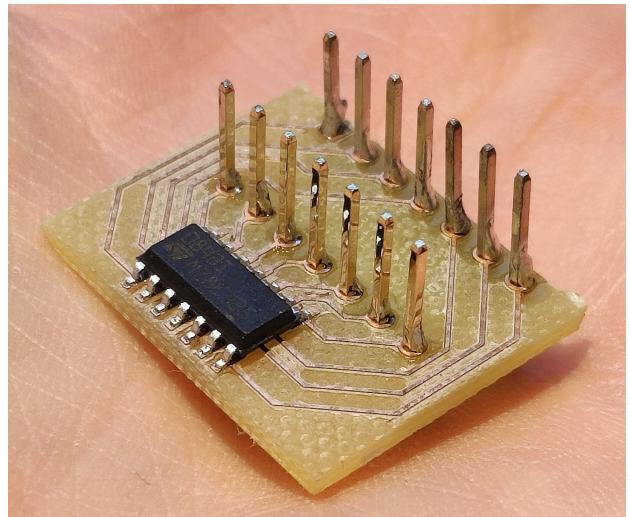


FIGURE 78 – PCB réel

La puce CMS étant trop large pour le support traversant, il a fallu la décaler et la placer à l'envers pour éviter un routage trop complexe.

De même, pour le connecteur VGA, nous avons réalisé une carte pour simplifier les connexions entre la carte et ce dernier :

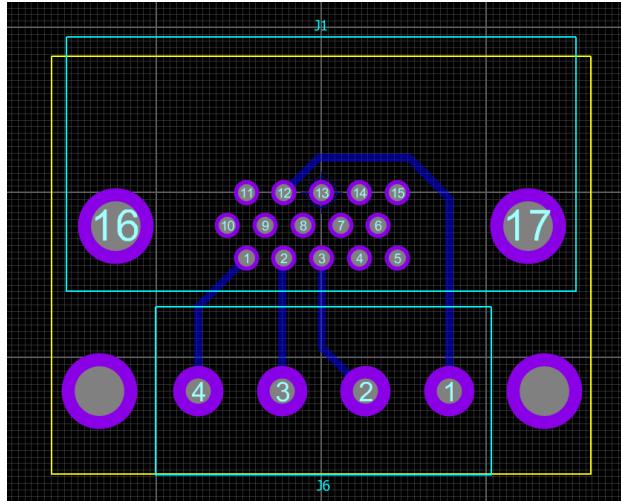


FIGURE 79 – Modélisation du PCB accueillant le connecteur VGA

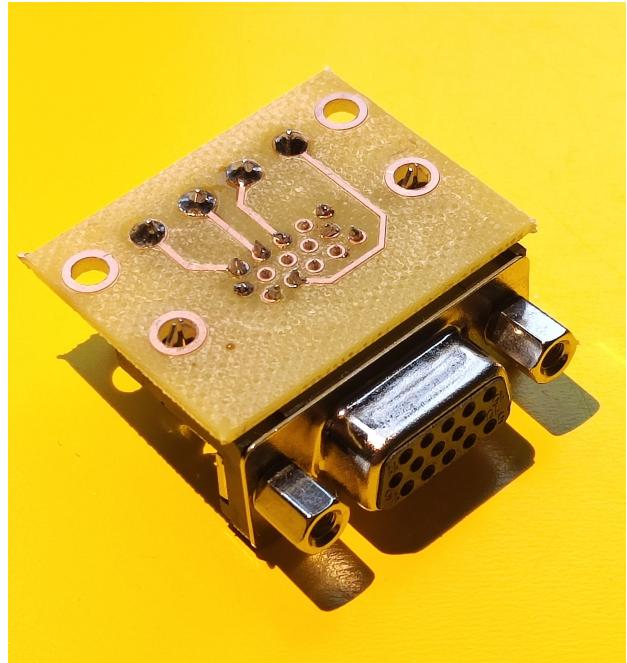


FIGURE 80 – PCB assurant le lien avec le connecteur VGA

Nous nous sommes aidés du schéma suivant, montrant comment router les différentes entrées sur les bons pins du connecteur :

