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 Génie Électrique

Spécialité : Automatique

Option : Automatique et Systèmes

THEME

Implémentation d'un régulateur floue dans un automate s7-300 pour la commande de la vitesse d'une machine asynchrone

Proposé et dirigé par,

Mr. Bettache Farouk

Préparé et présenté par,

Benyahia Abdesselam

Mouhoubi Salim

Remerciements

Tout d’abord, nous tenons à remercier **ALLAH** le tout puissant pour la volonté, la santé et la patience, qu'il nous a donné durant toutes ces longues années.

Nous tenons à exprimer toute notre gratitude et nos sincères remerciements à notre promoteur Dr **BETTACHE FAROUK**, pour avoir dirigé ce travail, pour ses grandes compétences scientifiques ainsi que pour ses conseils, ses remarques toujours constructives**.** aux enseignants qui ont contribué à la formation de notre connaissance depuis la première classe du primaire jusqu'à aujourd'hui , spécialement **Dr BENREKIA** Allah yarahmou.

Nos sincères remerciements s’adressent ensuite aux membres de jury, pour l’intérêt qu’ils ont manifesté pour notre travail et d’avoir acceptés de le juger.

Dédicaces

Je dédie mon modeste travail à tous Ceux qui ont pensé à mon avenir tous ceux qui se sont fatigué pour mon bonheur tous ceux qui se sont sacrifies pour mon éducation et ma culture

Mes très cher parents Dalila et Zinouba & Djamel pour toute l'affection tous les sacrifices et les encouragements

Très cher et doux petite princesse Hana pour son amour et son soutien illimités

Qu'ils n'ont pas cessé de nous prodiguer tout au long de nos études.

A mes chers soeurs Yassmine Ilham

A mon chèr frére Ismail et mon petit neveu Yanice .

A tout ma famille ainsi tout ma promotion 2020

A ceux qui m'aime & ceux que j'aime.

Dédicaces

Je rends grâce, à ***mon Dieu*** *de m'avoir donnée la force, la volonté, l'intelligence et la sagesse d'être patiente dans mes études.*

Je dédie ce travail à : Mes ***chers parents*** *qui m'ont beaucoup aidé et qui se sont sacrifiés pour mon bien et qui m'ont encouragé et soutenu le long de ma vie et durant mon cursus.*

A mon chers ***frère Walid*** *que j'aime beaucoup.*

A mon chers ***frère Djamel*** *et* ***sa femme****, et toute ma famille.*

A mon cher ami ***MOKHBAT AMINE****, chez qui j'ai trouvé l'entente dont j'avais besoin, et toute sa famille.*

A mon cher ami ***TALAMALI TARIK****, l'ami d'enfance qui est toujours là pour moi, et toute sa famille.*

A mes chers amis : ***Riadh*** *,* ***FAHD****,* ***Hocine*** *,****Adel*** *,****Dadou****,* ***Salim*** *Vous êtes les meilleures.*

A mes camarades : ***Youcef,Ismail,Ahmed,Mohammed****, et A tous mes collègues de la promotion d'automatique et systèmes. 2019/2020 ; ainsi que tous ceux qui ont contribué de près ou de loin à la réalisation de ce travail, spécialement Dr Cristian Guarnizo Lemus.*

Et tous ceux qui m'estiment et pensent à moi.

MOUHOUBI Salim

الملخص

يختص هذا المشروع في برمجة آلة غير متزامنة بواسطة (SIMULINK MATLAB)والتحكم فيه بالمتحكم التناسبي التكاملي PI و متحكم المنطق الضبابي (Fuzzy Logic controller) وذلك لضبط العوامل Kp و Ki ثم عمل برمجة بالمنطق الضبابي بواسطة لغة تحكم منظمة (SCL) في برنامج (TIA PORTAL) تحت جهاز التحكم الآلي (S7-300) و التأكد من نتائج المعاملات المذكورة سابقا .

الكلمات المفتاحية: لغة تحكم منظمة ، آلة غير متزامنة ،المتحكم، PI ، المنطق الضبابي ،S7-300.

Résumé

Ce projet consiste à la programmation d'une machine asynchrone par SIMULINK MATLAB avec un contrôleur proportionnel intégré PI et avec un contrôleur PI-floue pour ajuster les facteurs Kp et Ki, puis la programmation en logique floue en utilisant un langage de contrôle structuré (SCL) dans le logiciel TIA PORTAL sous l'automate de contrôle (S7-300) et vérifions les résultats des facteurs mentionnées précédemment.

Mots-clés : SCL, Logique floue, machine asynchrone, contrôleur PI ,S7-300.

Abstract:

This project involves in programming an asynchronous machine by SIMULINK MATLAB with an integrated proportional controller PI and a PI-fuzzy logic controller to adjust the Kp and Ki factors, then programming in fuzzy logic using a structured control language (SCL) in TIA PORTAL software under the PLC (S7-300) and we check the results of the previously mentioned factors.

KEY WORDS : SCL , Fuzzy Logic, asynchronous machine, PI controller ,S7-300.

SOMMAIRE

Introduction générale…...……..……..……..…01

Chapitre I

Modélisation de la MAS

I.1 Introduction 03

I.2 Principe de fonctionnement 03

I.3 Constitution de la MAS 05

I.3.1 Stator de la MAS 05

I.3.2 Rotor de la MAS 06

I.4 Modalisation et mise en équation 07

I.4.1 Hypothèses simplificatrices 07

I.4.2 Représentation schématique d'un moteur asynchrone triphasé 08

I.5 Modèle mathématique de la MAS 08

I.5.1 Equation mathématique de la MAS 09

I.5.2 Transformation De Park appliquée à la MAS: 12

I.6 Modélisation de l'onduleur de tension 18

I.7 Simulation De La MAS: 22

I.8 Conclusion 24

SOMMAIRE

Chapitre II

Le régulateur PI-Flou pour la commande du vitesse d'une machine Asynchrone

II.1 Introduction : 25

II.2 Généralité sur la commande scalaire 26

II.3 La commande scalaire 26

II.4 Modélisation de la machine asynchrone en régime permanent 28

II.5 Comportement de la MAS en régime permanent 28

II.6 Alimentation en tension 30

II.7 Contrôle en V/f de la machine asynchrone 31

II.8 Simulation de la commande scalaire de la MAS 32

II.9 Résultat de simulation 33

II.10 Principe général de la régulation 34

II.11 Performances des systèmes réglés 34

II.12 Système en boucle ouverte 35

II.13 Système en boucle fermée 36

II.14 La commande PI en régulation de vitesse 36

II.15 Simulation de la commande scalaire de la MAS avec régulateur 37

II.16 Résultats de simulation 38

SOMMAIRE

II.17 Optimisation Floue des paramètres du régulateur PI 39

II.17.1 Principes et définitions de la logique floue 39
II.17.2 Variable linguistique 40
II.17.3 Fonctions d'appartenance 40
II.17.4 Opérations sur les ensembles flous 41
II.17.5 Structure générale du superviseur flou 41

II.18 Description du contrôleur floue 42

II.19 Stratégie de contrôle de vitesse 42

II.19.1 Système de control adaptatif flou 42

II.19.2 Conception de contrôleur floue 43

II.20 Résultats de simulation 45

II.21 Conclusion 47

Chapitre III

Programmation d'un contrôleur flou dans PLC

III.1 Introduction 48

III.2 Présentation de quelques gammes SIMATIC 48

III.2.1 SIMATIC S7 48

III.2.2 SIMATIC S7-300 48

III.2.3 Présentation de l'automate S7-300 49

III.2.4 Description de l'Automate S7-300 50

III.3 Logiciel de programmation « TIA Portal V13 » 50

III.3.1 Présentation du logiciel 50

III.3.2 Conception d'un programme avec TIA PORTAL 51

SOMMAIRE

III.3.3 vue du portal 52

III.3.4 vue du projet 52

III.3.5 création du projet dans TIAPORTAL V13 53

III.4 Régulateur continue avec le FB 41 « CONT_C » 56

III.5 La Programmation d'un régulateur flou sous TIA-PORTAL 56

III.5.1. La Programmation du bloc FB 41..... 56

III.5.2 Description du bloc FB41 (CONT_C) : 57

III.5.3 Le fonctionnement de bloc PID FB41 (CONT_C) :..... 59

III.5.4 Les paramètres du bloc FB 41 « CONT_C » 59

III.6 Langage de programmation SCL 62

III.6.1 L'environnement de développement SCL..... 63

III.6.2 Éditeur/compilateur 63

III.6.3 Débogueur 64

III.6.4 Avantages offerts par SCL 64

III.7 La commande à base de la logique floue 65

III.7.1 La logique floue 65

III.7.2 Ensembles flous: 66

III.7.3 Haies: 67

III.7.4 Opérateurs flous : 70

III.8 Le raisonnement en logique floue: 72

III.8.1 Règles Floues..... 72

III.9 Commande par logique floue : 73

III.9.1 Constitution d'un système floue 73

III.9.1.2 Interface de fuzzification..... 74

III.9.1.3 Degré d'appartenance 74

SOMMAIRE

III.9.1.4 Mécanisme d'inférence floue 75

III.9.1.5 Interface de Défuzzification 75

III.9.1.6 Méthode du centre de gravité 75

III.9.1.7 Méthode du maximum 76

III.9.1.8 Méthode de la moyenne du maximum 76

III.9.2 Inférence Mamdani 76

III.9.3 Inférence Sugeno 77

III.10 La Programmation d'un régulateur flou sous TIA PORTAL 79

III.11 Comparaison du résultats du simulation..... 82

III.12 Conclusion 88

Conclusion générale 89

LISTE DES FIGURES

Chapitre I : Modélisation de la MAS

Figure I-1 : Le sens du flux par la règle de la main droite 4

Figure I-2 : Constitution d'une MAS 5

Figure I-3 : Stator d'une MAS 5

Figure I-4 : Le rotor à cage d'écureuil 6

Figure I-5 : Le rotor bobiné 7

Figure I-6 : Représentation d'une machine asynchrones triphasé 8

Figure I-7 : Représentation schématique d'une machine asynchrone triphasée 9

Figure I-8 : Transformation Des enroulements réels en enroulements..... 12

Figure I-9 :Schéma d'un onduleur de tension triphasé alimentant le stator du MAS..... 18

Figure I.10 : Schéma de principe de la commande «sinus -triangle ». 20

Figure I.11: Chronogramme d'une commande MLI «sinus -triangle» monophasé. 21

Figure I.12 : Schéma de principe d'une commande « sinus -triangle » triphasée..... 21

Figure I-13 : Modèle de la machine asynchrone alimentée en tension 22

Figure I-14 : Résultat du simulation du mas 23

Chapitre II : Le régulateur PI-Flou pour la commande du vitesse d'une machine Asynchrone

Figure II.1 : principe de réglage 27

Figure II.2: réglage avec autopilotage 27

Figure II.3 : Caractéristique couple vitesse 31

Figure II.4 : caractéristique V/f 32

Figure II.5 : Modèle Matlab/Simulink de la Commande Scalaire de la machine asynchrone sans régulateur PI..... 32

Figure II.6.Performances de la MAS sans régulateur PI 33

Figure II.7 : Stabilité du systéme 35

Figure II.8 : schéma bloc d'un systéme en boucle ouvert 35

Figure II.9 : schéma bloc d'un systéme en boucle fermé 36

Figure II.10. Commande scalaire avec le contrôle du rapport V/f 37

Figure II.11. Modèle Matlab/Simulink de la Commande Scalaire de la machine asynchrone avec régulateur PI 37

Figure II.12.Performances de la MAS avec régulateur PI 39

Figure II.13 Les Formes usuelles des fonctions d'appartenance 40

Figure II.14 Architecture du superviseur flou 41

Figure II.15.CONTROLLEUR PI-FLOU 43

Figure II.16 : La strécture d'un controleur PI-FLOU 44

Figure II.17 : Performances de la MAS avec l'application du régulateur PI-FLOU ……. ……. …..46

Chapitre III : Programmation d'un contrôleur flou dans PLC

Figure III.1 API SIMENS S7-300 50

Figure III.2 Constitution d'API S7-300 50

Figure III.3 Organisation pour la création d'un projet sous TIA PORTAL 52

Figure III.4 Vue détaillée du portal 53

Figure III.5 Vue du projet 54

Figure III.6 Schéma fonctionnel de CONT_C 57

Figure III.7 environnement du SCL 61

Figure III.8 Les blocs créés avec SCL 63

Figure III.9 Représentation du sous-ensemble flou F des petits entiers 65

Figure III.10 Comparaison de l'appartenance de la température en logique classique vs la logique floue 67

Figure III.11 Exemple de haies 68

Figure III.12 Exemples de haies et représentation mathématique 68

Figure III.13 Exemple d'opérations sur des ensembles flous de façon graphique. 69

Figure III.14 Aperçu synoptique d'un système flou 72

Figure III.15 Calcul du point d'appartenance graphiquement 73

Figure III.16 Agrégations des règles de sortie par coupure 75

Figure III.17 Conséquences de style Sugeno 76

Figure III.18 Evaluation des conséquences de style Sugeno 76

Figure III.19 L'organigramme du programme SCL 77

Figure III.20 Le bloc du controleur Flou. 78

Figure III.21 Le bloc FB41. 78

Figure III.22 Data bloc du programme 79

Figure III.23 Résultats du 1er teste 80

Figure III.24 Résultats du 2éme teste 81

Figure III.25 Résultats du 3éme teste 82

Figure III.26 Résultats du 4éme teste 83

Figure III.27 Résultats du 5éme teste 84

Liste des tableaux

Tab II.1 Régles du Δkp et Δki 44

Tab III.1 Les différents types de régulation PID sous TIA PORTAL. 55

Tab III.2 Paramètres d'entrée du BLOC FB41 60

Tab III.3 Paramètres de sortie du BLOC FB41 60

Tab.III.4 Les types de bloc supporté par SCL 64

Tab.III.5 Comparaison entre la logique classique et la logique floue 66

Tab III.6 Opérations sur des ensembles flous de façon mathématique. 70

Notation et Symboles

- **Acronymes**

- A.P.I **:** Automate Programmable Industriel.
- DB : Bloc de données.
- FB : Bloc fonctionnel.
- FC : Bloc Fonctions.
- CPU : Central Processing Unit.
- FBD : Function Blog Diagram.
- MLI : Modulation de Largeur d'impulsion.
- MAS : Machine Asynchrone.
- MPI : Interface Multi Points.
- PI : Proportionnel Intégrale.
- PLC : Programmable Logic Controller.
- SCL : Structured control language.
- SIMATIC : Siemens automatic.
- TIA PORTAL : Totally Integrated Automation
- TOR : Tout Ou Rien.

- **Paramètres de modélisation de la machine**

- Repère sa, sb, sc : Axes magnétique liés aux enroulements triphasés statorique ;
- (α, β) : Axes de référentiel de Concordia ;
- (d, q) : Axes de référentiel de Park ;
- Cem : Couple électromagnétique ;
- Cr : Couple résistant ;
- f : Coefficient de frottement visqueux.
- σ : Coefficient de dispersion (coefficient de Blondel)
- Tr : Constante de temps statorique et rotorique
- M : Inductances mutuelles cyclique entre stator et rotor.
- Lr : Inductances propres cyclique respectivement d'une phase statorique et d'une phase rotorique.
- $[Lss]$: Matrice des Inductances propres d'une phase statorique.

- $[Lrr]$: Matrice des Inductances propres d'une phase rotorique
- $[Msr]$: $[Msr]\ T$: Matrice des Inductances mutuelles du couplage entre stator et rotor.
- J : Moment d'inertie
- p : Nombre de pair de pole
- $[Rr]$: Résistance statorique et rotorique.
- (x, y) : Repère lié au champ au rotor.
- θ : Position angulaire du rotor par rapport au stator.

- **Grandeurs électrique**

- Vs : Tension statorique des phases d, q.
- Vrd : Tension rotorique de phase directe.
- Vrq : Tension rotorique de phase quadratique.
- Is : Tension statorique des phases d, q.
- Ir : Tension rotorique des phases d, q.

- **Grandeurs magnétiques**

- $\phi sa, \phi s$: Flux statorique des phases a, b et c ;
- $\phi ra, \phi r$: Flux rotoriques des phases a, b et c ;
- $\phi s\alpha, \phi s\beta$: Flux statoriques des phases α, β ;
- ϕr : Flux rotoriques des phases α, β ;
- ϕsd : ϕsq : Flux statoriques des phases d, q ;
- ϕr : Flux rotoriques des phases d, q.

- **Grandeurs mécaniques**

- Ω : Vitesse mécanique de rotation ;
- wr : Pulsation électrique correspondant à la vitesse de rotation ;

- **Grandeurs de commande et de l'observateur**

- $w*$: La valeur de référence de la vitesse de rotation.
- Kp : La composante proportionnelle du correcteur PI.
- Ki : La composante Intégrale du correcteur PI.

- **Les paramètres de la machine asynchrone**

Les données de la machine asynchrone sont :

- La tension 380V/220V
- Courant statorique nominal 6.9 A
- La fréquence f = 50 Hz
- Nombre de paire de pole p = 2
- Nombre de phases 3
- La vitesse nominale = 1420 Tr/mn
- Cr = 3 N.m
- J = 0.031 kg.m²
- ws = 314 rad/s
- Rr = 3.85 **Ω**
- Rs = 4.85 **Ω**
- Ls = 0.274 H
- Lr = 0.274 H
- Lr = 0.274 H

Chapitre I

Modélisation de la MAS

I.1.Introduction

La machine asynchrone, appelé aussi « machine à induction », est une machine électrique à courant alternatif sans connexion entre le stator et le rotor a été inventé aux USA par Nikola Tesla vers 1890. Sur le plan constructif, il est la plus simple des machines électriques. [3]

L'étude analytique de la MAS s'appuie sur deux approches différentes. La première est basée sur les grandeurs physiques, telles que les inductions magnétique et les densités de courant électriques, elle s'adresse de plus près aux constructeurs. Elle est plus précise mais très complexe . La deuxième est basée sur les grandeurs électriques, présente moins de précision et s'adresse aux concepteurs d'ensembles machine _ convertisseur.

Dans la première partie de ce chapitre, on va définir la machine asynchrone et Présenter le modèle mathématique de la machine asynchrone, cette machine fonctionne en moteur dont les phases du stator sont alimentées par un réseau triphasé de tension sinusoïdale à fréquence et amplitude constante au par un onduleur de tension au courant à fréquence et à amplitude variable.

Dans la deuxième partie de ce chapitre nous traiterons l'association onduleur-machine et nous présenterons un modelé général l'associant la machine asynchrone a son alimentation .

I.2.Principe de fonctionnement

Le principe de fonctionnement d'un moteur asynchrone repose :

- D'une part sur la création d'un courant électrique induit dans un conducteur placé dans un champ magnétique tournant. Le conducteur en question est un des barreaux de la cage d'écureuil ci-dessous constituant le rotor du moteur. L'induction du courant ne peut se faire que si le conducteur est en court-circuit (c'est le cas puisque les deux bagues latérales relient tous les barreaux).
- D'autre part, sur la création d'une force motrice sur le conducteur considéré (parcouru par un courant et placé dans un champ magnétique tournant ou variable) dont le sens est donné par la règle des trois doigts de la main droite. Comme le montre la figure (I.1).

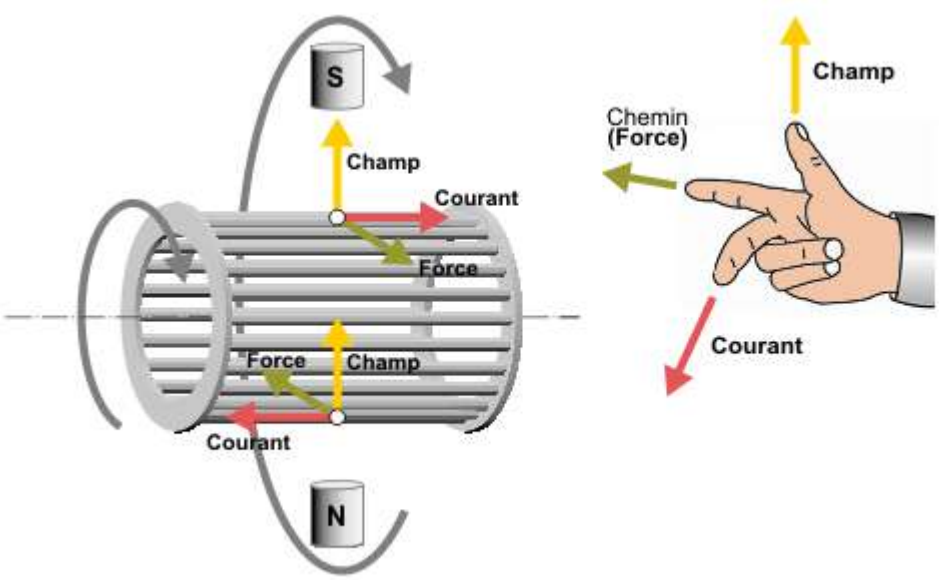


Figure I-1 : Le sens du flux par la règle de la main droite

Comme montré sur le schéma ci-dessus, le champ tournant , à un instant donné, est orienté vers le haut. En considérant deux conducteurs diamétralement opposés, on constate que les courants induits dans ces deux conducteurs sont en sens inverse et, associés au champ magnétique, créent des forces motrices en sens inverse. Le rotor étant libre de tourner sur l'axe X-Y, les deux forces s'associent pour imprimer aux deux conducteurs un couple permettant la rotation de la cage d'écureuil : le moteur électrique est inventé.

Pour entretenir la rotation du moteur, il est nécessaire de faire varier soit le courant dans les conducteurs de la cage, soit le champ magnétique. Dans un moteur asynchrone, c'est le champ magnétique qui varie sous forme de champ tournant créé dans le stator.Au démarrage le

champ tournant balaye les conducteurs de son flux à la vitesse angulaire de synchronisme. Le rotor mis en rotation tend à rattraper le champ tournant. Pour qu'il y ait un couple entretenu au niveau des conducteurs, la variation de flux doit être présente en permanence; ce qui signifie que si les conducteurs tournent à la vitesse de synchronisme comme le champ tournant, la variation de flux sur les conducteurs devient nulle et le couple moteur disparaît.

I.3.Constitution de la MAS

Ce type de machine est basé sur l'enroulement d'une masse métallique par l'action d'un champ tournant et comportant 2 armatures coaxiales l'une est fixe appelée ***stator*** et l'autre est mobile appelée ***rotor***; entre les 2 armatures il y a l'entrefer plus de details dans la figure (I.2)

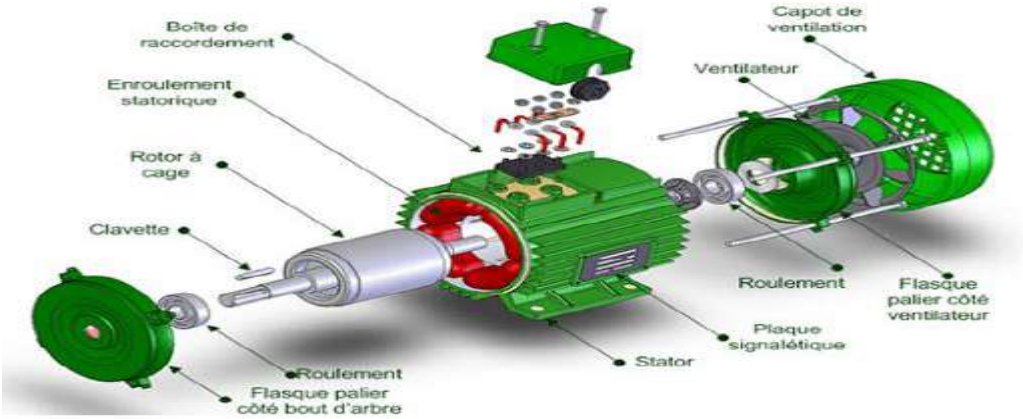


Figure I-2 : Constitution d'une MAS

I.3.1 Stator de la MAS

Le stator est la partie fixe "statique" du **moteur**. **Le stator** est formé d'une carcasse ferromagnétique qui contient trois enroulements électriques. Le passage du courant dans les trois enroulements crée un champ magnétique à l'intérieur du **stator .**figure (I.3)



Figure I-3 : Stator d'une MAS

I.3.2 Rotor de la MAS

Le rotor est la partie mobile du moteur asynchrone. Couplé mécaniquement à un treuil d'ascenseur par exemple, il va créer un couple moteur capable de fournir un travail de montée et de descente de la cabine d'ascenseur. Il se compose essentiellement :

- D'un empilage de disques minces isolés entre eux et clavetés sur l'arbre du rotor afin de canaliser et de faciliter le passage du flux magnétique.
- D'une cage d'écureuil en aluminium coulé dont les barreaux sont de forme trapézoïdale pour les moteurs asynchrones standards et fermés latéralement par deux "flasques" conductrices.

a) Le rotor à cage d'écureuil ou rotor en court-circuit

L'enroulement du rotor à cage d'écureuil est constituées de barres de cuivre nues introduites dans les encoches ; ces barres sont soudées ou rivées à chaque extrémité à deux anneaux qui les court-circuitent. L'ensemble ressemble à une cage d'écureuil d'où le nom de rotor à cage d'écureuil dans la figure (I.4). Dans les machines de petite moyenne puissance, les barres et les anneaux sont formés d'un seul bloc d'aluminium coulé.

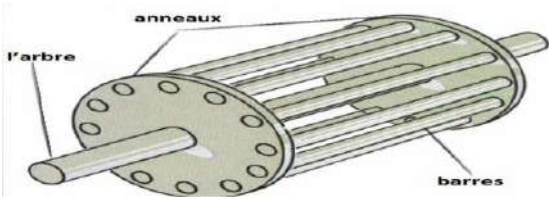


Figure I-4 : Le rotor à cage d'écureuil

b) Le rotor bobiné

Le rotor bobiné comprend un bobinage triphasé, semblable à celui du stator, placé dans les encoches. Il est composé de trois enroulements raccordés en étoile ; l'extrémité libre de chaque enroulement est reliée à une bague tournant avec l'arbre. Ces bagues permettent, par l'intermédiaire de trois balais, d'insérer une résistance extérieure en série avec chacun des trois enroulements lors du démarrage du moteur. En fonctionnement normal, les trois balais sont court-circuités. figure (I.5)



Figure I-5 : Le rotor bobiné

I.4 Modalisation et mise en équation

I.4.1 Hypothèses simplificatrices

L'étude de la modélisation des machines asynchrones repose sur le contexte habituel des hypothèses simplificatrices, à savoir :

· L'entrefer constant ;

· L'effet des encoches négligé ;

· La distribution spatiale sinusoïdale des forces magnétomotrices d'entrefer ;

· Le circuit magnétique non saturé et à perméabilité constante ;

· L'influence de l'effet de peau et de l'échauffement sur les caractéristiques n'est pas prise en compte.

Parmi les conséquences importantes de ces hypothèses, on peut citer:

· La constance des inductances propres;

· La loi de variation sinusoïdale des inductances mutuelles entre les enroulements statoriques et rotoriques en fonction de l'angle électrique de leurs axes magnétiques.

I.4.2 Représentation schématique d'un moteur asynchrone triphasé

A partir de ce qui a été dit précédemment, on peut représenter la machine par deux circuits couplés électro magnétiquement figure (I.6) . Il est cependant plus commode de mettre au point un schéma équivalent vu du stator, analogue à celui d'un transformateur. Les trois enroulements des phases du stator A_s, B_s, C_s, ainsi que les trois enroulements du rotor

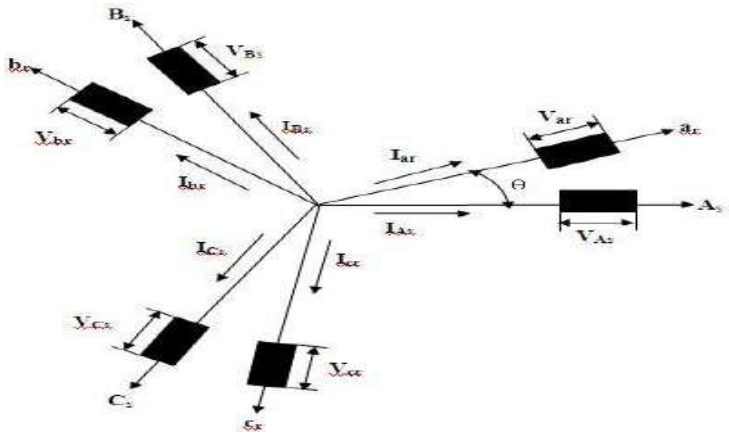


Figure I-6 : Représentation d'une machine asynchrone triphasée

I .5 Modèle mathématique de la MAS

La machine asynchrone représentée par la figure1.7 se compose :

- D'un circuit statorique fixe comportant trois phases identiques décalées entre elles 120° (**as ,bs, cs**: indices des trois phases du stator).

- D'un circuit rotoriques mobile comportant trois phases identiques en court circuits décalées entre elles de 120° (**ar ,br , cr** : indices des trois phases du rotor).

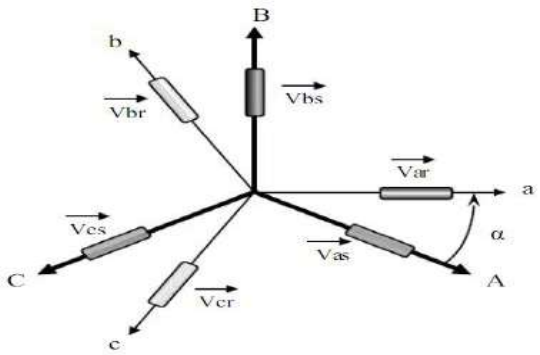


Figure I-7 : Représentation schématique d'une machine asynchrone triphasée

I.5.1 Equation mathématique de la MAS

Dans ces conditions, si on considère que le moteur à induction est triphasé au stator et au rotor. Les trois types d'équations traduisant le comportement du moteur sont[13]:

I.5.1.1 Equation Electrique

Pour le Stator

$$[V_s] = R_s[I_s] + \frac{d}{dt}[\emptyset_s] \quad \text{(I-1)}$$

Pour le rotor

$$[V_r] = R_r[I_r] + \frac{d}{dt}[\emptyset_r] \quad \text{(I-2)}$$

I.5.1.2 Equation magnétique

Pour le stator

$$[\emptyset_s] = [L_{ss}][I_s] + [L_{sr}][I_r] \quad \text{(I-3)}$$

Pour le rotor

$$[\emptyset_s] = [L_{rr}][I_r] + [L_{sr}][I_s] \quad \text{(I-4)}$$

Les valeurs instantanées des tensions et des courants de phase du stator et du rotor sont données respectivement par :

$$[V_s] = [V_{sa}\, V_{sb}\, V_{sc}]^T$$

$$[V_r] = [V_{ra}\, V_{rb}\, V_{rc}]^T \quad \text{(I-5)}$$

$$[I_s] = [I_{sa}\, I_{sb}\, I_{sc}]^T$$

$$[I_r] = [I_{ra}\, I_{rb} I_{rc}]^T \quad \text{(I-6)}$$

La matrice d'inductances statorique donné par :

$$[L_{ss}] = \begin{bmatrix} I_s & M_s & M_s \\ M_s & I_s & M_s \\ M_s & M_s & I_s \end{bmatrix} \quad \text{(I-7)}$$

La matrice d'inductances rotorique donné par :

$$[L_{rr}] = \begin{bmatrix} I_r & M_r & M_r \\ M_r & I_r & M_r \\ M_r & M_r & I_r \end{bmatrix} \quad \text{(I-8)}$$

La matrice des inductances mutuelles stator – rotor $[L_{sr}]$s'écrit sous la forme :

$$[L_{sr}] = M_{sr} \begin{bmatrix} \cos\theta & \cos(\theta + 2\frac{\pi}{3}) & \cos(\theta - 2\frac{\pi}{3}) \\ \cos(\theta - 2\frac{\pi}{3}) & \cos\theta & \cos(\theta + 2\frac{\pi}{3}) \\ \cos(\theta + 2\frac{\pi}{3}) & \cos(\theta - 2\frac{\pi}{3}) & \cos\theta \end{bmatrix} \quad \text{(I-9)}$$

$[L_{sr}]^T = [L_{sr}]$

Avec :

I_s : inductance propre des phases statoriques.

