الجمهورية الجزائرية الديمقراطية الشعبية

MINISTERE DE L'ENSEIGNEMENT SUPERIEUR
ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITE DR. YAHIA FARES DE MEDEA

FACULTE DE TECHNOLOGIE

Département du Génie Électrique



Mémoire de Fin d'Études

Présenté pour l'obtention du Diplôme
de Master en Électrotechnique

Option : Commande Électriques

THEME

ETUDE ET SIMULATION D'UN REGULATEUR PI NUMERIQUE ET IMPLIMENTATION DANS UN MICROCONTROLEUR PIC16F877A POUR LA COMMANDE DE VITESSE D'UN MOTEUR A COURANT CONTINU

Proposé et dirigé par,

Mr. BETTACHE FAROUK

Préparé et présenté par,

ABD ALLAH ALHADJ HAYET
TERRAS NAWEL

Année Universitaire 2017-2018

REMERCIEMENT

Au terme de ce projet nous tenons à remercier tout naturellement en premier lieu ALLAH le tout puissant qui nous adonné la force, le courage et la patience de bien mener ce travail.

Un hommage particulier ou promoteur de cette étude le Mr. BETTACHE qu'il trouve ainsi l'expression de notre profonde reconnaissance pour sa compétence, ses encouragements et ses précieux conseils durant le déroulement de ce travail.

Nulle œuvre n'est exaltante que cette réalisée avec le soutient moral et financier des personnes qui nous son proches.

Nous remercie nous pères et mères qui sont toujours entouré et motive à sans cesse devenir meilleur

Nos frères et sœurs, qui sont assisté dans ces moments difficiles.

Notre amies de par le monde qui n'ont cesse de nos encourage.

Nos sentiments vont également à toutes des personnes qui ont contribué de prés ou de loin à la réalisation de ce travail.



Dédicaces

Je dédie ce modeste travail à : La Source de joie et de bonheur, celui qui s'est toujours Sacrifié pour me voir réussir, que dieu te garde dans son vaste paradis, à toi Mon père.
A la lumière de mes jours, la source de mes efforts, la flamme de Mon cœur, ma vie et mon bonheur, maman que j'adore.
A ma sœur
A toute la famille TERRAS,

Et a tous mes amis sans exception.

NAWEL

Dédicace

Que ce travail témoigné de mes respects :
A mes parents :
Grace à leurs tendres encouragements et leurs grands sacrifices, ils ont pu créer le climat affectueux et propice à la poursuite de mes études.
Aucune dédicace ne pourrait exprimer mon respect. Mon considération et mes profonds sentiments envers eux.
Je prie de bon dieu de les bénir. De veiller sur eux. En espérant qu'ils seront toujours fiers de moi.
A mes frères.
A ma famille.
Ils vont trouver ici l'expression de mes sentiments de respect et de reconnaissance pour le soutient qu'ils n'ont cessé de me porter.
A mes professeurs.
A toutes mes amies et mes collègues :
Ils vont trouver ici le témoignage d'une fidélité
et d'une amitié infinie.

Hayet

الملخص:

الهدف من عملنا هذا هو إدماج مصحح(معدل) PI داخل معالجPIC 16F877A للتحكم في سرعة محرك ذو تيار مستمر(MCC) ، باستعمال لغة البرمجة MikroC لكتابة البرنامج الخاص لهذا التعديل ، وبرنامج Proteus isis للمحاكاة يتضمن العمل بعض المفاهيم الضرورية و الأساسية حول محرك ذو تيار مستمر، بالإضافة إلى لمحة عن المعالج pic16f877a ونبض تعديل العرض(PWM).

الكلمات المفتاحية : المعالج (PIC 16F877A) ،الإدماج ، تصميم المحاكاة، **المنظم MCC ،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.

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

Un aperçu sur le PIC 16F877A et modulation largeur impulsion(PWM), qui est dédié a cette application.

Mots clés : microcontrôleur **PIC : (Periphirol Interface contrôler),** Modélisation implémentation - Simulation, régulateur **PI,** moteur à 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.

In this work, some basics on the DC machine are given. An over view on the pic 16f877a which is dedicated to this application. And we realized these ideas by theoretical electronic components like digital and et pulse width modulation.

Is our peak 16F877A programming with the help of WIN 800 PIC programmer softwar

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

Liste des figures

Chaipitre I :

Figure(I.1) : Modes de fonctionnement d'uneMCC……...…2

Figure (I.2): Constitution de la machine à courant continu ……...……3

Figure (I.3): L'inducteur de la machine à courant continu ……3

Figure(I.4): Le rotor de la machine à courant continu ……...4

Figure (I.5) : Le dispositif collecteur / balais ……4

Figure(I.6) : Principe de fonctionnement d'un moteur à courant continu ……....5

Figure (I.7): Moteur à aimant permanent……6

Figure (I.8): Caractéristiques d'un moteur à excitation séparée……7

Figure (I.9): Le couple en fonction de courant d'alimentation……..……8

Figure(I.10): Caracteristique de vitesse d'un moteur série……9

Figure (I.11): Caracteristique de couple d'un moteur série……...….9

Figure (I.12): Schéma équivalent d'un MCC a excitation indépendante……..10

Figure (I.13) : a : Variation de la vitesse en fonction de la tension d'induit
b: variation de la vitesse en fonction du flux d'inducteur……13

Figure(I.14): Bilan des puissances d'un moteur à courant continu ……14

Figure(I.15) : Bilan des puissances d'un generateur a courant continu ……...14

Figure(I.16): Encodeur incremental ……16

Figure (I.17): Circuit interne du module……...17

Figure(I.18): Encodeur incremental..……..18

Chaipitre II :

Figure (II.1) : Boitier du PIC 16F877A ……...23

Figure (II.2) : Architecture interne du PIC 16F877A ……25

Figure(II.3) : Cycle de conversion……....30

Figure(II.4): Module du convertisseur……..…32

Chaipitre III :

Figure(III.1) : Schéma de principe d'une chaine de régulation……....38

Figure(III.2) : Schéma synoptique d'un régulateur PID……...39

Figure (III.3) : Schéma bloc de la fonction de transfert PI……..…42

Figure (III.4) : Schéma blocs du compensateur PI……44

Figure(III.5) : Approximation rectangulaire. ……...45

Figure (III.6) : Approximation rectangulaire devancée. ……..46

Figure (III.7). : Approximation trapézoïdale……47

Figure(III.8) : Schéma bloc en z du compensateur PI……..…..…48

Figure(III.9) : Procédure d'identification d'un modéle de système……..49

Figure(III.10) : Méthode "classique" d'identification……...51

Figure(III.11) : Réponse a l'échelon du système en boucle ouverte……52

Figure(III.12) : Paramètre a et L de la méthode Ziegler et Nichols……....……52

Chaipitre IV :

Figure(IV.1) : L'ouverture du mickroC……55

Figure(IV.2) : L'interface d'ISIS proteus v7……56

Figure (IV.3) : Choix des composants dans la bibliothèque……..…57

Figure(IV.4) : Connecter les composants……57

Figure(IV.5) : Schéma de la simulation……...58

Figure (IV.6): Schéma électrique sur proteus……...59

Figure (IV.7): Réponse de système……...60

Figure(IV.8): Export les donnees de système.....…60

Figure(IV.9): Import les donnees de système……...61

Figure(IV.10): L'utilisation de l'outil system identification……....62

Figure(IV.11): L'interface de l'outil system identification……..62

Figure(IV.12): Saisie des données relatives aux input et output du système……63

Figure (1V.13) : Choix de la description du système a estimer"tranfer function"……...63

Figure(IV.14): Choix du nombre des pOles et zéros de la fonction de transfert a estimer…..64

Figure(IV.15): Visualisation du résultat de l'estimation……..64

Figure(IV.16): Récupération de la fonction de transfert estimée……..65

Figure (IV.17): La comparaison entre la fonction de transfert estimé avec la réponse de système.....66

Figure (VI.18): Schéma synoptique de la chaine de régulation numérique de la vitesse de MCC……..……66

Figure (VI.19): Schéma synoptique detaillee de la chaine de régulation numérique de la vitesse de MCC……..……..…….67

Figure (IV.20): Potentiomètre……...68

Figure(IV.21) : Variation des largeurs de PWM……...…72

Figure (IV.22) : Organigramme de programme principale……...73

Figure(IV.23) **:** Organigramme de programme de la génération de la commande……...73

Figure(IV.24) : schéma de L293D……...74

Figure (IV.25) : simulation de l'encodeur……75

Figure(IV.26) : Representation de circuit LM331…….76

Figure(IV.27): réponse d'un systeme ……...……77

Figure(IV.28) : Schéma électrique complet du regulateur PI ……..81

Figure (IV.29) : La répense de système sous proteus……...81

Figure(IV.30) : schéma bloc globale de regulation de MCC par PI en Z…….82

Figure (IV.31) : La réponse de système sous simulink……82

Liste des tableaux

Tableau(II.1): Broches du PIC16F877A……..……..24

Tableau(II.2): Bits de registre ADCON1……..34

Tableau(II.3) : Bits de registre ADCON0.....35

Tableau(II.4): Module PWM de pic ……..……36

Tableau(III.1):Gain des compensateurs…….....52

Liste des acronymes et symboles

MCC: machine à courant continu.

PI: Proportionnelle intégral.

DC: Courant Continu (Direct Curent).

PWM : pulse width modulation (largeur d'impulsion modulation).

ADC : convertisseur analogique numérique.

PIC: (Perifirol Interface Controler).

CCP : (Capture / Compare / PWM)

RPM: round par minute.

Ie : le courant continu d'excitation.

E : La force électromotrice.

P : nombre de paire de pôles de la machine.

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

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

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

Φ: flux sous un pole de la machine en Webers.

V : la tension d'alimentation en volt.

I : l'intensité du courant d'alimentation en ampère (A)

Pu : la puissance utile en watts.

Pem : les pertes électromagnétiques.

Pa: puissance absorbé en Watts.

U : tension d'alimentation en Volt.

Ŋ : le rendement

Tu : le couple utile.

T : le couple.

R : résistance de l'induit du MCC.

L : inductance de l'induit du MCC.

e(t) : force électromotrice.

K_e : La constante électrique du moteur (constante de vitesse).

Ω(t) : la vitesse de rotation.

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

f : Coefficient de frottement visqueux.

Cr : Couple résistant.

C_p: Couple de pertes.

K_c : Constante de couple.

C_u : Couple moteur généré.

Cp (moment du couple de perte.

W0 : la pulsation propre.

A : le gain statique.

Ɛ: le facteur d'amortissement.

Vs: la tension de sortie.

Fe: la fréquence d'entrée.

I(t): Intensité traversant le moteur

Kp: Gain proportionnel

Ki: Gain intégral

Kd: Gain dérivé

Table de matières

Table de matières

Liste des figures…….....I
Liste des tableaux.....…….IV
Liste des acronymes……..…….……...….V
Table de matières…….VII
Introduction générale…….....…....1

Chapitre I : la machine a courant continu

I.1.Introduction ……...2
I.2.Description de la machine a courant continu (MCC) …….....2
I.3. Constitution ……..3
I.3.1.Le stator (l'inducteur)……3
I.3.2. Le rotor (l'induit) ……..4
I.3.3.Le collecteur et les balais ……...4
I.4 .Principe de fonctionnement …….5
I.5. Force contre electromotrice …….....5
I.6. Les differents types de moteurs ……...6
I.6.1. Les moteurs a inducteur a aimant permanent ……...…6
I.6.2. Les moteurs a inducteur bobine ……6
I.6.3.Modelisation d'un mcc a excitation independante ……..10
I.6.4.Fonction de transfert du moteur ……...…11
I.7. Variation de vitesse ……13
I.8.Bilan de puissance ……..14
I.9 .Les avantages et les inconvenients ……15
I.10. Utilisation de la machine a courant continu ……..…..16
I.11. Encodeur incremental ……..16
I.11.1.Principe de fonctionnement ……...18

I.11.2. Comptage du nombre d'impulsions……...18

I.12. Conclusion ……...20

Chapitre II : La présentation de PIC 16F877A

II.1.Introduction ……...21

II.2.Microcontroleur ……....21

II.3.Le choix du microcontroleur ……21

II.4.Les caracteristiques PIC 16F877A ……22

II.5. Description générale de PIC 16F877A ……23

II.5.1. Architecteur externe ……..23

II.5.2. Architecteur interne ……...24

II.6. Les timers ……26

II.7.Organisation de la mémoire du 16f877a……26

II.8.Les interruptions……..……..27

II.9.Les ports entrées/sortis ……..27

II.10.Le convertisseur analogique numérique ……..29

II.10.1.Déroulement d'une conversion……..……...29

II.10.2. Fonctionnement du convertisseur ……29

II.10.3.Configuration interne du module de convertisseur analogique/numérique ……..32

II.11. Module CCP ……...35

II.11.1.Mode compare ……..35

II.11.2.Mode capture ……36

II.11.3.Mode PWM……...36

II.12.L'USART ……37

II.13.Conclusion……37

Chapitre III : Régulation numérique et identification d'un système

III.1.Introduction ……..38

III.2.Principe général de la regulation ……..38

III.3.Les actions PID ……...…….39

II.3.1.L'Action proportionnelle ……40

III.3.2.L'action dérivée ……40

III.3.3.L'action intégrale ……..41

III.4. Régulation numérique ……42

III.4.1.FOnction de transfert discrete du PI ……43

III.4.2.Approximation rectangulaire ……44

III.4.3. Approximation rectangulaire devancée ……...45

III.4.4. Approximation trapézoïdale ……46

III.4.5.Fonction de transfert du PI avec l'approximation trapézoïdale ……47

III.5.Identification de processus et calcul des paramètres de correcteurs…….48

III.6.M2thodes de ziegler et Nichols (zn)…….51

III.7.Conclusion ……...53

Chapitre IV : Implémentation et simulation

IV.1. Introduction ……54

IV.1.1.Le mikroC PRO ……54

VI.1.2.Le proteus ISIS ……56

IV.1.2.1.Presentation de l'interface d'ISIS proteus V7……..56

IV.1.2.2.Les etape de travaille sur le proteus ……...56

IV.2.Implementation de la loi de commande ……...58

IV.2.1.Identification du système ……..58

IV.3.Régulation de vitesse d'un moteur à courant continu……...66

IV.3.1.Bloque de reference ……..68

IV.3.2.Bloque de regulation PI…….68

IV.3.2.1.Détermination de temps d'échantillonnage ……...70

IV.3.3.Bloque de driver ……74

IV.3.4.Bloque d'encodeur incrémentale ……75

IV.3.5.Bloque de LM331 ……...76

IV.4. Détermination des paramètres kp et ki du régulateur némurique ……...…77

IV.4.1.Méthode de Zeigler - Nichols ……...77

IV.4.2.IMplementation du regulateur PI dans µc ……78

IV.4.3.Implementation de la régulation sous simulink ……...82

IV.5.Conclusion……....83

Conclusion générale……..84

Introduction générale

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. [1]

Notre objectif dans ce mémoire est l'implémentation de régulateur PI, dans le microcontrôleur PIC 16F877A pour la commande de la vitesse d'un moteur à courant continu, pour ce la on a utilisé le logiciel MikroC pro pour la programmation du Pic et le logiciel Proteus ISIS pour la simulation. Notre travail est divisé en quatre chapitres.

Le premier chapitre est consacré à la constitution de la machine à courant continu et leurs principes de fonctionnements. Nous présentons les différents types du moteur à courant continu, et le domaine d'utilisation de cette machine, nous terminons ce chapitre par l'encodeur incrémental.

Dans le deuxième chapitre, nous allons présenté le microcontrôleur PIC 16F877A et son architecture (interne et externe), ses caractéristique et ses différents registres.

Pour le troisième chapitre on a fait une étude théorique sur le régulateur PI. Ensuite, on a présenté les principes de base de l'identification des modèles dynamiques de procédés, puis on a fait une régulation numérique avec le compensateur PI (méthode de Ziegler - Nichols).

Le quatrième chapitre est consacré à implémenté le régulateur numérique (PI) dans un microcontrôleur PIC16F877A, dont l'outil de programmation est le logiciel mikroC Pro et l'outil de test (simulation) est le logiciel Proteus ISIS.
Et enfin une conclusion générale.

Chapitre I :

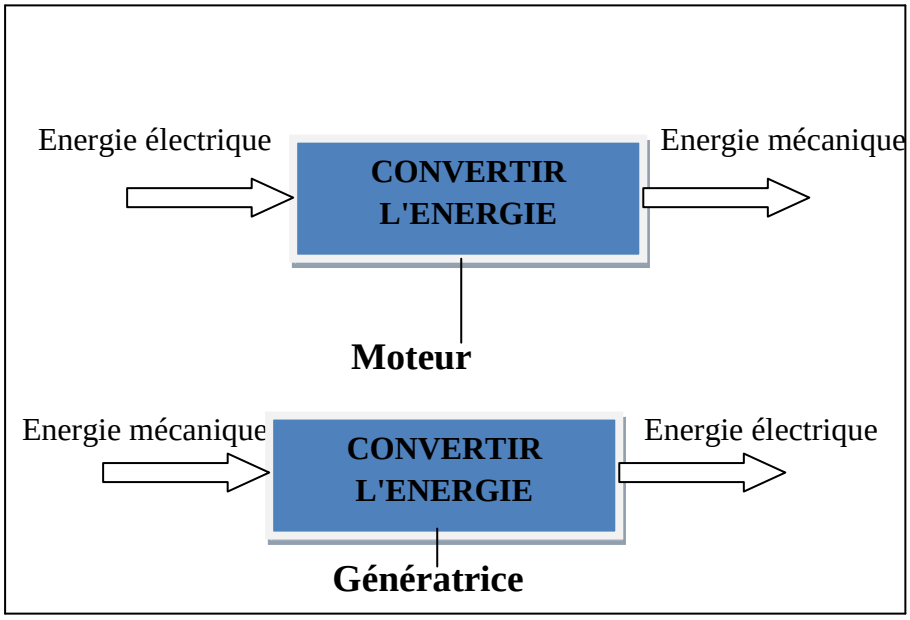
La machine à courant continu

I.1.Introduction :

Les moteurs à courant continu sont très utilisés dans les systèmes automatiques qui nécessitent une variation précise de la vitesse de rotation. [1][2] Dans ce chapitre, on présenté la constitution de la machine à courant continu et leur principe de fonctionnement. Il s'agit d'établir les différents types des moteurs à courant continu. Ensuit nous donnerons les avantage et les inconvénients. Enfin on fait une étude sur l'encodeur incrémentale.

I.2.Description de la Machine à Courant Continu (MCC):

Les machines à courant continu sont des convertisseurs électromécaniques d'énergie : Soit ils convertissent l'énergie électrique absorbée en énergie mécanique lorsqu'ils sont capables de fournir une puissance mécanique suffisante pour démarrer puis entraîner une charge en mouvement. On dit alors qu'ils ont un fonctionnement en moteur. Soit ils convertissent l'énergie mécanique reçue en énergie électrique lorsqu'ils subissent l'action d'une charge entraînante. On dit alors qu'ils ont un fonctionnement en générateur elle est représenté dans la figure suivante. [3]



Figure(I.1) : Modes de fonctionnement d'une MCC

I.3. Constitution :

La machine à courant continu est constituée de trois parties principales :

- l'inducteur.
- l'induit.
- le dispositif collecteur / balais.

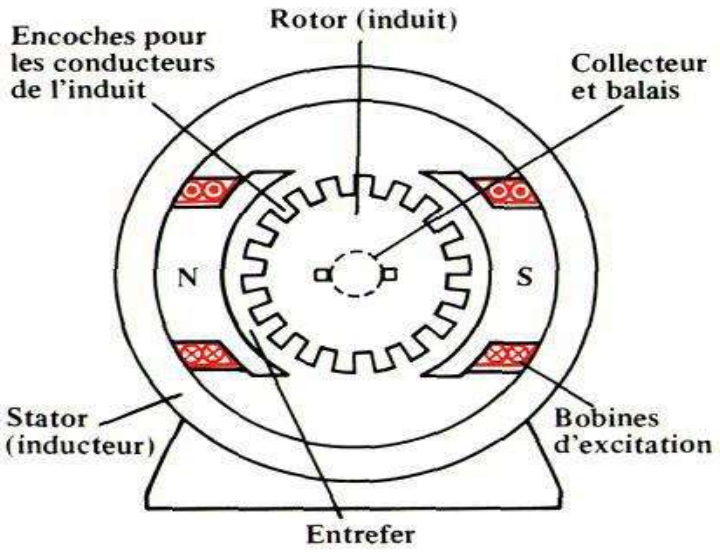


Figure (I.2): Constitution de la machine à courant continu. [4]

I.3.1.Le stator (l'inducteur):

Le stator est la partie fixe du moteur. Il est constitué d'un aimant permanent ou d'un électroaimant alimenté par le courant continu d'excitation (Ie). [1]



Figure (I.3): L'inducteur de la machine à courant continu**.** [5]

I.3.2. Le rotor (l'induit) :

Le rotor est la partie tournante du moteur. Il est constitué d'un enroulement de spires conductrices réunies en faisceaux, disposés de telle manière que lorsqu'un côté est soumis au pôle nord, l'autre est au pôle sud : les faisceaux sont tous reliés en série grâce aux lames du collecteur. [5]



Figure(I.4): Le rotor de la machine à courant continu. [5]

I.3.3.Le collecteur et les balais :

Le collecteur est un ensemble de lames de cuivre où sont reliées les extrémités du bobinage de l'induit. Les balais (ou charbons) sont situés au stator et frottent sur le collecteur en rotation. [5]

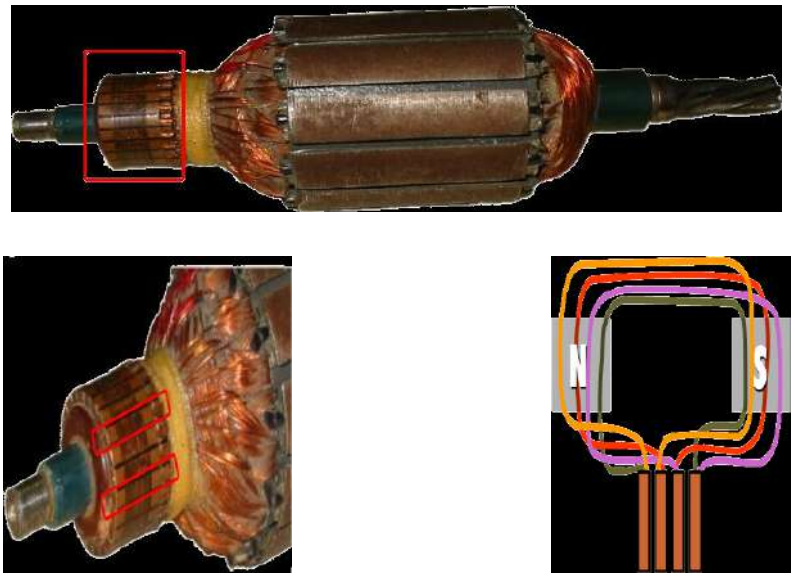
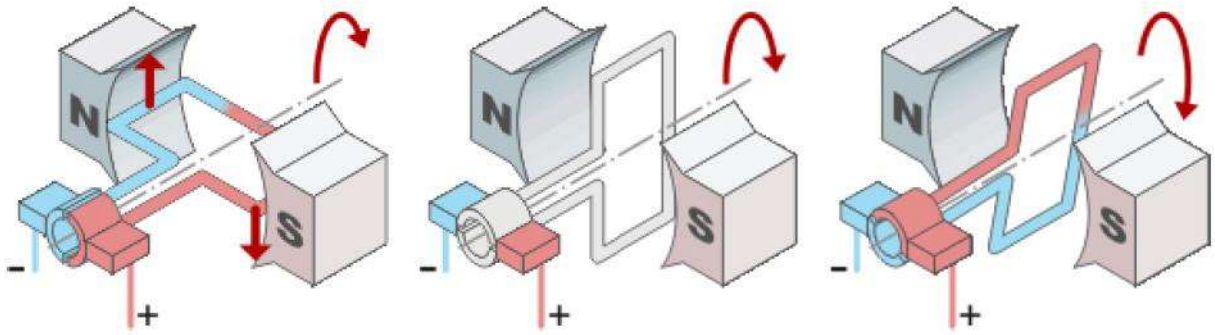


Figure (I.5) : Le dispositif collecteur / balais. [5]

I.4 .Principe de fonctionnement :

Un moteur à courant continu est constitué de deux parties électriques : le stator et le rotor, lorsqu'on alimente le moteur par un courant continu, il se crée une interaction magnétique (champ magnétique ou flux d'excitation) de direction Nord-Sud (comme le montre la Figure (I.6) qui met le moteur en mouvement. Lorsqu'on inverse le sens de la tension qui alimente le moteur, il tourne en sens inverse [6].

D'après la loi de Laplace, tout conducteur parcouru par un courant et placé dans un champ magnétique est soumis à une force, les conducteurs de l'induit placés de part et d'autre de l'axe des balais (ligne neutre) sont soumis à des forces F égales mais de sens opposé en créant un couple moteur, l'induit se met à tourner. [7]



Figure(I.6) : Principe de fonctionnement d'un moteur à courant continu. [7]

I.5. Force contre électromotrice :

Elle est donnée par la relation d'électrotechnique : La force électromotrice (E) est la tension produite par le rotor (l'induit) lors de sa rotation dans le flux magnétique produit par la partie fixe (l'inducteur). Elle dépend des éléments de construction de la machine. [1]

$$E = Pa.N.n .\Phi \quad (I.1)$$

P : nombre de paire de pôles de la machine.

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

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

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

Φ: flux sous un pole de la machine en Webers. Finalement :

$$E = K.\Omega.\Phi \quad (I.2)$$

$$K = \frac{P}{2\pi a} N \quad \text{(I.3)}$$

I.6. Les différents types de moteurs :

On distingue deux types de moteurs à courant continu :

I.6.1. Les moteurs à inducteur à aimant permanent :

Il n'y a pas de circuit inducteur, le flux inducteur est produit par un aimant permanent. Tous les moteurs à courant continu de faible puissance et les micromoteurs sont des moteurs à aimant permanent. Ils représentent maintenant la majorité des moteurs à courant continu. Ils sont très simples d'utilisation. [1]

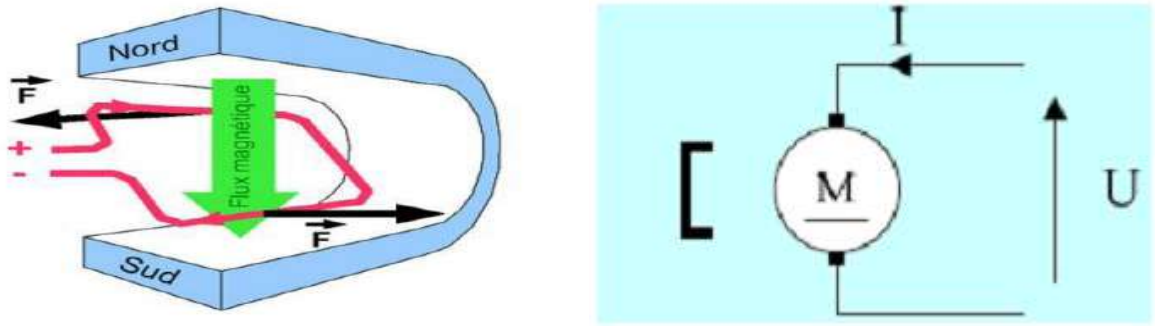


Figure (I.7): Moteur à aimant permanent. [1]

I.6.2. Les moteurs à inducteur bobiné :

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

- le moteur à excitation séparée.
- le moteur à excitation shunt.
- le moteur à excitation série.
- le moteur à excitation composée.

- **Le moteur à excitation séparée**

Dans un moteur à excitation séparée ou indépendante, le circuit d'excitation est séparé du circuit d'induit. Si l'inducteur est un aimant permanent, le flux (Φ) est constant. Si

l'inducteur est un électroaimant alimenté par une source de tension continue réglable, le flux (Φ) ne dépend que du courant dans l'inducteur appelé courant d'excitation (Ie).

