

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 des Sciences et de la 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 : COMMADE ELECTRIQUE
Thème

*Commande de la valeur numérique de la vitesse d'un moteur
à courant continu sous interface graphique
(GUI-MATLAB)*

Présenté Par :

- 1) Bouhadjela Wafaa
- 2) Benallal Wafaa

Devant les jurys composés de :

BENAZZA BAGHDADI	MAA	C.U.B.B (Ain Temouchent)	Président
TOUHAMI AICHA	MAA	C.U.B.B (Ain Temouchent)	Encadrant
GHOURAF DJAMEL EDDINE	MCB	E.N.P.O (Oran)	Co- Encadrant
AYACHE ZOUAOU	MCA	C.U.B.B (Ain Temouchent)	Examineur

Année universitaire 2019/2020

Remerciements

Nous remercions tout d'abord, Allah qui nous a donné la force et le courage de parvenir à élaborer ce modeste travail.

Nous tenons aussi à remercier nos encadreurs : Mme. TOUHAMI AICHA et Monsieur GHOURAF DJAMEL EDDIN pour ses conseils précieux et pour toutes les commodités et aisances qu'il nous a apportées durant notre étude et réalisation de ce projet.

Nous adressons nos remerciements aux membres du jury : Monsieur BENAZZA BAGHDADI et Monsieur AYACHE ZOUAOUI; pour avoir accepté de juger notre travail ce qui pour nous est un grand honneur.

Nous profitons de cette opportunité pour exprimer notre gratitude à tous nos enseignants qui ont contribué par leurs collaborations, disponibilité et sympathie à notre formation.

Enfin, nous tenons à remercier toute personne qui nous a Aidés de près ou de loin durant notre travail et en particulier tous nos amis de la promotion GE



Wafaa & Wafaa 

Dédicaces

A mes parents,

*Aucun hommage ne pourrait être à la hauteur de l'amour et
de l'affection dont ils ne cessent de me combler,*

*Qu'ils trouvent dans ce travail un témoignage de mon profond
amour et éternelle reconnaissance,*

Que dieu leur procure bonne santé et longue vie,

A mes chers frères

A mes chères sœurs

A mes chères cousines et cousins

A toute la famille petite et grande ainsi qu'à tous mes amis

A tous ce qu'il mon aidée de près ou de loin

A mon binôme que j'apprécie : Benallal Wafaa

Je dédie ce travail...



Wafaa.Bou

Dédicaces

A mes parents,

*Aucun hommage ne pourrait être à la hauteur de l'amour et
de l'affection dont ils ne cessent de me combler,*

*Qu'ils trouvent dans ce travail un témoignage de mon profond
amour et éternelle reconnaissance,*

Que dieu leur procure bonne santé et longue vie,

A mes chères sœurs

A mes chères cousines et cousins

A toute la famille petite et grande ainsi qu'à tous mes amis

A tous ce qu'il mon aidée de près ou de loin

A mon binôme que j'apprécie : Bouhadjela Wafaa

Je dédie ce travail...



Wafaa.Ben

Résumé

Les interfaces graphiques constituent aujourd'hui un véritable challenge dans le domaine industriel, car elles sont particulièrement un sujet d'actualité. Une interface graphique est souvent appelée GUI (Graphical User Interface), elle est formée d'une ou plusieurs fenêtres qui contiennent divers composants graphiques tel que boutons, menu, champs de texte, ect....et à l'aide de ces composants graphiques l'utilisateur peut interagir a tout moment. L'objectif de ce mémoire est de développer une interface graphique sous GUI-Matlab qui consiste à faire la régulation d'un moteur à courant continu à excitation séparée en utilisant un régulateur de type PI (Proportionnelle Intégral). On a effectué le contrôle du système à partir de régulateur PI et visualiser les résultats de la régulation avec simulation de notre système après on a calculé les paramètres dynamiques et statiques du système et analyser le système étudié(MCC) avant et après la commande.

Mots clés : interfaces graphiques,GUI,boutons, menu,champs de texte,GHI-Matlab, PI, MCC.

Abstract

Graphical interfaces today constitute a real challenge in the industrial domain, because they are particularly a topical subject. A graphical interface is often called GUI (Graphical User Interface), it is made up of one or more windows which contain different graphical components such as buttons, menu, text fields, etc. and the user using these graphical components and can interact at any time. The objective of this thesis is to develop a graphical interface on the GUI-Matlab which consists in regulating a DC motor with separate excitation using a PI (Proportional Integral) type regulator. We carried out the control of the system from PI regulator and visualized the results of the regulation with simulation of our system after we calculated the dynamic and static parameters of the system and analyzed the studied system (DC motor) before and after the command.

Keywords : Graphical interfaces, GUI, buttons, menu, text fields, GHI-Matlab, PI, DC Motor.

ملخص

تشكل الواجهات الرسومية اليوم تحديًا حقيقيًا في المجال الصناعي ، لأنها موضوع مستجد بشكل خاص. غالبًا ما تسمى الواجهة الرسومية GUI (واجهة المستخدم الرسومية) ، وتتكون من نافذة واحدة أو أكثر تحتوي على مكونات رسومية متنوعة مثل الأزرار والقائمة وحقول النص وما إلى ذلك وباستخدام هذه المكونات الرسومية يمكن للمستخدم التفاعل في أي وقت. الهدف من هذه الرسالة هو تطوير واجهة رسومية على (GUI-MATLAB) والتي تتكون من تنظيم محرك تيار مستمر DC بتفعيل منفصل باستخدام منظم نوع PI (تكامل متناسب). فقمنا بالتحكم في النظام من منظم PI ومثلنا نتائج التنظيم مع محاكاة نظامنا بعد أن قمنا بحساب العوامل لاديناميكية والثابتة للنظام ثم تحليل النظام المدروس(MCC) قبل وبعد الأمر.

الكلمات المفتاحية : الواجهات الرسومية, GUI , الأزرار, القائمة ,حقول النص,PI, DC motor, GHI-Matlab.

Table des Matières

Titre.....	I
Résumé.....	V
Table des matières.....	IIIV
Liste des figures	XI
Liste des symboles	IX
Acronyme.....	IIIX
Introduction générale.....	1

Chapitre I : état de L'art

I.1.Introduction.....	2
I.2.Description du système étudié.....	2
I.3. Définition de la machine à courant continue.....	3
I.4.Constitution d'une machine à courant.....	3
I.4.1.Inducteur.....	4
I.4.2.Induit.....	4
I.4.3.Collecteur et balais.....	4
I.5.Principe de fonctionnement	5
I.6.Les type des moteurs à courant continue.....	5
I.6.1.Excitation séparée.....	6
I.6.2.Excitation série.....	6
I.6.3.Excitation shunt.....	6
I.6.4.Excitation composée.....	7
I.7.Le bilan de puissance.....	7
I.8.Le rendement.....	8
I.9.Domaine d'utilisation.....	8
I.10.Avantages et inconvénient.....	9
I.10.1.Avantages.....	9
I.10.2.Inconvénient.....	9
I.11.Les méthodes de variation de la vitesse d'un moteur à courant continu.....	9
I.12.La commande de vitesse d'un moteur à courant continue.....	10

I.12.1.Par les techniques classique.....	10
I.12.1.Par les techniques avancées.....	10
I.13.Utilité des interfaces graphiques dans l'industrie.....	13
I.13.1.interfaces homme machine HMI	13
I.14. Conclusion	13

Chapitre II : Interface graphiques sous GUI-MATLAB

II.1. Introduction.....	16
II .2. Interface graphiques sous MATLAB	16
II.3.Présentation	17
II.4.Les étapes pour faire un programme simple.....	17
II.4.1.Démarrer GUIDE	18
II.4.2.Ajouter des composants.....	19
II.4.3 Définir les propriétés de chaque composant.....	20
II.4.4. Calback du bottons.....	20
II.4.5-Exécution de l'interface réalisée.....	23
II.5.Réalisation d'une calculatrice scientifiques et graphique.....	27
II.6.Conclusion.....	28

Chapitre III : Modélisation et simulation sous interface graphique

III.1.Introduction	29
III.2.Modélisation de source triphasé	29
III.3. Modélisation du redresseur.....	30
III.4. Modélisation du filtre	31
III.5.Convertisseur statique (hacheur).....	32
III.5.1.Définition d'un hacheur.....	32
III.5.2.Principe d'un hacheur série (abaisseur de tension).....	32
III.6.Générateur d'impulsion.....	33
III.7. Modélisation de MCC	33
III.8.MCC en boucles ouverte	35
III.9.Synthèse de régulateur PI	37

III.9.1.Principe générale de la régulation.....	37
III.9.2.Performances de système à réglés.....	38
III.10.La méthode de régulation idéale.....	39
III.10.1.Spécification de cahier charge	39
III.11. Modélisation sous interfaces graphique.....	47
III.12.Quelque exemples de fonctionnement de notre interface graphique.....	49
III.13.conclusion.....	52
Conclusion générale	53
Bibliographie	54

Liste des figures

Chapitre I

Figure I-1 : schéma du système étudié.....	2
Figure I-2:schéma synoptique de la machine à courant continu.....	3
Figure I-3:description de la machine à courant continu	3
Figure I-4 :l'inducteur de la machine à courant continu.....	4
Figure I-5:l'induit de la machine à courant continu.....	4
Figure I-6:le collecteur et les balais.....	5
Figure I-7:modélisation électrique d'un moteur à excitation séparée.....	6
Figure I-8:modélisation électrique d'un moteur à excitation série.....	6
Figure I-9:modélisation électrique d'un moteur à excitation parallèle.....	7
Figure I-10:modélisation électrique d'un moteur à excitation composée.....	7
Figure I-11:le bilan de puissance.....	8
Figure I-12:Schéma global de l'interface homme-machine	14

Chapitre II

Figure II -1:Présentation de l'interface de GUIDE.....	17
Figure II -2:Démarrage GUIDE.....	18
Figure II -3:fenêtre de l'interface.....	18
Figure II -4:Placement des éléments de l'interface.....	19
Figure II -5:Propriété du bouton.....	20
Figure II -6:activer l'interface.....	23
Figure II -7:Résultat du test.....	24
Figure II -8:tracer $\cos(t)$, $\sin(t)$ et $\exp(t)$	25
Figure II -9:Fonction 2D.....	26
Figure II -10:Fonction 3D.....	26
Figure II-11: Calcul numérique	27
Figure II-12: Traçage du graphe	27

Figure II-13: Calcul et traçage.....28

Chapitre III

Figure III -1:schéma bloc de la source triphasée sous MATLAB.....29

Figure III -2:Représentation schématique d'un redresseur PD3.....30

Figure III -3:schéma bloc de redresseur sous MATLAB.....30

Figure III -4:filtre passe-bas.....31

Figure III -5:schéma bloc du filtre passe-bas de Butter Worth sous MATLAB.....31

Figure III -6:Modélisation d'un hacheur série.....32

Figure III -7:Générateur d'impulsion sous MATLAB.....33

Figure III -8:Génération du signal de commande par la technique MLI.....33

Figure III -9:Schéma technologique de la machine à courant continu.....34

Figure III -10:schéma bloc du MCC sous MATLAB.....35

Figure III-11: schéma bloc du MCC en boucle ouverte.....35

Figure III-12: Schéma bloc de la commande de la MCC en boucle fermée39

Figure III-13: les schémas de source, redressement et filtrage.....42

Figure III-14: vitesse, courant, couple BO son charge avec $\alpha=0.3$43

Figure III-15: vitesse, courant, couple BO avec charge avec $\alpha=0.3$44

Figure III-16: vitesse, courant, couple BF son charge avec $\alpha=0.6$45

Figure III-17: vitesse, courant, couple BF avec charge avec $\alpha=0.6$46

Figure III-18: interface graphique sous GHI-MATLAB.....47

Figure III-19:interface graphique du premier exemple.....49

Figure III-20: interface graphique du deuxième exemple.....50

Figure III-21: interface graphique du troisième exemple.....51

Figure III-22: interface graphique du quatrième exemple.....52

Liste des symboles

Ω : la vitesse du rotor.

C_e : Le couple électromagnétique fourni par le moteur.

E : La force contre électromotrice générée par le moteur.

U : La tension d'induit.

R_a : La résistance d'induit.

L_a : L'inductance d'induit.

J : Le moment d'inertie du moteur.

f_r : Coefficient de frottement visqueux de l'ensemble moteur et charge.

K_t : Constante de couple électromagnétique.

K_e : Constante de Force contre électromotrice.

C_r : Couple résistant.

α : rapport cyclique.

Φ : le flux magnétique.

P_a : puissance absorbée.

P_u : puissance utile.

η : Rendement.

f : fréquence.

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

K_m : la constante de numérateur de la fonction du transfert du système.

T_m : la constante d'énumérateur de la fonction du transfert du système.

K_i : gain intégrale.

K_p : gain proportionnelle.

T : la constante du temps.

T_i : la constante d'intégration.

$R(p)$: la fonction du transfert du régulateur.

$G(p)$: la fonction du transfert du moteur.

Acronyme

ABCD

EFGH

FTBF : Fonction de Transfert en Boucle Fermée.

FTBO : Fonction de Transfert en Boucle Ouvert.

GUI : Graphical User Interface.

HMI : Humain Machine Interface.

FTBO : Fonction de Transfert en Boucle Ouvert.

LMNO

MCC : Moteur à Courant Continu.

MLI : Modulation de Largeur d'Impulsions.

MOSFET: Metal Oxide Semi-Conductor Field Effect Transistor.

OPQR

PI : Proportionnelle Intégrateur.

STUV

WXYZ

.

Introduction Générale

Depuis quelques années déjà la technologie moderne est basée sur les interfaces graphiques car elles visent également à rendre le plus naturel possible la relation entre l'homme et la machine. L'interface graphique permet en effet un dialogue avec la machine plus direct, plus intuitif et plus accessible.

Dans plusieurs applications électriques il est avantageux d'utiliser le moteur à courant continu car il est un convertisseur d'énergie, il est réversible. Ce moteur utilise une source d'énergie continue ou bien il est alimenté par un convertisseur statique qui transforme le courant alternatif en courant continu.

La supériorité de ces moteurs réside dans le fait qu'ils se prêtent facilement à un contrôle souple, continu et presque instantané de leur vitesse.

Ce projet de fin d'étude est une étude sur la commande de vitesse pour le moteur à courant continu sous interface graphique GUI-MATLAB. On a introduit le régulateur classique de type PI (proportionnelle intégrale) pour assurer un temps de réponse minimale du moteur et améliorer les performances qui sont spécifiées dans le cahier de charge. Puis on a intégré cette régulation dans une interface sous GUI-MATLAB.

Premier chapitre : on présente une étude générale du moteur à courant continu, la description du moteur et son principe de fonctionnement ainsi l'importance des interfaces graphique dans l'industrie.

Deuxième chapitre : on donne des généralités sur l'interface graphique sous GUI-MATLAB, les méthodes de programmation et les étapes pour créer des interfaces graphiques simples et faciles à réaliser.

Troisième chapitre : concerne la modélisation du moteur à courant continu alimenté par un hacheur série basé sur la technique MLI et commandé un régulateur de type PI sous MATLAB-Simulink implémenté sous interface graphique réalisée sous GUI-MATLAB.

Enfin, on conclut avec une conclusion générale résume tous les résultats obtenus dans ce projet.

Chapitre I

Etat De L'Art
Des Moteurs
à Couarnt
Continu

I.1- Introduction :

La machine électrique à courant continu (MCC) est un convertisseur électromécanique permettant la conversion bidirectionnelle d'énergie entre une installation électrique parcourue par un courant continu et un dispositif mécanique. Il est très utilisé en industrie est nécessite une régulation précise de la vitesse de rotation. Ce chapitre traite la modélisation en vue de la simulation d'un moteur à courant continu à excitation séparée. [1]

Nous allons présenter dans ce chapitre les différents types des machines à courant continu et leurs utilisations dans le domaine de l'industrie ainsi que les techniques de contrôle de vitesse de dernier.

I.2-Description du système étudié :

Le système étudié représenté dans la figure suivante est composé d'une source triphasé et un redresseur plus un filtre et un hacheur qui sert à alimenter le moteur à courant continu.

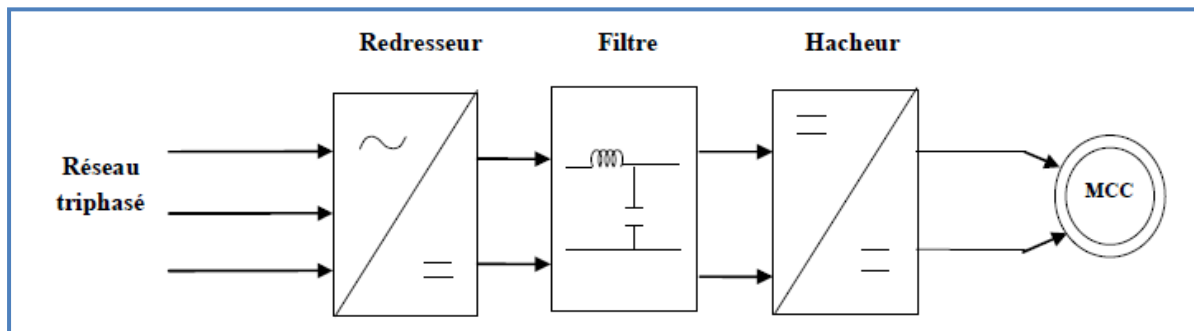


Figure I-1 : schéma du système étudié

- **Source triphasé :** Un système triphasé est un ensemble de trois grandeurs (tension ou courant) sinusoïdales de même fréquence, déphasées les unes par rapport aux autres.
- **Redresseur :** Le redresseur est un convertisseur « alternatif / continu ». Une conversion d'énergie électrique permet de disposer d'une source de courant continu à partir d'une source alternatif.
- **Filtre :** On utilise un filtre passe bas « LC », pour éliminer les harmoniques hautes fréquences.
- **Hacheur :** Un hacheur est un convertisseur statique continu – continu. On convertit une tension continue fixe U (en provenance d'une alimentation) en une tension continue réglable V par un signal de commande U_{com} de rapport cyclique variable.

I.3-Définition de la machine à courant continu :

Les machines à courant sont des convertisseurs électromagnétiques :

- Soit ils convertissent l'énergie électrique absorbée en énergie mécanique pour démarrer et entrainer une charge électrique, on dit alors « fonctionnement moteur ».
- Soit ils convertissent l'énergie mécanique reçue en énergie électrique on dit alors « fonctionnent génératrice ».

La figure suivante présente la conversion énergétique de la machine à courant continu.

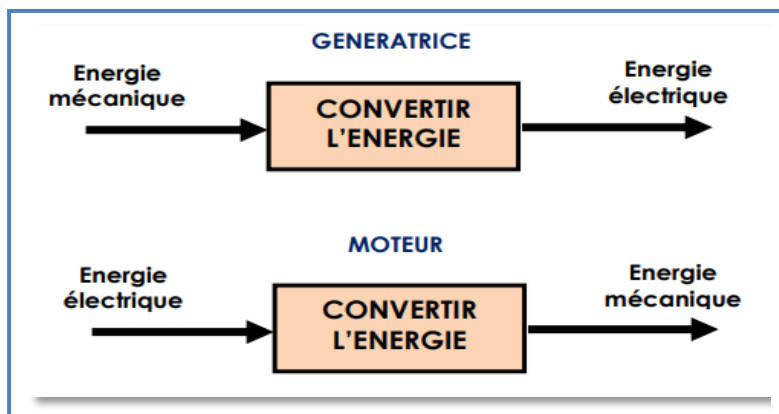


Figure I-2:schéma synoptique de la machine à courant continu

I.4- Constitution d'une machine à courant continu :

La machine à courant continu est constituée de trois parties principales : induit, inducteur séparé par un entrefer, des collecteurs et des balais.

La figure suivante présente la description de la machine à courant continu.

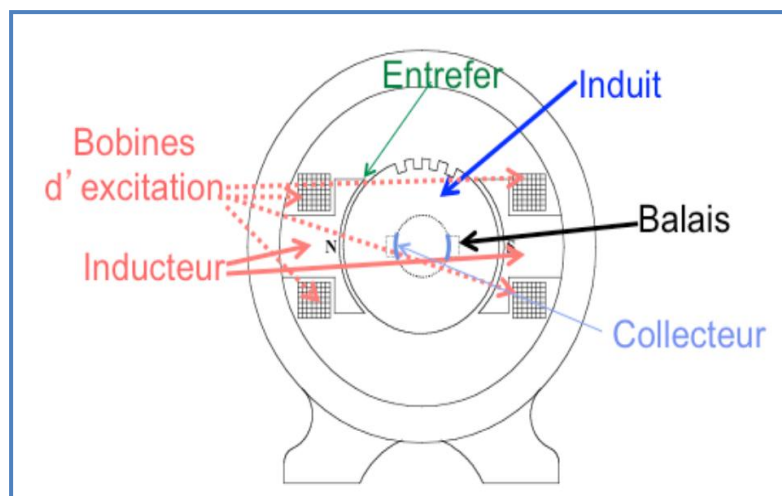


Figure I-3: description de la machine à courant continu

I.4.1- Inducteur:

C'est la partie fixe de la machine, il est constitué par des aimants permanents ou des enroulements bobinés parcourus par un courant continu. Ces derniers permettent de créer un champ magnétique statorique.

