REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITE YAHIA FARES DE MEDEA



Faculté de **T**echnologie

Mémoire de fin d'études de Master

Filière: Génie électrique

Spécialité: Automatique

Design and simulation of washing machine using fuzzy logic controller with MatLab

---

Presented by:

Mr. YKHLEF ABDESSEMED ABDELMADJID

Proposed and Directed by:

Dr. BETTACHE FAROUK

Promotion 2020

الملخص

خلال السنوات القليلة الماضية ، تم استخدام أنواع مختلفة من تقنيات التحكم في مختلف مجالات الصناعة. نظام التحكم المبني على المنطق الضبابي هو واحد منهم. يستخدم المنطق الضبابي عبارات بدلاً من النموذج الرياضي لحل مشكلة معينة. في هذا البحث ، تم تصميم الغسالة المنزلية العادية ، والتي يتم استخدامها كثيرًا ، بمساعدة المنطق الضبابي. تم إجراء كل من المحاكاة والتحكم في الجهاز المذكور أعلاه باستخدام مربع أدوات المنطق الضبابي MATLAB.

الكلمات المفتاحية :

الاستشعار التلقائي,التحسين ,MatLab ,الغسالة,المنطق الضبابي

Abstract:

For the past few years, different types of control techniques are being used in various fields of industry. Fuzzy logic based control system is one of them. Fuzzy logic uses statements instead of mathematical model for solving a given problem. In this paper, a normal household washing machine, which is used very often, is modeled with the help of Fuzzy logic. Both the simulation and the control of the aforementioned device have been done by using MATLAB's fuzzy logic toolbox.

Keywords: Fuzzy Logic, Washing Machine, MatLab, Optimization, Automatic Sensing

Résume :

Depuis quelques années, différents types de techniques de contrôle sont utilisés dans divers domaines de l'industrie. Le système de contrôle basé sur la logique floue est l'un d'entre eux. La logique floue utilise des énoncés au lieu d'un modèle mathématique pour résoudre un problème donné. Dans cet article, une machine à laver domestique normale, qui est utilisée très souvent, est modélisée à l'aide de la logique floue. La simulation et le contrôle du dispositif susmentionné ont été réalisés en utilisant la boîte à outils de logique floue de MATLAB.

Les mots clés : Logique floue, machine à laver, MatLab, optimisation, détection automatique

TABLE OF CONTENTS

List Of Figures ..... **I**

List Of Tables ..... **IV**

List Of Acronyms ..... **V**

GENERAL INTRODUCTION **1**

Chapter I : Three Phase Induction Motor **2**

1. Introduction 3
2. Induction Motor History 3
3. Three Phase Induction Motor Construction 4
   - 3.1 The Stator 4
   - 3.2 The Rotor 5
4. Working Principle Of Three Phase Induction Motor 7
5. Symbols of Three Phase Induction Motor 8
6. Synchronous Speed (Ns) 9
7. Slip (S) 9
8. The advantages and disadvantages of the asynchronous motor 10
   - 8.1 Advantage 10
   - 8.2 Disadvantage 11
9. Area of use 11
10. Modeling of the asynchronous machine 11
    - 10.1 Simplifying hypothesis 11
11. Modeling of MAS in the Three-Phase abc plan 12
12. General equations of the three-phase asynchronous machine 13
    - 12.1 Electrical equations 13
    - 12.2 Magnetic equations 14
    - 12.3 Mechanical equations 15
13. Park transformation 16
    - 13.1 Different benchmarks 16
    - 13.2 Modeling of MAS in the two-phase dq plane 17
    - 13.3 Electrical equations 19
      - a- Equations of tensions 19
      - b-Magnetic equation 19
      - c– Mechanical Equation 20
14. The asynchronous machine supplied with voltage 20

15. State representation of the MAS model 20
16. Modeling of the voltage inverter 22
16.1Introduction 22
16.2 Modeling of the voltage inverter 23
16.3 Mathematical model of the voltage inverter 23
16.4 Inverter control techniques 25
A. Command in full wave 25
B. Sinus-Triangle Pulse Width Modulation (PWM) 26
C. Hysteresis control 28
17. Simulation of the asynchronous machine model 29
18. Simulation result 31
19. Conclusion 33

Chapter II : Scalar control of induction motor 34

1. Introduction 35
2. The scalar command 35
3. Modeling of the asynchronous machine in steady state 37

3.1 Steady state asynchronous machine model 37

3.2 Equivalent electrical diagram of the MAS 39

3.3 Couple explanation 39

4. Scalar voltage control 41

4.1 The command law 41

4.2 V / f control of the asynchronous machine 43

5. Scalar Current Control 46

5.1 Control principle of the voltage inverter controlled by current 47

5.2 Power switch control principle 48

6. Simulation of MAS with Scalar Control 48
7. Regulator PI 50
8. Simulation of MAS with Scalar Control & Regulator PI 52
9. Conclusion 54

Chapter III : Fuzzy Logic Controller 55

1. Introduction 56

2. Fuzzy Logic History 56
3. Why do we use fuzzy logic? 57
4. Fuzzy Logic Architecture 58
4.1 Rules 58
4.2 fuzzification 58
4.3 Inference Engine 59
4.4 Defuzzification 59
5. Membership function 59
6. Fuzzy logic applications 60
7. Advantages and Disadvantages of Fuzzy Logic 61
7.1 Advantages 61
7.2 Disadvantages 62
8. Example of a Fuzzy Logic System 63
MATLAB Fuzzy Logic Toolbox 67
1. Introduction 67
2. Graphical User Interface (GUI) Tools 68
3. How to build a New FIS Mamdani? 71
3.1 The Fuzzy Logic Designer 71
3.2 The Membership Function Editor 74
3.3 The Rule Editor 82
3.4 The Rule Viewer 84
3.5 The Surface Viewer 86
4. Conclusion 88
Chapter IV : The Simulation 89
1. Introduction 90
2. Proposed Design 90
2.1 Fuzzification 92
2.2 Inference Engine 92
2.3 Diffuzification 93
3. The Simulation 96
4. Conclusion 100
GENERAL CONCLUSION 101
References 102

LIST OF FIGURES

Figure I.1: T.P.I.M construction 4
Figure I.2: Stator and winding of a three-phase induction motor 5
Figure I.3: Principle of a coil rotor 6
Figure I.4: squirrel-cage winding of a small induction motor 7
Figure I.5: working principal of three phase induction motor 8
Figure I.6: TPIM Symbols 9
Figure I.7:Schematic representation of a three-phase asynchronous machine. 12
Figure I.8: 3 Rotating frame of reference for axes $(\boldsymbol{d} - \boldsymbol{q})$ 17
Figure I.9: Block diagram of the asynchronous machine supplied with voltage 20
Figure I.10: the voltage inverter associated with the MAS 23
Figure I.11: Full wave control signals (180 °) 25
Figure I.12: 180 ° control inverter output voltages 26
Figure I.13: Principle of the Sine-Triangle PWM 27
Figure I.14: PWM principle 28
Figure I.15: voltage patterns obtained by the inverter by sine-triangle 28
Figure I.16: principle Control by hysteresis 29
Figure I.17:MAS simulink 30
Figure I.18: Speed Result 31
Figure I.19: Couple Result 31
Figure I.20: Current Result 32
Figure II.1 : setting principle.....36
Figure II.2: adjustment with autopilot 36
Figure II.3 : Steady state phase diagram 39
Figure II.4: diagram equivalent to the stator with the magnetic leaks totaled at the rotor 39
Figure II.5: variation of torque as a function of pulsation 40
Figure II.6: characteristic $(\boldsymbol{V} / \boldsymbol{f})$ 42
Figure II.7 : displacement of the torque-slip characteristic as a function of the supply frequency 44
Figure II.8: Autopilot and scalar control. Operating modes. 45
Figure II.9: Scalar control with voltage regulated voltage inverter 45
Figure II.10: characteristic Is according to ωr 46
Figure II.11: Scalar control with voltage inverter regulated while running 47
Figure II.12: Scalar control with current inverter 48

Figure II.13: MAS with Inverter and Scalar Control model 48
Figure II.14: MAS with Inverter and Scalar Control's speed result 49
Figure II.15: MAS with Inverter and Scalar Control's current result 49
Figure II.16: MAS with Inverter and Scalar Control's torque result 49
Figure II.17: speed control block diagram 50
Figure II.18: MAS with Scalar Control & Regulator PI's model 52
Figure II.19: MAS with Scalar Control & Regulator PI's speed result 52
Figure II.20: MAS with Scalar Control & Regulator PI's current result 53
Figure II.21: MAS with Scalar Control & Regulator PI's torque result 53
Figure III.1: Fuzzy Logic architecture. 58
Figure III.2: Fuzzy Logic Membership Function 60
Figure III.3: Fuzzy Logic Toolbox in MatLab 67
Figure III.4: Graphical User Interface (GUI) Tools 68
Figure III.5: The result of typing Fuzzy in the command window 70
Figure III.6: Fuzzy Logic Designer Explanation. 71
Figure III.7: Fuzzy Logic Designer. 72
Figure III.8: Saving In The Workspace. 73
Figure III.9: Fuzzy Logic Designer With Two Inputs 73
Figure III.10: Membership Function Editor Explanation. 75
Figure III.11: Membership Function Editor 77
Figure III.12: Membership Function Settings 78
Figure III.13: Membership Function Editor Settings 79
Figure III.14: Membership Functions Editor After Entering The Settings 81
Figure III.15: The Rules Editor 82
Figure III.16: The Rules Viewer 84
Figure III.17: The Surface Viewer 86
Figure IV.1: Show the design of the controller for washing machine. 91
Figure IV.2: Fuzzy logic inputs membership functions 94
Figure IV.3: Fuzzy logic output membership functions 94
Figure IV.4: New FIS 96
Figure IV.5: Updating our FIS 96
Figure IV.6: Save the current FIS to workspace 97
Figure IV.7: The Final Model 97
Figure IV.8: MAS parameters 98

Figure IV.9: The final speed result 98
Figure IV.10: The final torque result 99
Figure IV.11: The final current result **99**

LIST OF TABLES

Table III.1: Fuzzification table **.....58**

Table IV.1: Show the Input variable and membership **.....92**

Table IV.2: Show the Output variable and membership **.....93**

Table IV.3: Fuzzy Logic Rules Table for the Output ‘’ Washing speed ’’ **.....95**

LIST OF ACRONYMS

MAS :asynchronous machine

S,R: Index corresponding to stator and rotor

A, B, C : Index corresponding to the three phases of the stator

a, b, c : Index corresponding to the three phases of the rotor

$\boldsymbol{d}, \boldsymbol{q}$: Axis corresponding to the frame of reference linked to the rotating field

$\boldsymbol{\theta}$: Electrical angle

$\boldsymbol{Is\alpha}, \boldsymbol{Is\beta}$ **:** Stator current in the frame of reference (α, β)

$\boldsymbol{Isd}, \boldsymbol{Isq}$:Stator current in the frame of reference (d, q)

$\boldsymbol{Is}$:Stator current

$\boldsymbol{Ir}$: Rotor current

$\boldsymbol{Vsa}$, $\boldsymbol{Vsb}$: Stator voltage in the repository (α, β)

$\boldsymbol{Vrd}$, $\boldsymbol{Vrq}$ **:** Statoric tension in the frame of reference (d, q)

$\boldsymbol{Vs}$: Statoric tension

$\boldsymbol{fs}$: Statoric frequency

$\boldsymbol{\phi r\alpha}$, $\boldsymbol{\phi r\beta}$: Rotor flux in the repository (α, β)

$\boldsymbol{\phi rd}$, $\boldsymbol{\phi rq}$: Rotor flux in the frame of reference (d, q)

$\boldsymbol{\phi_R}$: Rotor flux

$\boldsymbol{\phi s}$: Stator flux

$\boldsymbol{\omega r}$: Electric rotor speed

$\boldsymbol{\omega s}$: Electric rotor speed

Ω : Mechanical rotor speed

σ : Total dispersion coefficient

Ce : Electromagnetic torque

Ls , Lr : Self-cyclic stator and rotor inductance per phase

Rr , Rs : Stator and rotor phase resistances

$[P(\theta_{obs})]$: Park Matrix

FIS : Fuzzy inference system

FLC : Fuzzy Logic Controller

Matlab : matrix laboratory

PWM : Pulse width modulation

$[Ki]$: Park's transformation matrix

$[Ki] - 1$: Park's inverse transformation matrix

GENERAL INTRODUCTION

Induction machines are by far the most common type of motor used in industrial, commercial or residential settings.

My problem is how to control the induction motor with command scalar then with fuzzy logic controller and get the same results

- In the first chapter a brief introduction to asynchronous motors, we give their constitution and operating principle followed by their mechanical and electrical characteristics and its modeling.

- In the second chapter I presented the command scalar of the asynchronous motors and more specifically the scalar control by indirect control of the flow. The regulation of the mechanical speed of the MAS with a PI regulator was presented.

- In the third chapter I talked about the FLC and its beginning, and its uses and the advantages and disadvantages and how it works

- In the fourth chapter we have the simulation of our project. We present the overall structure . after the connection between the system and the fis. Then the configuration and cabling used for the control and supervision of our system. Then we explain the different steps followed for controlling the asynchronous motor with FLC, And I finish with the different screens we created for the supervision of the system

Finally I close my document with a general conclusion which is dedicated to a synthesis of the work presented.

Chapter I:

Three Phase Induction Motor

1. Introduction

The induction motor (Asynchronous Motor) is an alternating-current motor in which torque is produced by the reaction between a varying magnetic field generated in the stator and the current induced in the coils of the rotor[1].

The induction motors are the most commonly used motors in many applications like pumps, compressors, small fans, hair dryers, mixers, high speed vacuum cleaners, electric shavers and the washing machine.

2. Induction Motor History

It took over 60 years to arrive at the invention of the induction motor. Here are some of the most prominent contributors to the invention:

In the 1820's JOSEPH HENRY and MICHAEL FARADAY both discovered and experimented the phenomena of induction. After that in 1830's HIPPOLYTE PIXII developed the AC generators. Then in 1870's ELIHU THOMSON, THOMAS EDISON, and WERNER VON SIEMENS were just a few who further developed our understanding of induction of dynamos. Next in 1880's about a dozen notable AC pioneers developed the transformers and AC generators.

In 1888 the induction motor was invented by the Italian GALEILO FERRARIS and the Croatian NIKOLA TESLA.

The induction motor is improved over time. Engineers at Siemens, General Electric and other small firms find ways to make the motors smaller and lighter while producing the same power output. Electric motors today are significantly smaller than those of the early 1900's. Some early motors are still in use today at historic factories and power plants[2].

3. Three Phase Induction Motor Construction

Like all motors, a three-phase induction motor contains a stator (the stationary part) and a rotor (which rotates). Each contains electrical windings that carry current and thus creates a magnetic field. The interaction of the magnetic fields creates the torque that rotates the rotor and the load.

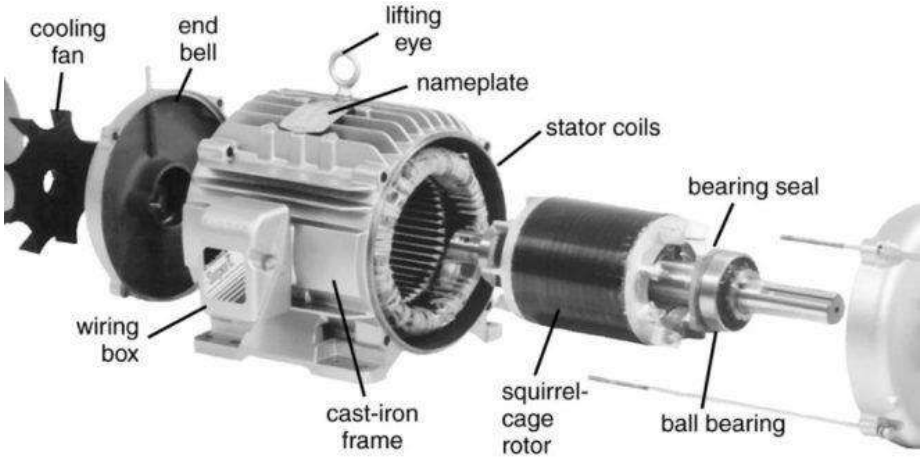


Figure I.1:T.P.I.M construction

3.1 The Stator

Figure I.2 shows a close-up View of the stator of an induction machine. The stator core is built of laminated, electrical-grade steel, and the coils are mounted in slots distributed around the circumference of the stator. The stator winding of the three-phase induction motor is very similar to the armature winding of a three-phase synchronous machine. The core and coil assembly is shown in Figure I.2 is placed in a frame, as shown in Figure I.1.

The frame (item 10), in this case, is cast iron and has cooling fins and mounting feet cast as part of the frame. The National Electrical Manufacturers' Association (NEMA) defines standard frame sizes so that motors will be interchangeable from one manufacturer to another. One coil of wire occupies two slots in the stator (one slot for each side of the coil). Only the portion of the coil in the slots contributes to the magnetic field. The coil sides are connected together by the end turns, which are shown in the front of the picture. In fact, the winding for a phase of the motor is typically composed of several coils of different sizes that occupy a number of slots around the stator. There will be a set of windings for each of the three phases times the number of pole pairs

in the machine. So a two-pole motor would have three-phase coils. While a four-pole motor would have six phase coils. Copper wire is used in the stator coils. The size of the wire is a function of the motor's expected full-load current, insulation type, and rated operating temperature



Figure I.2: Stator and winding of a three-phase induction motor

3.2 The Rotor

Two types of rotors are used in induction machines: wound and squirrel cage. The wound rotor has a winding that is virtually a mirror image of the stator winding. Connections to it are made via slip rings on the shaft. The wound-rotor machine is used for special purposes, usually involving speed control.

The squirrel-cage induction motor is by far the most common type of induction motor. The rotor is built of a stack of steel laminations, such as the one shown in figure I.3. The laminations are aligned and heat-shrunk onto the shaft, and then the molten aluminum is forced into the slots to form the squirrel-cage winding. It is essential that no air pockets be allowed to form as the squirrel cage is cast; otherwise, the resistance of the bar would increase, causing excessive heating. The bars are shorted together at the ends by shorting rings, which are formed at the same time as bars. The rotor winding thus appears like a squirrel or hamster cage.

Figure I.4 is a picture of a small, aluminum squirrel-cage rotor, from which the rotor steel was selectively etched away to show the aluminum bars. One of the shorting rings is visible at the left end of the squirrel cage. Larger rotors usually have fins cast onto them to provide additional cooling surface and to stir the air inside the motor, as shown in figure I.1.

The rotor of the machine must fit inside of the stator and be supported so that it can turn. This is done with bearings. The bearing assemblies are at the top and bottom of the rotor body in figure 1. The bearings shown there are ball bearings, which have less friction than cheaper sleeve bearings. Virtually all motors larger than one horsepower and many smaller ones use ball bearings [3].

In normal operation, the rotor of the asynchronous machine rotates at the speed of Ωr expressed in radians per second, and the magneto motive force produced by the stator currents rotates at theSpeed of synchronism at ΩS expressed in radians per second, but the speed Ω generally remains less than Ωs and we have:

$\Omega = \omega s/p$……(1)

ωS: Stator pulsation, linked to the stator frequency.

p: even number of poles of the resulting magnetomotive force.

The rotor is formed so as to obtain three windings having the same number of poles as that of the stator. The rotor windings are short circuited, the rotational speed of the rotor is less than ΩS, and we note:

$g = (\omega S - \omega)/\omega S$ ……..(2)

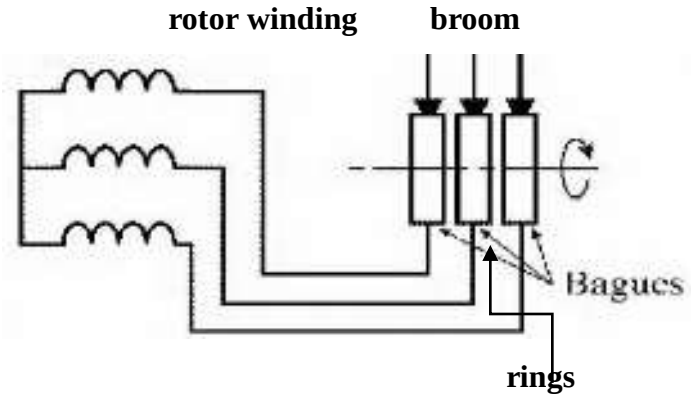


Figure I.3: Principle of a coil rotor

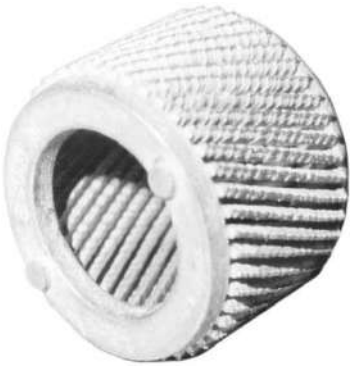


Figure I.4: squirrel-cage winding of a small induction motor

4. Working Principle Of Three Phase Induction Motor

The motor works on the principle of faraday's law of electromagnetic induction. According to 0this law, an EMF induced in a conductor due to the rate of change of flux linkage through the conductor.

In squirrel cage induction motor, the rotor bars are short-circuited by end rings and in slip-ring induction motor, the rotor winding is connected with external resistors. And it cut the stator rotating magnetic field.

According to Faraday's law, an EMF induced in the rotor winding. The rotor is a close circuit. Hence, due to this EMF current will flow through the rotor circuit. This current known as rotor current.

As the current flows through the rotor conductor, the flux induces on it. The direction of rotor flux is the same as the direction of the rotor current.

Now, we have two fluxes. One flux is induced in stator and second is induced in rotor. The stator rotating magnetic field rotates at synchronous speed. And the rotor flux lags behind the stator flux.

The relative motion between rotating stator flux and rotor flux, the rotor flux will try to catch up with the stator flux. According to the Lenz law, the rotor rotates in the direction of stator flux to minimize relative motion. And this is how the rotor starts rotating.

The rotor current produced by induced EMF. Therefore, this motor known as Induction motor.

The rotor never succeeds to catch the stator flux. So, the rotor rotates slights less than the speed of stator and that is synchronous speed.

If the rotor rotating at the speed of the synchronous speed, the relative motion between the stator and rotor is zero. So, no EMF induced in the rotor and no current will flow through the rotor conductors. And therefore, no torque will be generated.

As a result, the rotor cannot rotate at a speed of synchronous speed. The difference between the stator speed (synchronous speed) and rotor speed known as slip[4].

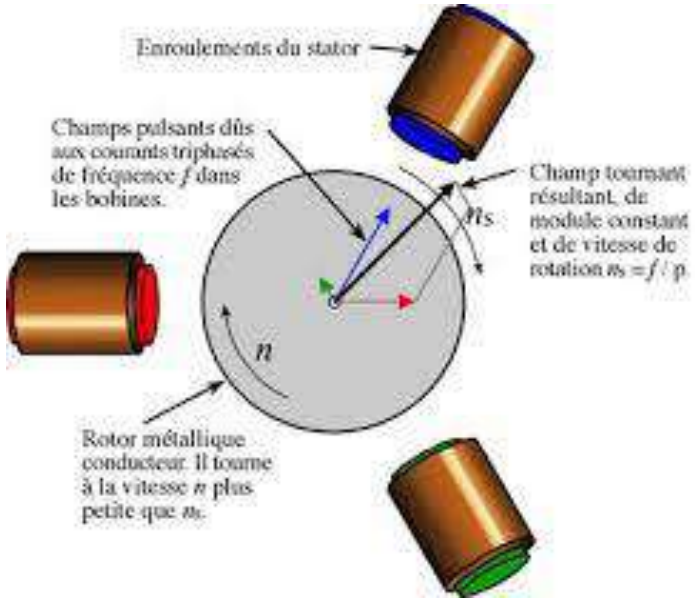


Figure I.5:working principal of three phase induction motor

5. Symbols of Three Phase Induction Motor

Here in this figures we can find the symbols of S.P.I.M:

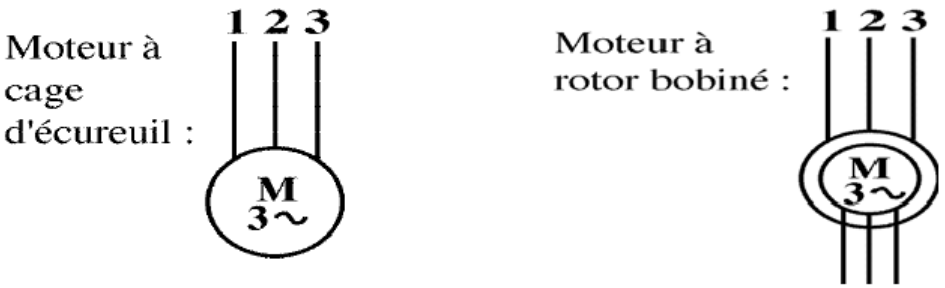


Figure I.6:TPIM Symbols

6. Synchronous Speed (Ns)

When the stator winding is given a single phase AC supply, the magnetic field is produced and the motor rotates at a speed (rotor speed (Nr)) slightly less than the synchronous speed (Ns) which is given by:

$Ns = 120 * f / p$

Ns =Synchronous Speed (the speed of the magnetic field) in rpm.

f =Supply voltage frequency.

P= Number of pole.

Exp:

Calculate the synchronous speed of 4 poles single phase induction motor. The line frequency is 60Hz

Sol:

$Ns = 120 * f / p$; $Ns = (120*60) / 4$; $Ns = 1800rpm$

7. Slip (S)

The slip speed is the difference between the synchronous speed (Ns) and the rotor speed (Nr) expressed as % or per-unit of synchronous speed. The slip (S) is a ratio and doesn't have units.

$S = (Ns – Nr) / Ns$

Slip speed = sync speed – rotor speed = Ns – Nr

- **If** the rotor runs at sync speed (no load) the S=0
- **If** the rotor is stationary (locked) then S=1

Exp:

6 poles induction motor is excited by a single phase, 60Hz source. If the full load speed is 1140 rpm.

Calculate: 1- The synchronous speed

2- Slip speed

3- The slip

Sol:

1) Synchronous speed (Ns):

$$Ns = 120 * f / p$$

$$Ns = (120*60) / 6$$

$$Ns = 1200 \text{ rpm}$$

2) Slip speed:

$$= Ns – Nr$$

$$= 1200 - 1140$$

$$= 60 \text{ rpm}$$

3) Slip:

$$= (Ns – Nr) / Ns$$

$$= (1200 – 1140) / 1200$$

$$= 0.05$$

7. The advantages and disadvantages of the asynchronous motor:

8.1 Advantage

- Robustness
- Simplicity of construction
- Inexpensive motor
- Few interviews
- Good yield

8.2 Disadvantage

- Speed depends on load
- The starting current equal to 6-8 times the rated current.
- vacuum cosφ is very low and not adjustable

9. Area of use :

Asynchronous machines are the most widespread alternative machines, using them in many devices:

- ✓ Pumping stations.
- ✓ Lifting devices.
- ✓ The lifts.
- ✓ Basic machine tools for working metal or wood.
- ✓ Handling devices such as conveyor belts.
- ✓ Ventilation of climatic rooms [5].

10. Modeling of the asynchronous machine:

10.1 Simplifying hypothesis:

The modeling of the asynchronous machine is based on a number of simplifying assumptions, which are:

- Magnetic circuits are symmetrical.
- The distribution of induction in the air gap is sinusoidal.
- The air gap is constant.
- The phenomena of saturation are neglected, which makes it possible to consider the magnetic flux as a linear function of the currents.
-
- The effect of notching is negligible.
- The influence of the skin effect and the heating on the characteristics is not taken into account.

Thus, among the important consequences of these assumptions, one can quote:

- The additivity of the flow.
- The constancy of the specific inductances.
- The law of sinusoidal variation of mutual inductances between stator and rotor windings as a function of the electrical angle between their magnetic axes.

