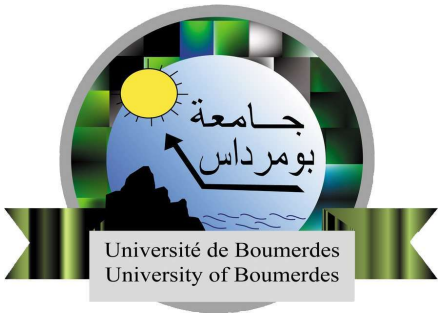
People's Democratic Republic of Algeria
Ministary of Higher Education and Scientific Research
University of M'Hamed BOUGARA - Boumerdes



Institute of Electrical Electronic Engineering
IGEE ex INILEC

Final Year Project Report Presented in Partial Fulfilment of the Requirements of the Degree of

MASTER

In **Power Engineering**

Option : **Power Engineering**

Title:

Optimization of Fuzzy Controller parameters for the vector controlled induction motor drive

Presented by :

- **AGUENAROUS Mohamed**
- **ZIDANE Houssam Eddine**

Supervisor :

Prof. KHELDOUN Aissa

Academic year : 2023/2024

Abstract

Due to its competitive features, particularly ruggedness and cost-effectiveness, the induction motor is the most widely used source of variable speed drives in the industry. Three common methods of control are employed based on the required performance level: scalar control, vector control, and direct torque control. The scalar controller is simple to implement but offers limited performance, while vector and direct torque control are used when high-performance speed control is required. However, both schemes typically rely on PID controllers, which are sensitive to the operating point and parameter variations.

This project investigates the use of Fuzzy Logic Controllers (FLC) due to their robust performance and model-free characteristics. Despite their advantages, the main drawback of FLCs is their complex tuning process, which limits their widespread utilization. To address this limitation, the project proposes optimizing the parameter selection process using meta-heuristic algorithms, specifically Particle Swarm Optimization (PSO) and Grey Wolf Optimization (GWO). Three optimization schemes were employed: gains optimization, membership functions (MFs) optimization, and combined gains and MFs optimization.

Indirect Rotor Flux-Oriented Control (IRFOC) was used to carry out this study. The simulation results indicated that the system's performance improved compared to classical control techniques such as scalar control method. Furthermore, the proposed control scheme significantly enhanced the system's robustness. This project acknowledges the challenges associated with the precise selection of fuzzy parameters and the optimization process. This work demonstrates that the implementation of Fuzzy Logic Controllers (FLCs) in industrial settings has led to significant advancements in induction motor technology. FLCs have enabled improved performance, increased robustness, and enhanced efficiency in the operation of induction motors, making them a more viable and attractive option for various industrial applications.

Keywords : Fuzzy Logic Control, FLC, Induction Motor, optimization, Particle Swarm Optimization, Grey wolf Optimization, PSO, GWO, IRFOC, MFs

---

Dedication

TO my loving parents, whose constant support and unwavering love have been the cornerstone of my life. This achievement wouldn't be possible without your dedication.
TO my wonderful grandmother, Her wisdom, kindness, and unwavering encouragement have been a guiding light throughout my journey.
TO my amazing sisters, your enduring support has been a constant source of strength throughout this journey.
TO my friends and classmates, your support and the shared learning environment made all the difference.

Thank you !

- Mohamed

TO my beloved family, honored parents ,and esteemed siblings whom without their love and continuous support i wouldn't be here.

TO my friends and classmates who provided the environment to grow and improve.

- Houssam

Acknowledgement

Praise to Allah Almighty the most merciful for guiding us all along this path and giving us the strength to proceed whenever we reached a dead end.

We would like to express our sincere gratitude to **Pr. KHELDOUN Aissa** for his invaluable contributions to this work. His insightful feedback and meticulous follow-up were instrumental in structuring the work and ensuring the highest quality in all sections. We are particularly grateful for his patience and professionalism throughout the process.

To the members of the jury, we would like to express our sincere gratitude for the honor you have bestowed upon us by taking the time to read and evaluate this work.

We would also like to thank the teaching and administrative staff of the university for their efforts in providing us with an excellent education.

We would like to extend our special thanks to **Mrs. GHERNAOUT Rayane** for her availability and guidance.

We also would like to express our gratitude **Dr. KHETTAB Soufiane** for his huge aide in carrying out motor tests. Many thanks to our dear friends who helped us with their ideas, and suggestions.

All our appreciation for our beloved parents who supported us with their words, prayers, and actions and were always there in our help.

Finally, we would like to thank everyone who has contributed in any way to the completion of this work.

Contents

List of Figures

List of Tables

Table of Variables

Variable	Description	Unit
σ	Leakage factor	-
τ	Time constant	s
ζ	Damping coefficient	-
θ_e	Angle between d^e and d^s required to ensure rotor field orientation	rad
μ	Membership degree	-
ψ	Flux	Wb
B_r	Rotor magnetic field	T
B_s	Stator magnetic field	T
C_1,C_2	Acceleration coefficient	-
ce , Δe, ΔE	Change in error	-
ΔT_e*	Incremental change in output reference torque	N.m
ΔU	Incremental change in output	-
e, E	Error	-
f, F	Friction coefficient	N.m.s/rd
f_s	Synchronous frequency	Hz
f_s	Switching frequency	Hz
G_e	Error gain	-
G_{ce}	Change in error gain	-
G_{CL}	Closed loop transfer function	-
G_{cu}	Incremental change in output gain	-
G_{OL}	Open loop transfer function	-
I, i	Current	A
J	Moment of inertia	$Kg.m^2$
K_i	Speed controller integral gain	-
K_{ii}	Current controller integral gain	-
K_P	Speed controller proportional gain	-
K_{Pi}	Current controller proportional gain	-
L_m	Mutual inductance	H
L_r	Rotor inductance	H
L_s	Stator inductance	H
n_r	Rotor electrical speed	Rpm
n_s	Stator electrical speed	Rpm

n_{syn}	Synchronous speed	rad/s
p	Number of poles	-
P	Number of pole pairs	-
r_1,r_2	Uniformly random number in the interval of [0, 1]	-
R_r	Rotor resistance	Ω
R_s	Stator resistance	Ω
s	Slip	-
t	Time	s
$T_{\alpha\beta 0}$	Clark transformation matrix	-
$T_{dq0}(\theta_e)$	Park transformation matrix	-
T_e, T_e^*	Electromagnetic output torque and reference torque	N.m
T_{ind}	Induced torque	N.m
T_L	Load torque	N.m
T_r	Rotor time constant	s
T_s	Sampling time	s
U	Universe of discourse	-
V	Voltage	V
w_e, w_s	Stator speed	rad/s
w_i	Firing strength	-
w_n	Natural frequency	Hz
w_r	Rotor speed	rad/s
w_{sl}	Slip speed	rad/s
$x(1), ..., x(5)$	Membership functions boundaries	-

List of Abbreviations

AC	*Alternating Current*
COA	*Center of Area*
COG	*Center of Gravity*
DC	*Direct Current*
DTC	*Direct Torque Control*
FIS	*Fuzzy Inference Systems*
FLC	*Fuzzy Logic Controller*
FLC-PI	*Fuzzy Logic Controller-Proptional Integral*
FOC	*Field Oriented Control*
GWO	*Grey Wolf Optimizer*
IAE	*Integral of Absolute Error*
IFOC	*Indirect Field Oriented Control*
IM	*Induction Motor*
IRFOC	*Indirect Rotor Field Oriented Control*
ITAE	*Integral Time Absolute Error*
MFs	*Membership Functions*
MOM	*Mean of Maximum*
PI	*Propotional Integral*
PSO	*Particle Swarm Optimization*
SPWM	*Sinusoidal Pulse Width Modulation*
SVPWM	*Space Vector Pulse Width Modulation*
VFD	*Variable Frequency Drive*
V/F	*Voltage/Frequency*

General Introduction

These days, Induction motors serve as the foundation of various industrial and commercial applications, because of their simplicity, sturdy and robust design. However, they required effective controlling in high performance utilization and one of the well known methods used for that is the indirect vector control, widely recognized as field-oriented control (FOC). This later permit the decoupling between the electromagnetic torque and the rotor flux, allowing the induction motor to be controlled as separate DC . Meanwhile, The speed controller is the key component of electrical drives [1] [2].

Induction motor drives are often controlled by standard Proportional plus Integral (PI) for both speed and current control. They have a basic construction and can provide adequate performance over a wide variety of operations. Even though, the constant PI controller is altered for some working conditions, it is sensitive to fluctuation of parameters, large load disturbance, and other factors like magnetic field, temperature,...etc. This problem can be solved by copious approaches such as self-tuning PI, in this manner the PI is susceptible to system's parameters and it's difficult to develop such controller in the absence of accurate mathematical models and due to inevitable parameter and load variations. The previously pointed disadvantage encourages the substitution of the PI by other alternatives that are adaptive to changes and based on artificial intelligence like the fuzzy logic controller. Replacing these controllers should primarily aim to reduce the standard error of signals or employ a sophisticated powerful control algorithm [3].

Fuzzy logic has been applied recently to a wide range of control fields. The primary benefit of the fuzzy logic control method over traditional control techniques that is based on linguistic rules with an IF-THEN general structure, which is the foundation of human logic, and it does not require an exact mathematical model of a system. It can also operate with non-linearity of random complexity [4].

The optimization techniques represented by PSO and GWO algorithms are involved to provide better tuning parameters for the Fuzzy controller, which enhances the control performance of the Indirect Vector Control method.

This thesis is organized into five chapters :

Chapter 1: **Modeling of Induction Motor** , which concerns with crucial aspects about the Induction Motors. This includes understanding the basis of operation, trans-

formations such as Clark and Park, and the development of a mathematical model to represent motor behavior accurately. This chapter forms the cornerstone of the upcoming chapters.

Chapter 2: **Vector Control of IM** , This chapter focus on the realm of vector control, a sophisticated control method that desires a precise regulation of motor speed and torque based on attaining decoupled control of the rotor flux magnitude and the torque producing current, with improved dynamic performances. Our focus lies specifically on indirect vector control, which is costly-efficient compared to other methods.

Chapter 3: **Theory of Fuzzy Logic** , In this chapter fuzzy logic concept is presented. It elucidate the working principle of fuzzy logic, basic definitions, history, advantages, and its structure. It discusses how fuzzy logic controllers (FLCs) utilize linguistic rules to adaptively adjust control parameters based on real-time operating conditions, thereby enhancing control performance and robustness. Then, it delves into the application of the FLC as speed controller with the vector control of induction motor to see its performance comparing to the conventional controllers.

Chapter 4: **Optimization Techniques** , this chapter concentrates on the optimization techniques which are aimed at fine-tuning the performance of the fuzzy logic controller. Two different optimization algorithms are introduced : particle swarm optimization (PSO) and Grey Wolf Optimizer (GWO), starting from the theoretical analysis to its integration with the optimization of gains and mfs of the fuzzy controller seeking to provide more precise control and stable speed regulation, enhanced torque performances, and increased efficiency for superior control over induction motors.

Chapter 5: **Simulation and Results** , This chapter cover the simulation of all previous concepts. It starts with the Indirect Vector Control method using PI, then replacing the speed PI controller with Fuzzy-PI controller manually tuned, revealing their results and comparing them. After that, optimization algorithms PSO and GWO are employed to optimize the gains, Mfs and both using one fitness function. At the end, the robustness of the best optimized Fuzzy is tested in different abnormal Operating Conditions like changing rotor resistance (R_r), stator resistance (R_s), and the moment of inertia (J) and compared with conventional PI controller to prove its adequate performance over a variety of working conditions.

Chapter 1

Modeling of Induction Motor

1.1 Introduction

This chapter aims to research and present a thorough model for induction machine that can be used for both analysis and control design. It serves as a general overview of the subject, reviewing the fundamentals of induction machines, including their construction and basic notions. Before delving into the modeling elements, we will be concentrating on the mathematical relations defining the induction machine in several reference frames, including Clark and Park transformations.

1.2 Description of induction motor

The induction machine is essentially a polyphase AC machine with either the stator or the rotor connected to the AC power grid. The AC power supply is usually three-phase, though it can also be single-phase. Under all circumstances, the machine's primary winding configuration or the stator that is connected to the grid should produce a traveling field in the machine's airgap. The machine's non-grid secondary conductors, or the mover in general, will experience voltage induction due to this traveling field. The rotor receives A.C. currents if the secondary windings are closed.

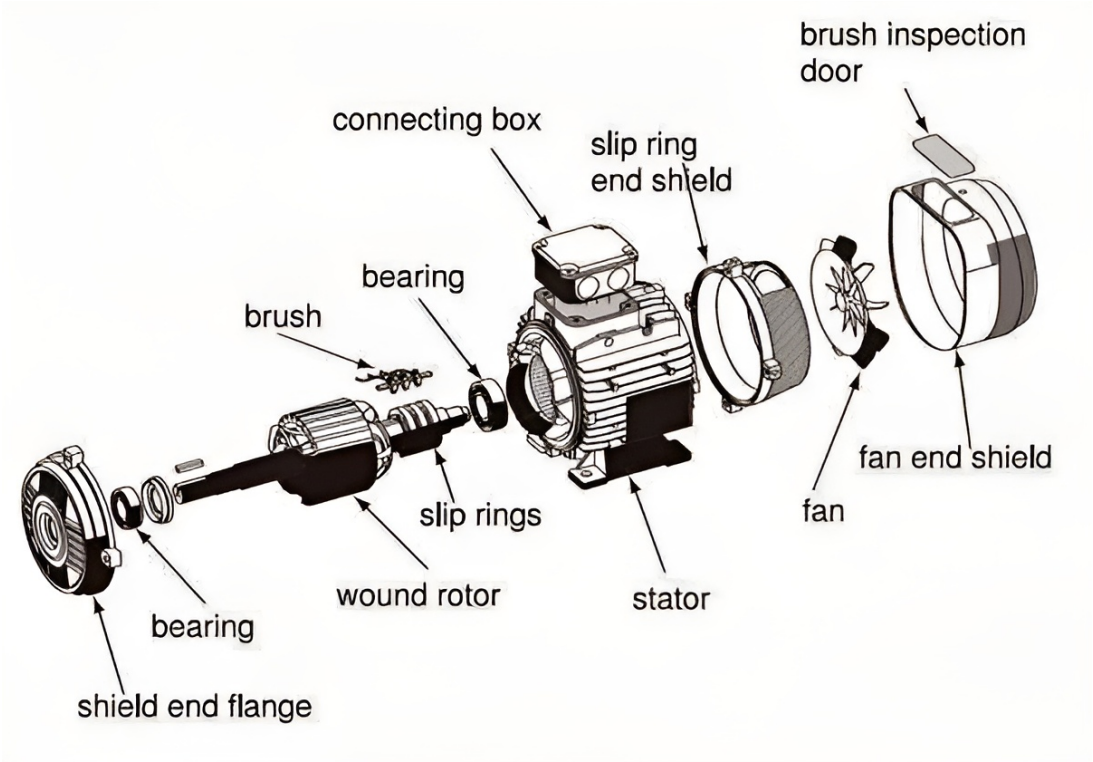


Figure 1.1: Cutaway view of an induction machine

At zero rotor speed, the interaction of the primary field and secondary currents produces torque. The ideal no-load (or synchronous) speed is the rotor speed at which the rotor currents are zero. The rotor winding might be composed of bars that are shortcircuited by end rings (cage rotors) or multiphase (wound rotors). The main and secondary

windings are arranged in identical slots that are stamped onto laminations, which are thin silicon steel sheets.[5]

1.3 Basic Induction motor concepts

The stator of the IM is connected to the 3-phase source and receives 3-phase set of voltages and currents which create a rotating magnetic field in the stator and by association will induce a current in the rotor.

The synchronous speed of the induction machine is defined by the frequency of the source and the number of poles of the particular machine and it's quantisized using the following equation :

$$n_{syn} = \frac{120 * f_s}{p} \tag{1.1}$$

Where:

n_{syn}: synchronous speed
f_s: synchronous frequency
p: number of poles of the IM

Slip (s) is defined as the difference between synchronous speed and operating speed, at the same frequency, expressed in rpm, or in percentage or ratio of synchronous speed.

$$s = \frac{n_s - n_r}{n_s} \tag{1.2}$$

Where :
n_s: stator electrical speed
n_r: rotor mechanical speed

The induced torque of the machine is given by:

$$T_{ind} = k.\vec{B_r}\vec{B_s} \tag{1.3}$$

Where:
k: is the motor constant
B_r: is the rotor magnetic field vector
B_s: is the stator magnetic field vector

1.4 Modelling the asynchronous machine

The structure of the induction machine under analysis consists of 3 identical phase windings positioned in a phase difference configuration of 120 electric degrees on the stator,

3 identical phase windings positioned in a similar phase difference on the rotor, and a constant air gap. (close slots in an optimal manner); an unsaturated (linear) magnetic circuit that permits the main and leakage inductances to be identified for every winding. Every phase winding has a harmonic distribution, W_s turns on the stator, and W_R turns on the rotor. Every inductance is regarded as constant. The machine's schematic view is displayed in fig 1.2. [5]

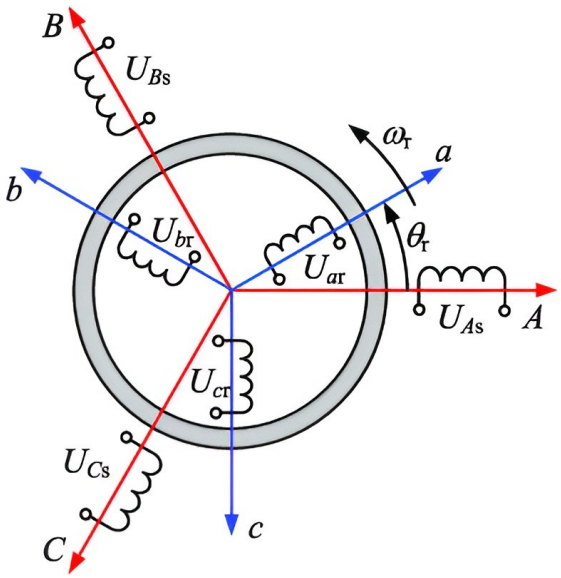


Figure 1.2: Representation of induction machine's windings

1.4.1 Mathematical model in the *abc* reference frame

The electrical equation of the induction motor in the *abc* reference frame is as follows:In the stator:

$$V_{as} = R_s i_{as} + \frac{d\psi_{as}}{dt} \tag{1.4}$$

$$V_{bs} = R_s i_{bs} + \frac{d\psi_{bs}}{dt} \tag{1.5}$$

$$V_{cs} = R_s i_{cs} + \frac{d\psi_{cs}}{dt} \tag{1.6}$$

We can write it in matrix form as follows

$$V_s = \frac{d}{dt} * \begin{bmatrix} \psi_{as} \\ \psi_{bs} \\ \psi_{cs} \end{bmatrix} + \begin{bmatrix} R_s & 0 & 0 \\ 0 & R_s & 0 \\ 0 & 0 & R_s \end{bmatrix} \begin{bmatrix} i_{as} \\ i_{bs} \\ i_{cs} \end{bmatrix} \tag{1.7}$$

In the rotor:

$$V_{ar} = R_r i_{ar} + \frac{d\psi_{ar}}{dt} = 0 \tag{1.8}$$

$$V_{br} = R_r i_{br} + \frac{d\psi_{br}}{dt} = 0 \tag{1.9}$$

$$V_{cr} = R_r i_{cr} + \frac{d\psi_{cr}}{dt} = 0 \tag{1.10}$$

We can write it in matrix form as follows

$$\begin{bmatrix} i_{ar} \\ i_{br} \\ i_{cr} \end{bmatrix} = \frac{d}{dt} * \begin{bmatrix} \psi_{ar} \\ \psi_{br} \\ \psi_{cr} \end{bmatrix} + \begin{bmatrix} R_r & 0 & 0 \\ 0 & R_r & 0 \\ 0 & 0 & R_r \end{bmatrix} \begin{bmatrix} i_{ar} \\ i_{br} \\ i_{cr} \end{bmatrix} \tag{1.11}$$

$$\psi_s = L_{ss} * i_s + L_{msr} * i_r \tag{1.12}$$

$$\psi_r = L_{rr} * i_r + L_{mrs} * i_s \tag{1.13}$$

$$L_{msr} = L_{ms} \begin{bmatrix} \cos(\theta_r) & \cos(\theta_r + \frac{2\pi}{3}) & \cos(\theta_r - \frac{2\pi}{3}) \\ \cos(\theta_r - \frac{2\pi}{3}) & \cos(\theta_r) & cos(\theta_r + \frac{2\pi}{3}) \\ cos(\theta_r + \frac{2\pi}{3}) & \cos(\theta_r - \frac{2\pi}{3}) & \cos(\theta_r) \end{bmatrix} = M(\theta_r) \tag{1.14}$$

$$L_{mrs} = L_{ms} \begin{bmatrix} \cos(\theta_r) & \cos(\theta_r - \frac{2\pi}{3}) & \cos(\theta_r + \frac{2\pi}{3}) \\ \cos(\theta_r + \frac{2\pi}{3}) & \cos(\theta_r) & cos(\theta_r - \frac{2\pi}{3}) \\ cos(\theta_r - \frac{2\pi}{3}) & \cos(\theta_r + \frac{2\pi}{3}) & \cos(\theta_r) \end{bmatrix} = M(-\theta_r) \tag{1.15}$$

The *abc* model can describe the behavior of the induction machine using six differential equations. These equations are intercoupled one to another by the mutual inductance between the different windings of the IM.The complexity of the system is due to the time dependence of the stator rotor coupling, or put simply, the position of the rotor .The induction motor ABC-model is subjected to mathematical transformations like "dq" or "$\alpha\beta$" in order to make the computation of the dynamic solution easier.[6]

1.4.2 Clark transformation:

It's the transformation of a stationary three-phase coordinate system to a stationary two-phase system through a space vector transformation where:[7]

$$[f_{\alpha\beta 0}] = T_{\alpha\beta 0}[f_{abc}] \tag{1.16}$$

where:

$$[f_{\alpha\beta 0}] = [f_\alpha f_\beta f_0]^T \tag{1.17}$$

and

$$[f_{abc}] = [f_a f_b f_c]^T \tag{1.18}$$

$$T_{\alpha\beta 0} = \frac{2}{3} \begin{bmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & \frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \end{bmatrix} \tag{1.19}$$

f represents voltage, current, and flux linkages
$T_{\alpha\beta 0}$ is the transformation matrix
The inverse transformation is given by:

$$T_{\alpha\beta 0}{}^{-1} = \begin{bmatrix} 1 & 0 & 1 \\ -\frac{1}{2} & \frac{\sqrt{3}}{2} & 1 \\ -\frac{1}{2} & -\frac{\sqrt{3}}{2} & 1 \end{bmatrix} \tag{1.20}$$

1.4.3 Park transformation

It is a change of variables that replaces variables such as voltages, currents, and flux linkages associated with fictitious windings rotating with the rotor. It refers the stator and rotor variables to a reference frame that is revolving at a random speed. From this reference frame's point of view, all the variables can be observed as constant values. With its special ability to remove all time-varying inductances caused by the spinning rotor from the voltage equations of three-phase ac machines, Park's transformation has revolutionized the field of machine analysis. Variable changes are used in the analysis of numerous static and constant parameters in power system components, in addition to being used in the analysis of ac machines to reduce time-varying inductances.[8]

$$[f_{dq0}] = T_{dq0}[f_{abc}] \tag{1.21}$$

The transformation matrix:

$$T_{dq0}(\theta_e) = \frac{2}{3}\begin{bmatrix} \cos(\theta_e) & \cos(\theta_e - \frac{2\pi}{3}) & \cos(\theta_e + \frac{2\pi}{3}) \\ \sin(\theta_e) & \sin(\theta_e - \frac{2\pi}{3}) & \sin(\theta_e + \frac{2\pi}{3}) \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \end{bmatrix} \tag{1.22}$$

where θ_e is the angular displacement of Park's reference frame and can be calculated by

$$\theta_e = (\int_0^t w_e(\zeta)d\zeta) + \theta(0) \tag{1.23}$$

Where w_e is the speed of rotation of the reference frame
The inverse park transformation is defined as follows:

$$[f_{abc}] = T_e(\theta_e)^{-1}[f_{dq0}] \tag{1.24}$$

1.4.4 Transformation between reference frames

It's normal to go between reference frames in the control system, and to achieve this we use the following transformations:

$$[f_{dq}] = \begin{bmatrix} \cos(\theta_e) & \sin(\theta_e) \\ -\sin(\theta_e) & \cos(\theta_e) \end{bmatrix} [f_{\alpha\beta}] \tag{1.25}$$

$$[f_{\alpha\beta}] = \begin{bmatrix} \cos(\theta_e) & sin(\theta_e) \\ \sin(\theta_e) & \cos(\theta_e) \end{bmatrix} [f_{dq}] \tag{1.26}$$

1.4.5 IM model in stationary $\alpha\beta$ reference frames

The IM is defined in the $\alpha\beta$ reference frame by the following equations:

$$v_{\alpha s} = R_s i_{\alpha s} + \frac{d\psi_{\alpha s}}{dt} \tag{1.27}$$

$$v_{\beta s} = R_s i_{\beta s} + \frac{d\psi_{\beta s}}{dt} \tag{1.28}$$

$$v_{\alpha r} = R_r i_{\alpha r} + \frac{d\psi_{\alpha r}}{dt} + w_r \psi_{\beta r} = 0 \tag{1.29}$$

$$v_{\beta r} = R_r i_{\beta r} + \frac{d\psi_{\beta r}}{dt} + w_r \psi_{\alpha r} = 0 \tag{1.30}$$

In this work, Copper losses are included but both magnetic losses and Harmonics are neglected.

1.4.6 IM model in rotating dq reference frames

The IM is defined in the dq reference frame, for an arbitrary speed w_e, by the following equations:

1) Electrical equations:

$$v_{ds} = R_s i_{ds} + \rho\psi_{ds} - w_e \psi_{qs} \tag{1.31}$$

$$v_{qs} = R_s i_{qs} + \frac{d}{dt}\psi_{qs} + w_e \psi_{ds} \tag{1.32}$$

$$v_{dr} = R_r i_{dr} + \rho\psi_{dr} - (w_e - w_r)\psi_{qr} = 0 \tag{1.33}$$

$$v_{qr} = R_r i_{qr} + \rho\psi_{qr} - (w_e - w_r)\psi_{dr} = 0 \tag{1.34}$$

2) Magnetic equations:

$$\psi_{ds} = L_s i_{ds} + L_m i_{dr} \tag{1.35}$$

$$\psi_{qs} = L_s i_{qs} + L_m i_{qr} \tag{1.36}$$

$$\psi_{dr} = L_r i_{dr} + L_m i_{ds} \tag{1.37}$$

$$\psi_{qr} = L_r i_{qr} + L_m i_{qs} \tag{1.38}$$

The equation of the electrical torque of the induction motor:[8]

$$T_e = \frac{3L_m P}{4}(i_{qs} i_{dr} - i_{ds} i_{qr}) \tag{1.39}$$

Rotor speed is computed in rad/sec using the following relation:

$$\frac{dw_r}{dt} = \frac{T_e - T_L - Bw_r}{J} \tag{1.40}$$

Where J is the moment of inertia of the rotor, T_L is the load torque, B is the friction coefficient.

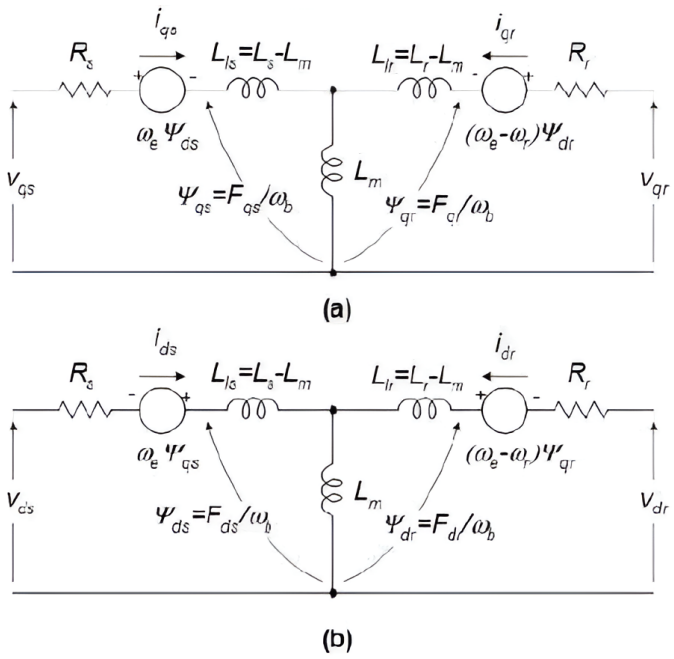


Figure 1.3: DQ modeling

From the equations above, we can ascertain that the speed of the IM is governed by a nonlinear differential equation, taking into consideration that the IM parameters (R_s, R_r, L_r and $L_s$are constant and are functions of w_r. The outputs of the aforementioned equations will change in accordance with the set speed of rotation of the reference frame. We can distinguish 3 distinct frames:

1) The model is referred to the stator frame when the speed of the reference is the same as that of the stator, thus $w_e = 0$.

2) The model is referred to the rotor frame: the speed of the reference frame is the same as that of the rotor $w_e = w_r$.

3) Model referred to the synchronously rotating reference frame: the speed of the reference frame is the same as the synchronous speed $w_e = w_s$.

1.5 Conclusion

This chapter dealt with the definition, reference frames, and mathematical modeling of Induction Machines. It represents the building block of the following chapters.The next chapter will discuss Vector Control which is a fundamental component of the effective control of IM.

Chapter 2

Vector Control of IM

2.1 Introduction

Asynchronous machine presents the benefit of being robust, costly efficient and simply designed. However, this simplicity is hidden behind a significant complication. From the dynamic model of induction motor, it is noticed the complexity of this later compared to the dc motor due to [9] :

1- The high level degree of the mechanical system in IM
2- The electromagnetic interaction between the stator and rotor
3- The inaccessibility to physical rotoric quantities in squirrel cage IM
4- Non linearity of the system

All these reasons raised a need for urgent practical solutions and therefor, several new techniques were introduced on the late of the last century such as : V/F, DTC, Scalar and Vector control methods. In this chapter focus on the vector control that has proven its reliability over the past years.

2.2 Vector Control Method

Vector control of induction motor is seen as the most used technique for the training of variable speed drives of asynchronous machines. It allows high performance command for the speed and torque, with excellent statically and dynamically functioning as well as great transitional mechanism. The main aim of this method is to maintain the stator field and rotor field perpendicular to each other to produce the maximum torque. So that it is possible to control, directly and separately, the Induction Motor's torque and flux. There are actually two methods of vector control. The first one called the direct method, was invented by Blaschke and the second one, known as the indirect was invented by Hasse [10]. Both methods vary in the way the rotor angle is calculated.

2.2.1 Direct Vector Control

In this type of FOC, Hall sensors or flux estimators are used to measure the rotor flux. Then, the rotor angle is acquired from the following formula [11] :

$$\theta_e = \tan^{-1}(\frac{\psi dr}{\psi qr}) \tag{2.1}$$

It's commonly recognized as feedback vector control scheme. Multiple controllers have been applied on this method to ameliorate its performance. Even though it's the most recommended method, it still deteriorates in certain points which are the high expense and the unreliability of the flux measurements [10].

2.2.2 Indirect Rotor Field Oriented Control

It is commonly abbreviated IRFOC, in this type the awareness of the rotor flux is not necessary. Thus, the flux is controlled in open loop and its reference value is either maintained constant along the control process or imposed by field weakening bloc which is known as "defluxage" [12]. Despite the fact that the flux is not controlled, on the contrary its position and reaction that ensure the decoupling between the torque and flux is evaluated with the following formula :

$$\theta_e = \int w_e dt = \int (pw_r + w_{sl})dt \tag{2.2}$$

where w_{sl} : slip frequency

As might be noticed, this method is quite appealing and cost-effective, thus accordingly it is more frequently used [9].