La figure suivante présente l'inducteur de la machine à courant continu.

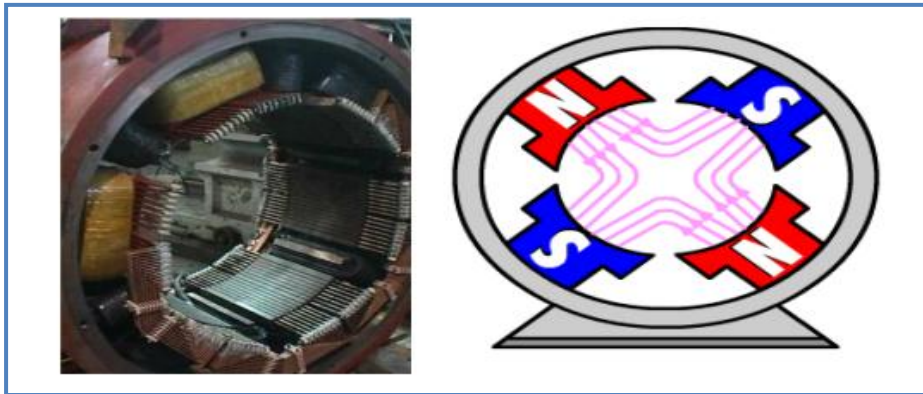


Figure I-4 : l'inducteur de la machine à courant continu

I.4.2- Induit :

C'est la partie mobile du moteur, c'est lui qui tourne. Il est constitué d'encoches dans lesquelles est enroulé un bobinage de (N) conducteur alimenté par un courant continu à travers de dispositif collecteur et balais.

La figure suivante présente l'induit de la machine à courant continu.

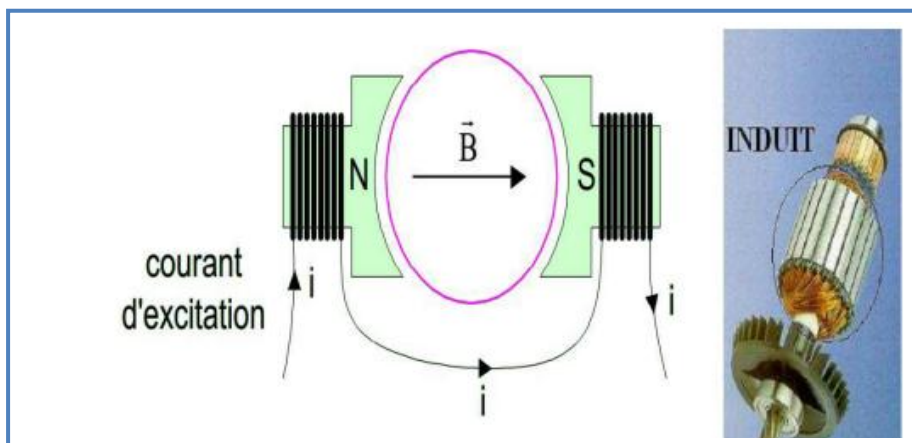


Figure I-5 : l'induit de la machine à courant continu

I. 4.3-Collecteur et Balais :

Le collecteur est un ensemble de lames en cuivre reliées les extrémités du bobinage.

CHAPITRE I-Etat de l'art des moteurs à courant continu (MCC)

Les balais sont appelés aussi les charbons sont situés au stator et frottent sur le collecteur en rotation.

Le collecteur et les balais permettent de faire circuler un courant dans l'induit.

La figure suivante présente le collecteur et les alais de la machine à courat continu.

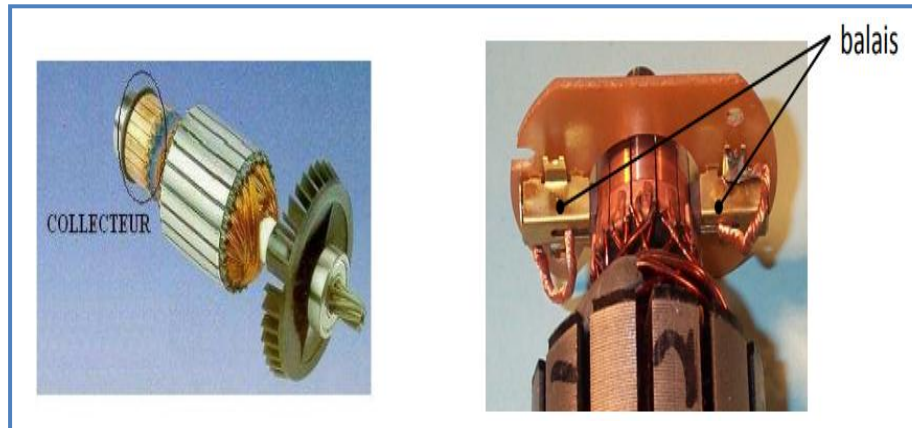


Figure I-6 : le collecteur et les balais

I.5-Principe de fonctionnement :

Le principe de fonctionnement d'un moteur à courant continu repose sur la création d'un couple électromagnétique sur un rotor porteur de conducteurs parcourus par un courant continu et qui se déplacent dans un champ magnétique.

Un inducteur (stator) crée un champ magnétique de direction fixe. Ce champ peut être obtenu par un aimant permanent ou par un électro-aimant.

L'induit (rotor) porte des conducteurs parcourus par un courant continu. Les spires sont le siège de force qui créent un couple entraînant la rotation du rotor. De cette rotation résulte une variation du flux traversant la spire, variation qui engendre une force électromotrice au borne de la spire.

Un dispositif permet l'alimentation des conducteurs, Ce dispositif est appelé collecteur et balais. [2]

I.6-Les types de moteur à courant continu :

Il existe 4 modes d'excitation de moteur à courant continu :

Excitation séparée, excitation série, excitation parallèle (shunt), excitation composée.

I.6.1- Excitation séparée :

L'inducteur et l'induit sont alimentés séparément. Le courant qui alimente l'inducteur est indépendant à celui qui alimente l'induit.

La figure suivante présente le schéma de l'excitation séparée.

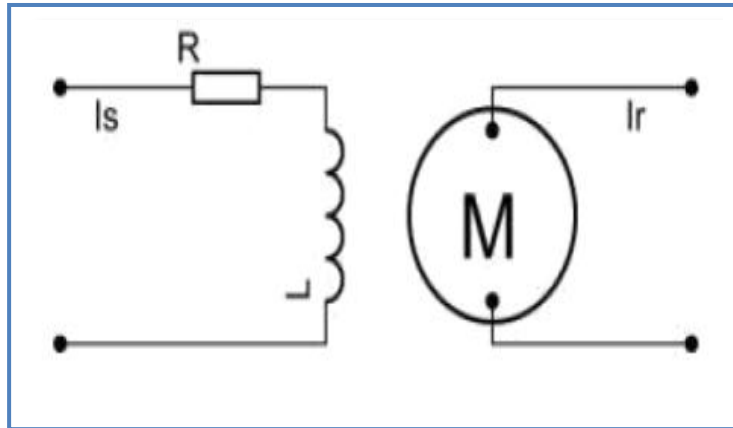


Figure I-7 : modélisation électrique d'un moteur à excitation séparée

I.6.2- Excitation série :

L'inducteur de moteur est en série avec l'induit, le courant d'induit est également le courant d'excitation.

Ce type de moteur présente un très fort couple de démarrage.

La figure suivante présente le schéma de l'excitation série.

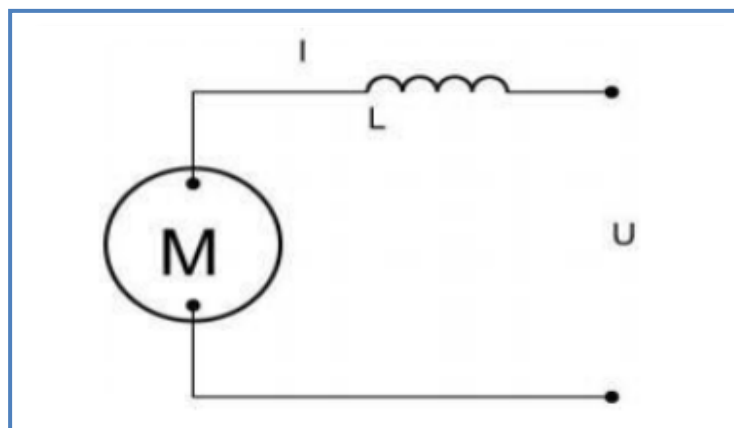


Figure I-8 : modélisation électrique d'un moteur à excitation série

I.6.3- Excitation shunt :

L'induit et l'inducteur sont alimentés par la même source de tension et sont mis en parallèle.

Ce type d'excitation présente un couple de démarrage moyen.

La figure suivante présente le schéma de l'excitation parallèle.

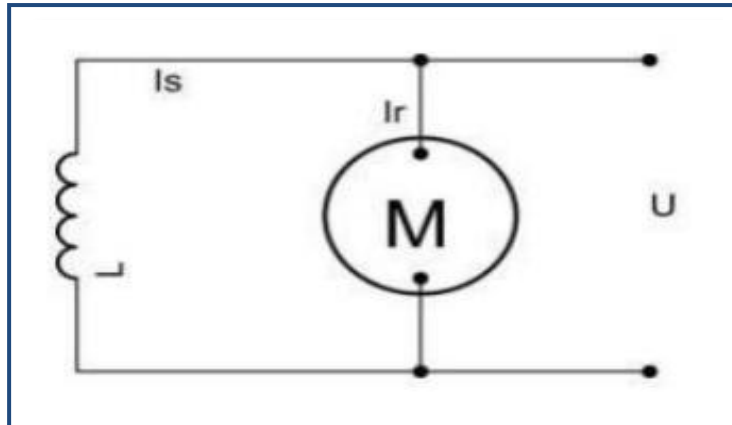


Figure I-9 : modélisation électrique d'un moteur à excitation parallèle

I.6.4-Excitation composée :

Le moteur à excitation composée porte dans son inducteur deux enroulements, le premier à excitation série et le deuxième à excitation shunt.

Ce type de moteur présente un couple de démarrage meilleur qu'en shunt mais plus faible qu'en série.

La figure suivante présente le schéma de l'excitation composée.

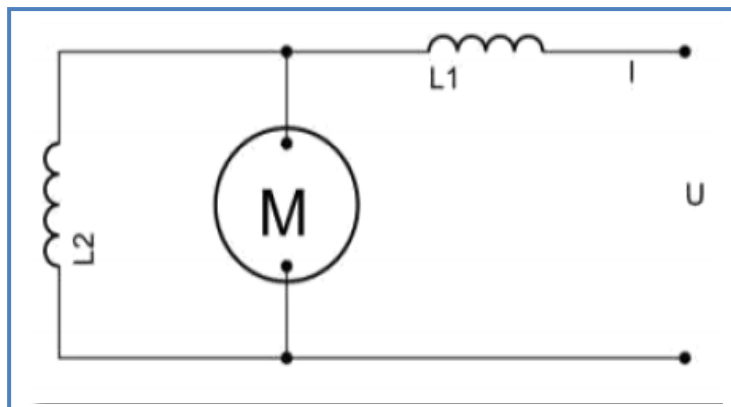


Figure I-10 : modélisation électrique d'un moteur à excitation composée

I.7-Le bilan de puissance :

Le moteur à courant continu absorbe une puissance électrique et fournit une puissance utile de nature mécanique. Entre ces deux termes, l'étude se portera sur toutes les pertes aussi bien mécaniques qu'électriques, et enfin une puissance sera étudiée tout particulièrement, elle

correspond au passage de la puissance électrique à la puissance mécanique.[3]Le bilan peut être résumé à l'aide schéma suivant :

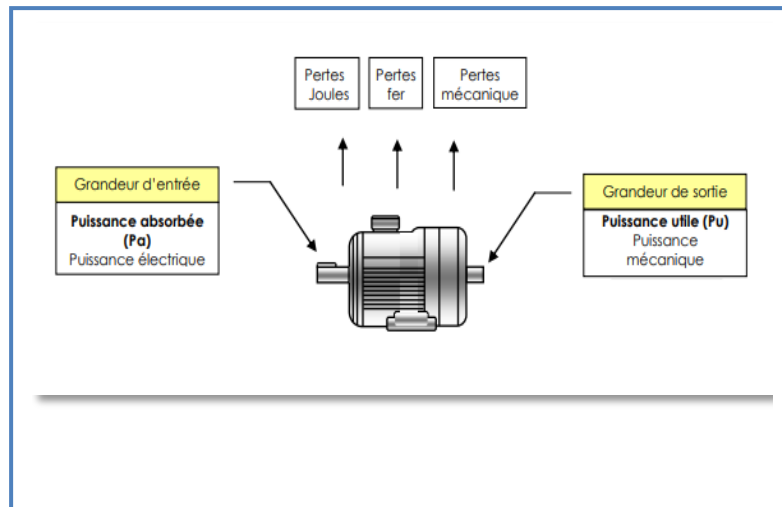


Figure I-11 : le bilan de puissance

I.8-Le rendement :

Le rendement de la machine à courant continu est la relation entre la puissance utile est la puissance absorbée :

$$\mu = \frac{p_u}{p_a} \quad (\text{I.1})$$

p_u : Puissance utile.

p_a : Puissance absorbée.

I.9-Domaine d'utilisation :

Malgré l'utilisation croissante et la grande rivalité avec les moteurs AC, lesquels, en tant que moteurs triphasés, font également partie des moteurs électriques, les moteurs DC jouent aujourd'hui encore un rôle très important. Ils sont souvent utilisés dans les applications industrielles, en raison de leurs propriétés notamment la possibilité qu'ils offrent d'ajuster leur couple et leur vitesse de manière précise et extrêmement variable.

Les moteurs à courant continu à excitation séparée sont utilisés en traction, sur les machines-outils et partout où les systèmes nécessitent des vitesses contrôlées.

CHAPITRE I-Etat de l'art des moteurs à courant continu (MCC)

Associés aux variateurs électronique, ils permettent d'obtenir différentes vitesses d'approche, de réglage ou de retour rapide sur les machines, de mettre en marche ou ralentir progressivement ou de maintenir constante une vitesse malgré les variations de charge.

I.10-Avantages et inconvénients:

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.10.1-Avantages :

- 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.10.2-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. [4]

I.11-Les méthodes de variation de la vitesse d'un moteur à courant continu :

Les variateurs de vitesses sont des systèmes qui convertissent les caractéristiques d'une alimentation en fonction d'une consigne donnée. Ils ont plusieurs fonctions parmi lesquelles :

- **Le démarrage :** le moteur passe de la vitesse nulle jusqu'à sa vitesse établie en un temps prédéfini et en évitant les pointes d'intensité.
- **La variation de vitesse :** modification de la fréquence de rotation du moteur par accélération ou décélération en un temps donné.

- **La régulation** : la fréquence de rotation du moteur est maintenue constante quelles que soient les fluctuations de la charge (dans certaines limites).
- **Le freinage** : le moteur passe d'une vitesse établie à une vitesse inférieure (ralentissement) ou à la vitesse nulle (arrêt) avec maintien en position possible.
- **L'inversion du sens de rotation** : permet de faire fonctionner le moteur dans les deux sens de rotation. [5]

I.12-La commande de vitesse d'un moteur à courant continu :

I.12.1-Par les techniques classiques :

1. Le régulateur PID :

PID est un régulateur qui support des trois actions P, I et D. il caractérise par réunir les effets positifs des trois correcteurs de base (Proportionnel, Intégrateur et Dérivateur). Avantage de terme I, il permet l'annulation d'une erreur statique tout en autorisant avantage de l'action D des performances de rapidité à celles d'un régulateur PI.

2. Le régulateur PI :

Ou contrôle proportionnel, nous pouvons ajouter l'intégration de l'erreur. Dans ce cas nous obtenons une régulation PI (proportionnelle et intégré).

ET bien, lors d'un simple contrôle proportionnel, il subsiste une erreur statique. Lorsque le système s'approche de sa consigne, l'erreur n'est plus assez grande pour faire avancer le moteur. Le terme intégral permet ainsi de compenser l'erreur statique et fournit, par conséquent, un système plus stable en régime permanent. Plus k_i est élevé, plus l'erreur statique est corrigée. [6]

3. Le régulateur PD :

On notera que l'action D ne permettant pas la transmission d'un signal constant, elle doit donc toujours s'accompagner au moins d'une action P en parallèle.

I.12.2-Par les techniques avancées :

1. La commande optimale :

La théorie de la commande optimale couvre toutes les activités dynamiques où une performance optimale est exigée.

CHAPITRE I-Etat de l'art des moteurs à courant continu (MCC)

Les systèmes à commander peuvent donc être d'origine diverses : mécanique, électrique, électronique, biologie, chimie, économie,...

Chaque problème de commande nécessite une description des propriétés dynamiques du processus à commander.

- Détermination du modèle mathématique du système :

Les systèmes étudiés sont décrits par des variables d'état.

Par exemple :

- ✓ Systèmes linéaires continus commandés
- ✓ Systèmes non linéaires continus commandés
- ✓ Systèmes discrets linéaire
- ✓ Système discrets non linéaire

Outre ce modèle, il faut formuler, pour un problème de commande optimale, le critère de performance à optimiser et les contraintes physiques.

- Formulation de l'indice de performances et des contraintes physiques :

Il s'agit d'une grandeur mathématique désignée dans la littérature technique selon le domaine, critère (en automatique), fonction coût (en économie), fonctionnelle (en mathématique). Dans ce qui suit, on utilise le mot critère.

Les critères les plus utilisés sont :

- ✓ Problème à temps minimal :

$$J = \int_{t_0}^T dt \quad (I.3)$$

- ✓ Cas linéaire quadratique :

$$J = \int_{t_0}^T \left\{ \frac{1}{2} X^T Q X + \frac{1}{2} U^T R U \right\} dt \quad (I.4)$$

- ✓ Cas générale :

$$J = \phi(X(T), X, t) + \int_{t_0}^T L(X, U, t) dt \quad (I.5)$$

Où $L(X, U, t)$ est une fonction non linéaire et $\phi(X(T), X, t)$ représente la fonction cout terminal. [7]

2. La commande adaptative :

La commande adaptative est une commande dont le but est de réagir à tout instant dans le sens désiré (en générale minimisation de l'erreur entre la consigne et la sortie) face aux variations que subit le système.

CHAPITRE I-Etat de l'art des moteurs à courant continu (MCC)

▪ Les approches de commande adaptative peuvent être classées en deux classes:

✓ Commande adaptative directe.

✓ Commande adaptative indirecte.

▪ Il existe principalement quatre types d'approches de commande adaptatives :

✓ Commande par gain programmé :

Dans plusieurs cas, il est possible de les variables mesurables ont une corrélation avec les changements dans la dynamique du système Ces variables peuvent donc être utilisées pour changer les paramètres du contrôleur. Cette approche est appelée commande par gain programmé La commande dispose de deux boucles Peut être vue comme une transformation de l'espace des paramètres du système à l'espace des paramètres du contrôleur.

✓ Commande adaptative à Modèle de Référence :

Cette approche de commande a été originalement proposée pour résoudre un problème dans lequel les spécifications de performances sont données en termes d'un modèle de référence. Ce modèle de référence donne une indication sur comment la sortie du système doit idéalement répondre à un signal de commande Son principe de base consiste à adapter les paramètres du contrôleur en fonction de l'erreur entre le système et le modèle.

✓ Contrôleurs auto-ajustable :

Cette commande fait partie des commandes adaptatives indirectes Les paramètres du système sont estimés et utilisés dans le calcul des paramètres du contrôleur. L'architecture de commande contient deux boucles : une boucle interne du contrôleur et une boucle externe d'ajustement de ses paramètres.

✓ Commande dual :

Dans les approches précédentes, l'estimation des paramètres (du contrôleur ou du système) est supposée certaine. Les incertitudes sur les paramètres ne sont pas considérées dans la synthèse du contrôleur.

Pour cela l'idée consiste à utiliser « la théorie de commande stochastique » non linéaire' Ceci conduit à ce qu'on appelle 'Commande duale'. L'approche est tellement compliquée qu'elle n'a pas été appliquée en pratique. [8]

3. La commande robuste :

Une loi de commande robuste a pour objectif d'obtenir un fonctionnement acceptable d'un système réel dans son contexte normal d'utilisation. En premier lieu il est fondamental que le comportement du procédé commandé soit stable.