Le courant crée un champ et une quantité de champ à travers une spire donne un flux. Si la tension (Ve) est constante, le courant d'excitation (Ie) est constant et le flux (Φ) est constant.

Dans ces conditions, la force contre électromotrice (E) ne dépend que de la fréquence (n) de rotation. Donc la fréquence de rotation (n) est proportionnelle à la tension (V) d'alimentation du moteur.

Compte tenu de la conservation de la puissance d'une machine (en négligeant les pertes), la puissance électrique absorbée est :

$$Pa = V.I \tag{I.4}$$

Et la puissance mécanique fournie est :

$$Pu = Tu.\Omega = Tu.2\pi.n \tag{I.5}$$

Si (V) est proportionnelle à (n) cela signifie que (Tu) est proportionnel à (I).

Donc le couple utile (Tu) est proportionnel à l'intensité du courant d'alimentation du moteur. Pour un moteur à excitation séparée, le flux (Φ) est constant donc la fréquence de rotation (n) est proportionnelle à la tension d'alimentation (V) :

$$n = \frac{E}{K\varphi} = \frac{V-R.I}{K\varphi} = K.V \tag{I.6}$$

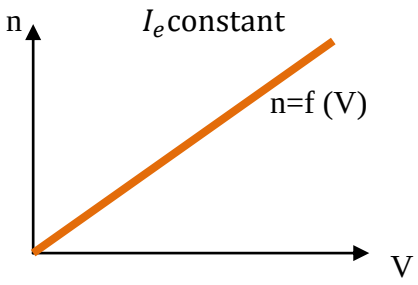


Figure (I.8): Caractéristiques d'un moteur à excitation séparée

Et le couple (Tu) est proportionnel à l'intensité du courant d'alimentation (I) :

$$T = E.I\Omega = K\Phi.I = K'.I \qquad (I.7)$$

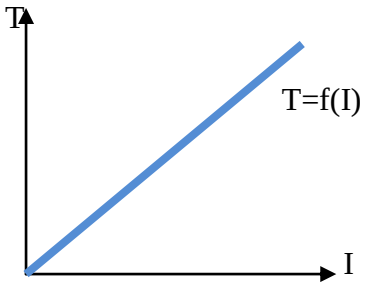


Figure (I.9): Le couple en fonction de courant d'alimentation

- **Le moteur à excitation série**

Dans un moteur à excitation série ou moteur série, le circuit d'excitation est en série avec le circuit d'induit.

Le flux n'est plus constant mais proportionnelle à (I)**.**

Dans ces conditions, la force contre électromotrice (E) dépend de la fréquence (n) de rotation et de l'intensité du courant (I)**.**

Donc la fréquence de rotation (n) est proportionnelle à la tension (V) et au courant (I) d'alimentation du moteur**.**

Compte tenu de la conservation de la puissance d'une machine (en négligeant les pertes), la puissance électrique absorbée est :

$$Pa = V.I \qquad (I.8)$$

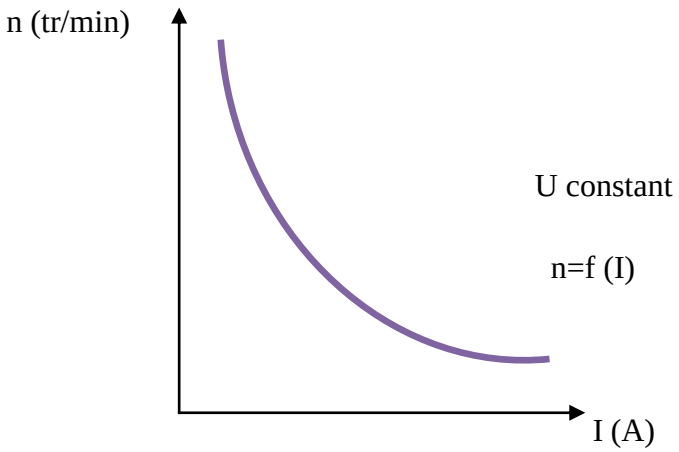
Et la puissance mécanique fournie est :

$$Pu = Tu.\Omega = Tu.2\pi.n \qquad (I.9)$$

Si (n) est proportionnelle à (V) et à (I) cela signifie que (Tu) est proportionnel au carré de l'intensité (I^2). Donc le couple (Tu) est proportionnel au carré de l'intensité du courant d'alimentation du moteur.

Pour un moteur à excitation série, le flux (Φ) n'est pas constant mais proportionnel à (I) donc la tension d'alimentation (V) est proportionnelle à la fréquence de rotation n et à l'intensité (I) :

$$E = k.n.N.\Phi = K.\boldsymbol{\Omega}.I \quad (I.10)$$



Figure(I.10): Caractéristique de vitesse d'un moteur série

Et le couple (T) est proportionnel au carré de l'intensité du courant d'alimentation (I)**.**

$$T = EI\ \Omega I^2 \quad (I.11)$$

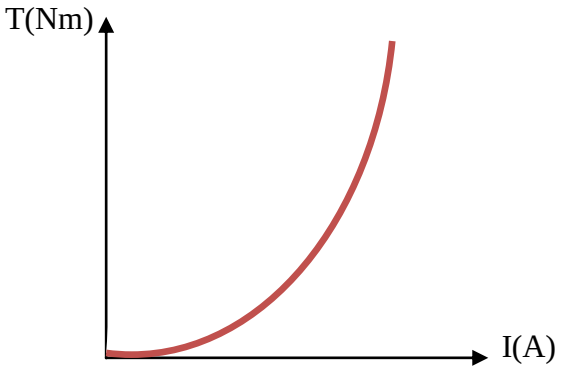


Figure (I.11): Caractéristique de couple d'un moteur série

- **Le moteur à excitation shunt**

La même source d'alimentation alimente l'induit et l'inducteur. L'inducteur est mis en parallèle avec l'induit. Les propriétés du moteur à excitation dérivation sont les mêmes que celle du moteur à excitation indépendante.

- **Le moteur à excitation composée**

Modes de branchement :

Courte dérivation : L'enroulement est shunté directement avec l'induit.

Long dérivation : L'enroulement est shunté en parallèle avec l'induit. [1]

I.6.3.Modélisation d'un MCC à excitation indépendante :

L'utilisateur d'un moteur s'intéresse en premier lieu à la caractéristique mécanique qui donne le couple produit en fonction de la vitesse. Il faut par ailleurs connaître le courant absorbé en fonction du couple demandé : c'est la caractéristique électromécanique, on utilise aussi la caractéristique de la vitesse.

On peut modéliser le moteur à courant continu par le biais d'équations électrique, électromécanique et mécanique.

Ces trois groupes d'équations nous permettront de mieux appréhender ce moteur dans son fonctionnement réel. Du coté électrique nous pouvons dire que le moteur à courant continu se définit par un circuit d'induit et un circuit inducteur; l'induit peut être vu comme une résistance et une inductance en série avec une source de tension commandée et proportionnelle à la vitesse.

Du coté mécanique, nous représentons la machine à courant continu par l'inertie de l'induit augmentée de celui de la charge entraînée. La figure suivante montre le schéma équivalent du moteur à courant continu [8].

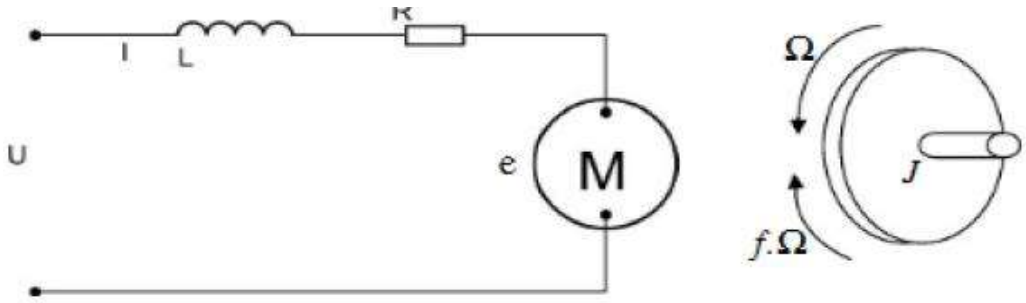


Figure (1.12): Schéma équivalent d'un MCC à excitation indépendante. [8]

D'après l'application des lois de Newton et Kirchhoff on a pu extraire les équations différentielles du premier ordre suivantes :

$$U(t) = Ri(t) + L\frac{di(t)}{dt} + e(t) \qquad (I.12)$$

Avec :

U(t) : tension aux bornes de l'induit.

R : résistance de l'induit du MCC.

L : inductance de l'induit du MCC.

e(t) : force électromotrice.

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

La vitesse de rotation du rotor est proportionnelle à la force électromotrice, sa formole est :

$$e(t) = k_e \Omega(t) \quad \text{(I.13)}$$

Ou:

K_e : La constante électrique du moteur (constante de vitesse).

$\Omega(t)$: la vitesse de rotation.

D'après le principe fondamental de la dynamique, nous avons:

$$J \frac{d\Omega}{dt} = C_u - C_r \quad \text{(I.14)}$$

$$Cu = K_c . i(t) - C_P \quad \text{(I.15)}$$

$$C_r = f . \Omega(t) \quad \text{(I.16)}$$

Avec :

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

f : Coefficient de frottement visqueux.

Cr : Couple résistant.

C_p: Couple de pertes.

K_c : Constante de couple.

C_u : Couple moteur généré

I.6.4.Fonction de transfert du moteur :

On utilise l'outil mathématique « transforme de Laplace » qui présente l'intérêt de transformer une équation différentielle en équation algébrique.

$$U(p) = Ri(p) + Lpi(p) + E \quad \text{(I.17)}$$

$$E = k_e \Omega(p) \quad \text{(I.18)}$$

$$JP\Omega(p) = C_u - C_r \quad \text{(I.19)}$$

On remplace E dans l'équation (I.17) et C_u et C_r par leurs valeurs on obtient:

$$U(p) = RI(p) + LPi(p) + k_e\Omega(p) \quad (I.20)$$

L'équation (I.14) devient:

$$JP\Omega(p) = k_c.i(p) - C_p - C_r \quad (I.21)$$

On déduit $\Omega(p)$:

$$\Omega(p) = \frac{k_c.i(p) - C_p}{f + JP} \quad (I.22)$$

On fait sortir l'expression de i(p) :

$$i(p) = \frac{f + JP}{k_c}\Omega(p) + \frac{C_p}{f + JP} \quad (I.23)$$

On inject i(p) dans (I.20)

$$U(p) = \Omega(p)\left[\frac{(R + LP)(f + JP)}{k_c} + k_e\right] + \frac{(R + LP)}{k_c}C_p \quad (I.24)$$

Supposant que la perturbation Cp (moment du couple de perte) est négligeable devant le moment du couple électromagnétique (Kc. i(t)), alors on prend Cp nulle pour simplifier le Système.

On aura:

$$U(p) = \Omega(p)\left[\frac{(R + LP)(f + JP)}{k_c} + k_e\right] \quad (I.25)$$

La fonction de transfert H(P) du système est le rapport entre la vitesse de rotation $\Omega(p)$ (qui est la sortie du système) et la tension entrant au ce dernier U(p).

$$H(p) = \frac{\Omega(p)}{U(p)} = \frac{k_c}{(R+LP)(f+JP)+k_ck_e} \quad (I.26)$$

Pour identifier le gain statique « A », la pulsation propre « W_0 » et le facteur d'amortissement « Ɛ » on réécrit la fonction de transfert sous forme canonique :

$$H(p) = \frac{H_c/(Rf + K_cK_e)}{\left[\frac{JL}{(Rf + K_cK_c}\right]p^2\left[\frac{(RJ + fL)}{Rf + K_cK_e}\right]p + 1} \quad (I.27)$$

On déduit à partir de l'équation (I.27) :

Le gain statique

$$A = \frac{K_c}{(Rf + K_c K_e)} \tag{I.28}$$

La pulsation propre :

$$W_0 = \sqrt{\frac{(Rf + K_c)}{Lf}} . \frac{2\pi}{360} \quad [rad/s] \tag{I.29}$$

Le facteur d'amortissement :

$$\mathcal{E} = W_0 \frac{Rj + fL}{2 * (Rf + K_c K_e)} \tag{I.30}$$

I.7. Variation de vitesse :

Pour faire varier la vitesse d'un moteur à courant continu, on peut agir sur la tension aux bornes de l'induit. La tension d'induit est directement proportionnelle à la vitesse de rotation. La puissance varie mais le couple reste constant. On dit alors que l'on fait de la variation de vitesse à couple constant.

$$\Omega = u\ Ra\ Ia / k \tag{I.31}$$

La figure(I.13) 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]

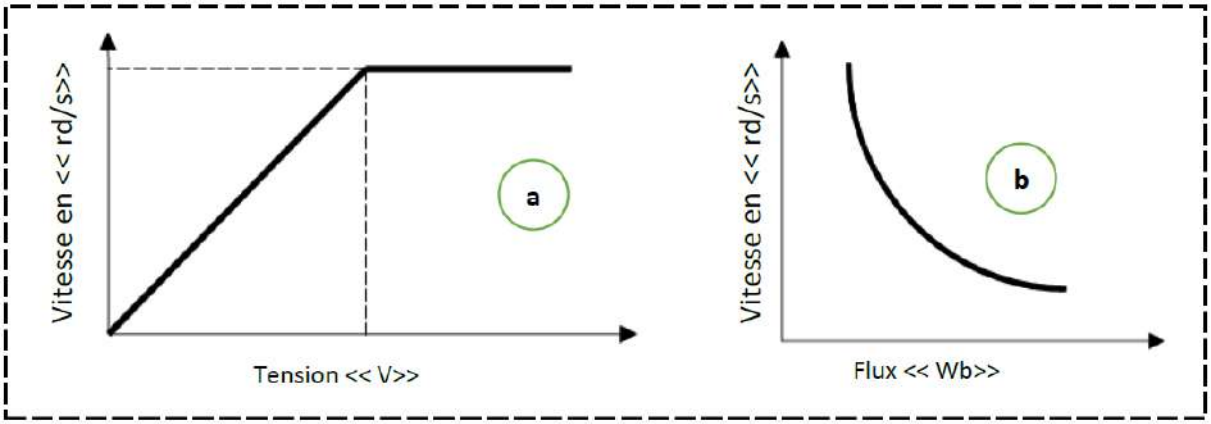
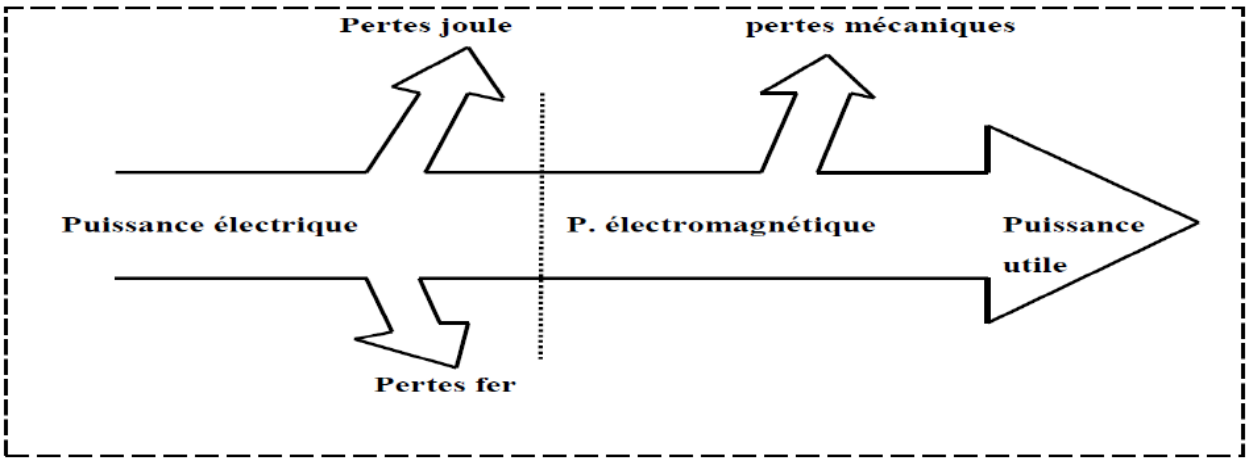


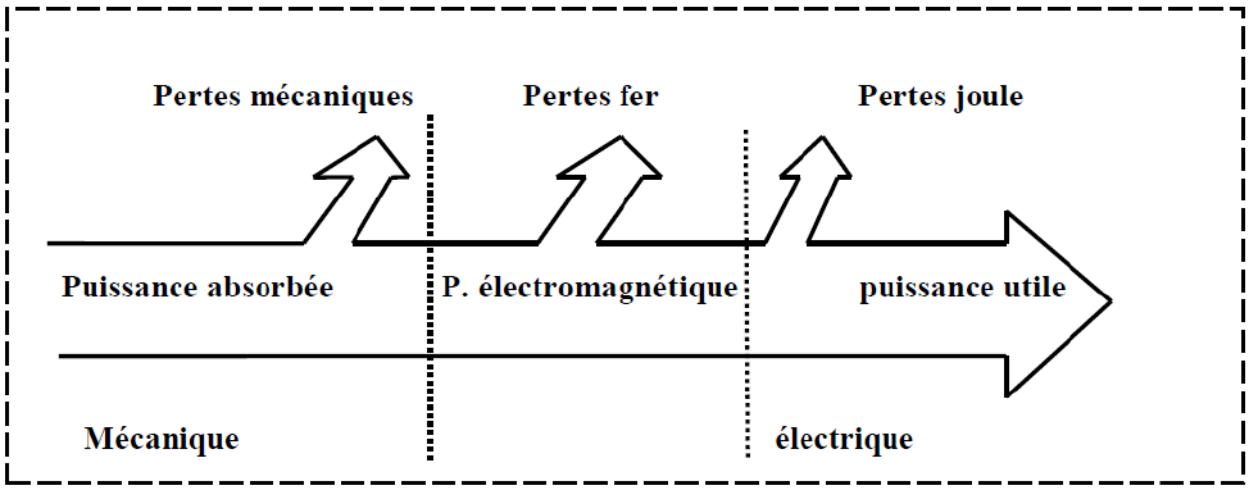
Figure (I.13) : a : Variation de la vitesse en fonction de la tension d'induit

b: Variation de la vitesse en fonction du flux d'inducteur. [2]

I.8.Bilan de puissance :



Figure(I.14): Bilan des puissances d'un moteur à courant continu. [2]



Figure(I.15) : Bilan des puissances d'un générateur à courant continu. [2]

- **Puissance utile sur l'arbre en sortie moteur :**

Pu = Pem – « pertes mécaniques + pertes magnétiques », en Watts

Pu : la puissance utile

Pem : les pertes électromagnétiques

- **Puissance absorbée en entrée :**

Pa = U.I

U : tension d'alimentation en Volt

I : courant absorbé en ligne,

Pa : puissance absorbé en Watts

❖ **Pertes :**

a. Pertes par effet joules « R.I»
b. pertes magnétiques
c. pertes mécaniques

❖ **Rendement :**

Les moteurs à courant continu consomment une partie de l'énergie absorbée pour leur fonctionnement. L'énergie mécanique fournie sera toujours plus petite que l'énergie électrique absorbée. Le rapport entre l'énergie fournie et l'énergie absorbée est le rendement. [2]

$$\eta = \text{Pu} / \text{Pa} \qquad \text{(I.32)}$$

I.9 .Les avantages et les inconvénients :

a. Les avantages

- Alimentation aisée dans les systèmes transportant leur réserve d'énergie (autonome) : pile ou batterie.
- La variation de fréquence de rotation est simple à réaliser.

b. Les inconvénients

Le principal problème de ces machines vient de la liaison entre les balais, ou charbons et le collecteur rotatif.

- Plus la vitesse de rotation est élevée, plus les balais doivent appuyer fort pour rester en contact et plus le frottement est important.
- Aux vitesses élevées les charbons doivent être remplacés très régulièrement.
- Le contact électrique imparfait à cause des arcs électriques, usant rapidement le commutateur et générant des parasites dans le circuit d'alimentation.

Pour des fonctionnements en moteur de petite puissance, ce problème peut être résolu grâce à la technologie du moteur à courant continu sans balai communément appelé moteur brushless. Un dispositif d'électronique de puissance remplace l'ensemble balai collecteur. La position du rotor est détectée par des capteurs à effet Hall et le courant est commuté par des transistors à effet de champ. [1]

I.10. Utilisation de la machine à courant continu :

Le moteur série est intéressant quand la charge impose d'avoir un gros couple, au démarrage et à faible vitesse de rotation. Le moteur sépare est particulièrement adapté aux entraînements de machines nécessitant des vitesses réglables (action sur la tension) et présentant un couple important en basse vitesse (machines outils).

- démarreur (automobile ...).
- moteur de traction (locomotive, métro ...).
- appareils de levage.
- Ventilateurs, pompes centrifuges, compresseurs, pompes à piston.
- Machines-outils à couple variable ou à mouvement alternatif (étaux-limeurs, raboteuses). [1]

I.11. Encodeur incrémental :

L'encodeur incrémental est un capteur de position extrêmement précis. Il est constitué de deux parties :

– un disque renferme des bandes opaques et transparentes sur sa périphérie.

– un module qui sert à acquérir les alternances (opaque/transparent) pour les produire en formes d'ondes analogiques.

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

– une diode luminescente LED (Light Emitting Diode).

– une lentille.[9]



Figure(I.16): Encodeur incrémental. [10]

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

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

Opaque (0 Volts) correspond à une période, transparente (5 Volts) sur la paire adjacente. Deux comparateurs reçoivent ces signaux et produisent les sorties finales pour les canaux A et B. Le canal A est en quadrature de phase avec le canal B. [9]

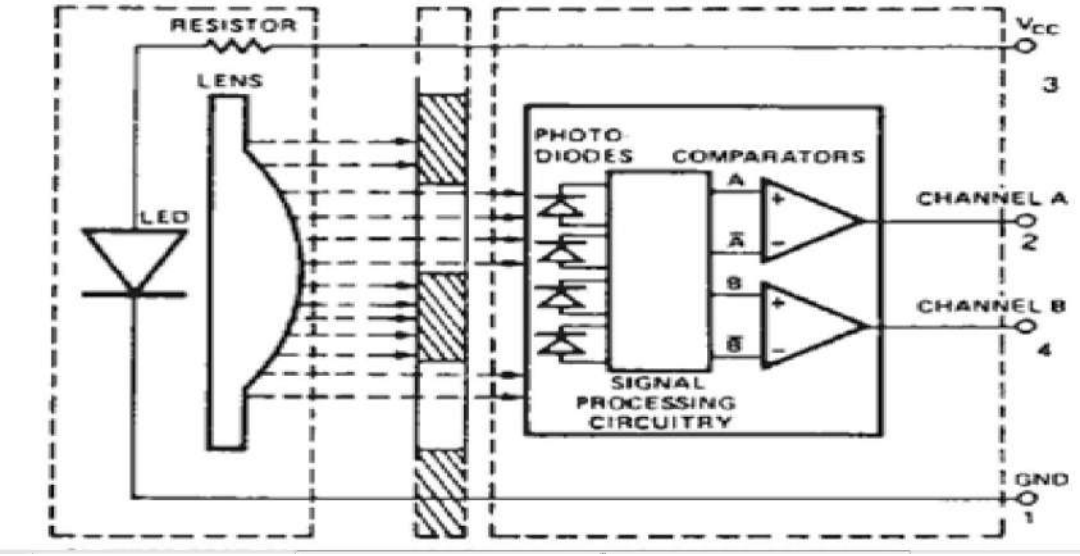


Figure (I.17): Circuit interne du module. [9]

I.11.1.Principe de fonctionnement :

La lumière émise par une diode électroluminescente passe à travers un disque en rotation solidaire de l'axe du moteur comportant des marques fixes transparentes ou opaques. Le signal délivré se présente alors comme une succession d'impulsions (en fait un signal carré), dont la périodicité est représentative de la vitesse.

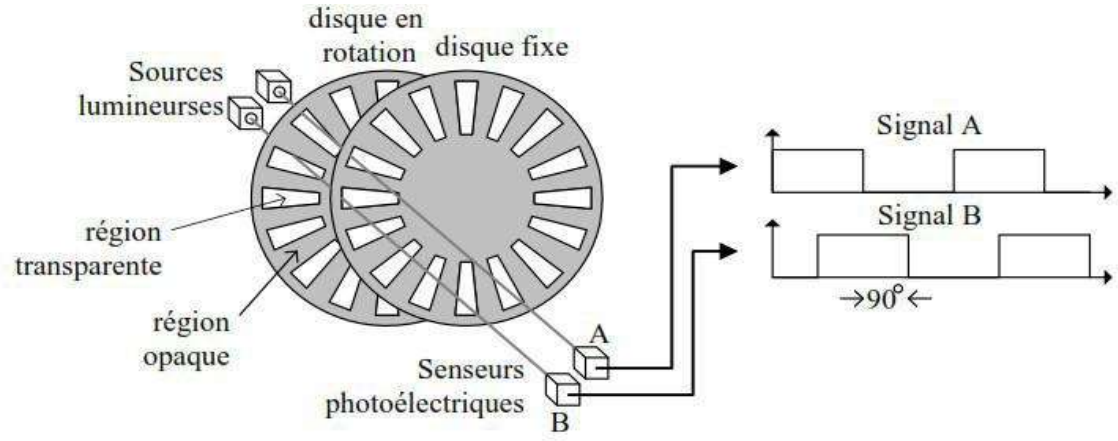


Figure (I.18): Encodeur incrémental. [2]

La mesure de la vitesse se fait simplement en comptant le nombre d'impulsions pendant un temps fixe. [9]

I.11.2. Comptage du nombre d'impulsions :

Il suffit pour cela d'observer les courbes ci-dessus, obtenues alors que le moteur tourne dans le sens positif. On constate que :

– Lorsque la voie A passe au niveau haut, la voie B est au niveau bas

– Lorsque la voie A passe au niveau bas, la voie B est au niveau haut

Quand le moteur tourne dans le sens positif, lors d'une interruption sur la voie A, les niveaux de A et B sont donc inversés.

Si le moteur tourne dans le sens négatif. On constate que :

– Lorsque la voie A passe au niveau haut, la voie B est au niveau haut.

– Lorsque la voie A passe au niveau bas, la voie B est au niveau bas.

Quand le moteur tourne dans le sens négatif, lors d'une interruption sur la voie A, les niveaux de A et B sont ont le même niveau.

Le désavantage de l'encodeur incrémentale est que la position mesurée est toujours relative à la position du rotor du moteur au moment où 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. [2]

Le gain de capteur est alors donné par :

$$k_E = 4N_t \frac{1tour}{2\pi rad} \tag{I.33}$$

Où :

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

Pour les positions rotatives :

Où :

$$\theta = \frac{nombre_de_fronts}{xN} * 360 \tag{I.34}$$

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

x = type d'encodage

I.12. Conclusion :

Ce chapitre à permis de rappeler les différents éléments qui constituent une machine à courant continu et leur principe de fonctionnement. Après notre étude nous avons constaté que les moteurs à excitation séparée et a aimant périmant sont les plus adaptée pour la variation de vitesse.

Chapitre II :

La présentation de PIC 16F877A

II.1.Introduction :

Ce chapitre est consacré à la présentation du microcontrôleur tout en illustrant ces différentes caractéristiques afin de mieux l'exploiter.

On aura à définir l'unité de contrôle et tout ce qui l'englobe de part et d'autre de sa structure. La partie la plus importante sera celle qui traite les différents ports d'entrée/sortie. Ainsi que le convertisseur et le module CCP.