2.3 Derivation of IRFOC

we have developed in the previous chapter the equations governing the induction machine in the dq reference frame and in this part we are going to build upon that. As it was mentioned the basic concept of the IRFOC is to align the rotor flux to either the d or q axis. In our application we align the rotor flux with d axis which will result in the development of the following relationships:

$$\psi_{qr} = 0 \tag{2.3}$$

$$\psi_{dr} = \psi_r = constant \tag{2.4}$$

Thus :

$$\rho\psi_r = 0 \tag{2.5}$$

Thus the equation (1.33) becomes :

$$v_{dr} = R_r i_{dr} = 0 \tag{2.6}$$

$$i_{dr} = 0 \tag{2.7}$$

and the equation (1.34) becomes:

$$v_{qr} = R_r i_{qr} + w_{sl}\psi_r = 0 \tag{2.8}$$

where :

$$w_{sl} = w_s - w_r \tag{2.9}$$

Thus equation (1.39) becomes:

$$T_e = \frac{3}{2}\frac{P}{2}\frac{L_m}{L_r}(\psi_{dr} i_{qs}) \tag{2.10}$$

$$i_{ds} = \frac{\psi_r}{L_m} \tag{2.11}$$

from equation (1.38)

$$i_{qr} = -\frac{L_m}{L_R} iqs \tag{2.12}$$

$$w_{sl} = -R_r \frac{i_{qr}}{\psi_{dr}} \tag{2.13}$$

thus:

$$w_s = w_r + R_r \frac{\frac{L_m}{L_r} i_{qs}}{\psi_{dr}} \tag{2.14}$$

in the end we get:

$$w_s = w_r + \frac{L_r}{R_r} \frac{i_{qs}}{i_{ds}} \tag{2.15}$$

$$T_e = \frac{3}{2} \frac{P}{2} \frac{L_m}{L_r} \psi_{dr} i_{qs} \tag{2.16}$$

$$\psi_{dr} = L_m i_{ds} \tag{2.17}$$

This control scheme is further explained by the fig 2.1

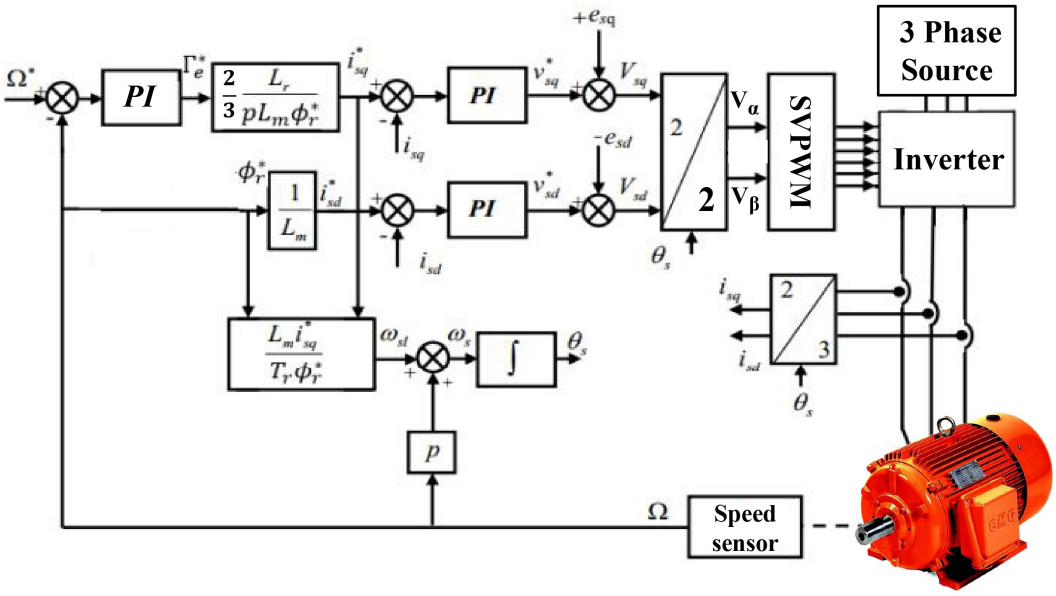


Figure 2.1: Block diagram of IRFOC

a) The decoupling between input and output

For voltage-supplied asynchronous machines, couplings between actions on axes d and q axes are presented by vector control laws. Decoupling is required because flux and torque depend on the voltages V_{ds} and V_{qs} concurrently. In order to represent the process as a collection of mono-variable systems evolving concurrently, the goal is, to the greatest extent feasible, to restrict the effect of an input to a single output. Controls are then non-interactive. Different decoupling techniques exist, such as using a controller, decoupling by state feedback, or decoupling by compensation. We are interested in decoupling by compensation.

b) Decoupling by compensation

It is paramount to define two new variables V_{ds}^* and V_{qs}^* where

$$V_{ds} = V_{ds}^* - e_{qs} \tag{2.18}$$

and

$$V_{qs} = V_{qs^*} - e_{ds} \tag{2.19}$$

with

$$e_{qs} = w_s \sigma L_s i_{qs} \tag{2.20}$$

$$e_{ds} = -(w_s \sigma L_s i_{ds} + w_s \frac{L_m}{L_r} \frac{d\psi_r}{dt}) \tag{2.21}$$

Thus, the voltages V_{ds}^* and V_{qs}^* are reconstructed from voltages V_{ds}^* and V_{qs}^* such that

$$V_{ds}^* = \sigma L_s \frac{di_{ds}}{dt} + (R_s + R_r \frac{{L_m}^2}{{L_r}^2}) i_{ds} \tag{2.22}$$

$${V_{qs}}^* = \sigma L_s \frac{di_{qs}}{dt} + w_s \sigma L_s i_{ds} \tag{2.23}$$

Once the two voltages are compensated, the flux will be controlled by V_{ds} while the torque will depend only on V_{qs}.
Using Park transformation, these output voltages (V_{ds},V_{qs}) are converted to V_α, and V_β which will be the inputs of the PWM controller.

2.4 Integration of SVPWM

To overcome the complexity of modulation required by the SPWM, a more effective approach is a technique known as SVPWM. Space vector pulse width modulation (SVPWM) is a method used in complex power electronics systems . It is an advanced technique that is extensively used in the precise and effective control of motor drive systems. To determine the duty cycles of the power switches in the VFD, SVPWM converts the desired voltage reference signal into a sequence of switch control signals. This is accomplished by simulating the three-phase voltage system in the $\alpha\beta$ plane, a two-dimensional rotating space vector. The stationary reference frame, usually the motor's stator reference frame, is where the $\alpha\beta$ plane is formed from.
The modulation period is divided into a number of smaller time intervals known as sectors by the SVPWM algorithm. Every sector is associated with a certain set of voltage vector combinations that the VFD is capable of producing. To get the specified output voltage, the algorithm determines how many duty cycles are needed for each sector.

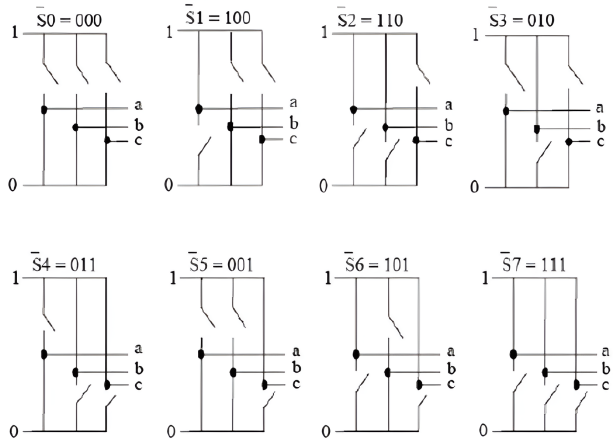


Figure 2.2: Inverter states

In the SVPWM technique, the three-phase voltage references are represented as a space vector v_{abc} in the complex plane, and this voltage reference vector is modulated by output voltage vectors available from an inverter. The SVPWM technique is now widely used in many three-phase inverter applications because it produces fundamental output voltage and gives less harmonic distortion of the load current, lower torque ripple in AC motors, and lower switching losses.[13]

Table 2.1: Relationship between voltages and switching vectors

Switch States			**Phase Voltages**			Space Voltage Vector
S_a	S_b	S_c	V_{as}	V_{bs}	V_{cs}	$V_n(n=1-7)$
0	0	0	0	0	0	$V_0 = 0\angle 0$
1	0	0	$\frac{2}{3}V_{dc}$	$-\frac{1}{3}V_{dc}$	$-\frac{1}{3}V_{dc}$	$V_1 = \frac{2}{3}V_{dc}\angle 0$
1	1	0	$\frac{1}{3}V_{dc}$	$\frac{1}{3}V_{dc}$	$-\frac{2}{3}V_{dc}$	$V_2 = \frac{2}{3}V_{dc}\angle 60$
0	1	0	$-\frac{1}{3}V_{dc}$	$\frac{2}{3}V_{dc}$	$-\frac{1}{3}V_{dc}$	$V_3 = \frac{2}{3}V_{dc}\angle 120$
0	1	1	$-\frac{2}{3}V_{dc}$	$\frac{1}{3}V_{dc}$	$\frac{1}{3}V_{dc}$	$V_4 = \frac{2}{3}V_{dc}\angle 180$
0	0	1	$-\frac{1}{3}V_{dc}$	$-\frac{1}{3}V_{dc}$	$\frac{2}{3}V_{dc}$	$V_5 = \frac{2}{3}V_{dc}\angle 240$
1	0	1	$\frac{1}{3}V_{dc}$	$-\frac{2}{3}V_{dc}$	$\frac{1}{3}V_{dc}$	$V_6 = \frac{2}{3}V_{dc}\angle 300$
1	1	1	0	0	0	$V_0 = 0\angle 0$

The aforementioned table shows the relation governing the switching vectors and voltages. Sectors are separated by 60° angles and from the eight defined states we recognize six as active voltage vectors ($V_1$to V_6 while V_0 and V_7 are known as the zero voltage vector. The magnitude and angle of the reference voltage vector is calculated using the following equation:

$$|V_{ref}| = \sqrt{{V_\alpha}^2 + {V_\beta}^2} \tag{2.24}$$

$$\alpha = \tan^{-1}\frac{\upsilon_\beta}{\upsilon_\alpha} \tag{2.25}$$

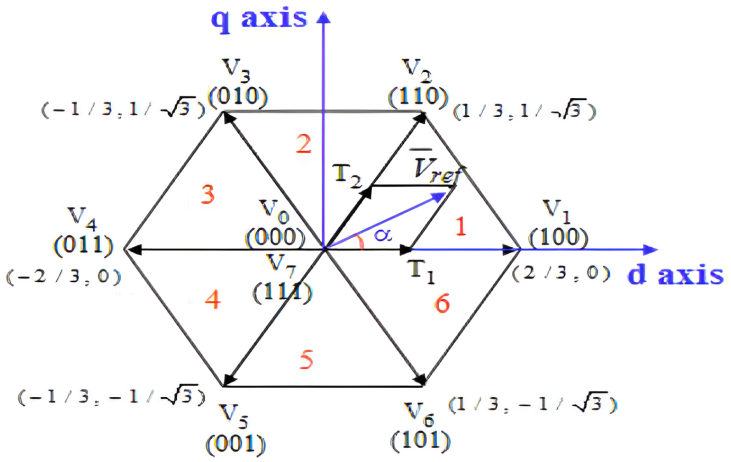


Figure 2.3: space vector diagram

The aforementioned angle α is the decider of which sector is the reference vector V_{ref} at.

The sampling time is given by:

$$T_s = \frac{1}{f_s} \tag{2.26}$$

the space vector in sector 1 is applied by the non-zero vector V_1 and V_2 during the T_1 and T_2 time durations, and the zero vector V_0 during the T_0 time duration[14]

$$\int_0^{T_s} \vec{V_{ref}}\, dt = \int_0^{T_1} \vec{V_1}\, dt + \int_{T_1}^{T_1+T_2} \vec{V_2}\, dt + \int_{T_1+T_2}^{T_s} \vec{V_0}\, dt \tag{2.27}$$

Where f_s and T_s are the frequency and the switching time respectively. the equations provided can be used to determine the time duration of switching vectors in each sector

$$T_1 = \frac{\sqrt{3}T_s|V_{ref}|}{V_{dc}} \sin(\frac{n}{3}\pi - \alpha) \tag{2.28}$$

$$T_2 = \frac{\sqrt{3}T_s|V_{ref}|}{V_{dc}} \sin(\alpha - \frac{n-1}{3}\pi) \tag{2.29}$$

$$T_0 = T_s - (T_1 + T_2) \tag{2.30}$$

The switching times of the six switches and the sector definition is depicted below

Table 2.2: Switching Time of Sectors

Sector	Upper Switches (S1, S3, S5)	Lower Switches (S4, S6, S2)
1	$S_1 = T_1 + T_2 + T_0/2$ $S_3 = T_2 + T_0/2$ $S_5 = T_0/2$	$S_4 = T_0/2$ $S_6 = T_1 + T_0/2$ $S_2 = T_1 + T_2 + T_0/2$
2	$S_1 = T_1 + T_0/2$ $S_3 = T_1 + T_2 + T_0/2$ $S_5 = T_0/2$	$S_4 = T_2 + T_0/2$ $S_6 = T_0/2$ $S_2 = T_1 + T_2 + T_0/2$
3	$S_1 = T_0/2$ $S_3 = T_1 + T_2 + T_0/2$ $S_5 = T_2 + T_0/2$	$S_4 = T_1 + T_2 + T_0/2$ $S_6 = T_0/2$ $S_2 = T_1 + T_0/2$
4	$S_1 = T_0/2$ $S_3 = T_1 + T_0/2$ $S_5 = T_1 + T_2 + T_0/2$	$S_4 = T_1 + T_2 + T_0/2$ $S_6 = T_2 + T_0/2$ $S_2 = T_0/2$
5	$S_1 = T_2 + T_0/2$ $S_3 = T_0/2$ $S_5 = T_1 + T_2 + T_0/2$	$S_4 = T_1 + T_0/2$ $S_6 = T_1 + T_2 + T_0/2$ $S_2 = T_0/2$
6	$S_1 = T_1 + T_2 + T_0/2$ $S_3 = T_0/2$ $S_5 = T_1 + T_0/2$	$S_4 = T_0/2$ $S_6 = T_1 + T_2 + T_0/2$ $S_2 = T_2 + T_0/2$

The main advantage of the SVPWM technique is to enable consistent switching while decreasing the switching losses.The switching sequence is organized such that only one inverter leg is switched when transitioning from one state to the next. This switching paradigm results in minimization of the switching frequency and achievement of optimal harmonic performance for each power device,.

2.5 Conclusion

In this chapter, the concept of vector control has been presented, and the indirect vector-controlled induction motor drive has been derived. Two stages of decoupling are required. The first one is to ensure the decoupling between i_{ds} and i_{qs}, and the second one eliminates the coupling between the control components V_{ds} and V_{qs}. At the end, these control outputs are converted to the stationary reference frame components. As it can be noticed, the control of current and speed requires the use of PI controllers. The latter can be conventional, band, or AI techniques. The next chapter focuses on the use of FL to design PI controllers.

Chapter 3

Theory of Fuzzy Logic

3.1 Introduction

As explained previously in chapter 2, Although Vector control has proven its performance among several other techniques and progressed significantly, It still faces some difficulties and challenges when it comes to variation of motor's parameters as the rotor resistance, inertia moment,...etc. This later raised an important problem to deal with and become the main interest of power technicians which force the researchers to find new techniques based on artificial intelligence. One of these techniques known is the fuzzy logic. It is widely used in the control of induction machines and the vector control [9].

In this chapter, we will present the general principle of fuzzy logic theory, some basic definitions, its history and advantages, then we integrate the fuzzy logic controller (FLC) with the vector control of induction motor to assess its performance comparing to the conventional controllers.

3.2 Principle and History of Fuzzy Logic

Fuzzy logic is based on the theory of fuzzy sets with a highly developed mathematical formalism [15]. Its first proposals appeared in the 40s by American researchers, however, the concept of fuzzy sets was proposed, at the first time, by **Lotfi ZADEH** [16] in 1965. His work made its initial appearance in 1974 where it has been realized in industrial application by **M.Mamdani** in the regulation of boiler. But the true exponent of this method started in Japan in 1980 by **TAKAGI-SUGENO** [17, 9]. After that, it has been applied in multiple areas such as : Economics, Medicine, Robotics, household appliances, cement plant conduits,...etc.
Fuzzy logic is distinguished from the conventional logic with its ability of processing the unclear and blurred information, which are frequently faced in the case of nonlinear systems [9].

3.3 Advantages and Disadvantages of Fuzzy Logic

The fuzzy logic brings together a number of advantages and weaknesses, which are as follows [18] :

3.3.1 Advantages

1) The ability to implement (linguistic) knowledge of the process operator.

2) Process mastery with complex behavior (highly non-linear).

3) It has human-like intuition, which allows it to adjust powerfully to control challenges.

4) Frequently obtaining better dynamic benefits (non-linear regulator).

3.3.2 Disadvantages

1) The lack of precise guidelines for the design of its parameters (choice of measurement sizes, determination of fuzzification, inferences and defuzzification).

2) The artisanal and non-systematic approach (implementation of knowledge of operators often difficult).

3) The impossibility of demonstrating the stability of its control law in general (in the absence of a valid model).

4) Consistency of inferences not guaranteed (possible appearance of contradictory inference rules).

3.4 Definitions and Basics

3.4.1 Fuzzy Set and Linguistic Variable

The theory of fuzzy logic is based on the concept of linguistic variable. It might be challenging to precisely define concept like medium temperature. The linguistic value medium is not accurate but thresholds can be set and considered to assign this or that qualifier based on the value if the variable in relation to these thresholds. The aspect of the word Medium is fuzzy, we can define the membership degree for the variable of temperature as medium. This later can take real values between 0 and 1.
Contrary, in boolean logic, the membership degree μ can only take two values either 0 or 1 [17].

• **Universe of Discourse**
We define the universe of discourse as the set of numerical or real values which can take the fuzzy variable, it is noted as **U**. For the fuzzy variable x, we define the fuzzy set A in the universe of discourse **U** with membership degree function [15]:
It's noted :

$$\begin{aligned} \mu_A : U &\Longrightarrow [0,1] \\ x &\Longrightarrow \mu_A(x) \end{aligned} \tag{3.1}$$

• **Membership Degree**
As mentioned earlier, in the boolean logic, there are only two possible values for the membership degree μ: 0 or 1 which means it can not take two qualifiers at the same time, we return back to the example of temperature (fig 3.1), if it is low, the following membership degree is considered [15] :

$$\mu_{low} = 1,\ \mu_{medium} = 0,\ \mu_{high} = 0 \tag{3.2}$$

Example :

- Temperature is low so **U**(low)=[0°C, 15°C]

- Temperature is medium so **U**(medium)=[15°C, 25°C]
- Temperature is high so **U**(high)=[25°C, 40°C]

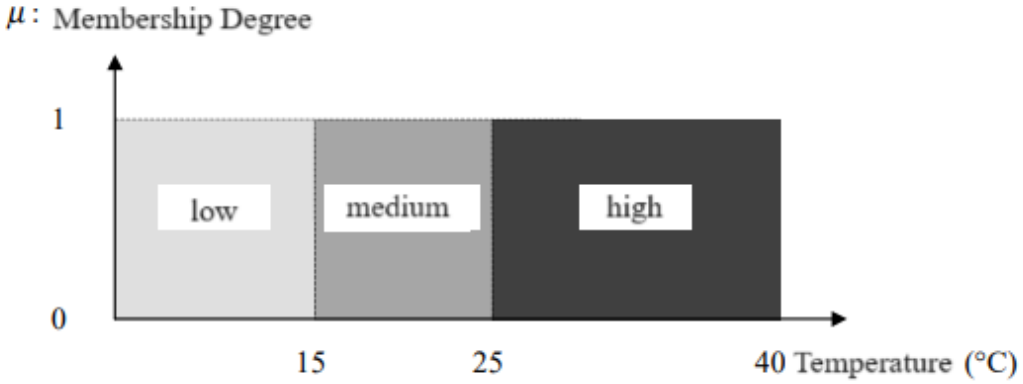


Figure 3.1: Example of Sets in Boolean Logic

On the other hand, in the fuzzy logic, the membership degree becomes a function taking a value between 0 and 1, thus, a temperature of 17°C can be considered to be of low with membership degree of 0.2 and as medium with membership degree of 0.8 (fig 3.2).

$$\mu_{low} = 0.2,\ \mu_{medium} = 0.8,\ \mu_{high} = 0 \tag{3.3}$$

Example :

- Temperature is low so **U**(low)=[0°C, 18°C]
- Temperature is medium so **U**(medium)=[12°C, 28°C]
- Temperature is high so **U**(high)=[22°C, 40°C]

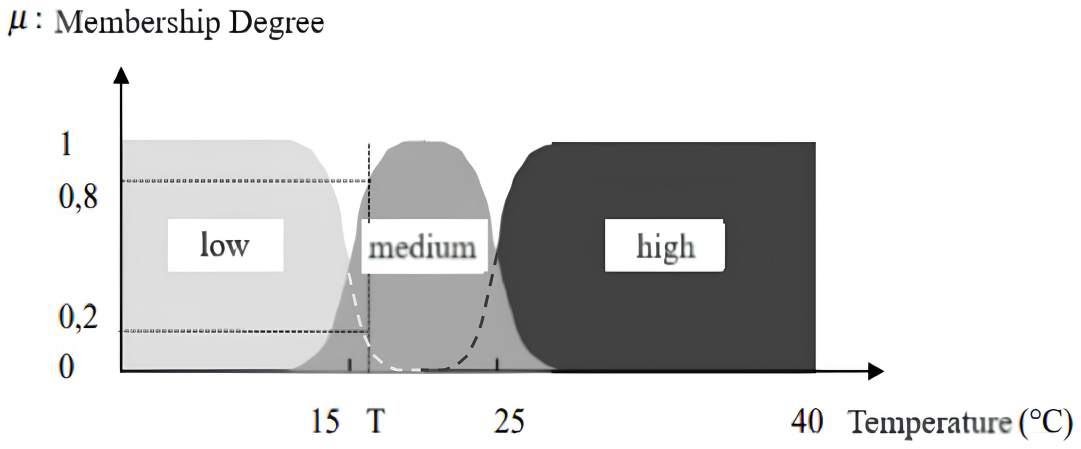


Figure 3.2: Example of Sets in Fuzzy Logic

3.4.2 Different Forms of Membership Functions

Usually to define the fuzzy sets, several membership functions can be used which are as follows [17] :

1) **Triangular Function :** this function is defined by three parameters which are {a,b,c} so :

$$\mu(x,a,b,c) = \begin{cases} 0, & a < x \\ \frac{x-a}{b-a}, & a \leq x \leq b \\ \frac{c-x}{c-b}, & b \leq x \leq c \\ 0, & otherwise \end{cases} \tag{3.4}$$

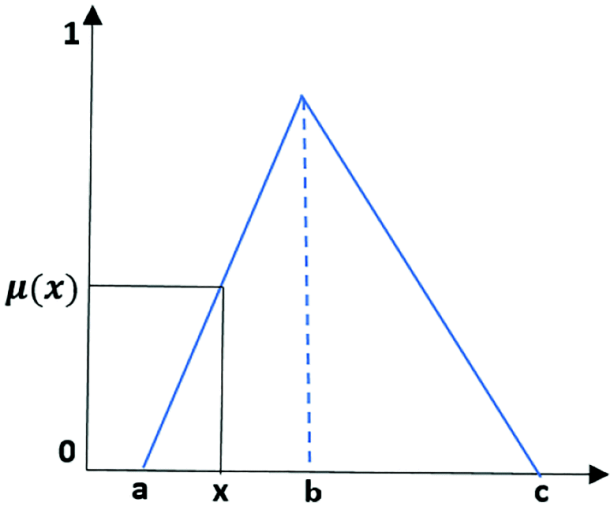


Figure 3.3: Triangular Membership Function

2) **Trapezoidal Function :** this function is defined by three parameters which are {a,b,c,d} so :

$$\mu(x,a,b,c,d) = \begin{cases} 0, & x < a \\ \frac{x-a}{b-a}, & a \leq x \leq b \\ 1, & b \leq x \leq c \\ \frac{d-x}{d-c}, & c \leq x \leq d \\ 0, & otherwise \end{cases} \tag{3.5}$$

3) **Gaussian Function :** this function is defined by two parameters which are {q,c} and it is noted by :

$$\mu(x,q,c) = \exp(\frac{-(x-c)^2}{2q}) \tag{3.6}$$

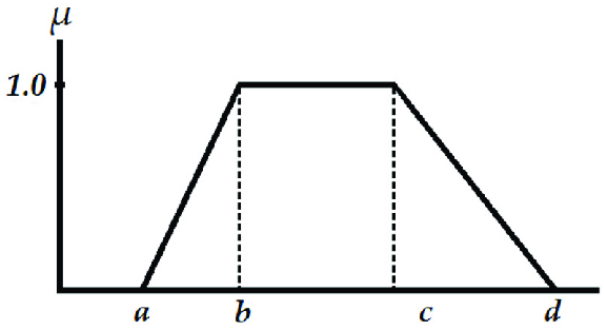


Figure 3.4: Trapezoidal Membership Function

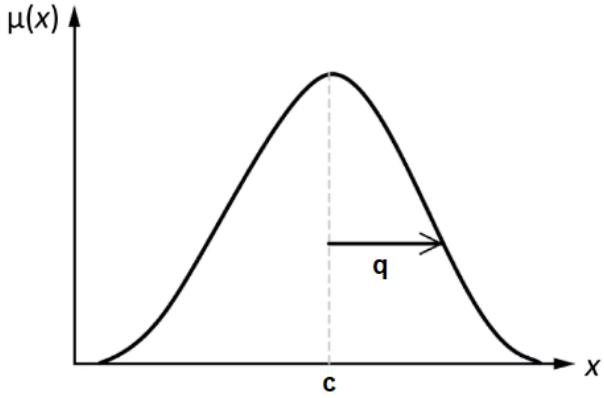


Figure 3.5: Gaussian Membership Function

4) **Singleton Function :** this function is defined by one parameter which is {c} and it is noted by :

$$\mu(x, c) = \begin{cases} 1, & x = c \\ 0, & otherwise \end{cases} \tag{3.7}$$

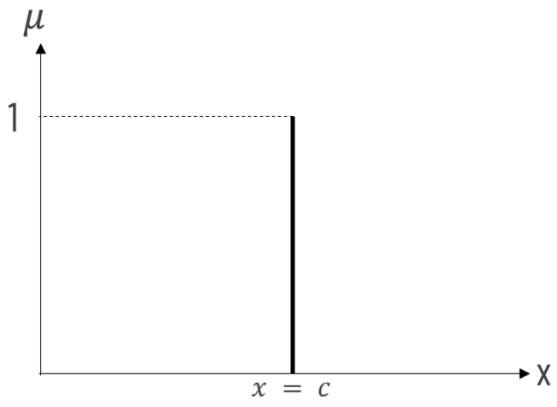


Figure 3.6: Singleton Membership Function

3.4.3 Fuzzy Logic Operators

Let's assume the membership functions μ_A *and* μ_B of the fuzzy sets A and B respectively, defined on the universe of discourse W. As in the theory of classical sets, we define the intersection $\cap$, the union $\cup$ of fuzzy sets as well as the complementary sets. These relationships are translated by the operators like AND, OR and NOT. New membership functions linked to these operators are established [15]:

- x belongs to A **AND** B $\Longleftrightarrow x \in \text{A} \cap \text{B} \Longleftrightarrow \mu_{A \cap B}(x)$
- x belongs to A **OR** B $\Longleftrightarrow x \in \text{A} \cup \text{B} \Longleftrightarrow \ \mu_{AUB}(x)$
- x belongs to the **complement of A** $\Longleftrightarrow x \in \bar{A} \Longleftrightarrow \mu(x)$

Example :

Let us take the previous sets, A characterizes the temperature sensor and B the humidity sensor; here is an example of "if the indoor air temperature is higher and moisture is strong" or "if temperature is high and moisture is high," indicating that you need to turn on the ventilation ventilators and coolers. As a result, you can see that the operators AND and OR in addition to the NOT appear and must be explicitly defined. Evaluation of the circumstances is required before determining which location to operate "on the coolers or the ventilates or both at the same time.

1) **AND Operator :** The AND operator is the intersection between two membership functions μ_A *and* μ_B. It can be realized by the function Min or the arithmetic function Product:

$$\mu_{A \text{ and } B}(x) = (\mu_A \cap \mu_B)(x) = min\{\mu_A(x),\ \mu_B(x)\} = \ \mu_A(x).\mu_B(x) \tag{3.8}$$

It is worth noting that the operator $\cap$ is commutative, associative and distributive.

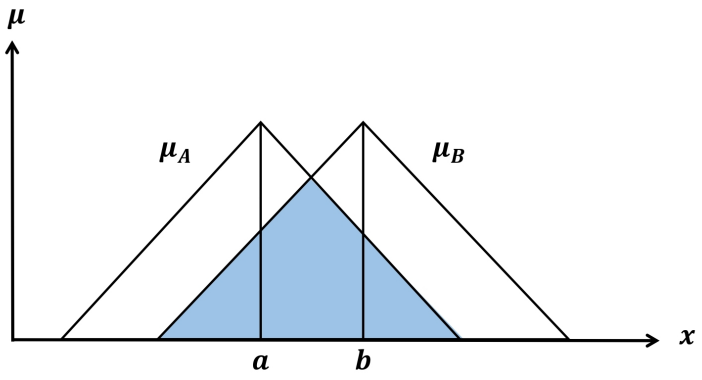


Figure 3.7: Intersection Between Two Membership Functions

2) **OR Operator :** The OR operator is the union of two membership functions μ_A *and* μ_B. It can be realized by the function Max or the arithmetic function Addition:

$$\mu_{A\ or\ B}(x) = (\mu_A \cup \mu_B)(x) = max\{\mu_A(x),\ \mu_B(x)\} = \ \mu_A(x)\ +\ \mu_B(x) \tag{3.9}$$

It is worth noting that the operator $\cup$ is commutative, associative and distributive.

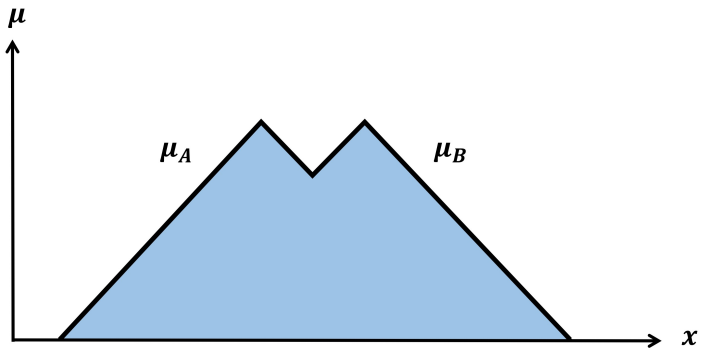


Figure 3.8: Union Between Two Membership Functions

3) **NOT Operator :** The NOT operator is characterized in fuzzy logic by the following function :

$$\mu(x) = \mathbf{NOT}\ \mu_A(x)\ = 1\ -\ \mu_A(x) \tag{3.10}$$

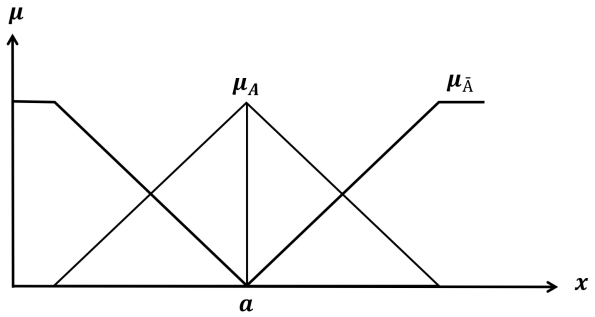


Figure 3.9: Complement of Membership Function

3.4.4 Deduction of Inferences

They can be described with different manners: linguistic, symbolic and using inference matrix [9].

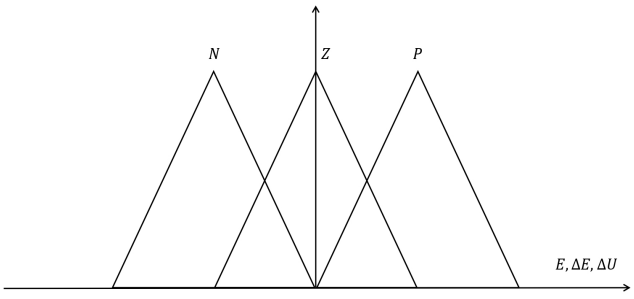


Figure 3.10: Fuzzy Sets

- **Linguistic Description :** the rules are written as follows :
  IF (E is positive) **AND** (ΔE is positive) **THEN**, (ΔU is positive) **OR**
  IF (E is positive) **AND** (ΔE is zero) **THEN**, (ΔU is positive) **OR**
  IF (E is positive) **AND** (ΔE is negative) **THEN**, (ΔU is zero) **OR**
  IF (E is zero) **AND** (ΔE is positive) **THEN**, (ΔU is positive) **OR**
  IF (E is zero) **AND** (ΔE is zero) **THEN**, (ΔU is zero) **OR**
  IF (E is zero) **AND** (ΔE is negative) **THEN**, (ΔU is negative) **OR**
  IF (E is negative) **AND** (ΔE is positive) **THEN**, (ΔU is zero) **OR**
  IF (E is negative) **AND** (ΔE is zero) **THEN**, (ΔU is negative) **OR**
  IF (E is negative) **AND** (ΔE is negative) **THEN**, (ΔU is negative) **OR**

- **Symbolic Description :**the previous rules can be simplified using the symbolic language with the substitution of the sets by abbreviations as follows :
  IF (E is P) **AND** (ΔE is P) **THEN**, (ΔU is P) **OR**
  IF (E is P) **AND** (ΔE is Z) **THEN**, (ΔU is P) **OR**
  IF (E is P) **AND** (ΔE is N) **THEN**, (ΔU is Z) **OR**
  IF (E is Z) **AND** (ΔE is P) **THEN**, (ΔU is P) **OR**
  IF (E is Z) **AND** (ΔE is Z) **THEN**, (ΔU is Z) **OR**
  IF (E is Z) **AND** (ΔE is N) **THEN**, (ΔU is N) **OR**
  IF (E is N) **AND** (ΔE is P) **THEN**, (ΔU is Z) **OR**
  IF (E is N) **AND** (ΔE is Z) **THEN**, (ΔU is N) **OR**
  IF (E is N) **AND** (ΔE is N) **THEN**, (ΔU is N) **OR**

- **Inference Matrix Description :** the previous rules can be arranged in a compact form as table such that the intersection of the row (Error) with the column (Change of error) gives the fuzzy set of the output shown on the Table (3.1).

