الجمهورية الجزائرية الديمقراطية الشعبية

MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITE DR.YAHIA FARES DE MEDEA

FACULTE DES SCIENCES ET DE LA TECHNOLOGIE

Département du Génie Électrique et de l'Informatique

Mémoire de Fin d'Études

Présenté pour l'obtention du Diplôme de Master en Génie Électrique

Option : Commande et électronique industrielle.

THEME

Implémentation de l'algorithme PI dans un microcontrôleur 16F877A pour la commande d'un moteur à courant continu

Proposé et dirigé par,

Mr. Farouk BETTACHE

Dr. Abdelhafid MOUALDIA

Préparé et présenté par,

Mme. Dalal TLEMSANI

Mlle. Yasmine OULD BABA ALI

AnnéeUniversitaire2014-2015

Remerciements

Nous remercions dans un premier Temps, notre DIEU qui nous a donné la volonté et le courage afin d'arriver à ce modeste travail.

Nous remercions nos promoteurs : Mr. F. BETTACHE et Dr. A. MOUALDIA qui nous ont orientés et aidé dans toute la durée de notre travail par leurs critiques et leurs suggestions nombreuses.

Nous tenons à remercier tous les membres de jury pour avoir accepter de juger notre travail.

Nos sincères remerciements vont également à tous les enseignants du département de Génie électrique de l'Université de Médéa, et à tous les enseignants qui ont participés à notre formation.

Dédicace

Je dédie ce modeste travail à tous ceux qui témoignent qu'il n'y a de dieu qu'Allah et MOHAMED est son envoyé et son prophète.

- *À Mes très chères parents, pour leurs soutient moral et financier et d'être la lumière de ma vie, qui m'ont toujours encouragé pour terminer mes études, en leur espérant une longue vie et que Dieu les gardes et mon mari Cherif.*
- *À mes sœurs et mes frères.*
- *À ma deuxième famille ZEMIR et surtout ma belle sœur Amira.*
- *À tous ceux qui m'ont aidé de près ou de loin à la réalisation de mon mémoire.*
- *À mon binôme OULD BABA ALI Yasmine*
- *À mes chères amies de l'université de MEDEA*
- *À tous les enseignants de faculté des sciences et de la technologie.*

Dalal TLEMSANI

Dédicace

Je dédie ce modeste travail à tous ceux qui témoignent qu'il n'y a de dieu qu'Allah et MOHAMED est son envoyé et son prophète.

Yamine OULD BABA ALI

الملخص

الهدف من عملنا هذا هو ادماج مصحح (معدل)PI داخل معالج PIC16F877A للتحكم في سرعة محرك ذو تيار مستمر باستعمال لغة البرمجة MikroC لكتابة البرنامج الخاص لهذا التعديل و برنامج Proteus Isis للمحاكاة قبل المرور الى التطبيق .

يتضمن هذا العمل بعض المفاهيم الضرورية و الاساسية حول محرك ذو تيار مستمر ،بالاضافة الى لمحة عن المعالج PIC16F877A والذي صمم لاجل هذا النوع من التطبيقات .

الكلمات المفتاحية : المعالج **(pic16f877a)**، الادماج ، تصميم المحاكات ،المنظم **PI** ، محرك ذو تيار مستمر.

Résumé

L'objectif de notre travail consiste à implémenter l'algorithme PI dans un microcontrôleur PIC 16F877A pour la commande d'un moteur à courant continu en utilisant la programmation MikroC pour programmer notre PIC et le programme Proteus ISIS pour la simulation avant la fait en pratique.

Dans ce travail, quelques notions de base sur la machine à courant continu sont données.

Un aperçu sur le PIC 16F877A qui est dédié a cette application.

Mots clés : microcontrôleur **PIC : (Periphirol Interface contrôler)-** Modélisation-implémentation –Simulation - régulateur **PI -** moteur a courant continu**.**

Abstract

The aim of our work is to implement the PI algorithm in a 16F877A PIC microcontroller to control a DC motor using MikroC programming for programming our PIC and Proteus ISIS simulation program before done in practice .

In this work, some basics on the DC machine are given. An overview on the PIC 16F877A which is dedicated to this aplications.

Keywords: PIC microcontroller(**Periphirol Interface contrôler**)- Modeling –simulation **- PI** regulator-**DC** motor

Sommaire

SOMMAIRE

Liste des figures…….…I

Liste des tableaux…….…..II

Liste des acronymes……...…III

Introduction générale…….. ..……..1

Chapitre I : machine à courant continu

I.1 Introduction ……...….2

I.2 Organisation de la machine……2

I.2.1 pôles inducteurs …….. ..2

I.2.2 Stator ou inducteur……..2

I.2.3 Induit « rotor »……..…..3

I.2.4 Collecteur……...…….3

I.3 Caractéristiques mécaniques et électriques ……..…….....5

I.4 Les différents types de moteurs……..6

I.5 Les organes mécaniques ……7

I.6 Principe de fonctionnement…….……..7

I.7 Force électromotrice……8

I.8 Fonctionnements……..8

I.8.1 fonctionnement en moteur……...…8

I.8.2 fonctionnement en génératrice……..……...9

I.8.3 Bilan des puissances……...……9

I.9 Réversibilité……10

I.10 Variation de la vitesse de la machine à courant continu……..……12

I.11 Procède de commande d'un moteur à courant continu…….……..12

I.11.1 réglage rhéostatique……13

I.11.2 réglage par le flux…….…..13

I.11.3 réglage par tension……..13

I.12 Modélisation en régime dynamique……..13

I.12.1 fonction de transfert du moteur à courant continu à excitation séparée…….……13

I.12.2 fonction de transfert du moteur en régime sous-excité……..…….16

I.12.3 fonction de transfert du moteur à excitation série…….……16

I.13 Sources…….17

I.14 avantages et inconvénients…….18

I.15 conclusion…….19

Chapitre II : Régulation analogique

II.1 Introduction ……20

II.2 Partie théorique ……..… 20

II.2.1 Fonction de transfert discrète du PI ……...… 20

II.2.2. Encodeur incrémental…….……....25

II.3 la partie de simulation …….……..... 27

II.3.1Les schéma bloc……29

II.3.2 Résultats de simulation…….30

II.3.3 Discussion…… 34

II.4 conclusion …….…….…….. 34

Chapitre III : microcontrôleur PIC16F877A

III.1 Introduction ……..35

III.2 Les caractéristiques du PIC16F877A ……..35

III.3 Description générale du PIC16F877A ……36

III.3.1. Architecture externe……..37

III.3.2 Architecture interne……..……...39

III.3.3 Organisation de la mémoire……...……40

III.3.4 Les registres du PIC16F877A……..……...41

III.3.5 Les timers……...……43

III.3.6 Les interruptions……45

III.3.7 Les ports d'entré sortie……..……45

III.3.8 Configuration des portx, les registres PORTX et TRISX…….….46

III.3.9 Le convertisseur analogique numérique ……...….47

III.4 Les avantages des microcontrôleurs ……..... 49

III.5 Conclusion ……..... 49

Chapitre IV : Implémentation et régulation

IV.1 Introduction …….. 50

IV.2 Les outils de développements ……...… 50

IV.2.1. Le proteus ISIS…….……51

IV.2.2 MikroC PRO …….…..52

IV.3 Implémentation du régulateur PI…….55

IV.3.1 principe de fonctionnement……...….56

IV.3.2.Organigramme…….…….……..62

IV.3.3. schéma électrique……..……..……..63

IV.3.4. Résultat de simulation…….…….……64

IV.4. Conclusion…….……67

Conclusion générale…….68

Annexe…….

Bibliographie .....

LISTE DES FIGURES

Figure I.1. ***a** : aiment, **b** : bobines* 3
Figure I.2. *Indui* …….*3*

Figure I.3. *Caractéristique magnétique d'une machine à courant con*……..……..5

Figure I.4. *Schéma équivalent du moteur à CC* ……..……..…… 5

Figure I.5. *Caractéristiques mécaniques et électriques* ……..…..….... 6

Figure I. 6. *Constitution de la MCC* …….……..7

Figure I. 7. *Principe de fonctionnement* ….....8

Figure I.8. *Bilan des puissances d'un moteur à courant continu* ……....……9

Figure I. 9. *Bilan des puissances d'un générateur à courant continu* ……..….....10

Figure I. 10. *Fonctionnement d'un moteur dans les quatre quadrants* ……...11

Figure I. 11. *Cycle de fonctionnement dans les quatre quadrants d'une machine tournante*...11

Figure I. 12 ***a**. Variation de la vitesse en fonction de la tension d'induit **b**. Variation de la vitesse en fonction du flux d'inducteur* …….....12

Figure I.13. *Classification des huit fonctions de transfert du moteur à courant continu (obtenus en négligeant L, f ou C0).* ……...……...14

Figure I.14.*Schéma fonctionnel linéarisé autour d'un point de fonctionnement* (Ω_0, I_0, Φ_0).....16

Figure I.15.*Schéma fonctionnel du moteur à courant continu à excitation série, sans affaiblissement.*……..17

Figure II.1. *Schéma blocs du compensateur PI.*……..... 21

Figure II.2. *Approximation rectangulaire* …….…….. 22

Figure II.3. *Approximation rectangulaire devancée.*…….……...…… 23

Figure II.4. *Approximation trapézoïdale* ……..……..…..…24

FigureII.5. *Schéma bloc en z du compensateur PI.*…….. 25

Figure II.6. *Encodeur incrémental* …….……..……26

Figure II.7. *Encodage X2.*……..26

Figure II.8. *Schéma bloc du MCC en boucle ouverte en charge* ……...……. 29

Figure II.9. *Fenêtre de réglage des paramètres de la machine à courant continu*……..….30

Figure II.10. Le couple en fonction de temps ……..…..…..…..…..…30
Figure II.11. *La vitesse en fonction de temps* ……..…..……... 31
Figure II.12. *Le courant en fonction de temps* ……..…..…31
Figure II.13. *Schéma bloc du MCC en boucle fermé en charge*……...32
Figure II.14. *Le courant en fonction de temps*……..32
Figure II.15. *le couple en fonction de temps*…….33
Figure II.16 **.** *la vitesse et la consigne en fonction de temps*…….33
Figure III.1.*Boitier du PIC 16F877A*……..……..37
Figure III.2. *Architecture interne du PIC 16F877A* ……..…..……40
Figure III.3. *Schéma du timer 0.*……..…..43
Figure III.4. *Schéma descriptive du timer 1.*……...44
Figure III.5. *Schéma du timer 2.* ……...44
Figure III.6.*Les différents PORT de PIC 16F877A*…….…46
Figure III.7 *Schéma fonctionnel analogique numérique*…….……47
Figure IV.1. *L'environnement IDE du compilateur microC PRO* …….……....53
Figure IV.2. *boite de dialogue « options »*…….. 54
Figure IV.3. *assistant de code* ……..... 54
Figure IV.4. *Schéma synoptique* ……..... 56
Figure IV.5 **.** *La conversion analogique numérique* ……....57
Figure IV.6 *Schéma bloc de la fonction de transfert* PI……...58
Figure IV.7. *Schéma électrique de CNA* ……..……...59
Figure IV.8. *Résultat de simulation*.....60
Figure IV.9. *Schéma électrique du timer1.*……..…..…61
Figure IV.10. *Organigramme de Programme principale.*……..…62
Figure IV.11. *Organigramme de Programme De comptage de nombre de tours*……..……62
Figure IV.12. *Organigramme de Programme du génération de la commande*……..……63
Figure IV.13. *Schéma électrique*…….……..……64
Figure IV.14. *Simulation du schéma électrique*……..…….65
Figure IV.15. *Résultat de simulation en (Vref =180 tr/mn)*…….66
Figure IV.16. *Résultat de simulation en (Vref=90tr/mn)*……..……..……67

LISTE DES TABLEAUX

Tableau III.1. *Broches du PIC.* ……..…….…….....*38*

Tableau.III.2. *Bits de registre ADCON1*…..……..……..…*48*

Tableau .III.3. *Bits de registre ADCON0*…….……..……..…..49

LISTE DES ACRONYMES

MCC: machine a courant continu.
PI: Proportionnelle intégral.
DC: Courant Continu (Direct Curent).
PIC: Periphirol Interface contrôler
CMOS: Complementary Metal Oxide Semi-conductor.
PWM:
RAM:
EPROM :
EEPROM :
ROM:
CAN: Convertisseur Analogique Numérique .
CNA : Convertisseur Numérique Analogique.
IDE : Integrated development environment.

Introduction générale

L'industrie, au sens le plus large du terme, ont plus en plus besoin de systèmes à vitesse variable, doués de souplesse et de précision, bien sur, les solutions mécaniques sont encore utilisées, mais les solutions électriques sont aujourd'hui plus appréciées, leurs succès viennent des caractéristiques incomparables que leur confère l'électronique sur le plan de la conversion de l'énergie.

L'application de la régulation de vitesse des moteurs électriques est très répondue dans les systèmes industriels et automatiques, l'un des critères d'efficacité d'une telle régulation est sa robustesse face aux changements. Elle permet d'effectuer une régulation des paramètres physiques (vitesse, position, température… etc.) des systèmes en boucle fermée, c'est à dire un système capable d'atteindre et de maintenir une certaine consigne en utilisant une mesure, dont il reste à déterminer la fonction permettant de corriger la commande en fonction de la consigne initiale et de l'erreur mesurée.

Notre objectif dans ce mémoire est l'implémentation le régulateur PI dans le microcontrôleur Pic 16f877a pour faire la commande de moteur a courant continu pour ce travail on doit utiliser le logiciel MikroC pro pour la programmation du Pic et le logiciel Proteus ISIS pour la simulation, on va réaliser quatre chapitres et une conclusion générale:

Le premier chapitre décrit une généralité sur le moteur a courant continu et ses principales caractéristiques.

Le deuxième chapitre contient deux parties, dans la première partie on a fait une étude théorique sur le régulateur PI et sur l'encodeur incrémental, dans la deuxième partie on a fait une régulation analogique avec le compensateur PI dont l'outil de simulation est le logiciel MATLAB simulink.

Dans le troisième chapitre on a présenté le microcontrôleur Pic 16f877a et son architecture (interne et externe), leur caractéristique et ses différents registres

Le quatrième chapitre est consacré à l'implantation et à la simulation de la commande de moteur a courant continu et faire les résultats de simulation.

Nous terminons notre travail par une conclusion générale.

Chapitre I :

Machine a courant continu

I.1. INTRODUCTION

L'évolution des technologies conduit à utiliser des machines nécessitant des vitesses de rotation précises et variables pour l'entraînement d'engins de manutention par exemple.

Une machine à courant continu est une machine électrique. Il s'agit d'un convertisseur électromécanique permettant la conversion bidirectionnelle d'énergie entre une installation électrique parcourue par un courant continu et un dispositif mécanique. Elle est aussi appelée dynamo.

En fonctionnement moteur, l'énergie électrique est transformée en énergie mécanique.
En fonctionnement générateur, l'énergie mécanique est transformée en énergie électrique, la machine se comporte comme un frein.

I.2. ORGANISATION DE LA MACHINE

Dans l'organisation d'une machine à courant continu, on peut distinguer les principaux éléments suivants :

I.2.1 *Pôles inducteurs*

Les pôles inducteurs ont pour rôle de créer le flux inducteur dans la machine «entre la culasse et l'induit ».
Ce flux est généré soit par des enroulements, soit par des aimants.
Ce flux est canalisé dans la machine par des matériaux ferromagnétiques «parties hachurées ».
Ce flux étant constant dans la partie portant les pôles inducteurs et dans les pôles inducteurs eux-mêmes.

I.2.2 *Stator ou inducteur*

Le stator partie fixe, constitué par un aimant ou un électroaimant appelé également inducteur qui crée un champ magnétique dirigé vers l'axe du rotor. Il est formé d'aimant figure (1.1-a) pour les petits moteurs et des bobines figure (1.1-b) parcourue par un courant continu. Les aimants ou bobines permettent la création d'un flux magnétique.



Figure I.1. ***a*** *: aiment,* ***b*** *: bobines* **[2]**

I.2.3 *Induit « rotor »*

Le rotor, partie mobile, appelé également induit, constitué par un cylindre d'acier doux à la périphérie du quel sont disposés des conducteurs reliés aux lames du collecteur sur lesquelles frottent deux charbons, ou balais, qui assurent la liaison avec les bornes du moteur.

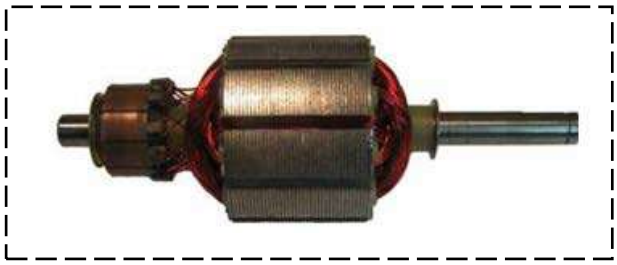


Figure I.2. *Induit* **[2]**

I.2.4 *Collecteur*

Il est formé d'un ensemble de lames de cuivre isolées latéralement les unes des autres et disposées suivant un cylindre en bout de rotor. Deux balais portés par le stator frottent sur les lames du collecteur**. [2]**

Prenons un cas particulier d'une machine à courant continu à vitesse variable dont la réaction du courant d'induit et du champ inducteur crée un couple électromagnétique **C** donné par la formule :

$$C = K \varphi i \qquad \text{(I.1)}$$

Avec $k = \left(\frac{2p_1}{2a}\right)\frac{n}{2\pi}$ coefficient de couplage électromécanique,

$2p_1$: nombre de pôles,

2a : nombres de voie d'enroulement,

n : nombre de conducteurs de l'induit.

La conservation de l'énergie appliquée à l'induit

$$C\Omega = e\, i \qquad \textbf{(I.2)}$$

Avec e force contre-électromotrice (f.c.é.m.) due à la vitesse angulaire de rotation Ω du rotor.

Cette relation n'a de sens que dans un système normalisé (Ω en rad/s).

Par suite, il vient :

$$e = k\,\varphi\,\Omega \qquad \textbf{(I.3)}$$

L'équation électrique relative à l'induit est : relative à l'induit est :

$$v = Ri + L\frac{di}{dt} + e = Ri + L\frac{di}{dt} + k\,\varphi\,\Omega \qquad \textbf{(I.4)}$$

Avec L et R inductance propre et résistance de l'induit.

Dans le cas des machines excitées par un courant i_f, il est nécessaire de connaitre la caractéristique magnétique **(Fig.I.3)**

$$\varphi = F(i_f) \qquad \textbf{(I.5)}$$

L'équation relative à l'inducteur, dont les n_f spires ont une résistance R_f, est :

$$V_f = R_f\, i_f + n_f \frac{d\varphi}{dt}$$

(I.6)

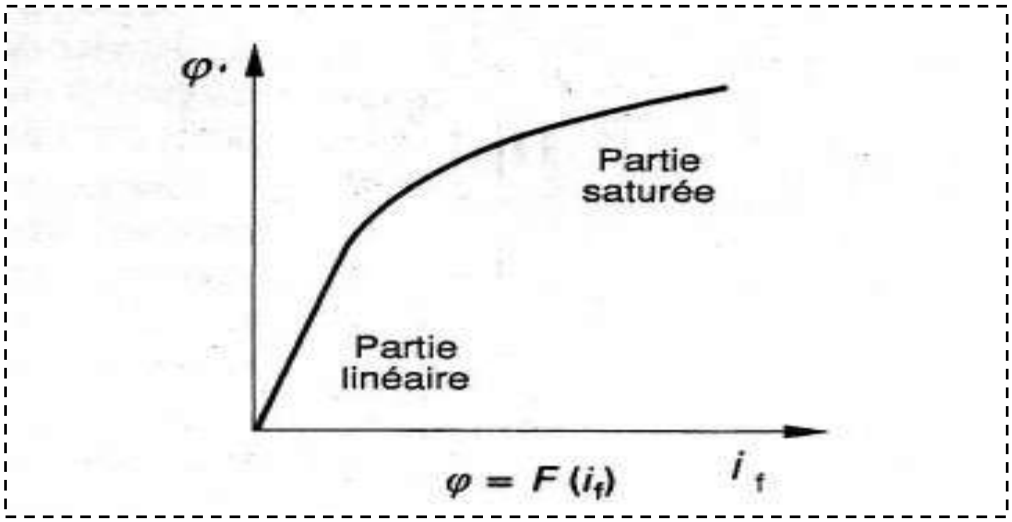


Figure I.3. *Caractéristique magnétique d'une machine à courant continu* **[1]**

Quand on réalise cette équation, on définit l'inductance propre $\mathbf{L_f}$ de l'inducteur **:**

$$\mathbf{L_f = n_f}\ \frac{d\varphi}{dt} \qquad \textbf{(I.7)}$$

La figure I.4 représente un schéma équivalent de la machine, dont le paragraphe **1.7** donnera le modèle dynamique complet**. [1]**

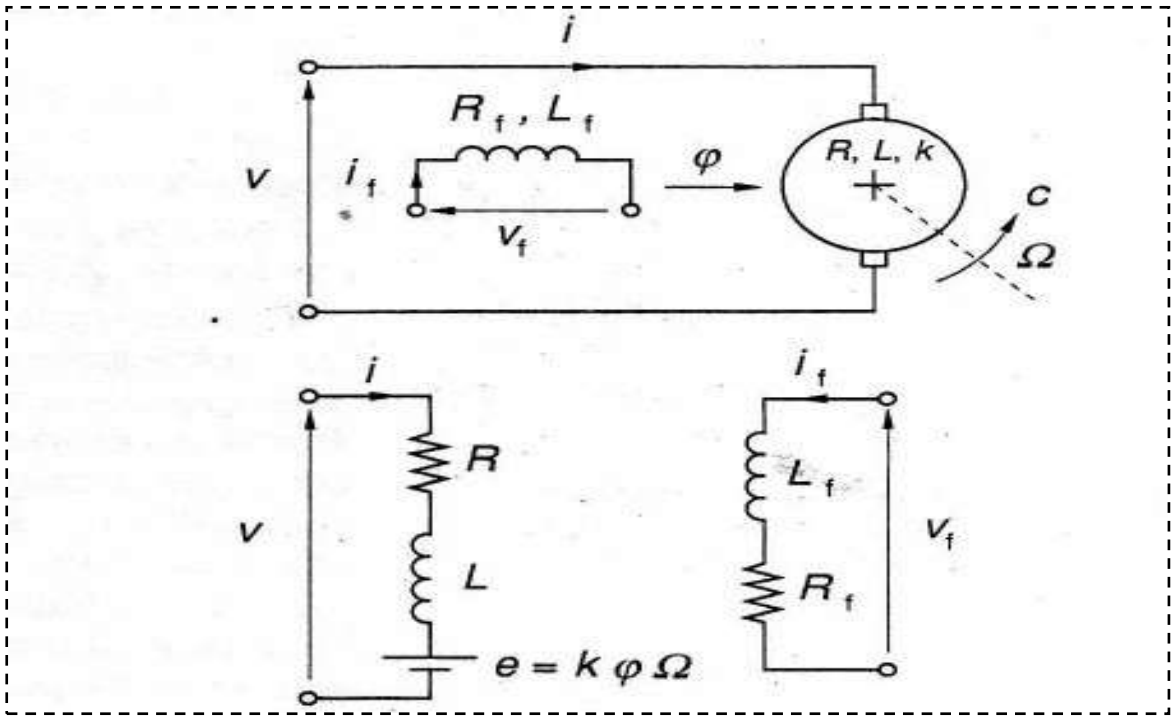


Figure I.4. *Schéma équivalent du moteur à CC.* **[1]**

I.3 CARACTERISTIQUES MECANIQUES ET ELECTRIQUES

On constate, voir figure (1.5), que la fréquence de rotation n diminue lorsque le couple C augmente « à tension U constante ».

Le graphe de droite montre que la fréquence de rotation est proportionnelle à la tension, à couple constant.

Remarque : Pour inverser le sens de rotation, on inversera le courant dans l'induit « on pourrait aussi inverser les courant inducteur, mais cela n'est pas possible si le flux inducteur est créé par des aimants permanents ». **[2]**

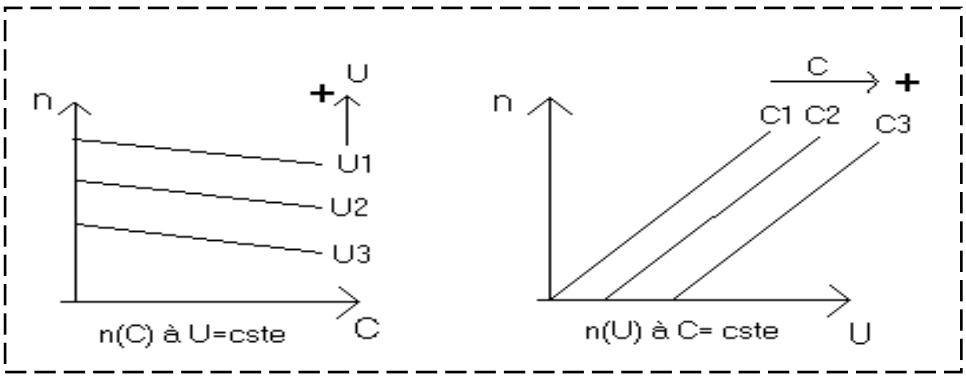


Figure I.5. *Caractéristiques mécaniques et électriques* **[1]**

I.4. LES DIFFERENTS TYPES DE MOTEURS

On peut réaliser l'excitation des moteurs de plusieurs façons :

a. En dérivation : C'est une excitation séparée, l'inducteur comporte beaucoup de spires de faibles sections.

b. En série : L'inducteur est traversé par le courant d'induit, il comporte peut de spires de grosses sections.

c. Composé : On trouve une partie des enroulements inducteurs en série et une partie en parallèle. On distingue deux types de branchements :

c.1 Courte dérivation : L'enroulement parallèle est aux bornes du moteur.

c.2 Longue dérivation : L'enroulement parallèle est aux bornes de la source d'alimentation.

On peut réaliser pour chaque type de branchement deux types d'excitations :

Une excitation à flux additif, le flux de l'enroulement série s'ajoute avec celui de l'enroulement parallèle.

Une excitation à flux soustractif, le flux série se soustrait de celui de l'enroulement parallèle.

Le type d'excitation a une influence sur le comportement du moteur. **[2]**

I.5. ORGANES MECANIQUES

Le stator autoporteur reçoit de chaque côté un flasque sur lequel le rotor sera positionné grâce à des roulements à billes ou à rouleaux suivant le type de charge « axiale ou radiale ». Les pôles inducteurs sont en général vissés peuvent aussi en faire partie intégrante ».

Un ventilateur est placé en bout d'arbre, sur le rotor, pour le refroidissement de la machine, il peut être complété par une ventilation forcée motorisée pour le refroidissement aux vitesses lentes. **[2]**

Le collecteur, l'ensemble porte-balais et les balais se situent de l'autre côté de la machine.