II.2.Microcontrôleur :

Les microcontrôleurs améliorent l'intégration et le coût (lié à la conception et à la réalisation) d'un système à base de microprocesseur en rassemblant ces éléments essentiels dans un seul circuit intégré. Un microcontrôleur est donc un composant autonome, capable d'exécuter le programme contenu dans sa mémoire morte dès qu'il est mis sous tension. Selon les modèles et les conditions de fonctionnement, les microcontrôleurs peuvent avoir besoin de quelques composants externes (quartz, quelques condensateurs, parfois une ROM), mais ceci reste très limité. [11]

II.3.Le choix du microcontrôleur :

Le choix d'un microcontrôleur est primordial car c'est de que dépendent en grande partie des performances, la taille, la facilité d'utilisation et le prix du montage.

Le PIC 16F877A possède en plus des instructions très puissantes donc un programme à développer réduit, surtout lorsqu'on utilise le logiciel de programmation micro pascal qui possède un nombre important de procédures et fonctions prédéfinit et dédié au PIC 16F877A. En fait la cause principale du choix de ce type de microcontrôleur est qu'il dispose de l'option du convertisseur A/D pour satisfaire coté acquisition, aussi la possibilité de l'adaptation au protocole I2C et la liaison RS232 mais aussi le nombre de ports d'entrées /sorties nous convient. [12]

II.4.Les caractéristiques PIC 16F877A :

Nous avons utilisé le microcontrôleur PIC 16F877A pour contrôler la vitesse du moteur, ce microcontrôleur c'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 8 ko « 1mot=14bit ».

- 33 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 : Module de Largeur d'Impulsion.
- 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. [2]

II.5. Description générale de PIC 16F877A :

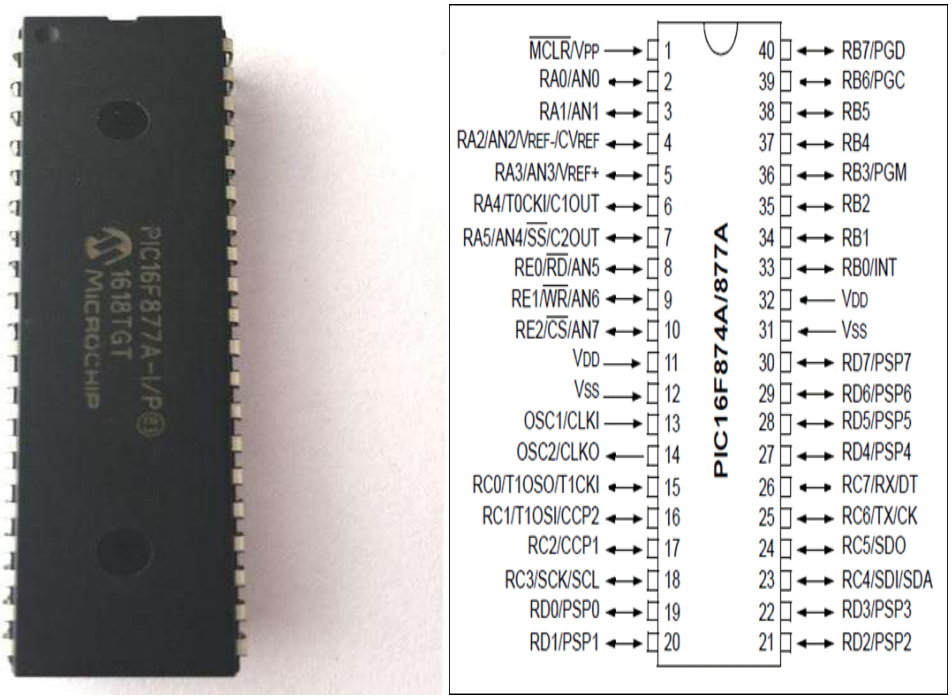


Figure (II.1) : boitier du PIC 16F877A. [13]

II.5.1. Architecteur externe :

Le microcontrôleur PIC 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. [3]

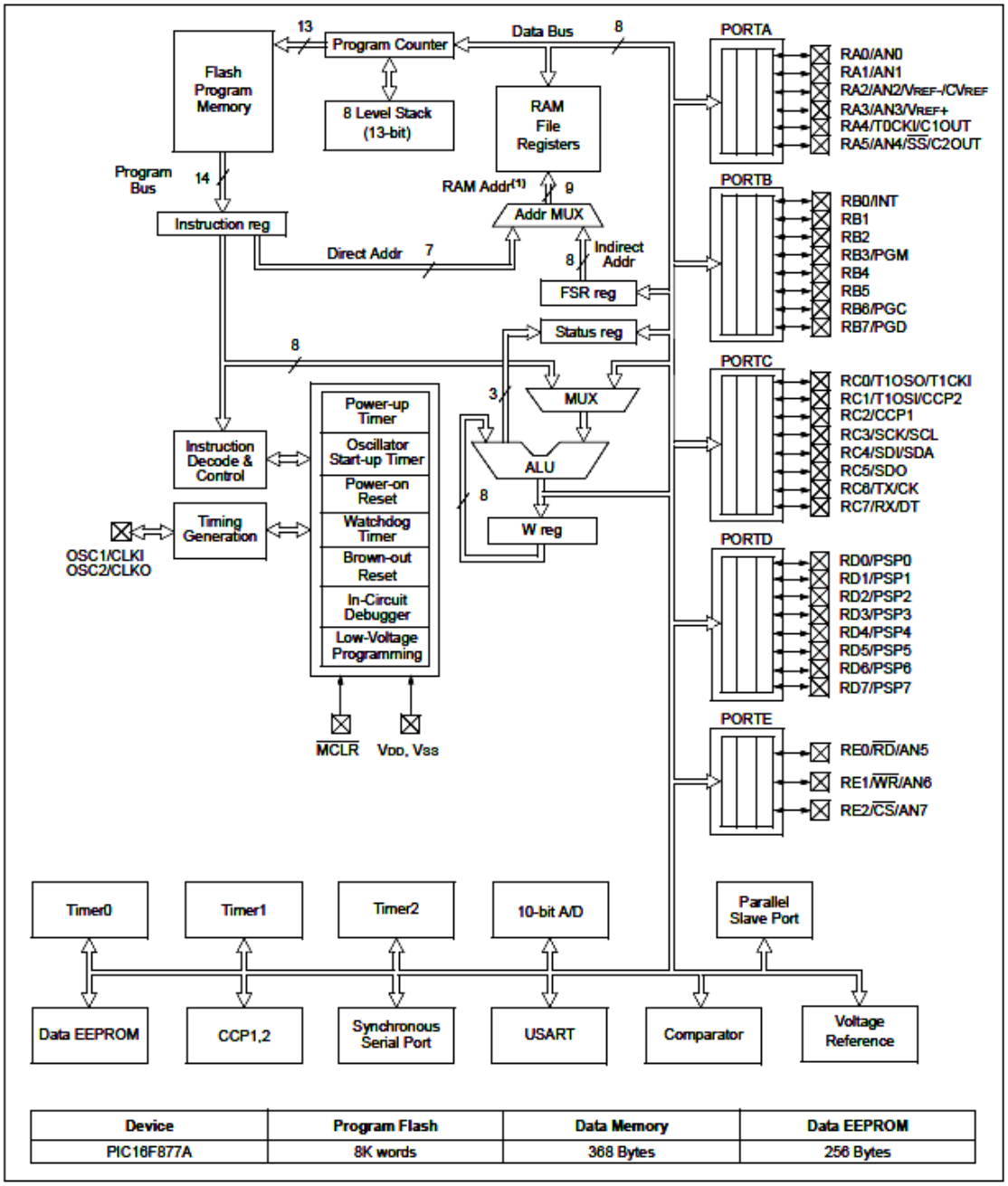
Tableau (II.1): Broches du PIC16F877A

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	Broch recevant quartz externe

II.5.2. Architecteur interne :

Cette architecture nous présente les principaux organes qui composent le µ c, la figure (II.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]



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

Figure (II.2) : Architecture interne du PIC 16F877A. [13]

II.6. Les Timers :

Notre pic possède 3 timers qui sont :

- Le Timer0 (8bits) : il peut être incrémenté par des impulsions extérieures via la broche (TOCKI/RA4) ou par l'horloge interne (Fosc/4).
- Le Timer1 (16 bits) : il peut être incrémenté soit par l'horloge interne par des impulsions sur la broche T1CKI/RC0 ou par un oscillateur (RC ou quartz) connecté sur les broches T1OSO/RC0 et T1OSI/RC1.
- Le Timer2 (8bits) : il est incrémenté par l'horloge interne, celle peut être pré divisée.

Tous ces timers peuvent déclencher une interruption interne, s'ils ont été autorisés.[14]

II.7.Organisation de la mémoire du 16F877A:

La mémoire du 16F877A est divisée en 3 parties :

- 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.

- La mémoire EEPROM :

La mémoire EEPROM (Electrical Ecrasable Programmable Read Only Memory), est constituée de 256 octets que nous pouvons lire et écrire depuis notre programme. Ces octets sont conservés après une coupure de courant et sont très utiles pour conserver des paramètres semi-permanents.

- La mémoire RAM et organisation :

La mémoire RAM est celle que nous allons sans cesse utiliser. 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 :

1) 80 octets en banque 0, adresses 0*20 à 0*6F
2) 80 octets en banque 1, adresses 0*A0 à 0*EF
3) 96 octets en banque 2, adresses 0*110 à 0*16F
4) 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. [15]

II.8.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. [14]

II.9.Les ports entrées/sortis :

Le PIC 16F877A dispose de 5 ports entrée/sortis et voici par la suite les particularités de chaque port :

Le port A :

Le 16F877A dispose de 5 canaux d'entrées analogiques. Nous pouvons donc échantillonner successivement jusque 5 signaux différents avec ce composant. Les pins utilisés sont les pins AN0 à AN4 (qui sont en fait les dénominations analogiques des pins RA0 à RA3 + RA5).

On peut noter également que les pins Anx sont des pins d'entées. Il n'est donc pas question d'espérer leur faire sortir une tension analogique. Ceci nécessairement un convertisseur numérique/analogique dont n'est pas pourvu notre PIC.

Le port B :

Hors de sa fonction principale autant que ports d'entrées/sorties, on note la pin RB0 qui en configuration d'entrée est de type « trigger de Schmitt » quand elle est utilisée en mode interruption « INT » ; la lecture simple de RB0 se fait d'une façon tout à fait classique, en entrée de type TTL. Encore il y a (RB3-RB6-RB7) qui peut servir dans la programmation en cas d'absence de programmateur commercial.

Le port C :

Le port C désigné par PORTC est un port bidirectionnel de 8 bits (RC0 à RC7). Toutes les broches sont compatibles TTL.

- La configuration de direction se fait à l'aide du registre TRISC, positionner un bit de TRISC à "1" configure la broche correspondante de PORTC en entrée et inversement. Au départ toutes les broches sont configurées en entrée.
- Toutes les broches du port C peuvent être utilisées soit comme E/S normales soit comme des broches d'accès à différents modules comme le TIMER1.
- Pour l'utilisation d'une broche du port C comme E/S normale, il faut s'assurer qu'elle n'a pas été affectée aux autres modules. Par exemple, si TIMER1 est valide, il peut utiliser les broches RC0 et RC1 selon sa configuration. **[11]**

Le port D :

Ce port fonctionne de façon identique aux autres, dans son mode de fonctionnement général. 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:

Ce port n'est présent que sur les PIC 16F877A. Il ne comporte que 3 pins RE0à 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 utilise comme port I/O ou comme port analogique. [15]

II.10.Le convertisseur analogique numérique :

Le CAN est un périphérique intégré destiné à mesurer une tension et la convertir en nombre binaire qui pourra être utilisé par un programme.

Le convertisseur analogique numérique est à approximations successives et il possède une résolution de 10 bits. Il est composé de :

- Un multiplexeur analogique **5** voies (PIC16F876) ou **8** voies (PIC16F877).
- Un échantillonneur bloqueur.
- Un Convertisseur Analogique Numérique de **10** bits. [16]

II.10.1.Déroulement d'une conversion:

La conversion d'un signal analogique numérique passe par deux phases :

- Echantillonnage blocage (simple and hold). Cette opération consiste à connecter l'entrée à convertir à un condensateur interne, qui va se charger à travers une résistance interne jusqu'à la tension appliquée : c'est le temps d'acquisition.
- Une fois le condensateur est chargé, déconnecté la tension appliquée en ouvrant l'interrupteur S, pour procéder à la phase ce conversion.

Pour résumer, les étapes de conversion sont :

- Configurer la conversion A/N
- Attendre TACQ.
- Lancer la conversion.
- Attendre conversion finie.
- Lire la valeur convertie. [17]

II.10.2. Fonctionnement du convertisseur :

La conversion se passe en 2 temps :

- 1^{er} temps : le signal à convertir est appliqué sur l'entrée à convertir, ce signal doit être présent au moins pendant le temps Tacq (temps d'acquisition environ 20µS pour 5V).
- $2^{ème}$ temps : la conversion, approximations successives.

Le temps de conversion minimum est de 12 TAD (TAD c'est le temps de conversion dépendant de l'horloge interne, typiquement 1,6µS).

Une conversion commence toujours par la mise à 1 du bit GO/DONE du registre ADCON0. Lorsque la conversion est terminée se bit repasse à 0.

Donc pour pouvoir lire le résultat dans les registres ADRESL et ADRESH il suffit d'attendre que le bit GO/DONE passe à 0. [16]

1. **Temps d'acquisition :**

Tacq = Tamp + Tc + Tcoff

Tamp : temps de réaction des circuits

Tc : temps de charge du condensateur

Tcoff : temps qui dépend du coefficient de température.

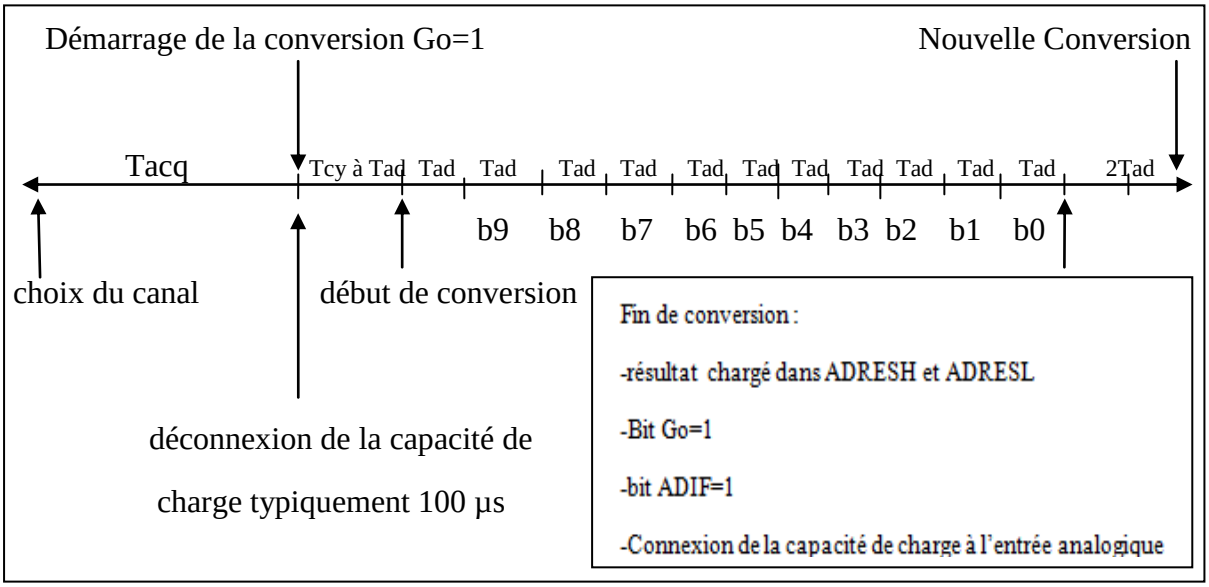
Le temps de réaction est fixé à 2µs ; de même Tcoff est limité à une valeur maximale de 1,25µs.

Dans les conditions normale et pour une tension de 5V, Tc = 16,47µs.

Le temps d'acquisition Tacq = 19,72µs |20µs [17]

2. **Temps de conversion :**

Le temps de conversion est égal à 12 TAD



Figure(II.3) : cycle de conversion. [16]

TAD est le temps de conversion d'un bit, il dépend de la fréquence du quartz et du pré diviseur (div) choisi :

$$TAD = div * \frac{1}{fosc}. \quad \text{(II.1)}$$

Le choix de div doit être ajusté pour que T_{AD} soit ≥ à 1,6 µs.

La valeur résultante N de la conversion ADRSH: ADRESL est égale à :

$$N = \frac{(Va - Vref-)}{Q} \quad \text{(II.2)}$$

Avec :

Va: tension analogique à numériser.

N: valeur numérique sur N bits.

Q = pas de quantification

$$Q = \frac{Vref_+ - Vref_-}{1024} \quad \text{(II.3)}$$

Mais avant de réaliser une conversion il faut définir la configuration du convertisseur :

- Le nombre d'entrées analogiques.
- Le nombre d'entrées logiques.
- Le type de tension de référence :
- Interne $Vref = VDD - VSS$
- Externe, soit : $Vref = Vref_+ - VSS$ ou $Vref = Vref_+ - Vref_-$

Le module de conversion utilise 4 registres disposés comme suit :

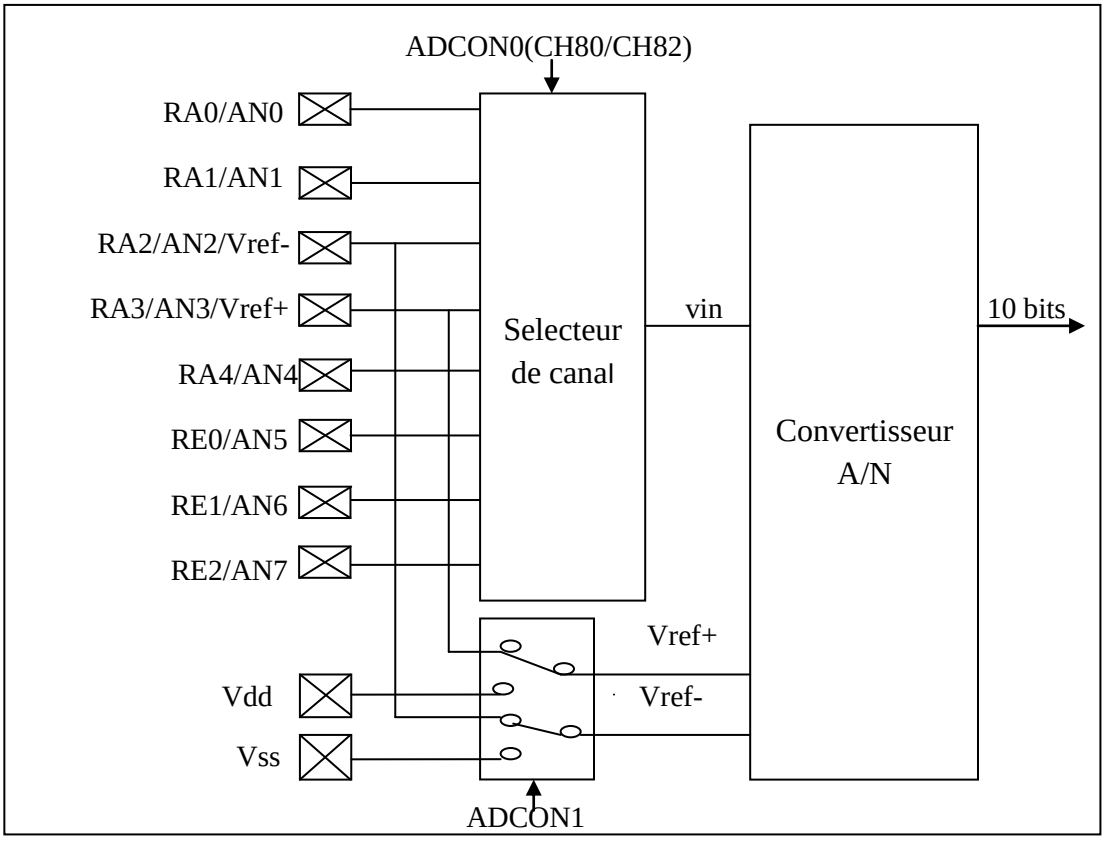
*Registre de Résultat High (ADRESH)

*Registre de Résultat Low (ADRESL)

*Registre0 de Contrôle (ADCON0)

*Registre1 de Contrôle (ADCON1). [16]

II.10.3.Configuration interne du module de convertisseur analogique/numérique :



Figure(II.4): module du convertisseur. [13]

On voit très bien sur ce schéma que les pins AN2 et AN3 servent selon la position du sélecteur d'entrée analogique ou de tension de référence. Le sélecteur de canal permet de sélectionner lequel des 8 canaux va être appliqué au convertisseur analogique/digital.

On voit que la sélection de la source des tensions de référence dépend de bits du registre ADCON1, tandis que le canal sélectionné pour être numérisé dépend d'ADCON0. C'est le registre ADCON1 qui détermine si ce port sera utilisé comme port I/O ou comme port analogique. Nous en aurons besoin dans la programmation. [15]

- **Les registres ADRESL et ADRESH :**

On attire votre attention sur le fait 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, Microchip 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**

La justification à droite sera principalement utilisée lorsque nous avons besoin de l'intégralité des 10 bits de résultat, tandis que la justification à gauche est très pratique lorsque 8 bits vous suffisent. Dans ce cas, les 2 bits de poids faibles se trouvent isolés dans ADRESL, il suffit donc de ne pas en tenir compte. Cette approche est destinée à nous épargner des décalages de résultats.[doc] Le choix de la méthode s'effectue à l'aide du bit 7 d'ADCON1.[15]

- **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 :

Tableau(II.2): Bits de registre ADCON1.

B7	ADFM	A/D result format select
B6	INUTILISTE	lu comme « 0 »
B5	INUTILISTE	lu comme « 0 »
B4	INUTILISTE	lu comme « 0 »
B3	PCFG3	Port configuration control bit3
B2	PCFG2	Port configuration control bit2
B1	PCFG1	Port configuration control bit1
B0	PCFG0	Port configuration control bit0

Le bit ADFM permet de déterminer si le résultat de la conversion sera justifié à droite (1) ou à gauche (0). Nous trouvons dans ce registre les 4 bits de configuration des pins liés au convertisseur analogique/numérique. Ces bits nous permettent donc de déterminer le rôle de chaque pin.

Comme nous avons 16 combinaisons possibles, nous aurons autant de possibilités de configuration (en fait, vous verrez que nous n'en avons que15). [15]

- **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. [16]

Tableau (II.3) : Bits de registre ADCON0.

B7	ADCS1	A/D conversion clock select bit
B6	ADCS0	A/D conversion clock select bit
B5	CHS2	Analog Channel select bit2
B4	CHS1	Analog Channel select bit1
B3	CHS0	Analog Channel select bit0
B2	GO/DONE	A/D conversion clock status bit
B1	INUTILISE	lu comme « 0 »
B0	ADON	A/D ON bit

II.11. Module CCP:

Le microcontrôleur PIC16F877A possède deux modules CCP (Capture / Compare / PWM) indépendants, nommés CCP1 et CCP2. Chaque module CCP possède deux résistances de 8 bits (CCPxH, CCPxL) pouvant être utilisées comme:

- Registre de capture 16 bits
- Registre de comparaison 16 bits
- Registre PWM 10 bits.

II.11.1.Mode compare :

Les deux modules CCP étant identiques on ne décrira que le module 1.

Les 16 bits des registres CCPR1 (CCPR1H et CCPR1L) sont constamment comparés avec le valeur sur 16 bits des registres du Timer 1 (TMR1H et TMR1L). Quand il y a égalité, la broche préalablement programmée en sortie PC2, passe soit à "1" soit à "0" suivant la configuration des 4 bits CCP1M du registre CCP1CON. Au même instant le Flag CCP1IF est mis à "1".

En mode Compare, les événements spéciaux générés quand il y a égalité sont:

- Pour CCP1: reset du Timer 1.
- Pour CCP2 : reset du Timer 1 et démarrage d'une conversion A/D.

Dans ce cas la broche de sortie n'est pas affectée, mais le Flag CCP1IF est mis à "1". Il est rappelé que ce Flag doit être remis à "0" par soft.

II.11.2.Mode capture :

Quand un événement extérieur apparaît sur la broche préalablement programmée en entrée PC2, la valeur du 16 bits des registres du Timer 1 (TMR1L et TMR1H) est recopiée dans les registres CCPR1 (CCPRIH et CCPR1L). Cet événement est programmable par les 4 bits CCP1M du registre CCP1CON. La capture peut avoir lieu à chaque front descendant, à chaque front montant, tous les 4 ou tous les 16 fronts montants.

Quand la capture a eu lieu, le flag CCP1IF est mis à "1". Ce bit doit être remis à "0" par soft. Si une nouvelle capture survient alors que la valeur dans CCPR1 n'a pas été lue, l'ancienne valeur est perdue.

Les fonctions de Capture et de Compare sur le Timer 1 par les modules CCP1 et CCP2 peuvent générer une interruption quand le Flag CCP1IF passe à "1" si le bit d'autorisation CCP1IE du registre PIE1 est mis à "1".

II.11.3.Mode PWM:

Le PIC a un module de PWM avec une résolution de 10 bits. Les bits 8-Msb sont stockés dans CCPRxL et les 2 bits restants dans le registre CCPxCON. Les tableaux ci-dessous montrent le module PWM de PIC.

Tableau(II.4): module PWM de PIC

Chaîne PWM	Port Pin	Registre de contrôle	Registre du cycle de service	Registre de période
PWM1	PC.2	CCP1CON	CCPR1L	PR2
PWM2	PC.1	CCP2CON	CCPR2L	PR2

Un signal PWM est caractérisé par une période, et un temps de travail ou le signal est à "1". Ce temps est appelé DUTY CYCLE.

II.12.L'USART :

L'USART (Universal Synchronous Asynchronous Receiver Transmitter) est l'un des deux modules de communication série dont dispose le PIC 16F876/877. L'USART peut être configuré comme système de communication asynchrone full duplex ou comme système synchrone half duplex (non étudié). La communication se fait sur les deux broches RC6/TX et RC7/RX qui doivent être configurés toutes les deux en entrée par TRISC. [18]

II.13.Conclusion:

Partant d'une présentation générale sur les microcontrôleurs, nous avons ensuite défini la famille de PIC et plus particulièrement le 16F877A.

En conclusion dans ce chapitre nous pouvons dire que le microcontrôleur 16F877A peut bien jouer le rôle d'une unité de contrôle pour notre système.

Chapitre III :

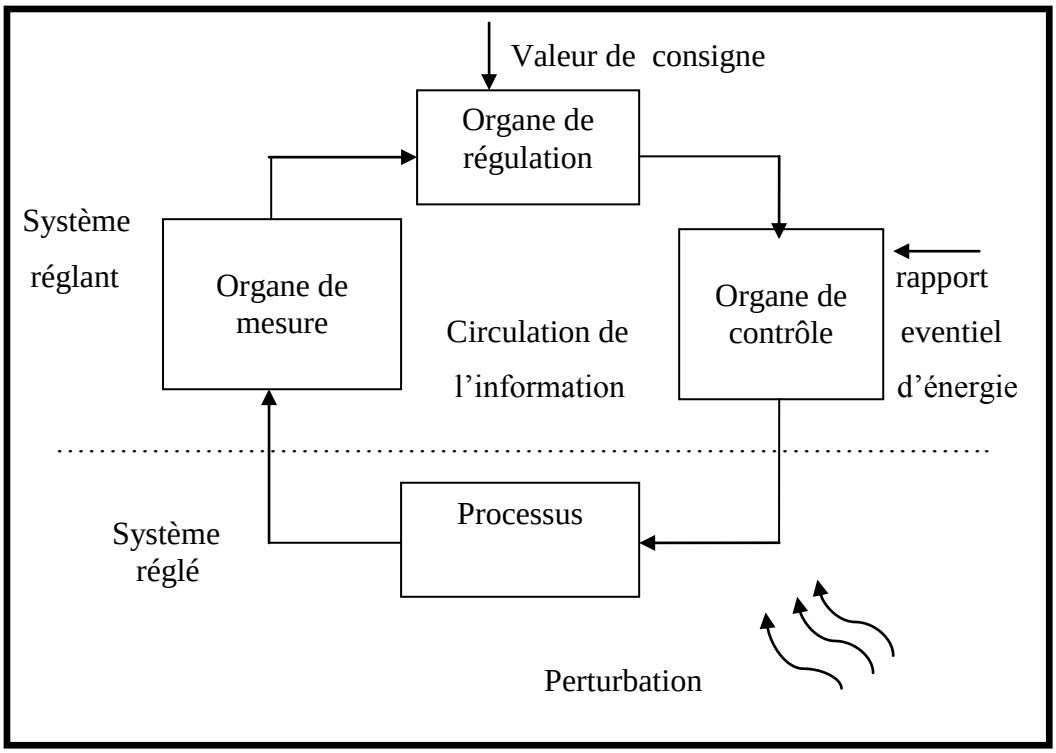
La régulation numérique et identification

III.1.Introduction :

Dans ce chapitre on a basé sur l'étude théorique de régulateur PI. Ensuite, nous allons présenter les principes de base de l'identification des modèles dynamiques de procédés. Puis on a fait une régulation numérique avec le compensateur PI (méthode de Ziegler -Nicholas).

