

الجمهورية الجزائرية الديمقراطية الشعبية
République algérienne démocratique et populaire
وزارة التعليم العالي والبحث العلمي
Ministère de l'enseignement supérieur et de la recherche scientifique
المركز الجامعي لعين تموشنت
Centre Universitaire Belhadj Bouchaib d'Ain-Temouchent
Institut de Technologie
Département de Génie Electrique



Projet de fin d'études
Pour l'obtention du diplôme de Master en :
Domaine : **SCIENCE ET TECHNOLOGIE**
Filière : **ELECTROTECHNIQUE**
Spécialité : **COMMANDE ELECTRIQUE**

Thème

**Contrôle du sens de la vitesse d'un moteur
à courant continu sous interface graphique
(GUI MATLAB)**

Présenté par :

- ABDALLAH ISMAIL
- AHMED BELKACEM ILYES

Devant les jurys composés de :

Dr AISSOU MASSINISSA	C.U.B.B (Ain Temouchent)	Président
Dr ATTOU AMINE	C.U.B.B (Ain Temouchent)	Examineur
Mdm TOUHAMI AICHA	C.U.B.B (Ain Temouchent)	Encadrante
Dr GHOURAF DJAMEL EDDINE	E.N.P.O (Oran)	Co-encadrant

Année universitaire : 2019/2020

Remerciements

Tous d'abord je tiens à remercier ALLAH, le tout puissant, de
m'avoir donné la santé, la volonté et la patience

Pour réaliser ce modeste travail

Je tiens à remercier vivement mon encadreur :

MADAME, TOUHAMI AICHA et Co-encadreur

Monsieur Ghouraf Djamel Eddine pour ses conseils

pertinents, ses orientations judicieuses, sa patience et

par ses suggestions et critiques qui

ont grandement facilité ce travail.

Mes remerciements s'adressent également à tous les membres

de jury, qui ont accepté de nous honorer de leur

Présence et de juger ce notre travail.

Je tiens à remercier tous ceux qui ont participé de près ou de loin à

l'élaboration de ce mémoire

Dédicace

Avant tout, je dois remercier Dieu le tout puissant qui m'a donné

L'envie et la force pour mener à terme ce travail.

Je tiens à dédier ce mémoire

À mes très chers parents, source de vie, d'amour et d'affection.

À mes chers frères, source de joie et de bonheur.

À ma Famille, source d'espoir et de motivation.

À mon binôme AHMED BELKACEM ILYES.

À tous mes Chers amis.

À tous que j'aime et à tous qui m'aime.

Et à tous ceux qui ont contribué de près ou de loin pour que ce

projet soit possible, je vous dis merci.

ABDALLAH ISMAIL

Dédicace

Tous d'abord je remercie ALLAH pour le courage et la patience
qu'il m'a donné afin de mener.

On dédie ce travail

A nos chers parents Pour leur soutien, leur patience, et leur
sacrifice, vous méritez tout éloge.

On espère être l'image que vous êtes fait de nous, que dieu vous
garde et vous bénisse.

On dédie aussi ce travail à nos chers frères et sœurs, pour leur
affection et leur encouragement qui ont toujours été pour nous des
plus précieux.

Et je voudrais dédier aussi ce travail à mon binôme «ABDALLAH
ISMAIL» et tous mes amis.

AHMED BELKACEM ILYES

Liste des symboles

$U(t)$: Tension appliquée au moteur.

$i(t)$: Intensité traversant le moteur.

C_e (T_e) : Couple moteur couple électromagnétique.

T_U : couple utile.

f : Coefficient de frottement visqueux.

K_m : Constante de couple électromagnétique.

K_e : Constante de Force contre électromotrice.

$E(t)$: Force contre électromotrice.

$\Omega(t)$: Vitesse de rotation du rotor.

C_r : Couple résistant.

α : rapport cyclique.

Φ : le flux magnétique

J : Moment d'inertie de l'axe du rotor.

F : force de Laplace.

B_s : champ magnétique.

P_a : puissance absorbée.

P_u : puissance utile.

P_{em} : puissance électromagnétique.

P_{JS} : pertes par effet joule dans l'inducteur.

P_{JR} : pertes par effet joule dans l'induit.

P_C : pertes collectif (pertes fer + pertes mécanique : dites pertes constantes).

V_{smoy} : valeur moyenne de tension.

R_a : résistance de l'induit.

L_a : Inductance de l'induit.

L_f : Inductance de l'inducteur.

R_f : résistance de l'inducteur.

T : période.

P : nombre de paires de pôles de la machine.

N : nombre de conducteurs actifs de la périphérie de l'induit.

a : nombre de paires de voies de l'enroulement entre les deux balais.

n : fréquence de rotation de l'induit (en t/s).

η : Rendement.

f : fréquence

$S(p)$: grandeur de (sortie)

$E(p)$: grandeur d'entrée.

$G(p)$: fonction de transfert.

$\varepsilon(p)$: erreur.

$C(p)$: fonction de transfert du régulateur.

$G(p)$: processus à commander (moteur à courant continu).

$E(p)$: grandeur d'entrée.

$e(t)$: signal d'entrée.

K_p : gain Proportionnel.

K_i : gain Intégral.

K_d : gain Dérivative.

Liste des abréviations

MCC : Moteur à courant continu

G : génératrice

M : moteur

DC-DC : Convertisseur continu - continu.

FEM : Force électromotrice.

MLI : Modulation de Largeur d'Impulsions.

MOSFET : Metal Oxide Semi-Conductor Field Effect Transistor.

GTO : Gate Turn- Off Thyristor.

HDGTO : Hard Driven GTO.

GCT : Gate-Commutated Thyristor.

IGBT : Insulated Gate Bipolaire Transistor.

Chapitre I généralités sur la machine à courant continue

Introduction générale	1
I.1-Introduction	4
I.2 Description générale du Système étudié.....	4
I.3 Constitution et principe de fonctionnement du MCC.....	4
I.3.1 Constitution de la machine à courant continu.....	4
I.3.2 Principe de fonctionnement de la machine.....	6
I.4 Les différents types de moteurs à courant continu.....	6
I.5 Domaine d'utilisation de moteur à courant continu.....	9
I.6 Avantages et inconvénients de MCC.....	9
I.6.1 Avantages	9
I.6.2 Inconvénients	10
I.7 méthodes utilisées pour varier la vitesse d'un moteur à courant continu.....	10
I.7. 1 Les dispositifs électroniques utilisés pour la variation de vitesse.....	10
I.8 principe et commande d'inversion de sens de rotation d'un moteur a courant continu.....	14
I.8.1Principe de l'inversion de sens.....	14
I.8.2 commande d'inversion de sens de rotation d'un moteur à courant continu.....	14
I.9 Commande de vitesse d'un moteur à courant continu par les techniques de commande avancé et les techniques classiques	16
I.9.1 les techniques de commande avancé	16
I.9.2 les techniques classiques	18
I.9.3 La différence entre les techniques de commande avancée et les techniques classiques	19
I.10 Conclusion	20

Chapitre II Développement et programmation des interfaces graphiques

II.1 Introduction22

II.2 Interaction homme-machine22

 II.2.1 Définition Interaction homme-machine.....22

 II.2.2 l'intérêt d'interaction homme-machine.....22

 II.2.3 l'utilité d'interaction homme-machine.....23

II.3 langages de programmations des interfaces graphiques23

 II.3.1 Boites à outils et Langages de programmation 23

II.4 Création d'une interface graphique sous MATLAB GUIDE.....25

 II.4.1 Définition Logiciel MATLAB 25

 II.4.2 Présentation du MATLAB GUIDE (GUI MATLAB).....25

 II.4.3 Création d'une interface graphique sous MATLAB GUIDE.....26

II.5 Les composants des éléments visuels.....27

II.6 Présentation d'un exemple réalisé dans MATLAB GUIDE.....30

 II.6.1 Programmation33

II.7 Conclusion36

Chapitre III Modélisation et simulation d'un MCC sous interface graphique

III.1 Introduction38

III.2 Implémentation du MCC sous SIMULINK-MATLAB38

 III.2.1 Modélisation et simulation du système sous SIMULINK-MATLAB38

 III.2.1.1 Modélisation de la source triphasée38

 III.2.1.2 Modélisation du redresseur 39

 III.2.1.3 Modélisation du filtre 40

 III.2.1.4 Modélisation hacheur41

 III.2.1.5 Modélisation du générateur d'impulsions :41

III.3 Notion de Boucle ouverte/Fermée 42

 III.3.1 Commande en boucle ouverte42

 III.3.2 Commande en boucle fermée43

III.4 Synthèse du régulateur PI45

 III.4.1 spécifications du cahier de charge45

 III.4.2 Détermination des gains de régulateur PI46

III.5 Modélisation de la MCC47

III.6 Résultats des simulations et interprétation49

 III.6.1 Simulation de la commande en boucle ouverte50

 III.6.1.1 Interprétations des résultats de simulation en boucle ouverte52

 III.6.2 Simulation de la commande en boucle fermée53

 III.6.2.1 Interprétations des résultats de simulation en boucle fermée56

III.7 Présentation et étude sous interface graphique GUI réalisé sous MATLAB56

 III.7.1 Présentation de l'interface graphique réalisée du système étudié.....57

 III.7.2 Etude sous interface graphique GUI réalisé sous MATLAB58

 III.7.2.1 Commande en boucle ouverte.....58

 III.7.2.2 Commande en boucle fermée.....63

 III.7.2.3 Commande de variation de sens de vitesse d'un moteur à courant continu....66

III.8 Interprétations des résultats de simulation67

 III.8.1 Interprétations des résultats de simulation en boucle ouverte.....67

III.8.2 Interprétations des résultats de simulation en boucle fermée.....	68
III.9 Conclusion	69
Conclusion générale.....	71
Références et bibliographie	73

Liste des figures

Figure (I.1) : Constitution d'une machine à courant continu	5
Figure (I.2) : Machine à excitation séparée.....	7
Figure (I.3) : Machine à excitation shunt.....	8
Figure (I.4) : Machine à excitation série.....	8
Figure (I.5) : Machine à excitation composé.....	9
Figure (I.6) : Principe de pont H.....	14
Figure (I.7) : Inversion du sens de rotation du moteur.....	15
Figure (I.8) : Pont en H avec des transistors bipolaires.....	15
Figure (II.1) : GUIDE MATLAB.....	26
Figure (II.2) : Présentation des outils d'une interface graphique sous GUIDE MATLAB.....	26
Figure (II.3) : Les propriétés d'un texte statique.....	31
Figure (II.4) : Calculatrice graphique et scientifique sous MATLAB GUIDE.....	32
Figure (II.5) : Exécution d'une calculatrice graphique et scientifique sous GUIDE MATLAB...35	
Figure III.1 : Schéma de principe d'un MCC.....	38
Figure III.2 : Schéma bloc de la source triphasée sous Simulink.	39
Figure III.3 : Schéma bloc du redresseur sous Simulink	39
Figure III.4 : Schéma du filtre passe bas	40
Figure III.5 : Schéma de filtre Butterworth	40
Figure III.6 : Schéma bloc du hacheur sous Simulink	41
Figure III.7 : Schéma du principe de génération des impulsions	41
Figure III.8 : Générateur d'impulsion	42
Figure III.9 : Commande en boucle ouverte	43
Figure III.10 : Commande en boucle fermée	44

Figure III.11 : Structure du système en boucle fermée	45
Figure III.12 : Schéma bloc du régulateur PI sous Simulink	47
Figure III.13 : Schéma block d'un MCC à excitation séparé	49
Figure III.14 : Schéma bloc global du système étudié sous Simulink	49
Figure III.15 : Simulation de la commande en boucle ouverte	50
Figure III.16 : Réponse indicielle en vitesse	50
Figure III.17 : Réponse indicielle en courant	51
Figure III.18 : Réponse indicielle en couple	51
Figure III.19 : Réponse indicielle en vitesse	51
Figure III.20 : Réponse indicielle en courant	52
Figure III.21 : Réponse indicielle en couple	52
Figure III.22 : Simulation de la commande en boucle fermé	53
Figure III.23 : Réponse indicielle en vitesse	54
Figure III.24 : Réponse indicielle en courant	54
Figure III.25 : Réponse indicielle en couple	54
Figure III.26 : Réponse indicielle en vitesse	55
Figure III.27 : Réponse indicielle en courant	55
Figure III.28 : Réponse indicielle en couple	55
Figure III.29 : Interface graphique réalisé sous GUI MATLAB pour visualiser les résultats de simulation	57
Figure III.20 : Simulation de l'interface graphique réalisée de la commande en boucle ouvert à vide.....	59
Figure III.31 : Résultats de simulation de l'interface graphique réalisée de la commande en boucle ouvert à vide	60
Figure III.32 : Simulation de l'interface graphique réalisée de la commande en boucle ouvert en charge.....	60

Figure III.33 : Résultats de simulation de l'interface graphique réalisée de la commande en boucle ouvert à vide.....	61
Figure III.34 : Simulation de l'interface graphique réalisée de la commande en boucle ouvert en charge.....	62
Figure III.35 : Résultats de simulation de l'interface graphique réalisée de la commande en boucle ouvert en charge.....	63
Figure III.36 : Simulation de l'interface graphique réalisée de la commande en boucle fermée à vide.....	64
Figure III.37 : Résultats de simulation de l'interface graphique réalisée de la commande en boucle fermée à vide.....	64
Figure III.38 : Simulation de l'interface graphique réalisée de la commande en boucle fermée en charge.....	65
Figure III.39 : Résultats de simulation de l'interface graphique réalisée de la commande en boucle fermée en charge.....	65
Figure III.40 : Simulation de l'interface graphique réalisée de la variation de sens de vitesse.....	66

Liste des tableaux

Tableau I.1 : Les différents types des hacheurs	11
Tableau I.2 : Les différents types des redresseurs	13

Introduction générale

Introduction Générale

De nos jours, du fait de l'utilisation croissante du moteur à courant continu dans les applications industrielles. Malgré ses inconvénients liés à son entretien, son poids, son prix et la présence du système balais- collecteur. Il possède l'avantage d'être facilement commandable en tension grâce au découplage naturel entre ses grandeurs : flux et couple.

Le moteur à courant continu MCC est une solution technologique possible pour réaliser la fonction de convertir une énergie électrique en énergie mécanique, et vice versa.

Pour les applications industrielles exigeant une régulation précise de la vitesse, on a d'abord utilisé des moteurs à courant continu commandés par des variateurs électroniques à semi-conducteurs. Cette technique consistait à faire varier la vitesse proportionnellement à la tension.

Dans le cas d'un Moteur à courant continu, plusieurs recherches portant sur la commande de vitesse et l'inversion de sens.

La régulation d'un moteur à courant continu se fait par plusieurs méthodes, parmi eux la méthode de régulation idéale, elle est utilisée dans l'industrie car elle permette de régler à l'aide de ses paramètres les performances (stabilité, précision, temps de réponse,...etc.) d'une régulation d'un processus modélisé par un premier ordre.

La liaison d'un moteur à courant continu à l'opérateur se fait par une interface graphique, cette dernière permettant de connecter une personne à une machine, à un système ou à un appareil, en théorie, il est donc possible d'utiliser ce terme pour définir n'importe quel écran permettant à un utilisateur d'interagir avec un appareil. Cependant, il est généralement utilisé pour le contexte d'un processus industriel.

Notre but est de faire une étude complète et détaillée d'un moteur à courant continu qui consiste la modélisation et la simulation pour commander la vitesse et le sens de vitesse d'un MCC.

Pour atteindre ce but, notre travail est articulé autour de trois chapitres :

- En premier chapitre nous avons fait la description générale du système étudié et présenté la constitution de la machine à courant continu et leur principe de fonctionnement, il s'agit d'établir les différents types des moteurs à courant continu. Ensuite nous donnerons les avantages et les inconvénients, le domaine d'utilisation. Enfin nous avons présenté les méthodes utilisées pour varier la vitesse, le sens de rotation et les techniques de commande de vitesse du MCC.

- Le deuxième chapitre est consacré à le développement et programmation des interfaces graphiques au début nous avons présenté l'interaction homme machine, son intérêt et son utilité dans l'industrie, après nous avons mentionné les langages de programmation des interfaces graphiques. Ensuite nous avons défini le logiciel MATLAB et présenté GUI MATLAB d'une manière générale pour avoir une idée de son fonctionnement. Enfin et à base des notions de création d'une interface graphique on a présenté dans la dernière partie un exemple d'une interface graphique d'une calculatrice graphique et scientifique qui répond aux exigences demandées avec sa programmation.
- Le troisième chapitre est consacré à la modélisation et la simulation d'un moteur à courant continu sous MATLAB SIMULINK et GUI MATLAB ensuite nous avons fait une étude sous interface graphique GUI réalisé sous MATLAB et montré les notions de boucle ouverte et fermé, Enfin nous avons obtenus les résultats des simulations et fait les interprétations.

Nous terminons par une conclusion générale.

Chapitre I

Généralités sur les moteurs à courant continu

I.1 Introduction

La machine à courant continu est un convertisseur d'énergie, totalement réversible, elle peut fonctionner soit en moteur, convertissant de l'énergie électrique en énergie mécanique, soit en génératrice, convertissant de l'énergie mécanique en énergie électrique.

Les moteurs à courant continu sont utilisés dans les asservissements ou la facilité de réglage de la vitesse, du moment, du couple et du sens de rotation permet une mise en œuvre aisée.

Le présent chapitre présentera les différents éléments constitué de la machine électrique à courant continu, le principe de fonctionnement de cette machine et les techniques de commande et réglage de vitesse. [1]

I.2 Description générale du système étudié

Un système électrique désigne l'ensembles des activités liées à l'industrie électrique ,la production d'électricité ,son transport, sa distribution aux consommateurs, notre système étudié il s'agit d'un convertisseur électromécanique permettant la conversion bidirectionnelle d'énergie entre une installation électrique parcourue par un courant continu et un dispositif mécanique, d'où l'énergie électrique est transformée en énergie mécanique contrôlé et suivi par un système de commande.

Il contient quatre systèmes interconnectés :

- Source triphasé.
- Convertisseurs statiques (redresseurs et hacheurs).
- Moteur à courant continu.
- Régulateur.

I.3 Constitution et principe de fonctionnement du MCC

La machine à courant continu est réversible ; c'est-à-dire que la constitution d'une génératrice est identique à celle d'un moteur.

I.3.1 Constitution de la machine à courant continu

Elle se compose de deux parties :

- a. Une partie fixe (stator), appelée inducteur, qui produit le flux magnétique nécessaire à la magnétisation de la machine.
- b. Une partie mobile (rotor), appelée induit, qui porte les conducteurs soumis au flux.

a. Parties fixe (inducteur)

C'est la partie fixe, Il est formé soit d'aimants permanents en ferrite soit de bobines placées autour des noyaux polaires. Lorsque les bobines sont parcourues par un courant continu, elles créent un champ magnétique dans le circuit magnétique de la machine notamment dans l'entrefer, espace séparant la partie fixe et la partie mobile.

La partie fixe se comporte des éléments suivants :

➤ **La carcasse (ou culasse)**

Elle forme la partie extérieure qui supporte les différentes parties de la machine, elle renferme le champ magnétique.

➤ **Les pièces polaires**

Les pièces polaires sont constituées de tôles feuilletées (pour réduire les pertes pas courant de Foucault) et permettent de supporter les bobines d'excitation de la machine. Elles permettent donc de produire un champ d'induction.

➤ **L'enroulement d'excitation**

L'enroulement d'excitation de la machine à courant continu sert à la création du champ magnétique.

b. Partie mobile (induit)

L'induit est la partie tournante de la machine, le noyau d'induit est en fer pour canaliser les lignes de champ, les conducteurs sont logés dans des encoches sur le rotor, deux conducteurs forment une spire.

La partie mobile se comporte des éléments suivants :

➤ **Le collecteur**

Il se compose de lames de cuivre isolées les une des autres, où sont reliées les extrémités du bobinage de l'induit.

➤ **Les balais**

Ce sont des tiges conductrices qui frottent sur la surface du collecteur, ils sont faits en graphite ou en charbon.

Le dispositif collecteur/ balais permet donc de faire circuler un courant dans l'induit [2].

La figure suivante présente la constitution d'une machine à courant continu :

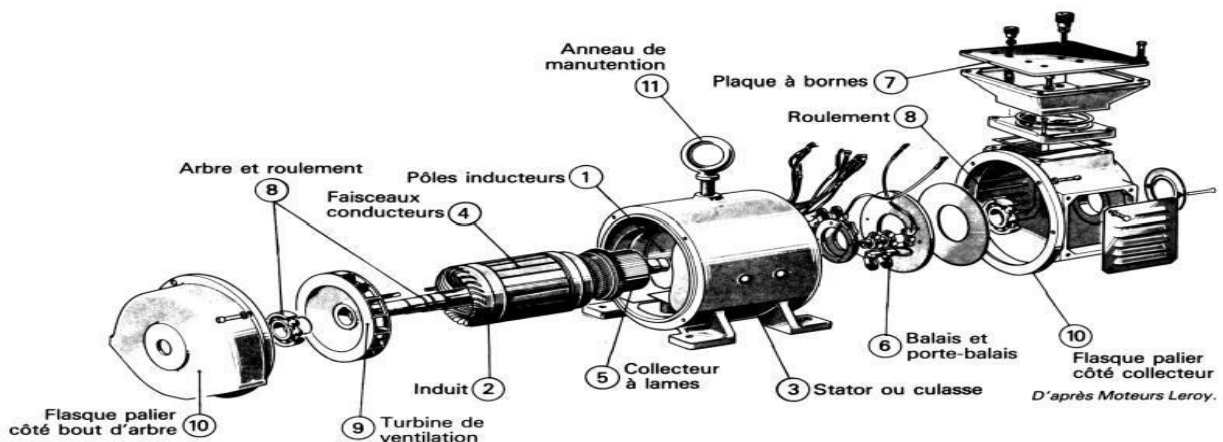


Figure. I.1 : Constitution d'une machine à courant continu. [3]

I.3.2 Principe de fonctionnement de la machine

La machine à courant continu, comme toutes les autres machines électriques, peut fonctionner en moteur ou en générateur.

- **Fonctionnement en génératrice**

Dans le cas d'une génératrice à excitation indépendante. Le rôle d'un générateur et de fournir de l'électricité sous tension continue variable.

Pour cela, il faut lui fournir de l'énergie mécanique sous forme d'un couple qui entraîne la rotation de la machine et également de l'électricité pour alimenter l'excitation, qui crée le champ magnétique. Comme l'enroulement d'excitation est fixe (situé au stator) et parcouru par un courant, le champ qu'il crée est de direction constante. De plus, le couple extérieur fait tourner le rotor.

L'interaction d'un champ fixe et d'un conducteur mobile provoque l'apparition d'une fem induite (E) aux bornes de l'enroulement d'induit. Cette force électromotrice (f.e.m) est recueillie et redressée par les balais qui frottent sur le collecteur.

Selon l'amplitude du champ créé par l'excitation (et donc selon la valeur du courant d'excitation), la tension induite sera d'amplitude variable [2], [4].

- **Fonctionnement en moteur**

Pour comprendre son principe de fonctionnement, nous allons considérer que l'inducteur et l'induit sont électriquement séparés. Nous sommes donc dans le cas d'un moteur à excitation indépendante.

On alimente l'excitation du moteur. Celui-ci crée un champ d'induction de direction constant. De plus, on fait circuler un courant dans le rotor.

En vertu du principe de Laplace, l'interaction du courant avec un champ qui est lui perpendiculaire crée un couple sur l'enroulement du rotor. Ce couple entraîne la rotation du moteur comme l'indique la formule suivante [2], [4] :

$$F = I a \cdot dl \wedge B \quad (I.1)$$

I.4 Les différents types de moteurs à courant continu [5]

On distingue deux types de moteurs à courant continu :

- **Les moteurs à inducteur à aimant permanent**

Dans le moteur à aimants permanents, le rotor et le stator sont constitués d'aimants dont les pôles identiques sont mis face à face afin de permettre le mouvement du moteur grâce au champ magnétique créé. Les pôles identiques se repoussent ce qui met le moteur en mouvement.

Le principal avantage de ce type de moteur réside dans sa durée de vie élevée qui correspond à celle des aimants (environ 400 ans). En revanche, les moteurs à aimants permanent peuvent être encombrants lorsque l'on veut obtenir des moteurs puissants car il faut des aimants plus puissants.

- **Les moteurs à inducteur bobiné**

Il existe 4 types différents de moteurs électriques qui sont classés en fonction du type d'excitation qui est employé, qui sont :

- a. Le moteur à excitation séparée.
- b. Le moteur à excitation shunt.
- c. Le moteur à excitation série.
- d. Le moteur à excitation composée.

a. Moteur à excitation séparée

Pour ce moteur deux alimentations sont nécessaires: une pour l'inducteur et une pour l'induit. Ici $E = U - RI = K \Phi \Omega$ (I.2), donc la vitesse de ce moteur est $\Omega = (U - RI) / K \Phi$ (I.3) et sa puissance est $P_u = P_{abs} - P_{je} - P_j - P_c$ (I.4) avec $P_{abs} = U \cdot I + U_e \cdot I_e$ (I.5) la puissance absorbée, $P_{je} = U_e \cdot I_e$ la puissance perdue par effet joule de l'inducteur, $P_j = R I^2$ (I.6) la puissance perdue par effet joule de l'induit et les pertes collectives.

Ainsi, son rendement est $\eta = P_u / P_{abs}$ (I.7).

La vitesse de ce moteur est réglable par tension et est indépendante de la charge. De plus, elle fournit un couple important à faible vitesse ce qui peut être utile pour les machines-outils ou le levage.