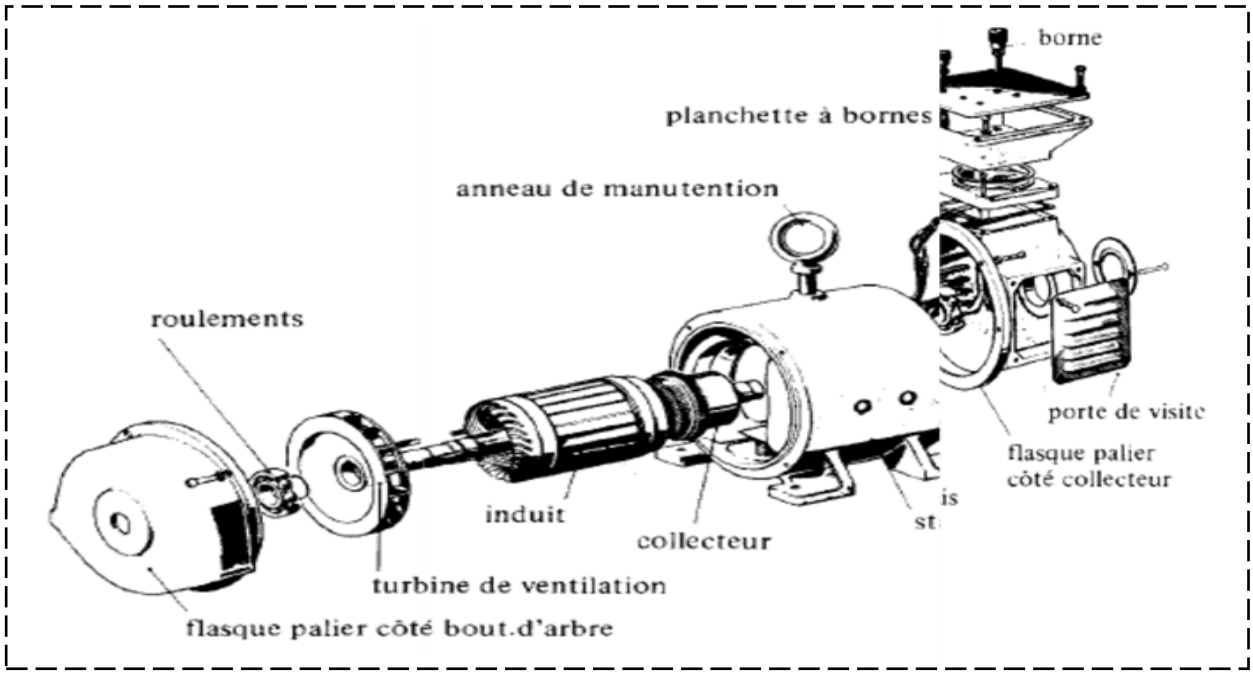


Figure I.6. *Constitution de la MCC* **[2]**

I.6. PRINCIPE DE FONCTIONNEMENT

Lorsque les conducteurs sont parcourus par un courant, ils sont soumis à des forces F1 et F2 qui tendent à faire tourner le rotor. Le collecteur permet d'inverser le sens du courant dans les conducteurs lorsque ceux-ci passent le plan vertical. Ainsi le sens du couple des forces F1 et F2 et donc le sens de rotation du moteur est conservé. **[2]**

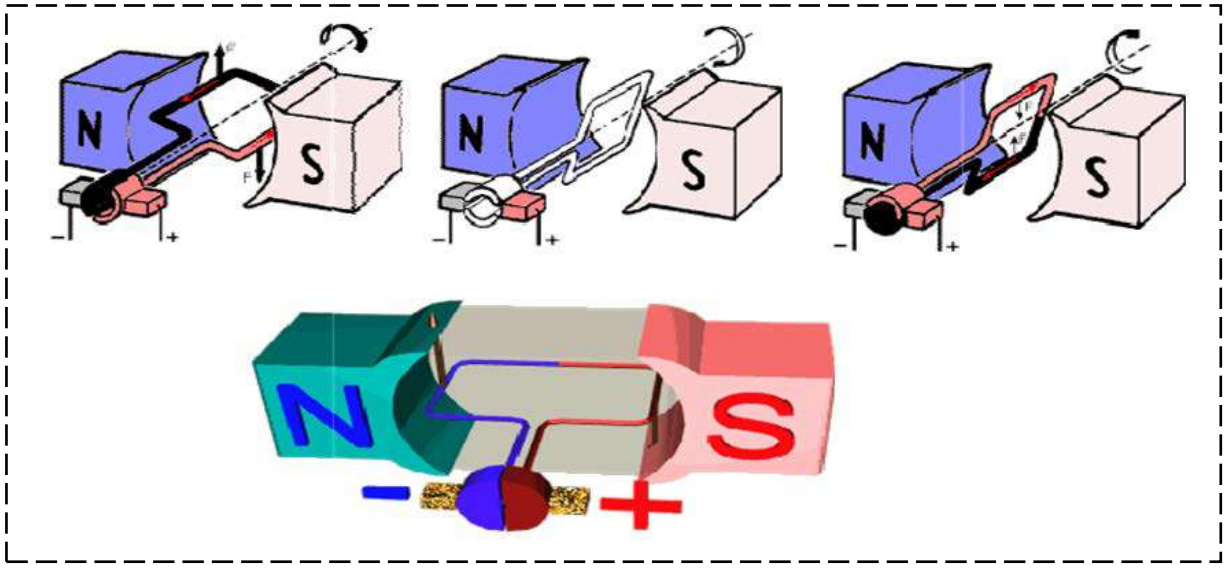


Figure I.7. *Principe de fonctionnement* **[2]**

I.7. FORCE ELECTROMOTRICE

Nous savons qu'une bobine en mouvement dans un champ magnétique voit apparaître à ses bornes une force électromotrice « **f.é.m**. » donnée par la loi de Faraday, sur ce principe, la machine à courant continu est le siège d'une f.é.m E :

$$E=\frac{p}{2\pi a} N \phi \Omega \qquad \textbf{(I.8)}$$

I.8 FONCTIONNEMENTS

Imaginons une machine électrique alimentée par une source de tension U constante. Lorsque le moteur tourne à vide il n'a pas besoin de fournir de couple, $I_{\underline{a}}$ est très faible et $U \approx E$. La vitesse de rotation est proportionnelle à U.

I.8.1 *fonctionnement en moteur*

Lorsque l'on veut la faire travailler, en appliquant un couple résistant sur son axe, cela la freine donc E diminue. Comme U reste constante, le produit $R_i.I_i$ augmente donc I_i

augmente, donc le couple *T* augmente lui aussi et lutte contre la diminution de vitesse : c'est un ***couple moteur***.

Plus on le freine, plus le courant augmente pour lutter contre la diminution de vitesse. C'est pourquoi les moteurs à courant continu peuvent « griller » lorsque le rotor est bloqué, si le courant de la source n'est pas limité à une valeur correcte.

I.8.2 *fonctionnement en génératrice*

Si une source d'énergie mécanique essaie d'augmenter la vitesse de machine, « la charge est entraînante : ascenseur par exemple », Ω augmente donc *E* augmente. Comme *U* reste constante, le produit $R_i.I_i$ devient négatif et augmente en valeur absolue, donc I_i augmente, donc le couple *T* augmente lui aussi et lutte contre l'augmentation de vitesse : c'est un ***couple frein***.

Le signe du courant ayant changé, le signe de la puissance consommée change lui aussi. La machine consomme une *puissance négative*, donc elle fournit de la puissance au circuit. Elle est devenue génératrice. **[2]**

Ces deux modes de fonctionnement existent pour les deux sens de rotation de la machine. Celle-ci pouvant passer sans discontinuité d'un sens de rotation ou de couple à l'autre.

On dit alors qu'elle fonctionne dans les quatre quadrants du plan couple vitesse.

I .8.3 *Bilan des puissances*

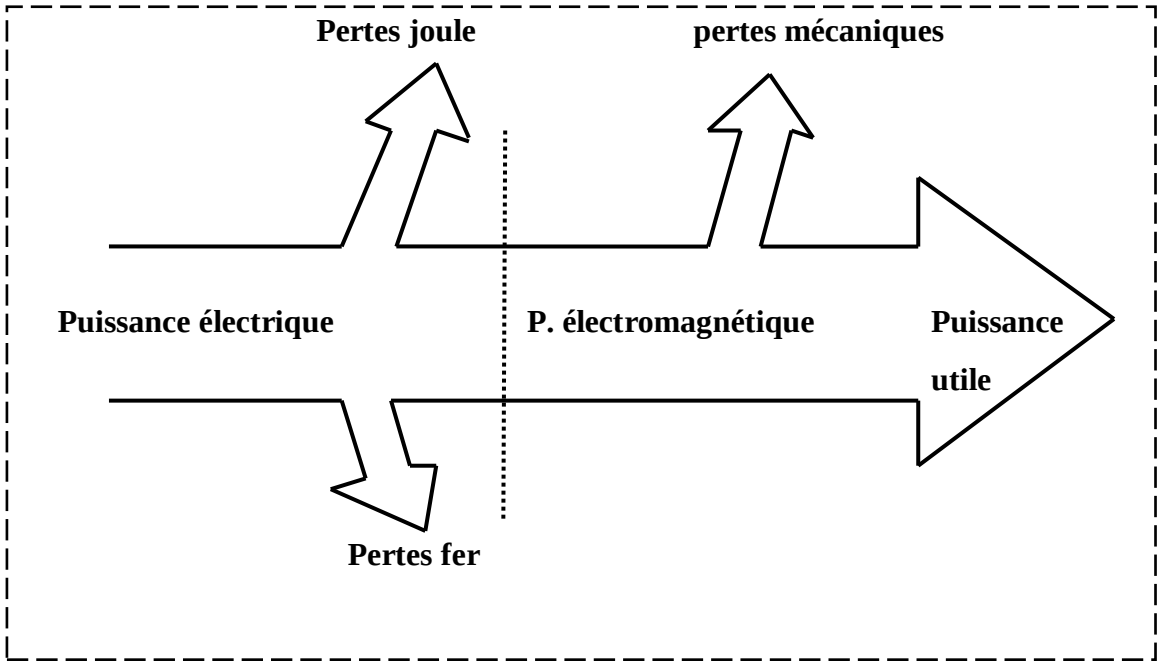


Figure I.8. *Bilan des puissances d'un moteur à courant continu* **[2]**

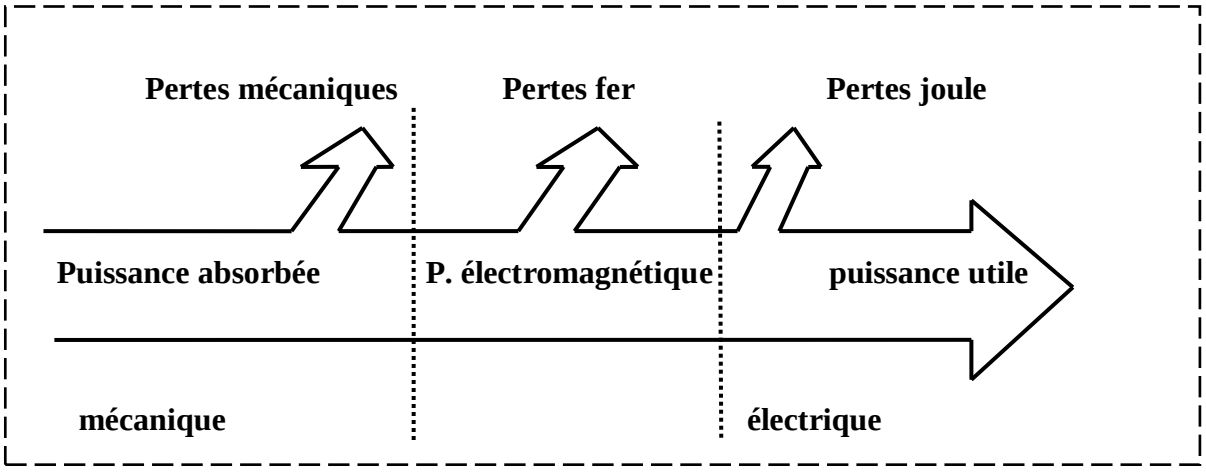


Figure I.9. *Bilan des puissances d'un générateur à courant continu* **[2]**

a. Puissance utile « sur l'arbre en sortie moteur » :

$P_u = P_{em}$ – « pertes mécaniques + pertes magnétiques », en Watts

b. Puissance absorbée « en entrée » :

P_{abs} = U.I « U : tension d'alimentation en Volt, I : courant absorbé en ligne, P_{abs} en Watts»

c. Pertes :

Pertes par effet joules « $R.I^2$ », pertes magnétiques, et pertes mécaniques

d. Rendement :

Le rendement = Puissance fournie « utile » / Puissance totale absorbée. = P_u / P_{abs}

P_{abs} = U.I « + puissance absorbée par l'inducteur » $P_u = C_u.\Omega$, d'où rendement = $(C_u*\Omega)/UI$

I.9 REVERSIBILITE

La MCC est fondamentalement réversible. Avec nos conventions, elle fonctionne en moteur si la puissance convertie est positive :

$$P_e = c\,\Omega = k\,\varphi\,i\,\Omega > 0 \qquad \textbf{(I.9)}$$

Et en génératrice (frein) si cette même quantité est négative.

La **figure I.10** définit le fonctionnement possible d'une machine dans les quatre quadrants :

Moteur : vitesse et couple positifs ;

Génératrice-frein : vitesse positive et couple négatif ;

Moteur : vitesse et couple négatifs ;

Génératrice-frein : vitesse négative et couple positif ;

Le passage du quadrant **I** au quadrant **IV** se rencontre dans le cas du levage : le couple ne change pas de signe, mais la vitesse est positive à la montée (charge résistante) et négative à la descente (charge entrainante).

La **Figure 8** donne un exemple typique de cycle de fonctionnement avec inversion du signe de la vitesse et passage du quadrant **I** au quadrant **II** (succession de phases d'accélération et de freinage, comme on en rencontre en traction ou en robotiques. **[1]**

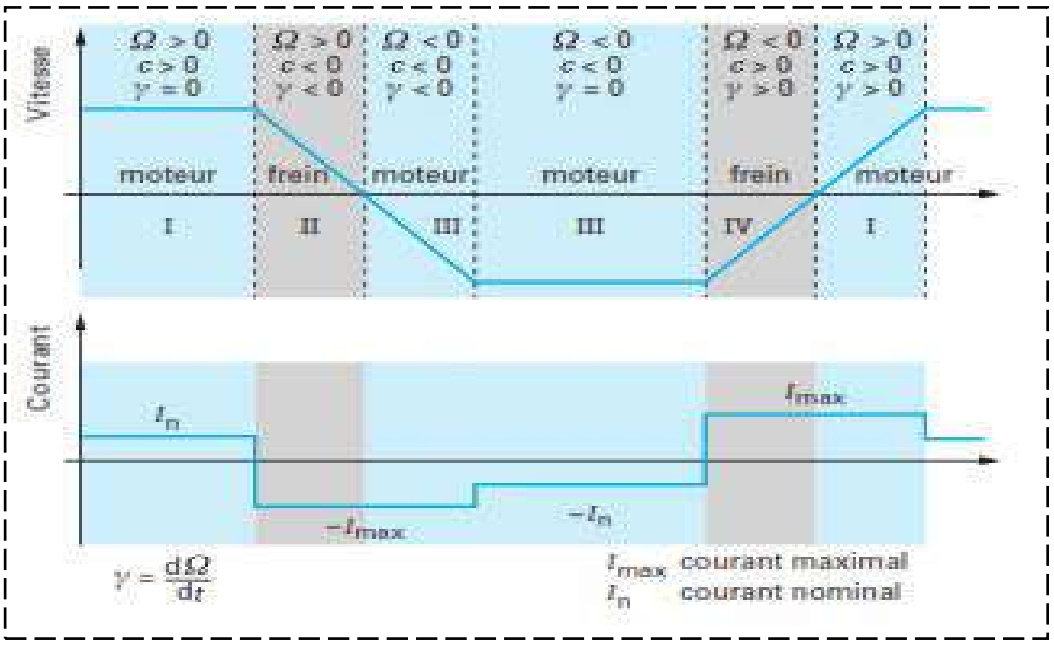


Figure I.10. *Fonctionnement d'un moteur dans les quatre quadrants* **[1]**

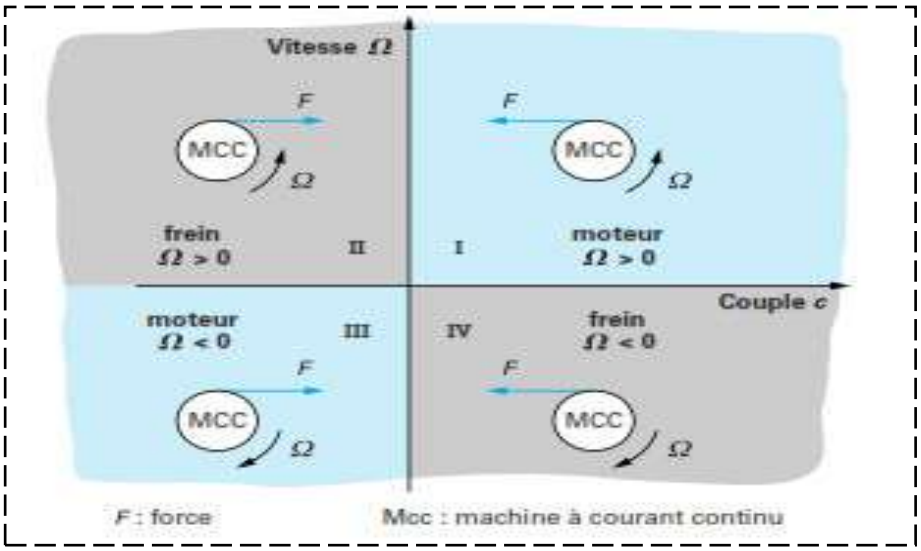


Figure. I.11. *Cycle de fonctionnement dans les 4 quadrants d'une machine tournante.* **[1]**

Cet exemple montre que, pour avoir de bonnes performances, le moteur doit pouvoir fonctionner dans les quatre quadrants. Cela exige que l'alimentation soit également réversible et nous examinerons cette question à propos des convertisseurs. En particulier, l'examen des équations [surtout la relation **(1)**] montre que la réversibilité exige la possibilité de changer le signe du courant d'induit, si le couple doit s'inverser et si on travaille à excitation constante.

I.10 VARIATION DE LA VITESSE DE LA MACHINE A COURANT CONTINU

$$\Omega = \frac{U\ Ra\ Ia}{K} \qquad \textbf{(I.10)}$$

La figure suivante représente : **a :** action sur la tension à flux constant- la figure représente la courbe de variation de la vitesse en fonction de la tension pour une charge constante.

b : action sur le flux à tension constante, la figure représente la courbe de variation de la vitesse en fonction du flux d'excitation pour une charge constante. **[1]**

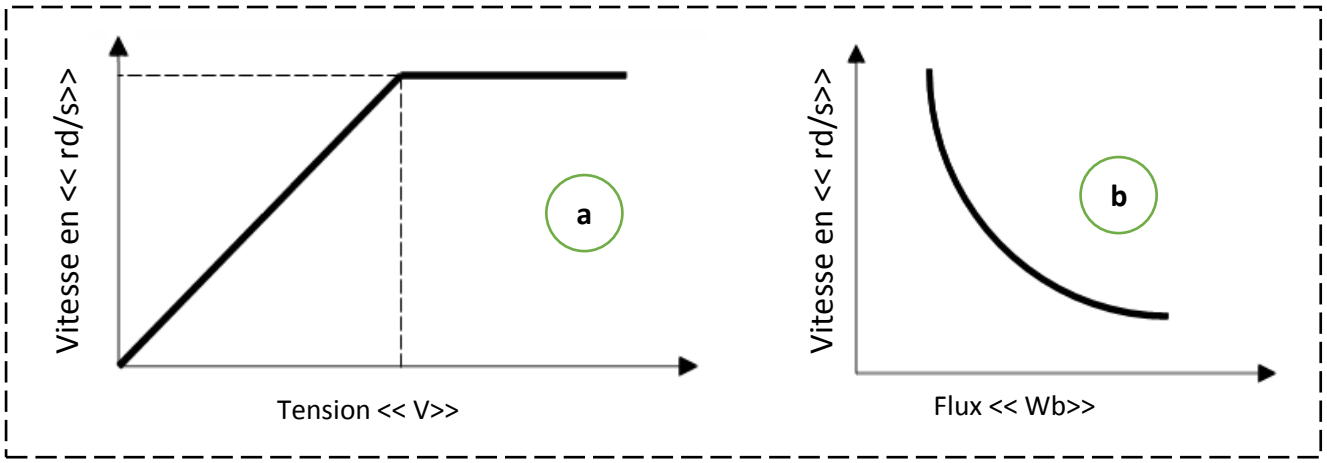


Figure I.12. ***a****. Variation de la vitesse en fonction de la tension d'induit* ***b****. Variation de la vitesse en fonction du flux d'inducteur* **[1]**

I.11 PROCEDE DE COMMANDE D'UN MOTEUR A COURANT CONTINUE

L'expression générale de la vitesse montre les trois paramètres R_a, R_{reg} et U que l'on peut régler ce qui donne les trois procédés pour la variation s de vitesse :

I.11.1 *Réglage rhéostatique*

Il est possible de réduire la vitesse en augmentant la résistance de l'induit avec un rhéostat branchée en série avec l'induit tout en fixant la tension et le flux à leurs valeurs

nominales, l'expression de la vitesse est :

$$\Omega = \frac{U\ Ra\ Rreg\ Ia}{K} \qquad \textbf{(I.11)}$$

Ce type de réglage est mauvais du point de vue rendement. Le rhéostat insère a une consommation d'énergie d'autant plus importants que la chute de vitesse est élevée, c'est ainsi que ce procède de réglage n'est utilisé que pour le démarrage et le freinage.

I.11.2 *Réglage par le flux*

Au démarrage, un moteur doit absorber un courant important pour pouvoir porter à la vitesse nominale toutes les masses et charges qui sont initialement au repos, d'où la nécessite d'appliquer un flux maximum pour permettre au moteur de développer une force électromotrice d'une valeur suffisante et d'une manière rapide.

Ce type de réglage est bon du point de vue technique, et également du point de vue économique car la puissance dissipée dans l'inducteur étant très faible par rapport à la puissance absorbée, le rendement du moteur ne sera pas modifier.

I.11.3 *Réglage par tension*

On applique une tension de valeur moyenne variable entre zéro et une valeur maximale tout en réglant le flux à sa valeur maximale. Ce procédé de réglage de vitesse est excellent du point de vue économique car aucune énergie n'est dissipe dans les rhéostats. Le réglage par la tension permet seulement de diminuer la vitesse.

I.12 MODELISATION EN REGIME DYNAMIQUE

I.12.1 *Fonction de transfert du moteur à courant continu à excitation séparée*

En vitesse variable, la machine est associée à des convertisseurs et à des régulateurs dont il faut déterminer les paramètres : c'est la synthèse du système, qui ne peut se faire raisonnablement que dans le cadre du système linéaire continu, à coefficients constants. Or, au paragraphe 1,61, nous avons vu qu'une analyse physique précise conduit à des modèles non linéaires. Pour simplifier les équations, on ne représente pas les effets de la saturation ou de la réaction d'induit. On considère que l'inductance de l'induit **L** est constante et onregroupe, en une expression du premier degré ($f\Omega+C_0$), l'ensemble des couples de frottement et du couple de charge. On se ramène à des équations différentielles linéaires à

coefficients constants. Le modèle dynamique complet est défini par :

$$V = R\,i + L\frac{di}{dt} + K\,\Omega \qquad \textbf{(I.12)}$$

$$C = K\,i = J\frac{d\Omega}{dt} + f\,\Omega + C_0 \qquad \textbf{(I.13)}$$

Avec $\mathbf{C_0}$ **:** couple de charge,

f : coefficient de frottement visqueux,

j : inertie des parties tournantes.

Il est fréquent (mais il faut le vérifier dans chaque cas) que les paramètres **L** et **f** soient négligeables. Par ailleurs, on ne peut pas s'intéresser à l'effet de $\mathbf{C_0}$ (par exemple quand le moteur est à vide) ; aussi, selon que l'on néglige **L**, **f** ou $\mathbf{C_0}$, on obtient huit fonctions de transfert possibles, correspondant à des niveaux de simplification différents

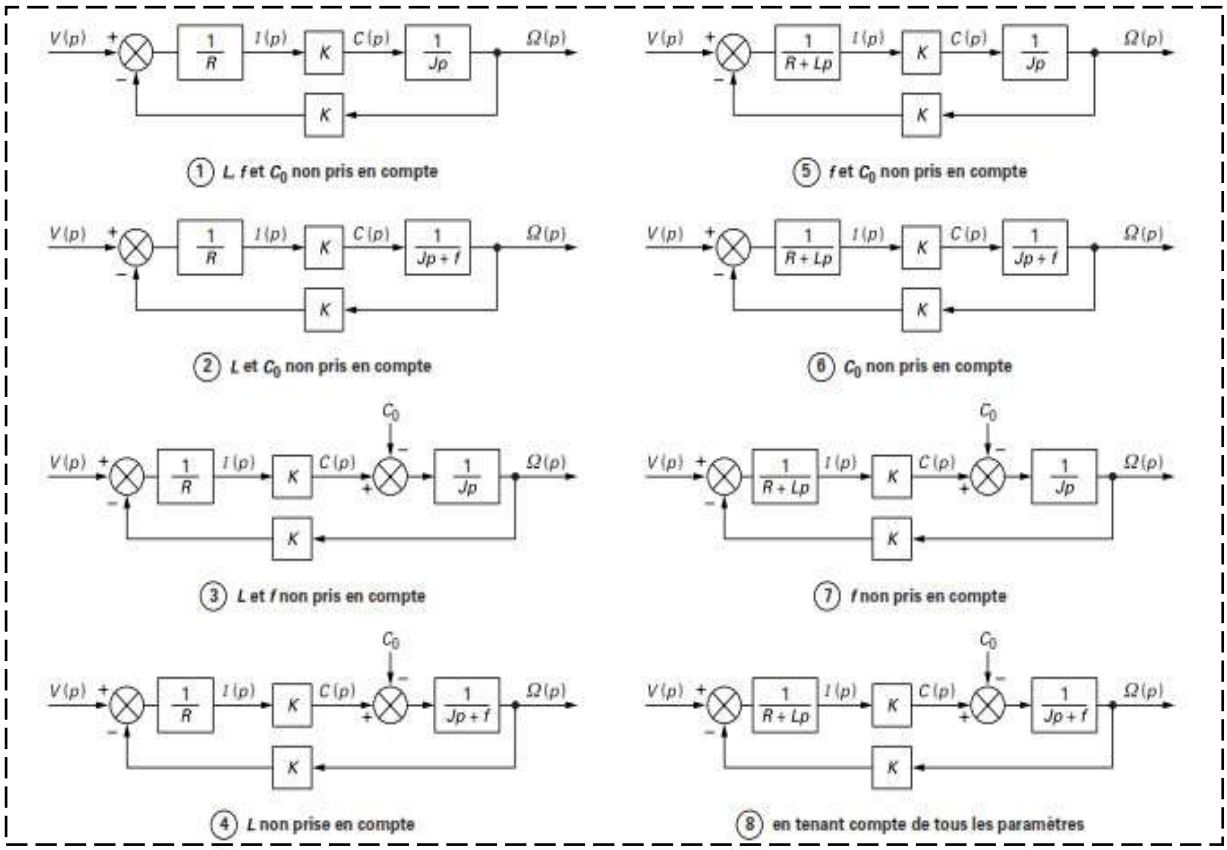


Figure I.13. *Classification des huit fonctions de transfert du moteur à courant continu (obtenus en négligeant L, f ou C0).* **[1]**

On reconnait, dans le modèle -1-, le modèle le plus simple qui nous a servi à définir les caractéristiques dynamiques des servomoteurs. A l'opposé, le modèle le plus complet est le modèle -8- déduit sans simplification des relations I.12 et I.13 .La fonction de transfert a pou expression complète :

$$\frac{\Omega(p)}{V(p)} = \frac{Km}{1+(\tau_{em}+\mu\tau_e)p+\tau_e\tau_{em}p^2} \qquad \textbf{(I.14)}$$

Avec $K_m = K/ (K^2+R\ f)$ gain statique,

$\mu = R\ f/(K^2+R\ f)$ coefficient habituellement petit,

$\tau_e = L/R$ Constante de temps électrique ;

$\tau_{em} = \frac{RJ}{K^2+Rf}$ Constante de temps électromécanique, habituellement (mais improprement) appelée constante de temps mécanique dans la littérature technique, ou on la noteτ_m.

Le plus souvent, mais pas toujours, les pôles de la fonction de transfert sont réels. On peut alors définir deux constantes de temps$\tau' et\ \tau''$. On obtient alors :

$\mu=0$

$K_m=1/K$

$\tau' \approx \tau_e$

$\tau'' \approx \tau_{em} = R\ J/K^2$

Dans ces conditions, la fonction de transfert est habituellement écrite :

$$\frac{\Omega(p)}{V(p)} = \frac{1}{K}\ \frac{1}{(1+\tau_e p)(1+\tau_{em}p)} \qquad \textbf{(I.15)}$$

I.12.2 *Fonction de transfert du moteur en régime sous-excité*

La figure 14 représente le schéma fonctionnel du moteur en régime sous-excité, la tension d'induit étant constante **(v = V_n)** et la tension d'excitation variable$(\mathbf{V_f})$.

