



Université ABBES LAGHROUR Khenchela
Faculté des Sciences et de la Technologie
Département de Génie Industriel
جامعة عباس لغرور خنشلة
كلية العلوم والتكنولوجيا
قسم الهندسة الصناعية



N° Série :.....

Mémoire de fin d'étude

Pour l'obtention du diplôme de Master

Filière : Electrotechnique

Spécialité : Commandes Electriques

Présenté par

BIBI NESREDDINE

THEME

**IMPLEMENTATION EN TEMPS REEL DE LA COMMANDE
D'UNE MACHINE A COURANT CONTINU (MCC) PAR DES
TECHNIQUES D'INTELLIGENCE ARTIFICIELLE.**

Soutenu le 09/09/2019 devant la commission d'examen composée de :

Dr. LAGGOUN Louanasse	MCB	à l'Université de Khenchela	Président
Dr. BOUTABBA Tarek	MCB	à l'Université de Khenchela	Encadreur
Dr. KHEMIS Abderrahmane	MCB	à l'Université de Khenchela	Examineur

Promotion 2018/2019

Confession et merci :

Je remercie tous ceux qui ont contribué à la réalisation de ce mémorandum de près ou de loin et je salue leur travail et leur souhaite du succès.

Dédicaces

Je dédie ce mémoire à :

Mes parents Toutes Utérus ,Toutes Mes amis

et tous ceux qui m'ont donné un coup de

main .

Liste des figures

Figure I.1	Construction d'un moteur à courant continu.....	8
Figure I.2	Schéma équivalent d'un MCC à excitation sépare	9
Figure I.3	Schéma bloc du Modèle de moteur électrique fermé.....	10
Figure I.4	Réponse de la MCC en boucle ouverte.....	11
Figure I.5	Schéma bloc de l'asservissement.....	12
Figure I.6	schéma bloc de la boucle de courant.....	15
Figure I.7	Schéma bloc de la boucle de courant identifié.....	15
Figure I.8	schéma bloc de la boucle de vitesse.....	17
Figure I.9	schéma bloc de la boucle de la vitesse identifié.....	17
Figure I.10	Détermination du temps de réponse à 2%.....	19
Figure I.11	Schéma bloc de l'asservissement des deux boucles (vitesse et courant).....	20
Figure I.12	régulation PI de la vitesse et de courant	21
Figure I.13	régulation PI de la vitesse et de courant avec filtre.....	21
Figure II.1	Différentes méthodes de l'intelligence computationnelle.....	24
Figure II.2	croisement en un point.....	29
Figure II.3	croisement en deux points.....	29
Figure II.4	Principe de la mutation Binaire.....	29
Figure II.5	Organigramme l'algorithme génétique.....	30
Figure II.6	Le schéma bloc de la boucle de courant.....	32
Figure II.7	interface GATOOL pour l'assertivement de courant	33
Figure II.8	réponse de courant à partir des paramètres régulateur PIAG.....	34
Figure II.9	Le schéma bloc de la boucle de vitesse.....	34

Figure II.10	interface GATool pour l'asservissement de la vitesse.....	35
Figure II.11	réponse de vitesse à partir des paramètres régulateurs PIAG	35
Figure II.12	Réponses en vitesse et en courant optimisées par l'AG.....	36
Figure II.13	Système à commande floue.....	39
Figure II.14	Structure interne d'un système floue.....	40
Figure II.15	La méthode d'inférence min-max.....	42
Figure II.16	La méthode d'inférence max-produit.....	42
Figure II.17	La méthode d'inférence somme-produit.....	43
Figure II.18	exemple sur la méthode du maximum	44
Figure II.19	Définition de la structure globale (entrée, sortie, type d'opérateurs).....	47
Figure II.20	Définition des fonctions d'appartenances pour 'e', 'de/dt' et 'u'.....	48
Figure II.21	Écran de saisie des règles.....	48
Figure II.22	Schéma bloc d'un asservissement de vitesse par RLF.....	49
Figure II.23	Réponses en vitesse et en courant régulées par logique floue.....	50
Figure II.24	allures de la vitesse et du courant (Comparaison entre les différentes méthodes PI, BLF, AG).....	51
Figure II.25	allures de la vitesse au moment d'application perturbation.....	51
Figure II.26	allures de la vitesse et courant filtre.....	52
Figure III.1	Essais au ralentissement.....	56
Figure III.2	Schéma bloc du Modèle de moteur électrique en vitesse.....	58
Figure III.3	Allure de la vitesse simulée.....	59
Figure III.4	Forme de signal MLI avec différent rapport cyclique.....	60
Figure III.5	La structure de convertisseur BUCK.....	61
Figure III.6	Formes d'ondes du convertisseur Buck (courants à gauche, tension à droite)..	61
Figure III.7	Oscillogramme de la tension aux bornes de la charge.....	62
Figure III.8	Description de la Carte Arduino "Uno"	63

Figure III.9	Description du signal PWM.....	64
Figure III.10	principe de fonctionnement d'un optocoupleur.....	65
Figure III.11	schéma fonctionnelle de l'optocoupleur.....	65
Figure III.12	Schéma on Proteus de système en boucle ouverte (hacheur série)	67
figure III.13	différant valeur de rapport cyclique.....	68
Figure III.14	Circuit interne du module.....	69
Figure III.15	Les deux signaux issus de l'encodeur optique.....	70
Figure III.16	Les Blocs d'ArduinoIO nécessaires pour la commande	71
Figure III.17	acquisition des impulsions sous Arduino IO Libra.....	72
Figure III.18	Modèle Simulink d'asservissement de vitesse.....	72
Figure III.19	Réponse de la vitesse en charge.....	73

Liste des Tableaux

Tableaux I.1	Caractéristiques de la MCC.....	10.
Tableaux I.2	Influence des paramètres K_p , K_i et K_d sur la réponse du système	14
Tableaux II.1	Règles à cinq fonctions d'appartenance.....	46
Tableaux II.2	Résumé des résultats des simulations.....	52
Tableaux III.1	: Paramètres identifiées.....	57

Liste des abréviations

MCC : Machine à Courant Continu

PID : Proportionnel intégral Dérive

PI : Proportionnel intégral

IA : intelligence artificielle

AG : Algorithmes génétiques

RLF : Régulateur logique floue

IGBT: Insulated Gate Bipolar Transistor

MLI : Modulation de Largeur d 'Impulsion

Table des Matières	1
Introduction Générale	5
Chapitre I : Modélisation et asservissement de la vitesse d'un MCC	
I.1	Introduction..... 7
I.2	La machine à courant continu..... 7
I.3	Constitution du moteur à courant continu 7
I.4	Modélisation du moteur à courant continue 8
I.5	Simulation en boucle ouverte du MCC..... 10
I.6	Asservissement du MCC avec un régulateur PI..... 11
I.7	Description des régulateur PID 12
I.7.1	Action Proportionnelle 13
I.7.2	Action intégral 13
I.7.3	Action Dérivée 13
I.8	L'effet des paramètres d'un contrôleur PID sur la réponse 13
I.8.1	Proportionnelle..... 13
I.8.2	Intégral 14
I.8.3	Dérive..... 14
I.9	Synthèse, calcul et choix du correcteur..... 15
I.9.1	Etude de la boucle du courant..... 15
I.9.1.1	Identification des blocs..... 15
I.9.1.2	Asservissement du courant..... 16
I.9.2	Etude de la boucle de la vitesse..... 17
I.9.2.1	Identification des blocs..... 17
I.9.2.2	Asservissement de la vitesse 18
I.9.3	Etude des deux boucles (courant et vitesse) 20
I.10	Conclusion 22
 Chapitre II : Commande MCC par l'intelligence artificielle	
II.1	Introduction..... 23
II.2	Intelligence artificielle 23
II.3	Techniques de Calcul intelligent 23

II.4	Algorithmes génétiques	24
II .4.1	Principe des Algorithmes génétiques	25
II.4.2	Conception d'un Algorithme génétique (procédé d'implantation desAG)	25
II.4.3	Paramètre des Algorithmes génétiques	26
II.4.4	Opérateurs des Algorithmes génétiques.....	26
II.4.4.1	Le codage.....	26
II.4.4.2	Population initiale.....	27
II.4.4.3	La Sélection	27
II.4.4.4	Reproduction avec des opérateurs de croisement et de mutation	28
II.4.4.4.1	Opérateur de Croisement	28
II.4.4.4.2	Opérateur de Mutation	29
II.4.5	Critère d'arrêt.....	30
II.4.6	Organigramme de la procédure l'algorithme génétique.....	30
II.4.7	Développement d'un AG pour le réglage PI.....	31
II.4.7.1	Écrire la fonction objective pour un algorithme génétique	31
II.4.7.2	Indices de performance.....	31
II.4.8	Application des AG à l'optimisation du régulateur PI pour la commande d'une machine à courant continu.....	32
II.4.8.1	Boucle interne de régulation du courant.....	32
II.4.8.2	Boucle externe de régulation de la vitesse.....	34
II.4.8.3	Application des paramètres du régulateur PI optimisés sur le modèle de la MCC	36
II.4.8.3.1	Résultats de simulation.....	36
II.4.8.3.2	conclusion.....	37
II.5	la logique floue.....	37
II.5.1	Principe de la logique floue	38
II.5.2	Le régulateur flou.....	39
II.5. 3	configuration générale d'un Régulateur flou.....	39
II.5.3.1	La fuzzification.....	40
II.5.3.2	Base de connaissance.....	40
II.5.3.3	Mécanisme d'inférence.....	41
II.5.3.3.1	Méthode Min-Max.....	41
II.5.3.3.2	Méthode d'inférence max-produit.....	42

II.5.3.3.3	Méthode d'inférence somme-produit.....	43
II.5.3.4	Défuzzification	43
II.5.3.4.1	méthode du maximum	44
II.5.3.4.2	Méthode du centre de gravité.....	44
II.5.4	Etapas de Conception d'un contrôleur flou pour le réglage de la vitesse	44
II.5.4.1	Choix des variables d'entrées et de sorties	44
II.5.4.2	Normalisation des variables d'entrée et de sortie	45
II.5.4.3	Interface de fuzzification	45
II.5.4.4	Inférence.....	45
II.5.4.5	Traitement numérique de l'inférence.....	46
II.5.4.6	Défuzzification.....	46
II.5.5	Application de la commande floue sur le moteur courant continue.....	46
II.5.5.1	Conception de contrôleur.....	47
II.5.5.2	Schema bloc de simulation de la MCC avec le controleur floue.....	49
II.5.5.3	Résultats de simulation et discussion.....	49
II.5.5.4	Conclusion	50
II.6	Comparaison entre les différentes méthodes	51
II.7	Conclusion	5 3

Chapitre III : Réalisation pratique d'un asservissement de vitesse

III.1	Introduction.....	54
III.2	Identification des paramètres du moteur.....	54
III.2.1	Mesure des paramètres électriques.....	54
III.2.1.1	La résistance de l'induit R.....	54
III.2.1.2	L'inductance de l'induit L.....	54
III.2.1.3	La constante K	54
III.2.2	Mesure des paramètres mécaniques	55
III.2.2.1	Le coefficient de frottement f	55
III.2.2.2	Le moment d'inertie J	56
III.2.3	Modélisation du système	57
III.2.3.1	Fonction de transfert du moteur	57

III.2.3.2	Simulation du MCC)	58
III.3	Commande du moteur à courant continu en boucle ouvert.....	59
III.3.1	La Commande par MLI.....	59
III.3.2	Principe de la commande MLI.....	59
III.3.3	Alimentation par convertisseur électronique.....	60
III.3.3.1	Hacheur.....	60
III.3.3.2	Hacheur série	60
III.3.3.3	Principe de fonctionnement	61
III.3.3.4	Le rapport cyclique	62
III.3.3.5	Valeur moyenne de la tension aux bornes de la charge.....	62
III.3.3.6	Présentation de la maquette.....	62
III.4	Réalisation de la commande en boucle ouverte.....	63
III.4.1	Commande du moteur à courant continu avec la carte Arduino	63
III.4.1.1	la carte Arduino UNO	63
III.4.1.2	Présentation des sorties analogiques (mode PWM)	63
III.4.2	Bloc d'isolation	64
III.4.2.1	Principe d'isolation	64
III.4.2.2	Isolation par optocoupleur	65
III.4.2.3	Calcul des résistances d'entrée et de sortie de l'optocoupleur.....	66
III.4.3	Bloc de puissance (Hacheur série)	66
III.4.4	Forme du signal de commande de MLI	67
III.5	Commande du moteur à courant continu en boucle ferme	68
III.5.1	Présentation du capteur de vitesse	68
III.5.2.1	Principe de fonctionnement.....	69
III.5.3	L'interfaçage Arduino↔Matlab/Simulink.....	70
III.5.3.1	L'environnement Matlab/Simulink	70
III.5.3.2	. Introduction sur Arduino IO.....	71
III.5.3.3	ArduinoIO	71
III.5.3.4	Exploitation de la bibliothèque ArduinoIO sous Simulink.....	71
III.5.4	Exploitation de l'encodeur optique avec Arduino/Simulink.....	72
III.6	Implémentation de la commande sur Simulink.....	72
III.6.1	Implémentation de la commande sous Simulink.....	73

III.7	Résultats de la réalisation expérimentale en boucle ferme.....	73
III.8	Conclusion	74
	CONCLUSION Générale	75
	BIBLIOGRAPHIE.....	

Introduction générale :

La régulation est l'un des volets de l'automatique qui connaît un essor considérable vu son importance dans l'industrie.

Au tout début, on faisait appel à la régulation classique à cause de ses avantages comme sa simplicité à utiliser, sa fiabilité et son efficacité, les méthodes classiques de régulation basées sur le dimensionnement du régulateur en utilisant le modèle mathématique du système, mais souvent les qualités d'un régulateur classique restent limitées à cause de la complexité réelle des systèmes à commander (non linéarité) et plus d'inconvénients dans le milieu industriel.

Plusieurs applications industrielles nécessitent l'utilisation des méthodes modernes de contrôle, permettant d'avoir une réponse rapide et des performances dynamiques élevées. Parmi ces méthodes, on cite le réglage par la logique floue (RLF) et les algorithmes génétiques.

Les concepts de ces deux approches sont basés sur la perception globale qu'un être humain à **l'environnement qui l'avoisine pour la logique floue**, et pour les algorithmes génétiques, sur une tentative de modéliser le comportement humain afin de pouvoir explorer ses capacités, de ce fait, ces deux approches, qualifiées alors d'intelligentes, font preuve de leur efficacité dans divers domaines d'application (traitement d'image, vision par ordinateurs, reconnaissance de la parole et de caractères manuscrits).

. Par ailleurs, le domaine de la commande des systèmes, et plus particulièrement des entraînements électriques, s'avère un champ d'application intéressant. Le travail présenté dans ce mémoire consiste à étudier ces deux approches et à les mettre en œuvre pour la commande de la vitesse d'une machine à courant continu, et ce dans le but de comparer leurs performances avec celles utilisant des techniques conventionnelles.

Dans le cadre de notre travail, nous allons étudier un moteur à courant continu commandé par l'intelligence artificielle et réaliser pratiquement un asservissement de vitesse (maquette feedback) de la machine à courant continu. Ce mémoire est donc constitué d'une introduction générale, de trois chapitres et d'une conclusion générale.

Au fil du chapitre **un**, nous allons tout d'abord présenter la modélisation de la machine et puis déterminer la commande PI. Dans le **deuxième** chapitre, nous détaillerons les formalismes de base de la logique floue et des algorithmes génétiques et la méthodologie

Introduction générale :

adoptée pour la conception du système de commande basé sur ces deux approches. L'application de ces commandes et l'étude comparative des performances des systèmes étudiés seront présentées. Le **troisième** chapitre sera consacré à l'étude et à la réalisation pratique d'un hacheur de puissance (hacheur série) et d'une carte de commande (maquette feedback) à base d'Arduino et du logiciel Matlab en vue d'étudier le comportement et l'asservissement numérique d'un moteur à courant continu.

Enfin, ce présent travail sera clôturé par une conclusion générale résumant les différents résultats obtenus .

I.1 Introduction

Les actionneurs électriques sont présents dans les usines automatisées, dans les moyens de transport et dans de nombreux objets de notre vie quotidienne, ils doivent avoir un comportement intelligent, ils sont donc commandés. Les commandes sont basées sur une bonne connaissance des actionneurs, machines électriques et convertisseurs statiques. Cette connaissance est à la base des modèles, qui sont évidemment basés sur des lois physiques, mais aussi sur des hypothèses constructives qu'il faut savoir discuter (linéarité, premier harmonique, symétrie...).

La technologie moderne des systèmes d'entraînement exige de plus en plus un contrôle précis et continu de la vitesse, du couple et de la position, tout en garantissant la stabilité, la rapidité et le rendement le plus élevé possible.

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 peuvent comporter un réseau de contrôle dont le principal élément de contrôle est un module de contrôle PID [0].

I.2 La machine à courant continu

Les machines à courant continu sont des convertisseurs électromécaniques d'énergie : Soit ils convertissent l'énergie électrique absorbée en énergie mécanique lorsqu'ils sont capables de fournir une puissance mécanique suffisante pour démarrer puis entraîner une charge en mouvement, On dit alors qu'ils ont un fonctionnement en moteur. Soit ils convertissent l'énergie mécanique reçue en énergie électrique lorsqu'ils subissent l'action d'une charge entraînée, On dit alors qu'ils ont un fonctionnement en générateur [01].

I.3 Constitution d'un moteur à courant continu

Un moteur à courant continu comprend quatre parties principales :

- 1) Le Stator composé d'un circuit magnétique comprenant une couronne annulaire massive ou feuilleté à laquelle sont fixés les pôles saillants, ces pôles sont de type, régulièrement alternés :
 - Les pôles inducteurs portent un ou plusieurs enroulements d'excitation alimentés en courant continu.

- Les pôles auxiliaires ou commutation disposés entre les pôles inducteurs selon les axes inter polaires dits axes neutres, Ils servent à améliorer la commutation.
- 2) Le Rotor ou l'induit comporte un circuit magnétique feuilleté muni d'encoches régulièrement distribuées à la périphérie.
 - 3) Le Collecteur est claveté sur l'arbre de l'induit, sur lequel frottent les balais. Il se compose de lame de cuivre de section trapézoïdale, portent chacune une ailette de connexion à l'enroulement de l'induit et isolé par du mica.
 - 4) Les balais sont fixes, en charbon graphitique, appuient sur des lames diamétralement opposées du collecteur, ils sont immobiles, isolés électriquement du bâti de la machine.

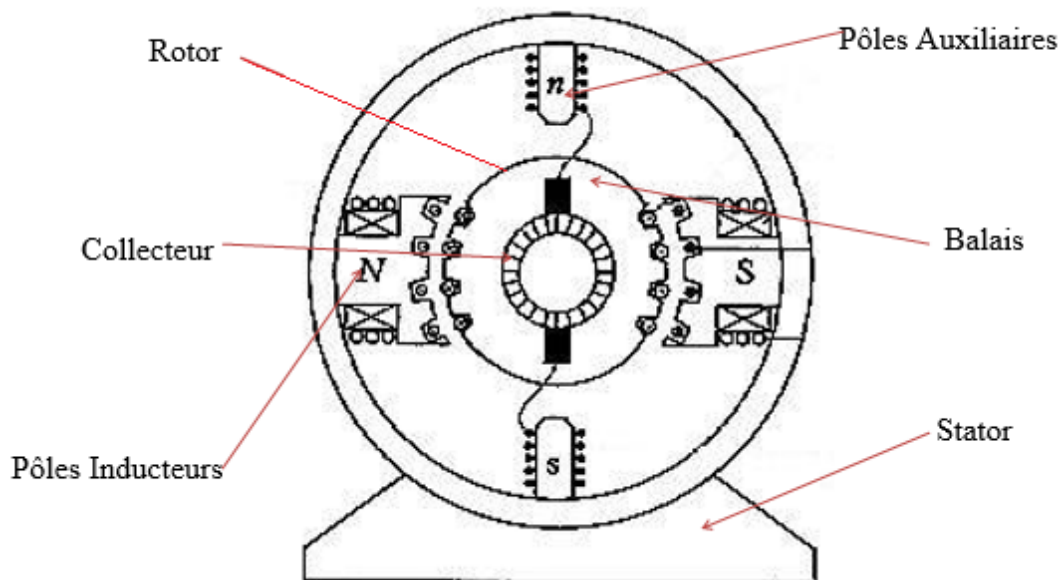


Figure I.1. Construction d'un moteur à courant continu

I.4 Modélisation du moteur à courant continu

Le moteur à courant continu peut être modélisé par le biais d'équations électriques, électromécaniques et mécaniques. Ces trois groupes d'équations nous permettront de mieux appréhender le moteur à courant continu dans son fonctionnement réel. Du côté électrique, nous pouvons dire que le moteur à courant continu se définit par un circuit induit et un circuit inducteur; l'induit de la MCC peut être vu comme une résistance et une inductance en série avec une source de tension commandée et proportionnelle à la vitesse [2]. Du côté mécanique, nous représentons la machine à courant continu par l'inertie de l'induit augmentée de celui de la charge entraînée. La Figure I.2 montre le schéma équivalent du moteur à courant continu.

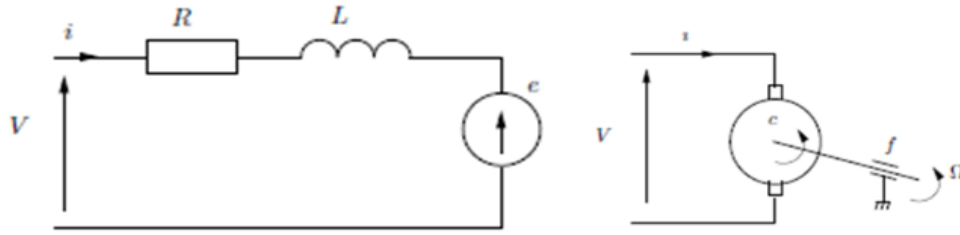


Figure I.2 : Schéma équivalent d'un MCC à excitation séparée

L'équation électrique, liant la tension $V(t)$ aux bornes de l'induit (rotor), le courant d'induit $i(t)$ et la force électromotrice $e(t)$ est donnée par :

$$Ri(t) + L \frac{di(t)}{dt} + e(t) = V(t) \tag{I.1}$$

Où : R est la résistance de l'induit du MCC, L son inductance et $e(t)$ la force électromotrice, qui est proportionnelle à la vitesse de rotation du rotor :

$$e(t) = K_e \Omega(t) \tag{I.2}$$

Où : K_e est la constante électrique du moteur (constante de vitesse) et $\Omega(t)$ la vitesse de rotation.

L'équation mécanique rendant compte des couples agissant sur le rotor s'écrit :

$$J \frac{d\Omega(t)}{dt} = C(t) - C_r(t) - f\Omega(t) \tag{I.3}$$

Où : $C(t)$ est le couple moteur, $C_r(t)$ le couple résistant (charge et perturbations), f le coefficient de frottement visqueux et J le moment d'inertie du rotor. Par construction, le couple $c(t)$ est proportionnel au courant d'induit $i(t)$:

$$C(t) = K_c i(t) \tag{I.4}$$

Où : k_c est la constante du couple moteur

En général, les coefficients K_e et K_c sont si proches qu'il est raisonnable de les considérer égaux, négligeant alors les pertes durant la conversion électromécanique de puissance.