11. Modeling of MAS in the Three-Phase abc plan:

Consider a three-phase asynchronous machine with the stator and the rotor shown schematically in figure I.2 and whose phases are denoted respectively by SA, SB, SC. The variable electrical angle θ as a function of time defines the instantaneous relative position between the magnetic axes of the SA and Ra phases chosen as reference axes:

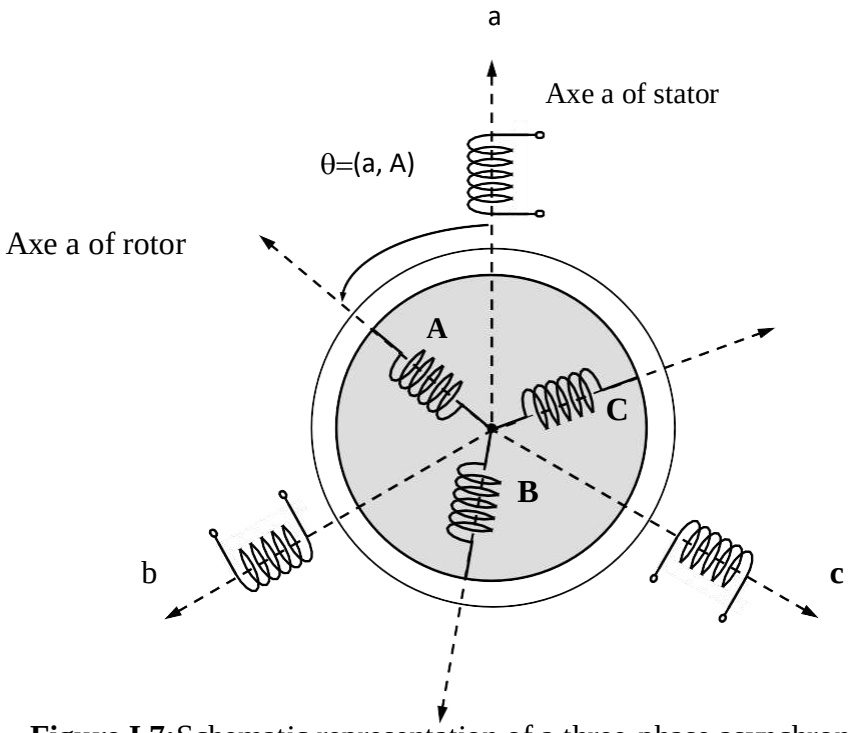


Figure I.7: Schematic representation of a three-phase asynchronous machine.

12. General equations of the three-phase asynchronous machine:

Under these conditions, if we consider that the induction motor is three-phase to the stator and the rotor. The three types of equations reflecting the behavior of the engine are

12.1 Electrical equations:

The voltage equations of the three stator phases and the three rotor phases are:

$$\begin{cases} Vsa = RsIsa + \frac{d}{dt}\theta sa \\ Vsb = RsIsb + \frac{d}{dt}\theta sb \\ Vsc = RsIsc + \frac{d}{dt}\theta sc \end{cases} \quad \text{.....(3)}$$

$$\begin{cases} Vra = RrIra + \frac{d}{dt}\theta ra \\ Vrb = RrIrb + \frac{d}{dt}\theta rb \\ Vrc = RrIrc + \frac{d}{dt}\theta rc \end{cases} \quad \text{.....(4)}$$

Vsa, Vsb, Vsc: Voltages applied to the three stator phases. isa, isb, isc: Currents which cross the three stator phases.

sa, sb, sc: Total fluxes through these windings.

Vsa, Vsb, Vsc: Rotor voltages isa, isb, isc: Rotor currents

sa, sb, sc: Rotor fluxes

RS: Resistance to an ephasestatoric. Rr: Resistance of a rotor phase

Equations (2) and (3) can be written in the following matrix form:

For the stator

$$\begin{bmatrix} Vsa \\ Vsb \\ Vsc \end{bmatrix} = \begin{bmatrix} Rs & 0 & 0 \\ 0 & Rs & 0 \\ 0 & 0 & Rs \end{bmatrix} \begin{bmatrix} isa \\ isb \\ isc \end{bmatrix} + \frac{d}{dt} \begin{bmatrix} \theta sa \\ \theta sb \\ \theta sc \end{bmatrix} \quad \text{.....(5)}$$

Or in the condensed form as follows:

$$[Vs(abc)] = [Rs][is(abc)] + \frac{d}{dt}[\theta s(abc)] \quad \text{.....(6)}$$

For the rotor:

$$\begin{bmatrix} Vra \\ Vrb \\ Vrc \end{bmatrix} = \begin{bmatrix} Rr & 0 & 0 \\ 0 & Rr & 0 \\ 0 & 0 & Rr \end{bmatrix} \begin{bmatrix} ira \\ irb \\ irc \end{bmatrix} + \frac{d}{dt} \begin{bmatrix} \theta ra \\ \theta rb \\ \theta rc \end{bmatrix} \quad \text{.....(7)}$$

Or in the condensed form as follows:

$$[Vr(abc)] = [Rr][ir(abc)] + \frac{d}{dt}[\theta r(abc)] \quad \text{.....(8)}$$

12.2 Magnetic equations:

The simplifying assumptions cited previously lead to linear relationships between flux and currents of the asynchronous machine, these relations are written in a matrix as follows:

For the stator:

$$\begin{bmatrix} \theta sa \\ \theta sb \\ \theta sc \end{bmatrix} = [Ls] \begin{bmatrix} isa \\ isb \\ isc \end{bmatrix} + [Msr] \begin{bmatrix} ira \\ irb \\ irc \end{bmatrix} \quad \text{.....(9)}$$

For the rotor:

$$\begin{bmatrix} \theta ra \\ \theta rb \\ \theta rc \end{bmatrix} = [Lr] \begin{bmatrix} ira \\ irb \\ irc \end{bmatrix} + [Msr] \begin{bmatrix} isa \\ isb \\ isc \end{bmatrix} \quad \text{.....(10)}$$

Such as: $[Msr]=[Mrs]^T$

We denote by :

$[Ls]$:Matrix of stator inductances.

$[Lr]$:Matrix of rotor inductances.

$[Msr]$:Matrix of mutual stator inductances.

$[Mrs]$:Matrix of mutual rotor inductances.

Or :

$$[Ls] = \begin{bmatrix} Ls & Ms & Ms \\ Ms & Ls & Ms \\ Ms & Ms & Ls \end{bmatrix} \text{.....(11)}$$

$$[Lr] = \begin{bmatrix} Lr & Mr & Mr \\ Mr & Lr & Mr \\ Mr & Mr & Lr \end{bmatrix} \text{.....(12)}$$

So :

$$[Msr]=[Mrs]^{T} = M_0 \begin{bmatrix} \cos(\theta) & \cos(\theta - 2\pi/3) & \cos(\theta + 2\pi/3) \\ \cos(\theta + 2\pi/3) & \cos(\theta) & \cos(\theta - 2\pi/3) \\ \cos(\theta - 2\pi/3) & \cos(\theta + 2\pi/3) & \cos(\theta) \end{bmatrix} \text{.....(13)}$$

With :

***Ls*:**Specific inductance of a stator phase.

***Lr*:** Specific inductance of a rotor phase.

Ms: Mutual inductance between stator phases.

Mr: Mutual inductance between rotor phases.

***θ*:**Electrical angle defines the instantaneous relative position between the axes

stator and rotor axes which are chosen as reference axe**s.**

M:Maximum mutual inductance between stator phase and rotor phase match

12.3 Mechanical equations:

The study of the characteristics of the asynchronous machine introduces variation not only electrical parameters (voltage, current, flux) but also mechanical parameters (torque, speed):

$$Ce = P[is(abc)]^{T}\frac{d}{dt}[Msr][ir(abc)] \text{.....(14)}$$

To have a complete model of the machine, it is necessary to introduce the equation of motion ofthe machine is expressed as follows:

$J\frac{d}{dt}\Omega r = Ce - Cr - f\Omega r$……..…(15)

J : Moment of inertia of rotating masses**.**

Cr : Resistant torque imposed on the machine shaft.

Ωr : Rotor speed.

Ce : Electromagnetic couple.

f: Viscous coefficient of friction.

The equations (4) thus obtained have variable coefficients resulting in the complexity of resolutionof the model defined by (3). This will lead to the use of the transformation of Park which will makeconstant these parameters.

13. Park transformation:

The purpose of the transformation of Park is to process a wide range of machines in a unified way. reducing to a single model, This conversion is often called transformation of axes, madecorresponding to the two windings of the original machine followed by a rotation, the equivalent windings from an electrical and magnetic point of view. This transformation thus, for the purpose of making themutual inductances of the model independent of the angle of rotation

13.1Different benchmarks:

The isotropy of the asynchronous motor allows great flexibility in the composition of the equationsof the machine in two axes using Park components, this requires the use of a benchmarkwhich allows to simplify the analytical expressions as much as possible. There are different possibilities forthe choice of axis coordinate system is practically reduced to three frames of reference (two-phase systems)orthogonal

1) Reference frame motionless with respect to the stator: $(\alpha - \beta) \rightarrow \omega obs = 0$.

2) Reference frame immobile compared to the rotor: $(x - y) \rightarrow \omega obs = \omega R$.

3) Reference frame motionless compared to the rotating field: (d - q)→ωobs = ωS.

Or :

ωobs: Angular speed of rotation of the two-phase axis system with respect to the axis system three-phase.

13.2 Modeling of MAS in the two-phase dq plane

Due to the existence of continuous trigonometric terms in the matrix of mutual inductances [Msr],the coefficients of the differential equations are variable and the analytical resolution of the system comes up againstpractically insurmountable difficulties To obtain a system of equations with constant coefficients, wetransforms the stator and rotor windings into two orthogonal two-phase windings dq according to thetransformation of PARK. The conversion involves the transformation of the equivalent windings from the point ofelectric and magnetic sight. Figure (8) represents the transformation of the real windings abc intod-q orthogonal windings

- Direct along axis (d).
- Quadrature (transverse) along the (q) axis.
- Homopolar

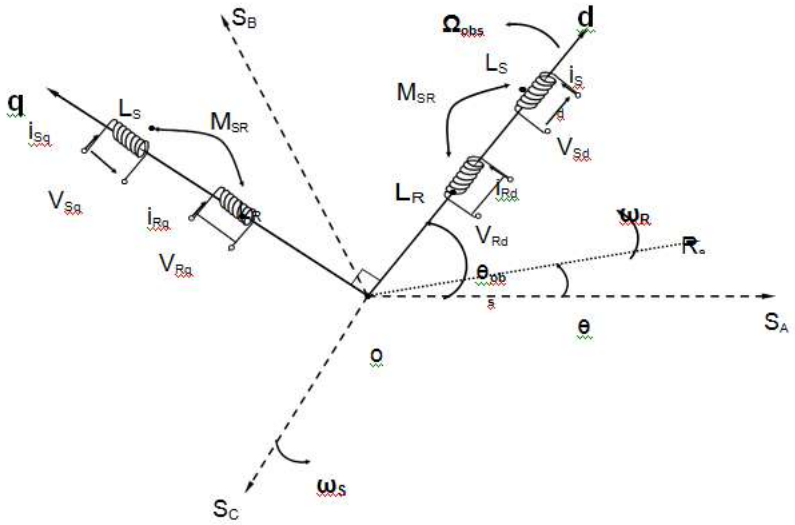


Figure I.8: 3 Rotating frame of reference for axes $(d - q)$

Or :

$\theta_{obs} = \int \omega_{obs} dt$: is any position of observation between the two-phase axis systems bycompared to the three-phase axis system.

The application of Park's transformation to the asynchronous machine corresponds to a transformationthree coil (stator and rotor) with two equivalent coil taking the same considerationor aspects in term, flow, torque, current or month an image that will be perfectly for themproportional

For the transition from the three-phase system to the two-phase system, we have the following equivalents:

The voltage equivalent :$[Vdqo] = [P(\theta obs)][Vabc]$

L'équivalent de courant : :$[idqo] = [P(\theta obs)][iabc]$……(16)

The equivalent of flux ::$[\phi dqo] = [P(\theta obs)][\phi abc]$

Or :

$[P(\theta obs)]$: is Park's matrix

In the case of a reverse passage, we have:

$$\begin{cases} [Vabc] = [P(\theta obs)] - 1[Vdqo] \\ [iabc] = [P(\theta obs)] - 1[idqo] \\ [\phi abc] = [P(\theta obs)] - 1[\phi dqo] \end{cases} \quad \text{……(17)}$$

The direct and inverse modified Park transformation matrix is then written:

$$[P(\theta obs)] = \sqrt{2/3}\begin{bmatrix} \cos(\theta obs) & \cos(\theta obs - 2\pi/3) & \cos(\theta obs + 2\pi/3) \\ -\sin(\theta obs) & -\sin(\theta obs - 2\pi/3) & -\sin(\theta obs + 2\pi/3) \\ 1/\sqrt{2} & 1/\sqrt{2} & 1/\sqrt{2} \end{bmatrix} \text{……..(18)}$$

The factor ($\sqrt{2/3}$): is there to conserve instantaneous electrical power

$$[P(\theta obs)]\mathrm{T} = \sqrt{2/3}\begin{bmatrix} \cos(\theta obs) & -\sin(\theta obs - 2\pi/3) & 1/\sqrt{2} \\ \cos(\theta obs - 2\pi/3) & -\sin(\theta obs - 2\pi/3) & 1/\sqrt{2}) \\ \cos(\theta obs + 2\pi/3) & -\sin(\theta obs + 2\pi/3) & 1/\sqrt{2} \end{bmatrix} \ldots\ldots\ldots\ldots\ldots(19)$$

When the angleθobs is assigned to the value zero, the transformation of Park is saidClarke's transformation and the passing matrix is written as follows

$$[C] = \begin{bmatrix} 1 & -1/2 & -1/2 \\ 0 & \sqrt{3}/2 & -\sqrt{3}/2 \\ 1/2 & 1/2 & 1/2 \end{bmatrix} \ldots\ldots\ldots\ldots(20)$$

The application of Park's transformation to electrical matrix models, (3) and (4), and magnetic, (5), (6), gives the following equations:

13.3 Electrical equations :

a- Equations of tensions:

$$\begin{bmatrix} Vsd \\ Vsq \end{bmatrix} = \begin{bmatrix} Rs & 0 \\ 0 & Rs \end{bmatrix}\begin{bmatrix} isd \\ isq \end{bmatrix} + \frac{d}{dt}\begin{bmatrix} \theta sd \\ \theta sq \end{bmatrix} + \begin{bmatrix} 0 & -ws \\ ws & 0 \end{bmatrix}\begin{bmatrix} \theta sq \\ \theta sd \end{bmatrix} \ldots\ldots\ldots\ldots(21)$$

$$\begin{bmatrix} Vrd \\ Vrq \end{bmatrix} = \begin{bmatrix} Rr & 0 \\ 0 & Rr \end{bmatrix}\begin{bmatrix} ird \\ irq \end{bmatrix} + \frac{d}{dt}\begin{bmatrix} \theta rd \\ \theta rq \end{bmatrix} + \begin{bmatrix} 0 & -(ws - wr) \\ (ws - wr) & 0 \end{bmatrix}\begin{bmatrix} \theta rq \\ \theta rd \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \ldots\ldots\ldots\ldots(22)$$

b-Magnetic equation:

$$\begin{bmatrix} \theta sd \\ \theta sq \end{bmatrix} = \begin{bmatrix} Ls & 0 \\ 0 & Ls \end{bmatrix}\begin{bmatrix} isd \\ isq \end{bmatrix} + \frac{d}{dt}\begin{bmatrix} \theta sd \\ \theta sq \end{bmatrix} + \begin{bmatrix} M & 0 \\ 0 & M \end{bmatrix}\begin{bmatrix} ird \\ irq \end{bmatrix} \ldots\ldots\ldots\ldots(23)$$

$$\begin{bmatrix} \theta rd \\ \theta rq \end{bmatrix} = \begin{bmatrix} Lr & 0 \\ 0 & Lr \end{bmatrix}\begin{bmatrix} ird \\ irq \end{bmatrix} + \frac{d}{dt}\begin{bmatrix} \theta rd \\ \theta rq \end{bmatrix} + \begin{bmatrix} M & 0 \\ 0 & M \end{bmatrix}\begin{bmatrix} isd \\ isq \end{bmatrix} \ldots\ldots\ldots\ldots(24)$$

With :

Ls = ls – M , Lr = lr – M :Self-cyclic inductance of the stator and the rotor respectively.

M = $3/2$ Mo : Mutual stator-rotor cyclic inductance

c– Mechanical Equation :

The electromechanical couple represented in equation (9) becomes:

$$\text{Cem} = \text{p}\frac{M}{Lr}\left(\phi rdisd - \phi rqisq\right)$$

The equation for mechanical speed is represented by equation (10).

$$\text{J}\frac{d}{dt}\Omega\text{r} = \text{Cem} - \text{Cr} - f\,\Omega\text{r}$$

14. The asynchronous machine supplied with voltage:

For an asynchronous machine supplied with voltage, if we consider the stator current iS and the fluxrotor ϕR as state variables, and the pulsation ωs and voltages Vsd, Vsq, as magnitude ofcommand and the torque Cr as a disturbance, we will have the following block diagram

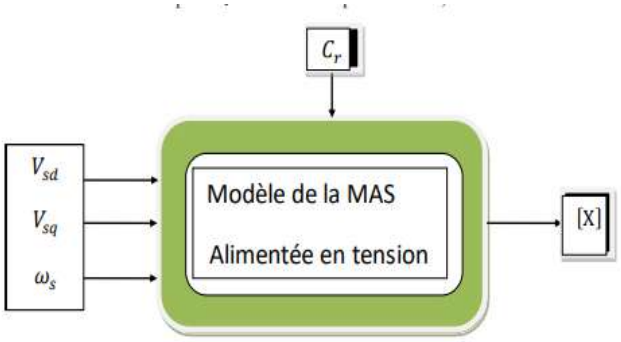


Figure I.9: Block diagram of the asynchronous machine supplied with voltage

The output vector [X], can have one of the forms of the different expressions

$$[X]=\begin{bmatrix} isd \\ isq \\ ird \\ irq \end{bmatrix} \quad \text{or} \quad [X]=\begin{bmatrix} \phi sd \\ \phi sq \\ \phi rd \\ \phi rq \end{bmatrix} \quad \text{or} [X]=\begin{bmatrix} \phi sd \\ \phi sq \\ ird \\ irq \end{bmatrix} \quad \text{or} \quad [X]=\begin{bmatrix} isd \\ isq \\ \phi rd \\ \phi rq \end{bmatrix}$$

15. State representation of the MAS model:

The general form of the equation of state is written as follows:

$$[\dot{X}] = [A][X] + [B][U] \text{……..(25)}$$

With :

$[\boldsymbol{X}] = [isa\ isb\ \phi sa\ \phi sb]^T$: state vector

$[\boldsymbol{U}] = [Vsa\ Vsb]$: Command vector

$[\boldsymbol{A}]$: System condition assessment matrix

$[\boldsymbol{B}]$: Control system matrix

$$\begin{cases} \frac{di_{s\alpha}}{dt} = -\left(\frac{R_t}{L_s\sigma}\right) i_{s\alpha} + \frac{1-\sigma}{MT_s\sigma}\Phi_{r\alpha} + \frac{1-\sigma}{M\sigma}\omega\Phi_{r\beta} + \frac{1-\sigma}{L_s\sigma}V_{s\alpha} \\ \frac{di_{s\beta}}{dt} = -\left(\frac{R_t}{L_s\sigma}\right) i_{s\beta} + \frac{1-\sigma}{MT_s\sigma}\Phi_{s\alpha} + \frac{1-\sigma}{M\sigma}\omega\Phi_{r\beta} + \frac{1-\sigma}{L_s\sigma}V_{s\beta} \\ \frac{d\Phi_{r\alpha}}{dt} = \frac{M}{T_r} i_{s\alpha} - \frac{1}{T_r}\Phi_{r\alpha} - \omega\Phi_{r\beta} \\ \frac{d\Phi_{r\beta}}{dt} = -\frac{M}{T_r} i_{s\beta} + \omega\Phi_{r\alpha} - \frac{1}{T_r}\Phi_{r\beta} \end{cases} \quad \text{.....(26)}$$

With :

$Ts = Ls / Rs$ and $Tr = Lr / Rr$: stator and rotor time constant

$\sigma = 1 - \frac{M^2}{LsLr}$: Coefficient of dispersion

$$[A] = \begin{bmatrix} -\frac{R_t}{L_s\sigma} & 0 & \frac{1-\sigma}{MT_s\sigma} & -\frac{1-\sigma}{M\sigma}\omega \\ 0 & -\frac{R_t}{L_s\sigma} & -\frac{1-\sigma}{M\sigma}\omega & \frac{1-\sigma}{MT_s\sigma} \\ \frac{M}{T_r} & 0 & -\frac{1}{T_r} & \omega \\ 0 & \frac{M}{T_r} & \omega & -\frac{1}{T_r} \end{bmatrix} \quad \text{.....(27)}$$

$$[B] = \begin{bmatrix} \frac{1}{Ls\sigma} & 0 \\ 0 & \frac{1}{Ls\sigma} \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \quad \text{.....(28)}$$

$$[A] = \begin{bmatrix} -\frac{Rt}{Ls\sigma} & ws & \frac{1-\sigma}{Mts\sigma} & -\frac{1-\sigma}{M\sigma}w \\ -ws & -\frac{Rt}{Ls\sigma} & -\frac{1-\sigma}{M\sigma}w & \frac{1-\sigma}{Mts\sigma} \\ \frac{M}{TR} & 0 & -\frac{1}{Tr}(ws-w) & \\ 0 & \frac{M}{Tr}(ws-w) & -\frac{1}{Tr} & \end{bmatrix} \quad \text{.....(29)}$$

The electromagnetic torque developed by the asynchronous machine given by the following relation:

$$Ce = \frac{PM}{Lr}(\emptyset r\alpha Is\beta - \emptyset r\beta Is\alpha) \quad \text{.....(30)}$$

16. Modeling of the voltage inverter:

16.1 Introduction:

A voltage inverter is a static converter ensuring the conversion of continuous energy to the form of alternating energy by a switching set [12], these switching sets relate to the control of the switches of the inverter. It must deliver variable voltages and frequencies in order to vary the speed of the machine, these variable voltages and frequencies must be the image of the references generated by the adopted control strategy. For this, several inverter control techniques are used, the main objective being to approach the alternating voltage as closely as possible to the reference calculated at the control level.

Pulse Width Modulation (PWM) is the most widely used technique for driving inverters. It is far from being an accessory element in the speed variation chain, it plays an essential role with consequences on all the performance of the system.

In this part of the chapter we will explain the modeling of the three-phase two-level voltage inverter used in our case, then we will describe the inverter control techniques.

16.2 Modeling of the voltage inverter:

The three-phase voltage inverter supplying the stator of the asynchronous machine is a Two-level voltage inverter formed by three independent arms each comprising two switching cells and two freewheeling diodes (figure 1.10). The voltage vector delivered depends on the state of the switchessa , sb and sc

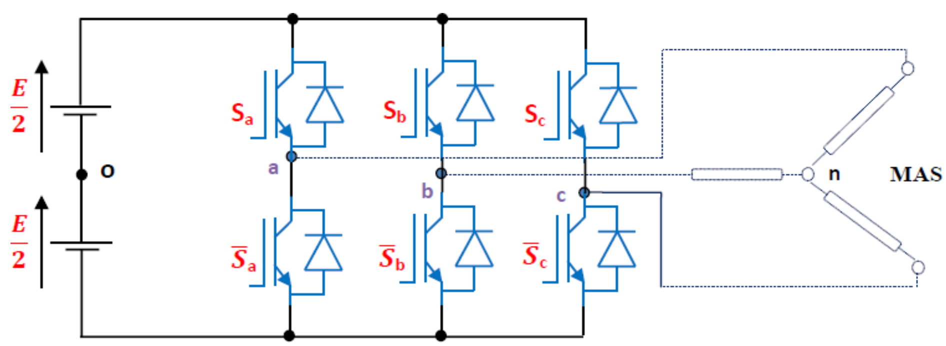


Figure I.10: the voltage inverter associated with the MAS

16.3 Mathematical model of the voltage inverter [13] :

The state of the switches, assumed to be perfect, can be defined by three quantities

command booleans Si (i = a, b, c):

- Si = 1 when the top switch is closed and the bottom switch is open.
- Si = 0 when the top switch is open and the bottom switch is closed.

Under these conditions we can write the voltages Vio as a function of the control signals Si and taking into account the fictitious point "o" represented in figure (10)

$$Vio = E\left(Si - \frac{1}{2}\right) \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots..(31)$$

Let ‘n’ be the neutral point on the AC side (MAS), then the three phase-to-phase voltages: Uab, Uca, and Ubc are defined by the following relations:

$$\begin{cases} Uab = Van - Vbn \\ Uca = Vcn - Van \\ Ubc = Vbn - Vcn \end{cases} \quad \text{.....(32)}$$

The load constituted by the machine is balanced ($Van + Vbn + Vcn = 0$), we will therefore have:

$$\begin{cases} Van = \frac{1}{3}(Uab - Uca) \\ Vbn = \frac{1}{3}(Ubc - Uab) \\ Vcn = \frac{1}{3}(Uca - Ubc) \end{cases} \quad \text{.....(33)}$$

By showing the point "o", the voltages between phases can also be written:

$$\begin{cases} Uab = Vao - Vbo \\ Ubc = Vbo - Vco \\ Uca = Vco - Vao \end{cases} \quad \text{.....(34)}$$

By replacing (34) in (33) we get:

$$\begin{cases} Van = +\frac{2}{3}Vao - \frac{1}{3}Vbo - \frac{1}{3}Vco \\ Vbn = -\frac{1}{3}Vao + \frac{2}{3}Vbo - \frac{1}{3}Vco \\ Vcn = -\frac{1}{3}Vao - \frac{1}{3}Vbo + \frac{2}{3}Vco \end{cases} \quad \text{.....(35)}$$

The use of expression (31) makes it possible to establish the instantaneous equations of the phase-to-neutral voltages as a function of the control quantities:

$$\begin{cases} Van = +\frac{2}{3}SaE - \frac{1}{3}SbE - \frac{1}{3}ScE \\ Vbn = -\frac{1}{3}SaE + \frac{2}{3}SbE - \frac{1}{3}ScE \\ Vcn = -\frac{1}{3}SaE - \frac{1}{3}SbE + \frac{2}{3}ScE \end{cases} \quad \text{.....(36)}$$

In matrix form:

$$\begin{bmatrix} Van \\ Vbn \\ Vcn \end{bmatrix} = \frac{E}{3} \begin{bmatrix} +2 & -1 & -1 \\ -1 & +2 & -1 \\ -1 & -1 & +2 \end{bmatrix} \begin{bmatrix} Sa \\ Sb \\ Sc \end{bmatrix} \quad \text{.....(37)}$$

16.4 Inverter control techniques:

Voltage inverters can be controlled using several control strategies.In order to produce an output voltage close to the sinusoid, various control strategies have been proposed by various authors for the voltage inverter. We study the following strategies[14]:

- Command in full wave.
- Sine-triangle command.
- Hysteresis control

A. Command in full wave.

To illustrate the operation of the inverter, it is interesting to study the behavior in full wave (180 °). This operation allows us to better understand the other control strategies. This command is also called the six-step command.

In this type of control each transistor conducts for (180 °). There are six modes of operation in each cycle, and each time mode (60 °) the transistors are numbered according to the manner in which they receive the sequences of moving each other by (60 °)[15].