La figure suivante présente le modèle d'une machine à excitation séparée :

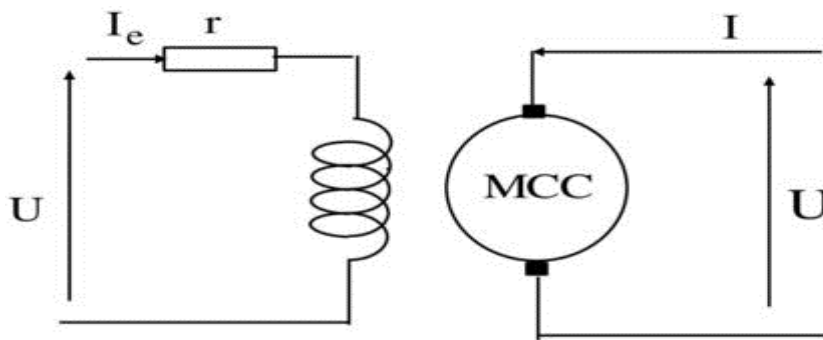


Figure. I.2 : Machine à excitation séparée.

b. Moteur à excitation shunt

Le moteur à excitation shunt ou parallèle est composé de deux enroulements : un enroulement autour du stator et un autour du rotor. Ils sont connectés en parallèle et sont donc parcourus par des courants d'amplitudes différentes.

La figure suivante présente le modèle d'une machine à excitation shunt :

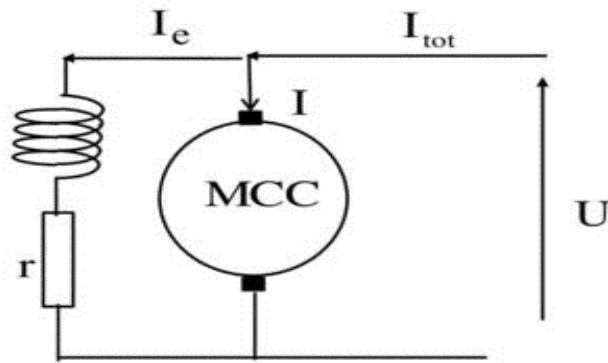


Figure. I.3 : Machine à excitation shunt.

Ce moteur présente beaucoup d'avantages. Sa vitesse est sensiblement constante et facile à régler. De plus, il suffit de changer le sens du courant pour changer sa vitesse de rotation. Ses utilisations sont diverses : machines outils, appareils de lavage, pompes, ventilateurs, etc....

Propriétés de ce couplage :

- Circuit inducteur et induit, alimenté sous la même tension.
- Proportionnalité entre fem et vitesse de rotation.
- Si U augmente, la vitesse augmente.
- Ne jamais couper le courant dans l'induit sinon le moteur s'emballe, en effet la vitesse du moteur est inversement proportionnelle à la tension dans le circuit induit.

c. Moteur à excitation série

Ce type de moteur est caractérisé par le fait que le stator (inducteur) est raccordé en série avec le rotor (induit), ainsi la même tension traverse le stator et le rotor. Par conséquent, le sens de rotation ne change pas avec le sens du courant; il suffit de brancher le rotor et le stator dans l'autre sens.

Le principal inconvénient de ce type de moteur est qu'il s'emballe à vide.

Ce type de moteur est utile dans l'électroménager par exemple ou dans les actions de traction ou de levage, la figure suivante présente le modèle d'une machine à excitation série :

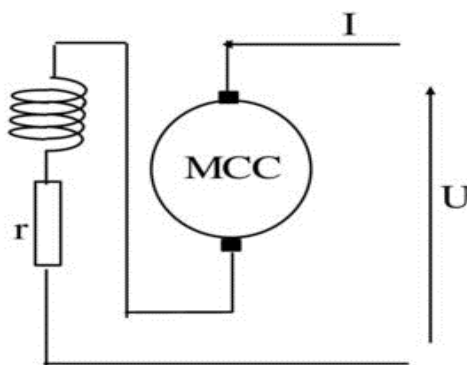


Figure. I.4 : Machine à excitation série.

d. Moteur à excitation composée

Ce type de moteur combine la structure à excitation série et celle du bobinage "shunt". Ainsi la polarité du bobinage shunt s'ajoute aux champs en série. Ce type de moteur possède un couple de démarrage élevé et offre une large variation de vitesse. Il est utilisé pour piloter des compresseurs, pompes centrifuges à tête variable, presses rotatives, scies circulaires, machines de cisaillement, ascenseurs et carrousels à bagages, la figure suivante présente le modèle d'une machine à excitation composé :

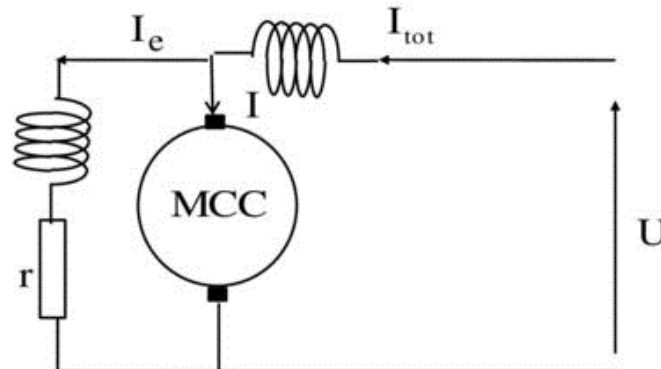


Figure. I.5 : Machine à excitation composé.

I.5 Domaine d'utilisation de moteur à courant continu :

Le moteur série est intéressant quand la charge impose d'avoir un gros couple, au démarrage et à faible vitesse de rotation. Le moteur sépare est particulièrement adapté aux entraînements de machines nécessitant des vitesses réglables (action sur la tension) et présentant un couple important en basse vitesse (machines outils) démarreur (automobile ...) moteur de traction (locomotive, métro ...) appareils de levage. Ventilateurs, pompes centrifuges, compresseurs, pompes à piston. Machines-outils à couple variable ou a mouvement alternatif (étaux-limeurs, raboteuses). [6]

I.6 Avantages et inconvénients de MCC : [1]

Les moteurs à courant continu sont utilisés dans de nombreuses applications industrielles, bien que leur construction soit plus complexe que celle des moteurs à courant alternatif.

I.6.1 Avantages :

- Alimentation aisée dans les systèmes transportant leur réserve d'énergie (autonome) : pile ou batterie.
- Une large gamme de variation de vitesse au-dessus de la vitesse de régime.
- Un fonctionnement avec des couples constant ou variables.

- Une accélération, un freinage et une inversion de sens de rotation très rapide, ce qui est avantageux dans le cas des appareils de levage et des machines-outils.
- Une vitesse de rotation qui peut être réglée par l'intermédiaire d'un système rétroaction.
- Une facilité de récupération d'énergie (fonctionnement dans les 4 quadrants).

I.6.2 Inconvénients :

- Une alimentation statorique et rotorique par courant continu (pont redresseur).
- Un mauvais facteur de puissance à basse vitesse.
- Nécessite beaucoup de maintenance, notamment pour le remplacement des balais et collecteur.
- Ne peut pas être utilisé dans des environnements explosifs, le frottement des balais sur le collecteur engendre des étincelles.

I.7 Méthodes utilisées pour varier la vitesse d'un moteur à courant continu

Il y a plusieurs façons de procéder pour varier la vitesse d'un moteur à courant continu. On peut le faire en variant la tension d'alimentation à ses bornes mais dans ce cas une partie importante de l'énergie est consommée par le dispositif d'alimentation. Pour cette raison, on préfère l'alimenter de façon discontinue avec un hacheur et faire ainsi varier la tension moyenne à ses bornes. On parle alors de Modulation par Largeur d'Impulsions (MLI), très utilisé dans le domaine de la régulation de vitesse de rotation d'un moteur courant continu. [7]

I.7.1 Les dispositifs électroniques utilisés pour la variation de vitesse des moteurs à courant continu:

Dans le cas des moteurs à courant continu nous avons deux types de convertisseurs :

- **Convertisseurs continu-continu (hacheurs)**

Un hacheur est un convertisseur statique continu-continu qui permet à partir d'un réseau à courant continu d'alimenter une charge sous une tension continue de valeur moyenne réglable. Le hacheur est basé sur principe d'ouverture et de fermeture régulière d'interrupteur statique, le réglage relatif des temps de l'ouverture et de la fermeture de l'interrupteur permet de contrôler l'échange d'énergie. [8]

Les différents types de convertisseur continu-continu sont présentés dans le (tableau I.1) :

Tableau I.1 : Les différents types des hacheurs. [8]

Type		Principe de fonctionnement	Les équations
Hacheurs non réversibles	Hacheur Série	L'interrupteur électronique H placé en série avec la source de tension, est périodiquement fermé pendant une durée αT et ouvert pendant $(1 - \alpha) T$	$U_s \text{ moy} = \alpha U_e$ (I.8) $I_e \text{ moy} = \alpha I_s$ (I.9) $P_e \text{ moy} = U_e \cdot I_e \text{ moy}$ (I.10) $P_e \text{ moy} = \alpha U_e I_s$ (I.11)
	Hacheur Parallèle	L'interrupteur électronique H, placé en parallèle sur la source de courant, est périodiquement ouvert pendant une durée βT et fermé pendant αT ($\alpha=1-\beta$).	$I_s \text{ moy} = \beta U_s = (1 - \alpha) I_e$ (I.12) $U_e \text{ moy} = \beta U_s = (1 - \alpha) U_s$ (I.13) $P_e \text{ moy} = U_e \text{ moy} I_e = U_s$ (I.14)

Type		Principe de fonctionnement	Les équations
Hacheurs Réversibles	Hacheur réversible en courant	Un hacheur deux quadrants associe un hacheur série et un hacheur parallèle. Il est réversible en courant mais pas en tension ; l'énergie est transférée de la source de tension continue vers la source de courant continu si $I_s > 0$, et réciproquement si $I_s < 0$. L'alimentation ($U_E > 0$) doit être réversible en courant.	$U_s moy = \alpha U_e$ (I.15) $I_e moy = \alpha I_s$ (I.16) $P_e moy = U_e I_e moy$ (I.17) $P_e moy = \alpha U_e I_s$ (I.18)
	Hacheur réversible en tension	De $t=0$ à $t=\alpha T$ les deux interrupteurs commandés H1 et H2 sont fermés et les diodes D1 et D2 ouvertes. La charge est sous tension $+E$.	$U_s = \alpha U_e + (1 - \alpha)(-U_e)$ (I.19) $U_s = 2\alpha - U_e$ (I.20)
	Hacheur réversible en courant et en tension	Un hacheur quatre quadrants associe deux hacheurs demi-point. Il est réversible en courant et en tension ; l'énergie est transférée de la source de tension continue vers la source de courant continu si $P_s Moy > 0$, et réciproquement si $P_s Moy < 0$.	$U_s moy = (2\alpha - 1)U_e$ (I.21) $I_e moy = (2\alpha - 1)I_s$ (I.22) $P_e moy = U_e I_e moy$ (I.23) $P_e moy = 2\alpha - U_e I_s$ (I.24)

➤ **Convertisseur alternatif-continu (redresseur)**

Les montages redresseurs, sont les convertisseurs de l'électronique de puissance qui assurent directement la conversion alternatif-continu. Alimenté par une source de tension alternative monophasée ou triphasée, ils permettent d'alimenter en courant continu le récepteur branché à leur sortie. Les différents types de convertisseur alternatif-continu sont présentés dans le (tableau I.2) [8] :

Tableau I.2 : Les différents types des redresseurs. [8]

Type	Principe de fonctionnement	Les équations
Redressement non commandé	redressement monophasé simple alternance	Charge résistive $U_s moy = \frac{U_s max}{\pi} \quad (I.25)$
	redressement monophasé double alternance	Charge résistive $U_s moy = \frac{U_s max}{\pi} \quad (I.26)$
	redressement triphasé simple alternance	Charge résistive $V_s = \frac{3\sqrt{3}.V_s max}{\pi} \quad (I.27)$
	redressement triphasé double alternance	Charge résistive $U_s = \frac{3U_s max}{\pi} \quad (I.28)$
Redressement Commandé	redressement monophasé simple alternance	La charge R, L, E $U_s moy = \frac{U_s max}{2\pi} (1 + \cos \alpha) \quad (I.29)$
	redressement monophasé double alternance	<p>➤ La charge R, L, E</p> <p>- Pont tout thyristors</p> $U_s moy = \frac{2U_s max}{\pi} (\cos \alpha) \quad (I.30)$ <p>-Ponts mixtes</p> $U_s moy = \frac{U_s max}{\pi} (1 + \cos \alpha) \quad (I.31)$
	redressement triphasé simple alternance	<p>➤ La charge R, L, E</p> $V_s moy = \frac{2\sqrt{3}U_s max}{2\pi} (\cos \alpha) \quad (I.32)$
	redressement triphasé double alternance	La charge R, L, E $U_s moy = \frac{3U_s max}{\pi} (\cos \alpha) \quad (I.33)$

I.8 principe et commande d'inversion de sens de rotation d'un moteur a courant continu

I.8.1 Principe de l'inversion de sens de rotation

Pour inverser le sens de rotation d'un moteur a courant continu, il faut inverser le courant soit dans l'inducteur, soit dans l'induit c'est-à-dire d'inverser la polarité aux bornes de l'induit ou de l'inducteur (pour les moteurs à inducteur bobiné).

L'inversion de sens s'effectue en :

- Inversant le sens du courant circulant dans l'induit pour un MCC à aimant permanent.
- Inversant le sens du courant circulant dans l'induit ou dans l'inducteur pour un MCC à bobinages.

De plus il est généralement préférable de pouvoir faire le sens de rotation du moteur. La solution est d'utiliser le pont en H. [9]

I.8.2 Commande d'inversion de sens de rotation d'un moteur à courant continu

De plus il est généralement préférable de pouvoir faire commande le sens de rotation du moteur, la solution est d'utiliser le pont en H.

Le pont en H est une structure électronique servant à contrôler la polarité aux bornes d'un récepteur. Il est composé de quatre éléments de commutation généralement disposés schématiquement en une forme de H d'où le nom. Les commutateurs peuvent être des relais, des transistors, ou autres éléments de commutation en fonction de l'application visée.

Il permet d'inverser le sens de rotation d'un moteur à courant continu, on utilise un pont en H. Le moteur est au milieu de ce pont d'interrupteurs comme la figure suivant :

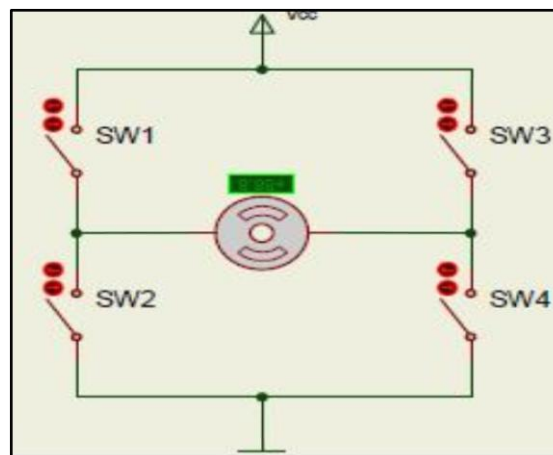


Figure I.6: Principe de pont H. [9]

Rotation sens 1 : SW1 et SW4 sont fermés, rotation sens 2 : SW2 et SW3 sont fermés.

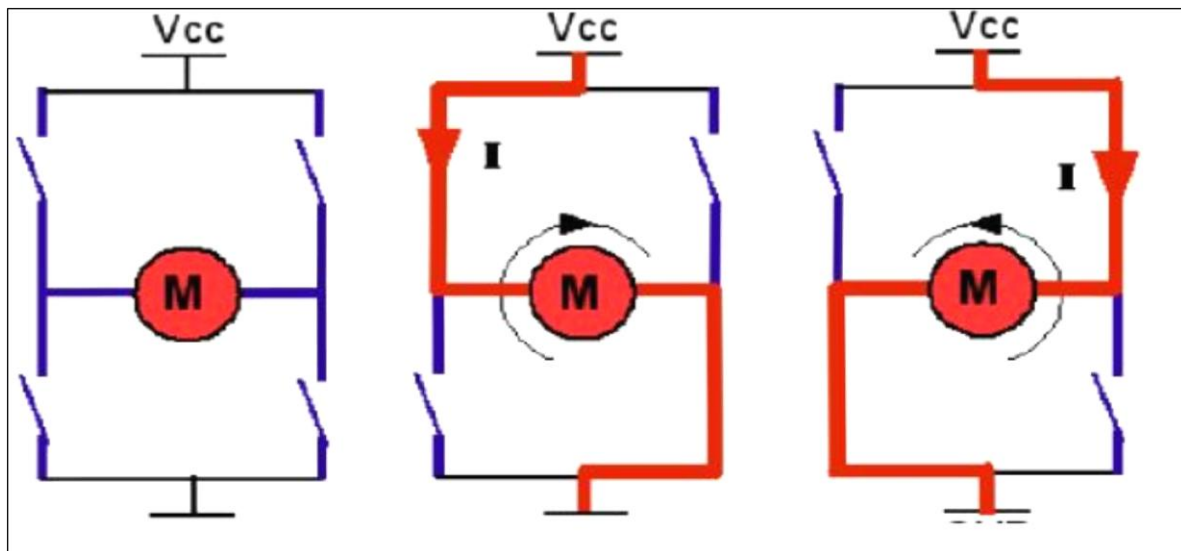


Figure I.7 : Inversion du sens de rotation du moteur. [9]

Ces interrupteurs peuvent être réalisés par les interrupteurs d'un relais, des transistors bipolaires (NPN/PNP), des transistors à effet de champ (MOSFET, canal N ou P) [9].

-Exemple d'un pont en H avec des transistors bipolaires (NPN/PNP) dans la figure suivant :

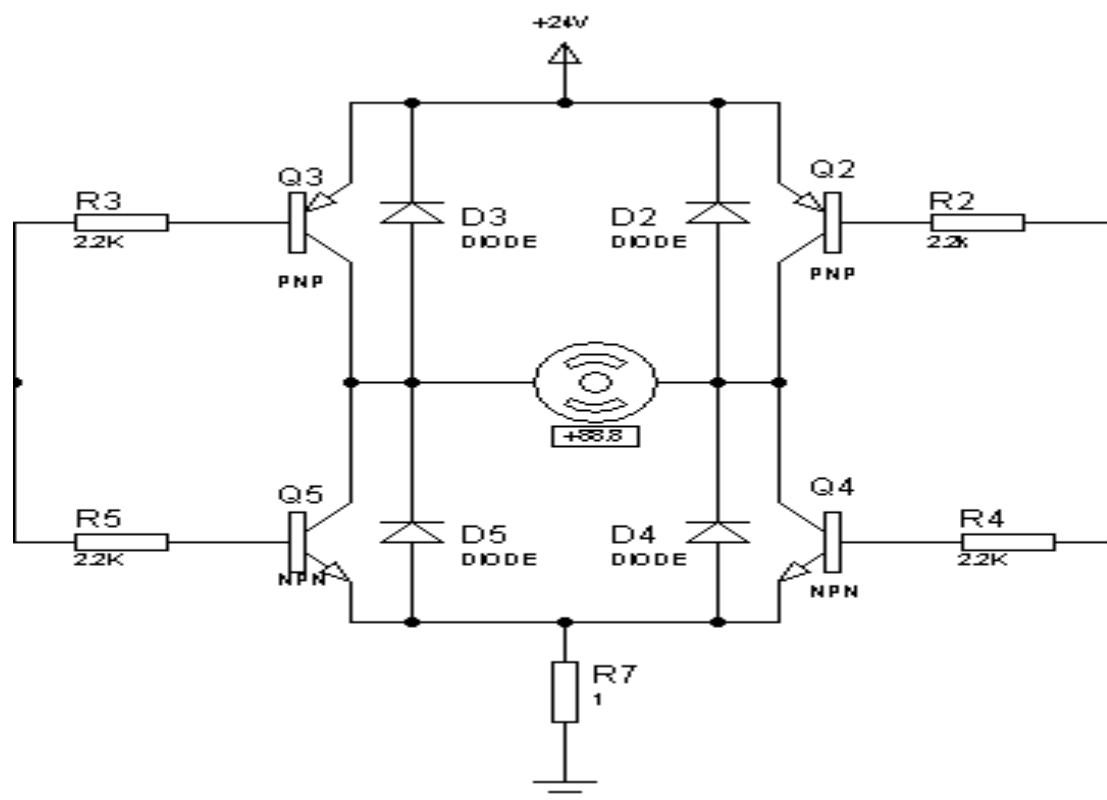


Figure I.8 : Pont en H avec des transistors bipolaires. [9]

I.9 Commande de vitesse d'un moteur à courant continu par les techniques de commande avancé et les techniques classiques

I.9.1 Les techniques de commande avancée

La théorie des systèmes a connu des progrès importants à travers les années. La plupart des techniques d'analyse et de synthèse sont basées sur des modèles linéaires des procédés commandés. Néanmoins, la nature non linéaire des systèmes physiques et en raison des performances de plus en plus croissantes exigées dans les applications industrielles, alors l'usage des techniques de commande avancée (commande adaptative, commande optimale, commande par mode de glissement, ...) devient indispensable. De nos jours, les techniques de commande avancée deviennent l'un des domaines de recherche les plus actifs. En parallèle, on dispose de calculateurs puissants et une variété d'outils logiciels. Ce qui facilite la synthèse de lois de commande avancées et leur exécution, sans difficultés en temps réel. La première étape dans l'étude d'un système de commande est la modélisation. Des modèles simples sont souvent adoptés. Le calcul de la loi de commande se fait ensuite d'une manière séparée à base du modèle élaboré, sans prise en compte de certains aspects physiques du processus à commander (variations paramétriques, consommation excessive d'énergie, ...). Ce qui conduit à des performances insuffisantes aux régimes transitoire et établi. Les variations paramétriques déstabilisent des systèmes de commande conventionnelle (régulateur à paramètres fixes) et dégrade les performances en boucle fermée. La solution efficace est de faire appel à des techniques de commande adaptative. La consommation abusive, la prolongation du régime transitoire ainsi que l'erreur au régime établi sont résolus par les commandes optimales. Le manque de robustesse par rapport aux fortes perturbations externes et variations importantes des paramètres des systèmes complexe et à constante de temps faible est compensé par la commande par mode de glissement. [10]

a. Commande adaptative

C'est l'ensemble des techniques utilisés pour l'ajustement automatique en ligne et en temps réel des régulateurs des boucles de commande, afin de réaliser ou de maintenir un certain niveau de performances, quand les paramètres du procédé à commander sont soit inconnus soit/et variables dans le temps. [11]

b. Commande optimale

La commande optimale se base sur un critère d'intégrale. il s'agit alors de minimiser ce critère. Dans ce but, on doit tenir compte des équations d'état du système à régler. Pour minimiser le critère d'intégrale, on applique le principe d'optimisation de Bellman. Ainsi, on

aboutit à la commande optimale et à la structure optimale. Comme on le verra, le procédé classique de la synthèse des réglages dans l'espace d'état ne tient pas compte de la présence de grandeurs de consigne et la perturbation. Pour cette raison, on limitera dans cette section à esquisser la marche à suivre, sans faire les développements en détail. On traitera alors ce cas général d'une manière approfondie.

Cette théorie est une généralisation du calcul des variations. Elle comporte deux volets : le principe du maximum (ou du minimum) dû à Lev Pontriaguine et à ses collaborateurs du Steklov Institute de Moscou, et l'équation de Hamilton-Jacobi-Bellman, généralisation de l'équation de Hamilton-Jacobi, et conséquence directe de la programmation dynamique initiée aux États-Unis par Richard Bellman. La théorie de la commande optimale fait partie de l'automatique et des mathématiques appliquées (optimisation des processus). En tant que cette théorie généralise le calcul des variations, elle a également un champ d'application en physique mathématique, et les développements théoriques actuels rejoignent les mathématiques pures. [12]

c. Commande par mode glissant

La commande par mode de glissement (CMG) est un mode de fonctionnement particulier des systèmes de commande à structure variable (SCSV). Ce type de systèmes a été étudié d'abord en ex-Union Soviétique par Utkin. V.I et Emeljavov. S.V. Par la suite, ces travaux ont été repris ailleurs, soit pour compléter l'étude théorique, soit pour étudier quelques applications possibles (machines électriques, convertisseurs électriques, robotique,...).

L'avantage majeur de la CMG est sa robustesse par rapport aux variations paramétriques. Son inconvénient, forte sollicitation de l'organe de commande et la nécessité d'utiliser la grandeur à régler et un certain nombre de ses dérivées, selon l'ordre du système.

La technologie des modes glissants consiste à amener la trajectoire d'état d'un système vers la surface de glissement et de la faire commuter à l'aide d'une logique de commutation appropriée autour de celle-ci jusqu'au point d'équilibre, d'où le phénomène de glissement.

La trajectoire dans le plan de phase se compose de trois parties distinctes :

- Le mode de convergence (MC).
- Le mode de glissement (MG).
- Le mode du régime permanent (MRP). [10]

I.9.2 Les techniques classiques

Les techniques classiques sont basées sur la commande PID est dite aussi (correcteur, régulateur, contrôleur), se compose de trois termes P, I et D, d'où le 'P' correspond au terme proportionnel, 'I' pour terme intégral et 'D' pour le terme dérivé de la commande.

Les régulateurs PID sont probablement les plus largement utilisés dans le contrôle industriel, même les plus complexes systèmes de contrôle industriel peut comporter un réseau de contrôle dont le principal élément de contrôle est un module de contrôle PID.

Le régulateur PID est une simple implémentation de retour d'information (Feedback). Il a la capacité d'éliminer la compensation de l'état d'équilibre grâce à l'action intégrale, et il peut anticiper le futur grâce à une action dérivée. [13]

a. Commande Proportionnelle

La commande de type 'Proportionnelle' est la plus simple qui soit. Il s'agit d'appliquer une correction proportionnelle à l'erreur corrigeant de manière instantanée, c'est-à-dire, lorsque le signal de commande est proportionnel au signal d'erreur:

$$Uc(t) = Kp\mathcal{E}(t) \quad \overline{TL} \quad Uc(p) = Kp\mathcal{E}(p) \quad (I.34)$$

Son rôle est d'amplifier l'erreur pour que le système réagisse plus vivement, comme si l'erreur était plus grande qu'elle ne l'est en réalité.

Il permet de vaincre les grandes inerties du système et diminue le temps de montée en donnant de la puissance au moteur (plus l'erreur est grande, plus on donne de puissance au moteur).