Table 3.1: Inference Matrix

ΔU		**ΔE**		
		N	**Z**	**P**
E	**N**	**N**	**N**	**Z**
	Z	**N**	**Z**	**P**
	P	**Z**	**P**	**P**

3.5 Types of Fuzzy Inference Systems

Fuzzy logic offers a powerful tool for dealing with complex systems. Fuzzy inference systems (FIS) translate these fuzzy concepts into decision-making processes. This summary explores two prominent FIS approaches: Mamdani and Sugeno [19].

3.5.1 Mamdani Version

Developed as one of the first fuzzy control systems, the Mamdani method excels at capturing expert knowledge. It leverages linguistic rules and fuzzy outputs, allowing for easy interpretation of the results before defuzzification (converting fuzzy outputs to crisp values). This makes Mamdani a popular choice for decision support applications where understanding the reasoning behind the decision is crucial. However, defuzzification can be computationally expensive.

3.5.2 Sugeno Version

It offers a computationally efficient alternative. It utilizes crisp outputs, either constant values or linear functions of the inputs. This eliminates the need for defuzzification, leading to faster processing. Additionally, Sugeno's compatibility with optimization and adaptation techniques makes it well-suited for control problems, particularly those involving dynamic and non-linear systems.

3.5.3 Key Difference

The core distinction lies in output generation. Mamdani employs defuzzification to convert fuzzy outputs into crisp values, while Sugeno uses weighted averages of pre-defined crisp outputs. This translates to Sugeno's membership functions being linear or constant, unlike Mamdani's fuzzy sets.

3.5.4 Sugeno's advantages

1) Computational Efficiency: faster processing due to the elimination of defuzzification stage.

2) Compatibility with Optimization and Adaptation: Well-suited for control problems involving dynamic systems.

3) Mathematical Analysis: Easier to analyze mathematically due to the use of crisp outputs.

3.6 Structure of Fuzzy Logic Controller

Unlike Conventional PI regulator, the fuzzy controller does not deal with a well-defined mathematical relationships, it uses inferences with multiple rules based on linguistic variables, these inferences makes use of specific operators related to fuzzy. The diagram in (figure 3.11) gives the internal structure of a fuzzy logic controller (FLC), its processing passes through three main parts : **Fuzzification, Rule base and inference engine, Defuzzification** [20].

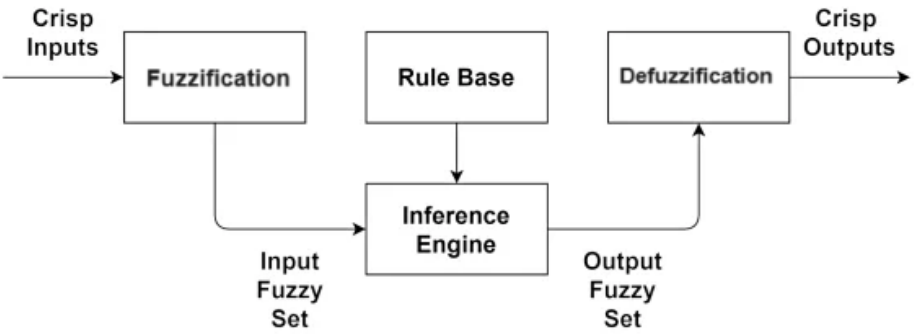


Figure 3.11: Internal Structure of Fuzzy Logic Controller

3.6.1 Fuzzification

At this step, the numerical entries (crisp) are transformed to linguistic variables (fuzzy) inputs, which means the fuzzification is the process of mapping linguistic labels of fuzzy collections to errors (e) and changes in errors (Δe) or (ce). Every label has a membership function that is linked to each physical crisp input variable [15] [9]. For the sake of simplification, 3 labels are used in this section Negative (N), Zero (Z) and Positive (P), further details about the suggested controller can be found later in the section of Application of Fuzzy Logic Controller to Vector Control.

Figure 3.12: Fuzzification Stage

3.6.2 Rule Base and Inference Engine

The main objective of this step is to determine the fuzzy outputs of the controller from fuzzy inputs resulting from the fuzzification. It is decomposed to : data base and decision logic [17].

a) Data base : This part contains generally all the data that is allowing the inference of the numerical output from a numerical input, which are : inputs-outputs variables and the inference rules.

1) **Inputs-Outputs Variables :** they are considered as numerical values, and attached to linguistic values, which has the following parameters :

- Name of the linguistic variable
- The linguistic value

- The universe of discourse
- The membership function
- The distribution over the universe of discourse

Example :
e : error
Δe : Change of error
Such that : Δe= e(t) - e(t-1)

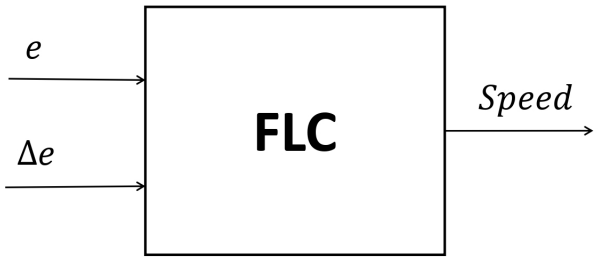


Figure 3.13: Example

2) **Rule Base :** The rules of inference are interpreted as follows:
Ri : **If**($X is$A$_i$) **AND** (Y is B_j) **Then** (Z is C_k).
$Such that$:i = 1,....,N$_A$; $j = 1, ..., N_B$; $k = 1, ..., N_C$; and Ri is the number of rules.
N_A, N_B, N_C : are the numbers of linguistic values for variables X, Y, Z respectively; and (Z) is the control signal of the machine.
The A_i, B_j, and C_k are the fuzzy sub sets.
In application, fuzzy variables have multiple membership sets, so several rules are treated at the same time; so we are interested in the inferences to several rules :

$$\begin{cases} R1 : \textbf{If } (X \ is \ A_1) \textbf{ AND } (Y \ is \ B_1) \textbf{ Then } (Z \ is \ C_1). \\ R2 : \textbf{If } (X \ is \ A_2) \textbf{ AND } (Y \ is \ B_2) \textbf{ Then } (Z \ is \ C_2). \\ | \\ | \\ | \\ RN : \textbf{If } (X \ is \ A_N) \textbf{ AND } (Y \ is \ B_N) \textbf{ Then } (Z \ is \ C_N). \end{cases}$$

The FLC rules are phrases realized by : $\mu_{Ri} = \mu(A_i, B_i) \longrightarrow C_i$
As mentioned previously, two types of rule systems can be used :

- **MAMDANI:** In this type, the output signal is fuzzy type.
  Ri: **If** (X1 is A_i) **AND** (X2 is B_j) **Then** (Y is C_k).
  Y : is the output signal or the consequence.
  Example:
  R1: **If** (e is N) **AND** (Δe is N) **Then** (Y is Low).

R2: **If** (e is N) **AND** (Δe is Z) **Then** (Y is Low).
R3: **If** (e is N) **AND** (Δe is P) **Then** (Y is Medium).
R4: **If** (e is Z) **AND** (Δe is N) **Then** (Y is Low).
R5: **If** (e is Z) **AND** (Δe is Z) **Then** (Y is Medium).
R6: **If** (e is Z) **AND** (Δe is P) **Then** (Y is High).
R7: **If** (e is P) **AND** (Δe is N) **Then** (Y is Medium).
R8: **If** (e is P) **AND** (Δe is Z) **Then** (Y is High).
R9: **If** (e is P) **AND** (Δe is P) **Then** (Y is High).

- **SUGENO:** In this type, the output signal is numerical type.
  Ri : **If**$(X1 is A_i)$**AND**$(X2 is B_j)$**Then**$(Y is f(A_i, B_j)$.
  $f(A_i, B_j)$: is a linear function s.t : Z=aX+bY
  Example:
  R1: **If** (e is N) **AND** (Δe is N) **Then** (Y = -1).
  R2: **If** (e is N) **AND** (Δe is Z) **Then** (Y = -1).
  R3: **If** (e is N) **AND** (Δe is P) **Then** (Y = 0).
  R4: **If** (e is Z) **AND** (Δe is N) **Then** (Y = -1).
  R5: **If** (e is Z) **AND** (Δe is Z) **Then** (Y = -1).
  R6: **If** (e is Z) **AND** (Δe is P) **Then** (Y = 0).
  R7: **If** (e is P) **AND** (Δe is N) **Then** (Y = 0).
  R8: **If** (e is P) **AND** (Δe is Z) **Then** (Y = 0).
  R9: **If** (e is P) **AND** (Δe is P) **Then** (Y = 1).

b) **Decision Logic :**
It consists of determining the interpretation, in other words, how to interpret a rule and the set of rules for the calculator. In general, a set of rules can be interpreted by **OR** logic, and a single rule can be construed as an **AND** logic.

- **Rules interpretation:** The set of operators which allow the interpretation of rule or set of rules is called **Inference Engine** (MAX-MIN or MAX-PROD) which are advanced methods. These methods are not used in Sugeno because the consequents already provide crisp outputs.

1) **MAMDANI :**

- **One Signle Rule :**
  For this case, the previously discussed notion is used, as follows:
  Ri: **If** (X is A_i) **AND** (Y is B_j) **Then** (Z is C_k).
  This later is interpreted with the method Min:
  $\mu_{Ri} = \mu_{AandB}(x) = (\mu_A \cap \mu_B)(x) = min\{\mu_A(x), \mu_B(x)\}$

- **Set of Rules :**
  It consists of calculation of the outputs μC_k from μR_i

$$\begin{cases} R1 : \textbf{If } (X \ is \ A_1) \textbf{ AND } (Y \ is \ B_1) \textbf{ Then } (Z is C_1). \\ R2 : \textbf{If } (X \ is \ A_2) \textbf{ AND } (Y \ is \ B_2) \textbf{ Then } (Z \ is \ C_2). \\ | \\ | \\ | \\ RN : \textbf{If } (X \ is \ A_N) \textbf{ AND } (Y \ is \ B_N) \textbf{ Then } (Z \ is \ C_N). \end{cases}$$

In order to interpret this set of rules, we use the method Max, so the engine is given by :
$\mu C_1 = \max(\mu R_i) \longrightarrow$ Ri for the output C_1
$\mu C_n = \max(\mu R_i) \longrightarrow$ Ri for the output C_n

2) **SUGENO :**

- **One Single Rule:**
  It is the same as MAMDANI except for the **AND** logic, it is interpreted as product. This later is used for weighting rule consequents. It multiplies the antecedent membership value (how well the input matches the fuzzy terms) with the consequent function, giving stronger rules a greater impact on the final crisp output. The value of output u can be described as singleton [21].
  Example :
  Imagine a Sugeno system controlling a room's temperature
  Input: Temperature (Temp)
  Output: Heating Power (Heat)
  Rule: If Temp is Cold (membership = 0.8) then Heat = 2*Temp + 1
  - Without Prod: If we directly used the membership value (0.8) as the weight, the consequent function wouldn't be scaled. The system might simply use Heat = 2*Temp + 1 regardless of how cold it actually is.
  - With Prod: we get a weighted consequent: (0.8) * (2*Temp + 1). This multiplies the membership value (0.8) by the entire consequent function. If the temperature (Temp) is 15°C (considered cold), the weighted consequent becomes: (0.8) * (2 * 15 + 1) = 24.8.
- **Set of Rules:**
  It is interpreted by :

$$Y = \frac{\sum y_i w_i}{\sum w_i} \tag{3.11}$$

  Where:
  w_i : is the firing strength
  y_i : is the output level of each rule

3.6.3 Defuzzification

In this step, the inverse of the fuzzification operation is performed. It consists of calculating the output physical value from the obtained membership degree. There exist several methods of deffuzification including : center of gravity, mean of maximum, and weighted average methods [22] [23].

- **Center of Gravity Method (COG) :**
  This is the most common method of defuzzification. The abscess of the center of gravity of the affiliation function resulting from the inference corresponds to the output value of the regulator. It appears that the more complicated the resulting affiliation function is, the longer and more costly the defuzzification process becomes in calculation time.

$$Z^* = \frac{\int z\mu_R(z)dz}{\int \mu_R(z)dz} \tag{3.12}$$

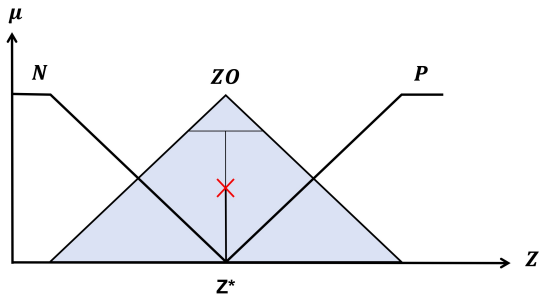


Figure 3.14: Center of Gravity (Area) Method

- **Mean of Maximum Method (MOM) :**
  This method is much simpler. The output value is chosen as the abscess of a maximum value of the affiliation function.

$$Z^* = \frac{\sum z_i \in M(z_i)}{|M|} \tag{3.13}$$

  where : M = { z_i $|\mu_A$ (z_i) is equal to the height of the fuzzy set A} and |M| is the cardinality of the set M.

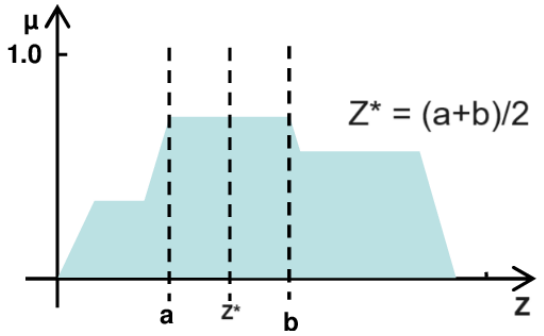


Figure 3.15: Mean of Maximum Method (MOM)

- **Weighted Average Method :**
  It corresponds to the center of gravity method when the membership functions do not overlap. This method is mainly used when the output variable's affiliation functions are singletons.

$$Z^* = \frac{\sum z\mu_{Ri}(z)}{\sum \mu_{Ri}(z)} \tag{3.14}$$

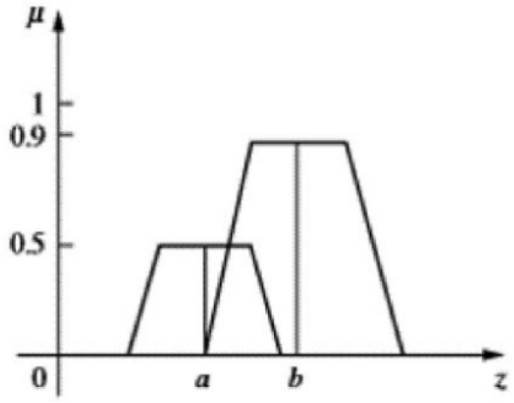


Figure 3.16: Weighted Average Method

3.7 Application of Fuzzy-PI Logic Controller to Vector Control

We will focus mainly on the speed controller that will be replaced by fuzzy and keep the current controllers with PI. This later has been chosen to overcome the Proportional and Integral limitations which are [24] [25] :

- **P Limitations :** Can't eliminate steady-state error, If the error remains constant, the controller output also remains constant, leading to a persistent difference between desired and actual value.
- **I Limitations :** Slow response: The controller output relies on accumulating past errors, so it can be slow to react to initial changes in the error.

Overall, a fuzzy PI controller offers a balance between fast response and good steady-state performance, while also being adaptable to non-linear systems.

Based on the fundamentals of fuzzy logic, the fuzzy speed controller is depicted in (figure 3.17). It can be noticed two linguistic input variables which are speed error (e) ,change of speed error (ce) or (Δe) and one output linguistic variable represented by the reference Torque (Te*), noted as follows [20] :

$$e(n) = w_r(n) - w_r(n-1) \tag{3.15}$$

$$ce(n) = e(n) - e(n-1) \tag{3.16}$$

$$T_e^* = \int \Delta T_e^*(n) = f(e(n), ce(n)) \tag{3.17}$$

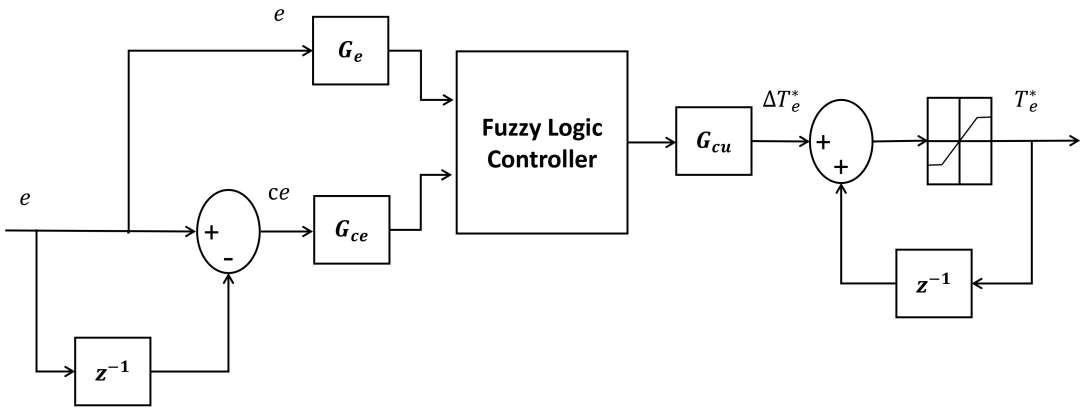


Figure 3.17: Fuzzy Speed Controller

3.7.1 Scaling Factors

For compatibility within the Fuzzy Logic Controller (FLC), the inputs (speed error, and change of speed error) and the output (reference torque) were scaled using three distinct factors (G_e, G_{ce}, and G_{cu}) to normalize them within range of [-1 ; 1] for efficient processing [3]. The coefficients were chosen manually by trial and error method.

3.7.2 Membership Functions

All membership functions (Mfs) for controller inputs (i.e., e and ce) and incremental change in controller output (i.e., ΔT_e^*) are defined on the common normalized domain [-1 ; 1] [1]. The membership functions are shown in Figures (3.18,19,20). The MFs for the inputs are defined on the normalized range [-1,1] as mentioned previously. Five triangular membership functions are used to denote the inputs and five singleton Mfs are employed for the output of the fuzzy controller ΔT_e^*. The Mfs are designed to be symmetrical and identical in terms of width and peak position.

Due to the use of sugeno type fuzzy controller, the singleton Mfs are used in the output. As stated before in the previous section, singletons in fuzzy control outputs bring efficiency and compatibility with linear methods, which represents a winning combination for real-world control problems.

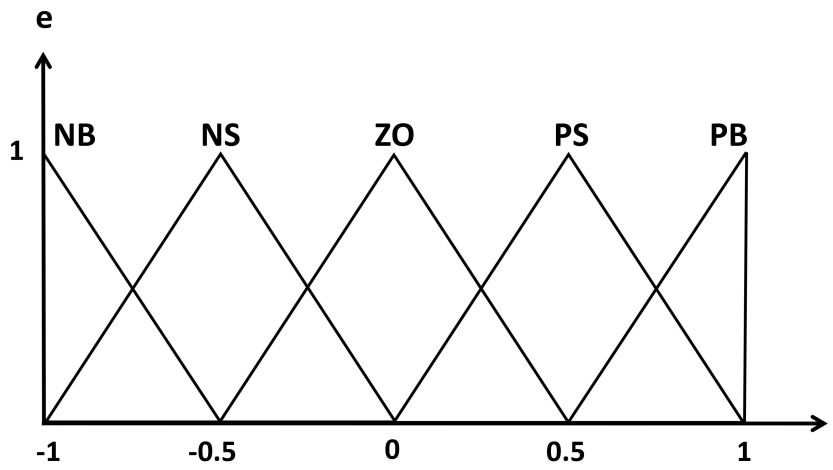


Figure 3.18: MFs of Error (e)

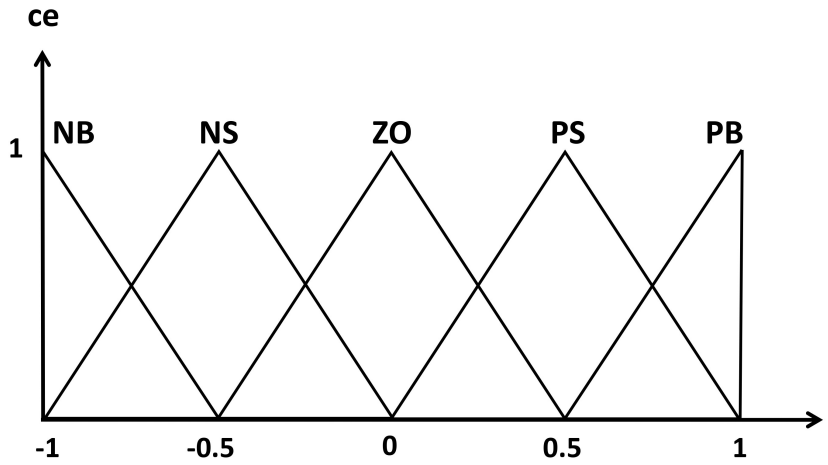


Figure 3.19: MFs of Change in Error (ce)

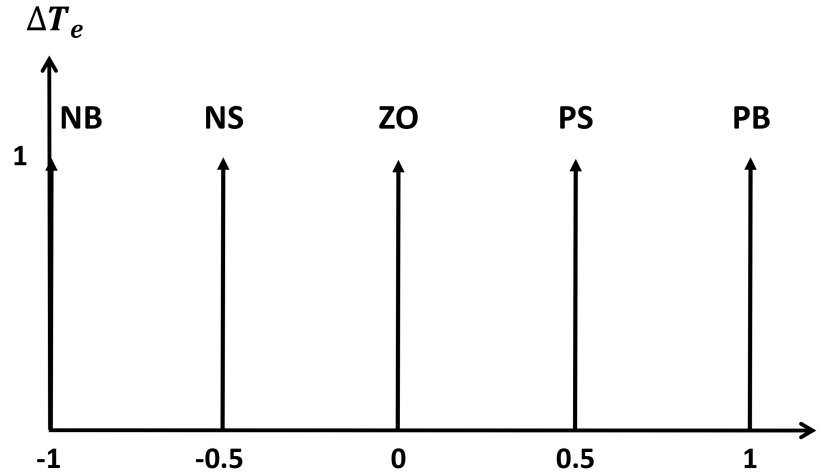


Figure 3.20: MFs of Output (ΔT_e)

3.7.3 Rule Base

Designing fuzzy control rules is crucial, but often relies on subjective expert knowledge. The Phase-Plane Trajectory method offers a systematic solution. This method relates the system's dynamic behavior (Fig 3.21) to the fuzzy rules. It ensures good performance across different operating speeds. The method analyzes a 5x5 membership function matrix to define rules. By dividing the step response into regions (Fig 3.21), we can track the change in error (e) and change of error (ce) (positive, negative, zero) across each region. This information is then mapped to a phase plane trajectory (Fig 3.22). This trajectory

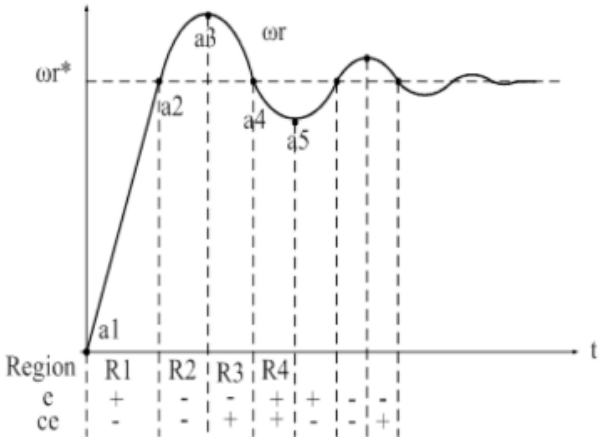


Figure 3.21: Dynamic Behaviour of a step Speed response

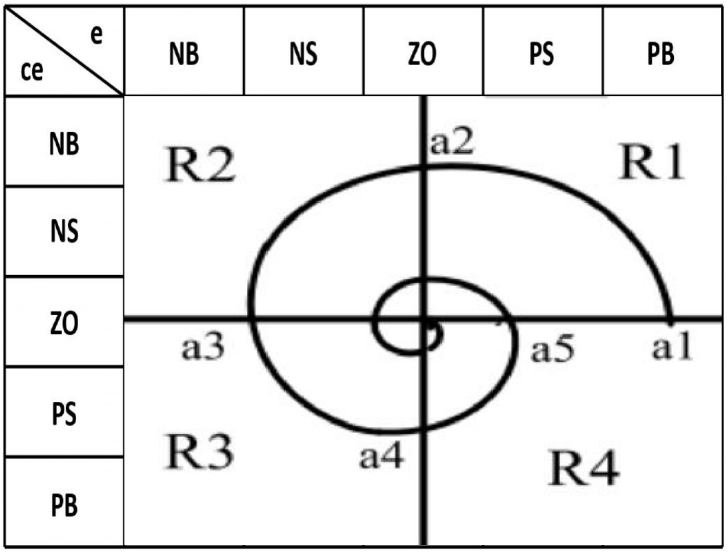


Figure 3.22: Phase Plane Trajectory Mapping

guides the control action (cu) based on pre-defined meta-rules. As the system reaches stability (e and ce become zero), the trajectory moves towards the center of the phase plane. Following this approach, 25 fuzzy rules are generated from the 5x5 membership function matrix, as shown in the standard fuzzy rule mapping frame (Fig 3.23) [26].

ce \ e	NB	NS	ZO	PS	PB
NB	1 NB R2	6 NB	11 NB a2	16 NS R1	21 ZO
NS	2 NB (-e , -ce)	7 NB	12 NS	17 ZO (+e , -ce)	22 PS
ZO	3 NB a3	8 NS	13 ZO	18 PS a1, a5	23 PB
PS	4 NS R3	9 ZO	14 PS a4	19 PB R4	24 PB
PB	5 ZO (-e , +ce)	10 PS	15 PB	20 PB (+e , +ce)	25 PB

Figure 3.23: Rule Based Mapping Frame

3.7.4 Defuzzification

Fuzzy controllers typically require defuzzification to convert their output into a usable value. This work does not need defuzzification because the sugeno consequents are mathematical functions (often linear) of the inputs, which provide directly crisp output values [19].

3.8 Conclusion

In this chapter, we have delved into the core principles of fuzzy logic and its application in the IRFOC of Induction motors. We examined the design of the fuzzy regulator, highlighting its advantages in achieving superior control performance. However, despite its impressive results, the fuzzy regulator holds potential for further refinement. To unlock this potential, the next chapter will explore the optimization techniques of the fuzzy PI controller using advanced algorithms.

Chapter 4

Optimization Techniques

4.1 Introduction

This chapter describes the mechanisms occurring in two representative nature-inspired optimization algorithms: particle swarm optimization (PSO) and gray wolf optimizer (GWO). These algorithms are inserted in two steps of the design approach dedicated to the optimal tuning of simple Takagi-Sugeno proportional-integral fuzzy controllers involved in the speed control of the induction motor.

4.2 Metaheuristic Algorithms

The simulation of biological behaviors in nature serves as an inspiration for metaheuristic algorithms, and the behavior of individuals within a population bears many similarities to the ecological behavior of biological groups in nature. The evolution of species and even the whole ecosystem is inseparable from the synergy between populations, which has the advantages of simplicity, parallelism, and high applicability, etc. Metaheuristic algorithms are very useful for tackling a variety of common optimization problems with high randomness, big scale, multi-objectivity, and multi-constraint features. The issue does not necessarily need to be continuously differentiable. Their distributed, self-organizing, cooperative, reliable, and straightforward implementation are what primarily distinguish them.[27]

4.3 Particle Swarm Optimization

Particle swarm optimization (PSO) was originally designed and introduced by Eberhart and Kennedy [28]. It is a population based search algorithm based on the simulation of the social behavior of birds, bees or a swarm of fishes, hence the term swarm used in the name. This algorithm originally intends to graphically simulate the unpredictable flying pattern of a bird flock. Each individual within the swarm is represented by a vector in multidimensional search space. This vector has also one assigned vector which determines the next movement of the particle and is called the velocity vector. The PSO algorithm also determines how to update the velocity of a particle. Each particle updates its velocity based on current velocity and the best position it has explored so far; and also based on the global best position explored by Swarm. Then it is iterated a fixed number of times or until a minimum error based on the desired performance index is achieved. [28]

4.3.1 Initialization

Initialization of PSO algorithm consists of initially randomly placing the particles according to a uniform distribution in the search space. This stage is common between virtually all the algorithms of stochastic iterative optimization. But here particles have velocities as well. By definition, a velocity is a vector applied to a position, thus giving another position. In practice, it is not desirable that too many particles tend to leave the search space as early as the first increment, or for that matter later. We will see below what

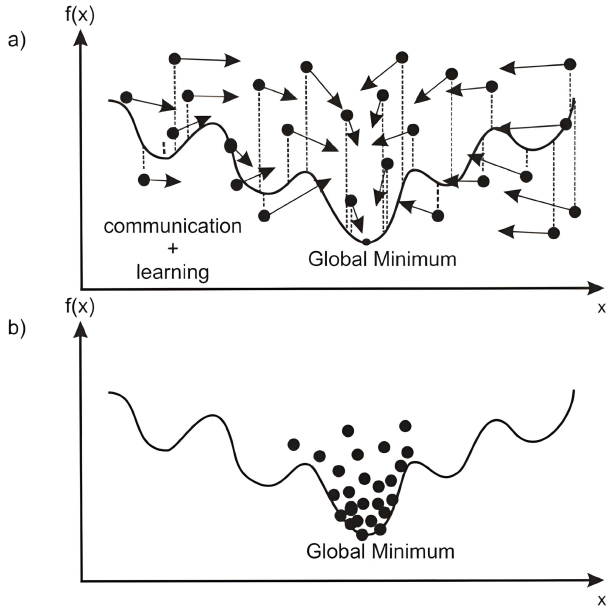


Figure 4.1: particle swarm representation

occurs in this case, but, for the moment, let us be satisfied with deriving at random the values of the components of each velocity, according to a uniform distribution in:[29]

$$Dist = [\frac{(x_{min} - x_{max})}{2}, \frac{(x_{max} - x_{min})}{2}] \tag{4.1}$$

The dimension of the search space is D. Thus, the current position of a particle in this space at the moment t is given by the vector $x(t)$ that moves with a speed defined by the vector $u(t)$. The best position found up to time t is given by a vector $p(t)$ and the best position found by informants of the particle is indicated by a vector $g(t)$

Thus, the equations of motion of a particle indicated by the index d are:

$$u_d = \theta u_d + c_1 r_1 (p_d - x_{i-1}) + c_2 r_2 (g_d - x_{i-1}) \tag{4.2}$$

$$x_i = x_{i-1} + u_i \tag{4.3}$$

4.3.2 PSO Algorithm

The pseudo-code of the PSO algorithm is presented

Algorithm 1 Particle Swarm Optimization (PSO)

Input : N,x_l,x_u,c_1,c_2,i_{max},f
Output : A swarm S of size N (N position vectors)
Initialize S, randomly generate the position $\mathbf{x}$ of each particle w.r.t the bounds x_l,x_u of the objective function;
Initialize all velocities $\mathbf{u}$ to zero;
Initialize best positions P_d (and respective values) for individual particles and find $g_d{}^*$;
Choose randomly two values in [0,1] for r_1 and r_2;
Iteration $i = 0$;
Initialize θ_{min},θ_{max};
Calculate fitness
While $i < i_{max}$
Calculate inertia: $\theta = \theta_{max} - \frac{\theta_{max} - \theta_{min}}{i_{max}} i$;
For each particle in S, the values for iteration i are :

1. Update velocity : $\mathbf{u}_i = \theta + \mathbf{u}_{i-1} + c_1 r_1[\mathbf{P}_d - x_{i-1}] + c_2 r_2[\mathbf{g}_d - x_{i-1}]$;
2. Update position $x_i = x_{i-1} + \mathbf{u}_i$;
3. Check fitness
4. Compute the value of the new position according to f;
5. Check/Update: $\mathbf{x}^*, \mathbf{g}^*$

Check for convergence;
Upgrade iteration $i = i + 1$; While
return S;

4.4 Grey Wolf Optimizer Algorithm

GWO is inspired by the behavior, Social structure, and ingenious hunting methods of grey wolves. Grey wolves are on top of the food chain in their respective hunting grounds, and they live in packs containing from 5 to 12 members. Mainly, grey wolf communal life is controlled by a strict social hierarchy. The leaders of the pack (*alphas*) are male and female wolves that are responsible for making decisions for their pack, such as their living quarters, hunting grounds, and even wake-up time. It's observed that they have some democratic behaviors, but most importantly, they must obey the *alphas*. In addition to the particular social structure, group hunting represents an interesting social behavior.[30]

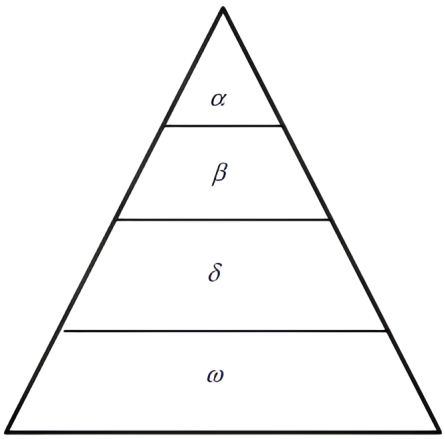


Figure 4.2: Grey Wolf hierarchy

4.4.1 Mathematical model

To mathematically model the social behavior of grey wolves in the algorithm, the best solution is considered *alpha*(α). Thus the second and third best solutions are considered *beta* (β) and *delta* (δ) respectively.[30]

a) Encercling the Prey

Grey wolves in the process of hunting encircle the prey.This behavior can be expressed by the following equations:

$$\vec{D} = |\vec{C}\vec{X}_p(t) - \vec{X}(t)| \tag{4.4}$$

$$\vec{X}(t+1) = \vec{X} - \vec{A}.\vec{D} \tag{4.5}$$

where:
t : iteration number
$\vec{A}$ and $\vec{C}$: are coefficient vectors
$\vec{X}_p$: vector of the prey's position
$\vec{X}$: Vector of Grey wolf's position
$\vec{D}$: Calculated vector used to specify a new position of the grey wolf
Vectors $\vec{A}$ and $\vec{C}$ are defined by the following equations:

$$\vec{A} = 2\vec{a}.\vec{r_1} - \vec{a} \tag{4.6}$$

$$\vec{C} = 2.\vec{r_2} \tag{4.7}$$

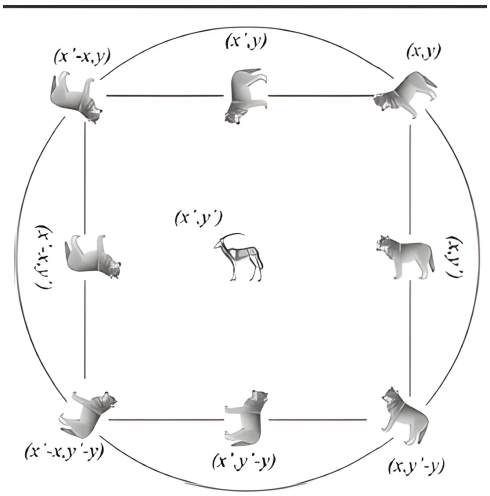


Figure 4.3: Attacking toward prey versus searching for prey

b) Attacking the Prey

Grey wolves bring their hunting to an end once their prey ceases all motion. This is modeled by decreasing the value of $\vec{a}$ mentioned in equation (4.6). When $|A| < 1$ grey wolves will attack the prey [30].