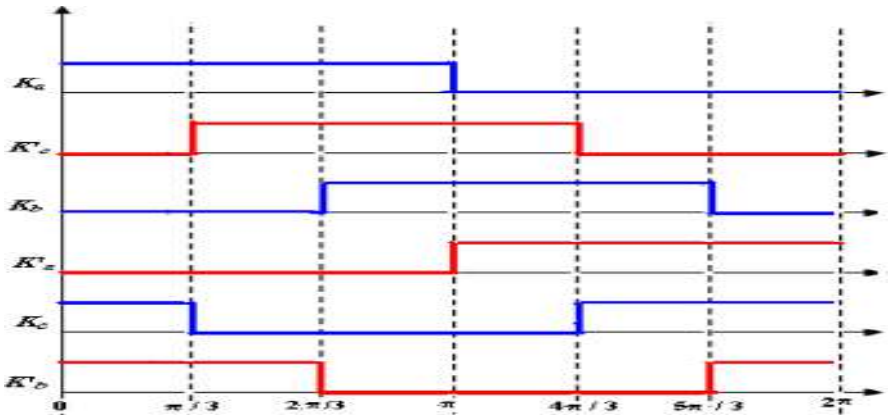


Figure I.11: Full wave control signals (180 °)

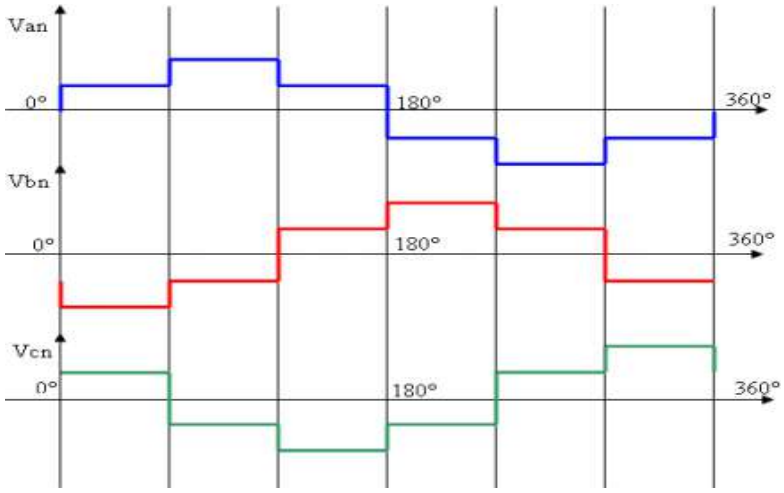


Figure I.12:180 ° control inverter output voltages

B. Sinus-Triangle Pulse Width Modulation (PWM)

Pulse width modulation is achieved by comparing a low frequency modulated wave (reference voltage) with a triangular shaped high frequency carrier wave. The switching times are determined by the points of intersection between the carrier and the modulant, the switching frequency of the switches is set by the carrier

With:

$$\begin{cases} V'as = \sqrt{2}V's\ Cos(wst) \\ V'bs = \sqrt{2}V's\ Cos(wst - \frac{2\pi}{3}) \\ V'cs = \sqrt{2}V's\ Cos(wst + \frac{4\pi}{2}) \end{cases} \quad \text{.....(38)}$$

And :

$$\begin{cases} Vp = \frac{E}{2}\left(\frac{4}{Tm}t - 1\right) & si: t < 0.5Tm \\ Vp = \frac{E}{2}\left(-\frac{4}{Tm}t + 3\right) & si: t < Tm \end{cases} \quad \text{.....(39)}$$

In three-phase, three sinusoidal references phase-shifted$2\pi/3$ at the same frequency f. As at the output of the inverter, the voltage is not purely sinusoidal, so it contains harmonics, which are solely responsible for interference which generates additional losses. This **PWM** is used to remedy these problems and it has the following advantages:

- Fine control of the amplitude of the fundamental output voltage generally on the largest possible range.
- Variation of the frequency of the output voltage.

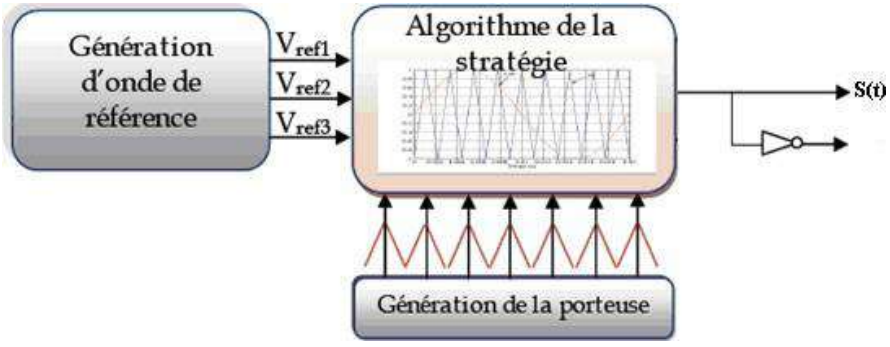


Figure I.13: Principle of the Sine-Triangle PWM

The essential parameters of PWM are:

- The modulation frequency.
- The modulation index "m" which is the image of the frequency ratio of the

Carrierfm on the reference frequency$f$ ..

- The modulation rate (the tuning coefficient) "r" which is the image of the ratio of the voltage amplitudes of the reference to the peak value of the modulation wave.

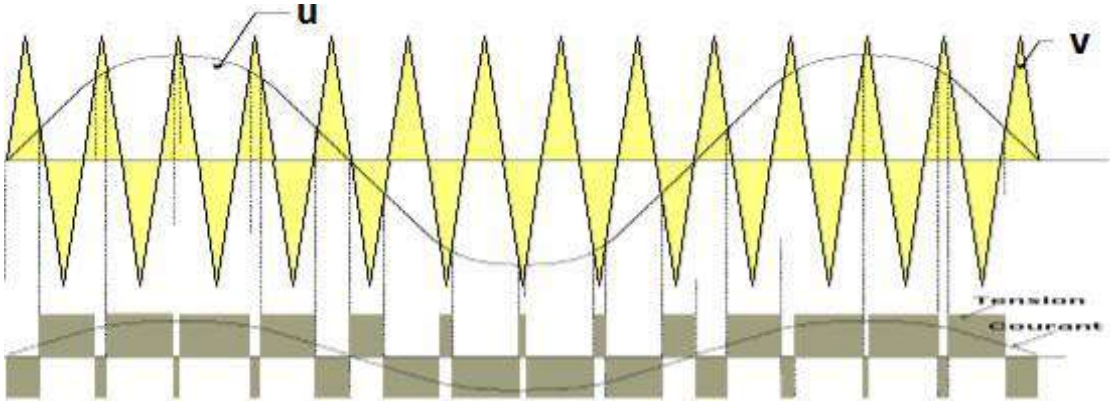


Figure I.14 : PWM principle

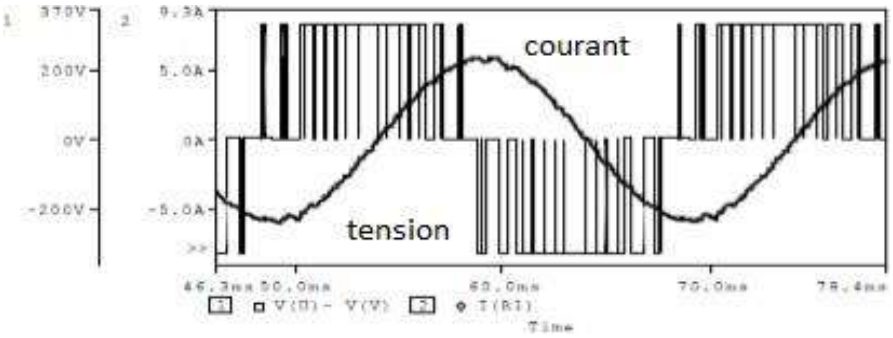


Figure I.15: voltage patterns obtained by the inverter by sine-triangle

We notice that the output voltage is gathered at the sine wave more than the control of the inverter in full wave.

C. Hysteresis control:

The output current $i\,(t)$ is the only control variable, however it is compared with two hysteresis thresholds coupled a $i(t) = iref \pm ei$; the result of this comparison determines the state of the switches.

We define the error ei by:

$$ei = i' - i$$

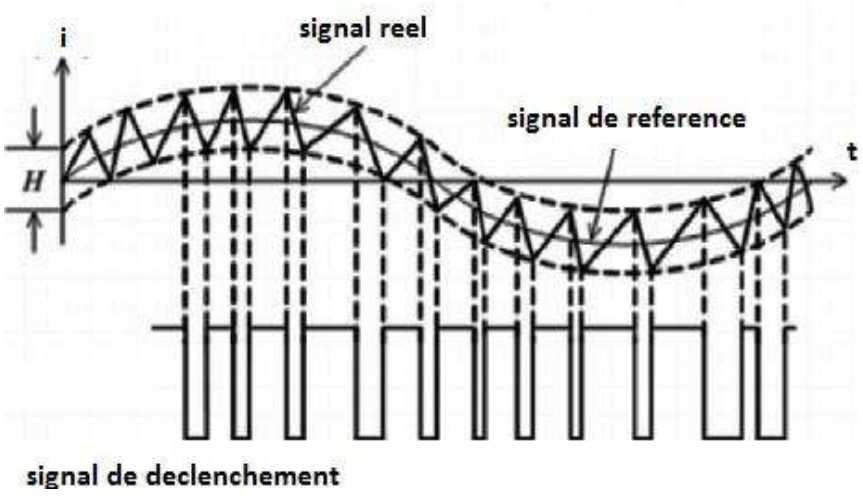


Figure I.16 : principle Control by hysteresis

When the load current is above the upper limit of the hysteresis, a negative voltage is applied by the inverter to the load, causing the current in the inductor to decrease. Once the current reaches the lower hysteresis limit, a positive voltage is applied by the inverter through the coil, causing the current to increase and repeating it.

***Hi*:** Hysteresis band the corrector generates at the output s = 1 to increase the current and s = 0 to reduce it

The corrector output has hysteresis and defined by:

$$\begin{cases} Si\ ei > Hi & s = 1 \\ Si\ |ei| < Hi\ and \frac{di}{dt} > 0 & s = 1 \\ Si\ |ei| < Hi\ and \frac{di}{dt} < 0 & s = 0 \\ Si\ ei < -Hi & s = 0 \end{cases} \quad \text{.....(40)}$$

17. Simulation of the asynchronous machine model

The asynchronous machine is normally supplied directly from the industrial network by abalanced three-phase voltage system.In certain applications for which the variation of the speed is necessary, the motor will besupplied by a three-phase voltage system or by a three-phase current system (injected) intothe stator windings, through an electronic power converter placed betweenthe motor and the industrial electrical network

Figure (I.17) represents the block diagram of the model obtained, the latter will be simulated using theSIMULINK software under MATLAB. The parameters of the MAS used in this work are givenin Annex

The simulation will be carried out in the two frames of reference $(d,q)\ and\ (\alpha,\beta)$ for a nominal load test after a no-load start. The supply voltages are assumed to be perfectly sinusoidal with equal and constant amplitudes, they can be presented as follows[6]:

$$\begin{cases} \boldsymbol{Vsa} = \sqrt{2}Vs\ \sin(wst) \\ \boldsymbol{Vsb} = \sqrt{2}Vs\ \sin(wst - \frac{2\pi}{2}) \\ \boldsymbol{Vsc} = \sqrt{2}Vs\ \sin(wst + \frac{2\pi}{2}) \end{cases} \quad \text{.....(41)}$$

$\boldsymbol{Vs}$: *RMS voltage value*

$\boldsymbol{Ws}$: *Feeding pulse*

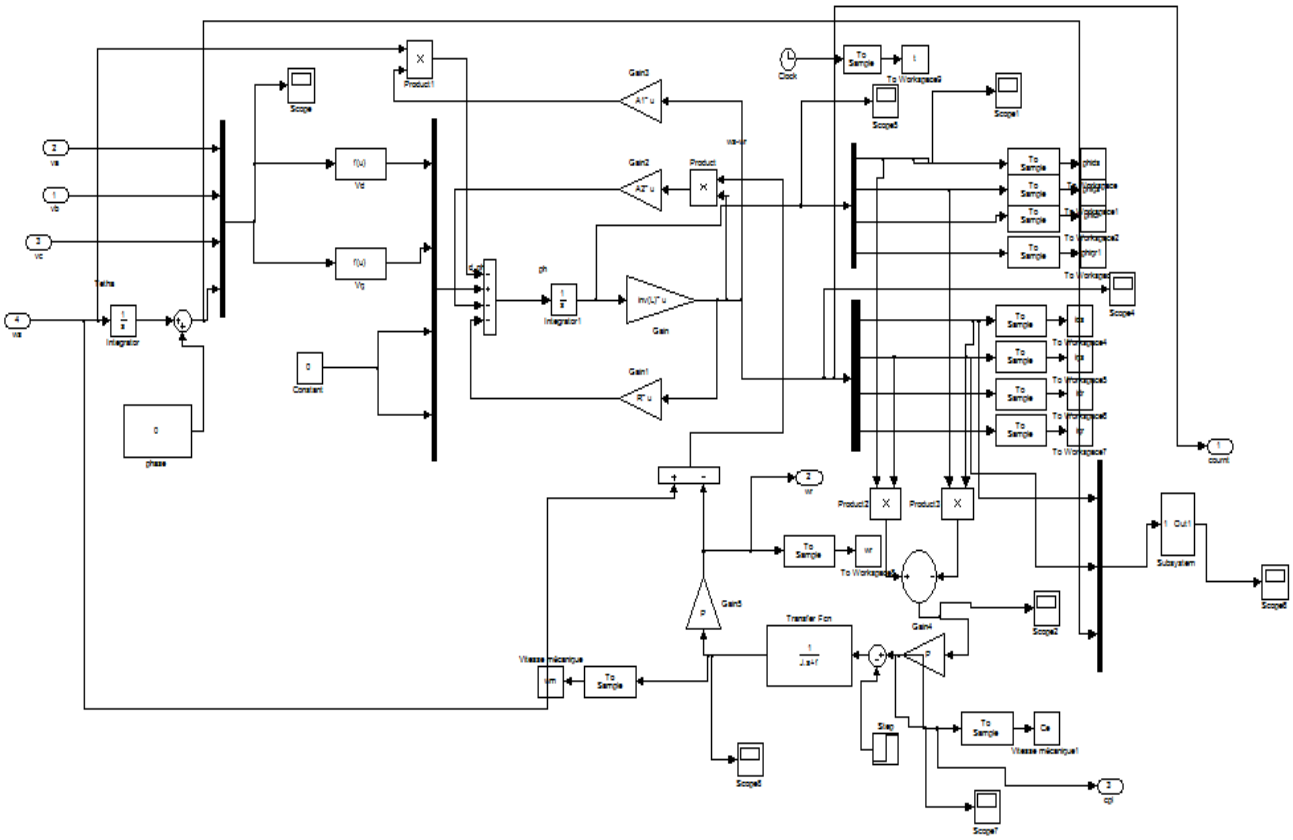


Figure I.17:MAS simulink

18. Simulation result:

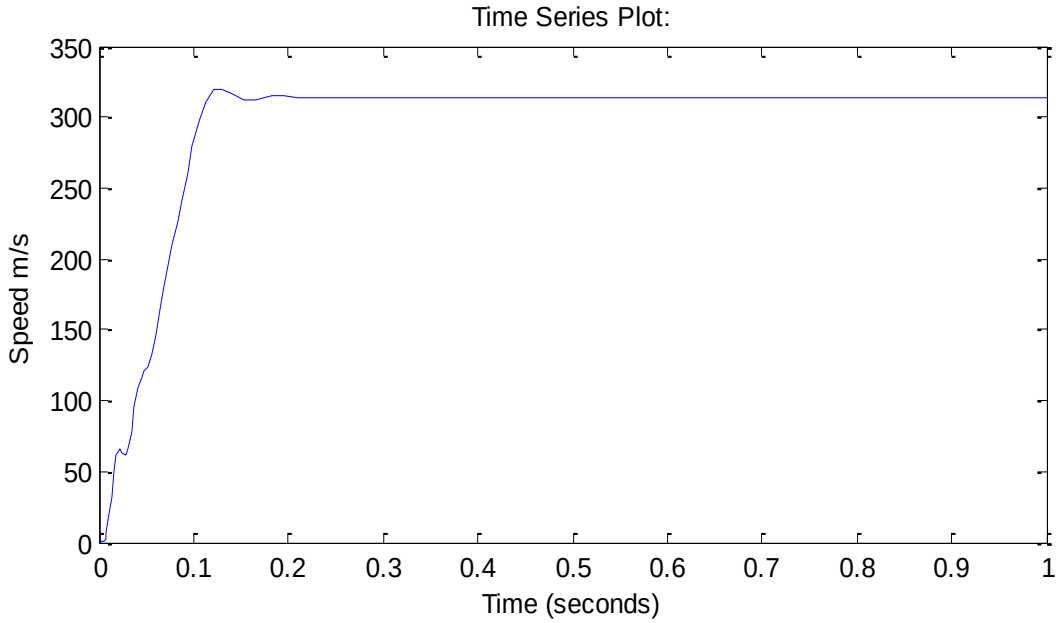


Figure I.18: Speed Result

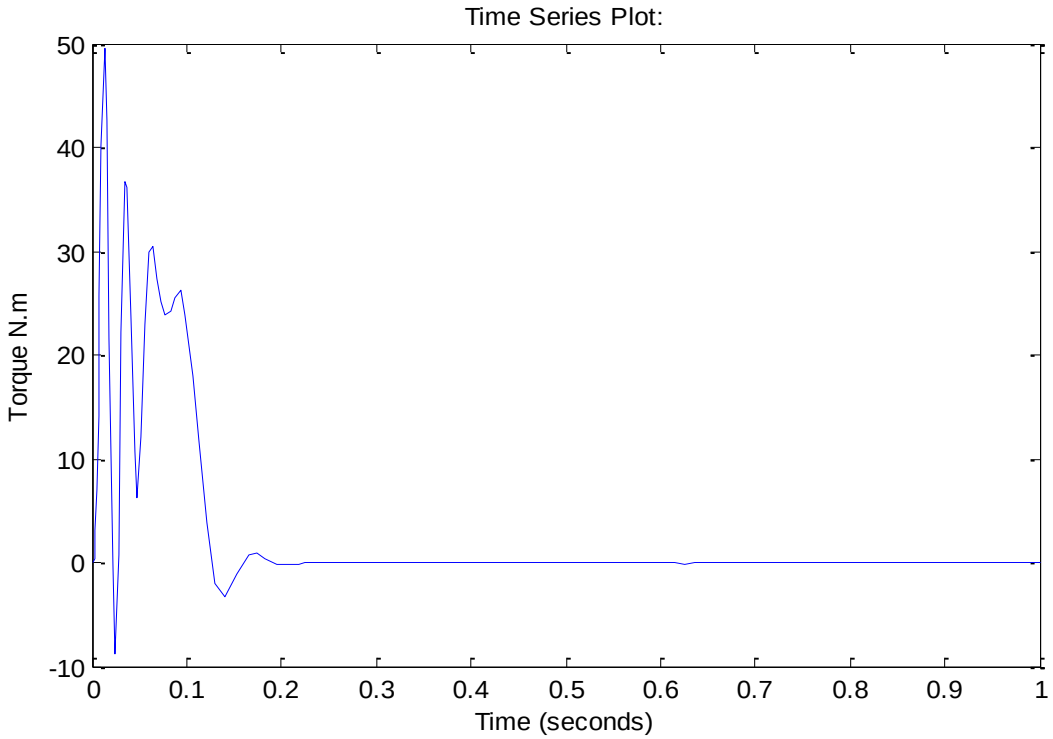


Figure I.19:torque Result

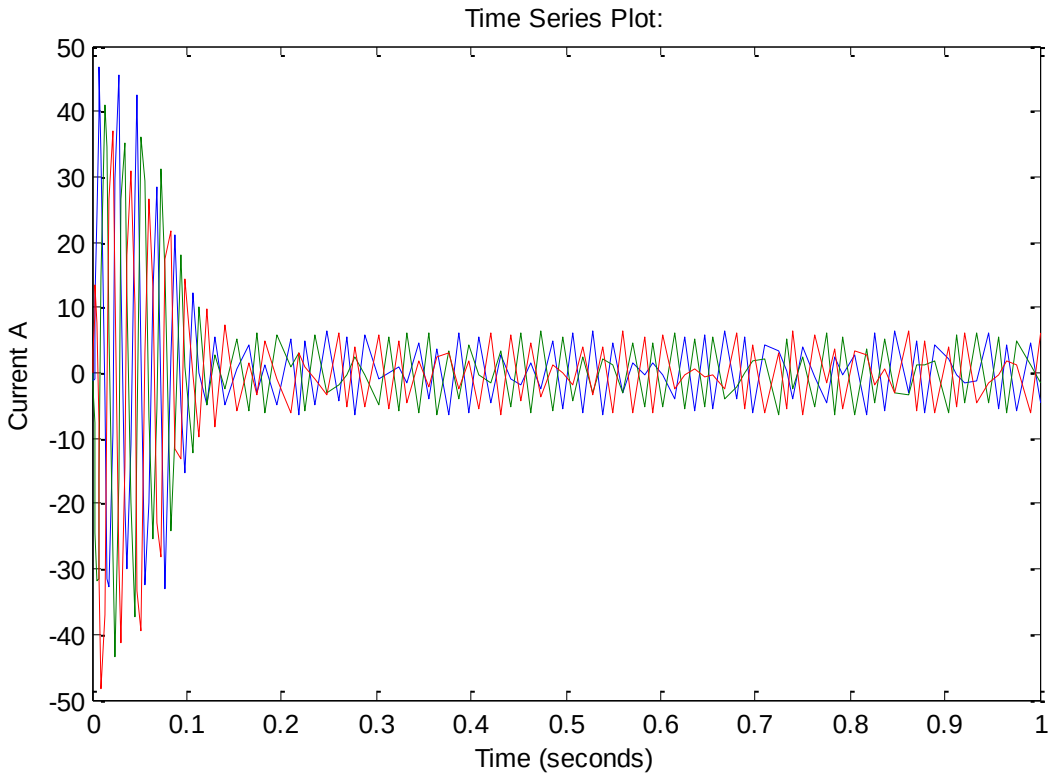


Figure I.20: CurrentResult

19. Conclusion:

In this chapter we have established a brief description of the three-phase asynchronous machine and its operating principle. Then, we presented the mathematical model of the asynchronous machine. Then we proceeded to the Park transformations, which resulted in the simplified models. Next, we have exposed a model of the machine in the dq reference frame linked to the rotating field and we ended this chapter with a study of the voltage inverter where we presented the mathematical model of the inverter then we studied the techniques control (full wave, sine triangle, by hysteresis).

In the next chapter we will develop the scalar control using the asynchronous machine model.

Chapter II:

Scalar control of induction motor

1. Introduction:

The asynchronous machine, due to its low cost and its robustness, is currently the most widely used machine for producing variable speed drives. By its structure, the asynchronous machine has a peculiarity compared to the direct current machine and synchronous type machines. This is because the supply by a single armature causes the same current to create the flux and the torque and thus, variations in torque cause variations in the flux. This coupling makes the model of the asynchronous machine nonlinear and its control is complex. Countless work has been done to develop high-performance controls for the asynchronous machine.

The oldest and most rustic scalar control is for applications requiring only medium static and dynamic performance. Many drives equipped with this control mode are used, in particular for industrial applications [7].

The scalar control of the asynchronous machine consists in imposing on the terminals of its armature, the modulus of the voltage or the current as well as the pulsation. This control mode is the simplest to implement, but also the least efficient, especially at low operating speeds.

In this chapter we first present the principle of the scalar command and then some types of this command

2. The scalar command

Scalar control is relatively simple compared to vector control. It is based on the model of the asynchronous machine in sinusoidal steady state. There are several types of scalar commands depending on whether you are acting on current or voltage. They mainly depend on the topology of the actuator used (voltage or current inverter). The voltage inverter is the most used in small and medium power. The most commonly used command is the U / F law.

The variable frequency power supply is provided by means of a static converter, generally DC-AC. The input source can be a current so.urce type or a voltage source type. At the output of the converter we check [8]:

- Either direct control with stator frequency adjustment and stator voltage réglage or stator current Is

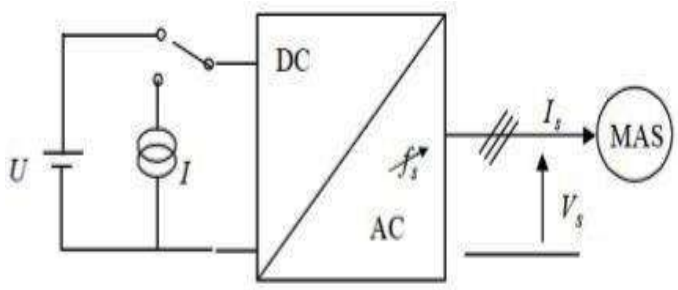


Figure II.1 : setting principle

- Either a control with autopilot which slaves the stator frequency fs and the voltage Vs or the current Is by carrying out $\omega s = \omega r + \omega$ using a mechanical speed sensor

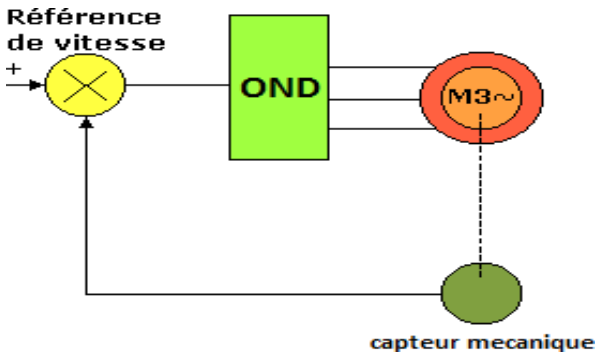


Figure II.2: adjustment with autopilot

We define the slip g as the ratio of the rotor and stator pulses $g = \omega r/\omega s$. The different pulsations are linked by the following autopilot relationship:

$\omega s = \omega r + p\Omega$…….. (1)

3. Modeling of the asynchronous machine in steady state:

To demonstrate the general principles of adjustment of the electromagnetic torque of the asynchronous machine, we will adopt the model of the sinusoidal steady state using the complex vector notation. The simplest speed control, called "U over F", allows machine speed to be varied over a wide range. The equations for calculating torque and predicting operating points are based on the 'steady state' model of the machine. For the equation of the machine model, we assume the assumptions of chapter 2

3.1Steady state asynchronous machine model:

If the supply voltages of the asynchronous machine are balanced three-phase and sinusoidal, they can be written in the following form:

$$\begin{cases} \boldsymbol{V}as = Vs\cos(wst) \\ \boldsymbol{V}bs = Vs\,cos(wst - \frac{2\pi}{3}) \\ \boldsymbol{V}cs = Vs\cos(wst - \frac{4\pi}{3}) \end{cases} \quad \text{.....(2)}$$

By choosing to fix reference $d-q$ to the rotating field at angular speed $\omega S = \frac{d\theta s}{dt}$

equation (2) can be written in the following form:

$$Vds = Uscos(\omega st - \theta s)$$
$$Vqs = Ussin(\omega st - \theta s) \quad \text{.....(3)}$$

By using the complex notation$X = Xd + Xq$, the electrical quantities can be put in the form:

$$\overline{V}s = Vds + jVqs = Rs(Ids + jIqs) + \frac{d}{dt}(\phi ds + j\phi qs) - ws(\phi qs - j\phi ds) \quad \text{.....(4)}$$

$$\overline{V}r = Vdr + jVqr = Rr(Idr + jIqr) + \frac{d}{dt}(\phi dr + j\phi qr) - wr(\phi qr - j\phi dr) \quad \text{.....(5)}$$

From :

$$\overline{V}s = Rs\overline{I}s + \frac{d}{dt}\overline{\phi}s \ + jws\overline{\phi}s \ldots\ldots(6)$$

$$\overline{V}r = Rr\overline{I}r + \frac{d}{dt}\overline{\phi}r \ + jwr\overline{\phi}r \ldots\ldots(7)$$

In steady state we have:

$$\frac{d}{dt}\overline{\phi}r = 0 \quad , \qquad \frac{d}{dt}\overline{\phi}s = 0$$

Consequently, equation (6) and (7) becomes:

$$\overline{V}s = Rs\overline{I}s + jws\overline{\phi}s$$

$$\overline{V}r = Rr\overline{I}r + jwr\overline{\phi r} = 0 \quad \ldots\ldots(8)$$

Magnetic flux equation:

$$\begin{cases} \overline{\phi}s = Ls\overline{I}s + Msr\overline{I}s \\ \overline{\phi}r = Lr\overline{I}r + Msr\overline{I}r \end{cases} \ldots\ldots(9)$$

By using equations (8) and (9) we can write the electrical quantities:

$$\begin{cases} Vs = Rs\overline{I}s + jwsls\overline{I}s + jwsMsr\overline{I}r \\ 0 = \frac{Rr}{g}\overline{I}r + jwslr\overline{I}r + jwsMsr\overline{I}s \end{cases} \ldots\ldots(10)$$

With : $Wr \ = \ gWs$

From these two equations (9) and (10), we deduce:

$$\overline{I}r = -j\frac{MsrWr}{Rr + jLrwr}\overline{I}s \ldots\ldots(11)$$

$$\overline{\phi s} = Ls\frac{Rr + jwr\sigma lr}{Rr + jLrwr}\overline{I}s \ldots\ldots(12)$$

3.2 Equivalent electrical diagram of the MAS:

Equations (10) can be represented by the following equivalent electrical diagram[9]:

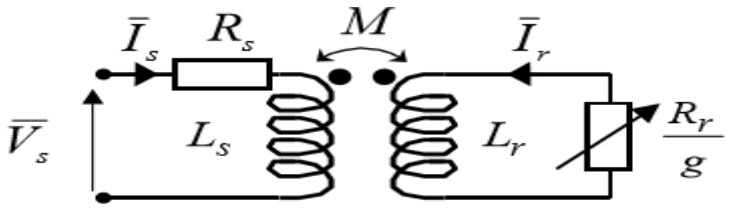


Figure II.3 : Steady state phase diagram

We can bring this diagram to the stator with the magnetic leaks totaled at the rotor ($Nr\omega s$):

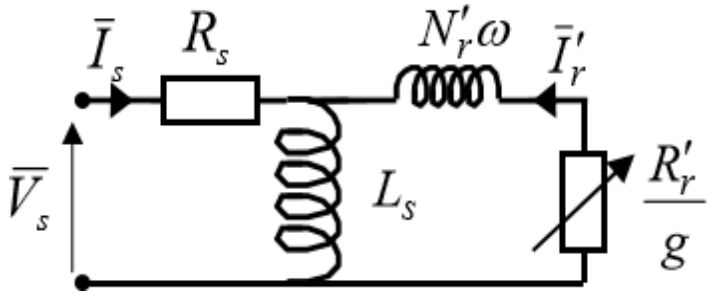


Figure II.4: diagram equivalent to the stator with the magnetic leaks totaled at the rotor

With:

$$\begin{cases} Nr = \sigma Lr = Lr - \frac{M}{Ls} \\ N'r = Nr\left(\frac{Ls}{Lm}\right)^2 \\ R'r = Rr\left(\frac{Ls}{Lm}\right)^2 \\ I'r = \left(\frac{Lm}{Ls}\right) Ir \end{cases} \quad \text{……(13)}$$

3.3 Couple Explanation:

The electromagnetic torque is given by:

$$Ce = \frac{3}{2}\ pLm\Im m(IsIr*) \text{……..…..(14)}$$

rom equations (11), (12) we obtain:

$$Ce = 3p\left(\frac{lm}{ls}\right)^2 \phi s^2 \frac{wr}{Rr\left(1+\left(\frac{\sigma wrlr}{Rr}\right)^2\right)} \quad \text{.....(15)}$$

This expression shows that in order to control the electromagnetic torque of the machine, it is necessary to control the stator flux donc (therefore the current) and the rotor pulsation Grand (Quantity which is not directly accessible).