Ainsi, on parlera de stabilité robuste pour une loi de commande lorsque la stabilité de la boucle fermée n'est pas altérée par les écarts entre le modèle utilisé pour la synthèse de la loi

de commande et le procédé à commander. Ces imperfections de modèle peuvent être dues à des imprécisions intrinsèques liées à l'incapacité d'avoir une connaissance totale de tous les phénomènes mis en jeu, ou bien à des imprécisions fixées a priori, découlant d'approximations ou de réductions d'ordre, et permettant de mettre en œuvre certaines techniques de synthèse.

Pour obtenir une loi de commande robuste il est donc nécessaire de prendre en compte ces imperfections de modèle.

Pour ce faire, le modèle du procédé se compose d'une famille de systèmes (éventuellement non linéaires et de dimension infinie) définie à partir d'un modèle nominal.

Cette famille est donnée en spécifiant soit des perturbations sur le système nominal soit des incertitudes sur le système. [9]

I.13-Utilité des interfaces graphiques dans l'industrie :

La gestion des bases de données et des différents dispositifs n'a jamais été une mince affaire. Sans une interface graphique adaptée, les intervenants ne peuvent pas visualiser les processus en cours.

Dans ce contexte, l'outil fait office de humain machine interface. En d'autres termes, il traduit les commandes saisies par l'opérateur en signaux.

De même, il transforme les données collectées en des informations concrètes. Cette fonctionnalité joue un rôle important au sein d'une usine.

L'utilitaire facilite la manipulation des principaux outils tout en simplifiant considérablement le suivi des activités. [10]

I.13.1-Interface Homme Machine HMI :

Le progrès dans les technologies d'affichage de dernière génération permet d'améliorer l'interaction des personnes et des machines, facilitant l'utilisation et le contrôle de processus et obtenant de plus en plus, l'information du système à contrôler, d'un simple coup d'œil.

HMI, acronyme d'Humaine Machine Interface ou Interface Homme Machine est le moyen par lequel, un usager peut communiquer avec une machine et qui comprend tous les points de contact entre ce dernier et la machine concernée.

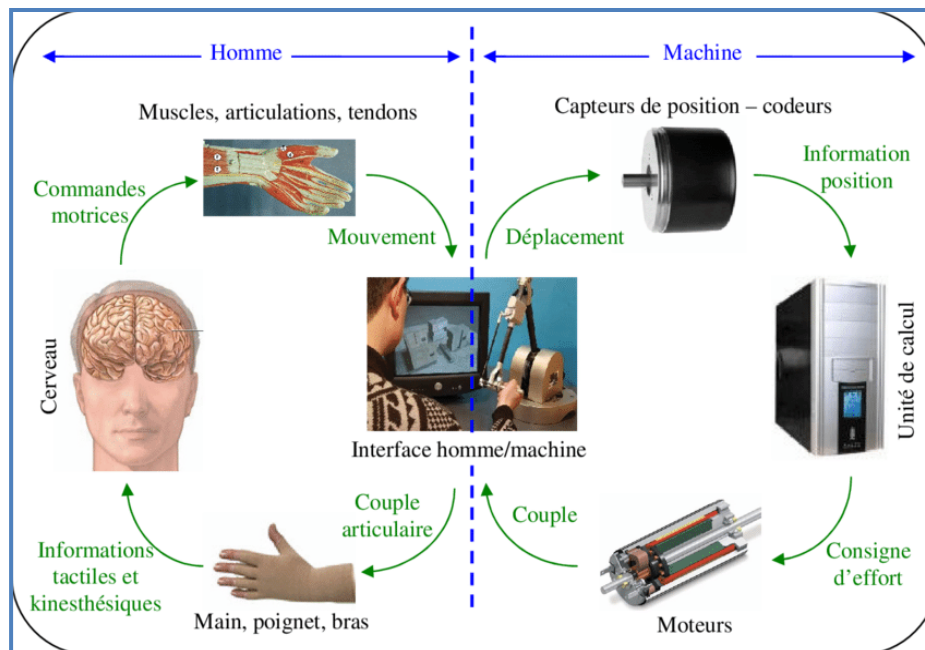


Figure I-12 : Schéma global de l'interface homme-machine

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.

I.14-Conclusion :

Le moteur à courant continu a été durant de nombreuses années la machine principalement utilisée dans les applications à vitesse variable, il est idéal pour un bon nombre d'applications industrielles. Alors, Dans ce chapitre nous avons présenté quelque généralité sur le moteur à courant continu, le principe de fonctionnement et ses caractéristiques.

CHAPITRE I-Etat de l'art des moteurs à courant continu (MCC)

Dans le chapitre qui se suit on va parler sur les interfaces graphiques et quelques exemples de leur programmation sur GUI-MATLAB.

Chapitre II

Les Interfaces

graphiques

sous GUI-Matlab

II.1-Introduction :

L'interface graphique désigne la manière dont est présenté un logiciel à l'écran pour l'utilisateur. C'est le positionnement des éléments : menus, boutons, fonctionnalités dans la fenêtre. Une interface graphique bien conçue est ergonomique et intuitive afin que l'utilisateur la comprenne tout de suite.

L'interface graphique est le langage d'échange entre l'humain (vous) et la machine (votre ordinateur). Votre ordinateur affiche à l'écran des éléments que vous comprenez et que vous interprétez. Chaque système dispose de sa propre interface graphique :

- Le tableau de bord de votre voiture
- Le distributeur de billet à la banque
- Votre téléviseur
- Votre ordinateur
- Votre tablette
- Votre Smartphone

La personne en charge de réaliser cette interface est le designer, parfois également appelé l'expert en expérience utilisateur. C'est lui qui définit l'apparence d'une interface à partir des besoins du système. Une interface se doit d'être intuitive et ergonomique. [11]

II.2-Interface graphique sous MATLAB :

Les interfaces graphiques sont appelées GUI (pour Graphical User Interface) sous MATLAB. 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. [12]

II.3-Présentation:

Le GUIDE est un constructeur d'interface graphique qui regroupe tous les outils dont le programmeur a besoin pour créer une interface graphique de façon intuitive. Il s'ouvre, soit en cliquant sur l'icône 'GUI', soit en tapant guide dans la Command Windows de MATLAB.

Le placement des objets est réalisé par sélection dans une boîte à outils. Leur mise en place et leur dimensionnement se font à l'aide de la souris. [12]

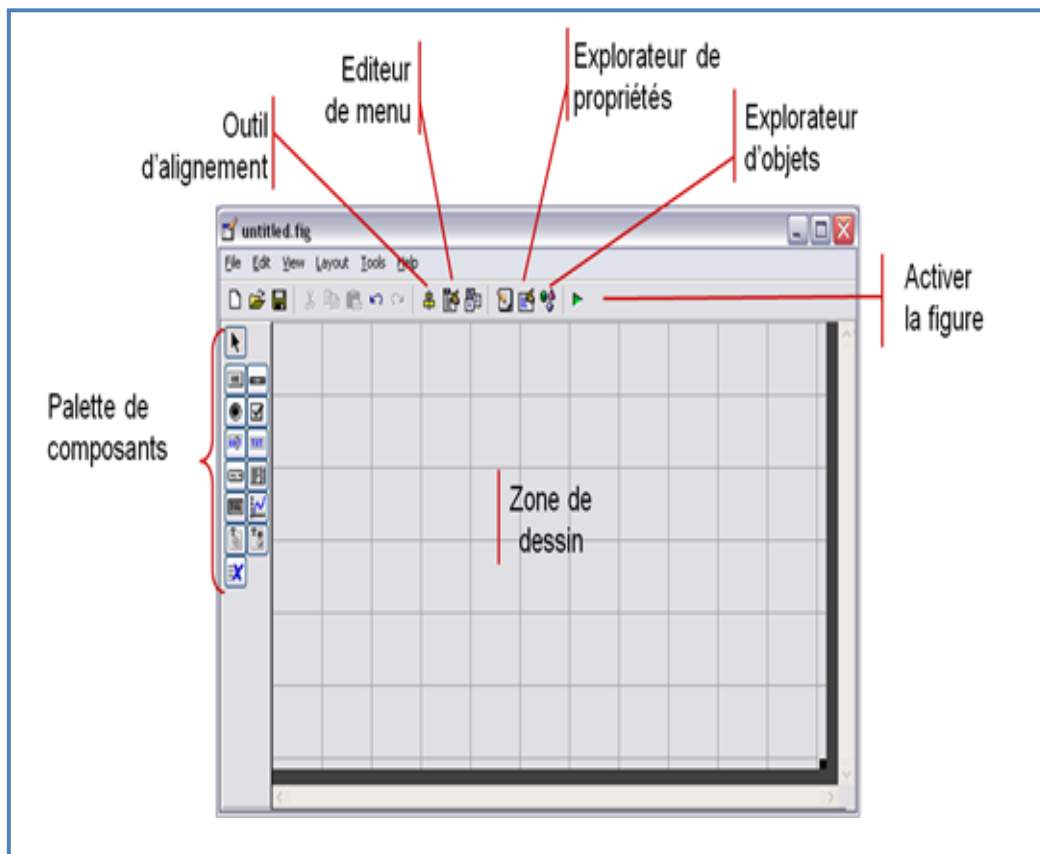


Figure II -1 : Présentation de l'interface de GUIDE

II.4-Les étapes pour faire un programme simple :

Réalisation d'une interface graphique sous GUI-MATLAB qui permet de tracer les graphes des différentes fonctions définies par un intervalle de définition (T_{min} , T_{max} et le pas) avec plusieurs options de traçage tel que le choix entre 2D et 3D, la couleur du graphe et l'épaisseur.

II.4.2-Ajouter des composants :

Pour ajouter un composant, il suffit d'appuyer sur le bouton correspondant à l'objet voulu, et en tracer un cadre sur la figure. L'objet sera de la taille du cadre dessiné.

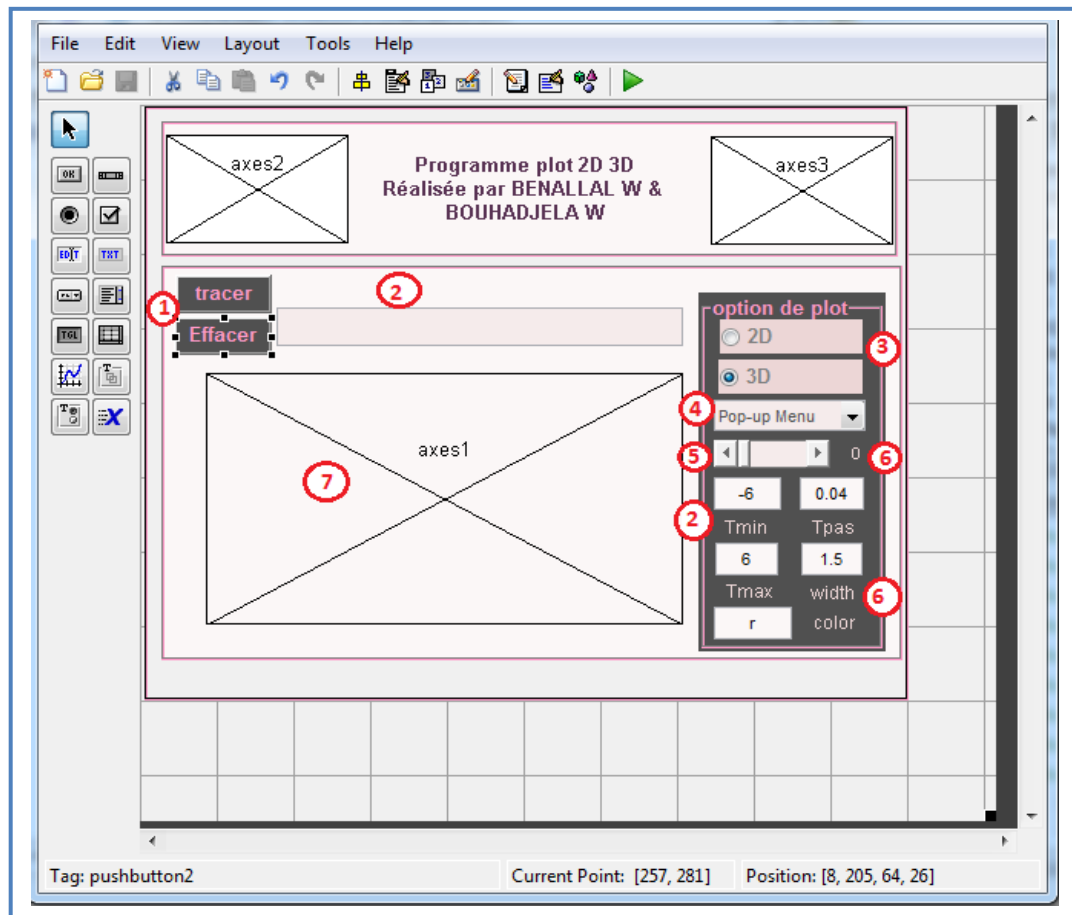


Figure II -4 : Placement des éléments de l'interface

1-Pushbuttons : le premier pour tracer les fonctions et le deuxième pour effacer.

2-Edit text : pour écrire les fonctions, modifier la couleur du graphe et l'épaisseur aussi pour définir l'intervalle de définition et le pas.

3-Radio button : le premier pour choisir le traçage en 2D et le deuxième pour choisir le traçage 3D.

4-Pop-up menu : présente la liste des fonctions.

5-Slider : pour modifier l'épaisseur du graphe.

6-Static text : pour définir les édites, pour afficher la valeur du Slider.

7-Axes : le premier pour le traçage des graphes, le deuxième et le troisième pour montrer le logo.

II.4.3-Définir les propriétés de chaque composant :

Un double-clic sur un objet permet de faire apparaître le Propreté Inspector où les propriétés des objets sont facilement éditables. Leurs modifications et la visualisation de ces modifications sont immédiates.

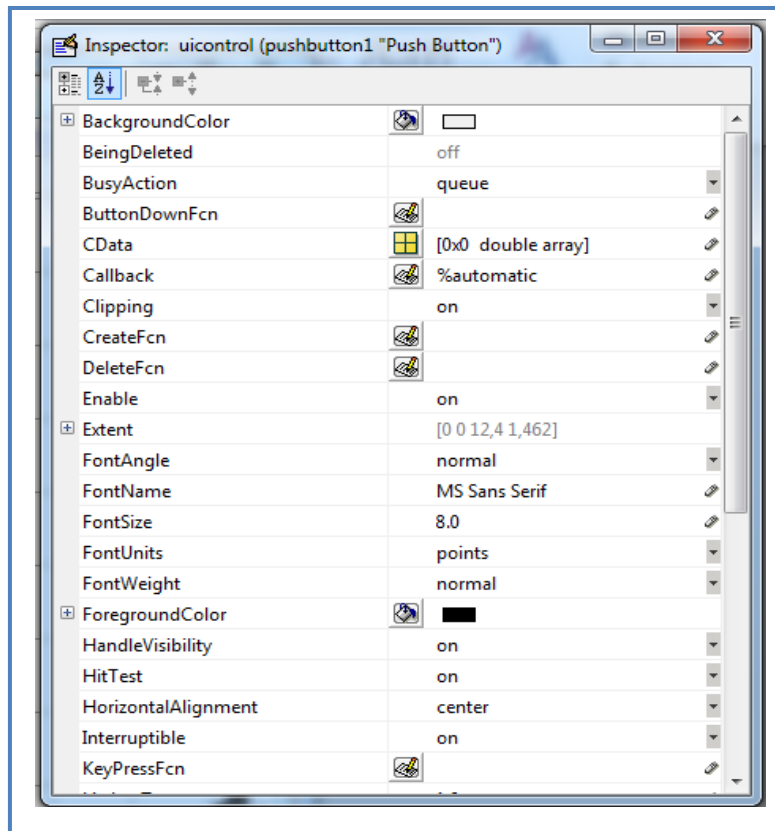


Figure II -5 : Propriété du bouton

Vous avez maintenant accès aux propriétés modifiables du bouton dans la fenêtre 'Inspector'. Parcourez cette liste jusqu'à atteindre 'String' du côté de la liste. La valeur du côté droit est le texte qui apparaît sur le bouton. Changez ce texte pour le nom du bouton qu'on veut et appuyez sur 'Entrée'. Regardez maintenant le texte sur le bouton.

Vous pouvez ainsi changer les autres propriétés de cette liste. Tous les objets ont une liste de propriétés modifiables à la main, comme on vient de faire, ou bien par programmation. Les propriétés les plus importantes pour chaque composant sont : le 'String' et le 'Callback'. [12]

II.4.4-Callback du bouton :

Vers la fin de notre programme, vous pourrez trouver le Callback qui a été généré pour un clic dans chaque élément.

- Pushbutton1, radiobutton1, radiobutton2

```
% --- Executes on button press in pushbutton.
function pushbutton_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
if get(handles.radiobutton1,'value')==1 && get(handles.radiobutton2,'value')==0
lw=str2num(get(handles.edit8,'string'));
col=get(handles.edit9,'string');
xmin=str2num(get(handles.edit2,'string'));
xmax=str2num(get(handles.edit3,'string'));
xi=str2num(get(handles.edit4,'string'));
xI=xmin:xi:xmax;
ymin=str2num(get(handles.edit2,'string'));
ymax=str2num(get(handles.edit3,'string'));
yi=str2num(get(handles.edit4,'string'));
yI=ymin:yi:ymax;
[x,y]=meshgrid(xI,yI);
z=eval(get(handles.edit1,'string'));
plot3(handles.axes1,x,y,z,col,'linewidth',lw)
grid(handles.axes1);
xlabel(handles.axes1,'x');
ylabel(handles.axes1,'y');
zlabel(handles.axes1,'z');
elseif get(handles.radiobutton1,'value')==0 && get(handles.radiobutton2,'value')==1
lin=str2num(get(handles.edit8,'string'));
col=get(handles.edit9,'string');
tmin=str2num(get(handles.edit2,'string'));
tmax=str2num(get(handles.edit3,'string'));
ti=str2num(get(handles.edit4,'string'));
t=tmin:ti:tmax;
y=eval(get(handles.edit1,'string'));
plot(handles.axes1,t,y,col,'linewidth',lin);
grid(handles.axes1);
xlabel(handles.axes1,'t');
ylabel(handles.axes1,'y');
end
```

- Pushbutton2

```
% --- Executes on button press in pushbutton2.
function pushbutton2_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
set(handles.edit1,'string','');
```

- Pompemenu

```
function popupmenu1_Callback(hObject, eventdata, handles)
% hObject    handle to popupmenu1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: contents = cellstr(get(hObject,'String')) returns popupmenu1 contents as cell array
%         contents{get(hObject,'Value')} returns selected item from popupmenu1
R=get(handles.popupmenu1,'Value')
if R==1
    plot(handles.axes1,1,1)
    grid(handles.axes1)

elseif R==2

    T1=str2num(get(handles.edit2,'String'))
    DT=str2num(get(handles.edit3,'String'))
    T2=str2num(get(handles.edit4,'String'))
    lw=str2num(get(handles.edit6,'String'))
    col=get(handles.edit5,'String')
    t=T1:DT:T2;
    y=cos(t)
    plot(handles.axes1,t,y,col,'LineWidth',lw)
    grid(handles.axes1)
elseif R==3
    T1=str2num(get(handles.edit2,'String'))
    DT=str2num(get(handles.edit3,'String'))
    T2=str2num(get(handles.edit4,'String'))
    lw=str2num(get(handles.edit6,'String'))
    col=get(handles.edit5,'String')
    t=T1:DT:T2;
    y=sin(t)
    plot(handles.axes1,t,y,col,'LineWidth',lw)
    grid(handles.axes1)
elseif R==4
    T1=str2num(get(handles.edit2,'String'))
    DT=str2num(get(handles.edit3,'String'))
    T2=str2num(get(handles.edit4,'String'))
    lw=str2num(get(handles.edit6,'String'))
    col=get(handles.edit5,'String')
    t=T1:DT:T2;
    y=exp(t)
    plot(handles.axes1,t,y,col,'LineWidth',lw)
    grid(handles.axes1)
end
```

- Slider :

```
% --- Executes on slider movement.  
function slider1_Callback(hObject, eventdata, handles)  
% hObject    handle to slider1 (see GCBO)  
% eventdata  reserved - to be defined in a future version of MATLAB  
% handles    structure with handles and user data (see GUIDATA)  
a=get(handles.slider1,'Value')  
set(handles.edit8,'String',a)  
set(handles.text1,'String',a)  
T1=str2num(get(handles.edit2,'String'))  
DT=str2num(get(handles.edit4,'String'))  
T2=str2num(get(handles.edit3,'String'))  
lw=str2num(get(handles.edit8,'String'))  
col=get(handles.edit9,'String')  
t=T1:DT:T2;  
y=eval(get(handles.edit1,'String'))  
plot(handles.axes1,t,y,col,'LineWidth',lw)  
grid(handles.axes1)
```

Les autres éléments on n'as pas besoin de programmation il suffit qu'on change directement sur l'interface graphique, et ces bouton on s'en sert pour :

- Edit2 : valeur maximal.
- Edit4 : modifier le pas de traçage.
- Edit3 : valeur minimal.
- Edit9 : changer la couleur.
- Edit1 : changer la fonction qu'on veut tracer.
- Edit8 : changer l'épaisseur
- Statictext: afficher la valeur de string.