III.2.Principe générale de la régulation :

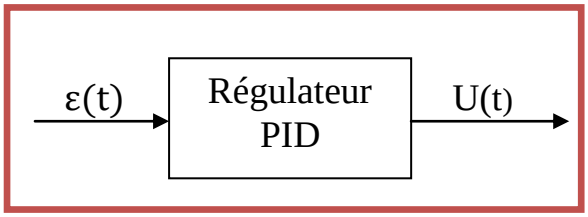
Dans la plupart des appareils et installations industrielles, tertiaires et mêmes domestiques, il est nécessaire de maintenir des grandeurs physiques à des valeurs déterminées, en dépit des variations externes ou internes influant sur ces grandeurs. [2]



Figure(III.1) : Schéma de principe d'une chaine de régulation.

III.3.Les actions PID :

En pratique, à une catégorie donnée de systèmes à asservir correspond un type de correcteur adopte. Pour effectuer un choix judicieux, il faut connaitre les effets des différentes actions : proportionnelle, intégrale et dérivée.



Figure(III.2) : Schéma synoptique d'un régulateur PID.

Un régulateur PID est obtenu par l'association de ces trois actions et il remplit essentiellement les trois fonctions suivantes :

- Il fournit un signal de commande en tenant compte de l'évolution du signal de sortie par rapport à la consigne.
- Il élimine l'erreur statique grâce au terme intégrateur
- Il anticipe les variations de la sortie grâce au terme dérivateur.

La commande U(t) donnée par le régulateur PID, dans sa forme classique est décrite par :

$$U(t) = Kp[\varepsilon(t) + \frac{1}{T_i}\int_0^t \varepsilon(\tau)d(\tau)d\tau + Td\frac{d\varepsilon(t)}{dt} \qquad \text{(III. 1)}$$

Elle est la somme détroits termes :

- Le terme proportionnel :

$$P = Kp\varepsilon(t) \qquad \text{(III. 2)}$$

- Le terme intégral :

$$I = Kp\frac{1}{T_i}\int_0^t E(\tau)d\tau \qquad \text{(III. 3)}$$

- Le terme dérivatif :

$$D = KpTd\frac{d\varepsilon(t)}{dt} \qquad \text{(III. 4)}$$

Les paramètres du régulateur PID sont le gain proportionnel Kp, le temps intégral Ti et le temps dérivatif Td, les temps étant exprimés en secondes. [1]

III.3 .1.L'action proportionnelle :

La sortie U(t) du régulateur proportionnel est donnée en fonction de son entrée Ɛ(t) qui représente l'écart entre la consigne et la mesure par la relation :

$$u(t) = Kp\varepsilon(t) \qquad \text{(III. 5)}$$

Pour le cas discret, cette relation reste la même telle que :

$$u(t) = Kp\varepsilon(k) \qquad \text{(III. 6)}$$

Le rôle de l'action proportionnelle est de minimiser l'écart Ɛ entre la consigne et la mesure et elle réduit le temps de montée et le temps de réponse. On constate qu'une augmentant du gain Kp du régulateur entraine une diminution de l'erreur statique et permet d'accélérer le comportement global de la boucle fermée. On serait tenter de prendre des Valeurs de gain élevées pour accélérer la réponse du procédé, mais on est limité par la stabilité de la boucle fermée. En effet, une valeur Trop élevée du gain augmente l'instabilité du système et donne lieu à des oscillations.

III.3.2.L'action dérivée :

Elle est une action qui tient compte de la vitesse de variation de l'écart entre la consigne et la mesure, elle joue aussi un rôle stabilisateur, contrairement à l'action intégrale. En effet, elle délivre une sortie variant proportionnellement à la vitesse de variation de l'écart.

$$(t) = T_d \frac{d_\varepsilon(t)}{dt} \qquad \text{(III. 7)}$$

Avec Td le dosage de l'action dérivée, exprimée minutes ou en secondes. L'action dérivée va ainsi intervenir uniquement sur la variation de l'erreur ce qu'augmente la rapidité du système (diminution des temps de réponses). L'action dérivée permet aussi d'augmenter la Stabilité du système par apport de phase $(+ \frac{\pi}{2}$ ce qui augmente la marge de phase). L'annulation de cette action en régime statique impose donc de ne jamais d'utiliser seule : L'action dérivée n'exerce qu'un complément à l'action proportionnelle.

En pratique, il est souhaitable de limiter l'action dérivée afin de ne pas amplifier les bruits haute fréquence et de limiter l'amplitude des impulsions dues aux discontinuités de l'écart.

Lorsque la période d'échantillonnage Te est petite, la différence vers l'arrière (Approximation d'Euler rétrograde) nous permettons d'approcher la dérivée d'un signal à temps continu par :

$$\frac{df(t)}{dt} = \frac{f(t) - f(t - T_e)}{T_e} \qquad (III.8)$$

Dans le domaine de Laplace, l'opération de dérivation se traduisait par une multiplication par la variable de Laplace ***p*** en continu.

$$Z\left\{\frac{f(t)-f(t-T_e)}{T_e}\right\} = \frac{1-z^{-1}}{T_e}F(z) \qquad (III.9)$$

Ceci conduit à établir l'équivalence linéaire entre la variable de *Laplace* ***p*** et la variable

$$P \longrightarrow \frac{1}{T_e}(1 - z^{-1}) \qquad (III.10)$$

Donc en discret, le terme dérive peut être remplace par :

$$u(k) = \frac{T_d}{T_e}.(\varepsilon(k) - \varepsilon(k-1) \qquad (III.11)$$

III.3.3.L'action intégrale :

L'action intégrale agit proportionnellement à la surface de l'écart entre la consigne et la mesure, et elle poursuit son action tant que cet écart n'est pas nul.

On dit que l'action Intégrale donne la précision statique (Elle annule l'erreur statique). L'action intégrale est conditionnée par le temps d'intégrale ***Ti***.

$$u(t) = \frac{1}{T_i}\int_0^t \varepsilon(\tau).d_\tau \qquad (III.12)$$

Comme dans le cas de l'action proportionnelle, un dosage trop important de l'action intégrale Engendre une instabilité de la boucle de régulation. Pour son réglage, il faut là aussi trouver un compromis entre la stabilité et la rapidité. L'ajout du terme intégral permet d'améliorer la précision mais en contrepartie, il introduit malheureusement un déphasage de (-π) : ce qui risque de rendre le système instable du fait de la diminution de la marge de phase.

Enfin, le correcteur intégral présente le défaut de saturer facilement si l'écart ne s'annule pas rapidement, ce qui est le cas des systèmes lents. En effet, tout actionneur est limité : un moteur est limité en vitesse, une vanne ne peut pas être plus que totalement ouverte ou totalement fermée. Il se peut que la variable de commande amène l'actionneur à sa limite ce qui suspend la boucle de retour et le système aura une configuration assimilable à une

boucle ouverte puisque l'actionneur demeurera saturé indépendamment de la sortie du système.

Quand l'erreur est réduite (action intégrale non saturée), il se peut qu'il faille un temps important pour que les valeurs des variables ne soient correctes de nouveau : on appelle ce phénomène l'emballement du terme intégral. Pour l'éviter, on peut par exemple :

- soit suspendre l'action intégrale quand la commande est saturée ;
- soit appliquer une méthode d'anti-saturation, qui consiste à recalculer le terme intégral pour ne pas saturer.

Pour le cas discret, le terme intégral peut être remplace par la somme des écarts et la différentielle dt par Te ce qui nous donne le résultat suivant :

$$u(n) = \frac{T_E}{T_i}\sum_{k=0}^{n}\varepsilon(k) = u(n-1) + \frac{T_E}{T_i}\varepsilon(n) \qquad \text{(III.13)}$$

Ti : Constante de temps intégrale et s'exprime en (s).

Le problème est que lorsque $\mathcal{E}$ redevient nul, le signal *U* peut avoir atteint une valeur trop élevée pour qu'il y ait équilibre. Autrement, elle permet d'éliminer l'erreur résiduelle en régime permanent.

En résumé, nous affirmons que l'action intégrale est utilisée lorsqu'on désire avoir en régime permanent, une précision parfaite, en autre, elle permet de filtrer la variable à régler d'où l'utilité pour le réglage des variables bruitées telles que la commande. [2]

III.4. Régulation numérique :

III.4.1 .fonction de transfert discrète du PI :

Le schéma bloc de la figure (III.3) 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 :

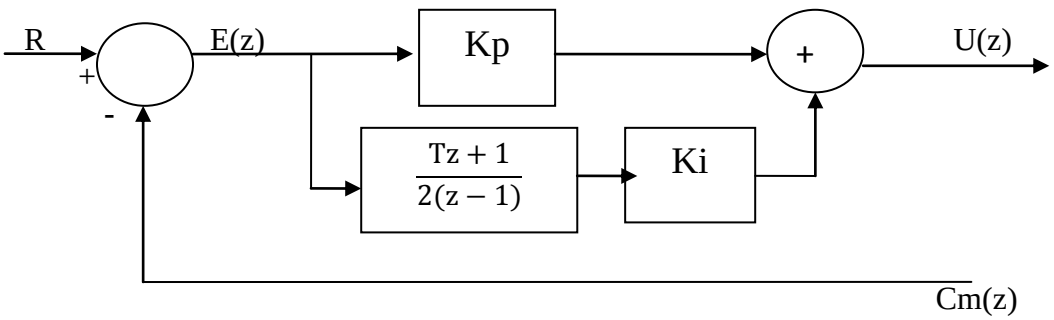


Figure (III.3) : Schéma bloc de la fonction de transfert PI.

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

$$\frac{u(z)}{E(z)} = Kp + Ki\frac{Tz+1}{2(Z^{-1})} = \frac{Kp(z-1)+Ki\frac{T}{2}(z+1)}{Z^{-1}} \quad \text{(III.19)}$$

$$\frac{u(z)}{E(z)} = \frac{Q0Z+Q1}{Z-1} \quad \text{Ou} \quad \begin{cases} Q0 = Kp + Ki\frac{T}{2} \\ Q1 = -Kp + \frac{T}{2} \end{cases} \quad \text{(III.20)}$$

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) **(III.21)**

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 :

$$U(k) = U(k-1) + Q0e(k) + Q1e(k-1) \quad \text{(III.22)}$$

$$m(KT) = Kpe(KT) + Kiei(KT) \quad \text{(III.23)}$$

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

$$ei(KT) = \int_0^{KT} e(t)dt = \int_0^{(k-1)T} e(t)dt + \int_{(k-1)T}^{KT} e(t)dt \quad \text{(III.24)}$$

Selon la définition de e_i *(kT),* la relation (III.24) peut être réécrite de la façon suivante :

$$ei(KT) = ei((k-1)T + \int_{(k-1)}^{KT} e(t)dt \quad \text{(III.25)}$$

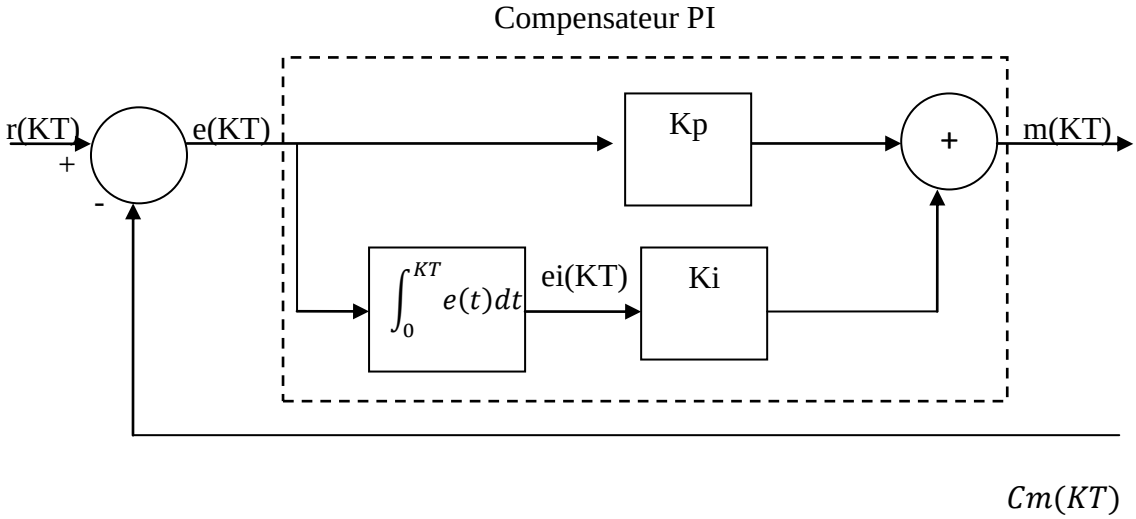


Figure (III.4) : Schéma blocs du compensateur PI.

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. [2]

III.4.2.approximation rectangulaire :

Telle que décrite par la figure (III.5), l'approximation rectangulaire de l'intégrale est donnée par la relation suivante :

$$\int_{(k-1)T}^{kt} e(t)dt \approx Te((k-1)T) \qquad \text{(III.26)}$$

Ainsi, en remplaçant la relation (III.26) dans la relation (III.25), on obtient la relation entre l'entrée et la sortie du système :

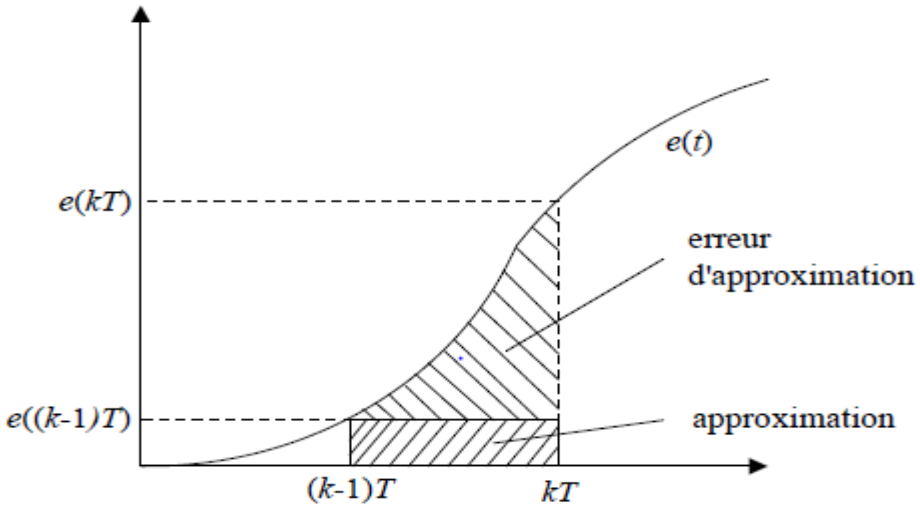
$$ei(KT) = ei((k-1)T + Te((k-1)T \qquad \text{(III.27)}$$

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 :

$$\mathrm{E_i(z)} = \mathrm{z^{-1}E_i(z)} + \mathrm{Tz^{-1}E(z)} \qquad \text{(III.28)}$$

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

$$\frac{\mathrm{E_i(z)}}{\mathrm{E(z)}} = \frac{\mathrm{T}}{\mathrm{Z-1}} \qquad \text{(III.29)}$$



Figure(III.5) : Approximation rectangulaire. [19]

III.4.3. approximation rectangulaire devancée :

Telle que décrite par la figure (III.6), l'approximation rectangulaire devancée de l'intégrale est donnée par la relation suivante :

$$\int_{(k-1)T}^{kt} e(t)dt \approx Te(kT) \qquad \text{(III.30)}$$

Ainsi, en remplaçant la relation (III.30) dans la relation (II.25), on obtient la relation entre l'entrée et la sortie du système :

$$ei(KT) = ei((k-1)T + Te((k-1)T \qquad \text{(III.31)}$$

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 :

$$Ei(z) = z^{-1}E_i(\mathrm{Z}) + \mathrm{T}z^{-1}E(Z) \qquad \text{(III. 32)}$$

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

$$\frac{Ei(z)}{E(z)} = \frac{Tz}{z-1} \qquad \text{(III.33)}$$

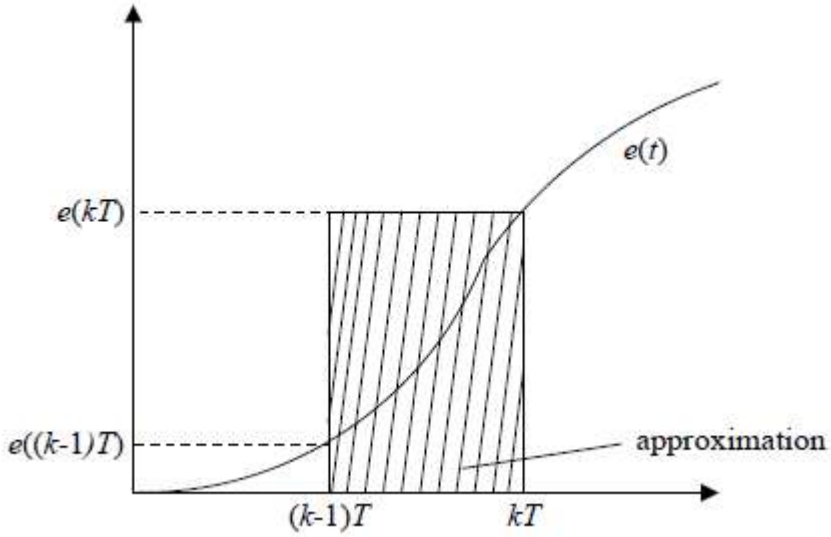


Figure (III.6) : Approximation rectangulaire devancée. [20]

III.4.4. approximation trapézoïdale :

Telle que décrite par la figure (III.7), l'approximation trapézoïdale de l'intégrale est donnée par la relation suivante :

$$(e((k-1)T + e(kt)) \qquad \text{(III.34)}$$

Ainsi, en remplaçant la relation (III.34) dans la relation(III.25), on obtient la relation entre l'entrée et la sortie du système :

$$ei(KT) = ei\big((k-1)T\big) + (e\big((k-1)T\big) + e(KT)) \qquad \text{(III.35)}$$

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 :

$$Ei(z) = Ei(z) + \big(E(z) + E(z)\big) \qquad \text{(III.36)}$$

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

$$\frac{Ei(z)}{E(z)} = \frac{Tz+1}{2(z-1)} \qquad \text{(III.37)}$$

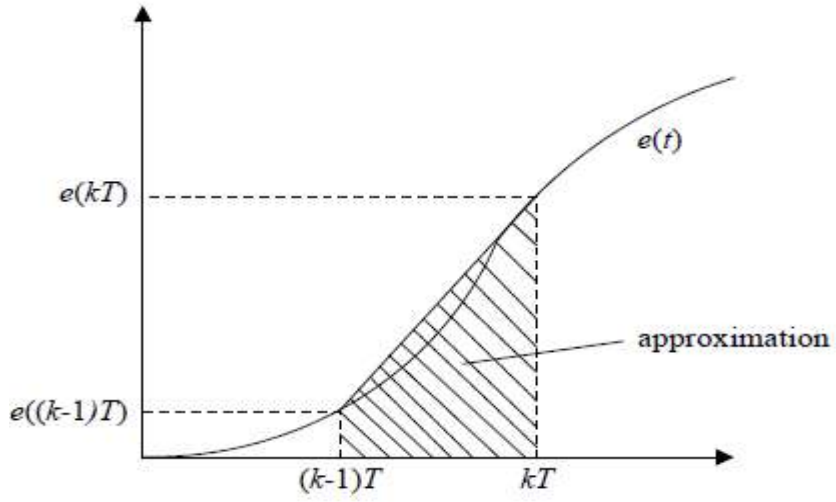


Figure (III.7). : Approximation trapézoïdale. [20]

III.4.5.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 (III.4). 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 (III.8). [20]

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

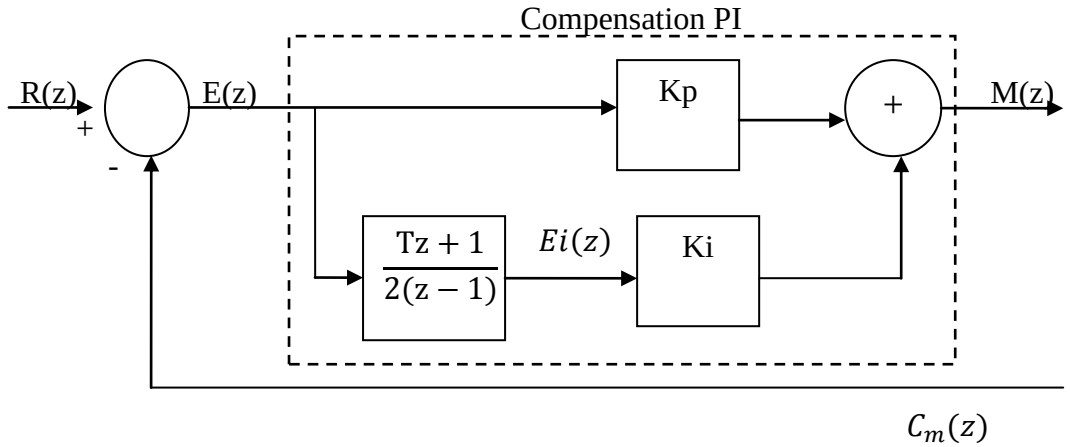
$$\frac{\mathrm{M(z)}}{\mathrm{E(z)}} = \mathrm{Kp} + \mathrm{Ki}\frac{\mathrm{Tz+1}}{\mathrm{2(z-1)}} \qquad \text{(III.38)}$$

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

$$\frac{\mathrm{M(z)}}{\mathrm{E(z)}} = \frac{\mathrm{Q0Z+Q1}}{\mathrm{Z-1}} \qquad \text{(III.39)}$$

Ou :

$$\begin{cases} Q0 = Kp + Ki\frac{\mathrm{T}}{2} \\ Q1 = -Kp + k_i\frac{T}{2} \end{cases} \qquad \text{(III.40)}$$



Figure(III.8) : Schéma bloc en z du compensateur PI.

II.5.Identification de processus et calcul des paramètres de correcteurs:

Identifier un processus (système), c'est chercher un modèle (dynamique) mathématique, appartenant à une classe de modèles connue, et qui, soumis à des signaux tests (en entrée), donne une réponse (dynamique et statique en sortie), la plus proche possible du système réel. [22]

- **Les modèles mathématiques d'un processus :**

La notion de modèle mathématique d'un système, d'un processus ou d'un phénomène, est un concept fondamental. Il existe une multitude de types de modèles, chacun étant destiné à une application particulière.

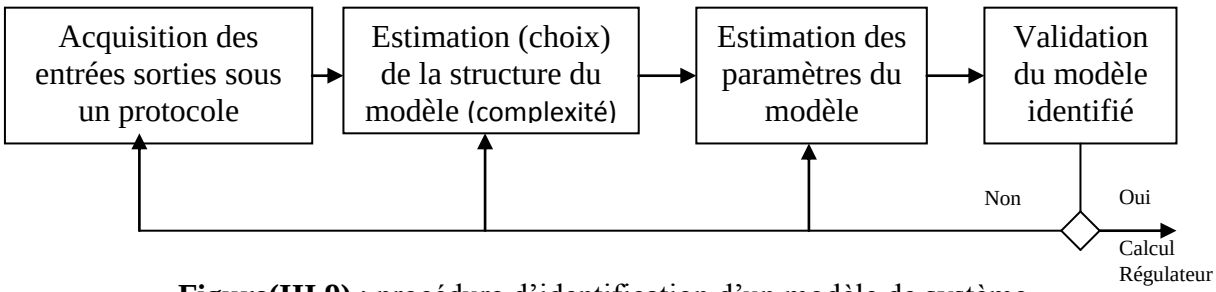
Nous pouvons les décliner en deux grandes catégories :

- **Les modèles de connaissance :** (basés sur les lois de la physique, de la chimie…), donnent une description complète des systèmes et sont utilisés pour la simulation et la conception des procédés. Ce sont souvent des modèles complexes.
- **Les modèles dynamiques de commande**, qui donnent la relation entre les variations des entrées d'un système et les variations de la sortie, sont utilisés en automatique.

Les modèles dynamiques sont de deux sortes :

- Modèles non paramétriques (réponse fréquentielle, réponse à un échelon)
- Modèles paramétriques (fonction de transfert, équations différentielles)

L'identification est une approche expérimentale pour la détermination du modèle dynamique d'un système. Cette approche peut être décomposée en quatre étapes :



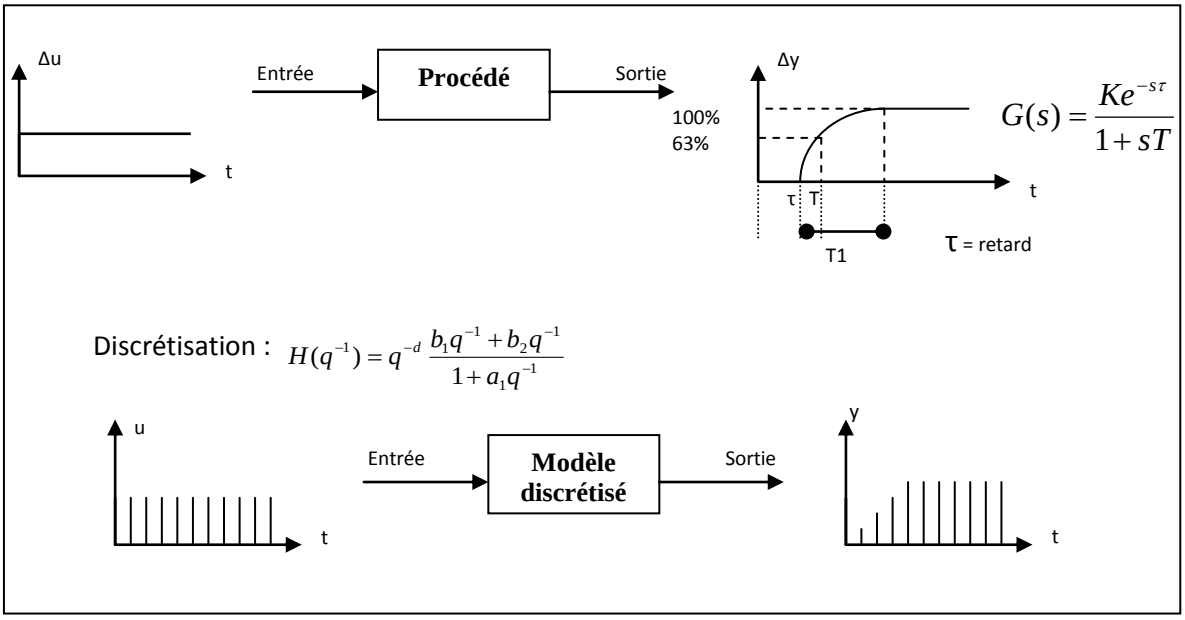
Figure(III.9) : procédure d'identification d'un modèle de système.