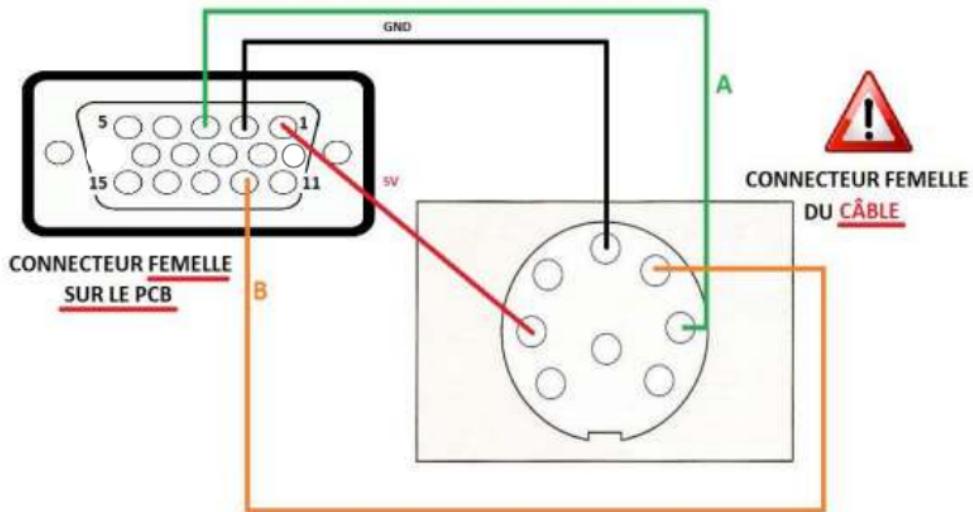


FIGURE 81 – Schéma de câblage du connecteur VGA

A noter que le codeur incrémental est alimenté en 15 V dans notre cas.

9.iii Soudures et Tests de la carte

Une fois la carte physique récupérée, nous avons dans un premier temps vérifié qu'aucun court circuit n'était présent. Nous n'en avons trouvé que deux, ce qui prouve que nous avons bien fait de mettre un écart minimal entre les pistes. Le cas échéant, il a suffit de gratter la piste avec un cutter pour résoudre le problème.

Ensuite, nous avons pu commencer les soudures.

Il fallait forcément commencer par les composants CMS car les traversants, munis de pièces en plastique, ne peuvent pas supporter les températures du four. Nous avons alors méthodiquement placé nos résistances, nos condensateurs, diodes, transistor et comparateur. Nous placions sur la carte d'abord toutes les résistances d'une certaine valeur avant de passer aux suivantes pour ne pas nous perdre dans les petits CMS. De plus, il fallait être vigilant au sens des diodes. Un trait fin sur le boîtier permet d'indiquer la cathode.

Ici, nous avons eu peu de problèmes. Seules deux résistances ont été mal soudées et donc ne faisaient pas le lien. Elles ont donc été ressoudées au fer et le problème était résolu.

Nous avons par la suite soudé les vias. Ceux-ci sont au nombre de 18 et ont été faits avec du fil issu de câbles multibrins.

Enfin, nous sommes passés aux composants traversants. La seule difficulté que nous avons rencontrée fut au niveau des supports des AOPs. En effet, nous ne disposions que d'un nouveau modèle dont les broches métalliques ne sont pas accessibles par le dessus. Ceci nous posa problème compte tenu que certaines soudures des pattes d'AOP devaient être faites des deux côtés de la carte.

Nous nous en sommes sortis en récupérant des supports sur d'anciens projets.