II.4.5-Exécution de l'interface réalisée :

Une fois l'interface graphique terminée cliquer sur le bouton 'Run' pour lancer la figure, le GUI réalisée transfert en fichier auto-exécution (.exe) démarre sas logiciel MATLAB par la fonction de Toolbox-MATLAB 'deploytool'

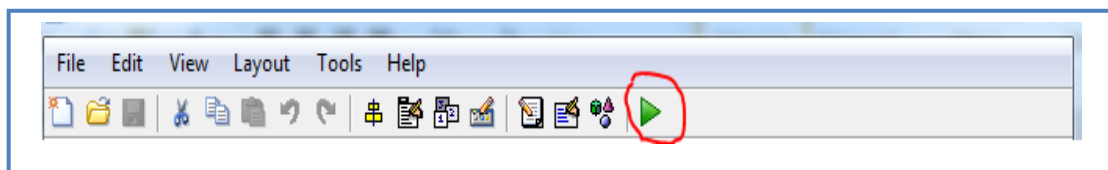


Figure II-6 : activer l'interface

CHAPITRE II-Interface graphique sous GUI MATLAB

Son enregistrement donne deux fichiers portant le même nom mais dont les deux extensions sont '.fig' et '.m'.

Le fichier '.fig' contient la définition des objets graphiques (positions et propriétés).

Ce fichier peut être ouvert ultérieurement avec le GUIDE pour modifier les objets graphiques.

Le fichier '.m' contient les lignes de code qui assurent le fonctionnement de l'interface graphique (actions des objets).

Ce fichier peut être édité dans le MATLAB Editor pour y ajouter des actions à la main. C'est ce fichier qui doit être lancé pour utiliser l'interface graphique.

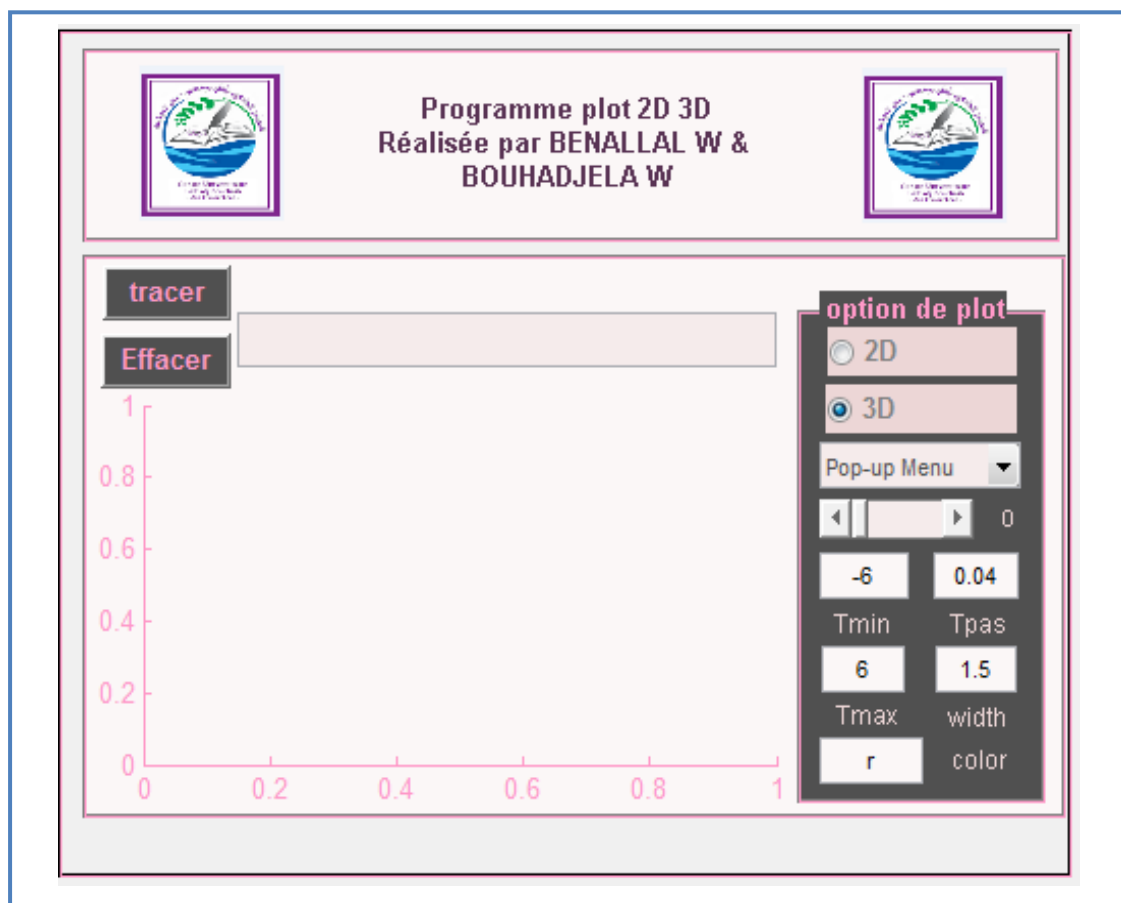


Figure II -7:Résultat du test

Dans ce moment on peut tracer les différentes fonctions sur l'interface par plusieurs méthodes :

- 1 ère méthode : on sélectionne la fonction choisie dans la liste des fonctions

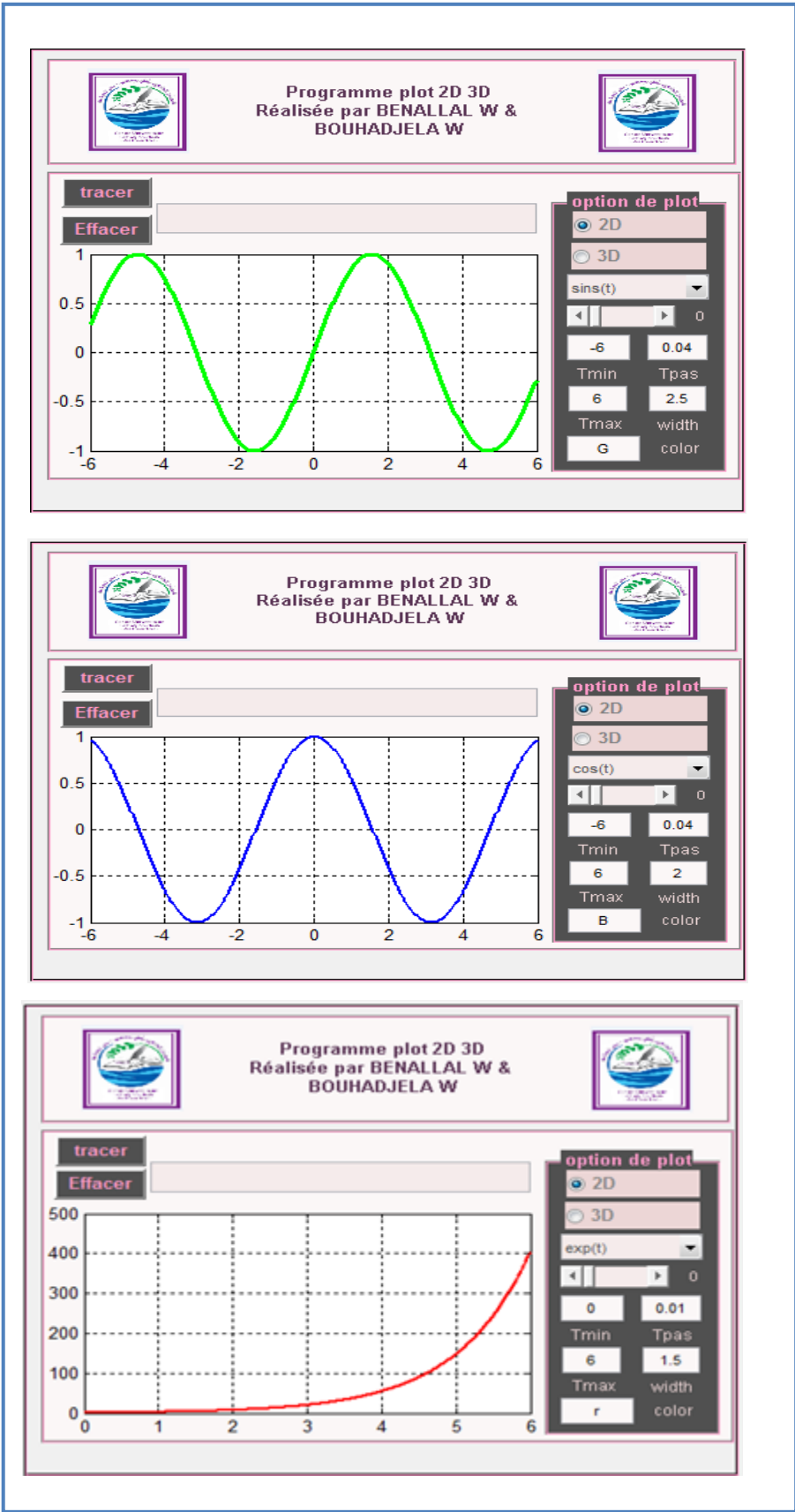


Figure II -8:tracer $\cos(t)$, $\sin(t)$ et $\exp(t)$

- 2ème méthode : écrire la fonction qu'on veut tracer dans 'edit' puis appuyer sur tracer.

Tracer en 2D la fonction : $f(t) = 10 \times e^{(-50 \times t)} \times \sin(314 \times t)$ (II.1)

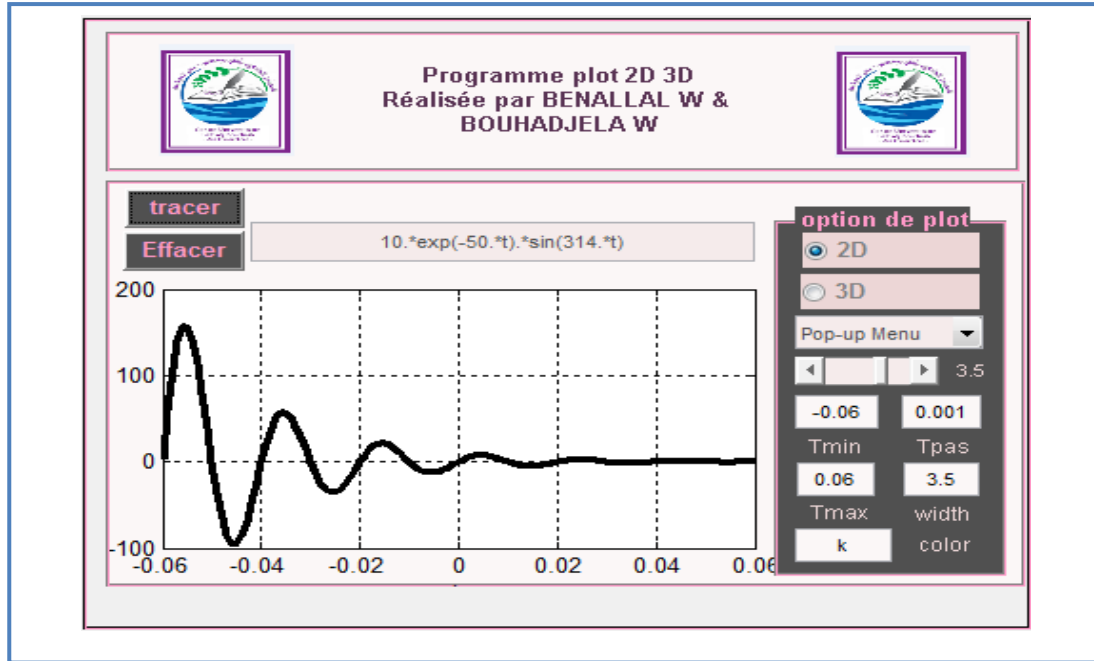


Figure II-9 : Fonction 2D

Tracer en 3D la fonction: $f(t) = (1 - x)^2 \times e^{(x^2 - (y+1)^2)} - (-x - x^3 - y^5) \times e^{(-x^2 - y^2)}$ (II.2)

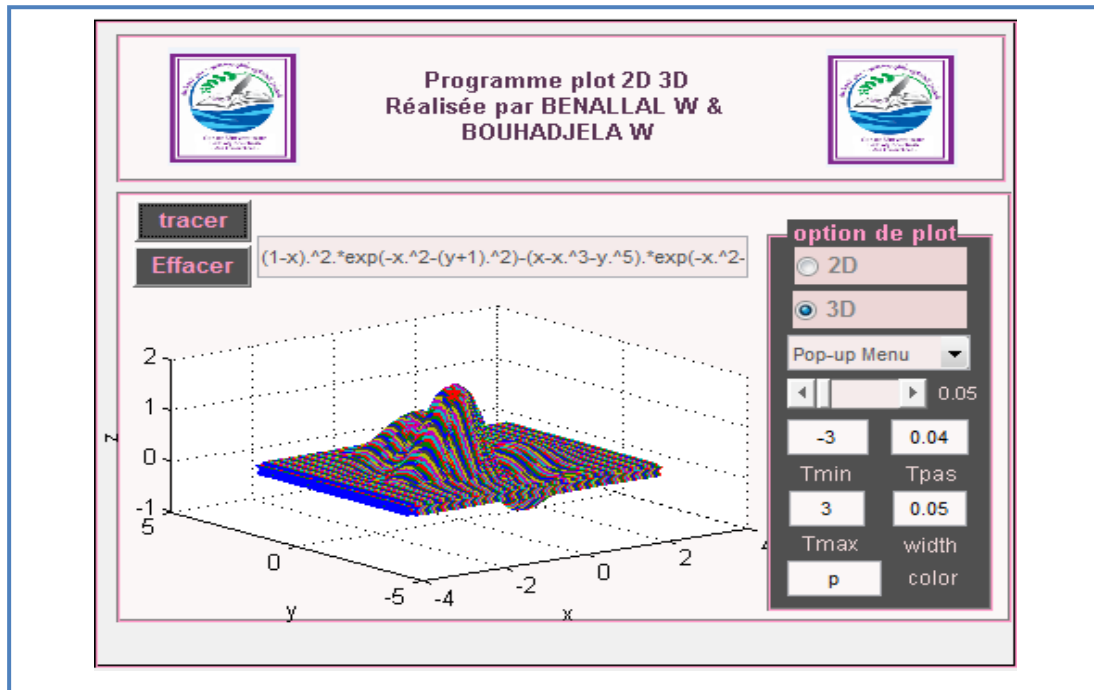


Figure II-10 : Fonction 3D

II.5-Réalisation d'une calculatrice scientifique et graphique :

On a réalisé une interface sous GUI-MATLAB d'une calculatrice scientifique et graphique.

Pour faire le calcul numérique on appuie sur les Pushbuttons pour déclarer l'opération dans 'edit' puis on appuie sur le Pushbutton (=), l'affichage effectuée aussi dans 'edit'.

Pour dessiner les graphes on déclare l'équation en fonction de 'x' dans 'edit' puis on appuie sur Pushbutton 'Graphe' et le graphe s'affiche dans 'axe'.

- **Exemple1 :** le calcul numérique de $V = 380/\sqrt{3}$ (II.3)

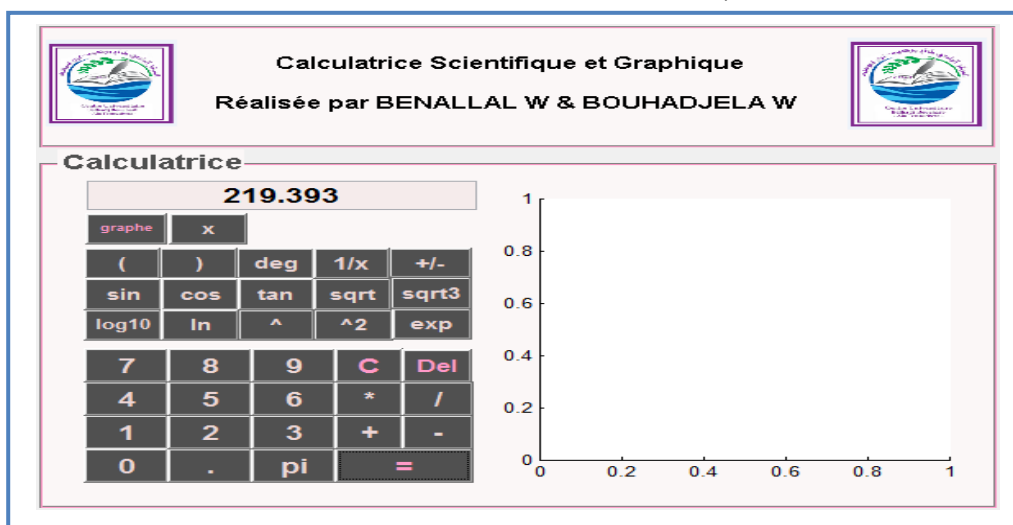


Figure II-11 : Calcul numérique

- **Exemple2 :** Tracer la fonction $f = x^2$ (II.4)

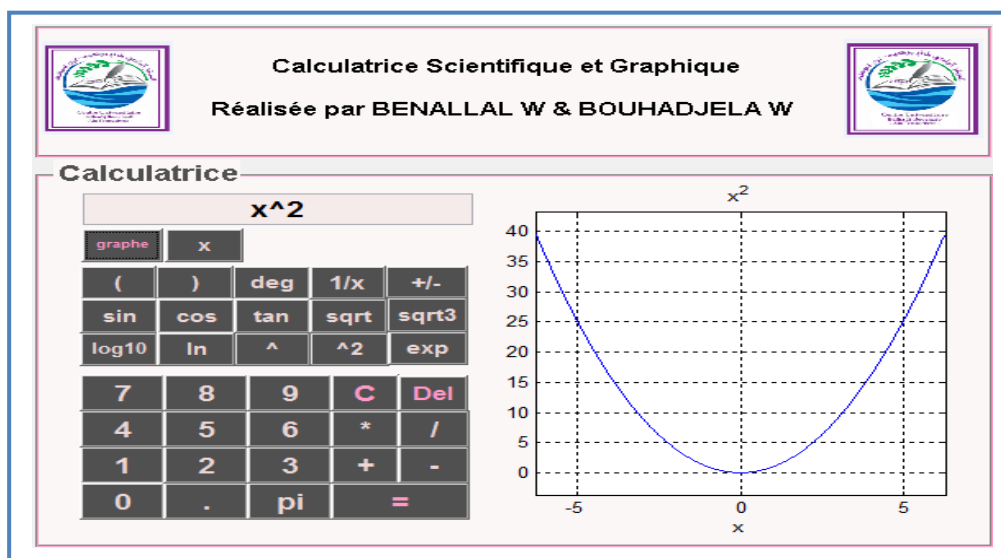


Figure II-12 : Traçage du graphe

- **Exemple 3** : calcul numérique et traçage du graphe de la fonction

$$f = (10 \times \cos(45) + \sin(90)) \times \log(10)/e^{100} \quad (\text{II.5})$$

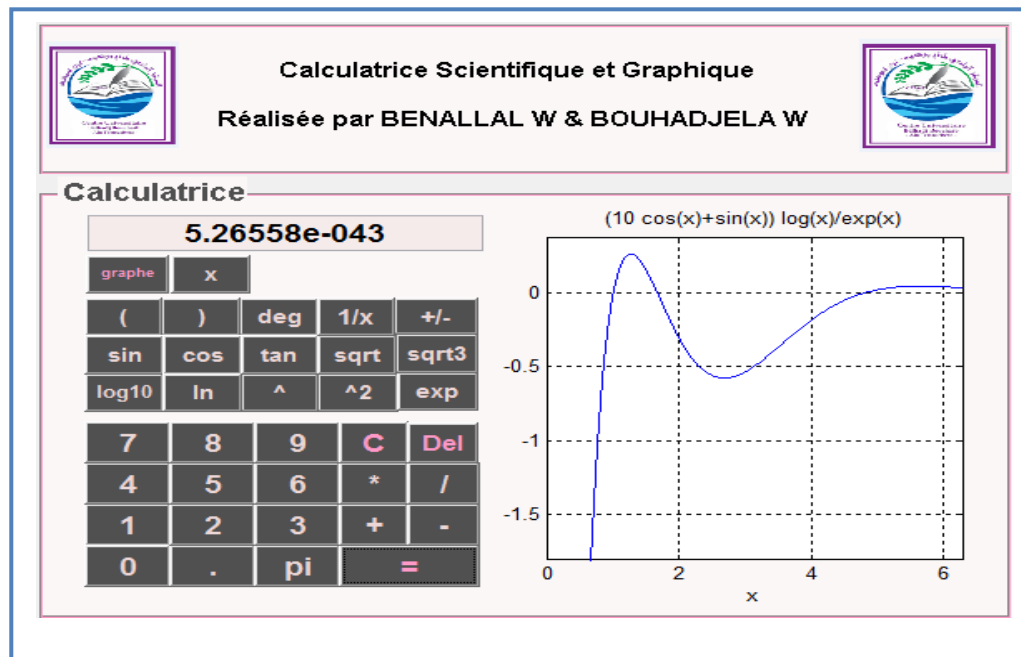


Figure II-13: Calcul et traçage

II.6-Conclusion :

L'environnement GUIDE de MATLAB permet de développer des GUI, des interfaces graphiques un GUI est défini dans MATLAB par deux fichiers dépendants. Respectivement une figure et un script. La programmation d'un GUI utilise des callbacks, et les échanges de données s'opèrent avec des 'handles'.