On pose : $K = K_e = K_c$

Les transformées de Laplace des équations (1), (2), (3) et (4) sont données par :

$$(R + Lp)I(P) = V(P) - E(P) \tag{I.5}$$

$$E(P) = k\Omega(P) \tag{I.6}$$

$$C(P) - C_r(P) = (f + Jp)\Omega(P) \tag{I.7}$$

$$C(P) = kI(P) \tag{I.8}$$

Les transformées de Laplace obtenues nous permettent de modéliser le moteur sous forme de schéma bloc ou schéma fonctionnel. Ces schémas présentent les fonctions de transfert suivant les paramètres d'entrée et de sortie considérés. Si on considère comme grandeur d'entrée la tension appliquée aux bornes de l'induit et comme grandeur de sortie la vitesse du moteur, nous obtenons le schéma bloc tension- vitesse. Ce dernier nous permet de voir l'évolution de la vitesse avec la tension, qui est donné par la Figure II.3 [3]

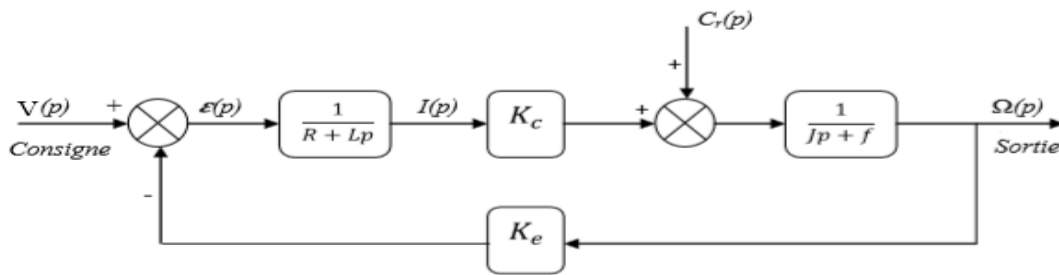


Figure. I.3 : Schéma bloc du Modèle du moteur électrique boucle fermée

I.5 Simulation en boucle ouverte du MCC

Les caractéristiques du Moteur à courant continu sont données par le tableau I.1

Tension nominale	V=220 V
Vitesse à la tension nominale	N=2100 tr/mn
Résistance de l'induit	R=0.6 Ω
Inductance de l'induit	L=6 mH
Constante de vitesse et de couple	K=1 V/rad/s
Coefficient de frottements visqueux	f=0.001 N.m/rd.s ⁻¹
Inertie du moteur	J=0.01 Kg.m ²

TableauxI.1 : Caractéristiques de la MCC.

Les résultats obtenus sont illustrés par la Figure I.4

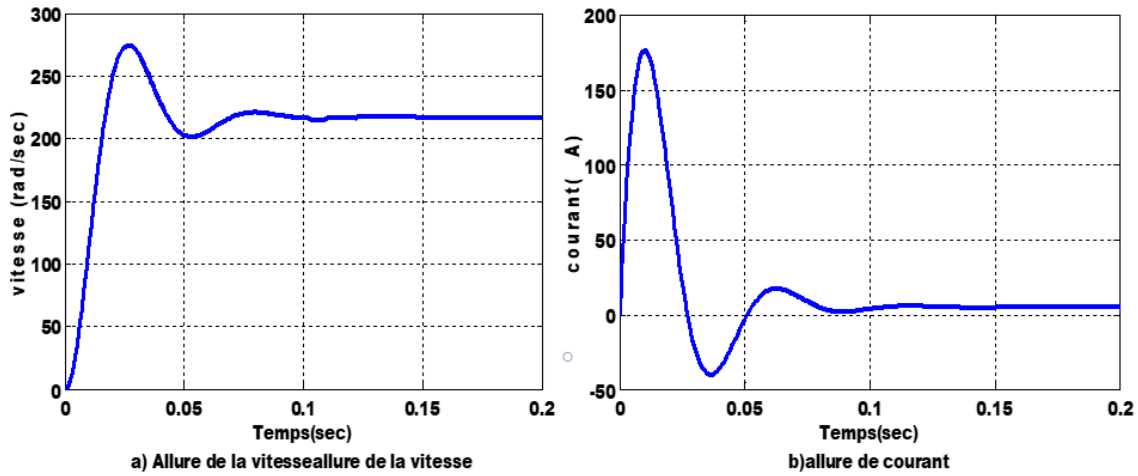


Figure I.4 : Réponse de la MCC en boucle ouverte.

❖ **Interprétation**

D’après la figure précédente on constate que : La réponse de vitesse oscille avec un dépassement=13.5%. Et un temps de réponse $\tau_{\text{repnse}} = 0.065 \text{ sec}$, et un temps de montée $\tau_{\text{repnse}} = 0.015 \text{ sec}$. Le courant subit au moment du démarrage un pic, puis atteint la valeur nulle

Pour un moteur électrique à courant continu, on a constaté que la réponse en vitesse est instable, d’où la nécessité d’une commande pour stabiliser le système et améliorer ses performances.

I.6 Asservissement du MCC avec un régulateur PI

L’asservissement est réalisé à l’aide d’un régulateur Proportionnel-Intégral-Dérivée PID qui se compose de trois termes P, I et D, d’où le P correspond au terme proportionnel, I pour le terme intégral et D pour le terme dérivé. Le régulateur PID est très utilisé pour l’asservissement des moteurs à courant continu soit en position ou en vitesse. On s’intéresse dans ce travail par l’asservissement en vitesse du MCC. La loi de commande est donnée par l’équation suivante :

$$u(t) = k_p \varepsilon(t) + k_i \int_0^t \varepsilon(t) dt + k_d \frac{d\varepsilon(t)}{dt} \tag{I.9}$$

- u désigne la commande.
- ε désigne l'erreur entre la consigne Entrée E et la sortie réelle S .
- k_p, k_i et k_d sont les paramètres du contrôleur PID.

Le régulateur PID est inséré dans la chaîne directe de l'asservissement, en série avec le processus, comme est indiqué dans la Figure I.6. Ce régulateur élabore, à partir du signal d'erreur $\varepsilon(t)$, une commande $u(t)$ en fonction de trois actions. 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 [4]. Ce travail a pour but, d'implémenter la commande PID classique pour un MCC. L'objectif est d'annuler l'erreur statique (Précision), de diminuer le dépassement (Stabilité), et de diminuer le temps de réponse et le temps de montée (Rapidité). L'ajustement des paramètres k_p, k_i et k_{id} de la commande PID est basé sur la méthode classique.

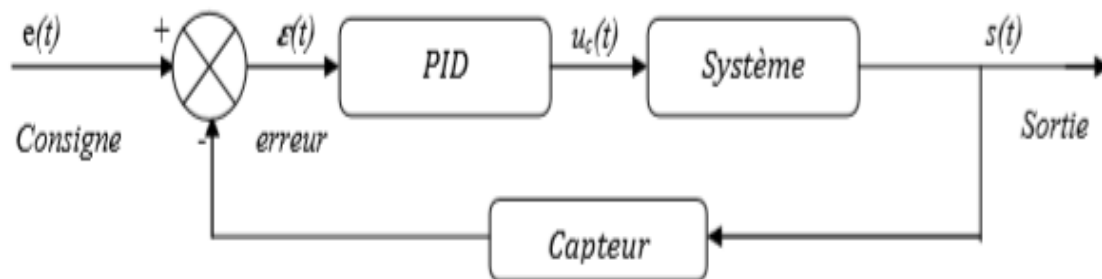


Figure I.5 : Schéma bloc d'asservissement.

I.7 Description des régulateurs PID :

Un régulateur PID remplit essentiellement trois fonctions :

- Il fournit un signal de commande $u(t)$ en tenant compte de l'évolution du signal de sortie $s(t)$ par rapport à la consigne $e(t)$.
- Il élimine l'erreur statique grâce au terme intégrateur.
- Il anticipe les variations de la sortie grâce au terme dérivateur [5].

Le PID représente les abréviations des trois actions qu'il utilise pour effectuer ses corrections, ce sont des ajouts d'un signal à un autre. Tous agissent sur la quantité régulée. Les actions aboutissent finalement à des soustractions de l'erreur de mesure.

I.7.1 Action Proportionnelle :

Pour cette action, l'erreur est multipliée par la constante P (pour Proportionnel) qui est négative, puis ajoutée (soustraction de l'erreur de mesure) à la quantité régulée. P est valide uniquement sur la bande dans laquelle le signal de sortie du régulateur est proportionnel à l'erreur du système. Noter que si l'erreur de mesure est égale à zéro, la partie proportionnelle de la sortie du régulateur est également à zéro [6].

I.7.2 Action Intégrale :

L'action intégrale fait intervenir la notion de temps. Elle tire profit du signal d'erreur passé qui est intégré (additionné) pendant un laps de temps, puis multiplié par la constante I (négative) ce qui en fait une moyenne, elle est enfin additionnée (soustraction de l'erreur de mesure) à la quantité régulée. La moyenne de l'erreur de mesure permet de trouver l'erreur moyenne entre la sortie du régulateur et la valeur de la consigne. Un système seulement proportionnel oscille en plus et en moins autour de la consigne du fait qu'en arrivant vers la consigne, l'erreur est à zéro, il n'enlève alors plus rien est dépassé la consigne, ou oscille et/ou se stabilise à une valeur trop basse ou trop élevée. L'addition sur l'entrée d'une proportion négative (soustraction) de l'erreur de mesure moyennée, permet toujours de réduire l'écart moyen entre la mesure en sortie et la consigne. Donc finalement, une boucle PI bien réglée verra sa sortie redescendre lentement à la valeur de la consigne [6].

I.7.3 Action Dérivée

L'action dérivée utilise aussi la notion de temps. Elle cherche à anticiper l'erreur future. La dérivée première (la pente de l'erreur) est calculée pour un laps de temps et multipliée par la constante (négative) D, puis elle est additionnée (soustraction de l'erreur de mesure) à la quantité régulée. L'action dérivée de la régulation fournit une réponse aux perturbations agissant sur le système. Plus important est le terme dérivé, plus rapide sera la réponse en sortie à une perturbation sur l'entrée.

I.8 L'effet des paramètres d'un contrôleur PID sur la réponse**I.8.1 Proportionnel**

L'erreur est multipliée par une constante K_p :

$$u(t) = k_p \varepsilon(t) \tag{I.10}$$

Plus K_p est grand, plus que la réponse est rapide et plus que l'erreur statique est corrigé.

I.8.2 Intégral

L'erreur est intégrée sur un intervalle de temps, puis multipliée par une constante K_i :

$$u(t) = k_i \int_0^t \varepsilon(t) dt \tag{I.11}$$

Plus K_i est élevé, plus que l'erreur statique est corrigée.

I.8.3 Dérive

L'erreur est dérivée par rapport au temps, puis multipliée par une constante K_d :

$$u(t) = k_d \frac{d\varepsilon(t)}{dt} \tag{I.12}$$

Réduit le dépassement et le temps de stabilisation [7].

Lorsque :

- K_p augmente \rightarrow montée plus rapide mais plus de dépassement.
- K_i augmente \rightarrow montée plus rapide mais régime stationnaire plus long, erreur statique plus faible.
- K_d augmente \rightarrow diminue le dépassement et le temps d'établissement du régime stationnaire, mais augmente la sensibilité au bruit

Le tableau suivant résume l'effet de chaque paramètre de contrôleur PID sur la réponse :

Coefficient	Temps de monté	Temps de stabilisation	Dépassement	Erreur Statique
K_P	Diminue	Augmente	Augmente	Diminue
K_I	Diminue	Augmente	Augmente	Annule
K_D	-	Diminue	Diminue	-

Tableau I.2: Influence des paramètres K_p , K_i et K_d sur la réponse du système [4].

I.9 Synthèse, calcul et choix du correcteur

Habituellement, les schémas de régulation de la vitesse sont constitués de deux étages, le premier comprend les boucles de régulation des courants, et le second; externe au premier, est destiné à la régulation de la vitesse. Ceci est justifié par le fait qu'en général, on considère que la dynamique de la vitesse est très lente devant la dynamique des courants, c'est pour cette raison que, dans les régulations classiques, la vitesse est toujours supposée constante pendant que les courants sont régulés.

Régler une boucle de régulation consiste à agir sur les paramètres des différentes actions (gain du proportionnel, gain de l'intégral, gain de la dérivée) sur des valeurs optimales pour obtenir la réponse désirée sur la sortie du procédé [6].

I.9.1 Etude de la boucle du courant

La boucle de régulation du courant peut se représenter par le schéma bloc suivant:

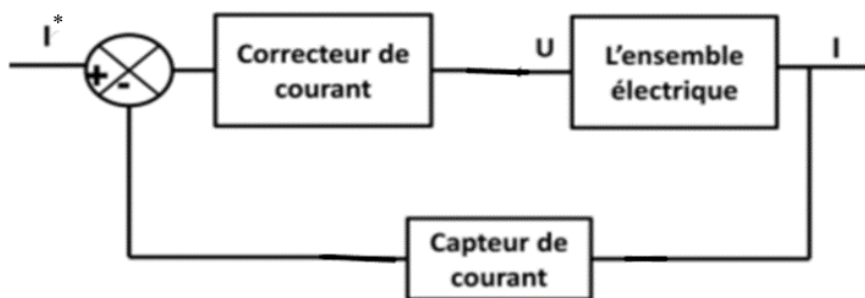


Figure I.6 : Schéma bloc de la boucle du courant

I.9.1.1 Identification des blocs

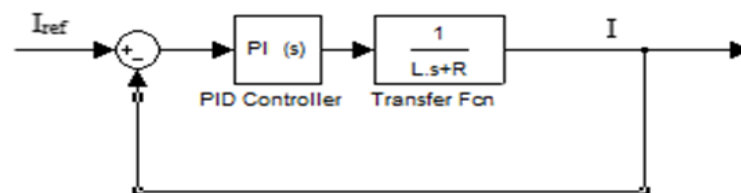


Figure I.7 : Schéma bloc de la boucle du courant identifié

Avec : La fonction de transfert du courant est :

$$FT_{\text{électrique}} = \frac{1}{L \cdot p + R} = \frac{1/R}{\frac{L}{R} p + 1} = \frac{1/R}{\tau p + 1} \quad (\text{I.13})$$

La fonction de transfert du correcteur PI est :

$$PI = K_p + \frac{K_i}{p} = \frac{K_p p + K_i}{p} = \frac{K_i(1 + \frac{K_p}{K_i} \cdot p)}{p} = \frac{K_i(1 + \tau_{\text{reg}} \cdot p)}{p} \quad (\text{I.14})$$

Et un retour unitaire égal 1

Notons : $\tau_{\text{reg}} = \frac{K_p}{K_i}$ et $\tau_{\text{elec}} = \frac{L}{R}$

I.9.1.2 Asservissement du courant

➤ **Boucle ouverte :**

La fonction de transfert en boucle ouverte est la suivante:

$$FT_{\text{BO}}(p) = PI \times FT_{\text{électrique}} = \frac{K_i(1 + \tau_{\text{reg}} \cdot p)}{p} \times \frac{1/R}{\tau_{\text{elec}} p + 1} \quad (\text{I.15})$$

On a choisit $\tau_{\text{elec}} = \tau_{\text{reg}}$ pour éliminer le dénominateur de la fonction de transfert de l'ensemble électrique.

Donc :

$$FT_{\text{BO}}(p) = \frac{K_i/R}{p} \quad (\text{I.16})$$

Notons : $\frac{K_i}{R} = K_1$

➤ **Boucle fermée:**

La fonction de transfert en boucle fermée est la suivante:

$$FT_{\text{BF}}(P) = \frac{FT_{\text{BO}}}{1 + FT_{\text{BO}}} = \frac{\frac{K_1}{p}}{1 + \frac{K_1}{p}} = \frac{K_1}{p + K_1} = \frac{K_1}{K_1 \left(\frac{1}{K_1} p + 1 \right)} = \frac{1}{1 + \tau_{\text{BF}} p} \quad (\text{I.17})$$

Notons : $\tau_{\text{BF}} = \frac{1}{K_1}$

Selon le cahier de charge, on veut que le temps de réponse en boucle fermée corrigée soit rapide 20 fois par rapport à la boucle ouverte non corrigée.

On pose : $\tau_{\text{BF}} = \frac{\tau_{\text{Bo}}}{20} = \frac{1}{K_1} = \frac{1}{K_i/R} = \frac{R}{K_i} = \frac{\tau_{\text{Bo}}}{20}$

$$K_I = \frac{20R}{\tau_{Bo}} \tag{I.18}$$

$$\tau_{reg} = \frac{K_P}{K_I} \rightarrow K_P = \tau_{reg} \times K_I = \tau_{elec} \times K_I = K_I \times \frac{L}{R}$$

$$K_P = K_I \times \frac{L}{R} \tag{I.19}$$

$$\tau_{rBF} = 3\tau_{BF} = \frac{3\tau_{sys}}{20} = \frac{R}{K_I} = \frac{3L}{20R} = \frac{R}{K_I}$$

Application numérique :

$$K_I = \frac{20R^2}{3L} = \frac{20 \times (0.6)^2}{3(0.006)} = 400 \tag{I.20}$$

$$K_P = K_I \times \frac{L}{R} = \frac{400 \times 0.006}{0.6} = 4 \tag{I.21}$$

Les résultats obtenus de régulateur PI du courant est :

$$PI = K_p + \frac{K_I}{p} = 4 + \frac{400}{p} \tag{I.22}$$

I.9.2 Etude de la boucle de la vitesse

Il s'agit d'étudier la boucle de vitesse sans implémenter la boucle de courant. Cet asservissement a pour schéma bloc:

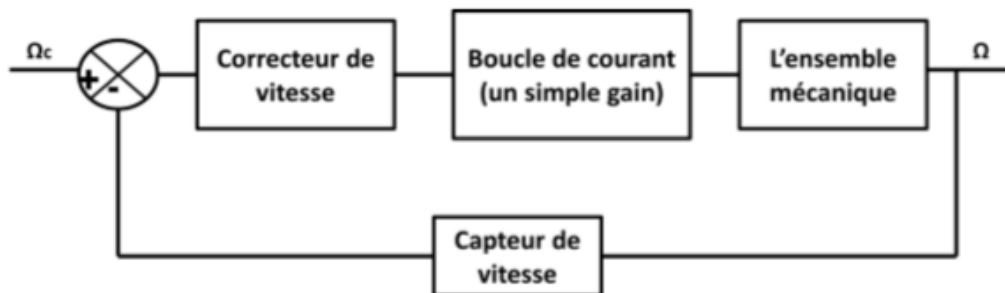


Figure I.8 : Schéma bloc de la boucle de vitesse

I.9.2.1 Identification des blocs:

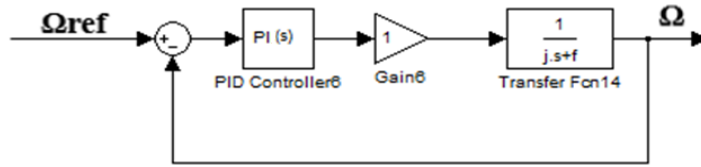


Figure I.9 : Schéma bloc de la boucle de la vitesse identifié

Avec : La fonction de transfert de la vitesse est :

$$FT_{mec} = \frac{1}{J \cdot p + f} \tag{I.23}$$

La fonction de transfert du correcteur PI est :

$$PI = K_p + \frac{K_I}{p} = \frac{K_p p + K_I}{p} = \frac{K_I(1 + \frac{K_p}{K_I})}{p} \tag{I.24}$$

Et un retour unitaire égal 1

Notons : $K_2 = \frac{K_p}{K_I}$; $K_v = K_I$

I.9.2.2 Asservissement de la vitesse

➤ **Boucle ouverte :**

La fonction de transfert en boucle ouverte est la suivante :

$$FT_{BO}(p) = PI \times FT_{mec} = \frac{K_v(1 + K_2 p)}{p} \times \frac{1}{j p + f} \tag{I.25}$$

$$FT_{BO} = \frac{K_v(1 + K_2 p)}{p(j p + f)} \tag{I.26}$$

➤ **Boucle fermée:**

La fonction de transfert en boucle fermée est la suivante:

$$\begin{aligned} FT_{BF} &= \frac{FT_{BO}}{1 + FT_{BO}} = \frac{\frac{K_v(1+K_2P)}{P(jp+f)}}{1 + \frac{K_v(1+K_2P)}{P(jp+f)}} = \frac{K_v(1 + K_2 P)}{P(j p + f) + K_v(1 + K_2 P)} \\ &= \frac{K_v(1 + K_2 P)}{j P^2 + (f + K_v K_2) P + K_v} \end{aligned} \tag{I.27}$$

En imposant toujours des pôles correspondants au critère utilisé

$$\frac{K_V(1 + K_2P)}{JP^2 + (f + K_VK_2)P + K_V} = \frac{\omega_n^2}{P^2 + 2\xi\omega_nP + \omega_n^2} \quad (\text{I.28})$$

Cette fonction peut s'écrire sous d'autres formes :

$$\frac{K_V(1 + K_2P)}{P^2 + \frac{(f+K_VK_2)}{J}P + \frac{K_V}{J}} = \frac{\omega_n^2}{P^2 + 2\xi\omega_nP + \omega_n^2} \quad (\text{I.29})$$

L'équation terme à terme de l'équation (I.27) entraîne que :

$$\begin{cases} 2\xi\omega_n = \frac{f + K_VK_2}{J} \\ \omega_n^2 = \frac{K_V}{J} \end{cases} \Rightarrow \begin{cases} K_V = \omega_n^2 \times J \\ K_2 = \frac{J2\xi\omega_n - f}{K_V} \end{cases} \quad (\text{I.30})$$

En posant au polynôme caractéristique deux pôles afin d'avoir en boucle fermée :

- Un amortissement $\xi=1$
- Une pulsation ω_n a temps de réponse a 2%

➤ **Détermination du temps de réponse à 2% d'un système 2^{eme} ordre**

La valeur de temps de réponse a 2% de la valeur final

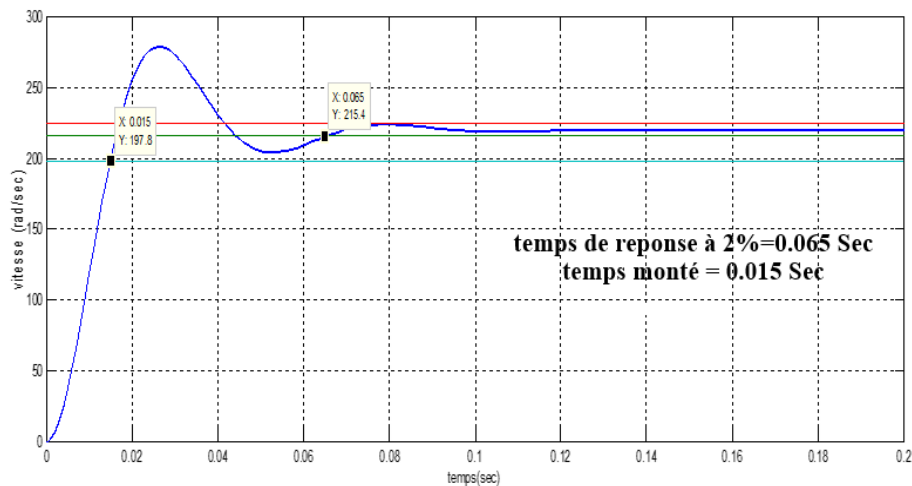


Figure I.10 : Détermination du temps de réponse à 2%

$$\tau_{\text{reponse}} = \frac{4}{\xi\omega_n} \Rightarrow \xi = 1 \text{ donc } \omega_n = \frac{4}{\tau_{\text{reponse}}} \quad (\text{I.31})$$

A partir de simulation : $\tau_{\text{reponse}} = 0.0653 \text{ sec}$

$$\text{donc: } \omega_n = \frac{4}{\tau_{\text{reponse}}} = \frac{4}{0.0653} = 61.25 \quad (\text{I.32})$$

Application numérique :

$$\begin{cases} K_V = K_I = \omega_n^2 \times J = 61.25 \times 0.01 = 37.51 \\ K_2 = \frac{J2\xi\omega_n - f}{K_V} = \frac{0.01 \times 1 \times 61.25 - 0.001}{37.5} = 0.0331 \end{cases} \quad (I.33)$$

$$K_2 = \frac{K_P}{K_I} \Rightarrow K_P = K_2 \times K_I = 0.0331 \times 37.51 = 1.244 \quad (I.34)$$

Les résultats obtenus de régulateur PI de la vitesse est :

$$PI = K_P + \frac{K_I}{p} = 1.244 + \frac{37.51}{p} \quad (I.35)$$

I.9.3 Etude des deux boucles (courant et vitesse)

De ce qui précède, on déduit le schéma complet de contrôle d'un moteur à CC, à l'aide d'une régulation cascade incluant deux fonctions de transfert. Comme la constante de temps électrique est plus faible que la constante de temps mécanique, la boucle interne (qui doit être la plus rapide) contrôle le courant, la boucle externe contrôlant la vitesse. On aboutit au schéma suivant:

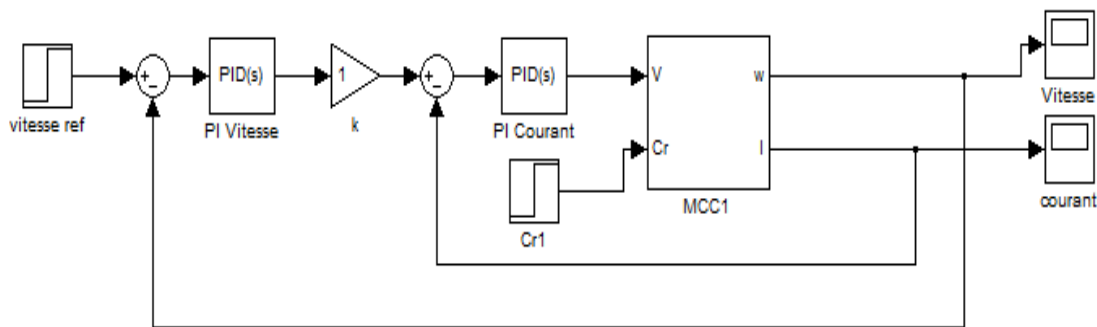


Figure I.11 : Schéma bloc de l'asservissement des deux boucles (vitesse et courant)