Voici à quoi ressemble la carte terminée :

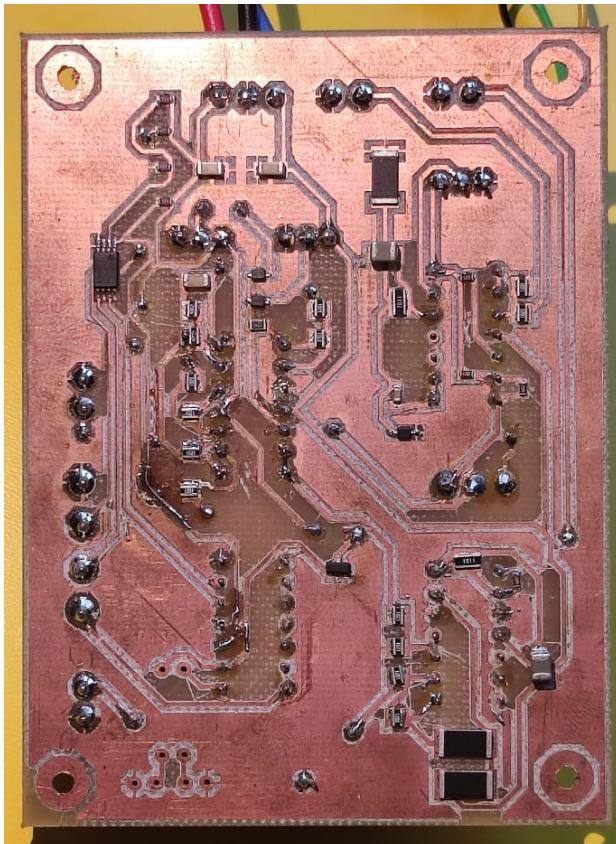


FIGURE 82 – Face CMS

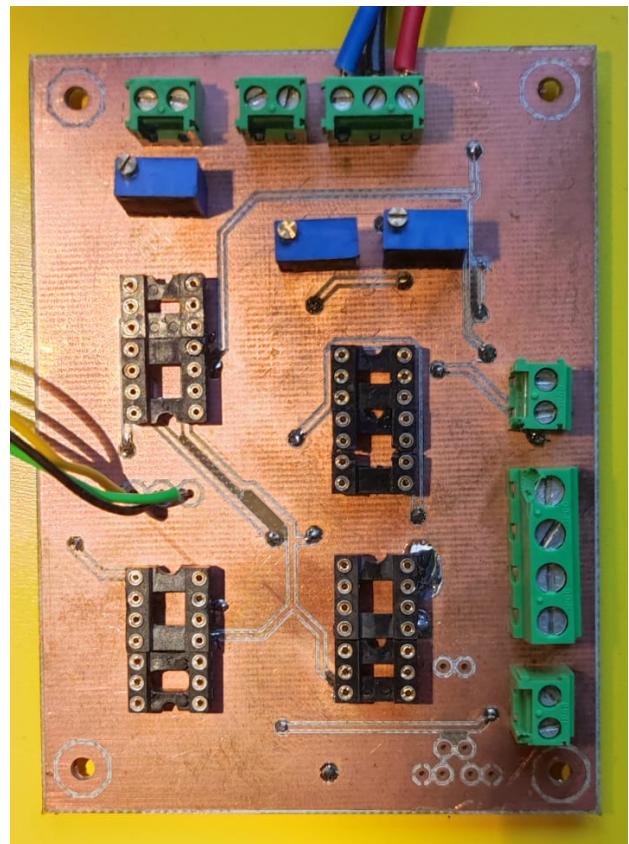


FIGURE 83 – Face traversants

Les PCB servant de support à la puce CMS vus précédemment se placent sur 3 des 4 supports. Trois fils sont également reliés à l'interrupteur pour changer de capteur de vitesse.

9.iv Tests préliminaires

Nous arrivons vers la fin de notre projet et nous devons maintenant réaliser certains tests préliminaires pour nous assurer du bon fonctionnement de la carte.

Nous allons tout d'abord nous assurer du bon fonctionnement du circuit d'asservissement par retour tachymétrique, puis nous vérifierons l'acquisition du signal issu du codeur incrémental.