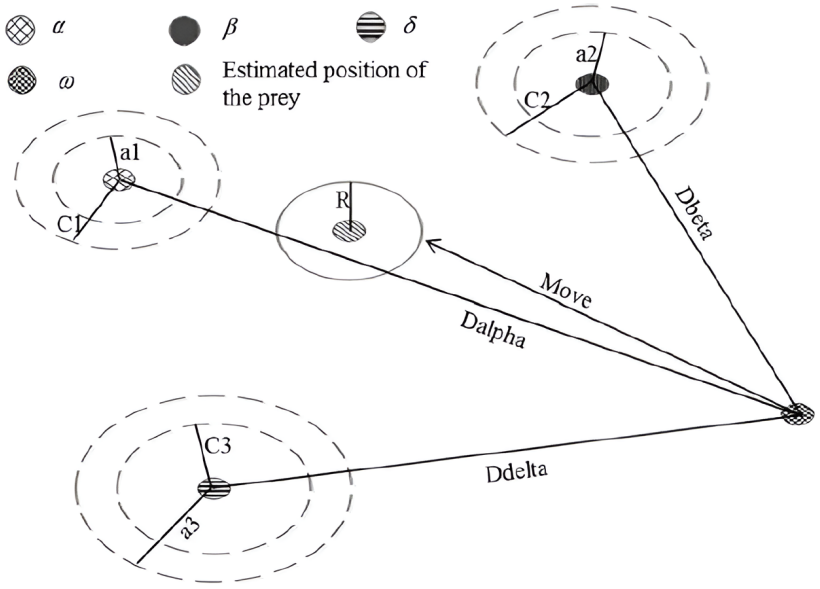


Figure 4.4: Updating the position

4.4.2 GWO Algorithm

Algorithm 2 Grey Wolf Optimization(GWO)

Input : Population size (pop), Max iteration (t_{max})
Output : Optimal grey wolf position X_α
Initialize the grey wolf population randomly
Initialize $\vec{a}$,$\vec{A}$,and $\vec{C}$;
Determine the fitness of each wolf X_i
X_α = the best solution
X_β = the second best solution
X_δ = the third best solution
while $i <= t_{max}$ **do**
    For each wolf X_i
update the position using equation (4.5)
Update $\vec{a}$,$\vec{A}$,and $\vec{C}$
Determine the fitness of each wolf X_i
Update X_α,X_β,and X_δ
$i = i + 1$

return X_α;

4.5 Fitness Function

As the name suggests, fitness functions are a measurement of how close is the output of the optimized system is to the desired output (the controller input). In our application, we utilized two parameters to construct the fitness function. We have used the difference between the actual speed of the system and the desired output as the error which was used to calculate IAE and ITAE . We have associated an appropriate weight with each variable in order to get an adequate cost function. Our optimization process is defined by two main constraints. The first one is the condition that ensures the crossing between membership functions of the fuzzy logic controller to not leave any uncovered combination of error and change of error. While the actual calculation of fitness was carried out using the following equation:

$$fitness = w_1 * max(IAE) + w_2 * max(ITAE) \tag{4.8}$$

where: w_1 and w_2: are weights
IAE Index : Integral of Absolute Error

$$IAE = \int |e(t)|\, dt \tag{4.9}$$

$ITAE$ Index : Integral Time Absolute Error

$$ITAE = \int t|e(t)|\, dt \tag{4.10}$$

As mentioned before, the adequacy of the system is determined by the value of the fitness function. The lower value (or closer to 0) of the fitness function the better.

4.6 Conclusion

In this chapter, we have been introduced to the Meta-heuristic algorithms used (PSO and GWO) and how they function. Also, we have defined the fitness function and how the measure of goodness of a solution is calculated through it. The next chapter will present the results of the different schemes applied and comparing them in order to find the most optimal control scheme (Controller to be used).

Chapter 5

Simulation and Results

5.1 Introduction

All simulations of the discussed concepts from earlier chapters are combined in this one. Here, we take a closer look at each case, evaluate the outcomes, and have lengthy discussions. We want to have a thorough grasp of the control of induction motor by combining these simulations. The used Motor's parameters can be found in Appendix A.

5.2 Vector Control of Induction Motor

In this part, we have simulated the Indirect Field Oriented Control (IRFOC) of IM shown on figure 5.1 based on the dynamic model of the induction motor in $\alpha\beta$ reference frame. The mathematical model of this later has been presented in chapter 1 and the control method in chapter 2. The Space Vector Pulse Width Modulation is introduced as a technique for generating PWM Signals for the inverter that feeds the induction motor. The simulation is done with a sampling time of $5 * 10^{-5}$s which corresponds to 20Khz. It is started by using 3 PIs controller for three control loops for speed, currents I_{ds} and I_{qs}. Then adding a reference filter to the speed PI controller, and we finished by replacing the PI controller by Fuzzy-PI controller.

- **Simulation Conditions :**
  Before presenting the simulation results, it is good to highlight some important points concerning the conditions for which the simulation was conducted in :

  1) Simulation Time = 5s
  2) The reference flux ψ_r^* was taken 1 Wb.
  3) The reference speed started with :

     - **at t=0s** $\longrightarrow$ w_r = 500 RPM with no load (T_L= 0 N.m)
     - **at t=1s** $\longrightarrow$ w_r = 500 RPM and T_L=7.78 N.m
     - **at t=2s** $\longrightarrow$ w_r = 1000 RPM
     - **at t=3s** $\longrightarrow$ w_r = -500 RPM (500 RPM in opposite direction)
     - **at t=4s** $\longrightarrow$ w_r = 800 RPM

Simulink Model

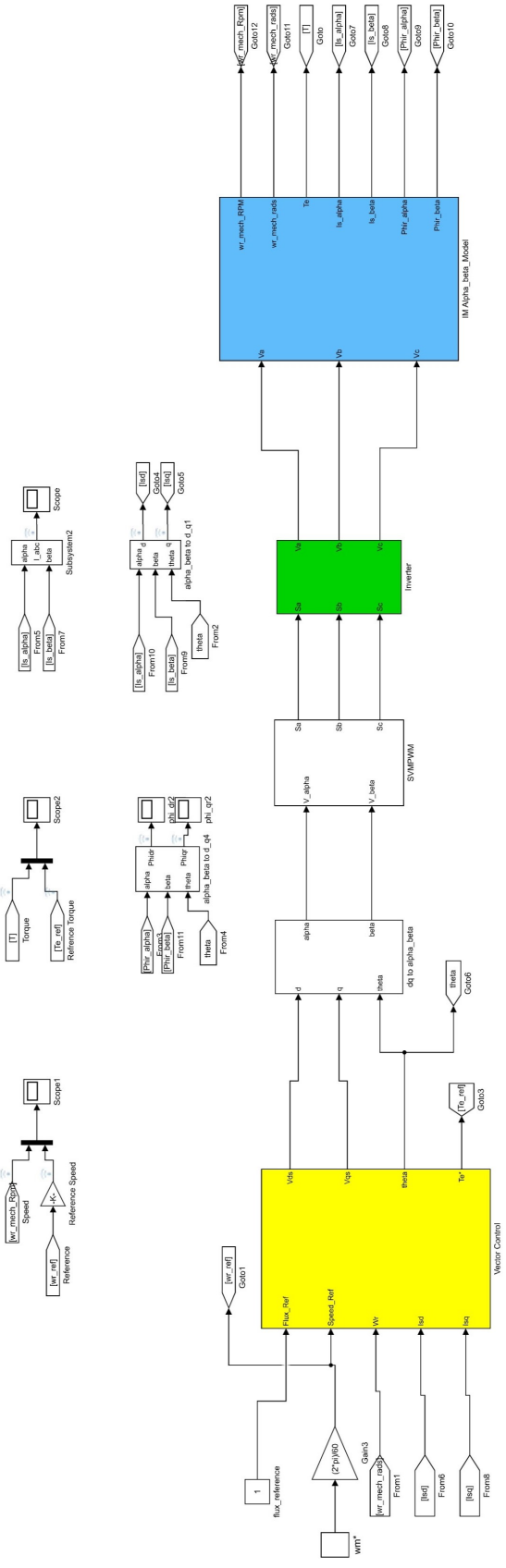


Figure 5.1: Indirect Vector Control

5.2.1 Conventional PI

As mentioned previously, We started by using the conventional PI with manual tuning, the analytical approach for tuning PI controller is shown in (Appendix B) but they don't perform well due to the non-linearity of the system, so we make those parameters as starting point for tuning with trial and error, the following (table 5.1) demonstrates the obtained gains Kp , Ki for the speed and stator currents controllers.

Table 5.1: Tuning Parameters

Controller	K_p	K_i
Speed	1.1	10
Currents	100	10000

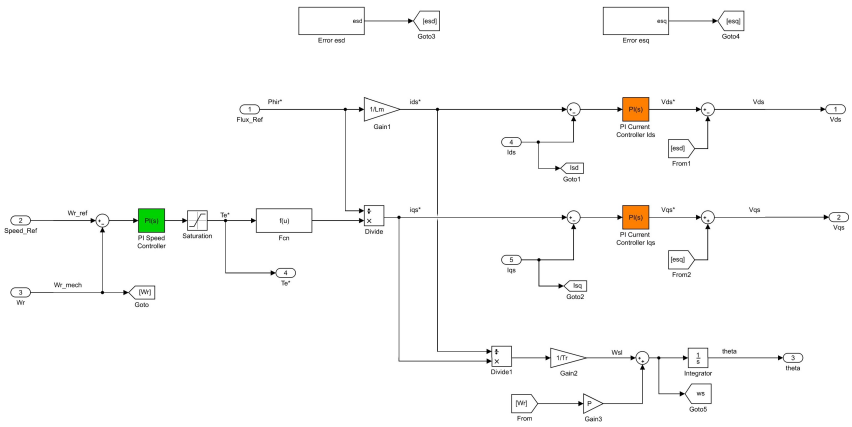


Figure 5.2: IRFOC with PI

- **Results :**
  According to the simulation of figure 5.2, the following results of speed, Torque, flux and currents are presented in figure 5.3 and subfigures (5.4a and 5.4b) and subfigures (5.5a, 5.5b, 5.5c, and 5.5d)

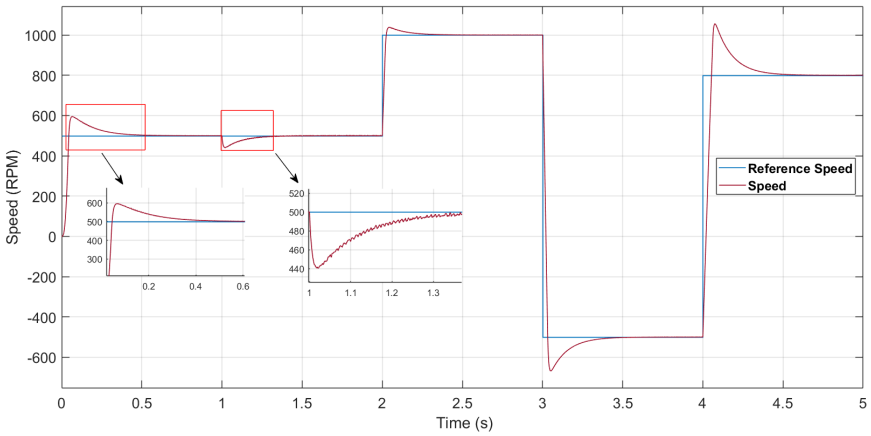
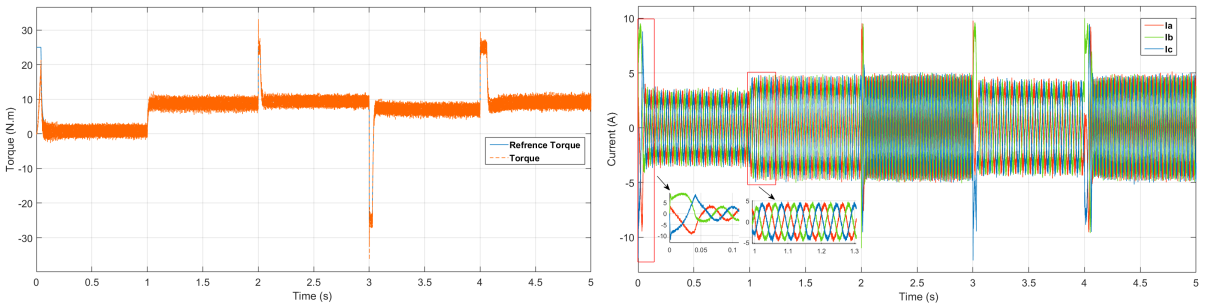
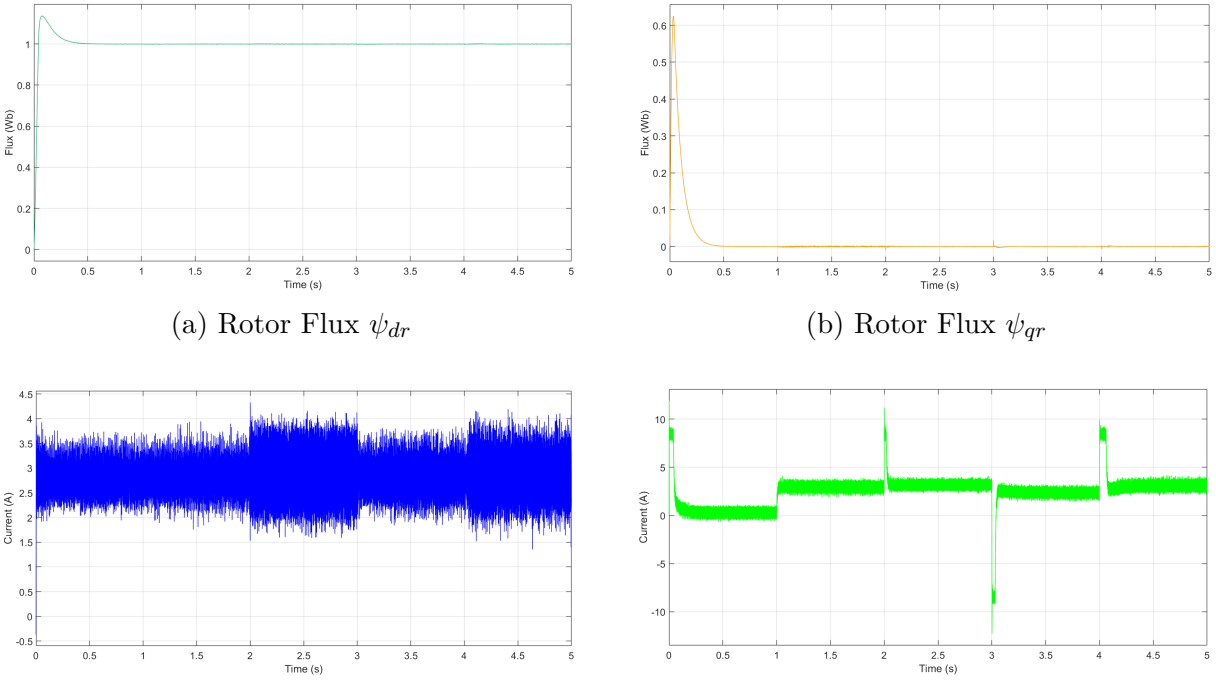


Figure 5.3: Speed (w_r)



(a) Torque (T_e) (b) Current (I_{abc})

Figure 5.4: Speed, Torque, and Current Curves using Conventional PI Controller



(a) Rotor Flux ψ_{dr} (b) Rotor Flux ψ_{qr}

(c) Stator Current I_{ds} (d) Stator Current I_{qs}

Figure 5.5: Rotor Fluxes and Stator Currents Curves using Conventional PI Controller

5.2.2 Conventional PI with Reference Filter

For this part, building upon the previous PI gains parameters we add a reference filter for which its transfer function is :

$$TF(s) = \frac{1}{(\frac{K_p}{K_i})s + 1} = \frac{1}{0.11s + 1} \tag{5.1}$$

and the following figure 5.7 shows the added reference block.

- **Results :**
  According to the simulation of figure 5.6, the following results of speed, Torque, flux and currents are presented in figure 5.7 and subfigures (5.8a and 5.8b) and subfigures (5.9a, 5.9b, 5.9c, and 5.9d)

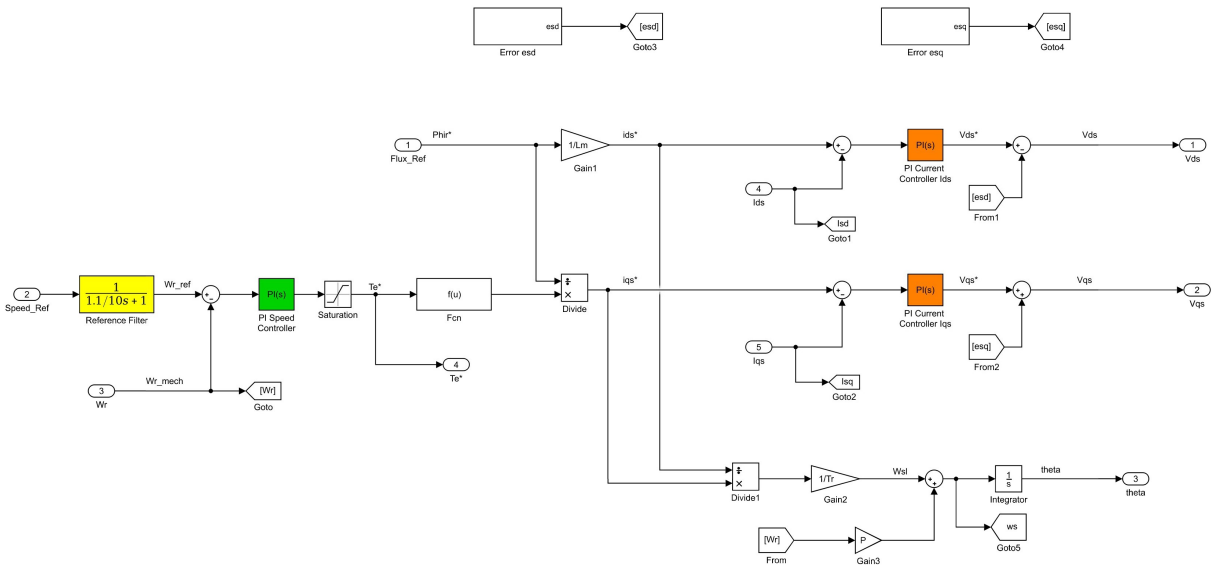


Figure 5.6: IRFOC with PI and Reference Filter

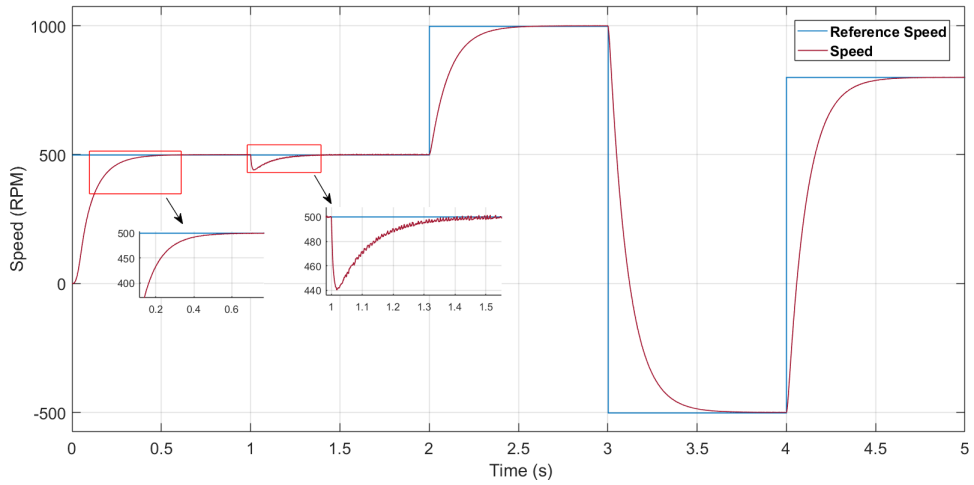
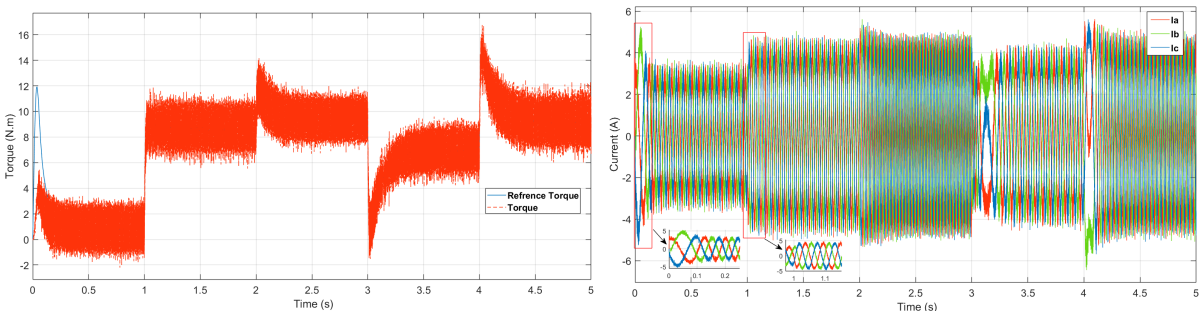
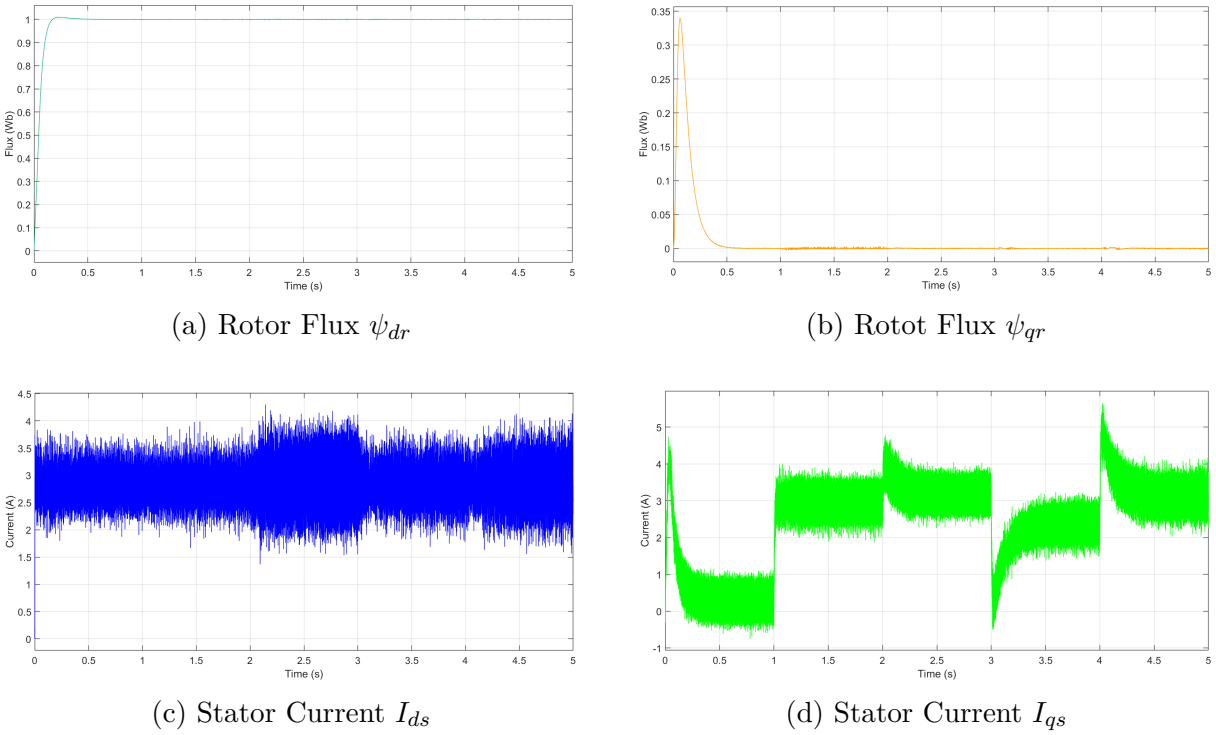


Figure 5.7: Speed (w_r) curve using Conventional PI with Reference Filter



(a) Torque (T_e) (b) Current (I_{abc})

Figure 5.8: Torque, and Current Curves using Conventional PI with Reference Filter



(a) Rotor Flux ψ_{dr}

(b) Rotot Flux ψ_{qr}

(c) Stator Current I_{ds}

(d) Stator Current I_{qs}

Figure 5.9: Rotor Fluxes and Stator Currents Curves using Conventional PI with Reference Filter

5.2.3 Using Fuzzy-PI with Manual tuning

In this part, we keep the current controllers and replace the speed controller with Fuzzy-PI Controller tuned manually as shown on the figures (5.10) and (5.11), the following table 5.2 demonstrates the obtained gains G_e ,G_{ce}, G_{cu} for the speed controller, the figure 5.12 shows an illustration of the chosen Membership functions.

Table 5.2: Tuning Parameters

Gains	Value
G_e	$\frac{1}{200}$
G_{ce}	$\frac{1}{3000}$
G_{cu}	5000

The used Membership functions (Mfs) for inputs error (e) and change of error (ce) in the fuzzy Sugeno code are Triagular and their ranges as follows :

```
'NB', [-1 -1 -0.5], ... %Trimf
'NS', [-1 -0.5 0], ... %Trimf
'ZO', [-0.5 0 0.5], ... %Trimf
'PS', [0 0.5 1], ... %Trimf
'PB', [0.5 1 1]; %Trimf
```

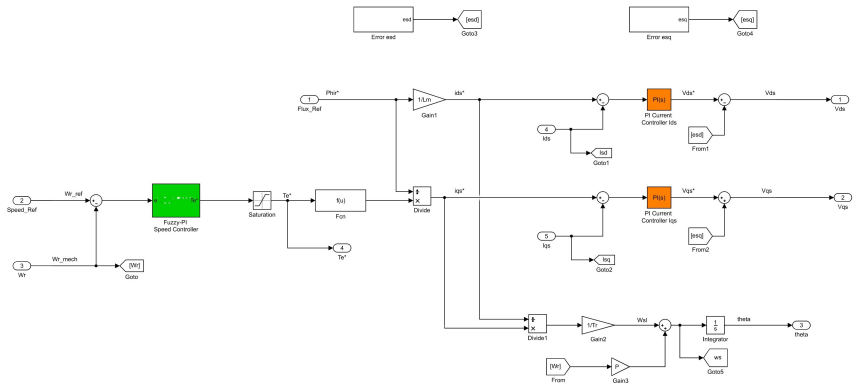


Figure 5.10: IRFOC with FLC

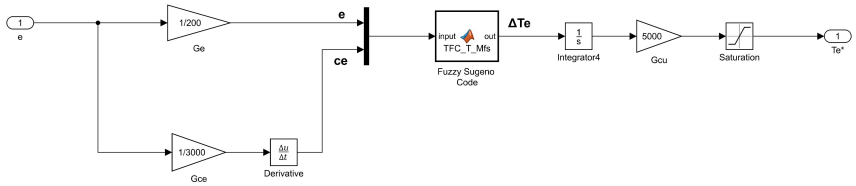
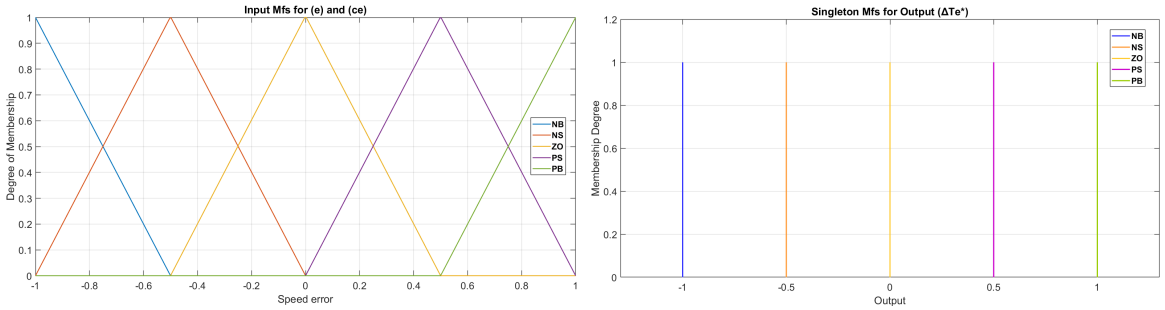


Figure 5.11: Fuzzy Logic Controller Simulink Block

and the Mfs for the output (ΔT_e^*) are Singleton presented in this manner :

```
'NB', -1, ... %Singleton
'NS', -0.5, ... %Singleton
'ZO', 0, ... %Singleton
'PS', 0.5, ... %Singleton
'PB', 1; %Singleton
```

The corresponding Graphs for the previous Mfs are demonstrated on the sub-figures (a), (b) of the (figure 5.14)

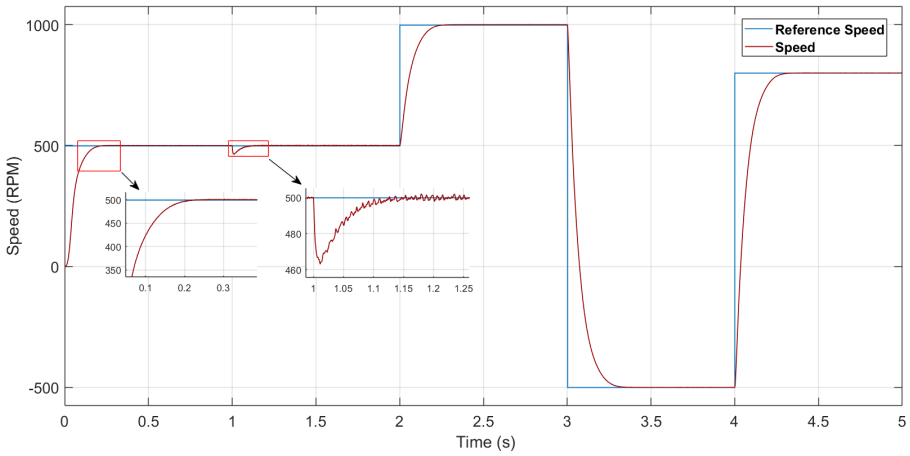


(a) Input Mfs for (e) and (ce)

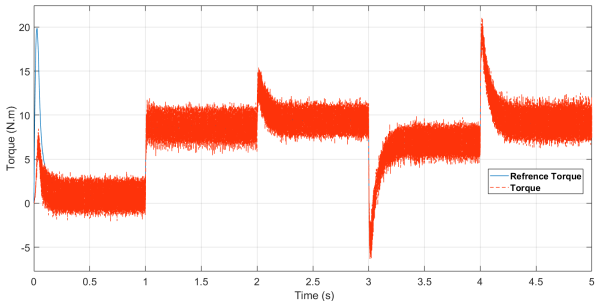
(b) Output Mfs of (ΔT_e^*)

Figure 5.12: Demonstration for Input and Output Mfs

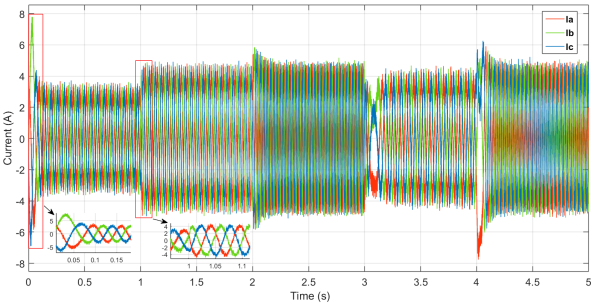
- **Results :**
  According to the simulation of figure 5.10, the following results of speed, Torque, flux and currents are presented in subfigures (5.13a,5.13b,5.13c) and subfigures (5.14a,5.14b,5.14c,5.14d).