Les résultats obtenus sont illustrés par la Figure I.10 avec un couple résistant de 5 N.m

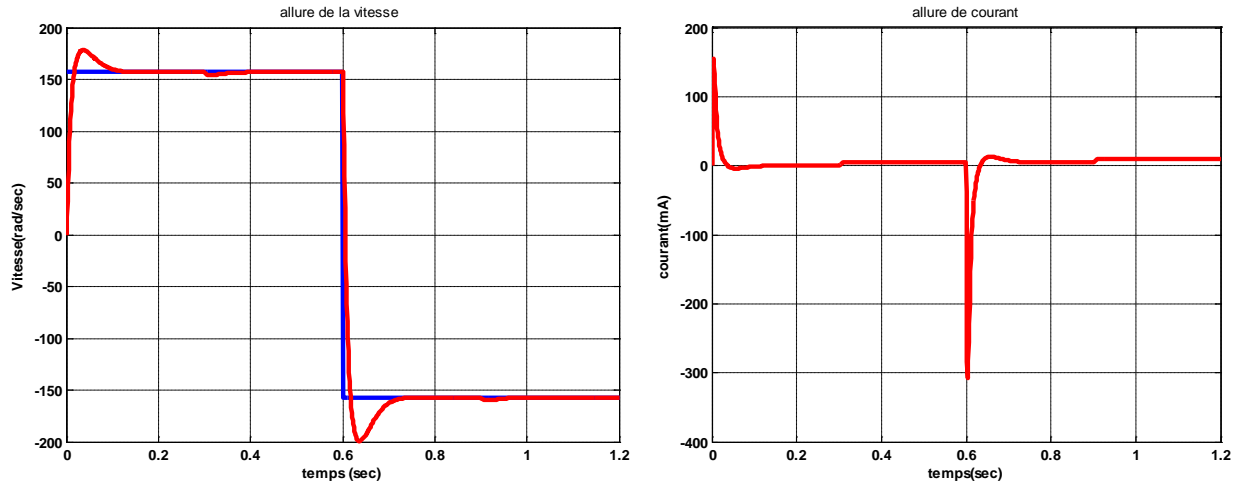


Figure I.12 : Régulation PI de la vitesse et de courant

❖ **Interprétation :**

D’après la figure précédente on constate que : La réponse de vitesse atteint la consigne (position désirée) en un temps de réponse $\tau_{repnse} = 0.015 \text{ sec}$, et un temps de montée $\tau_{repnse} = 0.013 \text{ sec}$, avec un dépassement=13.5%. On applique une charge après 0.3sec la vitesse chute de 3 rad/sec et retourne à sa valeur initiale avec un temps de rejet égal 0.08 sec. Le courant subit au moment du démarrage un pic, puis atteint rapidement la valeur nulle. On remarque qu’au moment d’application de la charge, le couple a augmenté pour équilibrer la charge (couple résistant)

Pour cette raison, on insère un filtre pour annuler le dépassement

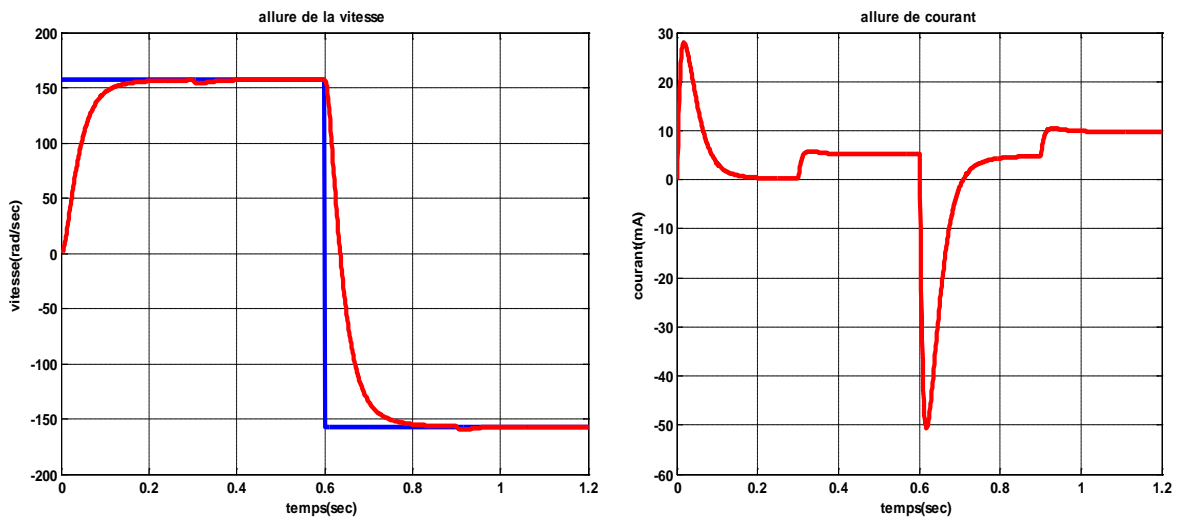


Figure I.13 : Régulation PI de la vitesse et du courant avec filtre

I.10 Conclusion

Dans ce chapitre nous avons vu que la commande PI appliquée aux moteurs électriques à courant continu était un régulateur qui pouvait offrir de bonnes performances, que ce soit au niveau de la précision, du temps de réponse, et de la stabilité. Cependant, il présente des dérives paramétriques.

II.1 Introduction :

Notre monde est intégralement constitué de systèmes vivants ou non-vivants, imbriqués et en interaction. Ainsi les ordinateurs et les machines intelligentes que nous connaissons aujourd'hui sont des applications de la cybernétique. Cette science cherche à expliquer, grâce à des concepts mathématiques, les comportements des systèmes naturels ou artificiels et les échanges d'informations entre ces systèmes. La cybernétique a été fondée en 1948 par le mathématicien américain Norbert Wiener, qui projetait la fondation d'une nouvelle discipline scientifique autonome, toutefois ces recherches n'ont abouti qu'à une contribution au développement de divers domaines parmi lesquels on distingue l'intelligence artificielle.

II.2 Intelligence artificielle

L'Intelligence Artificielle (IA) appartient au domaine de l'informatique et concerne les approches informatiques ayant pour but de faire apparaître un comportement intelligent. C'est une science qui s'intéresse à la conception de machines pouvant simuler la cognition humaine. En effet, elle tend à donner à la machine un comportement "humain", capable notamment de s'adapter à des situations nouvelles et d'apprendre en permanence. L'IA est directement issue des concepts cognitivistes, c'est à dire qu'elle envisage le fonctionnement cérébral sous un angle logico-déductif, et considère, de fait que l'acte cognitif s'effectue à travers une manipulation de symboles élémentaires. A travers le cognitivisme, l'IA établit une analogie entre le fonctionnement cérébral et celui de l'ordinateur [8]. Ce qui amène à dire que le but de l'IA est de construire des machines qui réalisent des choses qui requièrent de l'intelligence lorsqu'elles sont faites par des humains. Le terme d'IA est "né" officiellement durant la Dartmouth Conférence de l'été 1956 par Mc Carthy [9], mais était déjà dans les préoccupations des chercheurs depuis longtemps.

II.3 Techniques de Calcul intelligent

Le principe de base des méthodes de Calcul "Intelligent" consiste à considérer les êtres vivants comme modèles d'inspiration, le but étant de simuler à l'aide des machines leur comportement.

En général, ces techniques peuvent être regroupées en trois grandes classes : les réseaux de neurones artificiels qui utilisent l'apprentissage pour résoudre des problèmes complexes

tels que la reconnaissance des formes ou le traitement du langage naturel, la logique floue utilisée dans des applications d'intelligence artificielle, dans lesquelles les variables ont des degrés de vérité représentés par une gamme de valeurs situées entre 1 (vrai) et 0 (faux), et les méthodes de calcul évolutif pour la recherche de l'optimisation (Figure II.1). Ces différentes classes seront présentées dans les sections suivantes :

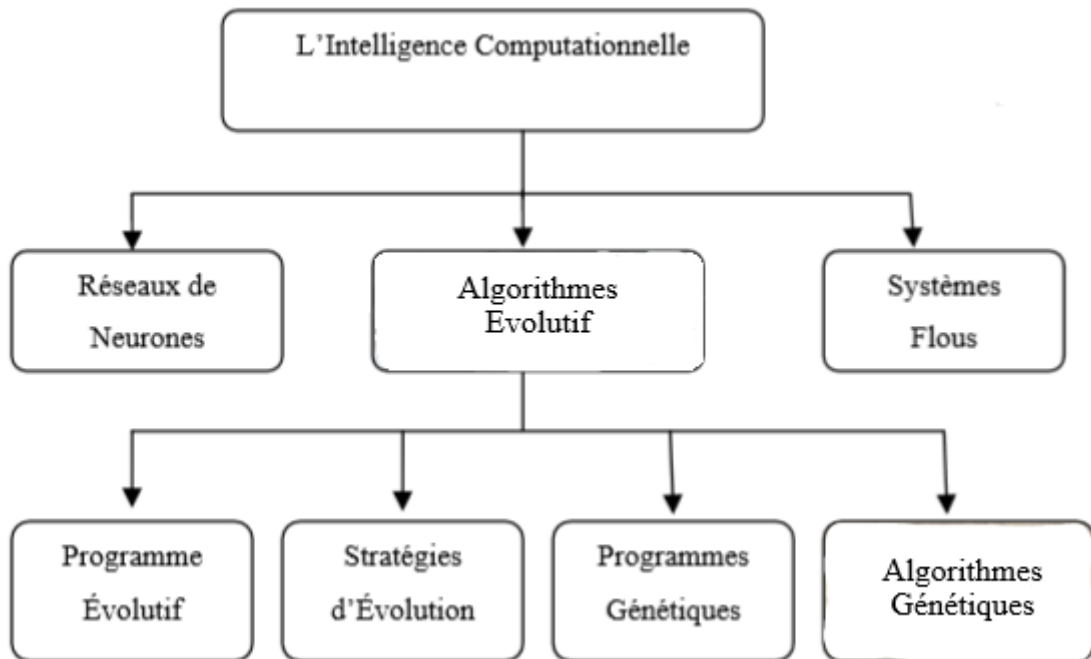


Figure II.1 : Différentes méthodes de l'intelligence computationnelle

II.4 Algorithmes génétiques

Les algorithmes génétiques sont une technique très adaptée aux problèmes d'optimisation et de recherche d'extrémums dans un environnement complexe. Ils sont basés sur les processus génétiques des organismes biologiques qui font évoluer une population d'une génération à l'autre suivant le principe de la sélection naturelle aussi appelée « la survie du plus fort » évoquée par Charles Darwin dans son livre «The origin of species ». Par imitation de ce principe, les algorithmes génétiques sont capables de faire évoluer une population de solution très faible vers une population très performante [10].

John Holland fut le pionnier et l'architecte des algorithmes génétiques, le premier à présenter leurs qualités intéressantes pour la résolution des problèmes d'optimisation

complexe. Compte à leurs fondements théoriques, ils furent exposés par Goldberg [11] qui est également le premier à les avoir employés dans la résolution d'un problème réel.

II.4.1 Principe des Algorithmes génétiques

Dans les algorithmes génétiques (AGs), l'auteur réalise une transposition artificielle des concepts régissant les organismes biologiques au niveau génétique, où chaque individu est représenté et identifié par un code constitué d'un nombre fini d'alphabets formant ce qu'on appelle «le chromosome ». L'ensemble des individus générés forment à leur tour une population au sein de laquelle, sévit une compétition à la survie et à la reproduction. Chaque individu détient un niveau de performance et d'adaptation aussi appelé « fitness de l'individu », c'est à base de cette fitness que les individus sont sélectionnés. Les plus performants sont maintenus pour être les géniteurs de la nouvelle population, tandis que les moins performants ont tendance à disparaître.

Le croisement plus connu sous le nom de reproduction, se fait par un simple mixage aléatoire des chromosomes parents donnant ainsi naissance à des enfants généralement plus performants. Mais certaines fois le patrimoine génétique de la population n'est pas correctement transmis aux enfants, entraînant ainsi l'apparition de nouvelles empreintes génétiques chez les enfants mutants, ce phénomène est appelé la mutation. Dans l'AG la mutation est provoquée aléatoirement dans les chromosomes des enfants avec une probabilité très faible, et ce en modifiant certaines allèles du chromosome [12].

II.4.2 Conception d'un Algorithme génétique (procédé d'implantation des AG)

La simplicité de mise en œuvre et l'efficacité constituent deux des caractéristiques les plus attrayantes de l'approche proposée par les AG. La mise en œuvre d'un algorithme génétique sollicite la disponibilité [13], [14] :

1. D'une représentation génétique du problème, c'est-à-dire un codage approprié des solutions sous la forme de chromosomes. Cette étape associe à chacun des points de l'espace de recherche une structure de données. Elle se place généralement après une phase de modélisation mathématique du problème traité. La qualité du codage des données conditionne le succès des algorithmes génétiques.

2. D'un mécanisme de génération de la population initiale. Ce mécanisme doit être capable de produire une population non homogène qui servira de base pour les générations futures. Le choix de la population initiale est important car il peut prendre plus ou moins rapidement la convergence vers l'optimum global. Dans le cas où l'on ne connaît rien sur le problème à résoudre, il est essentiel que la population initiale soit répartie sur tout le domaine de recherche.
3. D'une fonction d'évaluation pour mesurer la force de chaque chromosome.
4. D'un mode de sélection des chromosomes à reproduire.
5. Des opérateurs permettant de diversifier la population au cours des générations et d'explorer l'espace de recherche. L'opérateur de croisement recompose les gènes d'individus existant dans la population, l'opérateur de mutation a pour but de garantir l'exploration de l'espace de recherche.
6. Des valeurs pour les paramètres qu'utilise l'algorithme : taille de la population, nombre total de générations ou critère d'arrêt, probabilités de croisement et de mutation.

II.4.3 Paramètres des Algorithmes génétiques

En réalité, il n'existe pas de paramétrage universel pour la quantification de ce paramètre de dimensionnement. Cependant, certaines valeurs largement utilisées pour résoudre concrètement des problèmes, méritent d'être retenues :

- Taille de la population : entre $N=30$ et 50 individus
- Taux de croisement : la probabilité de croisement est choisie entre $[0.7 \quad 0.99]$;
- Taux de mutation : la probabilité de mutation est choisie entre $[0.001 \quad 0.01]$.

II.4.4 Opérateurs des Algorithmes génétiques

II.4.4.1 Le codage

Le codage est une modélisation d'une solution d'un problème donné sous forme d'une séquence de caractères appelée chromosome où chaque caractère, dit aussi gène, représente une variable ou une partie du problème. La tâche principale consiste à choisir le contenu des gènes qui facilite la description du problème et respecte ses contraintes [14]. La littérature définit deux types de codage : binaire et réel.

a. Le codage binaire

Le codage classique utilise l'alphabet binaire: [0 , 1]. Dans ce cas le chromosome représente simplement une suite de 0 et de 1. Le codage binaire est également indépendant des opérateurs génétiques (croisement et mutation) du moment où ces derniers ne nécessitent aucune spécification. En effet, toute manipulation d'un chromosome donne naissance à un nouveau chromosome valide. Dans la pratique, le codage binaire peut présenter des difficultés. En effet, il est parfois très difficile ou très lourd de coder des solutions de cette manière. En outre, dans certain cas , la taille mémoire requise peut devenir prohibitive [14].

b. Le codage réel

Pour certain problèmes d'optimisation, il est plus pratique d'utiliser un codage réel des chromosomes. Un gène est ainsi représenté par un nombre réel au lieu d'avoir à coder les réels en binaire puis de les décoder pour les transformer en solutions effectives. Le codage réel permet d'augmenter l'efficacité de l'algorithme génétique et d'éviter des opérations de décodage supplémentaires. En effet, un chromosome codé en réel est plus court que celui codé en binaire.

II.4.4.2 Population initiale

Une fois le codage choisi, une population initiale formée de solutions admissibles du problème doit être déterminée. Plusieurs mécanismes de génération de la population initiale sont utilisés dans la littérature. Le choix de l'initialisation se fera en fonction des connaissances que l'utilisateur a sur le problème. S'il n'a pas d'informations particulières, alors une initialisation aléatoire, la plus uniforme possible afin de favoriser une exploration de l'espace de recherche maximum, sera la plus adaptée. Mais dans d'autres cas, il est possible d'utiliser d'autres mécanismes. Par ailleurs, cette étape présente un problème principal qui est celui du choix de la taille de la population. En effet une population trop grande augmente le temps de calcul et demande un espace mémoire considérable, alors qu'une population trop petite conduit à l'obtention d'un optimum local [14].

II.4.4.3 La Sélection :

L'évaluation d'un individu ne dépendant pas de celle des autres individus, le résultat fourni par la fonction d'évaluation va permettre de sélectionner ou de refuser un individu pour ne garder que les individus ayant le meilleur coût en fonction de la population courante : c'est le rôle de la fonction de fitness. La Sélection permet de s'assurer que les individus

performants seront conservés, alors que les individus peu adaptés seront progressivement éliminés de la population. En d'autres termes, elle permet, à partir d'un chromosome, de calculer le coût d'un point de l'espace de recherche, de quantifier numériquement la validité de la solution qu'il représente et de mesurer la santé et le degré d'adaptation d'un individu à son environnement. Il existe plusieurs méthodes de sélection, mais la plus connue est celle du tournoi : on tire deux individus aléatoirement dans la population et on reproduit le meilleur des deux dans la nouvelle population. On refait cette procédure jusqu'à ce que la nouvelle population soit complète. Cette méthode donne de bons résultats. D'autre part la phase de sélection ne crée pas de nouveaux individus dans la population. Ceci est le rôle des opérateurs de croisement et de mutation.

II.4.4.4 Reproduction avec des opérateurs de croisement et de mutation :

II.4.4.4.1 Opérateur de Croisement :

Le croisement permet la création de nouveaux individus selon un processus fort simple. Il prend en entrée un couple d'individus parents P1 et P2 et renvoie un couple d'individus enfants E1 et E2 obtenus en choisissant aléatoirement un point de croisement (ou plusieurs points de croisement) dans les chromosomes et en recopiant dans le fils E1 les gènes de P1 jusqu'au point de croisement puis en complétant avec les gènes de P2. On effectue l'opération symétrique pour E2.

Voici deux méthodes de croisement classiques : le croisement en un point et le croisement en deux points. Notons qu'il existe une multitude de croisements en fonction de la nature du problème à traiter ; on pourrait aussi considérer le croisement en k points ou croisement uniforme.

a. Croisement en un point :

Considérons des chromosomes constitués de M bits. Initialement, on tire aléatoirement une position de croisement. On échange ensuite les deux sous chaînes terminales de chacun des deux chromosomes parents P1 et P2, ce qui produit deux nouveaux chromosomes enfants E1 et E2

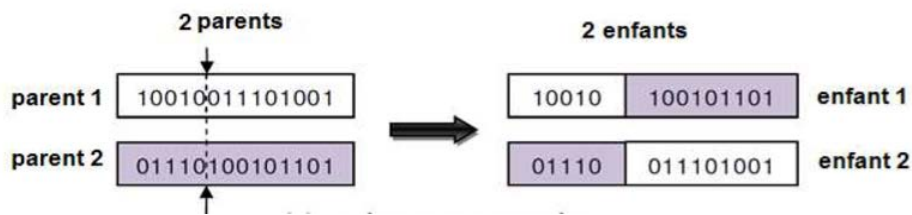


Figure II.2 : Croisement en un point

b. Croisement en deux points :

On choisit aléatoirement deux points de croisement. On échange ensuite les deux sous-chaînes situées entre les deux points de croisement de chacun des deux chromosomes parents P1 et P2, ce qui produit deux nouveaux chromosomes enfants E1 et E2.

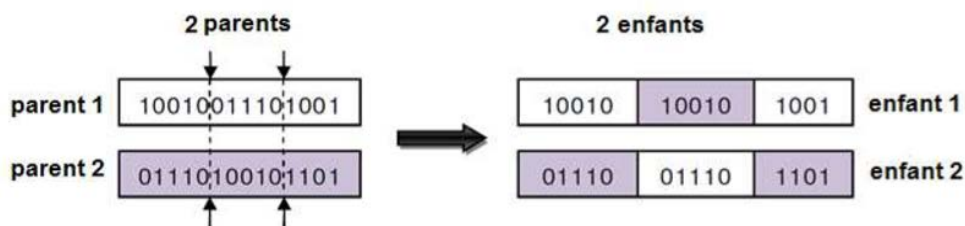


Figure II.3 : Croisement en deux-points

II.4.4.4.2 Opérateur de Mutation :

La mutation est un opérateur qui agit sur la population des enfants, son rôle est de garantir une diversité dans la population générée en introduisant de nouveaux gènes au patrimoine génétique de la population et ce avec une probabilité (P_m) souvent très faible.

L'opérateur de mutation diffère suivant le type du codage utilisé, ainsi dans le codage binaire la mutation est très simple à élaborer, il suffit de compléter à 1 toutes les allèles des chromosomes des enfants pour qui, la probabilité est inférieure à (P_m). Figure II.4 donne un exemple illustratif pour la mise en œuvre de cet opérateur

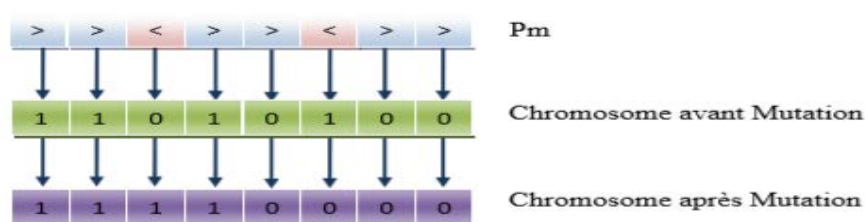


Figure II.4 : Principe de la mutation binaire

L'opérateur de mutation modifie donc de manière complètement aléatoire, les caractéristiques d'une solution, ce qui permet d'introduire et de maintenir la diversité au sein de notre population de solutions. Cet opérateur joue le rôle d'un "élément perturbateur", il introduit du "bruit" au sein de la population.

II.4.5 Critère d'arrêt

Déterminer l'arrêt d'un processus génétique est l'une des difficultés majeures de l'approche génétique. En effet, si l'on excepte le cas des problèmes artificiels, on ne sait jamais si l'on a trouvé l'optimum. Dans la pratique, l'utilisateur déclare un nombre de générations maximum. La recherche peut également être stoppée lorsque tous les individus d'une même population sont des copies d'un même individu. On dit alors qu'il y a "perte de diversité génétique" [15].

II.4.6 Organigramme de la procédure l'algorithme génétique

Le schéma suivant résume les étapes de l'algorithme génétique :

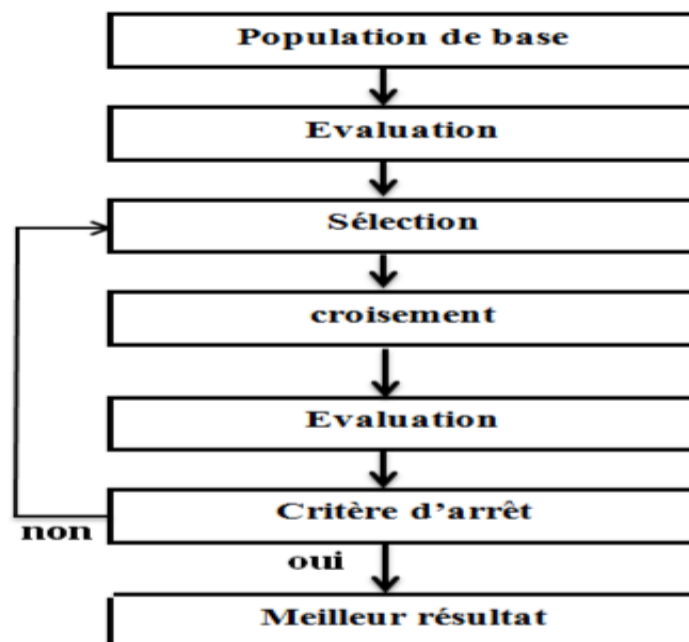


Figure II.5 : Organigramme de l'algorithme génétique

II.4.7 Développement d'un algorithme génétique pour le réglage PI

Pour faciliter le développement de ce projet, un système a été choisi et un contrôleur PI a été conçu à l'aide de méthodes conventionnelles. Un algorithme génétique a ensuite été créé pour évaluer les coefficients PI du même système et les résultats des deux techniques ont été comparés.