Lorsque l'on augmente le gain 'Kp', le système réagit plus vite et l'erreur statique s'en trouve améliorée, mais en contrepartie le système perd la stabilité. Le dépassement se fait de plus en plus grand, et le système peut même diverger dans le cas d'un 'Kp' démesuré. [14]

b. Commande Proportionnelle-Intégrale

Le correcteur de type PI est une régulation de type P auquel on a ajouté un terme intégral, il élabore alors une commande qui peut être donnée par la relation suivante :

$$Uc(t) = Kp\mathcal{E}(t) + \frac{1}{Ti} \int_0^t \mathcal{E}(t)dt \quad \overline{TL} \quad Uc(p) = Kp\mathcal{E}(p) + Ki \frac{d\mathcal{E}(p)}{p} \quad (I.35)$$

Le terme intégral complète l'action proportionnelle puisqu'il permet de compenser l'erreur statique et d'augmenter la précision en régime permanent. L'idée est d'intégrer l'erreur depuis

le début et d'ajouter cette erreur à la consigne, lorsque l'on se rapproche de la valeur demandée, l'erreur devient de plus en plus faible. Le terme proportionnel n'agit plus mais le

terme intégral subsiste et reste stable, ce qui maintient le moteur à la valeur demandée.

L'intégrale agissant comme un filtre sur le signal intégré, elle permet de diminuer l'impact des perturbations (bruit, parasites), et il en résulte alors un système plus stable.

Malheureusement, un terme intégral trop important peut lui aussi entraîner un dépassement de la consigne, une stabilisation plus lente, voire même des oscillations divergentes. [14]

c. Commande Proportionnelle-Intégrale-Dérivée

Les termes proportionnel et intégral peuvent amener un dépassement de la consigne et des oscillations dans des asservissements, cela implique pour le moteur des inversions de polarité, ce qui est loin d'être idéal. Pour limiter ce phénomène indésirable, on introduit un troisième élément, le terme dérivé. Son action va dépendre du signe et de la vitesse de variation de l'erreur, et sera opposée à l'action proportionnelle. Elle devient prépondérante aux abords de la valeur demandée lorsque l'erreur devient faible, que l'action du terme proportionnel faiblit et que l'intégrale varie peu, elle freine alors le système, limitant le dépassement et diminuant le temps de stabilisation.

L'action dérivée est surtout utilisée dans le cas de variables non bruitées, car la dérivation est très sensible au bruitage du signal, on diminuera donc son influence dans un asservissement de vitesse, pour lequel sa dérivée est une accélération, donc dans un asservissement de vitesse seulement l'action P et I sont utilisées, en revanche le terme 'D' peut être utilisé dans un seul cas où le gain 'Kd' prend une valeur précise pour ne pas trop influencer sur le système. [14]

I.9.3 La différence entre les techniques de commande avancée et les techniques classiques

Les commandes des techniques classiques synthétisées généralement par des formalismes mathématiques rigides ont été souvent mises à difficulté devant des contraintes multiples et des besoins accrus en performance.

Les systèmes réels sont dans la plupart du temps non linéaires, mal définis, ont des paramètres variables et sont soumis à des perturbations externes. La modélisation de ces systèmes relève très souvent d'une approximation des phénomènes physiques mis en jeu. Or, c'est à partir de cette représentation approximative des systèmes que l'on souhaite construire une commande pour le système réel. Cette commande doit être alors robuste dans le sens où elle devra assurer une faible sensibilité aux incertitudes sur les paramètres, à leurs variations et aux

perturbations externes. L'objectif de ce travail consiste à déterminer des commandes robustes, performantes et relativement simples à mettre en œuvre.

Par contre lorsque la partie commandée du processus est faiblement perturbée, les commandes classiques, par exemple à action proportionnelle intégrale dérivée, peuvent s'avérer suffisantes si les exigences en précision et en performance du système ne sont pas trop sévères. Dans le cas contraire, il faut concevoir des algorithmes de commande avancé assurant une robustesse du comportement, du processus vis-à-vis des incertitudes sur les paramètres et leurs variations.

De nombreuses approches robustes ont été proposées, on peut citer dans ce contexte, la commande adaptative, la commande par placement de pôle robuste, la commande optimale, la commande par mode glissant, la commande floue...etc. [15]

I.10 Conclusion

Dans ce chapitre on a présenté les constitutions et le principe de fonctionnement de moteur à courant continu et les divers types de ce moteur et leur mise en équation. Ainsi, les différentes méthodes utilisées pour varier sa vitesse et inverser le sens de rotation, ensuite on a expliqué la commande de vitesse par la technique commande avancée et les techniques classiques, et cela dans le but de détailler sa commande dans la partie qui suit.

Chapitre II

Développement et programmation des interfaces graphiques

II.1 Introduction

L'interface graphique, appelée communément la GUI (Graphical User Interface), est le lien entre l'utilisateur et la machine.

En effet, même si certains programmes tournent sur l'ordinateur sans aucune intervention de l'utilisateur, pour la majorité des scripts que vous allez coder, vous allez devoir interagir avec l'utilisateur. Pour cela, on utilise une interface graphique permettant de faire le lien entre l'utilisateur et le programme, et comme vous allez le voir, les GUIs offrent des possibilités quasiment illimitées, si ce n'est par votre imagination. [16]

II.2 Interaction homme-machine [17]

II.2.1 Définition Interaction homme-machine

Les interfaces homme-machine ou IHM (aussi appelées interactions personne-machine ou IPM ; en anglais, human-machine interfaces ou HMI).

Une Interface Homme-Machine (IHM) est une interface utilisateur permettant de connecter une personne à une machine, à un système ou à un appareil. En théorie, il est donc possible d'utiliser ce terme pour définir n'importe quel écran permettant à un utilisateur d'interagir avec un appareil. Cependant, il est généralement utilisé pour le contexte d'un processus industriel.

D'autres termes peuvent également être utilisés pour désigner cette technologie : Terminal d'Interface Opérateur, Interface d'Opérateur Local, ou encore Terminal Opérateur. En revanche, il ne faut pas confondre les IHM avec des Interfaces Utilisateur Graphiques (GUI). Ces dernières sont souvent utilisées au sein d'une interface IHM à des fins de visualisation.

Les IHM peuvent prendre différentes formes. Il peut s'agir d'écrans directement intégrés aux machines, d'écrans d'ordinateur, de tablettes tactiles, et bien plus encore. Quelle que soit la forme, le but de ces interfaces reste toutefois le même : fournir des insights sur les performances et les progrès des machines.

II.2.2 l'intérêt d'interaction homme-machine

Dans un contexte industriel, les IHM peuvent être utilisées pour plusieurs raisons. Elles peuvent servir à afficher les données de façon visuelle, ou pour suivre les temps de production ou les tendances. Elles peuvent aussi servir à surveiller les indicatifs de performances, ou à surveiller les entrées et sorties des machines. Il s'agit là de quelques exemples, mais les IHM peuvent avoir de nombreux autres cas d'usage.

L'opérateur d'une usine peut par exemple utiliser une IHM pour surveiller et contrôler la température d'un réservoir d'eau industriel, ou pour vérifier si les différentes pompes du complexe fonctionnent correctement. Ces interfaces servent donc de passerelles entre les opérateurs et les équipements industriels.

II.2.3 l'utilité d'interaction homme-machine

Les technologies IHM sont utilisées dans presque toutes les entreprises industrielles, ainsi que par d'autres entreprises, afin d'interagir avec les machines et d'optimiser les processus industriels, les interfaces homme-machine sont notamment utilisées dans les industries de l'énergie, de l'agroalimentaire, de la fabrication, du pétrole et du gaz, de l'énergie, du recyclage, ou encore du transport.

Les personnes chargées d'utiliser les IHM sont principalement, les opérateurs, les intégrateurs systèmes, et les ingénieurs (plus particulièrement les ingénieurs de système de contrôle). Ils utilisent ces interfaces comme des ressources essentielles pour surveiller les processus, diagnostiquer les problèmes, et visualiser les données.

II.3 langages de programmations des interfaces graphiques [18]

Pour programmer une interface graphique, il nous faut une boîte à outils.

La boîte à outils a été programmée en utilisant un langage de programmation donné (C, C++, Java, Matlab, python, etc.).

Ainsi donc le choix du langage de programmation à utiliser pour coder votre interface graphique peut être du même type que celui de la boîte à outil. Comme il peut être différent, si la boîte à outils le permet.

II.3.1 Boîtes à outils et Langages de programmation

A - Langage C/C++ :

Qt : développée par « Trolltech ».

Langage : C++.

Disponible sous licence GPL ou commerciale.

Disponible sous GPL depuis peu à la communauté Windows.

Exemple : KDE, gestionnaire de fenêtre sous Linux. Navigateur Opera.

wxWidgets : développée par la communauté « Open Source ».

Langage : C++.

L'ancien nom « wxWindows ».

Disponible sous licence LGPL.

Disponible depuis plus d'une dizaine d'années.

Exemple : AOL Communicator.

GTK+/GTKMM : développée par la communauté « Open Source ».

Langage : C (gtkmm C++)

Elle a vu le jour à cause des contraintes de licences associées à l'utilisation de Qt.

Version Windows chaotique , Il faut être sacrément très patient

Exemple : Gnome, gestionnaire de fenêtre sous Linux. Gimp (images).

B -Java :

- Développé par la compagnie SUN en 1995.
- Indépendant de la plateforme.
- AWT est le paquetage de base pour construire et manipuler des interfaces utilisateurs graphiques. Il est parmi les paquetages originaux de Java.
- Swing un nouveau paquetage dont les composantes sont écrites, manipulées et affichées complètement en Java ("pur" java).

C-python :

Python possède plusieurs modules disponibles pour la création de logiciels avec une interface graphique. Le plus répandu est Tkinter. Ce module convient à beaucoup d'applications et peut être considéré comme suffisant dans la plupart des cas. Néanmoins, d'autres modules ont été créés pour pouvoir lier Python à d'autres bibliothèques logicielles (« toolkit »), pour davantage de fonctionnalités, pour une meilleure intégration avec le système d'exploitation utilisé, ou simplement pour pouvoir utiliser Python avec sa bibliothèque préférée. En effet, certains programmeurs trouvent l'utilisation de Tkinter plus pénible que d'autres bibliothèques. Ces autres modules ne font pas partie de la bibliothèque standard et doivent donc être obtenus séparément.

Les principaux modules donnant accès aux bibliothèques d'interface graphique sont Tkinter et Pmw (Python megawidgets)⁵⁰ pour Tk, wxPython pour wxWidgets, PyGTK pour GTK+, PyQt et PySide pour Qt, et enfin FxPy pour le FOX Toolkit. Il existe aussi une adaptation de la bibliothèque SDL : Pygame, un binding de la SFML : PySFML, ainsi qu'une bibliothèque écrite spécialement pour Python : Pyglet (en).

Il est aussi possible de créer des applications Silverlight en Python sur la plateforme IronPython.

D-Matlab

Depuis l'introduction de l'outil GUIDE dans la version 5.0 de MATLAB, le développement des Interfaces Graphiques (GUI) est devenu courant. Elles permettent à l'utilisateur d'interagir avec un programme informatique, grâce à différents objets graphiques (boutons, menus, cases à cocher...). Ces objets sont généralement actionnés à l'aide de la souris ou du clavier.

Malgré le fait que les interfaces graphiques semblent secondaires par rapport au développement du cœur d'une application, elles doivent néanmoins être conçues et développées avec soin et rigueur.

Leur efficacité et leur ergonomie sont essentielles dans l'acceptation et l'utilisation de ces outils par les utilisateurs finaux.

Une bonne conception et un développement maîtrisé permettent également d'en assurer une meilleure maintenabilité.

II.4 Création d'une interface graphique sous MATLAB GUIDE

II.4.1 Définition Logiciel MATLAB

MATLAB est avant tout un logiciel de calcul matriciel. Il possède néanmoins une panoplie complète d'objets graphiques qui permettent d'une part d'afficher les résultats de calculs sous des formes variées (point, courbe, surface, graphique) et d'autre part, de créer des interfaces graphique (GUI) permettant à l'utilisateur d'interagir avec le programme. Dans ce présent chapitre, on va présenter d'une manière générale l'outil de programmation GUI MATLAB, de programmer notre interface et nous terminons par une réalisation virtuelle du système sous le logiciel V-Realm Builder et une simulation en 3D [19].

II.4.2 Présentation du MATLAB GUIDE (GUI MATLAB)

Les IHM (Interface Homme-Machine), appelées GUI (Graphical User Interfaces) dans MATLAB permettent à l'utilisateur d'interagir avec un programme informatique grâce à des objets graphiques (boutons, menus, cases à cocher,...). Du fait du nombre important d'objets et surtout du nombre encore plus élevé des paramètres associés, leur programmation "à la main" déroute généralement le débutant. Depuis l'apparition de la version 5.0 (1997), MATLAB possède un outil IDE dédié à la création des interfaces graphiques. Cet outil,

appelé GUIDE (Graphical User Interface Development Environment), permet de concevoir intuitivement ces interfaces graphiques, (Voir l'annexe) [19].

II.4.3 Création d'une interface graphique sous MATLAB GUIDE

- La commande `guide` permet de créer ou ouvrir un GUI.

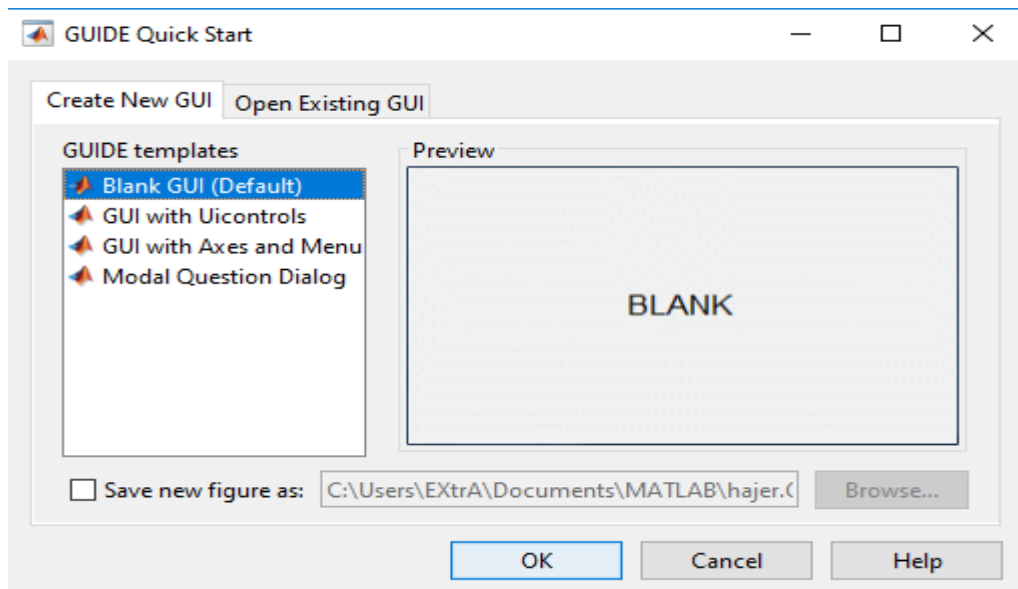


Figure II.1 : GUIDE MATLAB. [20]

➤ En créant un nouveau gui, Matlab affiche une figure avec des icônes de contrôles (boutons, menus, textes, ...). Le programmeur peut glisser ces icônes sur la forme afin de construire un GUI.,ci dessous, vous avez l'interface du GUIDE [20] :

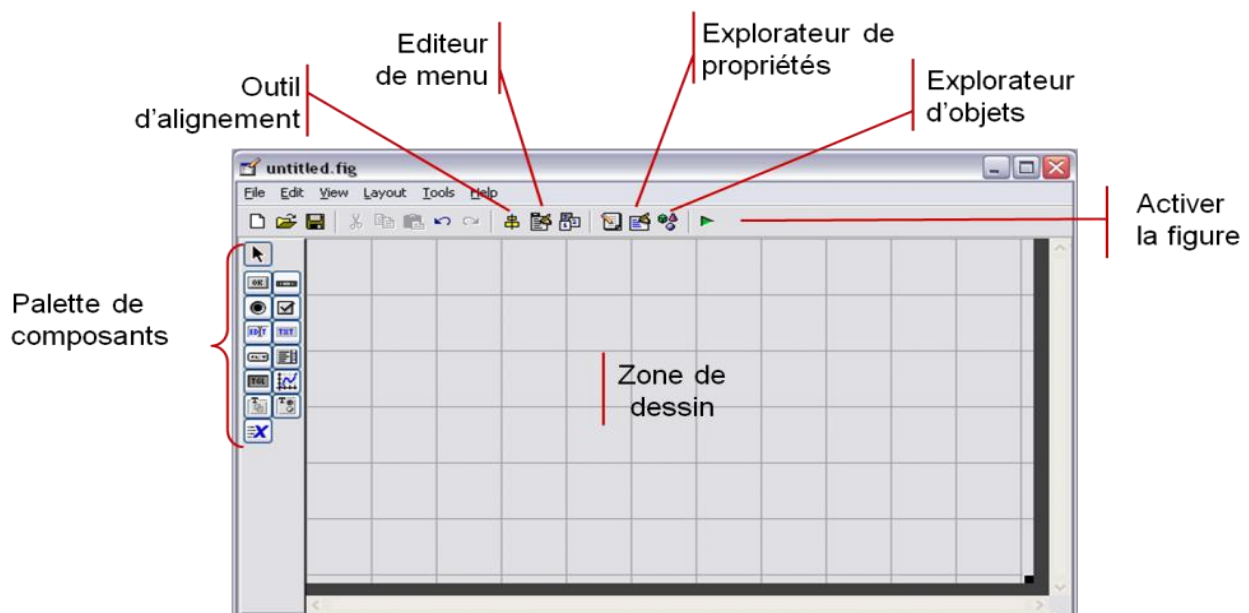


Figure II.2 : Présentation des outils d'une interface graphique sous GUIDE MATLAB. [21]

II.5 Les composants des éléments visuels [21]

➤ **Figure :**

- Plusieurs callbacks.
- Propriétés importantes:
 - NumberTitle, Name.
 - Menubar, Toolbar.
 - Units.
 - Resize.

➤ **Push button :**

- Bouton poussoir qui revient à l'état initial.
- Callback exécuté au relâchement de la souris.
- Propriétés importantes:
 - String : chaîne affichée sur le bouton.
 - Tag : nomme la fonction de callback.
 - CData : image sur le bouton.

➤ **Toggle button :**

- Bouton qui change d'état binaire.
- Callback exécuté au relâchement de la souris.
- Propriétés importantes :
 - les mêmes que push button.
 - Value indique l'état du bouton:
 - Si Value = Max, le bouton est pressé.
 - Si Value = Min, le bouton n'est pas enfoncé.

➤ **Radio button :**

- Bouton à 2 états : sélectionné et désélectionné.
- Callback exécuté au clic souris.

- L'exclusion mutuelle doit être codée.
- Propriétés importantes :
 - String : chaîne affichée à côté du bouton.
 - Value :
 - Si Value = Min, bouton désélectionné.
 - Si Value = Max, bouton sélectionné.
- **Checkbox :**
 - Bouton à 2 états : sélectionné et désélectionné.
 - Callback exécuté au clic souris.
 - Propriétés importantes :
 - String : chaîne affichée à côté du bouton.
 - Value :
 - Si Value = Min, bouton désélectionné.
 - Si Value = Max, bouton sélectionné.
- **Edit text :**
 - Champ éditable.
 - Callback exécuté à la perte du focus.
 - Propriétés importantes:
 - String : chaîne affichée dans le champ éditable:
 - La chaîne doit être convertie pour obtenir des valeurs numériques (str2num, str2double).
 - Tag.
 - Min, Max:
 - Si $\text{Max} - \text{Min} > 1$, édition multi-ligne.
- **Static text :**
 - Texte statique.
 - Propriété importante:
 - String : chaîne affichée sur le composant.

➤ Slider :

- Barre permettant à l'utilisateur de fournir une valeur numérique en la déplaçant.
- Orientation définie par le rapport Largeur/Hauteur.
- Propriétés importantes:
 - Value : Valeur courante.
 - Max : Valeur max possible.
 - Min : Valeur min possible.
 - SliderStep : Pas de quantification:
 - Lorsqu'on clique sur les flèches.
 - Lorsqu'on clique sur le composant.

➤ List Box :

- Affiche une liste de choix sur plusieurs lignes;
- Callback exécuté au relâchement de la souris;
- Propriétés importantes :
 - String : contient la liste des choix.
 - Value : contient les indices des éléments sélectionnés (le premier élément a l'indice 1).
 - Min, Max :
 - Si $Max - Min > 1$, la liste permet la sélection multiple.
 - Si $Max - Min \leq 1$, la liste ne permet pas la sélection multiple.

➤ Pop-up Menu :

- Affiche une liste de choix sur 1 seule ligne.
- Exécute le callback une fois la sélection terminée.
- Propriétés importantes:
 - String : contient la liste des choix.
 - Value: contient l'indice de l'élément sélectionné(le premier élément à l'indice1).

➤ **Axes :**

- N'a pas les mêmes propriétés que les composants précédents.
- Propriétés importantes:
 - ButtonDownFcn : callback des axes.
 - Box.
 - XGrid, YGrid, ZGrid.
 - XScale, YScale, ZScale.

➤ **Panel et Button group :**

- Permettent d'arranger les composants en groupe.
- Un button group assure l'exclusivité des radio buttons ou des toggle buttons.
- Callback au changement de la sélection.
- Propriétés importantes:
 - Title, TitlePosition.

II.6 Présentation d'un exemple réalisé dans MATLAB GUIDE

Par exemple, on veut créer un simple GUI pour une calculatrice scientifique et graphique, qui affiche tous les nombres réels et permet de faire les opérations mathématique (somme, produit, soustraction et division...etc) et pour tracer les graphes Voici alors les étapes de création et d'exécution :

- Cliquer sur l'icone 'panel' puis cliquer le sur la forme afin de créer une zone de travail.
- Cliquer sur l'icone 'push button' puis cliquer le sur la forme afin de créer un bouton poussoir Modifier son nom en double-cliquant sur 'push button' L'inspecteur d'objet se montre. Chercher la propriété 'String' et modifie la avec la valeur '0' et la même chose pour les numéros de 1 jusqu'a 9.
- Répéter la tache précédente pour créer les opérations mathématique d'une calculatrice (+ : -,x, ; / ;cos ; cos-1 ;sin ; sin-1 ;tan ;tan-1 ;e ;,ln ;log ; √ ; 1/x ; x^y ; x^2 ; x^3 ; x^-1 ; % ; π ; (;) ; = ; +/- ; .) et pour les paramètres du calculatrice (c ; ←).

- Cliquer sur l'icone 'Static Text' puis cliquer le sur la forme afin de créer un texte statique 'text1'. Modifier son nom en double-cliquant sur 'text1.L'inspecteur d'objet se montre. Chercher les propriétés :
 - 'FontSize' modifie avec la valeur '15' .
 - 'FontWeight' modifie en bold.
 - 'HorizontalAlignment' modifie en right.
 - 'String' et modifie la avec la valeur vide ' ' pour afficher les opérations arithmétiques.
- Les paramètres s'affichent comme suit :

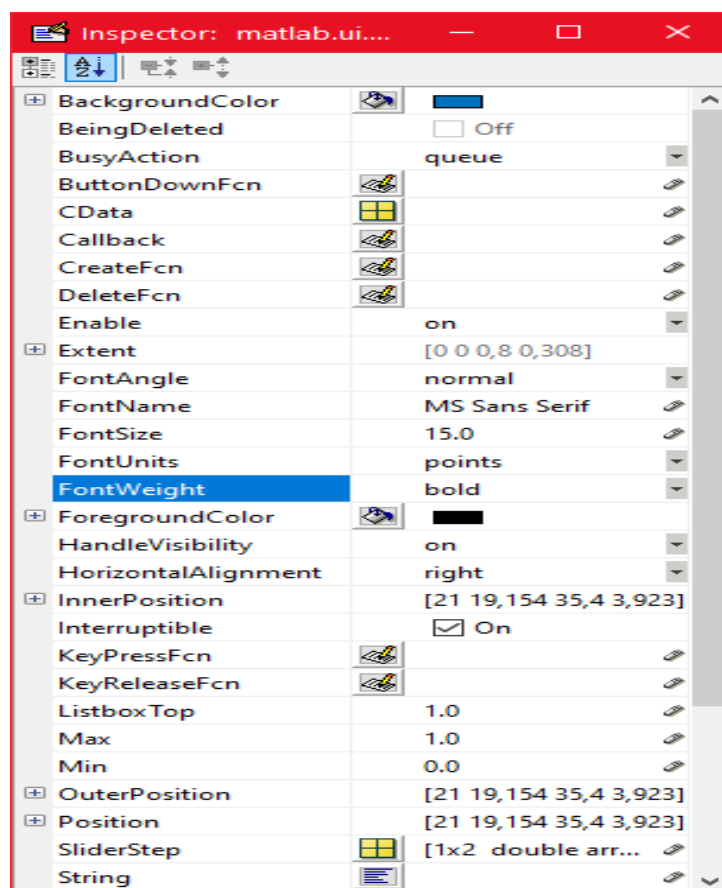


Figure II.3 : Les propriétés d'un texte statique.

- Répéter la tâche précédente pour créer 'text2' avec la valeur vide ' ' pour afficher les résultats des opérations arithmétiques.
- Cliquer sur l'icone 'axes' puis cliquer le sur la forme afin de créer une zone pour tracer les graphes sur calculatrice.
- Cliquer sur l'icone 'push button' puis cliquer le sur la forme afin de créer un bouton poussoir. Modifier son nom en double-cliquant sur 'push button'

L'inspecteur d'objet se montre. chercher la propriété 'String' et modifie avec le nom 'plot' pour tracer les graphes des fonctions mathématiques.