I_r: inductance propre des phases rotoriques.

L_{sr} : inductance mutuelle entre rotor et stator.

M_s: inductance mutuelle entre phases statoriques.

M_r : inductance mutuelle entre phases rotoriques.

θ : écart angulaire entre les axes des phases du stator et du rotor.

M_{sr} : maximum de l'inductance mutuelle entre une phase statorique et une phase rotorique.

I.5.1.3 Equation mécanique :

L'étude des caractéristiques de la machine asynchrone fait introduire du variation non seulement des paramètres électriques (tension, courant, flux) mais aussi des paramètres mécaniques (couple, vitesse) ,L'équation du mouvement de la machine s'écrit :

$$C_e = \frac{J}{p}\frac{d\omega}{dt} + \frac{f\omega}{p} + C_r \qquad \text{(I-10)}$$

Avec :

J : moment d'inertie de la machine [Kgm^2].

f : coefficient de frottement.[$N.m$ / rad / s] .

Cr : couple résistant impose àl'arbre de la machine [$N.m$] .

Ce : couple électromagnétique développe par la machine [$N.m$].

ω : vitesse rotorique électrique [rad / s] .

p : nombre de pairs de pôles.

Ω : vitesse mécanique donnée par :$\Omega = \frac{\omega}{p}$

On obtient finalement le modèle asynchrone triphasé :

$$\begin{cases} [V_s] = [R_s][I_s] + \frac{d}{dt}\{[L_{ss}][I_s]\} + \frac{d}{dt}\{[M_{sr}][I_r]\} \\ [V_s] = [R_s][I_s] + \frac{d}{dt}\{[L_{ss}][I_s]\} + \frac{d}{dt}\{[M_{sr}][I_r]\} \end{cases} \qquad \text{(I-11)}$$

I.5.2 Transformation de park appliquée à la MAS:

Le modèle de Park est la substitution du modèle triphasé (Equation différentielles à coefficients variables).par le modèle biphasé caractérise par deux axes orthogonaux U et V (U axe direct et V axe en quadrature) qui a pour effet de rendre les inductances mutuelles et propres du modèle indépendantes de la rotation. La figure (1.8) présente le système d'un axe réel triphasé (Xa, Xb, Xc) et le système d'axe fictif (U, V)

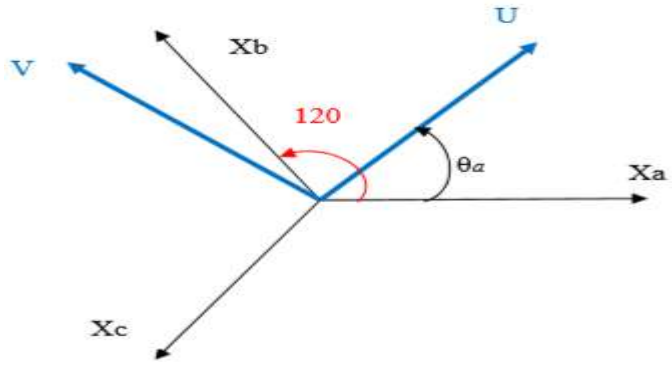


Figure I-8 : Transformation Des enroulements réels en enroulements équivalents (u,v)

$$\begin{bmatrix} X_U \\ X_V \\ X_0 \end{bmatrix} = [P(\theta_\alpha)] \begin{bmatrix} X_a \\ X_b \\ X_c \end{bmatrix} \quad \text{(I-12)}$$

(XU, XV) système biphasé équivalent se positionnant a θ_α près

X : représente une tension, un courant ou un flux

X_0 : Composante homopolaire (pour les systèmes triphasés équilibrés, la composante homopolaire est nulle).

$$P(\theta_\alpha) = \sqrt{(2/3)} \begin{bmatrix} \cos(\theta_\alpha) & \cos(\theta_\alpha - 2\frac{\Pi}{3}) & \cos(\theta_\alpha + 2\frac{\Pi}{3}) \\ -\sin(\theta_\alpha) & -\sin(\theta_\alpha - 2\frac{\Pi}{3}) & -\sin(\theta_\alpha + 2\frac{\Pi}{3}) \\ 1/\sqrt{2} & 1/\sqrt{2} & 1/\sqrt{2} \end{bmatrix} \quad \text{(I-13)}$$

D'où se matrice inverse $P(\theta_\alpha)^{-1}$

$$P(\theta_\alpha)^{-1} = \sqrt{(2/3)} \begin{bmatrix} \cos(\theta_\alpha) & -\sin(\theta_\alpha) & 1/\sqrt{2} \\ \cos(\theta_\alpha - 2\frac{\Pi}{3}) & -\sin(\theta_\alpha - 2\frac{\Pi}{3}) & 1/\sqrt{2} \\ \cos(\theta_\alpha + 2\frac{\Pi}{3}) & -\sin(\theta_\alpha + 2\frac{\Pi}{3}) & 1/\sqrt{2} \end{bmatrix} \quad \text{(I-14)}$$

$P(\theta_\alpha)$: Matrice de transformation de Park.

L'angle θ_α dans la matrice[$P(\theta_\alpha)$] prend la valeur θ_s pour les grandeurs statorique et la valeur θ_r pour les grandeurs rotorique

- θ_α angle de positionnement de l'axe (U, V) par rapport a l'axe réel (Xa, Xb, Xc)
- $\omega_\alpha = \frac{d\theta_\alpha}{dt}$ vitesse de coordonnée de l'ave (u,v)

I.5.2.1.Equation électrique :

La transformation de Park consiste à appliquer aux grandeurs courants, tensions et flux un changement de variables faisant intervenir l'angle entre l'axe des enroulements et les axes U et V. Les expressions des tensions statoriques et rotoriques dans ce système après transformation s'écrivent comme suit:

$$\begin{bmatrix} V_{US} \\ V_{VS} \end{bmatrix} = \begin{bmatrix} R_s & 0 \\ 0 & R_S \end{bmatrix} \begin{bmatrix} i_{us} \\ i_{vs} \end{bmatrix} + \frac{d}{dt} \begin{bmatrix} \phi_{us} \\ \phi_{vs} \end{bmatrix} + \begin{bmatrix} 0 & -\omega_\alpha \\ \omega_\alpha & 0 \end{bmatrix} \begin{bmatrix} \phi_{us} \\ \phi_{vs} \end{bmatrix} \quad \text{(I-15)}$$

$$\begin{bmatrix} 0 \\ 0 \end{bmatrix} = \begin{bmatrix} R_r & 0 \\ 0 & R_r \end{bmatrix} \begin{bmatrix} i_{us} \\ i_{vs} \end{bmatrix} + \frac{d}{dt} \begin{bmatrix} \phi_{ur} \\ \phi_{vr} \end{bmatrix} + \begin{bmatrix} 0 & -(\omega_\alpha - \omega) \\ \omega_\alpha - \omega & 0 \end{bmatrix} \begin{bmatrix} \phi_{ur} \\ \phi_{vr} \end{bmatrix} \quad \text{(I-16)}$$

1.5.2.2. Equation du flux :

$$\begin{cases} \phi_{us} = L_s i_{us} + M i_{ur} \\ \phi_{vs} = L_s i_{vs} + M i_{vr} \end{cases} \quad \text{(I-17)}$$

Avec :

$L_s = l_s - M_s$ Inductance cyclique propre du stator

$M = \frac{3}{2} M_{sr}$ Inductance cyclique mutuelle propre de l'effet des trois phases statoriques sur une phase du stator.

De la même manière pour le rotor ,on obtient :

$$\begin{cases} \phi_{ur} = L_r i_{us} + M i_{us} \\ \phi_{vr} = L_r i_{vr} + M i_{vs} \end{cases} \quad \text{(I-18)}$$

Avec $L_r = l_r - M_r$ Inductance cyclique propre du rotor

D'après les équations précédentes , on conclure l'équation matricielle suivante :

$$[V]=[I][R]+[L][I]+ \omega_s [A1][I]+(\omega_s- \omega)[A2][I] \quad \text{(I-19)}$$

I.5.2.3. Choix du repère :

Il existe différentes possibilités pour le choix de l'orientation du repère d'axe (u,v) qui dépendent généralement des objectifs de l'application. On peut fixer le repère (u, v) au stator ou au rotor ou au champ tournant, rappelons que le repère (u, v) est le repère mobile, c'est-à-dire qu'il nous appartient de calculer les angles de transformation de Park θs et θr afin d'effectuer les rotations, on peut donc le lier à un référentiel mobile comme le champ tournant

- Dans un référentiel lié à un champ tournant, l'avantage d'utiliser ce référentiel, est d'en faire aisément la régulation où on a besoin d'une orientation du flux.

- Dans un référentiel lié au stator pour l'étude des grandeurs rotoriques.

- Dans un référentiel lié au rotor pour l'étude des grandeurs statoriques.

I.5.2.4. Référentiel immobile au champ tournant (Système d, q) :

Les axes de référence sont tournantes au synchronisme d'où :

$$\theta_\alpha = \theta_s \Rightarrow \omega_\alpha = \omega_s \quad \text{(I-20)}$$

$$\omega_r = \omega_\alpha - \omega = \omega_s - \omega$$

$$\begin{cases} \dfrac{d\theta}{dt} = \omega \\ \dfrac{d\theta_s}{dt} = \omega_s \\ \dfrac{d\theta_r}{dt} = \omega_r \end{cases} \quad \text{(I-21)}$$

Ce qui permet d'avoir :

$$\begin{bmatrix} V_{ds} \\ V_{qs} \end{bmatrix} = \begin{bmatrix} R_s & 0 \\ 0 & R_S \end{bmatrix} \begin{bmatrix} i_{ds} \\ i_{qs} \end{bmatrix} + \frac{d}{dt} \begin{bmatrix} \phi_{ds} \\ \phi_{qs} \end{bmatrix} + \begin{bmatrix} 0 & -\omega_s \\ \omega_s & 0 \end{bmatrix} \begin{bmatrix} \phi_{ds} \\ \phi_{qs} \end{bmatrix} \quad \text{(I-22)}$$

$$\begin{bmatrix} V_{dr} \\ V_{qr} \end{bmatrix} = \begin{bmatrix} R_r & 0 \\ 0 & R_r \end{bmatrix} \begin{bmatrix} i_{dr} \\ i_{qr} \end{bmatrix} + \frac{d}{dt} \begin{bmatrix} \phi_{ds} \\ \phi_{qr} \end{bmatrix} + \begin{bmatrix} 0 & -\omega_r \\ \omega_r & 0 \end{bmatrix} \begin{bmatrix} \phi_{dr} \\ \phi_{qr} \end{bmatrix} \quad \text{(I-23)}$$

$$\begin{cases} V_{ds} = R_s i_{ds} + \dfrac{d\phi_{ds}}{dt} - \omega_s \phi_{qs} \\ V_{qs} = R_s i_{qs} + \dfrac{d\phi_{qs}}{dt} - \omega_s \phi_{ds} \\ V_{dr} = R_r i_{dr} + \dfrac{d\phi_{dr}}{dt} - \omega_r \phi_{qr} = 0 \\ V_{qr} = R_r i_{qr} + \dfrac{d\phi_{qr}}{dt} - \omega_r \phi_{dr} = 0 \end{cases} \quad \text{(I-24)}$$

$$\begin{cases} \phi_{ds} = L_s i_{ds} + M i_{dr} \\ \phi_{qs} = L_s i_{qs} + M i_{qr} \end{cases} \quad \text{(I-25)}$$

$$\begin{cases} \phi_{dr} = L_r i_{dr} + M i_{ds} \\ \phi_{qr} = L_r i_{qr} + M i_{qs} \end{cases} \quad \text{(I-26)}$$

L'avantage d'utiliser ce référentiel, est d'avoir des grandeurs constantes en régime permanent il est alors plus aisé d'en faire la régulation.

I.5.2.5 Equation du couple electromagnétique :

$$C_e = \frac{PM}{L_r}\left(\phi_{dr} i_{qs} - \phi_{qr} i_{ds}\right) \quad \text{(I-27)}$$

I.5.2.6 Equation d'état du modèle de la machine asynchrone:

Une machine asynchrone alimentée en tension, a comme variables de commande, les tensions statoriques V_{ds} et V_{qs} et comme perturbation le couple résistant Cr. Elle peut être décrite par Plusieurs variables d'état. Notre étude sera limitée à un seul cas représenté par les courants statoriques et les flux rotoriques

En remplaçant les expressions (I.20) et (I.21) dans l'équation (I.19), on obtient le système suivant :

$$\begin{cases} \dot{i}_{ds} = \left(\dfrac{-1}{\delta T_s} - [1-\delta]\dfrac{1}{\delta T_r} \right) i_{ds} + \omega_s i_{qs} + \dfrac{(1-\delta)}{\delta M T_r}\phi_{dr} + \dfrac{(1-\delta)}{\delta M}\omega_r \phi_{qr} + \dfrac{1}{\delta L_s} V_{ds} \\ \dot{i}_{qs} = -\omega_s i_{ds} \left(\dfrac{-1}{\delta T_s} - [1-\delta]\dfrac{1}{\delta T_r} \right) i_{qs} + \dfrac{(1-\delta)}{\delta M}\omega_r \phi_{dr} + \dfrac{(1-\delta)}{\delta M T_r}\phi_{qr} + \dfrac{1}{\delta L_s} V_{qs} \\ \dot{\phi}_{dr} = \dfrac{M}{T_r} i_{ds} - \dfrac{1}{T_r}\phi_{dr} + (\omega_s - \omega_r)\phi_{qr} \\ \dot{\phi}_{qr} = \dfrac{M}{T_r} i_{qs} - \dfrac{1}{T_r}\phi_{qr} - (\omega_s - \omega_r)\phi_{qr} \end{cases} \quad \text{(I-28)}$$

On cherche à obtenir un système d'équation écrit sous forme d'équation d'état Le modèle sera de la forme

$$\dot{X} = AX + BU \quad \text{(I-29)}$$

Avec :

$X = \left[i_{ds} i_{qs} \phi_{dr} \phi_{qr} \right]^T$ Vecteur d'état

$\dot{X} = \left[\dot{i}_{ds} \dot{i}_{qs} \dot{\phi}_{dr} \dot{\phi}_{qr} \right]^T$

$U = \left[V_{ds} V_{qs} \right]$ Vecteur de commande

Après tout calcul fait on trouve :

$$A=\begin{bmatrix} -\left(\frac{1}{T_S\delta}+\frac{1-\delta}{T_r\delta}\right) & \omega_S & \frac{1-\delta}{MT_r\delta} & \frac{1-\delta}{M\delta}\omega \\ -\omega_s & -\left(\frac{1}{T_S\delta}+\frac{1-\delta}{T_r}\right) & -\left(\frac{1-\delta}{M\delta}\right)\omega & \frac{1-\delta}{MT_r\delta} \\ \frac{M}{T_r} & 0 & -\frac{1}{T_r} & (\omega_s-\omega) \\ 0 & \frac{M}{T_r} & -(\omega_s-\omega) & \frac{-1}{T_r} \end{bmatrix} \quad \text{(I-30)}$$

la matrice de commande :

$$B=\begin{bmatrix} \frac{1}{\delta L_s} & 0 \\ 0 & \frac{1}{\delta L_s} \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \quad \text{(I-31)}$$

Avec :

$\omega_r=\omega_s-\omega$

$\delta=1-\frac{M^2}{L_rL_s}$ Coefficient de dispersion

$T_S=\frac{L_s}{R_s}$ Constante de temps statorique

$T_r=\frac{L_r}{R_r}$ Constante de temps rotorique

I.6 Modélisation de l'onduleur de tension

Un onduleur de tension triphasé est constitué de trois cellules (bras) de commutation de deux interrupteurs. Pour chaque interrupteur est formé par un transistor (ou un thyristor) monté en tête-bêche avec une diode de récupération. Pour assurer la continuité des courants alternatifs et éviter le court- circuitage de la source, les interrupteurs k11et k12, k21 et k22, k31 et k32 doivent être contrôlé de manière complémentaire. Le schéma structurel de l'onduleur de tension alimentant le stator du moteur asynchrone est donné par la figure (1-9)

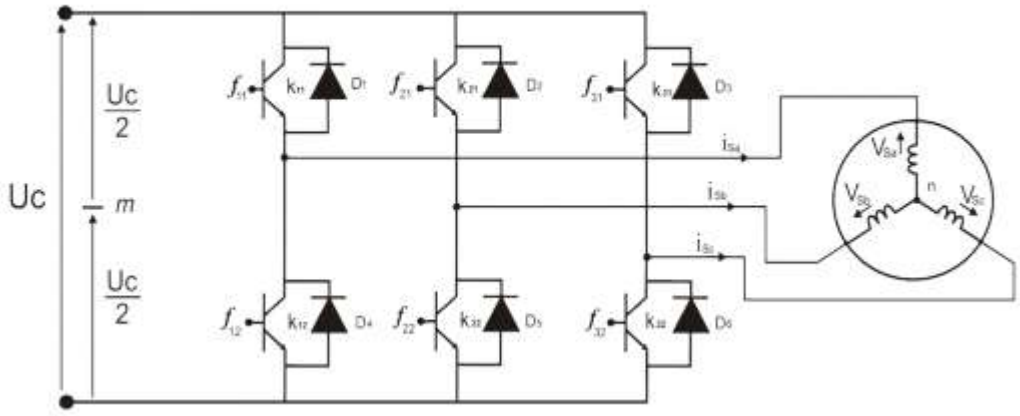


Figure I-9 :Schéma d'un onduleur de tension triphasé alimentant le stator du MAS

On définit la fonction de connexion fji (j ϵ {1, 2, 3}, i ϵ {1, 2}) comme l'état de l'interrupteur kji, on a :

$$\begin{cases} f_{11} = 1 - f_{10} \\ f_{21} = 1 - f_{20} \\ f_{31} = 1 - f_{30} \end{cases} \quad \text{(I-32)}$$

Avec :

fji =1 si l'interrupteur est fermé.

fji =0 si l'interrupteur est ouvert.

L'onduleur est alimenté par une source de tension continue constante, d'amplitude «Uc».Les potentiels des nœuds « a », « b » et« c » de l'onduleur triphasé par rapport au point milieu fictif «n » sont donnés par les tensions suivantes :

$$\begin{cases} V_{an} = f_{11} U_c \\ V_{bn} = f_{21} U_c \\ V_{cn} = f_{31} U_c \end{cases} \quad \text{(I-33)}$$

Les tensions composés délivrées par cet onduleur sont donnés par :

$$\begin{cases} u_{sab} = U_c(f_{11} - f_{21}) \\ u_{sbc} = U_c(f_{21} - f_{31}) \\ u_{sca} = U_c(f_{31} - f_{11}) \end{cases} \quad \text{(I-34)}$$

Pour une charge triphasé équilibrée, couplé en étoile avec un neutre isolé, les tensions statoriques simples sont reliés par :

$$V_{sa} + V_{sb} + V_{sc} = 0 \quad \text{(I-35)}$$

Les tensions simples sont liées aux tensions composées par :

$$\begin{cases} v_{sa} = 1/3(u_{sab} - u_{sca}) \\ v_{sb} = 1/3(u_{sbc} - u_{sab}) \\ v_{sc} = 1/3(u_{sca} - u_{sbc}) \end{cases} \quad \text{(I-36)}$$

Après arrangement les équations des deux systèmes (I. 32) et (I.34), on obtient le système matricielle suivant :

$$\begin{bmatrix} v_{sa} \\ v_{sb} \\ v_{sc} \end{bmatrix} = U_C/3 \begin{bmatrix} 2 & -1 & -1 \\ -1 & 2 & -1 \\ -1 & -1 & 2 \end{bmatrix} \begin{bmatrix} f_{11} \\ f_{21} \\ f_{31} \end{bmatrix} \quad \text{(I-37)}$$

I.6.1 La commande par modulation de largeur d'impulsion MLI

La Modulation en Largeur d'Impulsions (MLI) : en anglais : (Pulse Width Modulation, soit PWM) est une technique couramment utilisée pour synthétiser des signaux pseudo analogiques. De plus elle est utilisée pour le pilotage des convertisseurs statiques servant d'interface entre une charge (machine électrique, ...) et son dispositif d'alimentation (onduleur triphasé, ...). Elle est donc une technique utilisée pour la conversion de l'énergie.

I.6.2 Principe de la MLI sinus–triangle

Nous nous intéresserons à une technique analogique « la commande sinus triangle ». Cette technique consiste à comparer entre elles :

- Une onde de référence U_m, la modulatrice, de forme sinusoïdale et de fréquence f
- Une onde de modulation U_p, la porteuse, généralement triangulaire, de haute fréquence fp, telle que : $f_p \gg f$.

Le schéma de principe de cette commande est donné sur la figure (I.10). Les instants de fermeture des interrupteurs sont alors définis par les intersections entre les deux ondes, quant à la fréquence de commutation, elle est déterminée par celle de la porteuse.

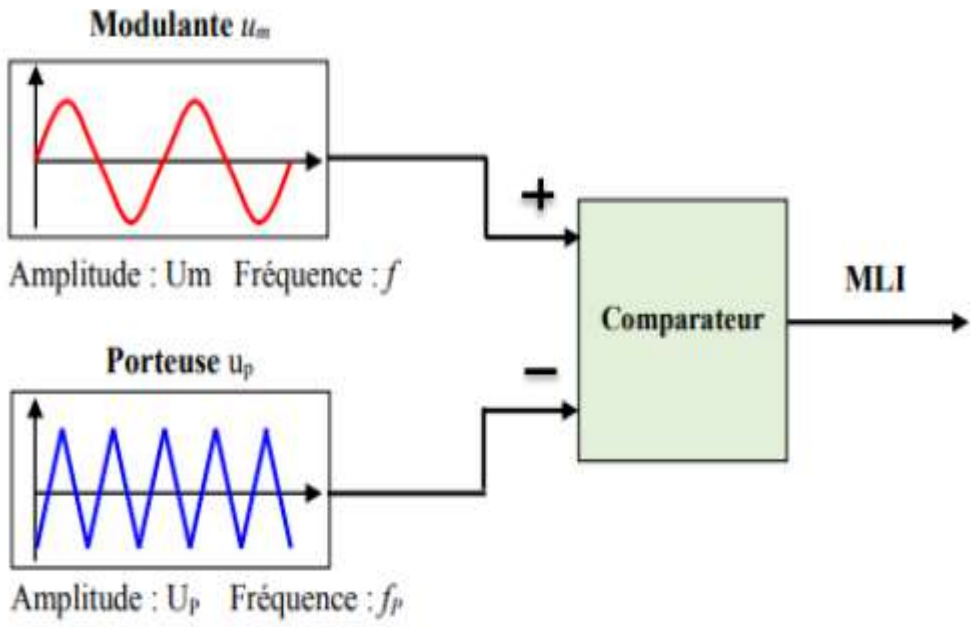


Figure I.10 : Schéma de principe de la commande «sinus -triangle ».

La figure (I.11) présente un exemple de chronogramme pour ce type de comparaison intersecté, généralement afin de réaliser une commande MLI triphasée on génère un système sinusoïdal de tension de référence V_{Ar}, V_{Br}, V_{Cr}

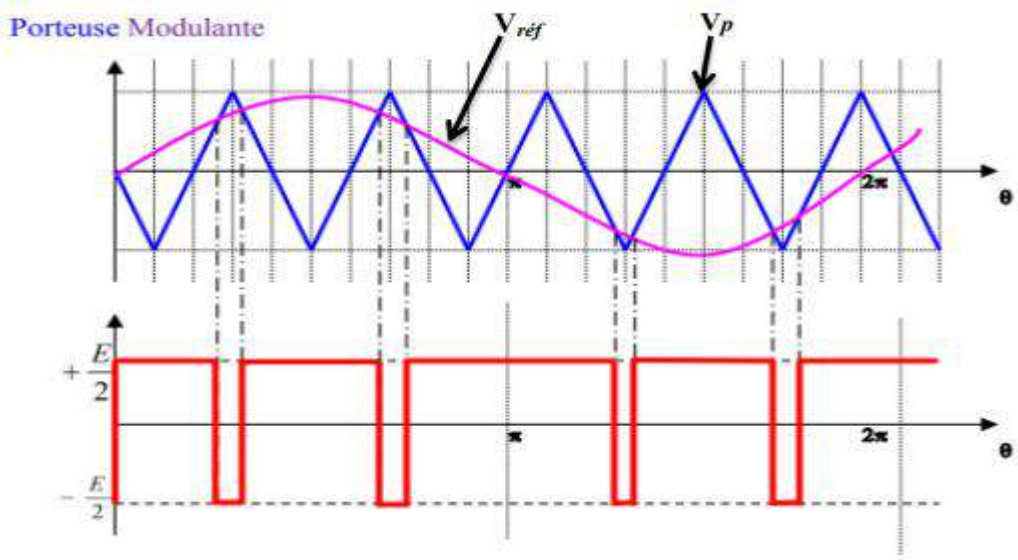


Figure I.11 Chronogramme d'une commande MLI «sinus -triangle» monophasé.

La figure (I.12) représente le schéma de principe d'une commande MLI sinus-triangle triphasée

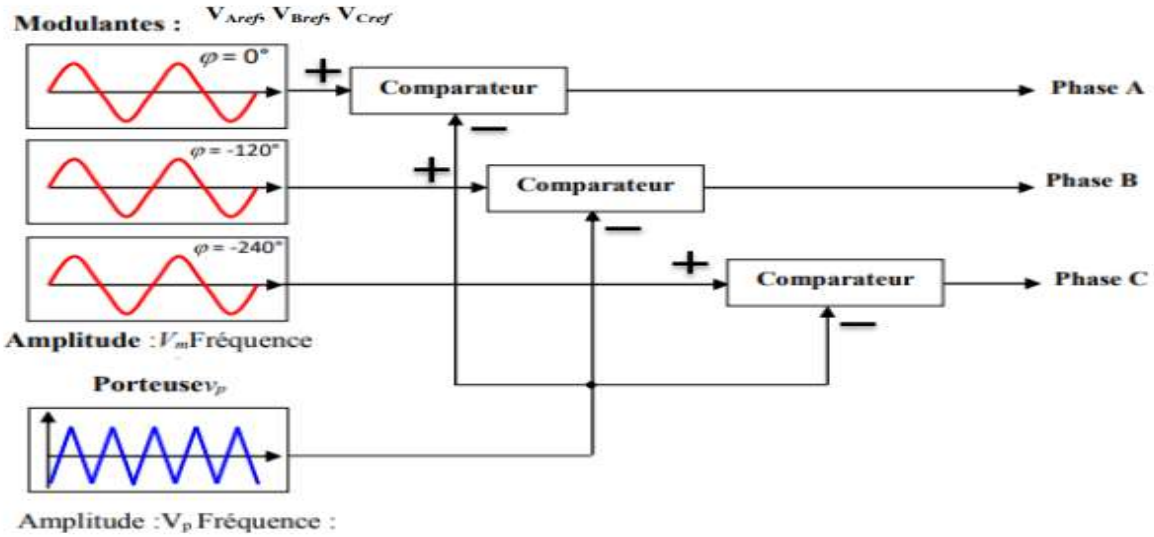


Figure I.12 : Schéma de principe d'une commande « sinus -triangle » triphasée**.**

I.7 Simulation de la MAS:

La structure en schéma bloc de cette simulation est présentée par la figure suivante :

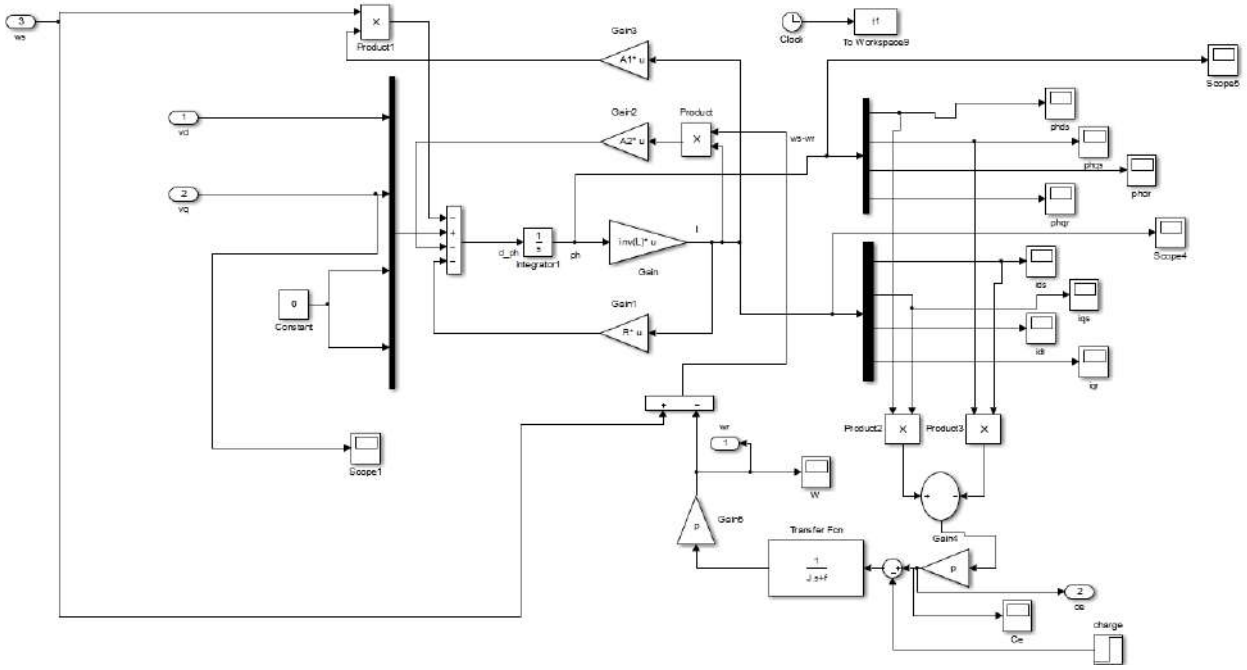
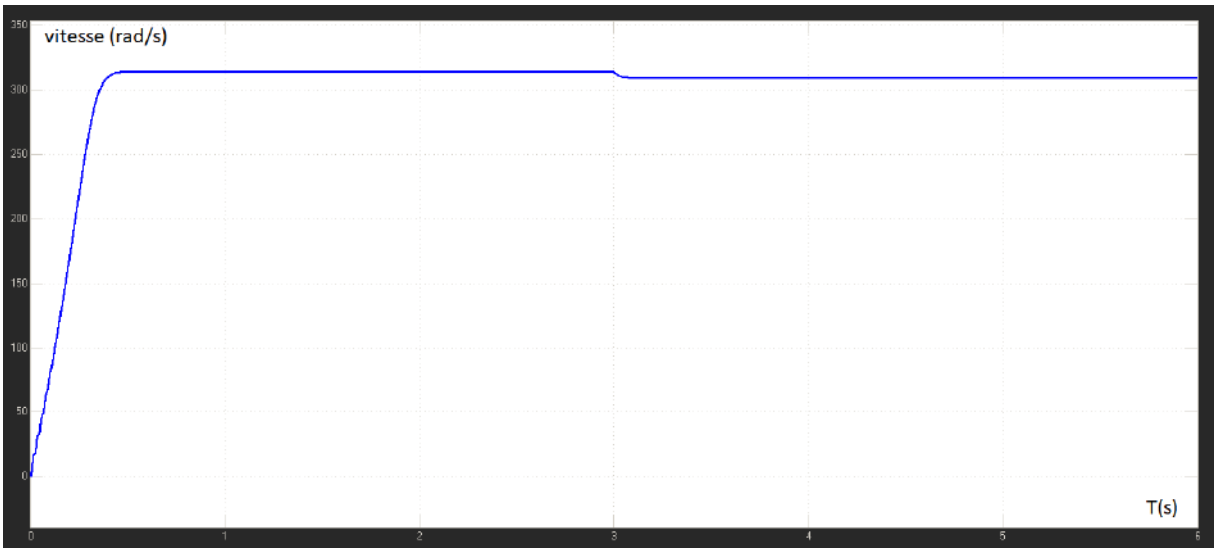


Figure I-13 : Modèle de la machine asynchrone alimentée en tension

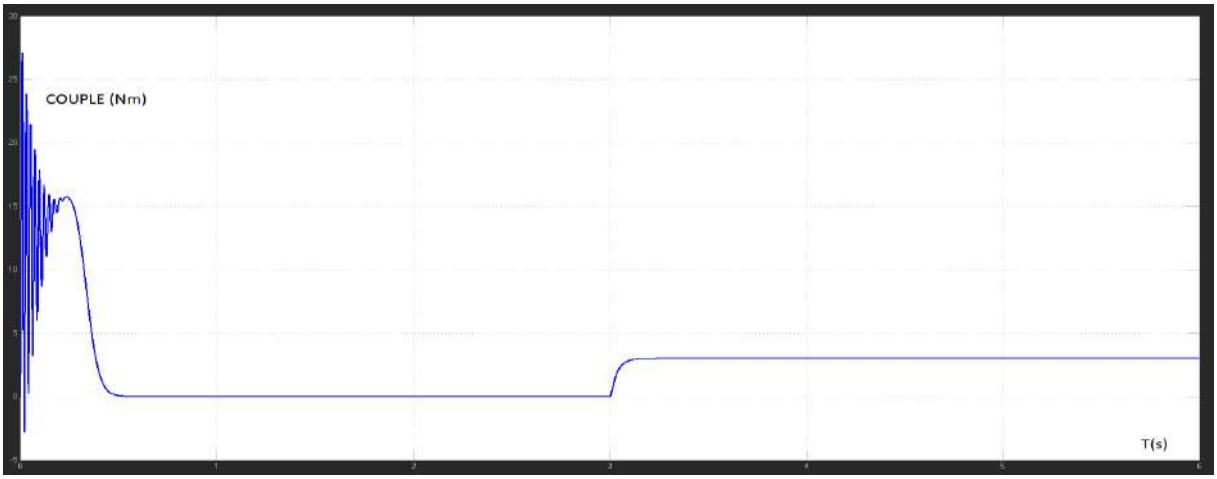
I.7.1 Résultat de simulation:

Dans un objectif de vérifier le modèle que nous avons développé dans cette partie, une simulation est réalisé avec le logiciel Matlab/Simulink.