Le GUIDE de MATLAB est un outil graphique qui regroupe tout ce dont le programmeur a besoin pour créer une interface graphique de façon intuitive.

Le chapitre suivant présentera une étude et simulation de notre système implémenté sous interface graphique développé sous MATLAB.

Chapitre III

**Modélisation et
Simulation sous
interface graphique
sous GUI-Matlab**

III.1-Introduction :

L'objectif de ce chapitre est la régulation de la vitesse d'un moteur à courant continu à excitation séparée, alimenté par un hacheur, en utilisant un régulateur de type Proportionnel-intégral (PI) implémenté sous interface graphique réalisé sous GUI-MATLAB a permet :

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

Le hacheur est un convertisseur statique permettant à partir d'une source de tension U continu fixe, d'obtenir une tension U_c continu de valeur moyenne variable. On utilise un interrupteur statique pour 'hacher' la tension continu et alimenter une charge. Nous allons nous intéresser au cas particulier d'un hacheur série dont on peut commander le rapport cyclique α .

III.2-Modélisation de la source triphasée :

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

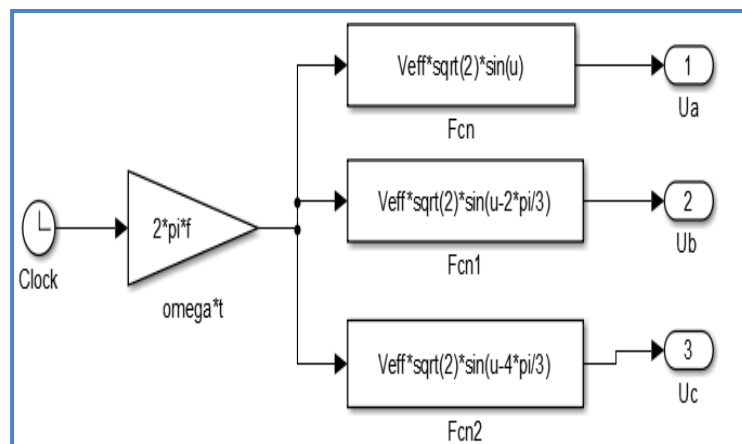


Figure III.1:schéma bloc de la source triphasée sous MATLAB

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

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

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

III.3-Modélisation du redresseur :

Un redresseur, également appelé convertisseur statique. Le redresseur consiste à transformer une tension ou un courant alternatif en son équivalent continu.

Ce redresseur (PD3) comporte trois diodes (D1, D2, D3) à cathode commune assurant l'allée du courant I_d et trois diodes (D4, D5, D6) à anode commune assurant le retour du courant I_d .

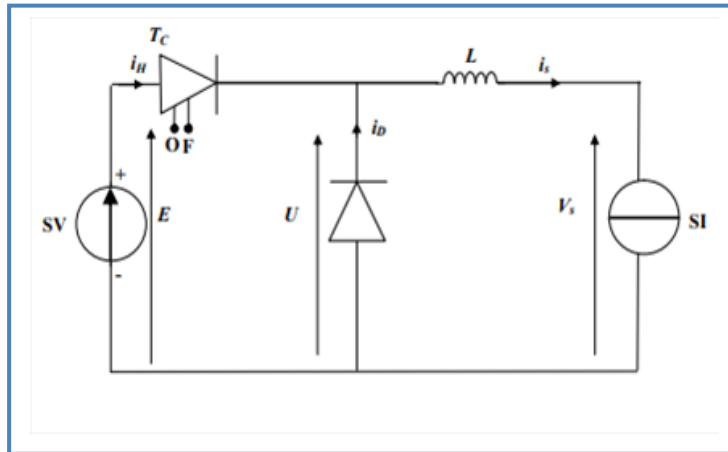


Figure III.2:Représentation schématique d'un redresseur PD3

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

$$U_{red} = \text{Max} [U_{a(t)} \cdot U_{b(t)} \cdot U_{c(t)}] - \text{Min} [U_{a(t)} \cdot U_{b(t)} \cdot U_{c(t)}] \tag{III.4}$$

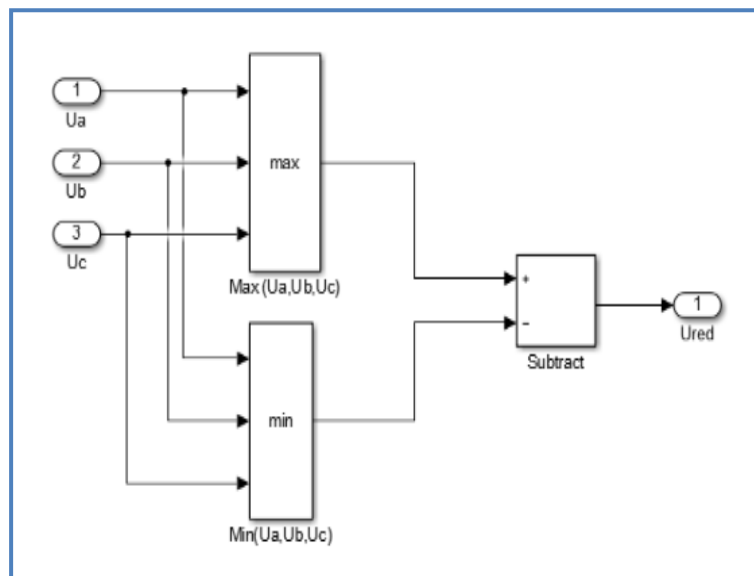


Figure III.3:schéma bloc de redresseur sous MATLAB

III.4-Modélisation du filtre :

Le filtre passe-bas est un dispositif qui donne une réponse en fréquence relativement constante à la basse fréquence.

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

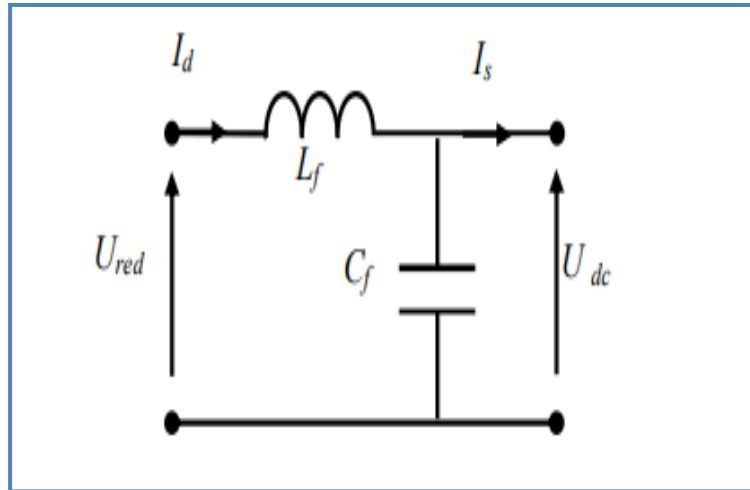


Figure III.4: filtre passe-bas

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

$$\omega_c = \frac{1}{\sqrt{L_f C_f}} \tag{III.5}$$

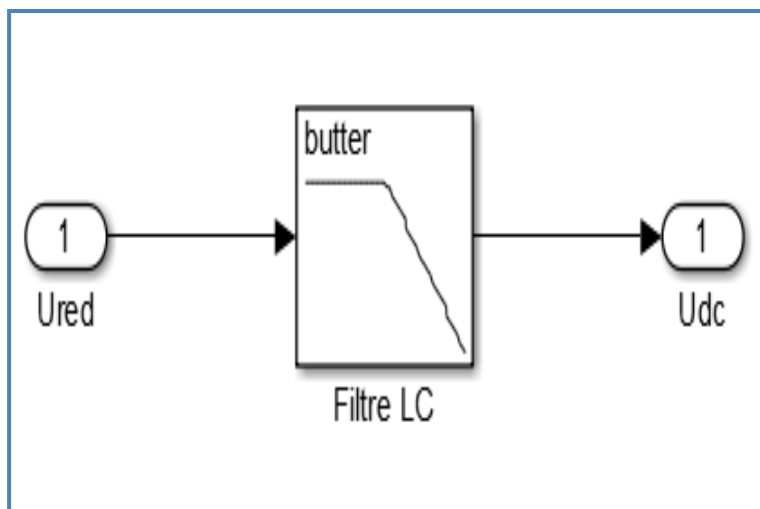


Figure III.5:schéma bloc du filtre passe-bas de Butter Worth sous MATLAB

III.5-Convertisseur statique (hacheur) :

III.5.1-Définition d'un hacheur :

Ces hacheurs sont utilisés pour faire varier la vitesse de rotation des moteurs à courant continu, les interrupteurs sont réalisés par des transistors bipolaires, transistors MOS, thyristor, etc. Les critères de choix d'hacheur sont :

- Types des interrupteurs utilisés
- Mode d'utilisation selon l'application
- Technique de commande appliquée au MCC.

Les convertisseurs généralement utilisés dans les applications des MCC à vitesse variable, sont de type hacheur ou pont à thyristors. L'élément de base de cet hacheur série est un transistor.

III.5.2-Principe d'un hacheur série (abaisseur de tension) :

Le générateur (SV) impose une tension qui reste sensiblement constante, le hacheur série permet de réduire de façon continue la tension moyenne U_{moy} . L'interrupteur unidirectionnel laisser passer i_H pendant la durée TF, ensuite il y'a ouverture de l'interrupteur et $i_H=0$, tandis que le courant continue à circuler dans la charge à cause de l'inductance dans la source de courant (SI) et de la diode de roue libre (continuité du courant dans l'inductance).

Pour obtenir α , on compare la tension d'entrée (Vcom : tension de commande) à un signal en dents de scie (Vscie) généré par la partie commande du hacheur. Le fruit de la comparaison est le signal de commande (Vint) envoyé sur les transistors (IGBT ou MOS). [14]

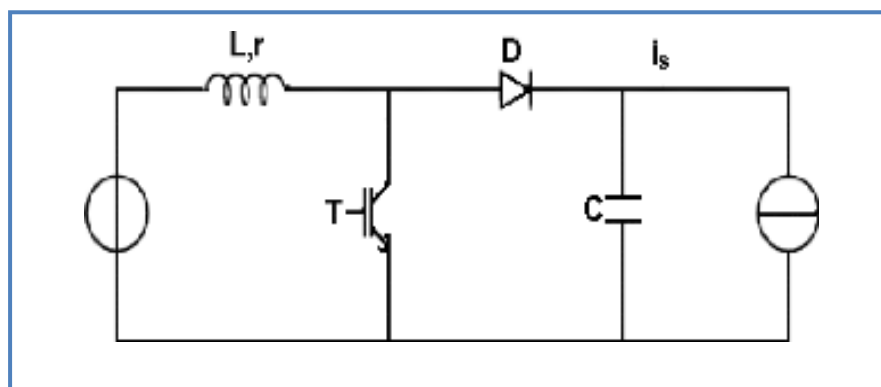


Figure III.6:Modélisation d'un hacheur série

III.6-Générateur d'impulsion :

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

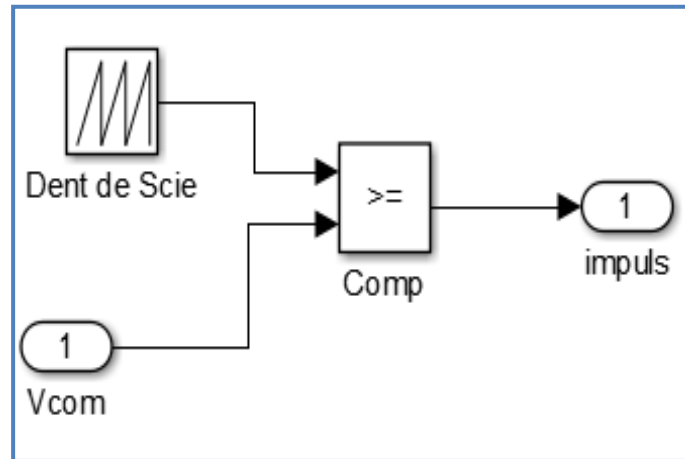


Figure III.7:Générateur d'impulsion sous MATLAB

Avec les impulsions ainsi produite, on attaque la gâchette du transistor, on remarque qu'on variant la tension V_{com} les imputions changerons à 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. [14]

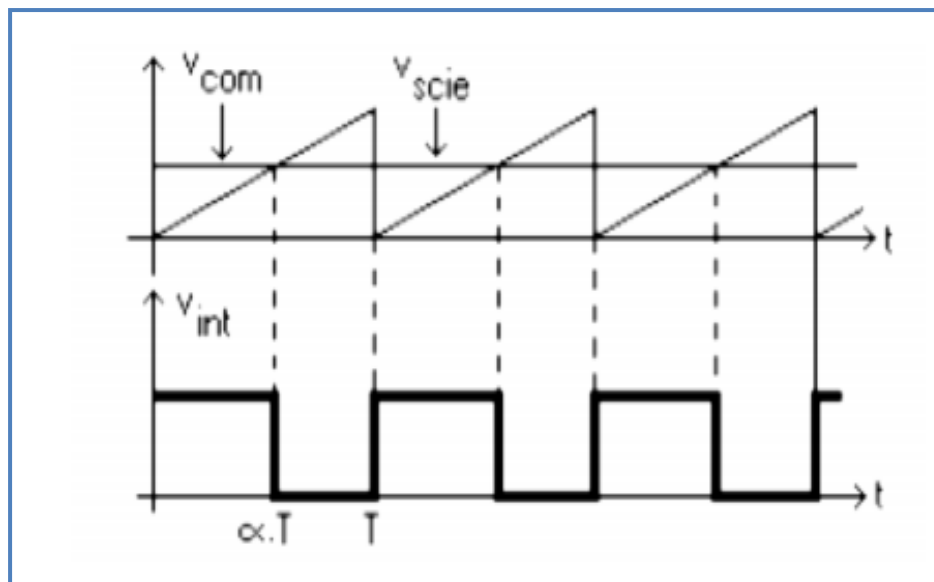


Figure III.8 : Génération du signal de commande par la technique MLI

III.7- Modélisation du MCC :

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

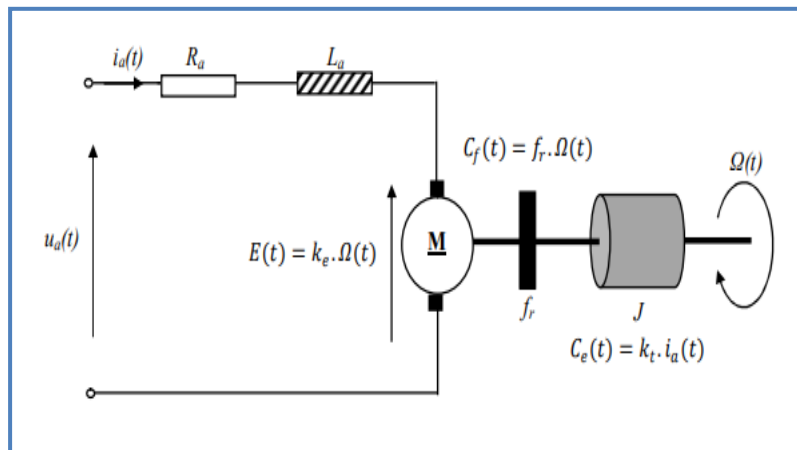


Figure III.9 : Schéma technologique de la machine à courant continu

Un moteur électrique à courant continu est régi par les équations physiques découlant de ses caractéristiques électrique, mécanique et magnétiques.

Equations électromécaniques :

$$C_e = K_t I_a \quad (\text{III.6}) \quad \text{Avec : } K_t = K\Phi \quad (\text{III.7})$$

$$E = K_e \Omega \quad (\text{III.8}) \quad \text{Avec : } K_e = K\Phi \quad (\text{III.9})$$

Equations électriques :

$$U = E + R_a I_a + R_a \frac{dI_a}{dt} \quad (\text{III.10})$$

Equations mécaniques :

$$J \frac{d\Omega}{dt} + f_r \Omega = C_e - C_r \quad (\text{III.11})$$

Ω : la vitesse de rotation du rotor.

C_e : Le couple électromagnétique fourni par le moteur.

E : La force contre électromotrice générée par le moteur.

U : La tension d'induit.

R_a, L_a : La résistance et l'inductance d'induit.

J : Le moment d'inertie du moteur.

f_r : Coefficient de frottement visqueux de l'ensemble moteur et charge.

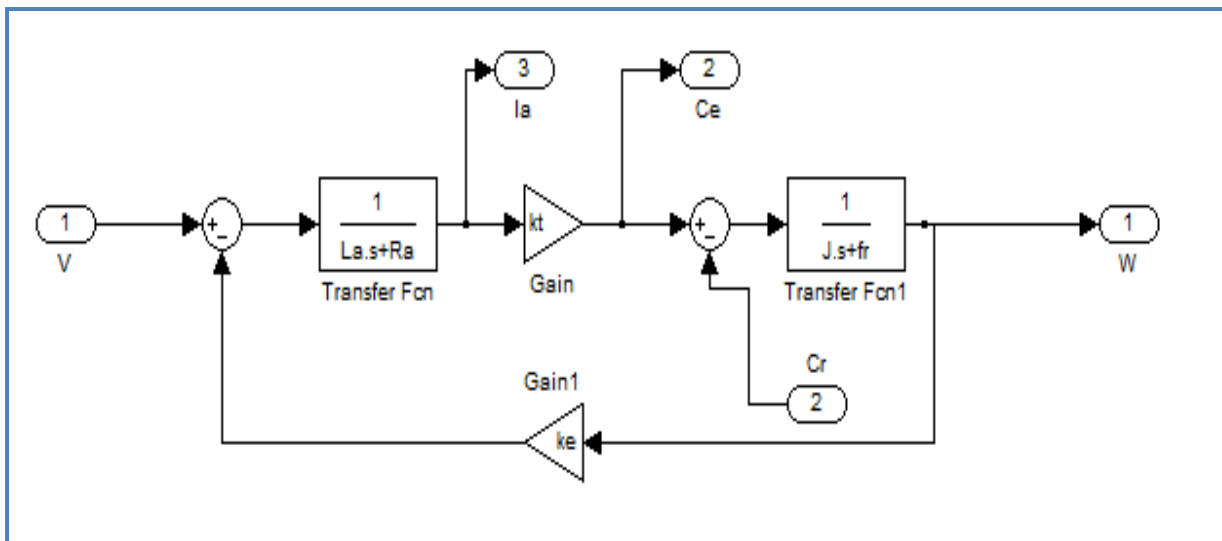
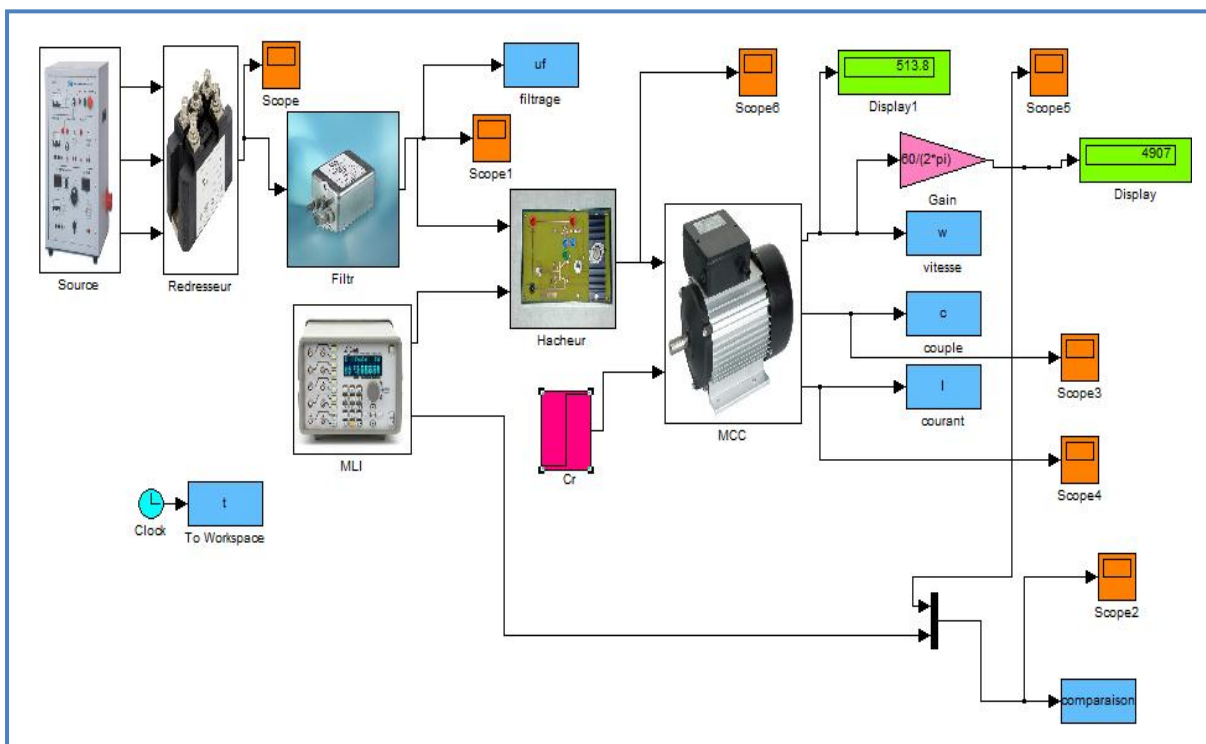