(a) Speed (w_r)

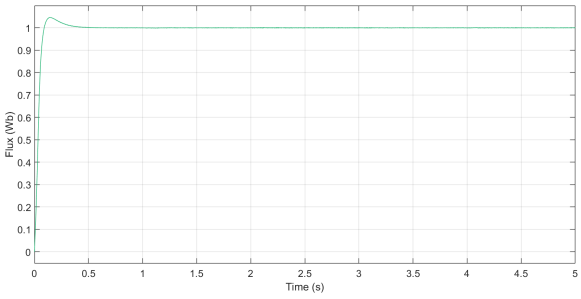


(b) Torque (T_e)

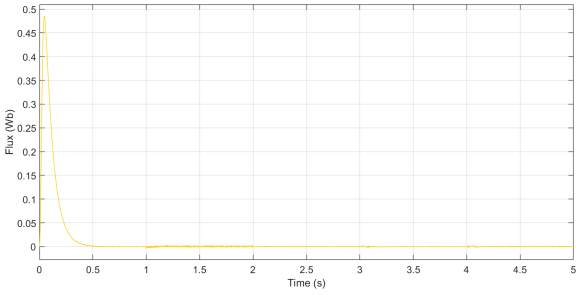


(c) Current (I_{abc})

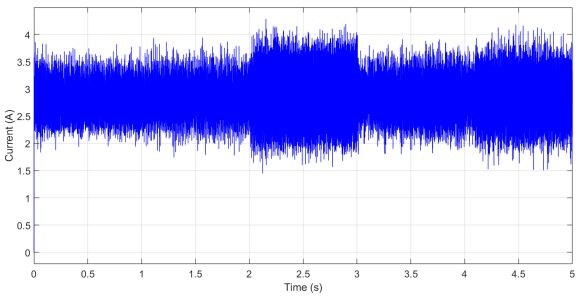
Figure 5.13: Speed, Torque, and Current Curves using Fuzzy-PI Controller



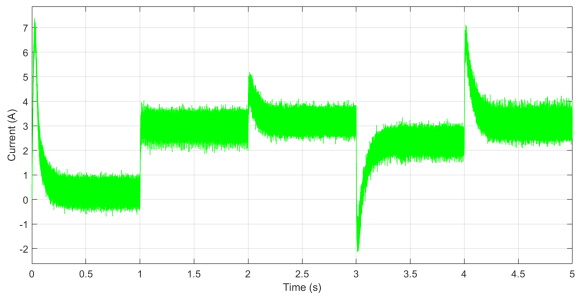
(a) Rotor Flux ψ_{dr}



(b) Rotot Flux ψ_{qr}



(c) Stator Current I_{ds}



(d) Stator Current I_{qs}

Figure 5.14: Rotor Fluxes and Stator Currents Curves using Fuzzy-PI Controller

5.2.4 Comparison

Since we are more interested in the speed controller, we summarized all the characteristics-values from previous simulation graphs in one table which compares the rise time, settling time, overshoot ,drop[1], and the recovery time[2] when applying TL=7.78 N.m.

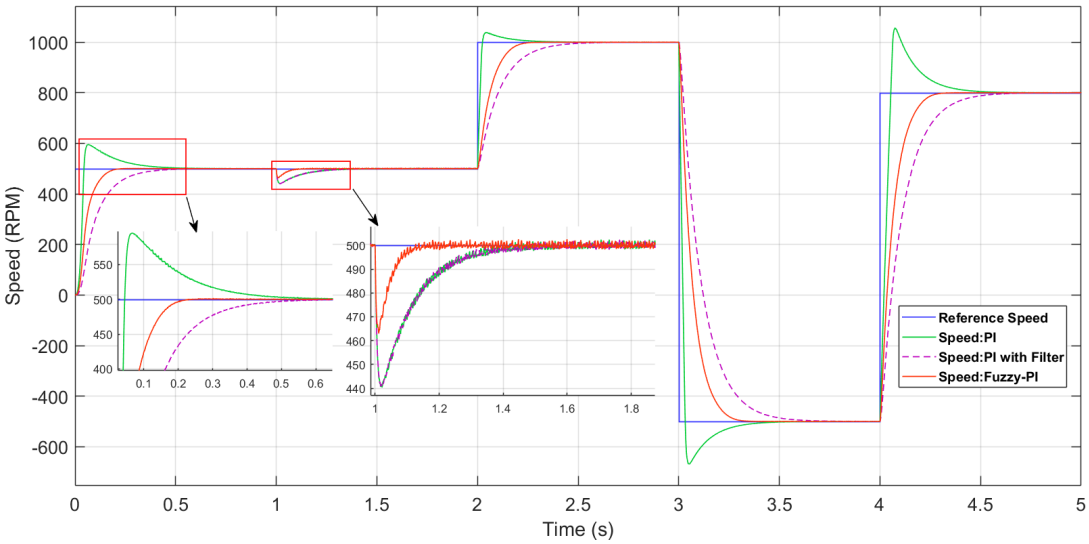


Figure 5.15: Comparison of Speed (w_r) curve between PI, PI with Filter, Fuzzy-PI

Table 5.3: Comparison between the different controllers

Controller/Charachteristics	%OS	T_r	T_s	Drop	$T_{Recovery}$
PI	19.14	0.0425	0.602	59	0.4
PI + Reference Fliter	0	0.2254	0.61	59.7	0.5
Fuzzy-PI	0	0.1202	0.2441	35.8	0.167

5.2.5 Discussion

Starting by the conventional PI and its respective speed graph depicted in figure 5.3 we can note that various overshoots and undershoots denote the speed curve when the speed changes. To mitigate this problem we added a filter to the input of the PI controller, which resulted in a smoother response with no overshoots or undershoots, but at the expense of both the rise and settling times.
When using the fuzzy PI denoted in figure 5.13 we can clearly note an improvement in the responsiveness of the system and lack of the anomalies observed in the first system(PI on its own). Furthermore, we notice empirically that the Fuzzy PI controller have better transient and steady-state response. Even when applying the load, we notice that the Fuzzy PI controller have a smaller drop and faster recovery.

[1]used to describe the decrease in speed that occurs when a load is applied to a motor
[2]the time it takes for the motor to accelerate back to its original speed after the load change

When investigating the Torque curve depicted in figure 5.4a,we notice that there is a spike when the motor starts which is not common in the Torque graphs for the other two controllers (PI and Fuzzy PI) which are represented in figure 5.8a and figure 5.13b respectively. We can also notice a spike in torque whenever the speed value changes. This behavior has been also noticed in the current graphs which is to be expected.

5.3 Optimization of Fuzzy-PI Controller Using Meta-heuristic Algorithms

5.3.1 Gains Optimization

In this subsection, the optimization of the gains has been performed as shown on the (figure 5.16). The gains x(1), x(2), x(3) corresponds to the gains G_e ,G_{ce}, G_{cu} respectively. The indexes IAE and ITAE are used for this purpose as explained previously in chapter 4. Building on the obtained data from the manually tuned Fuzzy-PI, we have estimated some near ranges for the gains to narrow down the space of research and get closer to better results. The ranges are arranged in (table 5.4) :

Table 5.4: Gains Ranges

Bounds/Gains	$\mathbf{G_e}$	$\mathbf{G_{ce}}$	$\mathbf{G_{cu}}$
lower bound (lb)	$\frac{1}{300}$	$\frac{1}{6000}$	5000
upper bound (ub)	$\frac{1}{100}$	$\frac{1}{3000}$	8000

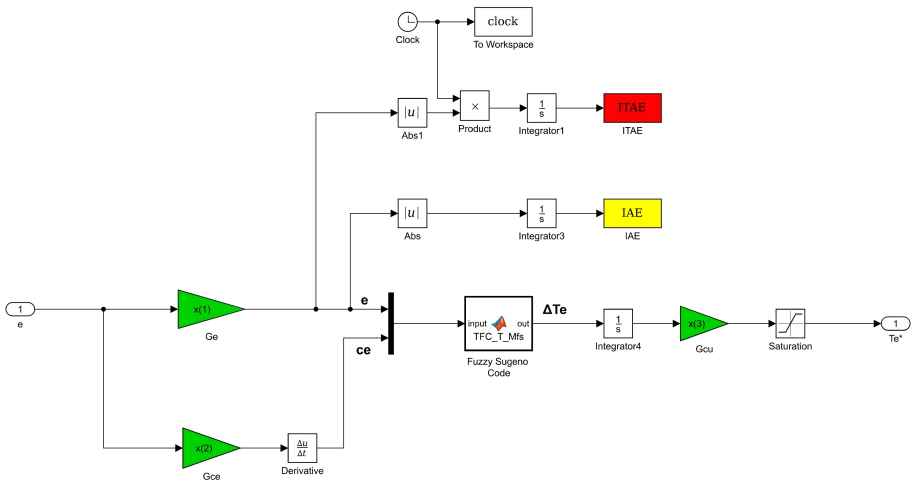


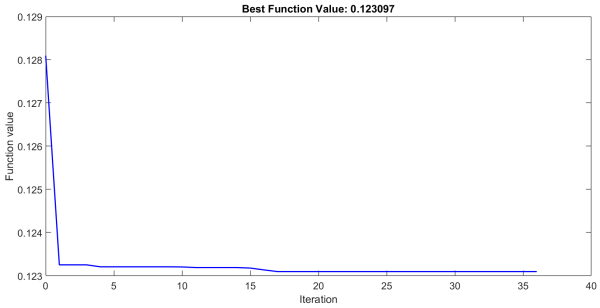
Figure 5.16: Simulink Block for Gains Optimization

a) Using PSO :
We have run the simulation using the following fitness function which was discussed previously in chapter 4.

$$fitness = 0.5 * max(IAE) + 0.5 * max(ITAE) \tag{5.2}$$

We chose 25 particles and 100 iteration in parameters of the simulation code but it stopped before reaching 100 iterations because whenever the fitness function doesn't change for several iterations the program will consider the last fitness function as the best one and stops the simulation.

- **Results :**
  According to the simulation done by the PSO algorithm and the simulink block (figure 5.16) the following gains results (figure 5.17) were obtained and its corresponding curves of speed, Torque, flux and currents are presented in subfigures (5.18a,5.18b,5.18c) and subfigures (5.19a,5.19b,5.19c,5.19d).



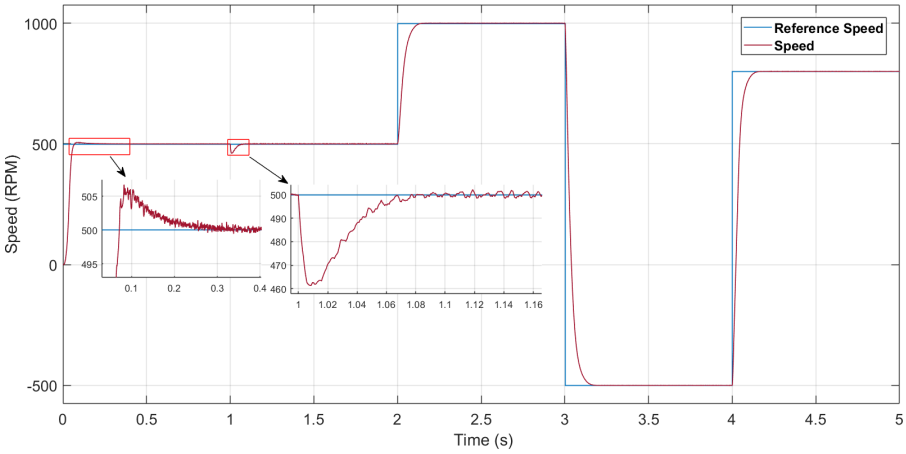

```
----------------------------------------
Optimization Results:
  - Best solution: [0.004693, 0.000167, 7992.706998]
  - Best fitness value: 0.123097
```

(a) Fitness Function Convergence (b) Gains

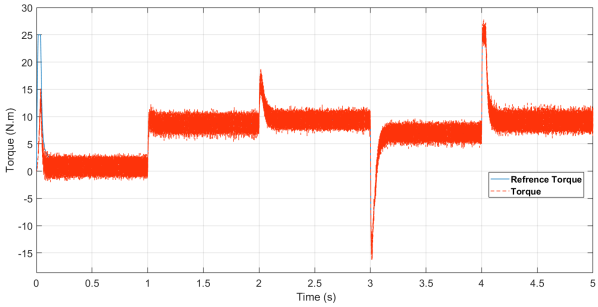
Figure 5.17: Gains Optimization Results

Table 5.5: Obtained Gains Using PSO

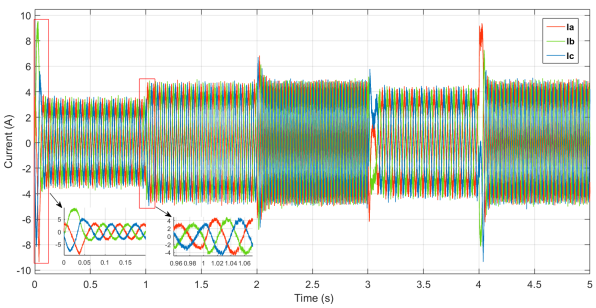
Gains	$\mathbf{G_e}$	$\mathbf{G_{ce}}$	$\mathbf{G_{cu}}$
Values	0.004693	0.000167	7992.706998



(a) Speed (w_r)

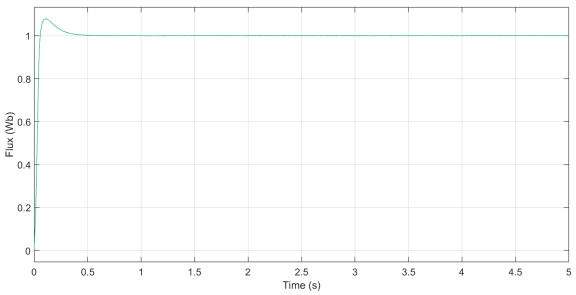


(b) Torque (T_e)

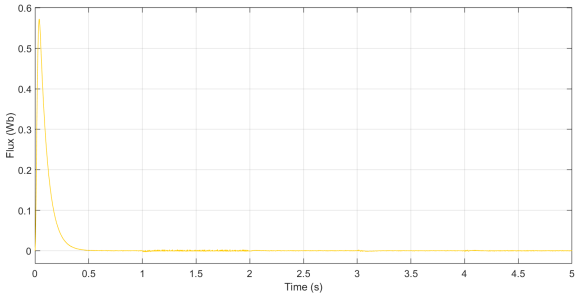


(c) Current (I_{abc})

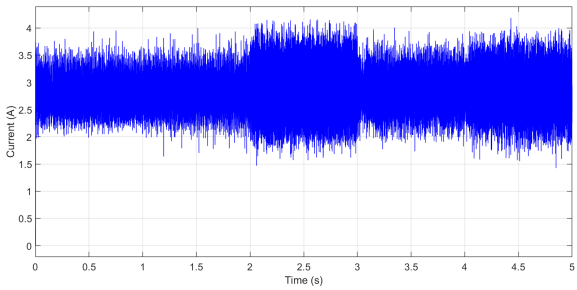
Figure 5.18: Speed, Torque, and Current Curves with Gains Optimization Using PSO



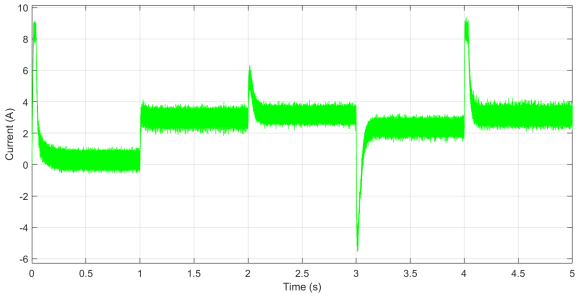
(a) Rotor Flux ψ_{dr}



(b) Rotor Flux ψ_{qr}



(c) Stator Current I_{ds}



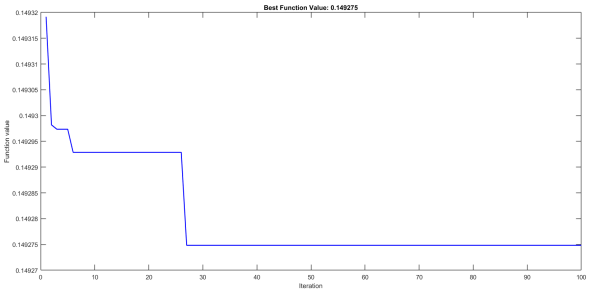
(d) Stator Current I_{qs}

Figure 5.19: Fluxes and Currents Curves with Gains Optimization Using PSO

b) Using GWO :

The procedures are the same like the particle swarm optimization, 100 iterations and the population size as 20, and we rund the GWO algorithm with same fitness function previously mentioned.

- **Results :**
  According to the simulation done by the GWO algorithm the following gains results in (table 5.6) and (figure 5.20) were obtained and its corresponding curves of speed, Torque, flux and currents are presented in subfigures (5.21a,5.21b,5.21c) and subfigures (5.22a,5.22b,5.22c,5.22d).



(a) Fitness Function Convergence

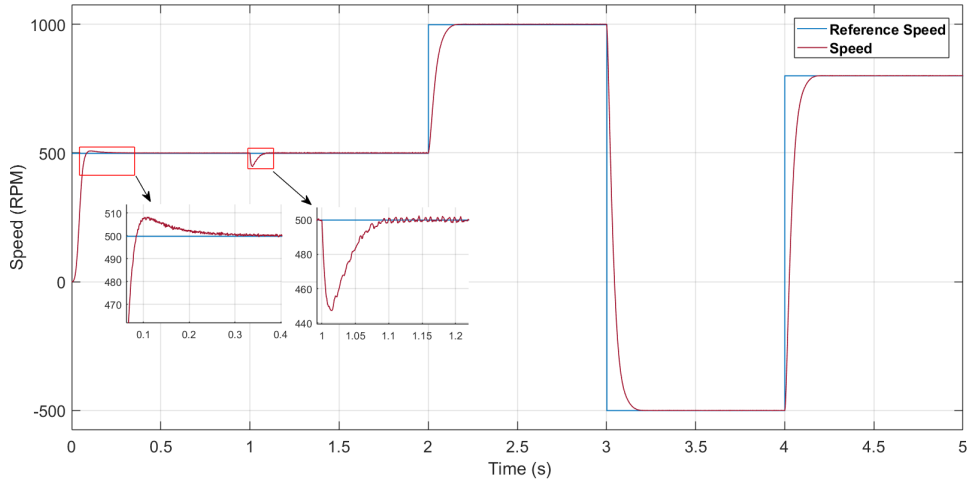
```
----------------------------------------
Optimization Results:
  - Best solution: [0.004000, 0.000167, 5808.269581]
  - Best fitness value: 0.149275
```

(b) Gains

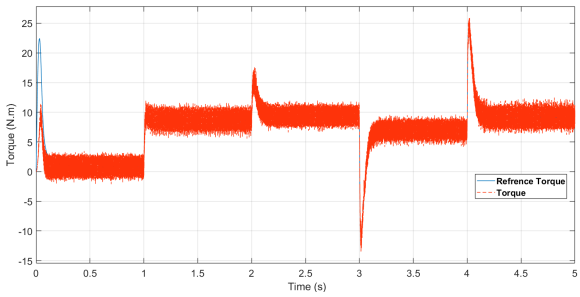
Figure 5.20: Optimization Results

Table 5.6: Obtained Gains Using GWO

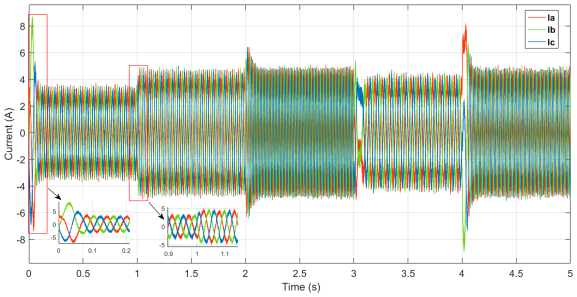
Gains	$\mathbf{G_e}$	$\mathbf{G_{ce}}$	$\mathbf{G_{cu}}$
Values	0.004	0.000167	5808.269581



(a) Speed (w_r)

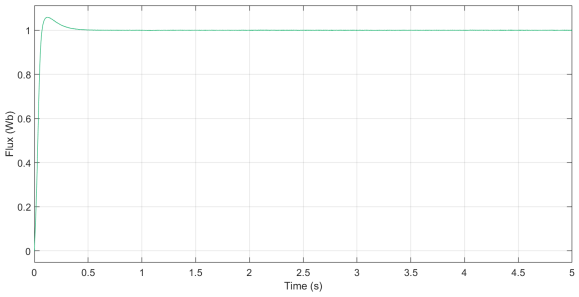


(b) Torque (T_e)

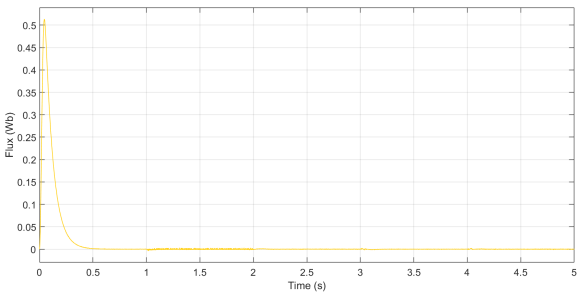


(c) Current (I_{abc})

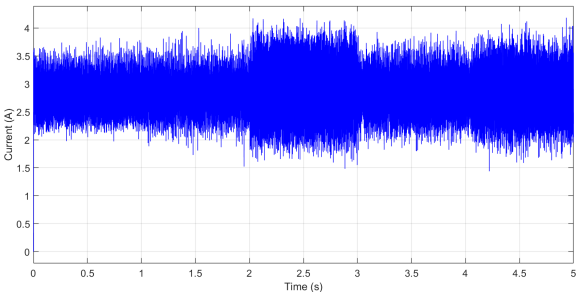
Figure 5.21: Speed, Torque, and Current Curves with Gains Optimization Using GWO



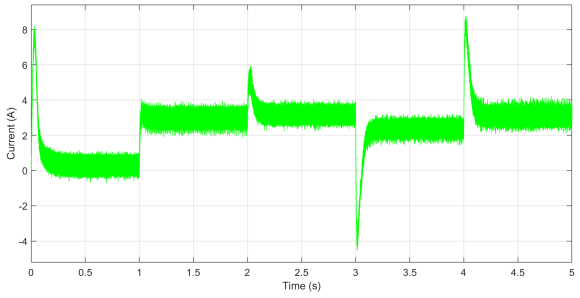
(a) Rotor Flux ψ_{dr}



(b) Rotor Flux ψ_{qr}



(c) Stator Current I_{ds}



(d) Stator Current I_{qs}

Figure 5.22: Fluxes and Currents Curves with Gains Optimization Using GWO

- **Discussion:**
  We notice no change in rotor fluxes and stator currents in comparison with the results before optimization. Our focus is mainly on the changed curves which correspond to Speed as a top event, torque, and phase currents as secondary events since they have a slight change. As a general look, the behavior of these later is better but we need to explain them individually. When comparing both algorithms speed results on the same graph (figure 5.23),we observe that the gains resulting from the PSO optimization yield a better rise time with a smaller overshoot, but they have a slightly slower settling time. This is made clear in the highlighted parts in figure 5.23. Furthermore, when the load is applied , we notice that the PSO gains have a smaller drop but an almost similar recovery time to that of GWO optimized system. Thus, we can conclude that the PSO gains perform better than their GWO counterparts.

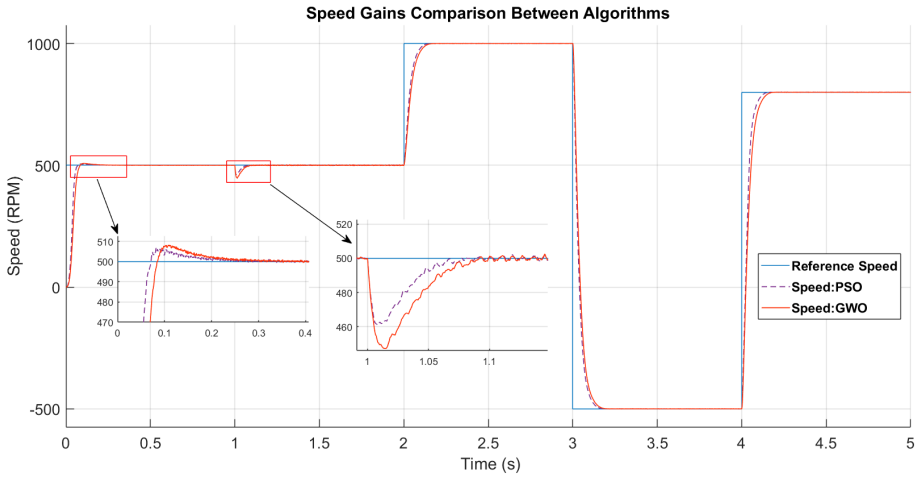


Figure 5.23: Comparison of Speed Curve Gains Optimization Between PSO and GWO

5.3.2 Mfs Optimization

In this subsection, the optimization of the input membership functions of error (e) and change in error (ce) has been performed as shown on the (figure 5.24). The gains now are fixed in their standard values as in section of (Fuzzy-PI). However, the Mfs are changing with 5 variables x(1), x(2), x(3), x(4), x(5) as shown on (figure 5.25). Building on the obtained data from the manually tuned Fuzzy-PI, we have estimated some near ranges for the Mfs to narrow down the space of research and get closer to better results, which are as follows :

Table 5.7: Membership Functions Ranges

Bounds/Mfs	**x(1)**	**x(2)**	**x(3)**	**x(4)**	**x(5)**
lower bound (lb)	0.01	0	0.35	0.49	0.5
upper bound (ub)	0.5	0.45	0.6	0.9	0.7

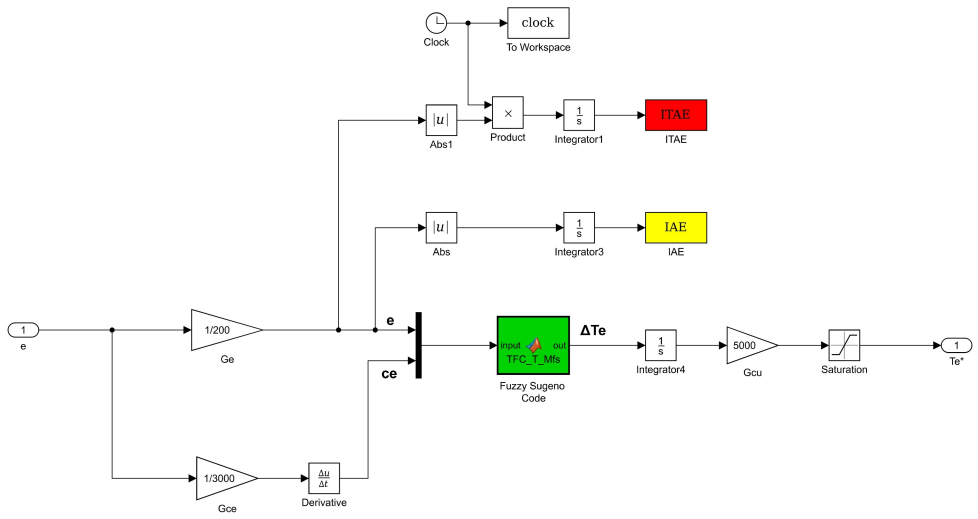


Figure 5.24: Simulink Block for Mfs Optimization

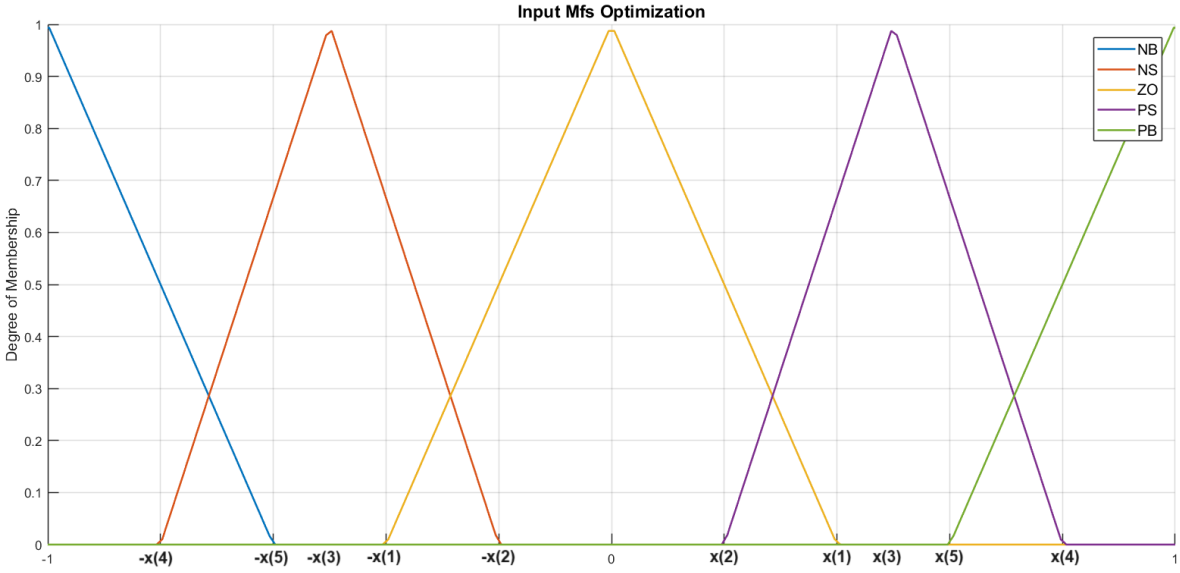


Figure 5.25: Input Mfs Optimization Scheme

The input Mfs inside the FLC code are modified as shown in the snippet below so the optimization algorithm can access the variable and modify it.

```
'NB', [-1 -1 -x(5)], ... %Trimf
'NS', [-x(4), -x(3), -x(2)], ... %Trimf
'ZO', [-x(1) 0 x(1)], ... %Trimf
'PS', [x(2), x(3), x(4)], ... %Trimf
'PB', [x(5) 1 1]; %Trimf
```

In addition to this, we added a penalty constraint to the optimization code in order to assure overlapping between Mfs which leads to continuous tracking of speed error over the whole interval [-1;1].

```
function fitness = func(x)
% Check the constraint
if x(2) >= x(1) || x(1) >= x(3) || x(3) >= x(5) || x(5) >= x(4)
    fitness = inf; % Penalize solutions that violate the constraint
    return;
end
```

a) Using PSO :

We have run the optimization using the same parameters of PSO Algorithm from previous subsection.

- **Results :**

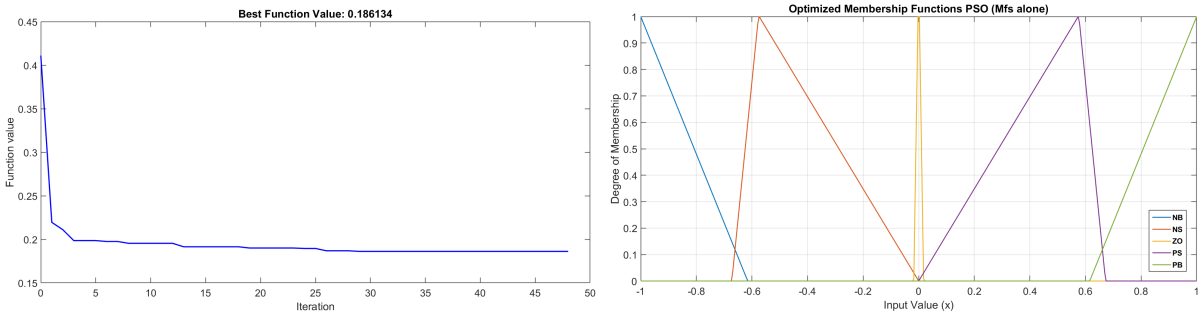
  According to the simulation done by the PSO algorithm and the simulink block (figure 5.19) the following Mfs results (table 5.8),(figure 5.26) and (figure 5.27) were obtained and its corresponding curves of speed, Torque, Phase current, Rotor fluxes and Stator currents are presented in (figure 5.28) and (figure 5.29).