When the flux modulus is constant, the torque depends only on ωr, The variation of electromagnetic torque as a function of the rotor pulsation is represented by the figure(5) :

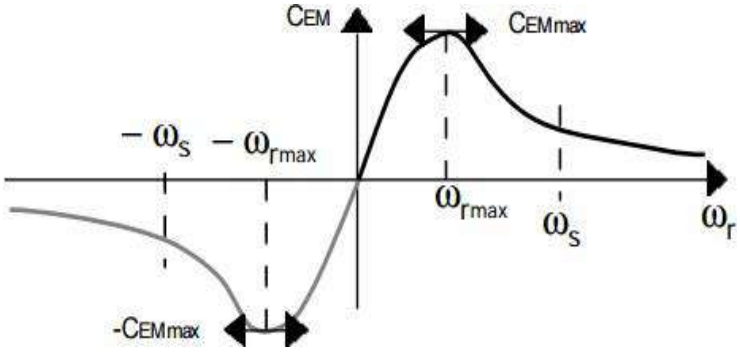


Figure II.5: variation of torque as a function of pulsation

The curve presents a maximum $Cemax$for the noted pulsation $Wrmax$:

$$Wrmax = \frac{Rr}{\sigma Lr} \quad \text{.....(16)}$$

And :

$$Cemax = 3p\left(\frac{lm}{ls}\right)^2 \phi s^2 \frac{1}{2\sigma lr} \quad \text{.....(17)}$$

With: σ is the dispersion coefficient of BLONDEL

We can see that the maximum torque is proportional or care of the stator flux.In the permanent mode the slip is sufficiently weak, the electromagnetic couple is proportional to the pulsation ωr:

$$Ce = \alpha\omega r \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (18)$$

With :

$$\alpha = 3p\left(\frac{lm}{ls}\right)^2 \phi s^2 \frac{1}{Rr} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (19)$$

From the knowledge of the law $Ce = f(\omega r)$ different control strategies for asynchronous machines have been developed.

4. Scalar voltage control :

4.1 The command law:

Let study the behavior of the asynchronous machine in steady state, the supply voltage is expressed by the relation[10]:

$$\overline{V}s = \frac{Rs}{1 + j\frac{Lr}{Rr}Wr}\left[\left(1 - \sigma\frac{lslr}{RsRr}\,WrWs\right) + j\left(\frac{Lr}{Rr}\,Wr + \frac{Ls}{Rs}\,ws\right)\right]\overline{Is} \ldots\ldots\ldots\ldots\ldots (20)$$

By deferring (12) in (20), the stator voltage modulus is then:

$$Vs = \phi s\frac{Rs}{ls}\sqrt{\frac{\left(1 - \sigma\frac{LsLr}{RsRr}\,WrWs\right)^2 + \left(\frac{Lr}{Rr}Wr + \frac{Ls}{Rs}Ws\right)^2}{1 + \left(\frac{\sigma Wrlr}{Rr}\right)^2}} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (21)$$

This relationship remains valid between the rms values of stator voltages and fluxes. It is the principle of constant flux control laws for machines supplied with voltage. We choose to maintain, if possible, the flux at its nominal value. Two control modes are possible:

- A command by controlling the stator frequency ωs and the stator current or voltage.
- A control with autopilot and control of the pulsation of the rotor currents The control law (2.21) makes it possible to maintain the flow constant, but it is too complex to be used without powerful means of calculation. It must be simplified
  - If the rotor pulsation is very weak (ωr = 0), then:

$$Vs = \phi s\omega s\sqrt{1+\left(\frac{Rs}{wsLs}\right)^2} \ldots\ldots\ldots (22)$$

  - If, moreover, Rs is negligible, then:

$$Vs = \phi s\omega s \ldots\ldots\ldots (23)$$

What characterizes a control law in $Vs\ /\ fs$ = constant.

If the stator frequency fs decreases, then the decreasing leakage flux reactances, on the other hand the ohmic voltage drop cannot be neglected. One then compensates by adding a term of tension $V0$

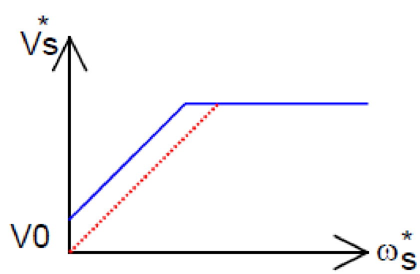


Figure II.6: characteristic $(V\ /\ f)$

A corrective term is often added to take into account the rotor pulsation:

$$Vs = \phi s(\omega s + k\omega r) \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (24)$$

With : $k = \frac{RsLr}{RrLs}$

To improve the behavior in low frequency one can use the following law:

$$Vs = \phi s\left(\omega s\sqrt{1 + \left(\frac{Rs}{WsLs}\right)^2} + KWr\right) \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (25)$$

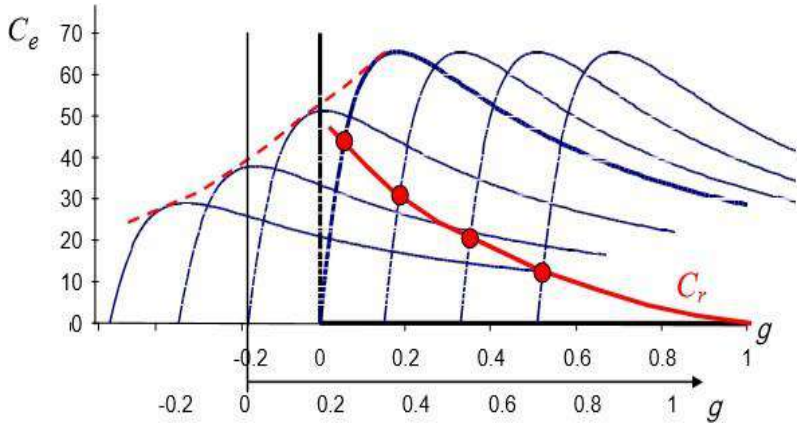
The preceding laws ensure that the modulus of the flux is maintained in a steady state in a transient state; they cause flux oscillations and therefore of the torque.

4.2 V / f control of the asynchronous machine:

Its principle is to keep $V / f = constant$, which amounts to keeping the flow constant. The torque is controlled by action on the slip. Indeed, according to the model established in steady state, the maximum torque is written

$$Cemax = \frac{3p}{2N'r}\left(\frac{Vs}{Ws}\right)^2 \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (26)$$

It can be seen that the torque is directly proportional to the square of the ratio of the voltage to the stator frequency. By maintaining this constant ratio and by adjusting the stator frequency, the curve of the electromagnetic torque (in quasi-static mode) of the asynchronous machine is shifted as shown in figure (7).



- Corresponds to the operating point `` intersection between the curve of the load torque of the motor "

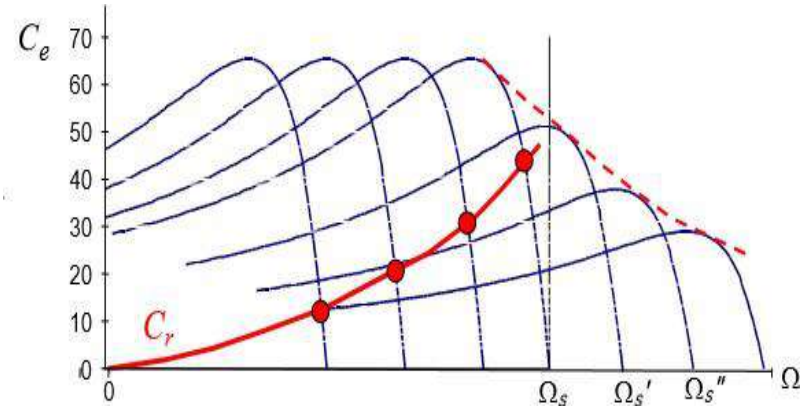


Figure II.7: displacement of the torque-slip characteristic as a function of the supply frequency

In fact, keeping the ratio constant is like keeping the flow constant. When the voltage reaches its maximum value, we then begin to decrease this ratio, which causes a reduction in the torque that the machine can produce. We are in a “defluxing” regime. This speed makes it possible to exceed the nominal speed of the machine, it is therefore also called the overspeed speed figure (8).

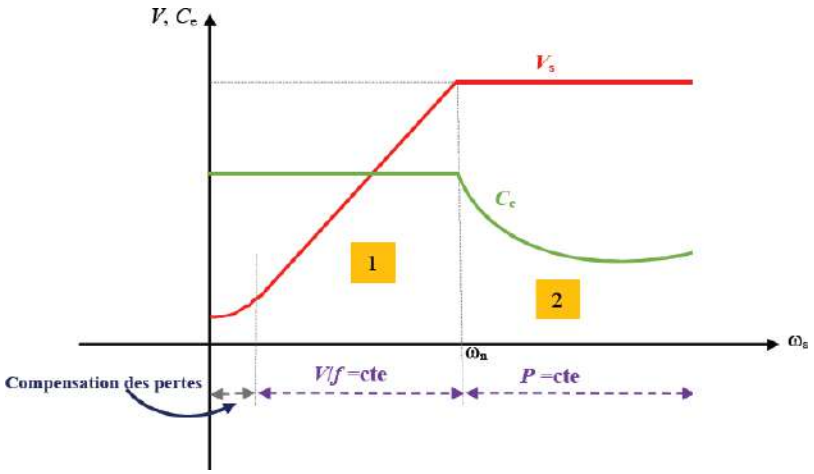


Figure II.8: Autopilot and scalar control. Operating modes.

The block diagram of this control is shown in figure (9). The asynchronous machine is supplied by a voltage inverter. The controller sets the frequency from the speed error and the voltage from the V / f law.

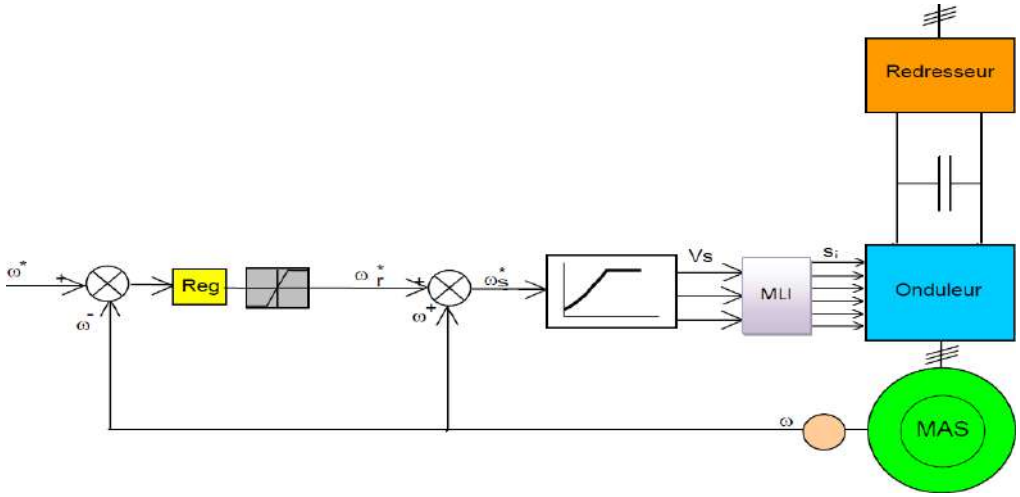


Figure II.9: Scalar control with voltage regulated voltage inverter

This control diagram shows how to regulate the speed of the machine by restoring the stator pulsation.

5. Scalar Current Control :

From equation (12), we define the modulus of the stator current by the following relation:

$$\bar{I}s = \frac{\bar{\phi}s}{Ls}\sqrt{\frac{1+\left(\frac{WrLr}{Rr}\right)^2}{1+\left(\frac{\sigma WrLr}{Rr}\right)^2}} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots..(27)$$

The characteristics$Is = f\ (\omega r)$ àϕs with constant are indicated in figure (10). For ϕs kept constant the stator current Is only depends on Wr:

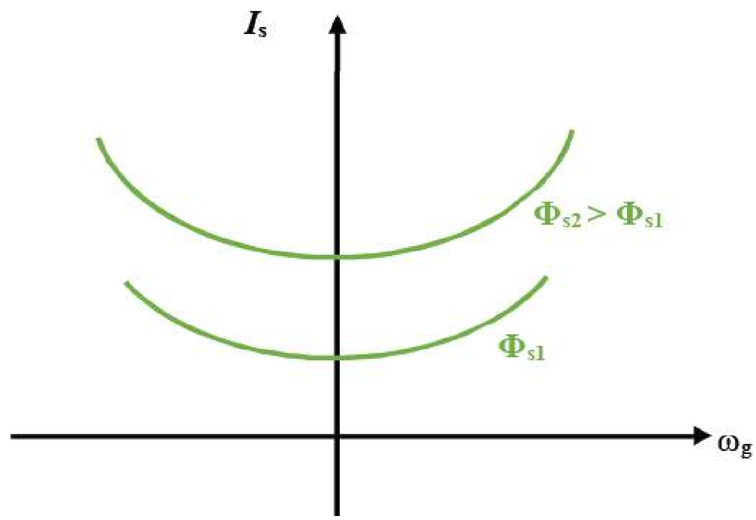


Figure II.10: characteristic Is according to ωr

For the control of this current we use:

- A voltage inverter controlled by currents
- A current switch

5.1 Control principle of the voltage inverter controlled by current:

From the speed reference the three reference currents$i'as\ , i'bs, i'cs$are computed by relation (28) and expressed by:

$$\begin{aligned} \boldsymbol{i}'as &= \sqrt{2}i's\ \sin(wst) \\ \boldsymbol{i}'bs &= \sqrt{2}i's\ \sin(wst - \frac{2\pi}{3}) \\ \boldsymbol{i}'cs &= \sqrt{2}i's\ \sin(wst + \frac{2\pi}{3}) \end{aligned} \quad \text{.....(28)}$$

After the current generation they are compared or measured stilt current the output of this comparison is the input of the regulators a hysteresis.The latter generates the inverter's control sequence.

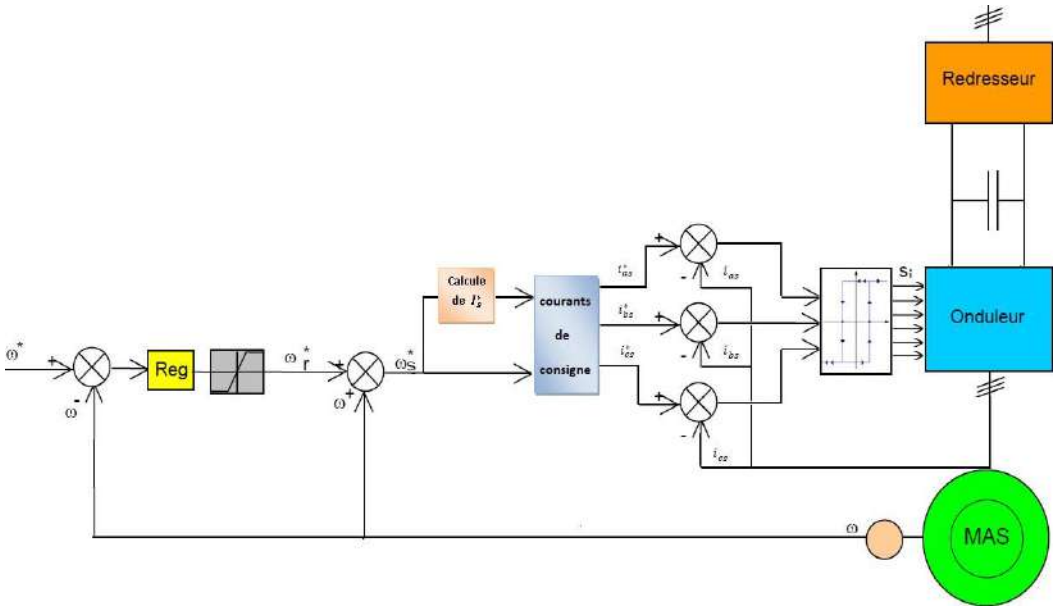


Figure II.11: Scalar control with voltage inverter regulated while running

5.2 Power switch control principle

The difference with the previous command, is that it is a current inverter (switch) that is used, Figure (12).Currents are directly imposed in the phases of the machine.The frequency of the fundamental is calculated in the same way.The value of the Id (continuous current) shelf current is equal to a constant close to the effective value of the current imposed Is.It is regulated using a controlled straightener bridge.The device is more complex than a scalar control of tension.

$$Is = \sqrt{6}/\pi \;\; Id$$

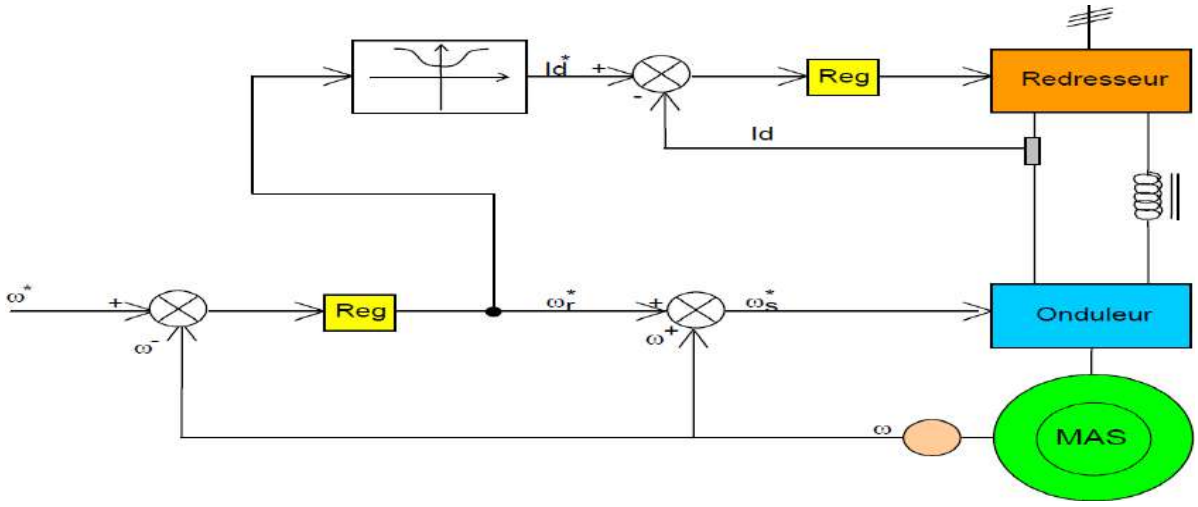


Figure II.12: Scalar control with current inverter

6. Simulation ofMAS with Inverter and Scalar Control :

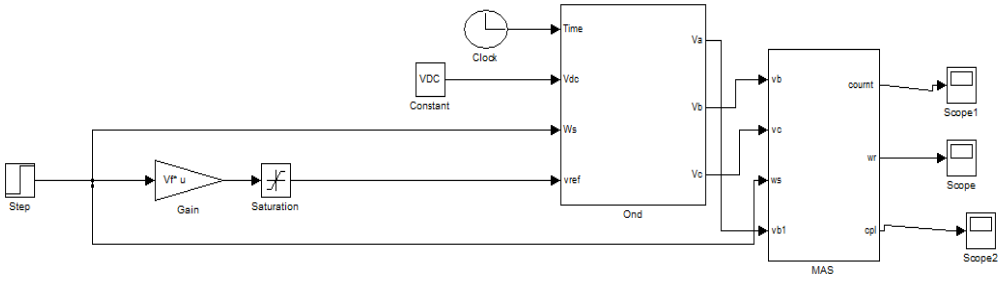


Figure II.13:MAS with Inverter and Scalar Control model

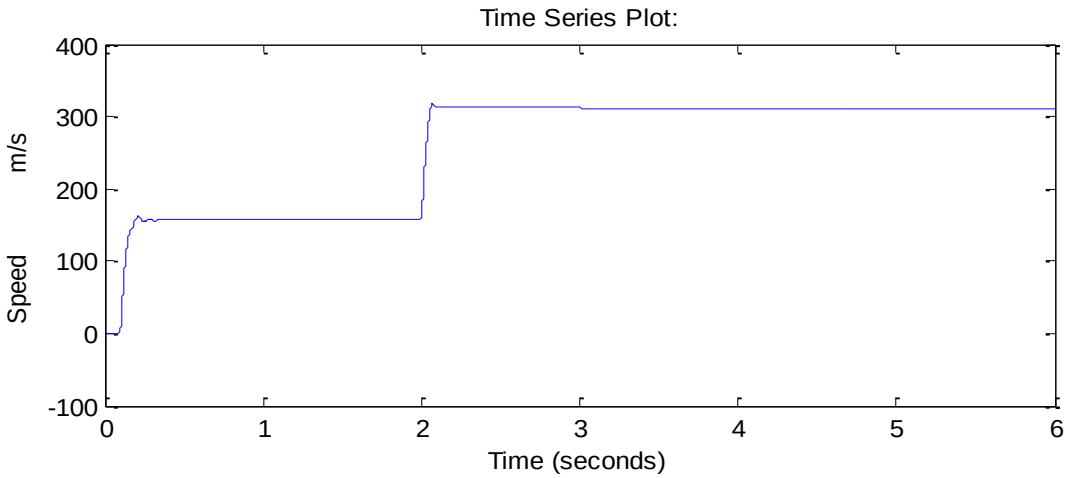


Figure II.14:MAS with Inverter and Scalar Control's speed result

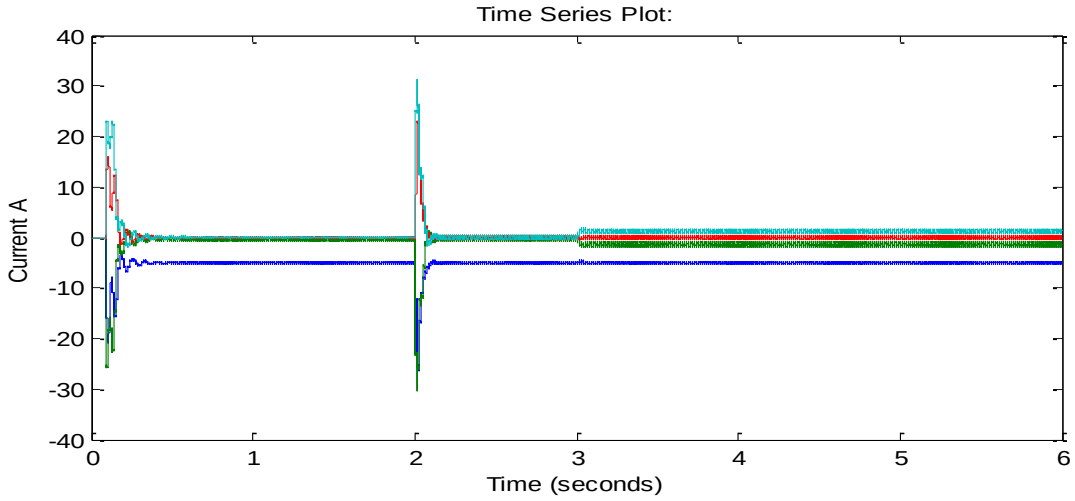


Figure II.15:MAS with Inverter and Scalar Control's current result

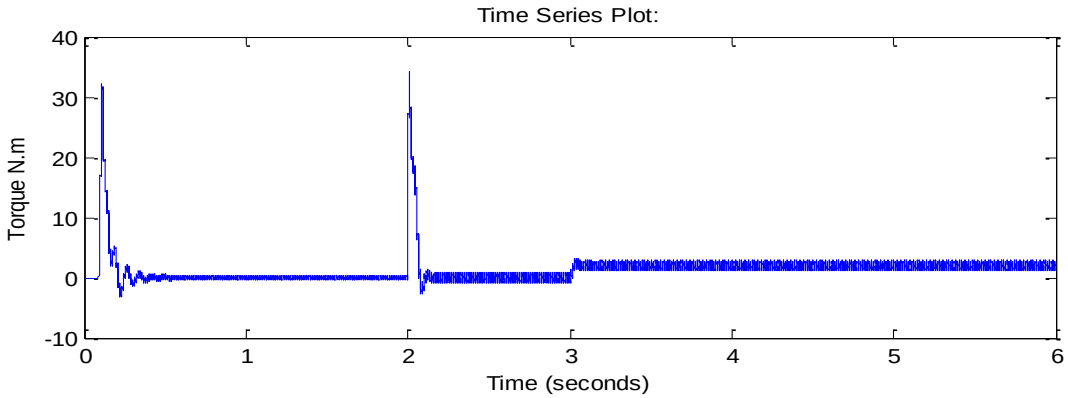


Figure II.16:MAS with Inverter and Scalar Control's torque result

7. Regulator PI:

Mechanical speed regulation is carried out using the IP regulator.The IP regulator controls the speed of the MAS and eliminates static error in position.Indeed the objective of the regulation and that the rotor reaches the reference speed in a given time.The desired dynamics of the closed-loop regulated system also ensures its stability[11].