- Cliquer sur l'icone 'slider' puis cliquer le sur la forme afin de créer une échelle graduée pour modifier l'épaisseur de la ligne des graphes.
- Cliquer sur l'icone 'Pop-up Menu' puis cliquer le sur la forme afin de créer une liste de choix pour les fonctions mathématiques.
- Le GUI se figure comme suit :

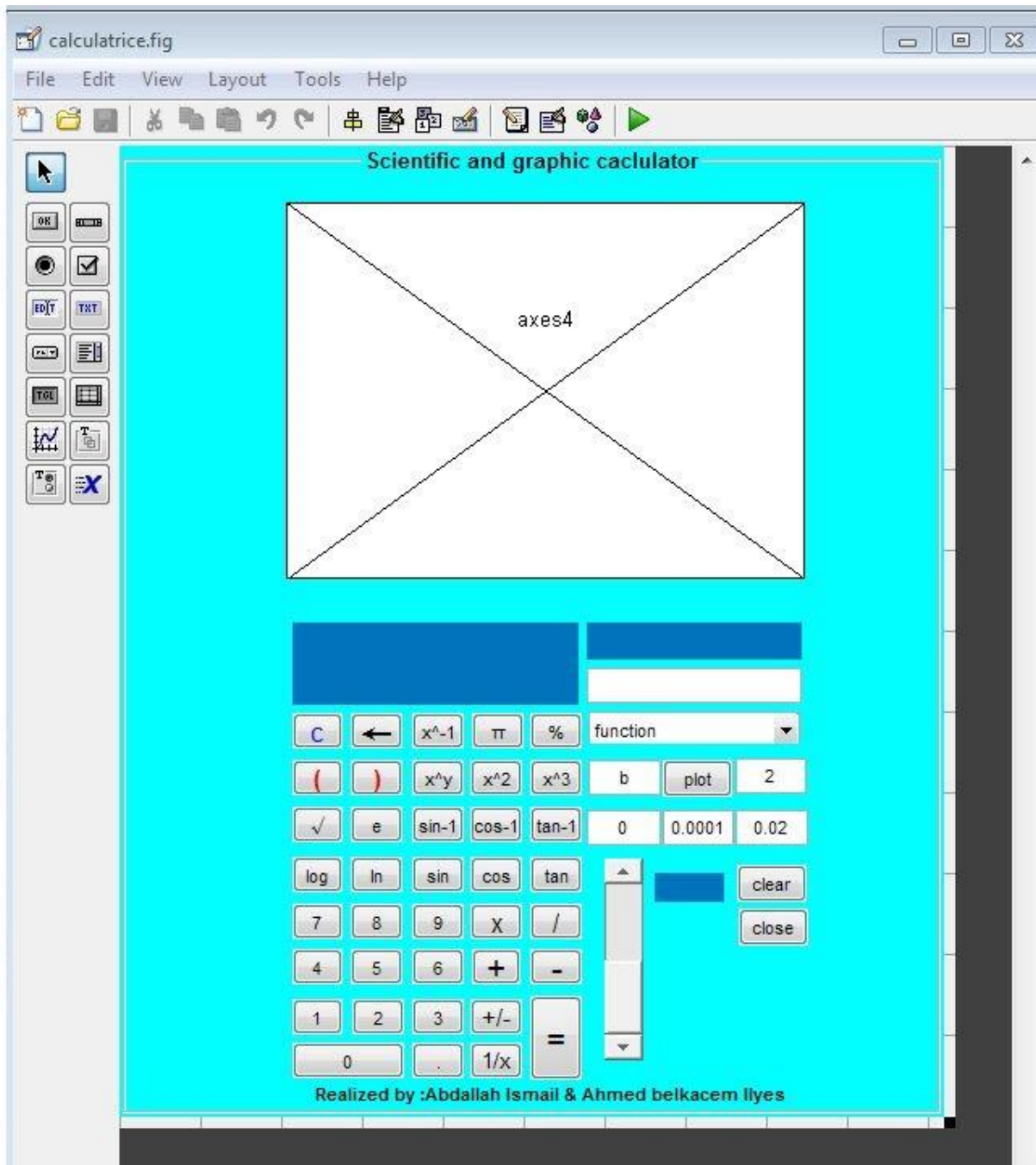


Figure II.4 : calculatrice graphique et scientifique sous MATLAB GUIDE.

II.6.1 Programmation

- Maintenant, cliquer avec le bouton droit de la souris sur le bouton '0' et sélectionner l'option 'View Callbacks->Callback' . Matlab interrompt l'action et demande de sauvegarder le gui (calculatrice).
- Matlab affiche sur l'éditeur le contenu de gui1.m qui est crée automatiquement par Matlab.
- Editer la fonction 'pushbutton1_Callback' comme suit :

```
% --- Executes on button press in p0.
function p0_Callback(hObject, eventdata, handles)
% hObject    handle to p0 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
ca=get(handles.t1,'string')
ca=strcat(ca,'0');
set(handles.t1,'string',ca)
```

- Répéter la tache précédente pour les autres numéros de 1 jusqu'a 9 et ensuite pour créer les opérations mathématiques d'une calculatrice (+ : -,x, ; / ;cos ; cos-1 ;sin ; sin-1 ;tan ;tan-1 ;e ;,ln ;log ; √ ; 1/x ; x^y ; x^2 ; x^3 ; x^-1 ; % ; π ; (;) ; = ; +/- ; .) et pour les paramètres du calculatrice (c ; ←).
- Maintenant pour calculer, cliquer avec le bouton droit de la souris sur le bouton '=' et sélectionner l'option 'View Callbacks->Callback'. Matlab interrompt l'action et demande de sauvegarder le gui (calculatrice).
- Matlab affiche sur l'éditeur le contenu de gui1.m qui est crée automatiquement par Matlab.

- Editer la fonction 'eql_Callback' comme suit :

```
% --- Executes on button press in eql.
function eql_Callback(hObject, eventdata, handles)
% hObject    handle to eql (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
ca=get(handles.t1,'string')
c=eval(ca);
set(handles.t2,'string',c)
```

- Editer la fonction du bouton plot 'pushbutton58_Callback' pour tracer les graphes comme suit :

```
% --- Executes on button press in pushbutton58.
function pushbutton58_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton58 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
t5=str2num(get(handles.edit1,'String'));
t6=str2num(get(handles.edit2,'String'));
Dt=str2num(get(handles.edit3,'String'));
t=t5:Dt:t6;
y=eval(get(handles.edit5,'String'))%sin(314.*t)
set(handles.text6,'String',get(handles.edit5,'String'))
cl=get(handles.edit4,'String')
LW=str2num(get(handles.edit6,'String'))
plot(handles.axes4,t,y,cl,'LineWidth',LW)
grid(handles.axes4)
```

- Sauvegarder, puis exécuter le script 'calculatrice. m' en cliquant sur 'Run'

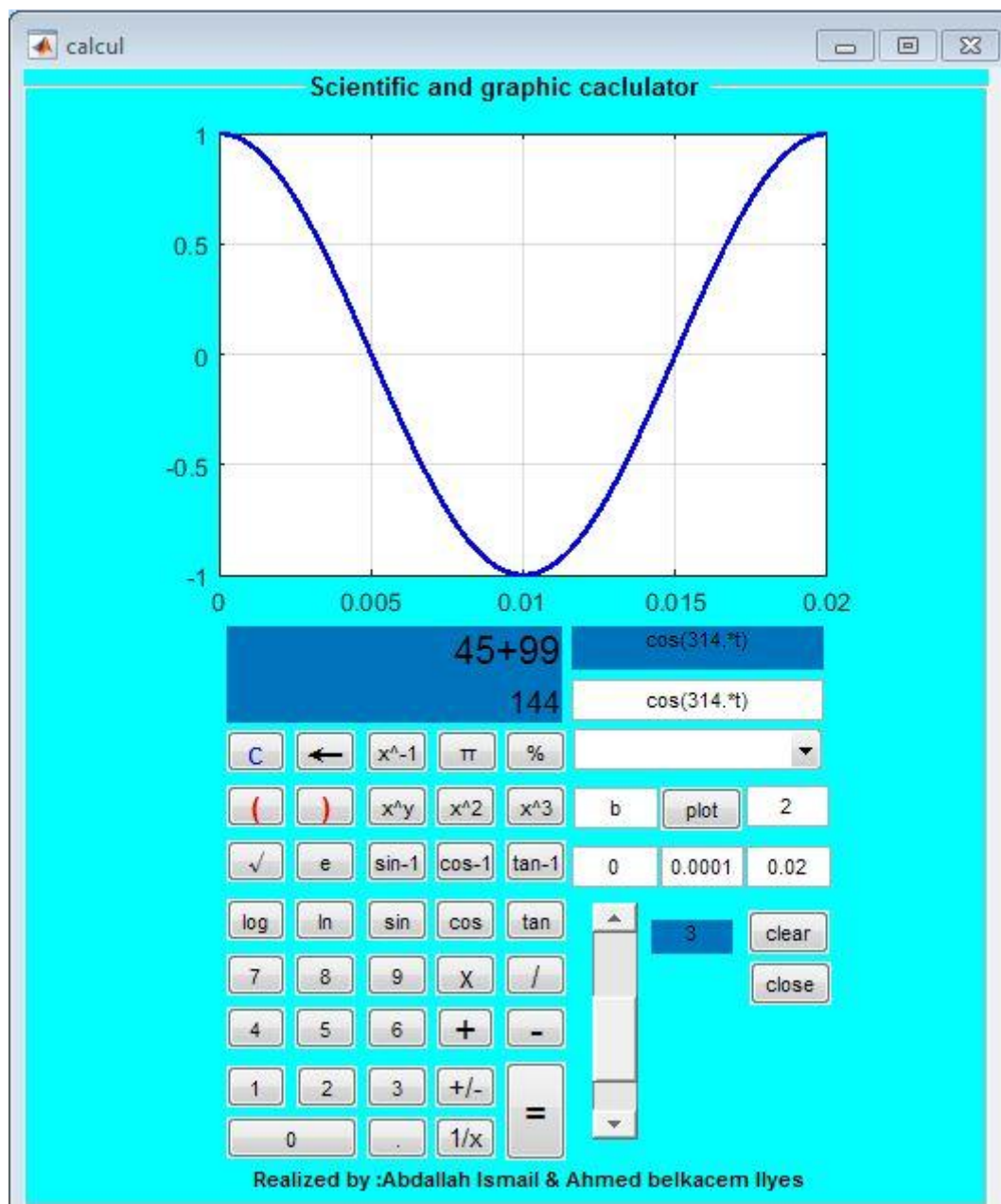


Figure II.5 : exécution d'une calculatrice graphique et scientifique sous GUIDE MATLAB.

I.7 Conclusion

Dans la première partie de ce chapitre, nous avons présenté interaction homme machine, son intérêt et son utilité dans l'industrie, après dans la deuxième partie nous avons mentionné les langages de programmation des interfaces graphiques.

Dans la troisième partie nous avons défini le logiciel MATLAB et présenté GUI MATLAB d'une manière générale pour avoir une idée de son fonctionnement, ensuite et à base des notions de création d'une interface graphique on a présenté dans la dernière partie un exemple d'une interface graphique d'une calculatrice graphique et scientifique qui répond aux exigences demandées avec sa programmation. Cette démarche est très utile pour concevoir, tester, contrôler, simuler des systèmes électriques avant de les réaliser pratiquement ce nous verrons dans la prochaine chapitre.

Chapitre III

Modélisation et simulation d'un MCC sous interface graphique

III.1 Introduction

Il s'agit, dans ce chapitre, de faire la modélisation et la simulation d'un moteur à courant continu, on a utilisé le logiciel MATLAB auquel est intégré l'outil SIMULINK qui est une plate-forme de simulation multi-domaine et de modélisation de systèmes dynamiques. Il fournit un environnement graphique et un ensemble de bibliothèques contenant des blocs de modélisation qui permettent le design précis, la simulation, l'implémentation et le contrôle de systèmes de communications et de traitement du signal et nous faire une liaison entre l'outil SIMULINK et l'outil GUIDE pour faire une interface graphique qui affiche les résultats des simulations obtenus et les performances du système étudié.

La finalité de la manipulation sera ainsi d'étudier le fonctionnement de moteur à courant continu (à vide, en charge) avec et sans régulation en simulant puis en interprétant leurs courbes de fonctionnement.

III.2 Implémentation du MCC sous SIMULINK-MATLAB

III.2.1 Modélisation et simulation du système sous SIMULINK-MATLAB [22]

III.2.1.1 Modélisation de la source triphasée

Il s'agit d'un MCC à excitation séparée commandé par la tension d'induit, dont le schéma technologique est illustré dans la figure au-dessous.

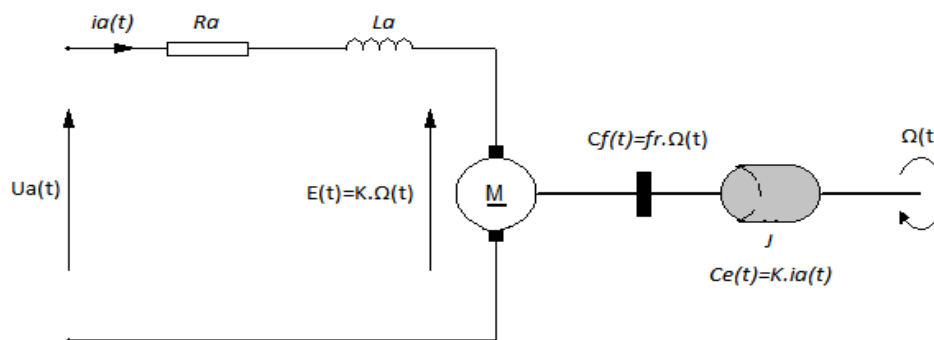


Figure III.1 : Schéma de principe d'un MCC.

On suppose que le redresseur est alimenté par un réseau triphasé équilibré de tension défini par le système d'équations suivantes :

$$U_a(t) = \sqrt{2}V_{eff} \sin (2\pi ft) \quad (III.1)$$

$$U_b(t) = \sqrt{2}V_{eff} \sin (2\pi ft - \frac{2\pi}{3}) \quad (III.2)$$

$$U_c(t) = \sqrt{2}V_{eff} \sin (2\pi ft - \frac{4\pi}{3}) \quad (III.3)$$

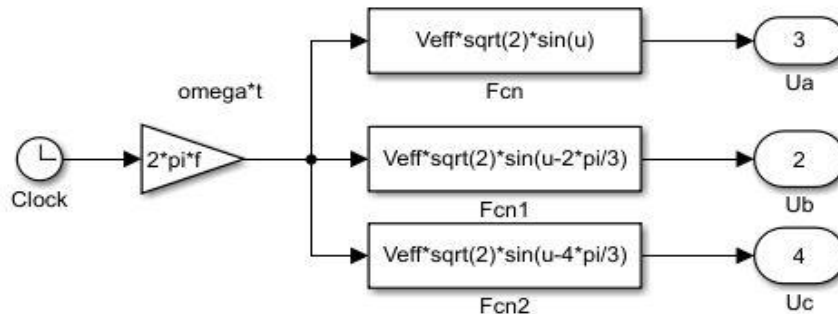


Figure III.2 : schéma bloc de la source triphasée sous Simulink.

Paramètre :

$$V_{eff}=220V$$

$$f=50Hz$$

III.2.1.2 Modélisation du redresseur

Et si on néglige l'effet d'empîtement, la tension de sortie du redresseur sera définie comme suite :

$$U_{red}(t) = \text{Max}[U_a(t), U_b(t), U_c(t)] - \text{Min}[U_a(t), U_b(t), U_c(t)] \quad (III.4)$$

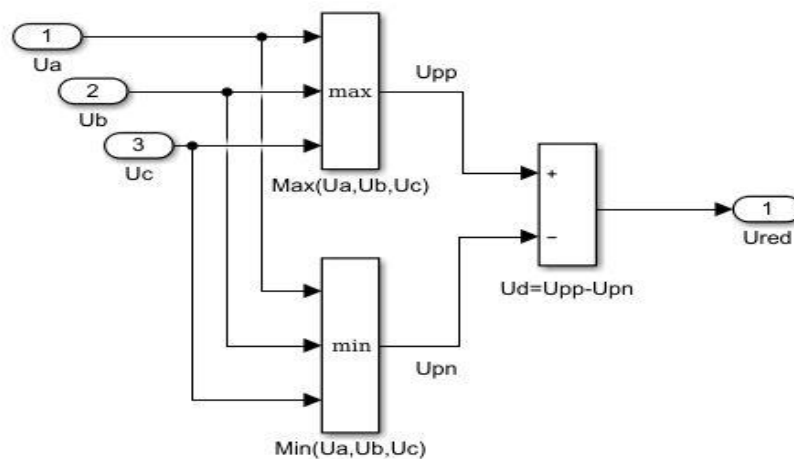


Figure III.3 : Schéma bloc du redresseur sous Simulink.

III.2.1.3 Modélisation du filtre

On utilise un filtre passe bas « LC », pour éliminer les harmoniques hautes fréquences. Ce filtre est schématisé par la figure suivante :

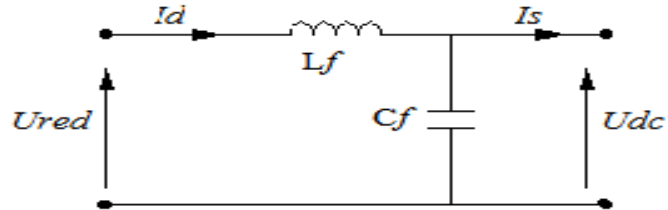


Figure III.4 Schéma du filtre passe bas.

C'est un filtre de deuxième ordre avec une pulsation de coupure égale à :

$$\omega_c = \frac{1}{\sqrt{C_f \cdot L_f}} \quad (III.5)$$

On peut tirer le filtre de «Butterworth» à partir de la bibliothèque Simulink/Signal processingBlockset/Filtering/Filter Designs/

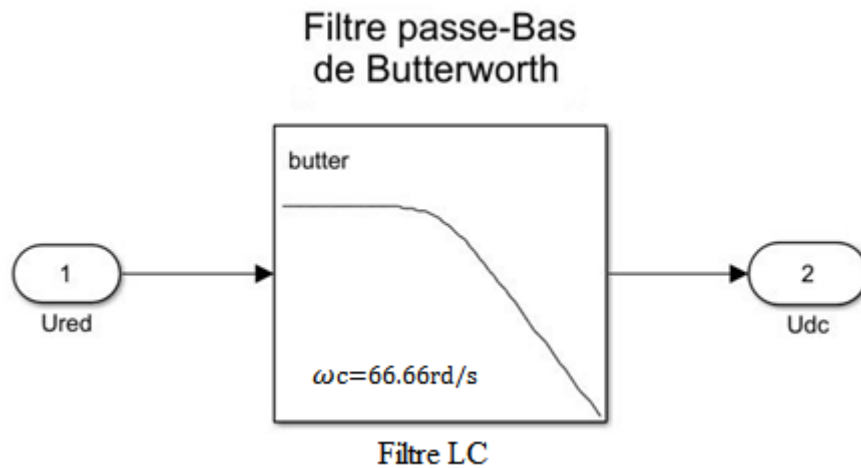


Figure III.5 : Schéma de filtre Butterworth.

Paramètre :

$L_f=400$ mH

$C_f=600$ mH

III.2.1.4 Modélisation hacheur

Les convertisseurs généralement utilisés dans les applications des MCC à vitesse variable, sont de type hacheur ou pont à thyristors. Dans ce travail on utilise un hacheur série, l'élément de base de cette hacheur est un transistor sous SIMULINK on utilise l'élément « Switch » dans la bibliothèque SIMULINK/Signal Routing.

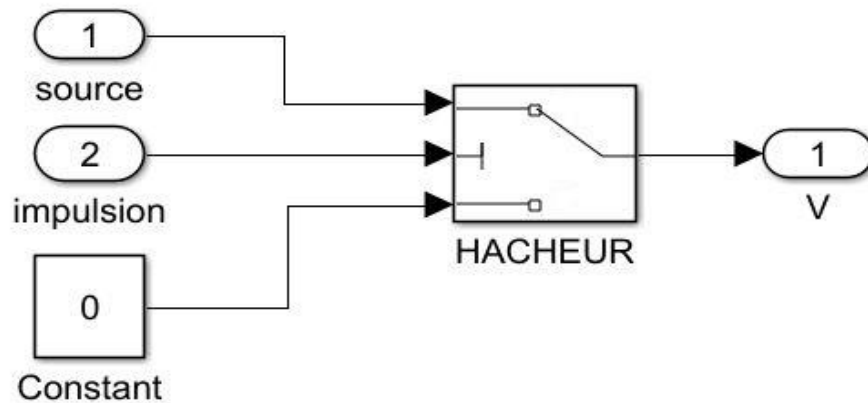


Figure III.6 : Schéma bloc du hacheur sous Simulink.

III.2.1.5 Modélisation du générateur d'impulsions

On utilise la technique MLI qui consiste à comparer la sortie (tension de commande) avec une porteuse de forme dents de scie.

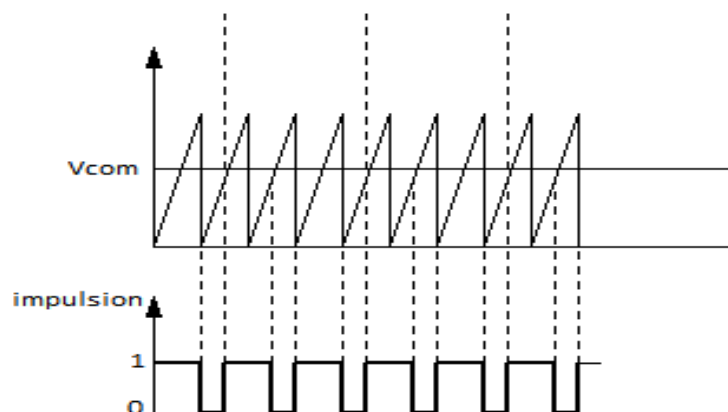


Figure III.7 : Schéma du principe de génération des impulsions.

Paramètre :

Intervalle du temps :

[0 1/1000]

Sorties :

[0 1]

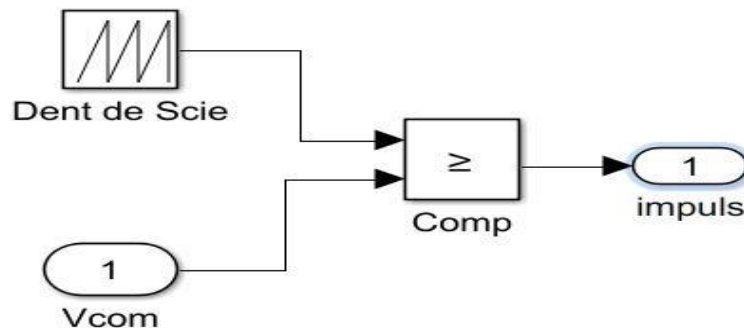


Figure III.8 : Générateur d'impulsion.

Avec les impulsions ainsi produite, on attaque la gâchette du transistor, on remarque qu'on variant la tension Vcom les impulsions changerons a leur tours ce qui provoque soit l'augmentation soit la diminution de la tension d'alimentation de la MCC donc une variation de la vitesse.

Le régulateur force cette vitesse à suivre la consigne.

III.3 Notion de boucle ouverte/fermée [23]

On distingue en général deux structures de commande : la commande en boucle ouverte et la commande en boucle fermée appelée également commande à contre-réaction.

III.3.1 Commande en boucle ouverte

Un système de commande est en boucle ouverte lorsqu'aucune mesure de sortie $y(t)$ n'est utilisée (ne comporte pas de contre-réaction) pour élaborer la commande $u(t)$ (figure III.9). Cela nécessite la connaissance d'un modèle de fonctionnement du système à commander. Par exemple la connaissance d'un modèle de fonctionnement d'un moteur à courant continu permettra de connaître la tension d'entrée qu'il faudra lui appliquer pour obtenir la vitesse de rotation désirée.

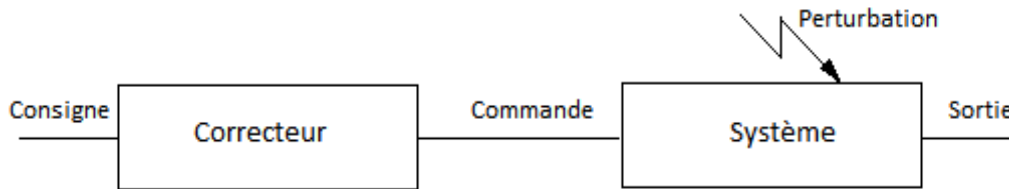


Figure III.9: Commande en boucle ouverte.

Cette solution est envisageable dans le cas où le système est parfaitement connu et modélisé et dans le cas où l'obtention d'une mesure de la sortie n'est pas économiquement possible.

Exemple :

Prenons l'exemple de l'asservissement de la température $y(t)$ d'une pièce chauffée par des radiateurs de chauffage central. Pour agir sur $y(t)$ on fait varier le débit d'eau dans les radiateurs par l'intermédiaire d'une petite vanne. Mais la température dans le local est aussi sensible à la température de l'eau qui circule, à la température extérieure, à l'ouverture d'une porte ou d'une fenêtre, au nombre de personnes qui entrent,etc.

Il s'agit d'une commande sans retour d'information de la grandeur de sortie.

La commande en boucle ouverte possède les inconvénients suivants :

- on ne peut pas commander un système qui subisse des problèmes d'instabilité (nous reviendrons plus en détail sur la notion de stabilité lors d'un prochain chapitre).
- correction difficile : n'ayant aucune information sur la sortie. L'opérateur ne peut élaborer aucune stratégie d'ajustement pour obtenir une sortie possédant la valeur souhaitée avec précision et rapidité.

III.3.2 Commande en boucle fermée

Afin de résoudre les problèmes de la commande en boucle ouverte et d'automatiser le système (supprimer l'action humaine) on introduit une boucle de retour (ou rétroaction). Dans cette stratégie de commande, une mesure de la sortie est utilisée et comparée avec la consigne par le correcteur (figure III.10).

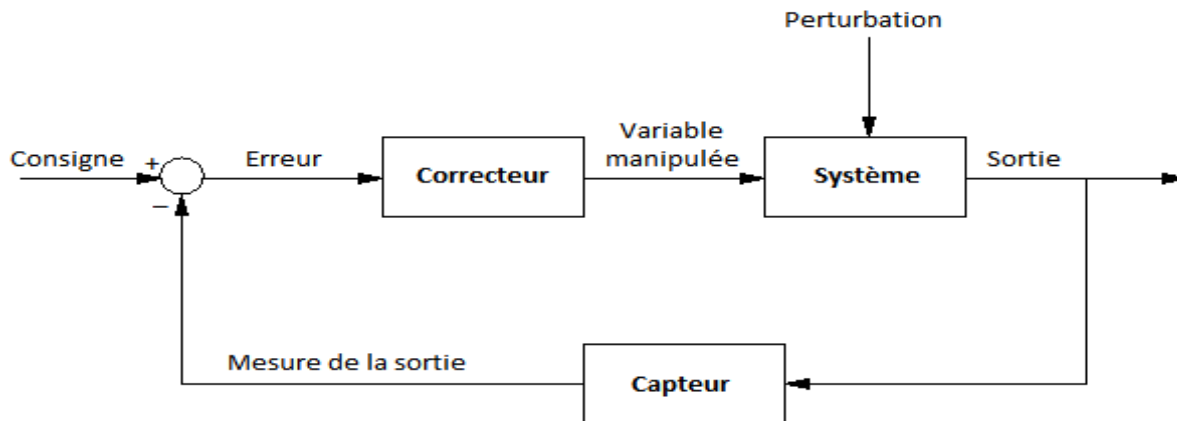


Figure III.10 : Commande en boucle fermée.