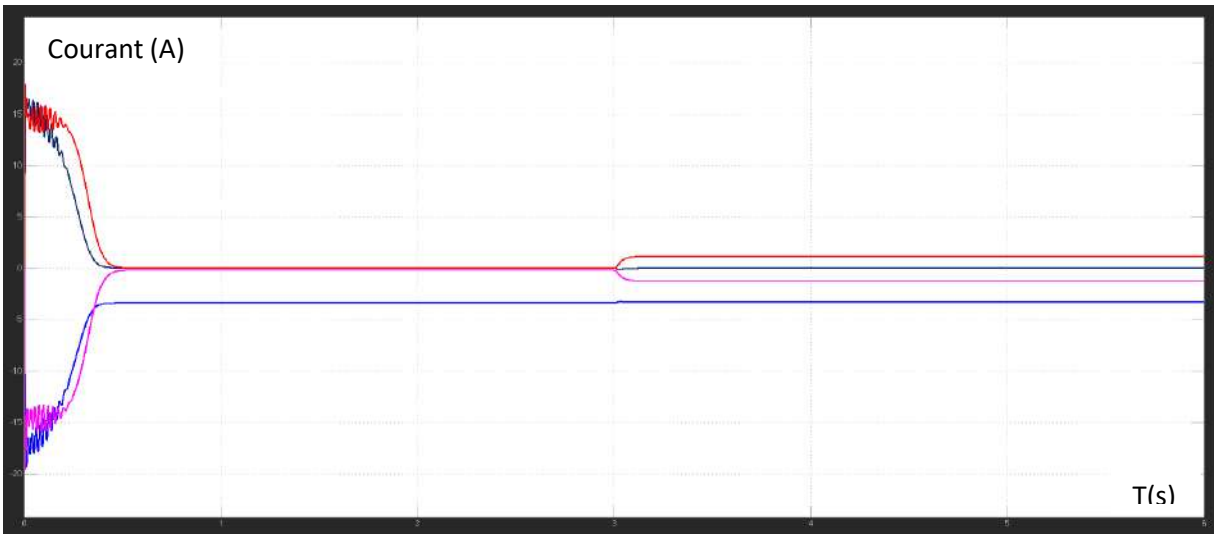
Après un démarrage à vide, on a appliqué, sur notre machine, un couple de charge Cr = 3 Nm à l'instant t =3s .



(a) Vitesse du rotation avec une charge en t=3s



(b) Couple électromagnétique



(c) Courant (i)

Figure I-14 : Résultat du simulation du mas avec un charge t=3s

I.7.2 Interprétation des résultats

A partir des résultats ci-dessus, nous pouvons constater que les performances sont satisfaisantes. Les figures (I.14.a) et (I.14.b) La vitesse et le flux convergent rapidement vers leurs références

Après t=3s le couple se stabilise à une valeur de 3 N.m qui correspond au couple de charge et une diminution permanente de la vitesse apparaît, ceci est dû au fait qu'il n'y pas de régulation.

I.8 Conclusion

Dans ce chapitre nous avons fait la modélisation de la machine asynchrone triphasé, Après, nous avons utilisé la transformation de Park pour éviter la complexité des équations différentielles. Cette transformation permet le changement du système triphasé réel au système biphasé linéaire équivalent de cette machine ce qui signifie une facilité de résolution et de simulation

Pour finir, on a fait l'association onduleur-machine

Dans le chapitre suivant, il sera question de s'intéresser à la commande de la MAS par la commande scalaire par régulateur PI-FLOU

Chapitre II

Le régulateur PI-Flou pour la commande du vitesse d'une machine Asynchrone

II.1 Introduction

Le moteur asynchrone est le moteur le plus utilisé dans les applications de la vitesse variable. Sa simplicité de conception, sa fiabilité, l'absence de collecteur sont les principales raisons de son succès. Mais son modèle mathématique est complexe, multivariable, non-linéaire, avec un fort couplage entre les variables d'entrée, de sortie et internes, ce qui a longtemps retardé l'apparition de commandes performantes et appropriés à ce type de motorisation [7], [8].

Bien que beaucoup d'algorithmes de commande aient été développés pendant plusieurs décennies, les contrôleurs PI sont toujours la majorité des régulateurs utilisés dans les systèmes de commande industrielles particulièrement quand les performances élevées du régulateur ne sont pas exigées.

La conception et l'analyse d'un tel contrôleur exigent de connaître les deux paramètres à savoir : le gain proportionnel (Kp) et l'intégrateur (Ki). Pour comprendre comment le comportement du système affecté par les deux paramètres des contrôleurs PI, beaucoup de temps doit être consommé pour trouver la règle optimale pour varier ces paramètres. La plupart de ces règles sont acceptées par les systèmes de commande industriels en général, mais elles réalisent rarement une bonne performance parce que le système à contrôler est difficile d'être modélisé exactement, [8], [9], [10], [11].

Les contrôleurs basés sur les concepts de l'intelligence artificielle, logique floue, réseaux de neurones, sont des alternatives intéressantes, et permettent d'améliorer la robustesse du processus à contrôler, [12].

Dans ce travail, on a amélioré les inconvénients du régulation classique du contrôleur PI en adoptant de commande floue en incluant l'expertise acquise par les opérateurs, [13]. Cette méthode permet d'obtenir une loi de commande souvent efficace, sans devoir faire appel à des développements théoriques importants. Elle présente l'intérêt de prendre en compte les expériences acquises par les utilisateurs et opérateurs du processus à commander.

La méthode suivante est utilisée pour déterminer les paramètres du régulateur PI adaptatif-flou,

-minimiser l'erreur statique.

-sélectionner les gains e et Δe jusqu'à l'obtention de la meilleure réponse du système

II.2. Généralité sur la commande scalaire

La commande scalaire est, la plus ancienne et la plus simple des lois de commande d'une machine asynchrone. Elle est utilisée essentiellement pour des applications ne nécessitant que des performances statiques ou dynamiques moyennes. De nos jours, grâce aux avancées de l'électronique de puissance, de nombreux variateurs utilisent ce mode de commande. On en retrouve essentiellement pour des applications industrielles de pompage, climatisation, ventilation.

Le principe de cette méthode est assez rustique. Il consiste à imposer aux bornes de l'induit de la machine, le module de la tension ou du courant ainsi que la pulsation. Plusieurs commandes scalaires existent selon que l'on agit sur le courant ou sur la tension. Elles dépendent surtout de la topologie de l'actionneur utilisé (Onduleur de tension ou de courant). Dans l'alimentation en tension, les onduleurs fournissent des tensions dont la forme et l'amplitude peuvent être considérées indépendantes de la charge. Par contre dans l'alimentation en courant, les courants fournis ont des formes et des amplitudes influencées par la nature de la charge. La variation de la vitesse est obtenue par la variation de la pulsation statorique qui est générée directement par le régulateur. Cette méthode de commande est basée sur le modèle de la machine en régime permanent[16].

II.3.La commande scalaire

La commande scalaire est relativement simple comparée à la commande vectorielle. Elle est basée sur le modèle de la machine asynchrone en régime permanent sinusoïdal. Il existe plusieurs types de commandes scalaires selon qu'on agit sur le courant ou la tension. Elles dépendent surtout de la topologie de l'actionneur utilisé (onduleur de tension ou de courant). L'onduleur de tension est le plus utilisé en petite et moyenne puissance

L'alimentation à fréquence variable se fait à l'aide d'un convertisseur statique généralement Continu–alternatif. La source d'entrée peut être de type source de courant ou de type source de tension. En sortie du convertisseur on contrôle [15] :

- Soit une commande directe avec réglage de la fréquence statorique et de la tension statorique ou du courant statorique . son principe represente dans la figure (II.1)

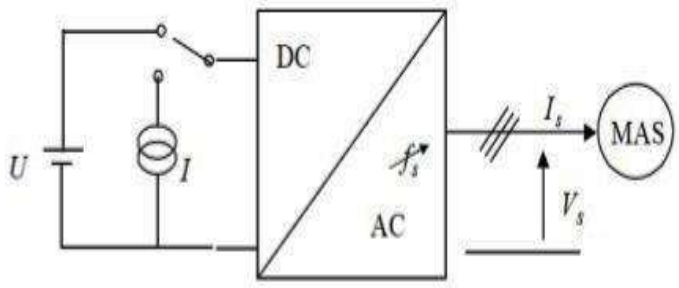


Figure II.1 : principe de réglage

- Soit une commande avec autopilotage figure (II.2) qui asservit la fréquence statorique et la tension ou le courant en réalisant = + à l'aide d'un capteur mécanique de vitesse

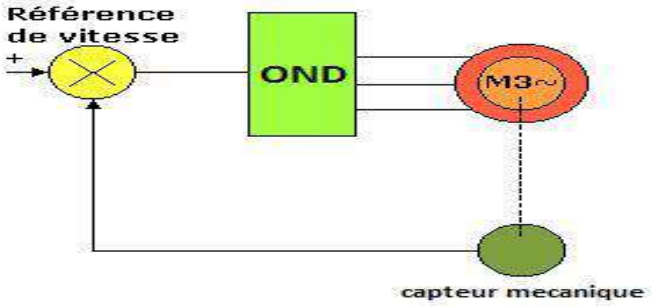


Figure II.2: réglage avec autopilotage

On définit le glissement comme le rapport des pulsations rotoriques et statoriques

$$g = \frac{\omega_r}{\omega_s} \qquad \text{(II-1)}$$

Les différentes pulsations sont reliées par la relation d'autopilotage suivante :

$$\omega_s = \omega_r + p\Omega \qquad \text{(II-2)}$$

II.4. Modélisation de la machine asynchrone en régime permanent

Pour mettre en évidence les principes généraux de réglage du couple électromagnétique de la machine asynchrone, nous allons adopter le modèle du régime permanent sinusoïdal en utilisant la notation de vecteur complexe. Le contrôle de vitesse le plus simple, dit 'U sur F', permet de varier la vitesse de la machine sur une large plage. Les équations qui permettent de calculer le couple et de prévoir les points de fonctionnement sont basés sur le modèle 'régime permanent' de la machine. Pour la mise en équations du modèle de la machine, nous supposons les hypothèses du chapitre1

II.5. Comportement en régime permanant de la MAS :

En étudions le comportement de la machine en régime permanent, pour alléger les notations, posons :

$$\bar{\mathrm{X}} = \overline{\mathrm{X}}_d + j\overline{\mathrm{X}}_q \tag{II-3}$$

En régime permanant et dans un repère lié au rotor, l'équation électrique rotorique s'écrit :

$$0 = R_r\overline{I}_r + j\omega_g L_r\overline{I}_r + j\omega_g M\overline{I}_s \tag{II-4}$$

Où ω_g la pulsation des courant rotoriques. Le flux statorique est donné par ses composantes dans les axes d et q du repère lié au rotor :

$$\overline{\psi}_s = L_s\overline{I}_s + M\overline{I}_r \tag{II-5}$$

A partir des é1quations ci–dessus, on en déduit :

$$\overline{I}_r = -\frac{jM\omega_g}{R_r + j\omega_{sl}L_r}\overline{I}_s \tag{II-6}$$

$$\overline{\phi}_s = L_s\frac{R_r + jL_r\sigma\omega_g}{R_r + jL_r\omega_g}\overline{I}_s \tag{II-7}$$

D'où l'expression du courant statorique en fonction du module du flux statorique :

$$I_s = \frac{\psi_s}{Ls}\sqrt{\frac{1+\left(\frac{\omega_g L_R}{R_r}\right)^2}{1+\left(\frac{\omega_g \sigma L_r}{R_r}\right)^2}} \qquad \text{(II-8)}$$

Cette relation est la base des lois de commande à flux constant des machine alimentée en courant. Rappelons qu'en régime sinusoïdal équilibré, la norme d'une grandeur triphasée X soit :

$$X = \hat{X}\begin{pmatrix} \cos\omega t \\ \cos(\omega t - \frac{2\pi}{3}) \\ \cos(\omega t + \frac{2\pi}{3}) \end{pmatrix} \qquad \text{(II-9)}$$

Que l'on peut encore représenter dans un repère (d,q) par :

$$\sqrt{X^2{}_d + X^2{}_q} = \sqrt{\frac{3}{2}}\hat{X} \qquad \text{(II-10)}$$

Le couple électromagnétique est donné par :

$$C_{em} = 3p\left(\frac{M}{L_s}\right)^2 \psi_{seff}{}^2 \frac{\omega_g}{R_r\left(1+\left(\sigma\frac{L_r}{R_r}\omega_g\right)^2\right)} \qquad \text{(II-11)}$$

Lorsque le module du flux est constant, le couple ne dépend que de .La valeur du couple est fixée par et par le module du flux. En fonctionnant au flux nominal, pour un couple donné, on peut déterminer le glissement donnant le couple maximal pour lequel la réactance de fuite et la résistance rotorique sont égales :

$$C_{em\max} = 3p\left(\frac{M}{L_s}\right)^2 \psi_{seff}{}^2 \frac{1}{2\sigma L_r} \qquad \text{(II-12)}$$

Pour :

$$\omega_{g\max} = \frac{R_r}{\sigma L_r} \tag{II-13}$$

Si le glissement est faible, on peut écrite :

$$C_{em} = \alpha\left(\psi_s\right)^2 \omega_g \tag{II-14}$$

ω_g Permet donc de régler le couple.

En régime permanent et dans un repère lé au stator :

$$\bar{V}_s = R_s\bar{I}_s + j\omega_s L_s\bar{I}_s + j\omega_s L_m\bar{I}_r \tag{II-15}$$

Et

$$\bar{V}_s = \frac{R_s}{1+ j\dfrac{L_r\omega_g}{R_r}}\left[\left(1-\sigma\frac{L_s L_r}{R_s R_r}\omega_g\omega_s\right)+ j\left(\frac{L_r}{R_r}\omega_g + \frac{L_s}{R_s}\omega_s\right)\right]\bar{I}_s \tag{II-16}$$

$$\bar{V}_s = \psi^*_s\frac{R_s}{L_s}\sqrt{\frac{\left(1-\sigma T_s T_r\omega_g\omega_s\right)^2+\left(T_r\omega_g + T_s\omega_s\right)^2}{1+\left(\sigma\omega_g T_r\right)^2}} \tag{II-17}$$

En reportant II.6 dans II.15, la tension statorique doit suivre la loi suivante :

Cette relation est la base des lois de commande permettant le contrôle du module du flux, donc deux modes de commande sont possibles :

- Commande par contrôle de la fréquence statorique Ws et du courant ou de la tension statorique ;

- Une commande avec autopilotage et contrôle de la pulsation des courants rotorique Wg .

II.6. Alimentation en tension :

La loi de commande précédente permet de maintenir le flux constant. Mais elle est trop complexe pour être exploitée sans moyen de calcul puissant. Elle doit être simplifiée.

- Si la pulsation rotorique est très faible, alors :

$$V_s = \psi^*_s \omega_s \sqrt{\left(1+\frac{R_s}{\omega_s L_s}\right)^2} \quad , \tag{II-18}$$

Avec le coefficient de dispersion :

$$\sigma = 1 - \frac{M^2_{sr}}{L_s L_r} \tag{II-19}$$

Si, de plus, Rs est négligeable, alors :

$$V_s = \psi^*_s \omega_s \tag{II-20}$$

Ce qui caractérise une loi en **Ve /F=Cst**

II.7.Controle en V/f de la MAS :

Son principe est de maintenir V/f=Constant ce qui signifie garder le flux constant. figure (II.3) . Le contrôle du couple se fait par l'action sur le glissement. Quand la tension atteint sa valeur maximale, on commence alors à décroître ce rapport ce qui provoque une diminution du couple que peut produire la machine. On est en régime de "défluxage". Ce régime permet de dépasser la vitesse nominale de la machine, on l'appelle donc aussi régime de survitesse (partie Ω>Ωs de la Figure II.3).

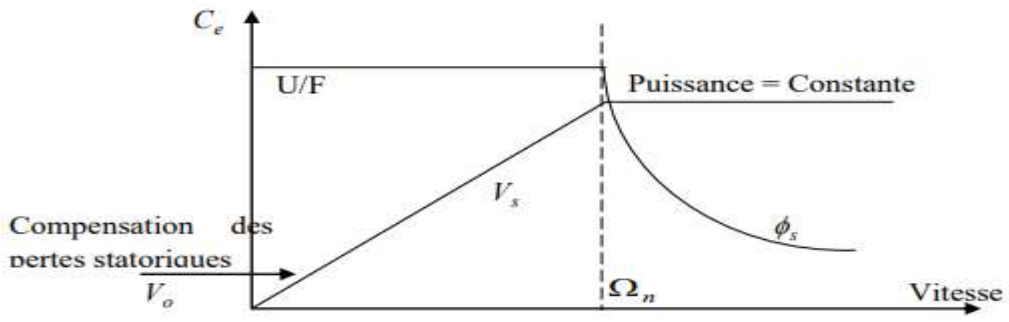


Figure II.3 : Caractéristique couple vitesse

À basse vitesse, la chute de tension ohmique ne peut pas être négligée. On compense alors en ajoutant un terme de tension V0 (Figure II.4).

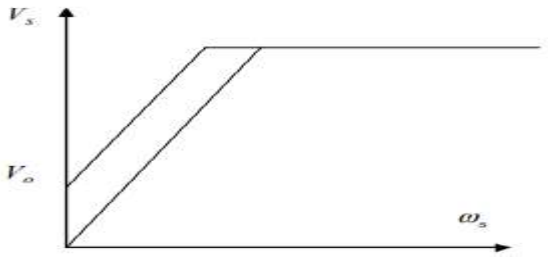


Figure II.4 : caractéristique V/f

II.8. Simulation de la commande scalaire de la MAS :

La simulation á été effectué par le logiciel «Simulink» sous «Matlab». Dans cette partie on présentera le cas d'une machine asynchrone triphasée alimentée par un onduleur triphasé de tension dont les paramètres de la MAS figure (II.5)

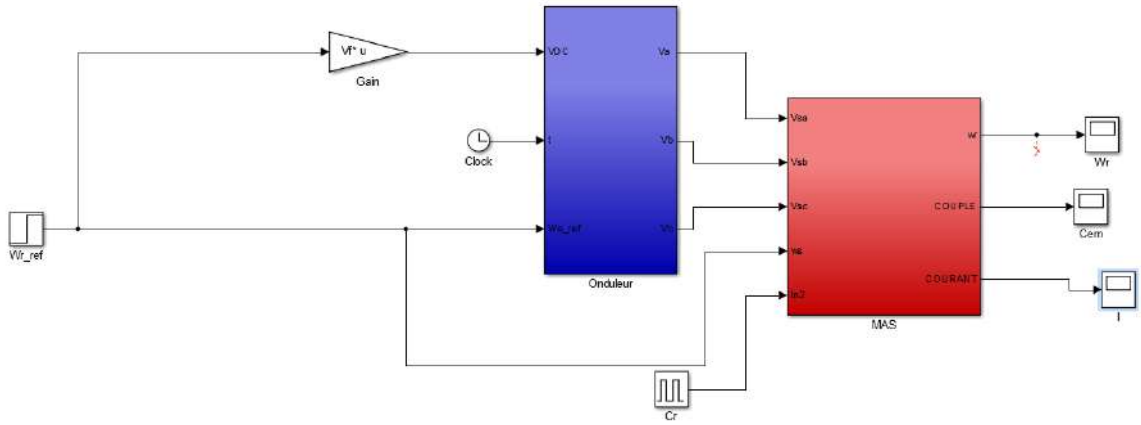
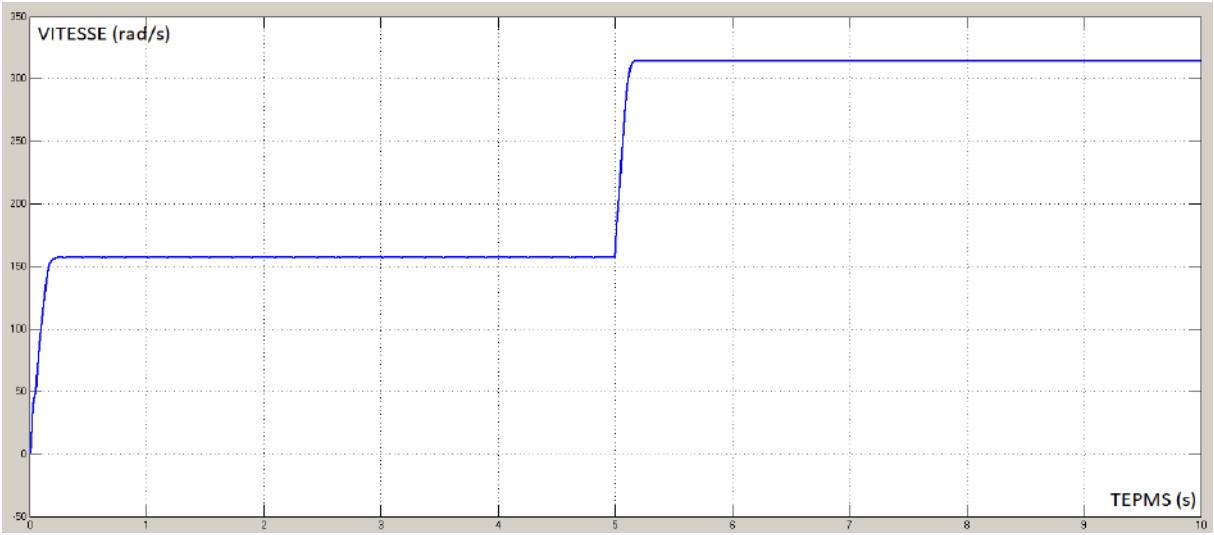
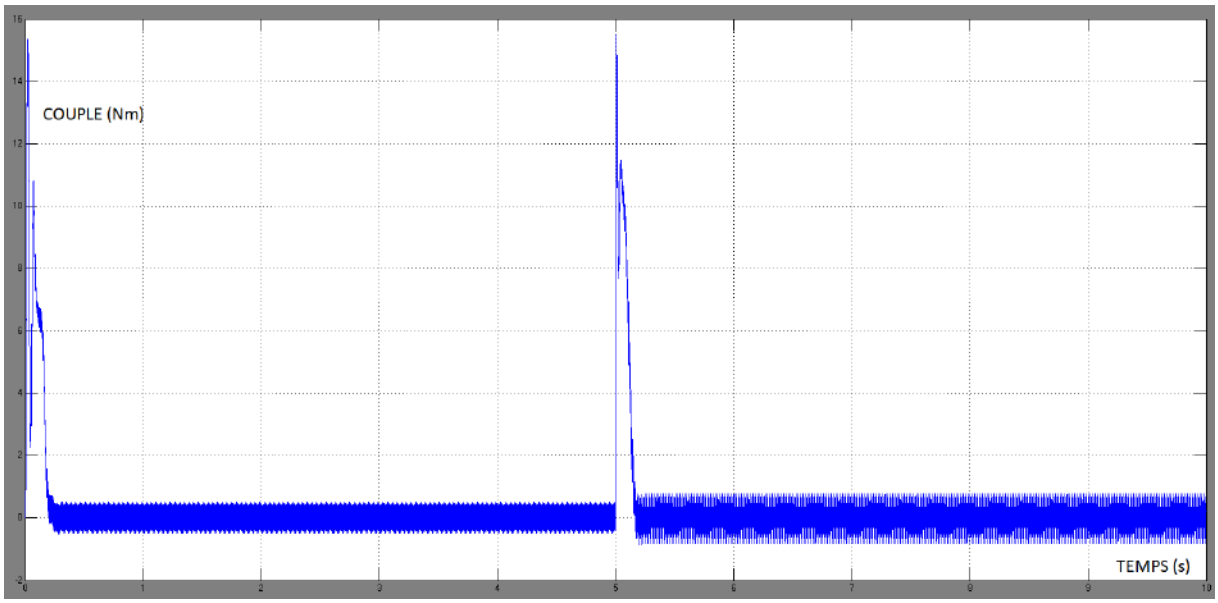


Figure II.5. Modèle Matlab/Simulink de la Commande Scalaire de la machine asynchrone sans régulateur PI

II.9. Résultats de simulation :



(a)Vitesse du rotation



(b) couple électromagnétique

Figure II.6.Performances de la MAS avec la commande scalaire sans régulateur PI

Interprétation des résultats :

A partir les résultats ci-dessus on remarque que la vitesse mécanique dans la figure (II.6.a) du la machine suit la consigne et après le changement du vitesse du référence a l'instant t=5s, la vitesse mécanique resuit la autre fois

Dans la figure (II.6.b) on constate que le couple et tjr approche au zéro (constant) malgré le changement du vitesse du référence au t=5s le couple envers l'approche du zéro (constant)

II.10. Principe général de la régulation

Dans la plupart des appareils dans des installations industrielles et domestiques, il est nécessaire de maintenir des grandeurs physiques à des valeurs déterminées, en dépit des variations extrêmes ou internes influant sur ces grandeurs. Par exemple, la vitesse et la position des moteurs, étant par nature variables, doivent donc être réglés par des actions convenables sur le processus considéré. Si les perturbations influant sur la grandeur à contrôler sont lentes ou négligeables, un simple réglage dit en boucle ouverte, permet d'obtenir et de maintenir la valeur demandée (par exemple : action sur un robinet d'eau). Dans la majorité des cas, ce type de réglage n'est pas suffisant, car trop grossier ou instable. Il faut alors comparer, en permanence, la valeur mesurée de la grandeur réglée à celle que l'on souhaite obtenir et agir en conséquence sur la grandeur d'action, dite grandeur réglant.

II.11 Performances des systèmes réglés

- Rapidité

La rapidité quantifie le temps de réponse du système. Elle correspond au temps de réaction de la sortie par rapport à la consigne. Le temps mis par la réponse pour ne plus dépasser 5% de la valeur finale. Ce temps est retenu comme critère de rapidité 5%

- Précision

La précision quantifié l'erreur lorsque l'équilibre est atteint, Avec l'entrée e(t) et la sortie s(t) de même nature. Autrement, un système est précis si la sortie suit la consigne en toutes circonstances avec un écart inférieur à la valeur définie dans un cahier des charges.

- Stabilité

On dit qu'un système est stable lorsque celui-ci tend à revenir à son état d'équilibre pour une consigne constante, la sortie doit être constante. figure (II.7)

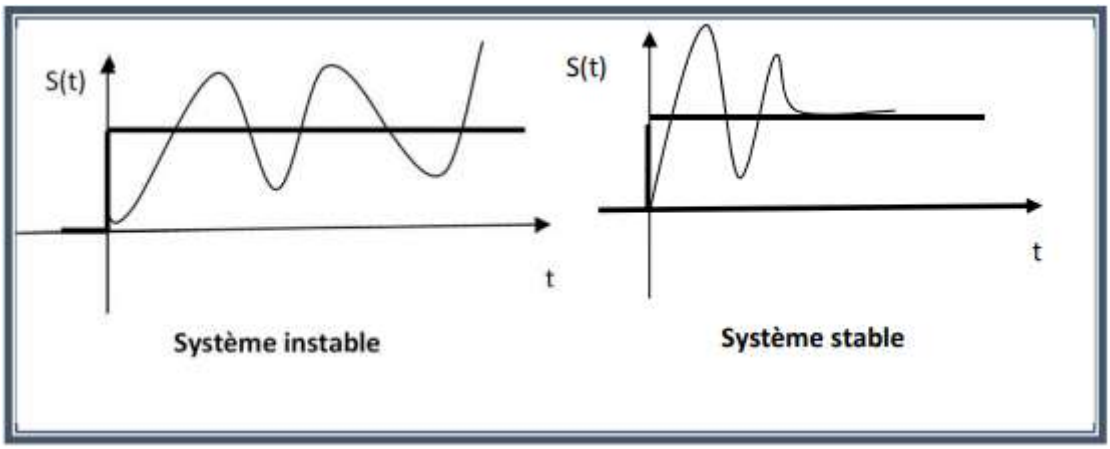


Figure II.7. Stabilité du système

II.12 Système en boucle ouverte

Un système est dit en boucle ouverte lorsque la commande est élaborée sans l'aide de la connaissance des grandeurs de sortie. Au niveau des inconvénients, il n'y a aucun moyen de contrôler, à plus forte raison de compenser les erreurs, les dérives, autrement dit, il n'y a pas de précision et surtout de fidélité qui dépendent de la qualité intrinsèque des composants. Enfin, le système en boucle ouverte ne compense pas les signaux de perturbation, le schéma bloc de système en boucle ouverte est donné par la figure suivante. figure (II.8)

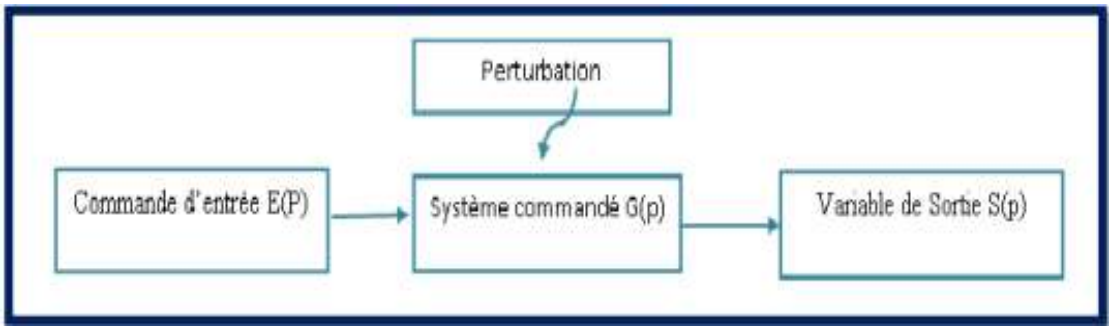


Figure II.8. Schéma bloc d'un système en boucle ouverte

Avec

S(p) : grandeur de (sortie)

E(p) : grandeur d'entrée. G(p) : fonction de transfert.