The diagram of the regulation of speed is represented by the following figure (13):

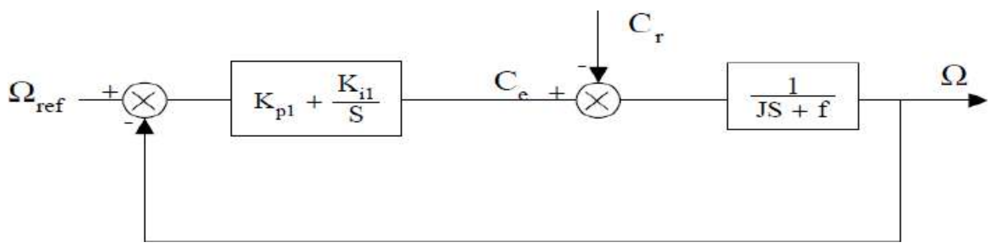


Figure II.17: speed control block diagram

The open loop equation without a regulator is defined by the equation:

$$\Omega(s) = \frac{1}{js + f}\,(Ce - Cr) \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (29)$$

In the case of the use of a conventional IP regulator, Ω it is then written:

$$\Omega = \frac{1}{js+f}\,(kp + \frac{Ki}{s})(\Omega ref - \Omega) - \frac{1}{js+f}Cr \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (30)$$

Is :

$$\Omega = \frac{KpS+Ki}{js^2+(kp+f)s+Ki}\Omega ref - \frac{S}{js^2+(kp+f)s+Ki}Cr \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (31)$$

The characteristic equation (pursuit and regulation) has a dynamic of the second order.In pursuit ($Cr = 0$) we will write:

$$\frac{\Omega}{\Omega ref}=\frac{\frac{Kp}{Ki}s+1}{\frac{j}{Ki}s^2+\frac{Kp+f}{Ki}s+1} \quad \text{.....(32)}$$

This transfer function has a 2nd order dynamic, including the standardized canonical form:

$$G(s)=\frac{1}{\frac{1}{Wn}s^2+\frac{2\xi}{Wn}s+1} \quad \text{.....(33)}$$

By comparison we then get:

$$\begin{cases} \dfrac{J}{Ki}=\dfrac{1}{Wn} \\ \dfrac{Kp+f}{Ki}=\dfrac{2\xi}{Wn} \end{cases} \quad \text{.....(34)}$$

With:

ξ:Slip Coefficient

$t_{r5\%}$: Response time (s)

Wn: Clean oscillation frequency ($Wn=\frac{3}{\xi*\mathrm{tr}5\%}$)

$$\begin{cases} Ki=Wn*J \\ Kp=\frac{2\xi*Ki}{wn}-f \end{cases} \quad \text{.....(35)}$$

8. Simulation of MAS with Scalar Control & Regulator PI:

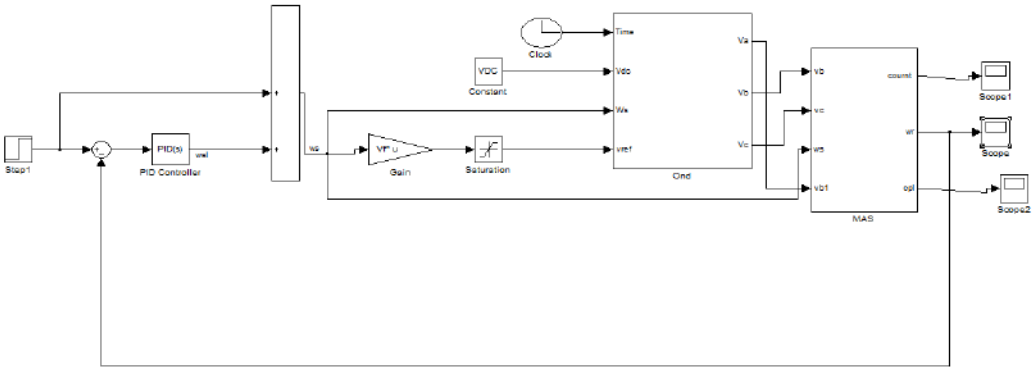


Figure II.18: MAS with Scalar Control & Regulator PI's model

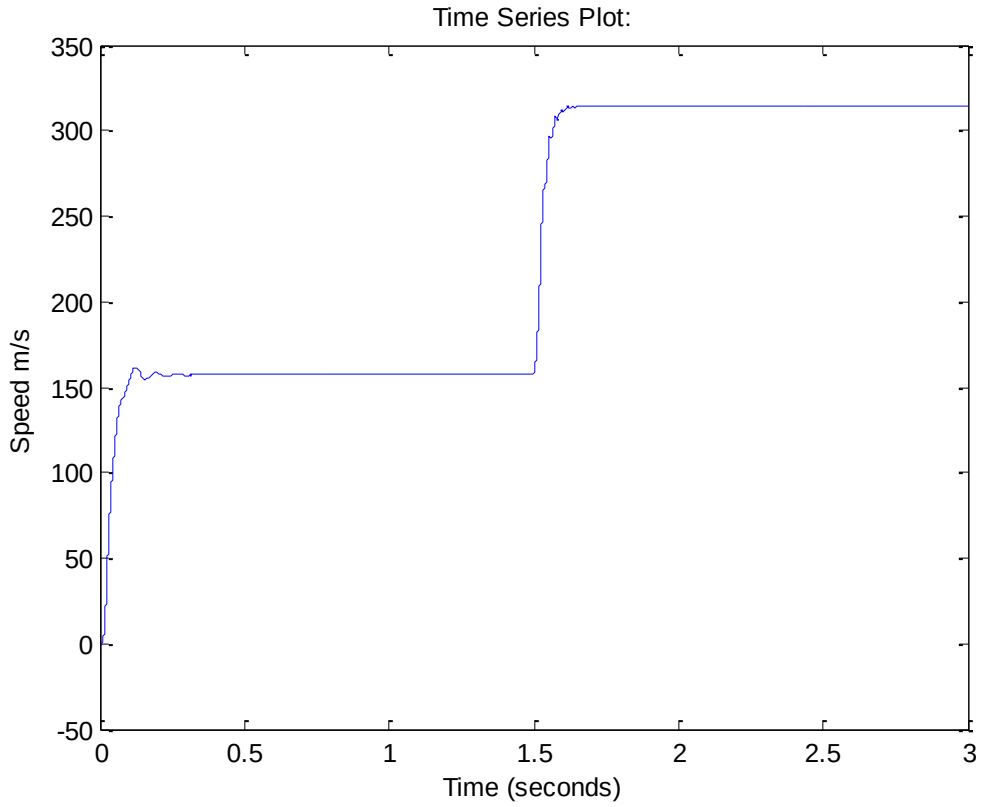


Figure II.19: MAS with Scalar Control & Regulator PI's speed result

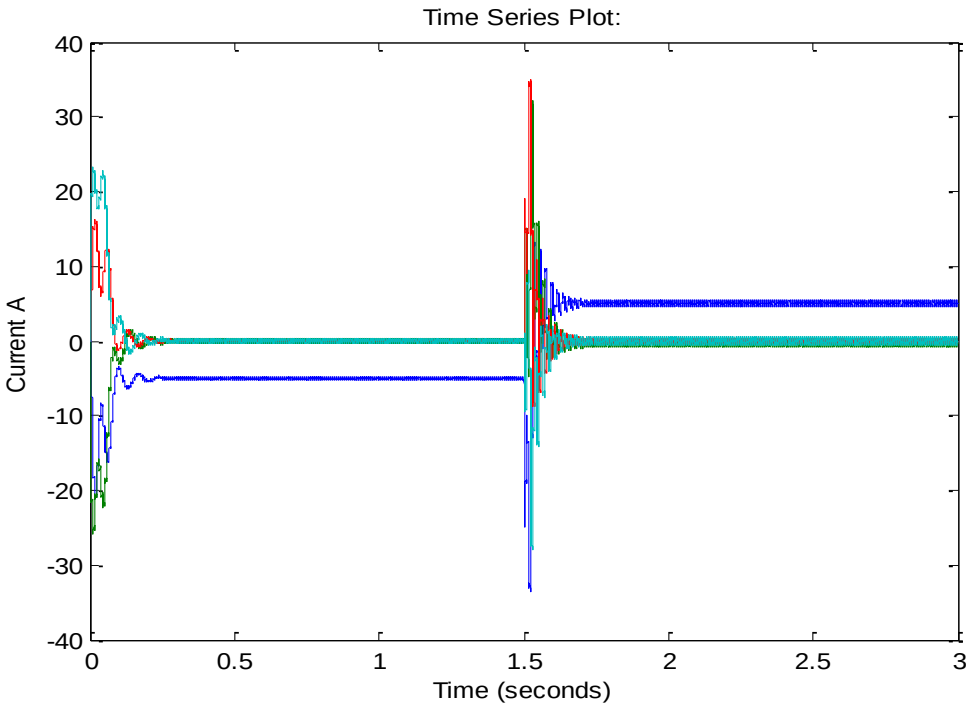


Figure II.20: MAS with Scalar Control & Regulator PI's current result

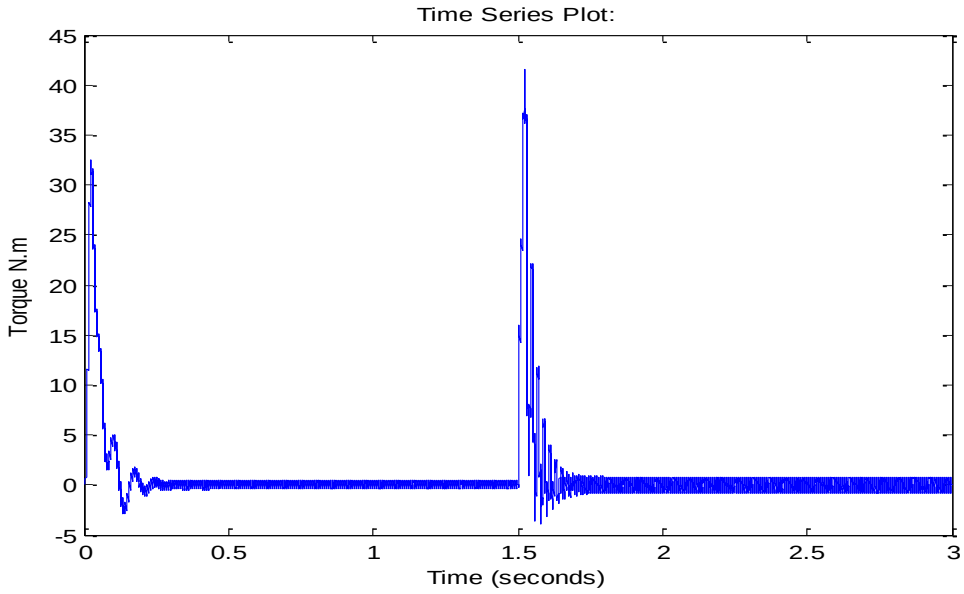


Figure II.21: MAS with Scalar Control & Regulator PI's torque result

9. Conclusion :

In this chapter we presented the basic concepts of scalar control and more specifically the scalar control by indirect control of the flow.The scalar control, which is a simple command, can be carried out with a voltage control or by running from the inverter.From the theoretical study it can be concluded that the scalar control linearized the flow of the machine at a certain point of operation and it was also found that this control strategy presents a major disadvantage in the base neck speeds of the importance of the statoric voltage drop.The regulation of the mechanical speed of the MAS with a PI regulator was presented.

Chapter III:

Fuzzy Logic Controller

1. Introduction:

Fuzzy logic is a concept which helps computers in making decisions in a way which resembles human behaviors. It helps industry in increasing productivity, creates the opportunity to make production more convenient and most importantly it helps industries in economical terms.

The concept of fuzzy logic was first proposed by Professor Lotfi A. Zadeh in 1965. It was presented in one of his research papers under the name Fuzzy logic or Fuzzy sets [20].

2. Fuz.zy Logic History:

When Aristotle and his predecessors devised their theories of logic and mathematics, they came up with the so-called Law of the Excluded Middle, which states that every proposition must either be true or false. Grass is either green or not green; it clearly cannot be both green and not green. But not everyone agreed, and Plato indicated there was a third region, beyond true and false, where these opposites "tumbled about."

In the Aristotelian world view, logic dealt with two values. In the 19th century, George Boole created a system of algebra and set theory that could deal mathematically with such two-valued logic, mapping true and false to 1 and 0, respectively. Then in the early 20th century, Jan Lukasiewicz proposed a three-valued logic (true, possible, false), which never gained wide acceptance.

In 1965, Lotfi A. Zadeh of the University of California at Berkeley published "Fuzzy Sets," which laid out the mathematics of fuzzy set theory and, by extension, fuzzy logic. Zadeh had observed that conventional computer logic couldn't manipulate data that represented subjective or vague ideas, so he created fuzzy logic to allow computers to determine the distinctions among data with shades of gray, similar to the process of human reasoning.

Although, the technology was introduced in the U.S. . U.S. and European scientist and researchers largely ignored it for years, perhaps because of its unconventional name. They refused to take seriously something that sounded so childlike. Some mathematicians argued that fuzzy logic was merely probability in disguise. But fuzzy logic was readily accepted in Japan, China and other Asian countries. The greatest number of fuzzy researchers today is found in

China, with over 10,000 scientists. Japan, though considered at the leading edge of fuzzy studies, has fewer people engaged in fuzzy research [21].

3. Why do we use fuzzy logic?

In all domains, from marketing to engineering, we need to deal with uncertainty in order to solve problems and make decisions. Fuzzy logic is a system that helps us do all of these things better by programming machines to think as humans do. It may not give us accurate reasoning, but it provides us with acceptable reasoning.

Fuzzy logic accepts that we are unable to provide a precise number for many uncertainties in life. Information is often imperfect, and data has errors. By replacing binary either/or options with if/then possibilities, fuzzy logic expands choices and considers subjective attributes.

The basis of fuzzy logic is also the basis of human communication. Because fuzzy logic is built on structures of qualitative description used in everyday language what's known as linguistic variables, such as "short," "medium" or "tall" in the height example, it is easier to understand. Moreover, because the mathematical concepts behind fuzzy reasoning are relatively simple, it represents an intuitive approach without far-reaching complexity.

A fuzzy logic system can be defined as the nonlinear mapping of an input dataset to scalar output data, a consequence of Zadeh's initial algebra. Such a system consists of four main parts: fuzzifier, a set of rules, an inference engine and a defuzzifier[22].

4. Fuzzy Logic Architecture :

The fuzzy logic architecture consists of four main parts:

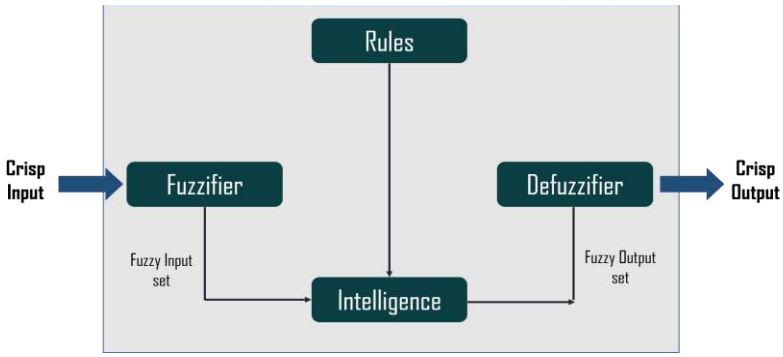


Figure III.1: Fuzzy Logic architecture.

4.1 Rules :

It contains all the rules and the IF-THEN conditions offered by the experts to control the decision making system. The recent update in the fuzzy theory provides different effective methods for the design and tuning of fuzzy controllers. Usually, these developments reduce the number of fuzzy rules.

4.2 fuzzification:

This step converts inputs or the crisp numbers into fuzzy sets. You can measure the crisp inputs by sensors and pass them into the **control system** for further processing. It splits the input signal into five steps such as:

LP	X is Large Positive
MP	X is Medium Positive
S	Small
MN	X is Large Negative
LN	X is Large Negative

Table III.1: Fuzzification table

4.3 Inference Engine :

It determines the degree of match between fuzzy input and the rules. According to the input field, it will decide the rules that are to be fired. Combining the fired rules, form the control actions.

4.4 Defuzzification:

The Defuzzification process converts the fuzzy sets into a crisp value. There are different types of techniques available, and you need to select the best suited one with an expert system [23].

So, this was about the architecture of fuzzy logic. Now, let's understand the membership function.

5. Membership function:

Membership functions allow you to quantify linguistic term and represent a fuzzy set graphically. A **membership function** for a fuzzy set A on the universe of discourse X is defined as $\mu A: X \rightarrow [0, 1]$.

Here, each element of X is mapped to a value between 0 and 1. It is called **membership value** or **degree of membership**. It quantifies the degree of membership of the element in X to the fuzzy set A.

- X axis represents the universe of discourse.
- Y axis represents the degrees of membership in the [0, 1] interval.

There can be multiple membership functions applicable to fuzzify a numerical value. Simple membership functions are used as use of complex functions does not add more precision in the output.

All membership functions for **LP, MP, S, MN,** and **LN** are shown as below :

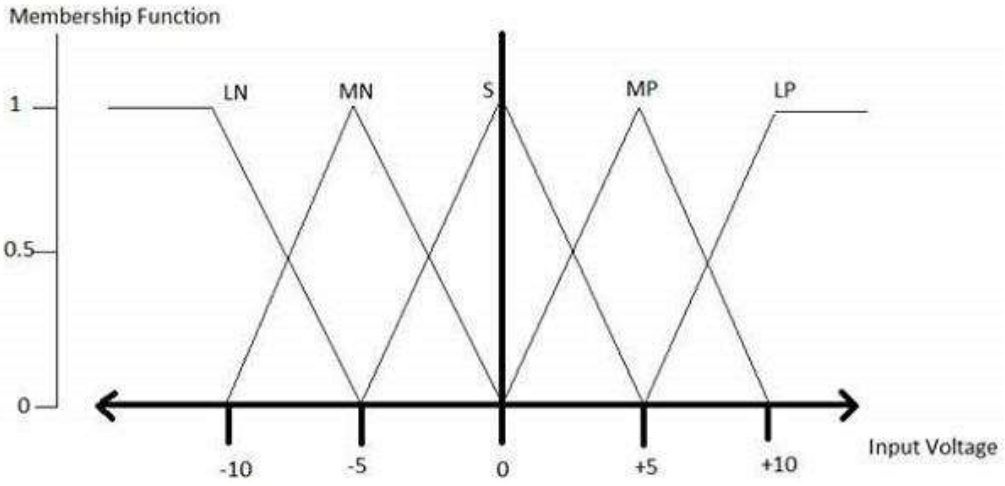


Figure III.2: Fuzzy Logic Membership Function

The triangular membership function shapes are most common among various other membership function shapes such as trapezoidal, singleton, and Gaussian.

Here, the input to 5 level fuzzifier varies from -10 volts to +10 volts. Hence the corresponding output also changes [24].

6. Fuzzy logic applications:

Fuzzy logic is extremely useful for many people involved in research and development including engineers (electrical, mechanical, civil, chemical, aerospace, agricultural, biomedical, computer, environmental, geological, industrial, and mechatronics), mathematicians, computer software developers and researchers, natural scientists (biology, chemistry, earth science, and physics), medical researchers, social scientists (economics, management, political science, and psychology), public policy analysts, business analysts, and jurists.

Indeed, the applications of fuzzy logic, once thought to be an obscure mathematical curiosity, can be found in many engineering and scientific works. Fuzzy logic has been used in numerous applications such as facial pattern recognition, air conditioners, washing machines, vacuum cleaners, antiskid braking systems, transmission systems, control of subway systems and unmanned helicopters, knowledge-based systems for multi-objective optimization of power

systems, weather forecasting systems, models for new product pricing or project risk assessment, medical diagnosis and treatment plans, and stock trading. Fuzzy logic has been successfully used in numerous fields such as control systems engineering, image processing, power engineering, industrial automation, robotics, consumer electronics, and optimization. This branch of mathematics has instilled new life into scientific fields that have been dormant for a long time [25].

7. Advantages and Disadvantages of Fuzzy Logic:

7.1 Advantages:

The benefits of using Fuzzy Logic systems are as follows:

- It is a robust system where no precise inputs are required.
- These systems are able to accommodate several types of inputs including vague, distorted or imprecise data.
- In case the feedback sensor stops working, you can reprogram it according to the situation.
- The Fuzzy Logic algorithms can be coded using less data, so they do not occupy a huge memory space.
- As it resembles human reasoning, these systems are able to solve complex problems where ambiguous inputs are available and take decisions accordingly.
- These systems are flexible and the rules can be modified.
- The structure of Fuzzy Logic Systems is easy and understandable.
- Fuzzy logic is widely used for commercial and practical purposes.
- It helps you to control machines and consumer products.
- It helps you to deal with the uncertainty in engineering.
- You can easily modify it to improve or alter system performance.
- Inexpensive sensors can be used which helps you to keep the overall system cost and complexity low.

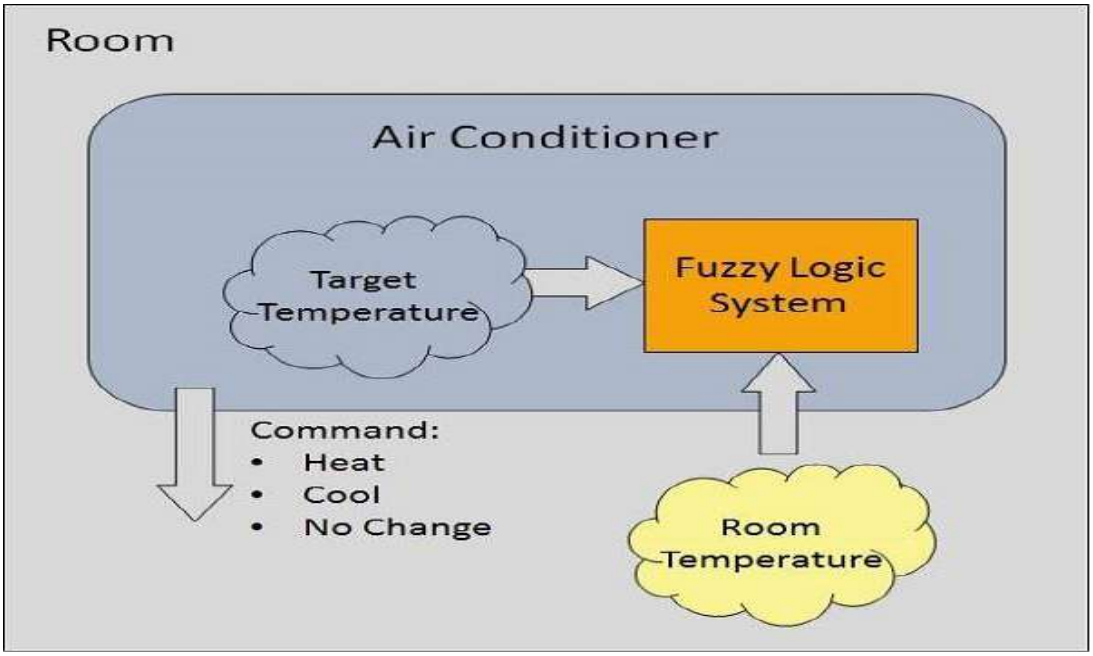
7.2 Disadvantages:

Let us look at the drawbacks of Fuzzy Logic systems:

- The accuracy of these systems is compromised as the system mostly works on inaccurate data and inputs.
- There is no single systematic approach to solve a problem using Fuzzy Logic. As a result, many solutions arise for a particular problem, leading to confusion.
- A major drawback of Fuzzy Logic control systems is that they are completely dependent on human knowledge and expertise.
- You have to regularly update the rules of a Fuzzy Logic control system.
- These systems cannot recognize machine learning or neural networks.
- The systems require a lot of testing for validation and verification.
- Fuzzy logic is not always accurate. So the results are perceived based on assumptions and may not be widely accepted.
- Validation and Verification of a fuzzy knowledge based system needs extensive testing with hardware.
- Setting exact, fuzzy rules and, membership functions is a difficult task.
- At times, the fuzzy logic is confused with probability theory[26].

8. Example of a Fuzzy Logic System:

Let us consider an air conditioning system with 5-level fuzzy logic system. This system adjusts the temperature of air conditioner by comparing the room temperature and the target temperature value.



ALGORITHM

- Define linguistic variables and terms.
- Construct membership functions for them.
- Construct knowledge base of rules.
- Convert crisp data into fuzzy data sets using membership functions. (fuzzification)
- Evaluate rules in the rule base. (interface engine)
- Combine results from each rule. (interface engine)
- Convert output data into non-fuzzy values. (defuzzification)

LOGIC DEVELOPMENT

Step 1: Define linguistic variables and terms

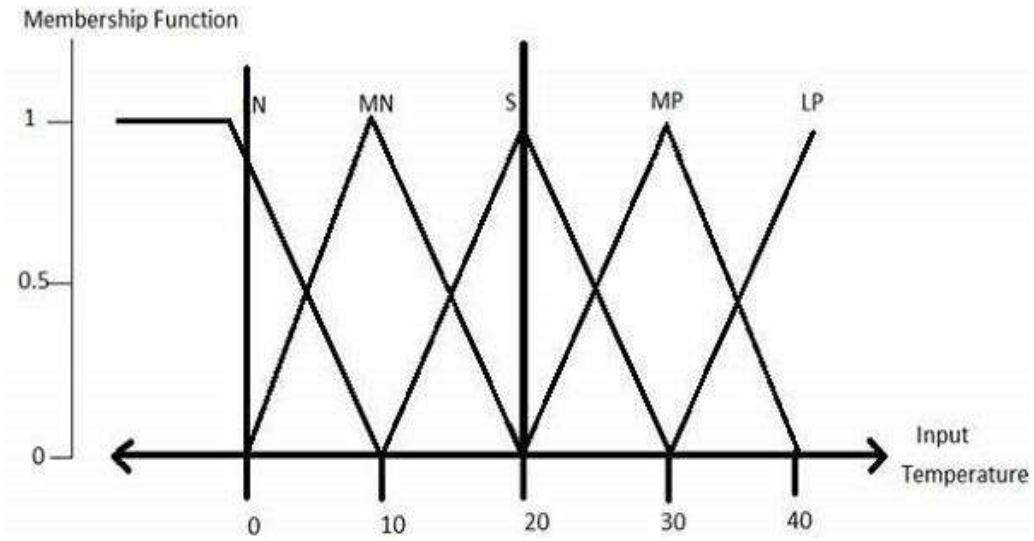
Linguistic variables are input and output variables in the form of simple words or sentences. For room temperature, cold, warm, hot, etc., are linguistic terms.

Temperature (t) = {very-cold, cold, warm, very-warm, hot}

Every member of this set is a linguistic term and it can cover some portion of overall temperature values.

Step 2: Construct membership functions for them

The membership functions of temperature variable are as shown:



Step3: Construct knowledge base rules

Create a matrix of room temperature values versus target temperature values that an air conditioning system is expected to provide.

ROOMTEMP. /TARGET	VERY COLD	COLD	WARM	HOT	VERY HOT
Very Cold	No Change	Heat	Heat	Heat	Heat
Cold	Cool	No Change	Heat	Heat	Heat
Warm	Cool	Cool	No Change	Heat	Heat
Hot	Cool	Cool	Cool	No Change	Heat
Very Hot	Cool	Cool	Cool	Cool	No Change

Build a set of rules into the knowledge base in the form of IF-THEN-ELSE structures.

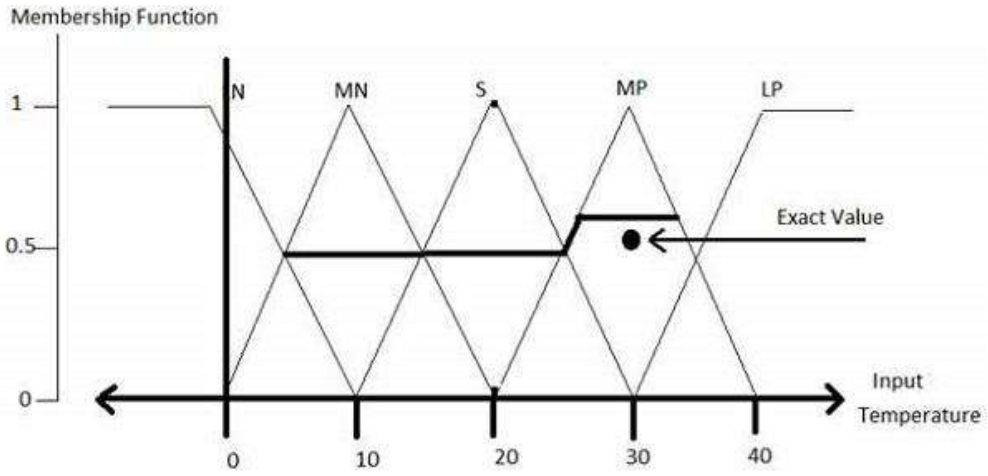
SR. NO.	Condition	Action
1	IF temperature=(Cold OR Very_Cold) AND target=Warm THEN	Heat
2	IF temperature=(Hot OR Very_Hot) AND target=Warm THEN	Cool
3	IF (temperature=Warm) AND (target=Warm) THEN	No_Change

Step 4 : Obtainfuzzy value

Fuzzy set operations perform evaluation of rules. The operations used for OR and AND are Max and Min respectively. Combine all results of evaluation to form a final result. This result is a fuzzy value.

Step 5: Perform defuzzification:

Defuzzification is then performed according to membership function for output variable [27].



