People's Democratic Republic of Algeria
Ministry of Higher Education and Scientific Research

University M'Hamed BOUGARA – Boumerdes

Institute of Electrical and Electronic Engineering
Department of Electronics

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

MASTER

In **Control**

Option: **Control**

Title:

Fuzzy-PID Speed Controller for an Induction Motor.

Presented by:

- BOUGHELOUM Dhya Eddine

- BENYAHIA Sofiane

Supervisor:

Dr. BOUSHAKI Razika

Registration Number:2019/2020

ACKNOWLEDGMENT

We would like to thank and appreciate the effort which was done by our supervisor Dr. Razika BOUSHAKI and the Control Engineering department at the Institute of Electrical and Electronic Engineering for giving us the privilege and the opportunity to do this project. Dr. BOUSHAKI’s expertise, understanding, and patience have strongly contributed to our project.

We would also like to thank our families for their support, motivation and help. Without the support and encouragement of our families and friends we would never have finished our degree.

Lastly, we would like to thank all the teachers and professors who have worked unconditionally to give us a high-quality education during our study.

DEDICATION

I, Dhya Eddine Bougheloum, would like to dedicate my work to:

My father, Abdelali, my beloved mother, my brothers, my friends and my supportive teachers,

Every person that helped me and supported me,

Every person who made my days at the institute enjoyable,

Every person who shared with me this journey,

Dr. Dalila CHERIFI who gave me her time, support, and knowledge,

Without all your help, support, and patience, I could not have done this effort.

I, Sofiane Benyahia, would like to dedicate this work to my supportive parents and family, my friends, my teachers and anyone who ever shared their knowledge with me.

ABSTRACT

The main objective of vector control or field-oriented control (FOC) control is to have decoupled control of flux and torque in three phase induction motors. FOC rotates the stationary stator reference frame into rotating reference frame attached to the rotor flux linkage space phasor which results in a decomposition of stator currents into torque and flux producing components under orthogonality. This will give fast dynamic response as compares to other scalar drives i.e. variable frequency drive (V/F).

This project presents a Fuzzy-PID control system for the speed control of a three-phase squirrel cage induction motor. The proposed method uses both Fuzzy logic and conventional controllers along with vector control technique. This method combines the advantages of the fuzzy logic controller and conventional controllers to improve the speed response of the induction motor. The FLC observes the closed loop error signal and then controls the PID input error signal so that the actual speed matches the reference speed with reduced rise time, settling time, and peak over shoot.

Implementation and simulation results using MATLAB of multiple controllers such as (PID, Fuzzy, and Fuzzy-PID) are compared along with conventional PI controller in terms of some performance measurements such as rise time (tr), maximum percent overshoot (Mp), settling time (ts), and steady state error (Ess) at various load conditions. The results of the simulation verified the effectiveness of the proposed speed controller model under different operating conditions and demonstrated improvements in performance in speed tracking and system's stability.

Keywords: Induction motor, vector control, Park and Clark transformation, fuzzy logic, PID controller.

TABLE OF CONTENTS

CHAPTER III: Field oriented control of induction motor

CHAPTER IV: Control of the induction machine

CHAPTER V: SIMULINK model and results

LISTE OF TABLES

LISTE OF FIGURES

Chapter I: General introduction

1.1. Introduction

Induction motors, particularly the squirrel cage induction motors (SCIM) as shown in figure 1.1 have been widely used in industry application such as hybrid vehicles, paper and textile mills, robotics, and wind generation systems because of their several inherent advantages such as their simple construction, robustness, reliability, low cost, and low maintenance needs. Without proper controlling, it is virtually impossible to achieve the desired task for any industrial application [1].