II.13 Système en boucle fermée

La boucle fermée (contre réaction) est capable de stabiliser un système instable en boucle ouverte. Dans une régulation en boucle fermée, une bonne partie des facteurs perturbateurs externes sont automatiquement compensés par la contre-réaction à travers le procédé. L'utilisation du retour d'information est le principe fondamental en commande électrique. La commande appliquée au système est élaborée en fonction de la consigne et de la sortie. On peut représenter une régulation de la manière dans la figure (II.9)

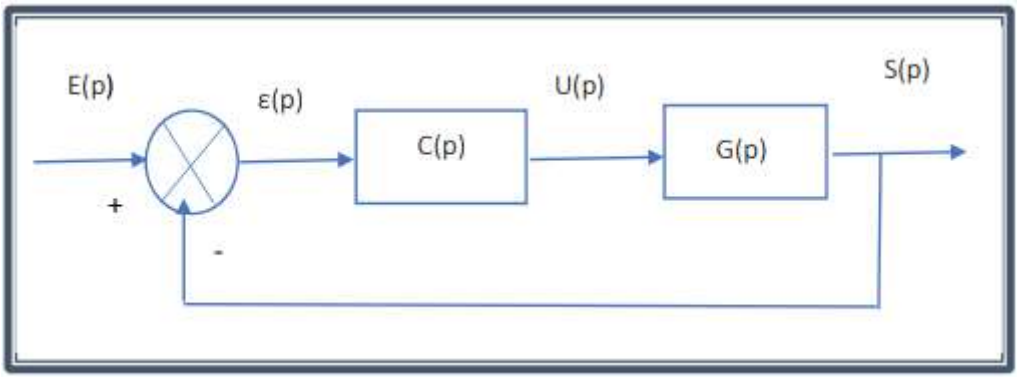


Figure II.9. Schéma bloc d'un système en boucle fermée

Avec

C(s) est la fonction de transfert du régulateur ;

G(s) est le processus à commander (machine asynchrone) ;

E(p) est la grandeur d'entrée ;

S(p) est la grandeur de sortie

ε(p) est l'erreur, ε(p) = E(p)-S(p)

II.14 La commande PI en régulation de vitesse

Le correcteur intégral est rarement utilisé seul, car son effet ne devient sensible que lorsque l'erreur dure depuis un certain temps .Pour obtenir une réponse initiale plus rapide, on l'utilise avec un correcteur proportionnel ,figure (II.10). La sortie d'un régulateur PI est de la forme suivante

$$U(t)=u0 +kc(Ɛ(t)+1/Ti\int Ɛ(t)dt \qquad \text{(II-21)}$$

La fonction de transfert du régulateur PI s'obtient en prenant la transformée de Laplace de l'équation précédente

$$C(p) = U(p) / \varepsilon(t) = Kc\ (1 + 1/Tip) \qquad \text{(II-22)}$$

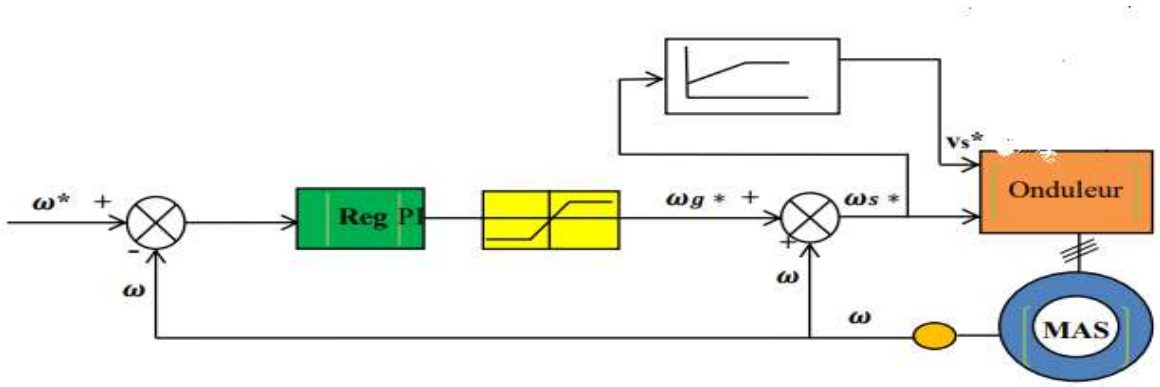


Figure II.10. Commande scalaire avec le contrôle du rapport V/f avec régulateur PI

II.15. Simulation de la commande scalaire de la MAS avec régulateur PI :

Une fois le modèle mathématique du notre moteur asynchrone ainsi que son alimentation est établi, nous pouvons aborder à la simulation de notre commande scalaire en utilisant le logiciel MATLAB / Simulink , figure II.11 , et en faisant plusieurs tests puis on visualise ces résultats.

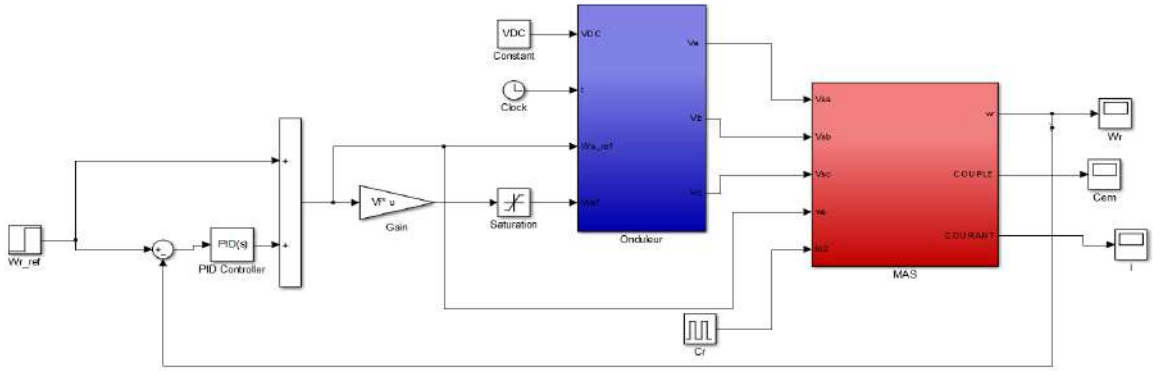
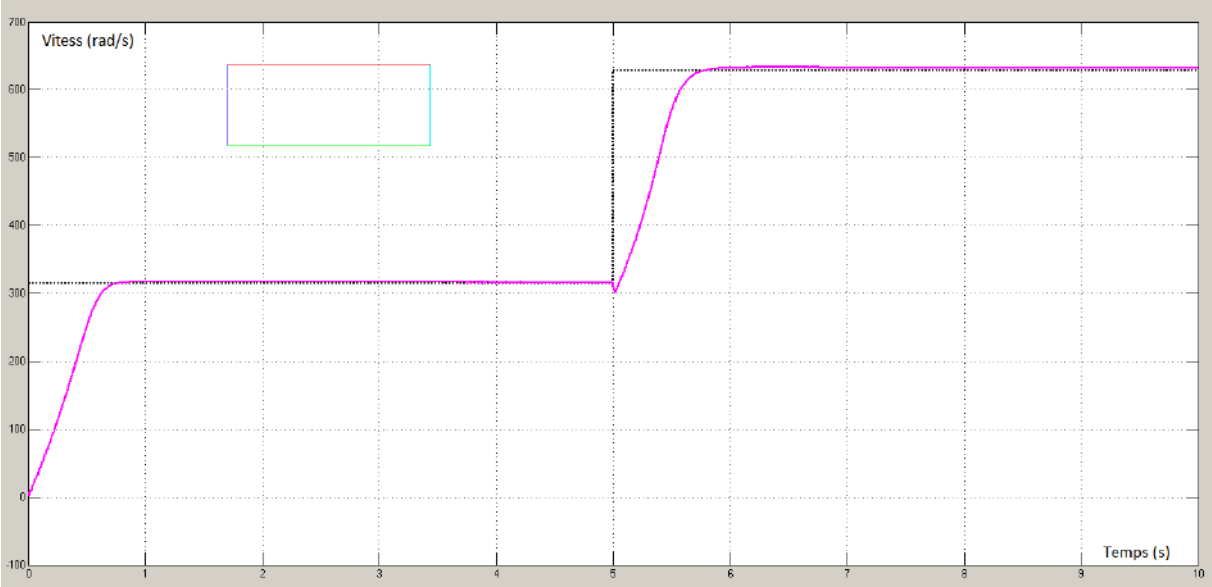


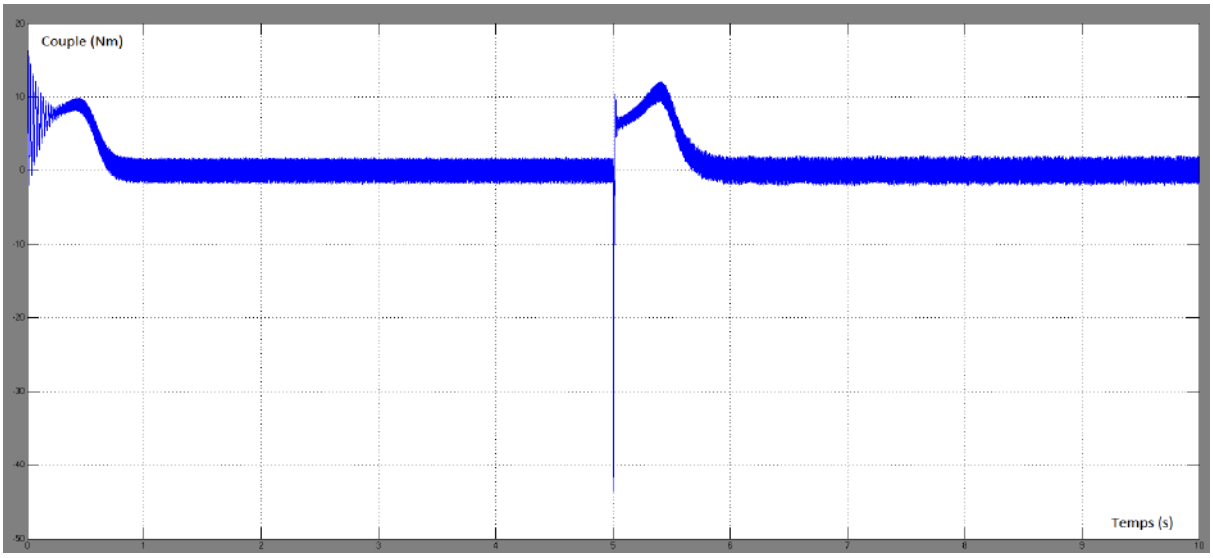
Figure II.11. Modèle Matlab/Simulink de la Commande Scalaire de la machine asynchrone avec régulateur PI

II.16. Résultats de simulation :

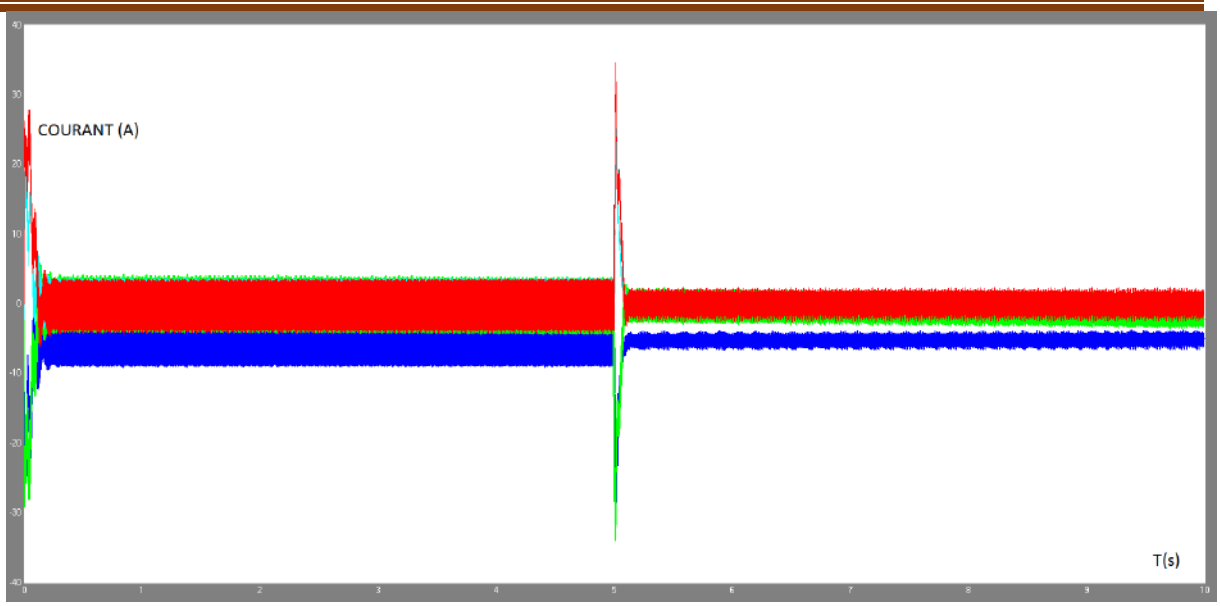
Pour valider notre algorithme de commande dans l'environnement Matlab/Simulink. Nous avons simulé la machine asynchrone pilotée par un contrôle scalaire et alimenté par un onduleur de tension à MLI.



(a) Vitesse du rotation



(b) couple électromagnétique



(c) Courant (i)

Figure II.12. Performances de la MAS avec régulateur PI

Interprétation des résultats

A partir les résultats ci-dessus on remarque que la vitesse mécanique dans la figure (II.12.a) du la machine suit la consigne et après le changement du vitesse du référence, elle faite un petit pic à cause du régulateur PI , la vitesse mécanique resuit la autre fois

Dans la figure (II.12.b) on constate que le couple et tjr approche au zéro (constant) malgré le changement du vitesse du référence au t=5s le couple envers l'approche du zéro (constant)

II.17.Optimisation Floue des paramètres du régulateur PI :

II.17.1 Principes et définitions de la logique floue : [17]

Une des caractéristiques du raisonnement humain est qu'il est généralement fondé sur des données imprécises ou même incomplètes. En effet, les connaissances dont nous disposons sur un système quelconque sont généralement incertaines ou vagues, soit parce que nous avons un doute sur leur validité ou alors nous éprouvons une difficulté à les exprimer clairement. Il est donc nécessaire de penser et de développer un nouveau type de raisonnement, le raisonnement approché, qui permettra de traiter mathématiquement l'imprécis et l'incertain.

Le premier à avoir souligné ces possibilités de développement est Lotfi A.Zadeh qui dès 1965 , introduit la théorie de la logique floue C'est une technique pour le traitement de connaissances imprécises et incertaines. Elle permet de prendre en considération des variables linguistiques dont les valeurs sont des mots ou des expressions du langage naturel, telle que faible, élevée, rapide, lent, grand, petit , etc….A cté d'un formalisme mathématique fort développé, nous préférons aborder la présentation de manière intuitive. Nous présentons un exemple simple pour comprendre l'intérêt de la logique floue sur la logique classique.

II.17.2 Variable linguistique

Une variable linguistique est représentée par un triplé (V, X, T_V), dans laquelle V est une variable (la vitesse, la température …) définie sur un ensemble de référence X (ses valeurs peuvent être n'importe quel élément de X) , On note $T_V(A_1, A_2, ...)$ n ensemble, fini ou infini, de sous-ensembles flous de X qui sont utilisés pour caractériser, qui sont utilisés pour caractériser. V Afin de permettre un traitement numérique, il est indispensable de les soumettre à une définition à l'aide de fonctions d'appartenance

II.17.3 Fonctions d'appartenance

Une définition des variables linguistiques à l'aide des fonctions d'appartenance est nécessaire dans le but de traiter des déductions floues par calculateur. Dans ce contexte, est attribuée à chaque valeur de la variable linguistique une fonction d'appartenance μ dont la valeur varie entre 0 et 1.Le plus souvent, nous utilisons des fonctions d 'appartenance de type triangulaire ou trapézoïdale figure VI.3.Fonction triangulaire : L'allure est complètement définie par trois paramètres $\{a,b,c\}$

$$\mu(x) = \max\left[\min\left[\frac{x-a}{b-a}, \frac{c-x}{c-b}\right], 0\right] \quad \text{(II-21)}$$

Fonction trapézoïdale : Elle est définie par quatre paramètres $\{a,b,c,d\}$

$$\mu(x) = \max\left[\min\left[\frac{x-a}{b-a}, 1, \frac{d-x}{d-c}\right], 0\right] \quad \text{(II-23)}$$



Figure II.13. Les Formes usuelles des fonctions d'appartenance

II.17.4 Opérations sur les ensembles flous

Puisque les fonctions caractéristiques des ensembles flous sont graduelles, l'extension de certaines opérations définies sur les ensembles classiques au cas des ensembles flous pose certaines difficultés. Il n'est donc pas possible d'appliquer directement la logique classique qui manipule des ensembles à fonctions caractéristiques binaires (0 ou 1). Pour pallier à ce problème, les définitions d'outils nécessaires au traitement des ensembles flous sont introduites.

Soient *B* et *A* deux ensembles flous dans *X* ayant respectivement μ_A, μ_B comme fonctions d'appartenance. L'union, l'intersection et le complément des ensembles flous sont définis à l'aide de leurs fonctions d'appartenance

Ces relations sont traduites par les opérateurs " Et " , "Ou " et " NON"

II.17.5.Structure générale du superviseur flou

La figure si déçue montre la structure générale du superviseur flou dont les entrées sont normalisées par un facteur d'échelle. On distingue trois parties :

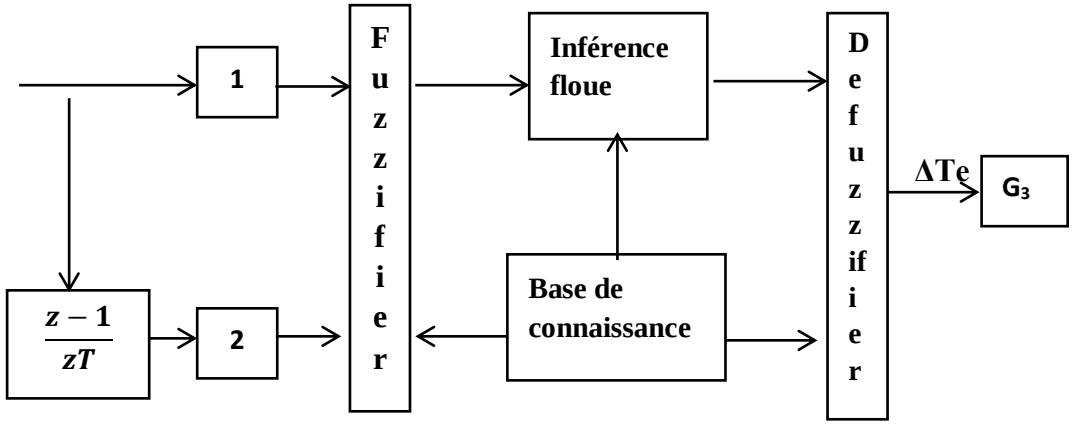


Figure II.14.Architecture du superviseur flou

II.18 Description du contrôleur flou :

Les paramètres du contrôleur PI utilisés sont pris normalisés dans l'intervalle [0, 1], en utilisant les transformations linéaires suivantes :

$$\begin{cases} K_p = K_{p0} + \Delta k_p \\ K_i = K_{i0} + \Delta k_i \end{cases} \quad \text{(II-23)}$$

Les entrées du contrôleur floue FLC sont : l'erreur e et la dérivé de l'erreur de, les sorties sont ; la valeur normalisée de l'action proportionnelle ' kp et la valeur normalisée de l'action intégrale ' ki.

Les sous-ensembles flous des variables d'entrées et les sorties sont définis comme suit :

- NB : grand négatif.
- NM: moyen négatif.
- NS : petit négatif
- ZO : zéro.
- PS : petit positif.
- PM : moyen positif.
- PB : grand positif

II.19 Stratégie du contrôle du vitesse

II.19.1 Système de control adaptatif flou

lorsque l'erreur est importante, augmenter le poids de la fonction de contrôle pour éliminer rapidement les erreurs, améliorer la vitesse de réponse du système; lorsque l'erreur est faible, augmente la quantité de changement dans le contrôle de la fonction de pondération d'erreur, et le dépassement du système en régime permanent sera réprimé. L'erreur e et le taux de variation de l'erreur ce sont des signaux d'entrée du contrôleur PI-FLOU, les différents moments de e et ce sont des erreurs pour les paramètres PI. La structure de contrôle PI-FLOU est représentée sur la figure (II.15). PI-FLOU adaptive consiste à identifier la floue entre Kp, Ki, e et ce du contrôleur PI, grâce à des tests continus sur e, ce en fonctionnement, en ressentant une théorie de contrôle floue pour modifier Kp, Ki deux paramètres en temps réel, il sera modifié par les exigences de différents paramètres de contrôle, et le système a une bonne performance de contrôle dans différentes situations.

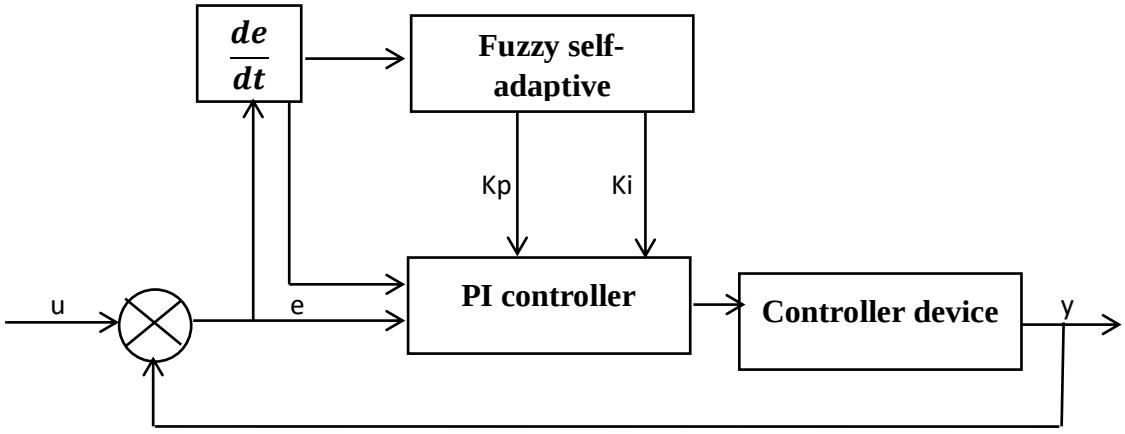


Figure II.15.CONTROLLEUR PI-FLOU

II.19.2 Conception de contrôleur floue

Cet partie construit une vitesse e et le taux de changement c e comme signaux d'entrée et de sortie comme signal de boucle de courant étant donné un contrôleur flou bidimensionnel. Le contrôleur flou sélectionne un contrôleur flou bidimensionnel pour entrer des variables pour la valeur de retour de vitesse et étant donné que la vitesse e et les valeurs d'erreur changent c e. La table matricielle floue en interrogeant les paramètres PI peut être ajustée. La procédure de conception du contrôleur PI-FLOU est la suivante: Prendre sept variables linguistiques: {grand négatif, milieu négatif, petit négatif, zéro, petit positif, milieu positif, CP}, les variables d'entrée et de sortie du sous-ensemble flou sont: {NB, NM, NS, ZO, PS , PM, PB}, sur le terrain sont: {- 3 - 2 - 1, 0,1, 2,3} .La fonction d'adhésion sélectionne la fonction d'appartenance au triangle,.La conception de base du contrôleur PI-FLOU est les règles de contrôle floues. Les règles de contrôle flou ont été acquises après de nombreuses expériences de contrôle moteur accumulées, ce qui est montré dans le tableau 1.

Tableau 1 :Base des régles du Δkp

Δe \ e	NB	NM	NS	EZ	PS	PM	PB
NB	PB	PB	PM	PM	PS	EZ	EZ
NM	PB	PB	PM	PS	PS	EZ	NS
NS	PM	PM	PM	PS	EZ	NS	NS
EZ	PM	PM	PS	EZ	NS	NM	NM
PS	PS	PS	EZ	NS	NS	NM	NM
PM	NS	EZ	NS	NM	NM	NM	NB
PB	EZ	EZ	NM	NM	NM	NB	NB

Tableau 2 : base des régles du Δki

Δe / e	NB	NM	NS	EZ	PS	PM	PB
NB	NB	NB	NM	NM	EZ	EZ	EZ
NM	NB	NB	NS	NS	NS	EZ	EZ
NS	NB	NM	NS	NS	EZ	PS	EZ
EZ	NM	NM	NS	EZ	PS	PM	PM
PS	NM	NS	EZ	PS	PS	PM	PM
PM	EZ	EZ	PS	PS	PM	PB	PB
PB	EZ	EZ	PS	PM	PM	PB	PB

II.1 Tableau de régles du Δkp et Δki

-Les tableaux 1 et 2 : [30] selon la règle de contrôle floue ont obtenu les valeurs Δkp, Δki, puis substituées dans l'équation (23) pour obtenir (Kp-Ki)

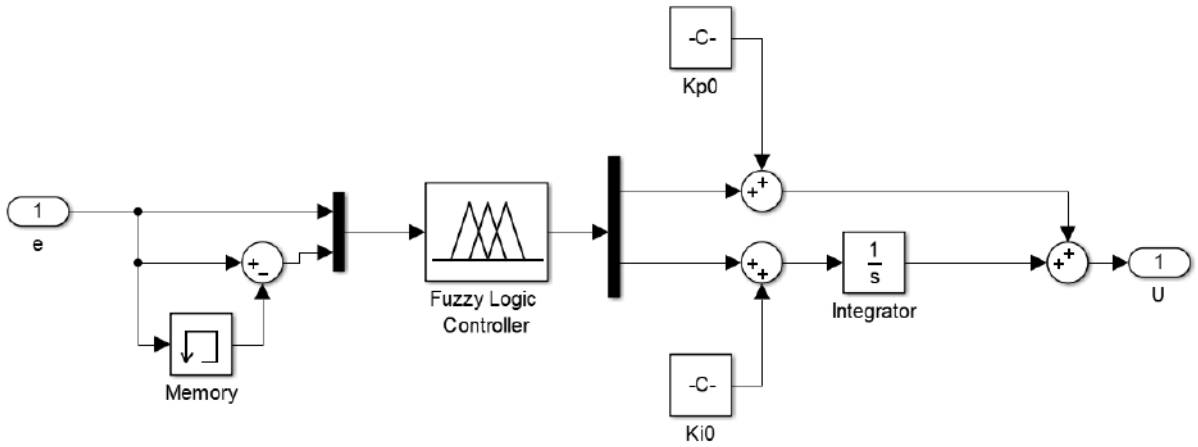
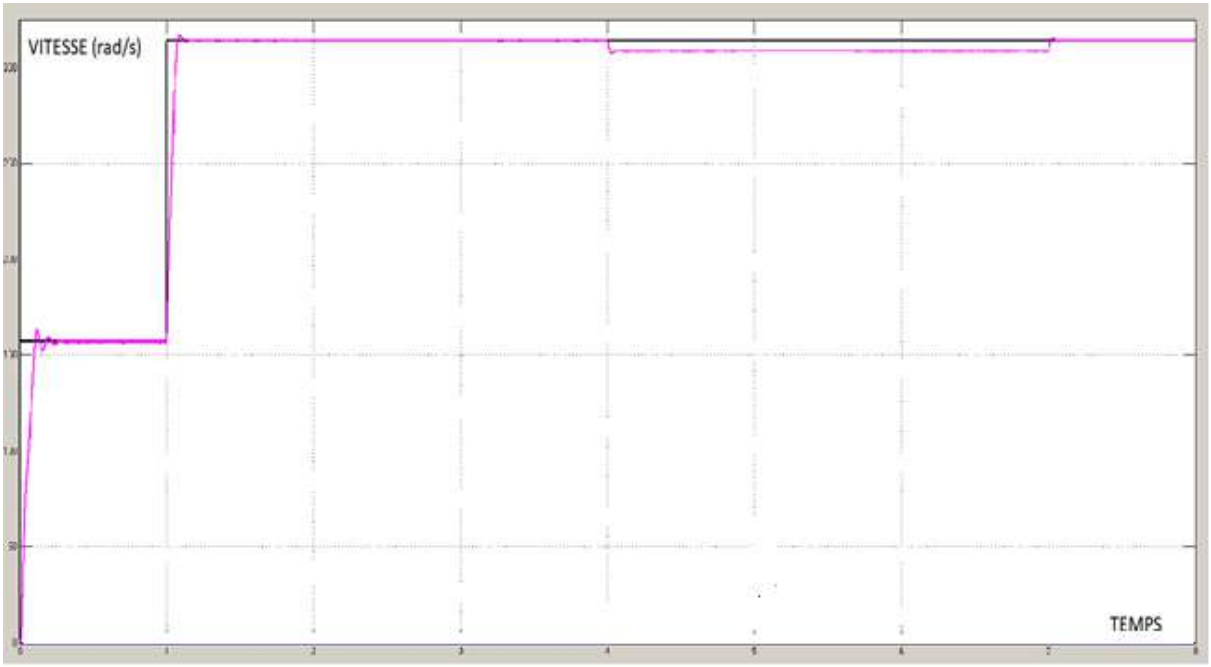
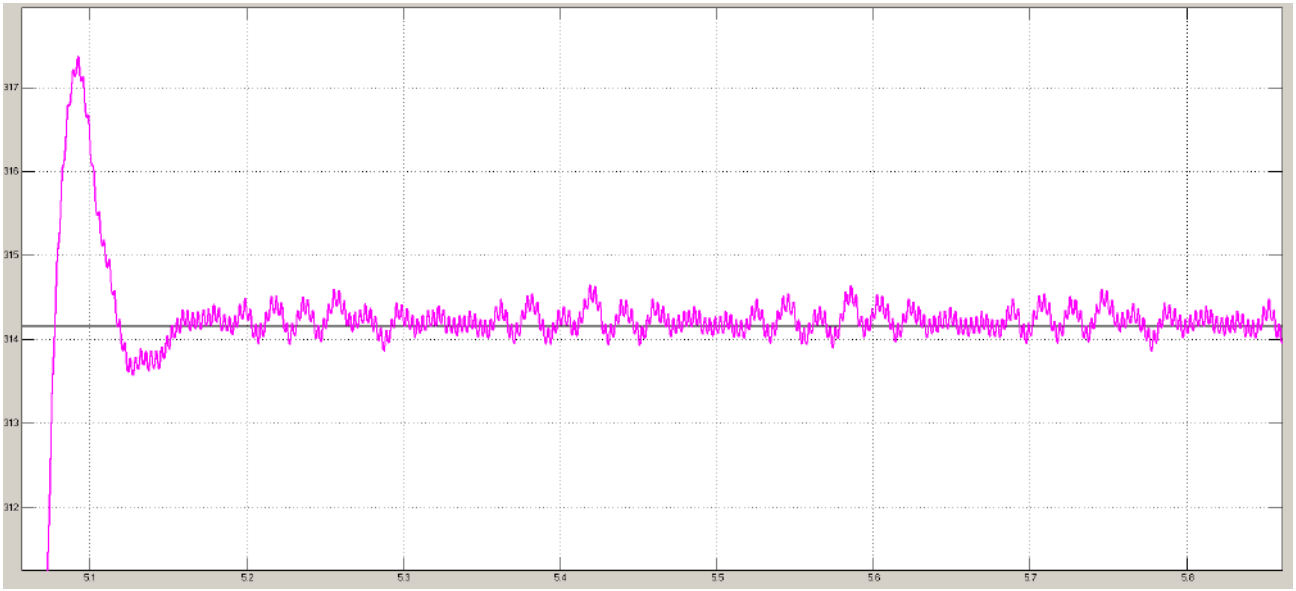


Figure II.16. La structure d'un contrôleur PI-FLOU

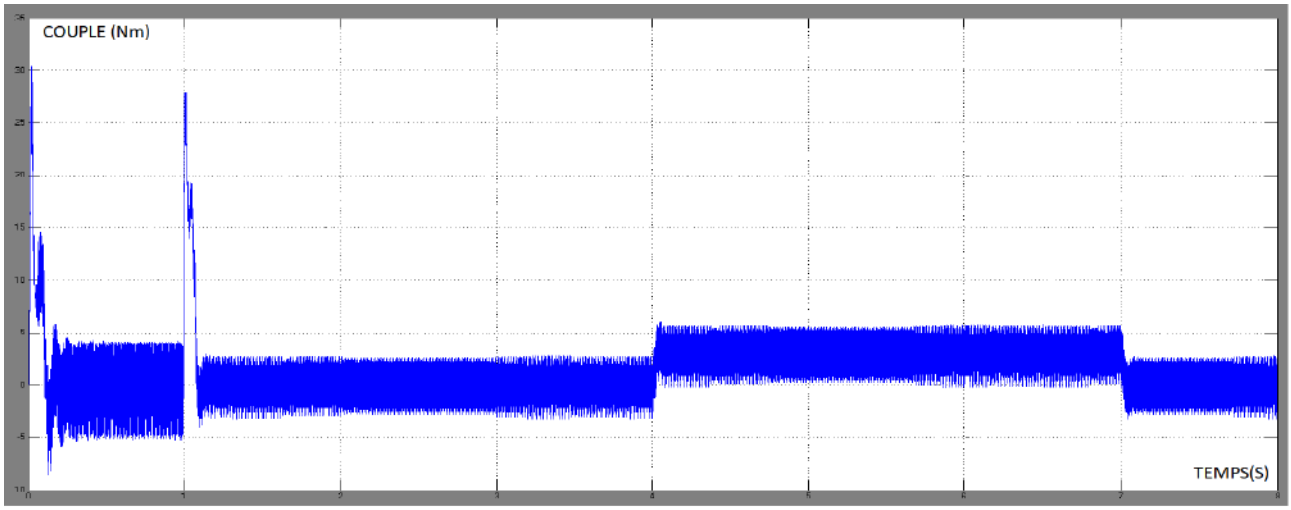
II.20. Résultats de simulation :



(a) La vitesse de rotation Wr(rad/s)



(b) Zoom a la vitesse de rotation lors de l'application du régulateur PI-FLOU



(c) Le couple électromagnétique (N.m) avec une charge du 3=Nm en t=4s

Figure II.17 : Performances de la MAS avec l'application du régulateur PI-FLOU.