La première étape fournit les données entrées/sorties susceptibles de permettre l'extraction d'un modèle de procédé significatif. Le protocole d'acquisition est un problème-clé car il conditionne en grande partie le succès d'une opération d'identification. Il doit tenir compte d'une part des contraintes pratiques, ce qui requiert en général l'utilisation de signaux d'excitation de faible amplitude et d'autre part de la nécessité d'exciter le système à identifier dans une bande de fréquences suffisamment large afin d'obtenir un modèle significatif pour le calcul de la commande.

L'estimation (ou le choix) de la complexité pour un type de modèle permet de déterminer quels sont les paramètres à estimer et quel est leur nombre. L'estimation paramétrique utilise des algorithmes qui, à partir des mesures entrées/sorties disponibles, fournissent les paramètres du modèle. Enfin, la dernière étape doit permettre de mettre en évidence si le modèle identifié est représentatif des comportements entrées/sorties du procédé.

Une opération d'identification complète doit nécessairement comporter les quatre phases indiquées plus haut. Les méthodes spécifiques utilisées dans chaque phase dépendent du type de modèle recherché (paramétrique ou non paramétrique, continu ou échantillonné).

La méthode d'identification "classique" utilisée pour obtenir des modèles paramétriques à partir des modèles non paramétriques type "réponse en échelon" est illustrée dans la figure(III.10) Cette méthode, initialement utilisée pour obtenir des modèles paramétriques continus, a été étendue pour l'identification des modèles échantillonnés.



Figure(III.10) : Méthode "classique" d'identification.

Nous nous intéresserons par la suite à l'identification des modèles dynamiques paramétriques échantillonnés qui sont les plus appropriés pour la conception et l'ajustement des systèmes numériques de commande et régulation.

Il existe deux principales classes de méthodes paramétriques :

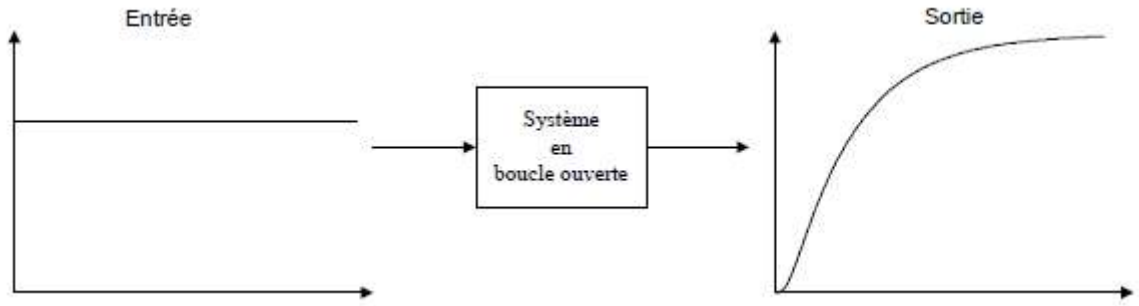
- Les méthodes paramétriques graphiques ou déterministes
- Les méthodes paramétriques statistiques (études stochastiques). [22]

Le but principal de l'identification en automatique est la mise en place d'un correcteur ou d'un régulateur afin d'améliorer les performances du système en boucle fermée. Aussi l'on est en droit de se demander si cette étape d'identification est bien nécessaire. C'est exactement le principe de la méthode de Ziegler-Nichols, qui n'est pas une méthode d'identification à proprement parler mais une méthode de réglage fondée sur des mesures effectuées directement sur le système. [15]

III.6.Méthodes de Ziegler et Nichols (ZN):

En 1942, Ziegler et Nichols ont proposé deux approches heuristiques basées sur leur expérience et quelques simulations pour ajuster rapidement les paramètres des régulateurs P, PI et PID. La première méthode nécessite l'enregistrement de la réponse indicielle en boucle ouverte, alors que la deuxième demande d'amener le système bouclé à sa limite de stabilité.

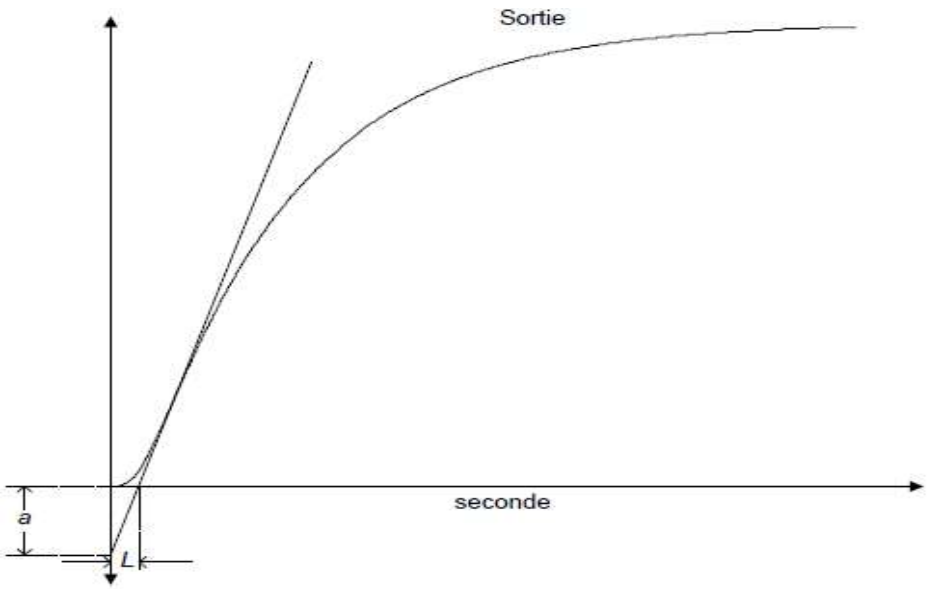
Il existe en fait deux façons d'appliquer la méthode de Ziegler – Nichols : une dans le domaine du temps et l'autre dans le domaine de la fréquence .dans notre travaille, nous allons uniquement introduire la méthode basé sur le domaine du temps comme l'indique la figure(III.11), la méthode dans le domaine du temps essentiellement basée sur la réponse à l'échelon du système en boucle ouverte. [22]



Figure(III.11) : Réponse à l'échelon du système en boucle ouverte.

- **Calcul des gains :**

Pour obtenir les paramètres du régulateur PID, il subit d'enregistrer la réponse indicielle du processus seul (c'est-à-dire sans le régulateur), puis de tracer la tangente au point d'inflexion de la courbe. On mesure ensuite sa pente **a**, le retard apparent L correspondant au point d'intersection de la tangente avec l'abscisse figure(III.12). On peut alors calculer les coefficients du régulateur choisi à l'aide du tableau(III.1).



Figure(III.12) : paramètre a et L de la méthode Ziegler et Nichols. [22]

Selon les parametres a et L obtenus à partir de la réponce a l'échelon du système ,les gains du compensateur peuvent finalement etre à partir du tableau siuvant.

Tableau(III.1): Gain des compensateurs

compensateur	Kp	Kp/Ki	Kd/Kp
P	$\frac{1}{a}$		
PI	$\frac{0.9}{a}$	$3L$	
PID	$\frac{1.2}{a}$	$2L$	$\frac{L}{2}$

III.7.Conclusion :

Dans ce chapitre nous avons présenté le modèle discret de compensateurs PI, puis nous avons fait une identification sur ce modèle pour déterminer tout les paramètres utilisés de ce système, après on a utilisée la méthode de Ziegler –Nichols pour obtenir les paramètres Kp et Ki.

Chapitre IV : Implémentation et simulation

IV.1. Introduction :

Dans ce chapitre, nous allons implémenter le régulateur numérique (PI) dans un microcontrôleur pour commander un moteur à courant continu, on utilisant les outils de développement software suivants :

- le mikroC pro pour la programmation du pic.
- Proteus ISIS pour faire des tests et la simulation.
- Matlab pour identification.

IV.1.1.Le mikroC PRO :

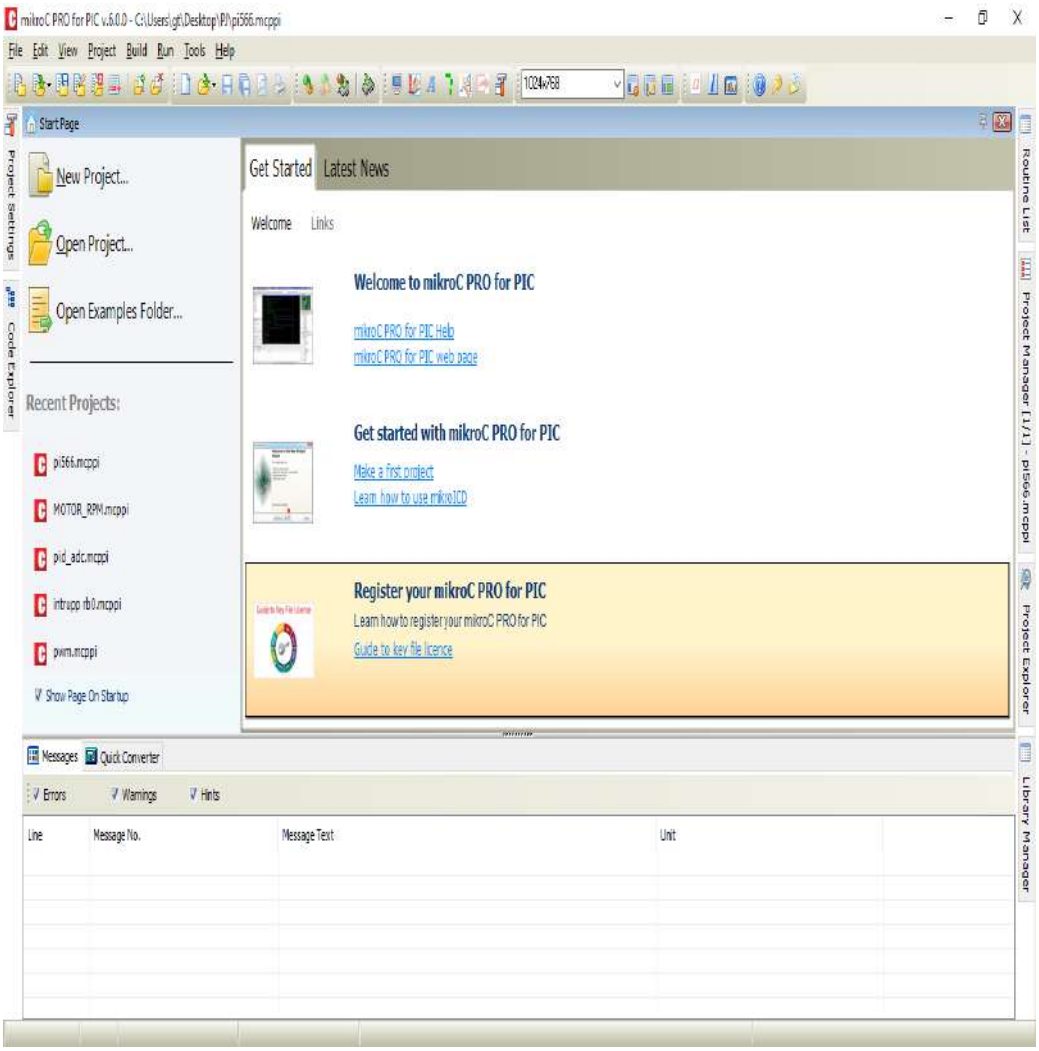
Le MikroC est un puissant outil pour les microcontrôleurs PIC. Il est conçu pour fournir au Programmeur la solution la plus simple possible pour développer des applications pour les systèmes embarqués, sans compromettre les performances ou le contrôle.

Le MikroC permet de développer et déployer des applications complexes:

- Écrire un code source C en utilisant le très avancé éditeur de code.
- Utiliser les bibliothèques du MikroC fait accélérer considérablement le développement (Acquisition de données, la mémoire, affichage, les conversions, les communications.....).
- Surveiller la structure du programme variables et fonctions dans l'explorateur de code.
- Inspecter le déroulement du programme et déboguer la logique d'exécution avec L'intégration debugger, obtenez des rapports détaillés et des graphiques sur les statistiques du Code.

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..[1]

- On obtient l'interface finale.

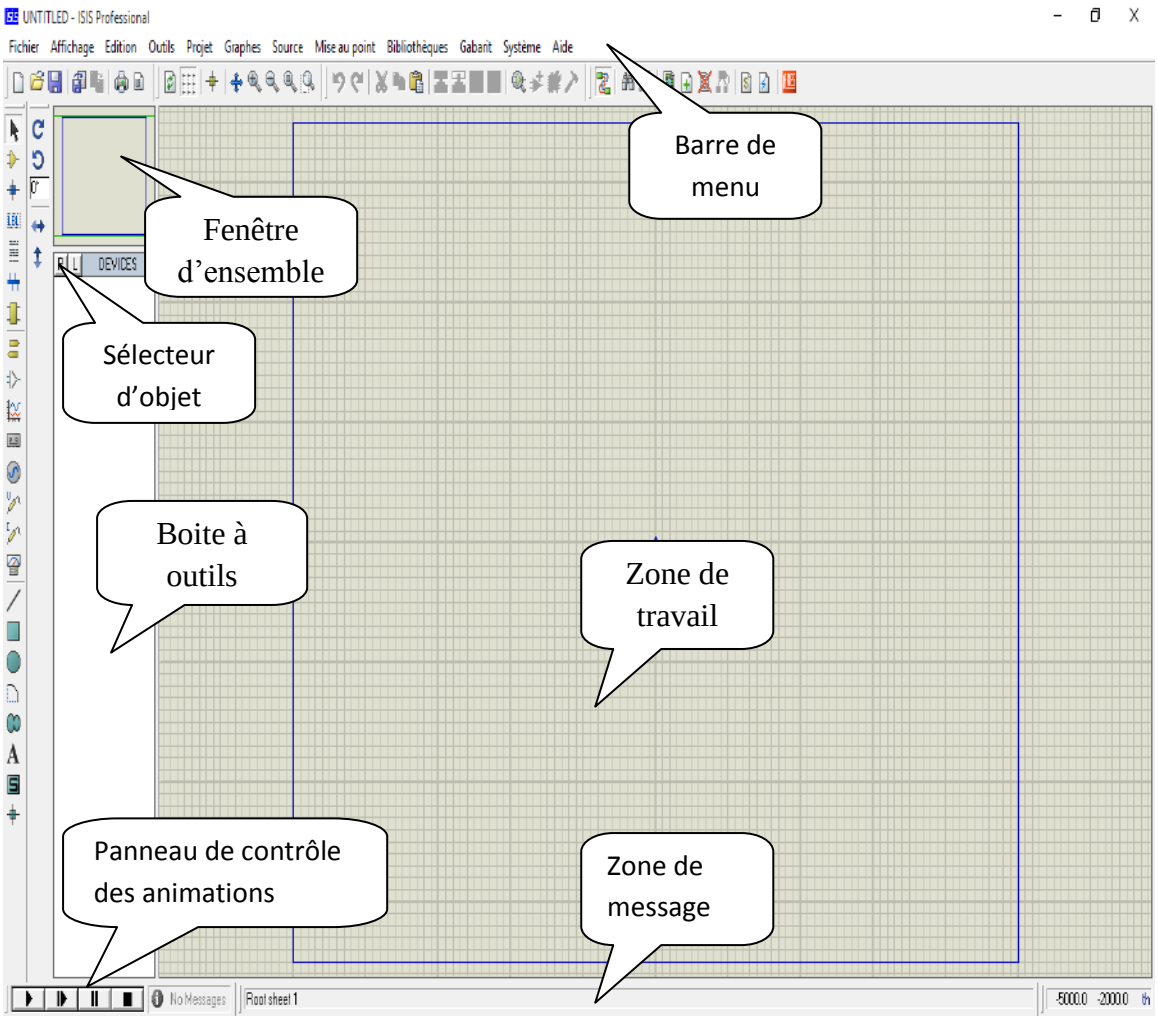


Figure(IV.1) : L'ouverture du mickroC**.**

- À la fin de la création de nouveau projet.
- On obtient un fichier .hex (forme hexadécimal).

VI.1.2.Le proteus ISIS :

IV.1.2.1.Présentation de l'interface d'ISIS proteus V7:



Figure(IV.2) : L'interface d'ISIS proteus v7.

IV.1.2.2.les étape de travaille sur le proteus :

- On lance le programme ISIS proteus v7.



- choisir les composants dans la zone «sélecteur d'objet »et on met les composants dans la zone de travaille.

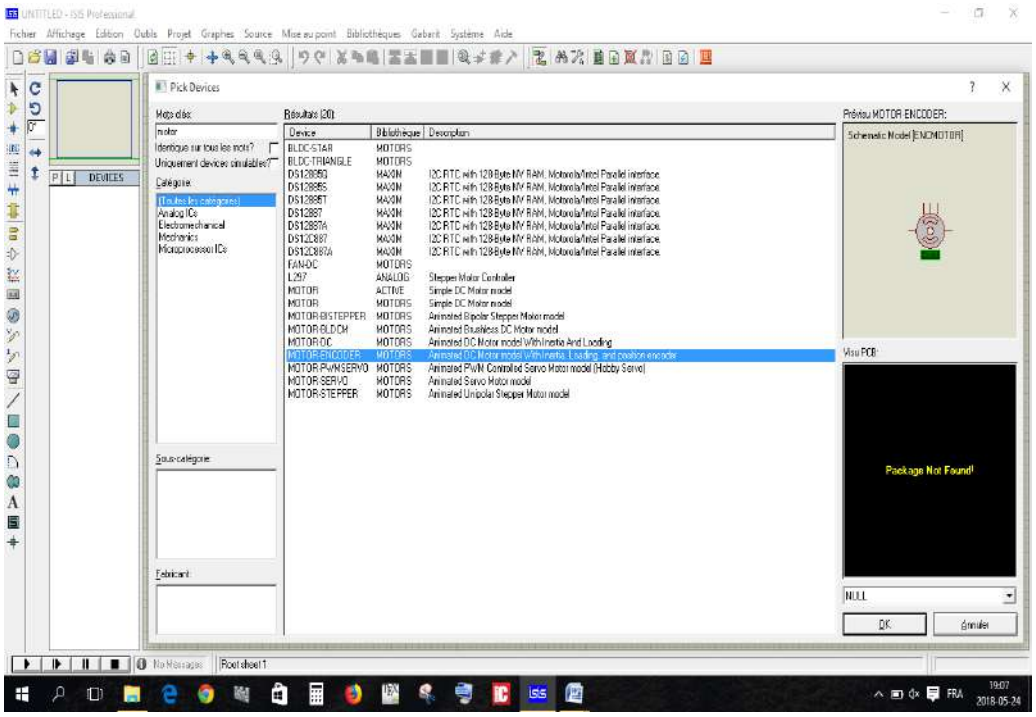
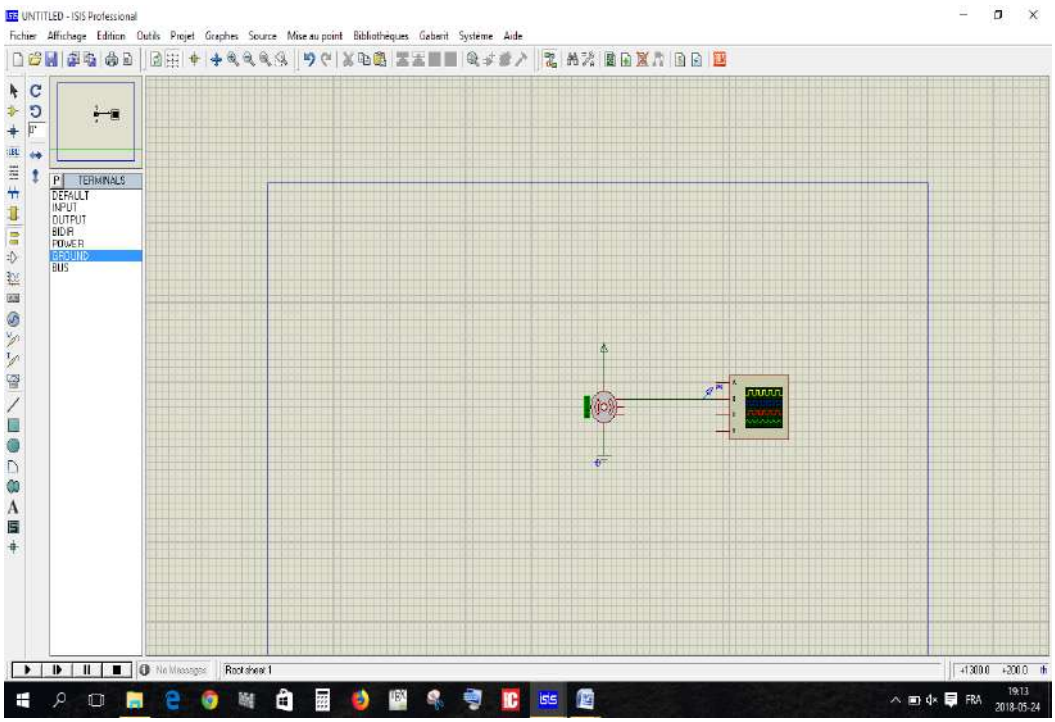


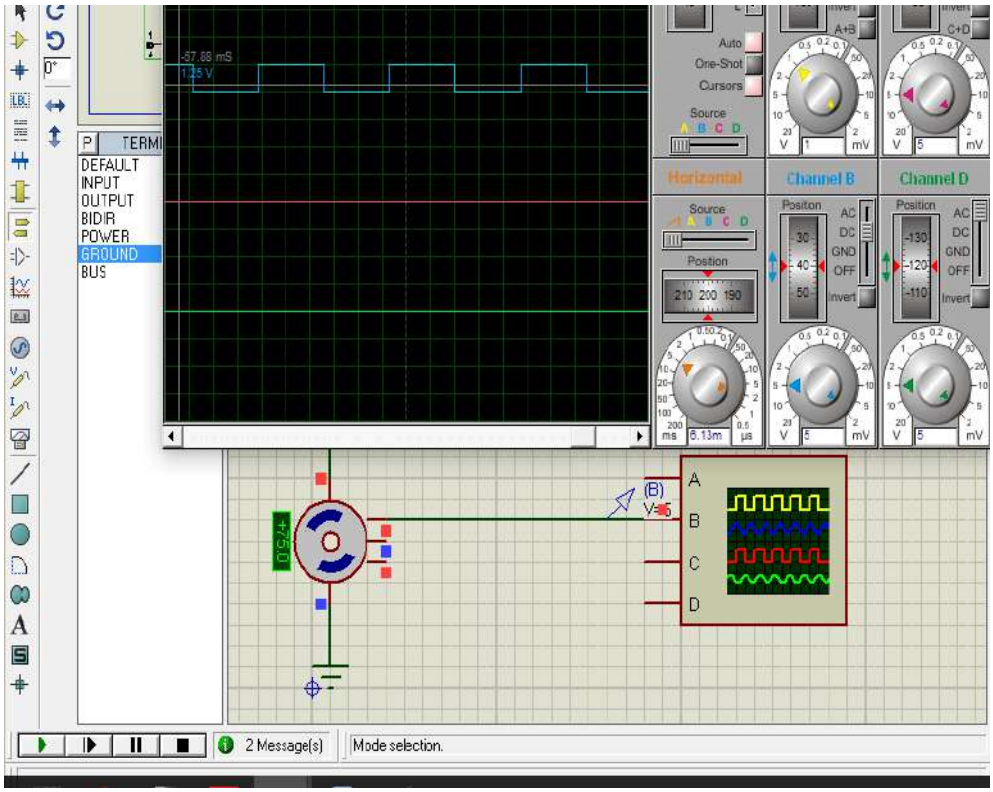
Figure (IV.3) : choix des composants dans la bibliothèque.

- Ensuite, on relie les composants qu'on a choisit.



Figure(IV.4) : connecter les composants.

- A la fin de la réalisation de montage on va faire la simulation, on utilisant la zone de panneau de contrôle des animations.



Figure(IV.5) : Schéma de la simulation.

IV.2.Implémentation de la loi de commande :

Pour arriver à faire cette implémentation, nous allons passer par les points suivant :

IV.2.1.Identification du système :

Le but de cette partie est de déterminer la fonction de transfert de notre système (MCC) en boucle ouvert notée *G*(s).pour obtenir les paramètres de régulateur numérique PI par la méthode de Zeigler-Nichols.

Tel que : $G(s) = \frac{u(s)}{v(s)}$ (IV.1)

Avec:

u (s): L'entrée du système (la tension).

v(s): La sortie du système (la vitesse de moteur).

On a passé par les étapes suivantes:

Etape 1:

On alimente le moteur par une tension nominale égale à 12V, nous faisons passer ensuite à la sortie de l'encodeur incrémentale sous forme d'une fréquence à travers un circuit électronique LM331 pour avoir une tension analogique qui représente l'image de la vitesse.

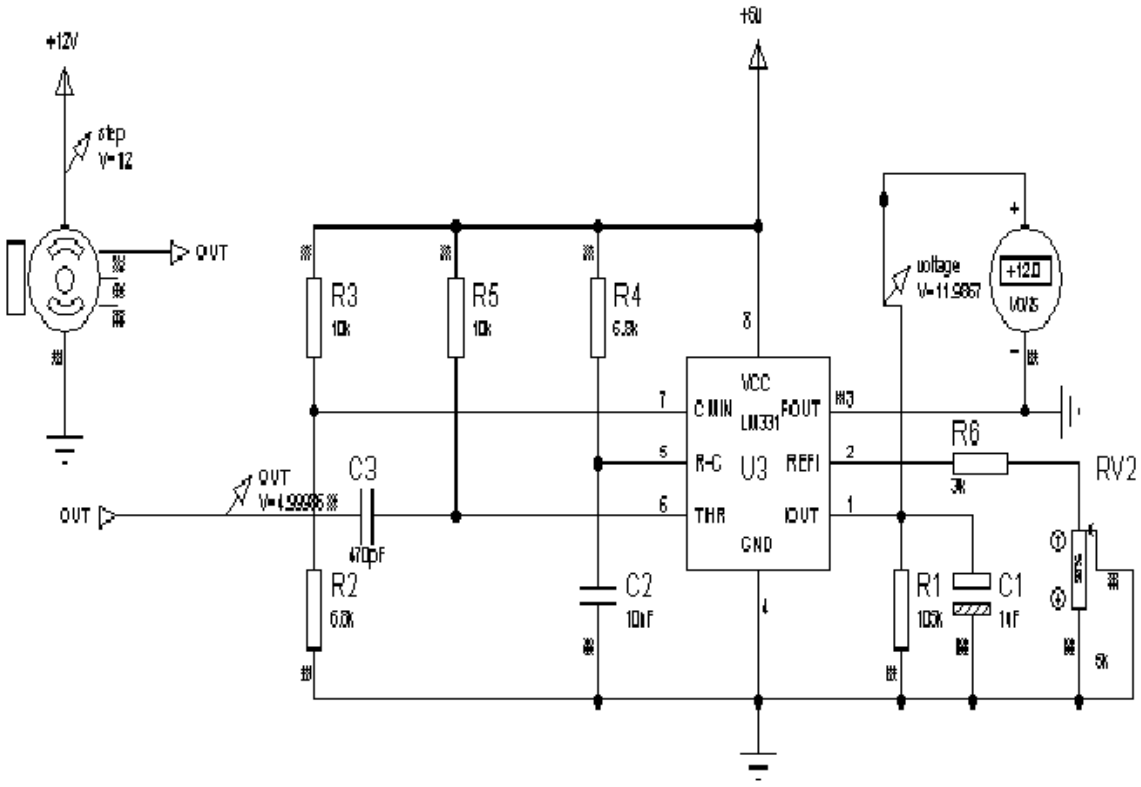


Figure (IV.6): schéma électrique sur proteus.

