

République Algérienne Démocratique et Populaire  
Ministère de l'Enseignement Supérieur Et de La Recherche Scientifique



Université de Ghardaïa

N° d'ordre :

N° de série :

Faculté des sciences et de la Technologie  
Département de science et technique  
**Mémoire présenté en vue de l'obtention du diplôme de**

## **MASTER**

**Domaine :** Science et Technique

**Filière :** Génie électrique

**Spécialité :** Automatique

**Présenté Par :** TERBAGOU Ramzi

## **Thème**

**Etude et Simulation d'une Commande de  
Vitesse d'un Moteur à Courant Continu**

**Soutenu publiquement le : 28/05/2015**

**Devant le jury :**

**Mr .BOUARAOUR Kamel**

Maître de Conférence A

Univ. Ghardaïa

**Président**

**Mr .TOUAFEK Khaled**

Maître de recherche A

URAER Ghardaïa

**Encadreur**

**Mr. ADOUANE Mabrouk**

Maître de recherche B

URAER Ghardaïa

**Examineur**

**Année universitaire 2014/2015**

# Dédicace

Avant tous,

Je remercie dieu le tout puissant de m'avoir donné le courage et la patience pour réaliser ce travail malgré toutes les difficultés rencontrées.

Je dédie ce modeste travail :

A mes très chers parents, que dieu les garde et les protège pour leurs soutien moral et financier, pour leurs encouragements et les sacrifices qu'ils ont endurés.

A mes frères,

A ma sœur

A ma grande famille

Aux chers amis,

A tous les amis (es) d'études surtout ceux de génie électrique promotion 2015

# REMERCIEMENTS

Nous remercions ALLAH qui nous a donné la volonté et la patience pour terminer ce modeste travail.

Nous tenons à remercier chaleureusement toutes les personnes qui, en particulier, à notre encadrement Mr. TOUAFEK Khaled et Mr. FIHAKHIR Amine Mehdi qui ont été présents non seulement par leurs conseils et leur savoir, mais aussi par tout un ensemble d'idées et de paroles qui nous ouvrent de nouveaux horizons vers le succès.

Nos remerciements sont également adressés à tous les enseignants du département de science et technique qui ont participé à notre formation pendant tout le cycle universitaire.

Nos vifs remerciements sont adressés à tous mes collègues dans l'option

Ce travail a été réalisé au sein du Département de science et technique -Université de Ghardaïa. Je tiens également à remercier Monsieur HADJ SAYED Abdelkader chef du département de science et technique

Enfin, nous adressons nos remerciements au président du jury Mr. BOUARAOUR Kamel ainsi qu'au membre du jury pour avoir accepté d'évaluer ce travail :

- Mr. ADOUANE Mabrouk

## Résumé

- Résumé

Dans ce projet on a fait l'étude et la conception de la commande en vitesse d'un moteur à courant continu. On a débuté par la modélisation du système à contrôler. la simulation matlab de plusieurs types de commandes synthétisés basés sur un cahier de charge bien définie (le régulateur PI , retour d'état , retour d'état avec intégrale, retour d'état observé, PI avec estimation de la mesure et par la logique floue). On a aussi fait la conception sur le logiciel PROTEUS les cartes de commande et de conditionnement avant la réalisation de ces cartes. la discrétisation du correcteur PI classique avec la routine PI pour l'implémentation sur le pic ont été développés .

Les résultats obtenus ont montré que la régulation sans capteur présente les même performances que celle avec capteur mais elle reste sensible aux bruit de mesure et à la précision de calcul.

- Abstract

In this project we did the study and design of the command speed of a DC motor. System modeling is started with at contrôler. la matlab simulation of several types of commands synthesized based on a well defined set of specifications (the PI controller, state feedback, state feedback with integral, observed state of return PI with estimate of the extent and fuzzy logic) .We also did the design on the PROTEUS software control and conditioning cards before achieving these discretization cartes. la corrector classic PI with PI routine for implementation on the peak were developed.

The results showed that the sensorless control offers the same performance as with sensor but remains sensitive to measurement noise and precision calculated.

- الملخص

في هذا المشروع تم إجراء دراسة وتصميم للتحكم في سرعة محرك كهربائي بتيار ثابت بدأنا بنمذجة النظام ثم انشاء محاكاة ببرنامج الماتلاب بعدة أنواع من الأوامر على أساس دفتر من الشروط(تحكم pi , تحكم ردد فعل ، تحكم pi مع التقدير والمنطق الضبابي). وكان أيضا التصميم على برنامج الايسيس الحاسوبي لبطاقة التحكم والتكيف.

وأظهرت النتائج أن التحكم في سرعة المحرك الكهربائي بدون أجهزة الاستشعار تقدم نفس الأداء كما هو الحال مع أجهزة الاستشعار ولكن لا تزال الحساسية للضجيج في القياس والدقة المحسوبة.

PI= Proportionnelle Intégrale

---

# Table des matières

---

## Table des matières :

Introduction Générale.....	1
Chapitre I : Notions et Généralités sur La régulation automatique.....	2
I.1. La régulation automatique.....	2
I.2. Eléments constitutifs d'une boucle de régulation .....	2
I.2.1. Le système à commander .....	2
I.2.2. L'actionneur .....	2
I.2.3. Le correcteur .....	2
I.2.4. Le capteur.....	3
I.3. Les problèmes communs des capteurs .....	3
I.4. Qu'est-ce qu'un observateur ? .....	3
I.5. Notions sur l'espace d'état :.....	4
I.5.1. L'état d'un système :.....	4
I.5.2. Variables d'état:.....	4
I.5.3. Vecteur d'état :.....	4
I.5.4. Espace d'état : .....	4
I.5.5. Commandabilité :.....	4
I.5.6. Observabilité : .....	4
I.6. La commande par retour d'état : .....	4
I.7. La synthèse du correcteur par retour d'état.....	6
I.7.1. Placement de pôles .....	6
I.7.2. Formule d'Ackermann .....	6
I.7.3. Remarque .....	7
I.8. La commande par retour de sortie ou l'observateur.....	7
I.9. La synthèse de l'observateur.....	8
I.10. Conclusion.....	8
Chapitre II : Etude et simulation du système à contrôler .....	9
II.1. Introduction.....	9
II.2. Description du système .....	9
II.3. Modélisation du MCC à aimant permanents .....	9
II.4. Représentation du MCC en boucle ouverte :.....	12
II.4.1. Modélisation du capteur.....	14
II.4.2. Modélisation du hacheur :.....	16
II.4.2.1. Hacheur série : .....	16
II.4.2.2. Principe de fonctionnement d'un hacheur : .....	16

II.4.2.3. Remarque :	17
II.5. Simulation du moteur :	19
II.5.1. Boucle ouverte :	19
II.5.2. Boucle fermée :	19
II.6. Régulation PI :	21
II.7. Représentation du système dans l'espace d'état :	24
II.8. Commande par retour d'état :	26
II.9. Commande à retour d'état et bouclage intégral :	28
II.10. Commande par retour d'état observé :	29
II.11. Régulation PI avec estimation de la mesure :	31
II.12. Commande par logique Floue :	32
II.12.1. Correcteurs flous de type PI :	32
II.12.2. Principe et structure de la commande :	33
II.13. Conclusion :	35
Chapitre III : Conception de la commande	37
III.1. Introduction :	38
III.2. Carte de commande :	38
III.2.1. Microcontrôleur PIC 18F458 :	39
III.2.1.1. Organisation externe du PIC 18F458 :	40
III.2.1.2. Les ports d'entrée /sortie :	41
III.2.1.3. Le convertisseur analogique numérique du PIC 18F458 :	42
III.2.2. Communication série asynchrone RS232 :	43
III.3. Carte de conditionnement :	46
III.3.1. Etage différentiel :	46
III.3.2. Etage inverseur :	47
III.3.3. Le filtrage :	48
III.3.4. Etage de protection :	48
III.3.5. Carte de sortie	49
III.3.6. Le signal PWM :	50
III.4. Programmation du Microcontrôleur PIC :	51
III.4.1. Introduction :	51
III.4.2. Discrétisation du correcteur PI :	51
III.4.3. Période d'échantillonnage :	52
III.4.4. Implémentation du PI :	53
III.5. La routine de traitement PI	54

III.6. Conclusion.....	55
Conclusion Générale .....	56

## Liste des figures :

<b>Figure I.1</b> : structure d'un système de régulation automatique .....	2
<b>Figure I.2</b> : représentation interne d'un système .....	5
<b>Figure I.3</b> : structure d'une commande par retour d'état .....	5
<b>Figure I.4</b> : structure d'une commande par retour de sortie .....	7
<b>Figure II.1</b> : Système à contrôler.....	5
<b>Figure II.2</b> Constitution du moteur à courant continu.....	10
<b>Figure II.3</b> :Schéma électrique d'un MCCAP.....	10
<b>Figure II.4</b> :Schéma fonctionnel du MCC.....	12
<b>Figure II.5</b> :la caractéristique de $V\omega = f(\omega)$ .....	15
<b>Figure II.6</b> :Convertisseur continu-continu.....	16
<b>Figure II.7</b> :hacheur série .....	16
<b>Figure II.8</b> :chronogramme de la fermeture et de l'ouverture .....	17
<b>Figure II.9</b> : moteur avec bobine de lissage .....	18
<b>Figure II.10</b> :modélisation du hacheur .....	18
<b>Figure II.11</b> : schéma bloc de la boucle ouverte .....	19
<b>Figure II.12</b> : réponse en boucle ouverte pour un échelon $\alpha=50\%$ .....	19
<b>Figure II.13</b> :schéma bloc de la boucle fermée.....	20
<b>Figure II.14</b> : réponse du moteur en boucle fermée.....	20
<b>Figure II.15</b> :représentation des pôles .....	22
<b>Figure II.16</b> :schéma bloc de la régulation PI.....	23
<b>Figure II.17</b> : réponse avec régulation pour une consigne de 10 (rad/sec) .....	24
<b>Figure II.18</b> : schéma bloc du retour d'état.....	26
<b>Figure II.19</b> : réponse avec retour d'état. ....	26
<b>Figure II.20</b> : retour d'état avec précompensateur .....	27
<b>Figure II.21</b> : réponse avec pré compensateur .....	27
<b>Figure II.22</b> : schéma du retour d'état intégral (PI) .....	28
<b>Figure II.23</b> : réponse avec retour d'état intégral.....	29
<b>Figure II.24</b> : retour d'état avec observateur.....	30
<b>Figure II.25</b> : réponse avec retour d'état observé .....	31
<b>Figure II.26</b> : schéma du correcteur PI avec estimateur de vitesse .....	31
<b>Figure II.27</b> : réponse du PI avec vitesse estimé.....	32
<b>Figure II.28</b> : schéma bloc de la structure de commande .....	33
<b>Figure II.29</b> : Fonctions d'appartenance des variables d'entrées.....	33
<b>Figure II.30</b> : Fonctions d'appartenance de la variable de sortie .....	34
<b>Figure II.31</b> : réponse avec régulateur flou .....	34
<b>Figure III.1</b> :Schéma synoptique de la réalisation .....	38
<b>Figure III.2</b> : Brochage du PIC18F458. ....	40
<b>Figure III.3</b> : Principe de conversion d'un ADC de PIC 18F458 .....	43
<b>Figure III.4</b> : Schéma fonctionnel d'une liaison série asynchrone de la norme RS232 .....	43
<b>Figure III.5</b> : Circuit du MAX232 .....	44
<b>Figure III.6</b> : Photo de la carte de commande.....	45
<b>Figure III.7</b> : Schéma synoptique du conditionnement.....	46
<b>Figure III.8</b> :Montage différentiel.....	47
<b>Figure III.9</b> : Montage inverseur .....	47

<b>Figure III.10</b> : Filtre passe-bas.....	48
<b>Figure III.11</b> : Montage de protection.....	49
<b>Figure III.12</b> : Carte de conditionnement .....	49
<b>Figure III.13</b> : Principe du signal PWM .....	50
<b>Figure III.14</b> : Boucle de régulation réel .....	54
<b>Figure III.15</b> : organigramme de la régulation.....	54

## Liste des tableaux

<b>Tableau I.1</b> : les tables d'inférence pour le PI floue.....	34
---	----

---

# **INTRODUCTION GENERALE**

---

## Introduction Générale

De nos jours les sciences de l'ingénieur font largement appel à l'estimation de l'état des systèmes. La connaissance complète de l'état d'un système est souvent nécessaire à l'élaboration d'une loi de commande ou à la mise en place d'une stratégie de surveillance. Or l'état d'un système n'est en règle générale que partiellement disponible et les signaux d'entrée et de sortie sont en pratique les seules grandeurs accessibles par mesure. La solution la plus répandue pour pallier ce problème consiste à coupler au système un système auxiliaire, appelé estimateur ou observateur d'état. L'observateur fournit une estimation de l'état du système à partir de son modèle et des mesures de ses entrées et sorties. Il est cependant bien connu que l'estimation d'état fournie par ce type d'observateur se dégrade considérablement si le modèle du système en question est mal connu ou si des perturbations inconnues agissent sur la sortie ou sur l'état du système.

Les progrès conjoints de l'électronique de puissance et de l'électronique numérique permettent aujourd'hui d'aborder la commande à vitesse variable dans des applications faibles puissances. Dans cette dernière, la suppression du capteur mécanique de vitesse peut présenter un intérêt économique et améliorer la sûreté de fonctionnement.

L'application choisie est la régulation en vitesse d'un moteur à courant continu, le travail était réparti sur quatre chapitres :

Le premier chapitre contient la partie théorique ainsi que des notions et de généralités sur la théorie exploitée.

Dans le second chapitre on a donné une description générale sur notre système à contrôler avec la modélisation de chaque élément constitutif, puis faire des simulations pour valider le modèle et la commande.

Le troisième a été consacré à la partie conception et simulation des différentes cartes nécessaires à la commande.

L'implémentation de différents algorithmes avec les résultats obtenus sont mentionnés dans le dernier chapitre avec leurs interprétations, et enfin on termine par une conclusion générale.

---

Chapitre I : Notions et  
Généralités sur La régulation  
automatique

---

---

---

## Chapitre I : Notions et Généralités sur La régulation automatique

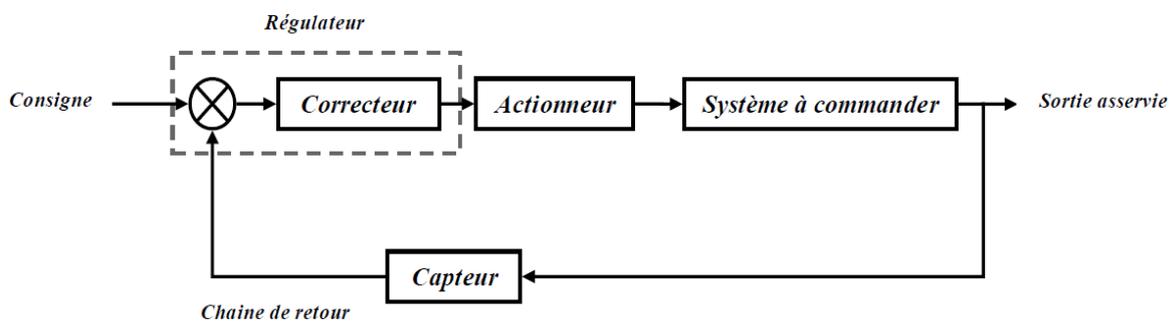
---

---

### I.1. La régulation automatique

La régulation automatique (automatic control , Figure I.1) est la technique de l'ingénieur offrant les méthodes et les outils nécessaires à la prise de contrôle d'une ou plusieurs grandeurs physiques d'un système en vue d'en imposer le comportement.

Les grandeurs physiques (vitesse, température, pression, courant, etc.), doivent être mesurées afin de vérifier leur état puis de déterminer à l'aide d'un traitement approprié l'action à entreprendre sur le système (installation de production, robot, etc.) pour qu'elles se comportent comme souhaité.



**Figure I.1 :** structure d'un système de régulation automatique

### I.2. Éléments constitutifs d'une boucle de régulation

#### I.2.1. Le système à commander

Ce système traduit en général un système physique possédant ses grandeurs d'entrées et ses grandeurs de sortie, il peut être décrit par diverse représentation.

#### I.2.2. L'actionneur

Est un organe qui convertit de l'énergie d'une forme en une autre pour agir sur le système à commander, il peut être électrique hydraulique ou pneumatique

#### I.2.3. Le correcteur

Est en quelque sorte le cerveau du système asservi, c'est lui qui génère les décisions pour accomplir la tâche désirée. Le choix de ce correcteur dépend principalement de la structure du système à commander et des performances désirées. En général, ce correcteur se base sur l'erreur du système bouclé pour générer l'action appropriée.

### **I.2.4. Le capteur**

Pour que le correcteur utilisé puisse assurer le suivi de la consigne imposée, l'emploi d'un organe de mesure est d'une grande utilité. Un tel organe est appelé capteur ; son rôle est de renseigner le correcteur sur le comportement dynamique de la grandeur commandée[3].

Malheureusement les capteurs physiques ont des points faibles qui peuvent dégrader le control d'un système.

### **I.3. Les problèmes communs des capteurs**

Il y a au moins quatre problèmes provoqués par les capteurs :

- Capteur trop cher ; le cout d'un capteur peut sensiblement augmenter tout le cout d'un système de contrôle.
- Le capteur et son câblage réduit la fiabilité d'un système de contrôle.
- La difficulté de mesurer quelque signaux pour raison d'environnement dure.
- Le capteur produit des erreurs significatives telles que le bruit stochastique, les erreurs cycliques et la réponse limitée.

On vue de l'importance et l'utilité des capteurs dans des systèmes de contrôle et de leurs problèmes indésirables il faut donc prévoir une solution.

Alors, pour augmenter ou remplacer les capteurs dans un système de contrôle des observateurs peuvent être utilisé.

### **I.4. Qu'est-ce qu'un observateur ?**

Les observateurs sont des algorithmes qui combinent les signaux disponibles avec d'autres connaissances du système de contrôle pour produire les signaux observés.

Ces signaux observés peuvent être plus précis, moins cher de produire et plus fiables que les signaux mesurés.

Pour comprendre la théorie des observateurs il faut passer par le contrôle des systèmes dans l'espace d'état (control modern).

## I.5. Notions sur l'espace d'état :

### I.5.1. L'état d'un système :

Est le plus petit ensemble de variables, de grandeurs, tel que la connaissance de cet ensemble à un instant quelconque  $t_0$ , ainsi que celle du signal d'entrée pour  $t \geq t_0$ , suffit pour la détermination complète de ce système pour  $t \geq t_0$ .

### I.5.2. Variables d'état:

Ce sont les variables, grandeurs qui constituent l'état du système, on considère que ces variables sont notées  $x_1, x_2, \dots, x_n$ ,  $n$  est l'ordre du modèle. L'ensemble  $\{x_1, x_2, \dots, x_n\}$  constitue l'état.

### I.5.3. Vecteur d'état :

On représente l'état par une concaténation de l'ensemble des variables d'état en un vecteur, de dimension  $n$ , que l'on note  $x = [x_1, x_2, \dots, x_n]^T$ .

### I.5.4. Espace d'état :

Il s'agit de l'espace vectoriel dans lequel le vecteur d'état  $x$  est susceptible d'évoluer, chaque instant de  $x$  étant associé à un point de cet espace.

### I.5.5. Commandabilité :

Un système est commandable si pour toute instance  $x_1$  du vecteur d'état, il existe un signal d'entrée  $u(t)$  qui permet au système de passer de l'état  $x_0$  à l'état  $x_1$  en un temps fini.