Table 5.8: Optimized Membership Functions Using PSO

Mfs	**x(1)**	**x(2)**	**x(3)**	**x(4)**	**x(5)**
Values	0.017668	0	0.576034	0.672343	0.615262

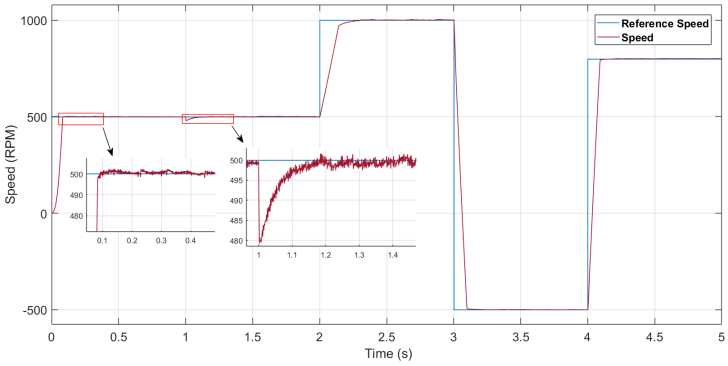
```
---------------------------------------
Optimization Results:
  - Best solution: [0.017668, 0.000000, 0.576034, 0.672343, 0.615262]
  - Best fitness value: 0.186134
```

Figure 5.26: Mfs Results

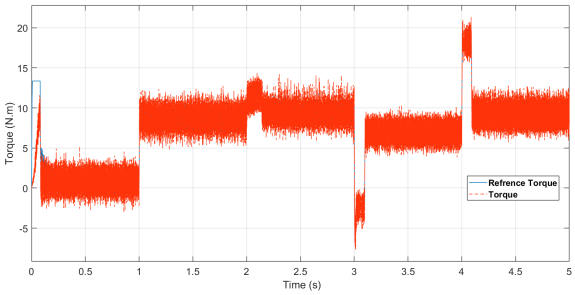


(a) Fitness Function Convergence Curve (b) Obtained Mfs Scheme Using PSO

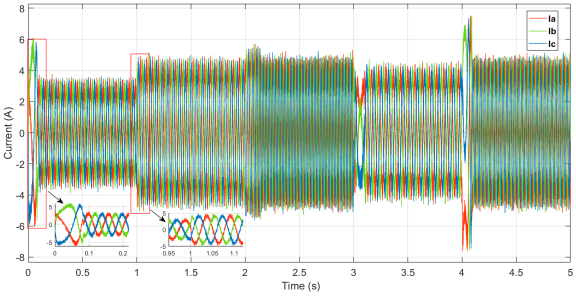
Figure 5.27: Mfs Optimization Results Using PSO



(a) Speed (w_r)

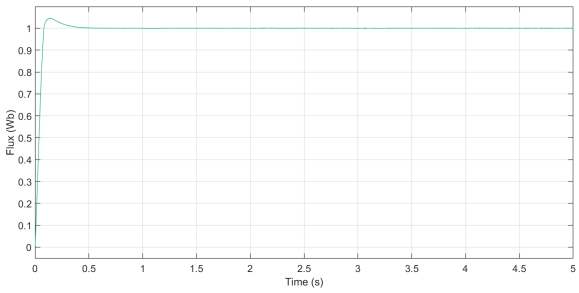


(b) Torque (T_e)

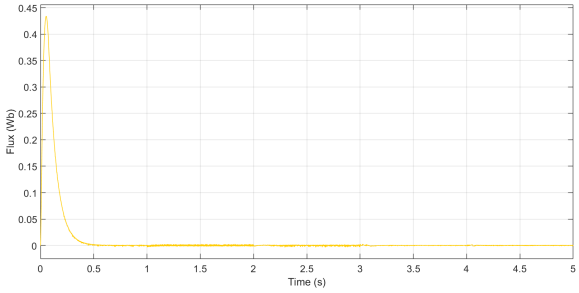


(c) Current (I_{abc})

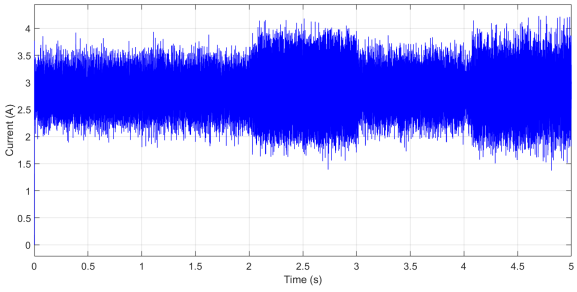
Figure 5.28: Speed, Torque, and Current Curves with Mfs Optimization Using PSO



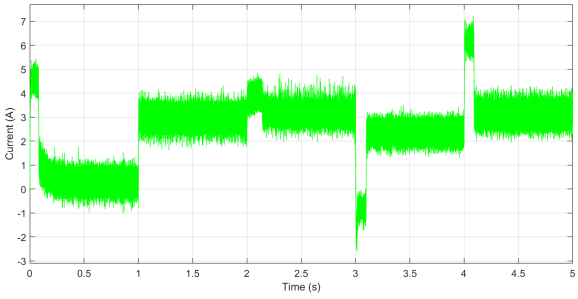
(a) Rotor Flux ψ_{dr}



(b) Rotor Flux ψ_{qr}



(c) Stator Current I_{ds}



(d) Stator Current I_{qs}

Figure 5.29: Fluxes and Currents Curves with Mfs Optimization Using PSO

b) Using GWO :

We have run the optimization using the same parameters of GWO Algorithm from previous subsection.

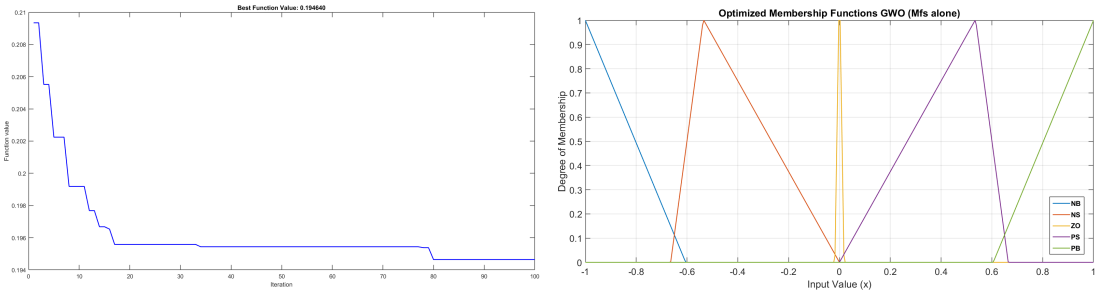
- **Results :**
  According to the simulation done by the GWO algorithm and the simulink block (figure 5.19) the following Mfs results (table 5.9),(figure 5.30) and (figure 5.31) were obtained and its corresponding curves of speed, Torque, Phase current, Rotor fluxes and Stator currents are presented in (figure 5.32) and (figure 5.33).

```
---------------------------------------
Optimization Results:
  - Best solution: [0.017984, 0.000000, 0.535811, 0.664358, 0.605264]
  - Best fitness value: 0.194640
```

Figure 5.30: Optimization Results

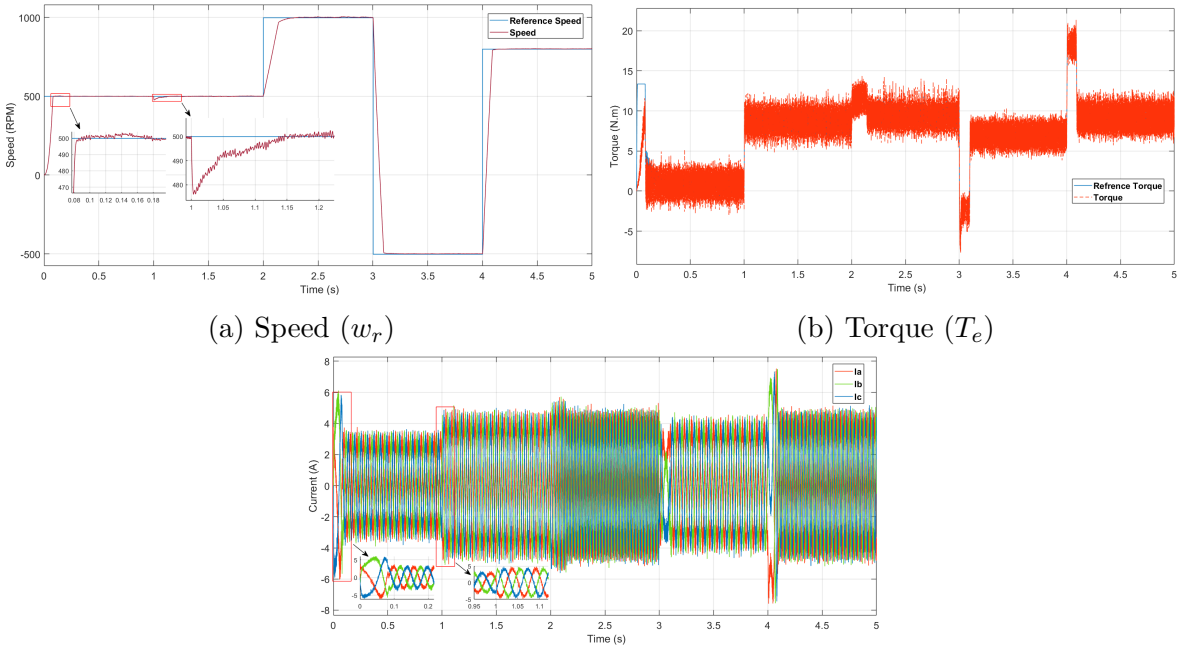
Table 5.9: Optimized Membership Functions Using GWO

Mfs	**x(1)**	**x(2)**	**x(3)**	**x(4)**	**x(5)**
Values	0.017984	0	0.535811	0.664358	0.605264



(a) Fitness Function Convergence Curve (b) Obtained Mfs Scheme Using GWO

Figure 5.31: Optimization Results

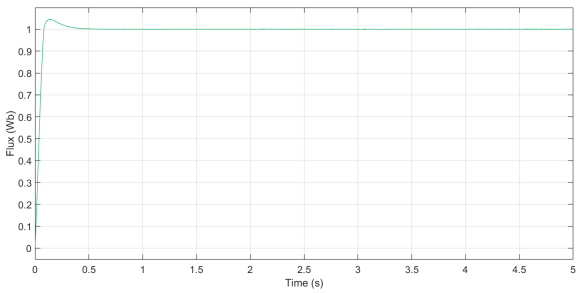


(a) Speed (w_r)

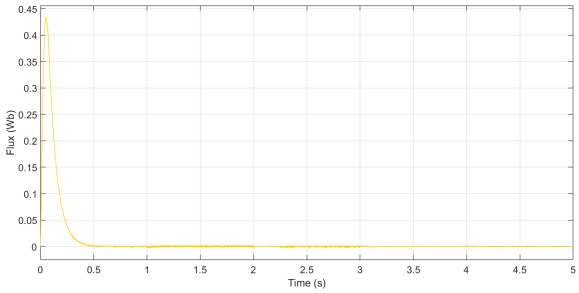
(b) Torque (T_e)

(c) Current (I_{abc})

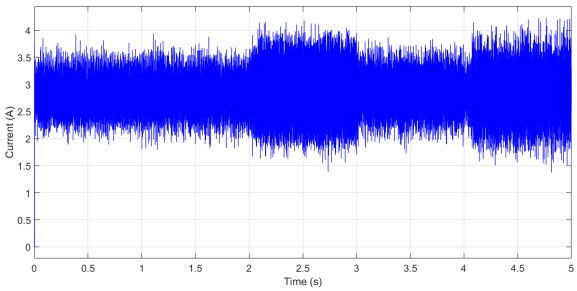
Figure 5.32: Speed, Torque, and Current Curves with Mfs Optimization Using GWO



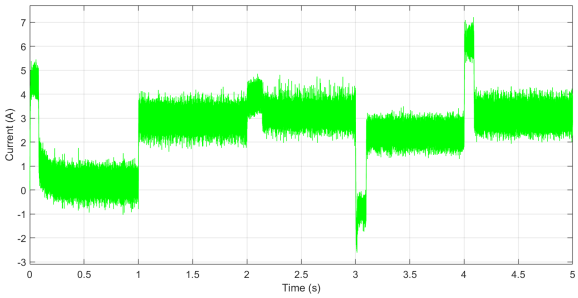
(a) Rotor Flux ψ_{dr}



(b) Rotor Flux ψ_{qr}



(c) Stator Current I_{ds}



(d) Stator Current I_{qs}

Figure 5.33: Fluxes and Currents Curves with Mfs Optimization Using GWO

- **Discussion:**
  We notice no change in rotor fluxes and stator currents in comparison with the results obtained before optimization. Our focus is mainly on the changed curves which correspond to Speed as top event, torque and phase currents as secondary events since they have a slight change. As general look, the behaviour of these later is better but we need to explain them individually. Firstly the speed, when comparing both algorithms speed results on the same graph (figure 5.34). We observe that the speed graph resulting from both optimization techniques are highly comparable. This is made most evident when we investigate (figure 5.26) and (figure 5.30), we notice from the aforementioned figures that the fitness values are similar.

5.3.3 Mfs and Gains together

In this subsection, we did the optimization of both Gains and input Mfs as shown on the (figure 5.35). The gains now are varying x(6), x(7), x(8) which correspond to G_e, G_{ce}, and G_{cu} respectively. On the other hand, the Mfs are changing with 5 variables x(1), x(2), x(3), x(4), x(5) as the on the previous subsection. However, in this part we are going to present the function value convergence curve, the corresponding Mfs scheme, and speed graph after 5 iterations, 30 iterations and last iteration to showcase how it converges towards the best solution. The ranges of search of both gains and Mfs are as shown on (table 5.10).

a) Using PSO :

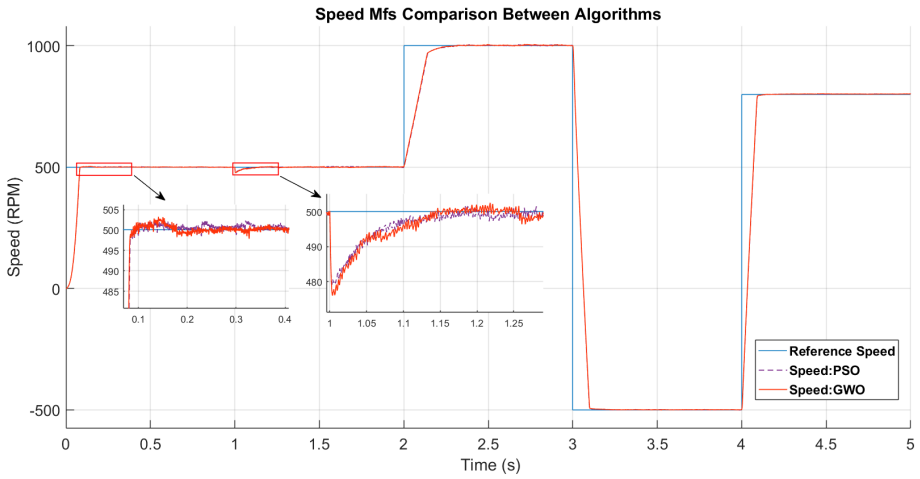


Figure 5.34: Comparison of Speed Curve Mfs Optimization Between PSO and GWO

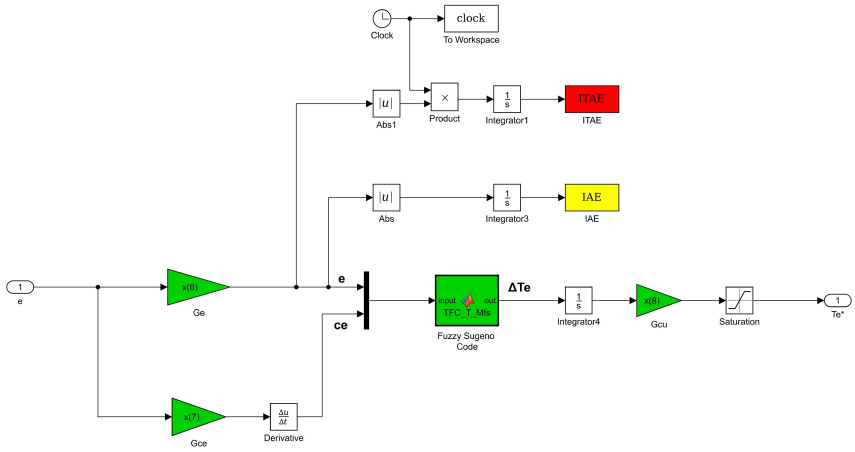


Figure 5.35: Simulink Block for Mfs and Gains Optimization

We have run the optimization using the same parameters of PSO Algorithm from previous subsections.

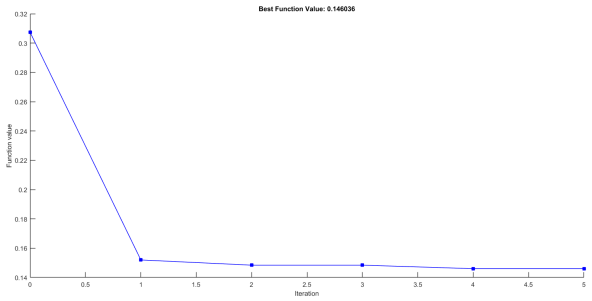
- **Results :**
  According to the simulation done by the PSO algorithm and the simulink block (figure 5.35) the following Mfs results and gains in iterations 5, 30 and last were obtained as shown in (figure 5.36), (figure 5.37) and (figure 5.38), and summarized in (table 5.11). the corresponding curves of speed, Torque, flux and currents for last iteration are presented in (subfigure 5.38d), (figure 5.39), and (figure 5.40).

Table 5.11: Results of Mfs and Gains Optimization Using PSO

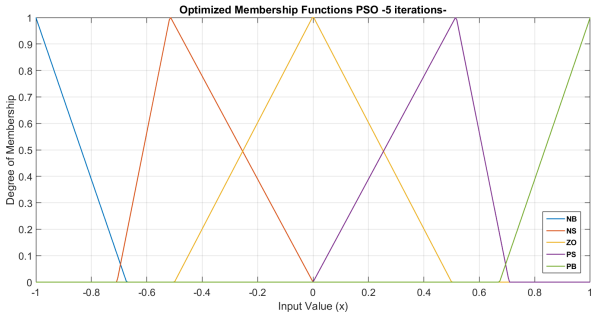
Values/Mfs-Gains	**x(1)**	**x(2)**	**x(3)**	**x(4)**	**x(5)**	**G_e**	**G_{ce}**	**G_{cu}**	**fitness**
5 iterations	0.5	0	0.516201	0.707616	0.672851	0.004618	0.0002	7999.022862	0.146036
30 iterations	0.124765	0.000001	0.369603	0.560259	0.537527	0.004551	0.0002	7487.175686	0.121766
Last iteration	0.019454	0	0.498829	0.604616	0.517720	0.004618	0.0002	7999.022862	0.117016

Table 5.10: Membership Functions Ranges

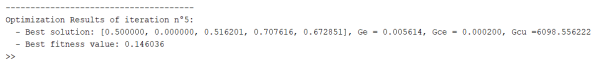
Bounds/Mfs-Gains	**x(1)**	**x(2)**	**x(3)**	**x(4)**	**x(5)**	$\mathbf{G_e}$	$\mathbf{G_{ce}}$	$\mathbf{G_{cu}}$
lower bound (lb)	0.01	0	0.35	0.49	0.5	$\frac{1}{300}$	$\frac{1}{6000}$	5000
upper bound (ub)	0.5	0.45	0.6	0.9	0.7	$\frac{1}{100}$	$\frac{1}{3000}$	8000



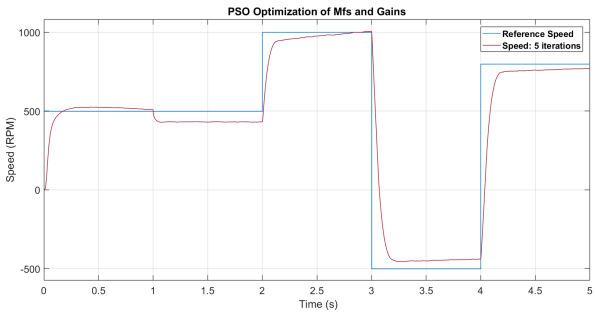
(a) Convergence Curve



(b) Mfs Scheme



(c) Optimization Results



(d) Speed Curve

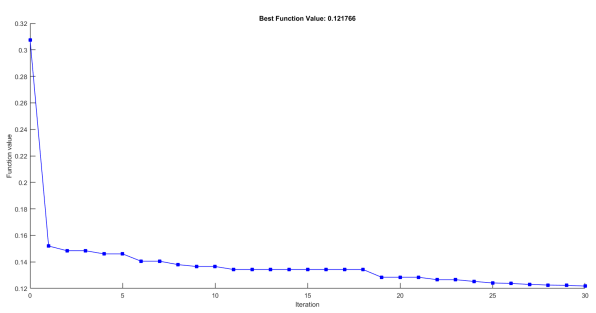
Figure 5.36: Results of optimization of Mfs and Gains in 5 iterations Using PSO

– The following results of rotor fluxes, stator currents, torque, and three phase current curves are taken in the last iteration.

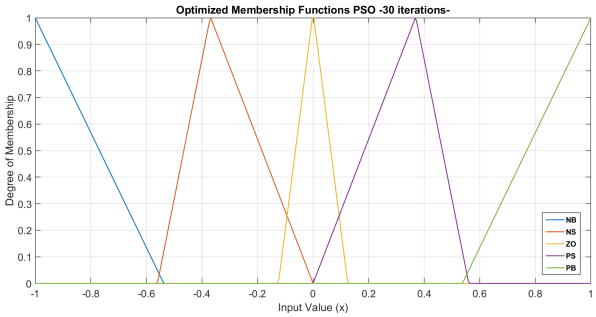
b) Using GWO :

We have run the optimization using the same parameters of GWO Algorithm from previous subsections.

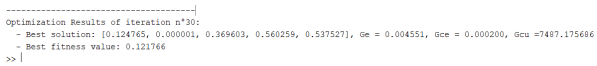
- **Results :**
  According to the simulation done by the GWO algorithm and the simulink block (figure 5.35) the following Mfs results and gains in iterations 5, 30 and last were obtained as shown in (figure 5.41), (figure 5.42) and (figure 5.43), and summarized in (table 5.12). the corresponding curves of speed, Torque, flux and currents for last iteration are presented in (subfigure 5.43d), (figure 5.45), and (figure 5.46) and comparison of speed between iterations using GWO in (figure 5.44).



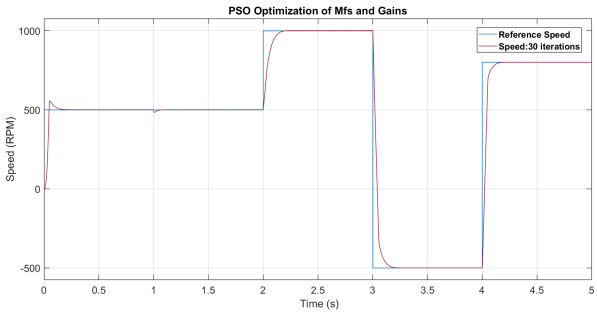
(a) Convergence Curve



(b) Mfs Scheme

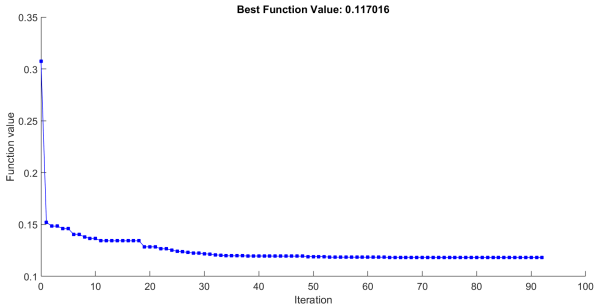


(c) Optimization Results

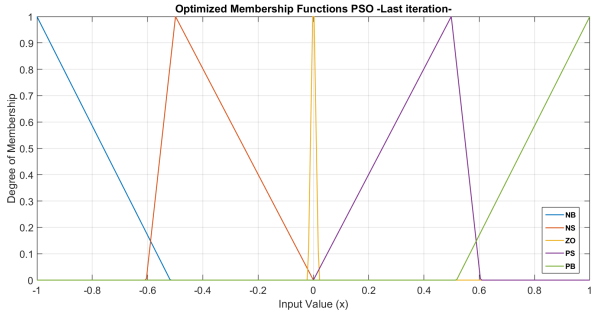


(d) Speed Curve

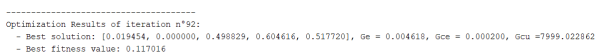
Figure 5.37: Results of optimization of Mfs and Gains in 30 iterations Using PSO



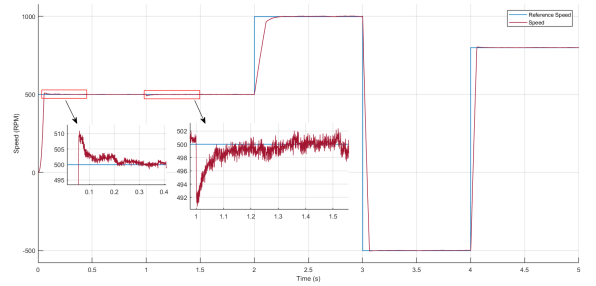
(a) Convergence Curve



(b) Mfs Scheme



(c) Optimization Results

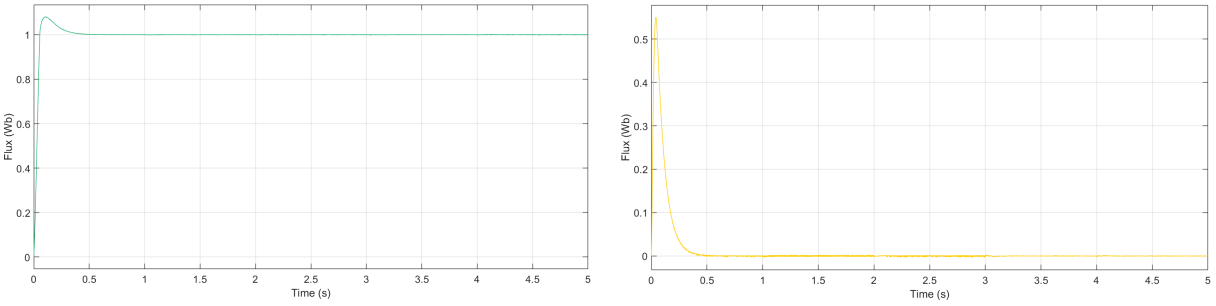


(d) Speed Curve

Figure 5.38: Results of optimization of Mfs and Gains in Last iteration Using PSO

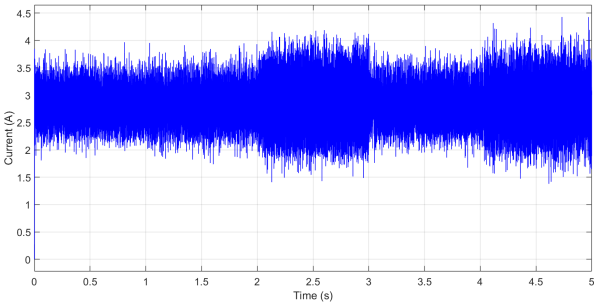
Table 5.12: Results of Mfs and Gains Optimization Using GWO

Values/Mfs-Gains	**x(1)**	**x(2)**	**x(3)**	**x(4)**	**x(5)**	**G_e**	**G_{ce}**	**G_{cu}**	fitness
5 iterations	0.076857	0	0.35	0.629526	0.5	0.004	0.0002	5000	0.141834
30 iterations	0.113215	0	0.35	0.606566	0.5	0.004	0.0002	5000	0.141641
Last iteration	0.076688	0	0.350951	0.515726	0.503249	0.004	0.0002	5000	0.141410

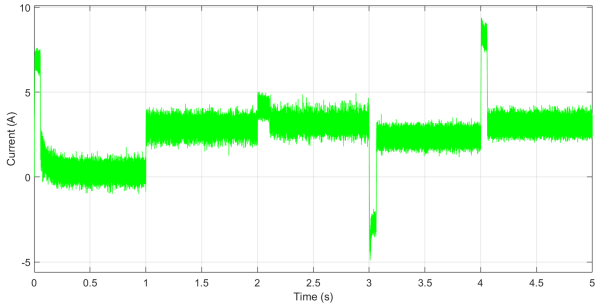


(a) Rotor Flux ψ_{dr}

(b) Rotor Flux ψ_{qr}

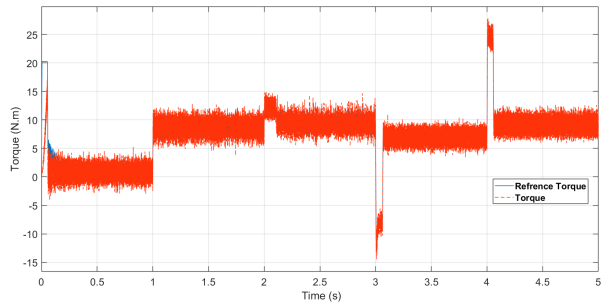


(c) Stator Current I_{ds}

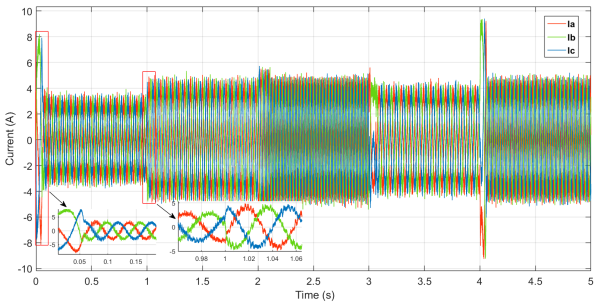


(d) Stator Current I_{qs}

Figure 5.39: Fluxes and Currents Curves with Mfs and Gains Optimization Using PSO



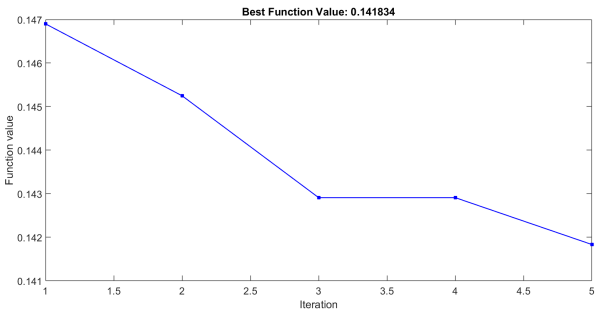
(a) Torque (T_e)



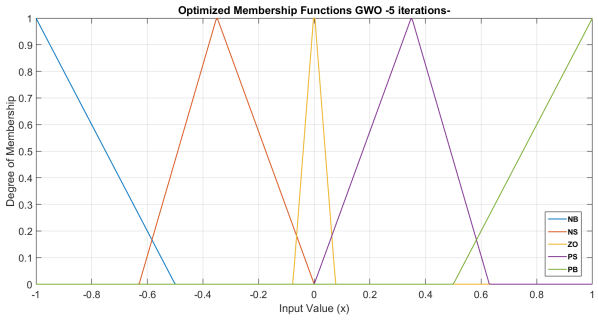
(b) Current (I_{abc})

Figure 5.40: Torque, and Current Curves with Mfs and Gains Optimization Using PSO

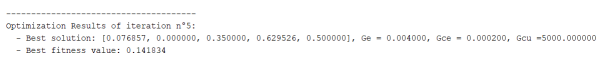
- **Discussion:**
  We notice no change in rotor fluxes and stator current compared to the results seen before the optimization. Our focus is mainly on the changed curves of which correspond to Speed as top event, torque and phase current as secondary events since their change is somewhat unperceivable. We can notice from figure 5.47 that the response gained from PSO optimization is faster. It has a shorter rise time but at the expense of having an overshoot. On the reverse side, both systems settles at approximately the same time. When applying the load, both systems respond with a drop but as noticed from the aforementioned graph that PSO system has a lower drop and a faster recovery in comparison with GWO system. We can also notice that the PSO system better follows the reference speed curve and is more responsive to speed changes.