- **la consigne ($y_c(t)$)** : elle correspond à la valeur souhaitée en sortie.
- **la sortie commandée $y(t)$ (asservie)** : représente le phénomène physique que doit asservir le système de commande ('asservissement).
- **Écart ou erreur ($\varepsilon(t)$)** : représente la différence entre la consigne et la sortie.
- **Comparateur** : compare en permanence ce que l'on obtient à ce que l'on souhaite obtenir en sortie (élabore l'écart $\varepsilon(t)$).
- **Correcteur** : le correcteur (contrôleur traite le signal d'écart et détermine le signal de commande.
- **Actionneur** : il reçoit du correcteur le signal de commande et agit directement sur le système à commander. Il représente le "muscle" qui va piloter l'évolution système (par exemple : amplificateur de puissance, moteur, vérin, vanne, etc.).
- **Capteur** : organe de mesure qui donne une image aussi fidele que possible de la sortie et la transforme en un signal compréhensible par le comparateur (le plus souvent électrique).
- La sensibilité du capteur impose donc les limites de la précision de l'asservissement (par exemple : accéléromètre, thermomètre, gyroscopes, ...).

On peut donc définir un asservissement (système asservi) comme un système bouclé (boucle fermée) dont le fonctionnement tend à annuler l'écart entre une grandeur commandée (la sortie $y(t)$) et une grandeur de commande (la consigne $y_c(t)$).

III.4 Synthèse du régulateur PI

On rappelle que le rôle de régulateur est de maintenir la grandeur régulée à une valeur de la consigne malgré la présence des perturbations dans le fonctionnement en régulation ou suivre la variation d'une consigne dans le fonctionnement en asservissement. Au moyen du choix des actions et de leurs paramètres, il est possible d'obtenir un comportement désiré en boucle fermée, caractérisant les performances du système. Plusieurs méthodes de la synthèse des paramètres du régulateur développées :

- Méthode de Ziegler et Nichols.
- Réglage en ligne des paramètres du régulateur PID.
- Méthode du modèle (régulation idéale, régulation parfaite).
- Méthode fréquentielle.

III.4.1 spécifications du cahier de charge [24]

Dans cette étude, la commande du MCC en boucle fermée pour assurer les spécifications du cahier de charge suivant :

- Stabilité du système.
- Système suivi trajectoire de référence.
- Dépassement négligeable.
- Rapidité (temps de montée moins de 0.01s, temps d'établissement moins de 0.03s).
- La précision (erreur statique nulle).

Pour traduire ces spécifications du cahier de charge en régulateur classique type PI on a utilisé les méthodes du modèle (régulation idéale) sont très nombreuses et reposent sur la connaissance d'un modèle précis du système à commander. Les performances réelles obtenues dépendent de la qualité du modèle et son aptitude à représenter le mieux possible le procédé. Parmi ces méthodes on propose deux variantes de la méthode du modèle : une basée sur la donnée de la fonction de transfert en boucle fermée répondant à un cahier de charge et l'autre basée sur la donnée de la fonction de transfert en boucle ouverte.

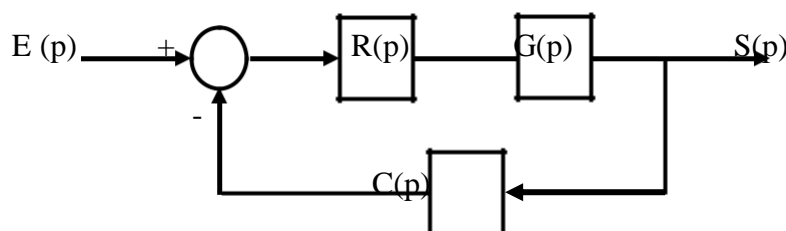


Figure III.11 : Structure du système en boucle fermée. [24]

La fonction de transfert en boucle ouverte est $FTBO = R(p) G(p)$ (III.6), $C(p)=1$

On appelle régulation idéale si la fonction de transfert en boucle ouverte est de la forme :

$$R(p) \cdot G(p) = \frac{1}{Tp} \quad (\text{III.7})$$

La constante de temps T est donnée et la fonction de transfert G(p) du procédé est connue, on déduit alors l'expression du régulateur R(p) par :

$$R(p) = \frac{1}{G(p) \cdot Tp} \quad (\text{III.8})$$

En conséquence, la fonction de transfert F(p) en boucle fermée est donnée par :

$$FTBF = \frac{R(p) \cdot G(p)}{1 + R(p) \cdot G(p)} = \frac{\frac{1}{Tp}}{1 + \frac{1}{Tp}} = \frac{1}{TP + 1} \quad (\text{III.9})$$

III.4.2 Détermination des gains de régulateur PI

$$G(p) = \frac{\Omega_s}{Uas} = \frac{\frac{K_T}{R_a f r + K_T \cdot K_e}}{1 + \frac{R_a J}{R_a f r + K_T \cdot K_e} S} \quad (\text{III.10})$$

On a :

$$Km = \frac{K_T}{R_a f r + K_T \cdot K_e} \quad (\text{III.11})$$

$$Tm = \frac{R_a J}{R_a f r + K_T \cdot K_e} \quad (\text{III.12})$$

Alors :

$$G(p) = \frac{Km}{1 + Tm \cdot S} \quad (\text{III.13})$$

$$R(p) \cdot G(p) = \frac{1}{Ts} \quad (\text{III.14})$$

$$R(p) = \frac{1}{Ts \cdot G(s)} = \frac{1}{Ts} \frac{1 + Tm \cdot S}{Km} = \frac{Tm \cdot S + 1}{Ts \cdot Km} \quad (\text{III.15})$$

$$R(p) = Kp + \frac{Ki}{S} \quad (\text{III.16})$$

$$Kp + \frac{Ki}{S} = \frac{Tm.S + 1}{TS.Km} \quad (III. 17)$$

$$Kp + \frac{Ki}{S} = \frac{Tm.S}{TS.Km} + \frac{1}{TS.Km} \quad (III. 18)$$

$$Kp = \frac{Tm}{Km.T} \quad (III. 19)$$

$$Ki = \frac{1}{Km.T} \quad (III. 20)$$

Le régulateur PI est modélisé par la fonction du transfert G(s), avec:

$$G_s = \frac{sKp+Ki}{s} = Kp + \frac{1}{s}Ki \quad (III. 21)$$

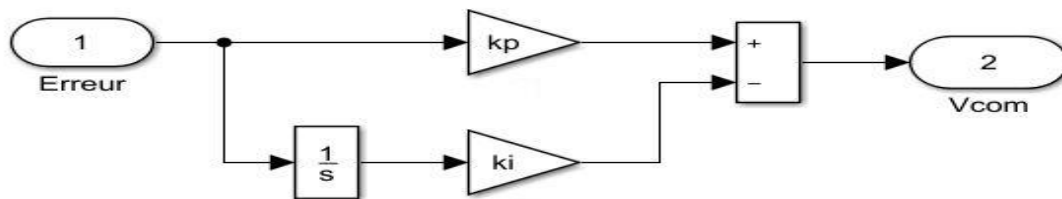


Figure III.12 : Schéma bloc du régulateur PI sous Simulink.

Paramètres :

- Kp=100

- Ki=0.7

III.5 Modélisation de la MCC [22]

La modélisation de l'ensemble moteur + charge est réalisable à partir des équations de base de la machine à courant continu et du principe fondamental de la dynamique (PFD):

Equations électromécaniques :

$$\left\{ \begin{array}{l} C_e = K \cdot I_a \\ E = K \cdot \Omega_m \end{array} \right\} \text{ Avec : } K=K_1 \cdot \phi \quad (III. 22)$$

Equation électrique (second loi de Kirchhoff):

$$Va = E + R_a I_a + L_a \frac{dI_a}{dt} \quad (III.23)$$

Equation mécanique (second loi de Newton) :

$$J \frac{d\Omega_m}{dt} + f_r \Omega_m = C_e - C_r \quad (III.24)$$

Ce: le couple électromagnétique fourni par le moteur.

E: la force contre électromotrice générée par le moteur.

V : la tension d'induit; Ra, La : la résistance et l'inductance d'induit.

J: le moment d'inertie du moteur.

fr: coefficient de frottement visqueux de l'ensemble moteur + charge.

On insère l'opérateur de Laplace p-noté «s» dans MATLAB-dans le système d'équation son obtient :

$$\begin{cases} Va = E + R_a I(p) + L_a p I_a(p) \\ J p \Omega_m(p) + f_r \Omega_m(p) = C_e - C_r \end{cases} \quad (III.25)$$

$$\begin{cases} Va - E = (R_a + L_a p) I_a(p) \\ (J p + f_r) \Omega(p) = C_e - C_r \end{cases} \quad (III.26)$$

$$\begin{cases} \frac{I_a(p)}{Va - E} = \frac{1}{R_a + L_a(p)} \\ \frac{\Omega_m(p)}{C_e - C_r} = \frac{1}{(J p + f_r)} \end{cases} \quad (III.27)$$

La figure suivante présente le schéma block d'un MCC à excitation séparé :

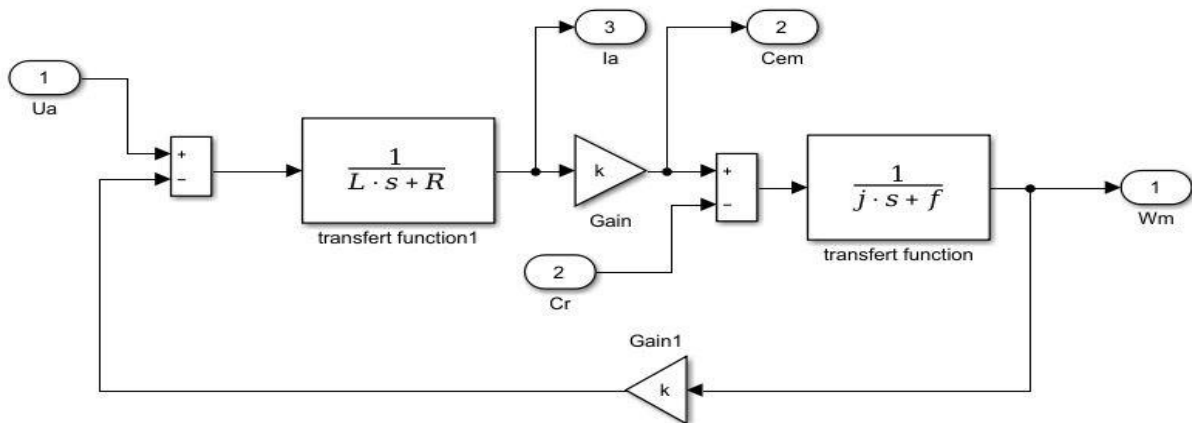


Figure III.13 : Schéma block d'un MCC à excitation séparé.

La figure suivante présente le schéma bloc global du système étudié sous Simulink :

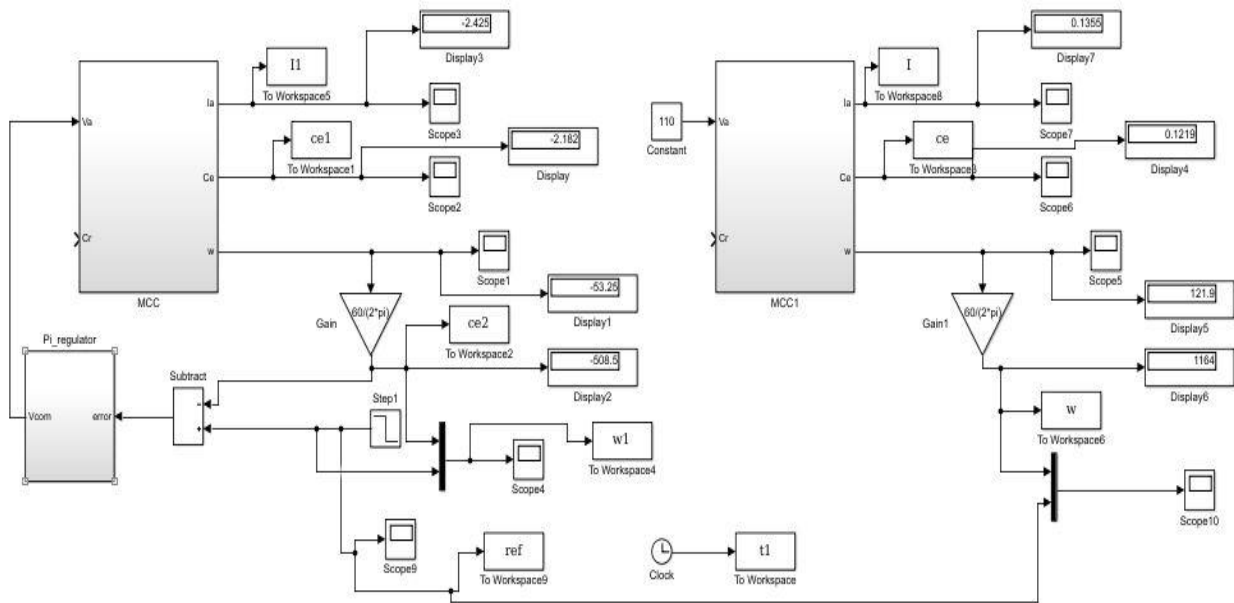


Figure III.14 : Schéma bloc global du système étudié sous Simulink.

III.6 Résultats des simulations et interprétation

Pour compléter notre étude, et faciliter la compréhension de ce qui à été fait précédemment, nous avons choisi, d'effectuer une simulation de chaque partie indépendamment, ensuite, un système globale de la commande.

III.6.1 Simulation de la commande en boucle ouverte

La figure suivante présente la simulation de la commande en boucle ouverte :

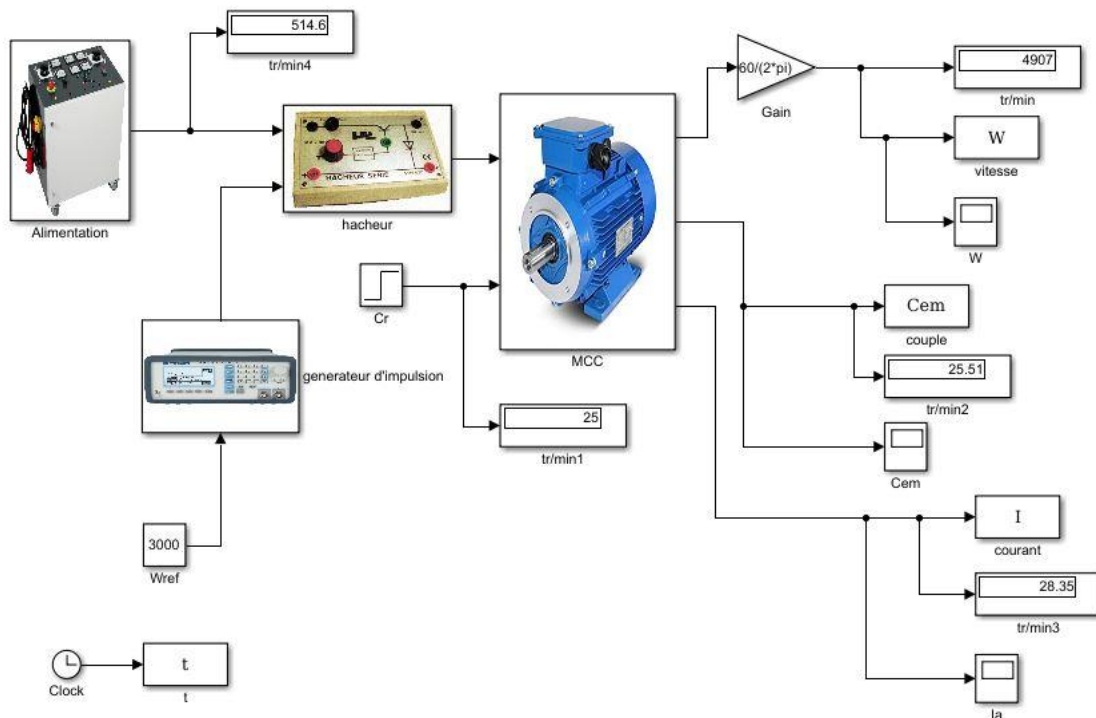


Figure III.15 : Simulation de la commande en boucle ouverte.

1- A vide :

On impose une consigne de vitesse avec un couple résistant nul ($Cr = 0N:m.$)

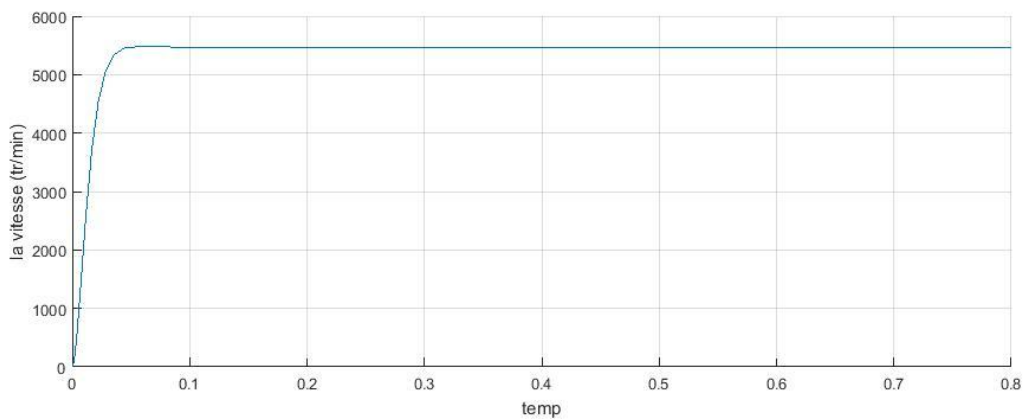


Figure III.16 : Réponse indicielle en vitesse.

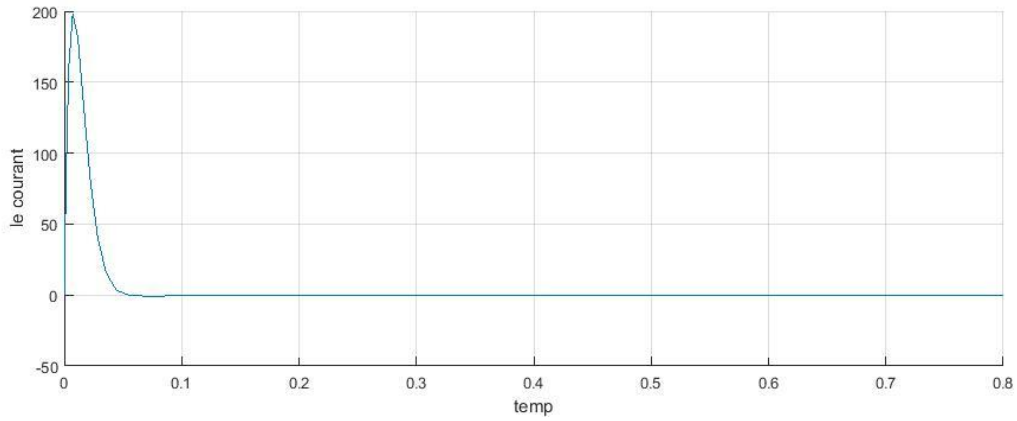


Figure III.17 : Réponse indicielle en courant.

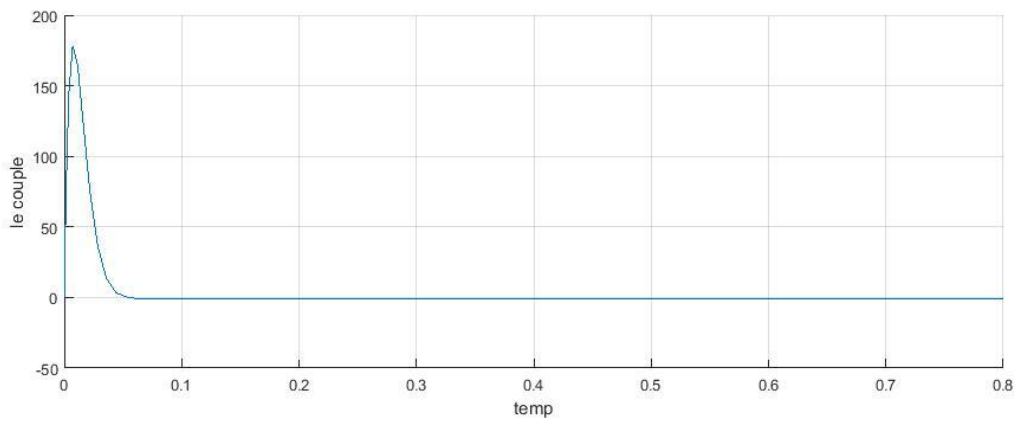


Figure III.18 : Réponse indicielle en couple.

1- En charge :

On impose la même consigne de vitesse ($V_{réf} = 3000 \text{ tr/min}$), mais cette fois on applique un couple résistant ($C_r = 25 \text{ N.m}$).

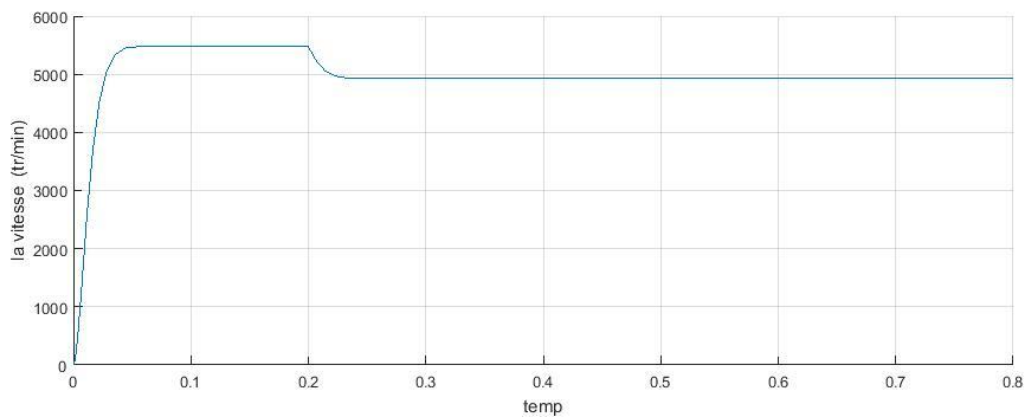


Figure III.19 : Réponse indicielle en vitesse.

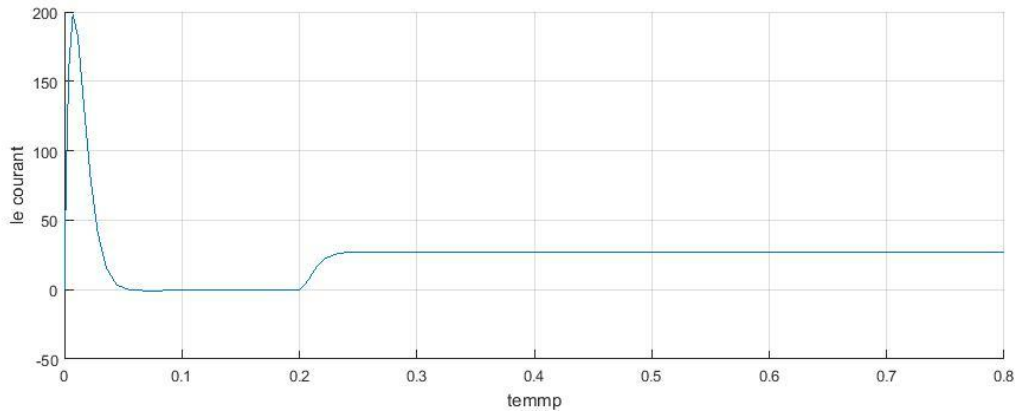


Figure III.20 : Réponse indicielle en courant.

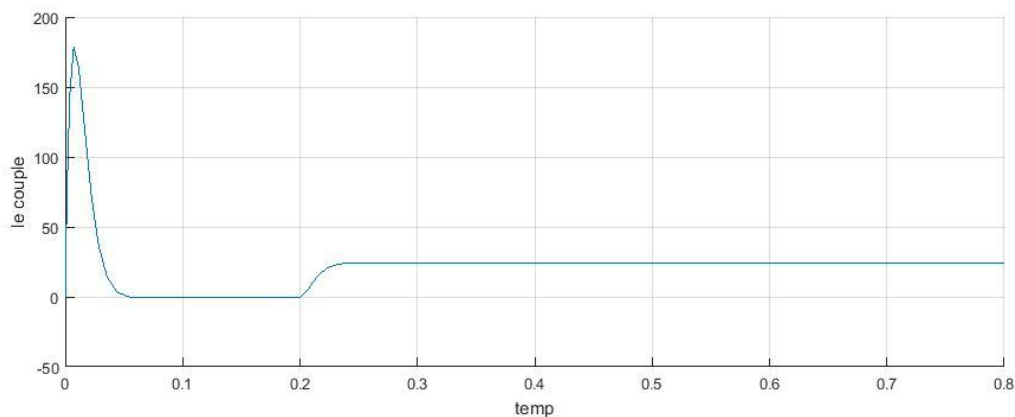


Figure III.21 : Réponse indicielle en couple.

III.6.1.1 Interprétations des résultats de simulation en boucle ouverte

1- A vide :

La vitesse de rotation atteint, la vitesse nominale (5448 tour/min) après une période transitoire qui dure environs 0.04 sec (figure III.16). Ainsi, la partie transitoire du courant est équivalente a celle de la vitesse (0.04 sec) avec une valeur crête d'environ 200 A, ensuite, le courant reprend le régime nominal avec une valeur de 0.02 A (figure III.17).