Ce modèle est non linéaire, à cause des caractéristiques magnétiques que l'on retrouve dans l'équation du circuit d'excitation **(5)** et des opérateurs multiplicatifs qui interviennent dans les expressions du couple et de ka force contre électromotrice

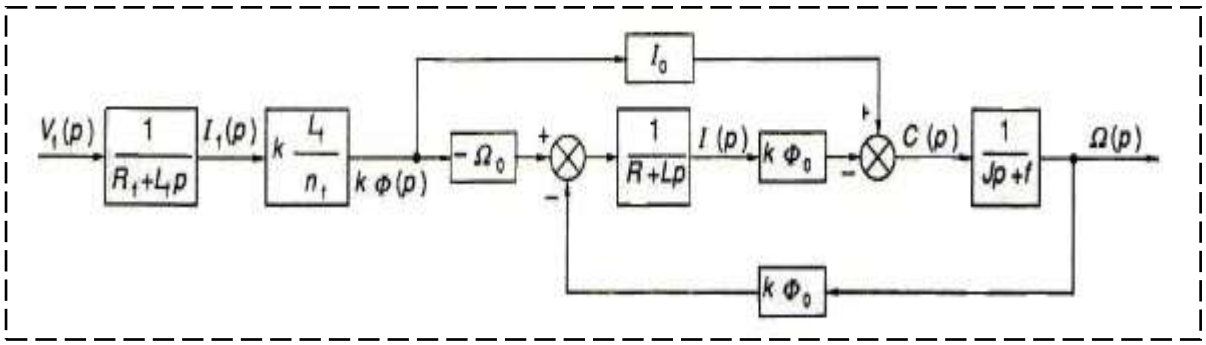


Figure I.14. *Schéma fonctionnel linéarisé autour d'un point de fonctionnement (Ω0, I0, Φ0)* [1]

Pour l'étude des régulations, il est naturel de linéariser ce modèle autour d'un point de fonctionnement défini par$(\mathbf{\Omega_0}, \boldsymbol{i_0}, \mathbf{\Phi_0})$**,** l'inductance propre de l'inducteur étant définie par la relation (6) :

$$\mathbf{L_f = n_f \frac{d_\varphi}{di_f}} \quad \textbf{(I.16)}$$

La **figure 14** donne le schéma fonctionnel linéarisé. D'où l'on peut déduire la fonction de transfert du système. Celle-ci est caractérisée par la présence de la constante de temps de l'excitation $\boldsymbol{\tau_f = L_f/R_f}$ habituellement très grande.

I.12.3 *Fonction de transfert du moteur à excitation série*

Nous avons déjà vu un schéma fonctionnel du moteur à excitation série (fig.15). Pour les mêmes raisons qu'au paragraphe précédent, ce modèle est fortement non linéaire. Pour l'étude de régulation, on peut :

- d'une part, considérer un modèle simplifié, sans affaiblissement, ce qui donne le modèle non linéaire de la figure 15 a, défini par les hypothèses (14) et (6) ;
- d'autre part, déterminer le schéma fonctionnel linéarisé de la figure 15 b, si on linéarise le modèle autour d'un point de fonctionnement $(\Omega_0, I_0, V_0, C_0)$, d'où l'on peut déduire sa fonction de transfert. **[1]**

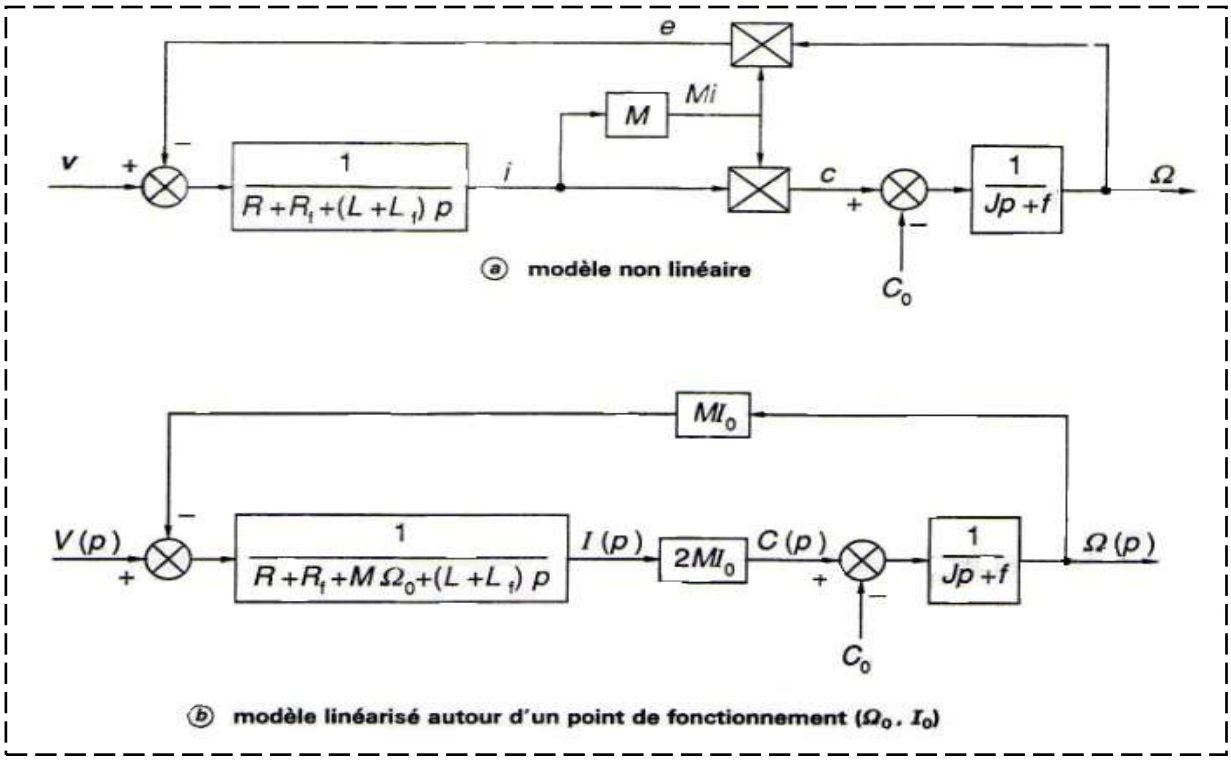


Figure I.15. *Schéma fonctionnel du moteur à courant continu à excitation indépendante.* **[1]**

I.13 SOURCES

On rencontre plusieurs types de sources ou de réseaux.

Les réseaux alternatifs sinusoïdaux industriels dont la fréquence *f* est 50 Hz. Pour les petites puissances (inférieures à 20 kW), on utilise le réseau monophasé 220-380V et pour les grandes puissances (atteignant plusieurs mégawatts), le réseau triphasé, éventuellement sous des tensions supérieures (660 V). En traction ferroviaire, on rencontre des alimentations monophasées sous des tensions élevées (25 kV par exemple).

Les réseaux de tensions continues redressées correspondent :

* Soit à des réseaux propres à la traction [SNCF (société nationale des chemins de fer français), 1500V ; RATP (régie autonome des transports parisiens), 750V], la réversibilité étant partielle, par échange d'énergie entre les machines fonctionnant en génératrice et celles fonctionnant en moteur sur les mêmes réseaux.

* Soit à des réseaux locaux, obtenus par redressement du réseau industriel alternatif, le plus souvent à l'aide de redresseurs à diodes (non réversible) ; on ne dispose plus alors que de la réversibilité dissipative.

Les sources électrochimiques (batteries) fournissent une tension continue, souvent assez basse (12V, 48V….) elles sont réversibles en courant. On les utilise dans les systèmes embarqués autonomes, les engins de manutention et autres véhicules électriques. **[1]**

I.14. AVANTAGES ET INCONVENIENTS

L'avantage principal des machines à courant continu réside dans leur adaptation simple aux moyens permettant de régler ou de faire varier leur vitesse, leur couple et leur sens de rotation : les ***variateurs de vitesse*.** Voire leur raccordement direct à la source d'énergie : batteries d'accumulateur, piles, etc.

Le principal problème de ces machines vient de la liaison entre les balais, ou « Charbons » et le collecteur rotatif. Ainsi que le collecteur lui-même comme indiqué plus haut et la complexité de sa réalisation. De plus il faut signaler que :

Plus la vitesse de rotation est élevée, plus la pression des balais doit augmenter pour rester en contact avec le collecteur donc plus le frottement est important.

Aux vitesses élevées les balais doivent donc être remplacés très régulièrement.

Le collecteur imposant des ruptures de contact provoque des arcs, qui usent rapidement le commutateur et génèrent des parasites dans le circuit d'alimentation, ainsi que par rayonnement électromagnétique.

Un autre problème limite les vitesses d'utilisation élevées de ces moteurs lorsque le rotor est bobiné, c'est le phénomène de « défrettage », la force centrifuge finissant par casser les liens assurant la tenue des ensembles de spires « le frettage ».

Un certain nombre de ces inconvénients ont partiellement été résolus par des réalisations de moteurs sans fer au rotor, comme les moteurs « disques » ou les moteurs « Cloches », qui néanmoins possèdent toujours des balais.

Les inconvénients ci-dessus ont été radicalement éliminés grâce à la technologie du moteur *brushless,* aussi dénommé « moteur à courant continu sans balais », ou moteur sans balais. **[2]**

I.15. CONCLUSION

La machine à courant continu a été durant de nombreuses années l'actionneur principalement utilisé dans les applications à vitesse variable. En effet, comme cela a été

mis en évidence, le contrôle de la vitesse de rotation peut être aisément réalisé par action sur la f.é.m. d'induit de la machine « en grandeur permanente, donc moyenne ». Sur un autre plan, on peut agir sur le flux inducteur qui est réglé par le courant dans l'inducteur « appelé aussi courant d'excitation ».

Chapitre II :

La régulation analogique

II.1 INTRODUCTION

Malgré tous les développements théorique de la commande depuis les 50 dernières années, les compensateurs proportionnel intégrale (PI), proportionnel dérivée (PD), et proportionnel intégral dérivée (PID) demeurent aujourd'hui les plus utiliser en milieu industriel.

Dans cette section, nous allons d'abord développer le modèle discret des compensateurs PI. Nous allons ensuite discuter à l'encodeur incrémental et finalement on voir la régulation analogique de la MCC a travers le compensateur PI.

II.2 LA PARTIE THEORIQUE

II.2.1 *fonction de transfert discrète du PI*

Le schéma bloc de la figure 2.1 illustre un compensateur PI discret. En effet, la sortie m(kT) du compensateur est la somme d'un gain proportionnel Kp multipliée par l'erreur entre la référence r(kT) et la mesure cm(kT) ainsi que l'intégrale de cette erreur multipliée par le gain intégrale Ki . La sortie de compensateur est donc donnée par :

$$m(kT) = Kp\ e(kT) + K_i\ e_i(kT) \quad \textbf{(II.1)}$$

ou e*(kT)* l'erreur et $e_i(kT)$ est l'intégrale de l'erreur qui peut être séparée sur deux intervalles de la façon suivante :

$$e_i(kT)=\int_0^{KT} e(t)dt=\int_0^{(k-1)T} e(t)dt+\int_{(k-1)T}^{kT} e(t)dt \quad \textbf{(II.2)}$$

selon la définition de $e_i(kT)$, la relation II.2 peut être réécrite de la façon suivante :

$$e_i(kT)= e_i((k-1)T)+\int_{(k-1)T}^{kT} e(t)dt \quad \textbf{(II.3)}$$

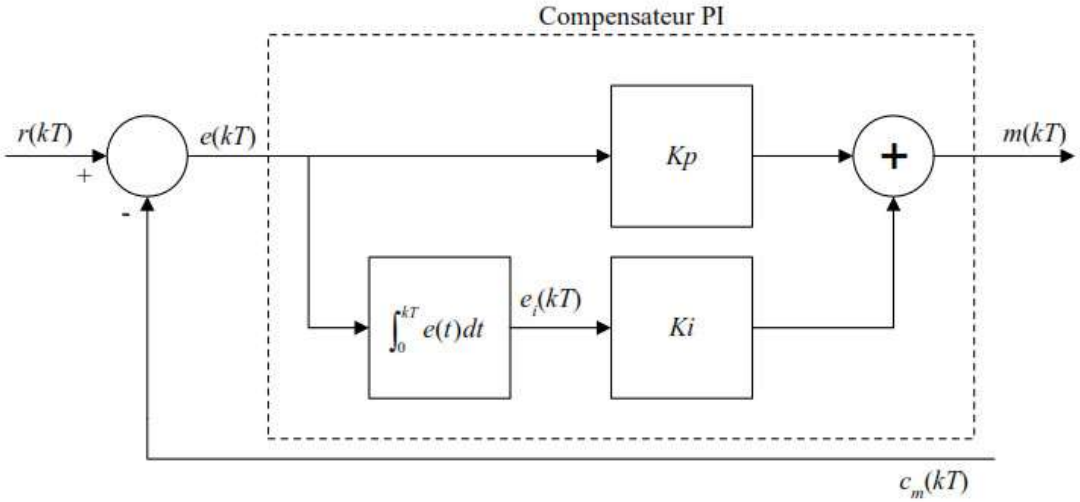


Figure II.1. *Schéma blocs du compensateur PI.* [11]

Le problème est alors d'obtenir l'intégrale de e(t) à partir du signal discret e(kT) qui est connu seulement aux temps *kT* pour k=1,2,3,….(aux périodes d'échantillonnages). En fait, l'intégrale de e(t) ne peut être obtenue de façon exacte ; elle ne peut qu'être approximée. Les trois approximations les plus populaires de cette intégrale sont l'approximation rectangulaire, l'approximation rectangulaire devancée et l'approximation trapézoïdale. [11]

II.2.1.1 approximation rectangulaire

Telle que décrite par la figure II.2, l'approximation rectangulaire de l'intégrale est donnée par la relation suivante :

$$\int_{(k-1)T}^{kT} e(t)dt \sim \mathbf{Te((K-1)T)} \qquad \mathbf{(II.4)}$$

Ainsi, en remplaçant la relation (II.4) dans la relation, on obtient la relation entre l'entrée et la sortie du système :

$$\mathbf{e_i(kT)=e_i((k-1)T)+Te((k-1)T)}$$

(II.5)

En appliquant la transformée en Z de part et d'autre de la relation tout en considérant des conditions initiales nulles, la propriété du retard nous permet d'obtenir la relation suivante :

$$\mathbf{E_i(z)=z^{-1}E_i(z)+Tz^{-1}E(z)}$$

(II.6)

La fonction de transfert entre la sortie du compensateur et l'erreur est donc donnée par relation suivante :

$$\frac{Ei(z)}{E(z)} = \frac{T}{z-1} \qquad \textbf{(II.7)}$$

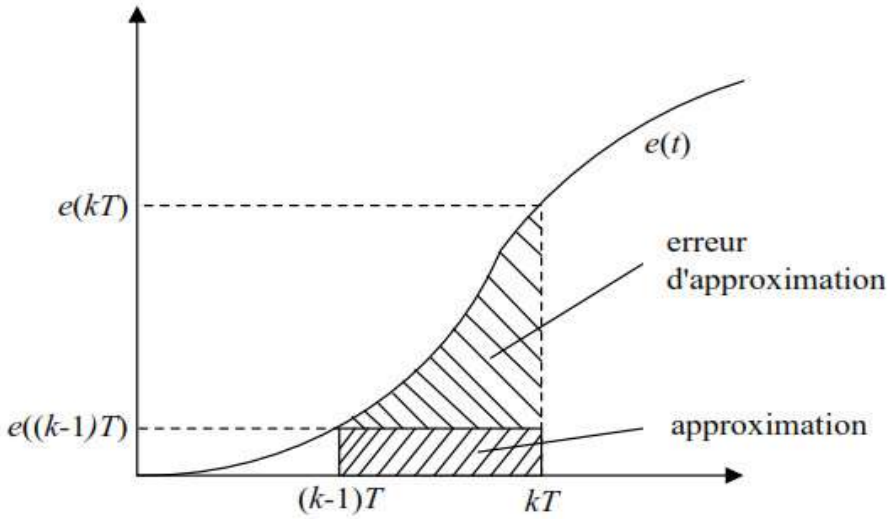


Figure II.2. *Approximation rectangulaire.*[11]

II.2.1.2 approximation rectangulaire devancée

Telle que décrite par la figure II.3, l'approximation rectangulaire devancée de l'intégrale est donnée par la relation suivante

$$\int_{(k-1)T}^{kT} e(t)dt = Te(kT) \qquad \textbf{(II.8)}$$

Ainsi, en remplaçant la relation (II.8) dans la relation (II.3), on obtient la relation entre l'entrée et la sortie du système :

$$\mathbf{e_i(kT)=e_i((k-1)T)+Te(kT)} \qquad \textbf{(II.9)}$$

En appliquant la transformée en Z de part et l'autre de la relation tout en considérant des conditions initiales nulles, la propriété du retard nous permet d'obtenir la relation suivante :

$$\mathbf{E_i(z)= z^{-1}E_i(z)+TE(z)} \qquad \textbf{(II.10)}$$

La fonction de transfert entre la sortie du compensateur et l'erreur est donc donnée par la relation suivante :

$$\frac{Ei(z)}{E(z)} = \frac{Tz}{z-1} \qquad \textbf{(II.11)}$$

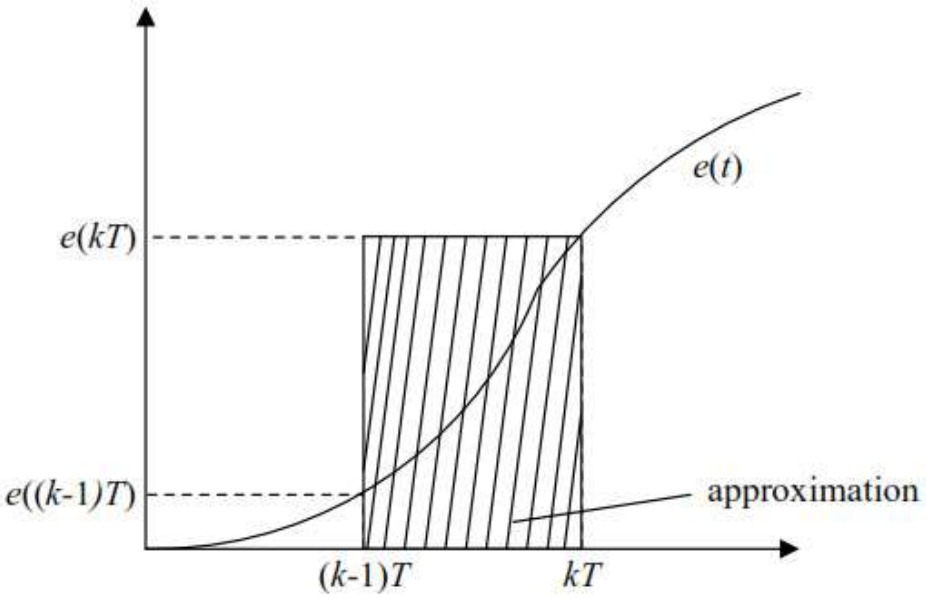


Figure II.3. *Approximation rectangulaire devancée.* [11]

II.2.1.3 approximation trapézoïdale

Telle que décrite par la figure II.4, l'approximation trapézoïdale de l'intégrale est donnée par la relation suivante :

$$\int_{(k-1)T}^{kT} e(t)dt = \frac{T}{2}(e((k-1)T)+e(kT)) \quad \textbf{(II.12)}$$

Ainsi, en remplaçant la relation (II.12) dans la relation(II.3), on obtient la relation entre l'entrée et la sortie du système :

$$e_i(kT)=e_i((k-1)T)+\frac{T}{2}(e((k-1)T)+e(kT)) \quad \textbf{(II.13)}$$

En appliquant la transformée en z de part et l'autre de la relation tout en considérant des conditions initiales nulles, la propriété du retard nous permet d'obtenir la relation suivante :

$$E_i(z)= z^{-1}E_i(z)+\frac{T}{2}(z^{-1}E(z)+E(z)) \quad \textbf{(II.14)}$$

La fonction de transfert entre la sortie du compensateur et l'erreur est donc donnée par la relation suivante :

$$\frac{Ei(z)}{E(z)} = \frac{Tz+1}{2\,z-1} \qquad \textbf{(II.15)}$$

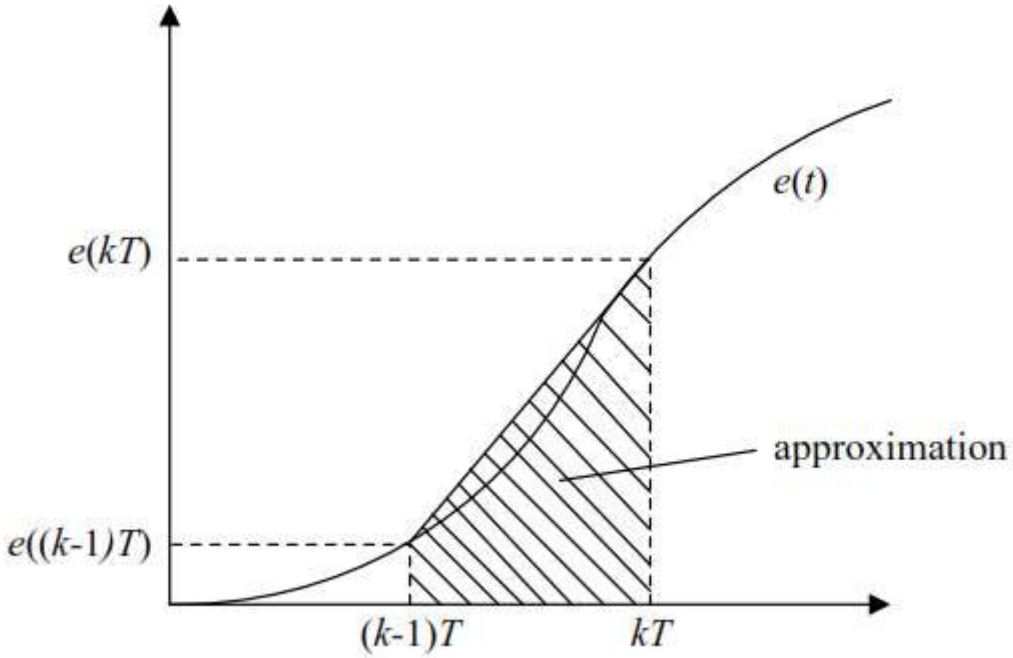


Figure II.4. *Approximation trapézoïdale.* [11]

II.2.1.4 fonction de transfert du PI avec l'approximation trapézoïdale

Maintenant que nous avons trouvé les fonctions de transfert qui approxime l'intégrale dans le domaine discret, il suffit de remplacer une de ces approximations dans le schéma bloc du compensateur PI de la figure II.1. Si on considère l'approximation trapézoïdale, qui est visiblement la plus précise, le schéma bloc du compensateur PI dans le domaine de z devient celui de la figure II.5. [11]

La fonction de transfert du compensateur PI avec une approximation trapézoïdale de l'intégrale est donc donnée par la relation suivante :

$$\frac{M(z)}{E(z)} = Kp + Ki\frac{T}{2}\frac{z+1}{z-1} \qquad \textbf{(II.17)}$$

En plaçant les deux termes sous un même dénominateur, on obtient

$$\frac{M(z)}{E(z)} = \frac{Q0Z+Q1}{z-1} \qquad \textbf{(II.18)}$$

Où

$$Q_0=K_p+K_i\frac{T}{2} \qquad Q_1=-K_p+K_i\frac{T}{2}$$

(II.19)

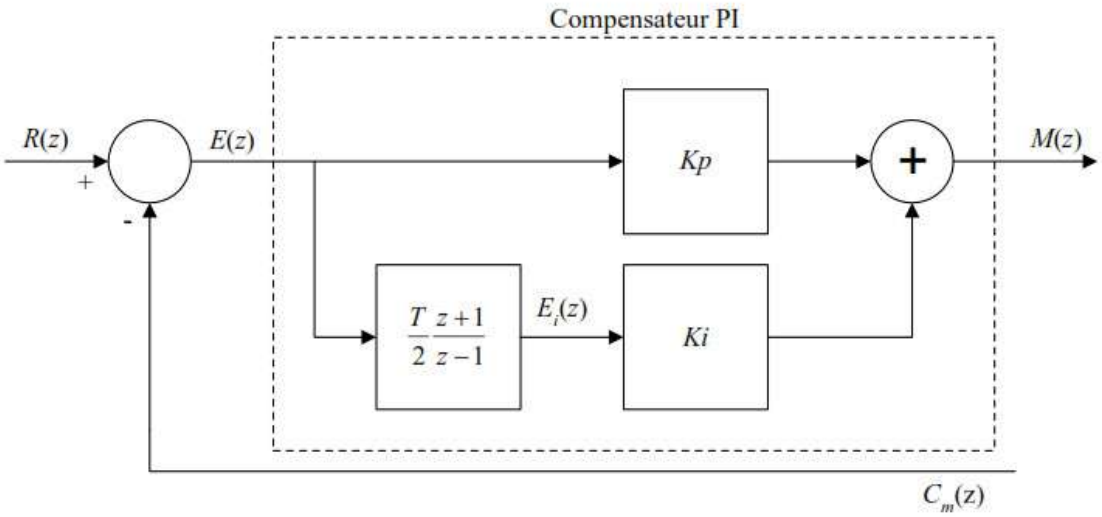


Figure II.5. *Schéma bloc en z du compensateur PI.* [11]

II.2.2 *Encodeur incrémental*

Certains signaux peuvent être convertis du domaine continu discret sans utiliser de convertisseur A/N. la position du rotor d'un moteur en est un exemple. En effet il existe plusieurs capteurs dédiés à mesurer la position du rotor d'un moteur directement dans le domaine numérique. Le moins dispendieux de ces capteurs est l'encodeur incrémental.