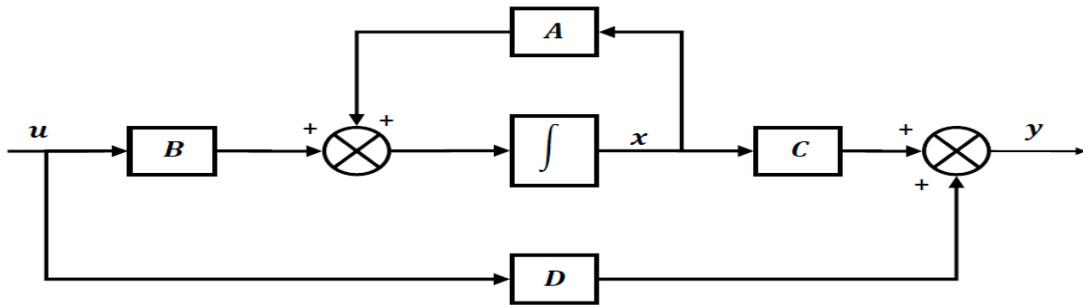
### I.5.6. Observabilité :

Un système est observable si, quel que soit  $t_0$ , il existe un intervalle de temps fini  $[t_0, t_1]$  tel que la connaissance de l'entrée  $u(t)$  et de la sortie  $y(t)$  sur cet intervalle permet de reconstituer  $x(t_0)$ .

## I.6. La commande par retour d'état :

Considérons le système linéaire invariant, dont la représentation d'état est :

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) + Du(t) \end{aligned} \tag{I.1}$$



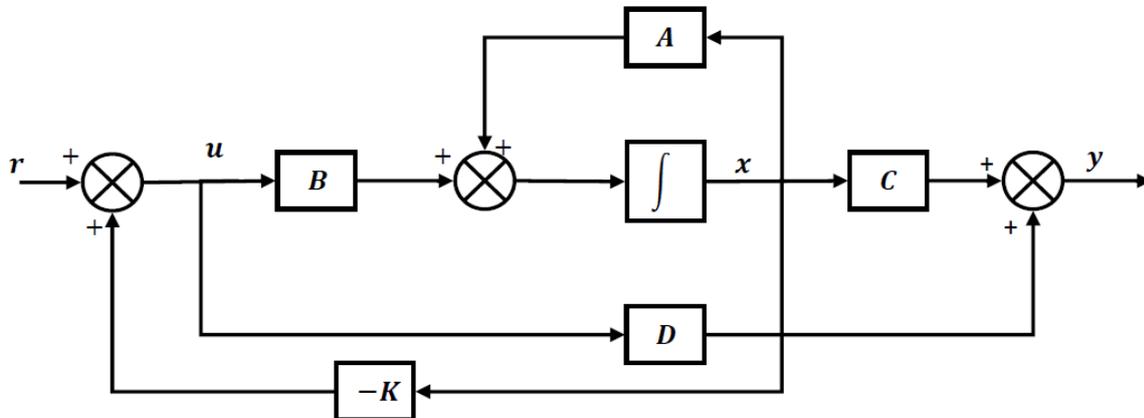
**Figure I.2 :** représentation interne d'un système

On définit la matrice  $M = [B, AB \dots \dots A^{n-1}B]$ , le système est commandable si et seulement si :  $\text{rang}(M) = n$

$n$  : L'ordre du système.

En désignant par  $r(t)$  le signal de référence, la commande par retour d'état consiste à choisir la matrice de gain tel que :  $u(t) = -Kx(t) + r(t)$ .

$K$  : Vecteur correcteur d'état.



**Figure I.3 :** structure d'une commande par retour d'état

Si on regarde attentivement le schéma d'asservissement dans la régulation moderne , on comprend que ce type de loi de commande ne correspond plus au schéma d'asservissement classique .

On cherche donc à modifier le comportement dynamique du processus en lui ajoutant une contre réaction calculée à partir du vecteur d'état.

## I.7. La synthèse du correcteur par retour d'état

Il existe pas mal de méthodes pour le calcul de K tel que le placement des pôles, méthode d'Ackermann ....etc.

### I.7.1. Placement de pôles

Le placement de pôles consiste à déterminer K de telle sorte que le système ait les pôles désirés ou, plus rigoureusement, de telle sorte que la matrice d'état en boucle fermée ait les valeurs propres spécifiées. Ceci permet d'agir de manière significative sur le comportement transitoire du système, en termes de temps de réponse, d'oscillations, etc [2].

La représentation du système en boucle fermée s'écrit comme suite :

$$\begin{aligned}\dot{x}(t) &= (A - BK)x(t) + B.r(t) \\ y(t) &= Cx(t)\end{aligned}\tag{I.2}$$

On considère que la matrice  $D = 0$ .

L'équation caractéristique du système est donné par :

$$|sI - (A - BK)| = 0\tag{I.3}$$

Supposant que nous décidons d'avoir des pôles en boucle fermée à,  $s = \mu_1, s = \mu_2 \dots s = \mu_n$  le polynôme caractéristique désiré est :

$$D_d(p) = \prod_{i=1}^n (s - \mu_i) = p^n + \alpha_{n-1} \cdot p^{n-1} + \dots + \alpha_1 \cdot p^1 + \alpha_0\tag{I.4}$$

Alors :

$$|sI - (A - BK)| = p^n + \alpha_{n-1} \cdot p^{n-1} + \dots + \alpha_1 \cdot p^1 + \alpha_0$$

La résolution de cette équation donne comme résultat les éléments de la matrice

### I.7.2. Formule d'Ackermann

$$K = [0 \ 0 \ \dots \ 0 \ 1] \cdot M^{-1} \cdot \phi(A)\tag{I.5}$$

$$\phi(A) = p^n + \alpha_{n-1} \cdot p^{n-1} + \dots + \alpha_1 \cdot p^1 + \alpha_0$$

Avec M la matrice de commandabilité,  $\alpha_i$  sont les coefficients de l'équation caractéristique de la boucle fermée désirée [1].

### I.7.3. Remarque

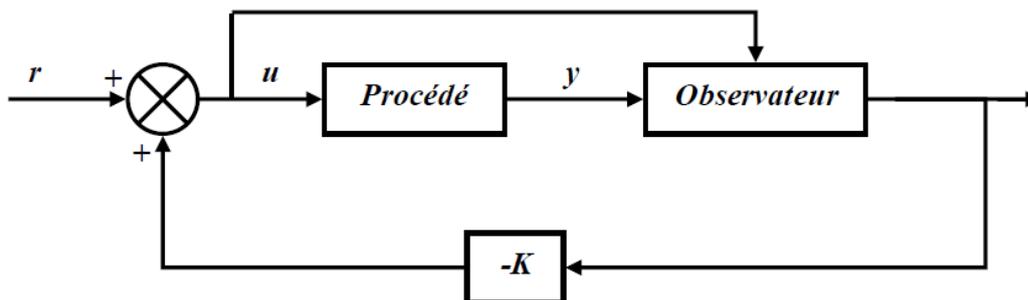
La loi de commande par retour d'état suppose que toutes les composantes  $x_i$  du vecteur d'état  $x$  sont accessibles à la mesure.

Dans le cas d'une commande sans capteur la grandeur à contrôler n'est pas accessible à la mesure, par conséquent la commande par retour d'état n'est plus applicable sans la disponibilité de cet état donc il fallait pensé à un autre moyen pour que le retour d'état soit fait.

### I.8. La commande par retour de sortie ou l'observateur

La théorie de l'observateur venait à point pour permettre la commande par retour d'état par l'estimation des états non accessibles.

L'observateur consiste à estimer les états du système à partir du signal de commande  $u$  et de la mesure des grandeurs de sorties  $y$ , pour reconstruire un vecteur  $\hat{x}$  qui soit aussi proche de  $x$  afin d'effectuer ensuite un retour d'état [3].



**Figure I.4** : structure d'une commande par retour de sortie

Le modèle de l'observateur s'appuie sur celui du système et il s'agit en fait de déterminer  $L$  de telle sorte que la dynamique de l'observateur soit plus rapide que celle du procédé.

Pour montrer le concept de l'observateur, considérons le système dont le modèle est donnée par :

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) \end{aligned} \quad (\text{I.6})$$

En notant par  $\hat{x}$  l'estimé de  $x$ , le modèle utiliser pour simuler la dynamique du système est le suivant :

$$\begin{aligned}\dot{\hat{x}}(t) &= A\hat{x}(t) + Bu(t) \\ \hat{y}(t) &= C\hat{x}(t)\end{aligned}\tag{I.7}$$

L'estimateur est décrit par l'équation suivante :

$$\dot{\hat{x}}(t) = A\hat{x}(t) + Bu(t) + L[y(t) - \hat{y}(t)]\tag{I.8}$$

## I.9. La synthèse de l'observateur

Les pôles dominants de l'observateurs doivent être placés à gauche des pôles dominants du correcteur, un rapport de 10 entre les pôles dominants de l'observateur et ceux de correcteur est largement suffisant pour assurer une très bonne convergence de l'estimation.

L'équation caractéristique est donné par :

$$|sI - (A - L.C)| = 0\tag{I.9}$$

Supposant que nous décidons d'avoir des pôles en boucle fermée pour l'observateur à

$s = \mu_1, s = \mu_2 \dots s = \mu_n$  le polynôme caractéristique désiré est :

$$D_d(p) = \prod_{i=1}^n (s - \mu_i) = p^n + \alpha_{n-1} \cdot p^{n-1} + \dots + \alpha_1 \cdot p^1 + \alpha_0$$

Alors :

$$|sI - (A - BK)| = p^n + \alpha_{n-1} \cdot p^{n-1} + \dots + \alpha_1 \cdot p^1 + \alpha_0\tag{I.10}$$

La résolution de cette équation donne comme résultat les éléments de la matrice L

## I.10. Conclusion

La structure de la commande en boucle fermée, peut prendre plusieurs configurations. La configuration la plus utilisée, quand le modèle employé est sous forme de fonction de transfert, souvent appelée la configuration cascade.

Par contre, quand le modèle est donné sous forme de modèle d'état, la structure utilisée est celle de la commande par retour d'état.

---

# Chapitre II : Etude et simulation du système à contrôler

---

## Chapitre II : Etude et simulation du système à contrôler

### II.1. Introduction

Notre but est de faire une commande en vitesse d'un moteur à courant continu sans capteur, nous allons commencer à bien connaître notre procédé qui est le moteur à courant continu à aimant permanent, avec une description générale, principe de fonctionnement ainsi que sa modélisation, ensuite en va attaquer la simulation avec différents structure de commande et au fur et à mesure on développera la théorie de l'estimation de vitesse.

### II.2. Description du système

La figure II.1 suivante représente le système à contrôler :



Figure II.1 : Système à contrôler

### II.3. Modélisation du MCC à aimant permanents

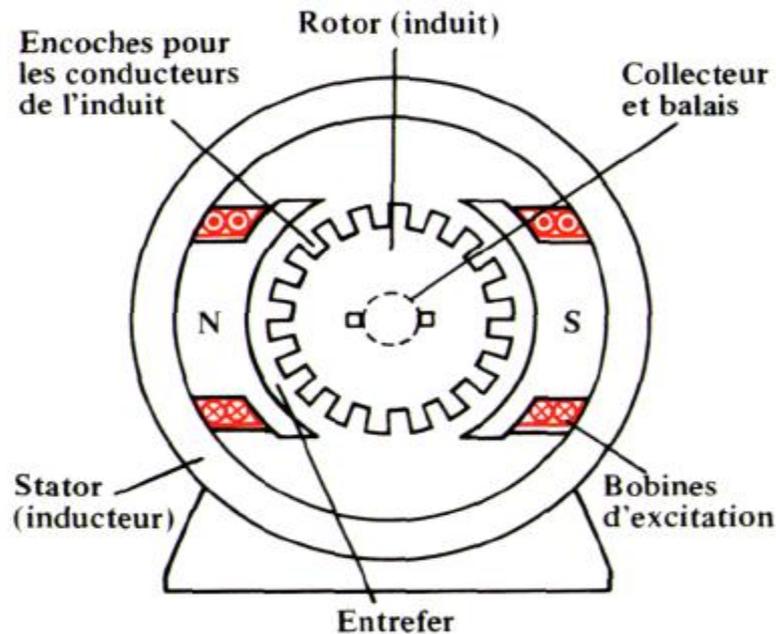
Le moteur à courant continu est un actionneur de type électrique, qui transforme l'énergie électrique en énergie mécanique.

#### ➤ Structure

Ces moteurs sont principalement constitués de deux parties principales( voir la figure II.2 ) :

- Un circuit magnétique avec son enroulement inducteur fixe, crée le champ inducteur ; l'ensemble est dit STATOR.
- Un enroulement d'induit tournant dit ROTOR.

A ces deux parties s'ajoute un troisième élément : le COLLECTEUR qui est constitué d'un ensemble de lames reliées, sur lesquelles sont soudés deux faisceaux de conducteur d'induit.



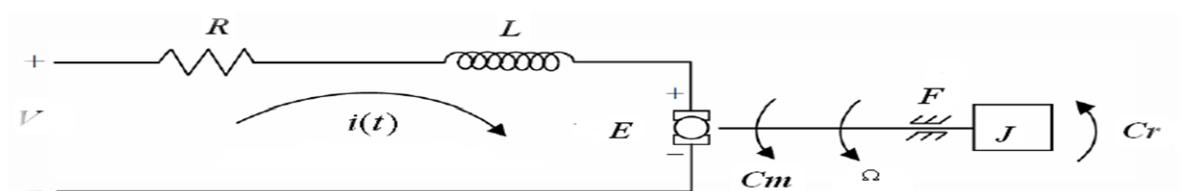
**Figure II.2** Constitution du moteur à courant continu

### ➤ Couplage

Il est possible de coupler de plusieurs manières les bobinages inducteurs et les bobinages induits, on distingue quatre types de couplage du moteur à courant continu :

- Moteur à excitation composée.
- Moteur à excitation séparée.
- Moteur à excitation shunt.
- Moteur à excitation série.

Puisqu'on travaille avec un MCC à aimant permanent, donc la partie inducteurs est négligeable, dans ce cas notre moteur peut être représenté par le schéma suivant :



**Figure II.3** : Schéma électrique d'un MCCAP

Avec :

- $V$  tensions appliquée au moteur [v]  
 $E$  force électromotrice [v]  
 $I$  le courant d'induit [a]  
 $R$  la résistance d'induit [ $\Omega$ ]  
 $L$  la self de l'induit [H]  
 $F$  le frottement [N.m]  
 $C_m$  Couple moteur généré [N.m]  
 $C_r$  Couple résistant [N.m]  
 $J$  l'inertie du système [Kg.m<sup>2</sup>]  
 $\Omega$  vitesse de rotation de moteur [rad/min]  
 $K_t$  Constante de couple moteur  
 $K_b$  Constante de force contre électromotrice

➤ **Les équations du moteur sont**

**1. Equation électrique de l'induit**

$$V(t) - E(t) = L \frac{dI(t)}{dt} + R \cdot I(t) \quad (\text{II.1})$$

Avec :

$$E(t) = K_b \cdot \Omega(t) \quad (\text{II.2})$$

**2. Equation mécanique**

$$C_m = J \frac{d\Omega(t)}{dt} + F \cdot \Omega(t) + C_r \quad (\text{II.3})$$

Avec :

$$C_m = K_t \cdot I(t) \quad (\text{II.4})$$

- Les équations dans le domaine laplacien donne :

$$V(s) - E(s) = R \cdot I(s) + L \cdot S \cdot I(s) \Rightarrow I(s) = \frac{V(s) - E(s)}{R + L \cdot S} \quad (\text{II.5})$$

On a  $\tau_{ele}$  : la constante de temps électrique

Avec :

$$\tau_{ele} = \frac{L}{R} \Rightarrow I(s) = \frac{V(s) - E(s)}{R(1 + \tau_{ele} \cdot S)}$$

On a  $\tau_{mec}$  : la constante de temps mécanique

$$C_m - C_r = JS\Omega(s) \Rightarrow \Omega(s) = \frac{C_m - C_r}{F + JS} \quad (\text{II.6})$$

Avec :

$$\tau_{mec} = \frac{J}{F} \Rightarrow \Omega(s) = \frac{C_m - C_r}{F(1 + \tau_{mec} \cdot S)} \quad (\text{II.6})$$

On a  $\tau_{mec}$  : la constante de temps mécanique

#### II.4. Représentation du MCC en boucle ouverte :

D'après les relations trouvées, on peut faire une représentation de notre moteur en boucle ouverte :

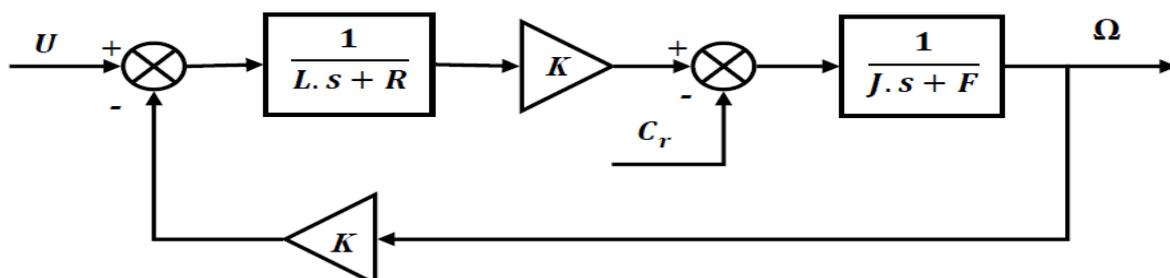


Figure II.4 :Schéma fonctionnel du MCC

On considère que le  $C_r = 0$  la fonction de transfert qui relie la sortie (vitesse) à l'entrée (tension) est la suivante :

$$\frac{\Omega(s)}{U(s)} = \frac{K}{(R + L.S).(F + J.S) + K^2} \Rightarrow \frac{\Omega(s)}{U(s)} = \frac{G}{1 + \tau_{em}.S + \tau_{em}.\tau_{el}.S^2} \quad (\text{II.7})$$

Avec :

$$K = K_t = K_b$$

$$G = \frac{K}{K^2 + R.F} \quad (\text{II.8})$$

Et :

$$\tau_{em} = \frac{R.J + L.F}{K^2 + R.F} \quad (\text{II.9})$$

Et :

$$\tau_{el} = \frac{L.J}{R.J + L.F} \quad (\text{II.10})$$

On peut utiliser l'approximation suivante :

$$\tau_{em} \cong \tau_{em} + \tau_{el} \quad (\text{II.11})$$

Ce qui donne :

$$1 + \tau_{em}.S + \tau_{em}.\tau_{el}.S^2 \cong (1 + \tau_{el}.S)(1 + \tau_{em}.S) \quad (\text{II.12})$$

Alors notre système devient :

$$G(s) = \frac{G}{(1 + \tau_{el}.S)(1 + \tau_{em}.S)} \quad (\text{II.13})$$

On peut l'écrire aussi sous la forme suivante :

$$G(s) = \frac{G/\tau_{em} \cdot \tau_{el}}{(S + 1/\tau_{el})(S + 1/\tau_{em})} = \frac{K \cdot \omega_n^2}{S^2 + 2 \cdot \zeta \cdot \omega_n \cdot S + \omega_n^2} \quad (\text{II.14})$$

Par identification on obtient :

$$S_1 = -1/\tau_{el} \quad S_2 = -1/\tau_{em} \quad \omega_n = \sqrt{1/\tau_{em} \cdot \tau_{el}} \quad \zeta = \frac{\tau_{em} + \tau_{el}}{2 \cdot \omega_n \cdot \tau_{em} \cdot \tau_{el}}$$

- Les paramètres du moteur sont [8]:

$$L = 100 \cdot 10^{-6} \text{ Henry}$$

$$R = 0.61 \text{ Ohm}$$

$$J = 1.84 \cdot 10^{-4} \text{ Kg} \cdot \text{m}^2$$

$$F = 1.3369 \cdot 10^{-2} \text{ N} \cdot \text{m} \cdot \text{s}$$

$$K_t = 0.1013 \text{ N} \cdot \text{m} / \text{A}$$

$$K_b = 0.1012 \text{ V} \cdot \text{s} / \text{rad}$$

Application numérique :

$$S_1 = -6172.83 \quad S_2 = -162.07 \quad \omega_n = 1000(\text{rad/sec}) \quad \zeta = 3.16$$

### II.4.1. Modélisation du capteur

La génératrice tachymétrique délivre une tension proportionnelle à sa vitesse de rotation, sa fonction de transfert est comme celle d'un système du premier ordre caractérisé par une constante de temps mécanique  $\tau_m$ :

$$G(s) = \frac{K_\omega}{1 + \tau_m \cdot S} \quad (\text{II.15})$$

Avec : gain de la tachymétrie  $K_\omega$

Généralement  $\tau_m$  est très petite, donc on peut la négliger en considérant l'approximation suivant :  $1 + \tau_m \cdot S \approx 1$ .