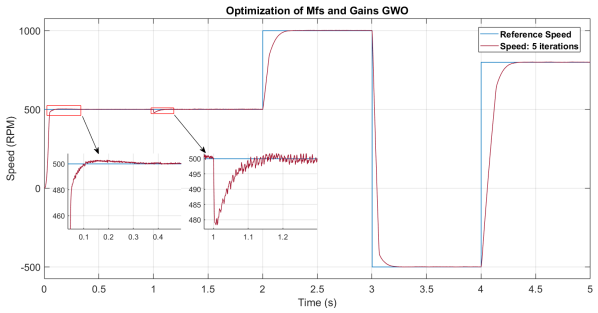
(a) Convergence Curve



(b) Mfs Scheme

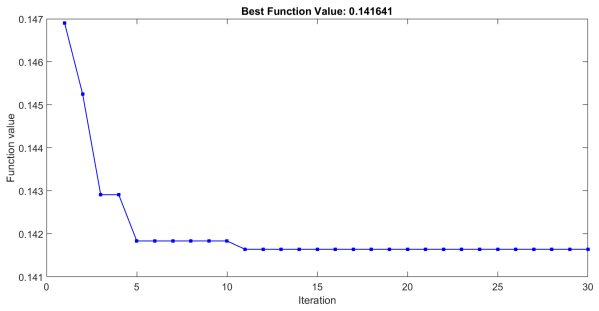


(c) Optimization Results

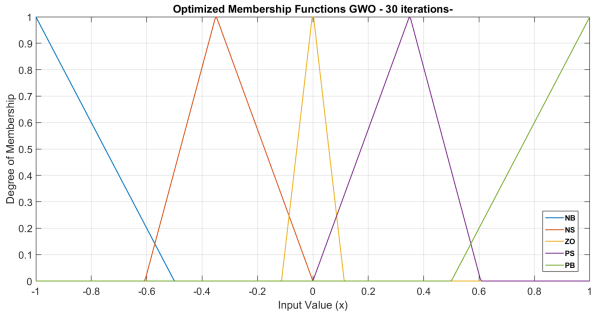


(d) Speed Curve

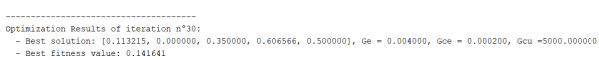
Figure 5.41: Results of optimization of Mfs and Gains in 5 iterations Using GWO



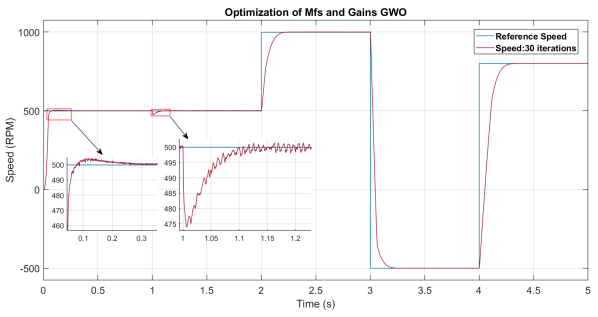
(a) Convergence Curve



(b) Mfs Scheme

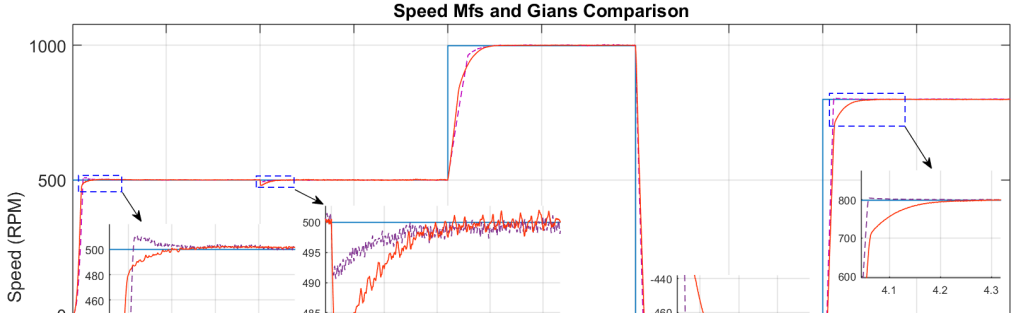


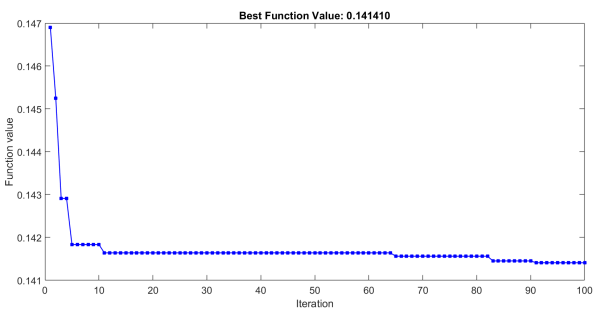
(c) Optimization Results



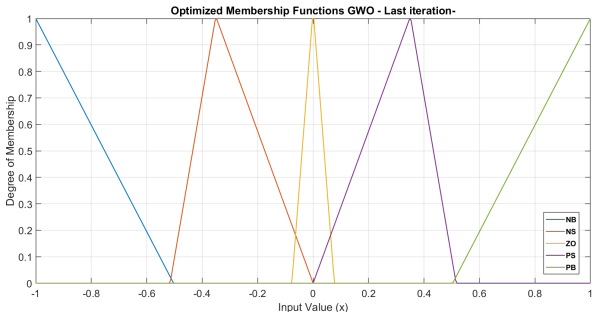
(d) Speed Curve

Figure 5.42: Results of optimization of Mfs and Gains in 30 iterations Using GWO

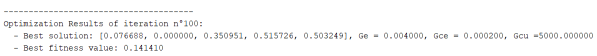




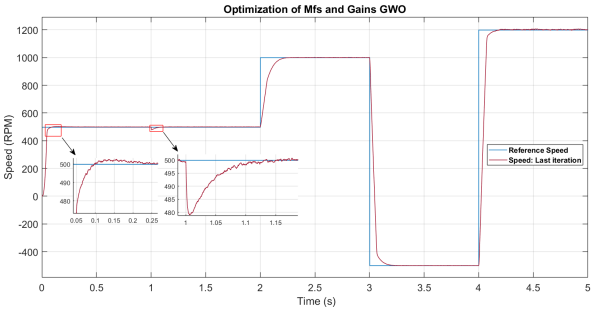
(a) Convergence Curve



(b) Mfs Scheme



(c) Optimization Results



(d) Speed Curve

Figure 5.43: Results of optimization of Mfs and Gains in Last iteration Using GWO

Table 5.13: Comparison between the Fuzzy-PI controllers

Controller/Charachteristics	**%OS**	$\mathbf{T_r}$	$\mathbf{T_s}$	**Drop**	$\mathbf{T_{Recovery}}$	**fitness**
Fuzzy-PI Manually tuned	0	0.1202	0.2441	35.8	0.167	/
Fuzzy-PI Optimized Gains PSO	1.4	0.0518	0.241	38.9	0.111	0.123097
Fuzzy-PI Optimized Gains GWO	1.78	0.0645	0.221	51.8	0.105	0.149275
Fuzzy-PI Optimized Mfs PSO	0	0.083	0.178	20.2	0.18	0.186134
Fuzzy-PI Optimized Mfs GWO	0	0.0831	0.189	23.4	0.148	0.194640
Fuzzy-PI Optimized Mfs & Gains PSO	1.4	0.05255	0.12	17.2	0.18	0.117016
Fuzzy-PI Optimized Mfs & Gains GWO	0.4	0.0465	0.13	20.8	0.112	0.141410

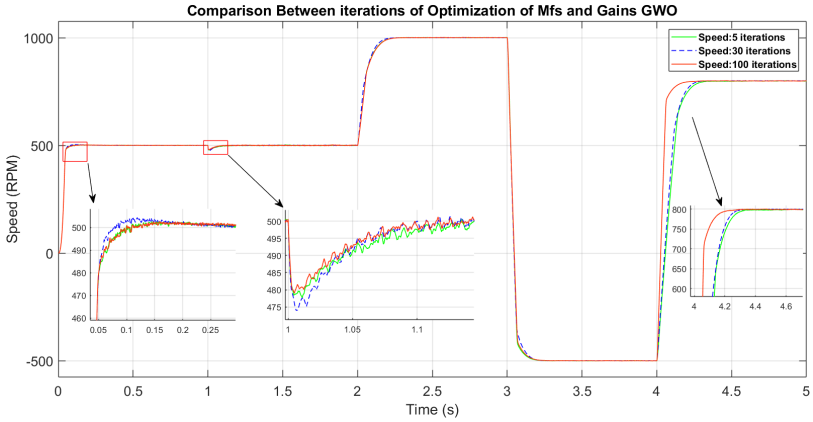
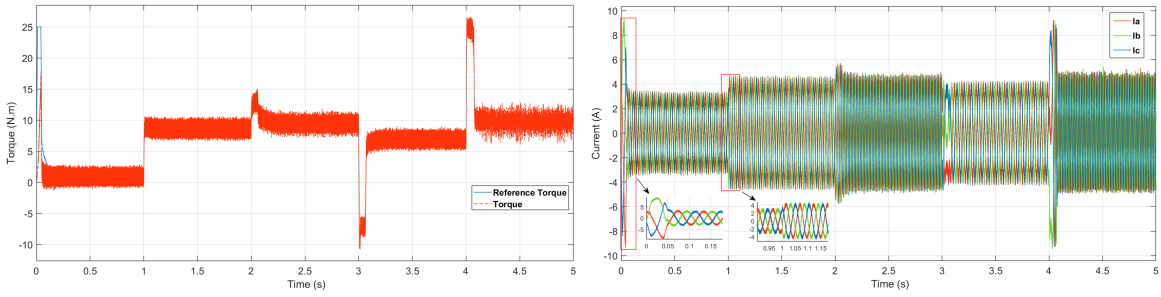
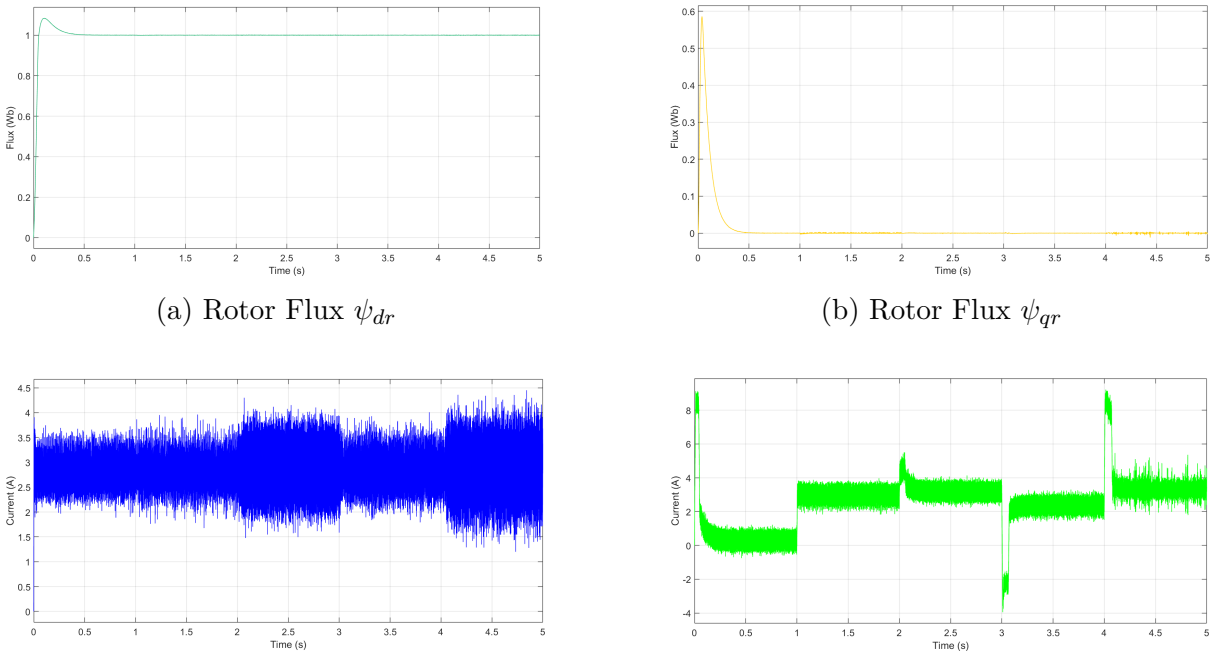


Figure 5.44: Comparison Between iterations of Optimized Mfs and Gains GWO



(a) Torque (T_e) (b) Current (I_{abc})

Figure 5.45: Speed, Torque, and Current Curves with Mfs and Gains Optimization Using GWO



(a) Rotor Flux ψ_{dr} (b) Rotor Flux ψ_{qr}

(c) Stator Current I_{ds} (d) Stator Current I_{qs}

Figure 5.46: Fluxes and Currents Curves with Mfs and Gains Optimization Using GWO

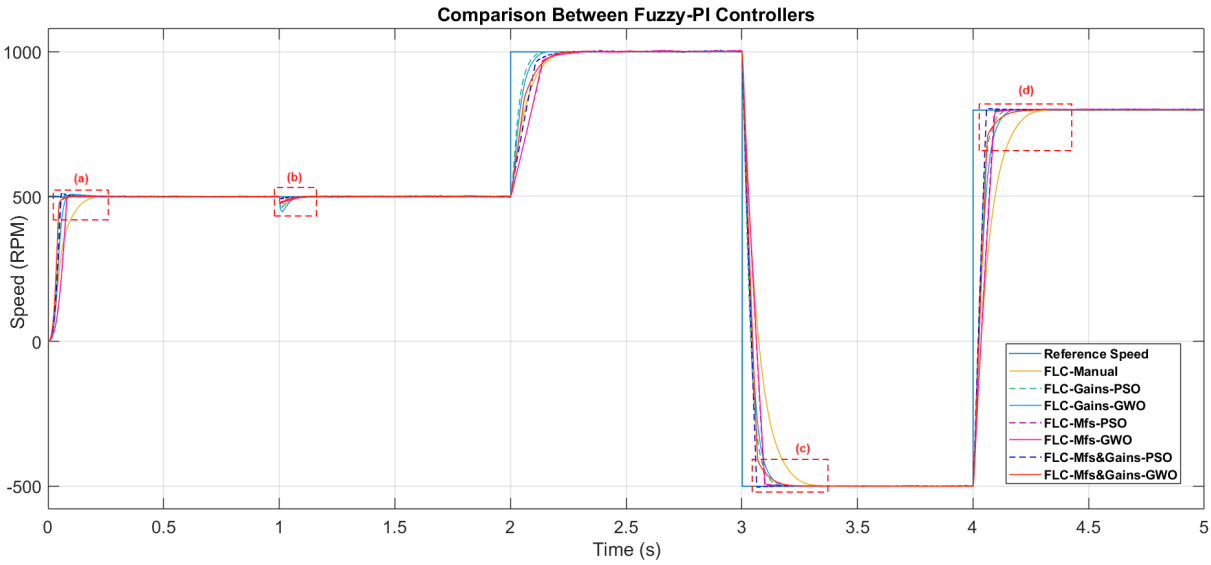


Figure 5.48: Comparison of Speed Between FLCs

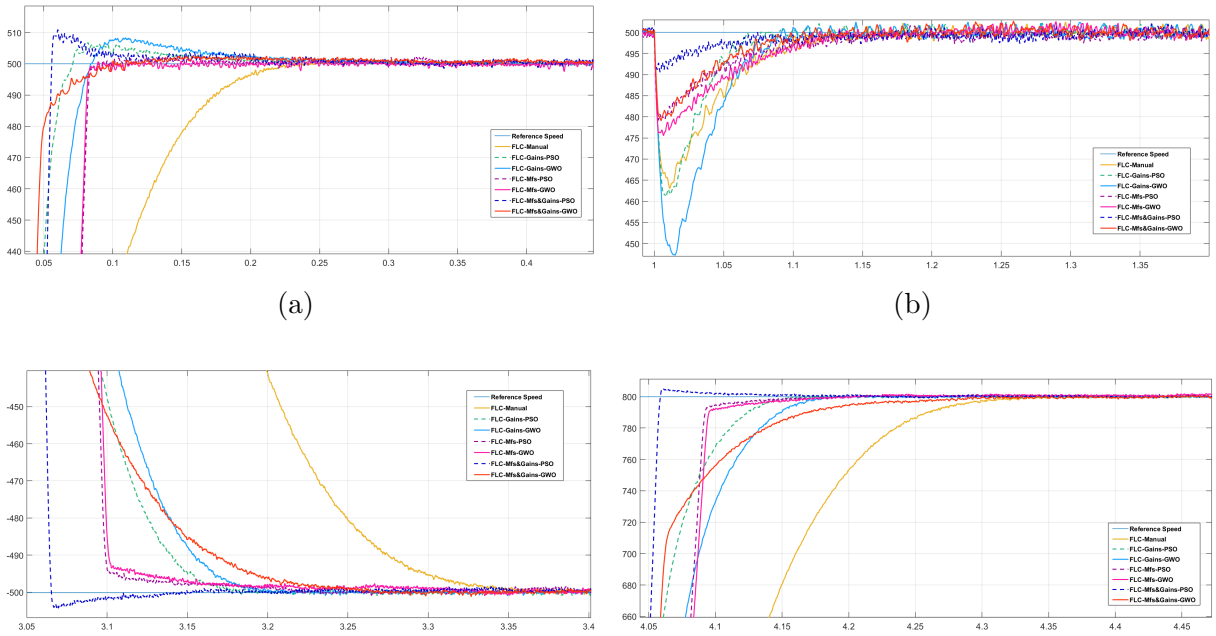


Figure 5.49: Speed Comparison several areas

5.4 Robustness of Fuzzy-PI Controller

Motor parameters are subjected to changes due to several reasons :

- Manufacturing tolerances (slight variations even in new motors)
- Wear and tear over time (friction, heat, etc.)
- Changes in operating conditions (temperature, load)

Thus, It is recommended to test if the controller can maintain the desired performances despite these variations. This is a way to stress-test the motor virtually to ensure it performs well under a variety of real-world conditions. To do so, we suggest three parameters changes which are : Stator resistance (R_s), Rotor resistance (R_r), Moment of inertia (J). The results shown below are for both PI and Optimized Fuzzy-PI controller[3] , to see which one has better performance and more robust to changes in motor's parameters.

5.4.1 Changing stator resistance (R_s)

This test is performed by increasing the value of stator resistance with 50% which means the $R_s = 7.89\ \Omega$, this variation will be applied at t = 1.5 (s) to easily distinguish it from speed changes.

- **Results :**

  a) Using PI Speed Controller :

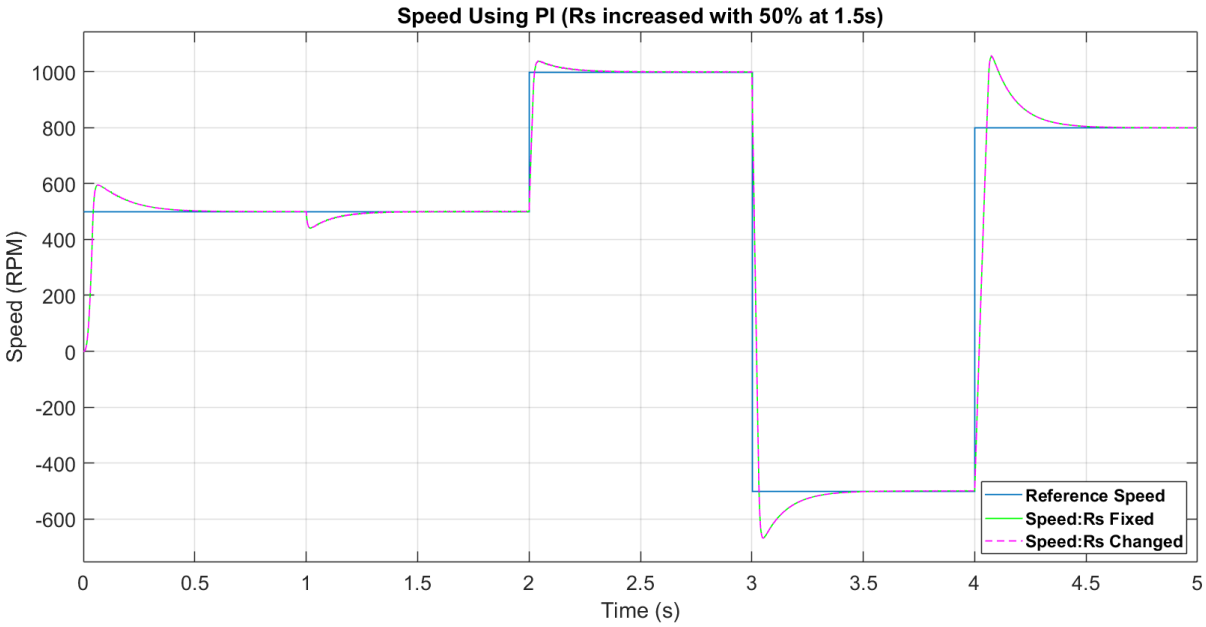


Figure 5.50: Speed Curve Using PI with Changed R_s

[3]Optimized GWO with Mfs and Gains is chosen for this part

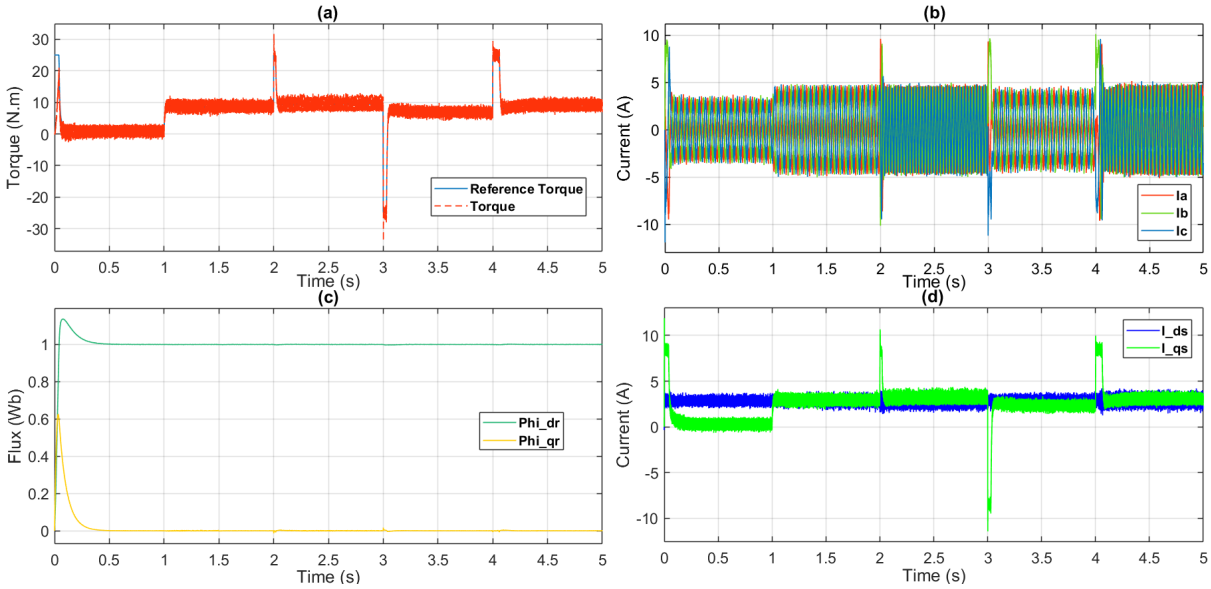


Figure 5.51: Torque, Fluxes, and Currents Curves Using PI with Changed R_s

b) Using Optimized Fuzzy-PI Speed Controller :

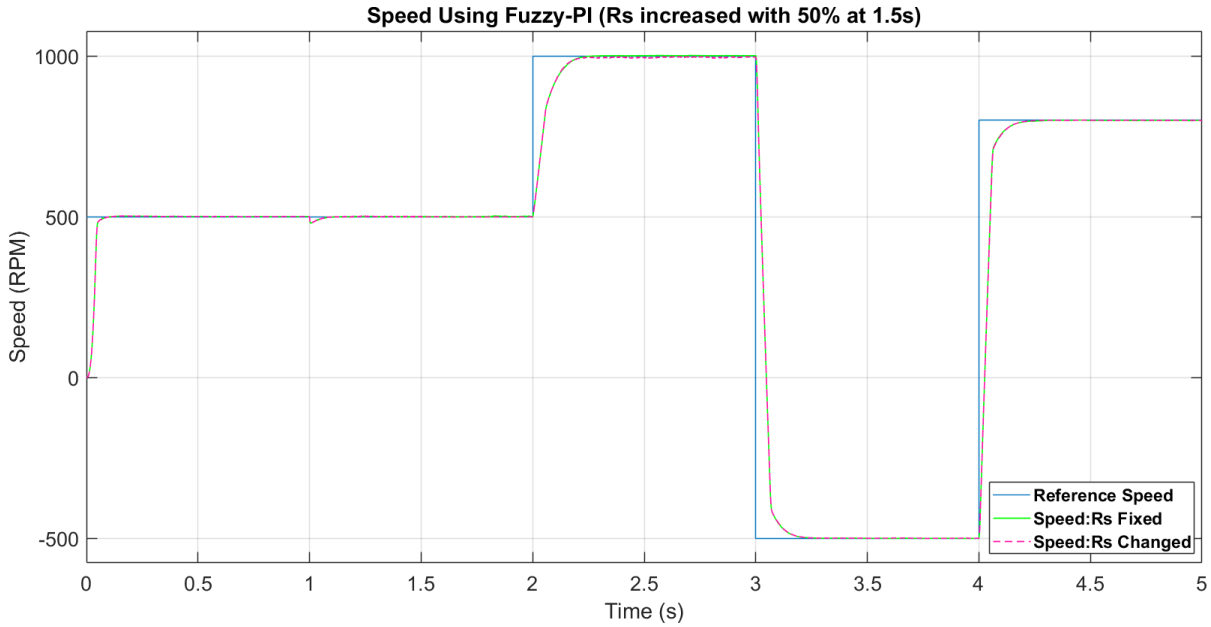


Figure 5.52: Speed Curve Using Optimized Fuzzy-PI with Changed R_s

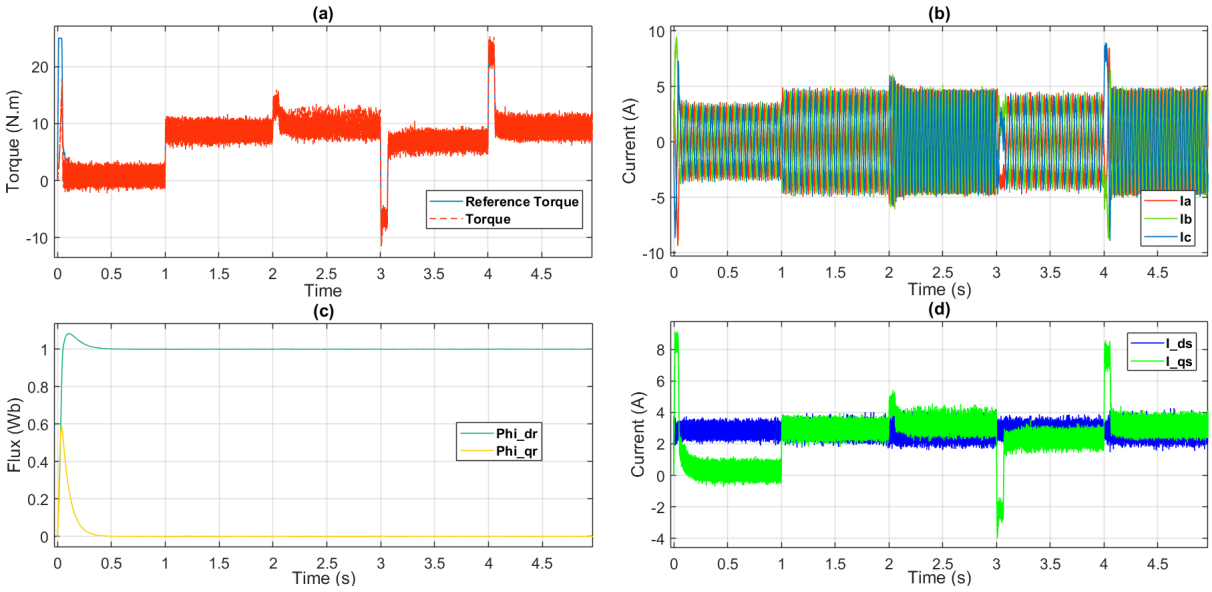


Figure 5.53: Torque, Fluxes, and Currents Curves Using Optimized Fuzzy-PI with Changed R_s

5.4.2 Changing rotor resistance (R_r)

This test is performed by increasing the value of rotor resistance with 50% which means the $R_r = 6.74205\ \Omega$, this variation will be applied at t = 1.5 (s) to easily distinguish it from speed changes.