L'encodeur incrémental peut mesurer le déplacement angulaire du rotor d'un moteur électrique. Le principe de fonctionnement de ce capteur est illustré par la figure II.6 le disque en rotation est branché mécaniquement avec le rotor du moteur. Ainsi, parce que les disques renferment des régions opaques et des régions transparentes, selon la position du rotor, les faisceaux lumineux peuvent atteindre les senseurs photoélectriques. Les signaux fournis par ces senseurs sont par conséquents parfois à une valeur maximum lorsque la lumière passe et parfois à une valeur minimum lorsque la lumière est bloquée. Connaissant le nombre de régions transparentes sur les disques, le déplacement du rotor peut être obtenu en comptent le nombre de valeurs maximales du signal A ou du signal B. [4]

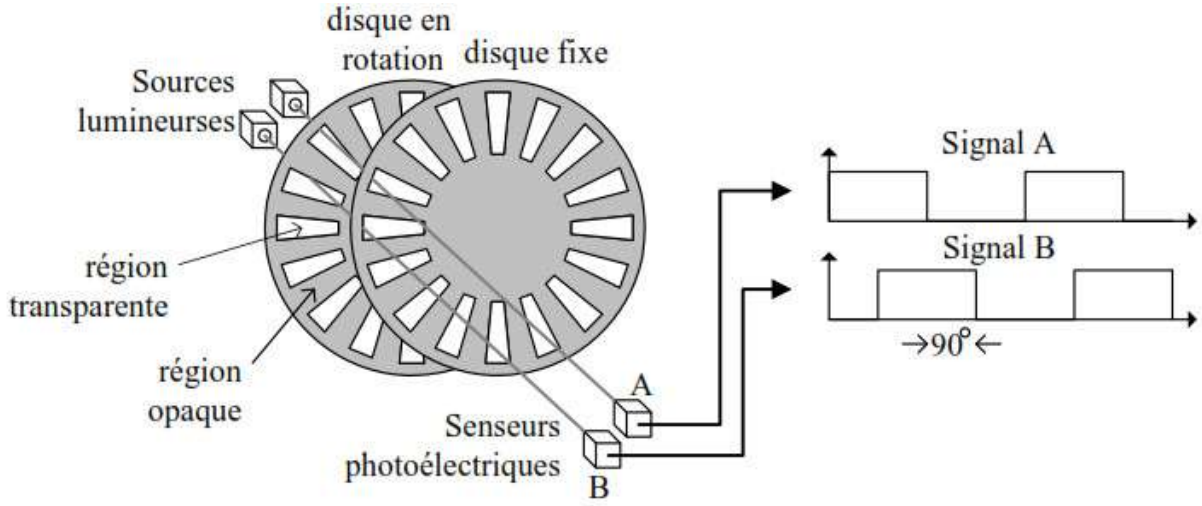


Figure II.6. *Encodeur incrémental.* [4]

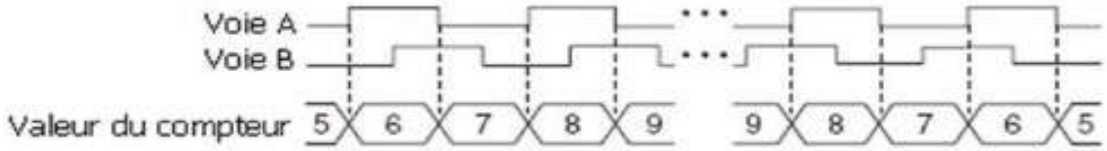


Figure II.7. *Encodage X2.* [3]

La figure II.7 représente un cycle de quadrature et les incréments et décréments résultants pour l'encodage X2 , [3]

La figure II.6 représente les sources lumineuses et senseurs A et B sont placés de façon à ce que les signaux provenant de ces capteurs soient en quadrature ; c'est-à-dire, déphasés de 90 degrés. Cette particularité nous permet de déterminer le sens, le signal B est en avance sur le signal A tandis que lorsqu'il tourne dans l'autre sens, B est en retard sur A. le décodage des signaux A et B peut donc se faire en utilisant un compteur branché sur le signal A ou B qui s'incrémente si B est en avance sur A et qui se décrémente si B est en retard sur A. pour augmenter la précision de la mesure par un facteur de quatre, il est également possible de compter tous les transitions des signaux A et B. [4]

Le gain de capteur est alors donné par :

$$K_E = 4\, N_t \, \frac{1\ tour}{2\,\pi\ rad} \qquad \textbf{(II.20)}$$

Où :

N_t : est le nombre total de régions transparentes sur les disques de l'encodeur incrémental.

Pour les positions rotatives :

$$\theta = \frac{nombre\ de\ fronts}{x\ N} * 360 \qquad \textbf{(II.21)}$$

Où : N = nombre d'impulsions générées par l'encodeur à chaque révolution de l'axe

x = type d'encodage

Le désavantage de l'encodeur incrémentale est que la position mesurée est toujours relative à la position du rotor du moteur au moment ou le système a subi une remise à zéro. Pour contourner ce problème, il est fréquent que l'encodeur incrémentale possède un troisième signal produisant une impulsion une seule fois par tour. Ce signal peut alors nous permettre de prévoir un mécanisme d'initialisation permettant de placer le rotor à une position connue lors de la remise à zéro. Il existe également d'autre type d'encodeur permettant de fournir directement la position du rotor du moteur et non son déplacement. Ces encodeurs beaucoup plus dispendieux sont appelés encodeurs absolus. [4]

II.3 LA PARTIE DE SIMULATION

Dans ce chapitre on a modélisé la machine a courant continu a travers d'un régulateur proportionnel intégral sous MATLAB /SIMULINK.

On a pour la machine à courant continu :

• Equations électriques :

Prenant en compte la résistance Ra et l'inductance La du circuit d'induit, du collecteur, des balais et des connexions, et en les supposant toutes deux constantes (pas de variation due à l'échauffement ni à la saturation magnétique), l'équation de tension induite

S'écrit :

$$u_a(t) = R_a.i_a(t) + \frac{d\Psi}{dt} = R_a.i_a(t) + \frac{d(N.\phi f)}{dt}$$

(II.21)

$$u_a(t) = R_a.i_a(t) + L_a\frac{dia}{dt} + e_m(t)$$

(II.22)

• Equations électromécaniques :

La tension induite em(t), appelée FEM ("force électromotrice" dans l'optique de l'exploitation en générateur) est proportionnelle à la vitesse angulaire w(t) et au flux inducteur Φf(t) :

$$em(t) = k.\phi f(t).\Omega(t) \quad \textbf{(II.23)}$$

K est une constante dépendant de la construction de la machine. La première équation montre que em(t) s'oppose à Ua(t), c'est à dire que le moteur réagit en créant une FEM

Le couple électromagnétique Tem(t) développé a pour expression :

$$Tem(t) = k.\phi f(t).ia(t) \quad \textbf{(II.24)}$$

• Equation mécanique :

Le moteur en rotation est décrit par l'équation (de la dynamique) d'équilibre suivante :

$$J_t \frac{d\omega}{dt} = T_{em}(t) - B\Omega(t) - T_r(t) - T(t) \quad \textbf{(II.25)}$$

Où

J : inertie totale entraînée ;

B: coefficient de frottement visqueux

Tr : couple résistant

T : couple de frottement de coulomb

Voilà en somme les équations régissant le fonctionnement du moteur. Mais pour mieux les exploiter, nous allons utiliser leurs transformées de Laplace.

Les transformées de Laplace des équations sont les suivantes :

$$u_a(t) = R_a.i_a(t)+L_a\frac{dia}{dt} +e_m(t) \rightarrow u_a-E_m = R_a.I_a+s.L_a.I_a$$

(II.26)

$$e_m(t)=k.\phi_f(t).\Omega(t)\rightarrow E_m =k.\ \phi\Omega$$

(II.27)

$$T_{em}(t)=k.\ \phi(t).i_a(t)\rightarrow T_{em} =k\ .\ \phi I_a$$

(II.28)

$$J_t\frac{d\omega}{dt} =T_{em}(t)-T_r(t)-B\Omega(t)-T\rightarrow T_{em}-T-B\Omega(t)-T=s.\ J_t.\Omega$$

(II.29)

Matlab est un logiciel de calcul matriciel à syntaxe simple. Avec ses fonctions spécialisées, Matlab peut être aussi considéré comme un langage de programmation adapté pour les problèmes scientifiques.

Matlab est un interpréteur: les instructions sont interprétées et exécutées ligne par ligne.

Matlab fonctionne dans plusieurs environnements tels que x-windows, windows, macintosh.

Il existe deux modes de fonctionnement :

- Mode interactif: Matlab exécute les instructions au fur et à mesure qu'elles sont données par l'usager
- Mode exécutif: Matlab exécute ligne par ligne un "fichier m" (programme en langage Matlab) .[5]

II.3.1 *Les schémas blocs*

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 considérées.

A notre travail, on considère la tension aux bornes de l'induit comme une entrée et la vitesse de moteur comme une sortie, nous obtenons le schéma bloc tension – vitesse. Ce dernier nous permet de voir l'évolution de la vitesse avec la tension.

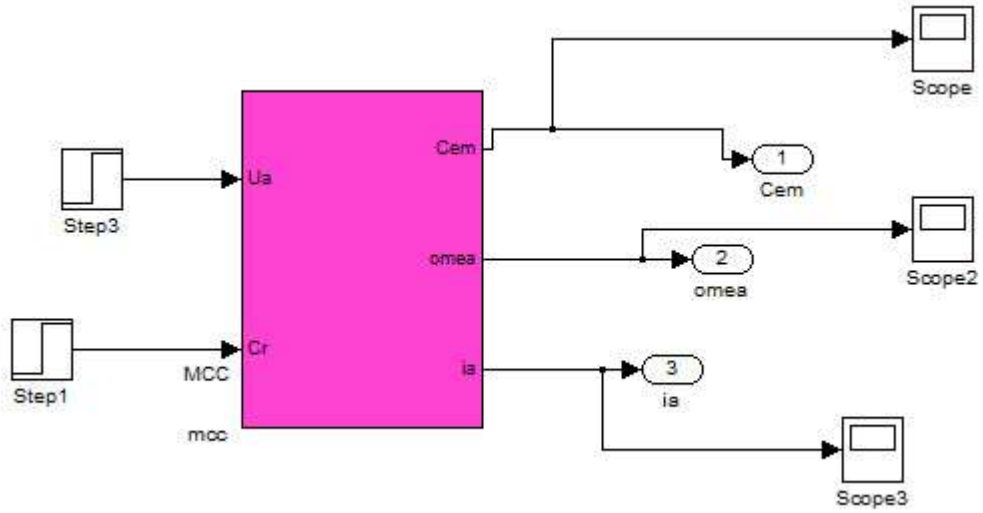


Figure II .8. *Schéma bloc du MCC en boucle ouverte en charge*

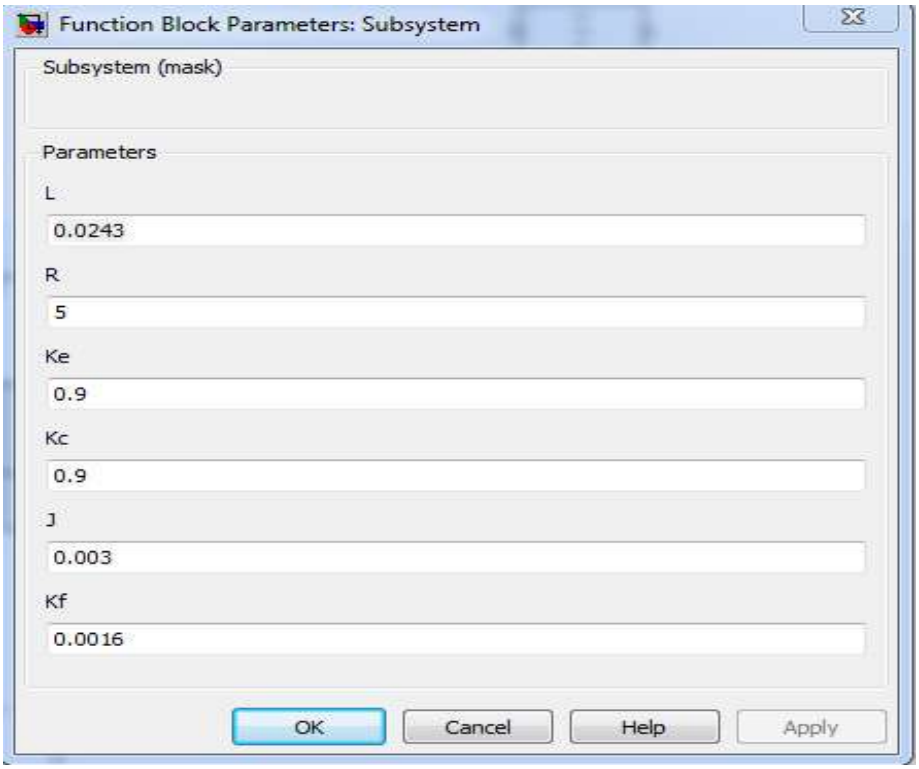


Figure II.9. *Fenêtre de réglage des paramètres de la machine à courant continu.*

II.3.2 *Résultats de simulation*

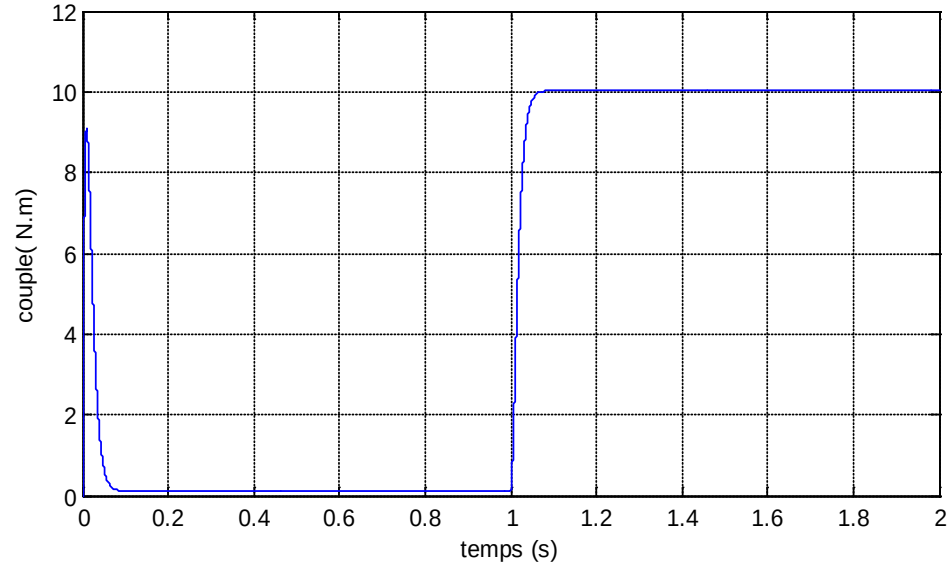


Figure II.10. *Le couple en fonction de temps*

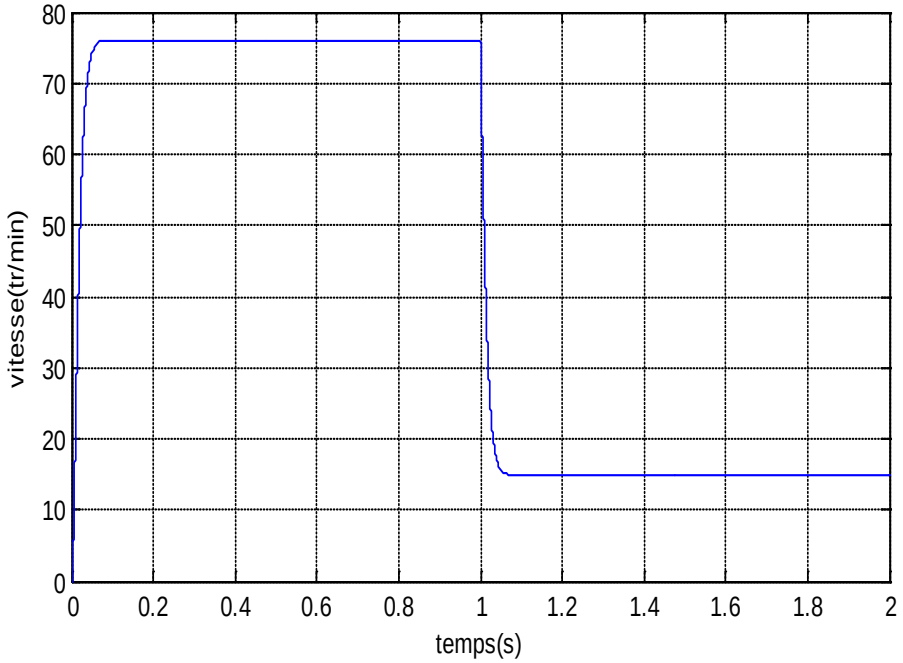


Figure. II.11. *la vitesse en fonction de temps*

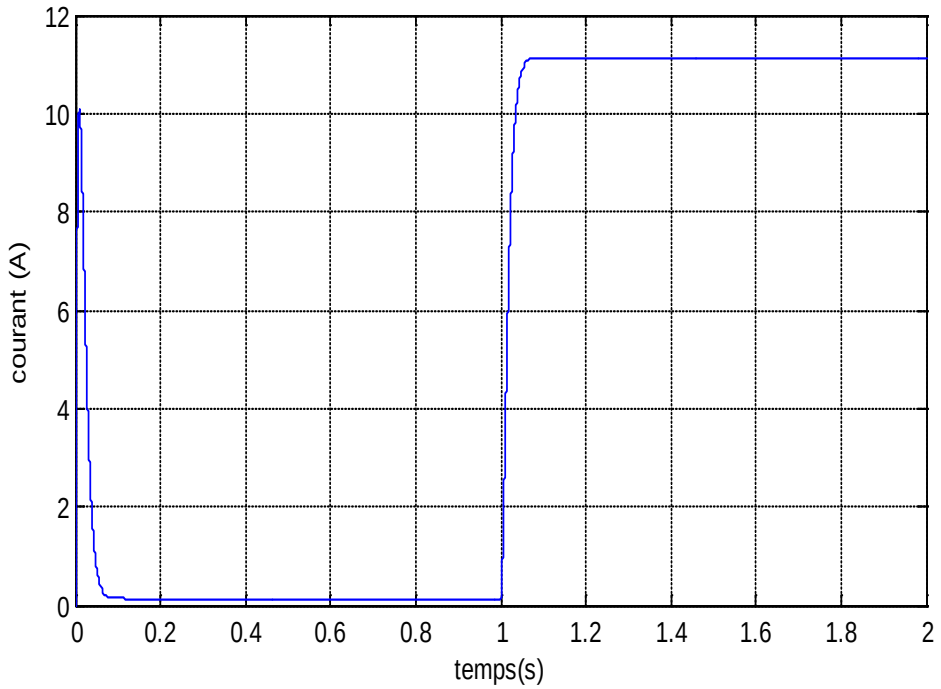


Figure II.12. *Le courant en fonction de temps*

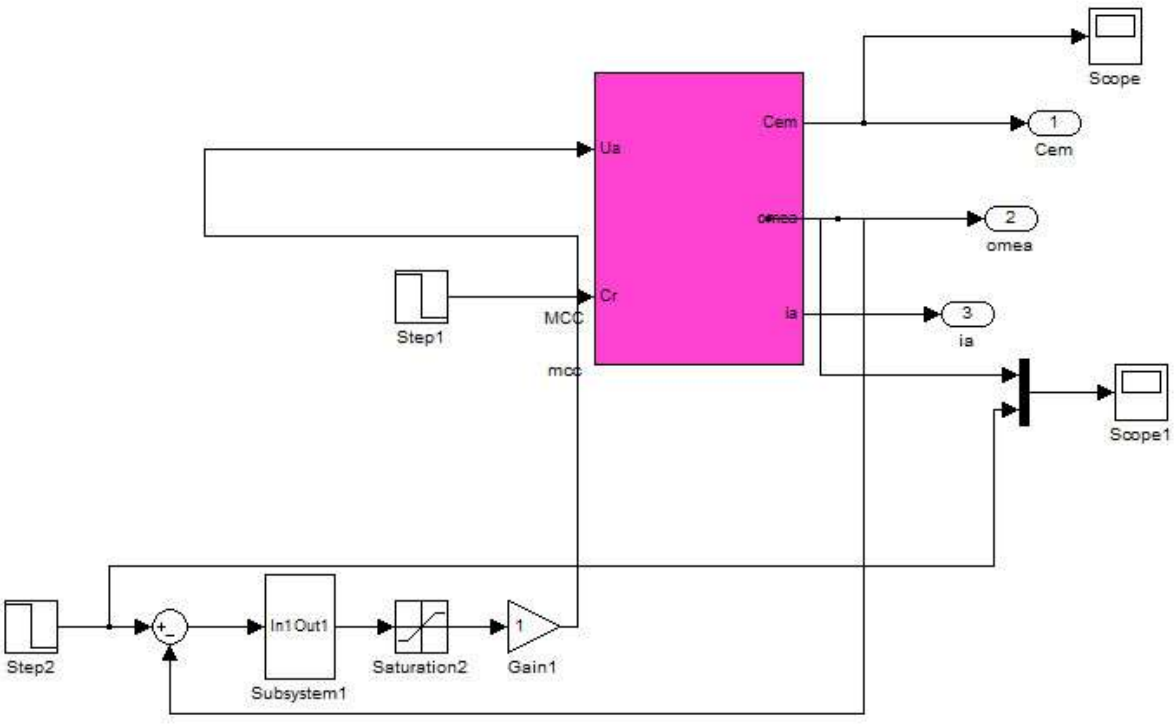


Figure II.13. *Schéma bloc du MCC en boucle fermé en charge*

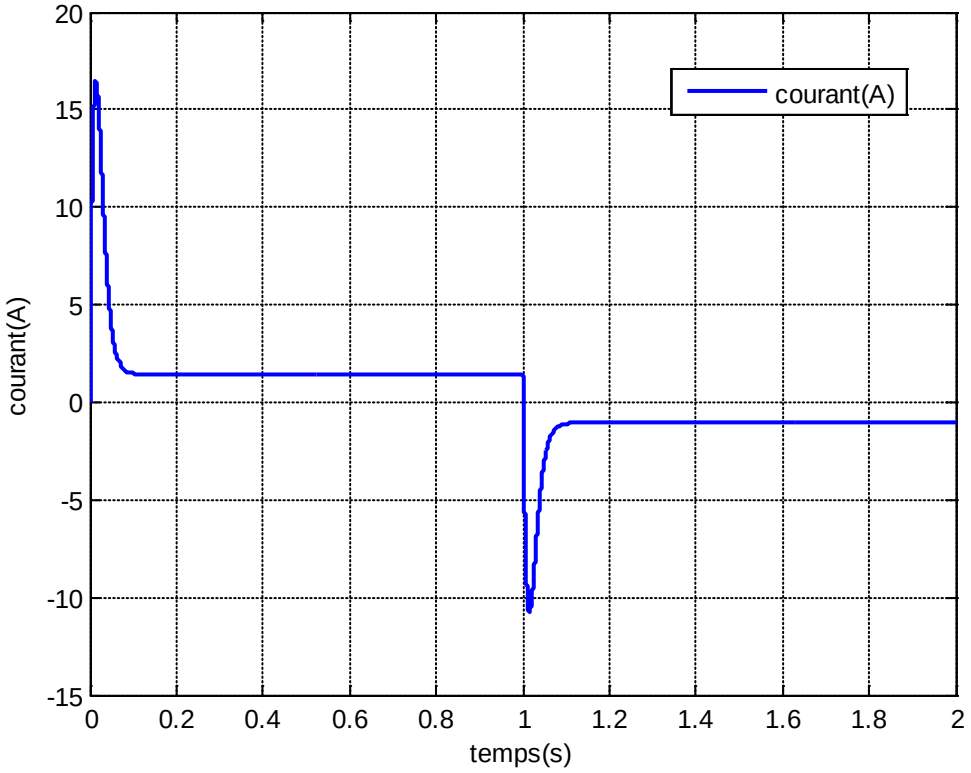


Figure II.14. *Le courant en fonction de temps*

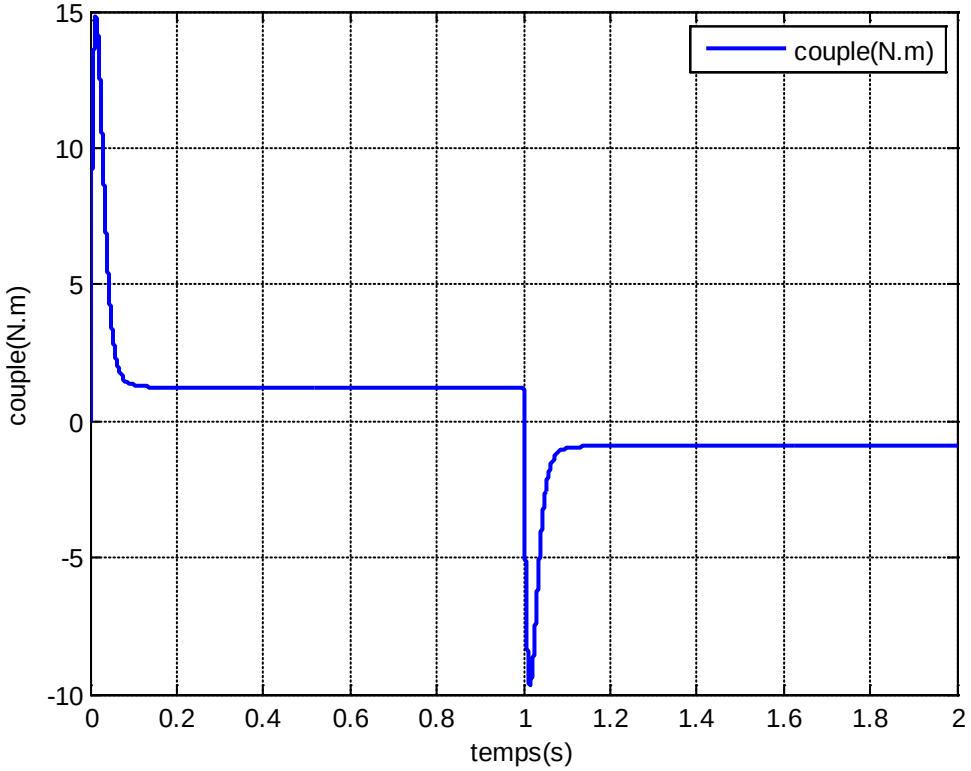


Figure. II.15. *le couple en fonction de temps*

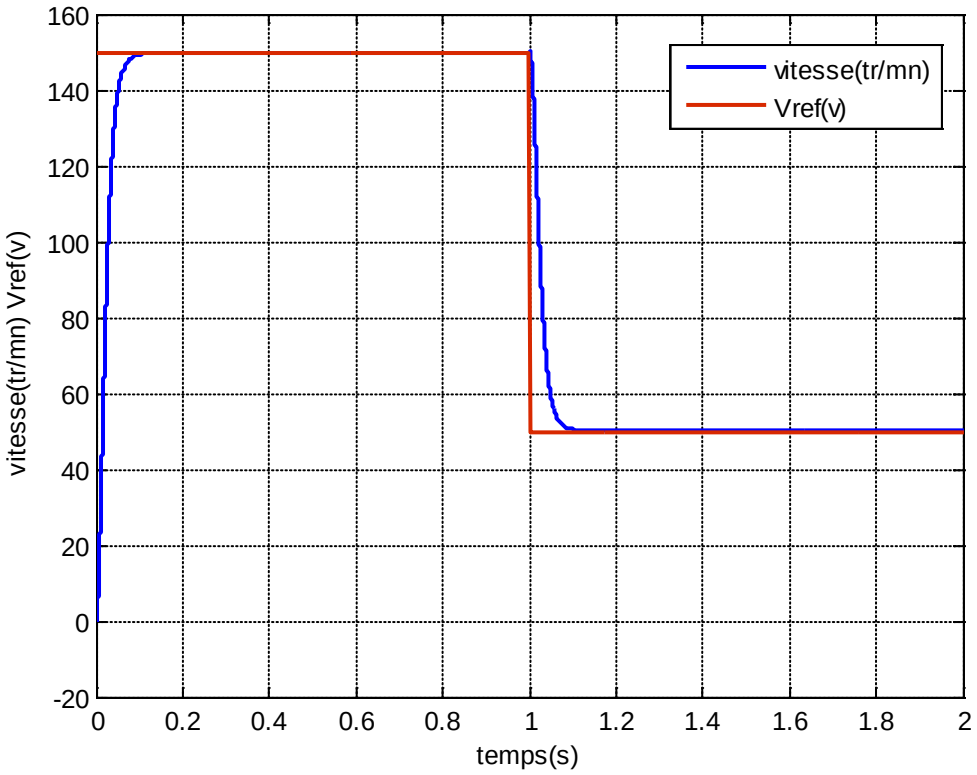


Figure. II.16. *la vitesse et la consigne en fonction de temps*

II.3.3 *Discussion*

On constate que la vitesse suit bien sa référence imposé (Ωref= 150rd/s entre l'instant (t=0 jusqu'à t=1s) et puis Ωref= 50 rd/s entre l'instant(t=1 jusqu'à t=2s) , cela confirme les performances obtenue par le régulateur PI.

Les coefficients Ki, Kp sont déterminer par la méthode d'emplacement de pôles

$$\begin{cases} Kp = 2\zeta J - Kf = 0.2684 \\ Ki = 2\zeta^2 J = 12.5 \end{cases}$$

II.4 CONCLUSION

Ce chapitre contient deux parties, dans la première partie on a fait une étude théorique sur le régulateur PI et sur l'encodeur incrémental, dans la deuxième partie on a fait une régulation analogique avec le compensateur PI dont l'outil de simulation est le logiciel MATLAB simulink.

Chapitre III :

Microcontrôleur PIC 16F877A

III.1 INTRODUCTION

L'objectif principal de ce chapitre est la présentation du cœur de notre application à savoir le microcontrôleur PIC16F877A.

L'architecture externe (brochage), l'architecture interne (registre interne) , configuration des registres et les interruption sont autant de paramètres auxquels nous y aborderons dans ce chapitre.