Nous nous représentons la réponse de notre moteur dans la figure suivante :

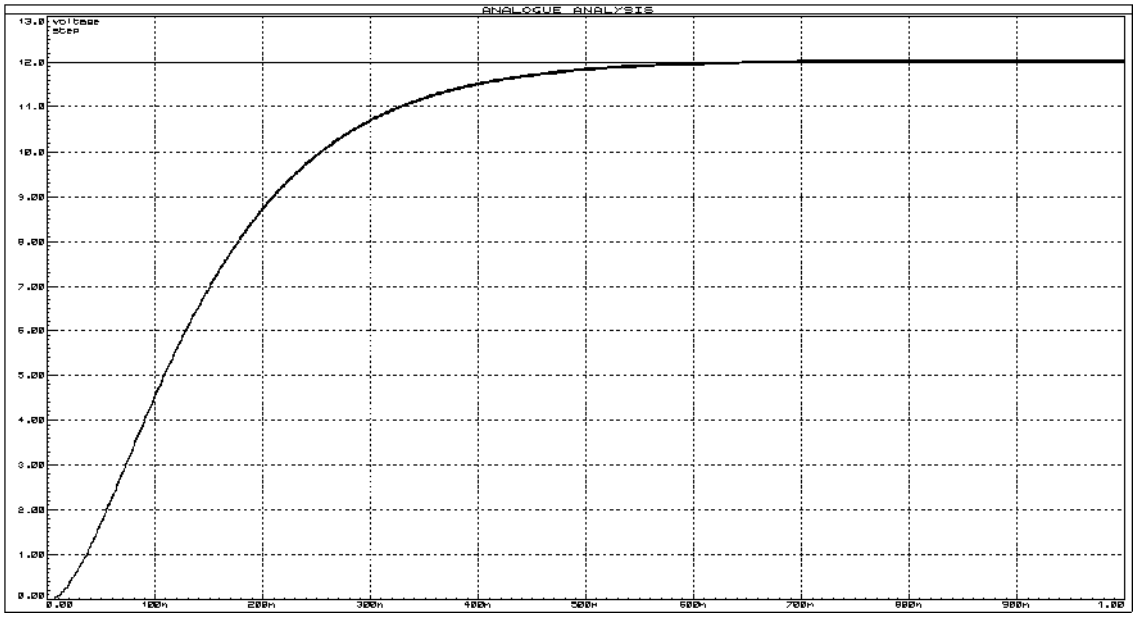


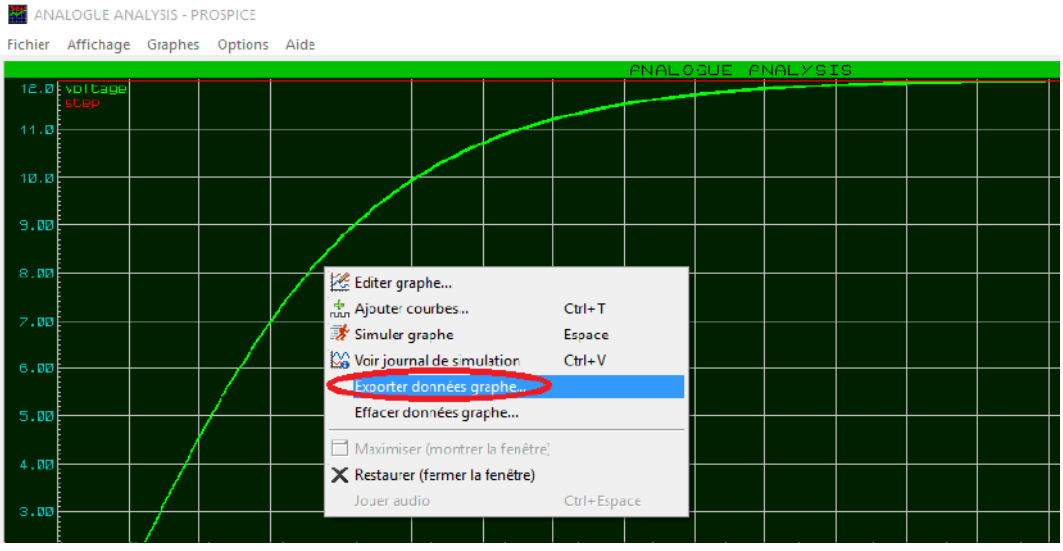
Figure (IV.7): réponse de système.

D'après les résultats, on remarque le temps de réponse (Tr=0.59s) est lent.

Etape 2 :

Elle consisté a exporté les donnée relevé par proteus comme suite:

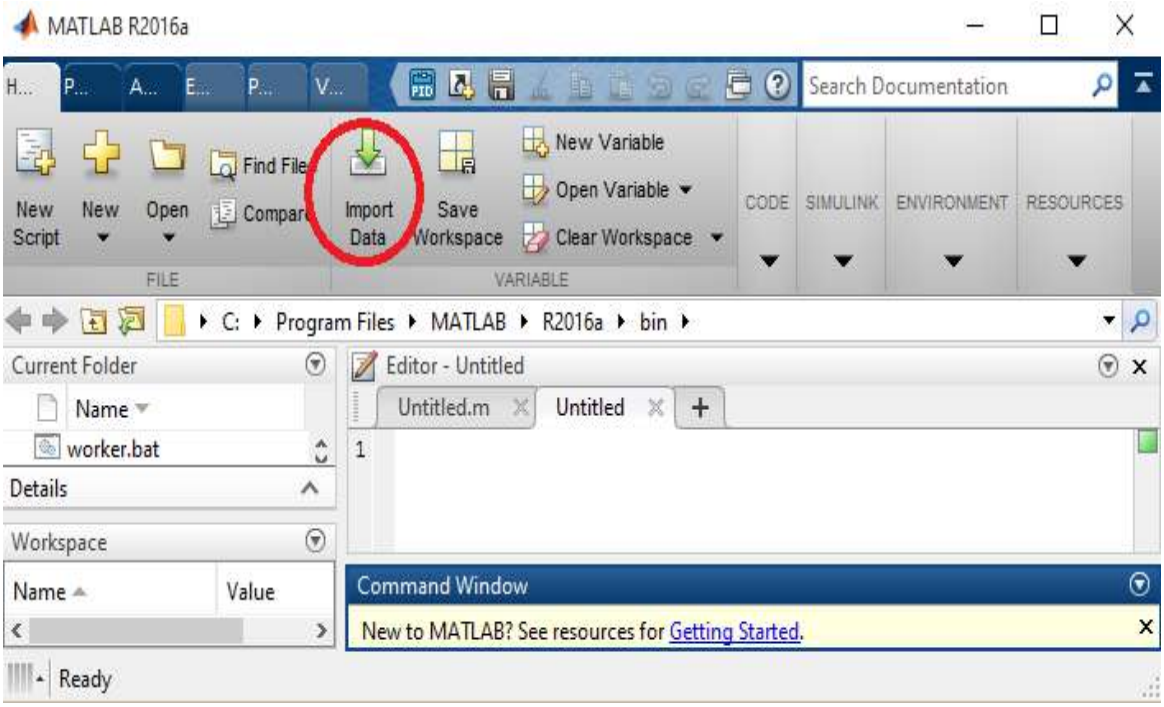
- Cliqué sur le bouton droit de la souris.
- choisir l'instruction export données graphe, comme la figure suivante:



Figure(IV.8): export les données de système.

Etape 3 :

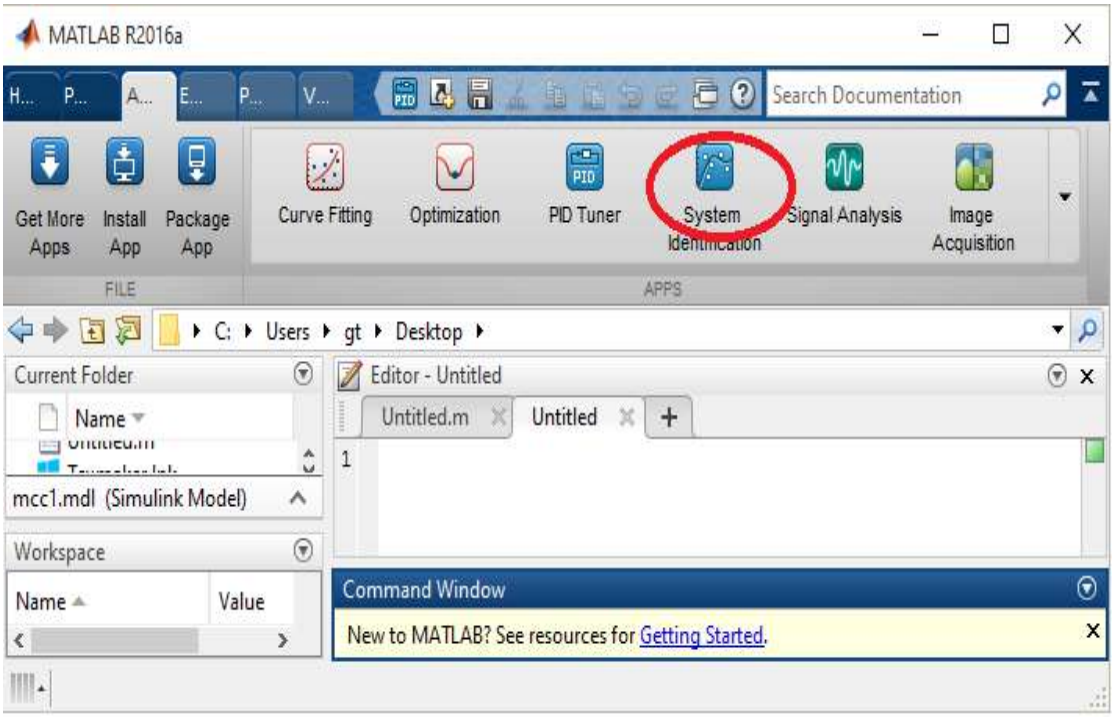
On importe ces données précédant dans le logiciel matlab par import Data.



Figure(IV.9): Import les données de système.

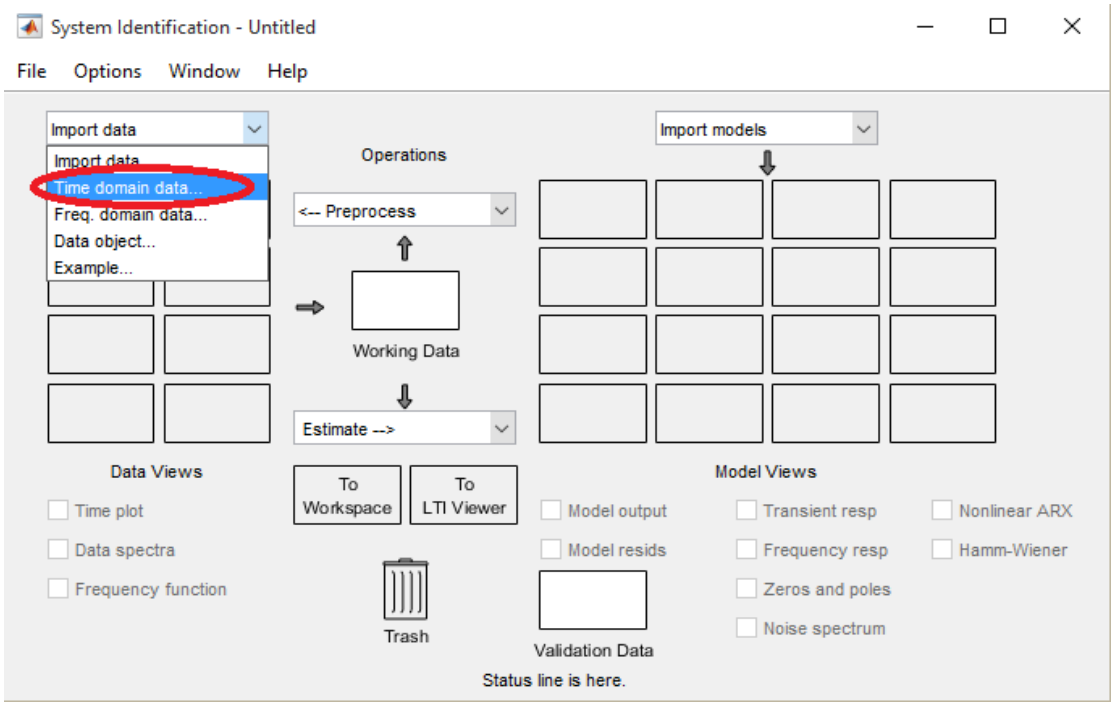
D'après avoir importé les données de Proteus à logiciel matlab, nous allons utiliser l'outil <<System Identification>> du matlab pour déterminer la fonction de transfert du MCC. Grâce à de cette données, Suivez les étapes ci dessous:

1. Ouvrir l'outil System identification Tool



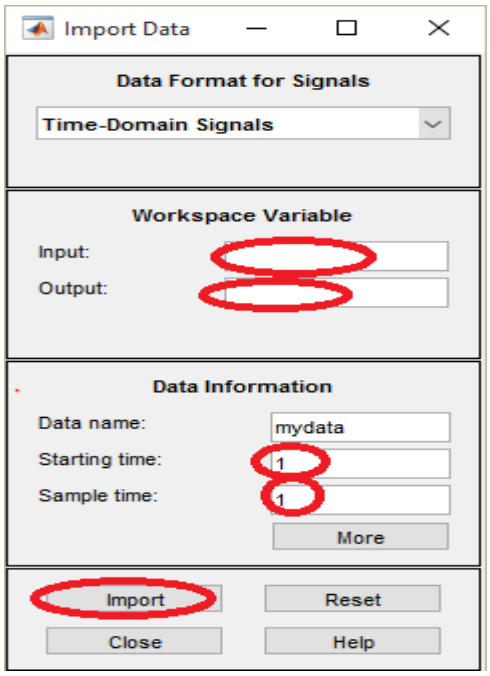
Figure(IV.10): L'utilisation de l'outil System Identification.

2. Cliquer sur import data et choisir Time domain data.



Figure(IV.11): L'interface de l'outil System identification.

3. Entrer le nom de la variable Input et la variable Output ainsi que temps de starting time et simple time. Enfin cliquer sur Import.



Figure(IV.12): Saisie des données relatives aux Input et Output du système.

4. Cliquer sur Estimate et choisir Transfer Function Models.

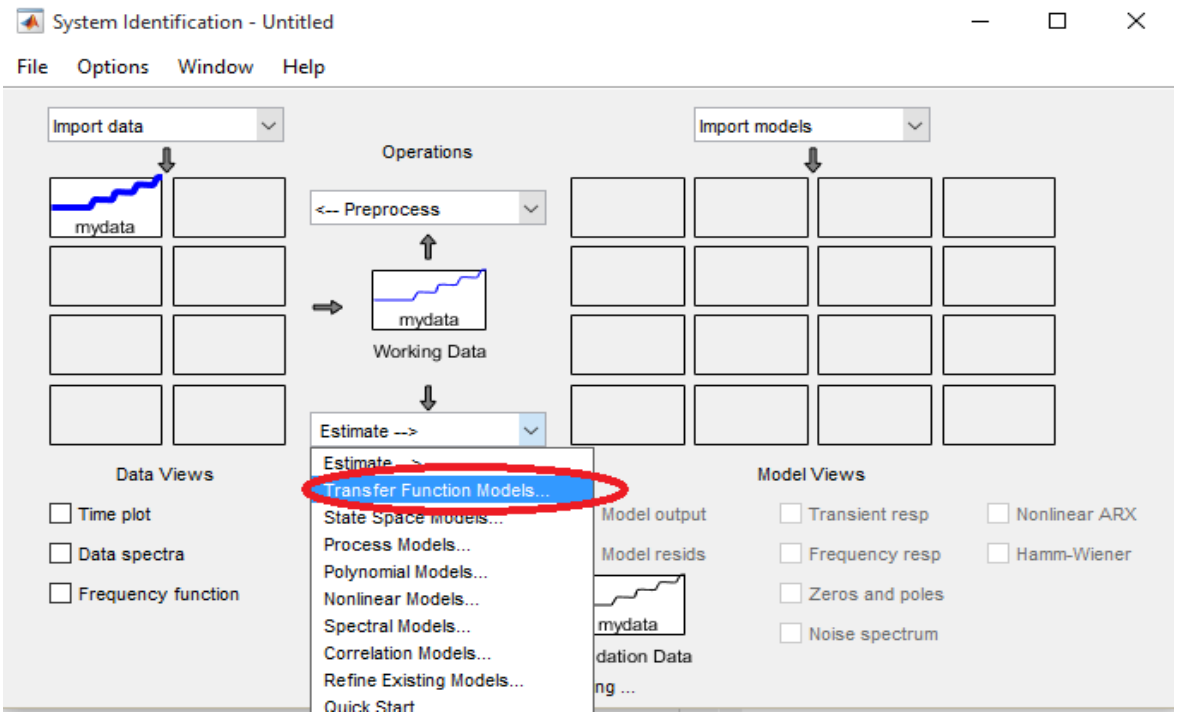
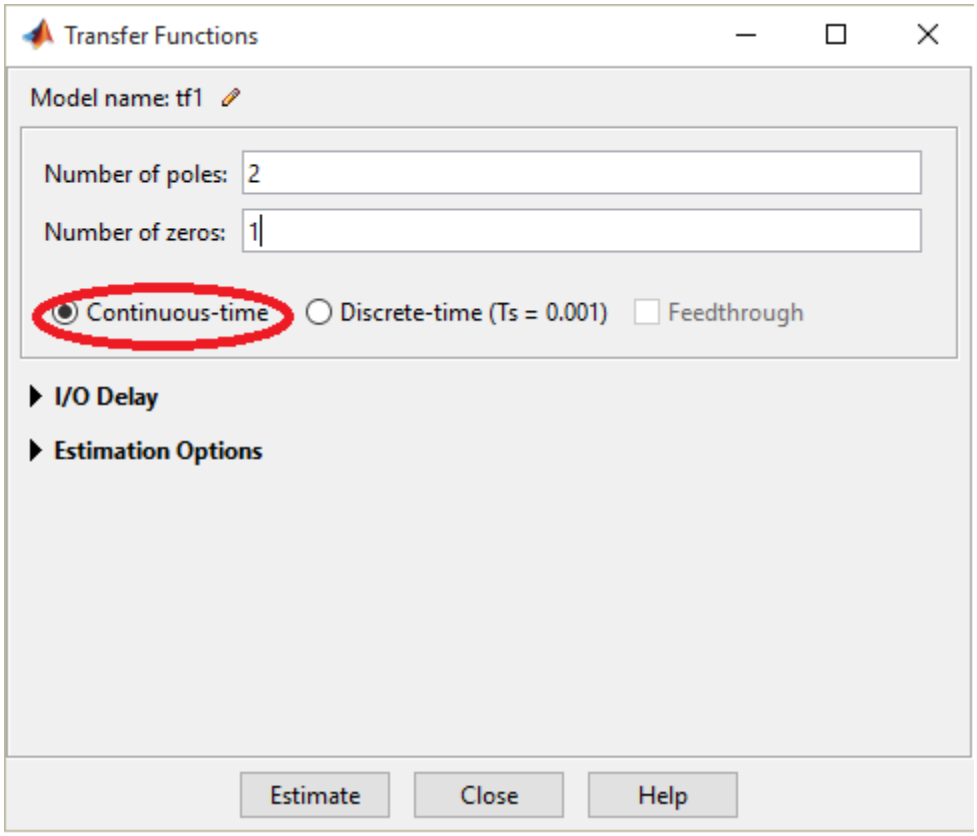


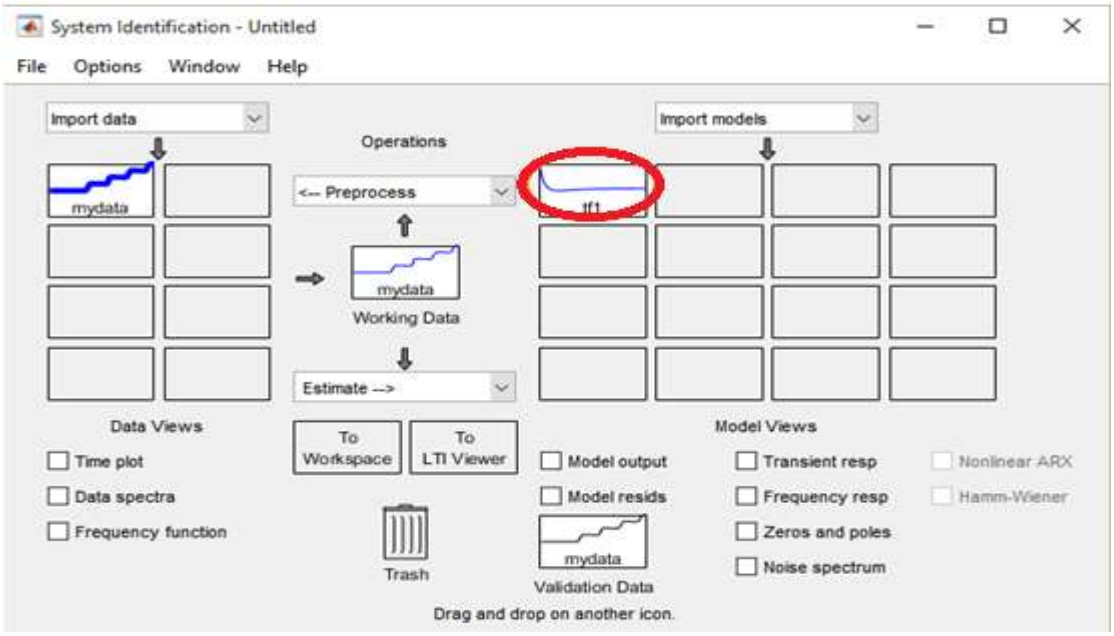
Figure (1V.13) : Choix de la description du système à estimer"Tranfer Function".

5. Entrer le nombre de pôle et de zéro et cliquer sur Discrete-Time ensuite cliquer sur Estimate.



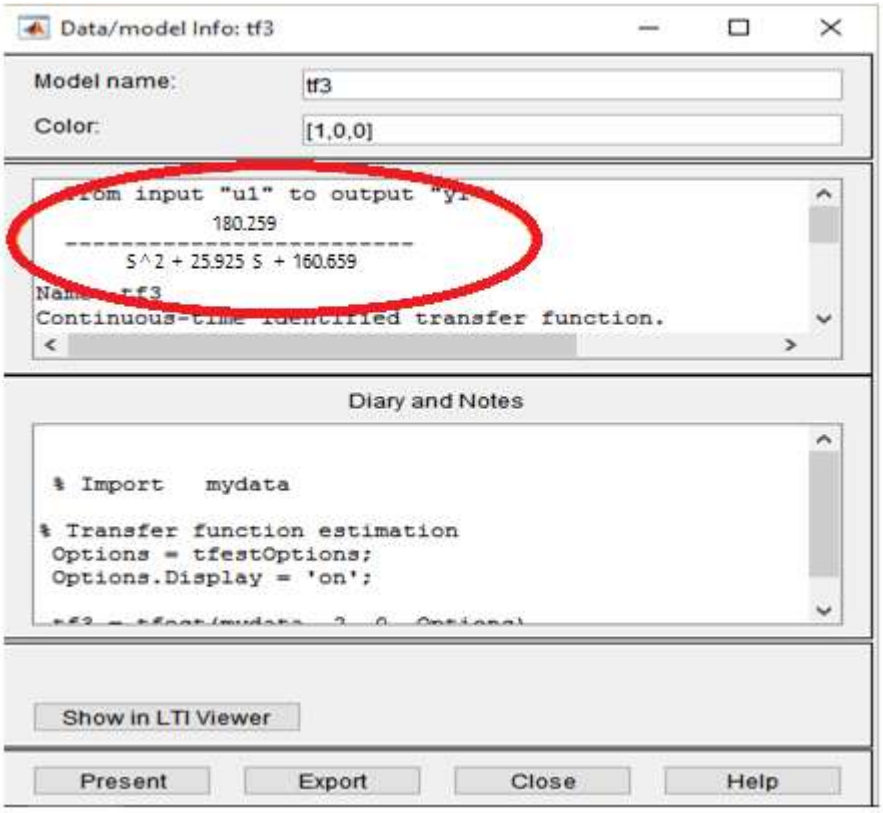
Figure(IV.14): Choix du nombre des pôles et zéros de la fonction de transfert à estimer.

6. Vous pouvez revenir à l'interface System Identification Tool et cliquer deux fois sur tf1 pour voir le résultat.



Figure(IV.15): Visualisation du résultat de l'estimation.

7. Une fenêtre apparait dans laquelle vous trouvez la fonction de transfert estimée *G*(s).



Figure(IV.16): Récupération de la fonction de transfert estimée.

Synthèse de résultats :

Après l'identification de système (MCC), on a comparé la fonction de transfert estimée avec la réponse de moteur relevé par proteus ISIS, et on a remarqué:

L'identification n'est pas parfaitement, donc on a ajusté sur les coefficients de fonction de transfert, puis on a trouvé la nouvelle fonction de transfert suivante :

$$G(s) = \frac{187.7}{s^2 + 29.925s + 187.7} \qquad (IV.2)$$

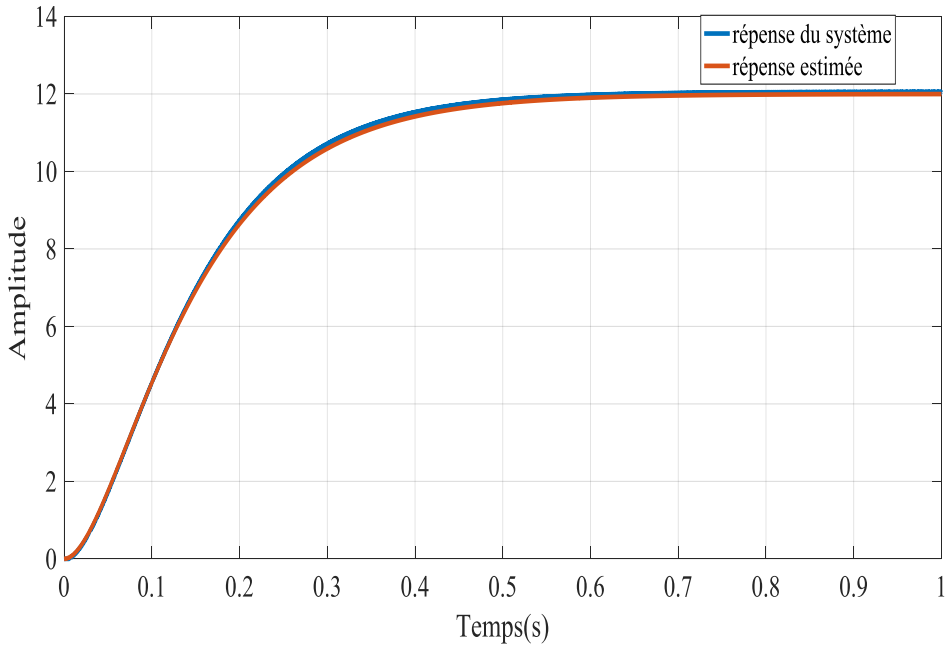


Figure (IV.17): la comparaison entre la fonction de transfert estimé avec la réponse de système.