En fin, la partie transitoire du couple est équivalente a celle de la vitesse (0.04 sec) avec une valeur crête d'environ 180 N.m, ensuite, le couple reprend le régime nominal avec une valeur de 0.02 N.m (figure III.18).

2- En charge :

On démarre le moteur à vide, puis on applique une charge $C_r = 25 \text{ N.m}$ à $t = 0.2 \text{ sec}$, par conséquent, la vitesse chute d'une valeur de 448 tr/min puis se stabilise aux alentours de

5000tr/min (figure III.19). Pour le courant en charge, on constate la présence d'un pic qui atteint 200A, puis reprend son régime permanent après 0.04 s. Lors de l'introduction de la charge a $t=0.2\text{sec}$, on remarque une augmentation considérable qui arrive jusqu'à 25A (figure III.20).

Pour le couple en charge, on constate la présence d'un pic qui atteint 180A, puis reprend son régime permanent après 0.04 s. Lors de l'introduction de la charge a $t=0.2\text{sec}$, on remarque une augmentation considérable qui arrive jusqu'à 25 N.m (figure III.21).

Cela est dû essentiellement à l'absence du feedback vu qu'on travaille en boucle ouverte. Pour rappel, les essais en boucle ouverte sont faits pour déterminer les caractéristiques de notre moteur.

III.6.2 Simulation de la commande en boucle fermée

La figure suivante présente la simulation de la commande en boucle fermé

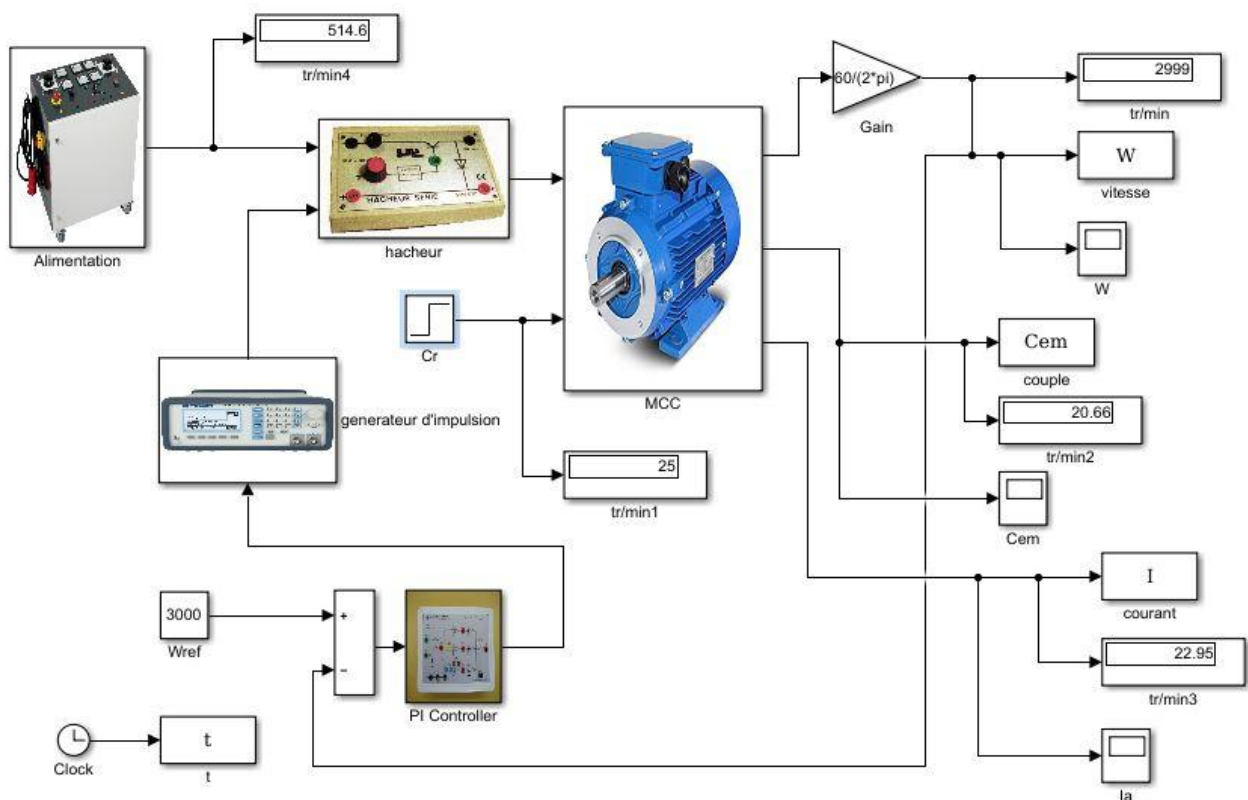


Figure III.22 : Simulation de la commande en boucle fermé.

1- A vide :

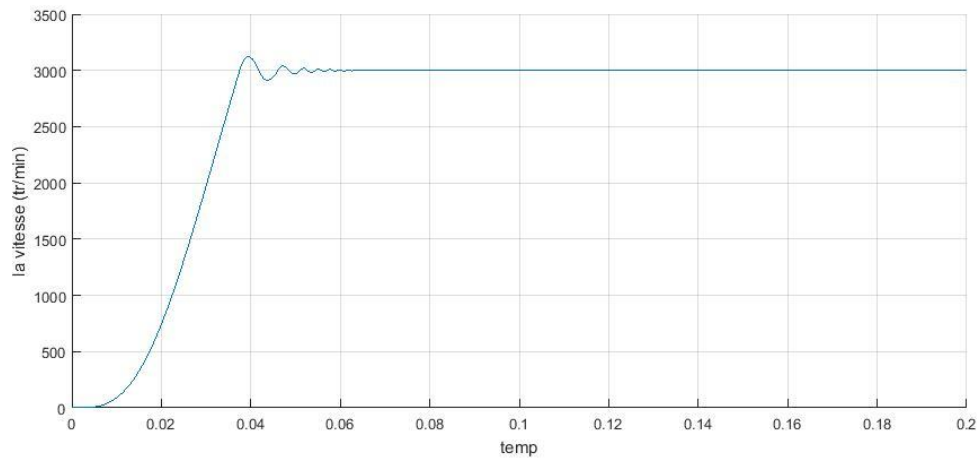


Figure III.23 : Réponse indicielle en vitesse.

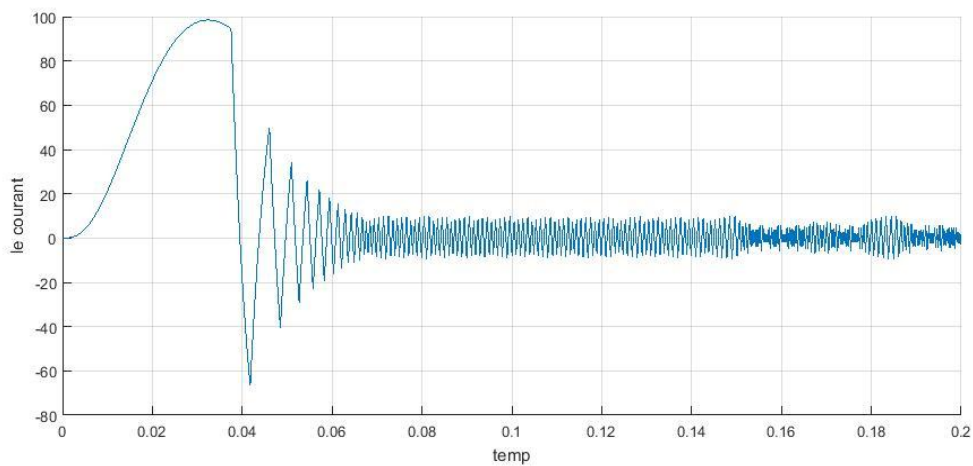


Figure III.24 : Réponse indicielle en courant.

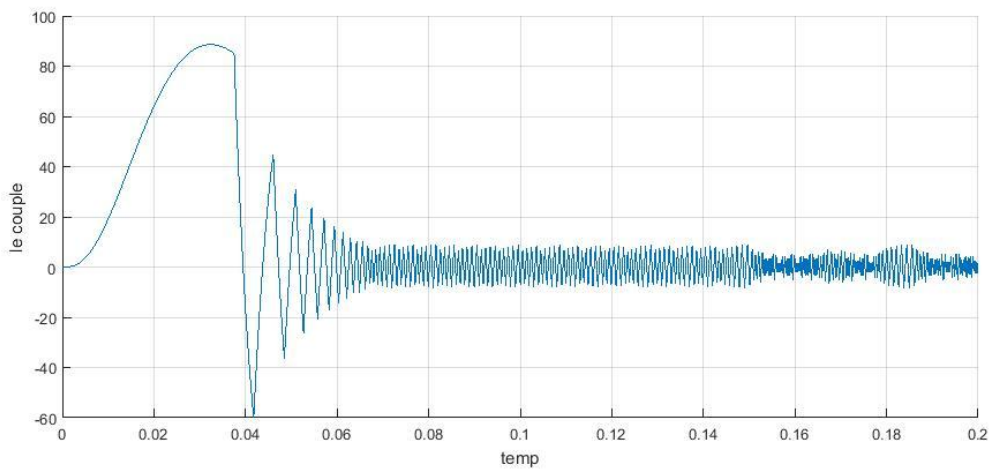


Figure III.25 : Réponse indicielle en couple.

2- En charge :

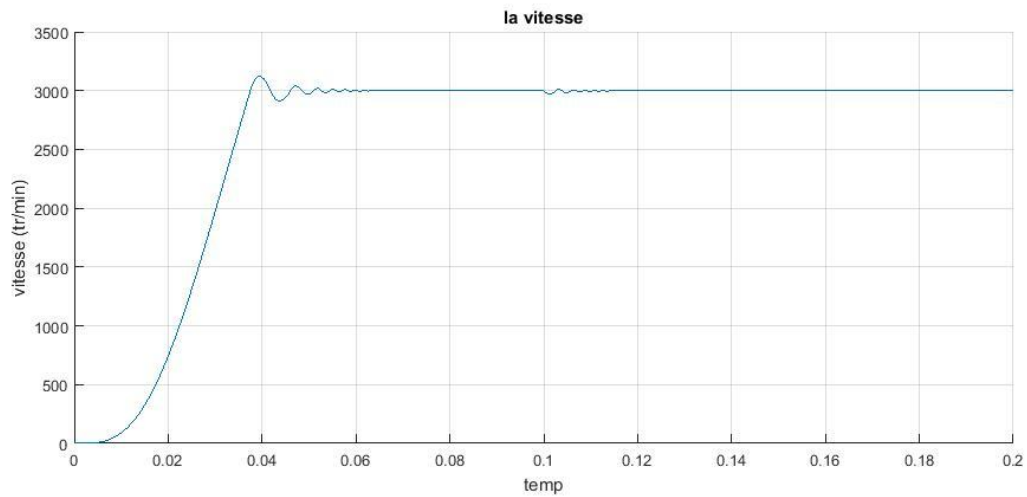


Figure III.26 : Réponse indicielle en vitesse.

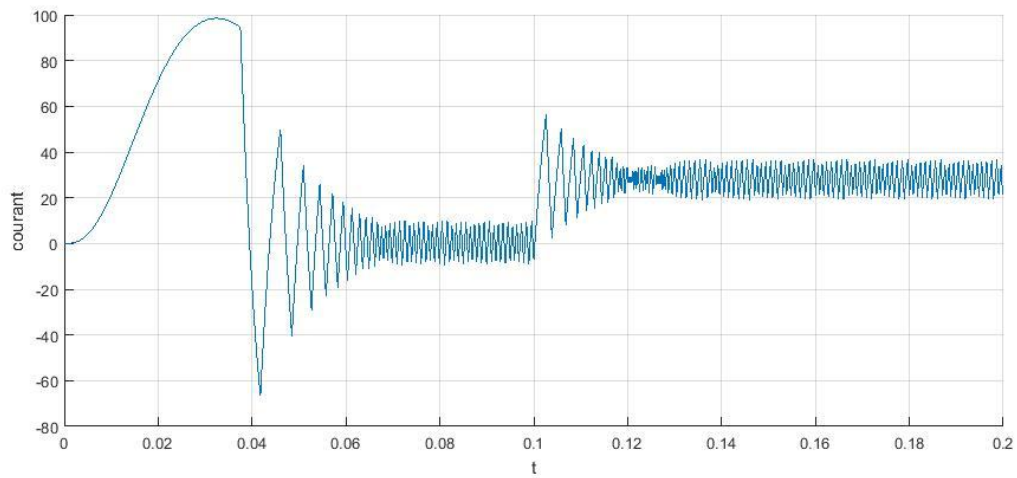


Figure III.27 : Réponse indicielle en courant.

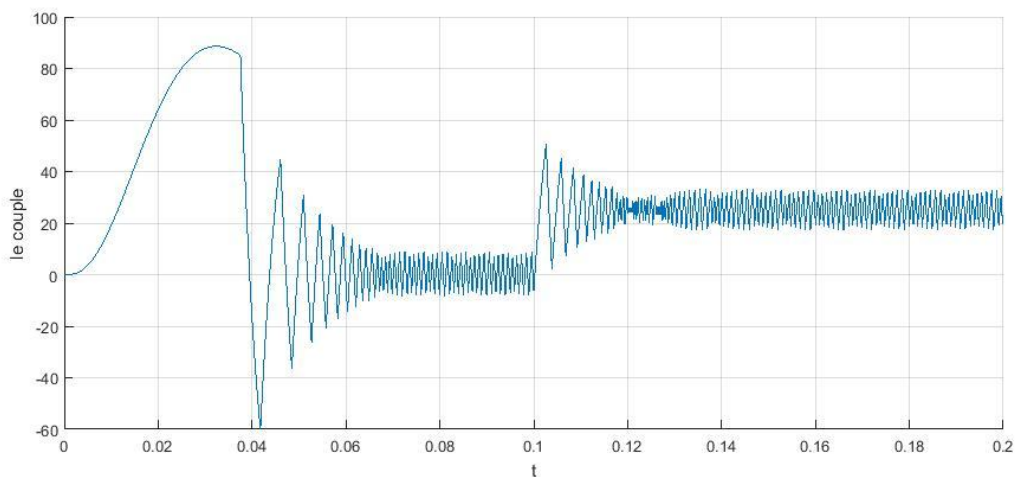


Figure III.28 : Réponse indicielle en couple.

III.6.2.1 Interprétations des résultats de simulation en boucle fermée

1- A vide :

L'analyse des courbes obtenues montre que la poursuite pour une consigne de vitesse de 3000 tour/min est satisfaisante avec un dépassement acceptable et un faible temps de réponse, environ 0.06s (figure III.23).

Le transitoire du courant a une durée équivalente au temps de démarrage avec une valeur de crête d'environ 100A. Ce courant atteint la valeur presque nulle (0.02 A) qui correspond au courant magnétisant (figure III.24).

Le transitoire du couple a une durée équivalente au temps de démarrage avec une valeur de crête d'environ 90N.m. Ce couple atteint la valeur presque nulle (0.03 N.m) qui correspond au couple électromagnétique (figure III.25).

2- En charge :

On applique une charge après 0.1 sec la vitesse chute puis retourne à sa valeur de référence (figure III.26).

D'après (figure III.27 et figure III.28), on remarque que le courant et le couple augmentent au moment de l'application de la charge à 2sec, afin de compenser le couple résistant.

III.7 Présentation et étude sous interface graphique GUI réalisé sous MATLAB

Pour commander le moteur à courant continu à excitation séparé en boucle ouvert et fermé et analyser et visualiser les différents comportements dynamiques du système étudié, nous commander le modèle réalisé sous Simulink, cette interface graphique affiche et visualise les résultats de simulation et les performances dynamiques et statiques.

Cette démarche avec le GUI MATLAB nous permet :

- D'analyser le système étudié (avant et après la commande).
- D'effectuer le contrôle du système à partir de régulateur PI.
- De visualiser les résultats de la régulation avec la simulation de notre système.
- De calculer les paramètres dynamiques du système.
- De varier la vitesse et sens de vitesse du MCC.
- De Contrôler la vitesse du MCC.

III.7.1 Présentation de l'interface graphique réalisée du système étudié

L'interface graphique GUI réalisé sous MATLAB pour l'exécution de différentes opérations, est représenté à la figure suivante :

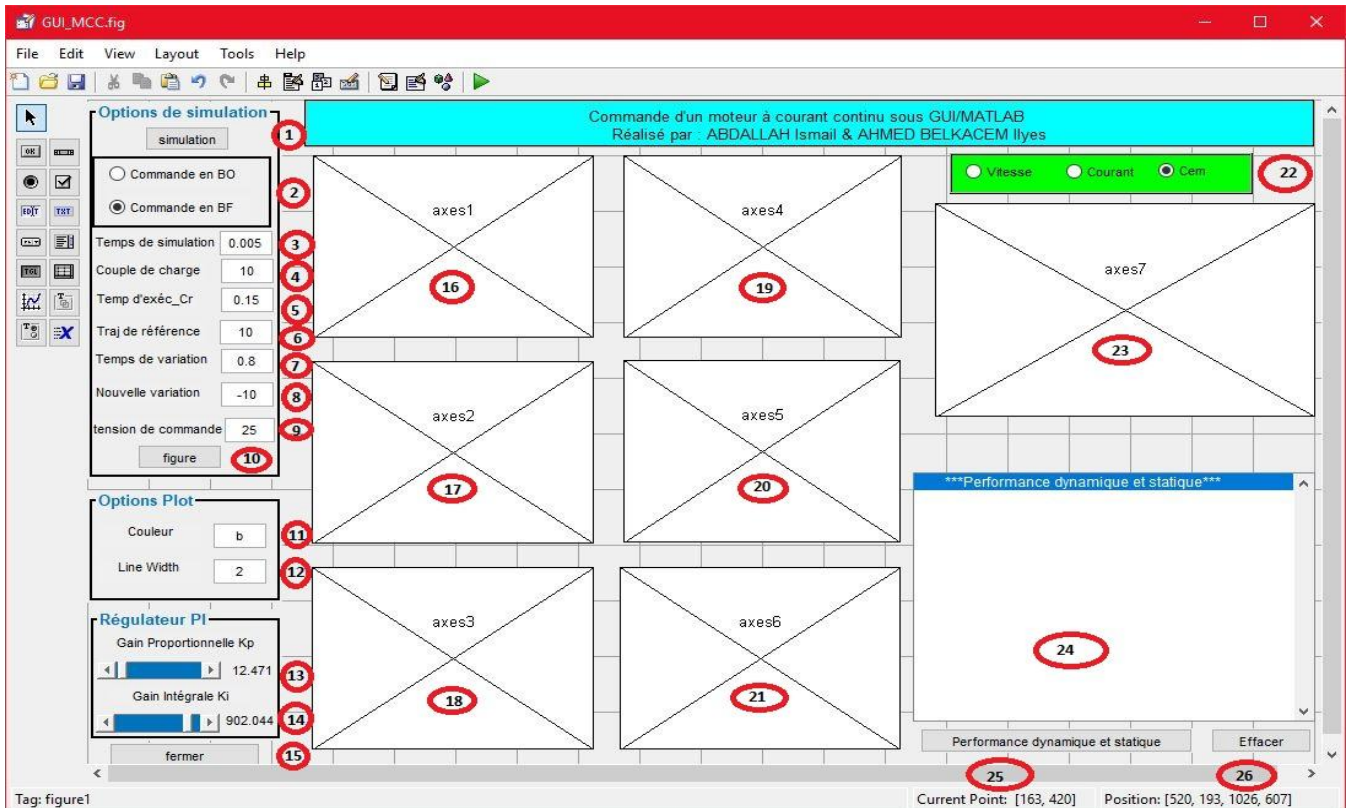


Figure III.29 : Interface graphique réalisé sous GUI MATLAB pour visualiser les résultats de simulation.

1. Push button pour lancer la simulation.
2. Deux radio buttons pour la commande en BO ou BF.
3. Edit text pour changer le temps de simulation.
4. Edit text pour changer le couple de charge appliqué.
5. Edit text pour changer le temps d'exécution de couple de charge.
6. Edit text pour changer la trajectoire de référence.
7. Edit text pour changer le temps de variation de la trajectoire de référence.
8. Edit text pour changer la nouvelle variation de la trajectoire de référence.
9. Edit text pour changer la tension de commande.
10. Push Button pour ploter les figures de simulation.

11. Edit text pour changer la couleur de courbes simulées.
12. Edit text pour changer l'épaisseur de la ligne des courbes simulées.
13. Slider pour changer les paramètres du régulateur Kp.
14. Slider pour changer les paramètres du régulateur Ki.
15. Push button pour fermer l'interface graphique.
16. Axes pour visualiser la courbe de la tension triphasée.
17. Axes pour visualiser la courbe de la tension redressée.
18. Axes pour visualiser la courbe de la tension filtrée.
19. Axes pour visualiser la courbe de la tension de commande et signal dent de scie.
20. Axes pour visualiser la courbe des impulsions.
21. Axes pour visualiser la courbe de l'induit.
22. Trois radios buttons pour tracer la courbe de la vitesse, le courant ou le couple.
23. Axes pour visualiser la courbe de la vitesse, le courant ou le couple.
24. Listbox pour visualiser les performances dynamiques et statiques du système.
25. Push button pour afficher les performances dynamiques et statique dans Listbox
26. Push button pour effacer les performances dynamiques et statique dans Listbox.

III.7.2 Etude sous interface graphique GUI réalisé sous MATLAB

Pour étudier un système il nous faire commande en boucle ouvert et fermé dans le mode de fonctionnement à vide et en charge.

III.7.2.1 Commande en boucle ouverte

a. Fonctionnement à vide :

Pour $\alpha=0.5$:

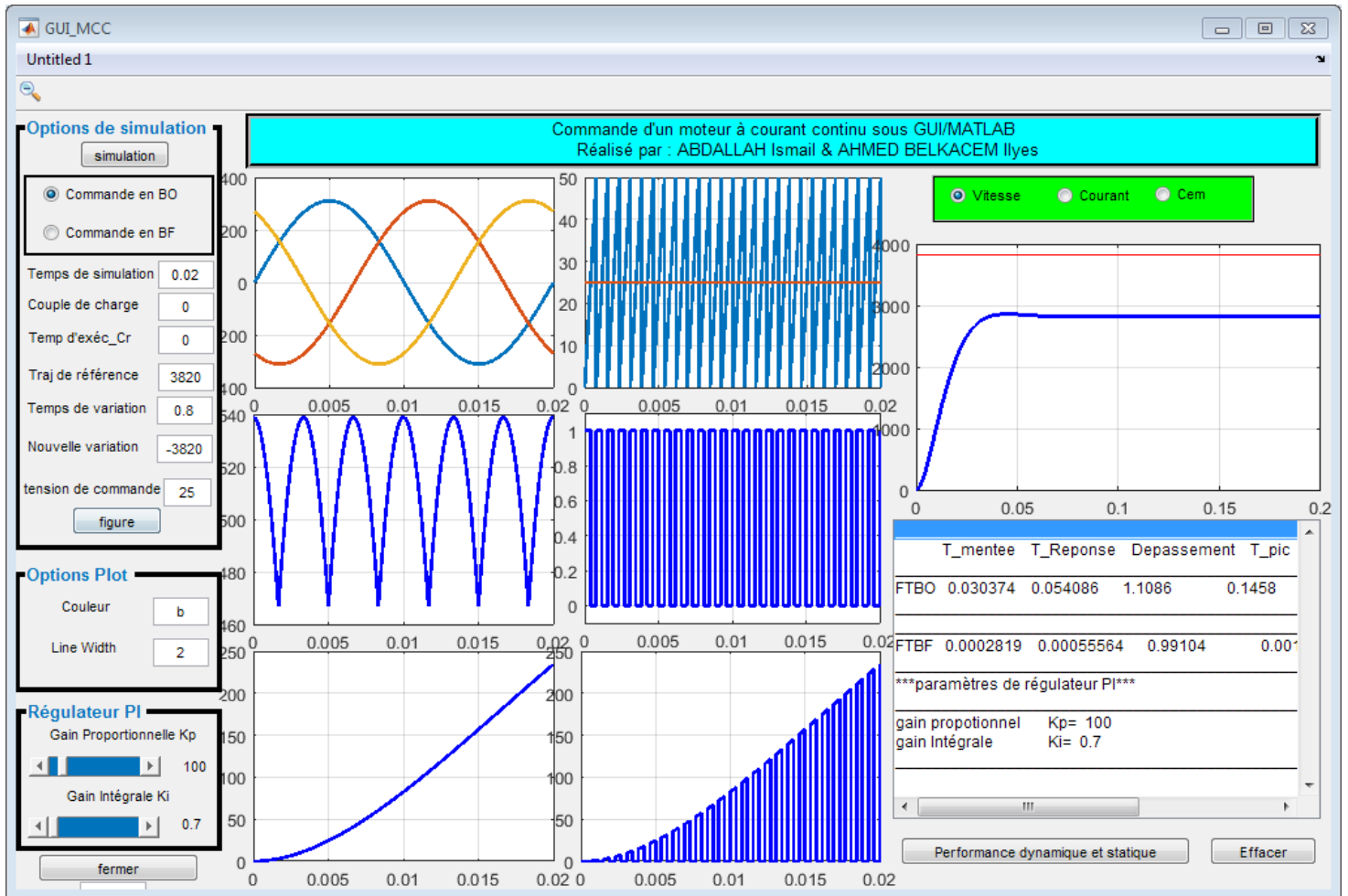
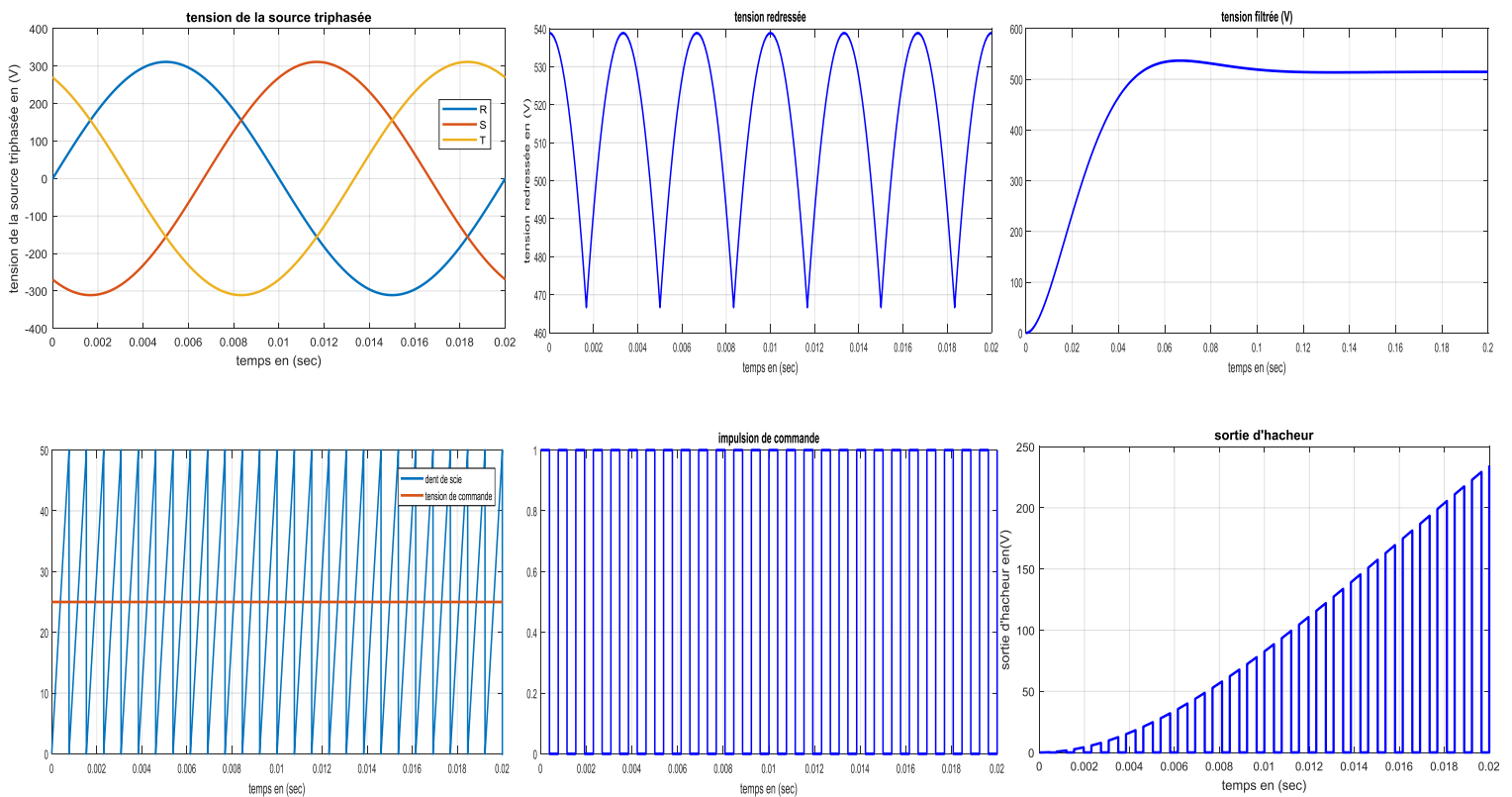


Figure III.40 : Simulation de l'interface graphique réalisée de la commande en boucle ouvert à vide.