Un microcontrôleur est un circuit intégré qui rassemble les éléments essentiels d'un ordinateur : processeur, mémoires « mémoire morte pour le programme, mémoire vive pour les données », unités périphériques et interfaces d'entrées-sorties. Les microcontrôleurs se caractérisent par un plus haut degré d'intégration, une plus faible consommation électrique « quelques milliwatts en fonctionnement, quelques nano watts en veille », une vitesse de fonctionnement plus faible « quelques mégahertz » et un coût réduit par rapport aux microprocesseurs polyvalents utilisés dans les ordinateurs personnels.

Par rapport à des systèmes électroniques à base de microprocesseurs et autres composants séparés, les microcontrôleurs permettent de diminuer la taille, la consommation électrique et le coût des produits. Ils ont ainsi permis de démocratiser l'utilisation de l'informatique dans le domaine d'industrie pour la commande et contrôle des systèmes.

Les microcontrôleurs sont fréquemment utilisés dans les systèmes embarqués, comme les contrôleurs des moteurs automobiles, les télécommandes, les appareils de bureau, l'électroménager, les jouets, la téléphonie mobile, etc.

Des nombreux fabricants se sont implantés sur le marché des microcontrôleurs, la société américaine Micro-chip à mis au point un microcontrôleur CMOS, « **C**omplementry **M**etal **O**xide **S**emi-conductor » appelé PIC ; cette dénomination est sous copyright de Micro-chip, donc les autres fabricants ont été dans l'impossibilité d'utiliser ce terme pour leur propre microcontrôleur. Actuellement ce microcontrôleur est très utilisé.

III.2 LES CARACTERISTIQUES DU PIC 16F877A

Nous avons utilisé le microcontrôleur **PIC 16F877A** pour contrôler la vitesse du moteur, ce µc est un circuit intégré à 40 broches dont les caractéristiques générales

peuvent être résumé comme suit

Vitesse d'exécution allant jusqu'à 20MHz permettant l'exécution rapide du programme, une instruction en cycle d'horloge de 200ns.

Une mémoire vive « **RAM** » de 368 octets, répartie sur 4 banque 0, 1, 2, 3 « mémoire de données ».

Une mémoire **EEPROM** pour sauver des paramètres de 256 octets « mémoire de données ».

Une mémoire morte « **ROM** » mémoire programme de type FLASH de 8ko « 1mot=14bit ».

30 Entrées/Sorties programmables multiplexés.

8 entrées « canaux » du module de conversion analogique numérique 10 bits.

2 Comparateurs analogiques programmables avec une référence de tension, programmable également.

Conservation des informations en mémoire vive jusqu'à 1.5v.

Une plage d'alimentation de 2.0V à 5V.

Module de capture **PWM** programmable permettant de capturer un signal « logique » en fonction du temps, ou d'offrir un signal **PWM : M**odule de **L**argeur d'**I**mpulsion.

De nouvelles fonctionnalités, comme les gestions de ports «Série».

Le PIC 16F877A parmi les plus puissants microcontrôleurs de la gamme « Mid-range » chez Micro-chip. Dans notre travail, on s'intéresse au convertisseur analogique/numérique au Timer1 pour la commande du moteur à courant continu. [6]

III.3. DESCRIPTION GENERALE DU PIC 16F877A

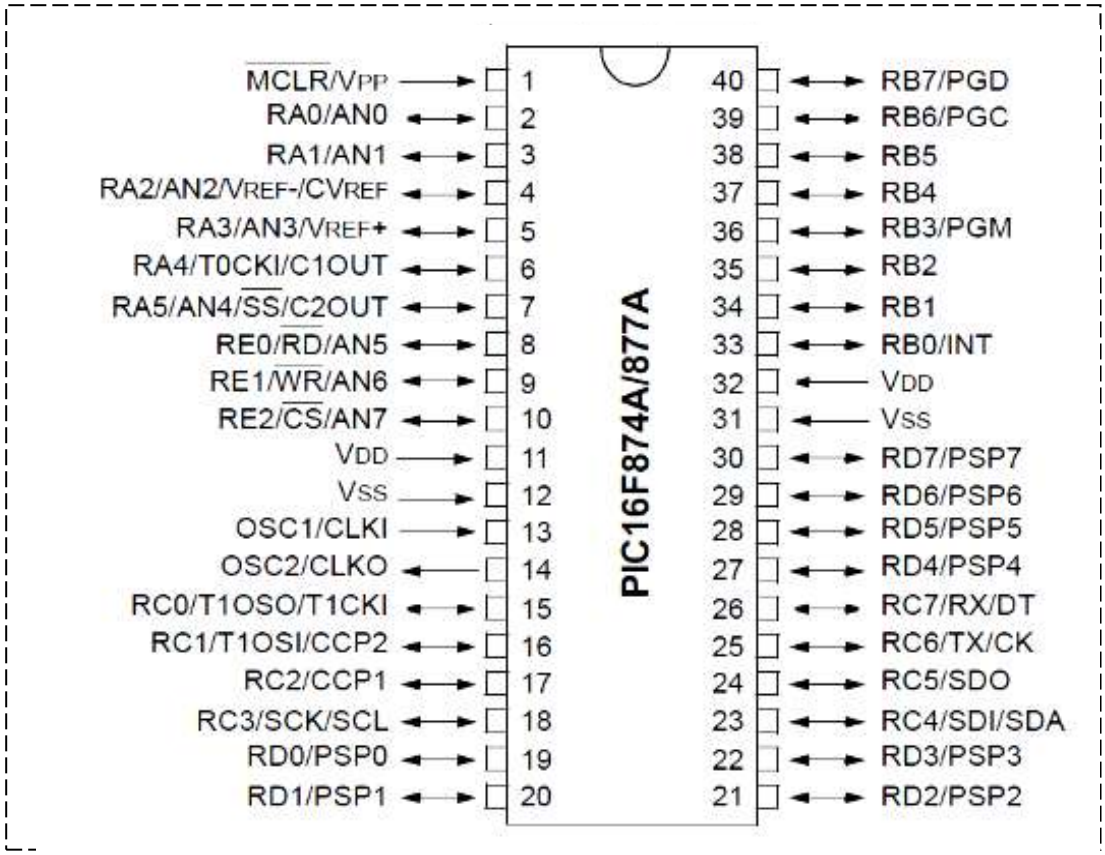


Figure III.1 : *boitier du PIC 16F877A* [6]

III.3.1. Architecture externe

Le microcontrôleur **16F877A** fait partie de la sous famille des **16F87X.** Cette branche fait partie intégrant de la famille des **Pics** Mid-Range, Le **PIC 16F877A** se présente sous la forme d'un boitier **PDIP 40** broches, Le numéro peut être suivi d'un «**A**», et d'un «**-XX**» qui Donne la fréquence d'horloge maximum du composant. A l'heur actuel, la version la plus courante est la version **-20.** Donc la fréquence maximale sera de **20MHz.** [6]

PIN	UTILISATION
RA0 à RA5	Lignes d'entrées /sorties du port A
RB0 à RB7	Lignes d'entrées /sorties du port B
RC0 à RC7	Lignes d'entrées /sorties du port C
RD0 à RD7	Lignes d'entrées /sorties du port D
RE0 à RE3	Lignes d'entrées /sorties du port E
VDD	Broche d'alimentation +5V

VSS	Broche du 0V
MCLR	Reset du circuit
OSC1 et OSC2	Broche recevant quartz externe.

Tableau III.1. *Broches du PIC.*

a. Alimentation « VDD ET VSS »

Ce sont les broches d'alimentation du circuit, les tensions qui peuvent être appliqués sont :

De **4V** à **6V** pour la gamme standard.

De **2V** à **6V** pour la gamme étendue.

L'intensité du courant consommé peut aller de **1µA** à **10mA.**

La consommation sera en fonction de :

La tension d'alimentation.

La fréquence interne.

Le mode de fonctionnement.

b. L'horloge

Deux pates **11** et **12** sont présentées sur le boitier du **PIC 16F877A.** La première nommée **OSC1/CLKIN** et la deuxième **OSC/CLKOUT.** Ces deux broches peuvent être utilisées de plusieurs manières; cela dépend de l'oscillateur ou de l'horloge employés:

XT : version à oscillateur à quartz «standard» fonctionnant jusqu'à une fréquence maximale de **4MHz**

HS : High Speed, version à oscillateur à quartz capable de fonctionner jusqu'à une fréquence maximale de 20**MHz.**

RC : capable de fonctionner jusqu'à **4MHz** mais avec une stabilité de fréquence plus faible que les versions à quartz.

LP « **Low Power** » : version à quartz également, mais prévue spécialement pour les applications à très faible consommation. C'est donc une version dont la fréquence maximale de fonctionnement est limitée à **20KHz.**

c. MCLR

Cette broche sert à initialiser le **PIC,** ce dernier dispose de plusieurs sources de **RESET. POR « Power on reset » :** mise sous tension. Un front descendant sur **MCLR** déclenche l'initialisation du µc**.** Cette bronche peut être simplement reliée à **VDD** si on n

a pas besoin de **RESET** externe.

EXTERNAL RESET « mis à l'état de **MCLR** ».Remise à zéro extérieur. Il faut appliquer un niveau bas sur l'entrée.

RESET au moins **2µs** pour que l'initialisation soit prise en compte.

WDT : chien de garde. Si le **WDT** arrive à la fin du temps de garde sans être rafraîchi il aura une initialisation du µc

BOR : baisse de l'alimentation. Si l'alimentation VDD chute en dessous de **4V** pendant **100µs** au moins, le microcontrôleur peut générer un RESET**.**
[6]

III.3.2. Architecture interne

Cette architecture nous présente les principaux organes qui composent le µc, la figure (III.2) illustre cette architecture **:**

La mémoire **FLASH** de **8K** mots à **14** bits.

La mémoire de donnée **EEPROM** à **256** octets.

La mémoire de donnée **RAM** à **368** octets.

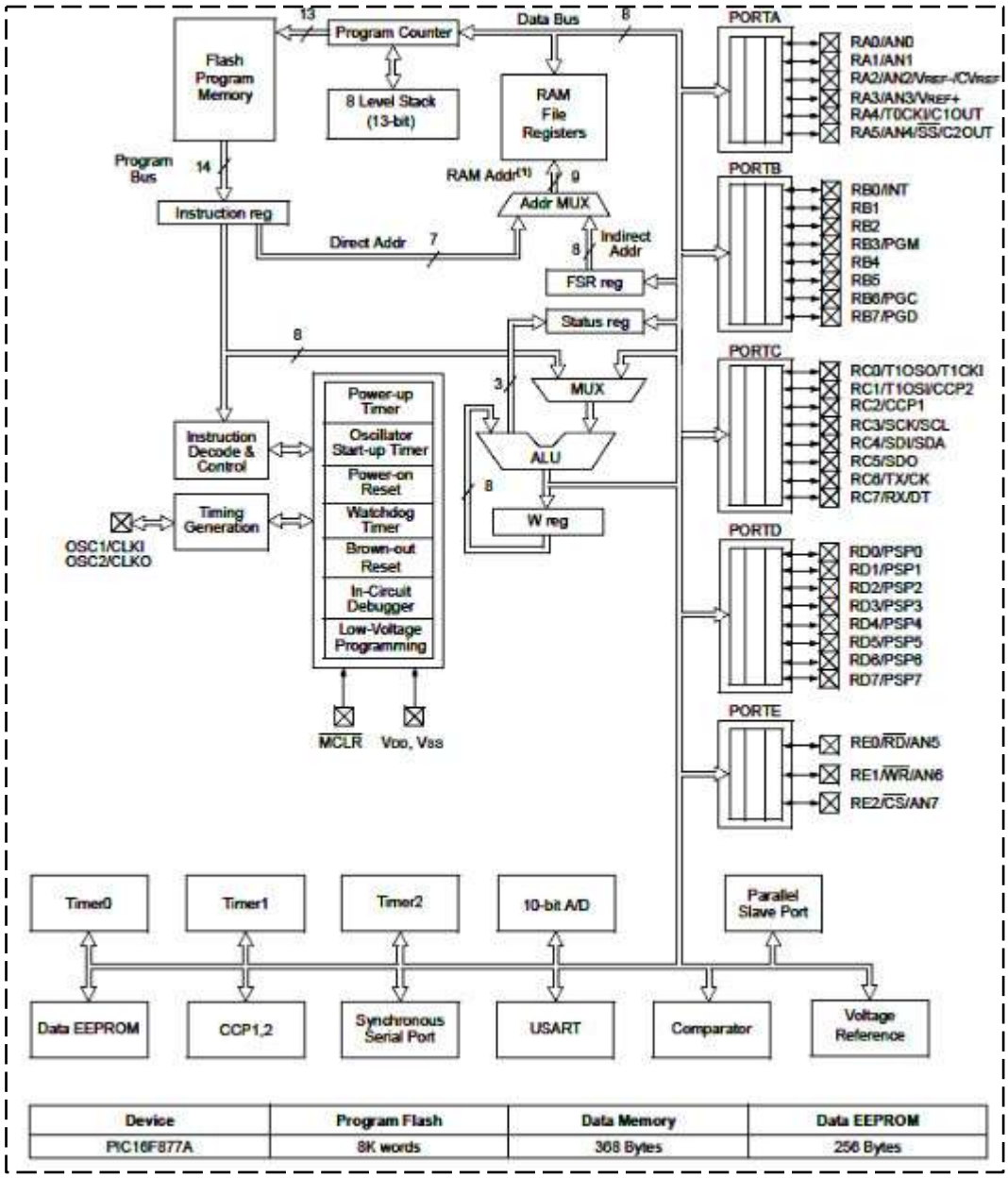
Les registres particuliers **W** « registre de travail »; **FSR** et **STATUS** « d'état ».

L'unité arithmétique et logique « **UAL ».**

Quatre ports bidirectionnels **E/S A, B, C, D.**

Huit niveaux de pile. [2]

Cependant, même si les op-codes sont codés sur **14** bits, les données et les résultats sont seront toujours sur **8** Bits.



Device	Program Flash	Data Memory	Data EEPROM
PIC16F877A	8K words	368 Bytes	256 Bytes

Figure III.2. *Architecture interne du PIC 16F877A*

III.3.3. Organisation de la mémoire

La mémoire du **PIC 16F877**A est divisée en 3 parties :

1. La mémoire programme

La mémoire programme est constituée de 8k mots de 14 bits. C'est dans cette zone que nous allons écrire notre programme. Le PIC exécute une à une les instructions logées dans la mémoire de programme.

2. La mémoire EEPROM

La mémoire EEPROM « Electrical Ecrasable Programmable Read Onlay Memory », est constituée de 256 octets, ces octets sont conservés après une coupure de courant et sont très utiles pour conserver des paramètres semi-permanents.

3. La mémoire RAM et organisation

La mémoire RAM est celle qui est sauvant utilisée. Toutes les données qui y sont stockées sont perdues lors d'une coupure de courant. La mémoire RAM disponible du 16F877 est de 368 octets. Elle est répartie de la manière suivante :

80 octets en banque 0, adresses 0*20 à 0*6F
80 octets en banque 1, adresses 0*A0 à 0*EF
96 octets en banque 2, adresses 0*110 à 0*16F
96 octets en banque 3, adresses 0*190 à 0*1EF
16 octets commun aux 4 banques, soit 0*70 à 0*7F = 0*F0 à 0*FF.

a. Plan mémoire pour les instructions « code programme »

Le plan mémoire est linéaire les adresses vont de **0000h** à **1FFFh « 8k mots** de **14** bits **»,** Par page de **2k mots.**

Les **Pics** n'ont qu'un seul vecteur d'interruption en **0004h**. Lors d'une interruption, le sous programme associé devra déterminer quel périphérique a demandé en mémoire de donnée comme avec les microcontrôleurs classiques, mais dans la mémoire programme. Elles sont utilisées lors d'appels de sous programmes, on ne peut pas imbriquer plus de **8** sous programmes.

b. Plan mémoire pour les données et registres interne

« SFR : Spécial function register »

Le plan mémoire des données et des registres internes est découpé en **4** zones ou Bank de **128** octets, pour accéder à une zone il faut positionner les bits **RB0 « bit 5 »** et **RP1 « bit 6 »** du registre **STATUS.**

Les registres appelés **General purpose Register** ne sont ni plus ni moins que des cases mémoires pour stocker les données. [2]

III.3.4. Les registres du PIC 16F877A

La mémoire **RAM** est organisée en **4** banques pour le **16F8777A**. La **RAM** est subdivisée de plus en quatre parties. Dans chacune des banques on trouve des «cases mémoires spéciales» appelées **REGISTRES SPECIAUX** et des cases mémoires «libres» dont nous pouvons nous en servir à notre guise.

a. Le registre statu d'adresse 0X03 ou registre d'états

Comme son nom l'indique ce registre contient un certain nombre de bits d'états de l'unité centrale mais aussi des bits de sélection de la page qui s'appellent ici RP1 et RP0. De ce fait, il peut être lu comme n'importe quel autre registre étant entendu que certains bits d'états restant non modifiables.

b. Le registre FSR d'adresse 0X04

Le **PIC 16F877A** possède une unité arithmétique et logique **« UAL »** et un registre de travail **W**. L'**UAL** effectue des fonctions arithmétiques booléennes entre les données se trouvant dans le registre de travail et n'importe quel registre. Pour les instructions à deux opérandes, typiquement une donnée est dans le registre de travail « **W** » et l'autre opérande dans un autre registre ou se trouve une valeur immédiate constante. Pour les instructions à un seul opérande, la donnée se trouve soit dans le registre de travail « **W** » soit dans un autre registre.

Le registre de travail « **W** » possède **14** bits, utilisé pour les opérations avec l'**UAL,** ce n'est pas un registre adressable. Elle « UAL » est composée :

- D'un accumulateur **8** bits **W : WORKING «** travail », c'est lui qui effectue toutes les opérations arithmétiques et logiques.
- Un registre d'état **8** bits **STATUS.**

IRP	RP1	RP0	TO	PD	Z	DC	C
Bit7							**Bit0**

- **Bit 0** ou bit C **Carry** c'set le bit de retenue en addition ou soustraction sur un octet. Il positionné a 1 si une retenue set générée depuis le bit de poids fort
- **Bit 1** ou bit DC comme **Digit Carry0** il est positionné à 1 si une retenue est générée du bit 3 « Bit de poids fort du quartet inférieur » vers le bit 0 du quartet supérieur. Il set utile pour corriger le résultat d'opérations effectuées en code BCD
- **Bit 2** ou **Z** comme **Zéro**, ce bit est mis à 1 si le résultat de l'opération arithmétique ou logique est nul.
- **Bit 3** ou **PD** comme **Power Dowen**, ce bit à 1 lors de la mise sous tension du circuit ou lors de l'exécution d'une instruction **CLWDT** relative au timer chien de garde il est mis à 0 par l'instruction **SLEEP.**
- **Bit 4** ou bit **TO** comme **Time Out**, ce bit est à 1 lors de la mise sous tension du circuit ou lors de l'exécution d'une instruction **CLWDT** ou **SLEEP ,**il passe à 0 si le

timer chien de garde déborde.

- **Bit 5** ou bit **RP0 « Register Bank Select0 »,** ce bit sert à sélectionner l'une des deux pages de la mémoire « Page 0 ou Page 1 » sachant que chaque page mesure **128** octet.

RP0 = 0 sélectionne la page mémoire **0** « adresse de **00** à **7F** »

RP0 = 1 sélectionne la page mémoire **1** « adresse de **80** à **FF** »

- **Bit 6** ou bit **RP1 « Register Bank Select1 »,** permet la sélection des pages **2** et **3**.
- **Bit 7** ou bit **IRP** ce bit est prévu pour un future adressage paginé indirect.

[6]

III.3.5. Les timers

Les **Pics 16F877A** disposent **3** timers :

a. Le timer 0 « 8 bits »: il peut être incrémenté par des impulsions extérieures via la broche **« TOCKI/RA4 »** ou par l'horloge interne « **Fosc/4 ».**

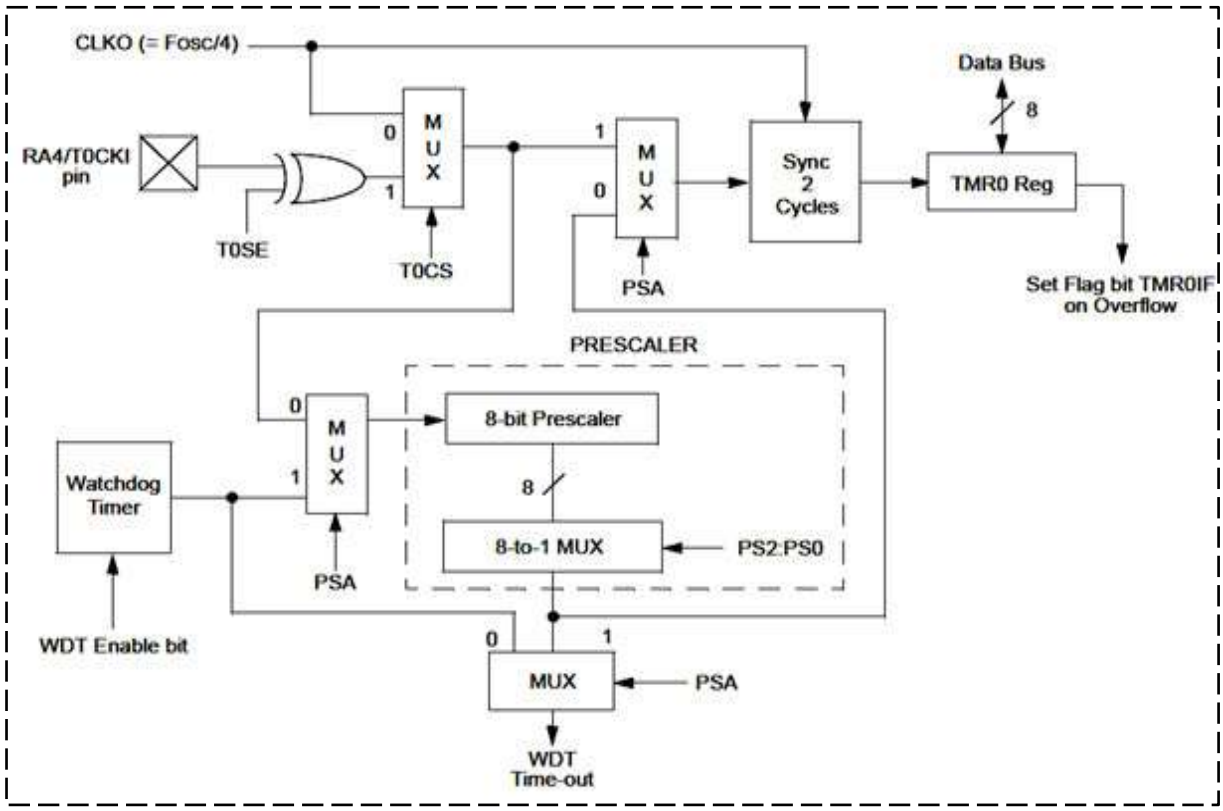


Figure III.3. *Schéma du timer 0.*[6]

b. Le timer 1 « 16 bits »: il peut être incrémenté soit par l'horloge interne, par des

impulsions sur les broches **TICKI/RC0** ou par un oscillateur **« RC ou quartz »** connecté sur les broches **TOSO/RC0** et **T1OS1**

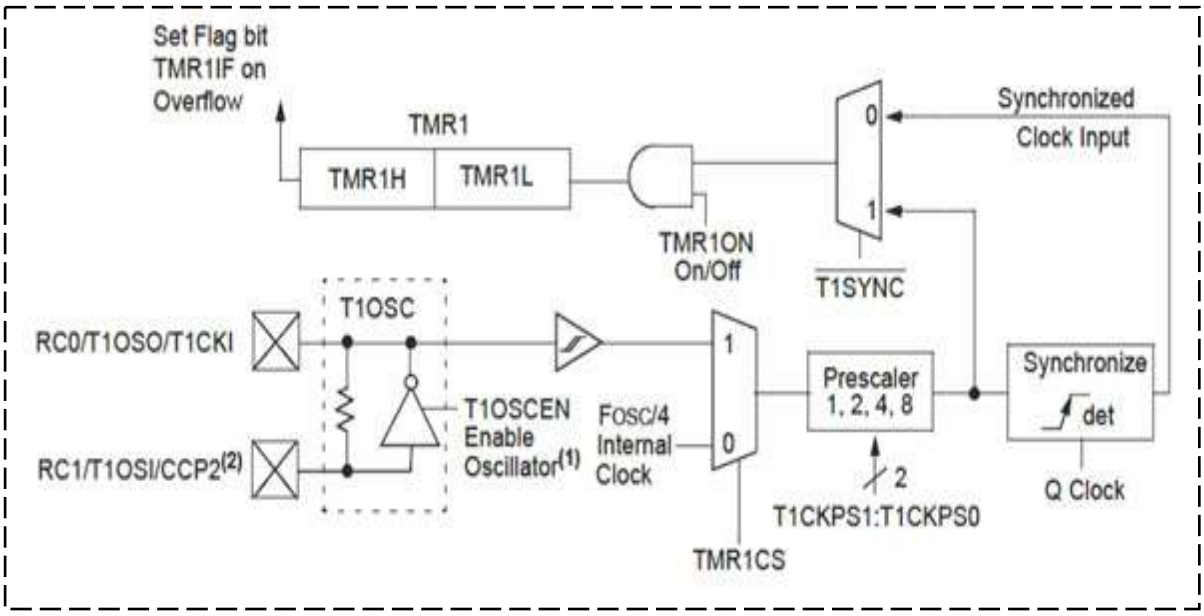


Figure III.4. *Schéma descriptive du timer 1* [2]

c. Le timer 2 « 8 bits »: il est incrémenté par l'horloge interne, celle peut être pré divisé.

Tous ces **timers** peuvent déclencher une interruption interne, s'ils ont été autorisés. [6]

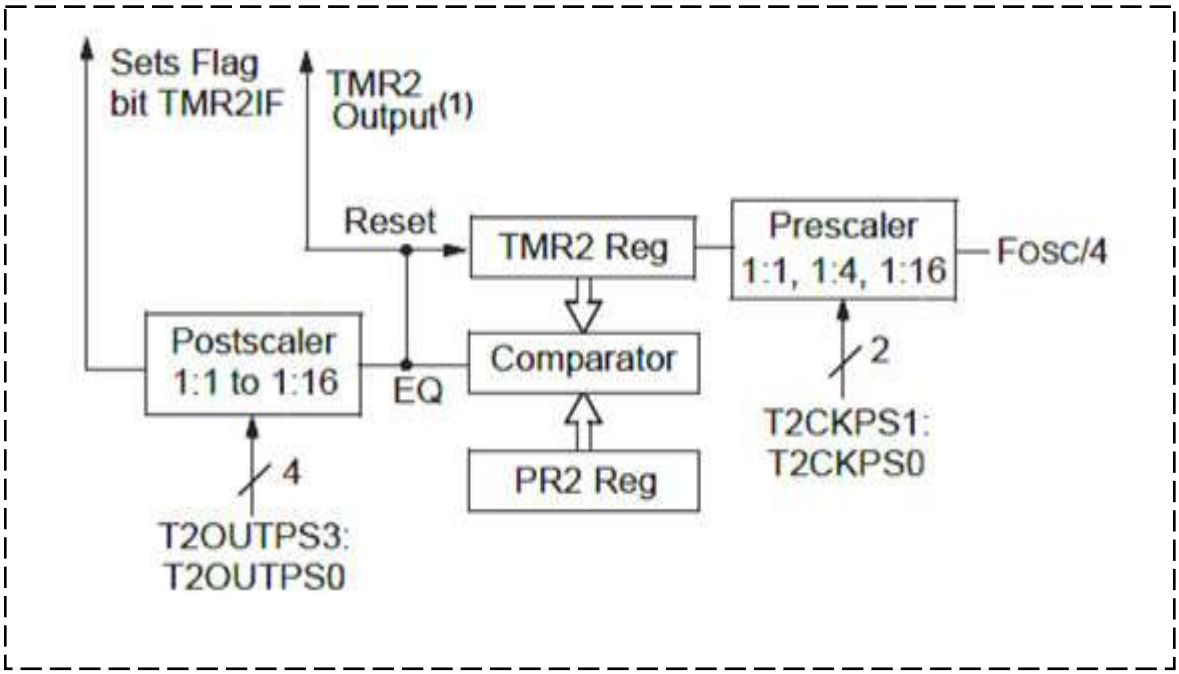


Figure III.5. *Schéma du timer 2.* [2]

III.3.6. Les interruptions

Le PIC 16F877A possède 15 sources d'interruptions. Les plus importantes à connaitre, car ce sont les plus utilisées en pratique, sont :

TMR0I : Dépassement du TIMER0 « débordement le passage de **255** à **0** »

TMR1I : Dépassement du TIMER1 « débordement le passage de **65535** à **0** »

TMR2I : Contenu du PR2 du TIMER2 « débordement le passage de la valeur qui contient dans le registre **PR2** à **0** »

INT : Interruption extérieure RB0

RBI : Changement d'état du port B « RB7 : RB4 »

PSPI : Lecture / Ecriture sur le port parallèle

ADI : Convertisseur Analogique / Numérique

TXI : Emission sur le port RS232

RCI : Réception sur le port RS232

SSPI : Bus SPI / I²C

CCP1I : Capture / Compare / PWM

CCP2I : Capture / Compare / PWM

EEI : Ecriture dans EEPROM

CMI : Comparateur

BCLI : Collision sur Bus [6]

III.3.7. Les ports d'entré sorties

Le microcontrôleur **16F877A** dispose **5 PORT « A, B, C, D, E ».** Tous ces ports sont bidirectionnels.