Pour ce faire, nous avons tout d'abord simulé sur PSIM ce qu'il se passerait si nous court circuitions les condensateurs des correcteurs intégrales. Ainsi, nous avions un ordre d'idée de ce à quoi nous devions nous attendre. Voici la simulation PSIM :

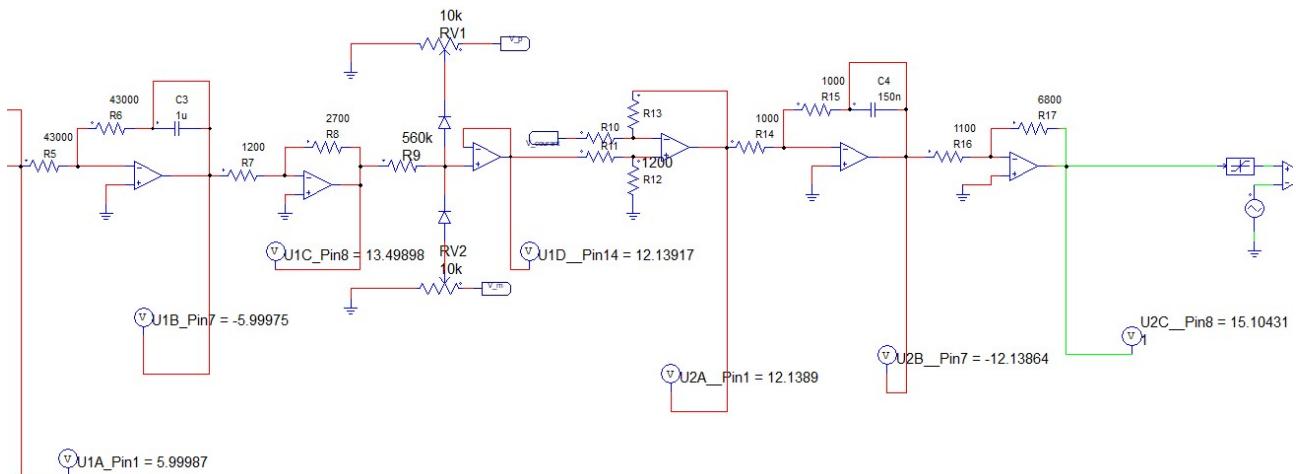


FIGURE 84 – Schéma de test préliminaire sur PSIM

Ensuite, nous avons réellement court-circuité ces condensateurs à l'aide d'une patte de résistance qui reliait leur borne. En plaçant en entrée un signal constant, nous avons pu mesurer la tension sur certains points importants et comparer à la simulation.

Si les valeurs étaient cohérentes avec PSIM, cela signifiait que notre carte fonctionnait correctement et que nous pouvions donc passer aux tests finaux.

Après avoir vérifié l'asservissement avec la dynamo tachymétrique sur le banc d'essai, nous avons pu nous assurer que l'acquisition du signal codeur ainsi que son traitement se passait comme convenu.

Pour cela, il a d'abord fallu tester la correspondance des entrées du câble VGA avec le connecteur branché sur le codeur selon le schéma de câblage.

Pour cela, nous devions vérifier au niveau de l'interrupteur que le signal issu du codeur était similaire à celui issu de la dynamo tachymétrique.

Ici, contrairement à la dynamo., nous avons rencontré plusieurs problèmes avant de pouvoir asservir correctement. Tout d'abord, le codeur incrémental n'était pas alimenté par le +15V. Nous nous sommes rendus compte qu'il s'agissait d'une soudure mal faite au niveau du bornier du VGA, il nous a alors suffi de la refaire.

Ensuite, nous avons remarqué que les signaux issus du codeur au niveau du comparateur n'avaient pas la forme en crêteau attendu. Ce problème venait de l'absence de résistances de pull-up en sortie du codeur. Nous les avons alors rajoutées ce qui a réglé le problème.

Enfin, une soudure mal faite au niveau de la bascule D empêchait l'acquisition du signal B du codeur. Nous l'avons refaite et avons finalement pu observer un signal similaire à celui issu de la dynamo. au niveau de l'interrupteur.

Maintenant que nous avons mis en accord les résultats théoriques de simulation et pratiques, nous pouvons passer aux vrais tests sur le banc d'essai.