La fonction de transfert devienne :

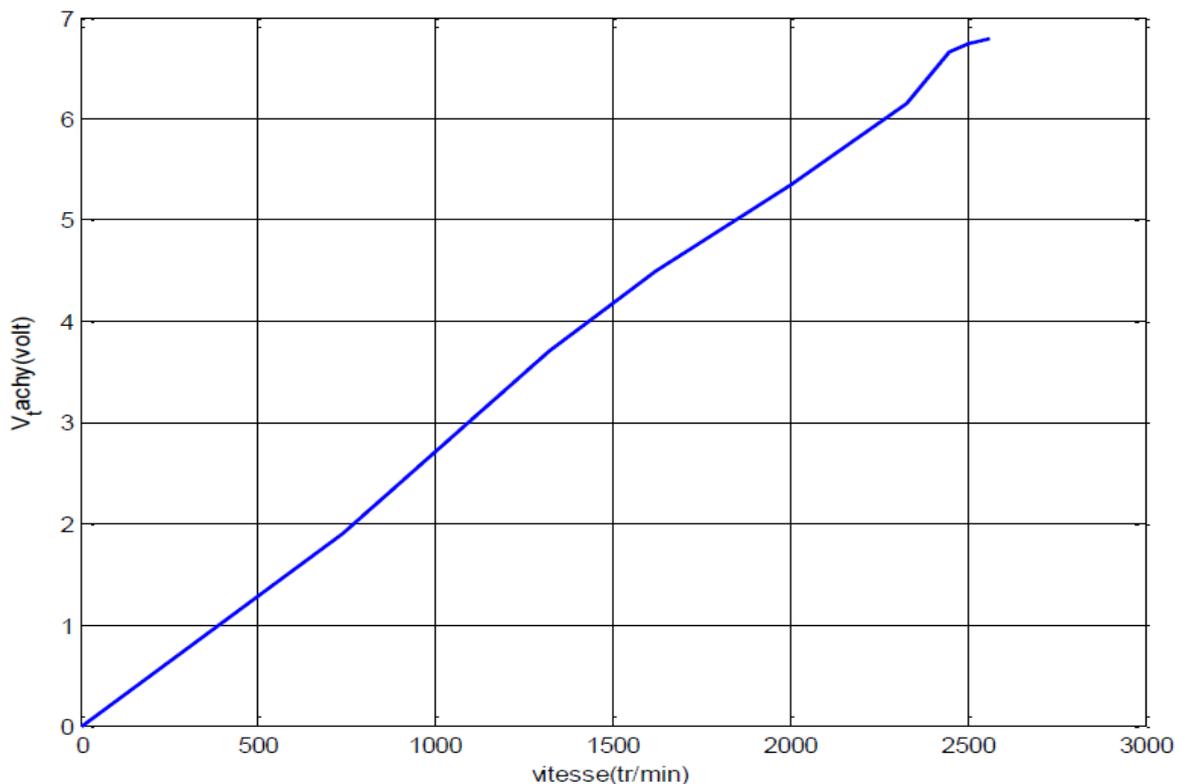
$$G(s) = K_{\omega} \quad (\text{II.16})$$

- **Détermination du gain**

Pour la détermination du gain de la tachymétrie on fait varier la tension d' induit du moteur et on a relevé la tension aux bornes de la tachymétrie, et on a tracé la caractéristique  $V_{\omega} = f(\omega)$ .

$V_{\omega}$ : Tension délivré par la génératrice tachymétrique.

$\omega$ : vitesse prise par la tachymétrie manuelle en tr/min.



**Figure II.5** : la caractéristique de  $V_{\omega} = f(\omega)$

On remarque bien que la caractéristique présenté sur la figure II.5 est linéaire, donc pour déterminer le gain il suffit de prendre 2 points et calculer la tangente de l'angle situé entre l'axe des 2 points et la caractéristique :

$$K_{\omega} = (5,3 - 2,6)/(2000 - 1000) = 0,0027 \text{ Volt/tr/min} \quad (\text{II.17})$$

### II.4.2. Modélisation du hacheur

Les hacheurs sont des convertisseurs continu-continu ont pour fonction de fournir une tension continue variable à partir d'une tension continue fixe et produisent aux bornes d'une charge une tension unidirectionnelle de valeur moyenne réglable. [6]

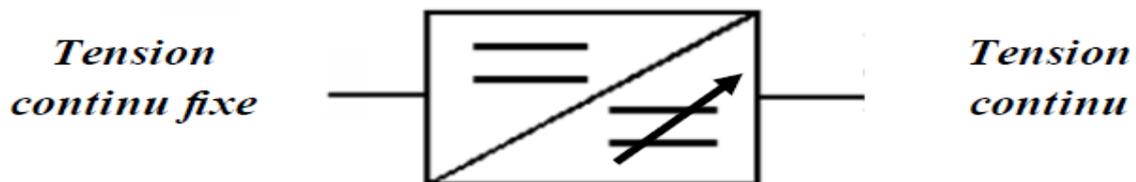


Figure II.6 : Convertisseur continu-continu

#### II.4.2.1. Hacheur série

Ce nom est lié au fait que la tension moyenne de sortie est inférieure à celle de l'entrée. Il comporte un interrupteur H à amorçage et à blocage commandés (transistor bipolaire, transistor MOS ou IGBT...) et un interrupteur à blocage et amorçage spontanés (DRL).

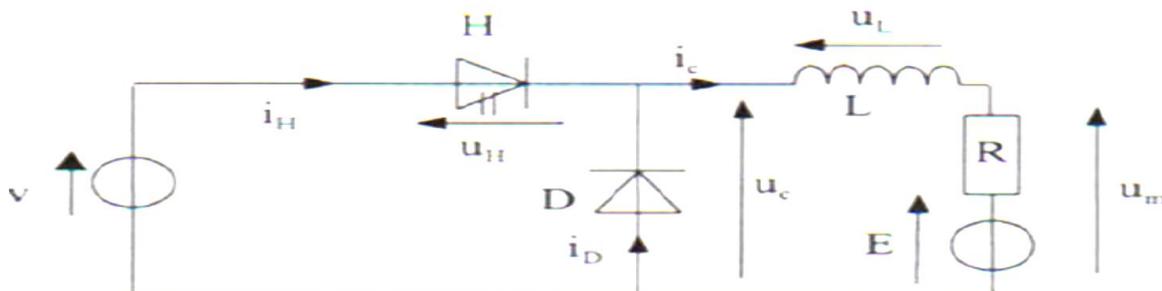


Figure II.7 : hacheur série

#### II.4.2.2. Principe de fonctionnement d'un hacheur

Le cycle de fonctionnement est de période de hachage  $T$  ( $T=1/f$ ), il comporte deux étapes :

- Lors de la première, l'interrupteur est fermé, la diode polarisée en inverse est bloquée. Cette phase dure de  $0$  à  $\alpha T$ . avec  $\alpha$  compris entre  $0$  et  $1$ .  $\alpha$  est appelé rapport cyclique.
- Lors de la seconde, l'interrupteur est ouvert, la diode devient passante. Cette phase dure de  $\alpha T$  à  $T$ .



**Figure II.8 :** chronogramme de la fermeture et de l'ouverture

Le hacheur série est souvent employé pour commander un moteur à courant continu, on rappelle que la vitesse d'un tel moteur est proportionnelle à la tension d'alimentation.

Pour un rapport cyclique  $\alpha$  la tension moyenne appliquée au moteur est :

$$V(t) = V_0 \cdot \alpha t \quad (\text{II.18})$$

La fonction de transfert du hacheur est :

$$\frac{V(S)}{\alpha(S)} = V_0 \quad (\text{II.19})$$

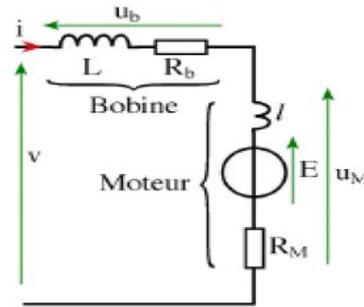
$V_0$ : Gain du hacheur (ou tension appliquée au hacheur)

#### II.4.2.3. Remarque :

Pour un bon fonctionnement du moteur, il est préférable que le courant d'induit soit continu, d'où la présence d'une bobine de lissage. Si son inductance est suffisamment grande, on pourra considérer le courant comme constant.

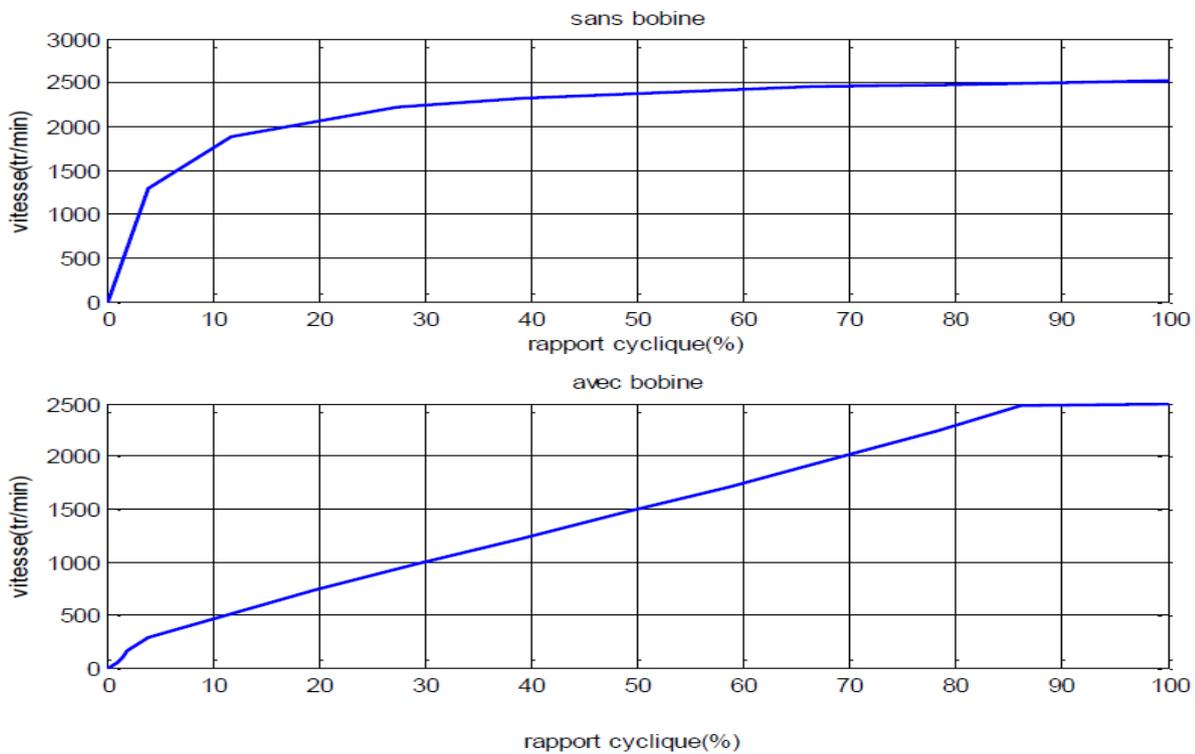
Finalement la f.é.m. du moteur et donc la vitesse peuvent être réglés grâce au rapport cyclique par la relation (II.19) .

Le modèle électrique complet du moteur et de la bobine de lissage est représenté sur la figure II.9:



**Figure II.9** : moteur avec bobine de lissage

Pour justifier cette remarque on a relevé expérimentalement la caractéristique  $\omega = f(\alpha)$  avec et sans bobine de lissage :



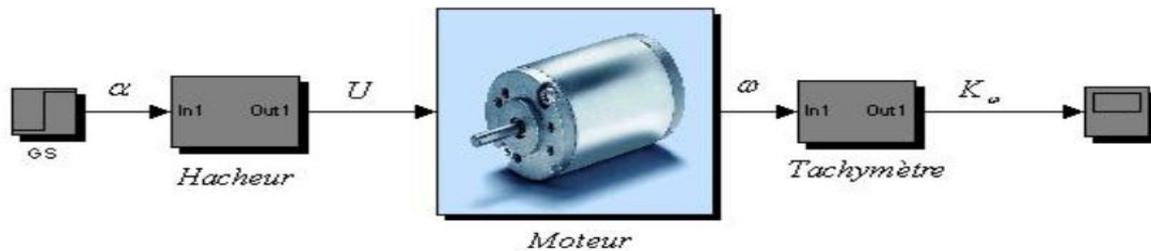
**Figure II.10** :modélisation du hacheur

On remarque bien que avant l'ajout de l'inductance la caractéristique  $\omega = f(\alpha)$  n'était pas linéaire, mais après l'ajout de l'inductance la caractéristique est devenue linéaire donc la vitesse est proportionnelle au rapport cyclique.

## II.5. Simulation du moteur :

### II.5.1. Boucle ouverte :

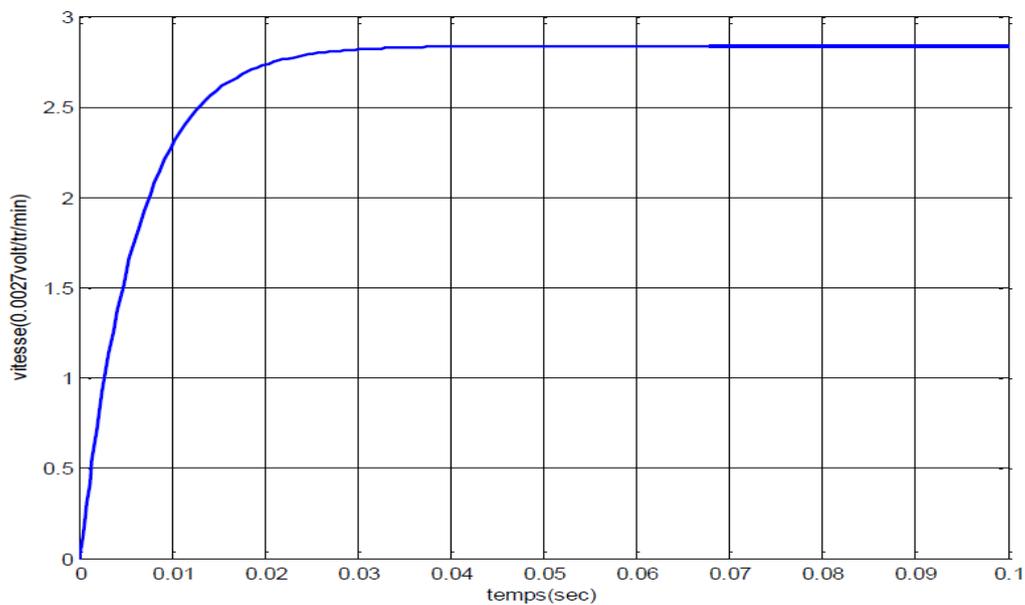
Pour qu'on puisse savoir le comportement dynamique du moteur on fait une simulation en boucle ouverte avec le simulink de MATLAB.



**Figure II.11** : schéma bloc de la boucle ouverte

La figure II.12 ci-dessus montre le comportement dynamique et statique de moteur avec un démarrage à vide et l'application d'un rapport cyclique de 50%.

réponse en boucle ouverte pour un échelon  $\alpha=50\%$

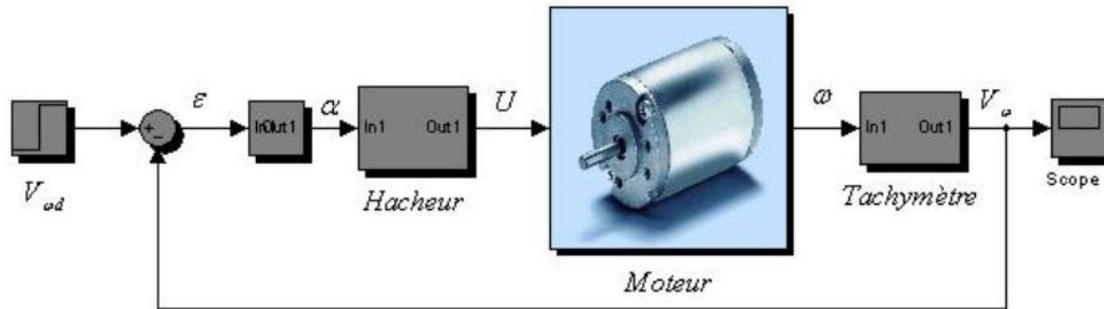


**Figure II.12** : réponse en boucle ouverte pour un échelon  $\alpha=50\%$

On remarque que le moteur atteint sa valeur finale au bout de 0.05s avec un temps d'établissement de 0.0236 s .