La plupart des lignes de **Ports** ont une double fonction.

Le **PORT A « 5 bits » I/O** pure et/ou convertisseur analogique et/ou **TIMER0.**

La broche **RA4** du **PORT A** « Entrée du **TIMER T0CKI »** est du type **DRAIN OUVERT**

Le **PORT B « 8 bits » I/O** pure et/ou programmation **ICSP/ICD** « broche **RB3/PGM, RB6/PGC et RB7/PGD** » et l'entrée d'interruption externe **RB0/INT.**

Le **PORT C « 8 bits » I/O** :C'est un port tout ce qu'il a de plus classique, or qu'il a Deux pins qu'on utilisera plus tard dans la communication série avec le

PC à travers **« TX** et **RX ».**

Le **PORT D « 8 bits » I/O** Le registre **TRISD** comportera donc les 8bits de direction, pendant que le registre port D correspond aux pins I/O concernés. D'où les 8 pins I/O, en mode entrée, sont du type « trigger de Schmitt ».

Le **PORT E « 3 bits » I/O** Ce port n'est présent que sur les PIC 16F877. Il ne comporte que 3 pins **RE0** à **RE2**, mais, contrairement aux ports, les bits non concernés de TRISE sont, cette fois, implantés pour d'autres fonctions. Les pins REX peuvent également être utilisés comme pins d'entrées analogiques. D'ou le registre ADCON1 qui détermine si ce port est utilisé comme port I/O ou comme port analogique. [6]

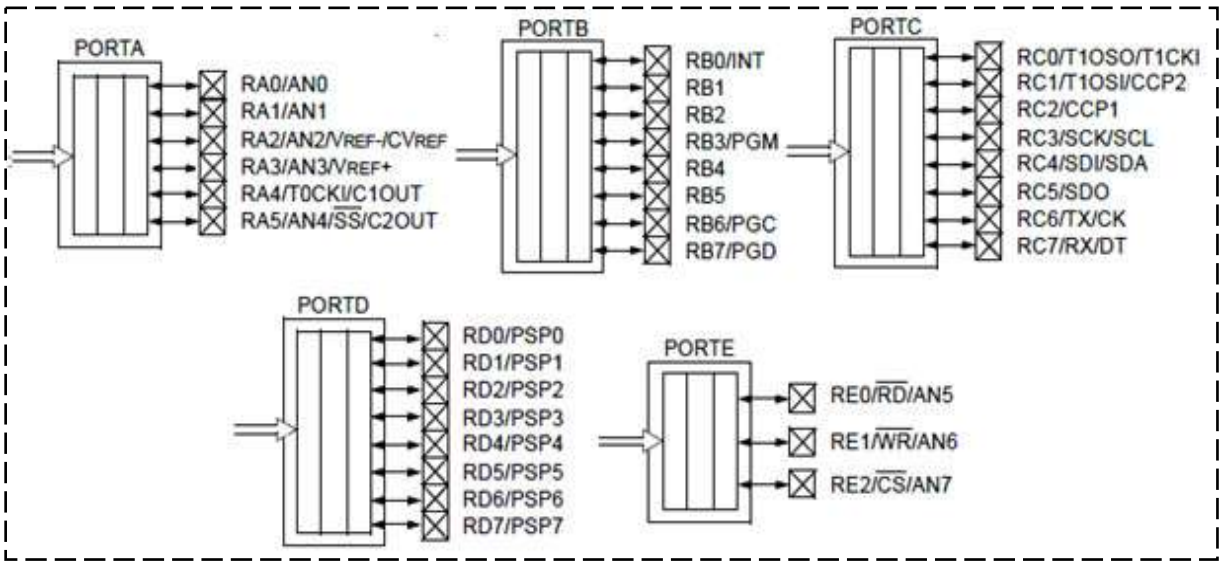


Figure III.6. *Les différents PORT de PIC 16F877A.* [6]

III.3.8. Configuration des portx, les registres PORTX et TRISX

Tous les ports sont pilotés par deux registres :

Le registre **PORTx,** si le **PORTx** ou certaines lignes de **PORTx** sont configurés en sortie, ce registre détermine l'état logique des sorties.

Le registre **TRISx**, c'est le registre de direction. Il détermine si le PORTx ou certaines lignes de port sont en entrée ou en sortie. L'écriture d'un 1 logique correspond à une **entrée** « **1** comme **I**nput » et l'écriture d'un **0** logique correspond à une sortie « **0** comme **O**utput ».

Au **RESET** toutes les lignes de ports sont configures en sorties.
[6]

III.3.9. Le convertisseur analogique numérique

Notre **16F877A** travaille avec un convertisseur analogique/numérique qui permet un échantillonnage sur 10 bits. Il est composé de :

Un multiplexeur analogique **5** voies **« PIC 16F877A ».**

Un échantillonneur bloqueur.

Un convertisseur analogique numérique de **10** bits.

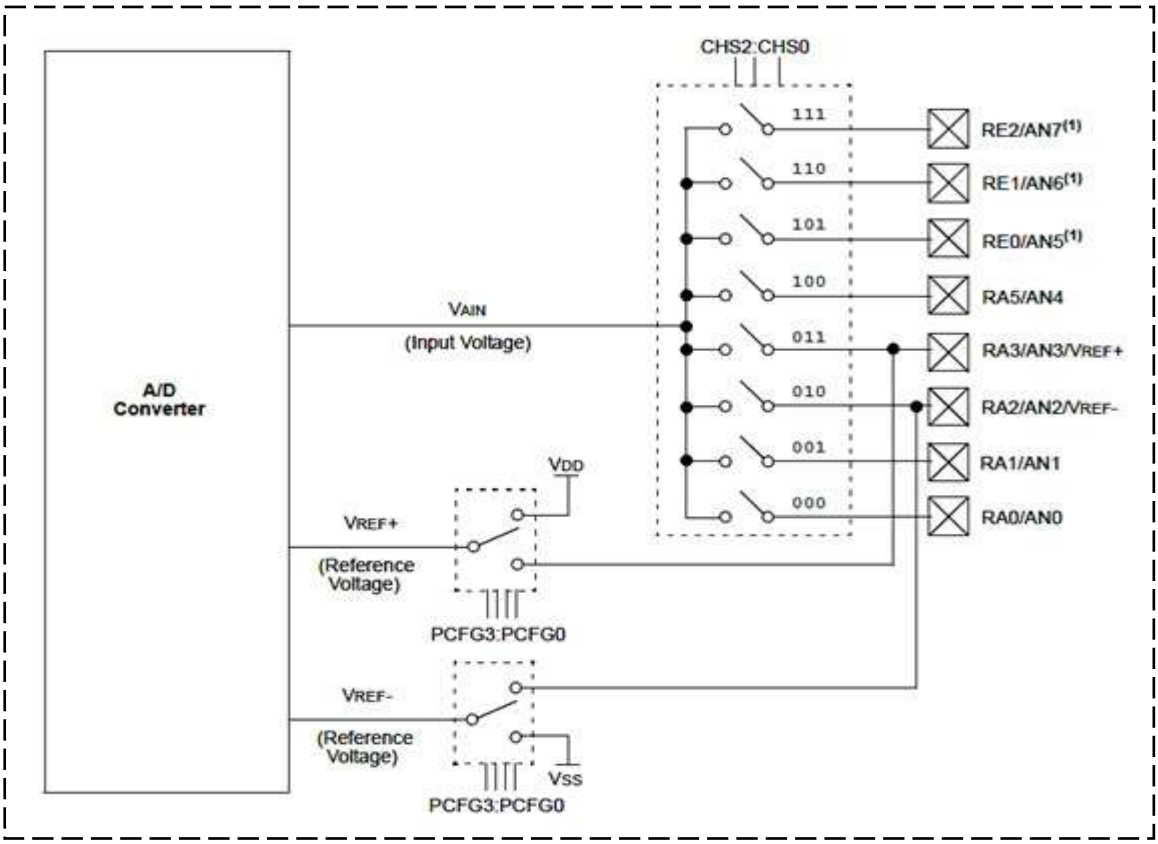


Figure III.7. *Schéma fonctionnel analogique numérique.* [6]

a. Les registres ADRESL et ADRESH

Il est important de savoir que le convertisseur donne un résultat sur 10 bits, et donc que ce résultat devra donc obligatoirement être sauvegardé dans 2 registres. Ces registres sont tout simplement les registres ADRESL et ADRESH. Comme 2 registres contiennent 16 bits, et que nous n'en utilisons que 10, Micro chip nous a laissé le choix sur la façon dont est sauvegardé le résultat, nous pouvons soit justifier le résultat à gauche, soit à droite.

La justification à droite complète la partie gauche du résultat par des « 0 ». Le résultat sera donc de la forme :

ADRESH										ADRESL					
0	**0**	**0**	**0**	**0**	**0**	**B9**	**B8**	**B7**	**B6**	**B5**	**B4**	**B3**	**B2**	**B1**	**B0**

La justification à gauche procède bien évidemment de la méthode inverse :

ADRESH										ADRESL					
B9	**B8**	**B7**	**B6**	**B5**	**B4**	**B3**	**B2**	**B1**	**B0**	**0**	**0**	**0**	**0**	**0**	**0**

b. Le registre ADCON1

Ce registre permet de déterminer le rôle de chacune des pins AN0 à AN7. Il permet donc de choisir si un pin sera utilisé comme entrée analogique, comme entrée/sortie standard, ou comme tension de référence. Il permet également de décider de la justification du résultat.

Pour pouvoir utiliser un pin en mode analogique, il faudra que ce pin soit configuré également en entrée par TRISA. Le registre ADCON1 dispose, comme tout registre accessible de notre PIC, de 8 bits, dont seulement 5 sont utilisés :

B7	ADFM	A/D result format select
B6	INUTILISE	lu comme « 0 »
B5	INUTILISE	lu comme « 0 »
B4	INUTILISE	lu comme « 0 »
B3	PCFG3	Port Configuration control bit 3
B2	PCFG2	Port Configuration control bit 2
B1	PCFG1	Port Configuration control bit 1
B0	PCFG0	Port Configuration control bit 0

Tableau III.2. *Bits de registre ADCON1*

c. Le registre ADCON0

Ce registre est le dernier utilisé par le convertisseur analogique/numérique. Il contient les bits que nous allons manipuler lors de notre conversion. Sur les 8 bits de notre registre, 7 seront utilisés. [2]

B7	ADCS1	A/D conversion Clock Select bit 1
B6	ADCS0	A/D conversion Clock Select bit 0
B5	CHS2	Analog Channel Select bit2

B4	CHS1	Analog Channel Select bit1
B3	CHS0	Analog Channel Select bit0
B2	GO/DONE	A/D conversion status bit
B1	Inutilisé	lu comme « 0 »
B0	ADON	A/D ON bit

Tableau III.3. *Bits de registre ADCON0*[2]

III.4. LES AVANTAGES DES MICROCONTROLEURS

L'utilisation des microcontrôleurs pour les circuits programmables à plusieurs points forts et bien réels. Il suffit pour s'en persuader, d'examiner la spectaculaire évolution de l'offre des fabricants de circuits intégrés en ce domaine depuis quelques années.
Nous allons voir que nombre d'entre eux découle du simple sens.

- Tout d'abord, un microcontrôleur intégre dans un seul et même boîtier ce qui, avant nécessitait une dizaine d'éléments séparés. Il résulte donc une diminution évidente de l'encombrement en matériel et du circuit imprimé.
- Cette intégration a aussi comme conséquence immédiate de simplifier le tracé du circuit imprimé puisqu'il n'est plus nécessaire de véhiculer des bus d'adresses et de donnée d'un composant à un autre.
- L'augmentation de la fiabilité du système puisque, le nombre des composants diminuant, le nombre des connexions composants/supports ou composants/circuits imprimés diminue.
- Réalisation des applications non réalisables avec d'autres composants. [7]

III.5. CONCLUSION

Le présent chapitre constitue le cœur de notre projet. En effet, nous avons présenté L'architecture et les caractéristiques du microcontrôleur PIC 16f877a

En conclusion, nous pouvons dire que le microcontrôleur 16F877A peut bien jouer le rôle d'une unité de contrôle (commande) pour notre système, il contient tous les modules nécessaires pour la régulation. Pour fonctionner cette unité de contrôle il faut la programmer et l'adapter à un compilateur de programmation.

Chapitre IV

Implémentation et simulation

IV.1 INTRODUCTION

Dans ce chapitre on va implémenter le régulateur numérique (PI) dans un microcontrôleur PIC16f877A dont l'outil de programmation est le logiciel mikroCPro et l'outil de test (simulation) est le logiciel ProteusISIS.

IV.2 LES OUTILS DE DEVELOPPEMENTS

Dans le but de faire commander le moteur à courant continue à travers le pic16F877A .on doit combiner les deux outils suivant :

a _ proteusISIS pour la simulation.

b _ le mikroC pro pour la programmation du pic.

IV.2.1. Le proteus ISIS



IV.2.1.1. Prise en main du logiciel ISIS

Dans notre application nous avons introduit ce logiciel de simulation. Proteus isis qui permet de simuler les différentes étapes de la conception.

ISIS: Intelligent Schematic Input System cette interface permet de concevoir et de simuler les applications. Ce logiciel possède une très riche bibliothèque, chose qui permet des applications très diverses.

IV.2.1.2. Saisi et simulation d'un schéma

Dans cette partie, nous expliquerons les différentes étapes permettant d'entamer la saisi et la simulation d'un schéma.

IV.2.1.2.1. Démarrer le logiciel

Lancer depuis le raccourci du bureau ou du menu **Démarrer** de votre ordinateur le logiciel **Proteus - ISIS.**

IV.2.1.2.2. Création de la liste des composants

-Sélectionner l'icône composant : 

-Dans la fenêtre du sélectionneur d'objets, cliquer sur P : 

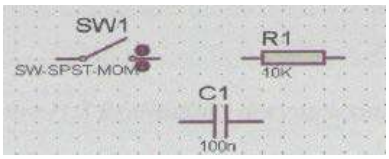
Une nouvelle fenêtre **(Pick Devices)** de choix de composants s'ouvre :

Trouver les composants désirés à partir d'un **mot clé,** ou dans la **catégorie**

proposée, puis dans la **sous-catégorie** (par défaut choisissez dans la sous-catégorie **Generic.** Cliquer une fois pour faire apparaître le composant, s'il convient double Cliquer pour qu'il s'inscrive sur la liste de travail **DEVICES.** Procéder ainsi, pour placer dès le début de l'application que vous voulez réaliser tous les composants que vous aurez besoin par la suite sur la liste de travail, puis fermer la fenêtre en cliquant sur **OK.**

IV.2.1.2.3.Placement des composants sur le schéma

Sélectionner le composant à placer dans la liste faite Précédemment et le placer dans la zone de travail de la fenêtre d'édition



IV.2.1.2.4.Placement des connexions externes

Pour GROUND ; OUTPUT ; INPUT...Après avoir cliqué sur cette fonction  Sélectionnez la connexion et après l'avoir mise dans le « panier », la placer sur le schéma. Pour notre schéma, nous utiliserons la masse (GROUND).

IV.2.1.2.5.Placement des alimentations et des générateurs

II faut maintenant placer les différentes alimentations ou générateurs utilisés.

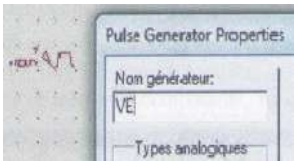
• Cliquer sur et sélectionner un générateur (DC, SINE, PULSE...).

• Ensuite cliquer sur le schéma pour le placer (ici Puise = rectangle).

• Les alimentations ou générateurs placés, il faut leur donner un nom et régler les différents paramètres qui sont fonctions du type de générateurs choisi.

Cliquer gauche sur le générateur, le symbole devient rouge, recliquer gauche, une fenêtre pour éditer les propriétés s'ouvre.

- ✓ Donner un nom (VE).
- ✓ Régler l'amplitude des tensions : VEmin =[V] et VEmax =[V]
- ✓ Régler pour avoir un signal carré de fréquence (désirée).



IV.2.1.2.6. Réalisations des connexions

II nous reste à relier les connexions entre les différents symboles. Sélectionner  dans le menu, Aller sur l'extrémité d'un composant, cliqué, le début de la liaison est créée, puis aller cliquer sur le composant à relier pour finir la liaison, le logiciel positionne seul le chemin.

IV.2.1.2.7.placement des instruments de mesure

Là il faut choisir les appareils de mesure que l'on souhaite utiliser en cliquant sur l'icône 



Pour un voltmètre continu (**DC VOLTMETER**), comme pour un voltmètre réel, il mesure une DDP (Différence de potentiel) entre les 2 points connectés, il faut donc le relier à 2 points.

IV.2.1.2.8. Animation et simulation du fonctionnement

Pour lancer l'animation. Cliquer sur la flèche, comme sur un appareil multimédia



IV.2.2 mikroC PRO

La nouvelle version appelée ***mikroC PRO*** dispose de très nombreuses améliorations du compilateur ***mikroC*** : nouvelles variables utilisables, nouvelle interface IDE, amélioration des performances du linker et de l'optimisateur, cycle de compilation plus rapide, code machine généré plus compact (jusqu'à 40 % suivant les cas), nouveaux PIC supportés, environnement de développement encore plus ergonomique, nouveaux exemples d'applications, etc...

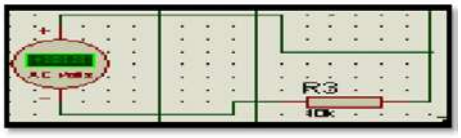
IV.2.2.1 IDE mikroC PRO

Lancer le compilateur mikroC PRO en cliquant sur l'icône



C'est sur la figure IV.2 que vous obtenez lorsque vous démarrez l'IDE *mikroC PRO* pour la première fois.

Une description détaillé de toutes les options disponibles dans ce compilateur prendre trop de temps, de sorte que nous allions sauter cette étape. Au lieu de cela, nous allons décrire salement le processus d'écriture d'un programme en langage *mikroC*, un simulateur de contrôle. Pour plus d'informations reportez-vous à l'aide [F1].



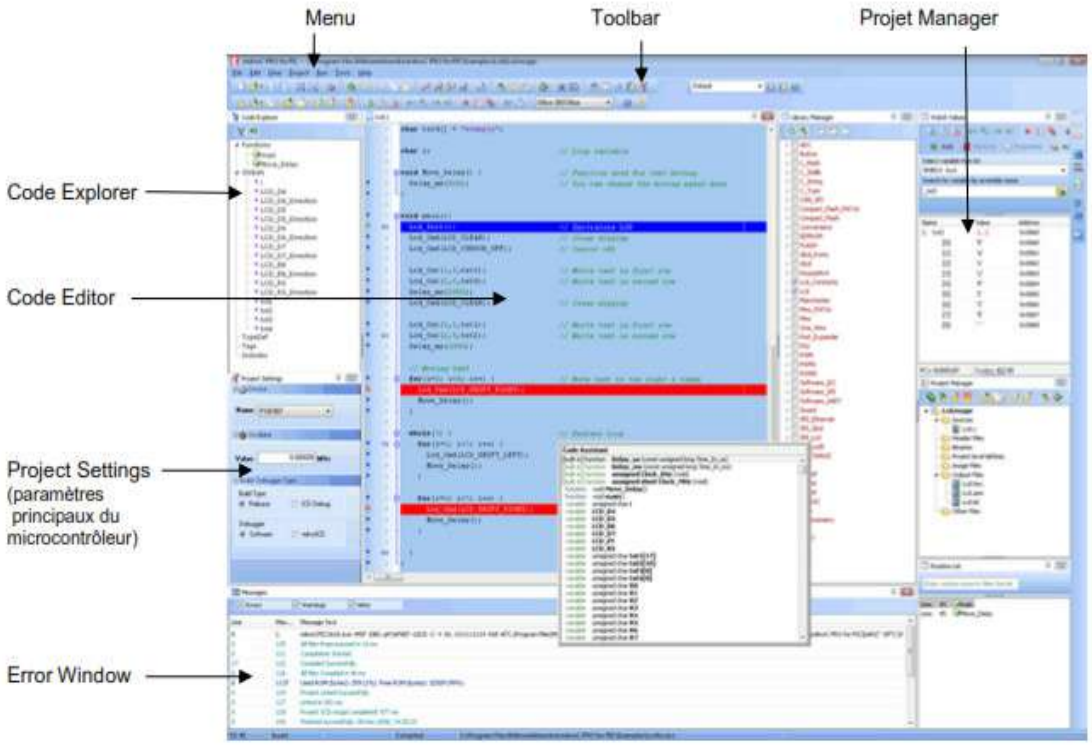


Figure. IV.1 : *L'environnement IDE du compilateur microC PRO*

Editeur de code (voire la figure IV.2 : Code Editeur)

L'éditeur de code est le même que toute éditeur de texte standard pour l'environnement de Windows, y compris Copie, Coller, Annuler les actions etc.… Il possède en plus des ressources comme suit :

- Coloration syntaxique réglable
- Assistant de code
- Assistant de paramètre
- Mise en forme automatique

Dans la boîte de dialogue *Obtions* (figure IV.3) vous pouvez configurer la coloration syntaxique, l'aide pour le code et paramètres, la correction automatique etc. Pour accéder à ce dialogue cliquez sur *Tools* > *Options* du menu déroulant ou sur l'icône 

Assistant de code : Si vous imprimez les premières lettres du mot et puis appuyez sur *Ctrl* + *Espace*, tous les identificateurs autorisés correspondant aux caractères imprimés seront offerts

dans une fenêtre (voir la figure IV.3). Maintenant, nous pouvons continuer à réduire le choix de taper ou d'utiliser la sourie pour sélectionner l'option appropriée dans la proposée et appuyez sur *Entrée*.

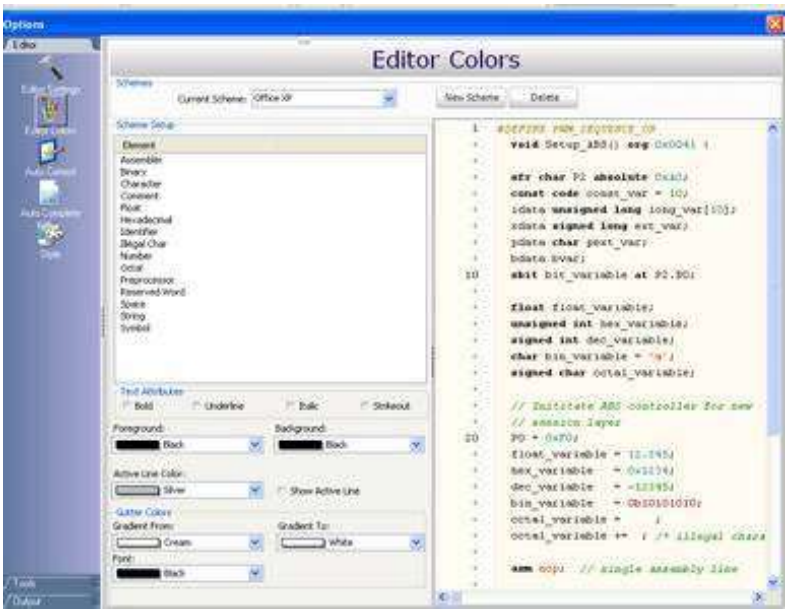


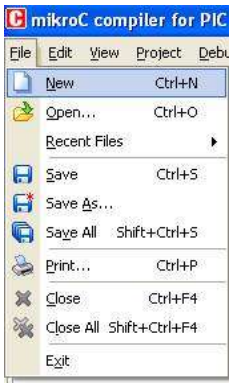
Figure. IV.2 : *boite de dialogue « options »*



Figure IV.3: *assistant de code*

Mise en forme automatique : Afin d'obtenir une lisibilité maximale, il est possible de générer automatiquement la mise en forme de certaines instructions. Par exemple, tapez l'instruction IF, puis *CRTL + J*. a ce stade, l'éditeur ajoute tout seul les instructions. A vous ensuite de compléter le programme.

Affichage syntaxique coloré : Toujours dans le but d'obtenir une lisibilité maximale de votre programme, il vous est possible de configurer entièrement les couleurs de chaque partie du listing (figure IV.3). Par exemple les commentaires en "vert", les instructions en "noir", les valeurs numériques en "bleu clair", etc.



IV.2.2.2 Création d'un nouveau fichier source

Pour créer un nouveau fichier source, procédez comme suit :

Sélectionne ***New*** depuis le menu ***File*** ou pressez *Ctrl+N* ou cliquez sur l'icône *New Fill* depuis la barre d'outils *File*.

Une nouvelle fenêtre s'ouvrira dans laquelle vous pourrez saisir votre code source. Sélectionnez alors ***Save*** depuis le menu ***File*** ou pressez *Ctrl+S* ou cliquez sur l'icône *Save File* depuis la barre d'outils *File* et nommez ce dernier comme vous le voulez.

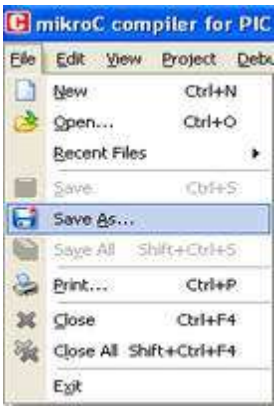
IV.2.2.3 Ouvrir un fichier existant



Sélectionnez ***Open*** depuis le menu ***File*** ou pressez *Ctrl+Q* ou cliquez sur l'icône *Open File* depuis la barre d'outils *File*. Dans la boite de dialogue ***Ouvrir,*** sélectionnez alors l'emplacement du fichier que vous désirez ouvrir et cliquez sur le bouton *Ouvrir*.

Le fichier sélectionné s'affiche dans sa propre fenêtre. Si le fichier sélectionné est déjà ouvert, sa fenêtre d'édition (Editor) deviendra alors active.

IV.2.2.4 Sauvegarder un fichier



Assurez-vous que la fenêtre contenant le fichier que vous voulez sauvegarder est active.

Sélectionnez ensuite ***Save*** depuis le menu ***File*** ou pressez *Ctrl+S* ou cliquez sur l'icône *Save Fail* de la barre d'outils *File*.

IV.2.2.5 Sauvegarder un fichier sous un nom différent

Assurez-vous que la fenêtre contenant le fichier que vous voulez



sauvegarder est active.

Sélectionnez ensuite ***Save As*** depuis le menu ***File***. Une boîte de dialogue '***Save*** [nom du fichier] ***.c as***' s'affichera alors.