II.4.7.1 Écrire la fonction objective pour un algorithme génétique :

Écrire une fonction objective est la partie la plus difficile de la création d'un algorithme génétique. Dans ce projet, la fonction objective est nécessaire pour évaluer le meilleur contrôleur PI pour la MCC. Une fonction objective peut être créée pour trouver un contrôleur PI offrant le plus petit dépassement, le temps de montée le plus rapide ou le temps de réglage le plus rapide. Toutefois, afin de combiner tous ces objectifs, il a été décidé de concevoir une fonction objective qui minimise l'erreur du système contrôlé.

Le contrôleur PI est placé dans une boucle de retour unité avec la fonction de transfert. Afin de réduire le temps de compilation du programme, la fonction de transfert est définie dans un autre fichier et importée en tant que variable globale. Le système contrôlé reçoit une entrée pas à pas et l'erreur est évaluée en utilisant un critère de performance d'erreur approprié, à savoir ITAE, ISE, IAE ou MSE.

II.4.7.2 Indices de performance

Diverses fonctions objectives ont été écrites sur la base de critères de performance d'erreur. Chaque fonction d'objective est fondamentalement la même, à l'exception de la section du code qui définit le critère de performance d'erreur spécifique mis en œuvre. Pour optimiser les performances d'un système contrôlé par PI, les gains PI du système sont ajustés pour maximiser ou minimiser un certain indice de performance. L'indice de performance est calculé sur un intervalle de temps T, normalement dans la région de $0 \leq T \leq t_s$ où t_s est le temps d'établissement du système. Les indices de performance utilisés étaient:

- Intégrale du temps multipliée par l'erreur absolue (ITAE) :

$$I_{ITEA} = \int_0^T t|e(t)|dt \quad \text{(II.1)}$$

- Intégrale de la grandeur absolue de l'erreur (IAE) :

$$I_{IAE} = \int_0^T |e(t)| dt \tag{II.2}$$

➤ Intégrale du carré de l'erreur (ISE) :

$$I_{ISE} = \int_0^T (e(t))^2 dt \tag{II.3}$$

➤ Moyenne du carré de l'erreur (MSE) :

$$I_{MSE} = \frac{1}{n} \sum_{i=1}^n (e(t))^2 \tag{II.4}$$

II.4.8 Application des AG à l'optimisation du régulateur PI pour la commande d'une machine à courant continu:

Dans cette partie nous présentons l'optimisation des paramètres du PI par l'utilisation des AG. L'objectif est toujours de trouver l'ensemble des paramètres optimisés PI de façon à ce que la réponse en vitesse et en courant du système en boucle fermée soit stable et plus robuste.

II.4.8.1 Boucle interne de régulation du courant :

Le régulateur de courant direct permet de définir la tension V, on aura :

$$FT_{elect} = \frac{1}{L.p + R} \tag{II.5}$$

La boucle de régulation du courant peut se représenter par le schéma bloc suivant :

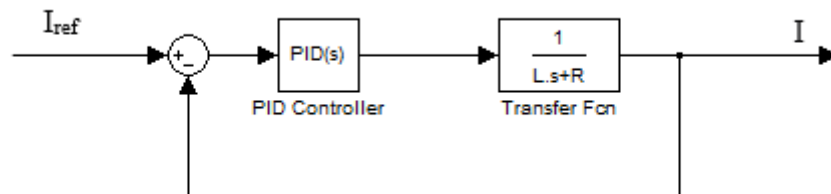


Figure II.6 : Schéma bloc de la boucle de courant

Une fois cette commande est entrée, et à partir de l'interface GATOOL, l'algorithme génétique effectue une itération jusqu'à ce qu'il réponde aux critères décrits par sa fonction de terminaison.

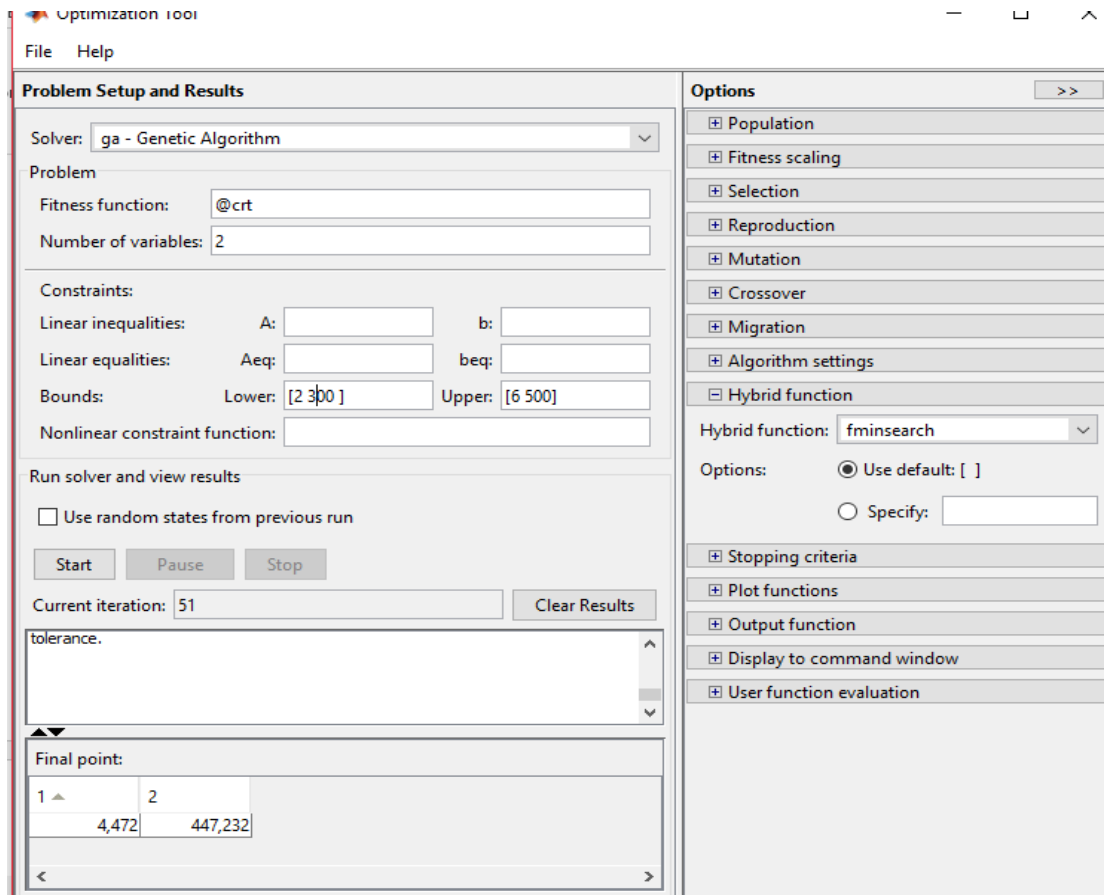


Figure II.7 : Interface GATool pour l'assertivement du courant

- **Fitness fonction :** est la fonction objective que nous souhaitons minimiser. Nous pouvons spécifier la fonction comme un descripteur de fonction de la forme @crt, où crt.m est un fichier M qui renvoie un scalaire.
- **Nombre de variables :** est le nombre de variables indépendantes pour la fonction fitness.
- **Bornes :** sont les limites inférieure et supérieure pour les variables:
 - Lower (Inférieur)= spécifie les limites inférieures en tant que vecteur.
 - Upper (Supérieur)=spécifie les limites supérieures en tant que vecteur.

Une fois l'algorithme génétique est terminé, il renvoie les variables K_p et K_i

Nous avons effectué plusieurs expériences pour déterminer ces paramètres.

→ $K_p = 4.47$ et $K_i = 447.23$

→ $PI = 4.47 + \frac{447.25}{p}$

La réponse de système avec ces paramètres est :

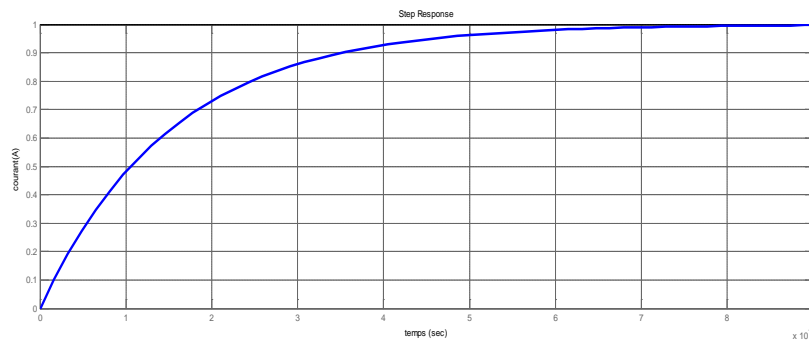


Figure II.8 : Réponse en courant à partir des paramètres du régulateur PI_{AG}

II.4.8.2 Boucle externe de régulation de la vitesse:

Le régulateur de vitesse permet de déterminer le courant de référence; soit l'équation mécanique donnée :

$$FT_{mec} = \frac{1}{Jp + f} \tag{II.6}$$

La boucle de régulation de la vitesse peut se représenter par le schéma bloc suivant :

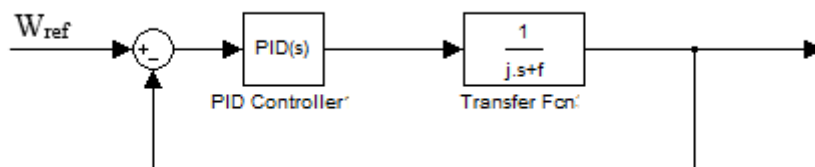


Figure II.9 : Schéma bloc de la boucle de vitesse

Une fois cette commande est entrée, et à partir de l'interface GATOOL, l'algorithme génétique effectuera une itération jusqu'à ce qu'il réponde aux critères décrits par sa fonction de terminaison.

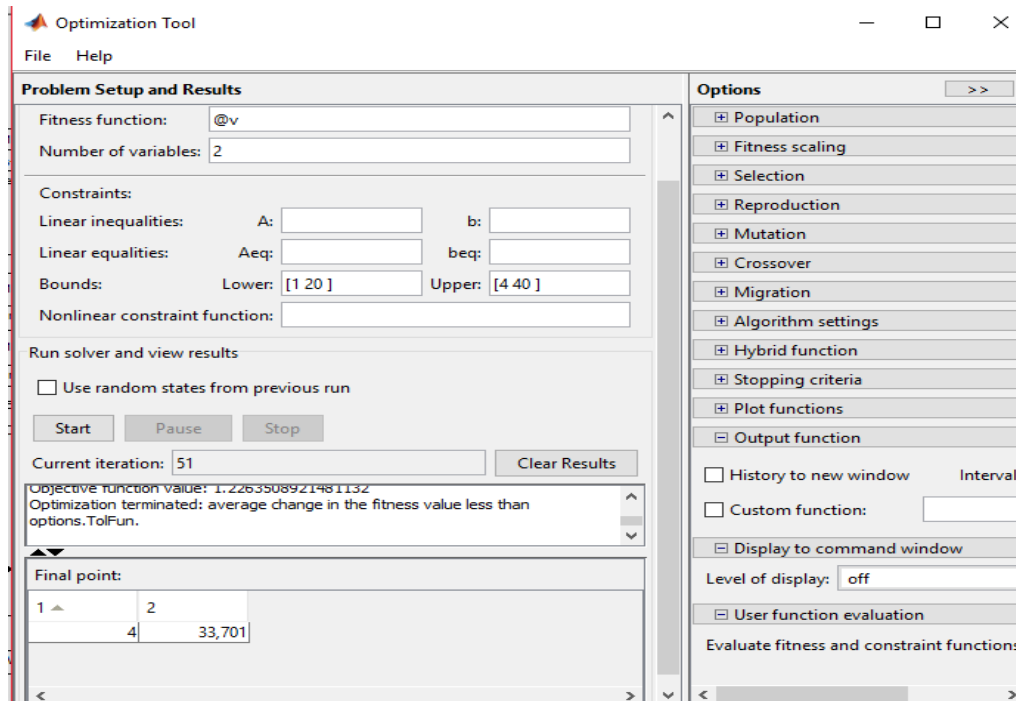


Figure II.10 : Interface GATOOL pour l'asservissement de la vitesse

Une fois l'algorithme génétique est terminé, il renvoie les variables K_p et K_i

Nous avons effectué plusieurs expériences pour déterminer ces paramètres.

→ $K_p = 4$ et $K_i = 33.7$

→ $PI = 4 + \frac{33.7}{p}$

La réponse du système en vitesse avec ces paramètres :

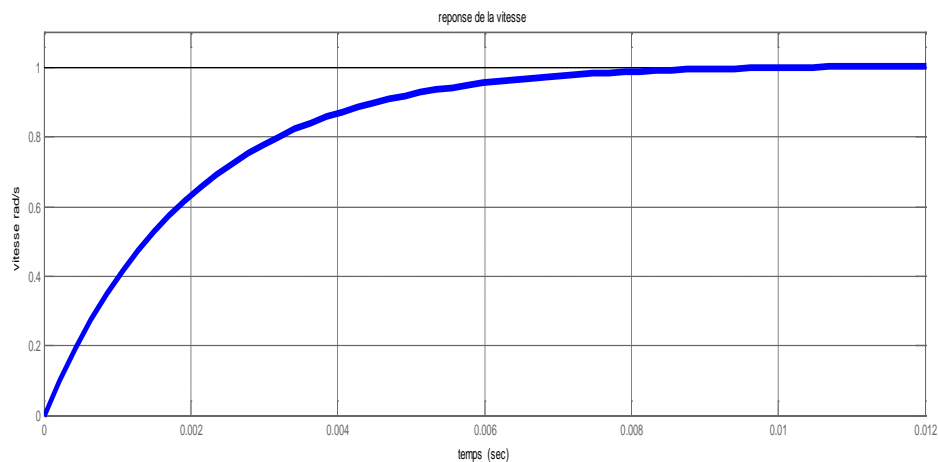


Figure II.11 : Réponse en vitesse à partir des paramètres du régulateur PI_{AG}

II.4.8.3 Application des paramètres du régulateur PI optimisés sur le modèle de la MCC :

Après avoir obtenu les différentes valeurs du régulateur optimisées, on a relevé les valeurs des paramètres : $Kp = 4.47$, $Ki = 447$ pour l'asservissement du courant et $Kp = 4$, $Ki = 33.7$ pour l'asservissement de la vitesse, on applique ces résultats sur la machine à courant continu.

II.4.8.3.1 Résultats de simulation

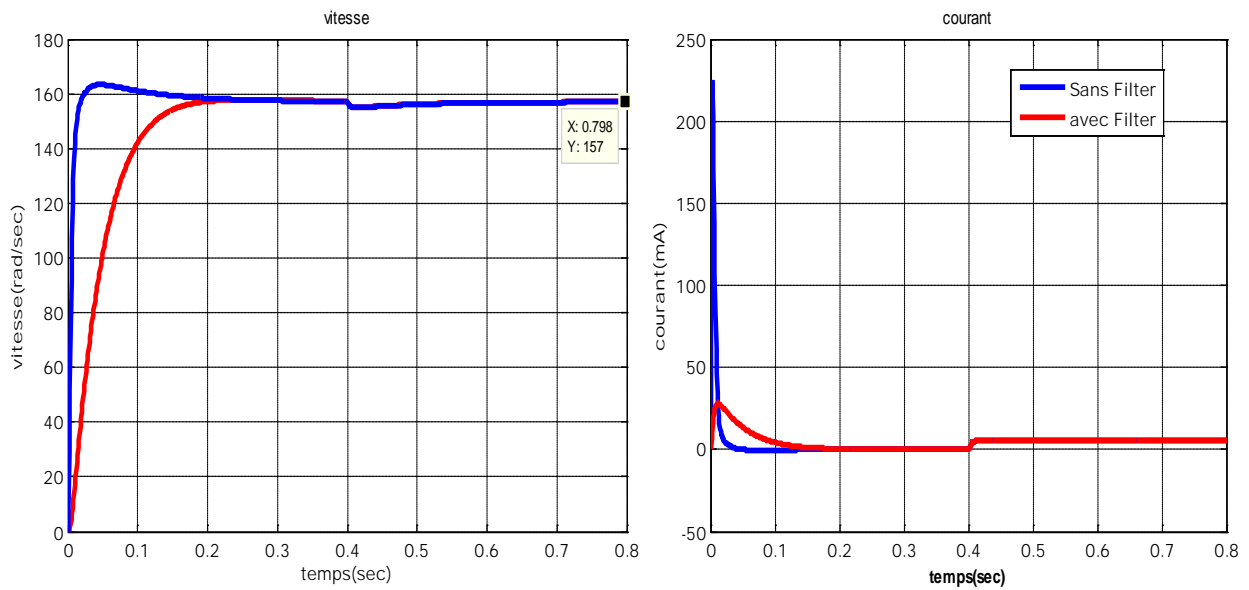


Figure II.12 : Réponses en vitesse et en courant optimisées par l'AG

❖ Interprétation :

- **Vitesse :** L'allure de la vitesse sans filtre possède une caractéristique presque linéaire et atteint la vitesse de référence dans un temps de réponse $t_r = 0.02 \text{ sec}$ et un temps de montée $t_m = 0.01 \text{ sec}$. avec un dépassement de 8%. Après l'application de la charge à l'instant $t=0.3 \text{ sec}$ de 5Nm, on ne constate presque aucune influence sur l'allure de la vitesse, et L'allure de la vitesse avec filtre possède une caractéristique exponentielle et atteint la vitesse de référence dans un temps de réponse $t_r = 0.15 \text{ Sec}$ et un temps de montée $t_m = 0.1 \text{ sec}$. avec un dépassement de 0%. Après l'application de la charge à l'instant $t=0.3 \text{ sec}$ de 5Nm,

on ne constate presque aucune influence sur l'allure de la vitesse par le régulateur PI.

- **le courant** : L'allure de courant sans filtre subit au moment du démarrage un grand pic de 400A contrairement l'allure de courant avec filtre subit un pic de 39A puis atteint rapidement la valeur nulle, au moment d'application de la charge, le courant a augmenté pour équilibrer la charge (couple résistant),

II.4.8.3.2 conclusion

Nous pouvons conclure que les AG sont des algorithmes simples de conception et peuvent résoudre des problèmes assez complexes. La résolution de ces problèmes est obtenue grâce aux opérateurs de reproduction. Ces AG sont des procédures assez robustes pour résoudre un problème d'optimisation pour la sélection des primitives. Nous avons observé que certains résultats obtenus sont mauvais à cause d'une mauvaise initialisation du domaine de recherche ou par un mauvais choix de paramètres. Lors de l'initialisation du domaine de recherche, il est difficile d'atteindre la solution. Pour déterminer des bons paramètres et un bon choix des méthodes de reproduction, il est nécessaire d'effectuer plusieurs expériences.

II.5 la logique floue

La logique floue constitue aujourd'hui une approche alternative intéressante. Elle présente plusieurs avantages tels que le raisonnement proche de celui de l'homme, sa capacité à contrôler des performances dynamiques et ses qualités intéressantes de robustesse. La logique floue inspire ses caractéristiques du raisonnement humain. Elle est basée sur la constatation que la plupart des phénomènes ne peuvent être représentés à l'aide de variables booléennes qui ne peuvent prendre que deux valeurs 0 ou 1.

La logique floue est donc un outil pour la manipulation et la représentation des connaissances imparfaitement décrites, ses objets sont modélisés par des ensembles flous support de base des notions vagues.

Enfin, elle présente aussi l'intérêt d'être plus facile à implémenter qu'une logique probabiliste.

II.5.1 Principe de la logique floue :

Présenté par le philosophe MAX BLACK dès 1937, le concept de la logique floue fut réellement introduit en 1965 par Lotfi Zadeh, un professeur d'électronique à l'université de Berkeley (USA), sa théorie `` fuzzy set theory `` n'eut pas un succès immédiat, elle fut développée surtout en Europe et au Japon. L'idée de base a pris naissance lorsqu'on a constaté la difficulté de programmer un automate en vue de sa réalisation d'une tâche, cependant jugée simple à réaliser par un être humain.

Le mode de raisonnement humain et le moyen de formaliser la connaissance humaine dans un langage accessible à une machine constituent, donc, les deux principaux sujets de réflexion qui ont conduit à l'apparition de la logique floue, freinée par la réticence à appréhender un nouveau concept et par des difficultés de réalisation pratique, la logique floue ne débouche réellement qu'en 1990 avec l'apparition de produits et de systèmes utilisant cette technique [16]-[17]. Aujourd'hui, la logique floue est arrivée à maturité et est utilisée dans de nombreux produits destinés au `` grand public `` notamment au Japon. Sa mise en œuvre est maintenant facilitée par la disponibilité des microprocesseurs dédiés et d'outils puissants de développement.

Le concept de la logique floue vient de la constatation que la variable booléenne, qui ne peut prendre que deux valeurs (vrais ou faux) est mal adaptée à la représentation de la plupart des phénomènes courants. En effet, cette logique floue classique considère qu'une proposition est soit vraie soit fausse. Cela permet de répondre à de nombreuses situations. En revanche, la logique floue distingue une infinité de valeurs de vérité (entre 0 et 1) [18]. Prenons l'exemple d'un système de chauffage. S'il fait froid, il chauffe, sinon il ne chauffe pas. Ensuite, il faut définir 'froid' par exemple, en dessous de 18°C. Donc, le système de chauffage va chauffer à pleine puissance jusqu'à 18°C puis s'arrêter car 'froid' n'est plus 'vraie'. Quelque temps plus tard, la température étant descendue en dessous de 18°C, il va repartir à pleine puissance pour quelques minutes. Un tel cycle peut être évité grâce à la logique floue en définissant 'très froid' en dessous de 10°C, le chauffage chauffe alors à pleine puissance et 'non froid' au-dessus de 20°C, température à laquelle le chauffage s'arrête. Entre 10°C et 20°, le chauffage va recevoir une instruction de chauffage modérée adaptée à la température, le chauffage va alors se stabiliser à une température constante apportant plus de confort qu'une alternance de cycle de chauffage. Lorsque plusieurs paramètres doivent être considérés (par exemple le taux d'humidité ou l'heure de la journée), la logique floue apporte aussi une réponse plus adaptée

Université Abbes laghrour khenchela

car un chauffage traditionnel programmé pour démarrer en dessous de 15°C ou à 18h ne va pas démarrer à 17h59 s'il fait 15.1°C et vous aurez en rentrant alors qu'un chauffage fonctionnant sur le principe de la logique floue comprendra qu'un démarrage modéré est nécessaire.

II.5.2 Le régulateur flou [19]

Lorsque le nouveau concept de la logique floue a été proposé, ses premières applications étaient dans le domaine du contrôle des systèmes des experts humains. Le contrôle de ces systèmes fait apparaître deux types d'information :

- Des informations numériques obtenues par les mesures des capteurs
- Des informations linguistiques obtenues par les experts humains

Le contrôle flou utilise la logique floue comme une démarche qui peut couvrir la stratégie du contrôle linguistique. Il est intégré dans la partie qui gère les données de commande et de contrôle de processus du système, appelée régulateur flou, Figure II.15.