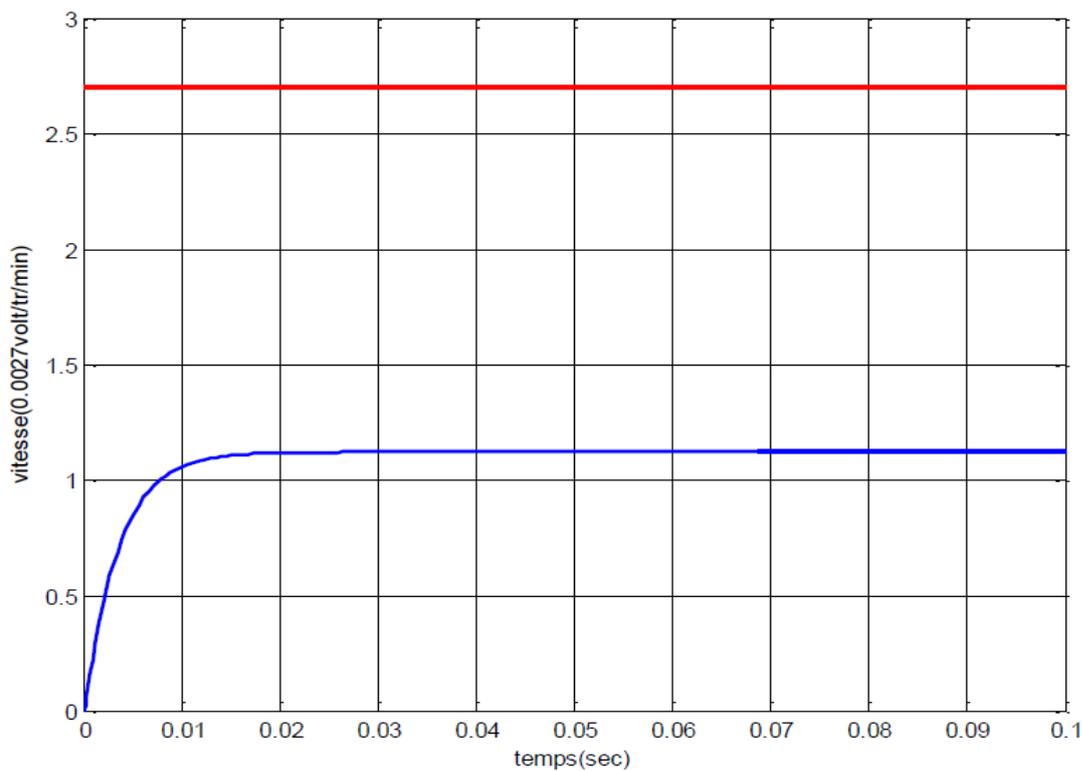
### II.5.2. Boucle fermée :

La boucle fermée permet de commander le moteur avec une vitesse désirée non pas avec la tension d'induit.



**Figure II.13** :schéma bloc de la boucle fermée

La figure II.14 ci-dessus montre la réponse du moteur en boucle fermée suite à une consigne de 2.7volt.



**Figure II.14** : réponse du moteur en boucle fermée

Une erreur d'environ 1.5 volt qui correspond à 555tr/min est introduite en régime permanent.

Suite à ce comportement non souhaitable du moteur on a pensée à améliorer les caractéristiques dynamiques et statiques du moteur, cela nous a amené à faire une régulation simple et utilisé dans la plupart des applications de control c'est l'introduction d'un correcteur PI dans la chaine directe.

## II.6. Régulation PI :

L'expression d'un tel correcteur est donnée par :

$$C(s) = k_p + \frac{k_i}{s} = k_p \cdot \frac{(s + z)}{s} \quad (\text{II.20})$$

Avec :

$$z = \frac{k_i}{k_p} \quad (\text{II.21})$$

En général, le correcteur PI est utilisé pour assurer les spécifications suivantes [2] :

- Une réponse apériodique.
- Un temps de réponse donné.
- Une erreur en régime permanent donnée.
- Un système stable.

Le choix des paramètres de réglage a besoin d'une méthode simple et suffisamment précise, il est évident que le choix des réglages dépend aussi des performances souhaitées.

Alors, pour cela on a décidé d'imposer un cahier de charges pour que le système réponde à ces conditions, notre cahier des charges est le suivant :

- Un dépassement  $MP \leq 5\% \Rightarrow \zeta = 0.707$ .
- Un temps d'établissement à 2%  $\leq 0.02s \Rightarrow \omega_n = 282.88 \text{ rad/s}$ .
- Une erreur statique nulle.

Le choix de la méthode de synthèse était de faire une synthèse dans le lieu des racines, la procédure à suivre est comme suite :

La fonction de transfert de notre système est :

$$G(s) = V_0 \frac{9807872}{(s + 6172.83)(s + 162.07)} K_\omega \quad (\text{II.22})$$

La fonction de transfert en boucle ouverte du système compensé est donnée par :

$$G(s) = \frac{K \cdot (s + z)}{s \cdot (s + 6172.83)(s + 162.07)} \quad (\text{II.23})$$

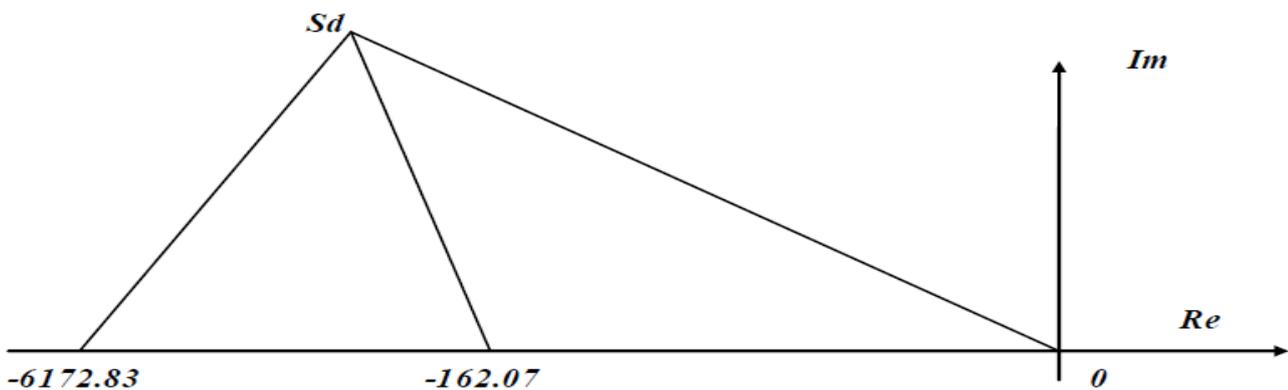
Avec :

$$K = 9807872 \cdot k_p \cdot V_0 \cdot K_\omega \quad (\text{II.24})$$

D'après les performances désirées, on déduit le pôle dominant suivant :

$$s_d = -\zeta\omega_n + \omega_n\sqrt{1 - \zeta^2} \quad (\text{II.25})$$

$$s_d = -200 + j \cdot 200$$



**Figure II.15** :représentation des pôles

L'emplacement du zéro du correcteur est déterminé à partir de la condition des angles :

$$\alpha = (1 + 2q)\pi - \sum \alpha_i + \sum \beta_i \quad (\text{II.26})$$

Avec  $q=0$

$$\alpha = \pi + \arg(S) + \arg(S_d + 162.07) + \arg(S_d + 6172.83) = 57.65 \quad (\text{II.27})$$

Et de la notation :

$$|z| = |\sigma_{s_d}| + \frac{I_m(S_d)}{\text{tg}(\alpha)} = 200 + \frac{200}{1.57} = 326.67 \quad (\text{II.28})$$

La valeur du gain  $K$  qui procure le pôle dominant  $S_d$  est donnée par :

$$K = \frac{|S_d||S_d + 162.07||S_d + 6172.83|}{|S_d + z|} = \frac{(282.84)(203.56)(5976.16)}{236.73} = 1453457 \quad (\text{II.29})$$

Les paramètres du correcteur PI sont alors donnés par :

$$k_p = \frac{K}{9807872 \cdot V_0 \cdot K_\omega} = 0.19 \quad (\text{II.30})$$

$$k_i = k_p \cdot z = 62.58$$

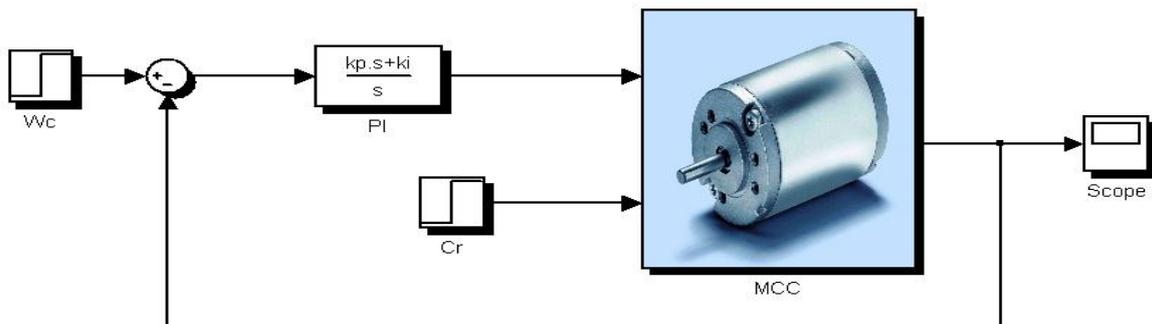
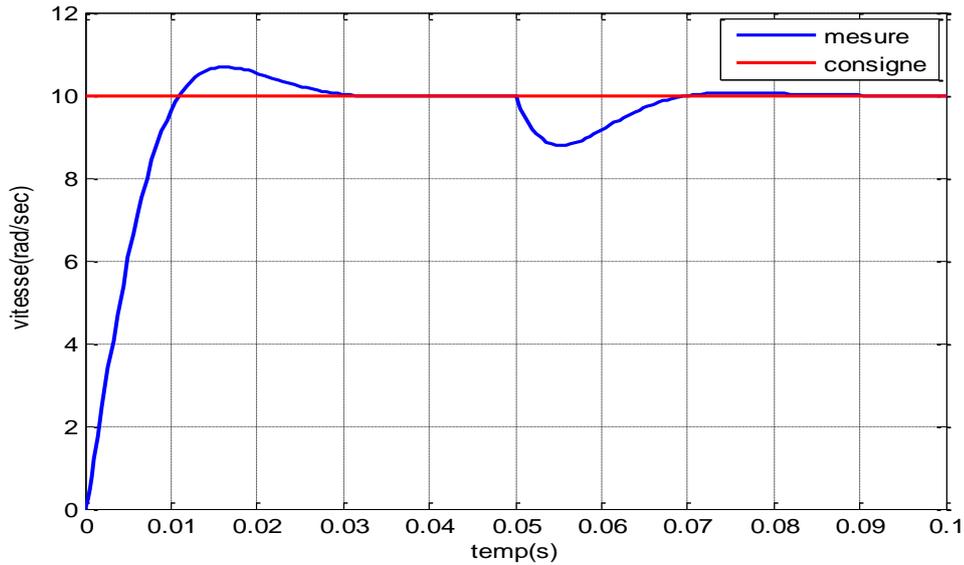


Figure II.16 :schéma bloc de la régulation PI

La réponse du système en boucle fermée est sur la figure II.17 suivante :



**Figure II.17** : réponse avec régulation pour une consigne de 10 (rad/sec)

On remarque que la vitesse atteint son point de consigne avec un temps d'établissement inférieur à 0.012s , un dépassement d'environ 5% et une erreur en régime statique nulle , le fonctionnement de notre régulateur PI est bon car il satisfait notre cahier des charges imposé.

Passons maintenant à notre problème qui est de faire une commande de vitesse sans capteur, commençons par la représentation du système dans l'espace d'état.

## II.7. Représentation du système dans l'espace d'état

Nous choisissons comme variables d'état le courant d'induit  $I$  et la vitesse  $\Omega$  notre vecteur d'état devient alors :

$$x(t) = \begin{bmatrix} I \\ \Omega \end{bmatrix} \quad (\text{II.31})$$

D'après les équations mécaniques et électriques du moteur on obtient la représentation interne suivante :

$$\begin{cases} \dot{x}(t) = Ax(t) + Bu(t) \\ y(t) = Cx(t) \end{cases} \Rightarrow \begin{cases} \dot{x}(t) = \begin{bmatrix} -R/L & -K/L \\ K/L & -F/J \end{bmatrix} x(t) + \begin{bmatrix} 1/L \\ 0 \end{bmatrix} u(t) \\ y(t) = [0 \ 1] \end{cases} \quad (\text{II.32})$$

Application numérique des paramètres du système :

$$\begin{cases} \dot{x}(t) = \begin{bmatrix} -6100 & -1012 \\ 550 & -72.6 \end{bmatrix} x(t) + \begin{bmatrix} 10000 \\ 0 \end{bmatrix} u(t) \\ y(t) = [01] \end{cases} \quad (\text{II.33})$$

### ➤ Commandabilité

Dans un premier temps on va vérifier si notre système est commandable ou non, le système est d'ordre 2. Pour cela on a utilisé la commande **Ctrb** de MATLAB :

```
A=[-R/L -K/L;K/J -F/J]
B=[1/L;0]
C=[0 1]
D=0
CO=Ctrb(A,B)
if rank(CO)==rank(A)
    display(' système commandable ')
else
    display(' système non commandable ')
end;
```

Le résultat du scripte montre que le rang de la matrice de commandabilité est égal au rang de la matrice A , donc le système est commandable .

### ➤ L'observabilité

Passons maintenant à l'étude de l'observabilité. La commande **Obsv** de MATLAB permet de vérifier l'observabilité :

```
A=[-R/L -K/L;K/J -F/J]
B=[1/L;0]
C=[0 1]
D=0
O=obsv(A,C)
if rank(O)==rank(A)
    display(' système observable ')
else
    display(' système non observable ')
end;
```

Le résultat du scripte montre que le rang de la matrice d'observabilité est égal au rang de la matrice A ,  
Donc le système est observable.

## II.8. Commande par retour d'état

On cherche à déterminer le vecteur de retour d'état  $k$ . On garde le même cahier de charges précédent alors le polynôme caractéristique désiré devient :

$$\begin{aligned}\psi_d(s) &= S^2 + 2\zeta\omega_n \cdot S + \omega_n^2 \\ \psi_d(s) &= S^2 + 400S + 80021\end{aligned}\quad (\text{II.34})$$

L'équation caractéristique du système est :

$$|sI - A + Bk| = 0 \quad (\text{II.35})$$

Pour le calcul de la matrice du correcteur d'état on a utilisé des commandes MATLAB , après calcul on obtient :

$$k = [-0.5773 \quad -0.0911] \quad (\text{II.36})$$

Le schéma de simulation est le suivant :

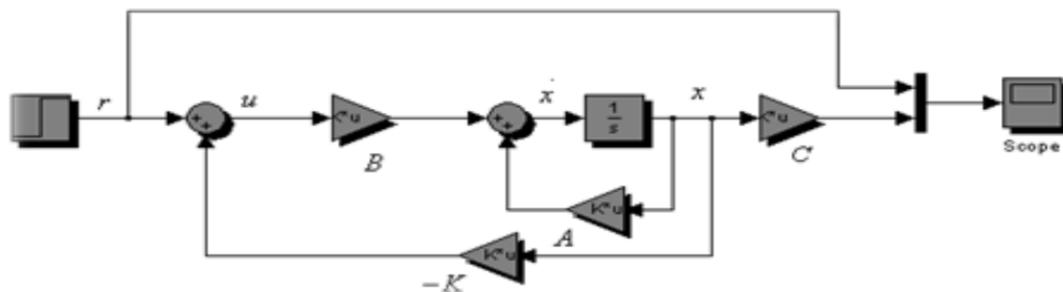


Figure II.18 : schéma bloc du retour d'état

La réponse du système est représentée sur la figure II.19 ci-dessus :

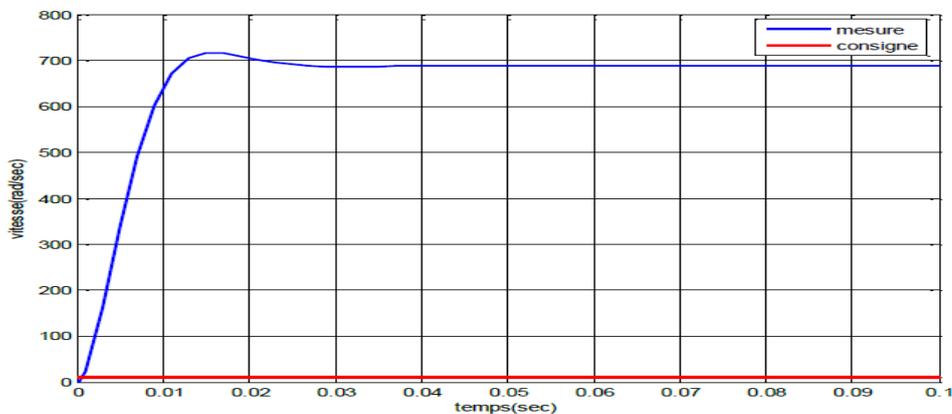
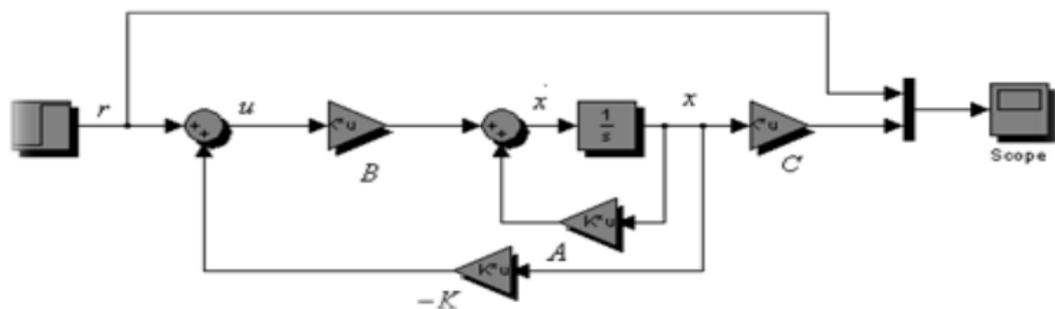


Figure II.19 : réponse avec retour d'état.

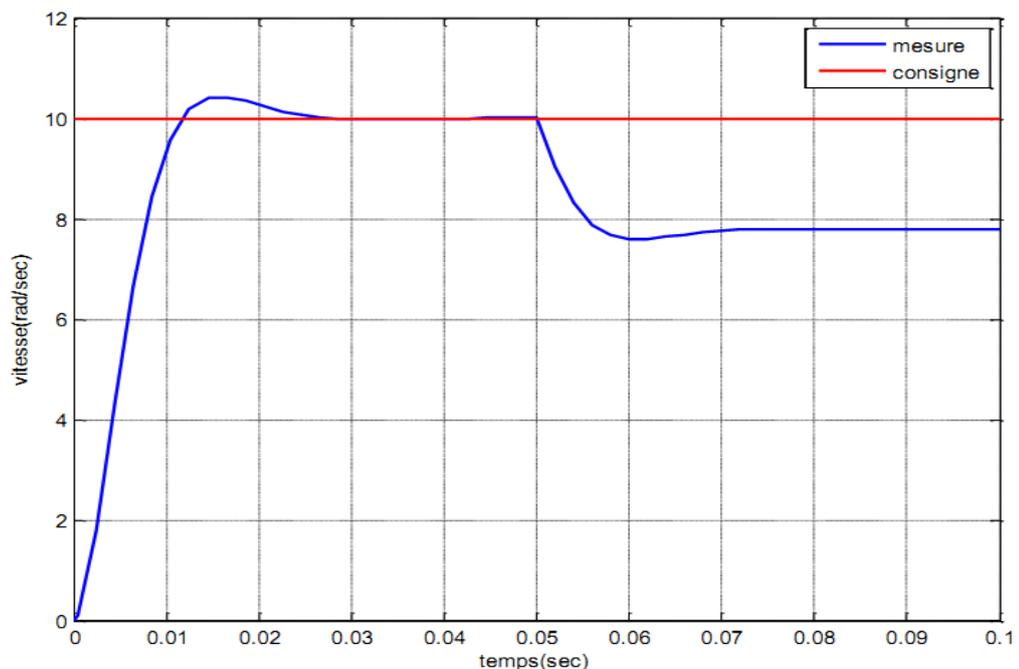
D'après le résultat on voit bien qu'il y a une erreur très importante entre la mesure et la consigne dans le régime statique, pour annuler cette erreur on doit introduire un **précompensateur** (gain) juste après la consigne. Ce précompensateur ne change pas les pôles du système bouclé, il permet de mettre en correspondance la consigne avec les variables d'états. Sa méthode de calcul est comme suite :

$$\begin{aligned} H &= (A - (B * K)) \\ A_{inv} &= \text{inv}(H) \\ H &= \text{inv}(-C * A_{inv} * B) \end{aligned}$$



**Figure II.20** : retour d'état avec précompensateur

La figure ci-dessus montre la réponse du système avec retour d'état et précompensateur plus l'application d'une charge à  $t=0.05s$ .