Figure III.10 : schéma bloc du MCC sous MATLAB

III.8-MCC en boucle ouverte:



- Sans utilisation du couple résistant C_r :

$$J \frac{d\Omega}{dt} C_e - f_r \Omega \quad (III.12)$$

$$JP\Omega(P) = C_e - f_r \Omega(P) \quad (III.13)$$

$$\Omega(P) \cdot (JP + f_r) = Ce \quad (\text{III.14})$$

$$\Omega(P) = \frac{Ce}{JP + f_r} \quad (\text{III.15})$$

$$Ce = k_t \cdot I_a(P) \quad (\text{III.16})$$

$$U(P) = R_a I_a(P) + L_a P I_a(P) + E \quad (\text{III.17})$$

$$U(P) - E = I_a(P) \cdot (R_a + L_a P) \quad (\text{III.18})$$

$$I(P) = \frac{U(P) - E}{R_a + L_a P} \quad (\text{III.19})$$

(16) dans (19)

$$Ce = \frac{k_t(U(P) - E)}{R_a + L_a P} \quad (\text{III.20})$$

$$Ce = \frac{k_t U(P) - k_t E}{R_a + L_a P} \quad (\text{III.21})$$

$$Ce = \frac{k_t U(P) - k_t k_e \Omega(P)}{R_a + L_a P} \quad (\text{III.22})$$

(III.22) dans (III.15)

$$\Omega(P) = \frac{k_t U(P) - k_t k_e \Omega(P)}{(R_a + L_a P)(JP + f_r)} \quad (\text{III.23})$$

$$\Omega(P) \cdot (R_a + L_a P)(JP + f_r) = k_t \cdot U(P) - k_t \cdot k_e \Omega(P) \quad (\text{III.24})$$

$$\Omega(P) \cdot (R_a + L_a P)(JP + f_r + k_t \cdot k_e) = k_t \cdot U(P) \quad (\text{III.25})$$

• Fonction de transfert du moteur :

La fonction de transfert cherchée H(p) est entre la tension U(p) et la vitesse de sortie Ωp) :

$$G(P) = \frac{S(P)}{E(P)} = \frac{\Omega(P)}{U(P)} \quad (\text{III.26})$$

$$G(P) = \frac{k_t}{(R_a + L_a P)(JP + f_r + k_t \cdot k_e)} \quad (\text{III.27})$$

$$G(P) = \frac{k_t}{L_a J P^2 + (R_a J + L_a f_r) P + R_a f_r + k_t \cdot k_e} \quad (\text{III.30})$$

On écrit l'équation sous la forme 2^{ème} ordre :

$$G(P) = \frac{k_t / R_a f_r + k_t \cdot k_e}{\frac{L_a J}{R_a f_r + k_t \cdot k_e} P^2 + \frac{R_a J + L_a f_r}{R_a f_r + k_t \cdot k_e} P + 1} \quad (\text{III.31})$$

• Application du couple résistant Cr :

$$J \frac{d\Omega}{dt} = Ce - Cr - f_r \Omega \quad (\text{III.32})$$

$$JP\Omega(P) = Ce - Cr - f_r \Omega(P) \quad (\text{III.33})$$

$$\Omega(P). (JP+f_r)= Ce - Cr \quad (III.34)$$

$$\Omega(P) = \frac{Ce}{JP+f_r} - \frac{Cr}{JP+f_r} \quad (III.35)$$

$$\Omega(P) = \frac{k_t.U(P)-k_t.k_e.\Omega(P)}{(R_a+L_aP)(JP+f_r)} - \frac{Cr}{JP+f_r} \quad (III.36)$$

$$\Omega(P) = \frac{k_t.U(P)}{(R_a+L_aP)(JP+f_r)} - \frac{k_t.k_e.\Omega(P)}{(R_a+L_aP)(JP+f_r)} - \frac{Cr}{JP+f_r} \quad (III.37)$$

$$\Omega(P) + \frac{k_t.k_e.\Omega(P)}{(R_a+L_aP)(JP+f_r)} = \frac{k_t.U(P)}{(R_a+L_aP)(JP+f_r)} - \frac{Cr}{JP+f_r} \quad (III.38)$$

$$\Omega(P) \frac{(R_a+L_aP)(JP+f_r)+k_t.k_e}{(R_a+L_aP)(JP+f_r)} = \frac{k_t.U(P)}{(R_a+L_aP)(JP+f_r)} - \frac{Cr}{JP+f_r} \quad (III.39)$$

$$\Omega(P) = \frac{k_t.U(P)-Cr(R_a+L_aP)}{(R_a+L_aP)(JP+f_r)+k_t.k_e} \quad (III.40)$$

$$\Omega(p) = \frac{k_t.U(P)-Cr(R_a+L_aP)}{JL_aP^2+(R_aJ+L_af_r)P+R_af_r+k_t.k_e} \quad (III.41)$$

On écrit l'équation sous la forme de Boode :

$$\Omega(P) = \frac{\frac{k_t.U(P)}{k_t.k_e+R_af_r} - \frac{Cr(R_a+L_aP)}{k_t.k_e+R_af_r}}{\frac{JL_a}{k_t.k_e+R_af_r}P^2 + \frac{(R_aJ+L_af_r)}{k_t.k_e+R_af_r}P + 1} \quad (III.42)$$

III.9-Synthèse de régulateur PI :

III.9.1-Principe générale de la régulation :

Dans la plupart des appareils dans des installations industrielles et domestiques, il est nécessaire de maintenir des grandeurs physiques à des valeurs déterminées, en dépit des variations extrêmes ou internes influant sur ces grandeurs.

Par exemple, la vitesse et la position des moteurs, étant par nature variables, doivent donc être réglés par des actions convenables sur le processus considéré.

Si les perturbations influant sur la grandeur à contrôler sont lentes ou négligeables, un simple réglage dit en boucle ouverte, permet d'obtenir et de maintenir la valeur demandée par exemple : action sur un robinet d'eau).

Dans la majorité des cas, ce type de réglage n'est pas suffisant, car trop grossier ou instable.

Il faut alors comparer, en permanence, la valeur mesurée de la grandeur réglée à celle que l'on souhaite obtenir et agir en conséquence sur la grandeur d'action, dite grandeur réglant.

On a, dans ce cas, constitué une boucle de régulation et plus généralement une boucle d'asservissement. Cette boucle nécessite la mise en oeuvre d'un ensemble de moyens de mesure, de traitement de signal ou de calcul, d'amplification et de commande d'actionneur, constituant une chaîne de régulation ou d'asservissement.

La consigne est maintenue constante se produit sur le procédé une modification des entrées perturbatrices.

L'aspect régulation est considéré comme le plus important dans le milieu industriel, car les valeurs des consignes sont souvent fixes.

Néanmoins, pour tester les performances et la qualité d'une boucle de régulation, on s'intéresse à l'aspect asservissement. [2]

III.9.2-Performances de systèmes réglés :

- **Rapidité :**

La rapidité quantifie le temps de réponse du système. Elle correspond au temps de réaction de la sortie par rapport à la consigne.

Le temps mis par la réponse pour ne plus dépasser 5% de la valeur finale. Ce temps est retenu comme critère de rapidité 5%.

- **Précision :**

La précision quantifie l'erreur lorsque l'équilibre est atteint, Avec l'entrée et la sortie de même nature.

Autrement, un système est précis si la sortie suit la consigne en toutes circonstances avec un écart inférieur à la valeur définie dans un cahier des charges.

- **Stabilité :**

On dit qu'un système est stable lorsque celui-ci tend à revenir à son état d'équilibre pour une consigne constante, la sortie doit être constante. [2]

III.10-La méthode de régulation idéale :

La figure ci-dessous présente le schéma bloc du moteur à courant continu en boucle fermée avec régulation idéale.

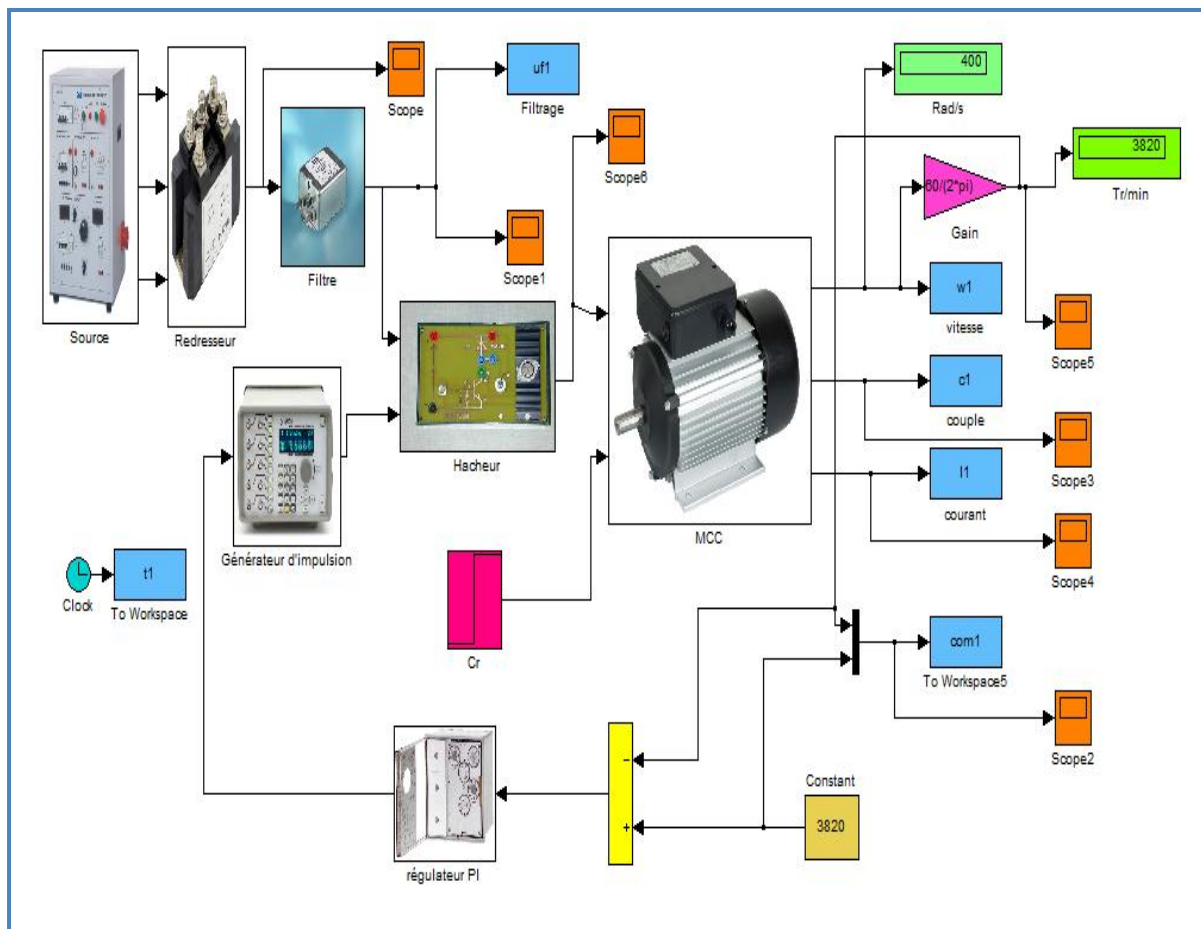


Figure III.12 : Schéma bloc de la commande de la MCC en boucle fermée

Plusieurs méthodes de la synthèse des paramètres du régulateur développé :

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

III.10.1-Spécification du cahier de charge :

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

- Stabilité du système

- Système suivi la trajectoire de la référence
- Dépassement négligeable
- Rapidité (temps de montée moins de 0.01s, temps d'établissement moins de 0.003)
- 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 du 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. [15]

Dans les deux la structure du système en boucle fermée est la suivante :

La fonction de transfert en boucle ouverte est : $FTBO=R_{(p)}G_{(p)}$ (III.43)

La fonction de transfert en boucle fermée est : $FTBF=\frac{R_{(p)}G_{(p)}}{1+R_{(p)}G_{(p)}}$ (III.44)

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

$$R_{(p)}G_{(p)} = \frac{1}{T_p} \quad (III.45)$$

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 :

$$R_{(p)} = \frac{1}{T_p G_{(p)}} \quad (III.46)$$

En conséquence de choix, la fonction du transfert en boucle fermée est :

$$FTBF=\frac{R_{(p)}G_{(p)}}{1+R_{(p)}G_{(p)}} = \frac{\frac{1}{T_p}}{1+\frac{1}{T_p}} = \frac{1}{1+T_p} \quad (III.47)$$

C'est-à-dire un comportement transitoire d'ordre un avec une précision statique parfaite.

Les performances sont alors :

- Stabilité inconditionnelle
- Précision statique parfaite
- Régime transitoire apériodique avec un temps de réponse fixe par le choix de T

$$G(P) = \frac{k_t / R_a f_r + k_t k_e}{\frac{L_a J}{R_a f_r + k_t k_e} P^2 + \frac{R_a J + L_a f}{R_a f_r + k_t k_e} P + 1} \quad (\text{III.48})$$

On néglige l'inductance (L) par rapport la résistance (R)

On aura:

$$G(p) = \frac{K_m}{T_m P + 1} \quad (\text{III.49})$$

$$\text{Avec } K_m = \frac{k_t}{R_a f_r + k_t k_e} \quad (\text{III.50})$$

$$T_m = \frac{R_a J}{R_a f_r + k_t k_e} \quad (\text{III.51})$$

$$R(p) = \frac{1}{TPG(p)} \quad (\text{III.52})$$

$$R(p) = \frac{1}{TP \left(\frac{K_m}{T_m P + 1} \right)} = \frac{1}{\frac{TP K_m}{T_m P + 1}} = \frac{T_m P + 1}{K_m TP} \quad (\text{III.53})$$

$$K_p = \frac{T_m}{K_m T} \quad (\text{III.54})$$

$$K_i = \frac{1}{K_m T} \quad (\text{III.55})$$

$$K_i = \frac{1}{T_i} \quad (\text{III.56})$$

$$T_i = K_m T \quad (\text{III.57})$$

C'est-à-dire un régulateur de type PI avec un gain proportionnel $K_p = T_m / K_m T$ et de constante d'intégration $T_i = K_m T$ La constante de temps T sera choisie de manière à fixer le temps de réponse en boucle fermée.

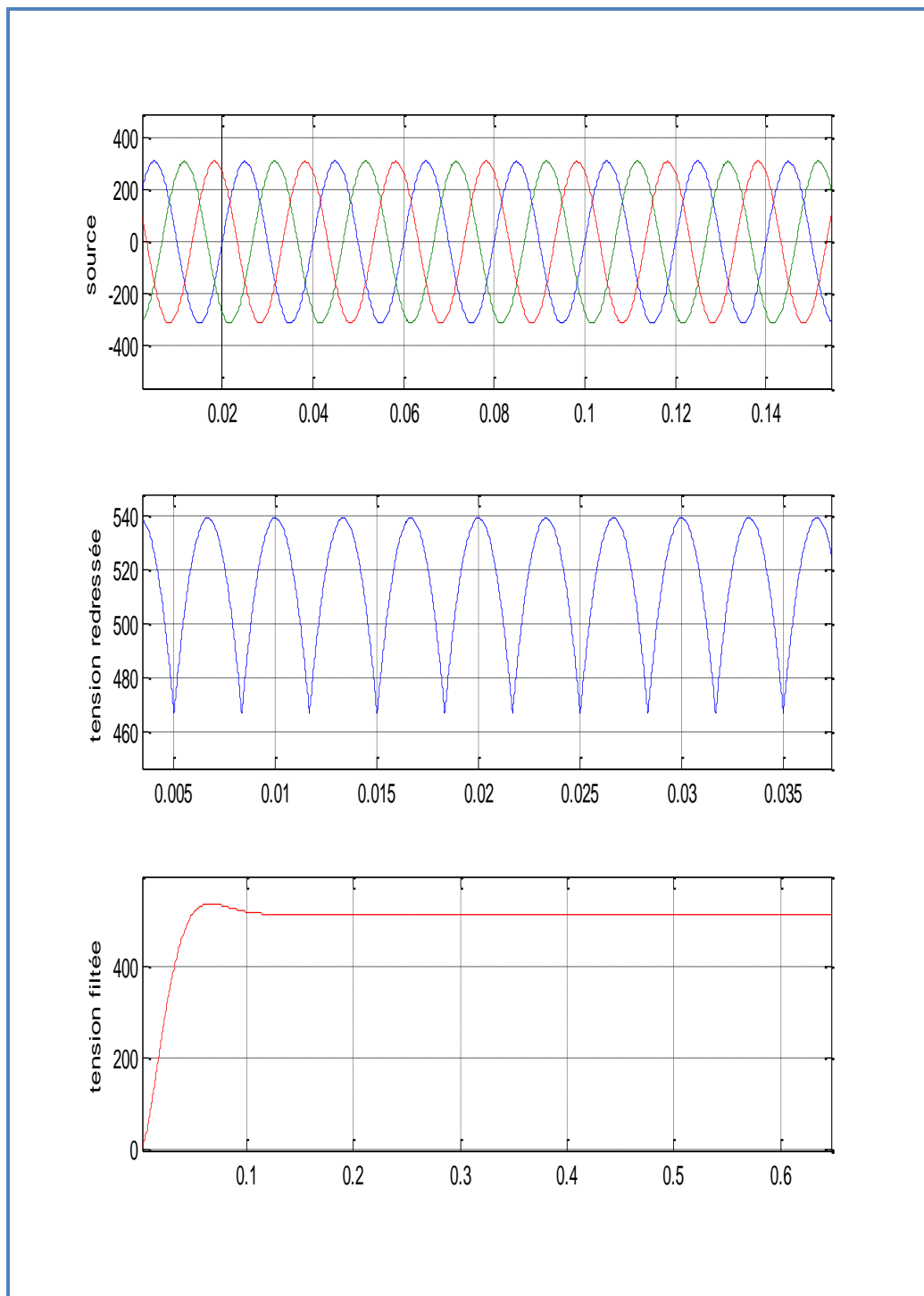


Figure III.13:les schémas de source, redressement et filtrage

❖ Interprétation :

Dans la source on remarque trois tensions sinusoïdales de même fréquence et même amplitude qui sont déphasées entre eux d'un tiers de tour. Après que les trois tensions passent par le redresseur leurs parties négatives vont éliminer et reste que les parties positives, le filtre va éliminer les harmoniques pour avoir un signal lisse.