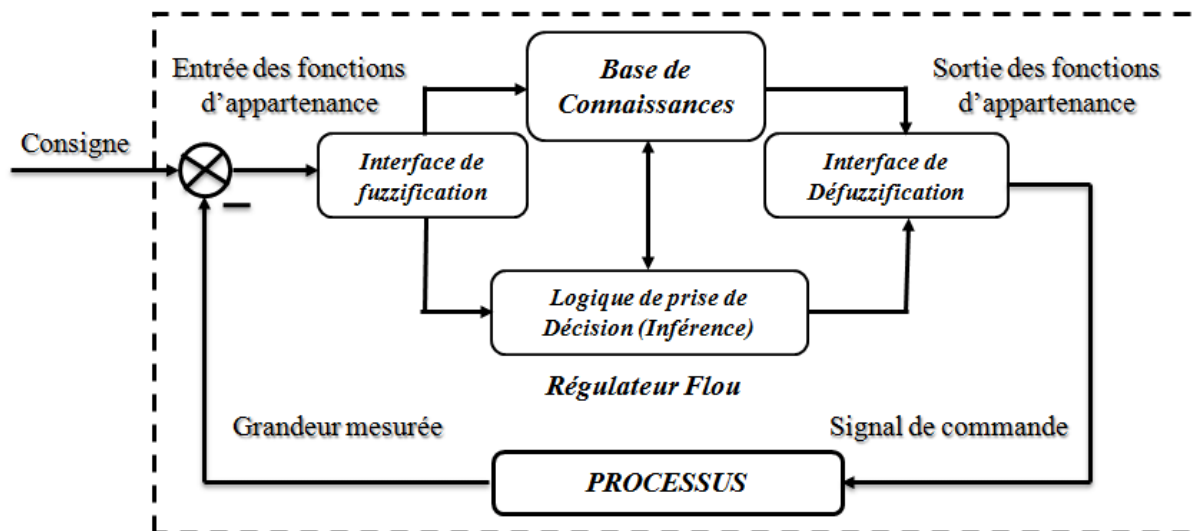


Figure II.13 : Système à commande floue

II.5.3 Configuration générale d'un Régulateur flou [20]

La logique floue est souvent associée à la notion de régulateur et plus particulièrement de régulateur flou (en anglais Fuzzy Logic Controller ou FLC). Le modèle flou permet de relier à la fois les observations scientifiques et l'expérience de l'opérateur à des représentations symboliques et qualitatives.

Un contrôleur flou est composé de quatre blocs principaux :

- Base de connaissance.
- Interface de fuzzification.
- Mécanisme d'inférence.
- Interface de défuzzification.

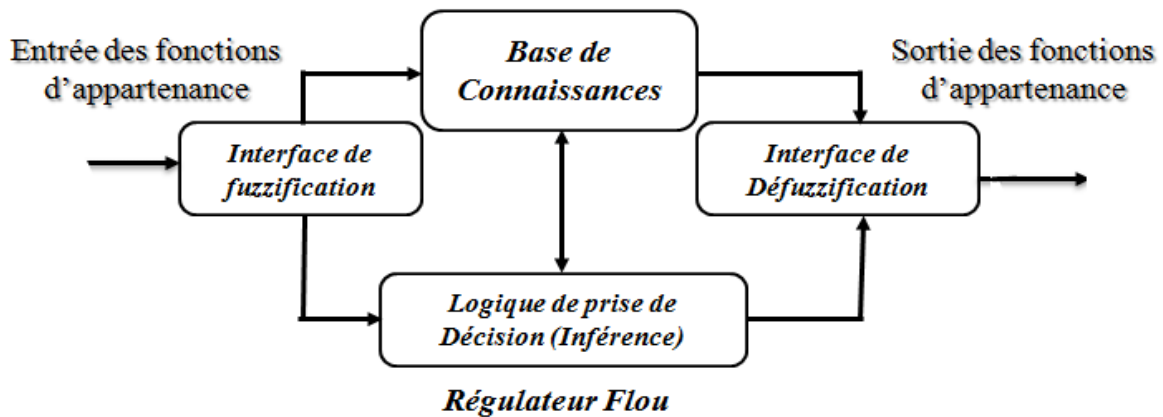


Figure II.14 : Structure interne d'un système flou

II.5.2.1 La fuzzification [21]

La fuzzification est le premier traitement qui entre en compte dans la structure d'un contrôle régi par la logique floue, durant cette phase, les informations issues du système sont tout d'abord normalisées. Ensuite, les données normalisées sont transformées en qualifications linguistiques.

Les entrées dans un système flou sont en général mesurées à l'aide d'organes de mesures qui sont le plus souvent de type analogique. Étant donné que, l'implémentation du système flou se fait toujours en numérique, il faut d'abord convertir les entrées analogiques en digital, puis on procède à la fuzzification. L'adaptation des entrées permet de garantir l'appartenance de celles-ci à l'univers de discours choisi. La dernière étape de fuzzification est la génération des degrés d'appartenance à chaque valeur linguistique définie par sa fonction d'appartenance.

II.5.2.2 Base de connaissance

Elle contient les définitions des fonctions d'appartenance (formes et paramètres) associées aux variables d'entrée/sortie ainsi que l'ensemble des règles floues. Le problème initial est alors décomposé en un ensemble de règles, de la forme :

“SI variable EST propriété, ALORS action”

Qui définit la réponse désirée du système en sortie pour les conditions d'entrée du système. Le nombre et la complexité des règles dépendent du nombre de paramètres d'entrée qui doivent être utilisés et du nombre de variables floues associées à chaque paramètre. Dans le cas où plusieurs données d'entrée sont retenues, les règles vont s'exprimer par exemple comme suit :

Si x_1 est A1 ET x_2 est A2 alors y est B

Si x_1 est A1 OU x_2 est A2 alors y est B

Où x_1 , x_2 et y sont les grandeurs physiques caractéristiques du système, A1, A2 et B sont les termes linguistiques, « OU » et « ET » les opérateurs de liaison des fonctions d'entrée.

II.5.2.3 Mécanisme d'inférence [22] [23]

Considéré comme le « cerveau » du contrôleur, il permet de lier les degrés d'appartenance des fonctions d'appartenance d'entrée aux fonctions d'appartenance de sortie. Le degré d'appartenance de la fonction de sortie peut être calculé par différentes méthodes :

- La méthode d'inférence min-max.
- La méthode d'inférence max-prod.
- La méthode d'inférence somme-prod

II.5.2.3.1 Méthode Min-Max

C'est la méthode la plus universelle mais qui n'est guère applicable en raison du temps de calcul très long. Cette méthode réalise, au niveau de la condition, l'opérateur OU par la formation du maximum et l'opérateur ET par la formation du minimum. La conclusion dans chaque règle, introduite par ALORS, lie le facteur d'appartenance de la condition avec la fonction d'appartenance de la variable de sortie est réalisé par l'opérateur ET, réalisé dans le cas présent par la formation du minimum.

Enfin l'opérateur OU qui lie les différentes règles est réalisé par la formation du maximum.

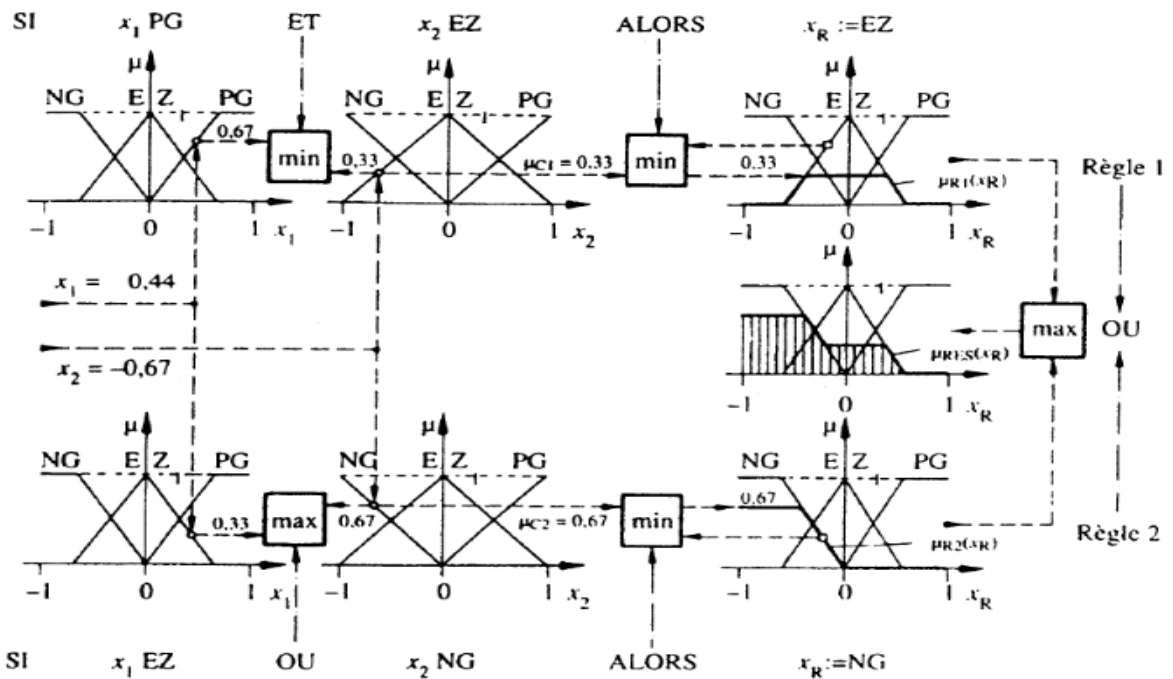


Figure II.15: Méthode d'inférence min-max

II.5.2.3.2 Méthode d'inférence max-produit

La différence avec la méthode précédente est la réalisation d' ALORS qui se traduit par la multiplication de la fonction d'appartenance considérée par la valeur de la règle

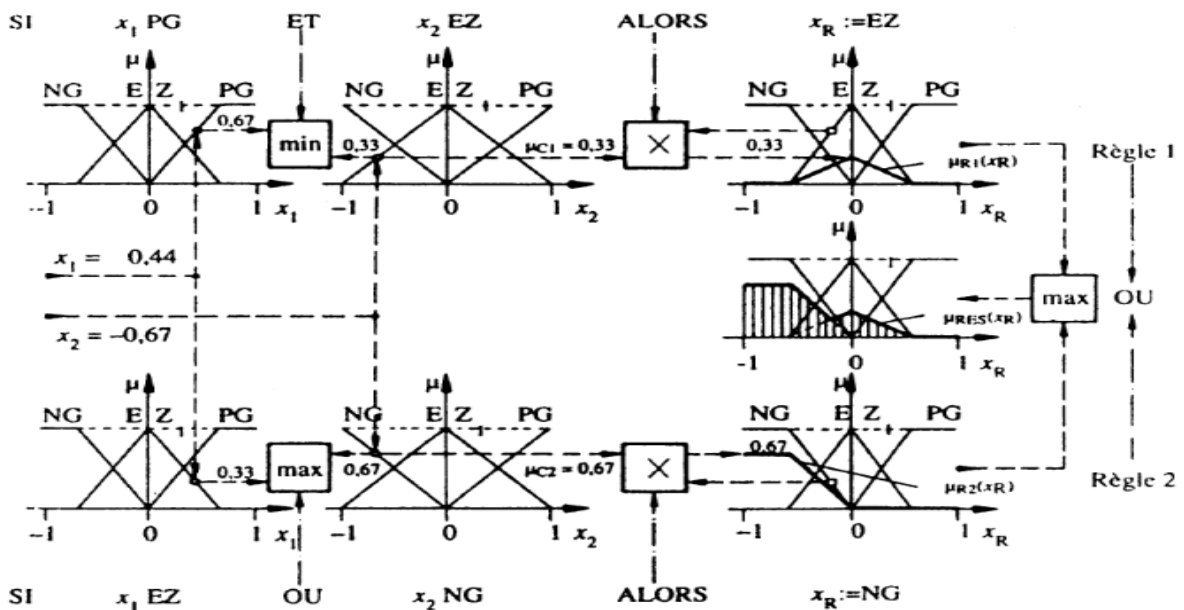


Figure II.16: Méthode d'inférence max-produit

II.5.2.3.3 Méthode d'inférence somme-produit

Proche de la méthode précédente, il suffit de remplacer la traduction d' ALORS par la somme des fonctions d'appartenance partielles.

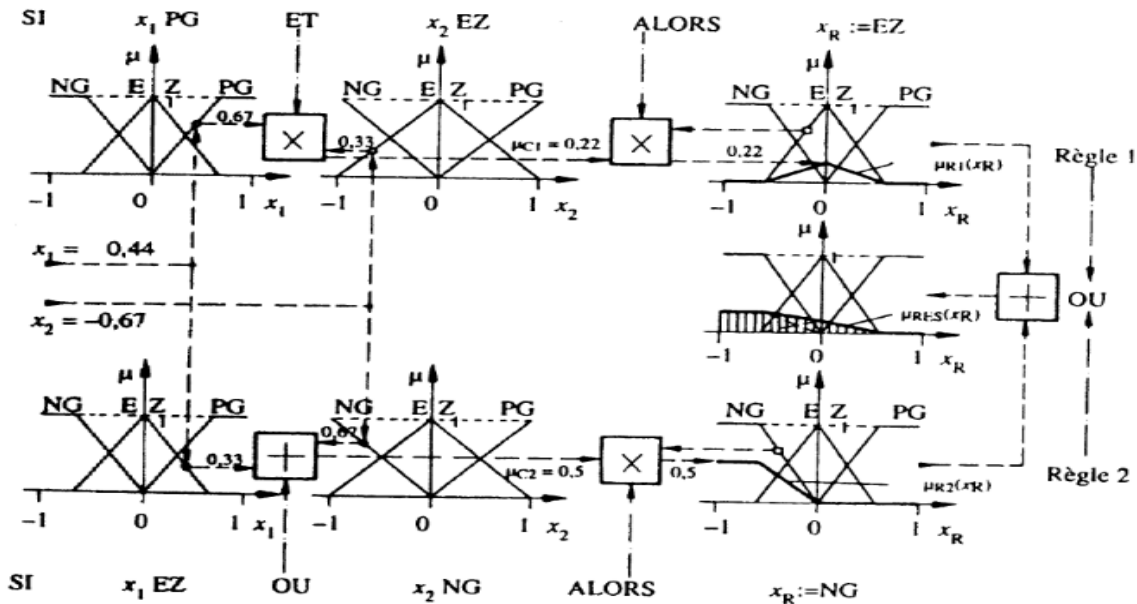


Figure II.17 : Méthode d'inférence somme-produit

II.5.2.4 Défuzzification

Pour bien définir la loi de commande, le contrôleur flou doit avoir une procédure de défuzzification jouant le rôle de la conversion de la commande floue en valeur physique pour chaque état du processus. Il existe plusieurs stratégies de défuzzification dont les plus utilisées sont :

- méthode du maximum ;
- méthode de la moyenne des maxima ;
- méthode du centre de gravité ;

II.5.2.4.1 Méthode du maximum

La sortie correspond à l'abscisse du maximum de la fonction d'appartenance résultante.

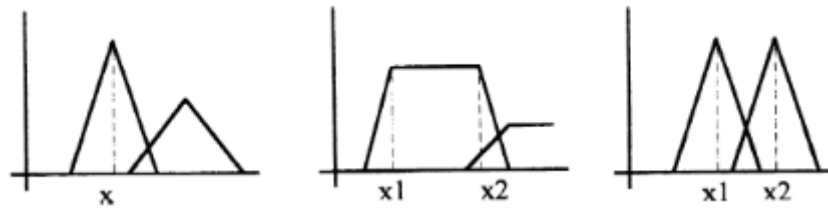


Figure II.18 : Exemple sur la méthode du maximum

Dans le cas où plusieurs sous-ensembles auraient la même hauteur maximale, on réalise leur moyenne une des ambiguïtés de la méthode du maximum est enlevée.

II.5.2.4.2 Méthode du centre de gravité

C'est la méthode la plus utilisée qui consiste à déterminer le centre de gravité de la fonction d'appartenance résultante $\mu_A(x)$

Dans ce contexte, il suffit de calculer l'abscisse U du centre de gravité comme suit :

$$CG = \frac{\sum_{x=a}^b \mu_A(x) \cdot x}{\sum_{x=a}^b \mu_A(x)} \quad (\text{II.7})$$

Où n est le nombre des niveaux de la sortie du contrôleur. Par cette méthode, on aboutit à une solution unique.

II.5.3 Etapes de Conception d'un contrôleur flou pour le réglage de la vitesse

Le choix des variables d'entrée et de sortie est la première étape à mener pour la conception d'un contrôleur flou. Les étapes suivantes : la normalisation de ces variables dans un domaine bien défini, la fuzzification, l'inférence et son traitement et finalement la défuzzification.

II.5.3.1 Choix des variables d'entrées et de sorties

Pour le réglage de la vitesse de la machine courante à continu, nous avons choisi comme variables d'entrées l'erreur 'e' qui est la différence entre la vitesse de référence Ω_{ref} imposée par l'opérateur, et la vitesse de rotation du rotor de la machine. L'autre variable d'entrée est la dérivée de l'erreur qui représente la différence entre l'erreur actuelle et l'erreur

précédente. Comme variable de sortie, une seule variable de commande qui est le couple de référence (C_{ref}) est choisie [24].

II.5.3.2 Normalisation des variables d'entrée et de sortie

La normalisation se fait par une division des variables e , Δe et U respectivement par les coefficients K_e et $K_{\Delta e}$, K_u de telle sorte qu'elles évoluent dans le domaine $[-1 \ 1]$ pour toutes les variables.

II.5.3.3 Interface de fuzzification

La fuzzification représente d'une part le choix de l'univers de discours des variables linguistique. Ce choix est généralement basé sur l'expérience de l'opérateur. L'intervalle de l'univers de discours de l'erreur est choisi entre $[-1 \ 1]$, pour la dérivée de l'erreur entre $[-1 \ 1]$ et pour la sortie entre $[-1 \ 1]$.

D'autre part la fuzzification consiste à choisir la forme des fonctions d'appartenance. Pour une raison de simplification en vue d'avoir le même effet de réglage dans les sens de rotation, les fonctions d'appartenance de forme triangulaire et trapézoïdale avec intersection de 50 % placées symétriquement par rapport à zéro sont utilisées. Le nombre des ensembles flous est cinq pour toutes les variables linguistiques qui sont successivement notées.

NG : Négatif Grand ; **N** : Négatif ; **EZ** : Environ Zéro ; **P** : Positif ; **PG** : Positif Grand.

II.5.3.4 Inférence

L'inférence ou la prise de décision est le noyau du contrôleur flou, elle est capable de simuler la prise de décision de l'être humain en se basant sur les concepts flous et les règles d'inférence, les règles d'inférence utilisées sont de type :

Si e est x_1 ET Δe est x_2 ALORS $ref \ C$ est x_3

Le nombre des ensembles flous pour e et Δe est cinq, donc il faut $5 \times 5 = 25$ règles d'inférence. Sachant qu'il n'existe pas une loi bien déterminée pour la détermination de la décision de chaque règle. L'expérience humaine, dans ce cas joue un rôle très important.

Chapitre II : Commande MCC par l'intelligence artificielle

Les règles d'inférence pour notre système sont représentées par une matrice d'inférence selon le tableau :

Δe \ e	NG	N	EZ	P	PG
NG	NG	NG	N	N	EZ
N	NG	N	N	EZ	P
EZ	N	N	EZ	P	P
P	N	EZ	P	P	PG
PG	EZ	P	P	PG	PG

Tableau II.1 : Règles à cinq fonctions d'appartenance

II.5.3.5 Traitement numérique de l'inférence

Après avoir choisi les règles d'inférence, les opérateurs de la logique floue permettent de choisir une méthode pour le traitement de l'inférence. Sachant que l'opérateur ET est le minimum, l'opérateur OU est le maximum et l'opérateur ALORS est le maximum. La méthode retenue sera la méthode min/max.

II.5.3.6 Défuzzification

La défuzzification consiste à transformer la décision de sortie linguistique en une décision numérique connue pour commander le système. Pour notre cas d'étude nous avons utilisé la méthode du centre de gravité

II.5.4 Application de la commande floue sur le moteur à courant continu

Après avoir énoncé et défini les concepts de base, les termes linguistiques utilisés en logique floue et la structure d'un régulateur flou. On s'intéresse donc à implémenter la commande floue pour un moteur à courant continu, pour un seul objectif est d'annuler l'erreur statique, diminuer le dépassement, diminuer le temps de réponse et le temps de montée afin d'obtenir une réponse adéquate du procédé et de la régulation et d'avoir un système précis, rapide, stable et robuste.

II.5.4.1 Conception de contrôleur

Le toolbox fuzzy logic de MATLAB possède un éditeur qui permet de créer des systèmes d'inférences floues : des FIS (pour fuzzy inference system). Cette boîte à outils permet de générer des fichiers « .fis », qui correspondent à des systèmes d'inférences floues et dont font partie les RLF. Cette boîte à outils possède 3 éditeurs (de fis, de règles et de fonctions d'appartenance) qui permettent de saisir l'ensemble des données du FIS ainsi que deux interfaces graphiques qui permettent de visualiser les inférences directement sur la base de règles, ainsi que des surfaces de contrôle.

Les fonctions d'appartenances sont du type triangulaire. L'intervalle d'intérêt des variables d'entrées est subdivisé en cinq classes pour l'erreur «e» et en cinq classes pour sa dérivée « $\frac{de}{dt}$ », alors que celui de la variable de sortie «u» est subdivisé en cinq classes

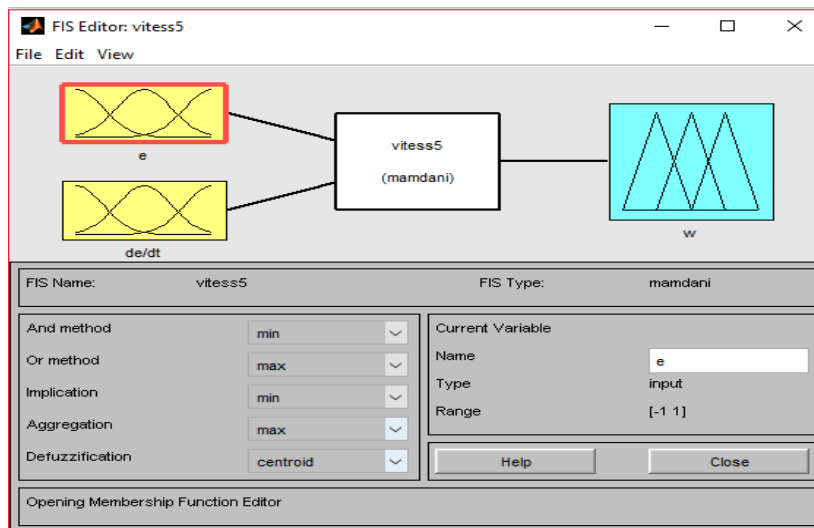


Figure II.19 : Définition de la structure globale (entrée, sortie, type d'opérateurs)

Grâce à cette interface, on définit les plages d'entrées et de sorties de façon à contraindre le comportement du bloc Fuzzy, illustré par la FigureII.23, pour définir ces fonctions :

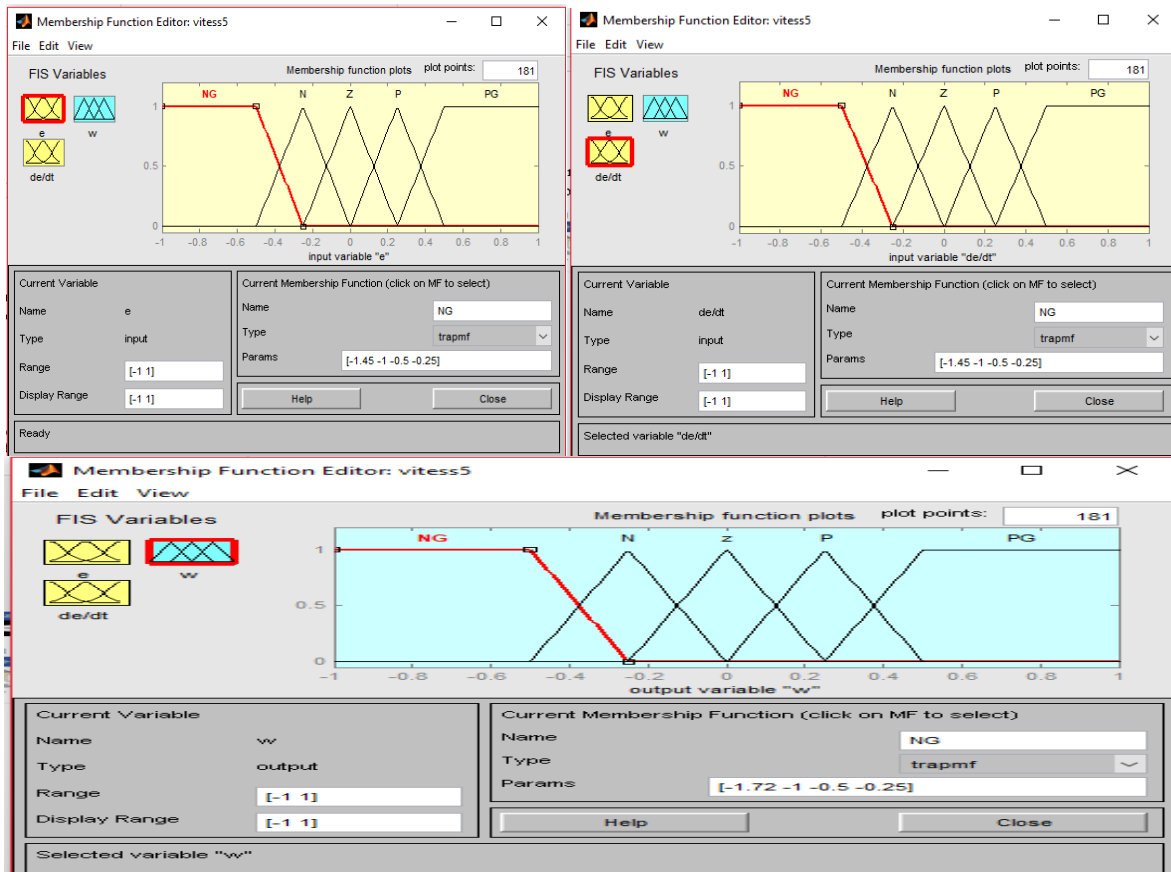


Figure II.20: Définition des fonctions d'appartenance pour 'e', 'de/dt' et 'u'

Il est également possible de configurer les lois du bloc flou, le principe celui d'actions conditionnelles «if » représenté sur la Figure II.24.