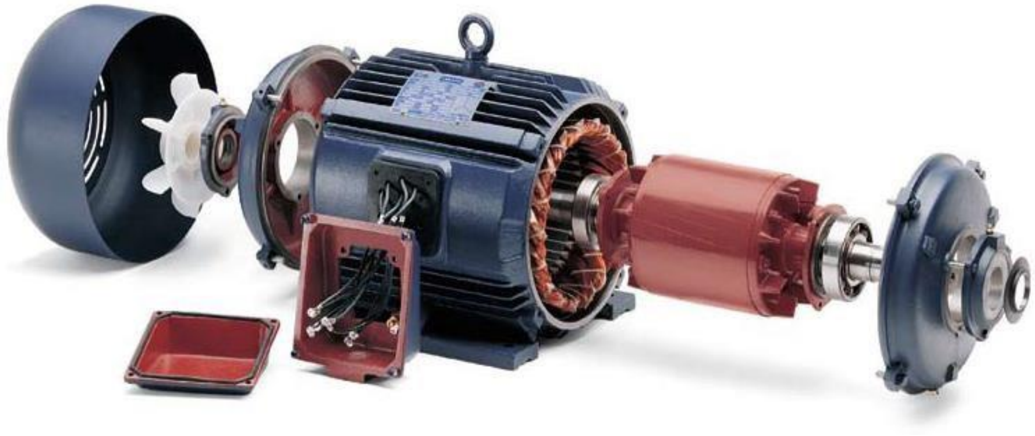


Figure 1.1: Three phase squirrel cage induction motor

Open loop control of induction motor (IM) with variable frequency and variable voltage amplitude provide a satisfactory variable speed motor for steady torque operation and without stringent requirements on speed regulation. However, for high performance drive requirements including fast dynamic response, accurate speed, and accurate torque control of induction motor is a challenging problem due to their highly coupled nonlinear structure and many of the parameters vary with the operating conditions such as load torque, reference speed set point, rotor resistance, and motor temperature [2].

To achieve optimal efficiency of induction motors, several control techniques have been developed to control the induction motor such as scalar control, vector or field oriented control, direct torque control. Scalar control is one of the first control techniques of induction motors. In this method the ratio of both the amplitude and frequency of the supply voltage is kept constant in order to maintain a constant air gap flux and hence provide maximum torque. Scalar control drives are easy to implement but does not yield satisfactory results for high performance applications because of inherent coupling effects between torque and flux give sluggish response and system is easily prone to instability. This problem can be solved by field oriented control or direct torque control. In most of industrial drive control applications, the standard method to control induction motor is based on the field oriented or vector control principle in order to achieve the best dynamic behavior. In this method the decoupling between the flux and torque allows the induction motor to be controlled in a similar method to that in the control of separately exited dc motors. Therefore, it can be used for high performance applications [3].

Over the years, the conventional control such as the proportional plus integral (PI), and proportional plus integral plus derivative (PID) controllers have been used together with vector control methods to better control the speed of induction motors. However, it must be pointed out that conventional controllers have major drawback such as performance sensitivity to variations in system's parameters, and the fact that when using fixed gains, the controller may not provide the required speed performance under variations in the motor parameters and operating conditions. In order to overcome these challenges, Fuzzy Logic controller (FLC) has been used for motor speed control [4].

The main advantage of fuzzy logic controller when compared to the conventional controller is that no mathematical model is required for the controller design. Fuzzy logic has been successfully used to control ill-known or complex systems where precise modeling is difficult or impossible [5]. It has been demonstrated that dynamic performance of electric drives as well as robustness with respect to parameter variations can be improved by adopting the nonlinear speed control techniques as in the ones fuzzy control provides [6] and [7]. Recently, hybrid control techniques based on combination of two or more control methods are proposed to enhance controller's performance. Our proposed method combines conventional controller with fuzzy logic controller and vector control technique to take advantage of the best attributes of both controllers and eliminate the drawbacks of conventional controller such as oscillation, overshoot, and undershoot and the drawback of FLC such as steady state error.