- **Boucle ouvert sans charge :**

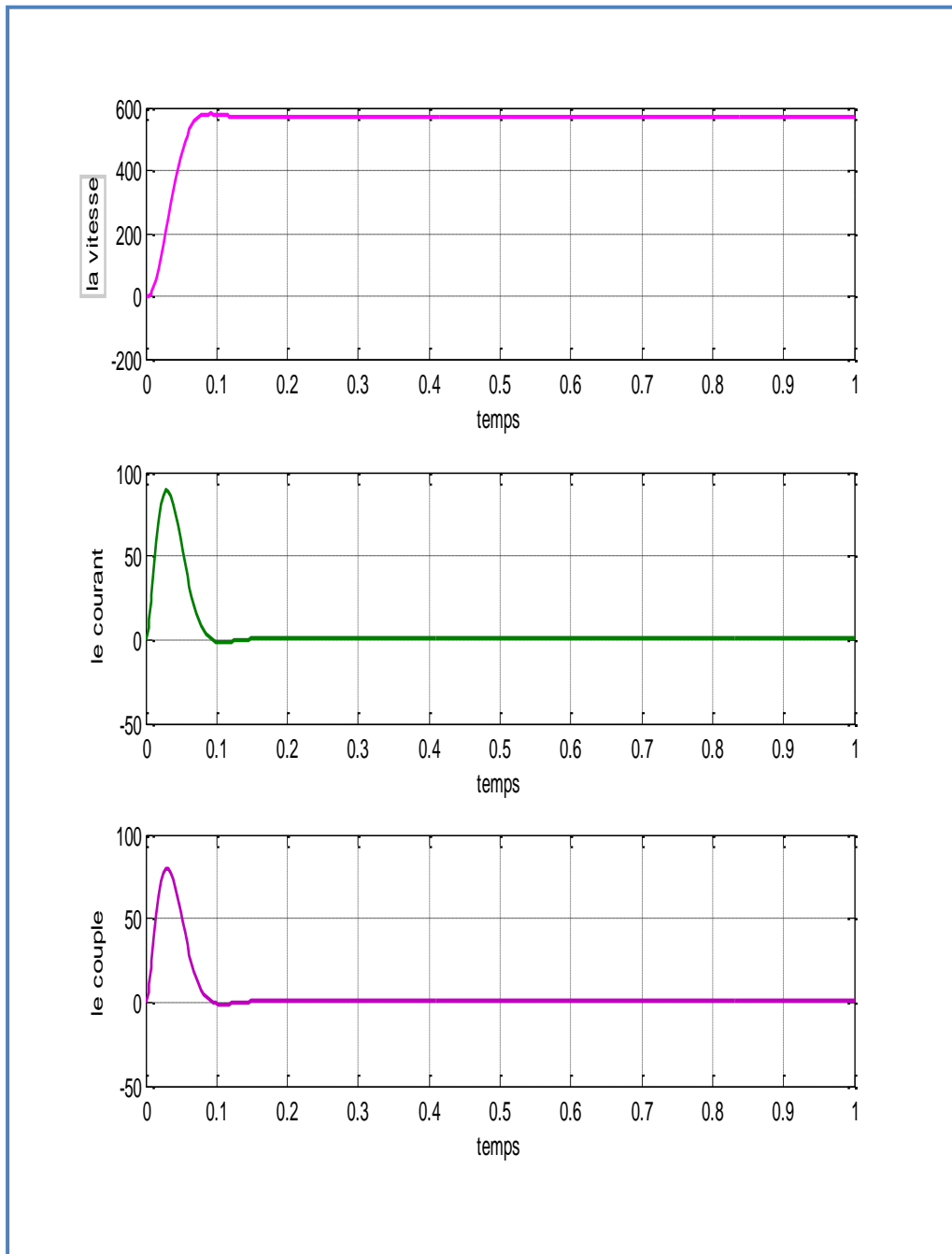


Figure III.14: vitesse, courant, couple BO sans charge avec $\alpha=0.3$

❖ **Interprétation :**

Dans la simulation du MCC sans charge ($C_r=0\text{Nm}$) on remarque un dépassement dans la vitesse, le courant et le couple électromagnétique avec un temps de réponse (0.1s)

Le couple et le courant ont la même réponse parce que le couple est proportionnel du courant ($C_e=K_I$).

- **Boucle ouverte avec charge :**

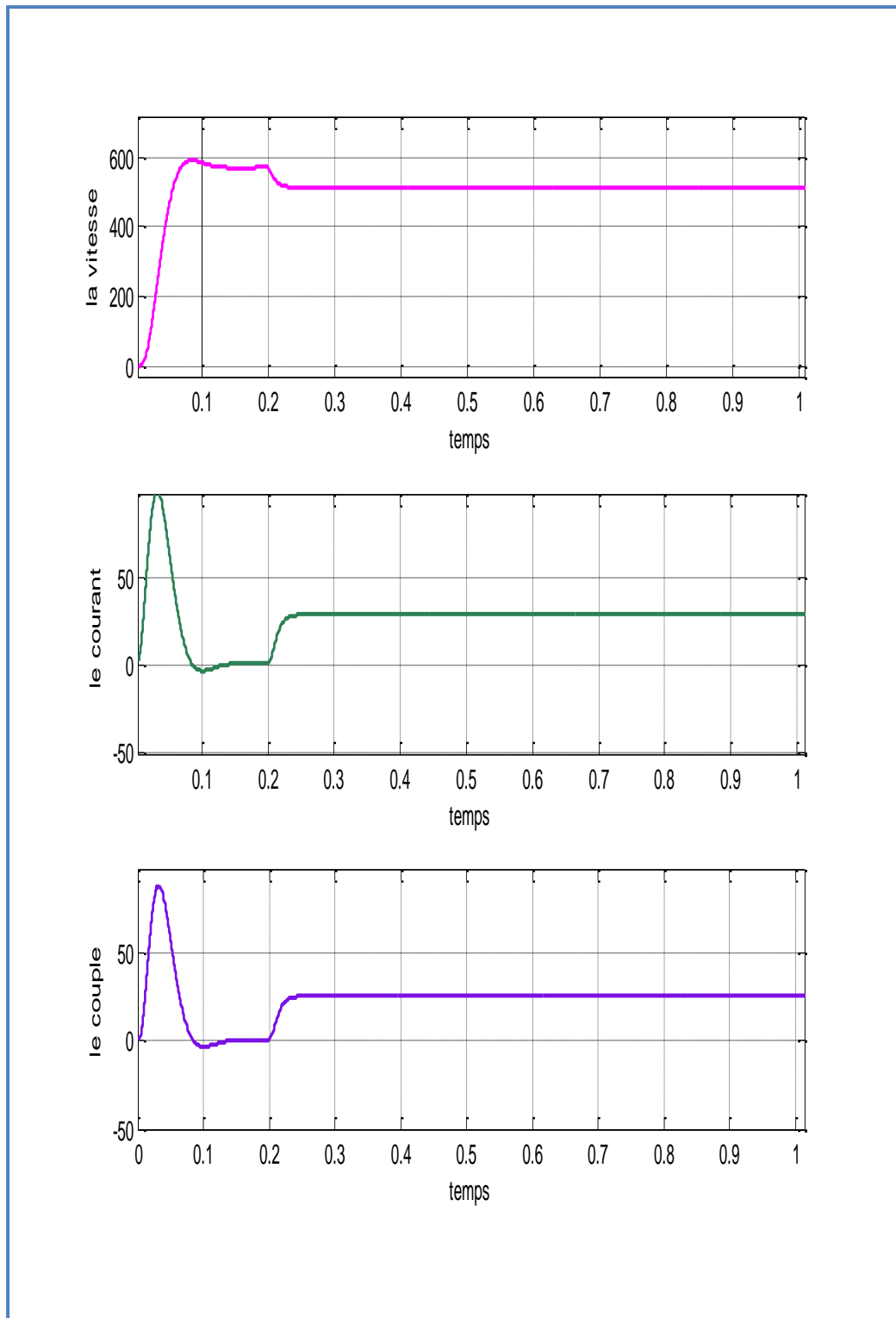


Figure III.15: vitesse, courant, couple BO avec charge avec $\alpha=0.3$

❖ Interprétation :

Dans la simulation du MCC avec charge on remarque la vitesse diminue lorsque on applique la charge, le couple électromagnétique prend la valeur de la charge, la réponse de couple et le courant est la même.

• Boucle fermée sans charge :

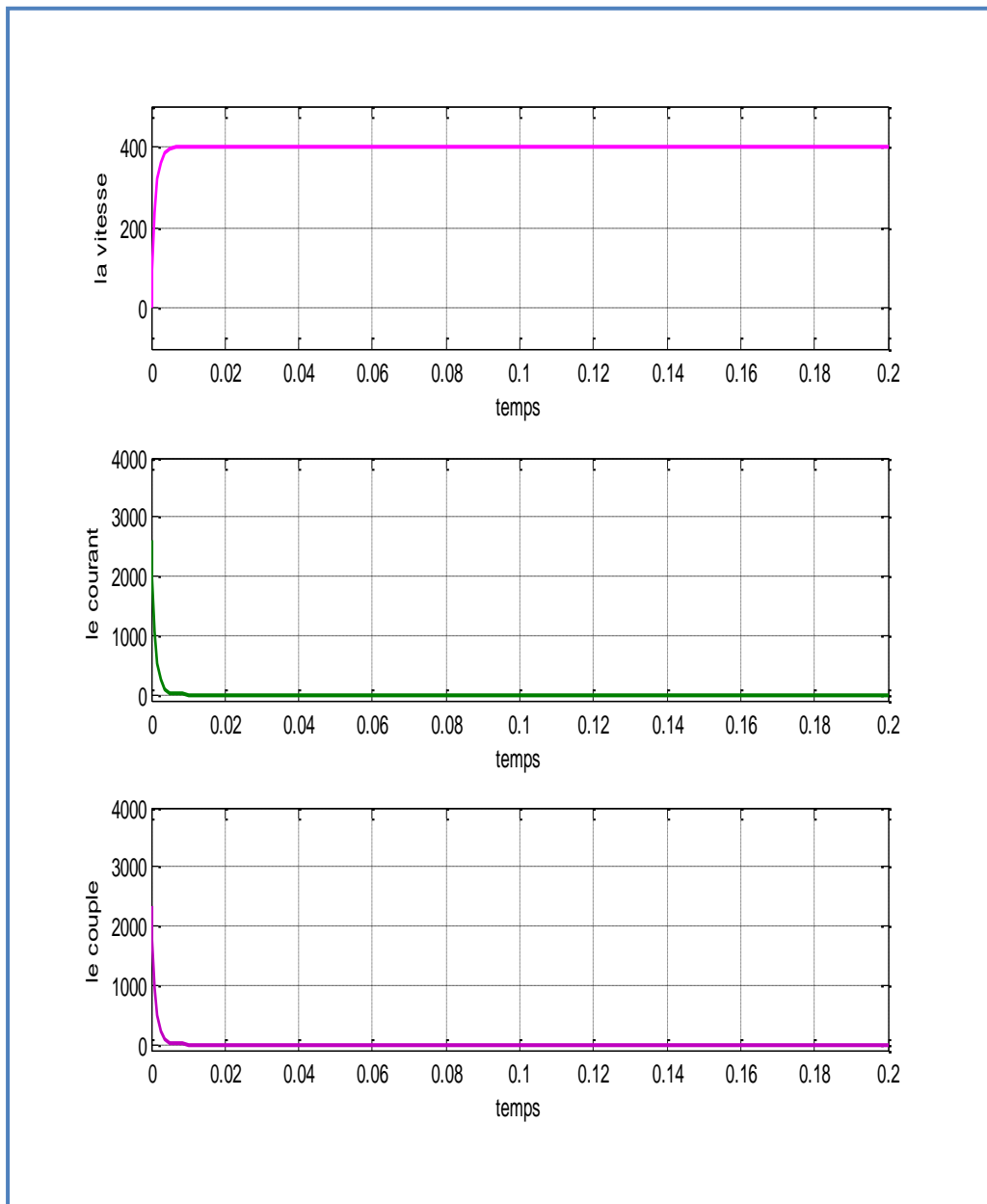


Figure III.16:vitesse, courant, couple BF sans charge avec $\alpha=0.6$

❖ Interprétation :

Dans la simulation du MCC en boucle fermée sans charge on remarque une amélioration dans le temps de réponse dans la vitesse et un dépassement négligeable, avec une erreur

statique nulle. Dans le couple électromagnétique et le courant on remarque que le dépassement augmente mais avec un temps de réponse très court (0.0001s). Un régulateur proportionnel (diminution de l'erreur statique et amélioration des performances), intégrale (erreur statique nulle).

- **Boucle fermée avec charge :**

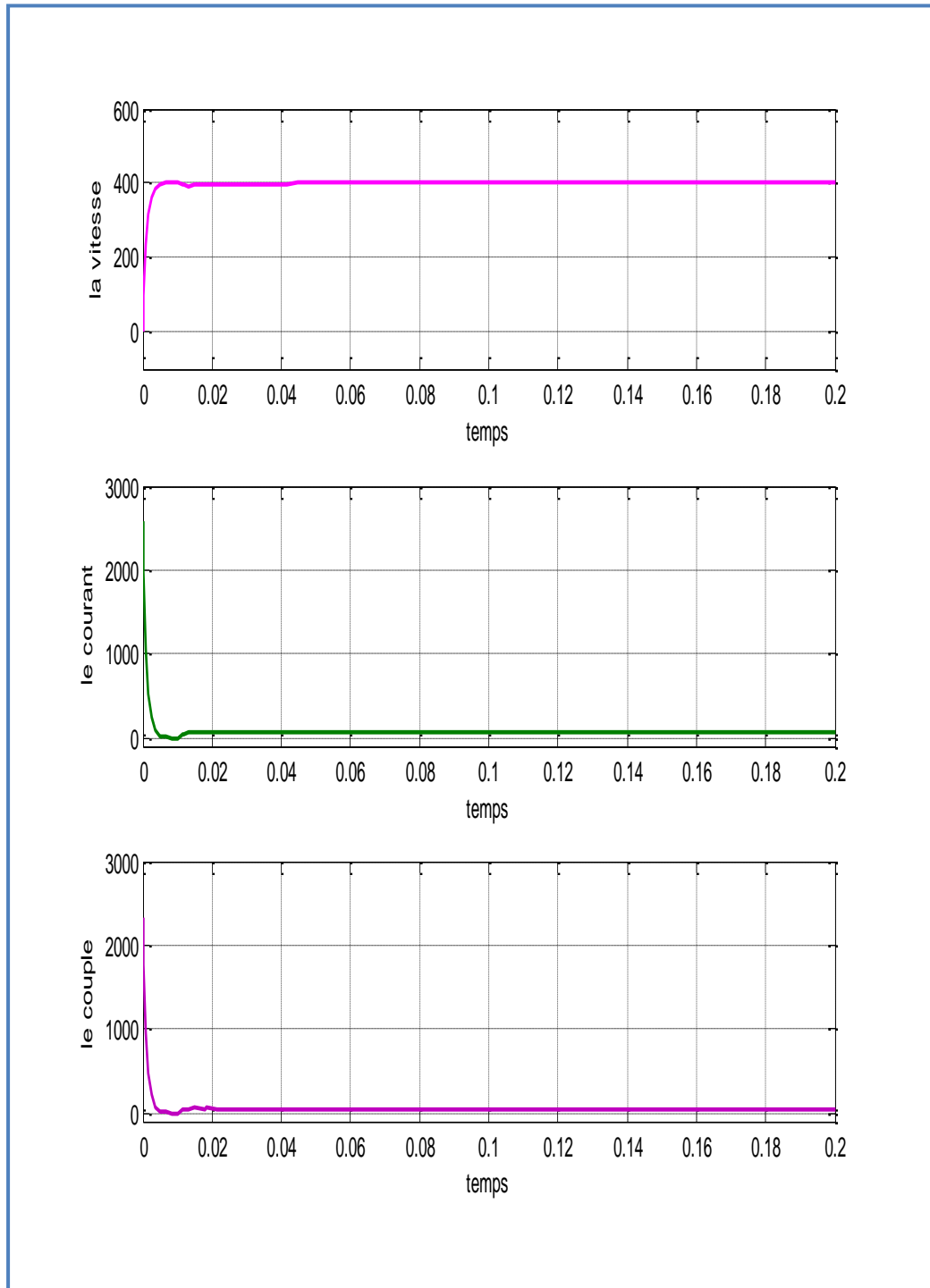


Figure III.17:vitesse, courant, couple BF avec charge avec $\alpha=0.6$

❖ Interprétation:

Lorsque on applique la charge on remarque une petite augmentation dans la valeur de vitesse, du couple et du courant mais après le système se stabilise très vite.

III.11-Modélisation sous interface graphique sous GUI-MATLAB :

On a implémenté la régulation de la vitesse du moteur à courant continu dans une interface graphique sous GUI-MATLAB présentée dans la figure suivante.

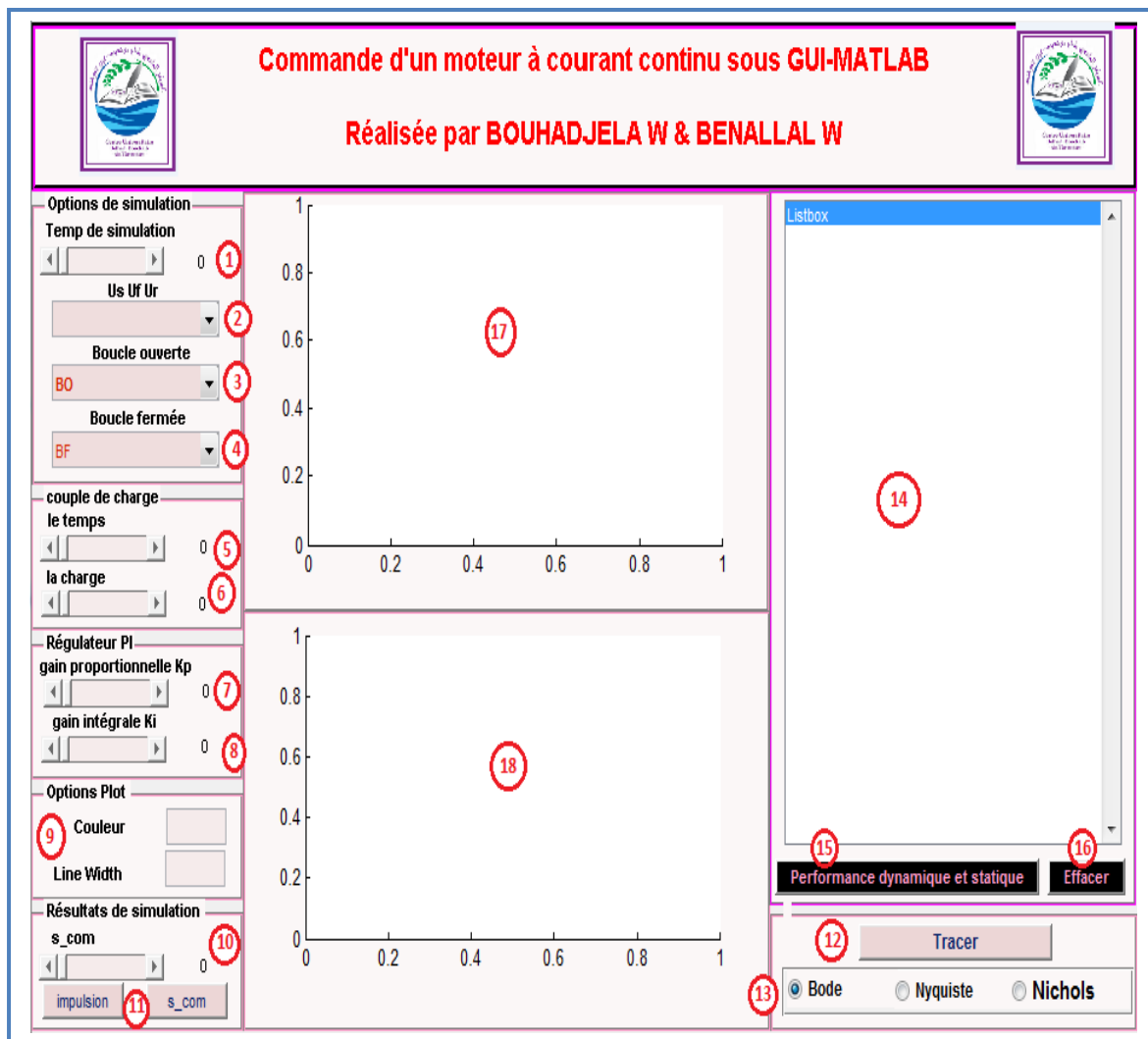


Figure III.18:interface graphique sous GUI-Matlab

1- Slider pour varier le temps de simulation.

2-Popupmnu pour choisir les tensions à tracer (tension de la source, tension redressée et tension filtrée).

3-Popupmnu pour tracer les graphes de la boucle ouverte (vitesse, couple, courant et comparaison entre la vitesse et la consigne).

4-Popupmnu pour tracer les graphes de la boucle fermée (vitesse, couple, courant et comparaison entre la vitesse et la consigne).

5- Slider pour varier le temps d'application du couple de charge.

6- Slider pour varier la valeur du couple de charge.

7-Slider pour varier la valeur de gain proportionnelle (K_p).

8- Slider pour varier la valeur de gain intégrale (K_i).

9- Deux édits pour la modification :

La première pour la couleur et la deuxième pour l'épaisseur.

10-Slider pour changer la valeur du signal de commande.

11- deux Pushbuttons pour tracer le signale de commande et l'impulsion.

12- Pushbutton pour tracer les diagrammes de Boode, Nyquiste et Nichols.

13- Des radios buttons pour choisir les diagrammes à tracer :

- Radio button1 pour tracer Boode.
- Radio button2 pour tracer Nyquiste.
- Radio button3 pour tracer Nichols.

14- List box pour afficher les performances dynamiques et statiques qui varient avec chaque variation des gains de régulateur (K_p et K_i).