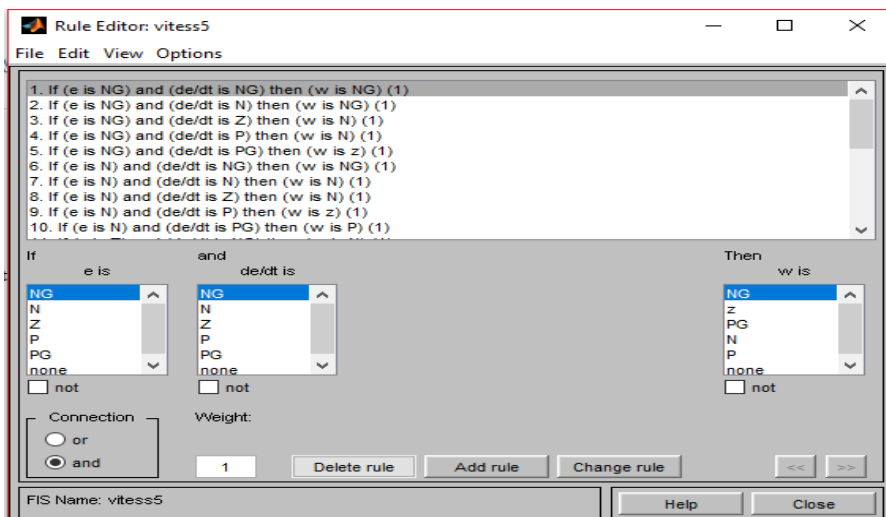


Figure II.21: Écran de saisie des règles

La sortie est ensuite déterminée suivant les poids respectifs des différentes lois d'entrées. En effet les lois se chevauchent, et donc une valeur de sortie est fonction de plusieurs lois d'entrées sur une même grandeur.

II.5.4.2 Schéma bloc de simulation de la MCC avec le contrôleur flou :

➤ Régulation en vitesse :

Pour illustrer les performances de réglage par la logique floue, nous avons simulé le schéma bloc de la commande en vitesse d'un moteur à courant continu qui est représenté par la Figure II.25, on conserve le régulateur de courant classique

Les valeurs des gains du régulateur sont choisis après plusieurs tests d'ajustement

$$K_e = \frac{1}{250} ; K_{\Delta e} = 1/9 \times 10^3 ; u = 25$$

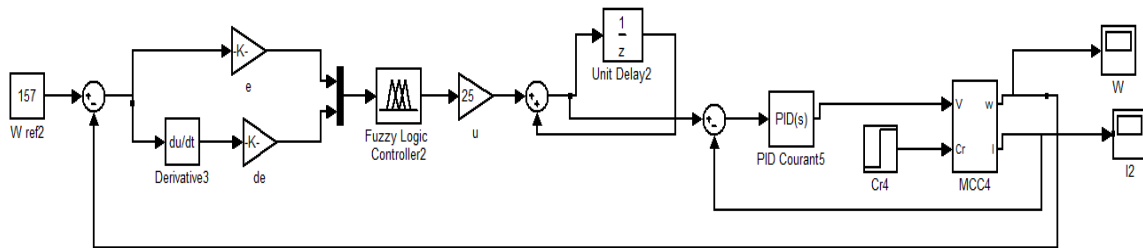


Figure II.22: Schéma bloc d'un asservissement de vitesse par RLF

II.5.4.3 Résultats de simulation et discussion

Une fois le régulateur est chargé dans le bloc fuzzy du simulink, on simule le système et on obtient la réponse en vitesse, Figure II.26 et en courant ,Figure II.27

Pour illustrer les performances de réglage par la logique floue, nous avons simulé un démarrage à vide, puis en charge avec un couple résistant à $t = 0.3$ sec

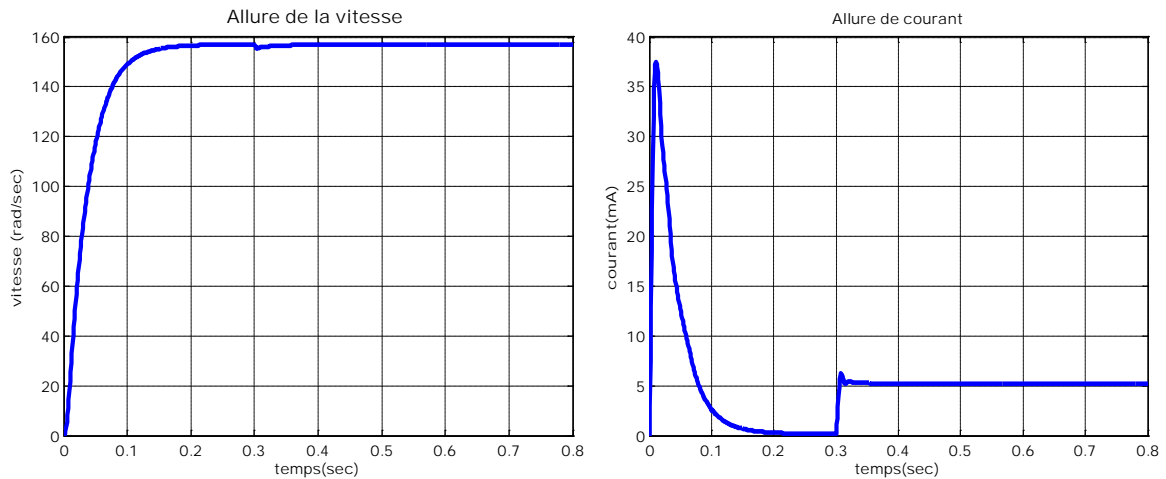


Figure II.23 : Réponses en courant et en vitesse.

❖ **Interprétation :**

L'allure de la vitesse possède une caractéristique presque linéaire et atteint la vitesse de référence dans un temps de réponse très petit $t_r = 0.15 \text{ Sec}$ et un temps de montée $t_m = 0.8 \text{ sec}$. avec un dépassement de 0% Après l'application de la charge à l'instant $t=0.3 \text{ sec}$ de 5Nm, on ne constate presque aucune influence sur l'allure de la vitesse, le courant subit au moment du démarrage un pic puis atteint rapidement la valeur nulle, au moment d'application la charge le courant à augmenter pour maintenir la charge (couple résistant)

➤ **Remarque :**

On remarque qu'à chaque fois on intervient sur les gains d'entrées de notre RLF on agit sur la rapidité du système et à chaque fois qu'on intervient sur le gain de la sortie du système on agit sur la précision

II.5.4.4 Conclusion :

Les résultats de la simulation nous ont permis de juger les qualités de la commande à base du régulateur flou. A travers les caractéristiques de la réponse, on observe de bonnes performances même en présence des perturbations grâce à la robustesse du régulateur flou, Ce qui nous a permis de conclure que la logique floue est un moyen très performant pour l'amélioration de la stabilité du moteur à courant continu.

II.6 Comparaison entre les différentes méthodes de régulation

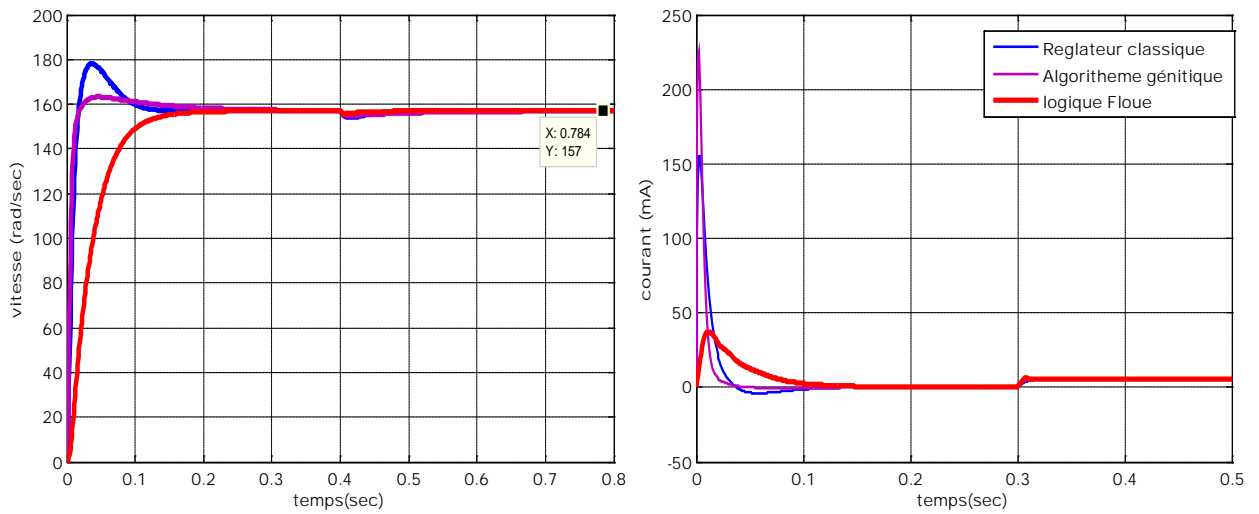


Figure II.24 : Allures de la vitesse et du courant (Comparaison entre les différentes méthodes PI, BLF, AG)

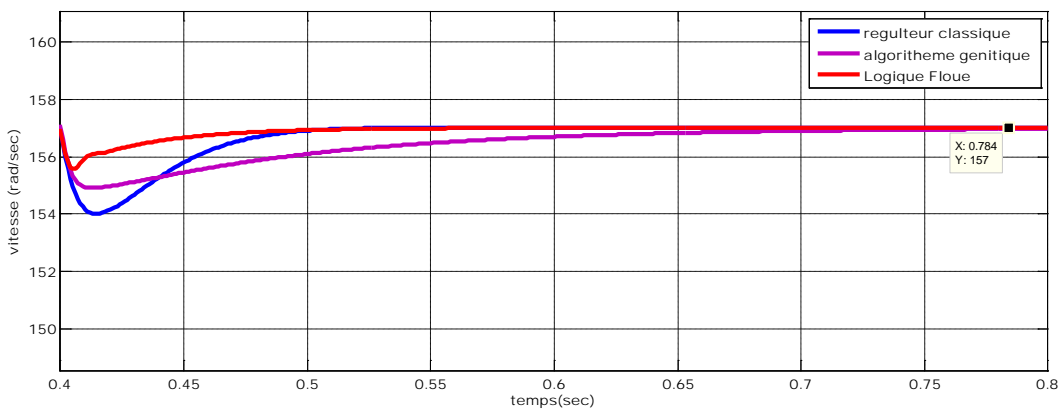


Figure II.25 : Allures de la vitesse au moment d'application de la perturbation

Avec filtre :

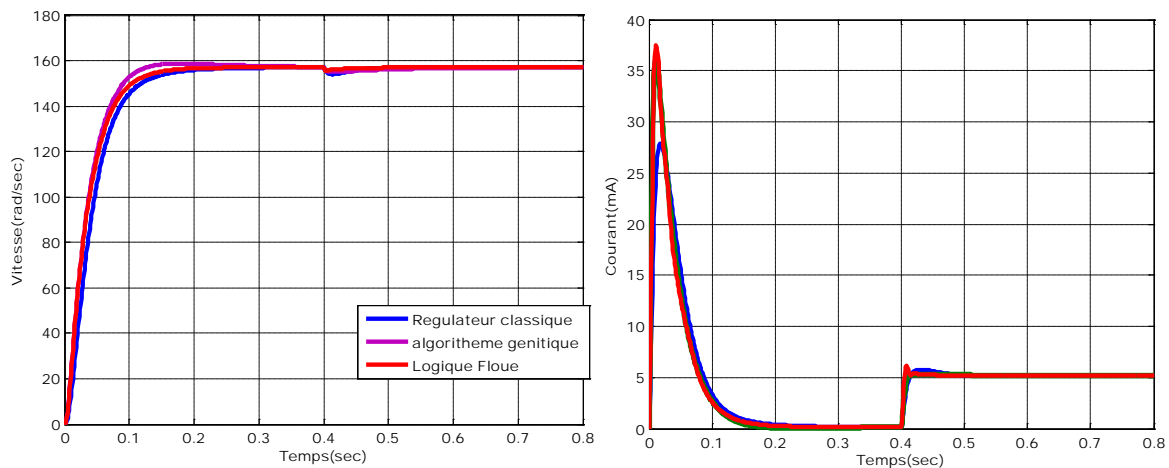


Figure II.26 : Allures de la vitesse et du courant avec filtre

Nous avons évalué les réponses du système à un échelon de vitesse et nous observons le comportement du système. tableau IV.2 résume les résultats obtenus.

	PI classique	PID AG	La logique Floue
Dépassement %	13.37%	3.8%	0%
Temps de reponse (Sec)	0.09	0.015	0.09
Temps de rejet (Sec)	0.08	0.15	0.05
Erreur %	0	0	0

Tableaux II.2 : Résumé des résultats des simulations

❖ **Interpritation**

Si on compare les résultats obtenus avec le régulateur à algorithmes génétiques à ceux de la logique floue et du régulateur classique PI :

Les résultats de simulation des algorithmes génétiques et la logique floue obtenus montrent l'amélioration de la réponse dynamique de la vitesse. Cette dernière atteint sa référence en un temps de réponse (temps de réponse $tr_{AG} = 0.015\text{Sec}$ et logique floue $tr_{LF} = 0.09\text{Sec}$). De plus, l'allure des courants marque des pics respectivement inférieurs à ceux marqués dans le cas du PI classique. Au démarrage et lors de l'application d'un couple de charge nominal. Lorsque nous adoptons les méthodes de commande en vitesse intelligentes, nous obtenons une réponse en vitesse qui suit parfaitement la référence demandée. Nous voyons que les impacts du couple de charge n'affectent pas la réponse en vitesse du système.

II.7 Conclusion :

Ce chapitre constitue une contribution à la mise en œuvre des techniques non conventionnelles de commande numérique. Nous avons ainsi élaboré deux stratégies de contrôle de vitesse de machines à courant continu basées sur l'algorithme génétique et la logique floue . Les résultats obtenus nous permettent de montrer que ces techniques, jusqu'à présent peu utilisées, surtout en milieu industriel, sont bien adaptées lorsqu'on a des contraintes sévères au niveau de la régulation. Les systèmes de contrôle de vitesse des machines à courant continu utilisant l'algorithme génétique ou la logique floue présentent des performances intéressantes

Diverses méthodes permettent d'optimiser les coefficients du PI. Mais le régulateur idéal n'existe pas, chaque optimisation d'un paramètre (précision, ...) se fait aux dépens d'un autre. Il convient alors de trouver le meilleur compromis en fonction des exigences initiales du cahier de charge.

III.1 Introduction

Ce chapitre est consacré à l'étude et à la réalisation pratique d'un hacheur de puissance (hacheur série) et d'une carte de commande (maquette feedback) à base d'arduino et du logiciel Matlab en vue d'étudier le comportement et l'asservissement numérique d'un moteur à courant continu.

III.2 Identification des paramètres du moteur

D'après ce qui précède dans le chapitre 1, le modèle d'un moteur à courant continu peut se caractériser par cinq grandeurs :

- Résistance R.
- Inductance L.
- Constante électromotrice (ou constante du couple) K.
- Moment d'inertie J
- Coefficient de frottement f

Ces paramètres sont généralement donnés par le constructeur. Toutefois, il peut être nécessaire de les caractériser [25].

III.2.1 Mesure des paramètres électriques

III.2.1.1 La résistance de l'induit Ra

Après avoir fait tourner le moteur pendant 3 mn, nous le déconnectons de l'alimentation et branchons un ohmmètre à ses bornes. On obtient la valeur de la résistance d'induit $R_a=12\Omega$.

III.2.1.2 L'inductance de l'induit La

La s'obtient simplement par la mesure de l'inductance via un PONT DE MESURE RLC METRE.

$L_a=10.7\text{mH}$.

III.2.1.3 La constante K

K est présent dans deux équations :

- l'équation (III.1) joignant la force électromotrice à la vitesse de rotation.

$$e(t) = K_e \Omega(t) \quad (\text{III.1})$$

- l'équation (III.2) joignant le couple au courant.

$$C(t) = K_c i(t) \quad (\text{III.2})$$

Il y a donc 2 manières de calculer K.

Chapitre III : Réalisation pratique d'un asservissement de vitesse

1. En calculant $e = V - Ri$ en régime établi, il faut donc connaître préalablement R et mesurer le courant i et la vitesse Ω .

2. En mesurant le couple fourni par le moteur et le courant absorbé. [25]

On détermine le constante K à partir d'un essai à vide effectué sur le moteur alimenté par la tension V .

Pour une tension d'alimentation $V=20V$ on aura $i=0.061A$ et $N=6000tr/mn$

En calculant :

$$e = V - Ri = 20 - (12 \times 0.061) = 19.268V \quad (\text{III.3})$$

Et à partir de l'équation (III.1) : $e = K\Omega$

$$K = \frac{e}{\Omega} = \frac{19.268}{\frac{6000 \times 2 \times \pi}{60}} = 0.03 V / rd/s \quad (\text{III.4})$$

III.2.2 Mesure des paramètres mécaniques

III.2.2.1 le coefficient de frottement f

La détermination de f est relativement aisée : il suffit pour cela de faire varier la vitesse de la machine (par action sur V) et de relever la valeur du courant i circulant dans la machine. A partir de l'équation (III.5) :

$$J \frac{d\Omega(t)}{dt} = C - C_r - f\Omega \quad (\text{III.5})$$

A vide le couple résistant $C_r(t) = 0$ et en régime permanent $\frac{d\Omega(t)}{dt} \rightarrow 0$ donc

L'équation (III.5) devient :

$$C = f\Omega \quad (\text{III.6})$$

Alors :

$$f = \frac{C}{\Omega} = \frac{K \times i}{\Omega} \quad (\text{III.7})$$

Pour la tension $V=20V$ on aura $i=0.061 A$ et $N=6000tr/mn$

$$f = \frac{K \times i}{\Omega} = \frac{0.03 \times 0.061}{\frac{6000 \times 2 \times \pi}{60}} = 2.81 \times 10^{-6} N. m / rd. s^{-1} \quad (\text{III.8})$$

III.2.2.2 le moment d'inertie J

Pour cela il faut faire un essai de ralentissement $\Omega=f(t)$

➤ Essais de ralentissement

On augmente la vitesse du moteur à une valeur constante, puis on la stabilise. A cet instant, on ouvre le circuit (on coupe toutes les alimentations en même temps) et on enregistre l'évolution de la vitesse par rapport au temps).

On effectue donc plusieurs essais et on calcule le temps de ralentissement, τ_{mec} correspondant à la moyenne, [26].

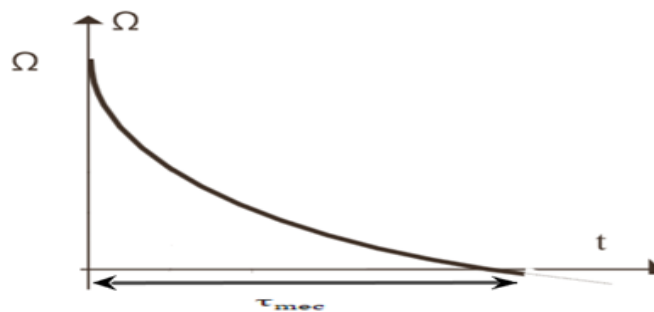


Figure III.1 : Essais de ralentissement

On mesure τ_{mec}

- 1^{er} essai :

Pour une tension $V=20V$ on aura $N=6000tr/mn$, on coupe l'alimentation $\tau_{mec} = 1.21Sec$

- 2^{eme} essai :

Pour une tension $V=19V$ on aura $N=5860tr/mn$, on coupe l'alimentation $\tau_{mec} = 1.17Sec$

A partir de cet essai :

$$\tau_{mec} = \frac{J}{f} \quad (III.9)$$

Alors :

$$J = \tau_{mec} \times f = 1.21 \times 2.81 \times 10^{-6} = 3.4 \times 10^{-6} kg/m^2 \quad (III.10)$$

D'après le peu des données constructeur que l'on avait, vitesse 11000tr/mn à 31V. On a réussi à déterminer les grandeurs du modèle équivalent du moteur.

Paramètres	Valeurs
Résistance de l'induit	$R = 12\Omega$
Inductance de l'induit	$L = 10.7\text{mH}$
Constant K	$K = 0.03 \text{ V} / \text{rd/s}$
Moment d'inertie J	$f = 2.81 \times 10^{-6} \text{ N.m/rd.s}^{-1}$
Coefficient de frottement f	$J = 3.4 \times 10^{-6} \text{ kg/m}^2$

Tableaux III.1 : Paramètres identifiés

III.2.3 Modélisation du système

Le but de cette partie est de déterminer la fonction de transfert échantillonnée de notre système en boucle ouvert notée $G(z)$. L'entrée du système est la tension $u(z)$ en volts et la sortie est la vitesse $V(z)$???

III.2.3.1 Fonction de transfert du moteur

On passe à Laplace :

$$Ri(P) + LPi(P) + e(P) = V(P) \tag{III.11}$$

$$e(P) = ke.\Omega(P) \tag{III.12}$$

$$JP\Omega(P) = C(P) - C_r(P) - f\Omega(P) \tag{III.13}$$

$$C(P) = kc.I(P) \tag{III.14}$$

En combinant (III.11) et (III.12) on obtient :

$$V(P) = Ri(P) + LPi(P) + ke.\Omega(P) \tag{III.15}$$

En combinant (III.13) et (III.14) on obtient :

$$JP\Omega(P) = kc.I(P) - C_r(P) - f\Omega(P) \tag{III.16}$$

On en déduit l'expression de $\Omega(P)$

$$\Omega(P) = \frac{kc.I(P) - C_r(P)}{f + JP} \tag{III.17}$$

On peut en sortir l'expression de $i(p)$:

$$i(P) = \frac{f + JP}{Kc} \left(\Omega(P) + \frac{C_r(P)}{f + JP} \right) \tag{III.18}$$

On l'injecte à présent dans (III.11)

Chapitre III : Réalisation pratique d'un asservissement de vitesse

$$V(P) = \Omega(P) \left(\frac{(R + LP)(f + JP)}{Kc} + Ke \right) + \frac{R + LP}{f + Jp} C_r(P) \quad (\text{III.19})$$

On suppose que le moment du couple de pertes (qui est vu comme une perturbation) est négligeable devant le moment du couple électromagnétique ($kc \cdot I(P)$) on peut alors prendre C_r nul pour simplifier le système. On a donc :

$$V(P) = \Omega(P) \left(\frac{(R + LP)(f + JP)}{Kc} + Ke \right) \quad (\text{III.20})$$

La fonction de transfert cherchée $H(p)$ est entre la tension alimentant le moteur (entrée) $V(p)$ et la vitesse acquise sur l'arbre (sortie) $\Omega(p)$, les valeurs numériques sont données dans le tableau III.1 [27], [28].

$$H(p) = \frac{V(P)}{\Omega(P)} = \frac{Kc}{(R + LP)(f + JP) + Kc \cdot Ke} \quad (\text{III.21})$$

III.2.3.2 Simulation du MCC

On peut établir le modèle mathématique de la réponse en vitesse du moteur électrique qui est donné par la figure suivante :

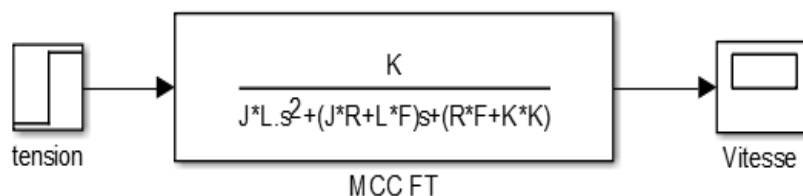


Figure III.2 : Schéma bloc du Modèle du moteur électrique.