1.2. Problem statement

Induction motors are most widely used in all industries. The speed of the induction motor has to be varied according to application requirement. Therefore, there is a need for more efficient and reliable induction motor drive systems for current and future applications. Induction motors are highly non-linear systems, having uncertain time varying parameters mainly rotor resistance and subjected to unknown load disturbance. In addition, the rotor flux is inaccessible for state feedback control. Taking these difficulties into account, various control strategies are proposed such as vector control method, direct torque control, and sliding mode control. The use of proportional plus integral (PI) controller for speed control of induction motor with above mentioned techniques gives reasonable performance in steady state operation. However, this performance characterized by an overshoot, slow transient response, steady state error and the main drawbacks of this kind of controllers is its sensitivity to variation in system's parameters and the fact that when using fixed gains, the controller may not provide the required speed performance under variations in the motor load torque and operating conditions. In order to overcome these challenges, Fuzzy-PID Logic controller has been proposed for induction motor speed control [2], [3], [8], and [9].

1.3. Scope and objectives

The main objective of this work is to design a control method to provide optimal dynamic response of squirrel cage induction motor. This will be achieved by incorporate fuzzy logic with conventional controllers and utilization of vector control technique. Implementation and simulation results using MATLAB/SIMULINK of various system controllers such as (PID, Fuzzy, and Fuzzy-PID) are analyzed and compared along with conventional PI controller in terms of several performance measurements such as rise time (tr), overshoot (Mp), settling time (ts) and steady state error at various load conditions.

1.4. Project outline

This project is broke down into five chapters. Chapter II talks about the modeling of the induction motor. Additionally, a brief description of space vector projection in stationary and rotating reference frame and the transformation between them. Chapter III gives an introduction, principle, and detailed mathematical model of field oriented control. The background and structure of conventional controllers and fuzzy controllers are presented in chapter IV. The SIMULINK model is presented and simulation results for the proposed controller techniques are analyzed and compared under different operating conditions in chapter V. Finally, the conclusion and recommendations for future work.

Chapter II: Induction motor modeling

2.1. Induction motor basics

Three phase induction motors are rugged, cheap to produce and easy to maintain. They can run at a nearly constant speed from zero to full load. The design of an induction motor is relatively simple and consists of two main parts, a stationary stator and a rotating rotor. There are two main classes of the induction motor differing in the way their rotors are wound: the wound induction motor and the squirrel cage induction motor.

The motor discussed in this project is a three phase squirrel cage induction motor. The rotor of a squirrel cage induction motor consists of aluminum bars which are short circuited by connecting them to two end rings so that rotor generates the induction current and magnetic field by itself. This makes the AC induction motor a robust, rugged and inexpensive candidate for motor drive systems [8].

The structure of a squirrel cage induction motor is shown in Figure 2.1. In an induction motor, the alternating currents feed from three phase terminals and flow through the stator windings, producing a rotating stator flux in the motor [8]. The rotating speed of this magnetic field is defined as synchronous speed, and related to the number of poles of the induction motor and the frequency of power source.

$$n_{sync} = \frac{120 f_e}{P} \quad rpm \tag{2.1}$$

where f_e is the power source frequency, P is the number of poles and n_{sync} is the synchronous speed in revolutions per minute.

The rotating magnetic field from the stator will induce a voltage in the rotor bars, since the rotor bars are short-circuited, a large circulating current will be generated in the rotor bars. This induced rotor current will then interact with the rotating magnetic field. Because of Lorentz's law, a tangential electromagnetic force will be generated on the rotor bars, and the sum of forces on each rotor bar produces a torque that eventually drives the rotor in the direction of the rotating field.