15- Pushbutton pour donner l'ordre à la List box d'afficher les performances.

16- Pushbutton pour effacer le contenu de la List box.

17- Axes1 pour tracer les graphes des Popupmnus.

18-Axes2 pour tracer les diagrammes, le signale de commande et l'impulsion.

III.12-Quelques exemples de fonctionnement de notre interface graphique :

• **Premier exemple :**

Lorsqu'on clique sur on clique sur le Popupmenu de la boucle ouverte, il s'affiche les quatre choix de traçage. On valide notre choix de la comparaison entre la consigne et la vitesse avec la possibilité de varier le temps et la valeur du couple résistant et de choisir la couleur et l'épaisseur. Puis on varie le temps de simulation par le slider, le graphe va s'afficher dans axes1.

On sélectionne le diagramme de Boode par le radio button1 puis on clique sur tracer, le diagramme va s'afficher dans axes2.

Pour les performances dynamiques et statiques on appuie sur Pushbutton ' performance dynamique et statique'.

La figure suivante présente notre exemple.

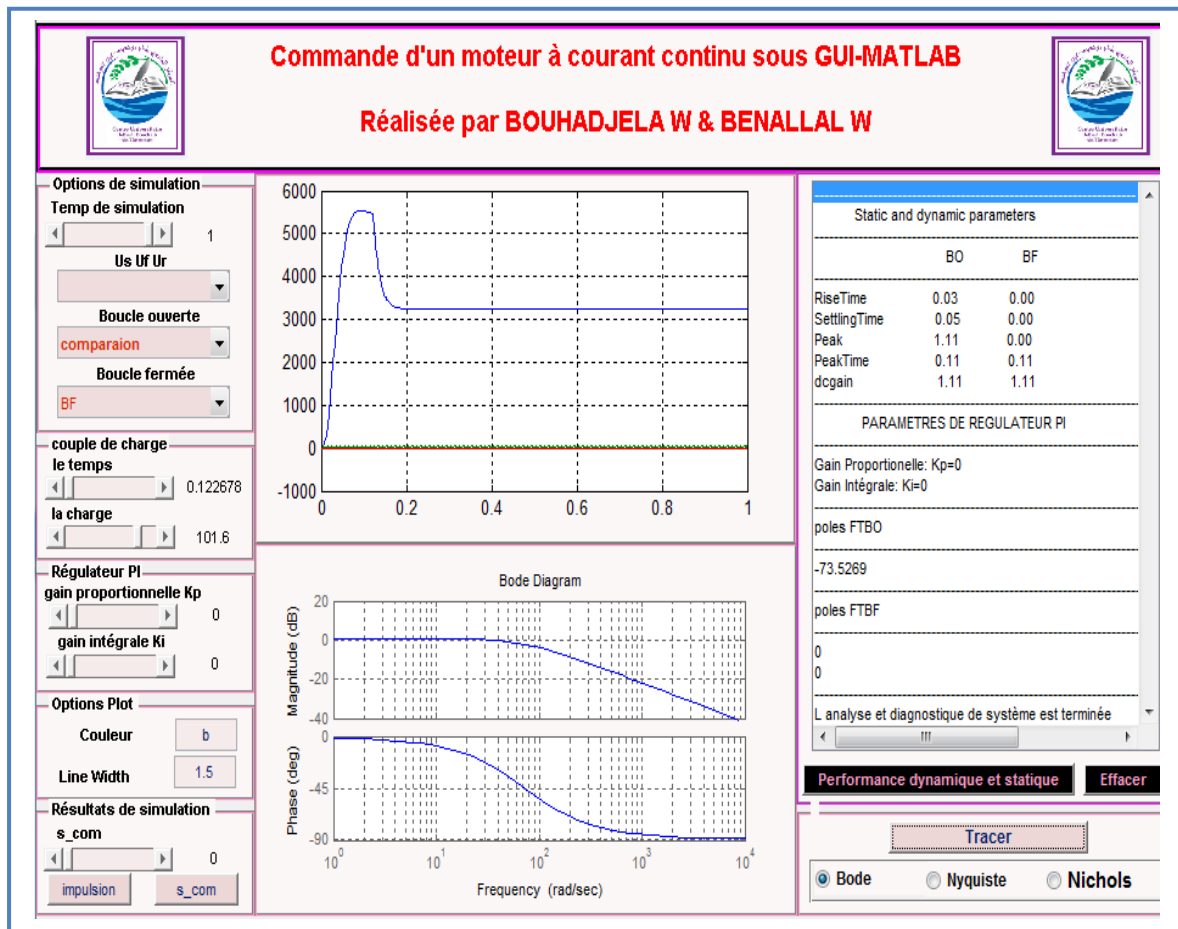


Figure III.19:interface graphique du premier exemple

• Deuxième exemple :

Lorsqu'on clique sur le Popupmenu de la boucle fermée, il s'affiche les quatre choix de traçage. On valide notre choix de la comparaison entre la consigne et la vitesse avec la possibilité de varier le temps et la valeur du couple résistant et de choisir la couleur et l'épaisseur. Puis on varie le temps de simulation par le slider, le graphe va s'afficher dans axes1.

On sélectionne le diagramme de Nichols par le radio button3 puis on clique sur tracer, le diagramme va s'afficher dans axes2.

Pour les performances dynamiques et statiques on appuie sur Pushbutton ' performance dynamique et statique'. Les performances de la boucle fermée plus les performances de la boucle ouverte et les pôles vont s'afficher dans la List box.

La figure suivante présente notre exemple.

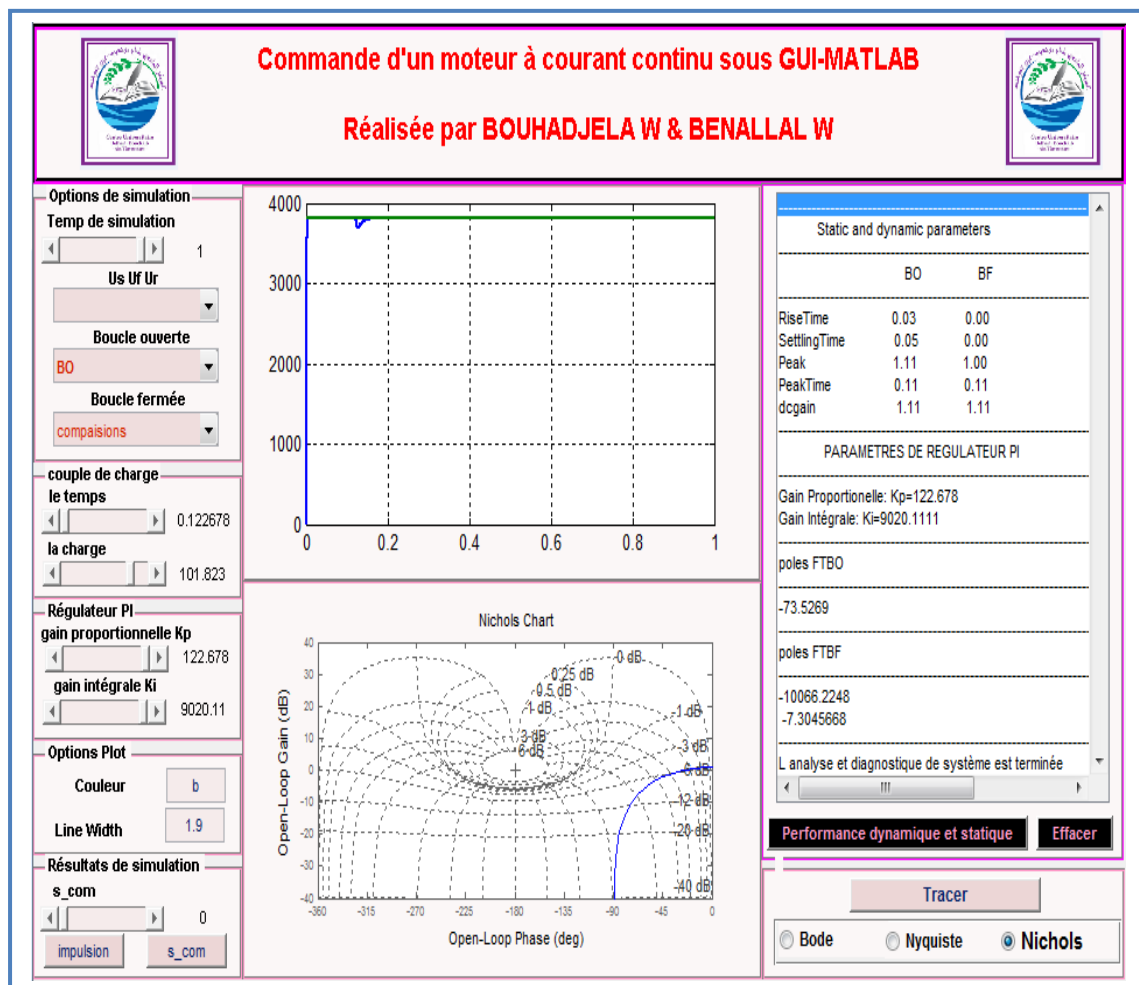


Figure III.20: interface graphique du deuxième exemple

• Troisième exemple :

Les mêmes étapes de l'exemple précédent sauf qu'on va varier les paramètres de régulateur K_p et K_i avec le changement de l'épaisseur, le temps de simulation et la valeur de couple résistant, le nouveau graphe va s'afficher dans axes1.

Le diagramme de Nyquist va se tracer dans axes2 après le clique dans radio button2 'Nyquist' puis sur le Pushbutton 'tracer'.

On clique sur le bouton 'Performances dynamique et statique', les nouvelles performances du système vont afficher dans la listbox.

La figure suivante présente notre exemple

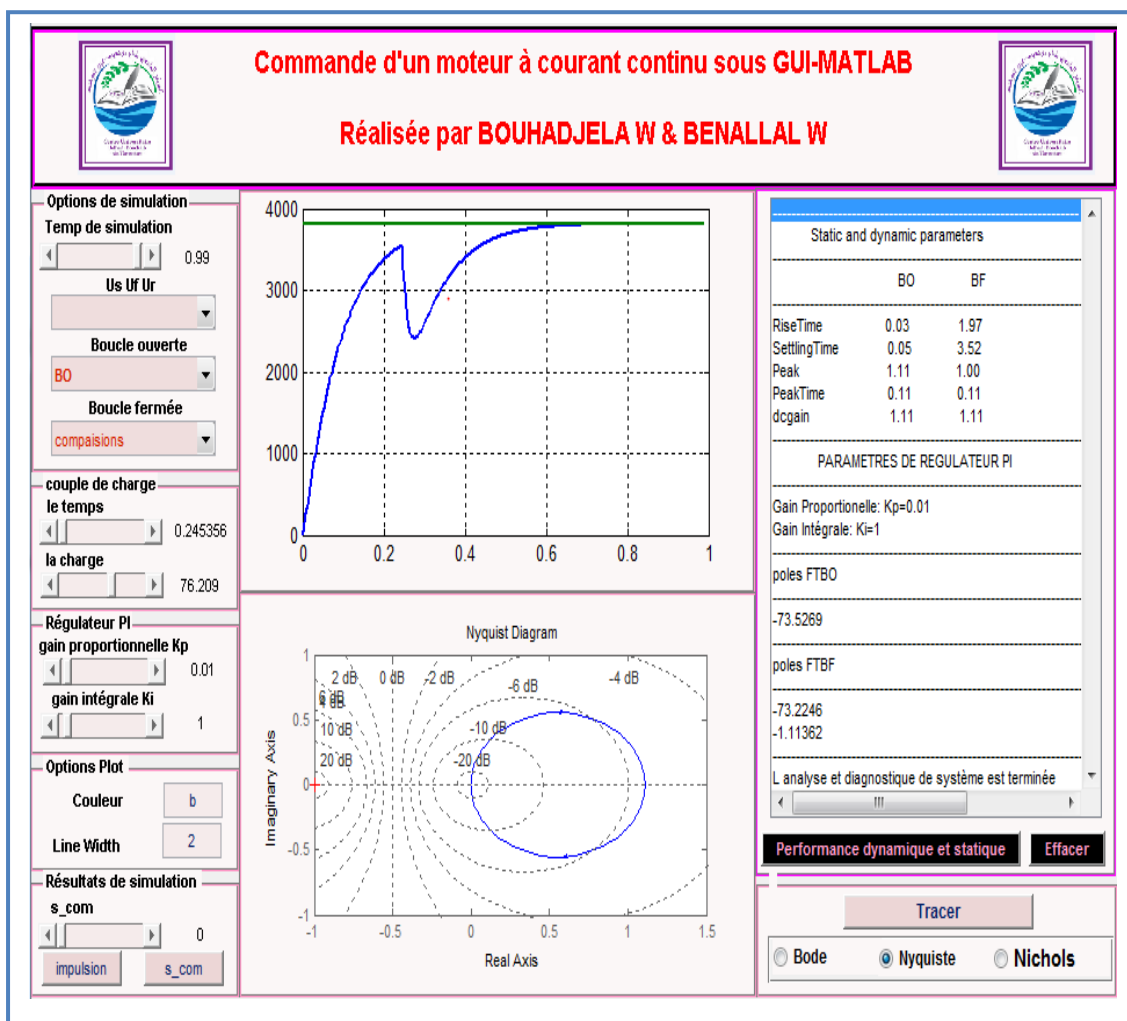


Figure III.21: interface graphique du troisième exemple

• Quatrième exemple :

On choisit dans Popupmenu1 la tension de la tension de la source (Us) et on change le temps de simulation, le graphe va s'afficher dans axes1.

On clique sur les Pushbuttons 'impulsion' et 's-com' les deux graphes vont s'afficher dans axes2 au même temps.

Puis on clique sur effacer pour vider la listbox.

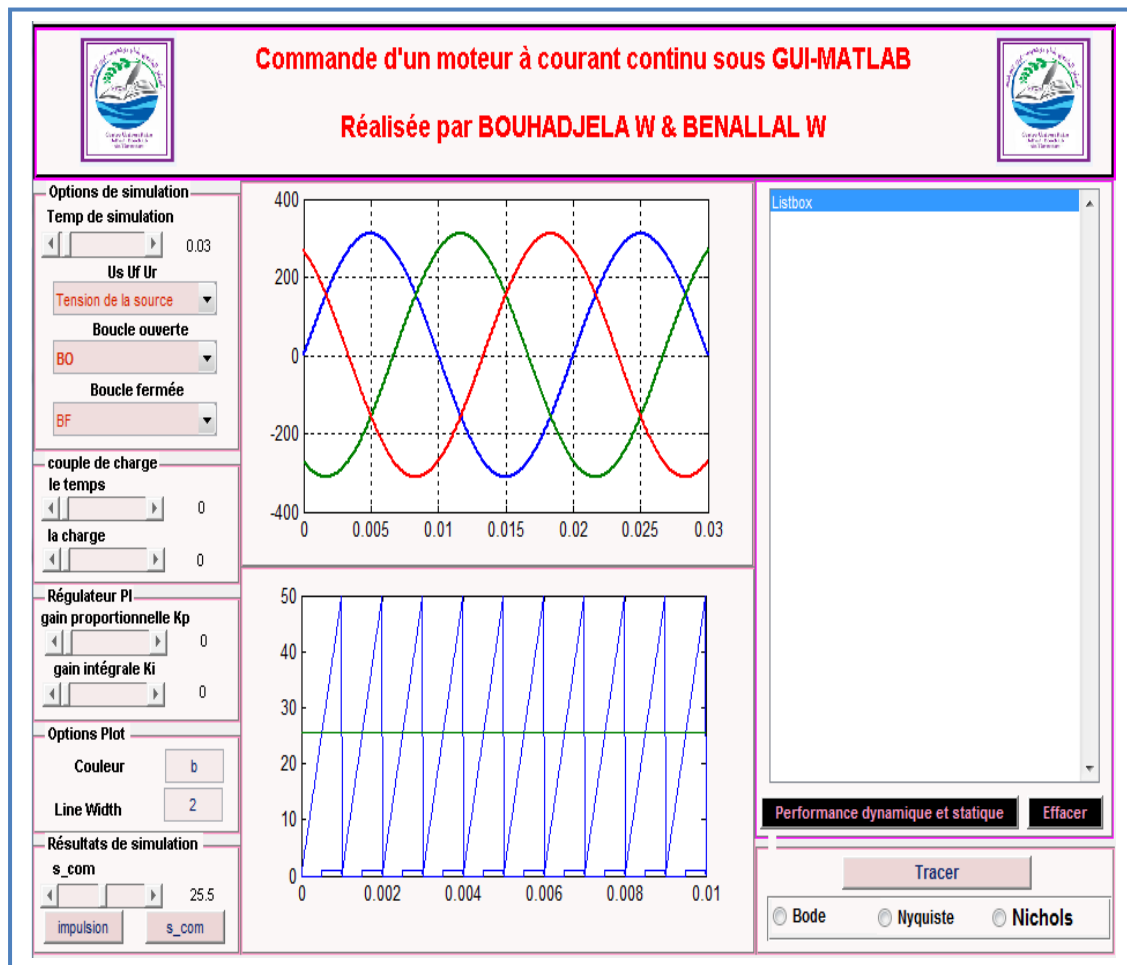


Figure III.22: interface graphique du quatrième exemple

III.13-Conclusion :

Dans ce chapitre on a fait la modélisation de système étudié (source, filtre, hacheur, moteur à courant continu) et on a étudié la commande de la vitesse du MCC à l'aide d'un régulateur proportionnel intégrale (PI).

On a utilisé la méthode de la régulation idéale à fin d'obtenir les réponses spécifiées dans le cahier de charge.

Puis on a créé une interface graphique sous GUI-MATLAB de cette régulation qui regroupe tous les boutons nécessaires pour faciliter les tâches de l'exploitation et le changement de données.

Conclusion Générale

Dans ce projet de fin d'étude on a fait une étude de régulation de la vitesse d'un moteur à courant continu avec un régulateur classique de type proportionnelle intégrale (PI) implémenté sous interface graphique sous GUI-MATLAB.

Plusieurs méthodes de régulation de la vitesse existent, parmi ces méthodes de régulations on a choisis la régulation idéale pour avoir le meilleur compromis en fonction des exigences du cahier de charge.

Les résultats obtenus ont montré que le moteur à courant continu est facile à commander par le régulateur PI avec la méthode idéale et on est arrivé aux exigences de cahier de charge voulus.

Les GUI (Graphical User Interface) sont très utilisées dans le domaine de l'industrie, c'est la technologie du moment qui facilite la relation entre l'homme et la machine. Pour cela on a implémenté notre régulation de vitesse du moteur à courant continu sous interface graphique sur MATLAB.

Après la création de notre interface de régulation, on a constaté que la régulation devient plus facile à étudier parce qu'elle nous permet de commander le système avec les paramètres de régulateur qu'on veut, aussi de faciliter la visualisation des résultats en boucle ouverte ainsi que la boucle fermé et le calcul des paramètres dynamiques et statiques du système étudié.

BIBLIOGRAPHIE

- [1] Boukerche Ghania « Etude et Synthèse d'un Contrôleur PI et Application ». *MEMOIRE DE MASTER UNIVERSITÉ BADJI MOKHTAR-ANNABA*, juin 2017.
- [2] colbertserv.lyceecolbert-tg.org. [En ligne] .
- [3] Benchiha Amira Saadia & Diabate fode « Cammande prédictive du moteur à courant continu » Centre Universitaire Belhadj Bouchaib d'Ain-Temouchent 2016.
- [4] Cour Dr Omar Fizazi « université Djilali liabes de sidi bel abasse ».
- [5] http://www.lyc-rostand-mantes.ac-versailles.fr/IMG/pdf/05032201_LA_VARIATION_DE_VITESSE.pdf.
- [6] Ben Kherif Oussama « Etude Comparative en Simulation entre un Régulateur PID Classique et un Régulateur PID Intelligent » mémoire UNIVERSITE MOHAMED BOUDIAF, M'SILA 2016.
- [7] https://jalelghabi.weebly.com/uploads/9/6/9/7/9697226/td_co_version1.pdf.
- [8] Cours système robotique « université paris est master 2 ».
- [9] Franck CASARANG « Commande robuste des systèmes plats application la commande d'une machine synchrone »
- [10] <https://france-ecologieindustrielle.fr/interfaces-graphiques-industriel>.
- [11] <https://cours-informatique-gratuit.fr/dictionnaire/interface-graphique>.
- [12] <https://briot-jerome.developpez.com/matlab/tutoriels/introduction-programmation-interfaces-graphiques/#LII-A>.
- [13] <https://matlabpourtous.com/matlab/les-interfaces-graphiques-sous-matlab/utilisation-de-guide>.
- [14] Dr. Youcef Djeriri & Abdelkader Meroufel « Techniques de Commande électrique » université Djilali liabes de sidi bila base.
- [15] Cour Dr Ghoraf Djamel Elddine « Ecole supérieure polytechnique Oran ».