La réponse en vitesse du moteur à courant continu obtenue par simulation est illustrée par la figure III.3 ;

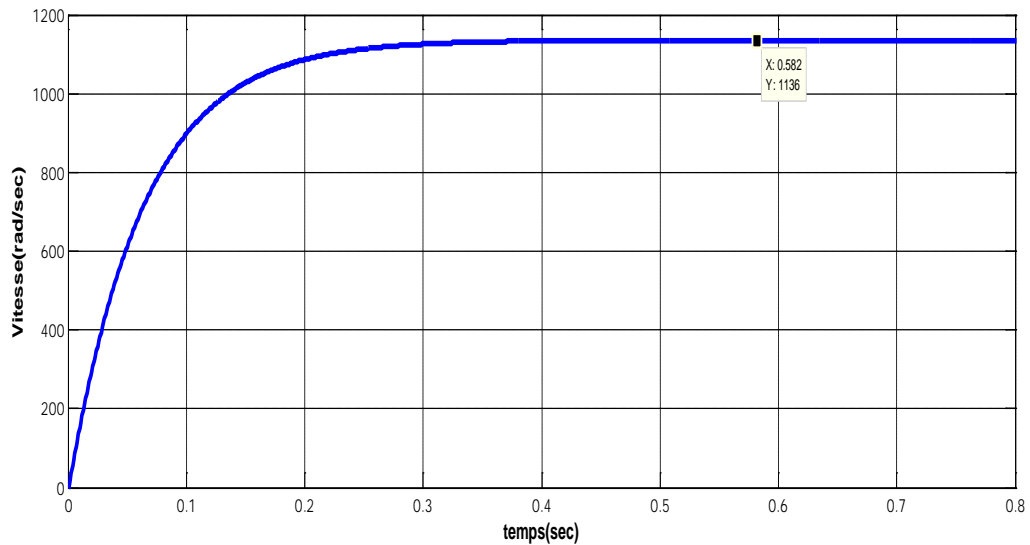


Figure III.3: Allure de la vitesse simulée.

III.3 Commande du moteur à courant continu en boucle ouverte

III.3.1 La Commande par MLI

La technique MLI (Modulation en Largeur d'Impulsion) ou PWM (Pulse With Modulation) est une technique de pilotage pour les convertisseurs statiques servant d'interface entre une charge (machine électrique, ...) et son dispositif d'alimentation (hacheur, onduleur...). Elle est donc une technique utilisée pour la conversion de l'énergie et consiste à générer un signal carré avec un rapport cyclique module en fonction d'un signal de commande. Le signal généré peut servir à commander un circuit de puissance à découpage (pont en H), associé à un filtrage passe-bas inductif, pour générer une onde sinusoïdale ou d'une autre forme. La technique est utilisée dans les onduleurs monophasés, diphasés ou triphasés [29].

III.3.2 Principe de la commande MLI

La méthode de contrôle la plus courante de la vitesse de moteur à courant continu consiste en la commande MLI [30]. La variation de la vitesse du moteur est effectuée en augmentant ou en diminuant l'intervalle de temps lorsque le signal de commande est à la valeur logique 1 (t_{ON}). Ainsi, à un rapport cyclique de 100%, le moteur atteindra le régime maximum, le (t_{ON}) sera maximal et le (t_{OFF}) sera nul. À un rapport cyclique de 50%, le moteur tournera à la moitié de la vitesse nominale, donc (t_{ON}) = (t_{OFF}). Dans le cas d'un rapport cyclique de 0%. Le moteur sera arrêté, donc (t_{ON}) sera nul. Le rapport cyclique est noté par α et peut être calculé à travers la relation :

$$\alpha = \frac{t_{ON}}{t_{ON} + t_{OFF}} \times 100 \quad (\text{III.22})$$

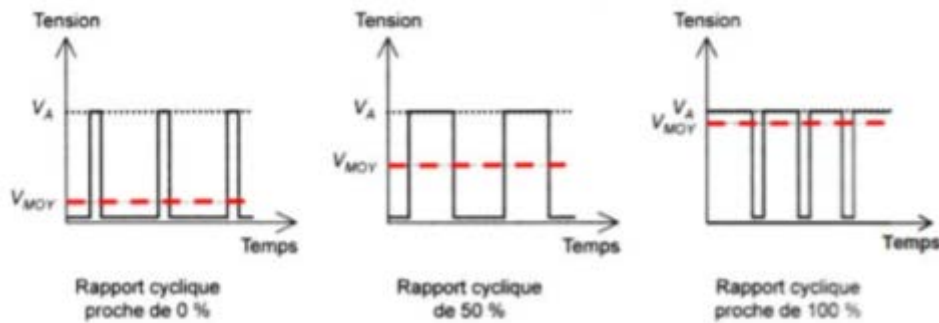


Figure III.4 : Forme de signal MLI avec différents rapports cycliques.

III.3.3 Alimentation par convertisseur électronique

Tous les moteurs à courant continu sont alimentés par des convertisseurs électroniques. Deux catégories de montages sont rencontrées : le hacheur et le redresseur commandé.

III.3.3.1 Hacheur

Le hacheur est rencontré dans tous les domaines de puissance, des petits actionneurs de la robotique jusqu'aux moteurs de traction ferroviaire. La technologie change suivant la puissance à commander : montages à transistors MOS ou IGBT pour les faibles ou moyennes puissances, montages à thyristors ou à GTO pour les puissances élevées. La présence ou non d'une bobine de lissage en série avec le moteur dépend de l'ordre de grandeur de la puissance du moteur : les interrupteurs utilisés pour les moteurs de puissance modeste sont assez rapides pour que la fréquence des commutations soit supérieure à la fréquence de coupure du moteur, pour se passer de cet élément encombrant, alors que ce n'est pas le cas pour des puissances plus importantes. La tension présentée à l'entrée du hacheur peut être obtenue à partir d'une source de tension continue (batteries, caténaire alimentée en continu) ou par redressement et filtrage d'une source de tension alternative.

III.3.3.2 Hacheur série

Le hacheur série, Buck, dévolteur ou hacheur abaisseur est un convertisseur direct $DC-DC$. La source d'entrée est de type tension continue et la charge de sortie continue de type source de courant.

C'est un hacheur « Un quadrant » qui n'a aucune réversibilité. L'énergie ne peut circuler que de l'entrée vers la sortie. Il ne comprend qu'un seul interrupteur commandé et une diode de roue libre.

Chapitre III : Réalisation pratique d'un asservissement de vitesse

Cependant ce transfert est réglable. Le paramètre de réglage est le rapport cyclique de la commande de l'interrupteur. Nous allons voir que cet hacheur est de type abaisseur, la tension de sortie étant toujours inférieure à la tension d'entrée. [31]

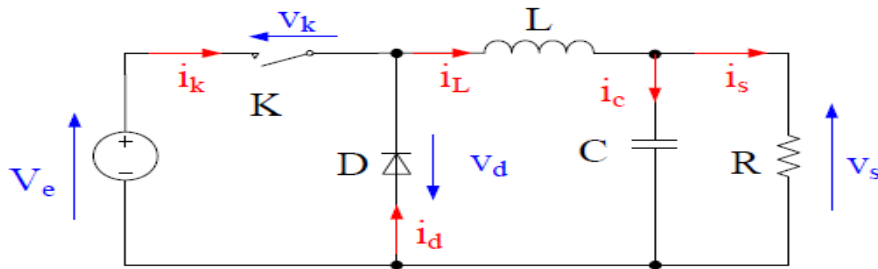


Figure III.5 : Structure de convertisseur BUCK

III.3.3.3 Principe de fonctionnement

L'interrupteur K se ferme et s'ouvre pendant une période T.

Il est fermé pendant l'intervalle $(0, \alpha T)$: la diode est bloquée, la source primaire fournit de l'énergie à l'inductance L et à la résistance R.

Il est ouvert de αT à T : la diode est passante et assure la continuité du courant et la décharge de L.

Lorsque ce convertisseur fonctionne en mode de conduction continue, on a les formes d'ondes de courant et de tension de la figure :

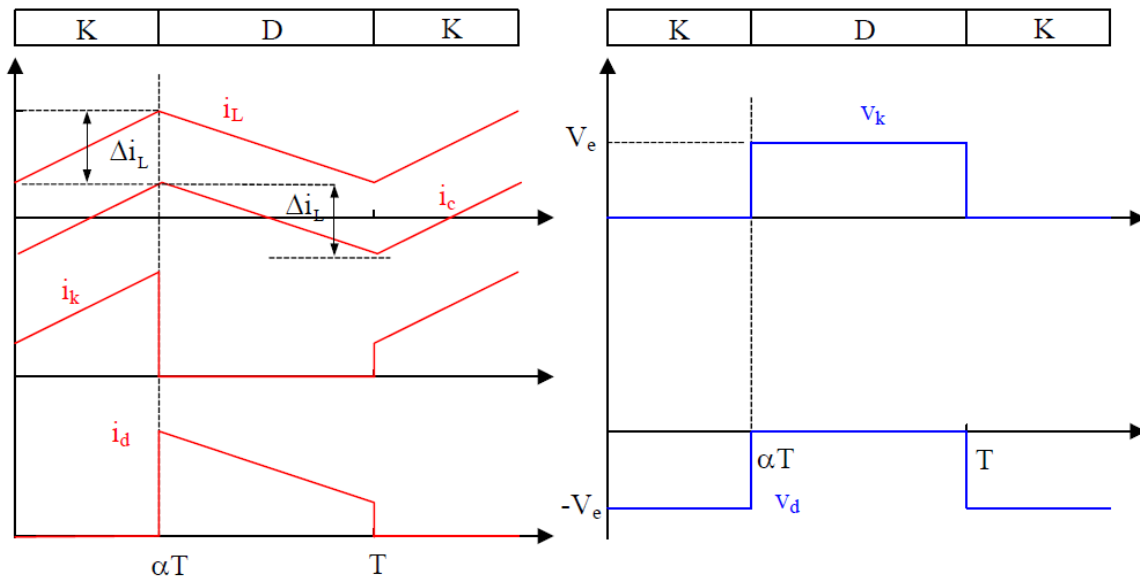


Figure III.6 : Formes d'ondes du convertisseur Buck (courants à gauche, tension à droite).

Chapitre III : Réalisation pratique d'un asservissement de vitesse

La tension de sortie en valeur moyenne :

$$V_s = \alpha \cdot V_e \quad [32] \quad (\text{III.21})$$

III.3.3.4 Le rapport cyclique

Le hacheur est commandé périodiquement. On appelle T période de hachage et f fréquence de hachage.

On appelle t_{OFF} la durée de fermeture de l'interrupteur sur une période et t_{ON} la durée d'ouverture de l'interrupteur sur une période. On a évidemment : $T = t_{ON} + t_{OFF}$

Le rapport cyclique est défini comme le rapport de la durée de fermeture à la période d'hachage:

$$\alpha = \frac{t_{ON}}{T}, \text{ le rapport cyclique est compris entre 0 et 1 : } 0 \leq \alpha \leq 1$$

III.3.3.5 Valeur moyenne de la tension aux bornes de la charge

La tension aux bornes de la charge est hachée, d'où le nom du montage.

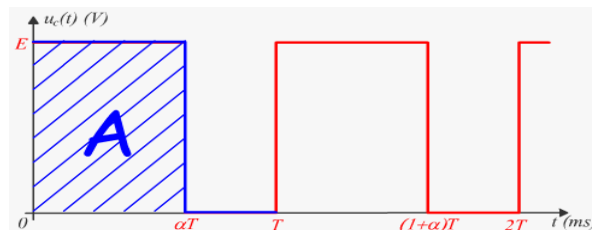


Figure III.7 : Oscillogramme de la tension aux bornes de la charge

On applique la méthode des aires : $\langle U_c(t) \rangle = U_{cmoy} = \frac{A}{T}$

Où : A est l'aire de la zone hachurée : $A = E \cdot \alpha T$, la valeur moyenne :

$$\langle U_c(t) \rangle = \alpha E \quad (\text{III.22})$$

Le rapport cyclique est toujours inférieur à 1, de ce fait la valeur moyenne de la tension aux bornes de la charge est toujours inférieure à la valeur de la tension de la source. Cet hacheur est aussi appelé hacheur dévolteur ou Buck convertie (pour les pays anglo-saxons).

III.3.3.5 Présentation de la maquette

La maquette est constituée d'un capteur de vitesse "encodeur" et un hacheur série capable de faire varier la vitesse d'un MCC. La Figure III.25 suivante schématise la connexion de la carte Arduino UNO avec l'entrée et la sortie du système (DC MOTOR + Hacheur série + encodeur). La figure ci-dessous montre une vue réelle de la maquette utilisée.

III.4 Réalisation de la commande en boucle ouverte

III.4.1 Commande du moteur à courant continu avec la carte Arduino

III.4.1.1 la carte Arduino UNO

C'est un circuit imprimé comportant tous les composants électroniques nécessaires pour faire fonctionner un microcontrôleur (Atmega 328) associé à une interface USB lui permettant de communiquer avec un ordinateur

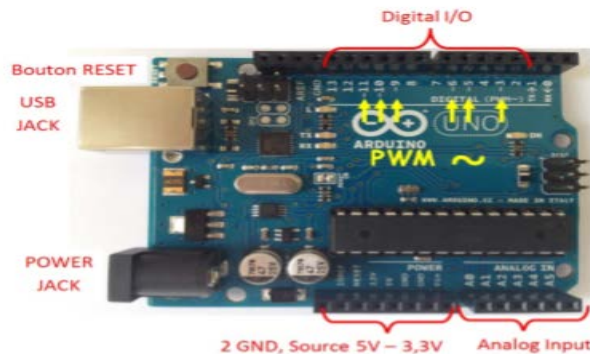


Figure III.8 : Description de la Carte Arduino "Uno"

- Microcontrôleur : ATmega328
- Operating Voltage : 5v
- Input Voltage (recommande) : 7-12 v
- Input Voltage (limites) : 6-20 v
- DC Current per I/O Pin : 40mA
- DC Current for 3.3V Pin : 50mA
- Flash Memory : 32 KB
- Clock Speed : 16MHz – Pins assignments :
- Analog Read (A0-A5)
- Analog Write [PWM] (3, 5, 6, 9, 10,11)
- Digital Read (2-19)
- Digital Write (2-19)

III.4.1.2 Présentation des sorties analogiques (mode PWM)

La carte Arduino Uno dispose de 6 sorties (3, 5, 6, 9,10 et 11) qui peuvent être utilisées en mode PWM, c'est-à-dire en modulation de largeur d'impulsion. Ce sont des signaux logiques binaires de fréquence constante (1000Hz) mais de rapport cyclique variable.



Figure III.9 : Description du signal PWM

Lorsqu'un moteur est alimenté par ce type de tension, tout se passe comme c'est-il était alimenté par une tension continue ajustable entre 0V (rapport cyclique= 0) et 5V (rapport cyclique=255). Ces sorties doivent être initialisées comme des sorties digitales

$$V_{out} = \frac{t_{ON}}{t_{OFF}} \times V_s \text{ avec } \tau_c = 2m \quad (\text{III.23})$$

La syntaxe de l'instruction permettant de générer le signal PWM est la suivante :

AnalogWrite (pin, valeur);

- pin : le pin sur laquelle on souhaite envoyer le signal (3, 5, 6, 9,10 ou 11).
- valeur : le rapport cyclique entre 0 et 255

III.4.2 Bloc d'isolation

De nombreux problèmes sont rencontrés lors du fonctionnement en liaison directe entre le circuit de commande et le circuit de puissance, tel que le retour de masse, le bruit électrique et le risque de destruction du circuit de commande par retour de courant de la charge.

Afin d'éviter tout problème de ce genre, le circuit de commande doit être électriquement isolé du circuit de puissance, cette isolation est réalisé par un optocoupleur.

III.4.2.1 Principe d'isolation

Pour isoler électriquement le circuit de commande de celui de puissance, on a recours à l'utilisation de deux sources d'alimentation ainsi que deux masses différentes, donc il n'y a aucune connexion entre les circuits cités.

Un avantage évident de l'utilisation de deux masses différentes est l'isolation contre le bruit électrique causé par le retour de masse, en effet une différence de potentiel entre deux points de masse cause de norme problèmes dans le milieu industriel, le courant de boucle induit dérange le signal.

III.4.2.2 Isolation par optocoupleur :

Un optocoupleur est considéré comme un élément de transfert de signal dont l'entrée et la sortie sont électriquement isolées l'une par un couplage optique .la Figure III.16 illustre le principe de fonctionnement d'un optocoupleur.

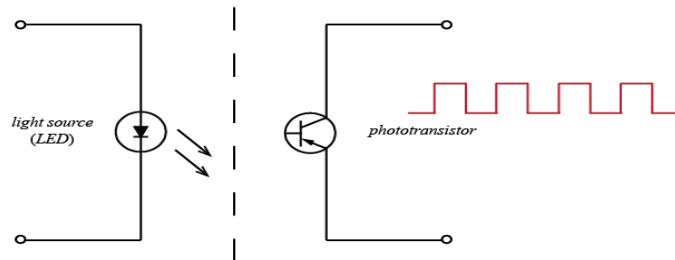


Figure III.10 : Principe de fonctionnement d'un optocoupleur

En général, l'entrée se compose d'une diode électroluminescent (LED) opérante dans l'infrarouge, la sortie est un photo-détecteur c'est –à –dire une photodiode, phototransistor ou éventuellement un photo-thyristor, La LED et la photo détecteur sont reliées optiquement, mais sont isolées électriquement dans un même boîtier.

Un signal électrique reçu d'un appareil de commande est transformé en un signal lumineux et devient alors émetteur optique, il se propage dans l'air, le plastique ou le verre.

Pré-amplification, ce signal électrique sera transmis au circuit de puissance

Lorsque on a utilisé est **TLP521-2** de type dont la structure interne est représentée sur la Figure III.17 et dont les caractéristiques principales sont données sur l'annexe

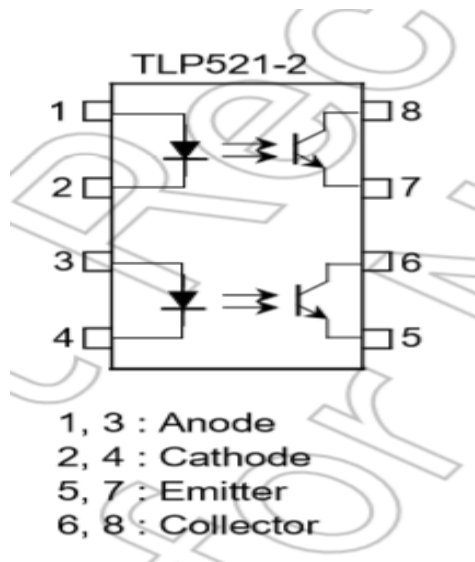


Figure III.11 : Schéma fonctionnel de l'optocoupleur

III.4.2.3 Calcul des résistances d'entrée et de sortie de l'optocoupleur

- On peut calculer la valeur de résistance d'entrée R_f qui protège la diode émettrice LED contre les surintensités .

$$V_e = V_f + R_d \times I_d \Rightarrow R_d = \frac{V_e - V_f}{I_f} \quad (\text{III.23})$$

Avec :

V_e : tension d'entre

V_f : tension aux bornes de la diode en conduction

I_f : courant direct de la diode

R_d : résistance d'entrée.

Dans notre cas : $V_c = 5V$; $V_f = 0.7V$ et $I_f = 50mA \rightarrow R_d = 300 \text{ Ohm}$

- Suivant la caractéristique du phototransistor on peut aussi calculer la valeur de la résistance de sortie

$$V_{cc} = V_{ce} + R_2 \times I_c \Rightarrow R_2 = \frac{V_{cc} - V_{ce}}{I_c} \quad (\text{III.23})$$

Avec :

V_{cc} : tension d'alimentation

V_{ce} : tension collecteur-émetteur

I_f : courant collecteur

Dans notre cas : $V_{cc} = 15V$; $V_f = 5V$ et $I_f = 2mA \rightarrow R_d = 3000 \text{ ohm}$

III.4.3 Bloc de puissance (Hacheur série)

Le circuit de puissance réalise est un hacheur de type dévolteur, il constitue essentiellement :

- D'un IGBT IRG4PH50S dont les caractéristiques sont données dans l'annexe
- Une charge qui dans notre étude est représentée par un moteur à courant continu de tension nominal 31V de courant nominal 100mA et de vitesse nominal 11000tr/mn
- Une diode de roue libre qu'il monté en parallèle avec la charge
-

Le circuit électronique réalise ci-dessous permet de contrôler un moteur à courant continu à partir des sorties PWM de la carte Arduino. Ce circuit doit varier le rapport cyclique du signal PWM.

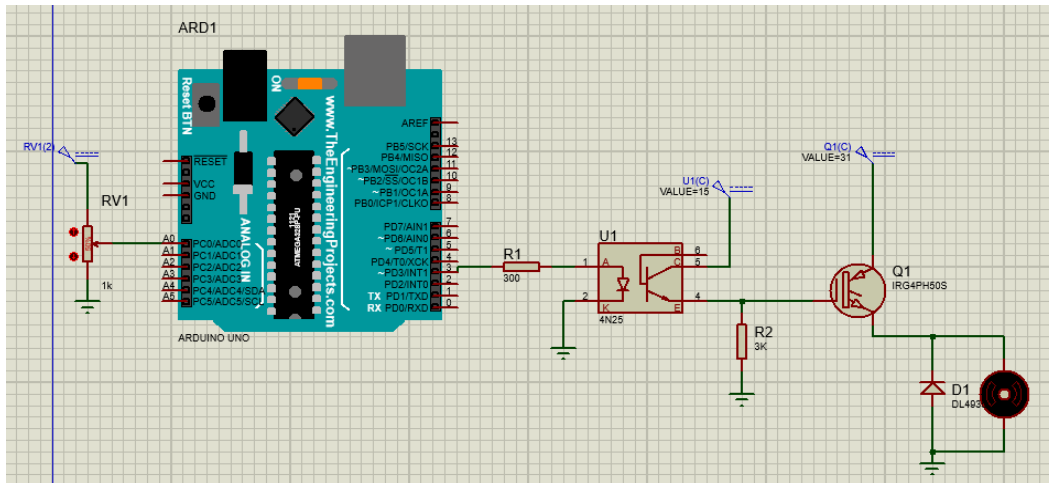


Figure III.12 : Schéma on Proteus de système en boucle ouverte (hacheur série)

Composant utilise :

- Arduino UNO
- Câbles de branchement
- Alimentation 15V
- Alimentation 31V
- IGBT
- Moteur à courant continue
- Optocoupleur
- Diode de roue libre
- 2 résistances

III.4.4 Forme du signal de commande de MLI

La Figure III.24 représente forme de signal de commande MLI (pwm)

Echelé :

le rapport cyclique Alpha = 0.85

le rapport cyclique Alpha = 0.25

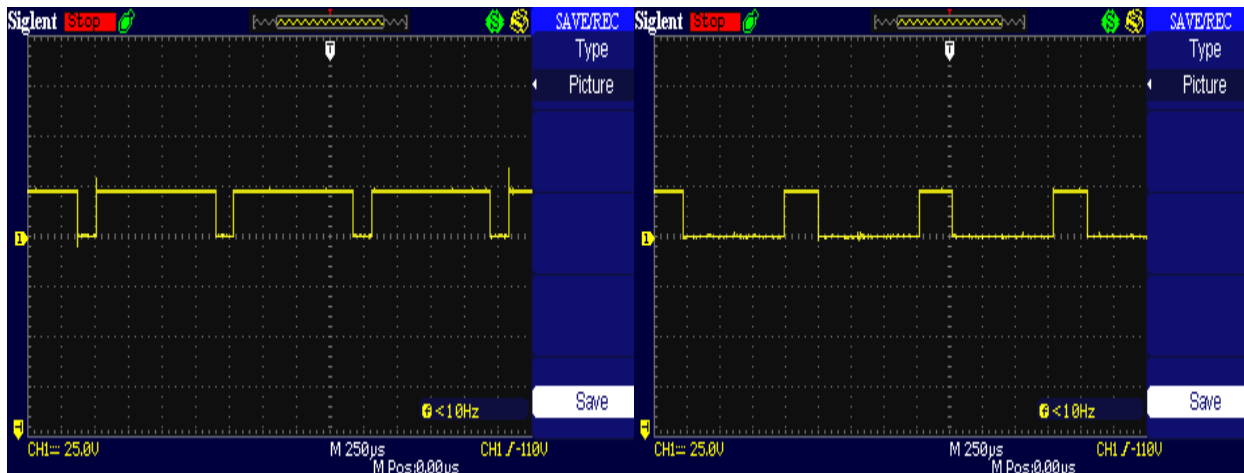


Figure III.13 : différent valeur de rapport cyclique

III.5 Commande du moteur à courant continu en boucle ferme

Dans cette activité, nous allons concevoir et mettre en œuvre un régulateur de vitesse pour un simple moteur à courant continu. En particulier, nous choisirons et réglerons les gains d'un contrôleur PI. Nous concevrons le contrôleur afin d'atteindre le niveau souhaité de réponse transitoire et examinerons en détail l'erreur en régime permanent produite par le système à boucle fermée obtenu, présence d'une perturbation constante.