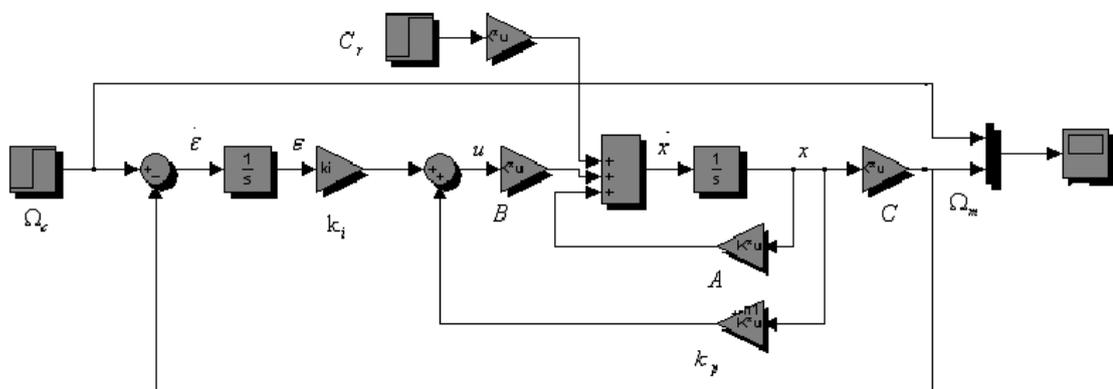
**Figure II.21** : réponse avec précompensateur

On remarque bien que l'erreur en régime statique est annulée, mais un autre problème est encore apparu, c'est que après l'application d'une charge le retour d'état est incapable de maintenir sa vitesse consigne.

Pour régler ce problème majeur le système peut être augmenté en utilisant un correcteur PI et retour d'état.

## II.9. Commande à retour d'état et bouclage intégral

La structure de commande par retour d'état et intégrale peut être représentée comme le montre la figure II.22 suivante :



**Figure II.22** : schéma du retour d'état intégral (PI)

Le système en boucle fermée est décrit de la manière suivante [3]:

$$\begin{cases} \dot{x} = A \cdot x + B \cdot u \\ y = \Omega = C \cdot x \end{cases} \quad (\text{II.37})$$

$$\begin{aligned} u &= k_i \cdot \varepsilon - k_p \cdot x \\ \dot{\varepsilon} &= \Omega_c - \Omega = \Omega_c - C \cdot x \end{aligned} \quad (\text{II.38})$$

Ce qui amène à la représentation d'état :

$$\begin{bmatrix} \dot{x} \\ \dot{\varepsilon} \end{bmatrix} = \begin{bmatrix} A & 0 \\ -C & 0 \end{bmatrix} \cdot \begin{bmatrix} x \\ \varepsilon \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} \cdot u + \begin{bmatrix} 0 \\ 1 \end{bmatrix} \cdot \Omega_c \quad (\text{II.39})$$

En remplaçant  $u$  par  $k_i \cdot \varepsilon - k_p \cdot x$  on obtient :

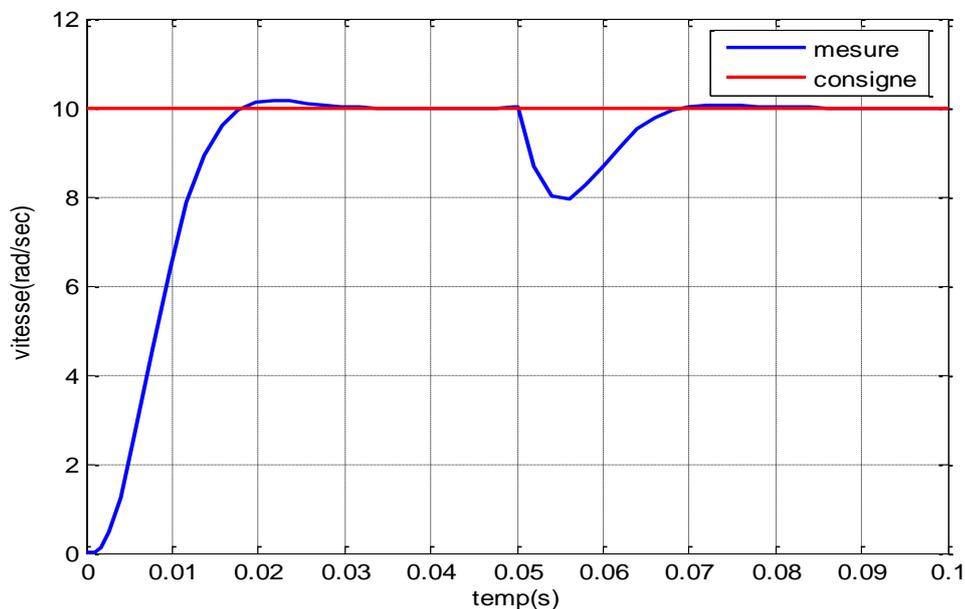
$$\begin{bmatrix} \dot{x} \\ \dot{\varepsilon} \end{bmatrix} = \begin{bmatrix} A - B \cdot k_p & B \cdot k_i \\ -C & 0 \end{bmatrix} \cdot \begin{bmatrix} x \\ \varepsilon \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} \cdot \Omega_c \quad (\text{II.40})$$

Notre système est devenue un système d'ordre 3, donc pour le calcul des paramètres  $k_i$  et  $k_p$  on doit imposer un 3<sup>ème</sup> pôle réel, stable et rapide.

Après l'application de la méthode de placement des pôles on a obtenu les valeurs suivantes :

$$\begin{cases} k_i = 4.3648 \\ k_p = [-0.5473 \quad -0.0731] \end{cases} \quad (\text{II.41})$$

La réponse du système est sur la figure II.23 suivante, avec l'application d'un échelon de 10rad/sec et un couple résistant de 0.1 à  $t=0.05$ .



**Figure II.23** : réponse avec retour d'état intégral

On remarque bien que le système suit la consigne, et annule l'effet du couple résistant. A ce niveau-là on a amélioré la réponse du système avec le retour d'état, maintenant on va ignorer la mesure de vitesse et la remplacer par un estimateur qui joue le rôle d'un capteur de vitesse.

## II.10. Commande par retour d'état observé

Dans ce cas-là il faut que la dynamique de l'observateur soit plus rapide de 2 à 5 fois que la dynamique du système en boucle fermée, alors l'équation caractéristique désirée de l'observateur devienne :

$$\begin{cases} \zeta = 0.707 \\ \omega_n = 1666 \text{ rad/s} \\ \varphi_d(s) = S^2 + 2\zeta\omega_n S + \omega_n^2 \\ \varphi_d(s) = S^2 + 2352S + 2765952 \end{cases} \quad (\text{II.42})$$

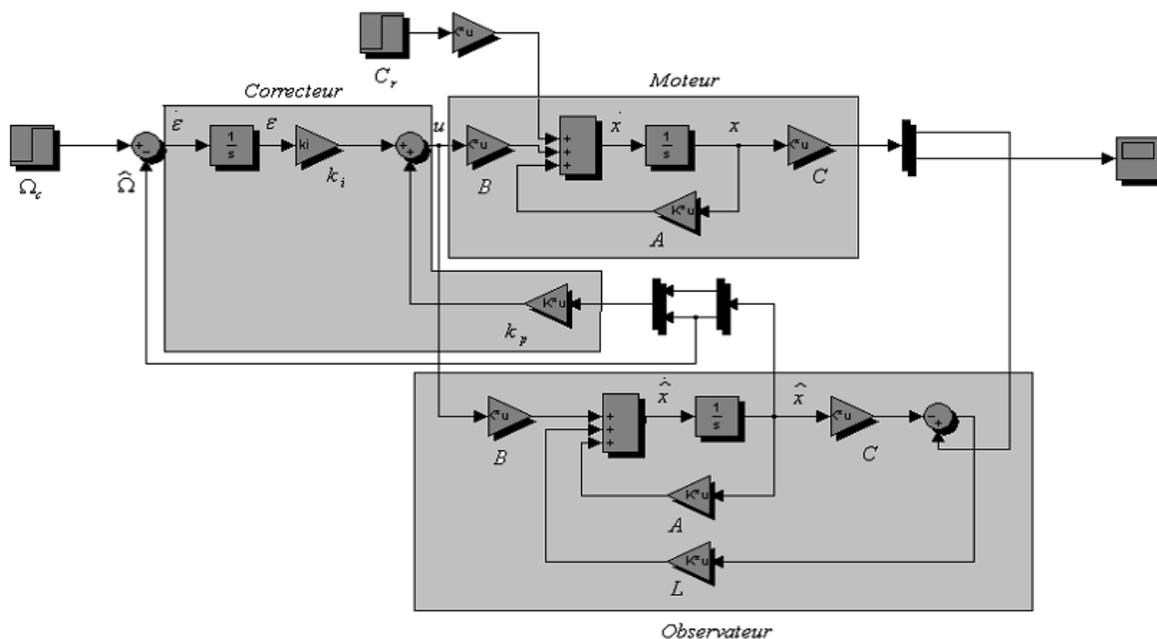
On va choisir comme sortie du moteur le courant d'induit I la matrice de sortie devienne :

$$C = [1 \ 0] \quad (\text{II.43})$$

Notre estimateur est basé sur l'erreur de sortie du système donc de l'erreur de courant d'induit I, le calcul de l'estimateur a été fait par la méthode de placement des pôles et on a obtenue la matrice d'observation suivante :

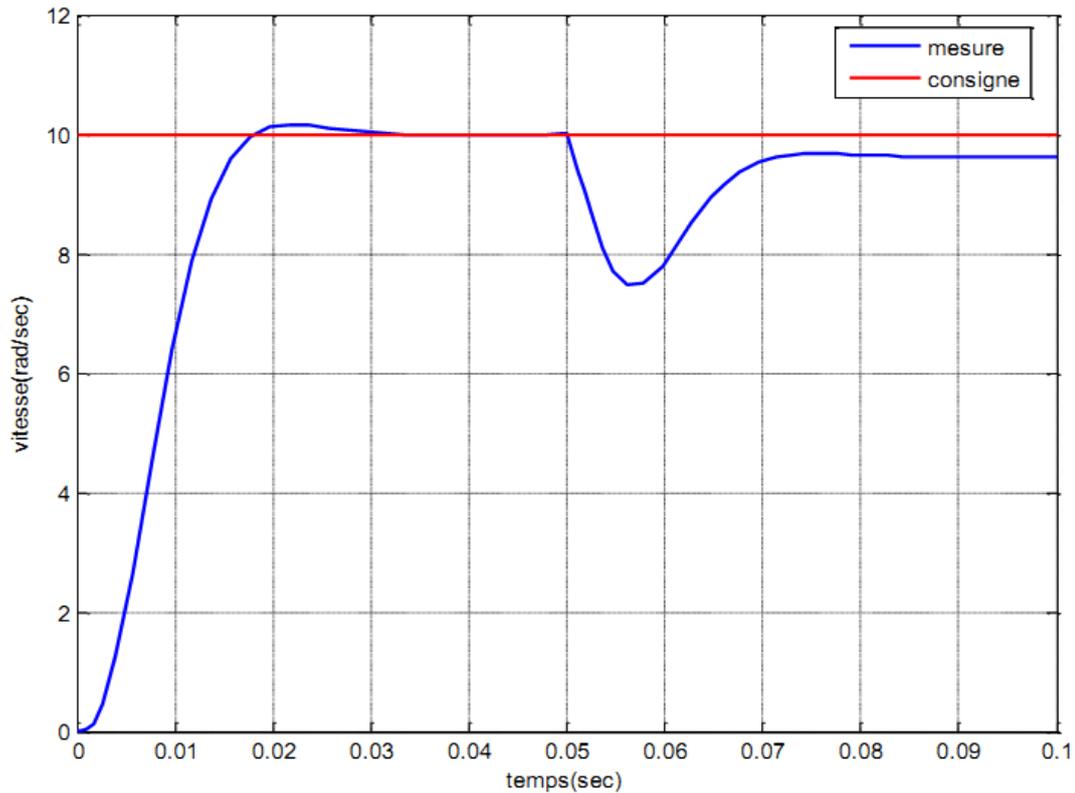
$$L = \begin{bmatrix} -3819.8 \\ -2019.3 \end{bmatrix} \quad (\text{II.44})$$

La structure de cette boucle de régulation est sur la figure II.23 suivante :



**Figure II.24** : retour d'état avec observateur

Les résultats suite à une simulation avec une consigne de 10rad/sec et un couple résistant de 0.1N.m à l'instant t=0.05s sont montrées sur la figure II.25 :



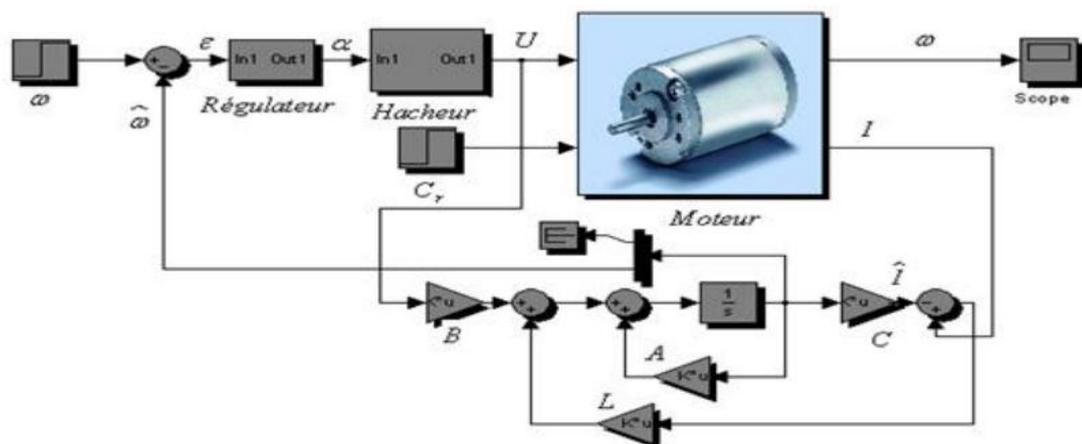
**Figure II.25** : réponse avec retour d'état observé

On remarque bien qu'après l'application d'un couple résistant la mesure suit la consigne mais avec une erreur de 0.3rad/sec qui est une erreur tolérable.

## II.11. Régulation PI avec estimation de la mesure

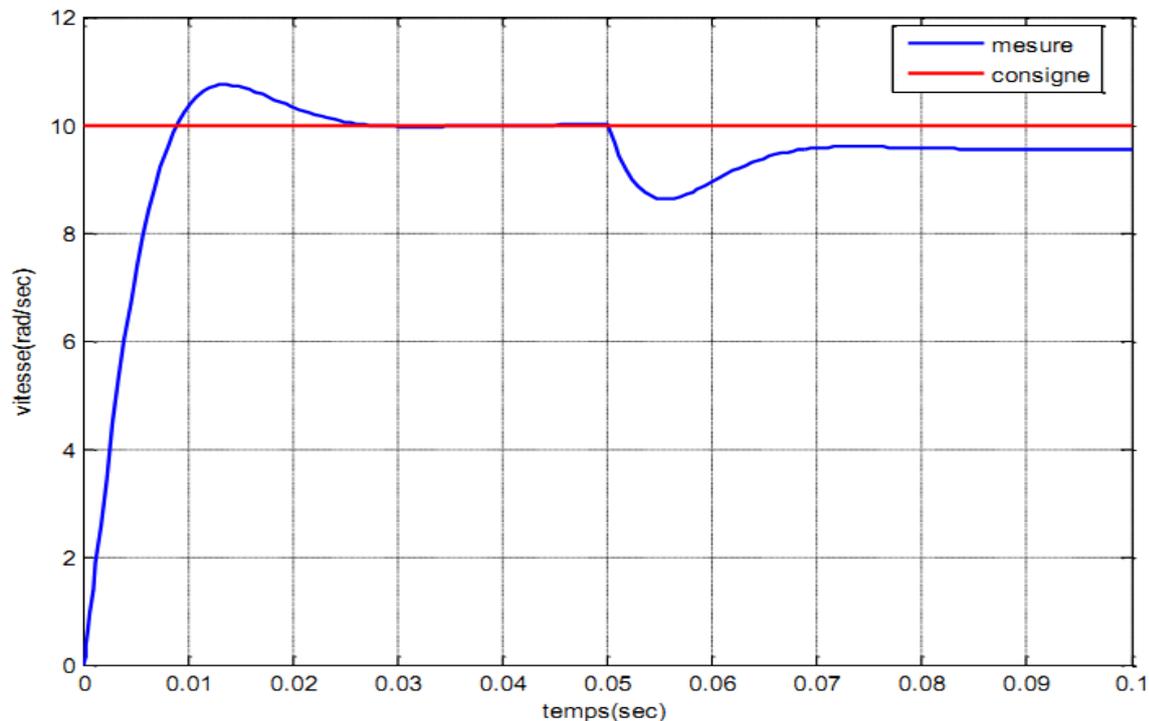
Passons maintenant à quelque chose de différent, c'est la régulation classique avec l'exploitation de l'observateur pour estimer la vitesse au lieu de la mesure avec un capteur.

Le principe est montré sur la figure II.26 suivant :



**Figure II.26** : schéma du correcteur PI avec estimateur de vitesse

La figure II.27 montre le résultat obtenu l'or d'une simulation avec une consigne de 10rad/sec et une charge de 0.8N.m à  $t=0.01s$ .



**Figure II.27 :** réponse du PI avec vitesse estimé

Le résultat obtenu est identique avec celui du retour d'état observé.

é.

## II.12. Commande par logique Floue

Contrairement aux techniques de réglage classique, le réglage par logique floue n'utilise pas des formules ou des relations mathématiques bien déterminées ou précises. Mais, il manipule des inférences avec plusieurs règles floues à base des opérateurs flous ET, OU, ALORS,... etc, appliquées à des variables linguistiques.