Dans cette boîte, sélectionnez l'emplacement dans lequel vous voulez sauvegarder le fichier. [8]

IV.3 IMPLEMENTATION DU REGULATEUR PI

IV.3.1 principe de fonctionnement

Le principe de la régulation numérique PI pour la commande du moteur à courant continu est basé sur la variation de l'alimentation (pilotage en tension) de ce dernier dans un intervalle de tension compris entre 0V et 12V, suivant la vitesse désirée.

On doit donc passer d'une façon successive et répétitive par les taches suivantes :

a- La lecture de la vitesse de référence
b- Traitement numérique de la loi de commande
c- Ecriture de la commande
d- Mesure de la vitesse du moteur à courant continu
e- Affichage de la vitesse (optionnel)

Le schéma synoptique suivant englobe les taches citées ci-dessus :

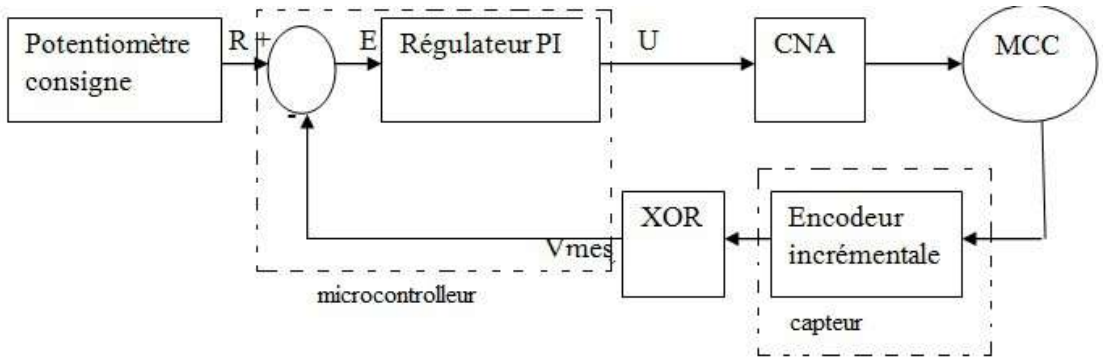


Figure IV.4. *Schéma synoptique*

IV.3.1.a la tache n°1 : lecture de la vitesse de référence pour réaliser cette tache on doit :

1- Configurer le microcontrôleur à recevoir une tension analogique (consigne)
2- Convertir cette tension en vitesse sous forme d'une valeur numérique (image de la tension)

IV.3.1.a.1. configuration du convertisseur A/D : nous avons configuré le convertisseur du microcontrôleur à travers le registre ADCON1 (ADCON1=143) de telle sorte que :

1	0	0	0	1	1	1	1
Bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0

- La première ligne du microcontrôleur de port A: RA0 est configure comme entrée analogique réserver à la consigne (M0=12Vmax)

- Les deux lignes de ports A ; RA2, RA3 comme entrées de tension de références (RA2= 0V, RA3= 12V)
- D'autre part la résolution du convertisseur est 10 bits cela signifie que 12V est représentée par 1111111111 et le pas du convertisseur est donc de 0.012V qui provient du fait que : $\frac{12V}{1023} = 0.012$ c-à-d :

0V ⟶ 0000000000

0.012V ⟶ 0000000001

0.024V ⟶ 0000000010

.

.

6V ⟶ 00011111110

.

.

12V ⟶ 1111111111

Exemple de 12V : exemple de 0V :

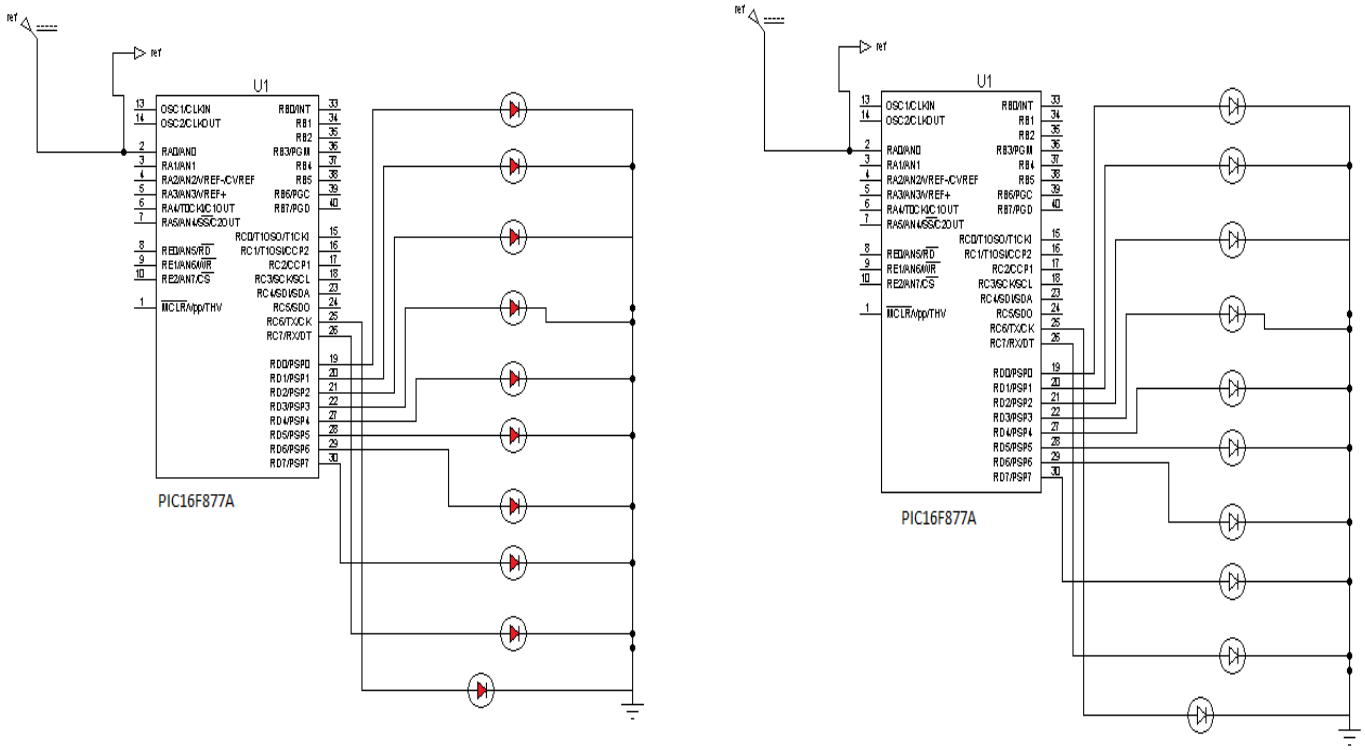


Figure IV.5. *La conversion analogique numérique*

IV.3.1.a.2. conversion tension ⟶ vitesse

La vitesse nominale du notre moteur est de 180 tr /min, nous avons donc donné la tension 12V pour présenter cette vitesse :

$$Vn = \frac{E.180}{1023} \text{ tr/min} \qquad \textbf{(IV.1)}$$

Ou :

E : consigne après conversion (consigne numérique)

Vn : vitesse nominale numérique

Exemple :

Si la consigne 6V alors

$E=(0001111110)_2 \longrightarrow (E=510)_{dec}$

D'où $V=\frac{510*180}{1023} = 89.7$ tr/min

IV.3.1.b. tache n°2 : traitement numérique de la commande

La fonction du transfert du régulateur numérique PI est donné par :

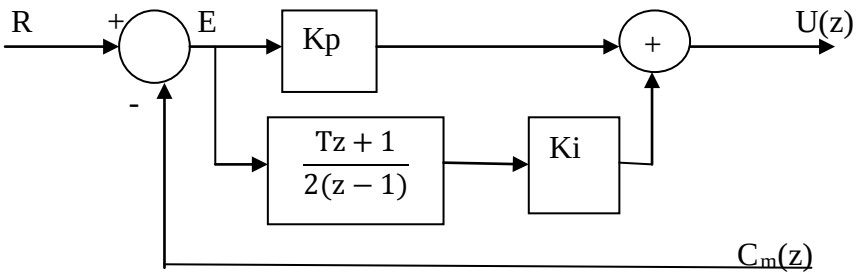


Figure IV.6. *Schéma bloc de la fonction de transfert PI*

$$\frac{U(z)}{E(z)} = Kp + Ki\ \frac{T}{2}\frac{z+1}{z-1} = \frac{Kp\,(z-1)+Ki\,\frac{T}{2}(z+1)}{z-1} \qquad \textbf{(IV.2)}$$

$$\frac{U(z)}{E(z)} = \frac{Q0\,z+Q1}{\boldsymbol{z-1}} \qquad \text{où} \begin{cases} Q0 = Kp + Ki\frac{T}{2} \\ Q1 = -Kp + \frac{T}{2} \end{cases} \qquad \textbf{(IV.3)}$$

Avec : E : erreur

U : la commande

De cette fonction de transfert nous avons mis en œuvre la loi de commande (équation récurrente) comme suite :

$z\,U(z) - U(z) = Q0\,z\,E(z) + Q1\,E(z) \qquad *z^{-1}$ **(IV.4)**

$U(z) - z^{-1}\,U(z) = Q0\,E(z) + Q1\,z^{-1}\,E(z)$

$U(k) - U(k-1) = Q0\,e(k) + Q1\,e(k-1)$

$U(k) = U(k-1) + Q0\,e(k) + Q1\,e(k-1)$

Cela signifie que la commande actuelle U(k) est égale à la somme de la commande précédente plus un coefficient Q0 multiplie par l'erreur actuelle plus un coefficient Q1 multiplie par l'erreur précédente donc notre traitement se base sur deux équations :

- $e(k) = Vref(k) - Vmes(k)$ (Vmes : vitesse mesurée) **(IV.5)**
- $U(k) = U(k-1) + Q0\,e(k) + Q1\,e(k-1)$ **(IV.6)**

IV.3.1.c. tache n°3: écriture de la commande

Comme la commande est présentée sur 10 bits nous avons configuré donc le port D plus deux lignes de port B du microcontrôleur comme sorties.

A ce stade la commande est toujours sous forme numérique alors nous avons interfacé un convertisseur N/A entre le microcontrôleur et le moteur.

Voici le schéma électrique de CNA :

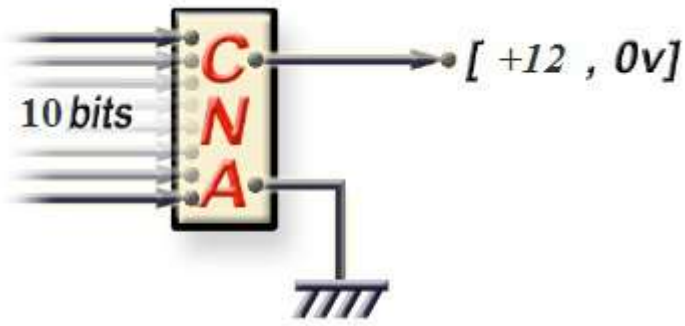


Figure IV.7. *Schéma électrique de CNA*

Dans ce schéma on voie que si Unum= $(1111111111)_2$, Alors Uana= 12V

IV.3.1.d. tache n°4 : mesure de la vitesse du moteur

Le moteur qu'on a utilisé est doté d'un encodeur incrémentale de deux rangées, chaque rangé compte 120 révolution , les deux sorties de ce encodeur sont introduit dans une porte XOR afin d'avoir un seul train d'impulsion (x2=2).

Exemple :

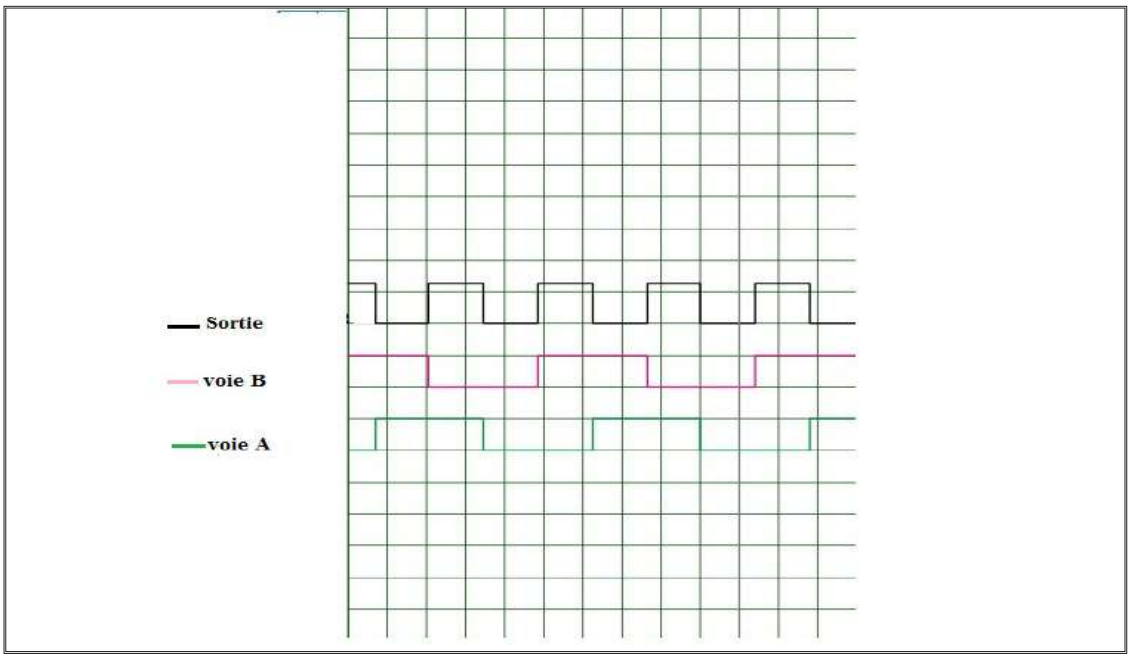


Figure IV.8. *Résultat de simulation*

la sortie de celle-ci est reliée à l'entrée RB0-int du microcontrôleur (utilisé comme entrée d'interruption externe) dans le but de comptée le nombre de fronts. De ce train d'impulsion le compte va nous permettre de savoir le nombre de tour fait par le moteur par l'application de la formule suivante :

$$\theta° = \frac{\text{nombre de fronts}}{2*120} 360° \quad \textbf{(IV.7)}$$

Par exemple le moteur fait 1 tour, alors le nombre de fronts sera 240 ce qui donne $\theta°=360°$

- Pour calculer maintenant la vitesse angulaire on prend le nombre de tour trouvé précédemment et on le divise sur un intervalle de temps fixe. (T=10ms dans notre cas) d'après l'équation suivante :

$$\frac{d\theta}{dt} = \frac{\theta}{T} \quad \textbf{(IV.8)}$$

L'intervalle de temps est mesuré par le timer de microcontrôleur ou ce dernier déclenche une interruption interne à chaque T. (T : période d'échantillonnage). Dans notre cas on a travaillé avec le timer1du 16 bits parce qu'il est plus précis par rapport au timer0 qu'il a 8 bits.

L'équation qui permet de calcule T est donner comme suite :

$T = 4 * T_h * \text{prédivision} *(65535\text{-VI})$ **(IV.9)**

Ou : T_h : période d'horloge

VI : valeur initiale

On a une fréquence de 8MHz ⟶ donc $T_h = 1/8*10^6$

Exemple :

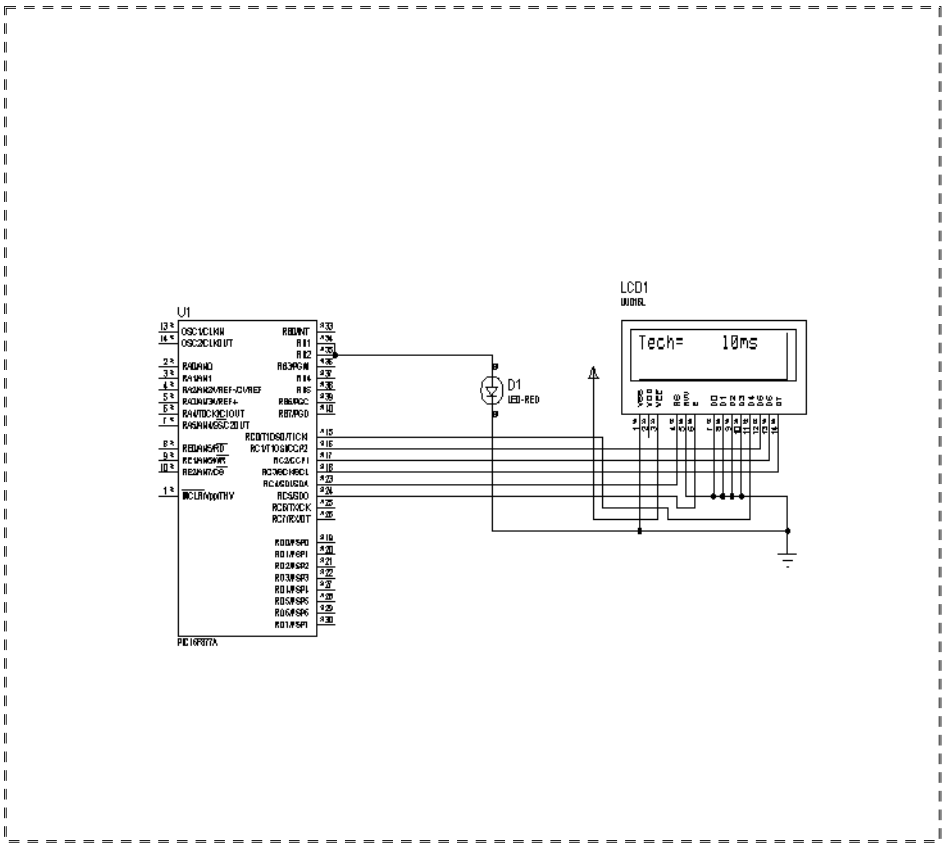
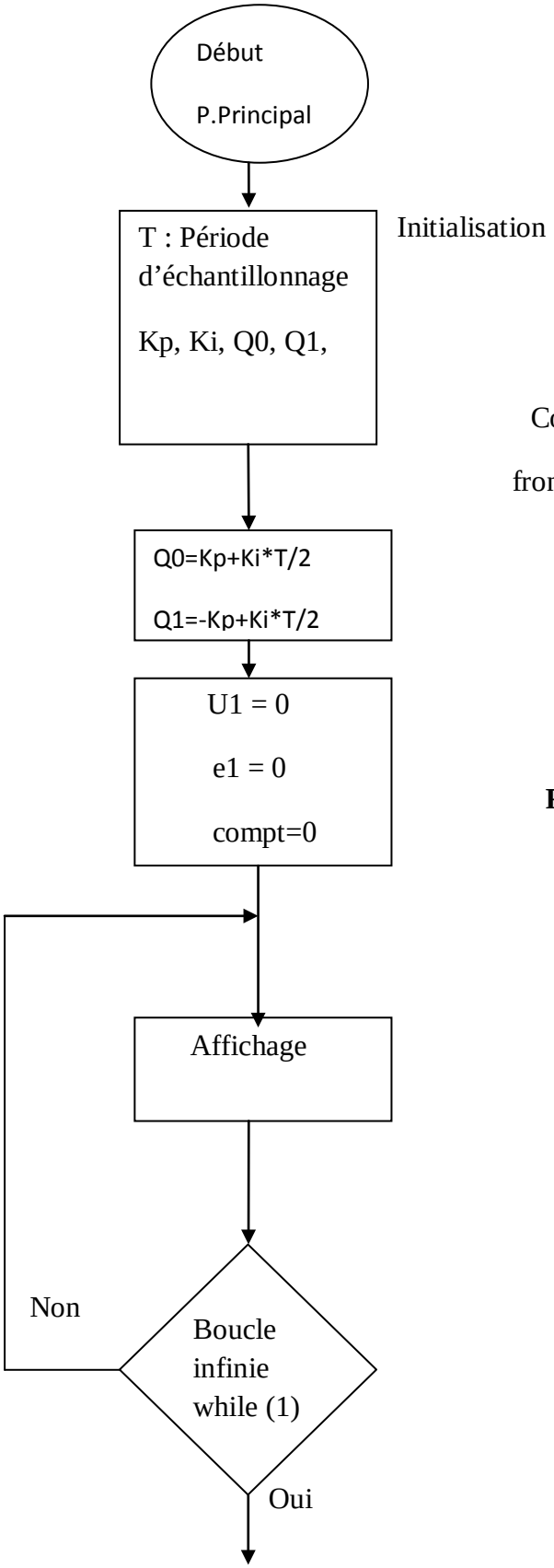


Figure IV.9. *Schéma électrique du timer1*

IV.3.1.e. tache n°5 : l'affichage

Cette tache est optionnelle, que nous avons ajouté pour afficher la vitesse angulaire