MATLAB Fuzzy Logic Toolbox

1. Introduction:

The MATLAB Fuzzy Toolbox facilitates the development of Fuzzy Logic System using :

1. Graphical User Interface (GUI) tools
2. Command Line Functionality

The Tool can be used for building :

1. Fuzzy Expert Systems
2. Adaptive Neuro-Fuzzy Inference Systems (ANFIS)

General Vision

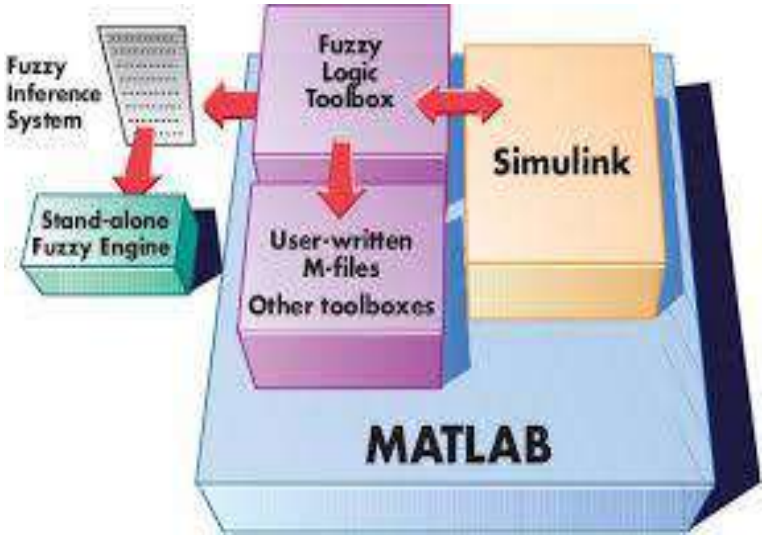


Figure III.3: Fuzzy Logic Toolbox in MatLab

2. Graphical User Interface (GUI) Tools:

There are five primary GUI tools for building , editing and observing Fuzzy Inference Systems in the Fuzzy Logic Toolbox:

- Fuzzy Inference System (FIS) Editor
- Membership Function Editor
- Rule Editor
- Rule Viewer
- Surface Viewer

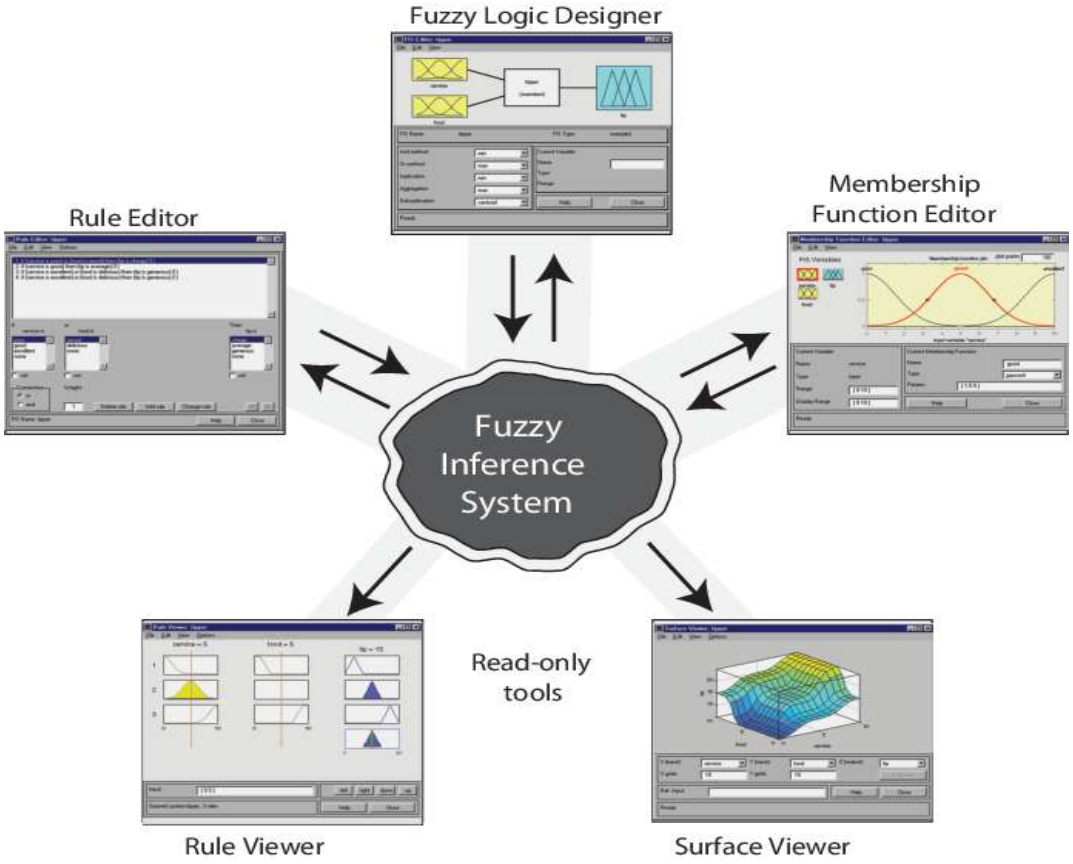


Figure III.4:Graphical User Interface (GUI) Tools

Fuzzy Logic Designer (FIS Editor) to handle the high-level issues for the system : How many input and output variables? What are their names?

Fuzzy Logic Toolbox software does not limit the number of inputs. However, the number of inputs may be limited by the available memory of your machine. If the number of inputs is too large, or the number of membership functions is too big, then it may also be difficult to analyze the FIS using the other tools.

Membership Function Editor to define the shapes of all the membership functions associated with each variable

Rule Editor to edit the list of rules that defines the behavior of the system.

Rule Viewer to view the fuzzy inference diagram. Use this viewer as a diagnostic to see, for example, which rules are active, or how individual membership function shapes influence the results

Surface Viewer to view the dependency of one of the outputs on any one or two of the inputs; that is, it generates and plots an output surface map for the system.

The Basic Tipping Problem

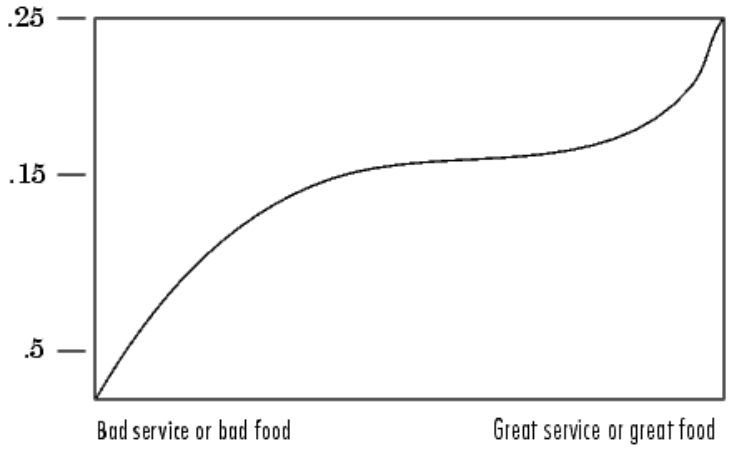
This example creates a Mamdani fuzzy inference system using on a two-input, one-output tipping problem based on tipping practices in the U.S. While the example creates a Mamdani FIS, the methods used apply to creating Sugeno systems as well.

Given a number between 0 and 10 that represents the quality of service at a restaurant (where 10 is excellent), and another number between 0 and 10 that represents the quality of the food at that restaurant (again, 10 is excellent), what should the tip be?

The starting point is to write down the three golden rules of tipping:

1. *If the service is poor or the food is rancid, then tip is cheap.*
2. *If the service is good, then tip is average.*
3. *If the service is excellent or the food is delicious, then tip is generous.*

Assume that an average tip is 15%, a generous tip is 25%, and a cheap tip is 5%.



The numbers and the shape of the curve are subject to local traditions, cultural bias, and so on, but the three rules are generally universal.

Now that you know the rules and have an idea of what the output should look like, use the UI tools to construct a fuzzy inference system for this decision process.

How to run it?

1. Run **MATLAB**
2. Type **Fuzzy** in command window, The push **Enter** in your Keyboard

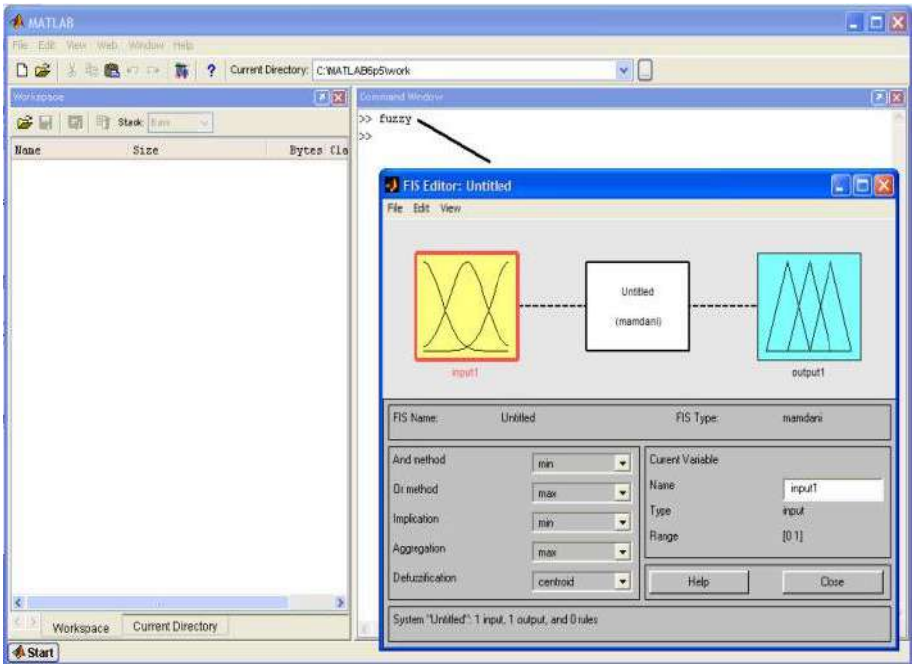
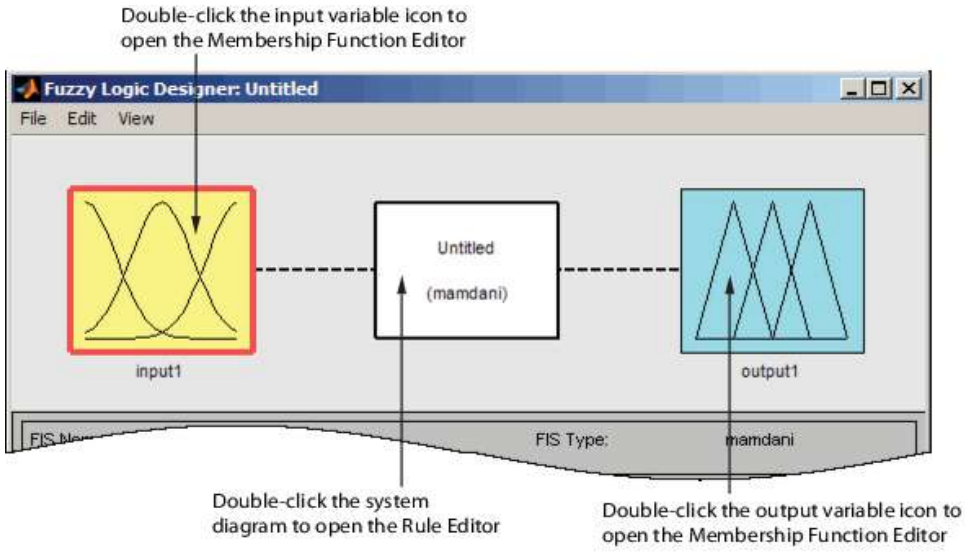


Figure III.5: The result of typing Fuzzy in the command window

3. How to build a New FIS Mamdani?

3.1 The Fuzzy Logic Designer

The **Fuzzy Logic Designer** opens and displays a diagram of the fuzzy inference system with the names of each input variable on the left, and those of each output variable on the right, as shown in the next figure. The sample membership functions shown in the boxes are just icons and do not depict the actual shapes of the membership functions.



Below the diagram is the name of the system and the type of inference used.

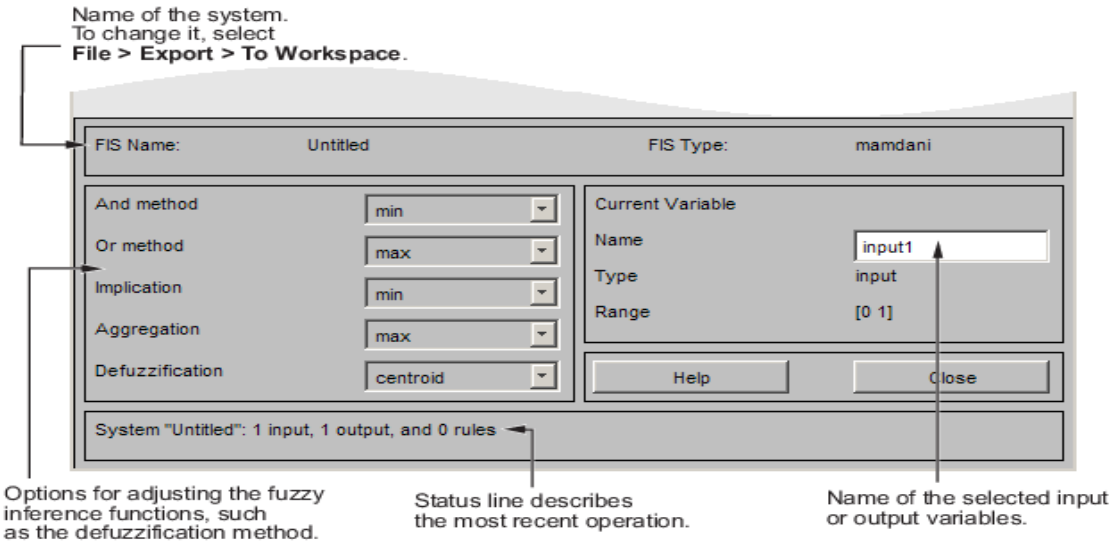


Figure III.6: Fuzzy Logic Designer Explanation.

In the **Fuzzy Logic Designer**:

- The drop-down lists let you modify the fuzzy inference functions.
- The **Current Variable** area displays the name of an input or output variable, its type, and default range.
- A status line at the bottom displays information about the most recent operation.

The generic untitled Fuzzy Logic Designer opens, with one input input1, and one output output1.

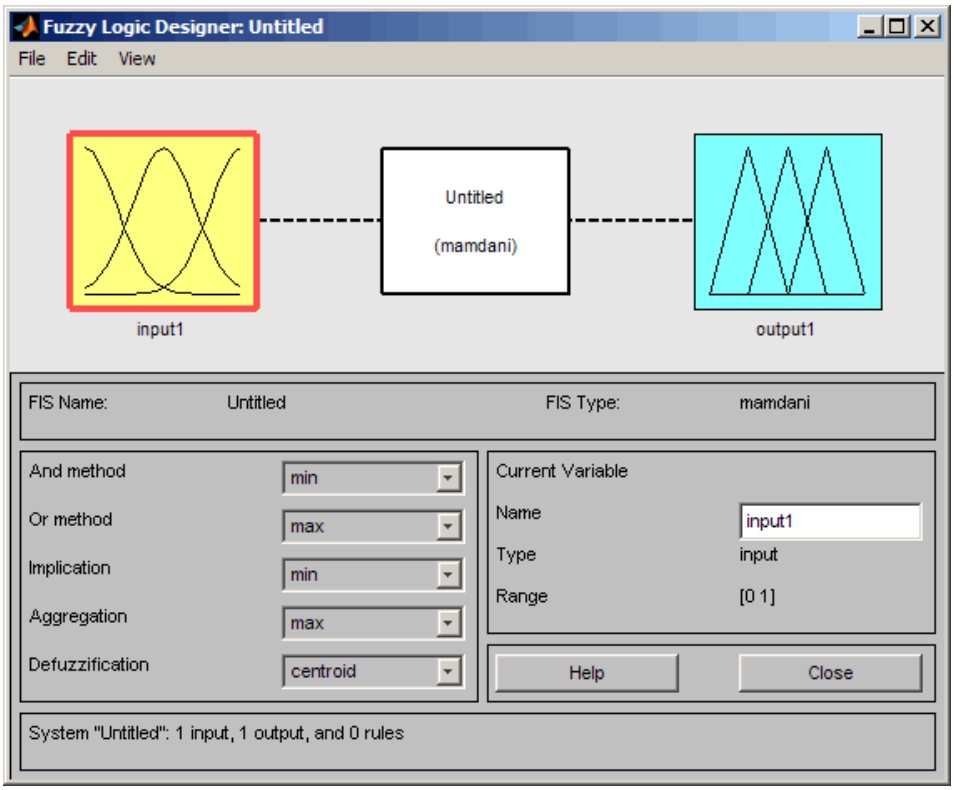


Figure III.7: Fuzzy Logic Designer.

In this example, you construct a two-input, one output system. The two inputs are **service** and **food**. The one output is **tip**

To add a second input variable and change the variable names to reflect these designations:

1. Select **Edit** > **Add variable** > **Input**.
    a. A second yellow box labeled **input2** appears.
2. Click the yellow box **input1**. This box is highlighted with a red outline.

3. Edit the **Name** field from input1 to service, and press **Enter**.
4. Click the yellow box **input2**. This box is highlighted with a red outline.
5. Edit the **Name** field from input2 to food, and press **Enter**.
6. Click the blue box **output1**.
7. Edit the **Name** field from output1 to tip, and press **Enter**.
8. Select **File** > **Export** > **To Workspace**.

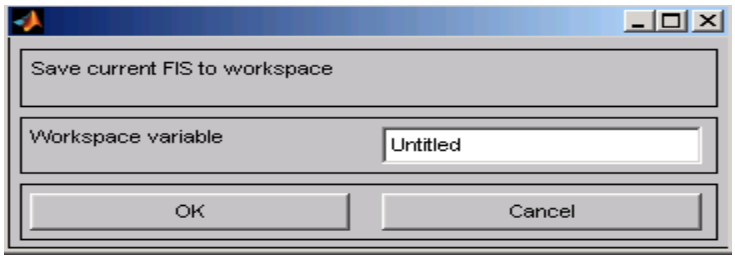


Figure III.8: Saving In The Workspace.

9. Enter the **Workspace variable** name tipper, and click **OK**.

The diagram is updated to reflect the new names of the input and output variables. There is now a new variable in the workspace called tipper that contains all the information about this system. By saving to the workspace with a new name, you also rename the entire system. Your Windows looks somethinglike the followingdiagram.

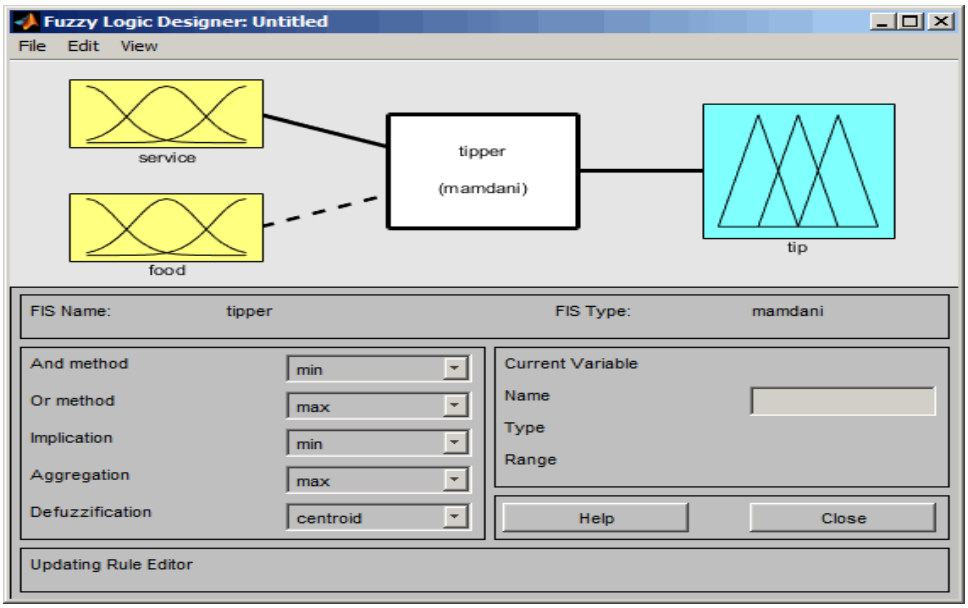


Figure III.9: Fuzzy Logic Designer With Two Inputs.

Leave the inference options in the lower left in their default positions for now. You have entered all the information you need for this particular UI. Next, define the membership functions associated with each of the variables. To do this, open the Membership Function Editor.

You can open the Membership Function Editor in one of three ways:

- Within the **Fuzzy Logic Designer** window, select **Edit** > **Membership Functions**.
- Within the **Fuzzy Logic Designer** window, double-click the blue icon called **tip**.
- At the command line, type mfedit.

3.2 The Membership Function Editor

The Membership Function Editor is the tool that lets you display and edit all of the membership functions associated with all of the input and output variables for the entire fuzzy inference system. The Membership Function Editor shares some features with the **Fuzzy Logic Designer**, as shown in the next figure. In fact, all of the five basic UI tools have similar menu options, status lines, and **Help** and **Close** buttons.

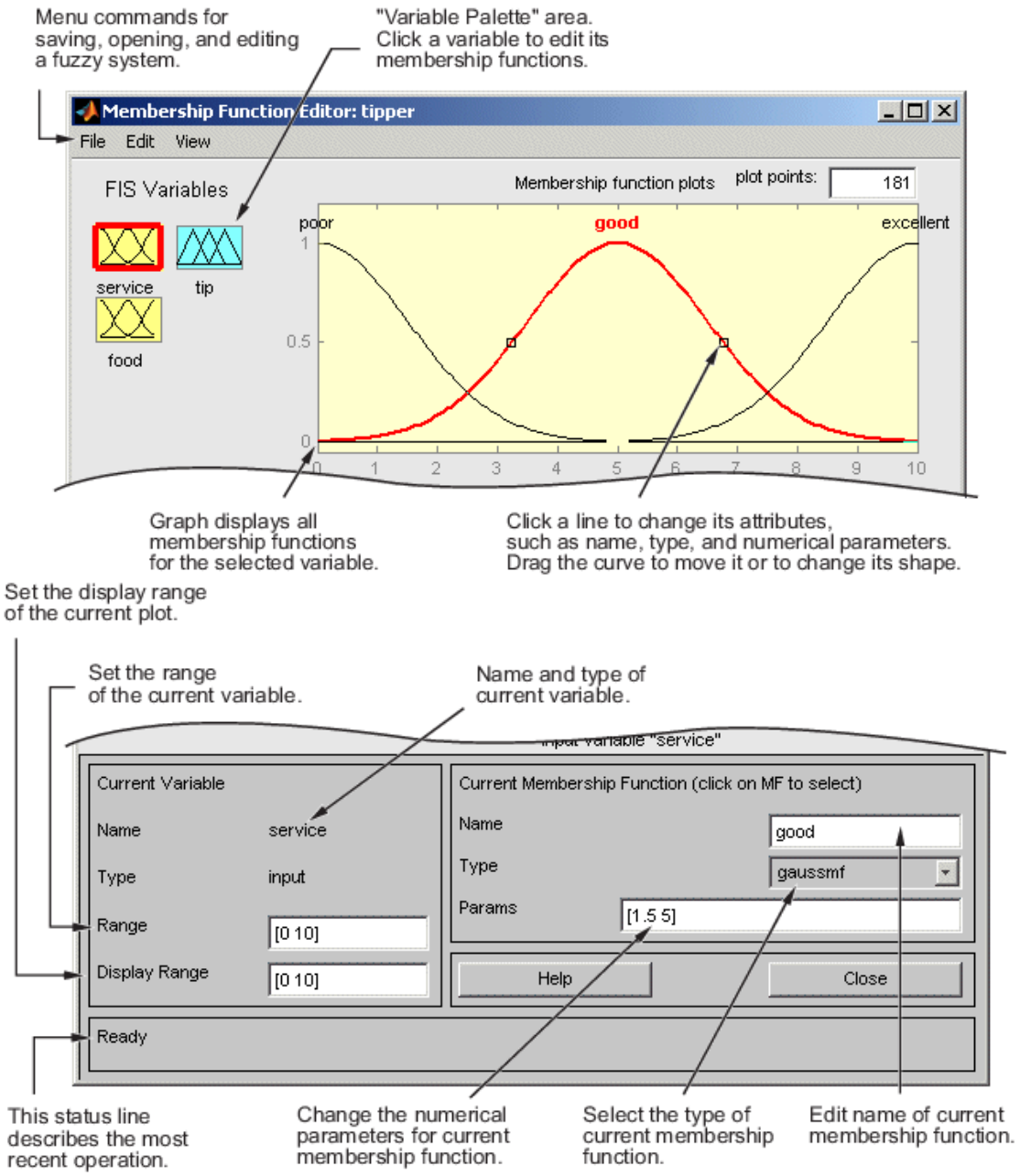


Figure III.10: Membership Function Editor Explanation.

When you open the Membership Function Editor to work on a fuzzy inference system that does not already exist in the workspace, there are no membership functions associated with the variables that you defined with the **Fuzzy Logic Designer**.

On the upper-left side of the graph area in the Membership Function Editor is a "Variable Palette" that lets you set the membership functions for a given variable.

To set up the membership functions associated with an input or an output variable for the FIS, select a FIS variable in this region by clicking it.

Next select the **Edit** pull-down menu, and choose **Add MFs...** A new window appears which allows you to select both the membership function type and the number of membership functions associated with the selected variable. In the lower-right corner of the window are the controls that let you change the name, type, and parameters (shape), of the membership function, after it is selected.

The membership functions from the current variable are displayed in the main graph. These membership functions can be manipulated in two ways. You can first use the mouse to select a particular membership function associated with a given variable quality, (such as poor, for the variable, service), and then drag the membership function from side to side. This action affects the mathematical description of the quality associated with that membership function for a given variable. The selected membership function can also be tagged for dilation or contraction by clicking on the small square drag points on the membership function, and then dragging the function with the mouse toward the *outside*, for dilation, or toward the *inside*, for contraction. This action changes the parameters associated with that membership function.

Below the Variable Palette is some information about the type and name of the current variable. There is a text field in this region that lets you change the limits of the current variable's range (universe of discourse) and another that lets you set the limits of the current plot (which has no real effect on the system).

The process of specifying the membership functions for the two-input tipping example, tipper, is as follows:

1. Double-click the input variable service to open the Membership Function Editor.

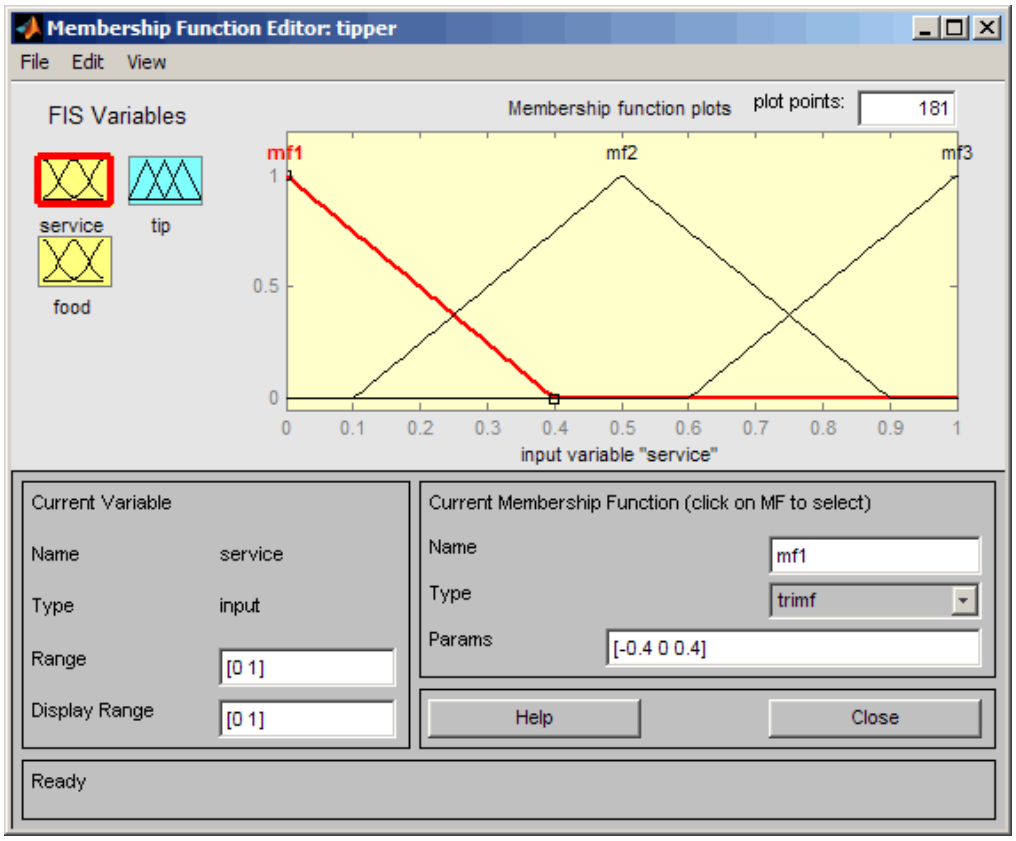


Figure III.11: Membership Function Editor.