Interprétation du résultats :

Dans la figure (II.17.a) on remarque que la vitesse mécanique du la MAS a une réponse rapide et suit toujours la consigne et elle le resuit malgré le changement du vitesse du référence et après t=4s on applique un couple de charge pendant 3s

La figure (II.17.b) le régulateur tourne avec l'estimée du vitesse , il la stabilise toujours vers la consigne donnée , on constate que le moteur donne de bonnes performances par rapport aux valeurs de consigne à l'entrée.

Concernant la figure (II.17.c) on constate que le couple envers l'approche du zéro (constant) et après l'application du couple de charge, il se stabilise approche du 3Nm.

II.21. Conclusion :

Le choix de la commande par logique floue présente plusieurs avantages : raisonnement proche du celui de l'homme c'est-à-dire traite l'imprécis, l'incertain et le vague, sa capacité à commander des systèmes non linéaire, obtention de meilleures performances dynamiques et ses qualités intrinsèques de robustesse.

on peut dire que le régulateur PI-FLOU ,offre une réponse satisfaisante par rapport au dépassement,

Chapitre III

Programmation d'un contrôleur flou dans PLC

III.1 Introduction

L'objectif de notre travail dans ce chapitre est d'implémenter une commande PI-floue pour une MAS. La programmation de la commande se fait à l'aide de logiciel TIA PORTAL V13 et le langage SCL (*Structured Control Language*).

- Première partie : Nous avons présenté dans ce chapitre la Gamme Simatic S7-300, le logiciel de programmation TIA PORTAL, PID FB41 (CONT_C), SCL, et une synthèse sur les théories des ensembles flous, ainsi que les outils mathématique nécessaire à leur manipulation.
- Deuxième partie : La Programmation d'un régulateur flou par langage SCL sous TIA PORTAL.

.

- Troisième partie : Comparaison les résultats avec SIMULINK MATLAB.

III.2 Présentation de quelques gammes SIMATIC :

Il existe plusieurs gammes de SIMATIC. On trouve le SIMATIC S7 et le SIMATIC M7 :

III.2.1 SIMATIC S7 :

Dans la gamme S7 on distingue trois grandes familles d'automates programmables industriels décrites dans ces paragraphes qui suivent :

III.2.2 SIMATIC S7-300 :

La famille S7-300 est constituée d'automates programmables de conception modulaire utilisés pour des automatismes de gamme, et peuvent être connectés entre eux au moyen d'un
câble-bus PROFIBUS.
Un automate S7-300 consiste en une CPU, un module d'alimentation PS, un module de signaux FM, un module de signaux SM et un processeur de communication. Comme indiqué dans la figure suivante [19].

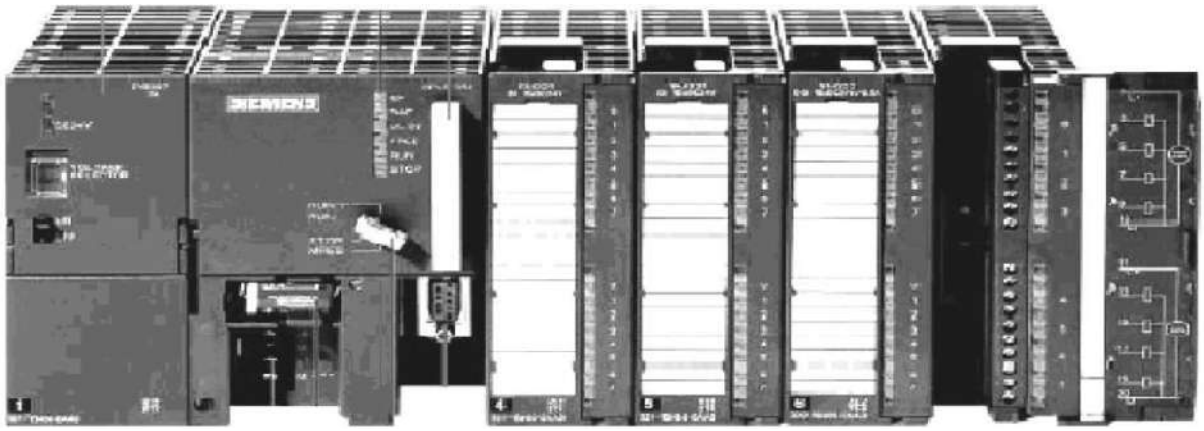


Figure III.1 API SIMENS S7-300

III.2.3 Présentation de l'automate S7-300 : [20]

L'automate **S7-300** est un mini automate modulaire, pour les applications d'entrée et de milieu de gamme, fabriqué par la firme SIMENS. On peut le composer en fonction de nos besoins à partir d'un vaste éventail de modules. La gamme des modules comprend :

- Des modules de signaux pour des entrées/sorties TOR et analogiques.
- Des modules de fonctions pour différentes fonctions technologiques.
- Des processus de communication (CP) pour les tâches de communications.
- Des modules d'alimentation pour le raccordement du S7-300 sur le réseau 230V

La figure ci-dessous représente la constitution d'un S7-300.

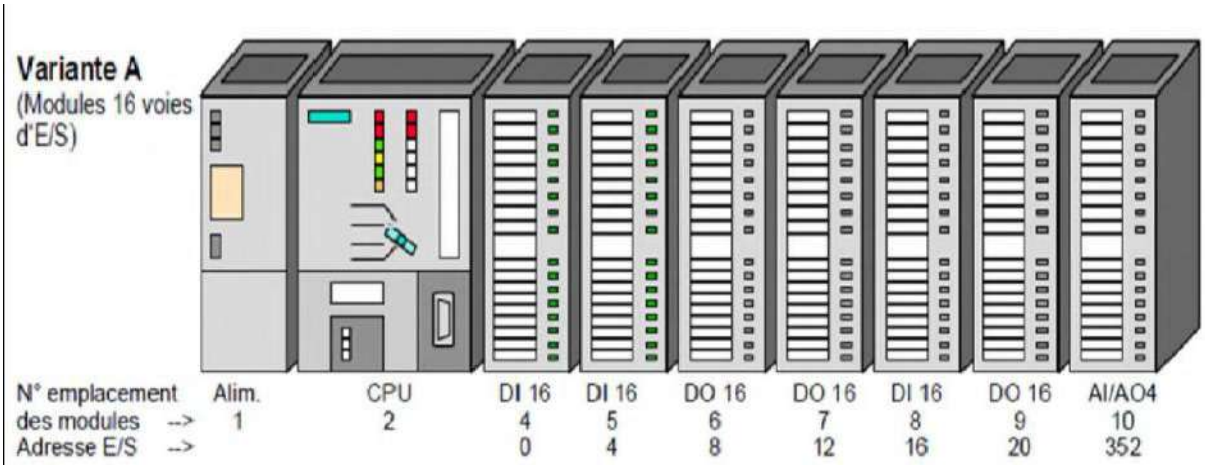


Figure III.2 Constitution d'API S7-300 [6].

- Emplacement 1 : alimentation 24V/5A
- Emplacement 2 : CPU 314
- Emplacement 3 : module de complémentaire
- Emplacement 4 : entrées TOR 16x24V
- Emplacement 5 : entrées TOR 16x24V
- Emplacement 6 : sorties TOR 16x24V 0.5A
- Emplacement 7 : sorties TOR 16x24V 0.5A
- Emplacement 8 : entrées TOR 16x24V
- Emplacement 9 : sorties TOR 16x24V 0.5A
- Emplacement 10 : module analogique 4AI/

III.2.4 Description de l'Automate S7-300 :

Le S7-300 est un automate modulaire d'une gamme excellente des produits SIMENS, il est synonyme de la nouvelle technologie des automates programmables. Le S7-300 est utilisé dans presque toutes les branches de l'industrie où les applications les plus variées. Sa modularité permet de réaliser des fonctions d'automatisations les plus diverses à partir des différents modules. Ses principales caractéristiques son :

- Sa puissance et sa rapidité.
- La possibilité d'intégration de nouvelles taches.
- Haute performance grâce aux nombreuses fonctions intégrées.

III.3 Logiciel de programmation « TIA Portal V13 » :

III.3.1 Présentation du logiciel :

La plate-forme TIA (Totally Integrated Automation) Portal est la dernière évolution des logiciels de travail Siemens qui permet de mettre en œuvre des solutions d'automatisation avec un système d'ingénierie intégré, dans un seul logiciel cette plate- forme regroupe la programmation des différents dispositifs d'une installation. On peut donc programmer et configurer, en plus de l'automate, les dispositifs HMI les variateurs… etc.

III.3.2 Conception d'un programme avec TIA PORTAL V13 :

La stratégie à suivre pour faire la conception d'un programme en utilisant la plate- forme TIA PORTAL V13 est :

- La création d'un nouveau projet.
- La configuration matérielle.
- La compilation et le chargement de la configuration.
- La création de la table des mnémoniques.
- L'élaboration du programme.
- La simulation avec le logiciel.
- La visualisation d'état du programme (le test) [21].

La conception d'une solution d'automatisation se fait par deux alternatives, soit on commence par la programmation ou par la configuration matérielle, dans notre cas on a commencé par la configuration [22].

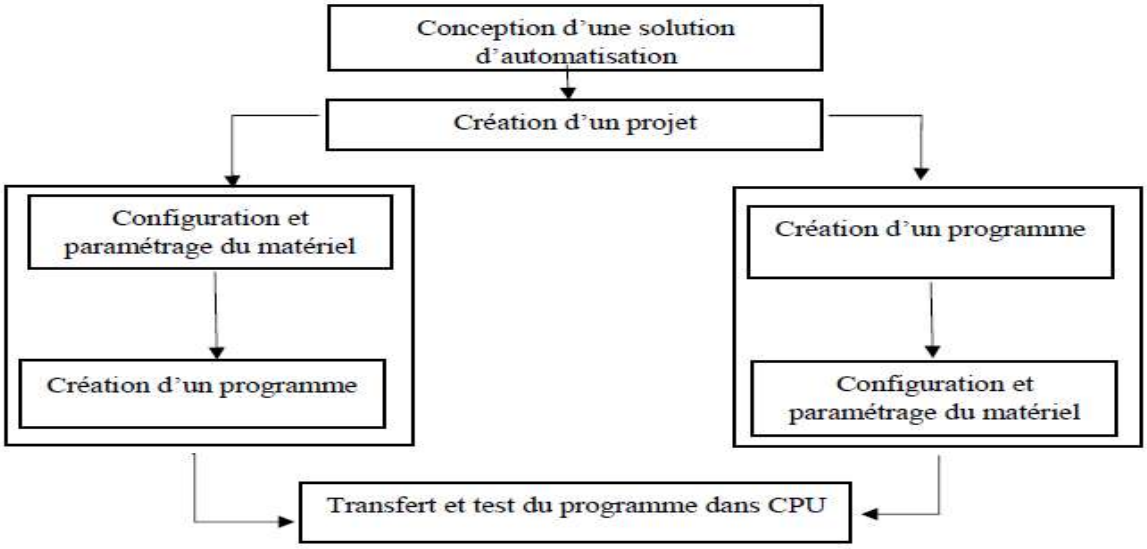
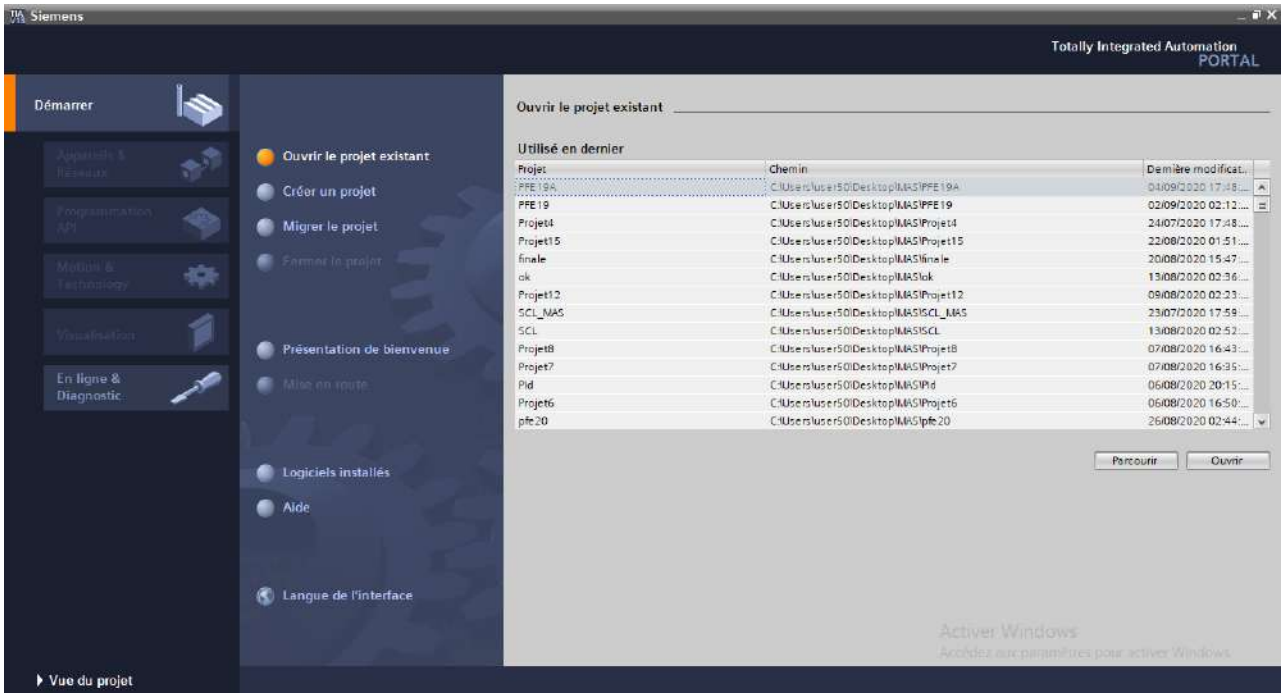


Figure III.3 Organisation pour la création d'un projet sous TIA PORTAL[22]**.**

III.3.3 Vue du portal :

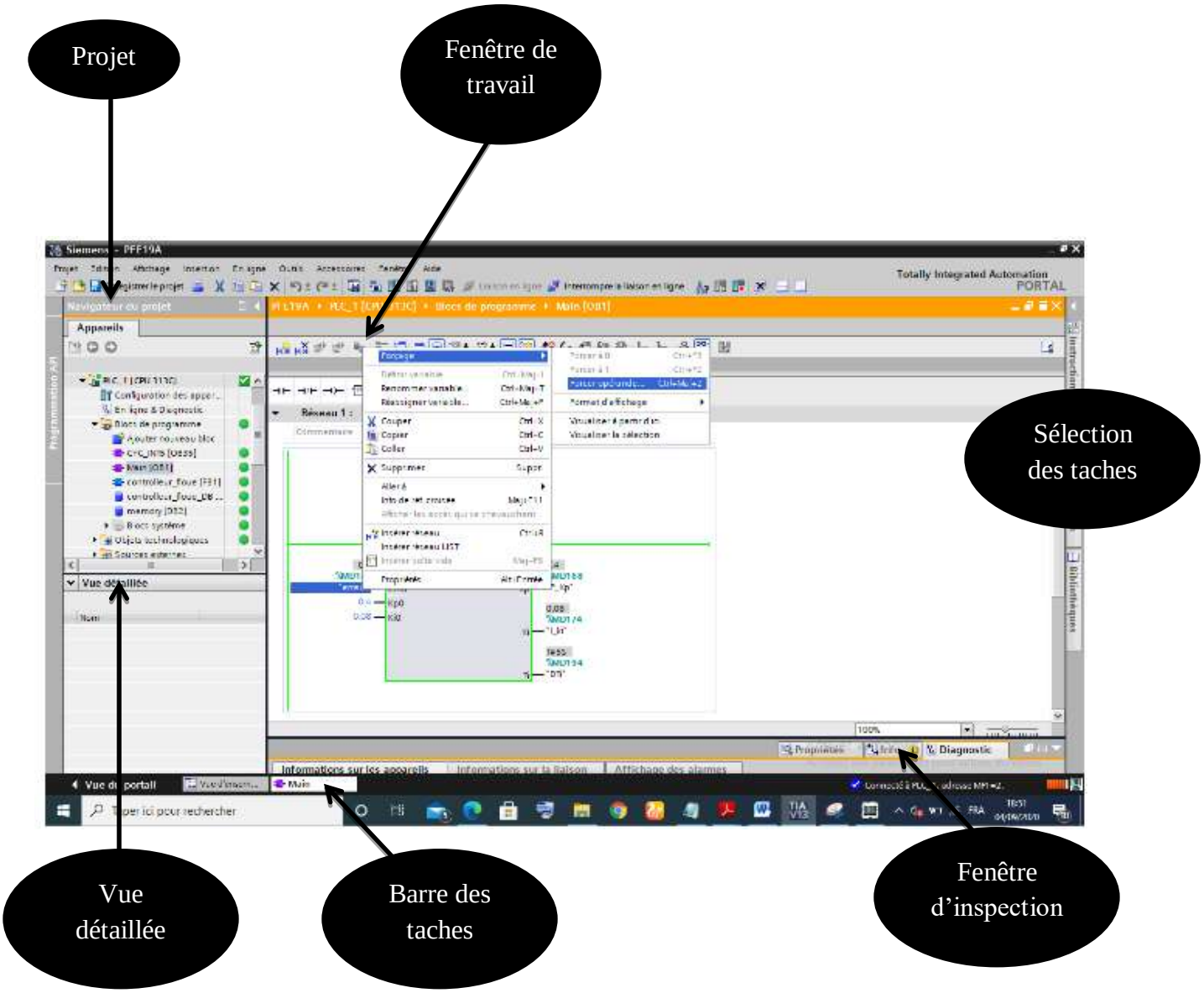
Chaque portail permet de traiter une catégorie de tâches (actions). La fenêtre affiche la liste des actions pouvant être réalisées pour la tâche sélectionnée.



FigureIII.4 Vue détaillée du portal

III.3.4 Vue du projet :

L'élément « Projet » contient l'ensemble des éléments et des données nécessaires pour mettre en œuvre la solution d'automatisation souhaitée [23].



FigureIII.5 Vue du projet

III.3.5 Création du projet dans TIA PORTAL V13 :

Afin de créer un nouveau projet TIA PORTAL V13, nous utilisant « l'assistant de création de projet », en cliquant sur « créer un projet » ce qui nous permet de commencer la configuration, cette méthode nous permet de gérer notre projet aisément. En sélectionnant l'icône « créer un projet », on affiche la fenêtre principale, on remplit le schnaps vides de notre fenêtre et on appuie sur le bouton « créer », la figure suivante représente la mise en route de notre projet.

1) Nom du projet
2) Spécifier le dossier dans lequel vous voulez enregistrer votre projet
3) Cliquez sur le bouton Créer

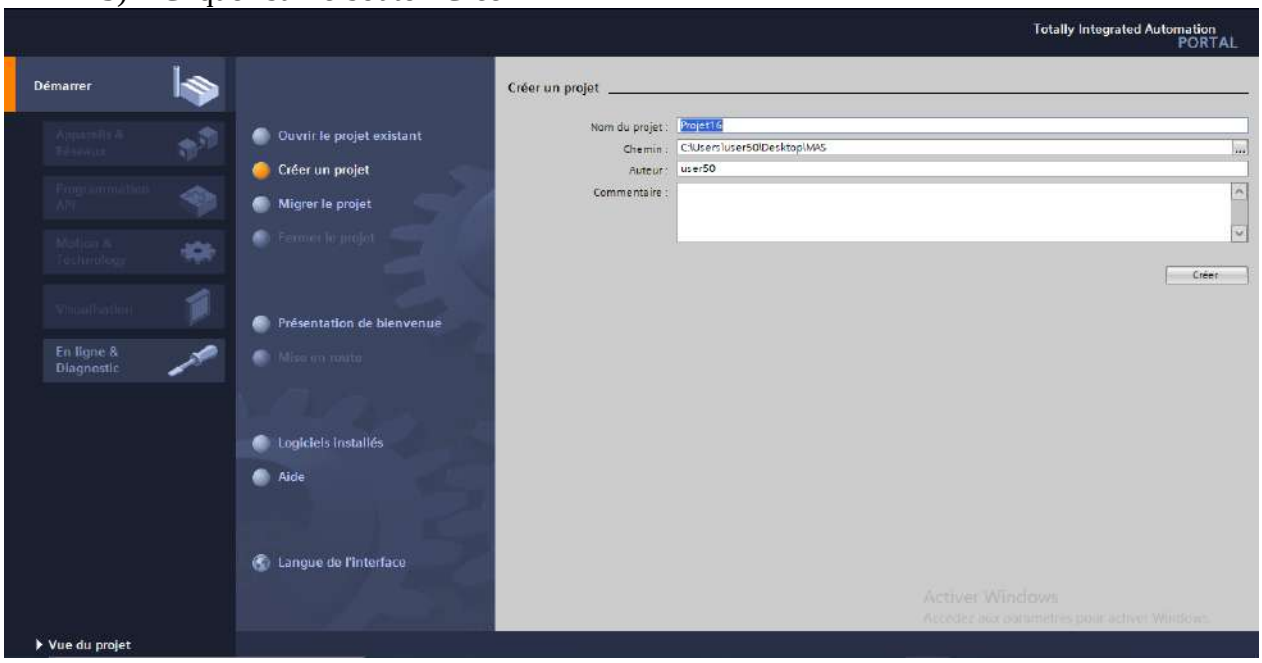


Figure III.6 Crée un projet

Après on clic sur le bouton Créer, notre projet est automatiquement sauvé sur le répertoire.

Il nous reste maintenant à choisir les appareils que nous voulons utiliser dans notre projet.

Pour cela :

1) Cliqué sur « Appareils & Réseaux »
2) « Ajouter un appareil »
3) « Contrôleurs »

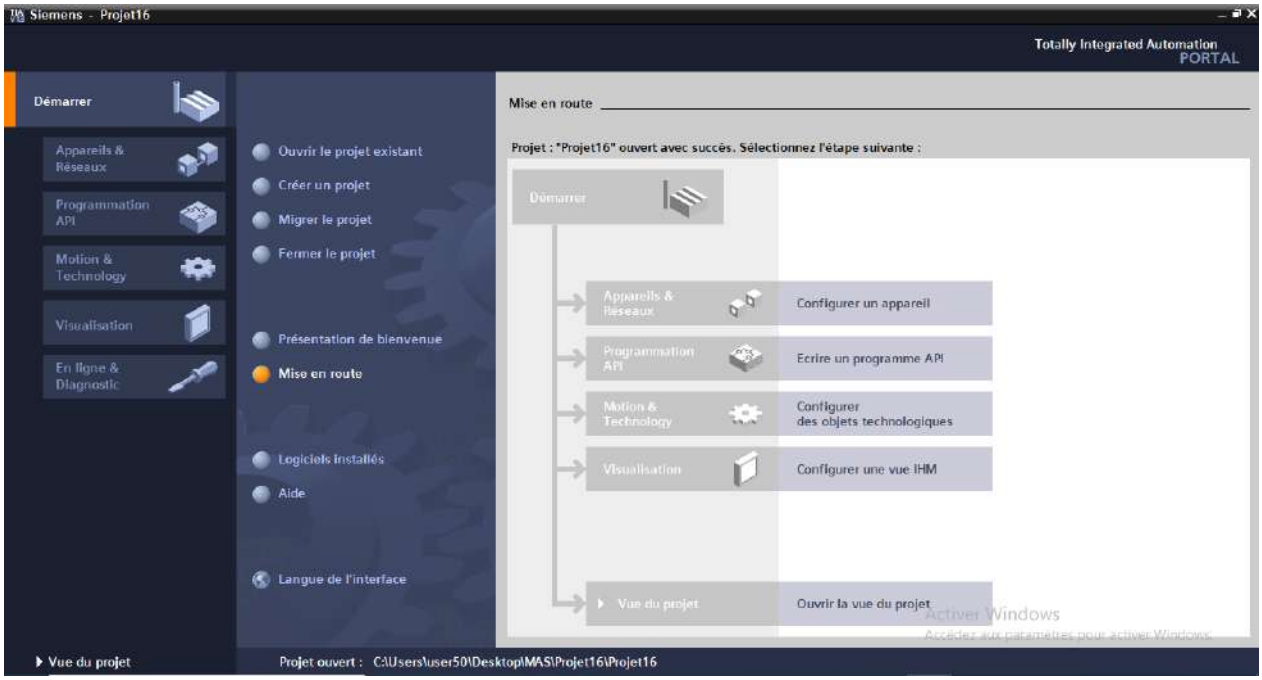


Figure III.7 Mise en route du projet

Dans notre cas, nous avons sélectionné un **CPU 313C** de type **S7-300** en faisant dérouler l'arborescence. Une fois notre automate choisi :

1) Donné un nom à notre automate

2) Double-cliquez sur le CPU choisi

Nous allons être basculés automatiquement vers la vue projet.

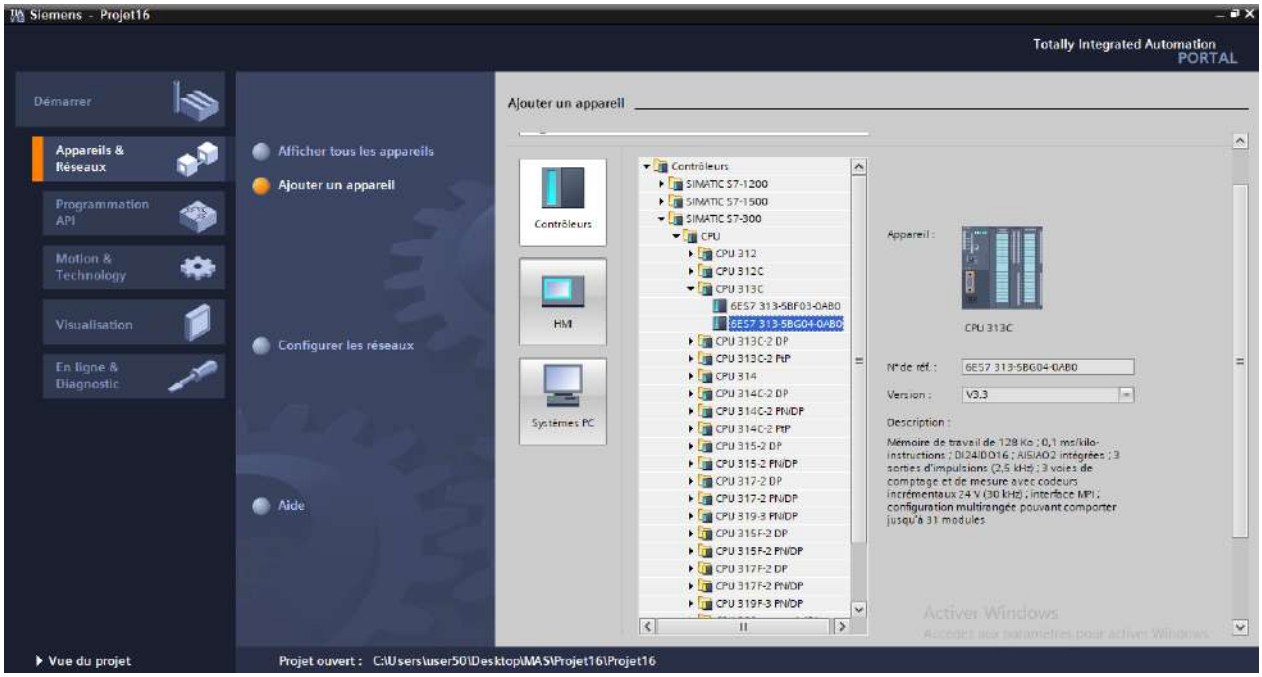


Figure III.8 Choisirons le CPU

III.4 Régulateur continue avec le FB 41 « CONT_C »

Le bloc FB 41 « CONT_C » sert à régler des processus industriels à grandeurs d'entrée et de sortie continues sur les automates programmables SIMATIC S7. Le paramétrage vous permet d'activer ou de désactiver des fonctions partielles du régulateur PID et donc d'adapter ce dernier au système réglé.