### II.12.1. Correcteurs flous de type PI

La première étape consiste à définir la base du correcteur, c'est à dire le moteur flou. La sortie de celui-ci, notée SPD, peut être définie à partir des fonctions  $k_p$  et  $k_d$ , qui sont respectivement les gains de l'action proportionnelle et dérivée du moteur[4].

$$S_{pd} = k_p(e, de)e + k_d(e, de)de \quad (\text{II.45})$$

Cette équation peut être réécrite en définissant des fonctions  $k_1$  et  $k_2$  de l'erreur  $e$  et la variation de l'erreur  $de$ .

$$S_{pd} = G\left(\frac{1}{G} + e(t)k_1(e, de) + \frac{1}{G_2} \frac{de(t)}{dt} k_2(e, de)\right) \quad (\text{II.46})$$

Pour réaliser un correcteur de type PI, il suffit d'intégrer la sortie du moteur flou. En notant  $S_{pi}$  la sortie du contrôleur flou de type PI, il vient

$$S_{pi} = \int S_{pd} dt \quad (\text{II.47})$$

### II.12.2. Principe et structure de la commande

Le contrôleur flou utilisé pour régler la vitesse du moteur est de type mamdani il reçoit comme entrée l'erreur ( $e$ ) et la variation de l'erreur ( $de$ ), le schéma bloc de la structure de commande est illustré par la figure II.28:

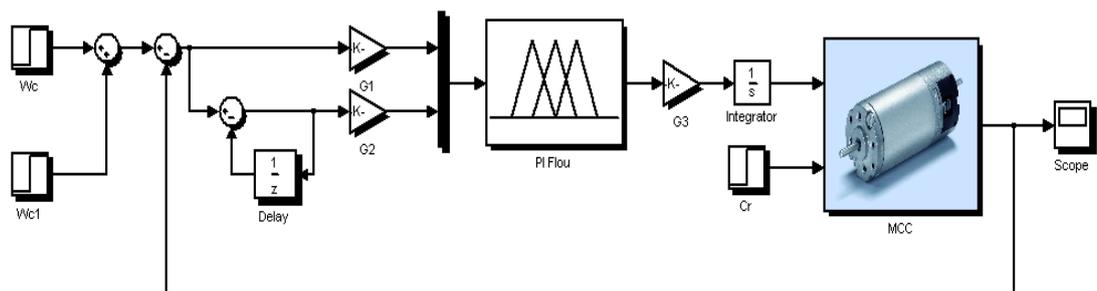


Figure II.28 : schéma bloc de la structure de commande

On définit les fonctions d'appartenance pour les variables d'entrées et de sortie comme suit :

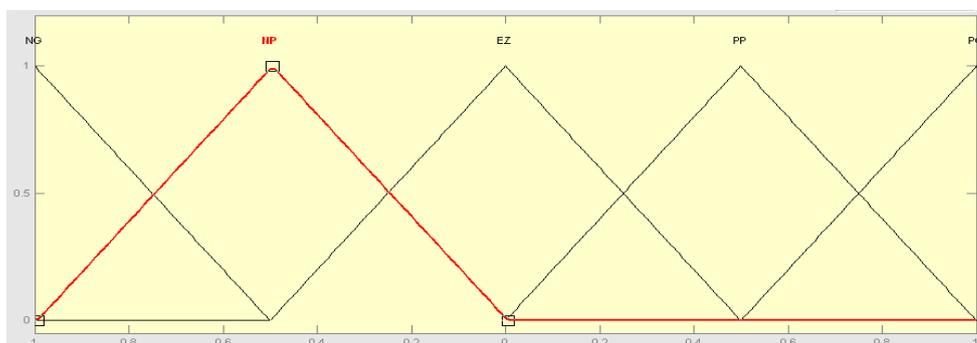
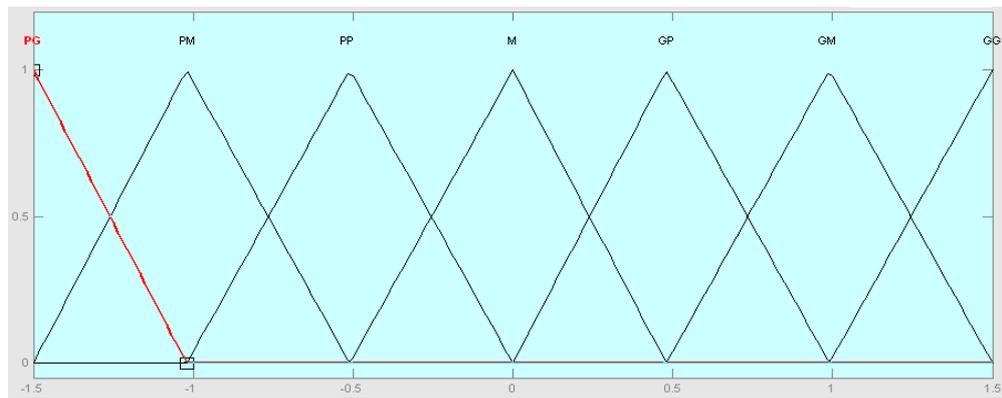


Figure II.29 : Fonctions d'appartenance des variables d'entrées



**Figure II.30 :** Fonctions d'appartenance de la variable de sortie

On résumera alors les tables d'inférence suivantes pour le PI flou :

$\Delta e \backslash e$	NG	NP	EZ	PP	PG
NG	<b>PG</b>	<b>PG</b>	<b>PM</b>	<b>PP</b>	<b>M</b>
NP	<b>PG</b>	<b>PM</b>	<b>PP</b>	<b>M</b>	<b>GP</b>
EZ	<b>PM</b>	<b>PP</b>	<b>M</b>	<b>GP</b>	<b>GM</b>
PP	<b>PP</b>	<b>M</b>	<b>GP</b>	<b>GM</b>	<b>GG</b>
PG	<b>M</b>	<b>GP</b>	<b>GM</b>	<b>GG</b>	<b>GG</b>

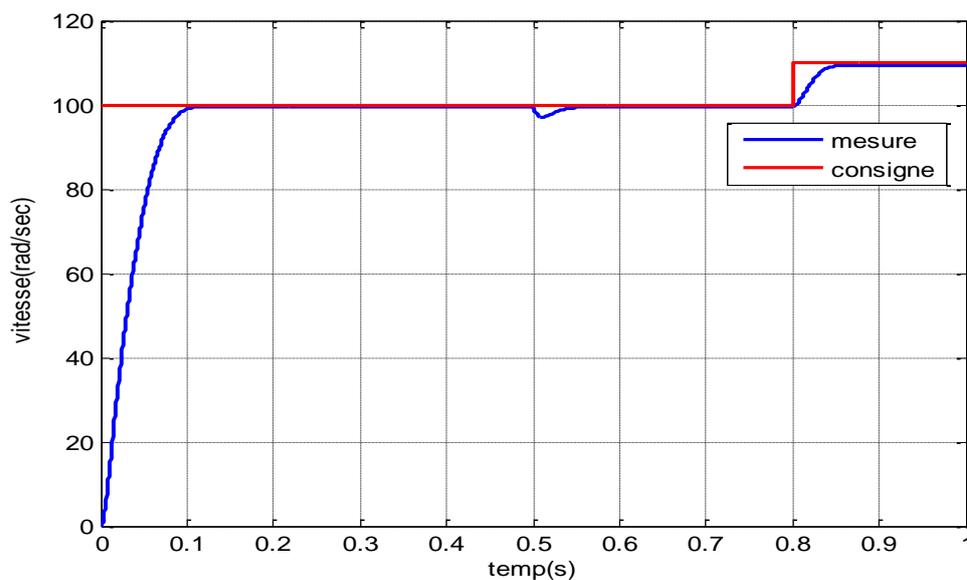
**Tableau I.2 :** les tables d'inférence pour le PI flou

L'action ou la commande floue de la sortie peut être exprimée ainsi :

$$\delta d = \{si e \text{ est } (NG) \text{ et } \Delta e \text{ est } (NG) \text{ alors } \delta d \text{ est } (NG) \text{ ou ; si } \dots \dots \dots \text{ ou ;} \} \quad (II.48)$$

Pour la déffuzification, on utilise la méthode de centre de gravité.

On passera maintenant à la simulation. :



**Figure II.31 :** réponse avec régulateur flou

La figure II.31 représentent la réponse du système avec régulateur flou ,la réponse est stable et atteint rapidement son point de consigne.

Après l'application d'une perturbation, la réponse du moteur montre que le système revient à suivre la consigne, avec une déformation sur la courbe de vitesse.

Le changement de consigne est illustré aussi sur la figure II.31 ,la régulation est bonne vue que la réponse arrive bien à atteindre son point de consigne.

### **II.13. Conclusion**

D'après les résultats obtenues par les différents régulateurs appliquer (le régulateur PI classique puis, par retour d'état PI, puis par la logique floue), on peut déduire que les résultats obtenus avec la logique floue sont très supérieurs à ceux obtenus par les régulateurs classique.

---

# Chapitre III :

## conception de la

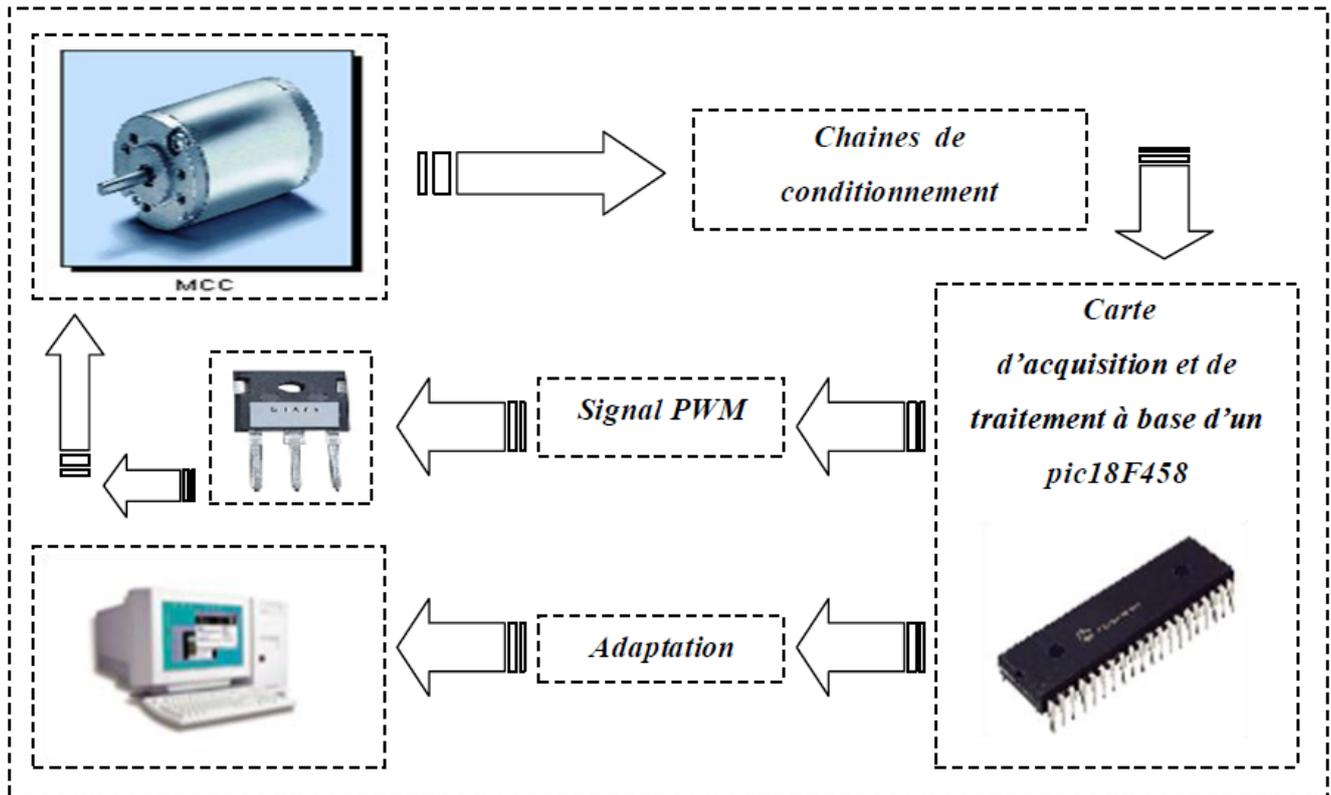
## commande

---

## Chapitre III : Conception de la commande

### III.1. Introduction :

Afin de pouvoir commander le moteur, avec une loi de commande numérique, il est nécessaire d'utiliser une carte d'interfaçage qui a pour but de transformer les signaux analogiques à des signaux numériques équivalents, ainsi que la nécessité d'un conditionnement pour faire l'adaptation des signaux nécessaire au traitement.



**Figure III.1 :** Schéma synoptique de la réalisation

Dans les paragraphes qui suivent on va expliquer chaque carte avec ces composants constitutifs.

### III.2. Carte de commande :

Cette carte contient :

- Un microcontrôleur de chez microchip le pic18F458.
- Adaptateur du niveau TTL 12V le MAX232.

Nous décrivons si dessous les caractéristiques de ces composant un par un

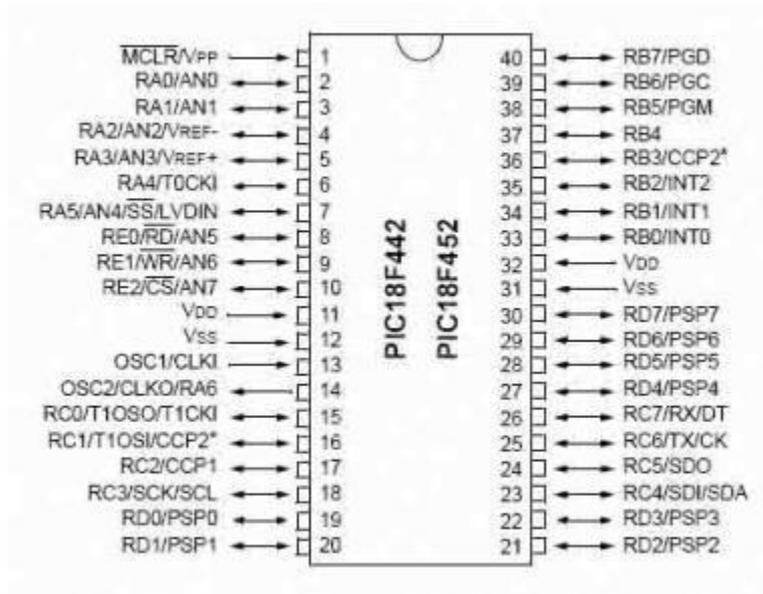
**III.2.1. Microcontrôleur PIC 18F458 :**

Un PIC est un microcontrôleur, c'est à dire une unité de traitement de l'information de type microprocesseur à laquelle on a ajouté des périphériques internes permettant de réaliser des montages sans nécessiter l'ajout de composants externes. Les PICs sont des composants dits RISC (Reduced Instructions Set Computer), ou encore (Composant à jeu d'instruction réduit).

Le PIC 18F458 est caractérisé par [10]:

- Une fréquence de fonctionnement élevée, jusqu'à 40 MHz
- Une mémoire vive de 1536 octets.
- Une mémoire EEPROM pour sauvegarder des paramètres de 256 octets
- Une mémoire morte de type FLASH de 32 Kmots, elle est réinscriptible à volonté
- Chien de garde WDT
- 33 Entrées et sorties
- Chaque sortie peut sortir un courant maximum de 25 mA
- 4 Temporisateurs : TIMER0 (16 bits avec pré diviseur),
- TIMER1 (16 bits avec pré-diviseur avec possibilité d'utiliser une horloge externe réseau RC ou QUARTZ) et TIMER2 (8 bits avec pré-diviseur et post-diviseur).
- 1 entrée de captures et de comparaison avec PWM (Modulation de largeur d'impulsions).
- Convertisseur analogique numérique 10 bits avec 8 entrées multiplexées maximum.
- Une interface de communication série asynchrone et synchrone (USART/SCI)
- Une interface de communication série synchrone (SSP/SPI et I2 C)
- Une tension d'alimentation entre 2 et 5.5 V.

### III.2.1.1. Organisation externe du PIC 18F458 :



**Figure III.2 :** Brochage du PIC18F458.

Le boîtier du PIC 18F458 décrit par la figure III.2 comprend 40 pins : 33 pins d'entrées/sorties, 4 pins pour l'alimentation, 2 pins pour l'oscillateur et une pin pour le reset (MCLR).

La broche MCLR sert à initialiser le  $\mu$ C en cas de la mise sous tension, de remise à zéro externe, de chien de garde et en cas de la baisse de tension d'alimentation.

Les broches VDD et VSS servent à alimenter le PIC.

Les broches OSC1 et OSC2 ou CLKIN et COUT permettent de faire fonctionner

L'oscillateur interne du PIC qui peut être un quartz, un résonateur céramique, un oscillateur externe ou un réseau RC. Dont le rôle est de créer des impulsions de fréquences élevées.

Lors de la programmation, la broche MCLR doit être portée à un niveau compris entre

12 V et 14 V et le PIC18F458 commence à programmer en appliquant un signal d'horloge sur la broche RB6 (broche 39) et les informations binaires transitent en série.

Sur la broche RB7 (broche 40). Chacune des informations qui transitent sur la broche RB7 est validée à la retombée du signal d'horloge sur la broche RB6.

### III.2.1.2. Les ports d'entrée /sortie :

Les  $\mu\text{C}$  18F458 contient les 5 ports suivants :

- **Le port A :**

Le port A est formé de six pins donc six entrées/sorties numérotées de RA0 à RA5 qui peuvent être utilisées comme des entrées pour le convertisseur analogique numérique ou utilisées pour le TIMER 0, dans ce dernier cas le pin RA4 sera utilisé comme entrée pour configurer TOCKI est de type drain ouvert. On peut utiliser ce port, soit pour la conversion analogique /numérique, soit en mode (I/O).

- **Le port B :**

Rien de particulier à dire sur ce port, qui possède 8 pins d'entrées/sorties classiques numérotées de RB0 à RB7. On note que le pin RB0 qui, en configuration d'entrée, est de type « trigger de Schmitt » quand elle est utilisée en mode interruption « INT ». La lecture simple de RB0 se fait, de façon tout à fait classique, en entrée de type TTL.

- **Le port C :**

Tout d'abord au niveau programmation, c'est un PORT tout ce qu'il y a de plus classique, comportant 8 pins de RC0 à RC7.

- **Le port D :**

Ce port fonctionne de façon identique aux autres, dans son mode de fonctionnement générale les 8 pins I/O, en mode entrée, sont de type « trigger de Schmitt » Ce port est très utilisé en mode parallèle esclave (slave).

- **Le port E :**

Le port E possède trois pins donc trois entrées/sorties, RE0 à RE2, il est utilisé comme entrée au convertisseur analogique numérique et aussi il peut contrôler le port parallèle slave c'est-à-dire le port D.

### III.2.1.3. Le convertisseur analogique numérique du PIC 18F458 :

La fonction conversion analogique-numérique consiste à transformer une grandeur électrique en une grandeur numérique exprimée sur N bits. Cette grandeur de sortie représente, dans le système de codage qui lui est affecté, un nombre proportionnel à la grandeur analogique d'entrée.

Le CAN intégré dans le 18F458 est un CAN 10 bits qui donne une précision en 5V de 5mV environ, ce qui est une précision tout à fait intéressante. La résolution 10 bits du PIC permet d'attribuer 1024 valeurs numériques à notre signal d'entrée.

D'une manière générale, le principe de conversion analogique/numérique nécessite

deux opérations :

#### A- La quantification :

Opération qui consiste à associer une valeur analogique à la plus petite variation mesurable entre deux valeurs codées distinctes en sortie.

Cette valeur est appelée quantum.

$$q = \frac{\Delta V_{e \max}}{2^n} = \frac{\Delta V_{ref+} - V_{ref-}}{2^n} \quad (\text{III.1})$$

$q$ : quantum (V), aussi appelé résolution.