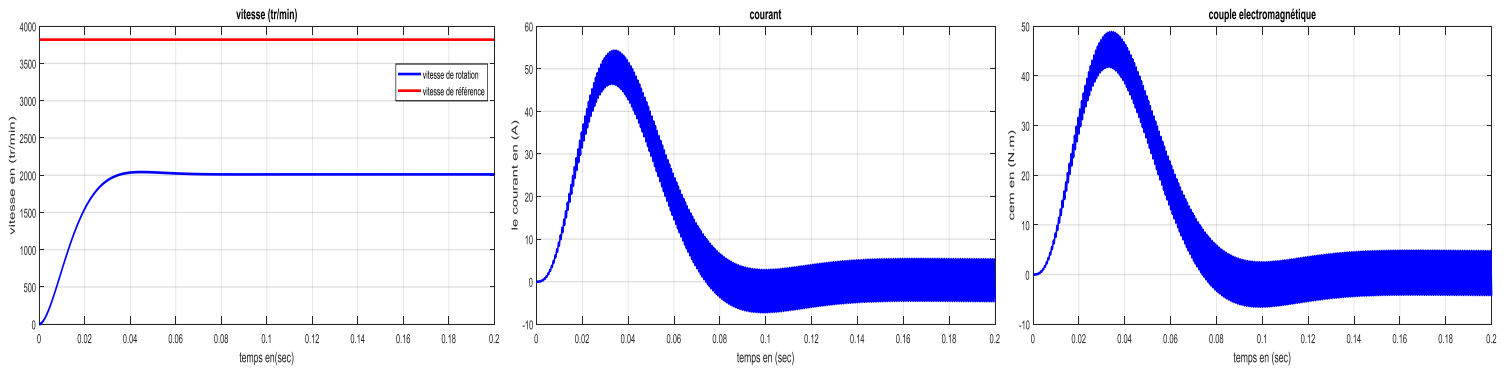


Figure III.31 : Résultats de simulation de l'interface graphique réalisée de la commande en boucle ouvert à vide.

Pour $\alpha=0.7$

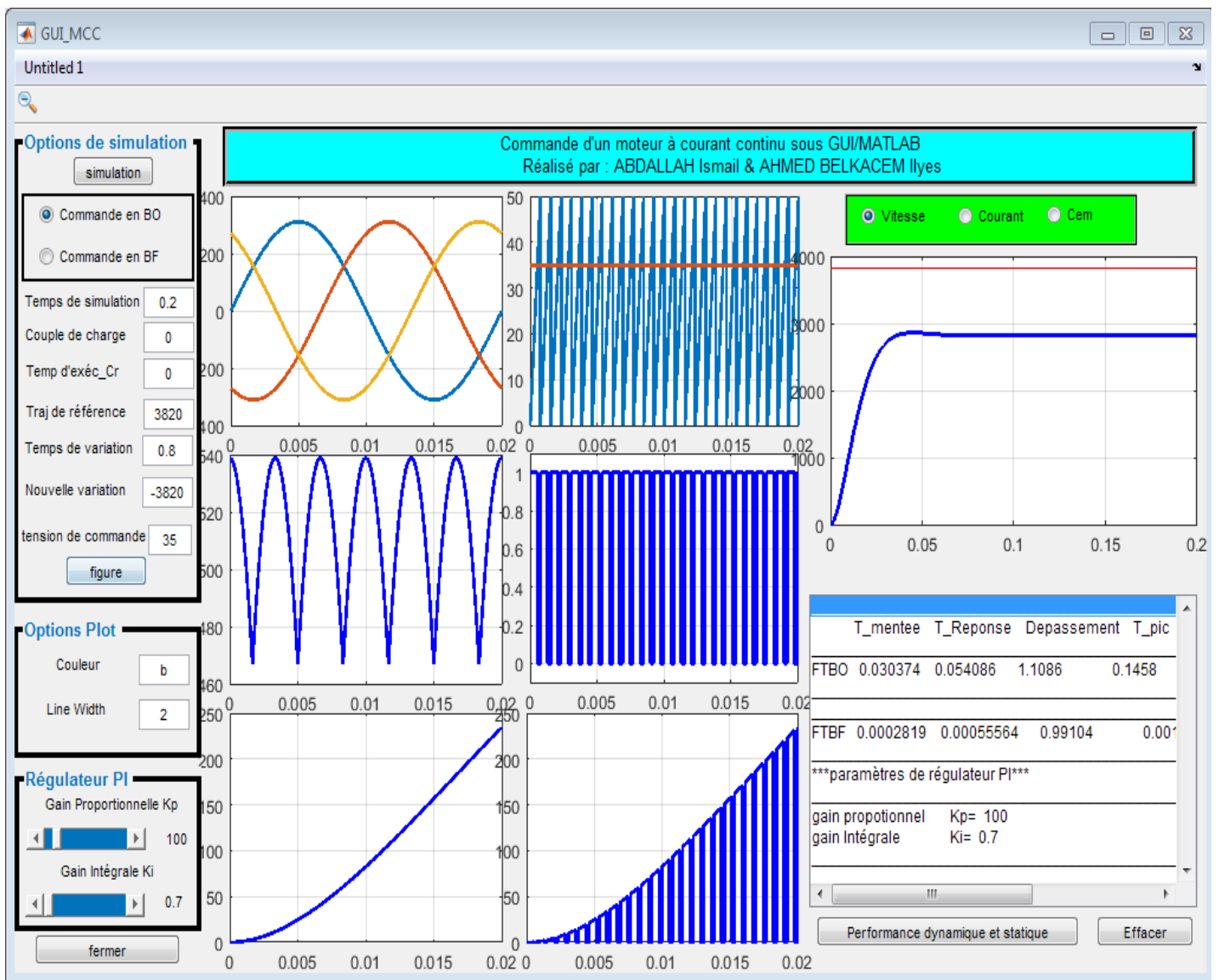


Figure III.52 : Simulation de l'interface graphique réalisée de la commande en boucle ouvert en charge.

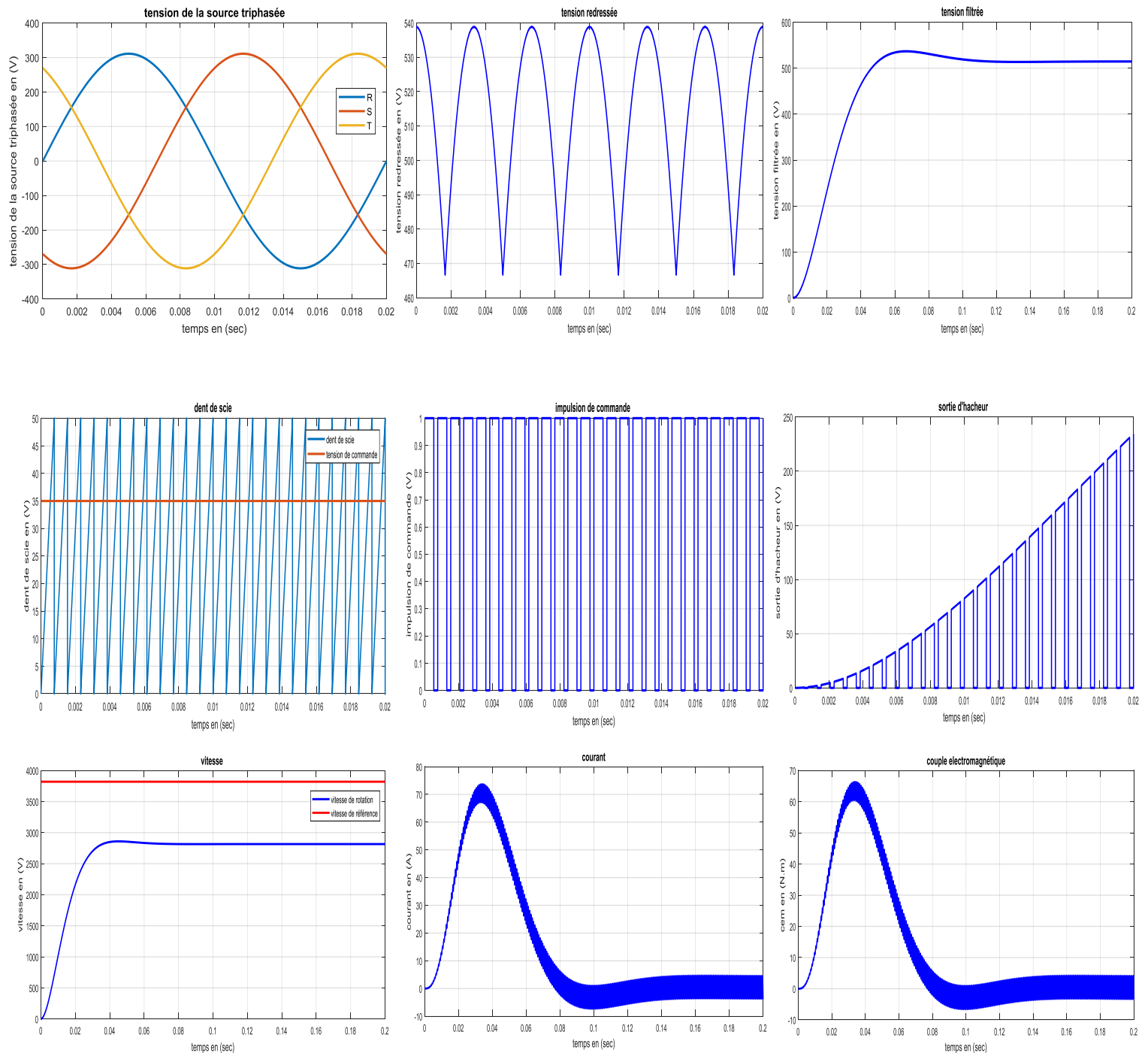


Figure III.33 : Résultats de simulation de l'interface graphique réalisée de la commande en boucle ouverte à vide.

b. Fonctionnement en charge

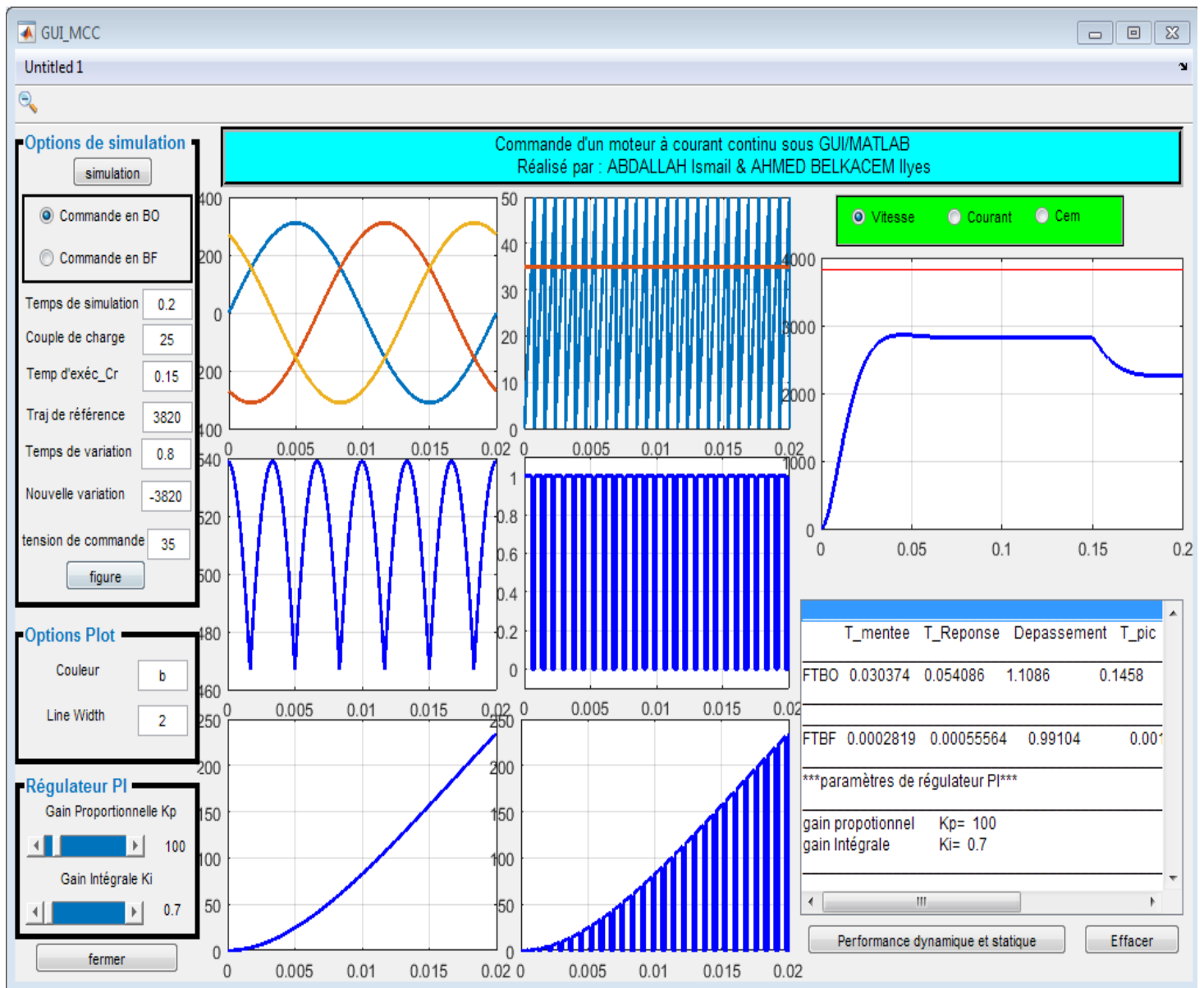
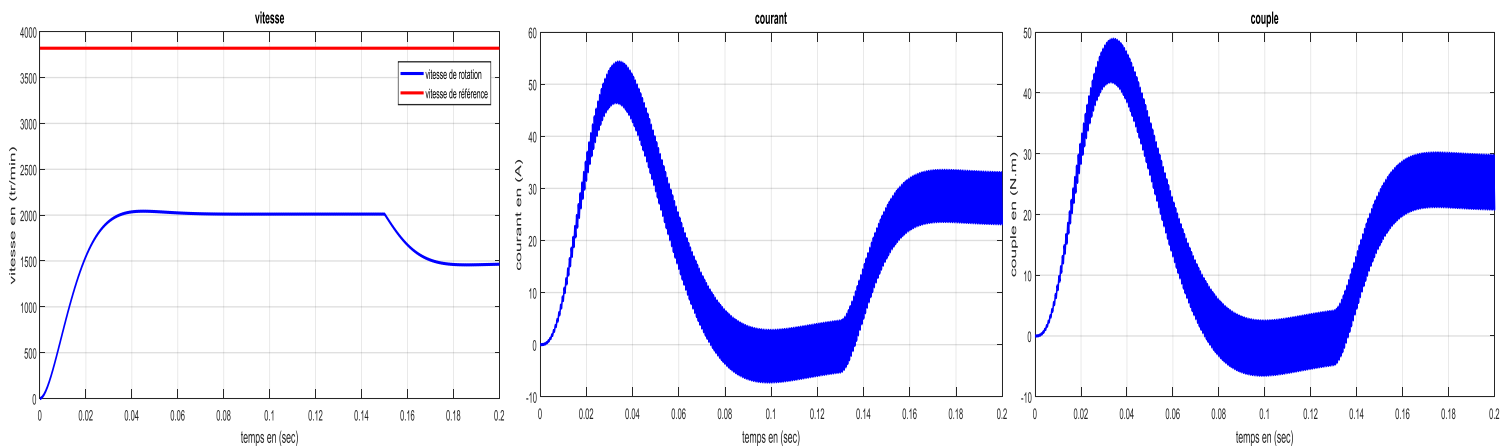


Figure III.34 : Simulation de l'interface graphique réalisée de la commande en boucle ouvert en charge.

Pour $\alpha=0.5$:



Pour $\alpha=0.7$:

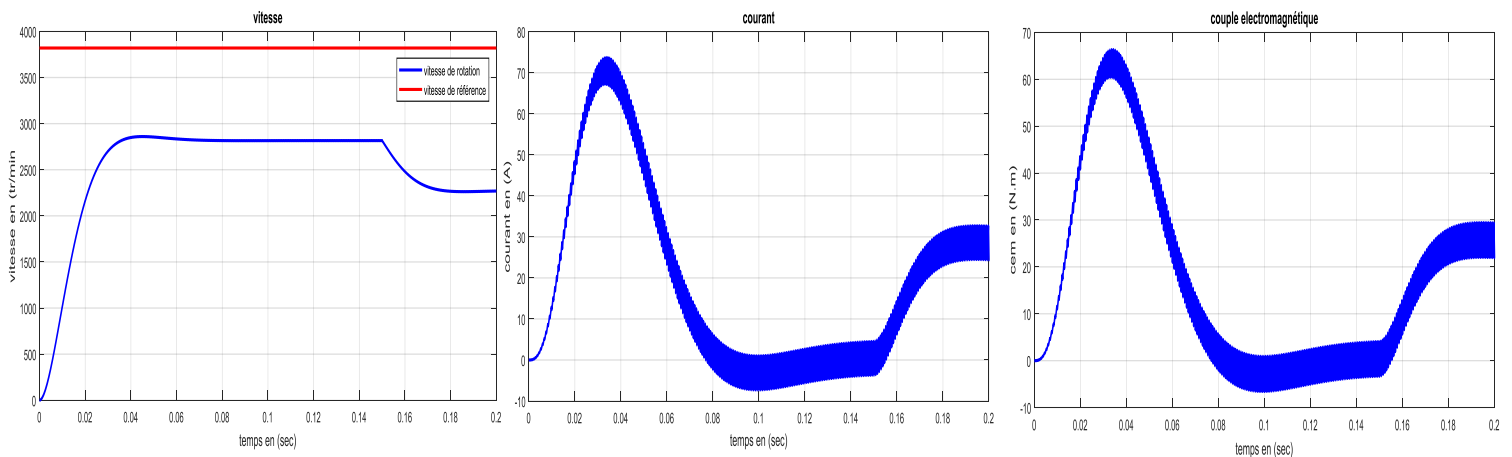


Figure III.35 : Résultats de simulation de l'interface graphique réalisée de la commande en boucle ouvert en charge.

III.7.2.2 Commande en boucle fermée

- calcul des paramètres du régulateur PI :

Pour synthétiser un régulateur satisfaisant les spécifications du cahier de charge citées précédemment en utilisant la méthode de la régulation idéale pour un régulateur PI avec une constante de temps $T=0.001$

D'après les équations obtenues dans la détermination des gains du régulateur PI pour la méthode de la régulation idéale :

On a :

$$Km = \frac{K_T}{R_a f_r + K_T \cdot K_e} \quad (III. 11)$$

$$Tm = \frac{R_a J}{R_a f_r + K_T \cdot K_e} \quad (III. 12)$$

$Kp = \frac{Tm}{Km \cdot T} \quad (III. 19)$	$Ki = \frac{1}{Km \cdot T} \quad (III. 20)$
--	---

Application numérique avec les paramètres choisis :

$$Km = 1.1086 \quad , Tm = 0.0138$$

$$Kp = 12.4711 \quad , Ki = 902.0444$$

a. Fonctionnement à vide

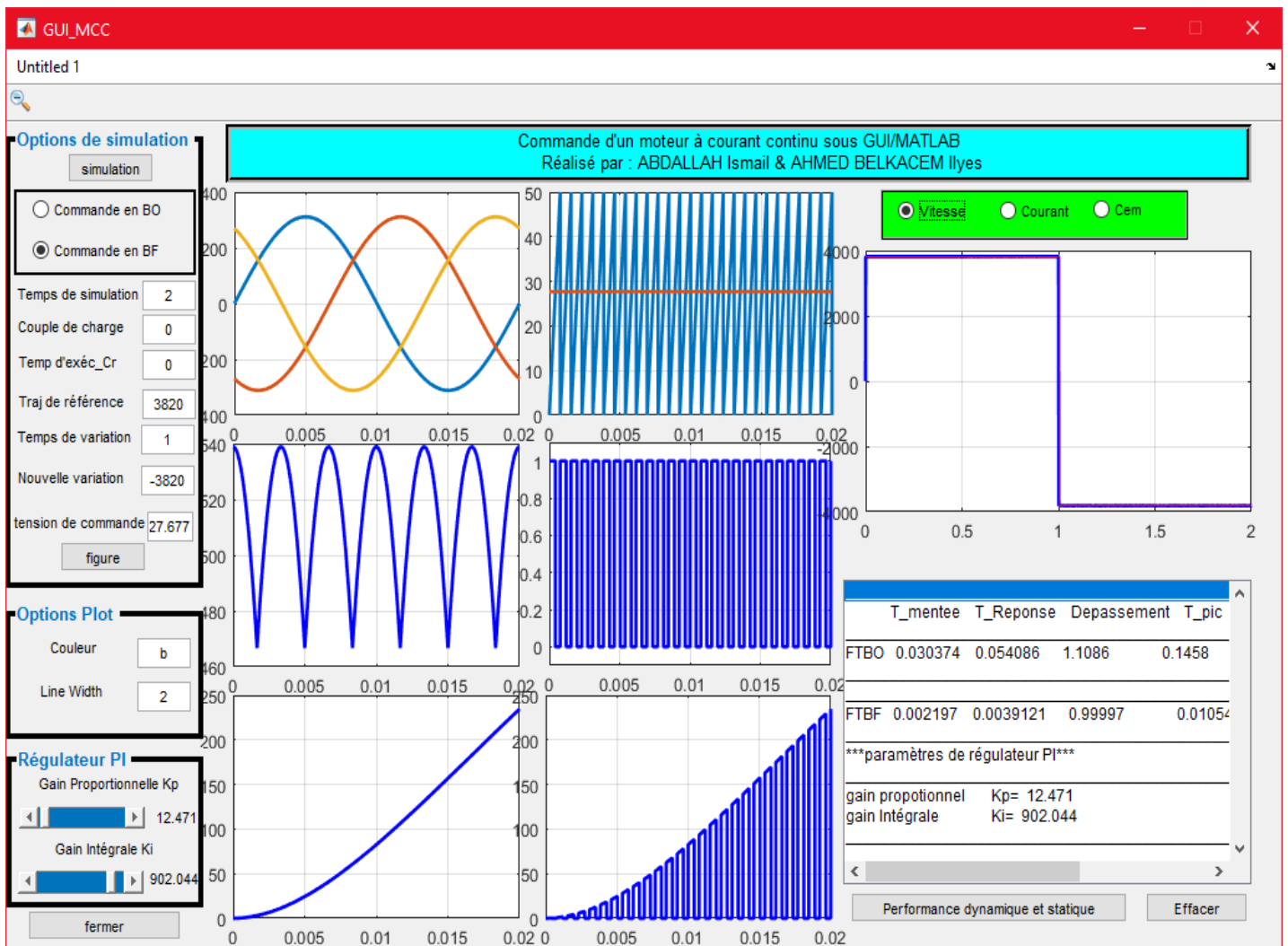


Figure III.36 : Simulation de l'interface graphique réalisée de la commande en boucle fermée à vide.