III.5 La Programmation d'un régulateur floue sous TIA PORTAL

III.5.1. La Programmation du bloc FB 41

Le logiciel **TIA PORTAL** offre des blocs fonctionnels (FB) de régulation
PID comprennent les blocs pour :

- La régulation continue.
- La régulation pas à pas.
- La modulation de largeur d'impulsion

Type de régulation	Bloc de régulation	Type de la sortie de régulation
Continue	FB41	Analogique
Pas à pas	FB42	Impulsion
Largeur d'impulsion	FB43	Analogique

Tab. III.1: Les différents types de régulation PID sous TIA PORTAL.

Les FB de régulation proposent une régulation purement logicielle, c'est-à-dire qu'un bloc contient toutes les fonctions du régulateur. Les données nécessaires au calcul cyclique sont stockées dans des blocs de données associés OB, les blocs de données d'instance, ce qui permet aux FB de les appeler plusieurs fois.

Le bloc FB41 sert à réguler des processus industriels à grandeurs d'entrée et de sortie continues sur les automates programmables SIMATIC S7. Le paramétrage du bloc

FB41 nous permet d'activer ou de désactiver des fonctions partielles du régulateur PID et donc d'adapter ce dernier au système régulé [24].

III.5.2 Description du bloc FB41 (CONT_C) :

Le FB réalise un régulateur PID prêt à l'emploi avec sortie continue de la grandeur de réglage et possibilité d'influencer à la main la valeur de réglage. Il propose les fonctions partielles suivantes [24].

A. Branche de consigne :

La consigne est entrée en format de virgule flottante à l'entrée **SP_INT**.

B. Branche de mesure :

La mesure peut être lue en format de périphérie ou de virgule flottante. La fonction

CRP_IN convertit la valeur de périphérie PV_PER en un nombre à virgule flottante compris entre –100 et +100 % selon la formule suivante :

$$\textbf{Sortie de CPR_IN} = \text{PV_PER} * \frac{100}{27648} \qquad \text{(III-1)}$$

La fonction PV_NORM normalise la sortie de CRP_IN selon la formule suivante :

Sortie de PV_NORM = (sortie de CPR_IN) _ PV_FAC + PV_OFF

La valeur par défaut de PV_FAC est 1 et celle de PV_OFF est 0.

C. Formation du signal d'erreur :

La différence entre consigne et mesure donne le signal d'erreur. Il traverse une zone morte (DEADBAND) pour supprimer la petite oscillation permanente due à la quantification de la grandeur de réglage (par exemple en cas de modulation de largeur d'impulsions avec PULSMGEN). Quand DEADB_W égale 0, la zone morte est désactivée.

D. Algorithme PID :

L'algorithme PID travaille dans l'algorithme de position. Les actions proportionnelle, intégrale (INT) et dérivée (DIF) sont en parallèle et peuvent être activées ou désactivées séparément. Ceci permet de paramétrer des régulateurs P, PI, PD et PID, mais aussi un régulateur I pur.

E. Traitement de la valeur de réglage manuelle :

On peut passer du mode automatique au mode manuel et inversement. En mode manuel, la grandeur de réglage est asservie à la valeur de réglage manuelle.

L'intégrateur (INT) est positionné et aligné de façon interne sur LMN – LMN_P –

DISV et le différentiateur (DIF) est forcé à 0. Ainsi, le passage au mode automatique s'effectue sans à-coup.

F. Traitement de la valeur de réglage :

La fonction LMNLIMIT permet de limiter la grandeur de réglage à des valeurs que vous indiquez. Le dépassement de ces limites par la grandeur d'entrée est signalé par des bits.

La fonction LMN_NORM normalise la sortie de LMNLIMIT selon la formule suivante :

LMN = (sortie de LMNLIMIT) _ LMN_FAC + LMN_OFF

La valeur par défaut de LMN_FAC est 1 et celle de LMN_OFF est 0.

La valeur de réglage est disponible aussi en format de périphérie. La fonction

CRP_OUT convertit la valeur à virgule flottante LMN en une valeur de périphérie selon la formule :

$$\mathbf{LMN_PER = LMN * \frac{100}{27648}} \qquad \text{(III-2)}$$

III.5.3 Le fonctionnement de bloc PID FB41 (CONT_C) :

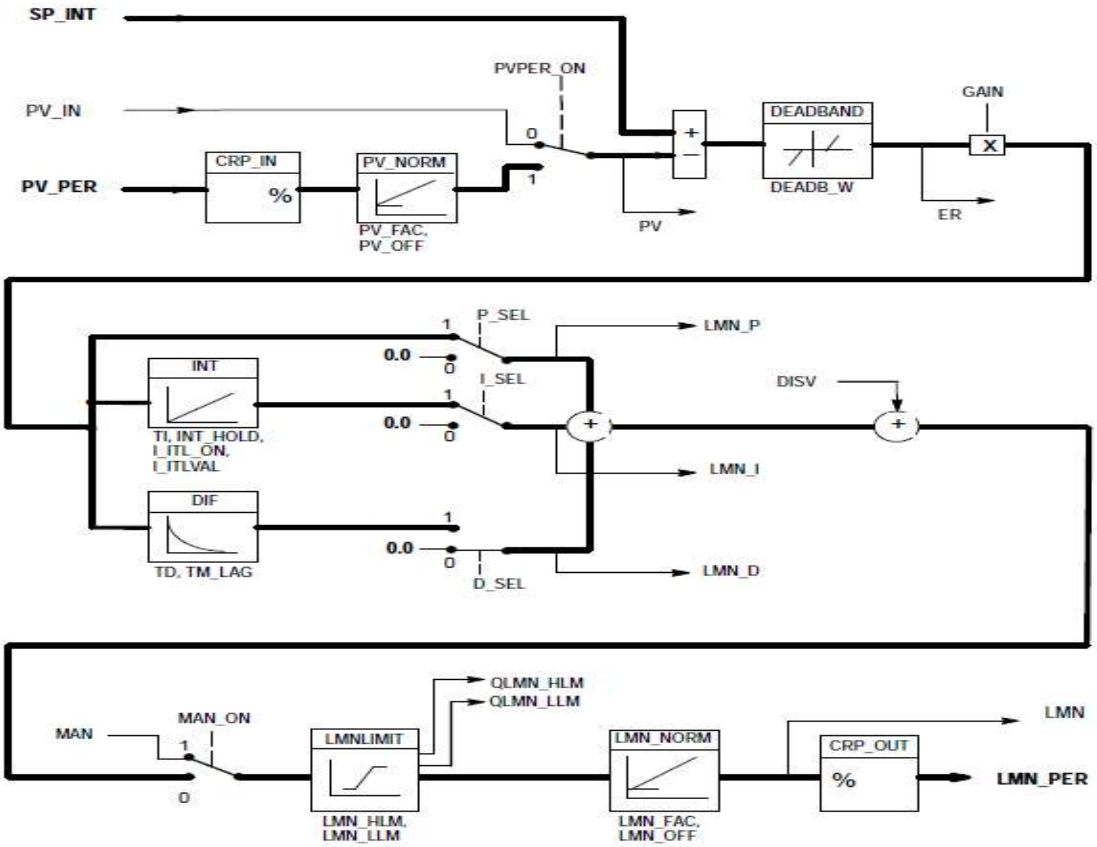


Figure III.9 Schéma fonctionnel de CONT_C [24].

III.5.4 Les paramètres du bloc FB 41 « CONT_C » :

A. Paramètres d'entrée :

Tab III.2 Paramètres d'entrée du BLOC FB41

Paramètres	Type de données	Valeur admises	Par défaut	Description
COM_RST	BOOL		FALSE	Le bloc renferme un sous-programme de démarrage qui est exécuté quand cette entrée est à 1.
MAN_ON	BOOL		TRUE	La valeur de réglage manuelle est sortie comme grandeur de réglage.

PVPER_ON	BOOL		FALSE	Pour que la mesure soit lue en périphérie, il faut relier l'entrée PV_PER à la périphérie et mettre à 1 l'entrée PVPER_ON.
P_SEL	BOOL		TRUE	L'action P est active quand cette entrée est à 1.
I_SEL	BOOL		TRUE	L'action I est active quand cette entrée est à 1.
INT_HOLD	BOOL		FALSE	La sortie de l'intégrateur peut être gelée. Pour cela, il faut mettre à 1 cette entrée.
I_ITL_ON	BOOL		FALSE	La sortie de l'intégrateur peut être positionnée sur la valeur initiale I_ITL_VAL. Pour cela, il faut mettre à 1 cette entrée.
D_SEL	BOOL		FALSE	L'action D est active quand cette entrée est à 1.
CYCLE	TIME	>= 1ms	T#1s	Le temps s'écoulant entre les appels de bloc doit être constant.
SP_INT	REAL	-100 à 100		Cette entrée sert à introduire une valeur de consigne.
PV_IN	REAL	-100 à 100		Cette entrée permet de paramétrer une valeur de mise en service ou d'appliquer une mesure externe en virgule flottante.
PV_PER	WORD			La mesure en format de périphérie est appliquée au régulateur par cette entrée.
MAN	REAL	-100 à 100		Cette entrée sert à introduire une valeur de réglage manuelle moyennant des fonctions de contrôle-commande.
GAIN	REAL			Cette entrée indique le gain du régulateur.
TI	TIME	>=CYCLE	T#20s	Cette entrée détermine la réponse temporelle de l'intégrateur.
TD	TIME	>=CYCLE	T#10s	Cette entrée détermine la réponse temporelle de l'unité de dérivation.

TM_LAG	TIME	>=CYCLE/2	T#2s	L'algorithme de l'action D contient un retard qui peut être paramétré à cette entrée.
DEADB_W	REAL	>=0	0.0	Cette entrée détermine la taille de la zone morte.
LMN_HLM	REAL	LMN_LLM à 100	0.0	Cette entrée indique la limite supérieure.
LMN_LLM	REAL	-100 à LMN_HLM	0.0	Cette entrée indique la limite inférieure
PV_FAC	REAL		1.0	Facteur de mesure Cette entrée est multipliée par la mesure.
PV_OFF	REAL		0.0	Cette entrée est ajoutée à la mesure. Elle sert à adapter l'étendue de valeur de mesure.
LMN_FAC	REAL		1.0	Cette entrée est multipliée par la valeur de réglage
LMN_OFF	REAL		0.0	Cette entrée est ajoutée à la valeur de réglage.
I_ITVAL	REAL	-100 à 100	0.0	Valeur d'initialisation pour l'action par intégration La sortie de l'intégrateur.
DISV	REAL	-100 à 100	0.0	Grandeur perturbatrice La grandeur perturbatrice est appliquée à cette entrée pour l'action anticipatrice.

B. Paramètres de sortie

Paramètres	Type de données	Valeur Admises	Par défaut	Description
LMN	REAL		0.0	Cette sortie donne en virgule flottante la valeur de réglage agissant réellement.
LMN_PER	WORD		W#16#0000	Cette sortie fournit la valeur de réglage en format de périphérie.
QLMN_HLM	BOOL		FALSE	Cette sortie signale le dépassement de la limite supérieure.
QLMN_LLM	BOOL		FALSE	Cette sortie signale le dépassement de la limite inférieure.
LMN_P	REAL		0.0	Cette sortie contient la composante proportionnelle de la grandeur de réglage.
LMN_I	REAL		0.0	Cette sortie contient la composante intégrale de la grandeur de réglage.
LMN_D	REAL		0.0	Cette sortie contient la composante différentielle de la grandeur de réglage.
PV	REAL		0.0	Cette sortie donne la mesure effective.
ER	REAL		0.0	Cette sortie donne le signal d'erreur effectif.

Tab III.3 Paramètres de sortie du BLOC FB41

III.6 Langage de programmation SCL :

SCL (Structured Control Language) est un langage de programmation évolué apparenté au langage PASCAL qui permet une programmation structurée. Le langage est conforme à la norme EN-61131-3 (CEI 61131-3) définissant le langage de programmation ST "Structured Text".

En plus des éléments de langage évolué, SCL contient également des éléments typiques de l'API comme éléments de langage tels que les entrées, sorties, temporisations, mémentos, appels de blocs, etc. Il prend en charge le concept de blocs de STEP 7 et permet ainsi une programmation des blocs

conforme aux normes en plus de la liste d'instructions (LIST), du schéma à contacts (CONT) et du logigramme (LOG). C.à.d. que SCL complète et étend le logiciel de programmation **TIA PORTAL** avec ses langages de programmation CONT et LOG.

Il n'est pas nécessaire de créer chaque fonction, vous pouvez utiliser des blocs préprogrammés, comme les fonctions systèmes et les blocs fonctionnels système qui se trouvent dans le système d'exploitation de l'unité centrale.

On peut mélanger des blocs programmés avec SCL avec des blocs CONT et LOG. Cela signifie qu'un bloc programmé avec SCL peut appeler un autre bloc programmé dans CONT ou LOG. De la même façon, les blocs SCL peuvent également être appelés dans des programmes CONT ou LOG.

Des réseaux SCL peuvent également être insérés dans des blocs CONT et LOG.

Les fonctions de test de SCL permettent de rechercher les erreurs logiques de programmation dans une compilation correcte.[25]

III.6.1 L'environnement de développement SCL

Pour utiliser et mettre en œuvre SCL, vous disposez d'un environnement de développement conçu pour répondre à la fois aux particularités de SCL. Cet environnement est composé d'un éditeur/compilateur et d'un débogueur. .[25]

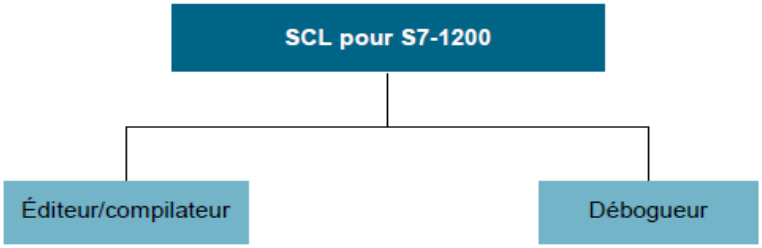


Figure III.10 environnement du SCL

III.6.2 Éditeur/compilateur

L'éditeur SCL est un éditeur de texte permettant d'éditer tous types de textes. Il vous permet principalement de créer et d'éditer des blocs pour des programmes STEP 7. Durant la saisie, un contrôle de syntaxe approfondi permet d'assurer avec facilité une programmation sans erreur. Les erreurs de syntaxe sont affichées en différentes couleurs.

L'éditeur offre les possibilités suivantes [25]:

- Programmation d'un bloc S7 dans le langage SCL.
- Ajout d'éléments de langage et d'appels de bloc par simple glisser-déposer.

- Contrôle de syntaxe directement pendant la programmation.
- éléments de langage en fonction de leur syntaxe
- Vérification du bloc terminé par compilation.
- Affichage des erreurs et avertissements qui surviennent pendant la compilation.
- Localisation de l'erreur dans le bloc, éventuellement avec description de l'erreur et proposition de correction.

III.6.3 Débogueur :

Le débogueur SCL peut contrôler l'exécution d'un programme dans l'AS et détecter ainsi les erreurs logiques éventuelles.

SCL propose à cet effet deux types de test [25]:

- Visualisation par étape.
- Visualisation en continu.

La "visualisati1on en continu" permet de tester un groupe d'instructions dans un bloc. Pendant le test, les valeurs des variables et des paramètres s'affichent chronologiquement et sont actualisées, autant que possible, de manière cyclique.

En mode "visualisation par étape", le programme est suivi dans son déroulement logique. Vous pouvez exécuter l'algorithme instruction par instruction et observer la modification des variables dans une fenêtre de résultats.

La "visualisation par étape" est possible ou non en fonction de la CPU utilisée. Celle-ci doit prendre en charge l'utilisation de points d'arrêt. La CPU utilisée dans ce document ne prend pas en charge les points d'arrêt.

III.6.4 Avantages offerts par SCL

Les avantages de programmation SCL sont :

- **Langage de programmation évolué :**

SCL propose non seulement tous les avantages d'un langage de programmation évolué, mais également des propriétés qui ont tout particulièrement été conçues pour permettre une programmation structurée.

- **Conservation de la structure de bloc :**

SCL permet de résoudre de manière optimale les diverses tâches qui composent un projet d'automatisation, si bien qu'il peut être mis en œuvre efficacement avec TIA dans chaque phase du

projet. En appliquant en particulier le concept de blocs de TIA, qui permet créer le bloc sous TIA comme dans LIST, LOG et CONT.

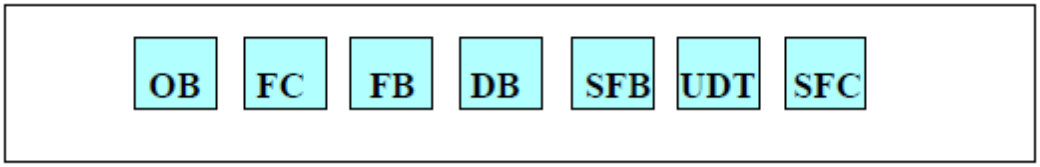


Figure III.11 Les blocs créés avec SCL

- **Types de blocs**

De par leur fonction, leur structure ou leur utilisation, les blocs de STEP 7 sont des sections limitées d'un programme utilisateur. SCL nous permet de créer les blocs suivants :

Type de bloc	Fonction
Bloc d'organisation	Interface entre le système d'exploitation et le programme utilisateur.
Fonction	Bloc permettant la transmission de paramètres sans mémoire.
Bloc fonctionnel	Bloc permettant la transmission de paramètres et possédant une mémoire.
Bloc de données	Bloc permettant d'enregistrer des données utilisateur.
Type de données utilisateur	Bloc permettant d'enregistrer un type de données utilisateur.

Tab.III.4 : Les types de bloc supporté par SCL

- **Compatibilité des blocs :**

Les blocs programmés dans SCL sont compatibles avec les blocs LIST, CONT et LOG. Ceci signifie qu'un bloc programmé dans SCL peut appeler un bloc programmé dans LIST, CONT ou LOG et inversement [25].

III.7 La commande à base de la logique floue :

III.7.1 La logique floue :

La logique floue repose sur la théorie des ensembles flous développée par Zadeh en 1965. A côté d'un formalisme mathématique fort développé, nous préférons aborder la présentation de manière intuitive.

Selon la logique floue, le raisonnement exacte est un cas limite du raisonnement approximatif , tout n'est qu'un degré. Tout système logique peut ˆêtre rendu flou. Les connaissances sont interprétées comme une collection de contraintes élastiques ou floues d'un ensemble de variables. L'inférence est un processus de propagation de contraintes élastiques.

La logique booléenne est un sous-ensemble de la logique floue.

La logique floue permet d'accommoder le concept de vérité partielle : des valeurs entre complétement vrai et complétement faux sont admises. On supporte des modes de raisonnement approximatifs plutôt qu'exacts. Son importance provient du fait que le raisonnement humain est approximatif.

La définition de Zadeh :

Fuzzy Logic is determined as a set of mathematical principles for knowledge

Representation based on degrees of membership rather than on crisp membership

Of classical binary logic.

En revanche, dans le domaine de la commande, la première application de la logique floue a été réalisée par Mamdani en 1974. Depuis 1987, date la première application utilisant la logique floue, le métro de Sendai, l'utilisation des techniques de commande floue n'a cessé de s'accroître [12].

III.7.2 Ensembles flous

En logique floue, un ensemble flou contient plusieurs valeurs. L'ensemble floue est concerné par un degré d'appartenance (ou degré de vérité). On utilise un continuum de valeurs logiques entre 0 (complétement faux) et 1 (complétement vrai). Une fonction d'appartenance est utiliser pour mapper un item X dans le domaine des nombres réels à un intervalle de 0 à 1, ce qui permet un degré de vérité.

L'appartenance à un ensemble représenté une valeur entre 0 et 1. Un ensemble flou peut ˆêtre défini comme un ensemble ayant des frontières floues. Un ensemble flou est définit comme suit : soit *S* un ensemble et *x* un membre de cet ensemble. Un sous-ensemble flou *F* de *S* est définit par une fonction d'appartenance _*F*(*x*) qui mesure le degré auquel *x* appartient à *F* [26].

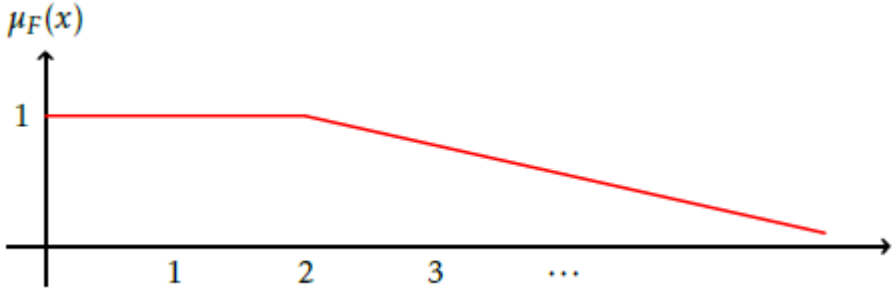


Figure III.12 Représentation du sous-ensemble flou F des petits entiers [26].

est définit comme une fonction : $\mu_A(x) : X \longrightarrow 0;1$

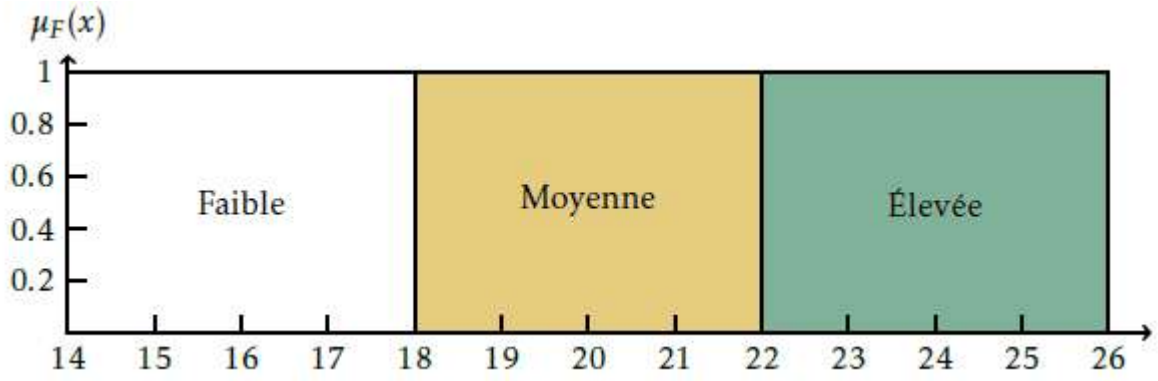
Où $\mu_A = 1$ si *x* est totalement dans *A*, $\mu_A(x) = 0$ si *x* n'est pas dans *A* et $0 < \mu_A(x) < 1$ si *x* est partiellement dans *A*. La fonction d'appartenance est une mesure :

- Du degré auquel un élément est membre d'un ensemble.
- Du degré d'appartenance.
- De la valeur de l'appartenance.
- Du degré de confiance.

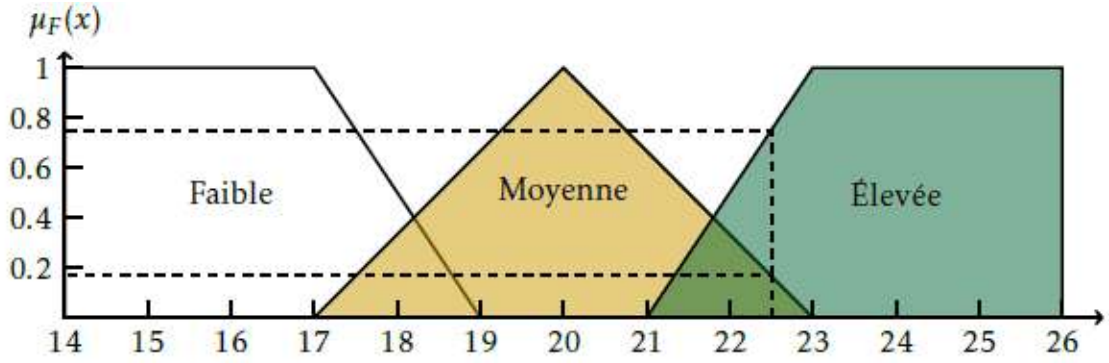
La logique floue permet de transformer plusieurs valeurs réelles en quelques variables floues avec différentes appartenances, ce qui permet de réduire le nombre de règles. On utilise ces règles pour faire la commande d'un système. Le tableau **TabIII.5** compare des règles de commande classiques à la logique floue pour un système de moteur, où *III* est la vitesse [26].

Logique Classique	Logique Floue
Si *temp*= 21°ALORS *III*= 30%	Si *temp*= élevé ALORS *III*= lent
Si *temp*= 22°ALORS *III*= 40%	Si *temp*= assez élevé ALORS *III*= rapide
Si *temp*= 23°ALORS *III*= 50%	Si *temp*= très élevé ALORS *III* = très rapide
Si *temp*= 24°ALORS *III*= 60%	
Si *temp*= 25°ALORS *III*= 75%	
Si *temp*>26°ALORS *III*= 100%	

Tab III.5 Comparaison entre la logique classique et la logique floue.



Représentation Classique



Représentation Floue

Figure III.13 Comparaison de l'appartenance de la température en logique classique vs la logique floue [26].

Selon la **Figure III.13**, en logique classique, une température de 22.5_ est considérée comme ´élevée. En logique floue, une température de 22.5°appartient au groupe "moyenne" avec un degré d'appartenance de 0.167, et appartient au groupe "´élevée" avec un degré d'appartenance de 0.75. Les variables floues ***faible, moyenne*** et ***élevée*** sont représentées par des fonctions linéaires. D'autres fonction auraient pu être utilisées, comme des trapézoïdes, des paraboles, etc.

Cependant, les fonctions linéaires sont beaucoup plus faciles `à implémenter de façon pratique, et donnent de bons résultats [26].

On utilise souvent une notation vectorielle pour représenter les fonctions. Pour les fonctions d'appartenance de la **figure III.13,**on peut utiliser la notation suivante :

- Température faible : (1/17, 0/19)
- Température moyenne : (0/17, 1/20, 0/23)
- Température élevée : (0/21, 1/23)

III.7.3 Haies

Les haies sont des modificateurs de valeurs floues et permettent la génération de déclaration floues à l'aide de calculs mathématiques. Elles modifient la forme des ensembles flous. Elles ont le même rôle que des adverbes et adjectifs en français. Selon leur impact sur la fonction d'appartenance, on les classifie selon leur effet [26]:

- Concentration : aide à intensifier un ensemble. Exemple : "très" crée une concentration et crée un nouveau sous-ensemble.
- Dilatation : agrandi l'ensemble. Exemple : "plus" ou "moins" rend l'ensemble plus grand que l'original.
- Contraste : change la nature de l'ensemble en l'intensifiant ou en l'agrandissant.

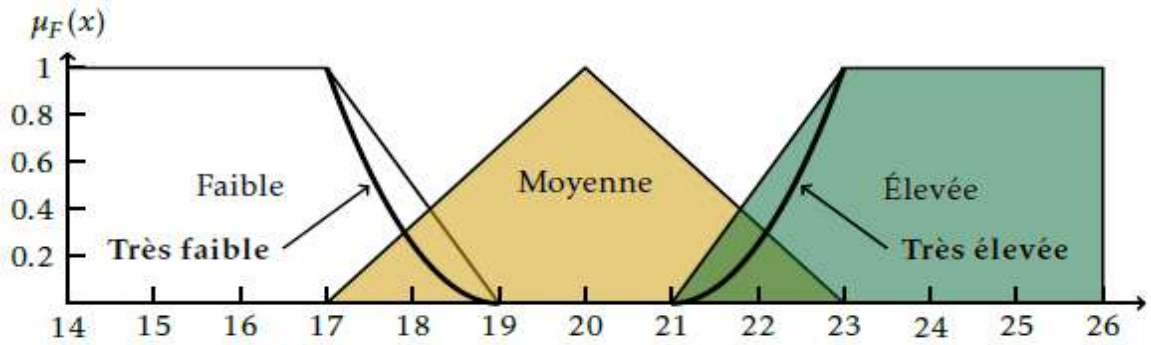


Figure III.14 Exemple de haies [26].

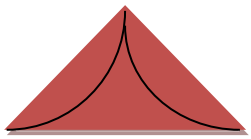
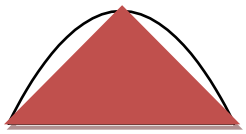
Haie	Représentation mathématique	Représentation graphique
Très	$[\mu_A(x)]^2$	
Plusoumoins	$\sqrt{\mu A(x}$	

Figure III.15 Exemples de haies et représentation mathématique

III.7.4 Opérateurs flous :

Une fois les ensembles flous définis, des mathématiques concernant ce type d'ensemble ont été développées. Les méthaniques élaborées ressemblent beaucoup à celles reliées à la théorie des ensembles conventionnels. Les opérateurs d'union, d'intersection et de négation existent pour les deux types d'ensemble. Les opérateurs habituels, soit l'addition, la soustraction, la division et la multiplication de deux ou plusieurs ensembles flous existent aussi. Toutefois, ce sont les deux opérateurs d'union et d'intersection qu'on utilise le plus souvent dans la commande par la logique floue.

Le complément permet de vérifier de combien un élément n'appartient pas à un ensemble.

Comme exemple, si on a l'ensemble des températures élevées, le complément est l'ensemble des températures qui ne sont pas élevées. Si ***A***est l'ensemble flou, son complément $\neg A$ est :

$$\mu_{\neg A}(x)=1-\mu_A(x) \qquad \text{(III-3)}$$

L'intersection de deux ensembles, en logique flou, est un peu différente des méthodes classiques. On cherche à savoir de combien un élément est-il dans les deux ensembles. On utilise alors la valeur minimale d'appartenance pour calculer l'intersection.

$$\mu_{A\cap B}(x) = MIN[\mu_A(x),\mu_B(x)] \qquad \text{(III-4)}$$

En logique floue, l'union est le contraire de l'intersection. On cherche à savoir de combien un est-il dans l'un des deux ensembles. On utilise la valeur maximale d'appartenance.

$$\mu_{A\cup B}(x)=MAX[\mu_A(x),\mu_B(x)] \qquad \text{(III-5)}$$

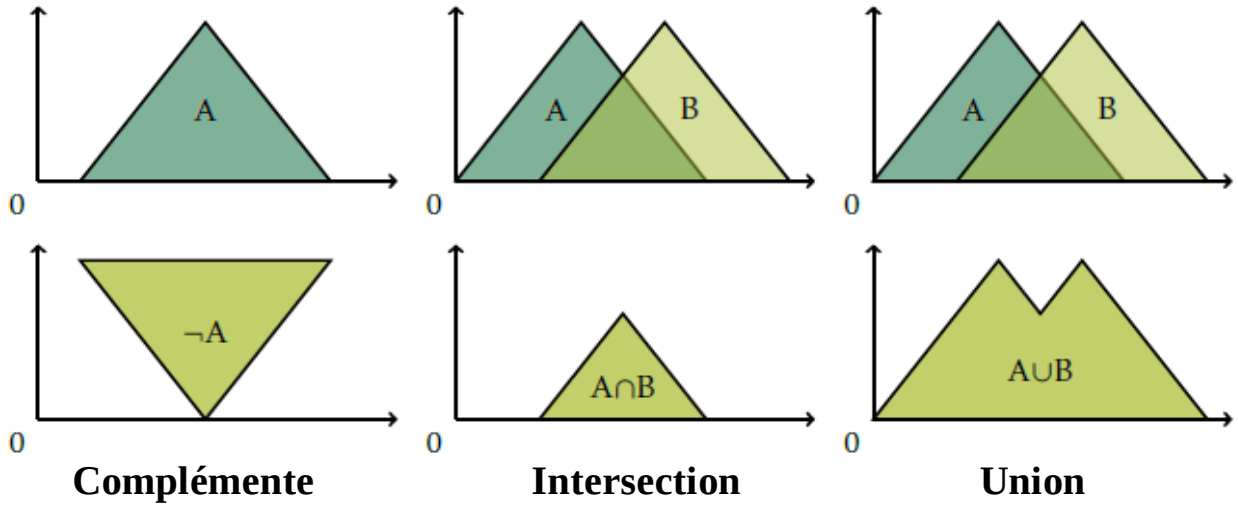


Figure III.16 Exemple d'opérations sur des ensembles flous de façon graphique [26].