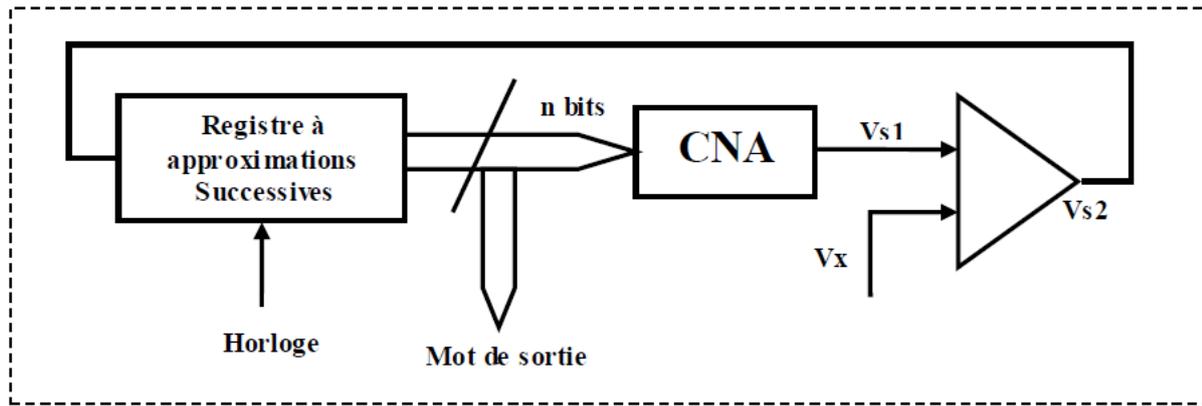
$\Delta V_e$ : C'est l'écart entre la valeur min et  $\max$  de  $V_e$  à numériser (V).

$n$ : Nombre de bits en sortie du convertisseur.

#### B- Le codage :

Opération qui assigne une valeur numérique à chacun de ces niveaux.

Le principe de conversion utilisé sur le PIC18F458 est celui de l'approximation successive dont le fonctionnement est détaillé ci-dessous :



**Figure III.3 :** Principe de conversion d'un ADC de PIC 18F458

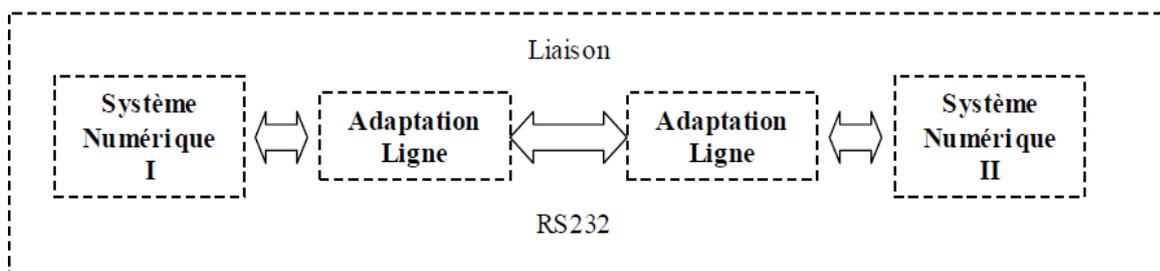
L'algorithme utilise un principe de dichotomie. Ceci signifie que l'intervalle de comparaison est divisé par deux à l'issue de chaque comparaison. Le registre à approximation successive génère une succession de mots numériques sur  $n$  bits. Il est commandé par les résultats de la comparaison entre  $V_x$  (tension à convertir) et la grandeur de sortie du CNA [9].

### III.2.2. Communication série asynchrone RS232 :

Les liaisons séries permettent la communication entre deux systèmes numériques en limitant le nombre de fils de transmission.

La liaison série aux normes de RS232 est utilisée dans tous les domaines de l'informatique. Elle est de type asynchrone, c'est-à-dire qu'elle ne transmet pas le signal de l'horloge [11].

Le schéma fonctionnel est le suivant :



**Figure III.4 :** Schéma fonctionnel d'une liaison série asynchrone de la norme RS232

La transmission série nécessite au moins 2 fils de communication, l'un pour la transmission (**Tx**) et l'autre pour la réception (**Rx**) et un fil de masse.

Passons maintenant au principe d'adaptation entre le PIC et le port série RS232.

Le PIC utilise les niveaux 0V et 5V pour définir respectivement des signaux « 0 »

et « 1 ». La norme RS232 définit des niveaux de +12V et -12V pour établir ces mêmes niveaux.

Nous aurons donc besoin d'un circuit charge de convertir les niveaux des signaux entre PIC et PC. La pin TX du PIC émettra en 0V/5V et sera convertie en +12V/-12V vers notre PC. La ligne RX du PIC recevra les signaux en provenance du PC, signaux qui seront converti du +12V/-12V en 0V/5V.

Notons que la liaison étant full-duplex, émission et réception sont croisées, chaque fil ne transitant l'information que dans un seul sens.

Nous utiliserons le célèbre circuit **MAX232** pour effectuer cette adaptation de niveaux.

Ce circuit contient un double convertisseur à double direction. Autrement dit, il dispose de :

- 2 blocs, dénommés T1 et T2, qui convertissent les niveaux entres en 0V/5V en signaux sortis sous +12V/-12V.
- 2 blocs, dénommés R1 et R2, qui convertissent les niveaux entres en +12V/-12V en signaux sortis sous 0V/5V.

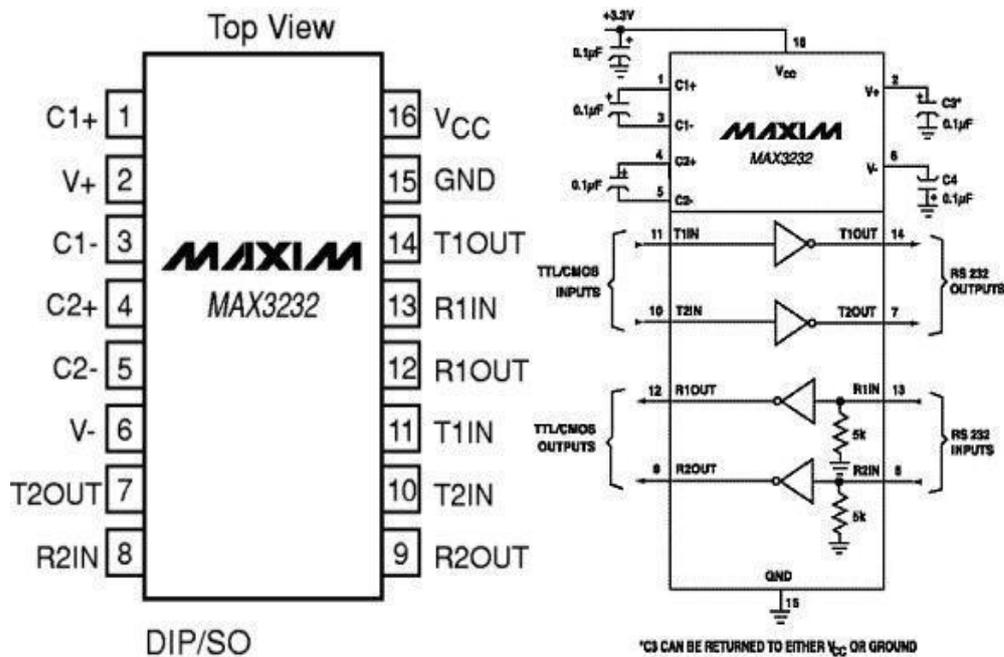


Figure III.5 : Circuit du MAX232

Nous avons utiliser cette communication pour le chargement du programme ainsi que la visualisation des grandeurs d'entrés ou de sorties sur l'HyperTerminal.

Durant tous notre travail, il nous est arrivé de reprogrammer notre pic plusieurs fois, pour corriger les erreurs et améliorer le fonctionnement.

Alors pour éviter de couper l'alimentation de la carte, enlever le pic, le placer dans le programmeur, le programmer, le remettre encor une fois sur la carte et relancer l'alimentation, on a utilisé la technique du bootloader.

Cette technique permet la programmation du pic sur circuit ICSP (In Circuit Serial Programming) à travers le port série du pc (voir annexe).

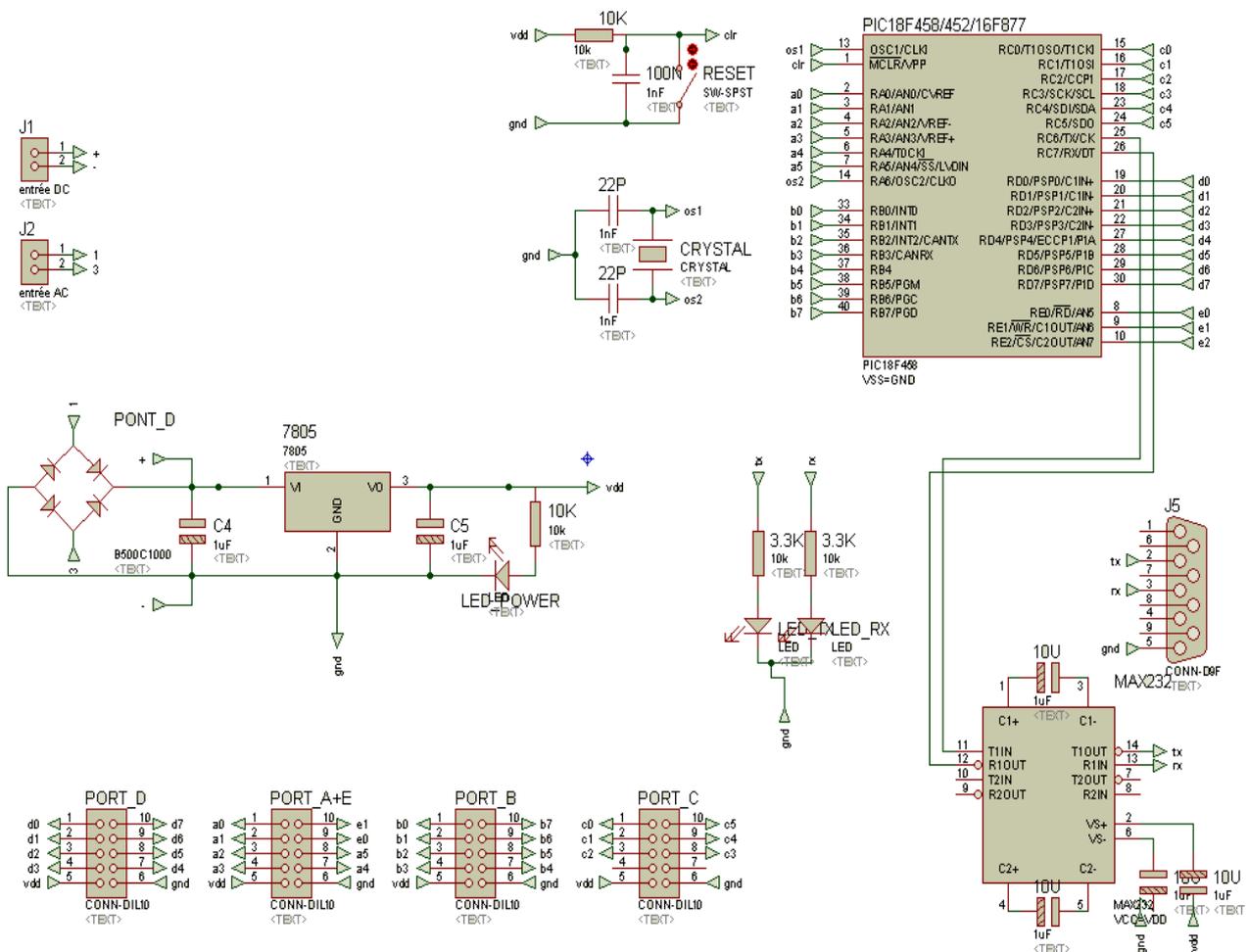


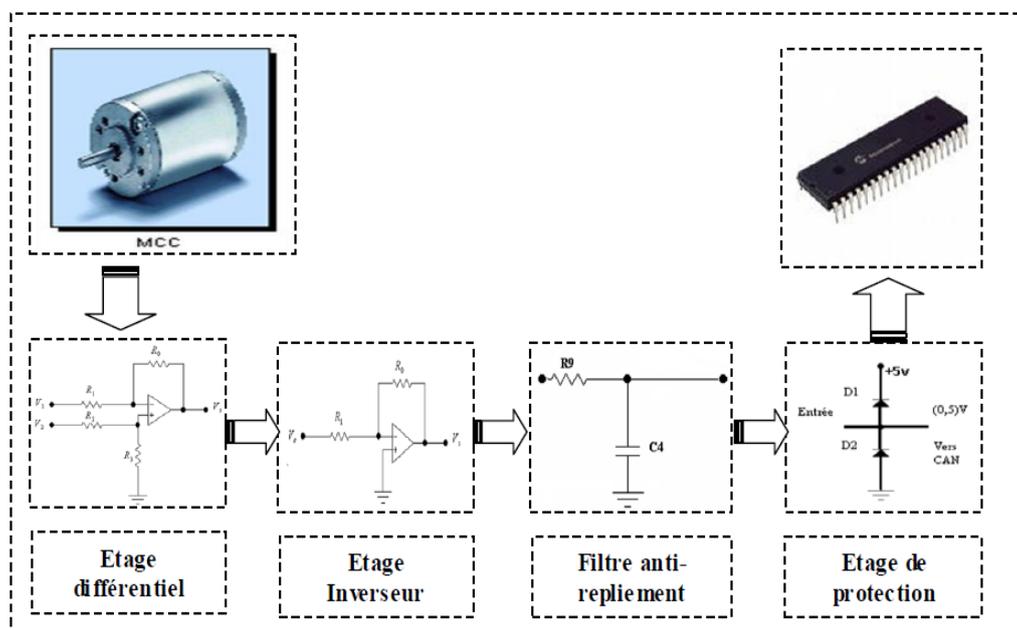
Figure III.6 : Photo de la carte de commande

### III.3. Carte de conditionnement

Pour pouvoir exploiter notre carte de commande, cette dernière doit être reliée avec une autre carte permettant sa liaison avec le processus considéré.

Le convertisseur analogique /numérique interne au 18F458 n'accepte que des entrées analogiques évoluant dans l'intervalle [0V +5V], il faut alors adapter les signaux d'entrées en les faisant passer par des chaînes de conditionnement.

Dans notre application nous avons eu besoin de trois grandeurs pour faire le traitement nécessaire à la génération de notre loi de commande, les 3 chaînes de conditionnement étant basé sur le même principe.

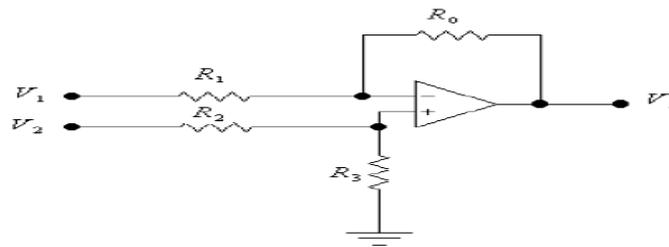


**Figure III.7** : Schéma synoptique du conditionnement

#### III.3.1. Etage différentiel

Lors de la transmission du signal issu du capteur, la présence d'un courant de masse peut entraîner une tension qui va se superposer à la tension due au capteur (signal de mesure) et sera donc, amplifiée de la même manière que le signal de mesure sans possibilité de l'éliminer.

Un amplificateur différentiel jouant, aussi le rôle de diviseur de tension est inclus dans la chaîne afin d'annuler l'effet de cette différence de potentiel, perturbatrice et pouvant rendre les mesures erronées [5].



**Figure III.8:**Montage différentiel

L'amplificateur différentiel est un étage de gain dont la fonction principale est d'amplifier une différence de tension ou de courant .il est utilisé pour éliminer l'effet des différents signaux parasites pouvant polluer le signal utile .les signaux d'entrées et de sorties sont liées par la relation :

$$VS = \frac{-R0}{R1} V1 + \frac{R3}{R2} V2 \cdot \frac{(1 + \frac{R0}{R1})}{(1 + \frac{R3}{R2})} \quad (\text{III.2})$$

En général, les valeurs des résistances  $R_0$ ,  $R_1$ ,  $R_2$ ,  $R_3$  sont choisies telles que :

$$\frac{R0}{R1} = \frac{R3}{R2} \quad (\text{III.3})$$

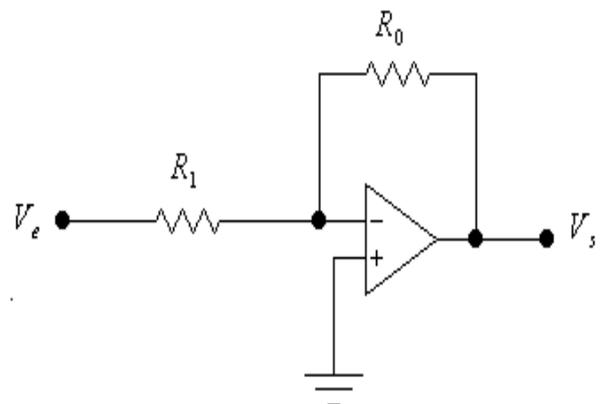
alors on obtient :

$$VS = \frac{-R0}{R1} (V2 - V1) \quad (\text{III.4})$$

### III.3.2. Etage inverseur

La tension de sortie de l'étage différentiel est comprise entre 0V et -5V ce qui implique le passage par un étage inverseur.

Equation :  $VS = \frac{-R0}{R1} Ve$  (III.5)



**Figure III.9 :** Montage inverseur

### III.3.3. Le filtrage

Les signaux à acquérir étant délivrés avec des oscillations gênantes pour le calcul, l'acquisition étant ponctuelle, elle risque alors de s'effectuer sur une valeur due à un état transitoire du processus, comme l'opération du filtrage peut assurer l'élimination des oscillations nous l'avons effectué avec un filtre anti-repliement.

Le filtre anti-repliement est un filtre passe-bas que l'on trouve dans les dispositifs de numérisation, qui permet de nettoyer le signal avant d'effectuer une conversion analogique numérique.

Son but est d'interdire toute présence, avant échantillonnage de fréquence supérieure à  $F_e/2$

$$F_e = 1/T_e = 1/0.001 = 1\text{Khz} \quad (\text{III.6})$$

La fréquence de coupure :

$$f_c = \frac{F_e}{2} = \frac{1}{2\pi RC} = 500\text{Hz} \quad (\text{III.7})$$

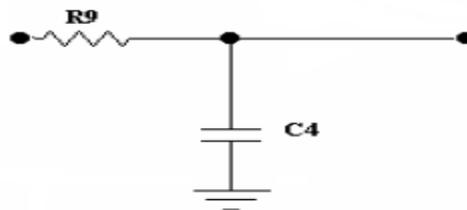


Figure III.10 : Filtre passe-bas

### III.3.4. Etage de protection

Pour la protection du PIC nous avons utilisé un circuit d'écrêtage à deux diodes (Bat49). Si une tension supérieure à +5V est présente à l'entrée de circuit elle se trouve écrêtée par la diode D1, de même, si une tension inférieure à 0V est présente à l'entrée elle sera écrêtée par la diode D2.