IV.3.2.Organigramme



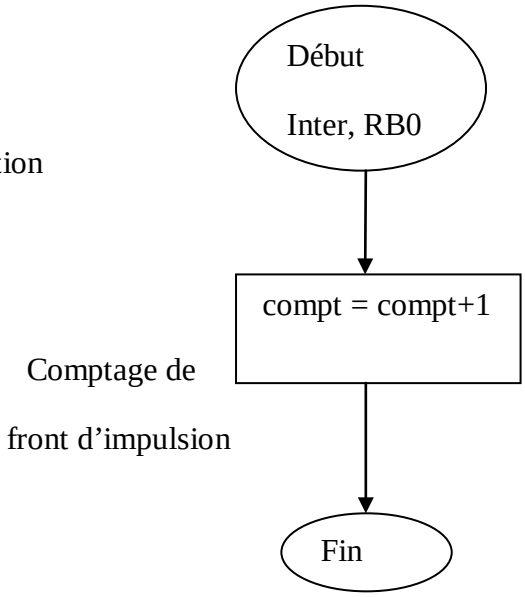


Figure IV.11. *Organigramme de Programme De comptage de nombre de tours.*

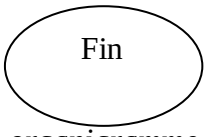


Figure IV.10.*organigramme de Programme principale.*

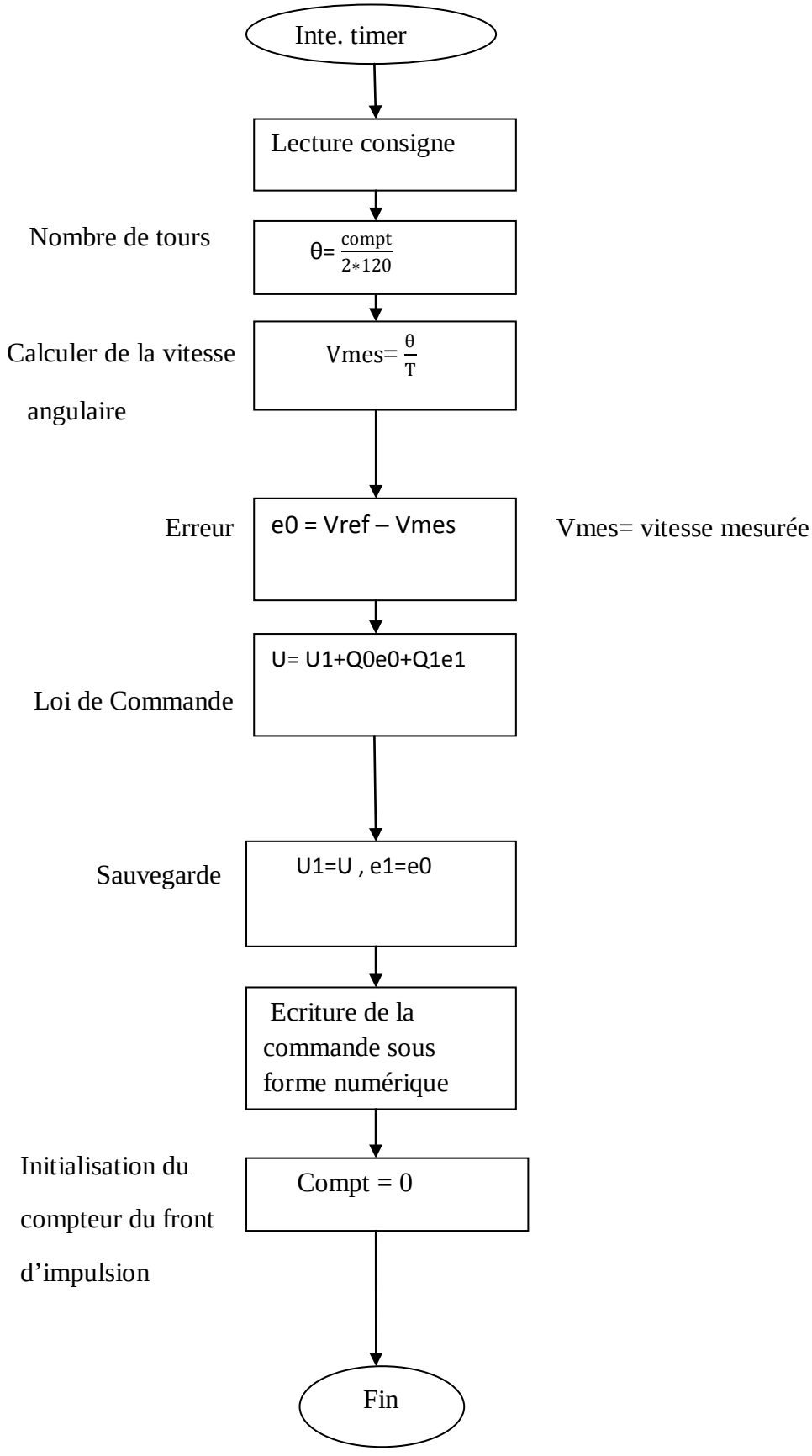


Figure .IV.12.*Organigramme de Programme du génération de la commande.*

IV.3.3. schéma électrique

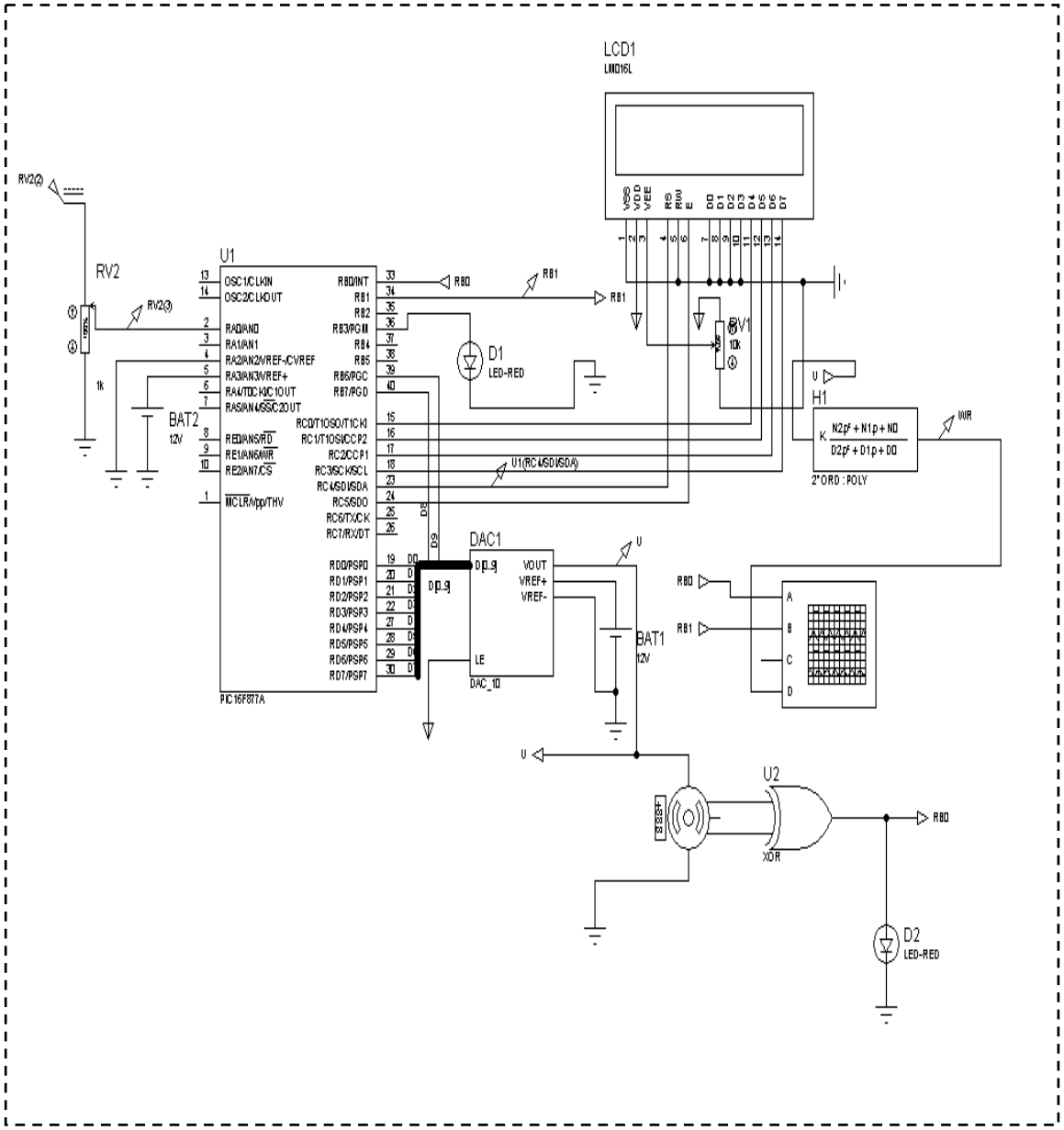


Figure IV.13. *Schéma électrique*

IV.3.4. Résultat de simulation

Pour Kp = 0.2684 et Ki = 12.5

On pend la valeur maximal de l'alimentation V_{ref} = 12V

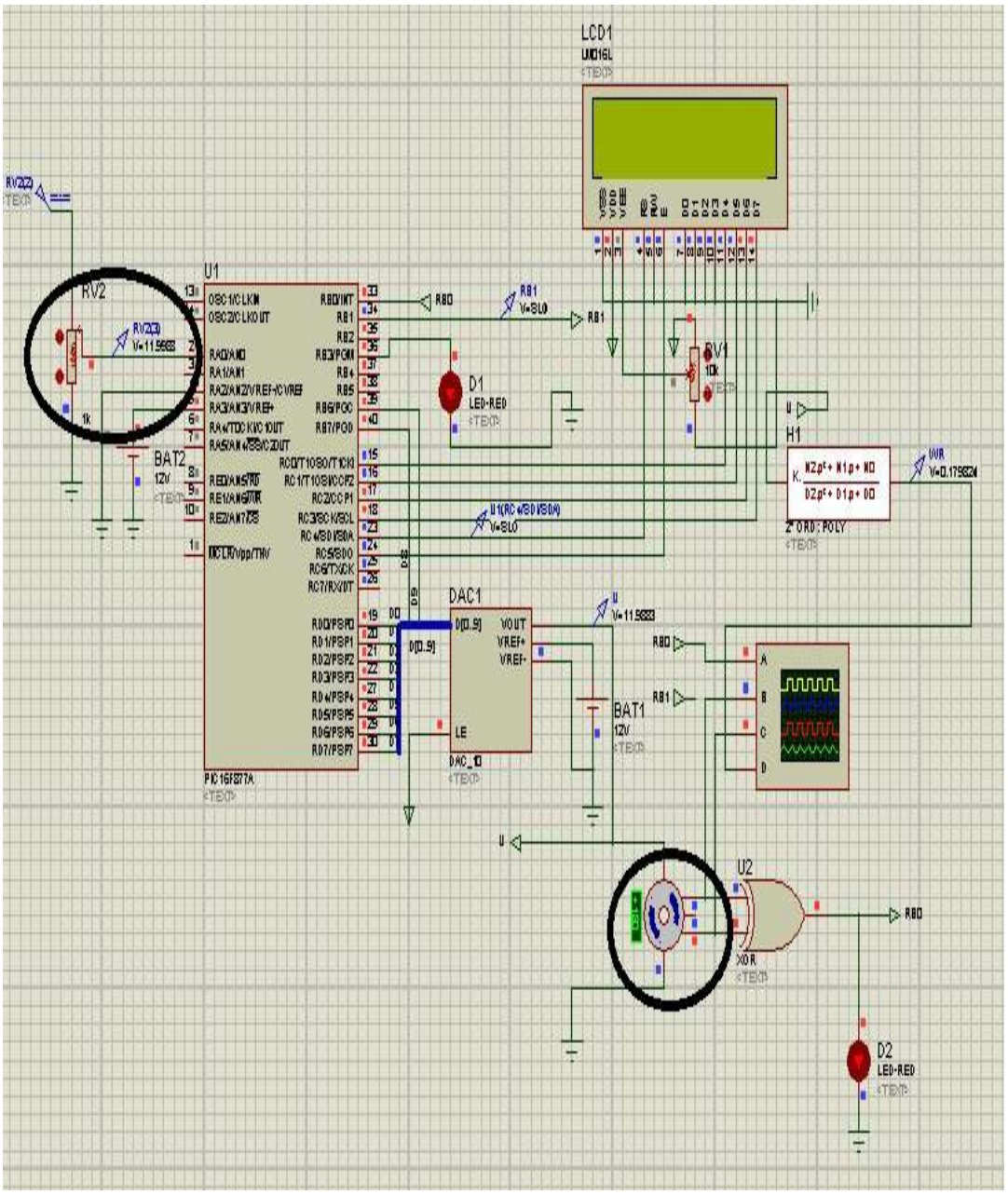
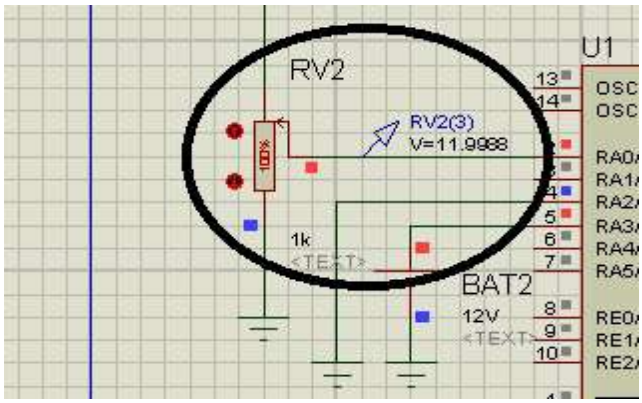
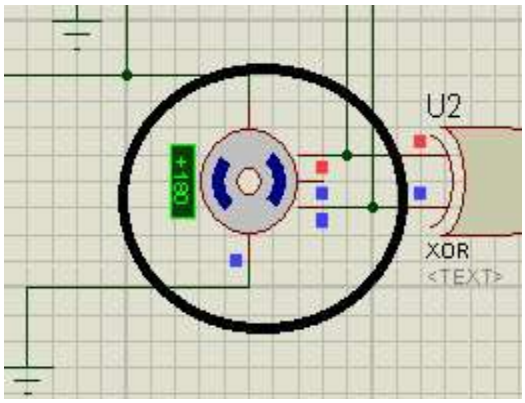


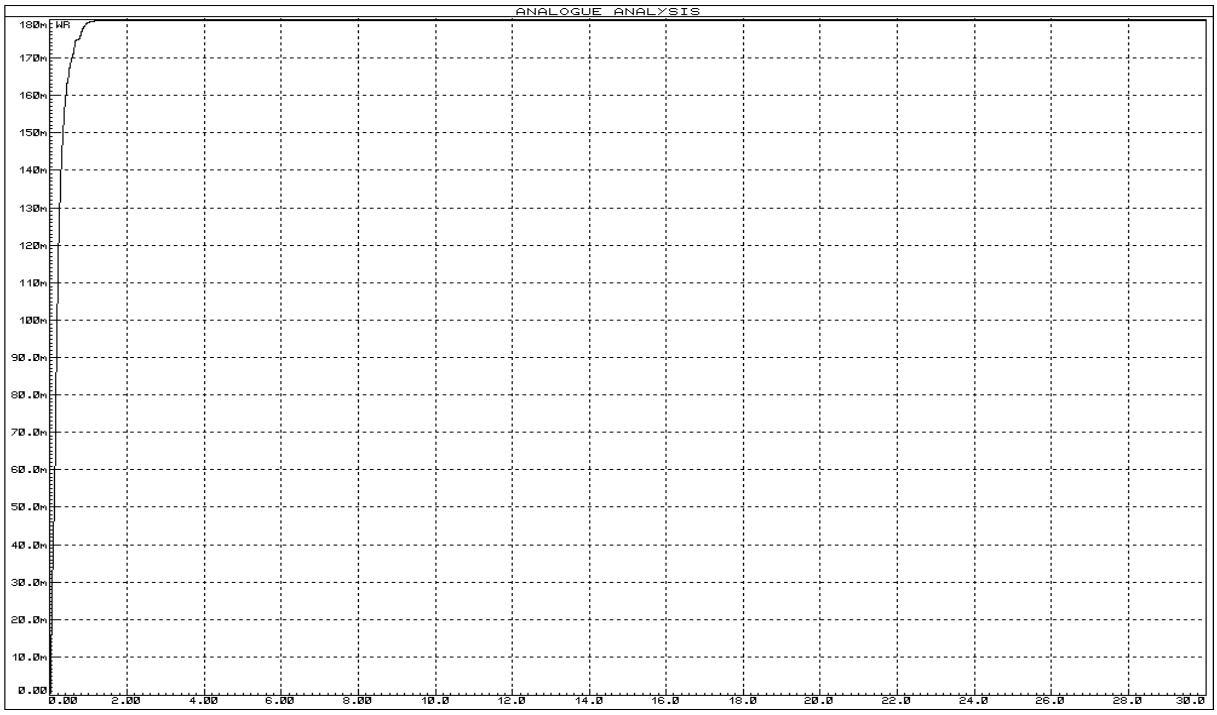
Figure IV.14. *Simulation du schéma électrique*



(a)



(b)

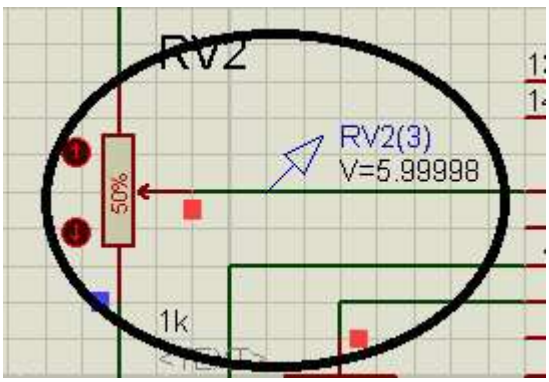


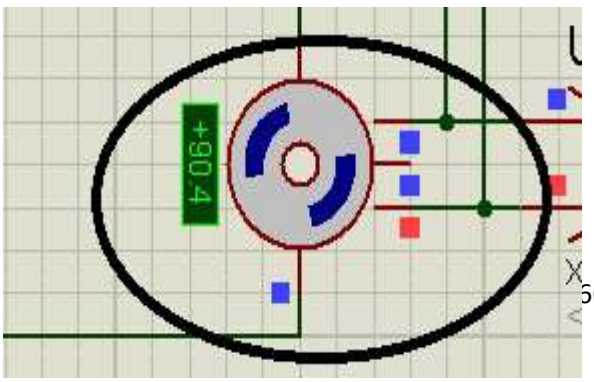
(c)

Figure IV.15. *Résultat de simulation (Vref =180 tr/mn)*

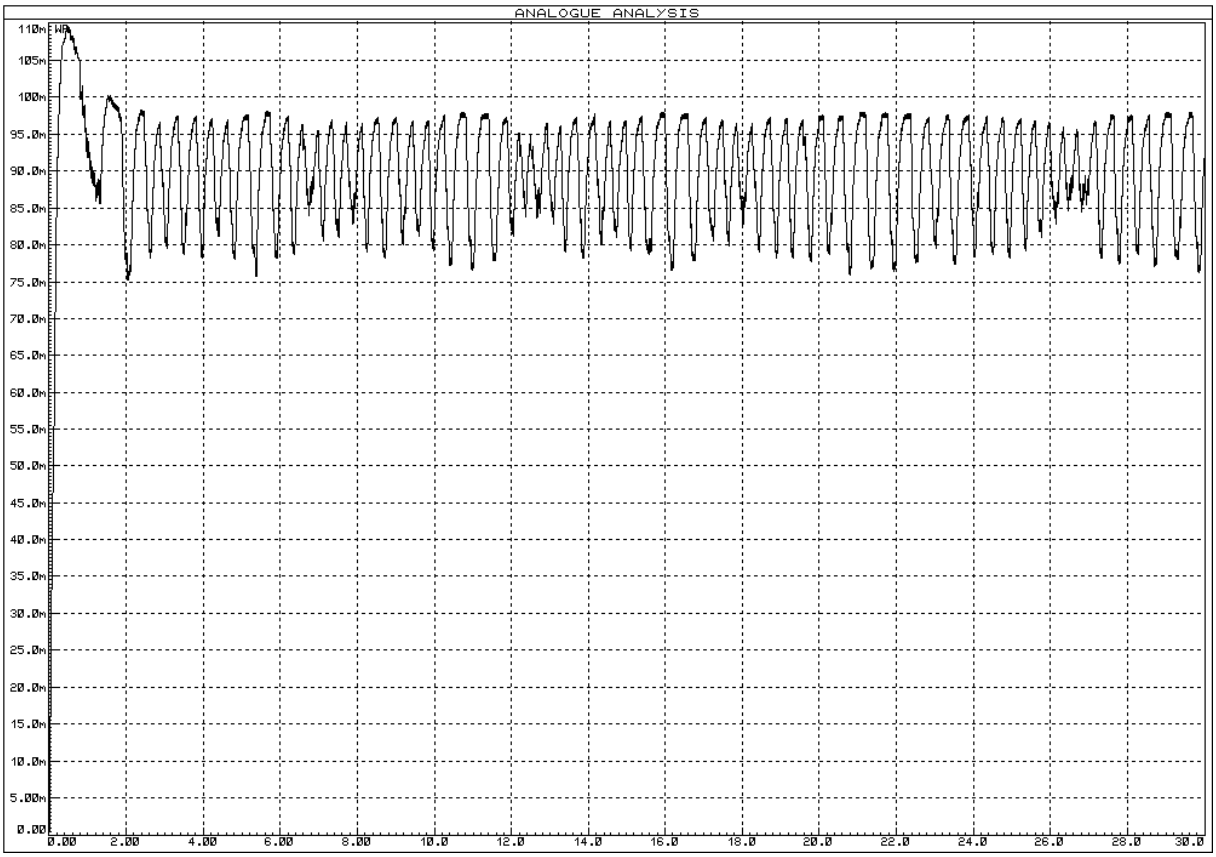
On voit que d'après la figure (b) et (c) que la vitesse suit bien notre vitesse de référence 180 tr/mn qui correspond à12V (voir la figure a)

Dans le deuxième cas on prend la valeur d'alimentation V_{ref} = 6V





(a) (b)



(c)

Figure IV.16. *Résultat de simulation (Vref=90tr/mn)*

on remarque que la vitesse tourne autour de la vitesse de référence (90tr/mn) ,ou ces ondulations sont due au mauvais choix de Kp, Ki (précision), comme il est montré dans la figure (b) et (c).

IV.4. CONCLUSION

Dans ce chapitre nous avons réalisé toutes les taches citées précédemment qui sont concrétisé par le programme proteus ISIS.

Conclusion générale

Ce travail à été pour nous, le premier pas vers un monde nouveau et complexe, mais très intéressant notre recherche bibliographique dans les domaines de l'électronique, de l'électrotechnique et de l'informatique tel que « les interruptions électronique, les machines à courant continu et la programmation », nous a permis d'enrichir nos connaissance et de développer notre base théorique.

Dans ce travail nous avons essayé d'apporter d'utiliser et d'employer les possibilités offertes par l'emploi des PICs dans le domaine des machines à courants continu. Notre objectif principal était donc de fournir à la fois un outil moins coûteux et d'une utilité.

- Nous avons implémenté le régulateur PI dans un microcontrôleur PIC pour la commande de la vitesse de la machine à courant continu.
- La régulation de la vitesse est réalisée à l'aide du PIC.

Toute fois le champ d'investigation est très large et les méthodes très nombreuses. Des améliorations peuvent être apportés à ce travail afin de le perfectionner et de l'enrichir.

_ Au niveau du circuit de commande.

_ Au niveau de la régulation.

Par exemple de compléter ce travail afin de commander et de réguler la vitesse d'un moteur dans les sens de rotation.

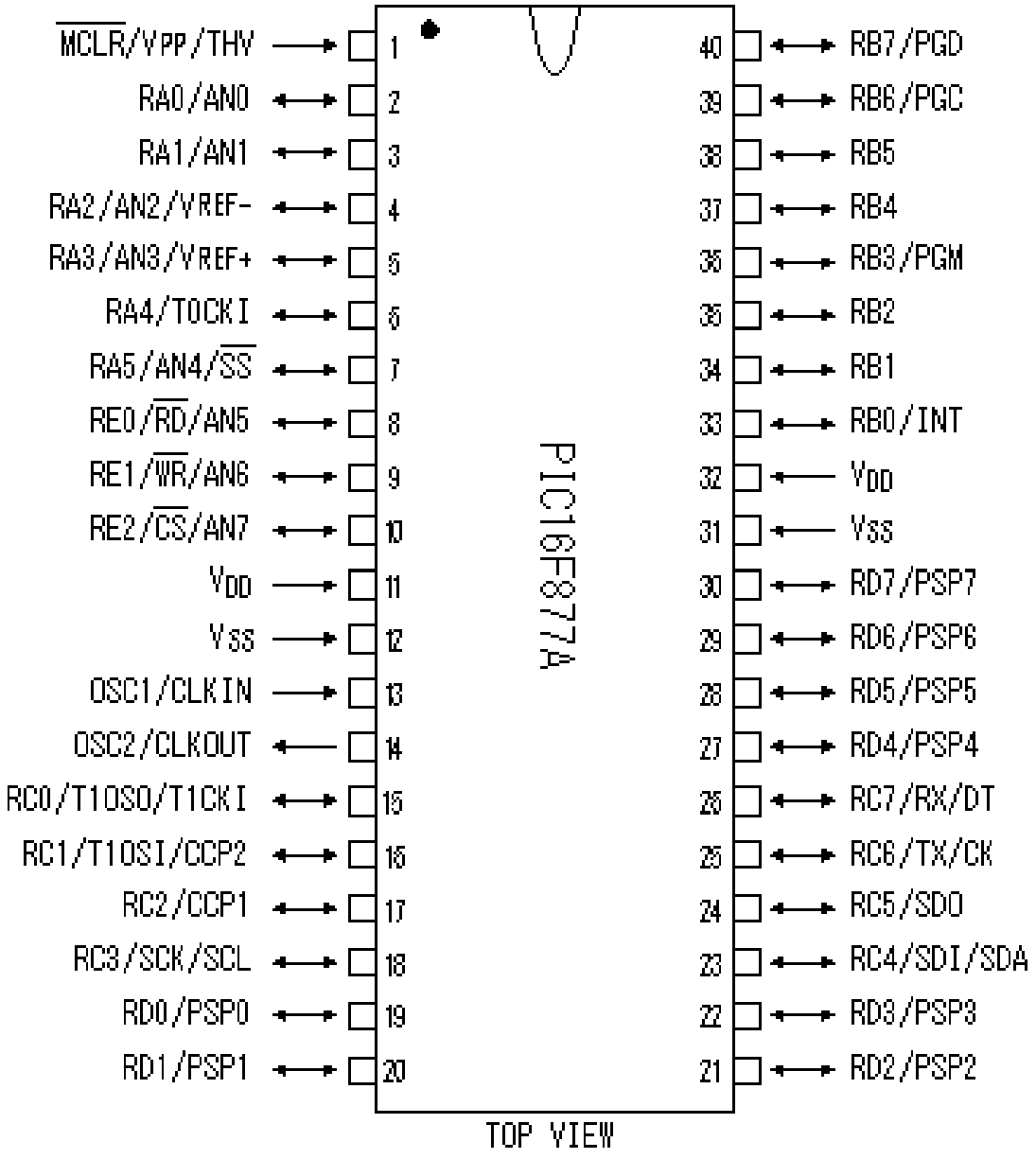
L'implémentation de l'algorithme PI par le microcontrôleur PIC16F877A a nécessité la maîtrise de l'architecture de ce dernier et une familiarisation avec son environnement de développement.

Le circuit PIC16F877A est très intéressant par son rapport qualité/prix.

Nous souhaitons dans le futur une réalisation pratique nécessitant une commande d'une machine à courant continu en temps réels, afin d'avoir une grande précision.

Annexe

L'annexe 1 : DATACHEET de microcontrôleur pic 16f877a.

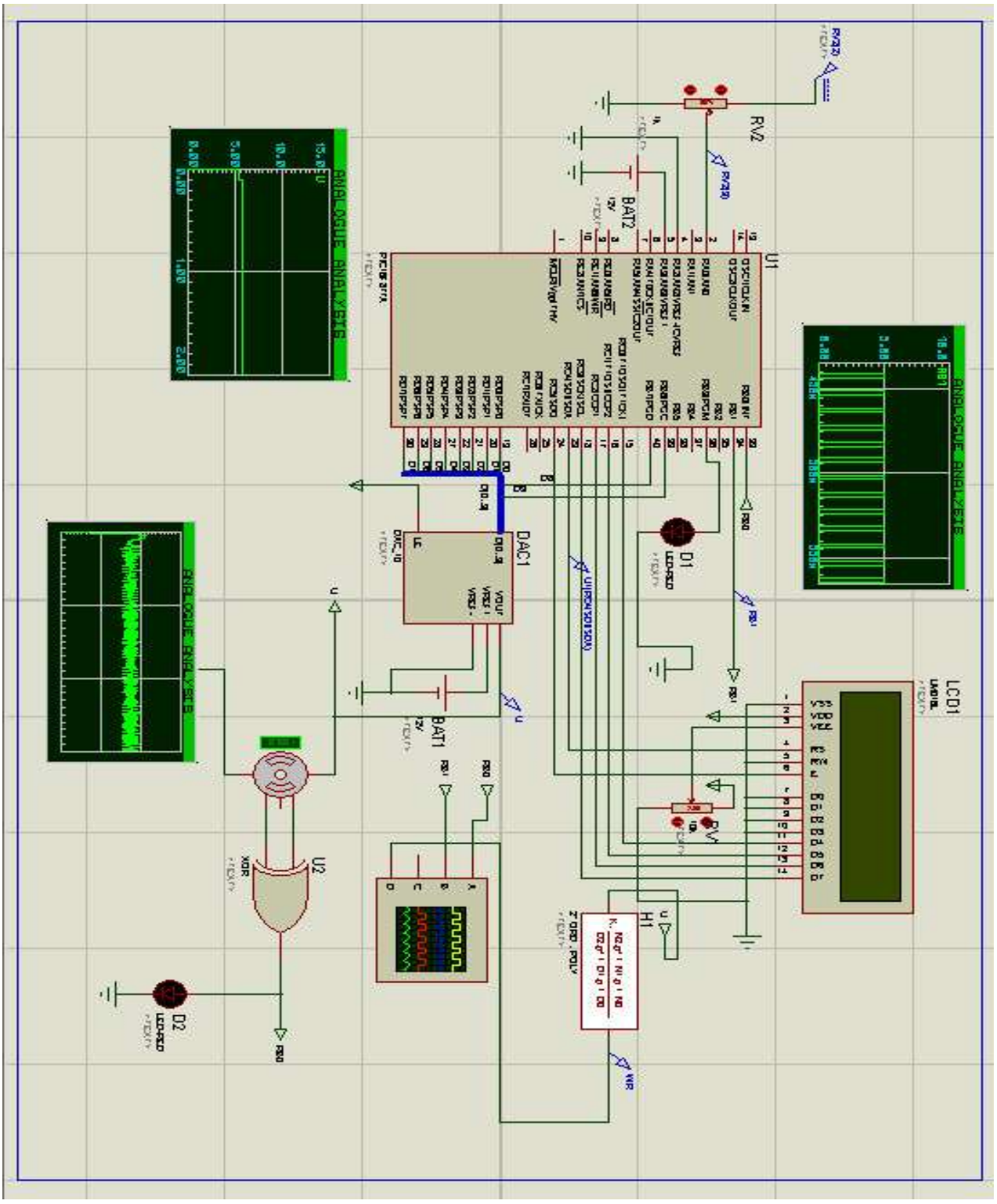


L'annexe 2 : tableau de configuration des pin input du convertisseur A / N

PCFG3: PCFG0	AN7	AN6	AN5	AN4	AN3	AN2	AN1	AN0	Vref+	Vref–
0000	A	A	A	A	A	A	A	A	VDD	VSS
0001	A	A	A	A	Vref+	A	A	A	RA3	VSS
0010	D	D	D	A	A	A	A	A	VDD	VSS
0011	D	D	D	A	Vref+	A	A	A	RA3	VSS
0100	D	D	D	D	A	D	A	A	VDD	VSS
0101	D	D	D	D	Vref+	D	A	A	RA3	VSS
0110	D	D	D	D	D	D	D	D	VDD	VDD
0111	D	D	D	D	D	D	D	D	VDD	VSS
1000	A	A	A	A	Vref+	Vref–	A	A	RA3	RA2
1001	D	D	A	A	A	A	A	A	VDD	VSS
1010	D	D	A	A	Vref+	A	A	A	RA3	VSS
1011	D	D	A	A	Vref+	Vref–	A	A	RA3	RA2
1100	D	D	D	A	Vref+	Vref–	A	A	RA3	RA2
1101	D	D	D	D	Vref+	Vref–	A	A	RA3	RA2
1110	D	D	D	D	D	D	D	A	VDD	VSS
1111	D	D	D	D	Vref+	Vref–	D	A	RA3	RA2

A = analog input D = digital input

L'annexe 3 : schéma globale de simulation sous le logiciel proteus ISIS.



Bibliographie

Bibliographies :

[1] Jean-Paul LOUIS, Bernard MULTON, Yvan BONNASIEUX, Michel LAVABRE, ‘‘commande des machines à courant continu (MCC) à vitesse variable’’, mémoire de master, Paris, 2002.

[2] Nadir MAATER, Lembarek BESEKHRIA, ‘‘conception d’une carte de commande de la machine à courant continu par un microcontrôleur pic16F877A ’’, mémoire de master, institut nationale de la formation professionnelle CHAHIB BOUKHLOUF MOHAMED BEN EL HADI, Batna, 2009.

[4] Pascal BIGRAS : GPA-783, asservissement numérique en temps réel, mémoire de master école de technologie supérieur, 2011.

[5] Mohamed ARHOUJDAM, ‘‘mise en œuvre d’une régulation de vitesse d’un MCC’’, thèse de master université Mohamed V souissi, 2013.

[6] Mr. Ramzi SAADI, Mr. SALHI Nassereddine, “réalisation de la carte à un microcontrôleur pour le contrôle de bras manipulateur via un pc ’’, thèse de master, université med khider, Biskra, 2010.

[7] BRAHIMI Abdelkarim, GUEZOULI Housseyne : ‘‘étude et réalisation d’une carte de commande à base d’un microcontrôleur pic16F877 pour ponts redresseurs triphasés à thyristor’’, thèse de master, université abou-bakr belkaid, tlemcen, 2014.

[8] V.TOURTCHINE . ‘‘programmation en MikroC application pour les microcontrôleurs de la famille PIC. Article. Université M’HAMED BOGARA’’. Boumerdes. 2012.

[11] BOUICHE Hachemi, BRAHIMI Mohamed, ‘‘commande PID d’un moteur à courant continu’’, mini projet, université abderrahmane Mira, Bejaya, 2010.

[12] BRUYERE Philippe, GOMBERT Benjamin, VENCE Eric, ‘‘ Conception et réalisation d’un module P.I.D. a sortie analogique’’. mémoire de master ESME Sudria 2003.

[13] Dr. Ibrahim, ‘‘Microcontroller Based Applied Digital Control’’, John Wiley & Sons ,article, Ltd. ISBN: 0-470-86335-8 2006.

Les sites :

[3] http://www.ni.com./tutorial/7109/it/.

[9] http://www.mikroe.com/chapters/view/3/chapter-2-core-sfrs/

[10] http://www.mikroe.com/chapters/view/5/chapter-4-timers/