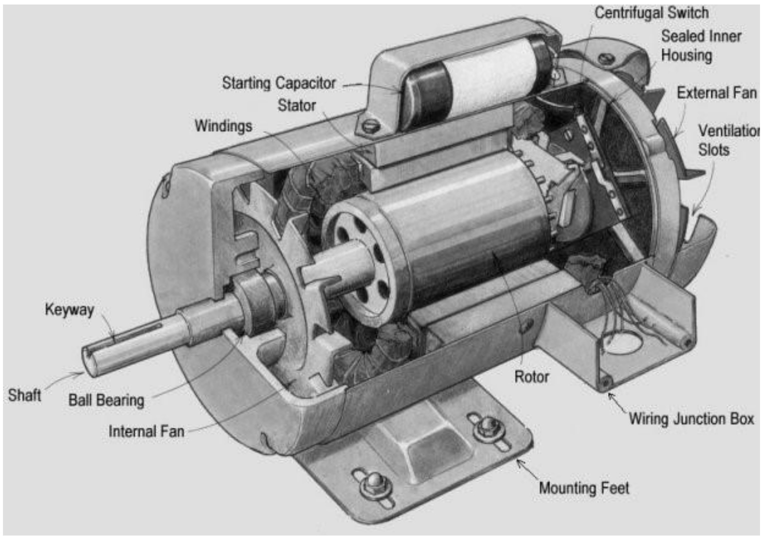


Figure 2.1: Squirrel cage induction motor cross section

When the rotating magnetic field is first generated, the rotor is still in its rest condition. However, the rotor will accelerate rapidly in order to keep up with the rotating stator flux. As the rotor speed increases, the rotor bars are not cut as much by the rotating field, so the voltage in the rotor bars decreases. If the rotor speed equals to the flux speed, the rotor bars will no longer be cut by the field and the rotor will start to slow down [8]. This is why induction motors are also called asynchronous motors because the rotor speed will never equal the synchronous speed. The difference between the stator and rotor speed is defined as the slip speed:

$$n_{slip} = n_{sync} - n_m \tag{2.2}$$

where n_{slip} is the slip speed

n_{sync} is the speed of the rotating magnetic field

n_m is the mechanical shaft speed of the motor

Also, a slip ratio can be defined as:

$$s = \frac{n_{sync} - n_m}{n_{sync}} \tag{2.3}$$

Notice that, if the rotor runs at synchronous speed: s = 0

if the rotor stops moving: s = 1

2.2. Space vectors

By using space vectors in the induction motor modeling, all the complex state variables can be efficiently defined [9]. Variables such as the three phase voltages, currents and fluxes of induction motors can be analyzed and described easily and conveniently.

The three phase axes are defined by the vectors: e^{j0° , e^{j120° and e^{j240° The stator windings and stator current space vector in the complex plane are shown in Figure 2.2 below.

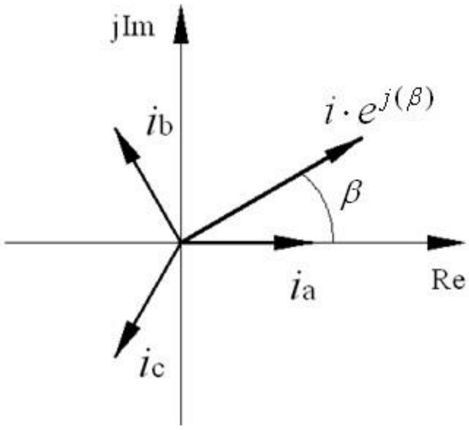


Figure 2.2: Current space vectors

The space vector of the stator current i_s can be described by:

$$i_s = i_{as} \cdot e^{j0^\circ} + i_{bs} \cdot e^{j120^\circ} + i_{cs} \cdot e^{j240^\circ} \tag{2.4}$$

where subscript s refers to the stator of the induction motor, a, b, and c are the three phase axes.

Furthermore, the rotor current can be described by:

$$i_r = i_{ar} \cdot e^{j0^\circ} + i_{br} \cdot e^{j120^\circ} + i_{cr} \cdot e^{j240^\circ} \tag{2.5}$$

where subscript r refers to the rotor of the induction motor.

2.3. The coordinate transformation of space vectors

The modeling, analysis and control design of induction can be significantly simplified by using coordinate transformations. A three-phase variable can be transferred into a two-phase variable; also a stationary variable can be transferred into a rotational one. This transformation usually includes the following two steps:

- The Clarke transformation
- The Park transformation

2.3.1. Forward and inverse Clarke transformation

The Clarke transformation converts a three-phase signals such as currents, voltage, and flux from three-phase coordinate system (a, b, c) into a two-phase coordinate orthogonal system (d^s, q^s). Figure 2.3 shows the graphical construction of the current space vector and its projection into stator reference frame (d^s, q^s). Where, the real part of the current space vector is equal to instantaneous value of the direct-axis current componentI_d^s and imaginary part is equal to the quadrature-axis current component I_q^s [10] and [11].