- **Results :**

  a) Using PI Speed Controller :

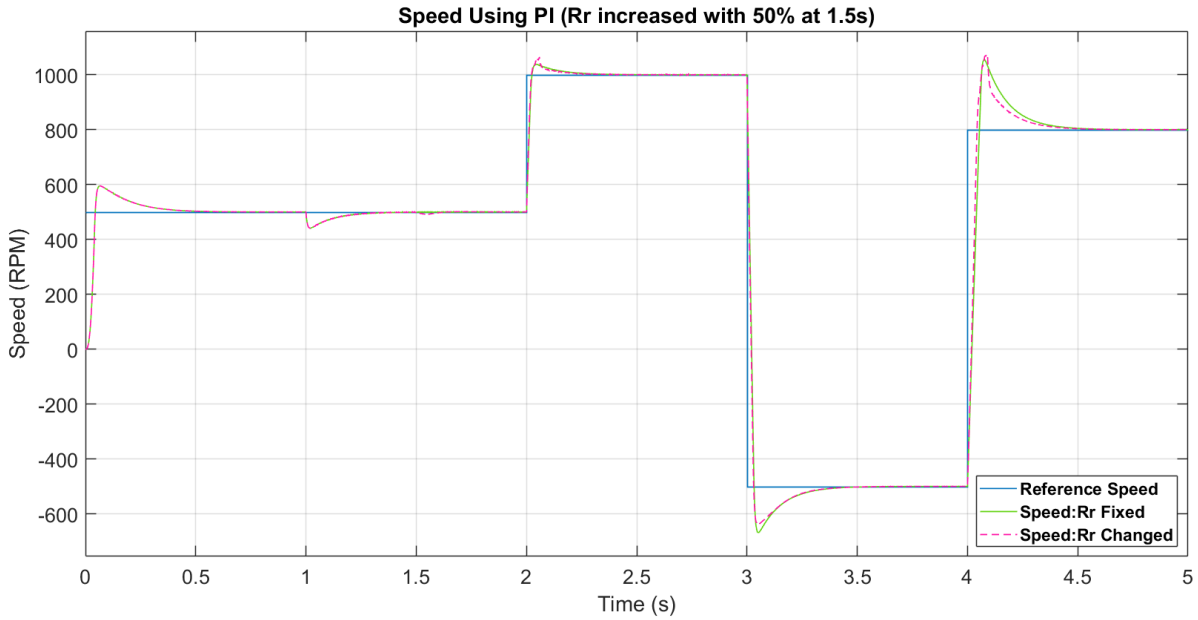


Figure 5.54: Speed Curve Using PI with Changed R_r

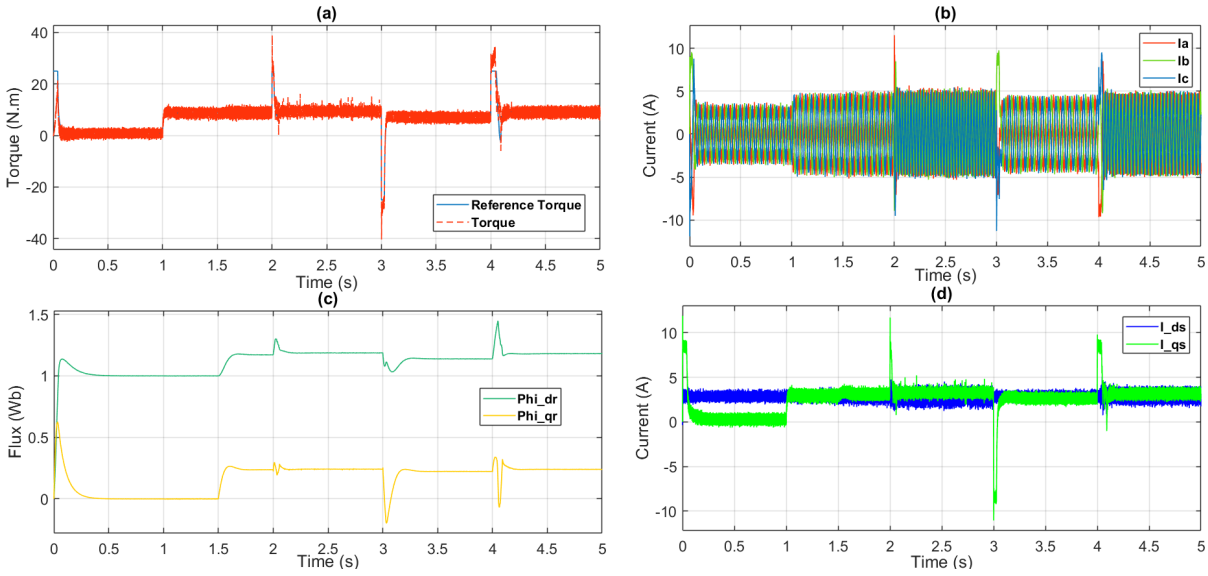


Figure 5.55: Torque, Fluxes, and Currents Curves Using PI with Changed R_r

b) Using Optimized Fuzzy-PI Speed Controller :

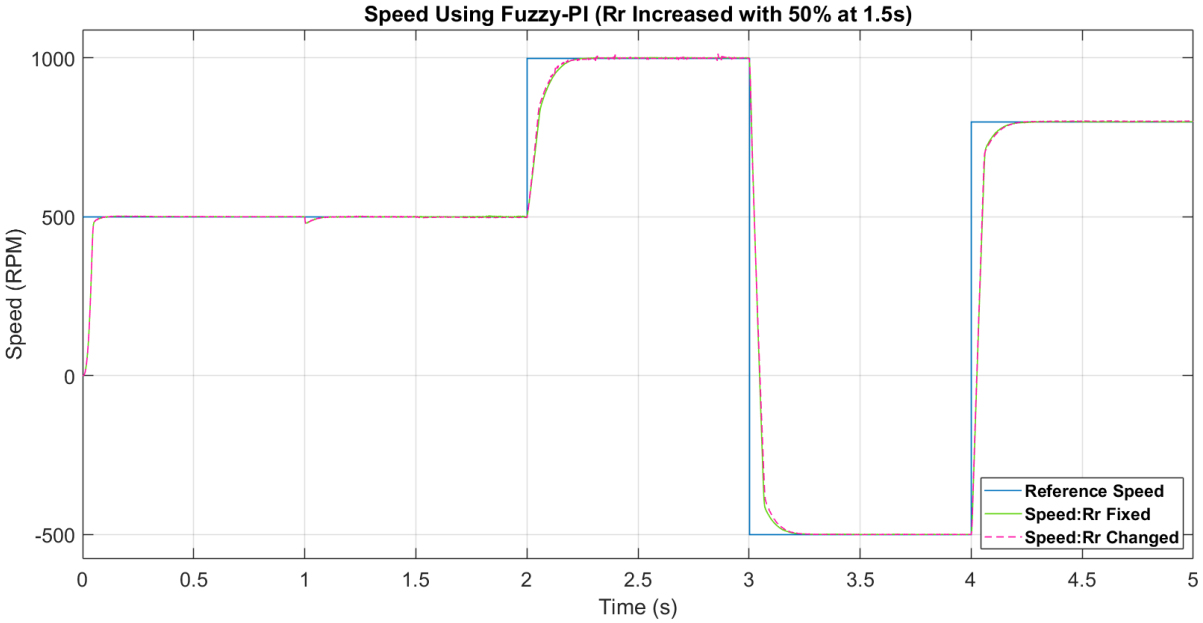


Figure 5.56: Speed Curve Optimized Fuzzy-PI with Changed R_r

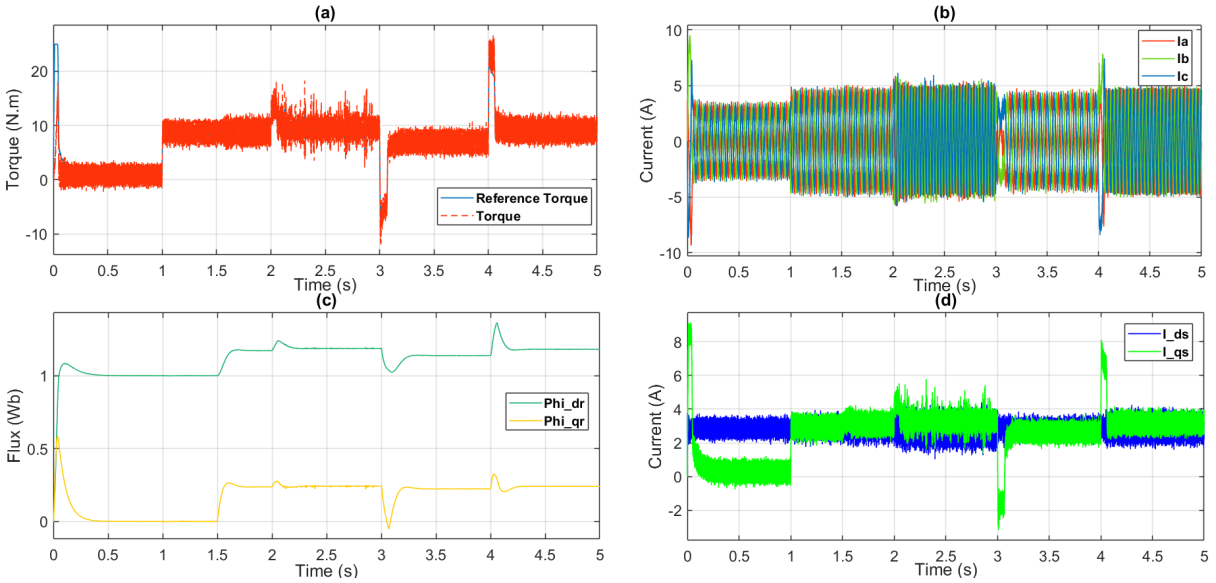


Figure 5.57: Torque, Fluxes, and Currents Curves Optimized Fuzzy-PI with Changed R_r

5.4.3 Changing moment of Inertia (J)

This test is performed by increasing the value of moment of inertia with 50% which means the J = 0.01008255 kg/m^2, this variation will be applied at t = 1.5 (s) to easily distinguish it from speed changes.

- **Results :**

  a) Using PI Speed Controller :

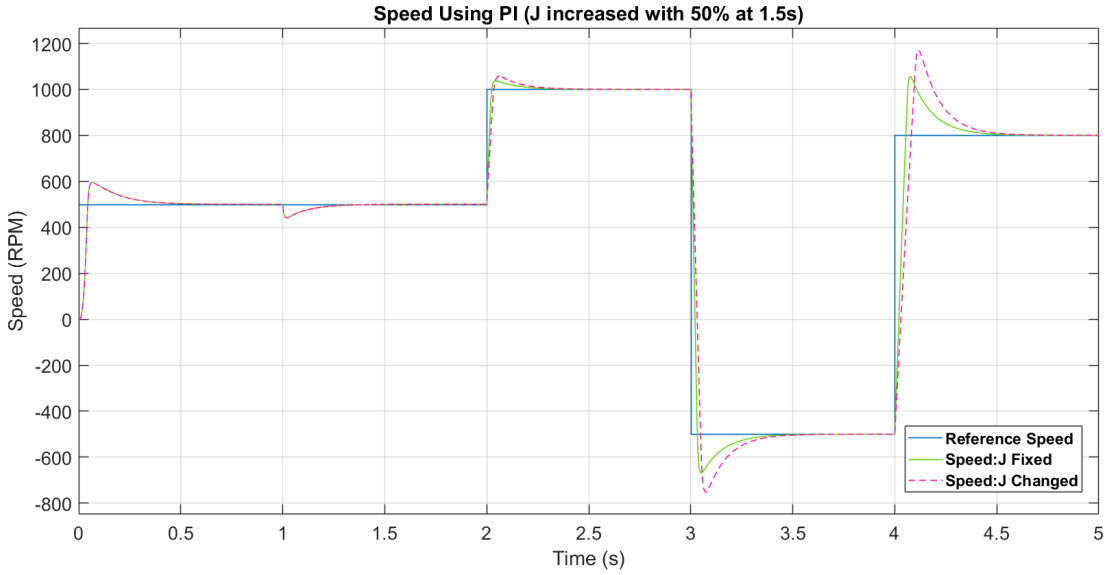


Figure 5.58: Speed Curve Using PI with Changed J

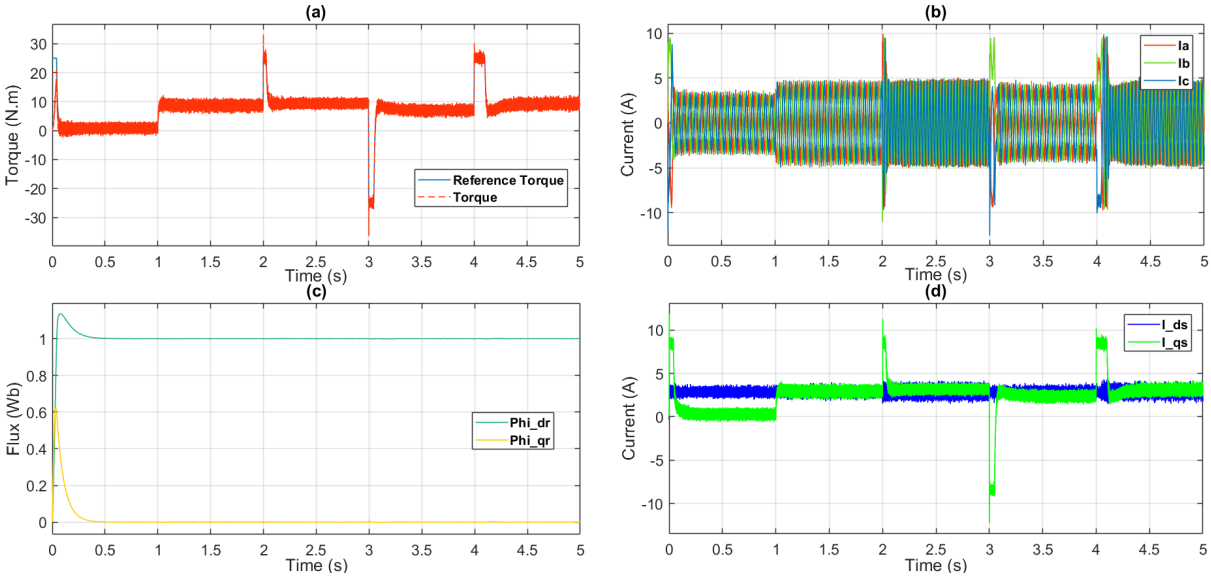


Figure 5.59: Torque, Fluxes, and Currents Curves Using PI with Changed J

b) Using Optimized Fuzzy-PI Speed Controller :

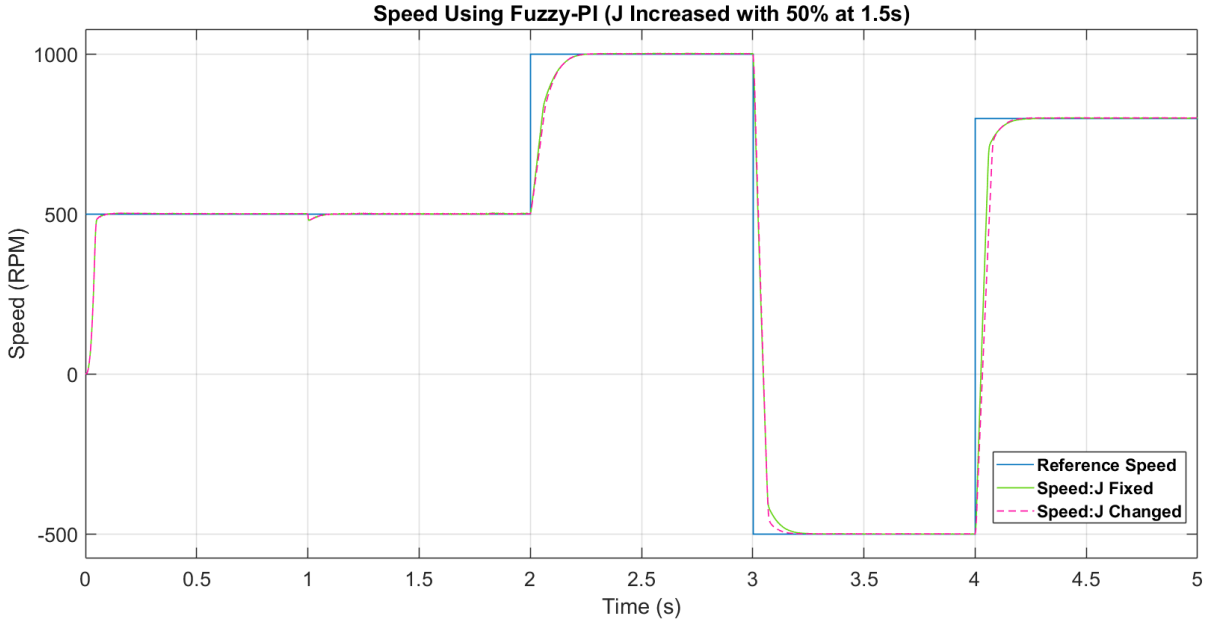


Figure 5.60: Speed Curve Using Optimized Fuzzy-PI with Changed J

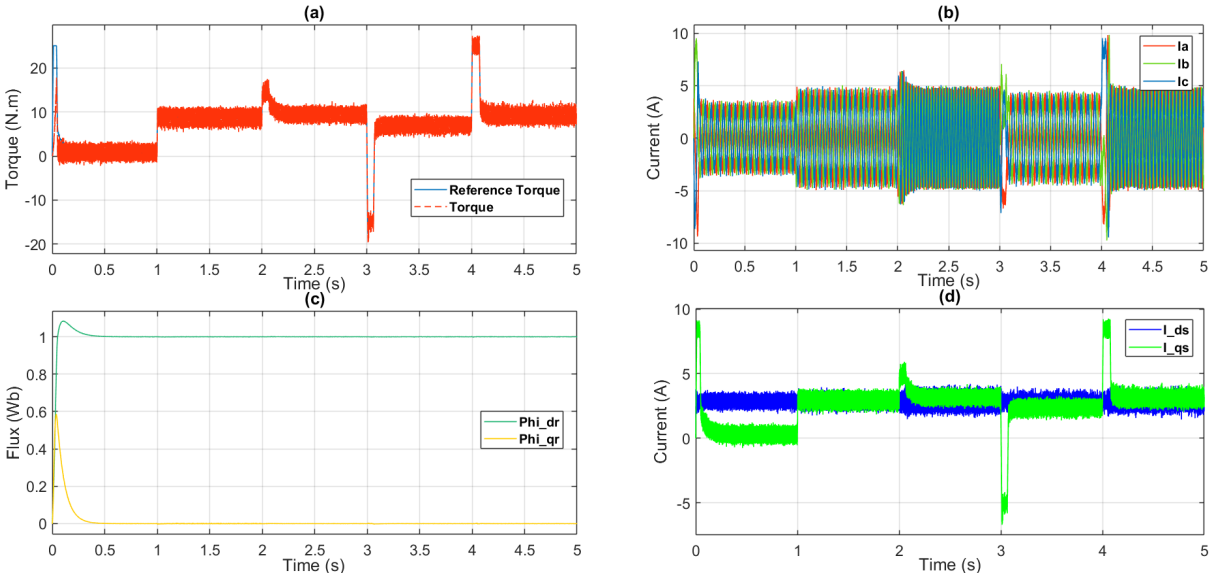


Figure 5.61: Torque, Fluxes, and Currents Curves Using Optimized Fuzzy-PI with Changed J

- **Comparison with average percentage deviation:**

  This table summarizes the previous graphs. It represents the percentage deviation of Overshoot and Rise time between the speed curve with varying parameter relative to fixed parameter for both conventional PI and Fuzzy-PI in different cases.

Table 5.14: Comparison of Percentage Deviation Between Conventional PI and Fuzzy-PI

	Rs		Rr		J	
	PI	Fuzzy-PI	PI	Fuzzy-PI	PI	Fuzzy-PI
PO %	0	0	1.515	0.370	10.795	0
Rise Time %	0	0	0.240	0.047	0.767	0.41

5.4.4 Discussion

Investigating figure 5.50 we note that the change of R_s have no effect on the PI controller, but on the other hand the change of inertia moment affects the speed profile which is depicted in figure 5.58. From figure 5.59 we can note that this change is not accompanied by any change in the Torque or the decoupling. On the other hand when changing R_r, not only that the speed profile has changed, but we note loss of decoupling.
Applying the same changes to our Fuzzy PI controller we note from figure 5.52 and figure 5.53 that there is no effect on either the speed profile or the Torque and Flux respective profiles. The same can be said about the change in inertia moment J. When changing R_r, we note that speed remains constant, but there is a loss of decoupling.

5.5 General Discussion

Initially, we compared the performance of the classical PI controller, PI controller with reference filter, and Fuzzy PI controller.It was observed that the conventional tuning method results were unsatisfactory in terms of both steady state and transient responses. Hence, the need to find another control method. The Fuzzy PI controller resulted in the most adequate speed graph out of the three and achieving better response. It yielded a smaller rise time and no overshoot indicating a better transient response.
After establishing that Fuzzy PI controller is more suitable, we investigated the effects of optimizing the gains using PSO and GWO algorithms. This yielded in an even better output with a better fitness value. For PSO-optimized gains, we noticed a smaller overshoot coupled with a faster rise time and settling time. In General the PSO system was better than the GWO system. Empirically, both systems were an improvement over the previous discussed ones.
Next, we moved to optimizing the MFs using both meta-heuristic algorithms and we noticed a deterioration in the quality of our results. In this part the fitness function was the main indicator of the better system because the speed graph showed the two systems almost superimposed . The new findings coincide with the previous ones in which the performance of the PSO was better than that of the GWO.
In the third optimization scheme, we optimized both the MFs and gains. We found that the PSO system has an overshoot but a better following of the reference signal; even its response to the load was better than that of the GWO system.
Lastly, we checked the robustness of our control scheme for various situations (changing R_r, changing R_s, and changing moment of inertia J). We noticed that the changing R_s, and the changing moment of inertia had little to no no effect on our system. In contrast, the change of R_r caused the loss of decoupling between ψ_{qr} and ψ_{qr} . We found that even if we lose the decoupling, the speed retains its shape.

5.6 Conclusion

In conclusion, the performed simulations and results showcased the superiority of the FLC-PI with meta-heuristic optimization algorithms compared to the conventional PI controller. FLC-PI achieved faster reference tracking accuracy, improved transient response, and superior robustness to parameter variations. These findings underscore the effectiveness of FLC in handling non-linearities and uncertainties inherent in induction motors. These meta-heuristic algorithms represent a viable approach to streamline the tuning process of the FLC-PI by reducing the time-consuming in the manual tuning and efforts to understand the linguistic knowledge.

General Conclusion

General Conclusion

The work presented in this thesis concerns the study of Fuzzy-PI based on Indirect vector control for speed control of induction motor, and the optimization of this later using meta-heuristic algorithms for the gains and membership functions (Mfs).

It started by modeling the induction motor which is a necessary step in designing an adequate control system that can be adapted to speed drivers, enabling us to deduce precise control laws by manipulating the mathematical equations describing the behavior of the induction machine accurately. It merits bringing up that the design of IM is very intricate on the inside, due to a lack of a perfect model that incorporates all variables involved in its working mechanism. The model chosen to work within this project is the alpha-beta model, which can be connected with the inverter.

Indirect vector control method has been introduced as an efficient-technique for the control of induction motor due to its reliability and cost-effectiveness. It aims to maintain decoupled torque and flux to produce maximum torque separated from speed control. The most commonly used controller for this method is the conventional PI, which rely heavily on the mathematical model of the system and it is challenging for it to deal with non-linear systems. It's worth mentioning that it is very sensitive to motor's parameters variations, which represents a challenge for our designed controller.

The fuzzy logic controller with its independence from mathematical model is becoming increasingly important in the field of electric machines, and it has become a valid replacement for the conventional PI controller. However, because of its operational needs, which depend on linguistic rules, its design characteristics provides a challenge to specialist.

The objective of this work is to optimize the fuzzy logic controller using meta-heuristic algorithms in the scope of controlling of the induction motor using Indirect Rotor Fied oriented control (IRFOC).

Meta-heuristic optimization algorithms, represented in this case by Particle Swarm Optimization (PSO) and Grey Wolf Optimization (GWO), are employed to automate the design of FLCs, leading to improved performance and robustness. This optimization approach involves defining a fitness function that represents the desired control performance and then using the metaheuristic algorithm to search for the optimal FLC parameters.

The simulation results have shown that FLC-PI performances are superior to those of conventional PI. Furthermore, they demonstrate the effectiveness of combining FLC-PI with meta-heuristic optimization for induction motor control drives. The proposed approach outperforms traditional PI controllers in terms of reference tracking accuracy, even at low speeds or reversed direction the tracking is always fast and satisfactory. In addition to maintaining a balance between transient response and steady-state errors. At the end, we tested its robustness to parameter uncertainties.

Future Work

This research effectively showed that Fuzzy Logic Control (FLC) combined with metaheuristic optimization offers robust and high-performance control for induction motors using IRFOC. There's significant potential for further exploration. This section proposes promising future research directions to expand the capabilities and applications of FLC with metaheuristic optimization in induction motor control.

- **Diverse Motor Types:** Investigate the effectiveness of FLC with metaheuristic optimization for controlling different types of electric motors, such as synchronous motors, permanent magnet synchronous motors (PMSMs), or stepper motors. Analyze how the control strategies and optimization techniques might need to be adapted for each motor type.
- **Real-World Implementation:** Future work can focus on translating the research findings into practical implementations by developing FLC-based control systems for induction motors operating in real-world industrial environments. This would involve addressing hardware integration challenges, and real-time control considerations.
- **Hybrid Optimization Approaches:** Exploring the potential of combining multiple metaheuristic algorithms could leverage the strengths of each individual approach. This could involve a two-stage optimization process where one algorithm performs a coarse search, followed by another for fine-tuning. Investigating such hybrid approaches could potentially lead to improved optimization results for FLC design.

Appendices

Appendix A

Motor's Parameters

Table A.1: Induction Motor Parameters

Parameter	**Value**
Power	1500 W
Pole Pairs	2
Stator Resistance (R_s)	5.26 Ω
Rotor Resistance (R_r)	4.4947 Ω
Stator inductance (L_s)	0.37632 H
Rotor inductance (L_r)	0.35912 H
Mutual inductance (L_m)	0.35444 H
Inertia torque (J)	0.0067217 kg/m^2
Friction coeffecient (F)	0.016107 N.m.s/rad

Appendix B

Analytical Tuning of PI Controller

In classical control of IRFOC, PI controllers are widely used but they need to be tuned to operate well and give good control of the system. The conventional method of tuning PI is very effective with linear systems. However, with non-linear systems as in the case of IRFOC, most likely it can not converge to the right tuning but it may help sometimes. For this reason, it is necessary to start with the conventional tuning before trials and errors method. In our system, we have 3 PI controllers to regulate the speed w_r and the two stator currents components i_{sd} and i_{sq}.

B.1 Speed Controller

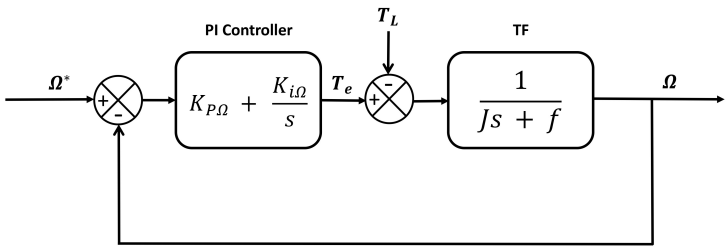


Figure B.1: Speed control loop

Let $K_{p\Omega}$ be K_p and $K_{i\Omega}$ be K_i
By considering the T_L=0 so the open loop transfer function is :

$$G_{OL} = (K_P + \frac{K_i}{s}) * \frac{1}{Js + f} \tag{B.1}$$

Thus, the closed loop transfer unction will be :

$$G_{CL} = \frac{G_{OL}}{1 + G_{OL}} = \frac{K_P s + K_i}{s^2 + \frac{K_P + f}{J} s + \frac{K_i}{J}} \tag{B.2}$$

We use a first order reference filter of the equation (B.3) to eliminate the overshoot.

$$TF_{Filter} = \frac{1}{\frac{K_p}{K_i} s + 1} \tag{B.3}$$

So the system will be reduced to :

$$G_{CL} = \frac{1}{s^2 + \frac{K_P+f}{J}s + \frac{K_i}{J}} \tag{B.4}$$

As one can see, the second-order differential equation represents the motor's speed. Its closed loop transfer function is characterized as follows :

$$TF = \frac{1}{s^2 + 2\zeta w_n s + w_n^2} \tag{B.5}$$

where : ζ is the damping coefficient and w_n is the natural frequency.
By identification of (B.4) with (B.5) we find:

$$\begin{cases} \frac{K_P+f}{J} = 2\zeta w_n \\ \frac{K_i}{J} = w_n^2 \end{cases} \tag{B.6}$$

Let's consider $\zeta = 1$ and $w_n = 8$, this leads to : $K_i = 0.43$ and K_p= 0.0914

B.2 Current Controllers

The torque and the flux requires maintaining the loops of direct (d) and quadratic (q) statoric currents under control. To do the implementation of the controller, we will use the system of static equations derived from the induction motor model. We have two currents loops i_{sd} and i_{sq}, which are identical in their plants so, the values we get for i_{sd} will be applied on i_{sq}.

B.2.1 Current i_{sd} Controller

The current regulator i_{sd} provids the reference voltage V_{sd}^*. The control loop is shown in the (figure B.2)

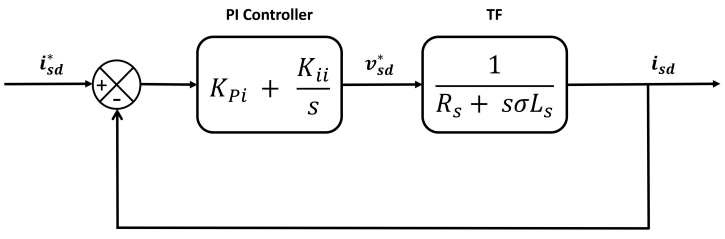


Figure B.2: Current i_{sd} control loop

$$G_{OL} = (K_{Pi} + \frac{K_{ii}}{s}) * \frac{1}{L_s\sigma s + R_s} = K_{Pi}(s + \frac{K_{ii}}{K_{Pi}})\frac{1}{s}\frac{\frac{1}{\sigma L_s}}{\frac{R_s}{\sigma L_s} + s} \tag{B.7}$$

By applying zero pole cancellation such that :

$$\frac{Kii}{KPi} = \frac{R_s}{\sigma L_s} \tag{B.8}$$

we find:

$$G_{OL} = \frac{K_{Pi}}{s} * \frac{1}{\sigma L_s} \tag{B.9}$$

Thus, the closed loop transfer unction will be :

$$G_{CL} = \frac{G_{OL}}{1 + G_{OL}} = \frac{\frac{K_{Pi}}{s\sigma L_s}}{\frac{KPi}{s\sigma L_s} + 1} = \frac{1}{\frac{\sigma L_s}{K_{Pi}s+1}} = \frac{1}{\tau s + 1} \tag{B.10}$$

$$With : \tau = \frac{\sigma L_s}{K_{Pi}} \tag{B.11}$$

From motor parameters $\sigma = 1$ - $\frac{L_m^2}{Lr*Ls} = 0.0704$

Let's take : τ=0.001s; from equation (B.8) and (B.11) we find :

$$\begin{cases} K_{Pi} = \frac{\sigma L_s}{\tau} = 26.4929 \\ K_{ii} = K_{Pi}\frac{R_s}{\sigma L_s} = 5259.9944 \end{cases} \tag{B.12}$$

B.2.2 Current i_{sq} Controller

The i_{sq} current loop regulation scheme is shown on the (figure B.3), it is noted that it is the same as for the current i_{sd}. Using the same procedures as for the i_{sd} regulator, the coefficients of the current regulator i_{sq} that provides the reference voltage V_{sq}^* will be determined.

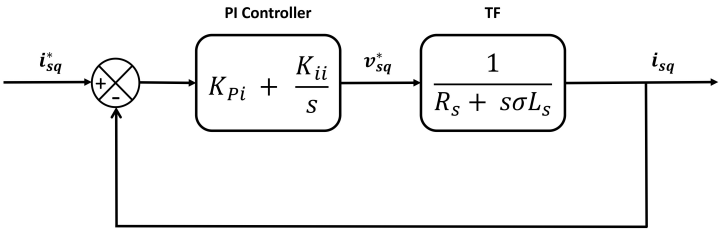


Figure B.3: Current i_{sq} control loop

Bibliography

[1] Rajesh Kumar, RA Gupta, and SV Bhangale. Indirect vector-controlled induction motor drive with fuzzy logic based intelligent controller. 2007.

[2] Alaa Tahhan and Feyzullah Temurtaş. Enhanced fuzzy logic control model and sliding mode based on field oriented control of induction motor. *World Journal of Engineering and Technology*, 12(01):65–79, 2024.

[3] Zulhisyam Salleh, Marizan Sulaiman, Rosli Omar, and Fizatul Aini Patakor. Optimization of fuzzy logic based for vector control induction motor drives. In *2016 8th Computer Science and Electronic Engineering (CEEC)*, pages 83–88. IEEE, 2016.

[4] Biranchi Narayan Kar, KB Mohanty, and Madhu Singh. Indirect vector control of induction motor using fuzzy logic controller. In *2011 10th International Conference on Environment and Electrical Engineering*, pages 1–4. IEEE, 2011.

[5] Prof. Rui Esteves Araújo. Induction motors, Nov 2012.

[6] Pr.Kheldoun Aissa. *Machine and Drives. University course.*

[7] Hamid A Toliyat. *DSP-based electromechanical motion control.* CRC Press, 2004.

[8] Moshekwa Malatji. Derivation and implementation of a dq model of an induction machine using matlab/simulink*, 03 2020.

[9] Aissa Khaldoun. Commande vectorielle d'un moteur asynchrone à cage avec adaptation par logique floue de la résistance rotorique et minimisation des pertes totales, 2001.

[10] Venu Gopal BT. Comparison between direct and indirect field oriented control of induction motor. *Int. J. Eng. Trends Technol*, 43(6):364–369, 2017.

[11] Kwang Hee Nam. *AC motor control and electrical vehicle applications.* CRC press, 2018.

[12] Abdelkader Mechernene, Mokhtar Zerikat, Nordine Benouzza, Soufyane Chekroun, and N Benharir. Contrôle sans capteur de vitesse d'un moteur asynchrone basé sur l'approche du mras-mutuel avec flux rotorique orienté.

[13] Kim Sang-Hoon. *Electric Motor Control.* Elsevier Science Technology, 2017.

[14] Rayane Ghernaout. Identification and control of asynchronous motor using metaheuristic algorithms, 2022.

[15] Lotfi Baghli. *Contribution à la commande de la machine asynchrone, utilisation de la logique floue, des réseaux de neurones et des algorithmes génétiques*. PhD thesis, Université Henri Poincaré-Nancy I, 1999.

[16] Lotfi Asker Zadeh. Fuzzy sets. *Information and control*, 8(3):338–353, 1965.

[17] HAMDOUD Hamza. Commande vectorielle par la logique floue de la machine triphasé asynchrone, 2012.

[18] Atallah Mohamed Djemoui Guediri Abdelhafid, Guediri Mourad. Commande par logique floue appliquée à la machine asynchrone, 2019.

[19] Mohammed Blej and Mostafa Azizi. Comparison of mamdani-type and sugeno-type fuzzy inference systems for fuzzy real time scheduling. *International Journal of Applied Engineering Research*, 11(22):11071–11075, 2016.

[20] Ashish Kushwaha and Madan Gopal. Fuzzy logic controller performance in vector control of induction machine. In *2017 International Conference on Technological Advancements in Power and Energy (TAP Energy)*, pages 1–5. IEEE, 2017.

[21] Jeydson Lopes Da Silva. Fuzzy logic control with pso tuning. *Fuzzy Systems*, page 185, 2022.

[22] Debasis Samanta. Defuzzification methods. *Indian Institute of Technology Kharagpur[online]. Dostupné z: http://cse. iitkgp. ac. in/~ dsamanta/courses/archive/sca/Archives/C hapter% 205% 20Defuzzification% 20Methods. pdf [Accessed 31 Jul. 2019]*, 2019.

[23] N Mogharreban and LF Dilalla. Comparison of defuzzification techniques for analysis of non-interval data. In *NAFIPS 2006-2006 Annual Meeting of the North American Fuzzy Information Processing Society*, pages 257–260. IEEE, 2006.

[24] S Bhattacharya, A Chatterjee, and S Munshi. An improved pid-type fuzzy controller employing individual fuzzy p, fuzzy i and fuzzy d controllers. *Transactions of the Institute of Measurement and Control*, 25(4):352–372, 2003.

[25] J Immanuel, CS Parvathi, L Shrimanth Sudheer, and P Bhaskar. Matlab graphical user interface based fuzzy logic controllers for liquid level control system. *Sensors & Transducers*, 148(1):52, 2013.

[26] MHN Talib, Z Ibrahim, N Abd Rahim, R Zulhani, N Nordin, Nabil Farah, and AM Razali. An improved simplified rules fuzzy logic speed controller method applied for induction motor drive. *ISA transactions*, 105:230–239, 2020.

[27] Gai-Ge Wang, Xiaoqi Zhao, and Keqin Li. Metaheuristic algorithms. Not available:Not available, FEB 2024.

[28] Aleksandar Lazinica. *Particle swarm optimization*. InTech, 2009.

[29] MAURICE CLERC. *PARTICLE SWARM OPTIMIZATION*. ISTE, 2006.

[30] Omid Bozorg-Haddad. *Advanced Optimization by Nature-Inspired Algorithms*. Springer, Jul 2017.