IV.3.Régulation numérique de vitesse d'un moteur à courant continu :

Pour faire la régulation numérique de vitesse on a utilisé le schéma synoptique suivant**:**

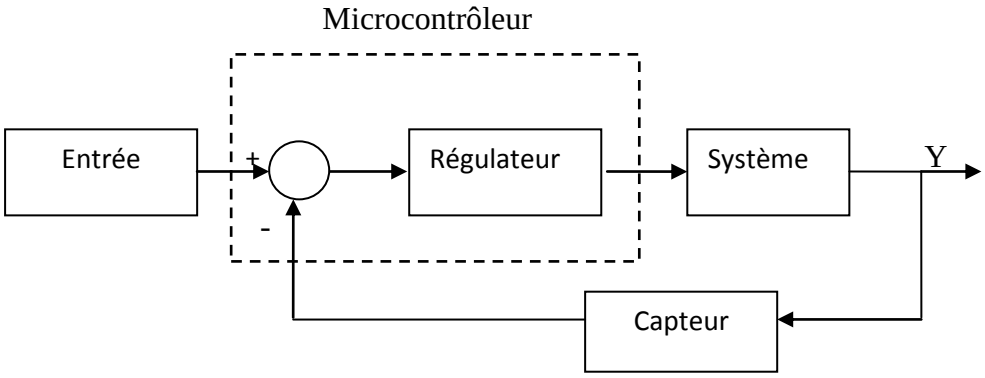


Figure (VI.18): Schéma synoptique de la chaine de régulation numérique de la vitesse de MCC

Notre travaille est basée sur le schéma synoptique détaillée comme représente dans la figure ci-dessous:

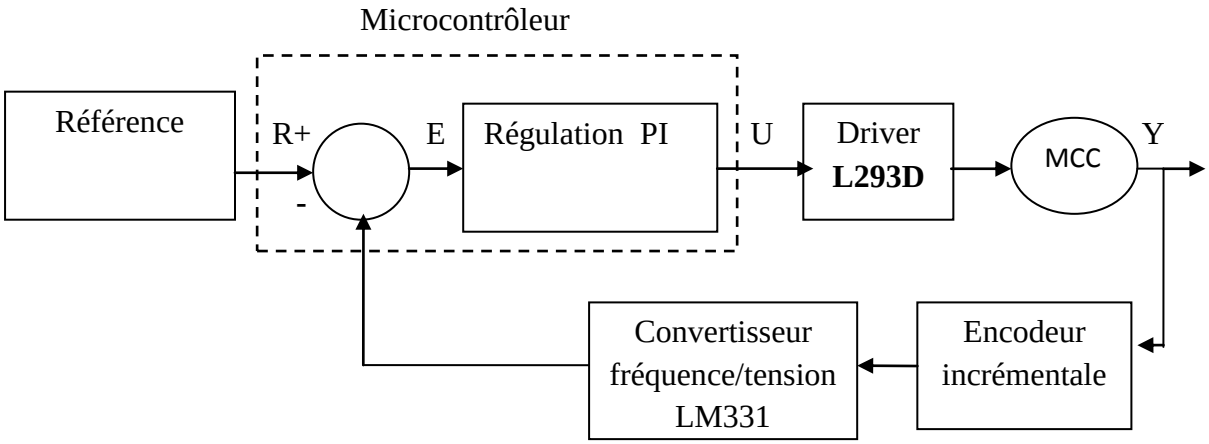


Figure (VI.19): Schéma synoptique détaillée de la chaîne de régulation numérique de la vitesse de MCC**.**

Dans ce schéma le principe de base est comme suit :

Le régulateur numérique permet de générer une tension de commande à chaque instant d'échantillonnage. Pour relier ce dernier avec notre MCC, il faut d'un intermédiaire c'est bien le Driver L293D pour amplifier la puissance, le moteur qu'on a utilisé est doté d'un encodeur incrémentale, son rôle est donné l'information sur sa vitesse puis la fréquence de cette vitesse est converti à une tension par le circuit LM331. Après, on a implémenté un régulateur ou correcteur PI pour corriger l'erreur entre la tension de référence et la vitesse de moteur sous forme une tension.

A partir de la figure(IV.19), le rôle de chaque bloque sera décrit comme suit :

IV.3.1.Bloque de référence :

Le bloque de référence est sous forme d'un potentiomètre qui va donnée une tension entre 0V à 5V, qui correspond à des vitesses entre 0 tr/min et 1500 tr/min.

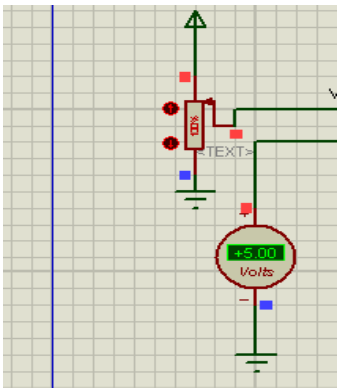


Figure (IV.20): Potentiomètre.

IV.3.2.bloque de régulation PI:

Ce bloque est le cœur de notre travaille qui représente un microcontrôleur 16F877A, le choix de ce microcontrôleur grâce à ces caractéristiques, parmi ses caractéristiques on cite les trois modules utilisés pour notre programmation qui sont les suivants : les portes d'entrée sorties, le convertisseur ADC et USART.

- **Les portes d'entré\ sorties:**

• on a la ligne A0 de porte A est configure comme entrée analogique afin de l'utiliser comme entrée de référence, et la ligne A1 de porte A est configuré comme entrée analogique afin de l'utilisée comme entrée de vitesse de moteur sous forme de tension.

- On a la ligne RC2 de porte C est configuré comme sortie numérique de commande (PWM).

- **le convertisseur ADC:**

Le convertisseur de microcontrôleur permet de convertir les deux tensions analogiques qui sont la tension de référence et la tension de la vitesse délivre pour le circuit LM331 à des tensions numériques.

Pour cela on a configurée les deux registres : ADCON0 et ADCON1 qui caractérisent comme suit :

ADCON0=81, pour déterminer la fréquence de conversion.

ADCON1=04, pour sélectionner A0 et A1 comme des entrées analogiques.

- la résolution du convertisseur est 10 bits cela signifie que 5V est représentée par 1111111111 et le pas du convertisseur est donc de 0.0048V.

c-à-d :

0V ⟶ 0000000000
0.0048V ⟶ 0000000001
0.096V ⟶ 0000000010
3V ⟶ 1001100110
5V ⟶ 1111111111

Calcule de la vitesse :

La vitesse nominale du notre moteur est de 1500 tr /min, nous avons donc donné la tension 5V pour présenter cette vitesse :

$$\mathrm{Vn} = \frac{\mathrm{E}*1500}{1023}\ \mathrm{tr/min} \qquad \text{(IV.3)}$$

Ou :

E : consigne après conversion (consigne numérique)

Vn: vitesse nominale numérique

Exemple :

Si la consigne 2.5V alors

$\mathrm{E}=(0111111111)_2 \rightarrow \mathrm{E} = (510)_{\mathrm{dec}}$

D'où :

$\mathrm{Vn} = \frac{510*1500}{1023}$ = 774.8 tr/min

IV.3.2.1.Détermination de temps d'échantillonnage :

Le théorème d échantillonnage permet de choisir la période d'échantillonnage de façon à éviter la perte d'information essentielle lors de l'échantillonnage, SHANNON arrive à la conclusion que, pour ne pas perdre d'information essentielle, un signal doit être échantillonné avec une période d'échantillonnage inferieur à la moitié de la période du signal sinusoïdal de fréquence maximale.il est établit dans la règle suivante [21]:

$$T < \frac{1}{2f_{max}} \quad \text{(IV.4)}$$

Ou :

T : la période d'échantillonnage.

f_{max} : La fréquence maximale de la décomposition du signale.

Tel que : $f_{max} = 4f_c = \frac{4w_c}{2\pi} = \frac{\pi}{4w_c}$ (IV.5)

Ou :

f_c : La fréquence de coupure du système de premier ordre en Hz.

w_c : La fréquence de coupure en rad/s.

En cas de deuxième ordre, on utilise les mêmes arguments dans un système qu'est sous la forme suivante :

$$G(s) = \frac{{w_n}^2}{s^2 + 2\xi w_n s + {w_n}^2} \quad \text{(IV.6)}$$

Ou :

w_n: La fréquence naturelle du système rad/s.

ξ : Facteur d'amortissement.

On suppose que $\xi \leq 1$ devrait être échantillonnée avec une période d'échantillonnage qui respecte le critère suivant [22] :

$$T < \frac{\pi}{4w_n} \quad \text{(IV.7)}$$

Pour notre système :

$$G(s) = \frac{187.7}{s^2 + 29.925s + 187.7} \quad \text{(IV.8)}$$

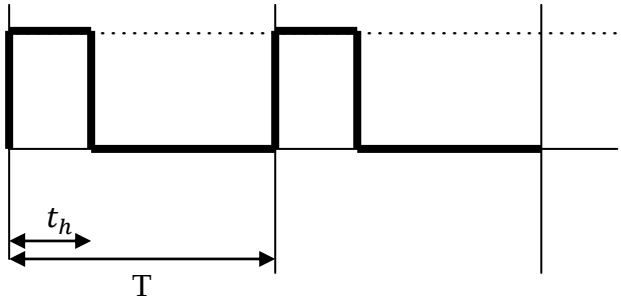
On a: $w_n = \sqrt{187.7} = 13.7003$ rad /s (IV.9)

D'après la relation (IV.5) on trouve que : $T < \frac{\pi}{4 * 13.7003}$ (IV.10)

Donc : $T < 0.0573\ s$ (IV.11)

- **l'USART :**

  On a utilisé le module USART pour délivrer PWM avec une fréquence de base 1KH, et pour commander la vitesse de moteur à travers un driver.



T: Le pseudo période.

t_h: Période change.

La vitesse de moteur est varié pendant la variation de largeur et la fréquence de signaler numérique de PWM, c-à-d que :

0 bit ⟶ 0tr/min.

127 bit ⟶ vitesse égale 750tr/min

255 bit ⟶ vitesse maximale (1500tr/min)

Dans la figure ci-dessous on a représenté la variation des largeurs de PWM pendant le nombre de bit.

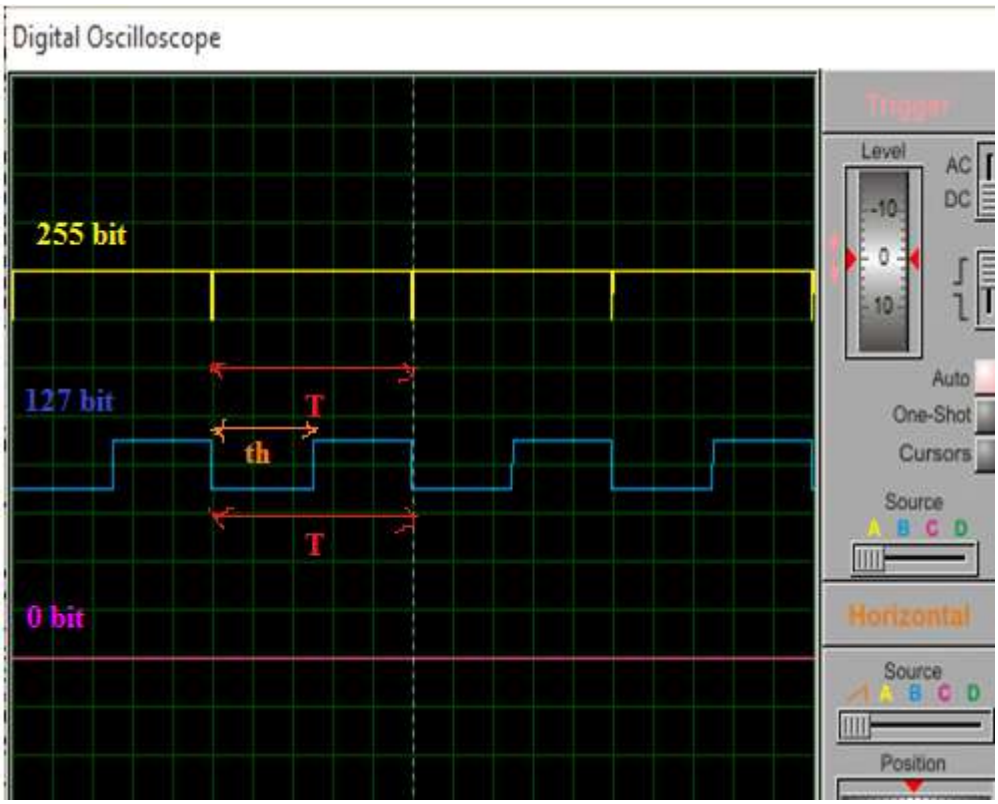
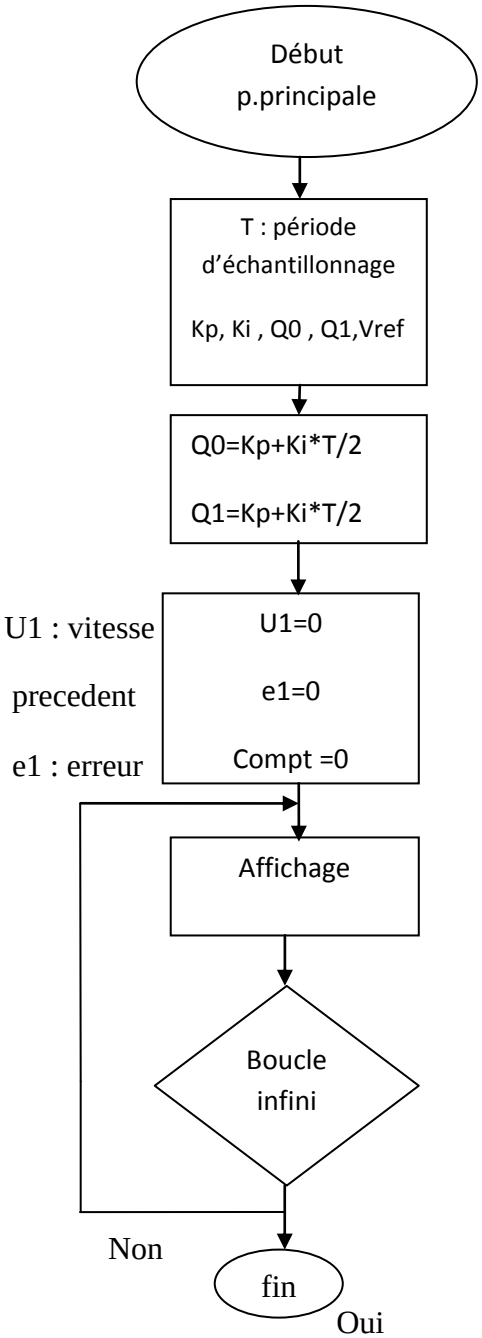
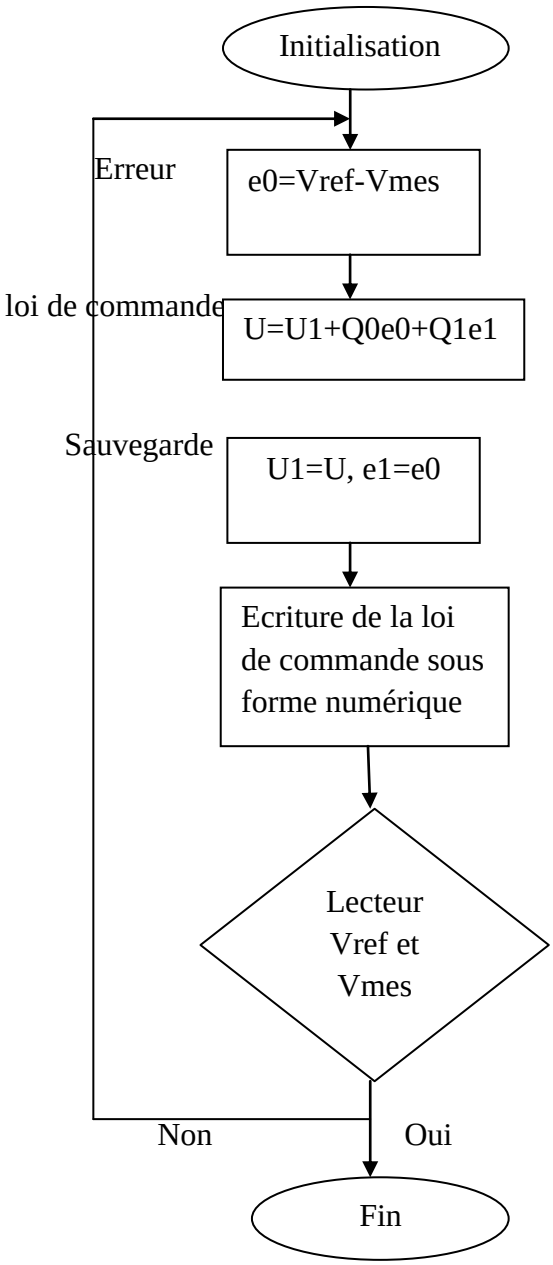


Figure (IV.21): Variation des largeurs de PWM.

D'autre part, la partie software est représente par l'organigramme suivant :



Figure(IV.22) : Organigramme de programme principale.



Figure(IV.23) : Organigramme de programme de la génération de la commande.

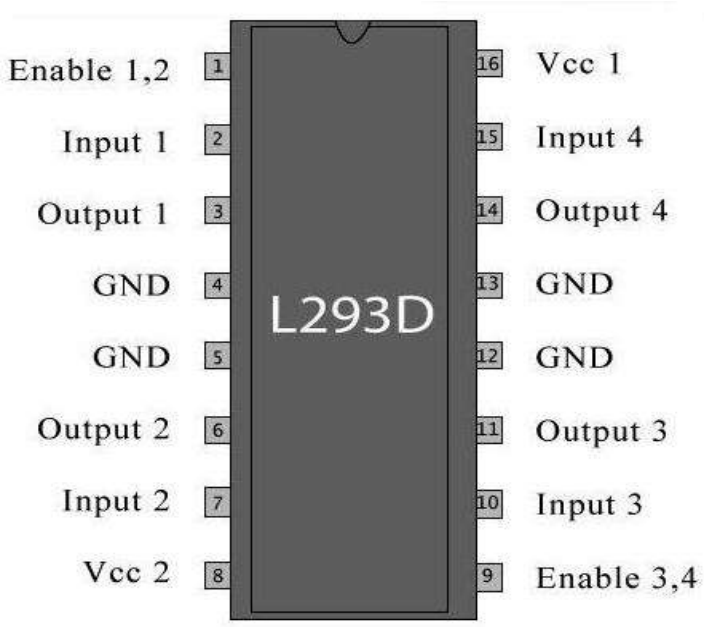
IV.3.3.bloque de driver :

Afin de passé de faible puissance à une grande puissance, on a ajouté entre le microcontrôleur pic 16F877A et le moteur un driver L293D.

Le L293D sont quadruples demi-conducteurs H haute intensité, est conçu pour fournir des courants d'attaque bidirectionnels jusqu'à 600 mA à une tension de 4,5V à 36V.[23]

Caractéristiques L293D :

- Large gamme de tension d'alimentation : 4,5V à 36V.
- Entrée-logique séparée d'alimentation.
- Arrêt thermique.
- Entrées-immunité Haute-bruit.
- Courant de sortie 1A canal par (600 mA).
- Courant, sortie crête 2 A par canal (1,2A).
- Pince de sortie pour Diodes inductif de suppression des transitoires.



Figure(IV.24) : Schéma de L293D.

IV.3.4.bloque d'encodeur incrémentale :

L'encodeur incrémentale permet de donner des informations sur la position d'un mobile et sa vitesse.

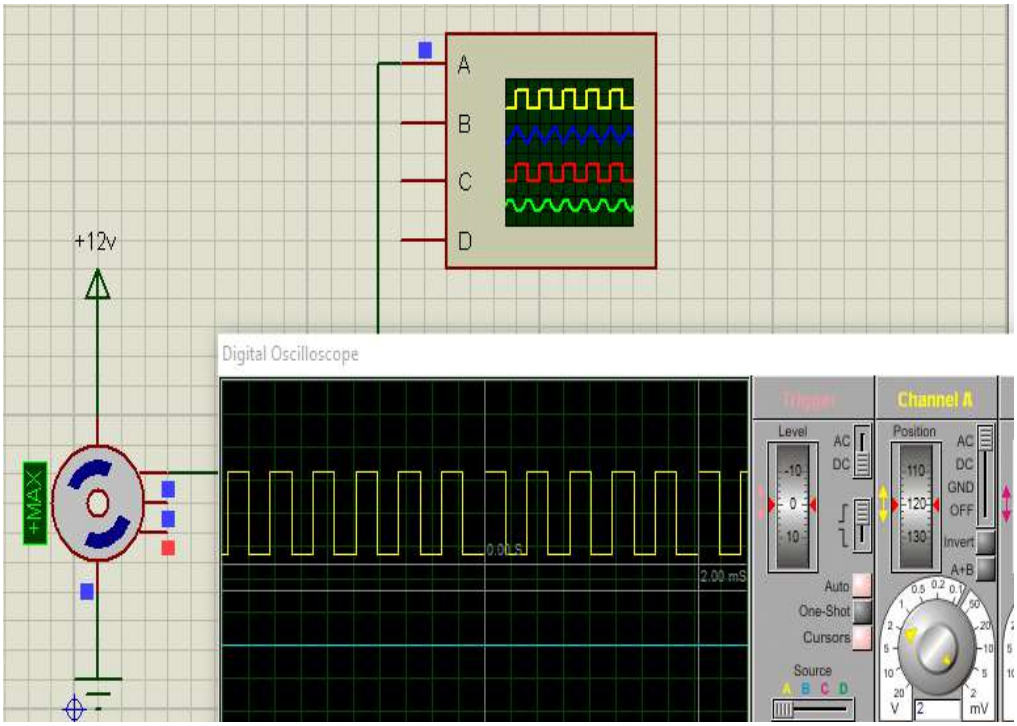


Figure (IV.25) : Simulation de l'encodeur

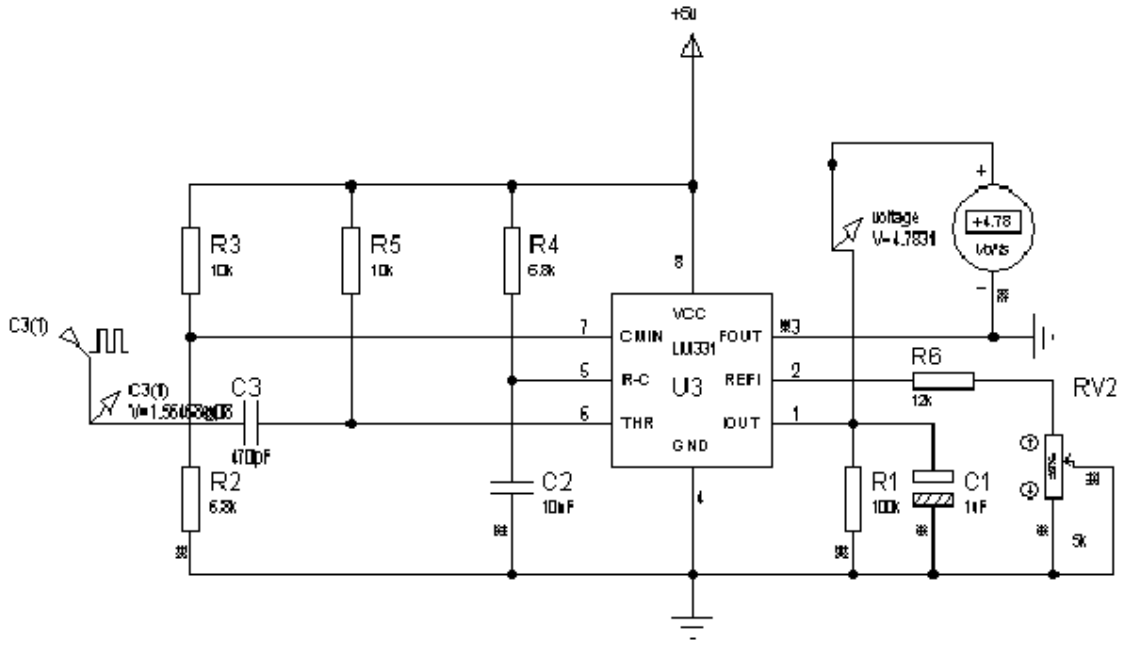
A partie de cette figure on a calcule la vitesse de MCC :

$$\theta = \frac{\text{compteur}}{100} = \frac{5}{100} = 0.05 \qquad \text{(IV.12)}$$

$$V = \frac{0.05}{2} * 60 = 1500 \text{ tr/ min} \qquad \text{(IV.13)}$$

IV.3.5.Bloque de LM331 :

Afin de lire la vitesse d e moteur, on utilise LM331 qu'il assure la conversion de la fréquence d'un signal périodique en une tension continue cette tension attaque le microcontrôleur pic16F877A.



Figure(IV.26) : Représentation de circuit LM331.

❖ **Calcul de la tension de sortie :**

La tension de sortie varie en fonction de la fréquence d'entrée, selon la formule suivante :

$$Vs = Fe * 2.09 * \left(\frac{R1}{R6+RV2}\right) * R4 * C2 \quad (IV.14)$$

Vs: la tension de sortie.

Fe: la fréquence d'entrée.

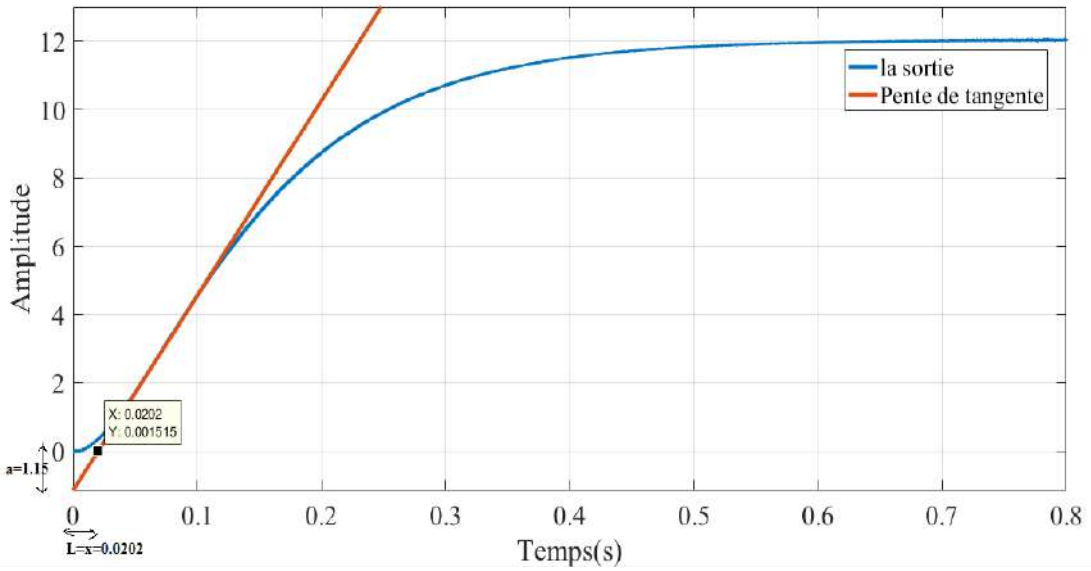
Si par exemple Fe=5Khz à l'entrée du circuit LM331, alors suivant l'équation ci-dessus la tension de sortie sera Vs=4.75V.

IV.4. Détermination des paramètres Kp et Ki du Régulateur numérique :

Plusieurs méthodes (placement de pole, Chien-Hrones-Reswick……) sont envisagées dans les littératures, parmi ces méthodes on a choisi la méthode du Zeigler Nichols**.**

IV.4.1.Méthode de Zeigler - Nichols :

Cette méthode permet de déterminer les paramètres Kp et Ki, pour cela on prend la réponse de la sortie (Vs=f(t).u(t)) puis on trace une droite tangente à la courbe de la fonction de transfert comme il est montré sur la figure suivante. Les paramètres **a** et **L** sont obtenus graphiquement d'après la figure(III.12).



Figure(IV.27): Réponse d'un système.