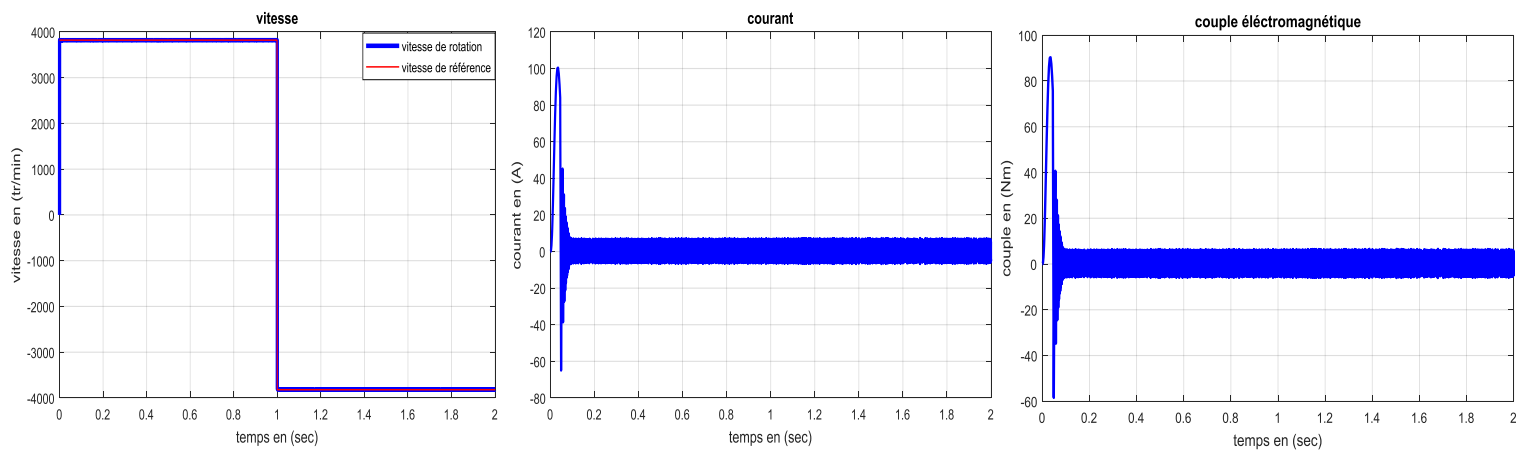


Figure III.37 : Résultats de simulation de l'interface graphique réalisée de la commande en boucle fermée à vide.

b. Fonctionnement en charge

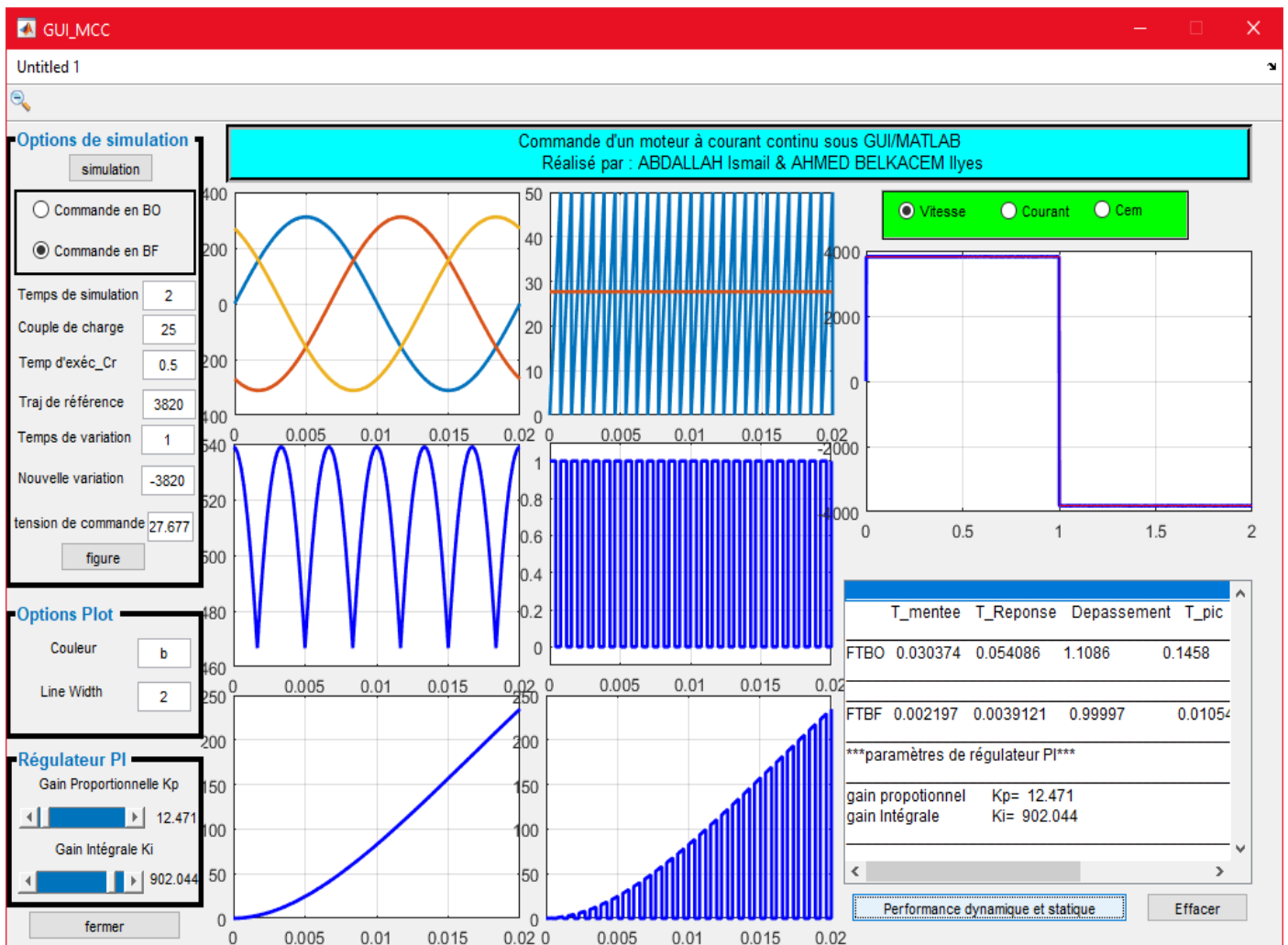


Figure III.38 : Simulation de l'interface graphique réalisée de la commande en boucle fermée en charge.

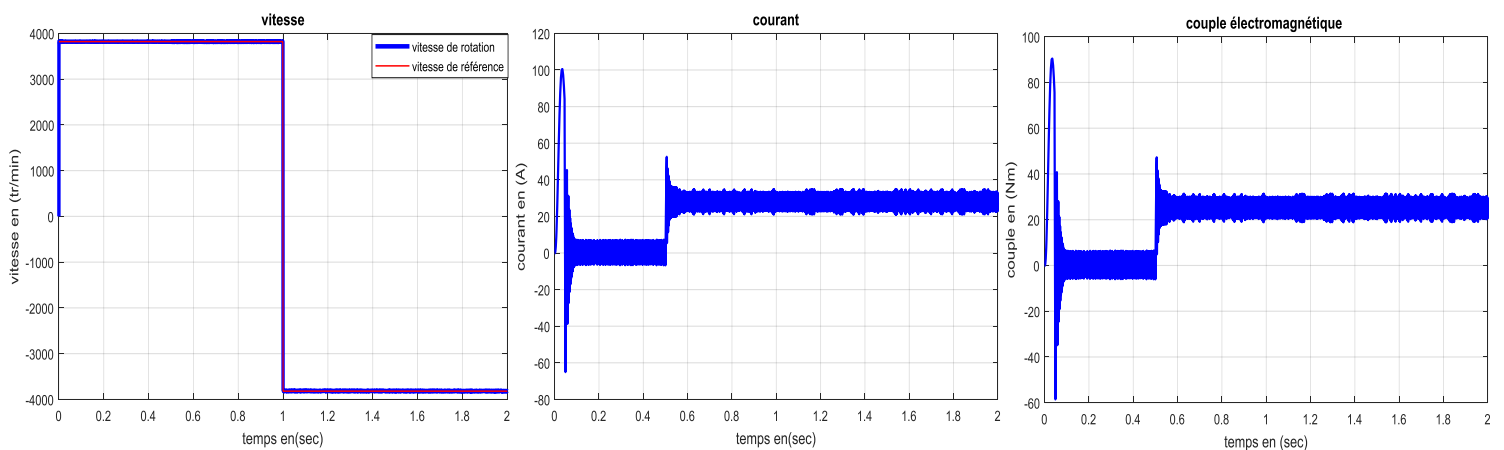


Figure III.39 : Résultats de simulation de l'interface graphique réalisée de la commande en boucle fermée en charge.

III.7.2.3 commande de variation de sens de vitesse d'un moteur à courant continu

Pour faire varier le sens de vitesse d'un moteur à courant continu nous utilisons le pont en H à contrôler la polarité aux bornes d'un récepteur. Il est composé de quatre éléments de commutation généralement disposés schématiquement en une forme de H d'où le nom.

Nous introduisons la variation de sens de vitesse dans l'interface graphique réalisé qui permet de commander deux paramètres :

-temps de variation : c'est le temps quand on a varié le sens de vitesse.

-la nouvelle variation de vitesse : c'est la nouvelle vitesse variée dans un temps de variation.

La figure suivante doit montrer la variation de sens de vitesse avec les deux paramètres suivants :

-Temps de variation : 1 sec

-Nouvelle variation : -3820 tr/min

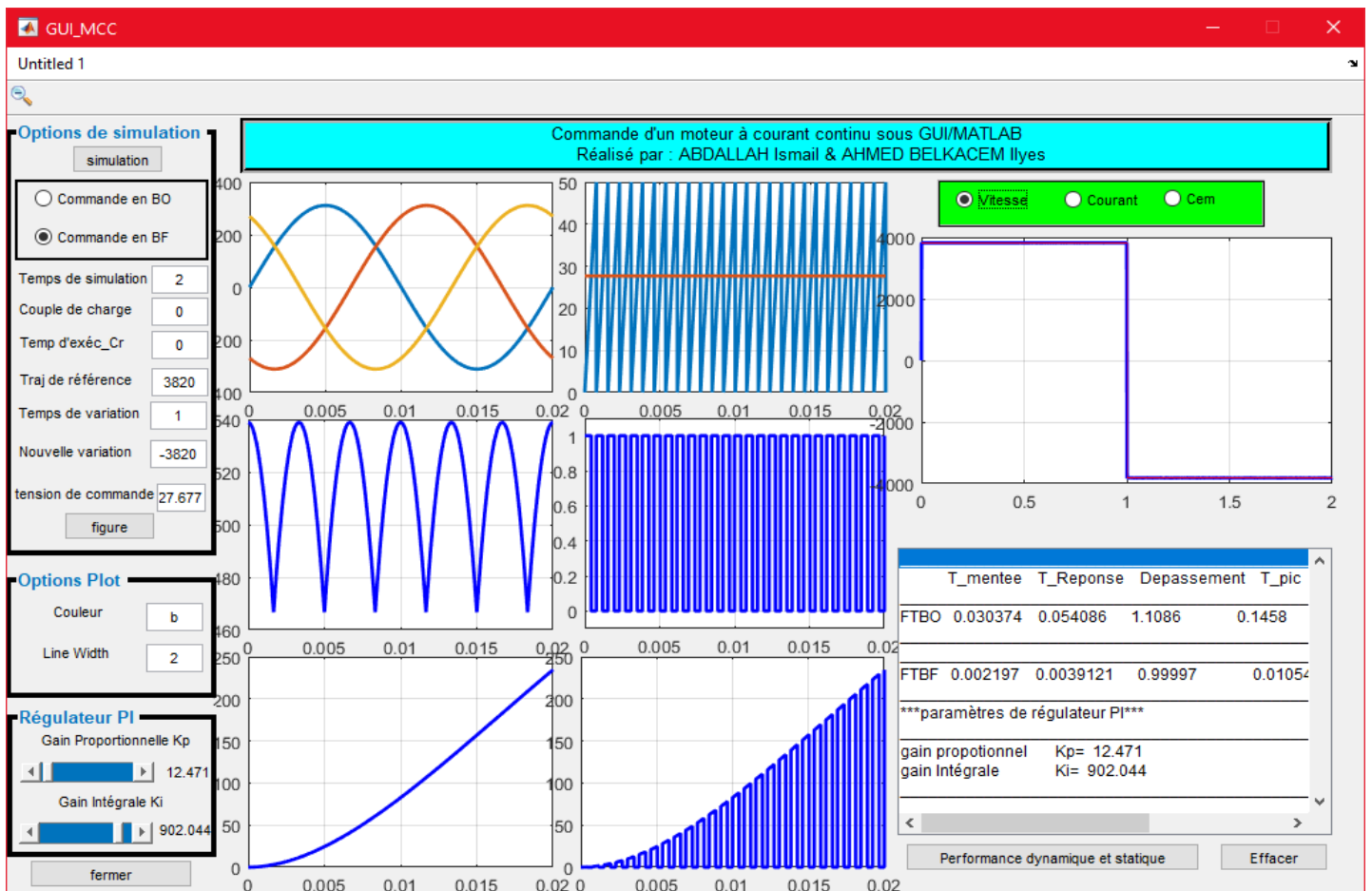


Figure III.40 : Simulation de l'interface graphique réalisée de la variation de sens de vitesse.

III.8 Interprétations des résultats de simulation

III.8.1 Interprétations des résultats de simulation en boucle ouverte

a. Fonctionnement à vide

En faisant varier α on fait varier la durée de fermeture de l'interrupteur. Cette durée augmente avec l'augmentation du rapport cyclique α , pour $\alpha=0.5$ et 0.7 la tension moyenne U_{ch} varie entre 0 à V_e et cette dernière est toujours inférieure à la tension d'entrée ($U_{ch} = \alpha V_e$).

La vitesse commence de zéro pour se stabiliser à 2000 tr/min pour $\alpha=0.5$ et 2800 tr/min pour $\alpha=0.7$, qui est la vitesse du régime établie, la courbe a une forme exponentielle $\omega(t) = Km \left(1 - e^{-\frac{t}{T_m}}\right)$, ce qui correspond à la présence des phénomènes transitoires des courants dans les enroulements de la machine.

Le moteur démarre avec un courant de démarrage important de l'ordre 55 A pour $\alpha=0.5$ et 74 A pour $\alpha=0.7$, puis passe du régime transitoire ou régime établi pour se stabiliser autour de 0.08 A (pas de charge mécanique).

Le moteur démarre avec un couple de démarrage important de l'ordre 49 N.m pour $\alpha=0.5$ et 67 N.m pour $\alpha=0.7$ puis passe du régime transitoire ou régime établi pour se stabiliser autour de 0.09 N.m.

Des fluctuations sont présentées au niveau de courant dues à la technique MLI à cause de la commutation des interrupteurs de l'onduleur.

Remarque: pour éliminer les ondulations (fluctuations) de courant on peut utiliser deux méthodes, soit ajouter une inductance de lissage ou une boucle de régulation du courant.

b. Fonctionnement en charge

- Au régime établi à l'instant $t=0.15$ sec nous appliquons un couple de charge sur le moteur de 25 N.m, la vitesse va décroître pour se stabiliser à 1500 tr/min pour $\alpha=0.5$ et 2250 tr/min pour $\alpha=0.7$.

- Au régime établi à l'instant $t=0.15$ sec à cause du couple de charge appliqué au moteur, le courant augmente pour se stabiliser à la valeur moyenne 28A pour $\alpha=0.5$ et 29A pour $\alpha=0.7$.

- À cause du couple de charge appliqué au moteur le couple augmente pour se stabiliser à la valeur moyenne 26 N.m pour $\alpha=0.5$ et 27 N.m pour $\alpha=0.7$, il a toujours la même forme que le courant.

III.8.2 Interprétations des résultats de simulation en boucle fermée

a. Fonctionnement à vide

- La vitesse commence de zéro et augmente directement jusqu'à 3820 tr/min se stabilise, puis elle est suivie la trajectoire de référence même à la variation de sens de vitesse à $t=1\text{sec}$.
- Le moteur démarre avec un courant de démarrage important de l'ordre 100 A puis passe en régime établi à $t=0.1\text{ sec}$.
- Le moteur démarre avec un couple de démarrage important de l'ordre 90N.m, ce dernier est proportionnel au courant par ce que $C=K_c.I$, alors la courbe du couple prends la même forme du courant.

b. Fonctionnement en charge

- Au régime établi à l'instant $t=0.5\text{sec}$ nous appliquons un couple de charge sur le moteur de 25 N.m, La vitesse commence de zéro et augmente directement jusqu'à 3820 tr/min se stabilise, puis elle est suivie la trajectoire de référence même à la variation de sens de vitesse à $t=1\text{sec}$, donc le couple ne influe pas sur la vitesse car cette dernière suivie la trajectoire de référence.
- Au régime établi à l'instant $t=0.5\text{sec}$ a cause du couple de charge appliqué au moteur, le courant augmente pour se stabilise à la valeur moyenne 28A.
- A cause du couple de charge appliqué au moteur le couple augmente pour se stabiliser à la valeur moyenne 26 N.m, il a toujours la même forme que le courant.

➤ Paramètres choisis appliquées sur le moteur à courant continu :

- Paramètres du moteur: $R_a=1.8402$; $L_a=0.0077\text{ H}$; $J=0.0061$; $\alpha=0.001$; $K=0.9$
- Paramètre de la MLI: Fréquence de la porteuse : 1000 Hz.
- Charge constante : 25N.m (application en régime permanent) par exemple un ascenseur ou grue.
- Charge variable : (avec $K=0.0002$) ; (ex. pompe ou ventilateur).
- Echelon de commande de vitesse : 3820 tr/min.

III.9 Conclusion

Dans ce chapitre on a fait la modélisation et la simulation d'un moteur à courant continu sous MATLAB SIMULINK et visualiser les résultats de simulation sous GUI MATLAB qui nous permet de commander la vitesse et inverser le sens de vitesse du moteur à courant continu.

D'après les résultats de simulation, on constate que la commande en boucle fermée par un régulateur PI est très robuste et plus stable par rapport à celle de la commande en boucle ouvert.

L'action proportionnelle est la plus simple qui soit. Il s'agit d'appliquer une correction proportionnelle à l'erreur corrigeant de manière instantanée, ainsi l'action intégrale complète l'action proportionnelle puisqu'il permet de compenser l'erreur statique et d'augmenter la précision en régime permanent.

Conclusion générale

Conclusion générale

Le travail effectué de ce mémoire rentre dans le cadre de la modélisation et la simulation d'une commande d'un moteur à courant continu à excitation séparée, puis les techniques de la régulation de vitesse et sens de vitesse sous Matlab-Simulink avec la méthode de régulation idéale, et les résultats de simulation visualisent sous GUI MATLAB.

En effet la variation de la vitesse d'un moteur à courant continu se fait d'une manière simple et souple en variant la tension d'induit qui varie le rapport cyclique α , puis la variation de sens se fait en inversant la polarité aux bornes de l'induit ou de l'inducteur.

Durant la simulation sans régulateur, les réponses relevés ont été très divergeant, on a constaté que le système est instable et ne suivre pas la trajectoire de référence au préalable à des valeurs admissibles, en utilisant un régulateur de type proportionnelle intégrale car c'est le mieux approprié contre les perturbations du système, et donne bonne précision. Les paramètres du régulateur ont été choisis par la méthode de la régulation idéale.

Ensuite on a passé à la simulation avec régulateur PI qui a donné des résultats acceptables sur les réponses de vitesse, couple et courant, le système devient stable et suivre la trajectoire de référence.

La synthèse du régulateur PI a été faite par la méthode de la régulation idéale, les performances dynamique et statique du système se changent en variant les paramètres de régulateur PI (K_p et K_i), la simulation du système global montre des performances du système dimensionné.

Espérant que ce travail trouve une suite avec l'utilisation d'autres types de régulateur et d'autres techniques de commande.

Références et bibliographie

Références et Bibliographie

- [1] Bennoura.A et Djabri.K<<Etude et simulation d'un variateur de vitesse d'une Machine à courant continu>> Mémoire de master Centre Universitaire Belhadj Bouchaib d'Ain Temouchent, 2018.
- [2] T.Wildi et G. Sybile, << Electrotechnique >>, Edition de Boeck, 2005. K. Farid <<Identification et contrôle de la machine à courant continu par la Dspace 1103>>.
- [3] TIR Zoheir << Modélisation et Simulation de la Machine à Courant Continu >>, Mémoire Magister en électrotechnique, 2008.
- [4] M. Kostenko et L. Piotrovski, << Machines à courant continu et transformateurs>>, Edition Mir Moscou, 1979. Mémoire de Fin d'Etudes de MASTER ACADEMIQUE UNIVERSITE MOULOUD MAMMERI DE TIZI-OUZOU, septembre 2015.
- [5] Craste.A et Khatchadourian.P ; <<le moteur à courant continu>>,2017.
- [6] Boulakroune.S et CHINE.A <<Commande à vitesse variable d'un moteur à Courant continu alimenté par hacheur Bidirectionnel en courant>> Mémoire de master UNIVERSITE BADJI MOKHTAR DE ANNABA, Juin2018.
- [7] A.AYAD, << Etude et réalisation d'une commande MLI pilotant un hacheur série>>, mémoire de f d'étude de Master, Université Abou Bekr Belkaid, Tlemcen, 2014.
- [8] Guyin chateigner, Michel Boés, Daniel Bouix, Jacques Vaillant, Daniel Verkindère «MANUEL DE GENIE ELECTRIQUE », DUNOD, Paris, 2006.
- [9] V Chassilian; cours<< distribuer l'énergie électrique vers un mcc : changement de sens & variation de vitesse>>, ST JO AVIGNON France.
- [10] Mr ACHOUR AbdelYazid ; cours<< Techniques de Commande avancée>>, Université A.MIRA-BEJAIA , 2016.

- [11] MANSERI Sonia <<Commande adaptative indirecte d'un moteur à courant continu>>, mémoire de fin d'étude de Master, Université A.MIRA Bejaïa, 2012.
- [12] V. Alexéev, V. Tikhomirov et S. Fomine, <<Commande optimale>>, Mir, 1982.
- [13] DEBOUZE Aymen et KELAIAIA Achref, << Commande floue des machines électriques>> Mémoire de fin d'étude de Master, Université 8 Mai 1945 – Guelma, Juillet 2019.
- [14] Ayeb Fakhreddine ,<< Conception et implémentation d'une commande PID numérique d'un moteur à courant continu>> , Mémoire de fin d'étude de Master , UNIVERSITE LARBI BEN M'HIDI D'OUM EL BOUAGUI, juin 2014.
- [15] ABDERREZEK Hadjer << Contribution à la commande robuste des systèmes non linéaires>>, Mémoire de magistère, UNIVERSITÉ FERHAT ABBAS SETIF-1, octobre 2014
- [16]<https://openclassrooms.com/fr/courses/148061-la-programmation-avec-autoit/147131-votre-premiere-interface-graphique-gui>.
- [17] <https://www.lebigdata.fr/interface-homme-machine-tout-savoir-sur-les-ihm>
- [18] Mohamed Lokbani; Cours << Interfaces Graphiques de Base>> Université de Montréal 2008.
- [19] JEROME Briot, <<A guide to MATLAB for Beginners Epecienced Users>>, Handback, Cambridge, 2000.
- [20]<https://matlabpourtous.com/matlab/les-interfaces-graphiques-sous-matlab/utilisation-de-guide>.
- [21]<https://matlabpourtous.com/matlab/les-interfaces-graphiques-sous-matlab/les-objets-graphiques-sous-matlab>.
- [22] Dr GHOURAF Djamel Eddine, TP <<commande des machines électriques >>, Ecole Nationale Polytechnique, Oran.
- [23] Dr. Aggoune Lakhdar, Cours << Systèmes Asservis>>, Université Ferhat Abbas Sétif 1, 2018.
- [24] Dr GHOURAF Djamel Eddine, Cours <<Correction des systèmes asservis linéaires-méthodes de synthèse>>, Ecole Nationale Polytechnique, Maurice Audin, Oran, 2020.

Résumé

L'évolution des technologies conduit à utiliser des machines nécessitant des vitesses de rotation précises et variables. L'avantage principal des moteurs à courant continu réside dans leur adaptation simple aux moyens permettant de régler ou de faire varier leur vitesse, leur couple et leur sens de rotation ainsi que leur raccordement direct à une source d'énergie. Dans ce travail, on a présenté les généralités sur les moteurs à courant continu, puis la technique de la variation de vitesse et le sens de rotation, Ainsi le développement et programmation des interfaces graphiques et enfin la simulation de notre model sur logiciel de simulation MATLAB, les résultats obtenus sont visualises sur l'interface graphique réalise sous GUI MATLAB

Mots clés : Moteur à courant continu, interface graphique, MATLAB

abstract

The evolution of technologies leads to use machines requiring precise and variable speeds. The main advantage of direct current machines lies in their simple adaptation to adjust or vary their speed, torque and direction of rotation as well as their direct connection to a power source.

In this work, we presented the generalities of direct current motors, then the technique of speed variation and the direction of rotation, Thus the development and programming of graphic interfaces and finally the simulation of our model on MATLAB simulation software. , the results obtained are visualized on the graphical interface produced under GUI MATLAB

Keywords : DC motor, graphic interface, MATLAB

ملخص

أدت تطور التكنولوجيا إلى استخدام الآلات التي تتطلب سرعات دقيقة ومتغيرة على سبيل المثال قطار فائق السرعة .

الميزة الرئيسية لمحرك التيار المستمر تكمن في تكيفها السهل في سبل تعديل أو تغيير في سرعته وعزمه و اتجاه الدوران بالإضافة إلى اتصاله المباشر لمصدر الطاقة.

في هذا العمل، قدمنا عموميات لمحركات التيار المستمر، ثم أساليب تغيير السرعة و اتجاه الدوران، و بالتالي تطوير و برمجة واجهات الرسوم وأخيرا محاكاة نموذجنا على برامج ماتلاب، يتم عرض النتائج التي تم الحصول عليها على الواجهة الرسومية المنتجة على غي ماتلاب

الكلمات المفتاحية: محركات التيار المستمر، الواجهة الرسومية، ماتلاب