1. In the Membership Function Editor, enter [0 10] in the **Range** and the **Display Range** fields.
2. Create membership functions for the input variable service.
   a. Select **Edit** > **Remove All MFs** to remove the default membership functions for the input variable service.
   b. Select **Edit** > **Add MFs** to open the Membership Functions dialog box.
   c. In the Membership Functions dialog box, select gaussmf as the **MF Type**.

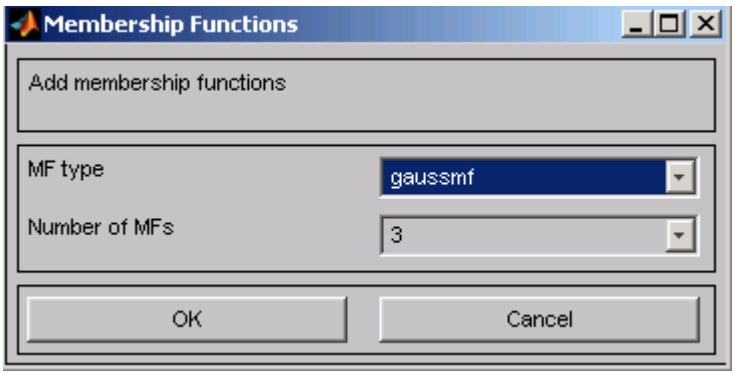


Figure III.12: Membership Function Settings.

d. Verify that 3 is selected as the **Number of MFs**.

e. Click **OK** to add three Gaussian curves to the input variable service.

Rename the membership functions for the input variable service, and specify their parameters.

a. Click on the curve named mf1 to select it, and specify the following fields in the **Current Membership Function (click on MF to select)** area:

- In the **Name** field, enter **poor**.
- In the **Params** field, enter [1.5 0].

The two inputs of **Params** represent the standard deviation and center for the Gaussian curve.

a. Click on the curve named mf2 to select it, and specify the following fields in the **Current Membership Function (click on MF to select)** area:

- In the **Name** field, enter **good**.
- In the **Params** field, enter [1.5 5].

b. Click on the curve named mf3, and specify the following fields in the **Current Membership Function (click on MF to select)** area:

- In the **Name** field, enter **excellent**.
- In the **Params** field, enter [1.5 10].

The Membership Function Editor window looks similar to the following figure:

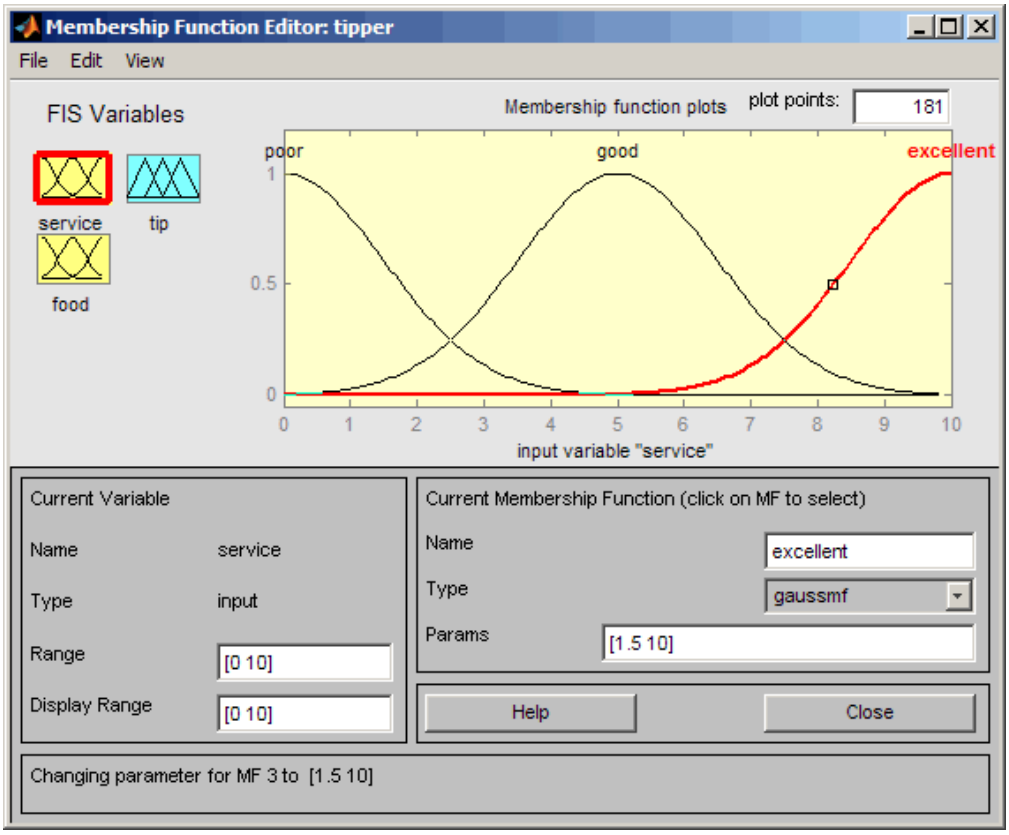


Figure III.13: Membership Function Editor Settings.

1. In the **FIS Variables** area, click the input variable food to select it.
2. Enter [0 10] in the **Range** and the **Display Range** fields.
3. Create the membership functions for the input variable food.
   a. Select **Edit** > **Remove All MFs** to remove the default Membership Functions for the input variable food.
   b. Select **Edit** > **Add MFs** to open the Membership Functions dialog box.
   c. In the Membership Functions dialog box, select trapmf as the **MF Type**.
   d. Select 2 in the **Number of MFs** drop-down list.
   e. Click **OK** to add two trapezoidal curves to the input variable food.
4. Rename the membership functions for the input variable food, and specify their parameters:
   a. In the **FIS Variables** area, click the input variable food to select it.

b. Click on the curve named mf1, and specify the following fields in the **Current Membership Function (click on MF to select)** area:

- In the **Name** field, enter rancid.
- In the **Params** field, enter [0 0 1 3].

c. Click on the curve named mf2 to select it, and enter delicious in the **Name** field.

Reset the associated parameters if desired.

5. Click on the output variable tip to select it.
6. Enter [0 30] in the **Range** and the **Display Range** fields to cover the output range.

The input ranges from 0 to 10, but the output is a tip between 5% and 25%.

7. Rename the default triangular membership functions for the output variable tip, and specify their parameters.

a. Click the curve named mf1 to select it, and specify the following fields in the **Current Membership Function (click on MF to select)** area:

- In the **Name** field, enter cheap.
- In the **Params** field, enter [0 5 10].

b. Click the curve named mf2 to select it, and specify the following fields in the **Current Membership Function (click on MF to select)** area:

- In the **Name** field, enter average.
- In the **Params** field, enter [10 15 20].

c. Click the curve named mf3 to select it, and specify the following:

- In the **Name** field, enter generous.
- In the **Params** field, enter [20 25 30].

The Membership Function Editor looks similar to the following figure.

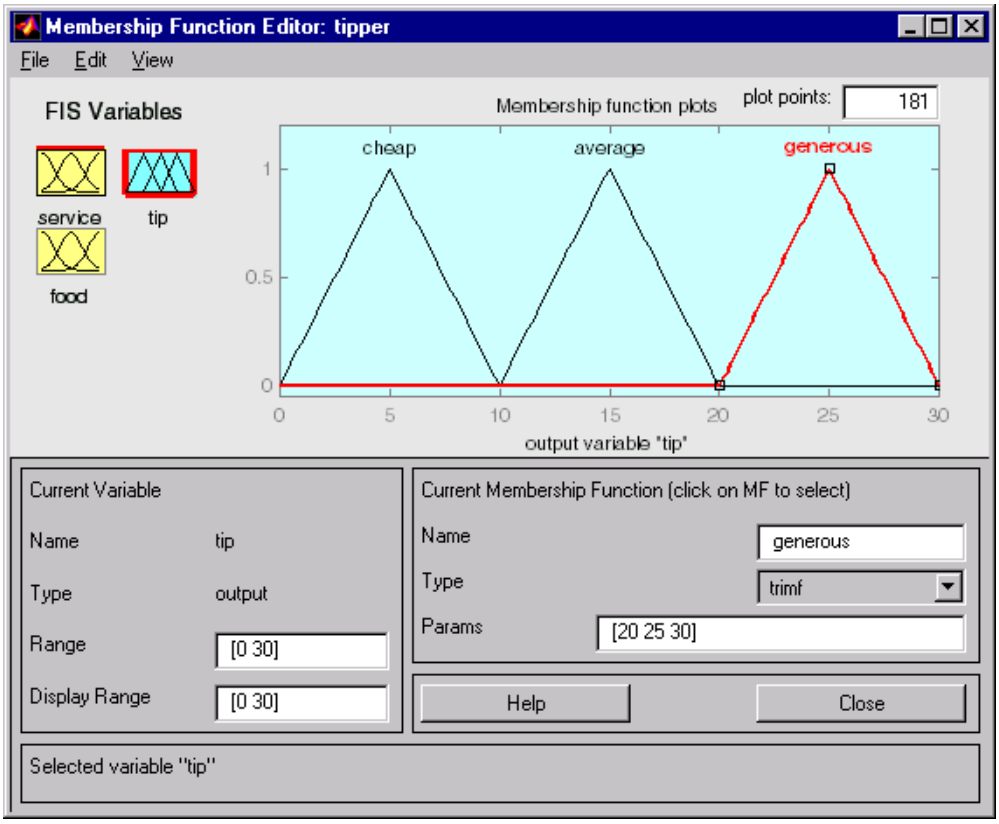


Figure III.14: Membership Functions Editor After Entering The Settings..

Now that the variables have been named and the membership functions have appropriate shapes and names, you can enter the rules. To call up the Rule Editor, go to the **Edit** menu and select **Rules**, or type rule edit at the command line.

3.3 The Rule Editor

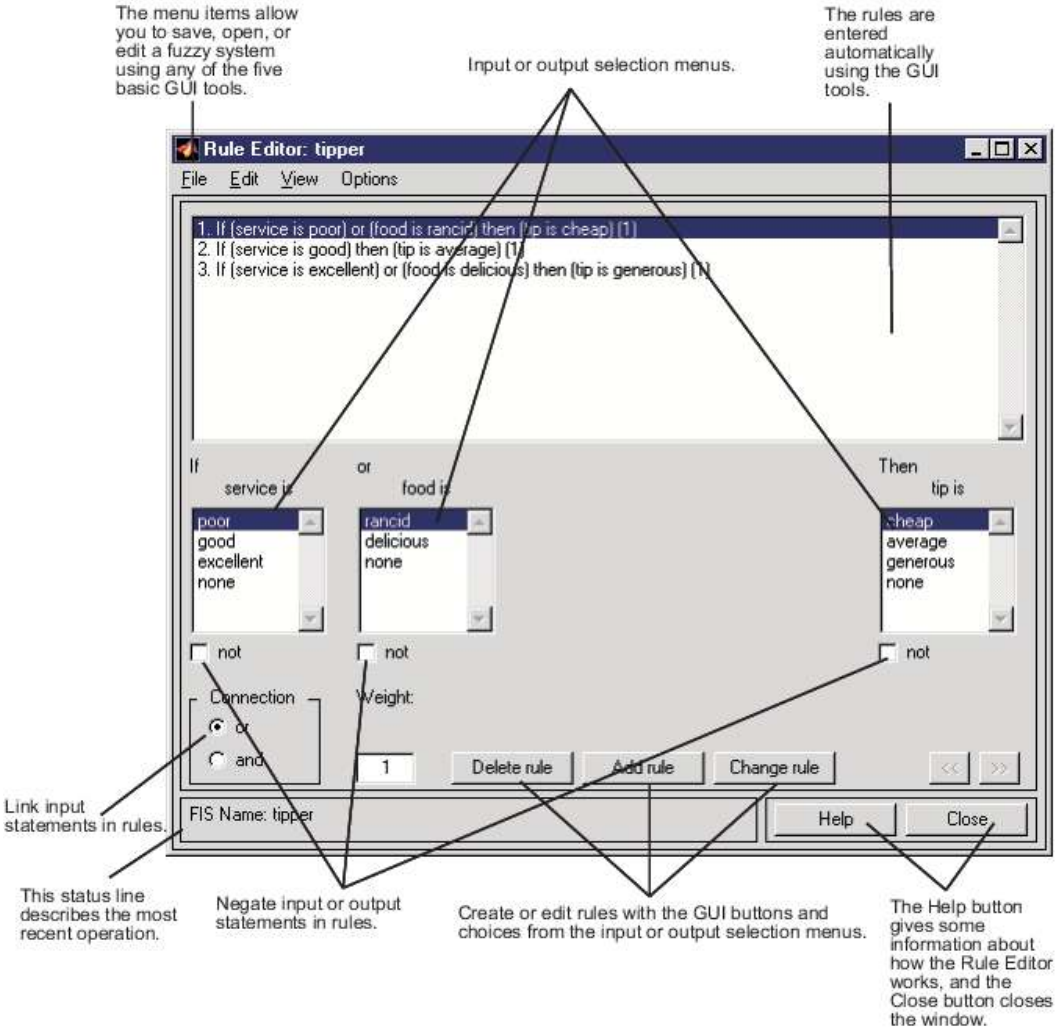


Figure III.15: The Rules Editor.

Constructing rules using the graphical Rule Editor interface is fairly self evident. Based on the descriptions of the input and output variables defined with the **Fuzzy Logic Designer**, the Rule Editor allows you to construct the rule statements automatically. You Can :

- Create rules by selecting an item in each input and output variable box, selecting one **Connection** item, and clicking **Add Rule**. You can choose none as one of the variable qualities to exclude that variable from a given rule and choose not under any variable name to negate the associated quality.
- Delete a rule by selecting the rule and clicking **Delete Rule**.

- Edit a rule by changing the selection in the variable box and clicking **Change Rule**.
- Specify weight to a rule by typing in a desired number between 0 and 1 in **Weight**. If you do not specify the weight, it is assumed to be unity.

Similar to those in the **Fuzzy Logic Designer** and the Membership Function Editor, the Rule Editor has the menu bar and the status line. The menu items allow you to open, close, save and edit a fuzzy system using the five basic UI tools. From the menu, You Can also :

- Set the format for the display by selecting **Options** > **Format**.
- Set the language by selecting **Options** > **Language**.

You can access information about the Rule Editor by clicking **Help** and close the UI using **Close**.

To insert the first rule in the Rule Editor, select the following:

- poor under the variable **service**
- rancid under the variable **food**
- The **or** radio button, in the **Connection** block
- Cheap, under the output variable, **tip**.

Then, click **Add rule**.

The resulting rule is

The numbers in the parentheses represent weights.

Follow a similar procedure to insert the second and third rules in the Rule Editor to get

1. *If (service is poor) or (food is rancid) then (tip is cheap)*
2. *If (service is good) then (tip is average)*
3. *If (service is excellent) or (food is delicious) then (tip is generous)*

The Format pop-up menu from the Options menu indicates that you are looking at the verbose form of the rules. Try changing it to symbolic. You will see

1. (Service==poor) | (food==rancid) => (tip=cheap)

2. (Service==good) => (tip=average)

3. (Service==excellent) | (food==delicious) => (tip=generous)

3.4 The RuleViewer

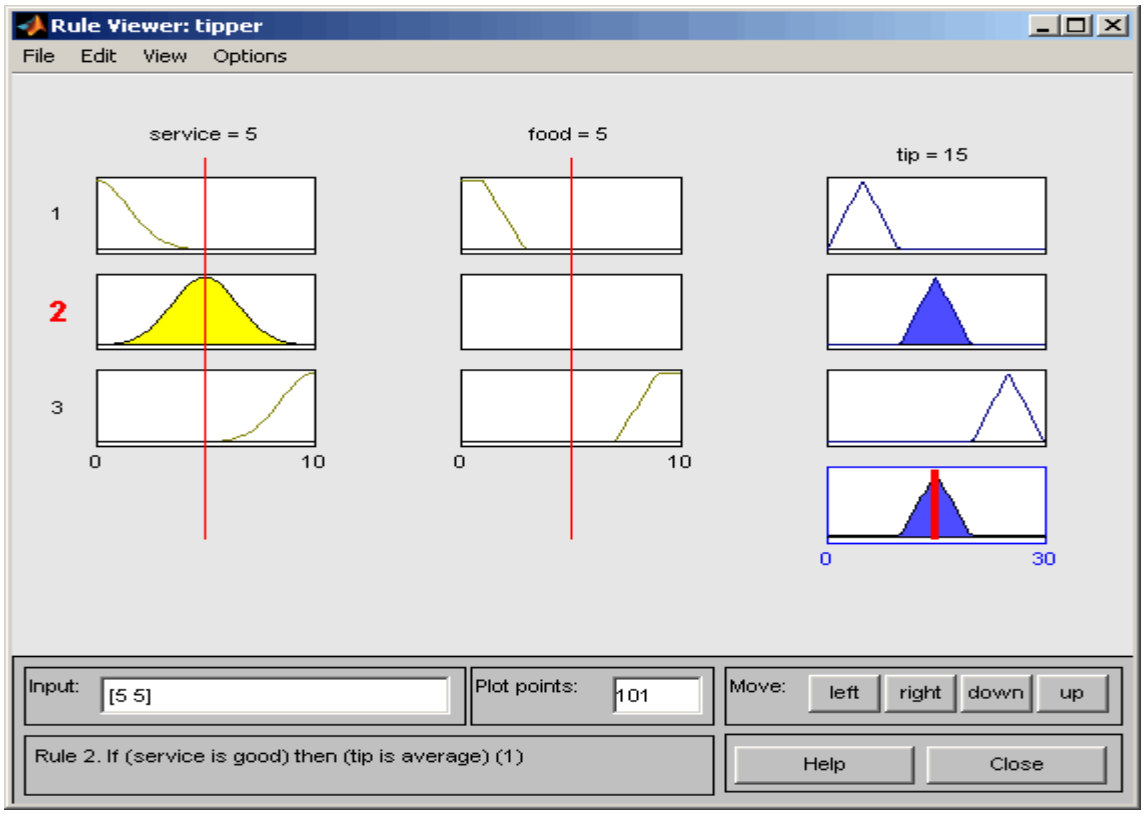


Figure III.16: The Rules Viewer.

The Rule Viewer displays a roadmap of the whole fuzzy inference process. It is based on the fuzzy inference diagram described in the previous section. You see a single figure window with 10 plots nested in it. The three plots across the top of the figure represent the antecedent and consequent of the first rule. Each rule is a row of plots, and each column is a variable. The rule numbers are displayed on the left of each row. You can click on a rule number to view the rule in the status line.

- The first two columns of plots (the six yellow plots) show the membership functions referenced by the antecedent, or the if-part of each rule.
- The third column of plots (the three blue plots) shows the membership functions referenced by the consequent, or the then-part of each rule.

Notice that under **food**, there is a plot which is blank. This corresponds to the characterization of none for the variable **food** in the second rule.

- The fourth plot in the third column of plots represents the aggregate weighted decision for the given inference system.

  This decision will depend on the input values for the system. The defuzzified output is displayed as a bold vertical line on this plot.

  The variables and their current values are displayed on top of the columns. In the lower left, there is a text field **Input** in which you can enter specific input values. For the two-input system, you will enter an input vector, [9 8], for example, and then press **Enter**. You can also adjust these input values by clicking on any of the three plots for each input. This will move the red index line horizontally, to the point where you have clicked. Alternatively, you can also click and drag this line in order to change the input values. When you release the line, (or after manually specifying the input), a new calculation is performed, and you can see the whole fuzzy inference process take place:

- Where the index line representing service crosses the membership function line "service is poor" in the upper-left plot determines the degree to which rule one is activated.
- A yellow patch of color under the actual membership function curve is used to make the fuzzy membership value visually apparent.

  Each of the characterizations of each of the variables is specified with respect to the input index line in this manner. If you follow rule 1 across the top of the diagram, you can see the consequent "tip is cheap" has been truncated to exactly the same degree as the (composite) antecedent — this is the implication process in action. The aggregation occurs down the third column, and the resultant aggregate plot is shown in the single plot appearing in the lower right corner of the plot field. The defuzzified output value is shown by the thick line passing through the aggregate fuzzy set.

  You can shift the plots using **left**, **right**, **down**, and **up**. The menu items allow you to save, open, or edit a fuzzy system using any of the five basic UI tools.

  The Rule Viewer allows you to interpret the entire fuzzy inference process at once. The Rule Viewer also shows how the shape of certain membership functions influences the overall result. Because it plots every part of every rule, it can become unwieldy for particularly large systems,

but, for a relatively small number of inputs and outputs, it performs well (depending on how much screen space you devote to it) with up to 30 rules and as many as 6 or 7 variables.

The Rule Viewer shows one calculation at a time and in great detail. In this sense, it presents a sort of micro view of the fuzzy inference system. If you want to see the entire output surface of your system — the entire span of the output set based on the entire span of the input set — you need to open up the Surface Viewer. This viewer is the last of the five basic Fuzzy Logic Toolbox UI tools. To open the Surface Viewer, select **Surface** from the **View** menu.

3.5 The Surface Viewer

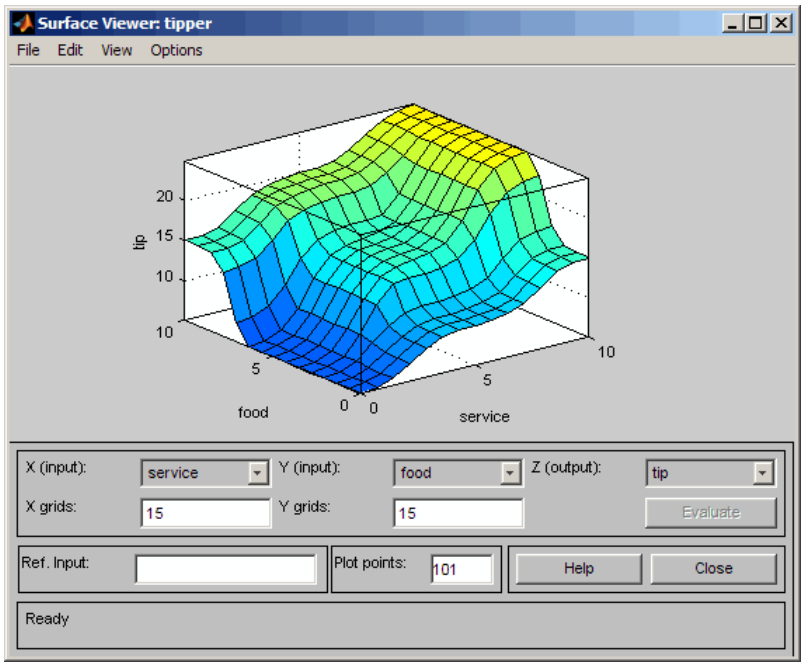


Figure III.17: The Surface Viewer.

Upon opening the Surface Viewer, you see a three-dimensional curve that represents the mapping from food and service quality to tip amount. Because this curve represents a two-input one-output case, you can see the entire mapping in one plot. When we move beyond three dimensions overall, we start to encounter trouble displaying the results.

Accordingly, the Surface Viewer is equipped with drop-down menus **X (input)**, **Y (input)** and **Z (output)** that let you select any two inputs and any one output for plotting. Below these menus

are two input fields **X grids** and **Y grids** that let you specify how many x-axis and y-axis grid lines you want to include. This capability allows you to keep the calculation time reasonable for complex problems.

By default, the surface plot updates automatically when you change the input or output variable selections or the number of grid points. To disable automatic plot updates, in the **Options** menu, clear the **Always evaluate** option. When this option is disabled, to update the plot, click **Evaluate**.

If you want to create a smoother plot, use the **Plot points** field to specify the number of points on which the membership functions are evaluated in the input or output range. This field defaults to the minimum number of plot plots, 101. If you specify fewer plot points, the field value automatically resets to 101. When you specify the number of plot points, the surface plot automatically updates.

By clicking on the plot axes and dragging the mouse, you can manipulate the surface so that you can view it from different angles.

The **Ref. Input** field is used in situations when there are more inputs required by the system than the surface is mapping. You can edit this field to explicitly set inputs not specified in the surface plot.

Suppose you have a four-input one-output system and would like to see the output surface. The Surface Viewer can generate a three-dimensional output surface where any two of the inputs vary, but two of the inputs must be held constant because computer monitors cannot display a five-dimensional shape. In such a case, the input is a four-dimensional vector with NaNs holding the place of the varying inputs while numerical values indicates those values that remain fixed.

The menu items allow you to open, close, save and edit a fuzzy system using the five basic UI tools. You can access information about the Surface Viewer by clicking **Help** and close the UI using **Close** [28].

4. Conclusion:

The emergence of fuzzy logic and its applications has dramatically changed the face of industrial control engineering. Fuzzy logic has given an opportunity to control engineers to meet and overcome the challenges of developing effective controllers for complex systems. The area of fuzzy logic is still in its infancy, and is a very fertile area of engineering research. Comparison of genetic algorithms with other evolutionary techniques viz. simulated annealing, chaotic theory is a grey area of research in the field of optimization. In nutshell, the results reported in this thesis outlined a research direction for software engineering researchers, who had not paid much interest in the application domain of soft computing techniques, where fuzzy logic and genetic algorithms, play a visible role. The aim of this work is to raise the awareness of good results in soft computing and bridge the gap between software engineering and soft computing.

Chapter IV:

Simulation

1. Introduction:

Washing machines are common household appliances in Algeria but washing machine that efficiently controls the wash time is vital. Conventional, proportional, integral and differential [PID] controllers have proven to be less capable. So fuzzy logic enables designers to control complex systems more effectively than traditional approaches. Electronic appliances manufacturer companies are working on the sensor based fully-automatic washing machines which can sense type of fabric, type of dirt, dirtiness of clothes and volume of clothes to decide the ideal the washing time andthe speed of washing as per the information by the sensor systems to the machine . The success of sensor based washing machine depends upon the availability of reliable and durable sensors. The presently available sensors for washing machine in the market are load sensor, turbidity sensor, temperature sensor and optical sensor. The users of washing machines have been facing the problem of selecting the length of wash time based on the type of clothes, type of dirt, dirtiness of clothes and volume of clothes. Most of the people find it very difficult to decide that which cloth needs what amount of washing time. To overcome these problems fully-automatic washing machines based on fuzzy logic offers the advantage of performance, simplicity and less cost.

In this chapter , we have introduced four input variables and one output fuzzy logic controller to get correct wash time. Fuzzy logic toolbox in MATLAB is used for FLC simulation. The results so obtained

2. Proposed Design:

To model a system using fuzzy logic, the first step is to determine the inputs and outputs of it. A washing machine’s most important duty is to clean the cloths without damaging them. In order to achieve it, the output parameters of fuzzy logic, which are related to washing, must be paid importance.