Dénomination	Intersection ET :	Réunion OU :	Complément NON :
Operateurs de Zadeh MIN/MAX	$\mu_{A\cap B}(x)$ $=MIN[\mu_A(x),\mu_B(x)]$	$\mu_{A\cup B}(x)=MAX[\mu_A(x),\mu_B(x)]$	$\mu_{\neg A}(x)=1-\mu_A(x)$
Probabiliste PROD/PROBOR	$\mu_A(x) * \mu_B(x)$	$\mu_A(x)+\mu_B(x) - \mu_A(x)*\mu_B(x)$	$\mu_{\neg A}(x)=1-\mu_A(x)$

Tab III.6 Opérations sur des ensembles flous de façon mathématique.

III.8 Le raisonnement en logique floue :

En logique classique, les raisonnements sont de la forme :

$$\begin{cases} \text{Si p alors q} \\ \text{p vrai alors q vrai} \end{cases}$$

En logique floue, le raisonnement ou également appelé raisonnement approximatif,
Se base sur des règles floues qui sont exprimées en langage naturel en utilisant les
Variables linguistiques dont nous avons donné la définition précédemment. Une règle floue aura cette forme :

Si A et y ∈ B alors z ∈ C, avec A, B et C des ensembles flous [27].

III.8.1 Règles Floues

Une règle floue est une déclaration de la forme suivante :

SI x est A ALORS y est B

Où x et y sont des variables linguistiques, et A et B sont des valeurs linguistiques, déterminées par les ensembles flous sur les ensembles X et Y. Une variable linguistique est une variable floue. Par exemple : La tension est haute. La variable linguistique *tension* prend la valeur linguistique *élevée*. La plage de valeurs linguistiques possibles d'une règle représente l'univers de cette variable. Un exemple de règle floue est :

SI *vitesse* est *lente* ALORS *arrêt* est *court*

La variable *vitesse* peut avoir une plage de valeurs entre 0 et 220 km/h. On peut inclure des sous-ensembles flous (très lent, lent, moyenne, rapide, très rapide) pour modifier cette règle. Chaque sous-ensemble flou représente une valeur linguistique pour la variable.
La logique classique (SI – ALORS) utilise la logique binaire. La logique floue permet d'associer une plage de valeurs (un ensemble flou) `a des variables linguistiques. On peut réduire le nombre de règles jusqu'à 90% en utilisant la logique floue [26].

III.9 Commande par logique floue :

La commande floue est certainement le domaine d'application de la logique floue le plus utilisé. Son but est, comme en automatique classique de gérer un processus, selon une consigne désirée, par action sur des grandeurs physiques. Sa particularité est de reproduire le comportement d'un opérateur humain, plutôt que de réaliser un modèle mathématique du système [28]

III.9.1 Constitution d'un système floue :

Un système flou est composé de quatre parties essentielles :

- La base de connaissance constituée d'une base de données et d'une base des règles.
- Le système d'inférence.
- L'interface de fuzzification.
- L'interface de déffuzzification

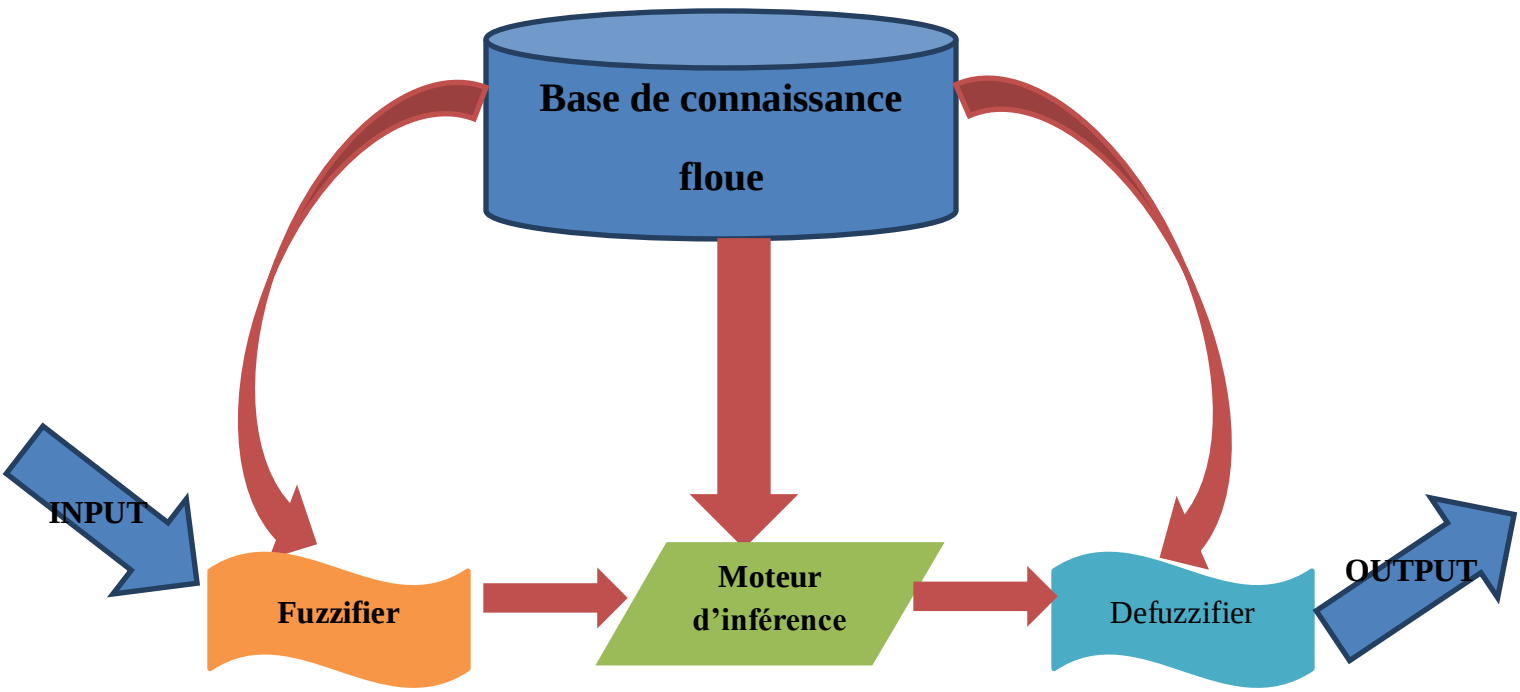


Figure III.14 Aperçu synoptique d'un système flou

III.9.1.2 Interface de fuzzification

La fuzzification est l'opération de projection des variables physiques réelles sur des ensembles flous caractérisant les valeurs linguistiques prises par ces variables. Le bloc de fuzzification effectue les fonctions suivantes:

- établit les plages de valeurs pour les fonctions d'appartenance à partir des valeurs des variables d'entrées.
- effectue une fonction de fuzzification qui convertit les données d'entrée en valeurs linguistiques convenables qui peuvent être considérées comme l'étiquette des ensembles flous. Cette opération doit être effectuée dans un domaine normalisé généralement par l'intervalle [-1,1] afin de faciliter le calcul.

III.9.1.3 Degré d'appartenance :

Les degrés d'appartenance des du contrôleur sont également déterminés pour les groupes flous, car chaque entré a deux degrés d'appartenance.

Les relations mathématiques (1 et2) et la figure X montrent comment les degrés d'appartenance sont calculés.

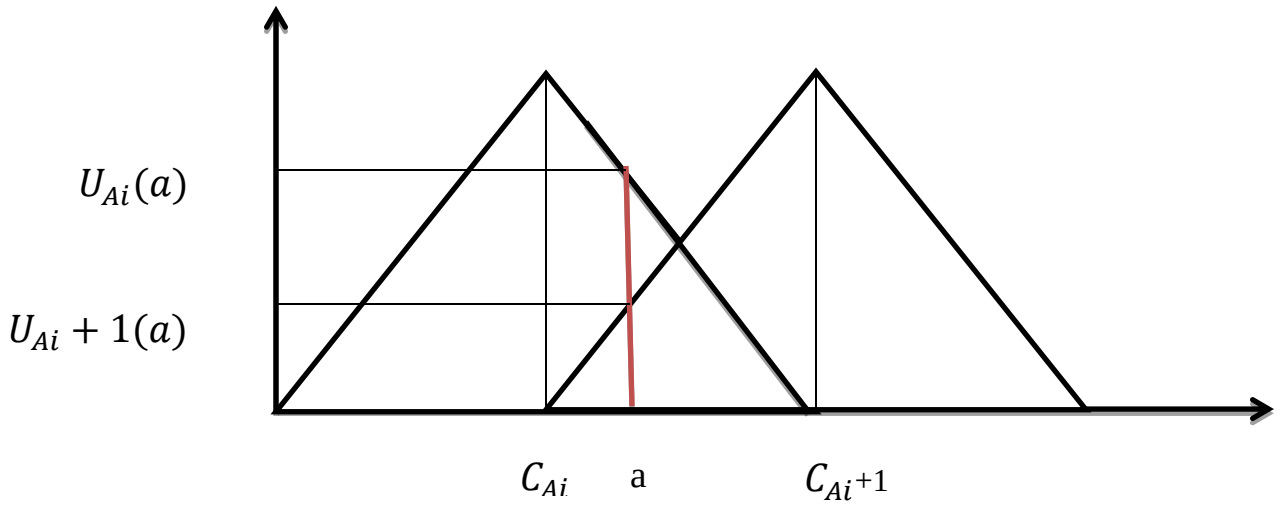


Figure III.18 Calcul du point d'appartenance graphiquement

$$U_{Ai}(a) = \frac{C_{Ai+1} - a}{C_{Ai+1} - C_{Ai}} \quad \text{(III-6)}$$

$$U_{Ai+1}(a) = \frac{a - C_{Ai}}{C_{Ai+1} - C_{Ai}} \quad \text{(III-7)}$$

III.9.1.4 Mécanisme d'inférence floue

Le principe de cette opération consiste à établir la liaison entre les grandeurs d'entrée du régulateur exprimés par des termes linguistiques et la variable de sortie sous sa forme linguistique à partir d'une parfaite connaissance du système à régler et d'une expertise concernant le fonctionnement du processus à régler. Les trois méthodes d'inférence les plus usuelles sont [29] :

- Max-Produit.
- Somme-produit.
- Max-Min (Implication de Mamdani).

III.9.1.5 Interface de défuzzification

Comme nous avons vu dans la section précédente, les méthodes d'inférence fournissent un résultat qui est une fonction d'appartenance. Or, la sortie du contrôleur est en général une grandeur continue, prenant sa valeur dans un intervalle. La défuzzification est le traitement qui permet de définir une correspondance entre le résultat de l'inférence et la grandeur continue fournit en sortie. Plusieurs stratégies de défuzzification ont été développées.

III.9.1.6 Méthode du centre de gravité

La méthode du centre de gravité c'est la méthode de Défuzzification la plus utilisée, son principe repose sur la détermination de l'abscisse du centre de gravité x* de la fonction d'appartenance résultante de l'inférence$\mu_A(x)$. Cette abscisse correspond à la valeur de sortie du régulateur.

III.9.1.7 Méthode du maximum

Comme son nom l'indique, la commande en sortie est égale à la commande ayant la fonction d'appartenance maximale. La méthode du maximum simple, rapide et facile, mais elle introduit des ambiguïtés et une discontinuité de la sortie (parfois on trouve deux valeurs maximales).

III.9.1.8 Méthode de la moyenne du maximum

Elle considère, comme valeur de sortie, les moyennes de toutes les valeurs pour lesquelles la fonction d'appartenance issue de l'inférence est maximale.

III.9.2 Inférence mamdani

L'inférence Mamdani procède selon quatre étapes :

1. Fuzzification des variables d'entrée
2. Evaluation des règles
3. Agrégations des sorties des règles
4. Défuzzification

Si on utilise la méthode de coupure pour combiner les règles, il faut créer un nouveau polygone à partir des trois fonctions d'appartenance de la conséquence C. La hauteur du polygone est déterminée à partir de la valeur d'appartenance calculée plus haut.

L'agrégation pour cet exemple est montrée à la figure III.13

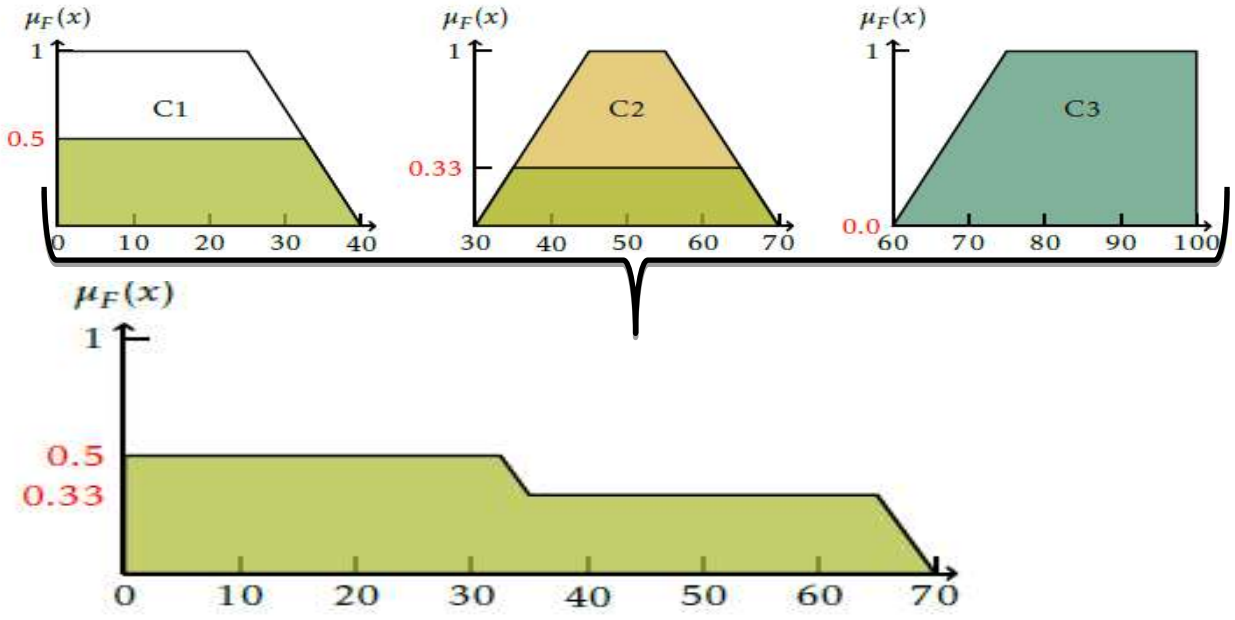


Figure III.19 Agrégations des règles de sortie par coupure [26].

Après avoir combiné les règles, il faut produire un chiffre net comme sortie.
La technique la plus populaire est la méthode du Centroide : on cherche le centre de gravité du polygone obtenu :

$$CG = \frac{\sum_{x=a}^{b} \mu_A(x).x}{\sum_{x=a}^{b} \mu_A(x)} \qquad \text{(III-8)}$$

III.9.3 Inférence Sugeno :

La méthode de Sugeno permet de simplifier le calcul de l'agrégation, afin d'obtenir plus rapidement une solution nette. Cette méthode est souvent utilisée pour des applications à temps réel, où le temps de calcul est important.
La méthode de Sugeno utilise un seul pic comme fonction d'appartenance, plutôt qu'un polygone. Ce pic est un ensemble flou ayant une fonction d'appartenance 1 à un point particulier de l'espace, et 0 ailleurs. Le résultat de l'évaluation des règles devient l'amplitude du pic. La figure III.20 montre un exemple de fonction de style Sugeno pour le même exemple.
Les conséquences sont évaluées selon le résultat de l'évaluation des règles. Dans l'exemple précèdent, on a obtenu une appartenance de 0.5 pour la règle 1, de 0.33 pour la règle 2, et 0 pour la règle 3. Ces trois valeurs deviennent alors l'amplitude des pics, comme à la figure III.21.

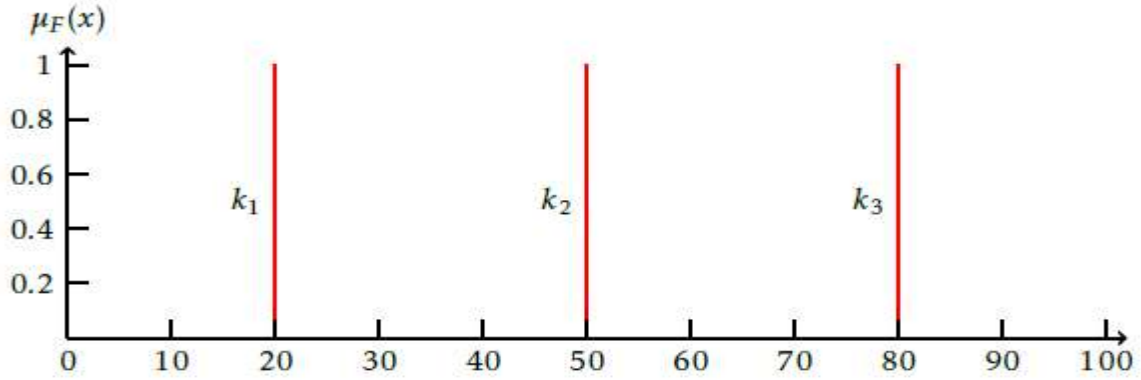


Figure III.20 Conséquences de style Sugeno

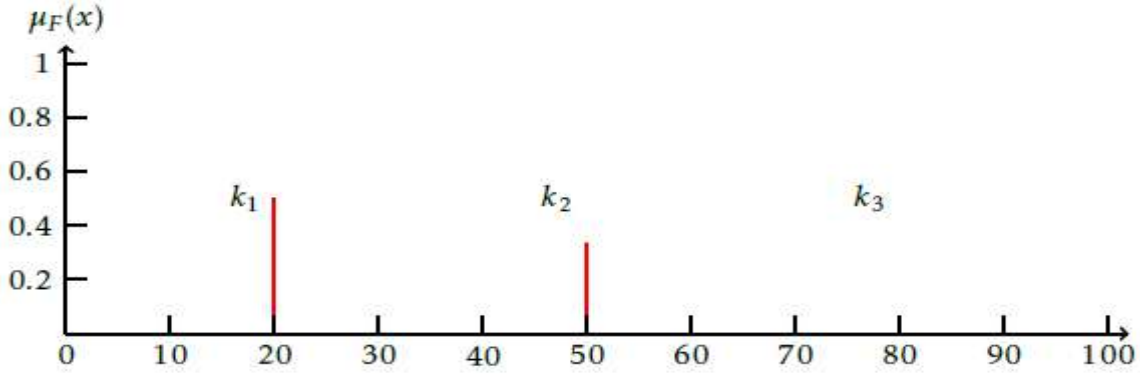


Figure III.21 Evaluation des conséquences de style Sugeno

Pour faire l'agrégation des règles, on utilise une moyenne pondérée

$$CG_{sugeno} = \frac{\sum_i \mu_A(Ki).Ki}{\sum_i \mu_A(Ki)} \qquad \text{(III-9)}$$

III.10 La programmation floue sous TIA PORTAL

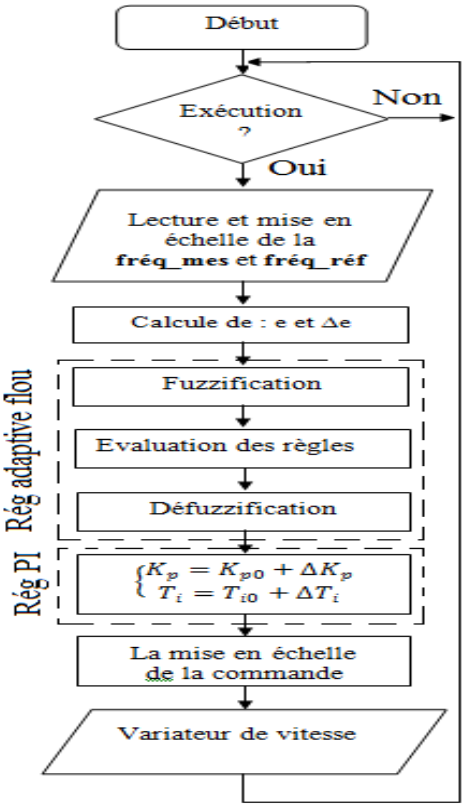


Figure III.22 Organigramme générale du programme sur TIA PORTAL

L'organigramme ci-dessous représente notre programme sur SCL TIA PORTAL en raisonnement floue, Où :

µe et µΔe : degré d'appartenance de l'erreur ***e*** et la variation de l'erreur Δ***e***

re = [-3,-2, -1, 0, 1, 2, 3] : Abscisses de l'intervalle discours de l'erreur ***e***

rΔe = [-3,-2, -1, 0, 1, 2, 3] : Abscisses de l'intervalle discours de la variation de l'erreur Δ***e***

rkp (l, k) : règle qui se trouve à la ligne ***l*** et colonne ***k*** du Tab II.1.1 (table d'inférence Kp)

rΔkp (l, k) : règle qui se trouve à la ligne ***l*** et colonne ***k*** du Tab II.1.2 (table d'inférence Ki)

m = [m(1) m(2) m(3) m(4)] : vecteur des minimaux.

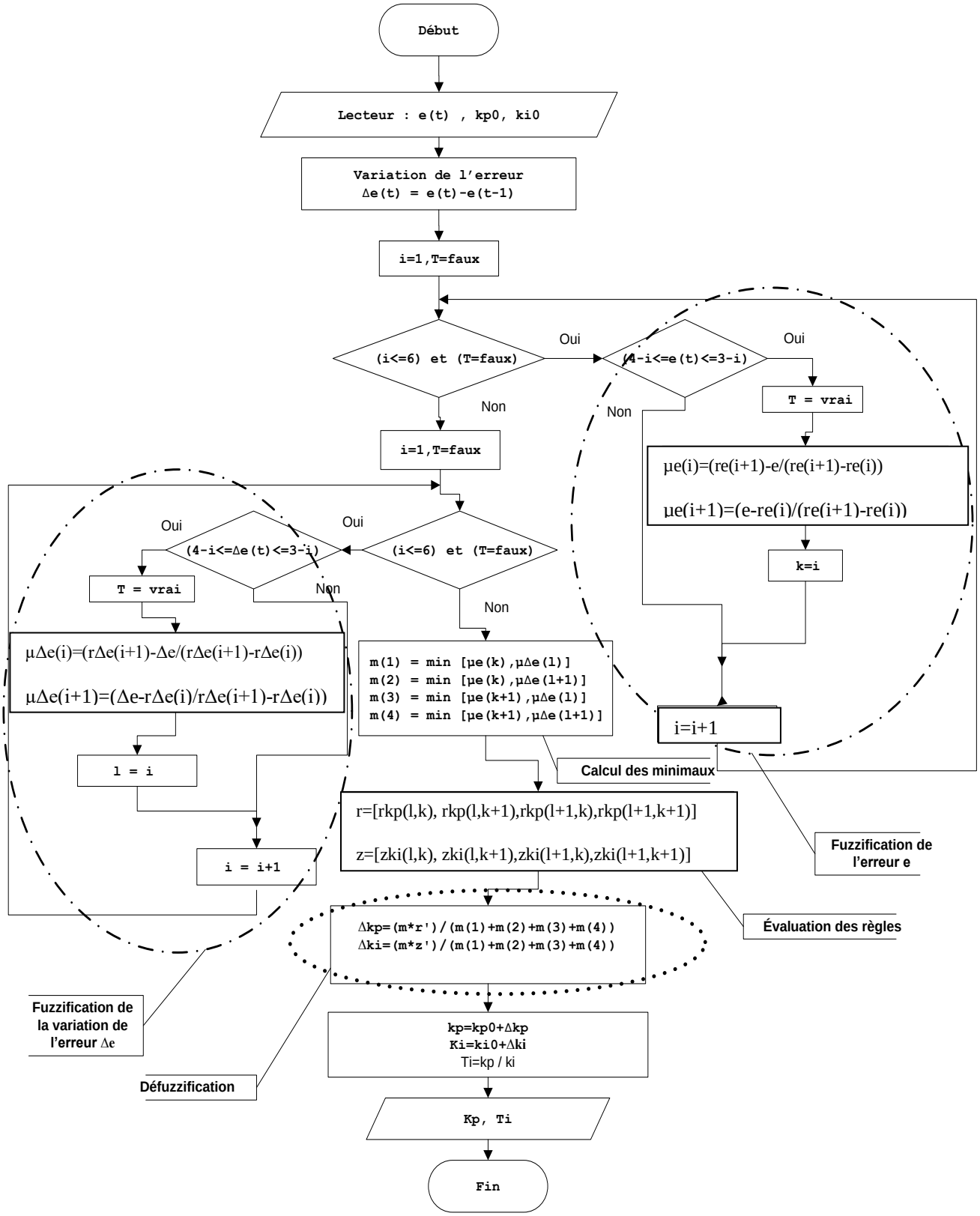


Figure III.23: Organigramme d'un régulateur floue sous SCL-Tia-Portal

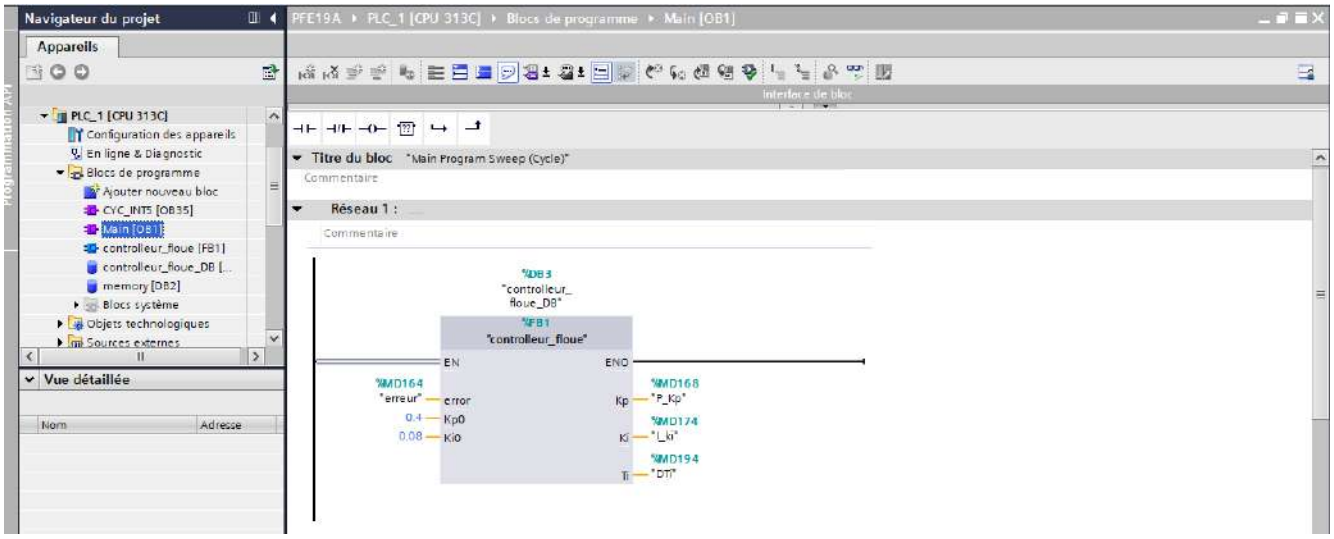


Figure III.24 le bloc du contrôleur floue

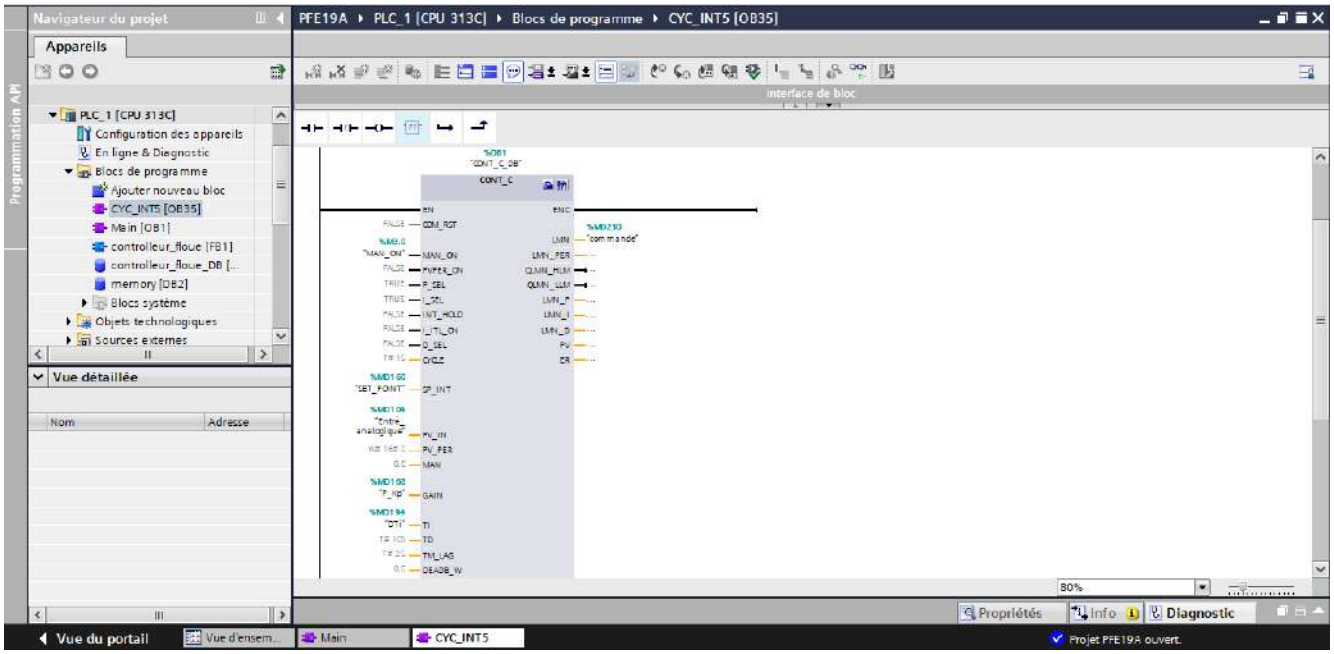


Figure III.25 le bloc FB41

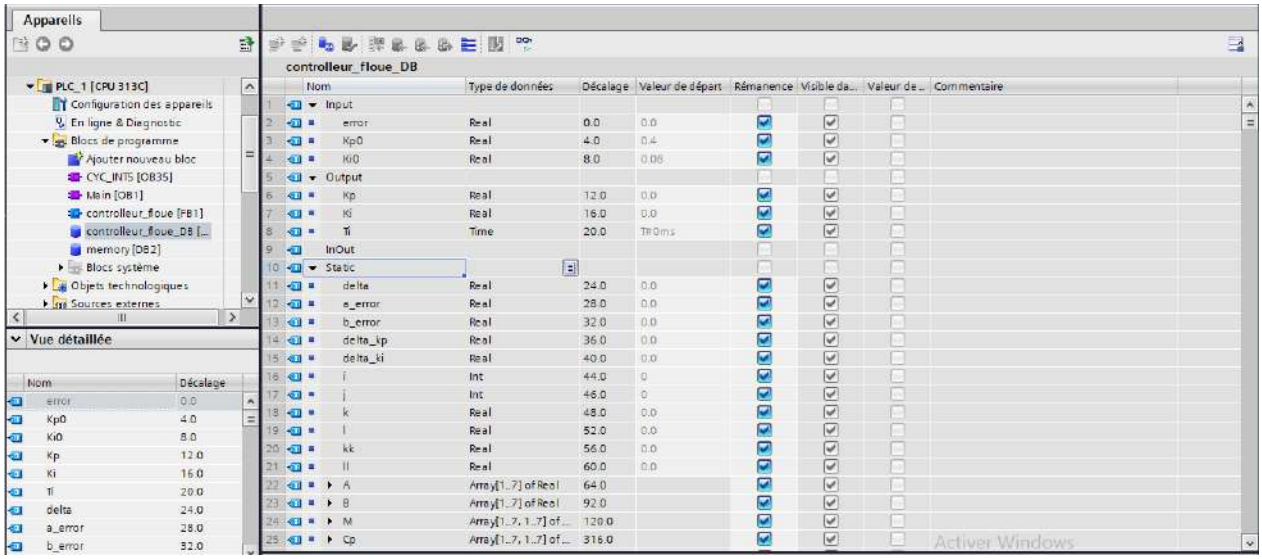
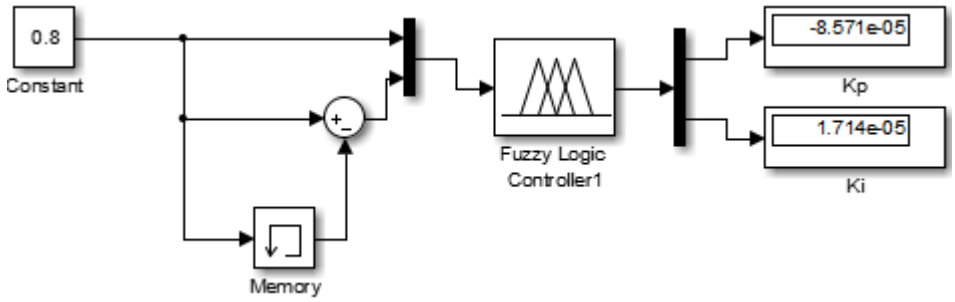