Selon la figure (IV.26) ces paramètres sont donnée par **a**=1.15 et **L**=0.0202. Les gains du compensateur sont alors obtenir à partir de tableau (III.1) :

$$Kp = \frac{0.9}{a} = \frac{0.9}{1.15} = 0.78 \qquad \text{(IV.15)}$$

$$Ki = \frac{Kp}{3L} = \frac{0.78}{3*0.0202} = 12.87 \quad \text{(IV.16)}$$

Si on considère que la période d'échantillonnage du système est T =390e-6s, on peut finalement obtenir les constantes Q0 et Q1 du régulateur PI (voir les équations (III.40), (III.41))

$$Q0 = Kp + \frac{T}{2}Ki = 0.782 \quad \text{(IV.17)}$$

$$Q1 = -Kp + \frac{T}{2}Ki = -0.777 \quad \text{(IV.18)}$$

IV.4.2.Implémentation du régulateur PI dans µc :

- **Partie software:**

La partie software, après détermination des paramètres Kp et Ki est représentée par le programme sur mikroC suivant :

```
double a=1.15,l=0.0202;

double Q0=0.,Q1=0.,e1=0.,m0=0.,m1=0.,e0=0.0,T,TAD=4e-6,k;

double duty_cycle=0.0,setpoint=0.0,Input=0.0;

void interrupt(){

 if( PIR1.ADIF) {

 e0=setpoint-Input;        //calcule d'erreur

 m0 = m1 + Q0*e0 +Q1*e1;

 m1 = m0;

e1 = e0;

duty_cycle=m0;

if(duty_cycle>255.) duty_cycle=255.;

if(duty_cycle<0.) duty_cycle=0.;
```

```
porta.ra3=1;

PWM1_Start();        //démarrage PWM

PWM1_Set_Duty(duty_cycle);

PIR1.ADIF =0;

porta.ra3=0;

TRISD=0x00;

PORTD=0x00;

  } }

void main(){

double Kp=(0.9/a);  //kp:

double Ki=kp/(3.0*l);

T=19.7+12.*TAD+2.*TAD;

      //19.7:phase d'acquisition

      //12*Tad: phase de conversion

      //2*Tad:  temps mort

T=390e-6;

Q0 = Kp + Ki*T/2.  ;

Q1 = -Kp + Ki*T/2. ;


ADCON0 = 0x81;        //Fosc=8mhz   Fad=8/32=250khz  Tad=4us

ADCON1 = 0x04;        // A0,A1: analogiques

INTCON.GIE=1;

INTCON.PEIE=1;
```

```
 PIE1.ADIE=1;

PIR1.ADIF=0;

PORTA = 0xFF;

TRISA = 0x03;         // RA0 i RA1 are analog inputs

PWM1_Init(1000);         //Initializing the PWM1 module at 1kHz

/*ADC_Init();

//PORTA.RA3=0; */

 while(1) {

        PIR1.ADIF=1;

        setpoint=ADC_Read(1);         //valeur consigne

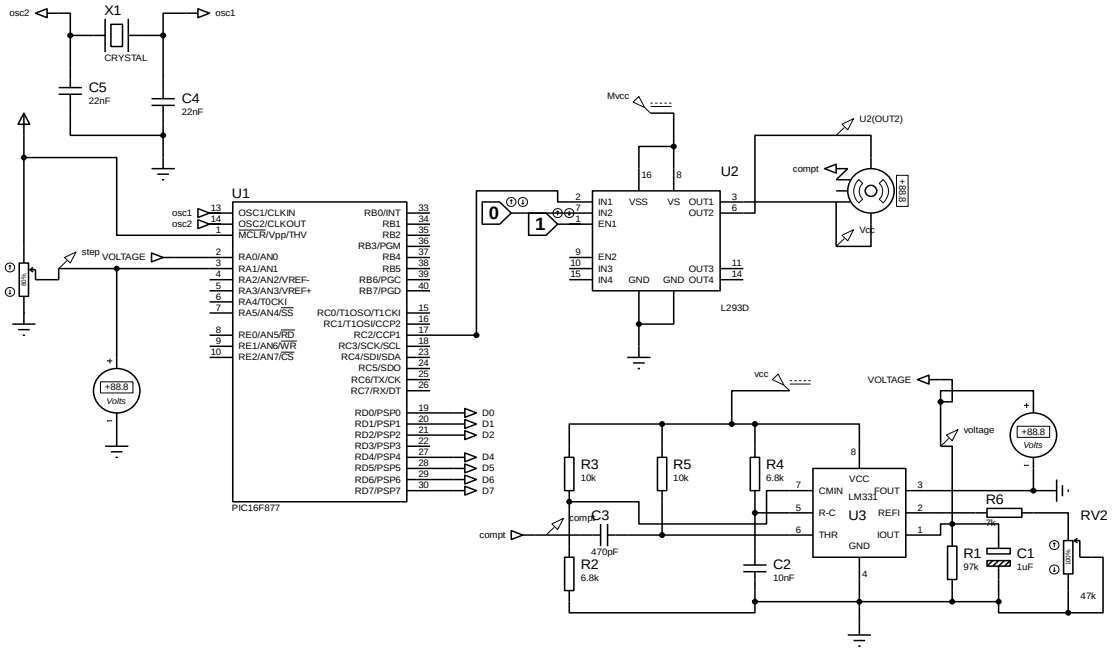
       PIR1.ADIF=0;

        Input= ADC_Read(0);         //valeur de retour V-to-F convertissuer lm331}

}
```

- **Partie hardware:**

Cette Partie est représentée sous ISIS par le montage suivant:



Figure(IV.28) : Schéma électrique complet du régulateur PI.

D'après les résultats, on remarque la diminution du temps de réponse (Tr=0.25s) avec un dépassement de 15% :

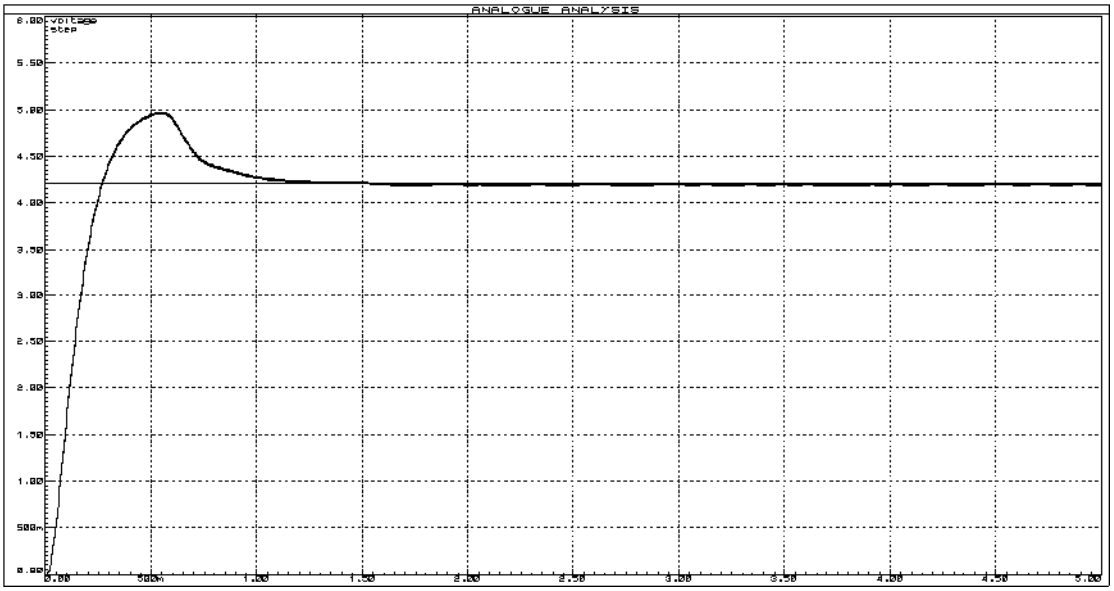
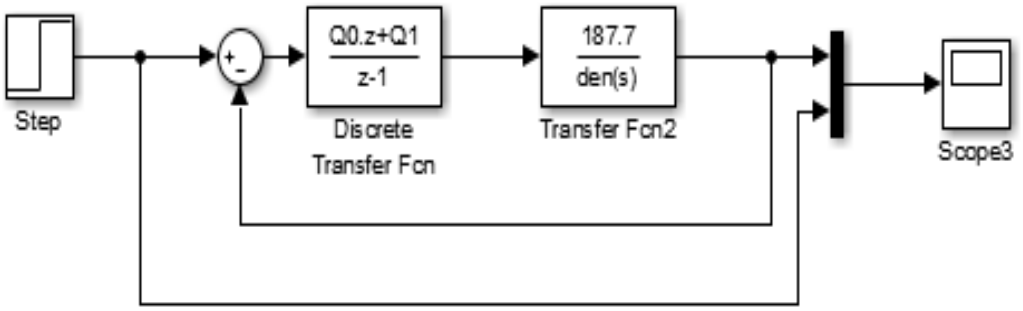


Figure (IV.29) : La réponse de système sous proteus.

IV.4.3.Implémentation de la régulation sous Simulink :

La régulation de notre procédé est assurée par le schéma Simulink ci-dessous qui regroupe la consigne, le comparateur, le correcteur PI en Z, la fonction de transfert de MCC.



Figure(IV.30) : Schéma bloc globale de régulation de MCC par PI en Z**.**

D'après la simulation sur matlab on obtient le signal de vitesse suivant :

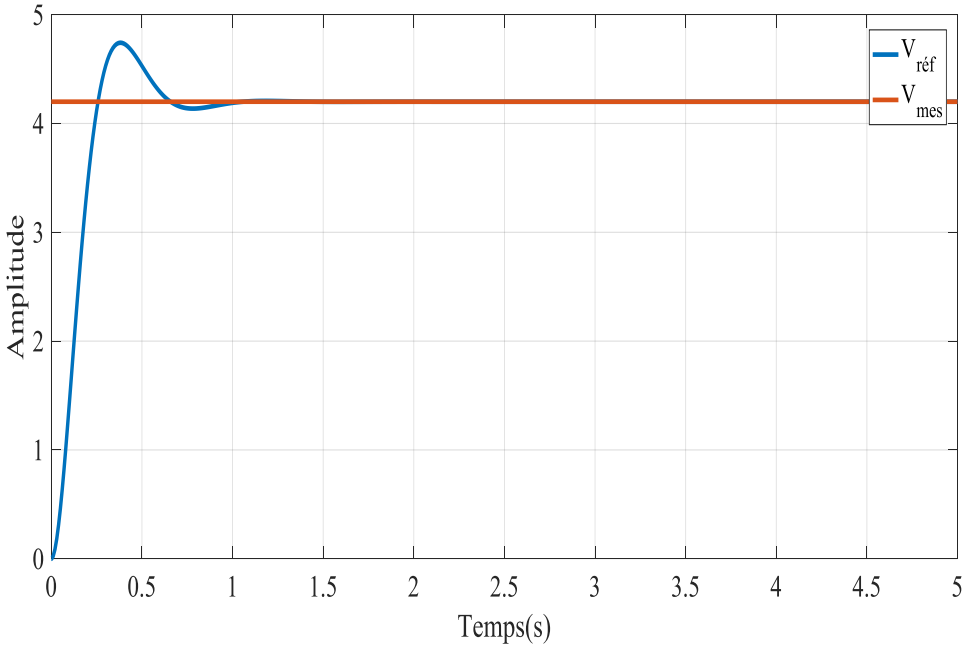


Figure (IV.31) : La réponse de système sous simulink.

Comparaison:

D'après implémentation de régulateur numérique PI dans le microcontrôleur, et d'après les résultats de simulation obtenu sous le proteus et logiciel matlab, on remarque que les deux réponses sont identiques(Tr=0.25s avec un dépassement égale a 15%).

IV.5.Conclusion :

Dans ce chapitre nous avons expliqué toutes les étapes citées précédemment qui sont concrétisé par le programme proteus ISIS et MATLAB.

La première partie a était consacré a l'identification de système pour déterminer les paramètres de MCC.

La deuxième partie a était consacré a la configuration des paramètres Kp et Ki par la méthode de Zeigler - Nichols et a l'implémentation de la régulation sous matlab et proteus pour commander la vitesse de MCC.

.

Conclusion générale

Le but principal de notre travail concernait l'étude et simulation et implémentation d'un régulateur numérique PI dans microcontrôleur PIC 16F877A, pour la commande d'un moteur à courant continu. Les moteurs à courant continu sont très utilisés de nos jours, notamment dans les applications automobiles, dans des applications de faible puissance utilisant des batteries ou encore pour la traction électrique.

Les structures et les principes de fonctionnement des machines à courant continu sont relativement bien connus. Aussi, il y a peu de travaux de recherche qui sont développés sur ce type de machine depuis plusieurs années. En fait, l'essentiel de la recherche sur ces machines est mené par des industriels. Cette recherche porte plus particulièrement sur les matériaux qui sont utilisés pour réaliser les collecteurs et les balais. [1]

D'après l'étude des différents types de moteur et suivant ces caractéristiques, on a vu que le moteur à excitation indépendante c'est lui qui s'adapte pour notre projet, il est caractérisé par une vitesse réglable par tension et indépendante de la charge.

Dans ce travaille, nous avons présenté l'étude du moteur à courant continu, puis nous avons implémenté le régulateur PI dans un microcontrôleur PIC pour la commande de la vitesse de la machine à courant continu, et enfin la simulation de notre modèle.

Dans le premier chapitre, nous avons rappelé les principaux éléments de constitution de la machine à courant continu. Puis nous avons présenté les différents types du moteur à courant continu ainsi que les avantages et les inconvénients.

Dans le deuxième chapitre, nous avons rappelé le microcontrôleur PIC 16F877A et son architecture, leur caractéristique et ses différents registres.

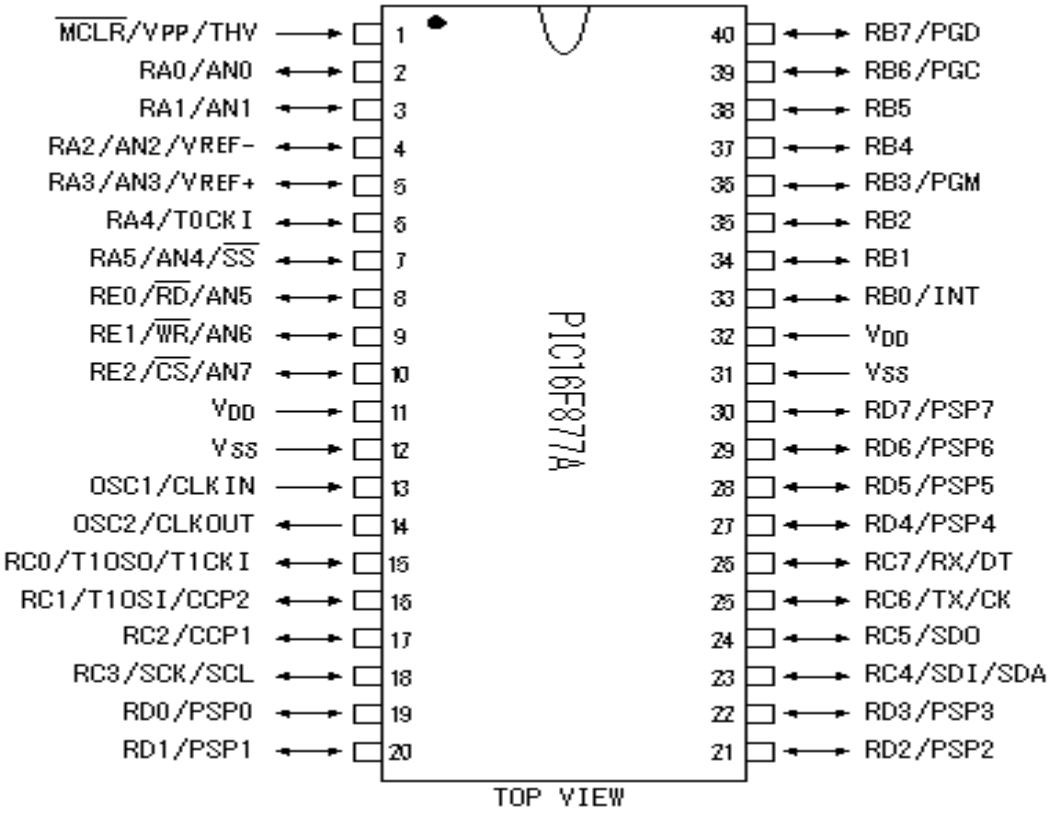
Dans le troisième chapitre nous avons fait une identification sur ce modèle, pour déterminer tout les paramètres utilisés de ce système, après on a utilisée la méthode de Ziegler –Nichols pour obtenir les paramètres Kp et Ki.

Dans le quatrième chapitre nous avons identifié le moteur a courant continu pour obtenu la fonction de transfert. D'après on a appliqué la méthode de Zeigler – Nichols sur la

fonction de transfert pour déterminé les paramètres de régulateur PI, et enfin l'implémentation de la régulation sous matlab et proteus pour commander la vitesse de MCC.

Annexe

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

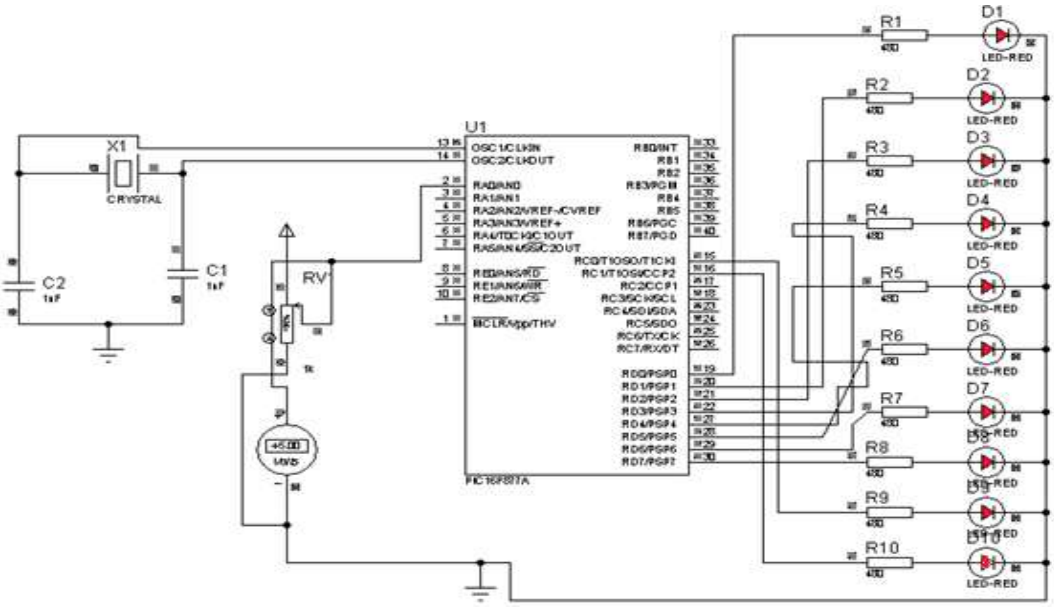


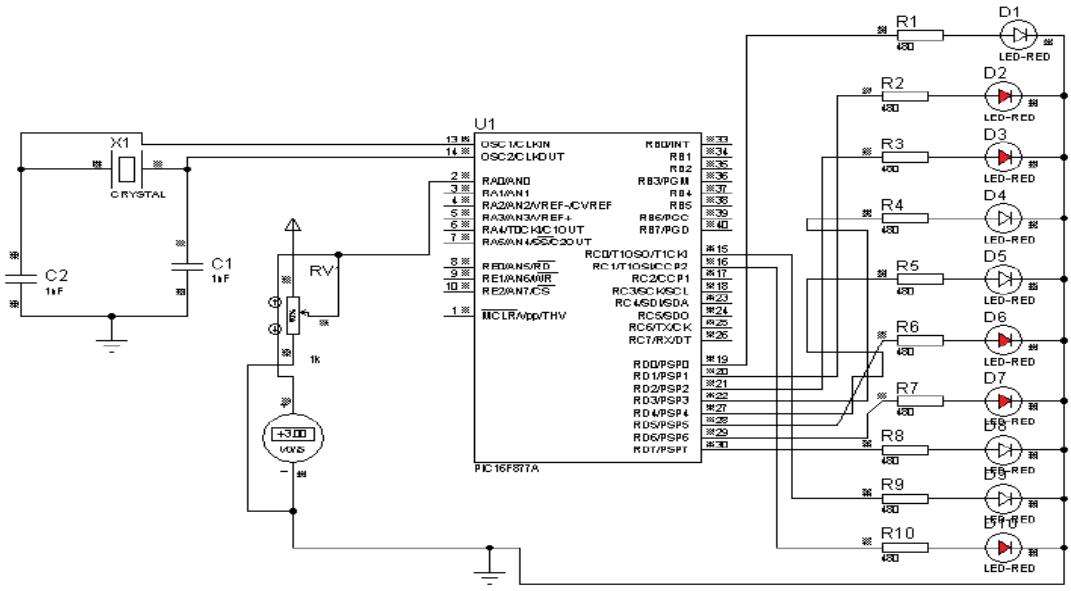
L'annexe 2: tableau de configuration des pin input du convertisseur A/N

PCFG <3:0>	AN7	AN6	AN5	AN4	AN3	AN2	AN1	AN0	VREF+	VREF-	C/R
0000	A	A	A	A	A	A	A	A	VDD	VSS	8/0
0001	A	A	A	A	VREF+	A	A	A	AN3	VSS	7/1
0010	D	D	D	A	A	A	A	A	VDD	VSS	5/0
0011	D	D	D	A	VREF+	A	A	A	AN3	VSS	4/1
0100	D	D	D	D	A	D	A	A	VDD	VSS	3/0
0101	D	D	D	D	VREF+	D	A	A	AN3	VSS	2/1
011x	D	D	D	D	D	D	D	D	—	—	0/0
1000	A	A	A	A	VREF+	VREF-	A	A	AN3	AN2	6/2
1001	D	D	A	A	A	A	A	A	VDD	VSS	6/0
1010	D	D	A	A	VREF+	A	A	A	AN3	VSS	5/1
1011	D	D	A	A	VREF+	VREF-	A	A	AN3	AN2	4/2
1100	D	D	D	A	VREF+	VREF-	A	A	AN3	AN2	3/2
1101	D	D	D	D	VREF+	VREF-	A	A	AN3	AN2	2/2
1110	D	D	D	D	D	D	D	A	VDD	VSS	1/0
1111	D	D	D	D	VREF+	VREF-	D	A	AN3	AN2	1/2

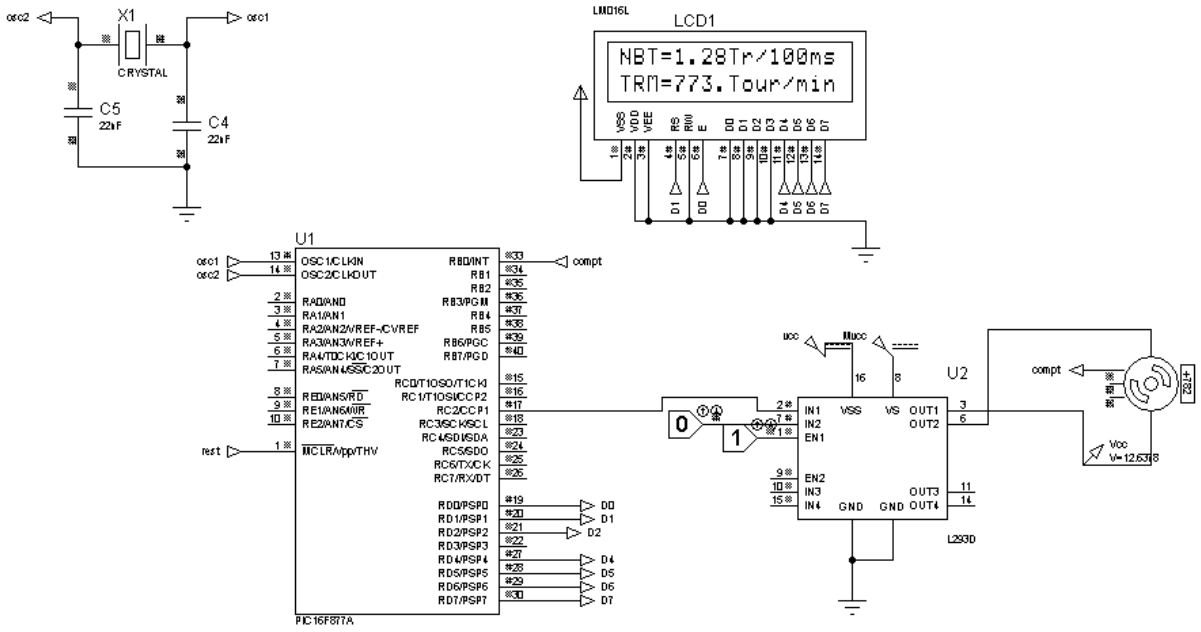
A=analogique input D=digitale input

L'annexe 3:schémas électriques des conversions analogiques numériques pour 5v et 3v.





L'annexe 4: Schéma de la simulation pour PWM égale a 128 bits.



Bibliographie

[1] H.BOUDJELLAH, "Etude et simulation d'un variateur de vitesse commande un moteur à courant continu ", mémoire master, Université MERBAH Ouargla, 2015/2016.

[2] S.LAIMACHE. F.MOUHOUBI."Réalisation d'un régulateur numérique PI pour commander un moteur à courant continu à base d'un microcontrôleur PIC16f877a", mémoire master, Université Dr YAHIA FARES de Médéa, 2015/2016.

[3] D. TLMSANI. Y. OULD BABA ALI."Implémentation de l'algorithme PI dans un microcontrôleur 16f877a pour la commande d'un moteur à courant continu", mémoire master, Université Dr YAHIA FARES de Médéa, 2014/2015.

[4] A.LEBATT, N. MOUSSA NADJEM "commande d'un moteur à courant continu en utilisant un circuit FPGA", mémoire master, Université Dr YAHIA FARES de Médéa, 2016/2017.

[5] [en ligne], 29/05/2018, Enseignement.allais_en/.

[6] N. MAATER, L.BESEKHRIA, "conception d'une carte de commande de la machine à courant continu par un microcontrôleur pic16F877A ", institut nationale de la formation professionnelle CHAHIB BOUKHLOUF MOHAMED BEN EL HADI, Batna, 2009.

[7] A.AYAD, "Etude et réalisation d'un commande MLI pilotant un hacheur série", Mémoire de Master, Université Abou BekrBelkaid Tlemcen, 2014.

[8] S. MOUSSAOUI,"commande floue d'un moteur à courant continu (Mcc) comparaison avec la commande PID", mémoire master, Université Abderrahmane Mira-Bejaia, 2016/2017.

[9] [en ligne], 30/04/2018, https://www.researchgate.net/publication/314236426.

[10][enligne],29/04/2018http://www.lmdindustrie.com/content/guides/achat/Codeurs_optiques_201003.pdf.

[11] DEHABA HOUSSEM." Réalisation d'un cardiofréquencemètre à base du PIC 16F877", mémoire master Université FERHAT ABBAS –SETIF 2014.

[12] A.BRAHIMI, H.GUEZOULI," Etude et réalisation d'une carte de commande à base d'un microcontrôleur PIC 16f877 pour ponts redresseurs triphasés à thyristors", Mémoire de Master, Université Abou-Bekr Belkaïd Tlemcen, 2013 – 2014.

[13] [en ligne] 15/05/2018 https://components 101.com.

[14] R.SAADI, T. GUESBAYA," Réalisation de carte à microcontrôleur pour le contrôle de bras manipulateur via un pc", mémoire master, Université Mohamed Khider Biskra, 2009/2010.

[15] S.BERNAOUI "Etude et réalisation d'un kit de développement d'applications à base de pic de la famille 16Fxxx", mémoire master, Université Mohamed Khider Biskra, 2012.

[16] [en ligne], 30/04/2018, https://www.researchagate.net/.

[17] [en ligne], 28/05/2018, https://sti.discip.ac.caen.fr/.

[18]. [en ligne], 28/05/2018, https://www.technologlog nepro.com.

[19] [en ligne], 28/05/2018, http://www.technologuepro.com/ .

[20]. H. BOUICHE, M. BRAHIMI, ‘‘commande PID d'un moteur à courant continu ’’, mini projet, université Abderrahmane Mira, Bejaia, 2010.

[21] Mohamed ARHOUJDAM, ‘‘mise en oeuvre d'une régulation de vitesse d'un MCC’’, université Mohamed V souissi, 2013.

[22] [en ligne], 22/04/2018, http://www.apc.univ-paris7.fr/~colonges/identifbf5.doc.

[23] [en ligne], 22/04/2018, https://electrosome.com/dc-motor-l293d-pic-microcontroller/.