The proposed Fuzzy Logic Controller for washing machine consists of four Linguistic Inputs:

- Dirtiness of clothes
- Type of dirt
- Types of fabric
- Volumeof clothes

And two outputs:

- Washing speed

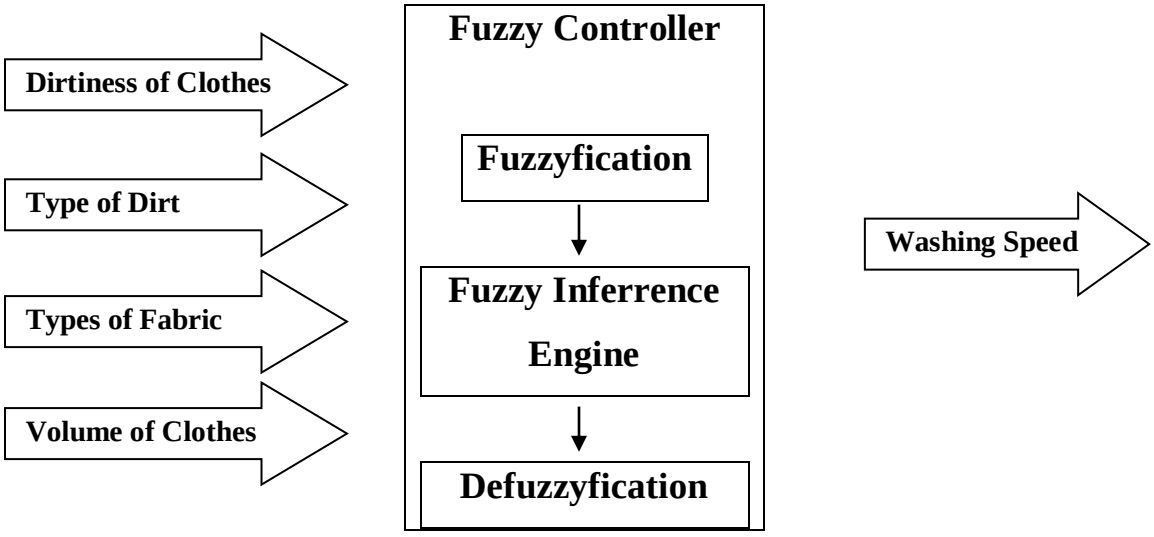


Figure IV.1: Show the design of the controller for washing machine.

The washing machine inference techniques used to design the controller system is Mamdani method, with triangular membership function. The Mamdani style of fuzzy inference process is performed in four steps as follow fuzzification of the input variable, Rule evaluation (Inference engine), aggregation of the rule outputs (composition) and defuzzification to get the output.

2.1 Fuzzification

At this step take the crisp inputs, the dirtiness of the clothes, type of dirt, the volume of clothes, type of fabric used to make clothes to determine the degree to which these inputs belong to each of the appropriate fuzzy sets. The membership value and range are shown in Table 1 below.

Input Variable	Membership Value	Membership Range
Dirtiness of Clothes	[Small,Medium, Large]	[0 100]
Type of Dirt	[Notgreasy, Medium, Greasy]	[0 100]
Type of Fabric	[Silk, Woolen, Cotton]	[0 100]
Volume of Clothes	[Small, Medium, Large]	[0 100]

Table IV.1: show the Input variable and membership

After knowing the membership of each input variable and the output variable as shown in Table 2 below, then the inference engine is created.

2.2 Inference Engine

At this step, the Fuzzified inputs above are taken and apply them to the antecedents of the fuzzy rules. Because there are multiple antecedents AND operator is used to connecting and obtaining the single number that represents the result. Because of space here is the list of the example of the rule created to determine the washing speed.

1. If (Dirtiness is Small) and (TypeOfDirt is Not greasy) and (TypeOfFabric is Silk) and (volume of clothes is Small) then (WashingSpeed is VeySlow)

2. If (Dirtiness is Large) and (TypeOfDirt is Greasy) and (TypeOfFabric is Cotton) and (volume of clothes is Large) then (WashingSpeed is very fast)

3. If (Dirtiness is Large) and (TypeOfDirt is Not greasy) and (TypeOfFabric is Woolen) and (volume of clothes is Medium) then (WashingSpeed is Medium)

2.3 Diffuzification

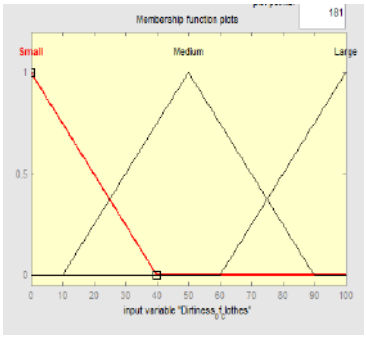
The results are combined to give a crisp answer that means the actual washing speed andwashing time from the rule the process is called deffuzifaction. The defuzzification method used is the centroid technique.

Output Variable	Membership Value	Membership Range
Washing Speed	[VeryLow, Slow, Medium, Fast, VeryFast]	[0 1200]

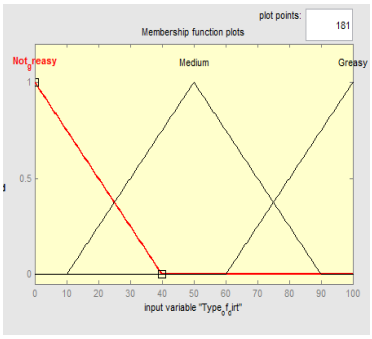
Table IV.2: show the Output variable and membership

Using fuzzy logic, input parameters such as dirtiness of clothes, type of dirt, type of fabric and volume of clothes, will help washing machine to achieve economical wash. Automation, Control and Intelligent Systems. The input and output parameter's membership function values, names, upper and lower limits are being set based on a given problem. The membership functions, with upper and lower limits, of input and output parameter's are shown in figure 3 and 4 respectively. Figure 3 and figure 4 showmembership functions of input and output respectively and also their upper and lower limits.

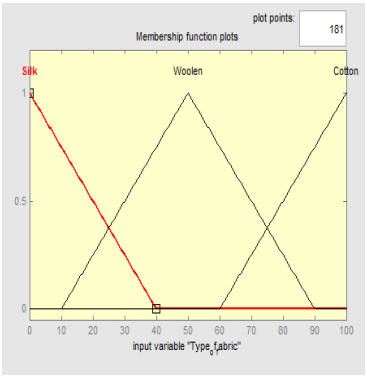
After determining the membership functions and their upper and lower limits required for the modeling of necessary parameters, a total of 81 rules have been established to define relationship among those parameters. In order to apply fuzzy logic to washing, it is necessary to establish fuzzy logic rules. These rules can be seen inTable 3 below.



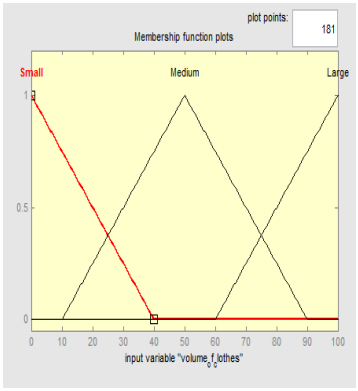
Input variable ‘’Dirtiness of clothes’’



Input of ‘’ Type of dirt ‘’

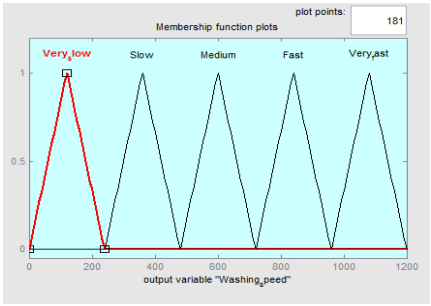


Input variable ‘’Type of fabric’’



Input of ‘’ Volume of clothes ‘’

Figure IV.2: Fuzzy logic inputs membership functions



Output variable ‘’Washing speed’’

Figure IV.3: Fuzzy logic output membership functions

	Inputs				Output
	1	**2**	**3**	**4**	**1**
1	**S**	**NG**	**SK**	**S**	**VSL**
2	**S**	**M**	**SK**	**M**	**SL**
3	**S**	**G**	**SK**	**L**	**F**
4	**M**	**NG**	**W**	**S**	**VSL**
5	**M**	**M**	**W**	**M**	**M**
6	**M**	**G**	**W**	**L**	**F**
7	**L**	**NG**	**C**	**S**	**SL**
8	**L**	**M**	**C**	**M**	**M**
9	**L**	**G**	**C**	**L**	**VF**

Table IV.3: Fuzzy Logic Rules Table for the Output ‘’ Washing speed ’’

S = Small, M = Medium, L = Large, NG = Not Greasy, G = Greasy, SK = Silk, W= Woolen, C = Cotton, VSL = Very Slow, SL = Slow, F = Fast, VF = Very Fast.

3. The Simulaion:

In the first step we open the MATLAB and type ‘ fuzzy ‘ in the command window and push ENTER then we get the page of FIS with just one input and output

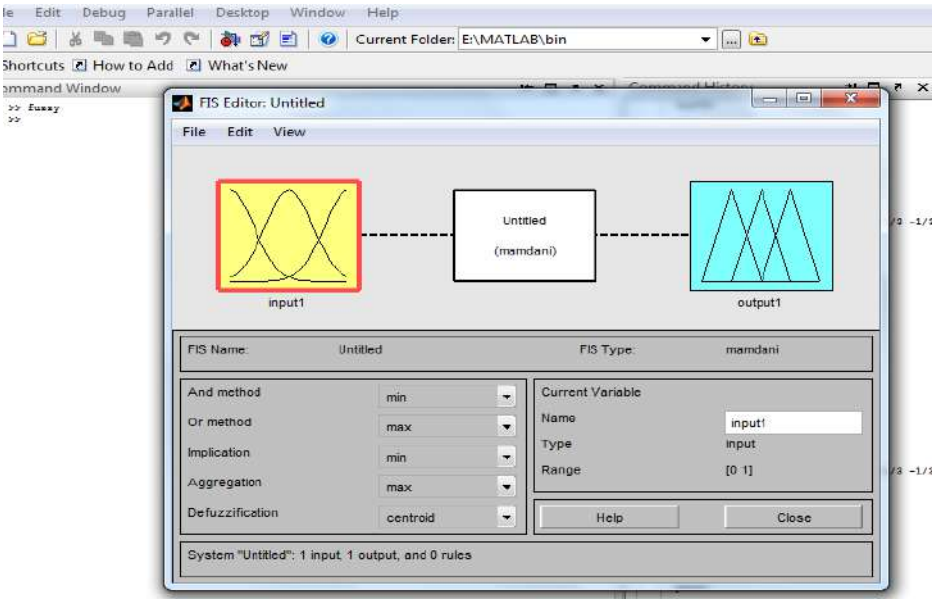


Figure IV.4: New FIS

After that w update our FIS from the file that we made before

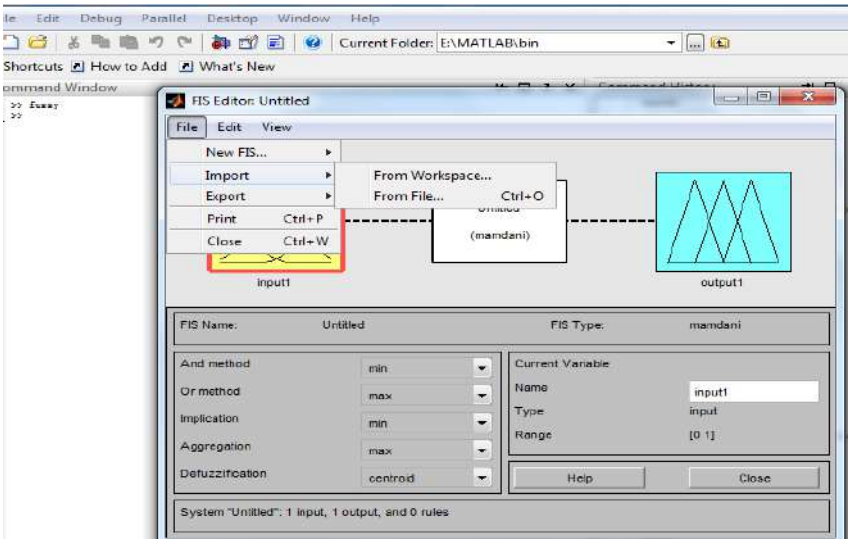


Figure IV.5: Updating our FIS

Here the FIS that we’ll work for Then, we save the FIS to workspace

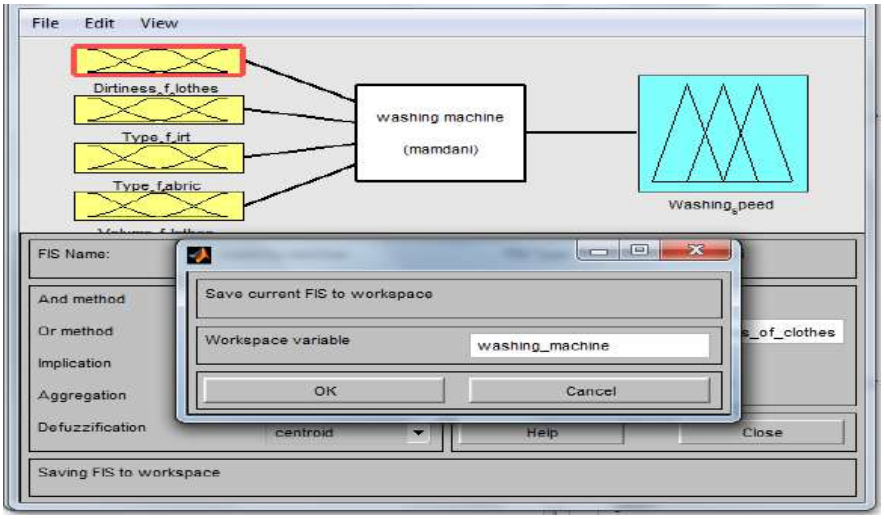


Figure IV.6: Save the current FIS to workspace

We must open the previous simulation pae (chapter II) coupled with our fuzzy system and a gain to convert it from($Tr/m\ to\ Rad/s$)

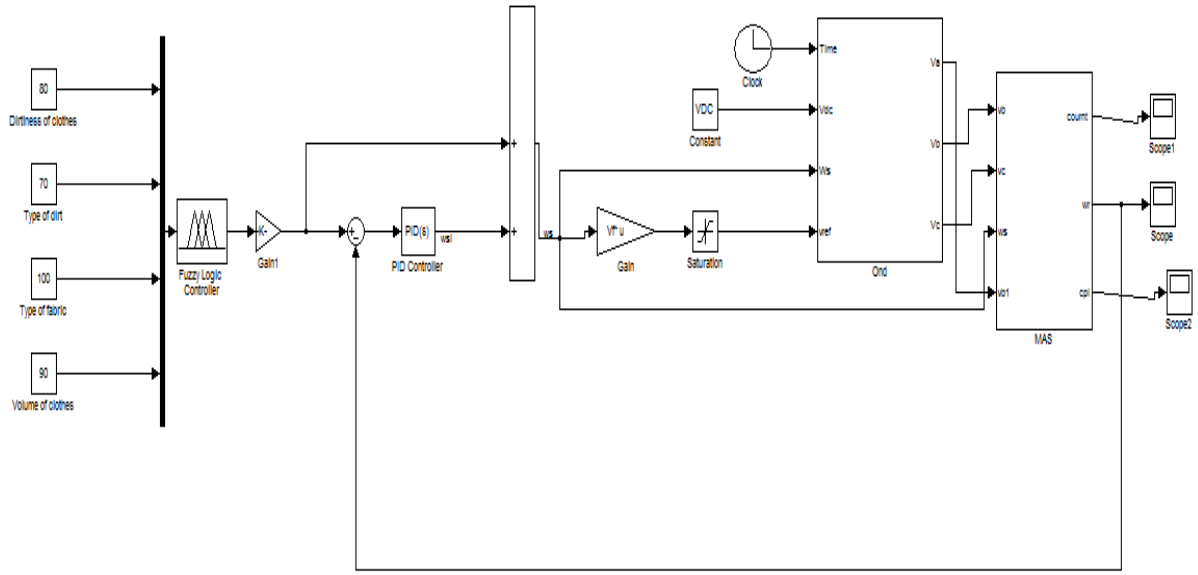
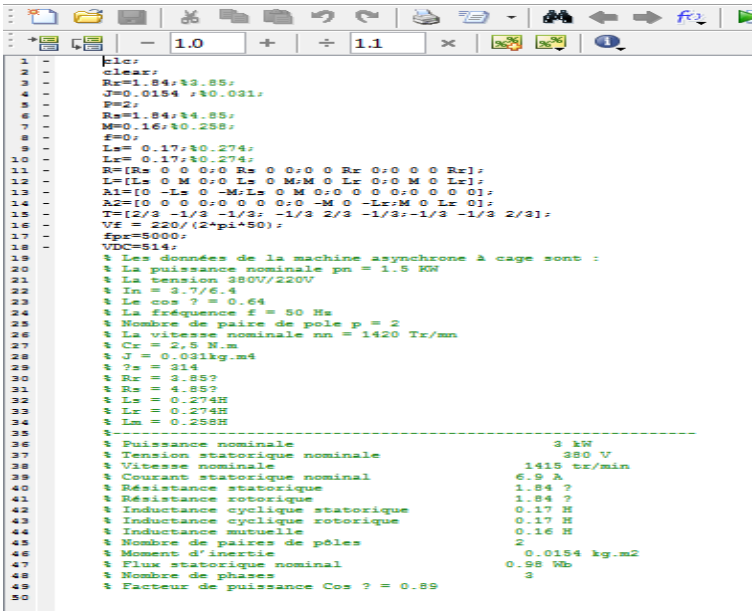


Figure IV.7: The Final Model

And ofcourse we have to run the MAS parameters



```
clc;
clear;
Rr=1.84;%3.85;
J=0.0154 ;%0.031;
P=2;
Rs=1.84;%4.85;
M=0.16;%0.258;
f=0;
Ls= 0.17;%0.274;
Lr= 0.17;%0.274;
R=[Rs 0 0 0;0 Rs 0 0;0 0 Rr 0;0 0 0 Rr];
L=[Ls 0 M 0;0 Ls 0 M;M 0 Lr 0;0 M 0 Lr];
A1=[0 -Ls 0 -M;Ls 0 M 0;0 0 0 0;0 0 0 0];
A2=[0 0 0 0;0 0 0 0;0 -M 0 -Lr;M 0 Lr 0];
T=[2/3 -1/3 -1/3; -1/3 2/3 -1/3;-1/3 -1/3 2/3];
Vf = 220/(2*pi*50);
fpr=5000;
VDC=514;
% Les données de la machine asynchrone à cage sont :
% La puissance nominale pn = 1.5 KW
% La tension 380V/220V
% In = 3.7/6.4
% Le cos ? = 0.64
% La fréquence f = 50 Hz
% Nombre de paire de pole p = 2
% La vitesse nominale nn = 1420 Tr/mn
% Cr = 2,5 N.m
% J = 0.031kg.m4
% ?s = 314
% Rr = 3.85?
% Rs = 4.85?
% Ls = 0.274H
% Lr = 0.274H
% Lm = 0.258H
%-------------------------------------------------------------------
% Puissance nominale                          3 kW
% Tension statorique nominale                 380 V
% Vitesse nominale                            1415 tr/min
% Courant statorique nominal                  6.9 A
% Résistance statorique                       1.84 ?
% Résistance rotorique                        1.84 ?
% Inductance cyclique statorique              0.17 H
% Inductance cyclique rotorique               0.17 H
% Inductance mutuelle                         0.16 H
% Nombre de paires de pôles                   2
% Moment d'inertie                            0.0154 kg.m2
% Flux statorique nominal                     0.98 Wb
% Nombre de phases                            3
% Facteur de puissance Cos ? = 0.89
```

Figure IV.8: MAS parameters

At the end and according to our 4 inputs we run thesimulated model and we get this figures:

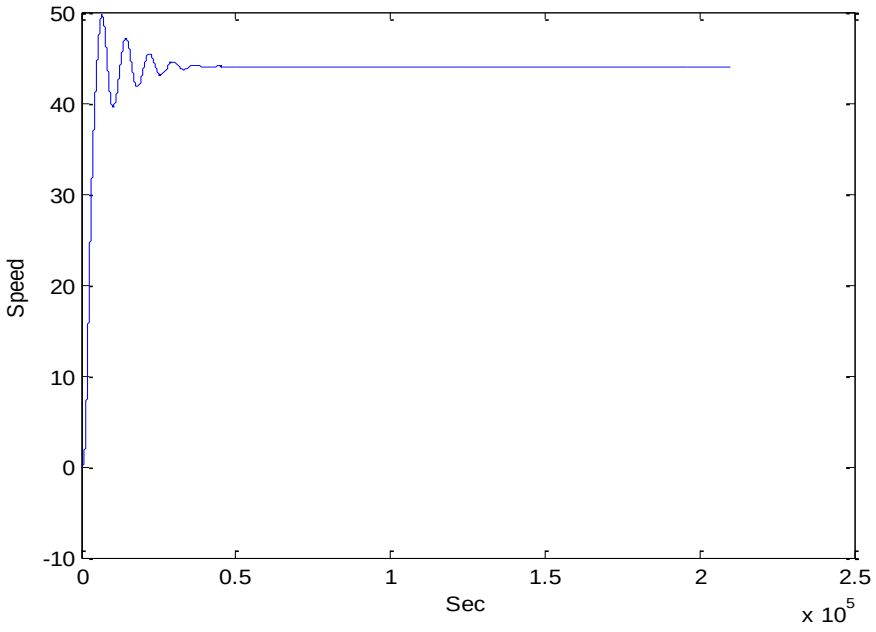


Figure IV.9: The final speed result

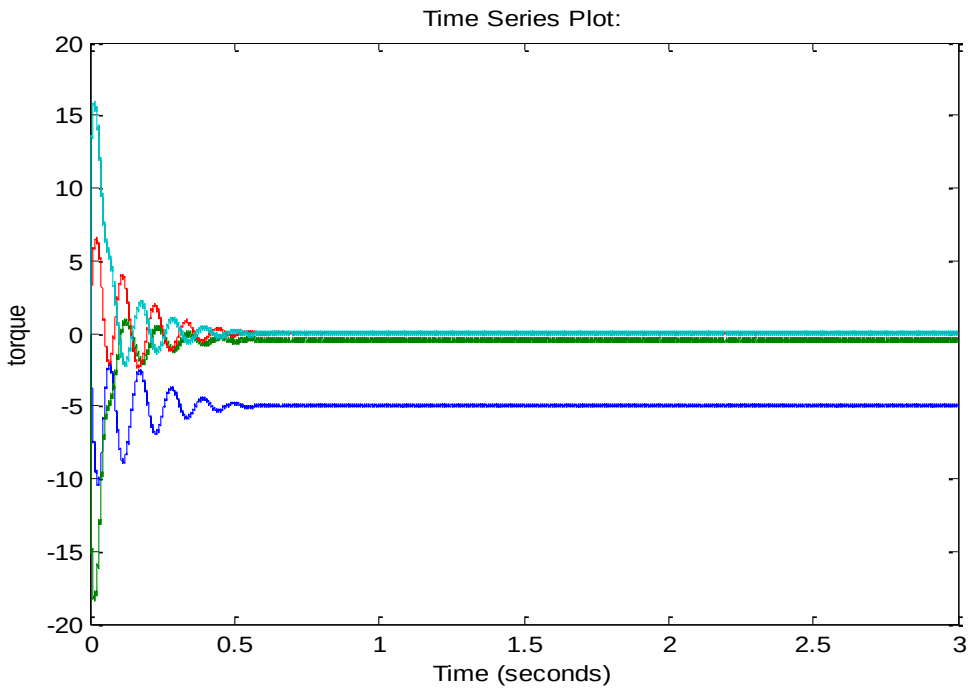


Figure IV.10: The final torque result

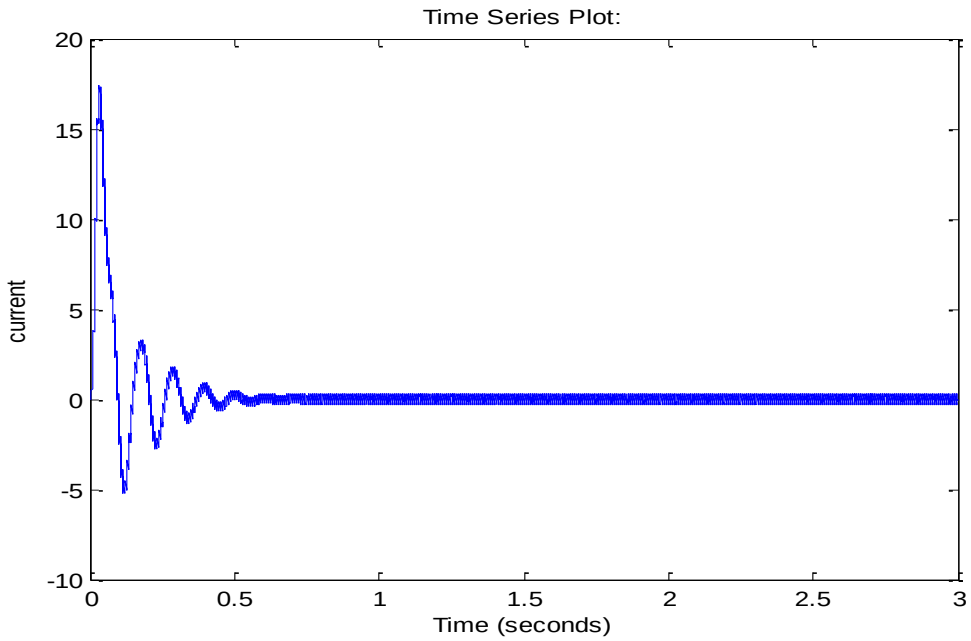


Figure IV.11: The final current result

Conclusion

This chapter focuses on the design and simulation of a washing machine on MATLAB. By the use of proposed fuzzy logic Controller, we have been able to obtain different Wash Speed (output Variable) for different Type of Dirt, Dirtiness of Clothes, Types of Clothes and Volume of Clothes (Input Variable). This analysis also leads to saving in electrical energy and time for washing the clothes by using proposed Washing Machine. This proposed advance and automatic sensor system based Washing Machine depicts the advantage of Fuzzy Logic Controller in the conventional Washing Machine.

GENERAL CONCLUSION

My project allowed me to know the operation of an asynchronous motor and its different methods to vary its speed, for this I used the scalar command which allowed me to know the operation of the latter and how to use it to vary the speed.

At the end I used the fuzzy logic controller to contol the speed.

I carried out the whole command program using MATLAB / SIMULINK

I hope that our work can serve as a support for future students, who want to discover these technologies

REFERENCES

[1] https://www.merriam-webster.com/dictionary/induction%20motor
[2] https://edisontechcenter.org/inductionMotors.html
[3] https://electricalacademia.com/induction-motor/three-phase-induction-motor-construction/
[4] http://electricalarticle.com/working-principle-of-three-phase-induction-motor/
[5] http://4electromecanique.blogspot.com/2016/06/moteur-asynchrone.html
[6] univ el oued commande vectorielle d'un machine a induction pdf
[7] BERNARD DE FORNEL. Machines Electriques Tournantes Conception Construction Et Commande
[8] Dr A MEROUFEL. Contrôle De La Machine Asynchrone. Université Djillali Liabés Sidi Bel- ABBES.2008\2009.
[9] L. BAGHLI. Modélisation Et Commande De La Machine Asynchrone. Notes DeCours.IUFM De Lorraine.2003\2004.
[10] Dr. MANIR HADEF. Commande Electrique. Notes De Coures. Université De Jijel.2015
[11] Mr. MIHOB YOUCEF. Commande de Flux Adaptative De La Machine Asynchrone. Thèse De Doctorat. Oran.2012.
[12] CHEKIMA MESSAOUD. Simulation du Fonctionnement du moteur Asynchrone avec Asymétrie Rotorique. Mémoire De Master Commande Electrique. Université Mohamed Khider Biskra.08 juin2014.
[13] M. LAAMAYAD TAHAR. Commande Optimale D'une Machine Asynchrone Apport DeLALogique Floue. Thèse De Magister Commande Electrique. Université De Batna.2008
[14]http://elearning.univeloued.dz/courses/EPA01/document/Cours_Master2_ChapitreI.pdf
[15] La Commande Pleine Ordre Des Onduleur Association avec une Machine Asynchrone

[20] https://fr.scribd.com/document/257622220/10-11648-j-acis-20140203-11-pdf
[21]https://www.computerworld.com/article/2566110/sidebar--the-history-of-fuzzy-logic.html
[22]https://www.weareworldquant.com/en/thought-leadership/reshaping-the-world-with-fuzzy-logic/
[23]https://www.edureka.co/blog/fuzzy-logic-ai/
[24]https://nejattukenmez.wordpress.com/2016/05/24/bulanik-mantik-nedir/
[25]https://www.hindawi.com/journals/afs/2013/581879/
[26] https://www.upgrad.com/blog/fuzzy-login-in-artificial-intelligence/
[27] https://nejattukenmez.wordpress.com/2016/05/24/bulanik-mantik-nedir/

[28]https://www.mathworks.com/help/fuzzy/building-systems-with-fuzzy-logic-toolbox-software.html#brzqs45