Figure III.26 Data bloc du programme

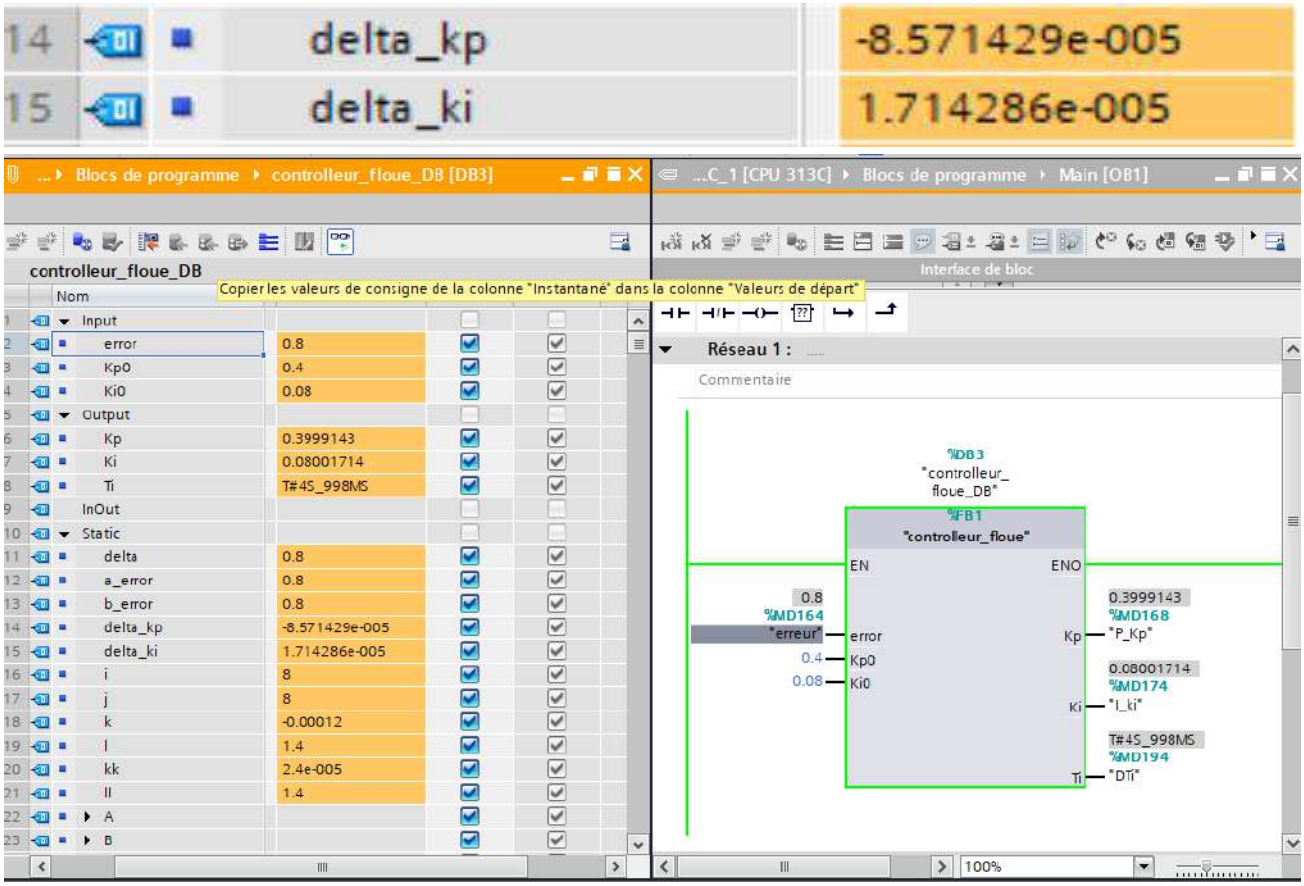
III.11 Comparaison du résultats du simulation :

On va discuter l'influence de la variation du Kp et Ki pour faire une comparaison entre la commande PI-Floue sous **SIMULINK MATLAB** et la programmation du raisonnement floue sous **TIA PORTAL** V13 avec langage SCL, sur les réponses numériques du ΔKp et ΔKi.

- **Le premier Test ; erreur = 0.8**



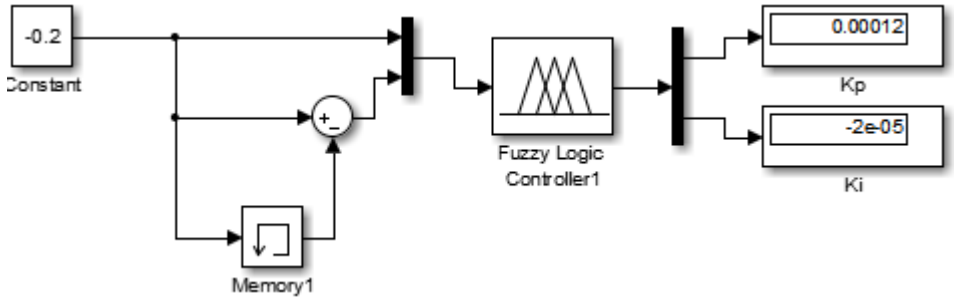
Résultat sur SIMULINK MATLAB



Résultat sur TIA PORTAL

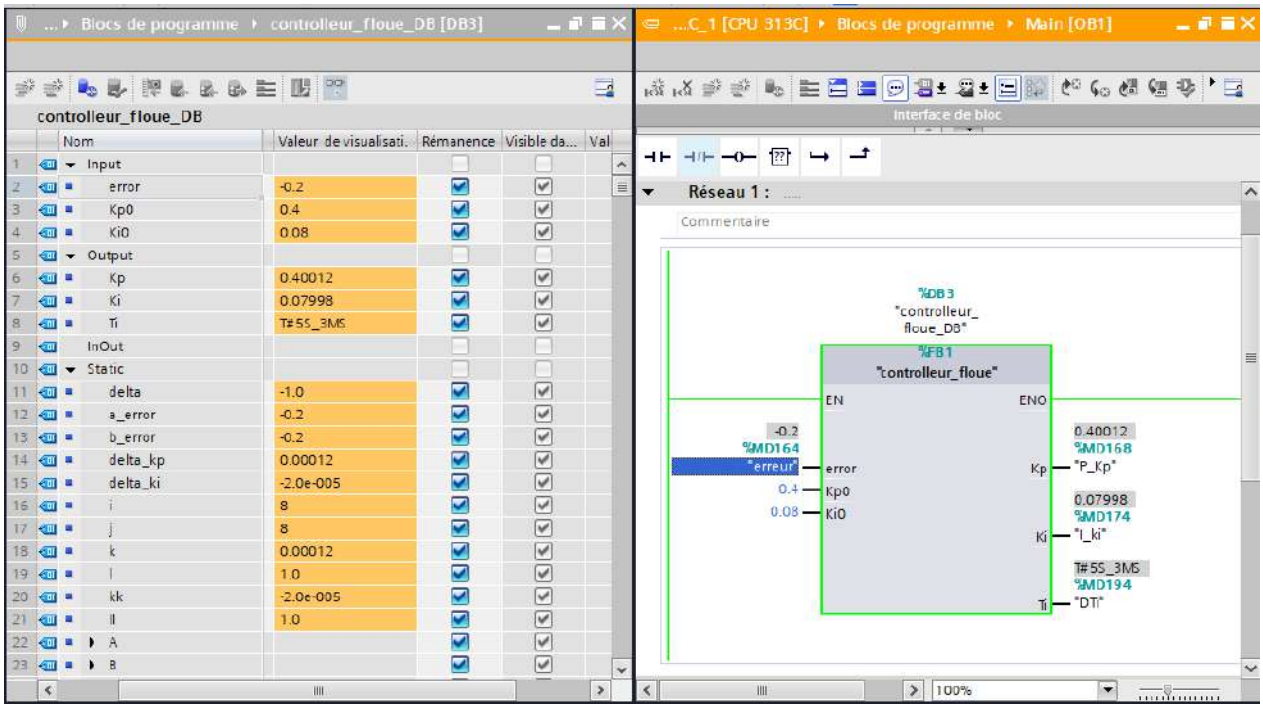
(a) Résultats du 1er test

- **Le Deuxième test ; erreur = -0.2**



Résultat sur SIMULINK MATLAB

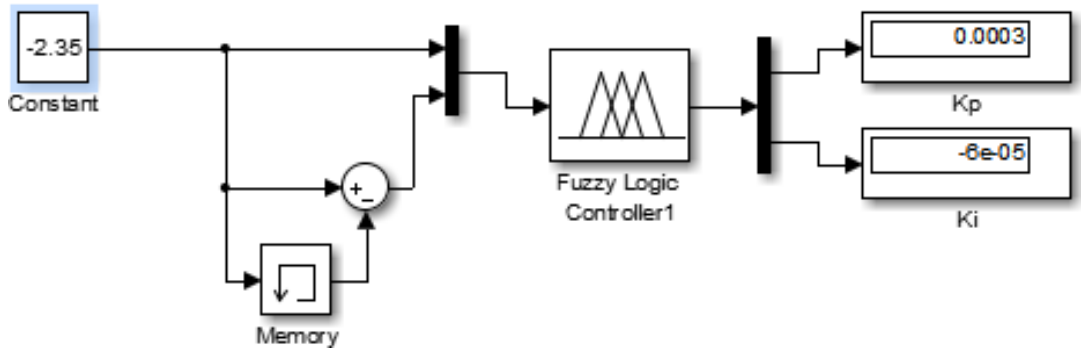




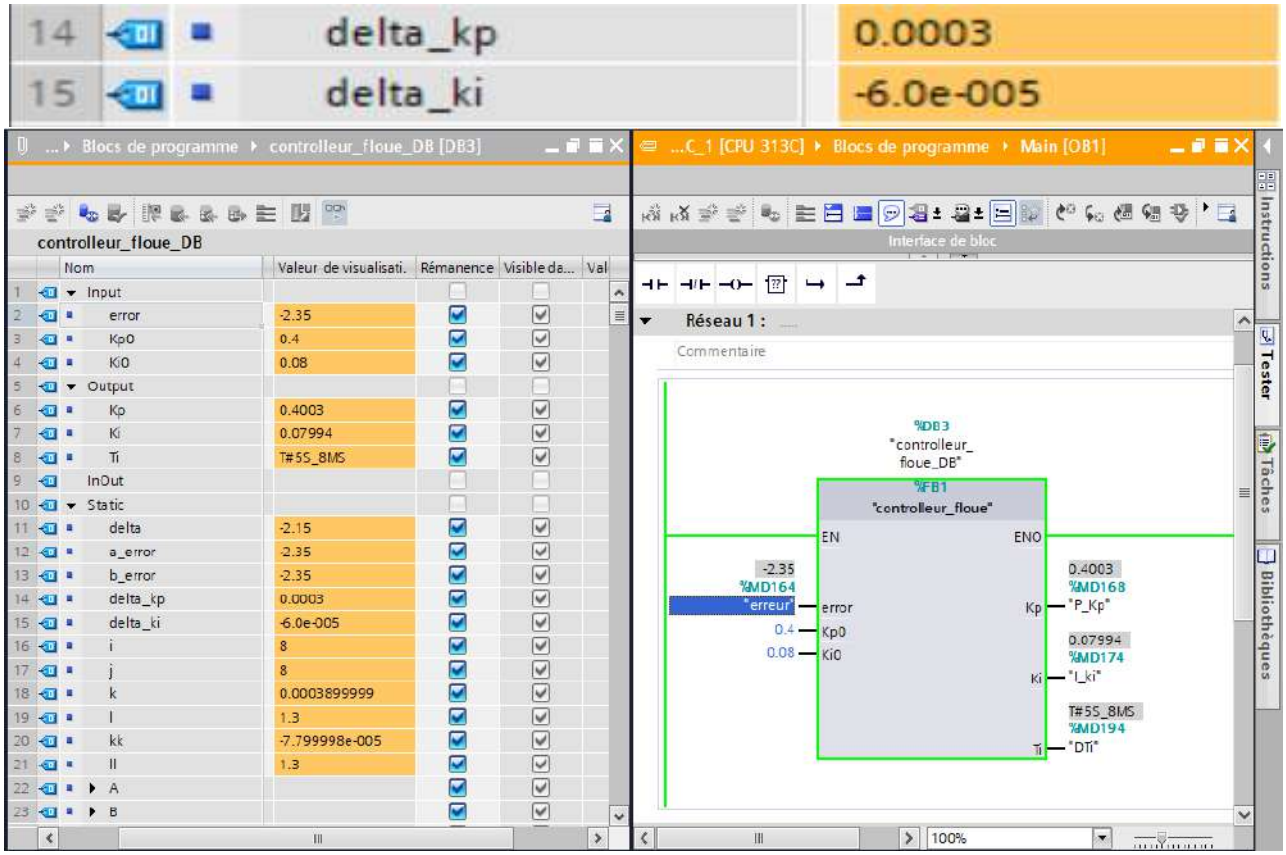
Résultat sur TIA PORTAL

(b) Résultats du 2éme test

- **Le troisième test erreur = -2.35**



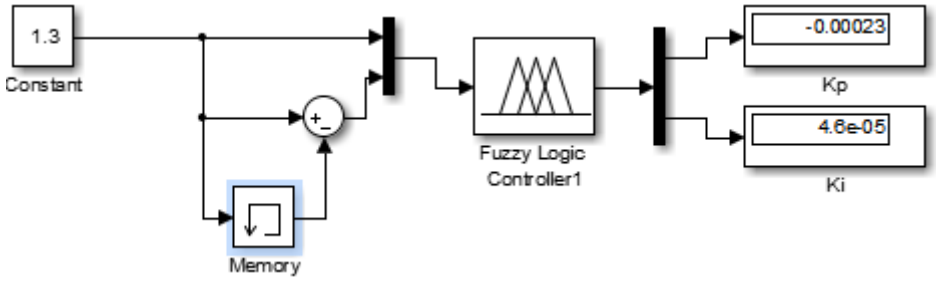
Résultat sur SIMULINK MATLAB



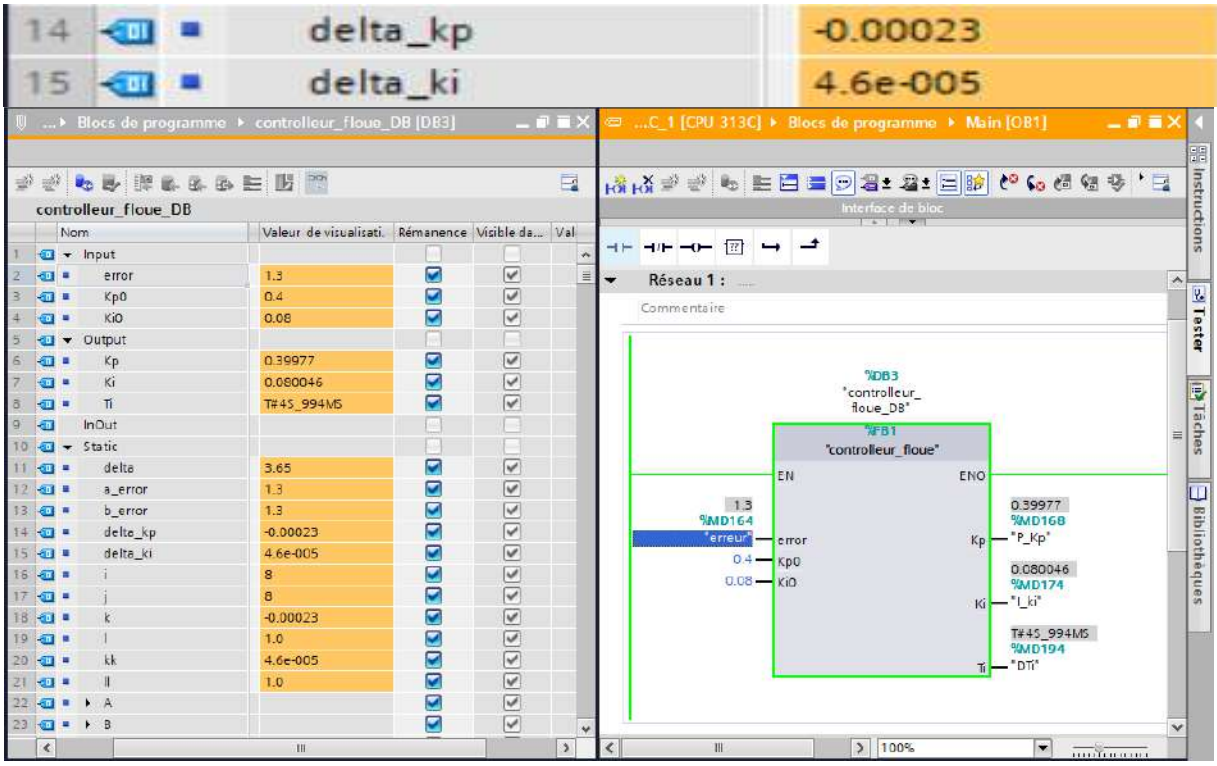
Résultat sur TIA PORTAL

(c) Résultats du 3éme test

- **Le quatrième test ; erreur =1.3**



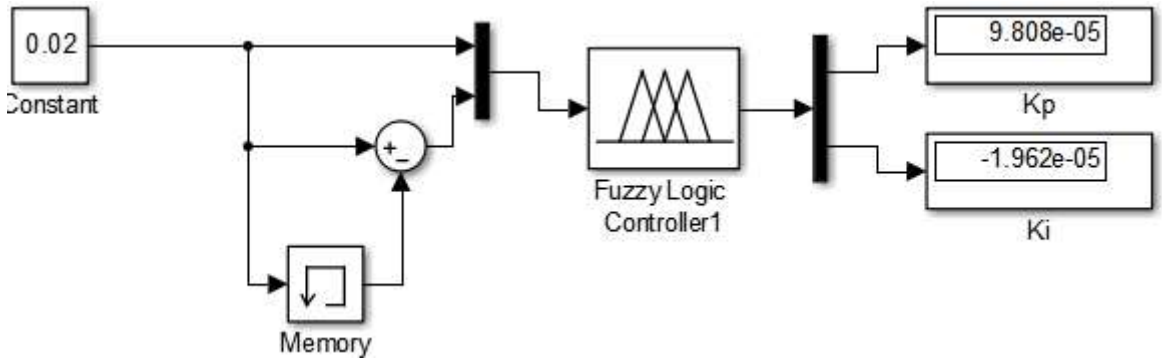
Résultat sur SIMULINK MATLAB



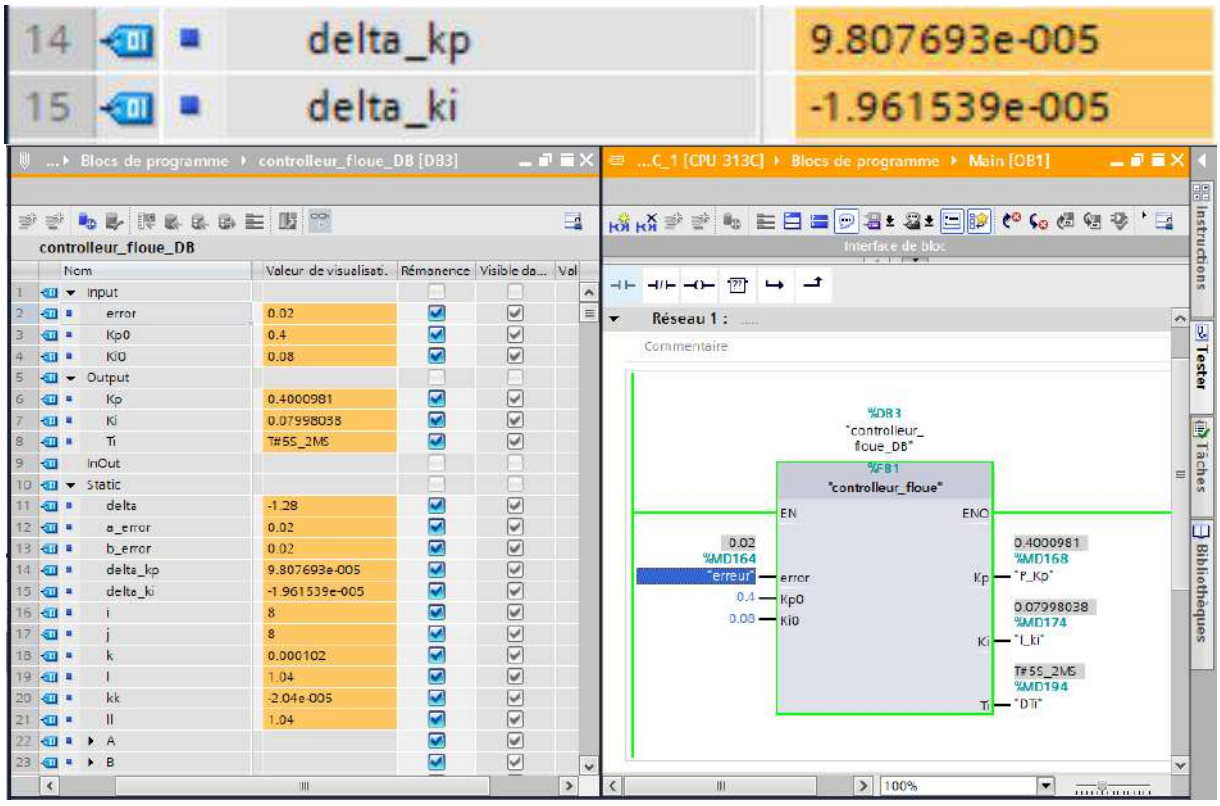
Résultat sur TIA PORTAL

(d) résultat du 4éme test

- **Le cinquième test ; erreur = 0.02**



Résultat sur SIMULINK MATLAB



Résultat sur TIA PORTAL

(e) Résultat du 5éme test

Figure III.27 les résultats du comparaison entre SCL TIA POTAL et SIMULINK MATLAB

III.11 Conclusion

On peut conclure que les résultats du l'algorithme du raisonnement flou sur TIA-PORTAL sont validés par SIMULINK MATLAB.

Cette méthode sur TIA PORTAL sera donc celle qui sera améliorée par les techniques modernes afin de garder les avantages et supprimer les inconvénients.

L'algorithme du raisonnement flou TIA PORTAL est caractérisée principalement par unerésultat bien précis

Conclusion générale :

Ce mémoire étudie la modélisation de la machine asynchrone avec l'onduleur et leurs simulations en SIMULINK MATLAB.
Puis on a fait la commande scalaire de la MAS avec une régulation classique par régulateur PI. Ensuite, nous avons implémenté la logique floue pour optimiser les paramètres du régulateur PI.
En fin on a programmé un algorithme d'un régulateur flou en TIA-PORTAL par langage SCL et validé les valeurs du ΔKp et ΔKi qu'on a obtenue sur SIMULINK MATLAB ce que est confirmé notre algorithme d'un régulateur flou sur TIA PORTAL.

Ce travail nous a permis d'acquérir une expérience enrichissante dans le domaine de la régulation des moteurs électriques, et leur logiciels MATLAB et TIA PORTAL de SIEMENS, sur le plan plus personnel, on a pu de découvrir le travail d'équipe (collectif) et de s'intégrer dans un milieu professionnel.
Finalement, les connaissances théoriques apprises durant notre cursus étant complétées dans ce travail notamment que la communication et le respect d'un planning préalablement établi restent les éléments essentiels de la réussite d'un projet.

Bibliographie

[1] : Lallouani Hellali, Saad Belhamdi "Direct Torque Control of Doubly Star Induction Motor Using Fuzzy Logic Speed Controller" IAES International Journal of Artificial Intelligence (IJ-AI) Vol. 7, No. 1, March 2018

[2] : SEDDAR Yahia , GHEZAL Katia 'Contribution à l'Implémentation d'un Régulateur Flou sous un API S7-300 , 03 juillet 2017

[3] : Bernard Schneider et Alain Beuret « Électricité au service des machines », Hute Ecole d'Ingénierie et de Gestion du Canton de Vaud, 2007

[4] : Alain Charbonnel, «Le moteur asynchrone triphasé», 27/11/2010

[5] OTHMANI Ahmed Mehdi , 'COMMANDE VECTORIELLE D'UNE MACHINE ASYNCHRONE ASSOCIÉE A UN OBSERVATEUR ADAPTATIF ' , Année 2019

[6] : Ségier GUY « Electronique de puissance les fonction de bases et leurs principales applications »,DUNOD , Paris, 1999.

[7] L.BAGHLI, Contribution à la commande de la machine asynchrone, utilisation de la logique floue, des réseaux de neurones et des algorithmes génétiques, thèse de doctorat, Université Henry Poincaré, Nancy-I, Janv. 1999.

[8] J.P.CARON, J.P.HAUTIER, Electrotechnique Modélisation et commande de la machine asynchrone, Editions Technip, Paris 1995.

[9] P.J. Mac Vivar Whelen, « Fuzzy Sets Man-Machine Interaction », In.J.Man-Machine Studies, Vol. 8, pp.687-697,1976.

[10] S. Chand, "on-line, Setf-Monitiring Tuner for Proportional Integral Derivative Controllers". IEEE Proceedings of 30th ICEO'1171 Conference on Decision and Control, December, pp. 1905-1906, 1991.

[11] S. Z. He, S. Tan, F. L. Xu, and P. Z. Wang. "fuzzy Self-tuning of PID controllers", fuzzy Sets and Systems, Vol. 56, pp. 37-46, 1993.

[12] Z. Y. Zhao, M. Tomizuka, and S. Isaks, "Fuzzy Gain Scheduling of PID Controller", IEEE Trans, On Systems, Man, and Cybernetics, Vol. 23, No. 5, Sep –Oct, pp. 1392-1398, 1993.

Bibliographie

[13] Yuwana, M. and Seborg D. E., "A new Method for On-line Controller Tuning", AICHE J, Vol. 28,pp. 434-440

[14] E. NELSON et P.WATCHOU , «Contrôle en V/F d'une machine asynchrone », Rapport de stage, l'université de Montpellier, 2015-2016.

[15] E. NELSON et P.WATCHOU , «Contrôle en V/F d'une machine asynchrone », Rapport de stage, l'université de Montpellier, 2015-2016.

[16] BENGUESMIA et Toufik GUERAR Khalil ' IMPLEMENTATION DE LA COMMANDE SCALAIRE POUR UNE MACHINE ASYNCHRONE TRIPHASE', UNIVERSITE MOHAMED BOUDIAF - M'SILA , 2017 /2018

[17] « Introduction à La Logique Floue » Franck Dernoncourt, Avril 2011,Paris.

[18] Jingwei Xing, Zhaolun Liu , Ying yang, Zhaozhao Zheng Yanshan University, Qinhuangdao, Hebei, 066004, China.

[19] : MICHEL. G, Les API, Architecture Et Application Des Automates Programmables. Industriels. Dunod, Paris, [1987].

[20] : Documentations techniques internes. "Manuel d'instructions et d'entretien". CAA.

[21] : ALAIN GONZAGA « Les automates Programmables Industriels ».

[22] : Aide Du Logiciel TIA PORTAL.

[23] : HATRAF LAHCENE MOHAMED AMINE.TOUZALA ABDERRAHMANE « automatisation et supervision d'une station de remplissage des bouteilles ».

[24] : SIMATIC Logiciel de base pour S7-300/400 Régulation PID, manuel Siemens AG, 1996.

[25] : SIMATIC SCL pour S7-300/400 Programmation de blocs, manuel Siemens AG, 1998 consulté le 29/04/2015, p. 1-2,1-3.

[26] : Documentations GELE5313 CHAPITRE 11. Gabriel Cormier.

[27] : Klalib Abdelmalk. Remidi Fateh. Modélisation et simulation d'un système photovoltaïque à base de logique floue. UNIVERSITE YAHIA FARES DE MEDEA 2017-2018

[28] : Mme BELHADJER Iman AMELIORATION DE PERFORMANCES D'UN SYSTEME DE POMPAGE PHOTOVOLTAIQUE PAR LA LOGIQUE FLOUE TYPE-2. UNIVERSITE YAHIA FARES DE MEDEA 2019

Bibliographie

[29] : Zerhouni Mohammed Chérif, Boudjemaa Mourad , Contrôle la température et l'humidité d'une serre agricole à base de la logique floue autour d'un microcontrôleur de type Arduino, Université de Medea , 2018.

[30] : Jingwei Xing, Zhaolun Liu , Ying yang, Zhaozhao Zheng Yanshan , Fuzzy Self-Adaptive PI Control Method for Permanent Magnet Synchronous Motor Drive ,University, Qinhuangdao, Hebei, 066004, China.