La vitesse angulaire du moteur est estimée à l'aide d'un codeur en quadrature. Les impulsions du codeur sont comptées sur la carte Arduino via deux des entrées numériques de la carte. Une des sorties numériques de la carte est également utilisée pour activer et désactiver un transistor, connectant et déconnectant ainsi le moteur à une source de tension continue. La carte Arduino communique les données enregistrées à Simulink pour les visualiser et les analyser. La logique d'estimation de la vitesse du moteur en fonction du nombre de codeurs et de la logique de contrôle de la vitesse du moteur est implémentée dans Simulink. Initialement, cette logique est exécutée sur l'ordinateur hôte, mais plus tard, nous téléchargeons toute la logique sur la carte Arduino.

III.5.1 Présentation du capteur de vitesse

L'encodeur optique est un capteur de position extrêmement précis, il est constitué de deux parties :

- un disque renferme des bandes opaques et transparentes sur sa périphérie.
- un module qui sert à acquérir les alternances (opaque/transparent) pour les produire en formes d'ondes analogiques.

Le module contient un émetteur et un récepteur. La partie émettrice se compose :

Chapitre III : Réalisation pratique d'un asservissement de vitesse

- une diode lumineuse LED (Light Emitting Diode).
- une lentille.

La lumière émise par la diode se transforme en faisceaux parallèles à l'aide de cette lentille située directement devant la diode. A l'opposé de l'émetteur (LED +Lentille) on trouve la partie réceptrice englobant les photodiodes et les comparateurs ayant comme sortie les chaînes A et B. Lorsque le moteur tourne, le disque solidaire l'axe du moteur permet l'alternance entre bandes opaques et bandes transparents d'où l'interruption des faisceaux lumineux. Les photodiodes qui détectent ces interruptions sont l'origine de la formation du signal de sortie.

Ces détecteurs sont également espacés de telle sorte que sur une paire de détecteurs une période

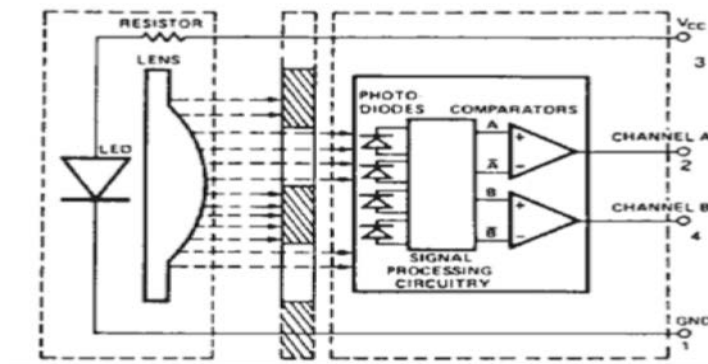


Figure III.14 : Circuit interne du module

Opaque (0 Volts) correspond à une période transparente (5 Volts) sur la paire adjacente. Deux comparateurs reçoivent ces signaux et produisent les sorties finales pour les canaux A et B. Le canal A est en quadrature de phase avec le canal B.

III.5.2.1 Principe de fonctionnement

L'encodeur optique fournit deux signaux carrés en quadrature, comme sur la capture ci-dessous : Ces deux signaux permettent de mesurer à la fois la vitesse et le sens de rotation

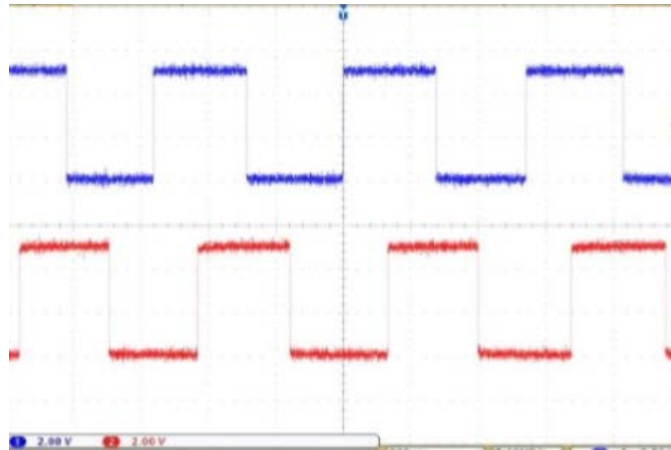


Figure III.15 : Les deux signaux issus de l'encodeur optique

La mesure de la vitesse se fait simplement en comptant le nombre d'impulsions pendant un temps fixe. La cadence d'échantillonnage utilisée pour l'asservissement sera de 0.02 s.

III.5.3 L'interfaçage Arduino↔Matlab/Simulink

III.5.3.1 L'environnement Matlab/Simulink

C'est un logiciel de calcul mathématique pour les ingénieurs et les scientifiques créé par MathWorks.

MATLAB est un environnement de programmation pour le développement d'algorithme, d'analyse de données, de visualisation, et de calcul numérique. En utilisant MATLAB, la résolution des problèmes de calcul complexes se fait plus rapidement qu'avec des langages de programmation traditionnels, tels que C, C++, et le Fortran.

SIMULINK est un environnement pour la simulation, il fournit un environnement graphique interactif et un ensemble de bibliothèques de bloc qui permettent de concevoir, simuler, mettre en application, et examiner une variété de systèmes, tel que les systèmes de communications, de commandes, de traitement des signaux, de traitement visuel, et de traitement d'image. Il existe trois possibilités d'interfacer la carte Arduino avec Matlab/Simulink, à savoir :

1. Programmation de la carte Arduino Uno comme une carte d'interface.
2. Utilisation du package ArduinoIO.
3. Utilisation du package Arduino Target.

III.5.3.2 Introduction sur Arduino IO :

Arduino est une plate-forme de prototypage d'objets interactifs à usage créatif constituée d'une carte électronique et d'un environnement de programmation.

III.5.3.3 ArduinoIO

Cette solution consiste à utiliser la carte arduino comme une interface d'entrées (Analog Input)/sorties (Analog/Digital Output). Ce package permet de communiquer Matlab ou Simulink avec la carte Arduino via un câble USB.

Elle consiste à pré-charger un programme dans la carte Arduino afin que celle-ci fonctionne en serveur.

Ce programme consiste à "écouter" les requêtes envoyées via la liaison série (USB) et de répondre à ces requêtes en renvoyant l'état d'une entrée ou en modifiant l'état d'une sortie. Ces mêmes entrées/sortie sont vues dans Matlab comme des entrées logiques ou analogiques (utilisation du CAN) ou des sorties analogiques (mode PWM)

III.5.3.4 Exploitation de la bibliothèque ArduinoIO sous Simulink

Les blocs nécessaires pour notre objectif d'asservissement sont les suivants :



Figure III.16 : Les Blocs d'ArduinoIO nécessaires pour la commande

- **Real-Time Pacer** : Ce bloc permet de ralentir le temps de simulation de sorte qu'il synchronise avec le temps réel écoulé. Le coefficient de ralentissement est défini par l'intermédiaire du paramètre Speed up.
- **Arduino IO Setup** : Pour configurer sur quel port la carte Arduino UNO est connectée. Pour cela il suffit de sélectionner le port dans le Gestionnaire des périphériques.
- **Encodeur Read** : Pour configurer à partir de quel pin on va réaliser l'acquisition des signaux issue du capteur (channel A et channel B).
- **Arduino Analog Write**: Pour configurer à partir de quel pin [3, 5, 6, 9, 10,11] on va envoyer la commande en PWM vers l'actionneur.

III.5.4 Exploitation de l'encodeur optique avec Arduino/Simulink

Nous utiliserons Simulink pour lire la sortie du codeur et pour tracer la vitesse en temps réel.

En particulier, nous utiliserons le package arduino IO de Math Works

1. Pré-chargement de adiosrv.pde sur la carte Arduino UNO
2. Développement du modèle Simulink

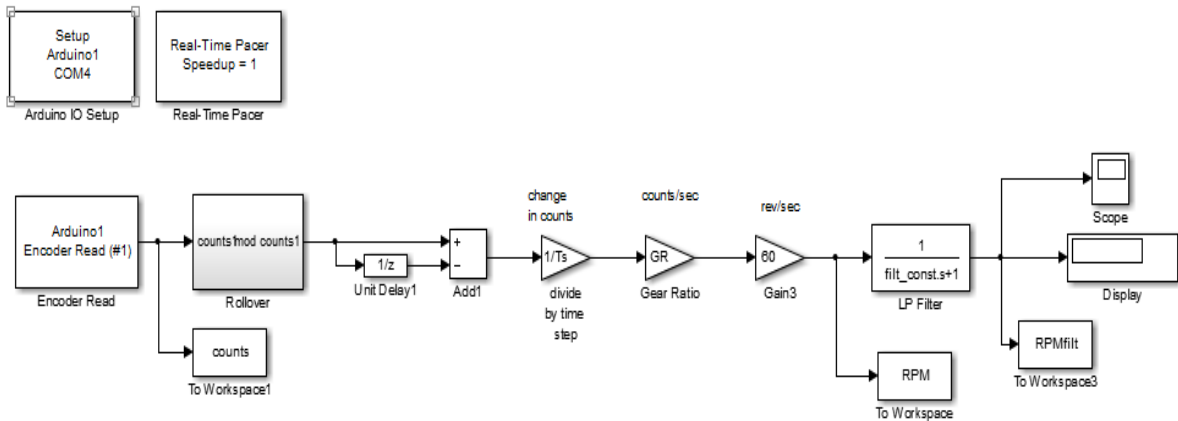


Figure III.17 : acquisition des impulsions sous Arduino IO Library

III.6 Implémentation de la commande sur Simulink

Dans cette partie l'implémentation du régulateur se fera sur Simulink.

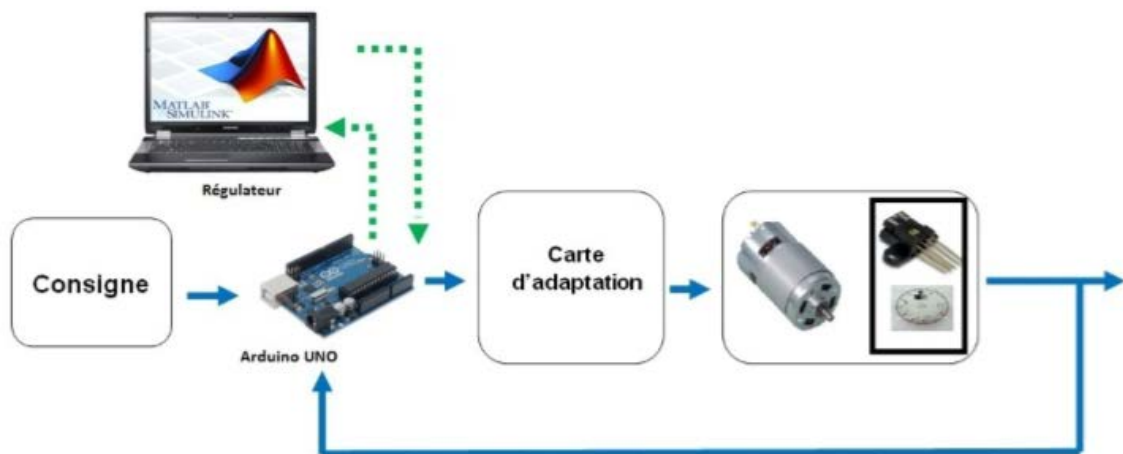


Figure III.18 : Schéma synoptique de l'asservissement à implémenter

L'étape de la commande du système est constituée de deux parties.

Chapitre III : Réalisation pratique d'un asservissement de vitesse

- La première partie consiste à utiliser l'outil Matlab PID Tuning pour déterminer les différents paramètres du correcteur et l'objectif de commande.
- La deuxième partie consiste à implémenter le régulateur sur Simulink

III.6.1 Implémentation de la commande sous Simulink

La boucle d'asservissement à implémenter sur Simulink se traduit par le schéma suivant : L'asservissement de notre procédé est assuré par le schéma Simulink ci-dessous qui regroupe la consigne, le comparateur, le correcteur PI(z), l'acquisition de vitesse et l'envoi de la commande PWM.

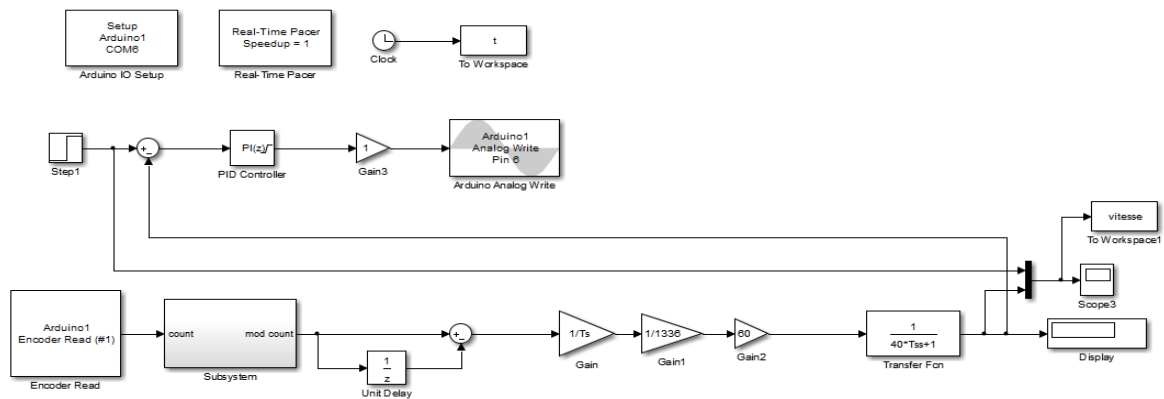


Figure III.18 : Modèle Simulink d'asservissement de vitesse

III.7 Résultats de la réalisation expérimentale en boucle ferm

Réponse de la vitesse :

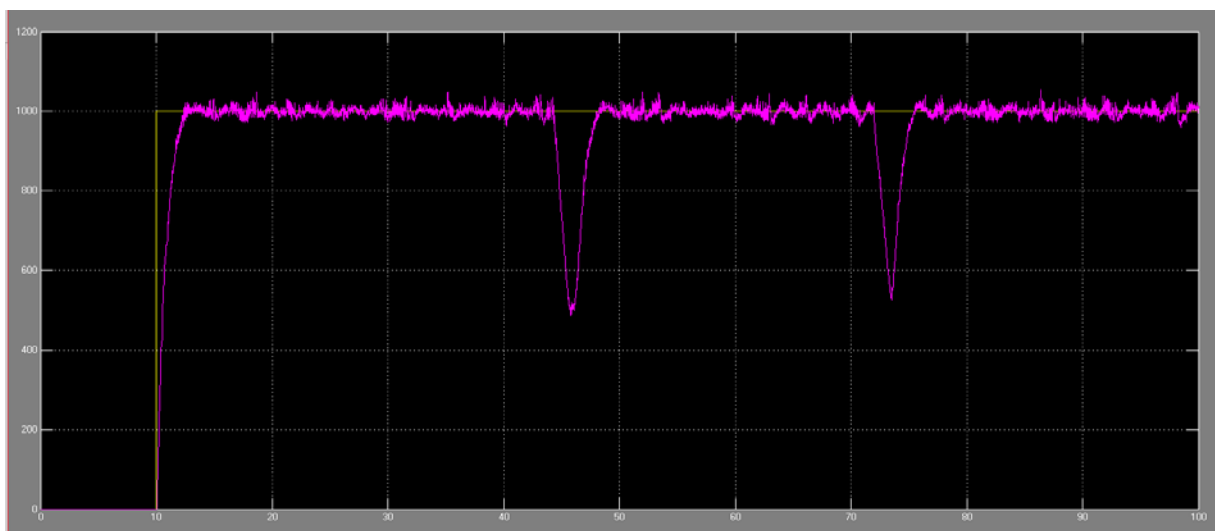


Figure III.19 : Réponse de la vitesse en charge

❖ **Interprétation :**

Au démarrage, le moteur atteint sa valeur de consigne 1000tr/mn avec un temps de réponse satisfaisant (2 s). Lors de l'application de la charge à l'instant ($t=45s$), la vitesse diminue puis reprend immédiatement sa consigne.

Dans ce contexte, les sorties rejoignent avec succès nos consignes vu le travail excellent que fait notre PI. Ainsi, cela argumente clairement sur la puissance et le potentiel de notre système de commande.

III.8 Conclusion

D'après les résultats pratiques, on constate que la commande en boucle fermée par un régulateur PI est très robuste.

Ainsi, les résultats obtenus par la réalisation montrent les avantages qu'apporte la technique de commande en boucle fermée proposée dans ce chapitre.

Conclusion générale

Le travail présenté dans ce mémoire est constitué de deux parties :

La première partie constitue une contribution à la mise en œuvre des techniques non conventionnelles de commande numérique. Nous avons ainsi élaboré deux stratégies de contrôle de vitesse des machines à courant continu basées sur la logique floue et les algorithmes génétiques. Les résultats obtenus nous permettent de montrer que ces techniques, jusqu'à présent peu utilisées, surtout en milieu industriel, sont bien adaptées lorsqu'on a des contraintes sévères au niveau de la régulation. Les systèmes de contrôle de la vitesse des machines à courant continu utilisant la logique floue ou les algorithmes génétiques présentent des performances intéressantes.

La deuxième partie constitue la réalisation de la commande numérique d'un moteur à courant continu. Pour réaliser ce travail, nous avons jugé utile de le diviser en plusieurs parties à savoir :

1. On a réalisé une carte de commande pour la génération du signal du PWM à base d'un Arduino, dont les avantages résident dans une fiabilité plus élevée, une réduction de cout de la réalisation et une augmentation de la flexibilité de la commande avec la possibilité d'atteindre des performances élevées.
2. La carte de puissance est un hacheur série (dévolteur) à IGBT qui présente des avantages permettant de le considérer comme un commutateur favorable pour la conversion de l'énergie électrique tel que :
 - La commande en tension.
 - Simplicité de commande (il nécessite une faible puissance de commande).
 - Une forte impédance d'entrée.
 - Pertes de la commutation faibles
3. La régulation de la vitesse est réalisée par un microcontrôleur Arduino, après avoir mesurée la vitesse par l'encodeur optique, comparée avec celle de référence et traitée par le régulateur PI.

Références Bibliographiques

- [0]. MEMOIRE Présentée pour l'obtention du diplôme de MASTER
Par BENSADALLAH Ahmed
- [1]. CH.Bekkouche et AK.Touggourti, «Etude et simulation d'un variateur de vites commande un moteur à courant continu », mémoire de master option Electrotechnique industrielle, Université Kasdi Merbah Ouargla, 2016.
- [2]. P. Barret, 'Machines électriques', Edition Ellipses, 2002
- [3]. BENOUMESSAD Ismahane ET DJEBBAR Ahlem (ETUDE ET REALISATION D'UN HACHEUR SERIE POUR LA COMMANDE D'UN MOTEUR A COURANT CONTINU) mémoire de master option Electrotechnique industrielle, Université batna 2 2018.
- [4]. M. A. Johnson & Mohammed H.Moradi «PID Control (New identification and design methods) », Springer, 2005
- [5]. Freddy Mudry, Ajustage des paramètres d'un régulateur Pid, Département d'électricité et Informatique - Ecole d'ingénieur du Canton de Vaud, Mars 2006
- [6]. Réglages D'une Boucle Pid [En Ligne], The national institutes of standards and technology, Url =Www.Linuxcnc.Org/Docs/Html/Motion/Pid_Theory_Fr.Html, Date de consultation le : 01/11/2015.
- [7]. ELMIR Younes, Optimisation multi objectif par les algorithmes génétiques et approche Pareto de paramètres d'un contrôleur PI- mémoire de Magistère - Université d'Oran Mohamed Boudia
- [8]. Lamia Mamlouk et Fouad Benouirane. "Vie Artificielle". 2003
- [9]. Laurent Bougrain. "Intelligence artificielle". Projet CORTEX, LORIA/INRIA LORRAINE, 2000
- [10]. D. Beasley, « An Overview of Genetic Algorithms Part1: Fundamentals », University of Cardiff, Cardiff, CF2, 4YN, United Kingdom, 1993.
- [11]. D.E. Goldberg, « Algorithmes génétiques. Exploration, optimisation et apprentissage automatique », Addison-Wesley. France, 1994.
- [12]. M. MITCHELL, « An Introduction to Genetic Algorithms », London: Cambridge, Massachusetts. London, England, 1999.

- [13]. David E. Goldberg. “Algorithmes génétiques Exploitation, optimisation et apprentissage automatique”. Addison-Wesley, France, SA, 1994.
- [14]. Youssef Harrat. “Contribution à l’ordonnancement conjoint de la production et de la maintenance : Application au cas d’un job Shop”. Thèse de Doctorat, L’U.F.R des Sciences et Techniques, Université de Franche-comté, 2003.
- [15]. Les critères d’arrêt se résument alors en : 1. Arrêt après un nombre de générations fixé à priori. 2. Arrêt lorsque la population cesse d’évoluer ou en présence d’une population homogène
- [16]. O. Landolt, P. Marchal et E. Vittoz, « processeurs flous », Eisi-Micro Swiss Semaine d’enseignement spécifique, Exposé on st-Inier, Décembre 2000.
- [17]. J. C de Barros, « Application de la logique floue à la commande optimale du moteur asynchrone », thèse de doctorat, université Aix- Marseille III, France, 2003.
- [18]. R. Benabdennour, P. Borne, M. Ksouri, et F. M’sahli « Identification et Commande des procédés Industrielles », Edition technique, Paris 2001.
- [19]. LAOUFI Mourad, BABA Nawel, DJAMAI Hayet, Planification de la trajectoire D’un robot mobile par la logique floue. Mémoire de fin d’étude, l’université de Bechar Promotion 2010
- [20]. S.KHELKHAL, La logique floue et le contrôle avancé en milieu industriel, Mémoire de fin d’étude. Casablanca / Maroc .2004.
- [21]. GUENOUNOU Ouahib, Méthodologie De Conception De Controleurs Intelligents Par L’approche Génétique-application à un Bioprocédé, THESE De Doctorat De l’Université De Toulouse.2009.
- [22]. DJAABOUB Salim, Logique floue et SMA : Aide à la décision floue dans les systèmes multi- agents, Mémoire Magister en informatique. Universitaire Mentouri de Constantine 2009. [4] S.KHELKHAL, La logique floue et le contrôle avancé en milieu industriel, Mémoire de fin d’étude. Casablanca / Maroc .2004.
- [23]. Gonzalo Cabodevila, Commande et identification Floue. Ecole Nationale Supérieure de Microtechnique.1993.
- [24]. Olivier CAZIER, Instabilités dynamiques de systèmes frottants en présence de variabilités Paramétriques, Application au phénomène de crissement, Thèse de doctorat, l’université de VALENCIENNES.2012

- [25]. Lycée Gustave Eiffel de Dijon Classe préparatoire P.T.S.I. Année 2016 – 2017
- [26]. H .Sellam et A.Roubache, « Etude et réalisation d'un système de commande à base d'un microcontrôleur pour la régulation de la vitesse d'un moteur à courant continu», PFE électronique N6, 2005.
- [27]. B. Deforge & Q. David «Asservissement en position d'un axe linéaire», projet d'automatique, 2008.
- [28]. C. le Lann «Le PID utilisé en régulation de position et/ou de vitesse de moteurs électriques», Projet de fin d'étude, 2007.
- [29]. Luc Lasne, Electronique de puissance, Dunod, 2015
- [30]. Pathik A. Kapadiya, Ishan R. Shah, Nirav D. Mehta, STUDY AND IMPLEMENTATION OF DC DRIVE USING PIC16F877A MICROCONTROLLER, International Journal of Research in Engineering and Technology (IJRET), Volume : 03 Issue : 05,2014.
- [31]. M.GARNERO la conversion Dc Dc : les hacheurs
- [32]. Touaref Mostefa. (Etude et simulation d'un convertisseur AC/DC type BUCK (Abaisseur de tension)) Mémoire de Fin d'Etudes MASTER. UNIVERSITE DE BATNA 2014