Arborescence des fichiers par figure :

Si le visualisateur de PDF utilisé est Adobe Acrobat, alors il est possible d'accéder au fichier directement en cliquant sur le lien. Autrement, il faudra suivre l'arborescence.

- Figure (4) :[donnees_moteur.m](#)
- Figure (5) :[/Modelisation et simulation du systeme en boucle ouverte/MATLAB/simu_moteur_a_vide_SIMULINK.slx](#)
- Figure (6) :[/Modelisation et simulation du systeme en boucle ouverte/PSIM/Moteur_a_vide_PSIM.psimsch](#)
- Figure (7) :[/Modelisation et simulation du systeme en boucle ouverte/PSIM/Moteur_a_vide_PSIM.psimsch](#)
- Figure (8) :[/Modelisation et simulation du systeme en boucle ouverte/MATLAB/Comparaison_courant_psim_MATLAB.m](#)
- Figure (9) :[/Modelisation et simulation du systeme en boucle ouverte/MATLAB/Comparaison_vitesse_psim_MATLAB.m](#)
- Figure (11) :[/Frottements et charges/PSIM/Simulation_frottement_et_charges.psimsch](#)
- Figure (12) :[/Frottements et charges/MATLAB/Simulation_frottement_et_charges.slx](#)
- Figure (13) :[/Frottements et charges/MATLAB/Comparaison_courant_et_vitesse_frottements_et_charges.m](#)
- Figure (14) :[/Frottements et charges/MATLAB/Comparaison_courant_et_vitesse_frottements_et_charges.m](#)
- Figure (15) :[/Frottements et charges/MATLAB/Comparaison_courant_et_vitesse_frottements_et_charges.m](#)
- Figure (16) :[/Frottements et charges/MATLAB/Comparaison_courant_et_vitesse_frottements_et_charges.m](#)
- Figure (19) :[/Systeme complet avec hacheur/PSIM/Systeme_complet_BO.psimsch](#)
- Figure (21) :[/Prise en compte de la commande/PSIM/Systeme_complet_BO.psimsch](#)
- Figure (22) :[/Prise en compte de la commande/MATLAB/Systeme_complet_avec_commande.slx](#)
- Figure (23) :[/Systeme complet avec hacheur/PSIM/Systeme_complet_BO.psimsch](#)
- Figure (24) :[/Systeme complet avec hacheur/PSIM/Systeme_complet_BO.psimsch](#)
- Figure (25) :[/Systeme complet avec hacheur/PSIM/Systeme_complet_BO.psimsch](#)
- Figure (26) :[/Systeme complet avec hacheur/PSIM/Systeme_complet_BO.psimsch](#)
- Figure (29) :[/Asservissement en courant/MATLAB/comparaison_FT_courant.slx](#)
- Figure (30) :[/Asservissement en courant/MATLAB/comparaison_FT_courant.slx](#)
- Figure (31) :[/Asservissement en courant/MATLAB/asserv_courant.slx](#)
- Figure (32) :[/Asservissement en courant/MATLAB/asserv_courant.slx](#)
- Figure (33) :[/Asservissement en courant/PSIM/Boucle_courant_asservie_hacheur.psimsch](#)
- Figure (34) :[/Asservissement en courant/PSIM/Boucle_courant_asservie_hacheur.psimsch](#)
- Figure (35) :[/Asservissement de vitesse/MATLAB/Fonction_de_transfert_courant_et_vitesse_MATLAB.m](#)
- Figure (36) :[/Asservissement de vitesse/MATLAB/Comparaisons_asservissement_de_vitesse_SIMULINK.slx](#)
- Figure (39) :[/Asservissement de vitesse/MATLAB/Comparaisons_asservissement_de_vitesse_SIMULINK.slx](#)
- Figure (40) :[/Asservissement de vitesse/PSIM/Systeme_complet_asservie_hacheur.psimsch](#)
- Figure (41) :[/Asservissement de vitesse/PSIM/Systeme_complet_asservie_hacheur.psimsch](#)