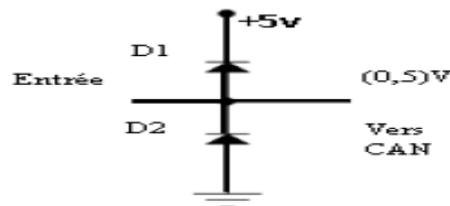


Figure III.11 : Montage de protection

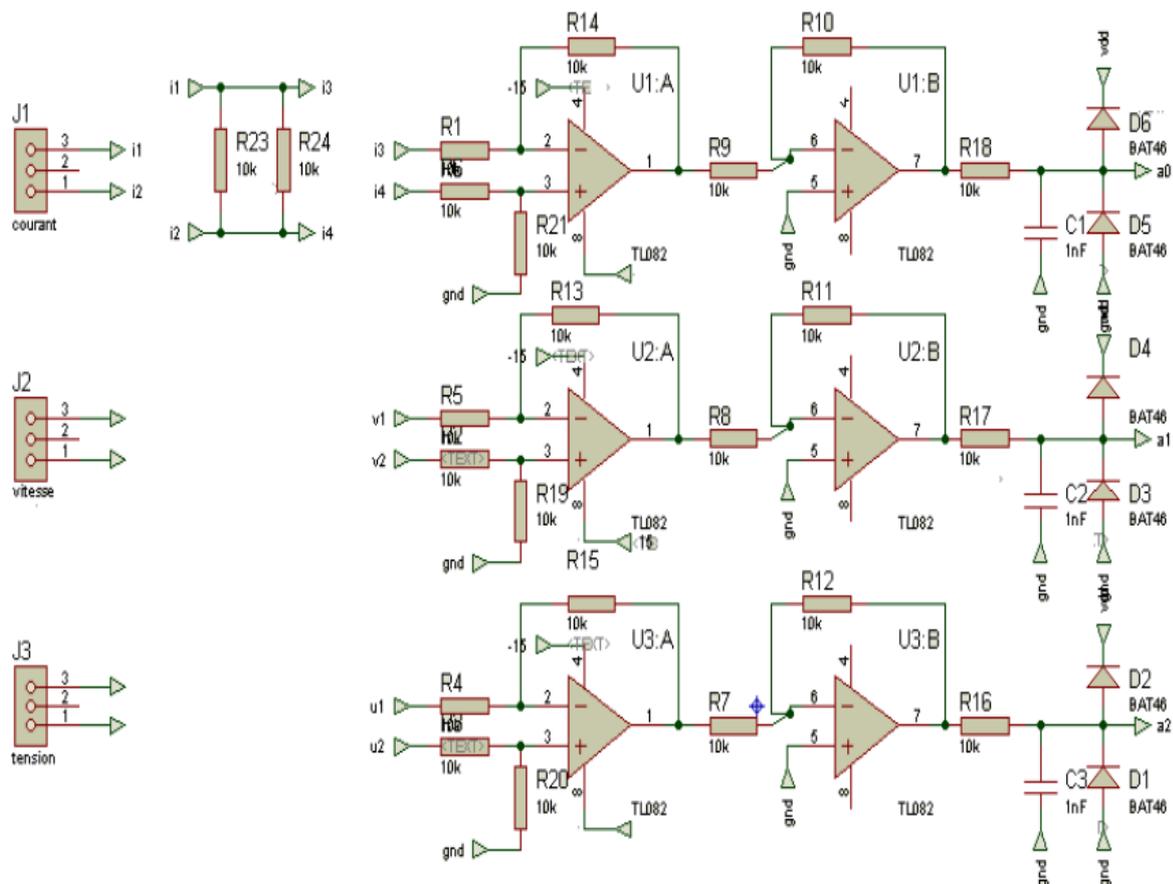


Figure III.12 : Carte de conditionnement

### III.3.5. Carte de sortie

Après l'acquisition des différents mesures, notre carte doit délivrer une commande numérique suivant l'algorithme de commande qu'elle exécute, cette commande est mis en forme

suivant un signal PWM généré par le PIC afin qu'elle puisse attaquer le MCC à travers un hacheur série.

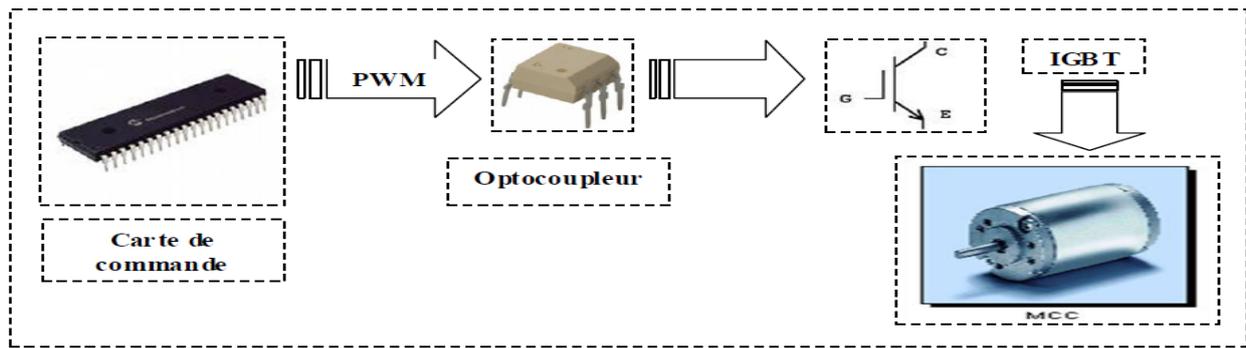


Figure III.13 : Schéma synoptique de commande

### III.3.6. Le signal PWM

Généralement, tous les microcontrôleurs récents savent générer des signaux PWM ce qui signifie Pulse Width Modulation (**ou** modulation de largeur d'impulsion). Ces

signaux permettent en effet de doser avec une grande efficacité la puissance appliquée à des charges alimentées sous une tension continue.

Grâce au timer 2 du PIC et à une possibilité de configuration interne particulière faisant appel au module de capture CCPx, il est très facile de générer des signaux PWM avec le rapport cyclique de notre choix.

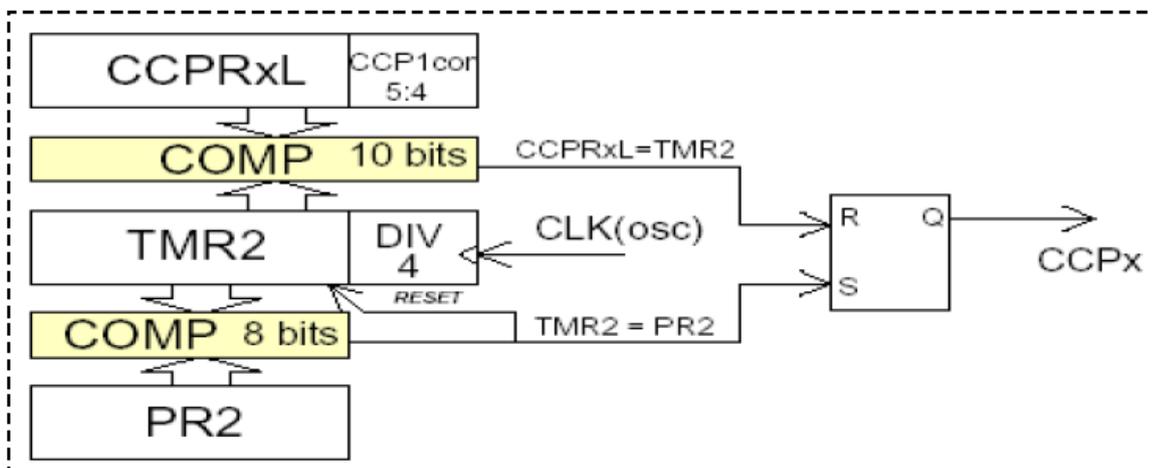


Figure III.13 : Principe du signal PWM

Le principe de fonctionnement est le suivant. Le timer 2 permet de définir, grâce à son registre PR2 et à son pré-diviseur d'horloge, la fréquence de génération des signaux PWM qui sont prélevés au niveau de sa sortie horloge. Cependant, pendant la phase de comptage, le registre de ce timer est comparé au contenu du registre de comparaison CCP1 et, s'il y a égalité, la sortie PWM est remise prématurément à zéro alors que le comptage du timer 2 continue normalement, nous passons maintenant à la partie suivante où on développera l'implémentation des programmes en langage C.

## III.4. Programmation du Microcontrôleur PIC

### III.4.1. Introduction

L'utilisation d'un calculateur numérique comme structure de commande d'un système dynamique continu implique la discrétisation du correcteur à implémenté selon une approximation choisi pour trouver une loi de commande discrète dans le temps.

Dans cette chapitre on va faire la discrétisation du correcteur PI classique avec une implémentation sur pic.

### III.4.2. Discrétisation du correcteur PI

Le principe de cette discrétisation consiste à trouver un correcteur  $C(z)$  à partir du correcteur  $C(s)$  en choisissant une approximation pour la variable  $p$ .

Soit  $T$  la période d'échantillonnage, les approximations les plus utilisé sont :

- Approximation avancé :

$$p \rightarrow \frac{z - 1}{T} \quad (\text{III.8})$$

- Approximation retardé :

$$p \rightarrow \frac{z - 1}{z \cdot T} \quad (\text{III.9})$$

- Approximation de Tustin (ou bilinéaire) :

$$p \rightarrow \frac{2}{T} \cdot \frac{z - 1}{z + 1} \quad (\text{III.10})$$

Prenons le choix de cette dernière approximation, la fonction de transfert du correcteur PI est

$$C(s) = \frac{k_p s + k_i}{s} \quad (\text{III.11})$$

Sa structure en discret devienne :

$$C(z) = \frac{\alpha(z)}{\varepsilon(z)} = \frac{k_p \left( \frac{2}{T} \cdot \frac{z-1}{z+1} \right) + k_i}{\left( \frac{2}{T} \cdot \frac{z-1}{z+1} \right)} \quad (\text{III.12})$$

Après calcul et simplification on obtient :

$$\frac{\alpha(z)}{\varepsilon(z)} = \frac{(k_i T - 2k_p)z^{-1} + (2k_p + k_i T)}{2 \cdot (1 - z^{-1})} \quad (\text{III.13})$$

La loi de commande générale du correcteur est donnée comme suite :

$$\alpha(k) = \alpha(k - 1) + A \cdot \varepsilon(k - 1) + B \cdot \varepsilon(k) \quad (\text{III.14})$$

Avec :

$$\begin{cases} A = 0.5k_i T - k_p \\ B = k_p + 0.5k_i T \end{cases} \quad (\text{III.15})$$

### III.4.3. Période d'échantillonnage

La période d'échantillonnage **T** doit être choisie en fonction de la dynamique du système. Si le processus est très lent, il est inutile de choisir des périodes d'échantillonnages très petites, par contre si l'évolution du processus est très rapide, il est nécessaire d'échantillonner à des cadences élevées pour pouvoir suivre l'évolution des processus avec le temps tout en respectant le théorème de **SHANON** qui s'énonce ainsi [6]:

«Toute fonction du temps **f(t)** possédant un spectre de fréquence limitée à  $\pm 1/T_c$  peut être transformée par échantillonnage périodique, de période inférieur ou égale à **Tc/2** sans aucune perte d'information »

En pratique, la période d'échantillonnage **T** est choisie telle que : **Tc/5 > T > Tc/10**

Et si l'on désigne par **Fc** la fréquence de coupure du processus à commander, et par **F** la fréquence d'échantillonnage, on a : **5 · Fc < F < 10 · Fc**

Pour la régulation des moteurs, la période d'échantillonnage conseillée est comprise entre 0.001 sec à 0.1 sec, cela est donné comme une règle empirique.

Dans notre application le choix de cette période était de prendre **T=1ms**, ce qui est un choix judicieux car la constante de temps du moteur est de l'ordre de 10ms.

### III.4.4. Implémentation du PI

Le schéma suivant montre la structure de la boucle de régulation :

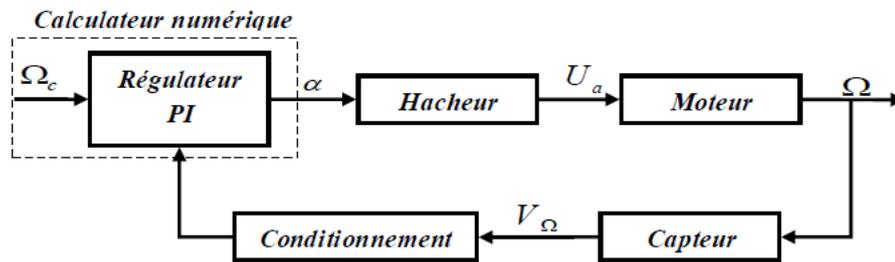


Figure III.14 : Boucle de régulation réel

L'algorithme de régulation se compose de deux parties, une partie contenant le programme principal et l'autre partie contenant la routine d'interruption utilisée pour le calcul et l'élaboration de la loi de commande. L'interruption utilisée est celle du timer, le programme de traitement d'interruption est exécuté à chaque temps d'échantillonnage, l'organigramme suivant montre l'algorithme de régulation :

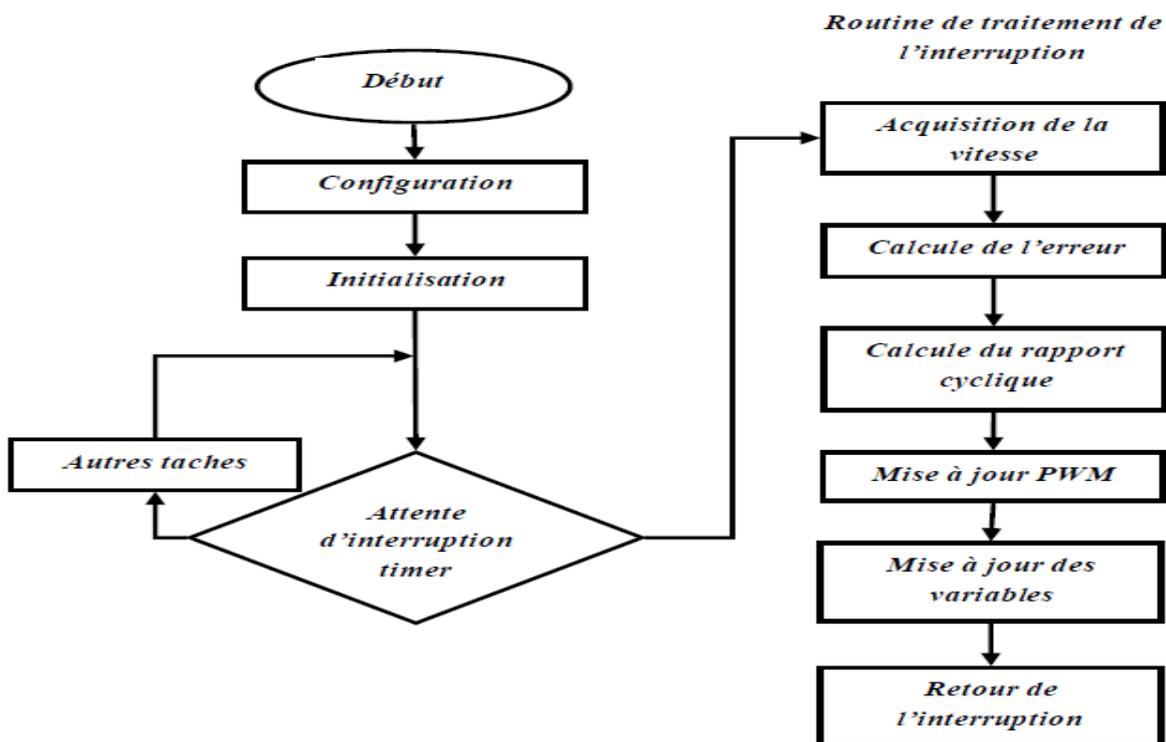


Figure III.15 : organigramme de la régulation

### III.5. La routine de traitement PI

```

#int_timer0
void timer0_isr(){
// initialisation du timer0 pour le compte
  set_timer0(63036);
// acquisition de la vitesse réelle
  set_adc_channel(0);
  delay_us(10);
  w_mes=read_adc()*0.0048;
// filtrage numérique
  w_mesf=(w_mes+w1+w2+w3+w4+w5)/6.0;
// acquisition de la consigne
  set_adc_channel(1);
  delay_us(10);
  w_cons=read_adc()*0.002248289;
// calcule de l'erreur
  w_err=w_cons-w_mesf;
// calcule de la loi de commande
  u_com=u_com_2+b0*w_err+b1*w_err_1+b2*w_err_2;
// mise à jour du signal PWM
  dt=(unsigned int8)(u_com*255);
  set_pwm1_duty(dt);
// affichage sur l'hyper terminal
  printf(" %f  %f \r\n",w_cons,w_mesf);
// mise à jour des variables
  w_err_2=w_err_1;
  w_err_1=w_err;
  u_com_2=u_com_1;
  u_com_1=u_com;
  w5=w4;
  w4=w3;
  w3=w2;
  w2=w1;
  w1=w_mes;
}

```

### III.6. Conclusion

Dans le cadre de notre travail on va utiliser ce signal pour la commande d'un hacheur ou plus précisément un composant de l'électronique de puissance IGBT

Dans ce chapitre nous avons donné une idée générale sur les différentes cartes a etudie dans notre projet, carte de commande, carte de conditionnement et la description du hacheur a utilisé.

---

# CONCLUSION GENERALE

---

# Conclusion Générale

L'objectif du projet qu'on a élaboré était la commande en vitesse d'un moteur à courant continu .

On a débuté ce travail par la modélisation du système à contrôler (hacheur, moteur, capteur), le modèle obtenu a servi à l'étude de la dynamique du système en boucle ouverte .

On a synthétiser plusieurs types des commandes basés sur un cahier de charge bien définie (le régulateur PI classique , retour d'état , retour d'état avec intégrale, retour d'état observé, PI avec estimation de la mesure , puis par la logique floue).

D'après les résultats obtenus par les différents régulateurs, on peut déduire que les résultats obtenus avec la logique floue sont très supérieurs à ceux obtenus par les régulateurs classiques.

On a aussi la conception sur le logiciel les cartes de commande et de conditionnement avant de réaliser les cartes.

Le dernier chapitre on a fait la discrétisation du correcteur PI classique avec la routine PI pour l'implémentation sur le pic.

Les résultats obtenus ont a montré que la régulation sans capteur présente les même performances que celle avec capteur mais elle reste sensible aux bruit de mesure et à la précision de calcule.

On propose comme travail future de :

- implémentés expérimentalement les cartes de commande réalisée à base d'un microcontrôleur pic18f458 et d'une carte de conditionnement des signaux de mesures.
- Utiliser un observateur pour implémentés expérimentalement les commande basé sur l'observateur.

---

# **BIBLIGRAPHIE**

---

## BIBLIOGRAPHIE

- [1] **Katsuhiko Ogata, Univ of Minnesota** « modern control engineering \_ 3rd Ed » prentice hall,liver,1997.
- [2] Aymen Chaaira « Représentations d'état linéaires des systèmes monovariabiles » ESIP, **Olivier BACHELIER** ,Oct 19- 2013
- [3] – **El-kebir BOUKAS**« systèmes asservis » école polytechnique de Montréal, 1999.
- [4] **Nadia Martaj·Mohand Mokhtari** « application de MATLAB 5 et SIMULINK 2 » Le premier chapitre ,Springer, 1998.
- [5] - **H.HAMMOUMI** « Etude et réalisation d'une carte prototype à base du microcontrôleur MC68HC11F1 » Département d'électrotechnique, 2004
- [6] – **D.Jaume\ S.Thelliez\ M.Vergé**«Commande des systèmes dynamiques par calculateur » [Texte imprimé] : BTS, IUT, Écoles d'ingénieurs EYROLLES, 1991.
- [7] – **S.MOKHTARI\ A.MORSLI** « Régulation de vitesse d'un moteur à courant continu par Microcontrôleur HCS12DP512 » Département d'Automatique, 2008.
- [8] – **U-Séries Servodisc** ,Kollmorgen ,Motion Technologies Group • Commack, New **York -2001**
- [9] - **P.Poulichet**,« les convertisseurs analogiques numériques » , livre T2-ACQDO-Chapitre -2005
- [10] – **Data sheet Microchip** ,« PIC18FXX8 28/40-Pin High Performance, Enhanced FLASH Microcontroller with CAN »,site,2004 Microchip Technology Inc.