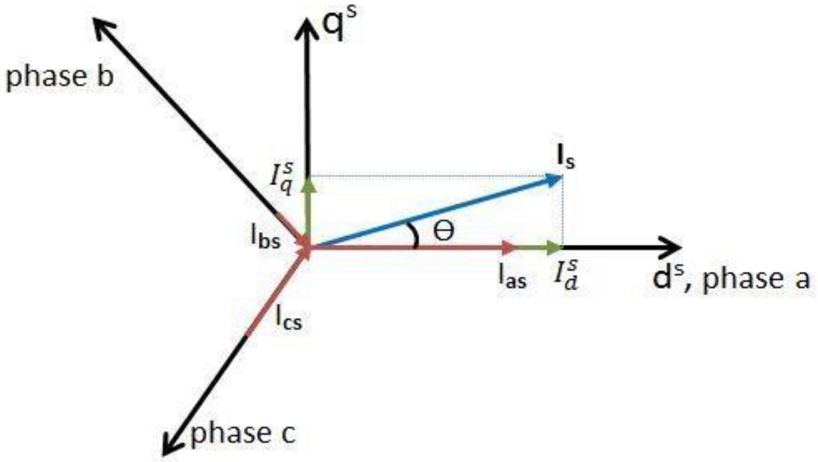


Figure 2.3: Stator current space vector and its component in the stationary reference frame [11]

Assuming that the a-axis and d^s-axis are in the same direction, the quadrature phase stator current components I_d^s and I_q^scan be expressed in term of the instantaneous values of the actual phase currents as:

$$\begin{bmatrix} I_d^s \\ I_q^s \end{bmatrix} = \begin{bmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & \frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} \end{bmatrix} \begin{bmatrix} I_{as} \\ I_{bs} \\ I_{cs} \end{bmatrix} \tag{2.6}$$

Where:

I_{as}, I_{bs}, and I_{cs} are the actual stator currents. I_d^s , I_q^s are the direct and quadrature currents respectively [11].

The quadrature-phase stator current components I_d^s and I_q^s can be easily converted from a two-phase coordinate orthogonal system (d^s, q^s) into a three-phase coordinate system (a, b, c) by using the inverse Clarke transformation. The matrix equation is expressed as [10]:

$$\begin{bmatrix} I_{as} \\ I_{bs} \\ I_{cs} \end{bmatrix} = \frac{2}{3} \begin{bmatrix} 1 & 0 \\ -\frac{1}{2} & \frac{\sqrt{3}}{2} \\ -\frac{1}{2} & -\frac{\sqrt{3}}{2} \end{bmatrix} \begin{bmatrix} I_d^s \\ I_q^s \end{bmatrix} \tag{2.7}$$

2.3.2. Forward and inverse Park transformation

Park transformation is the most important transformation in the field oriented control. It converts a two-phase orthogonal system (d^s, q^s) into rotating reference frame (d^e, q^e) [12]. Current space vector equations can be formulated in a rotating reference frame that rotates at a synchronous speed ω in the same direction as does the stator current space vector. As a result, in the steady state, coordinates of the stator current components in the new reference do not vary in time [10] and [11].

In rotating reference frame d^e - q^e the direct and quadrature axes (d^e and q^e) rotate at synchronous speed ω, as shown in figure 2.4, where Θ_e is the angle between the d^s axis of the stationary reference frame and the d^e axis of the rotating reference frame [11]. If the d^e axis is aligned with the rotor flux vector, the transformation is expressed as

$$\begin{bmatrix} I_d^e \\ I_q^e \end{bmatrix} = \begin{bmatrix} \cos\Theta_e & \sin\Theta_e \\ -\sin\Theta_e & \cos\Theta_e \end{bmatrix} \begin{bmatrix} I_d^s \\ I_q^s \end{bmatrix} \tag{2.8}$$

Where:

I_d^s, I_q^s are the direct and quadrature currents respectively in stationary reference frame,

I_d^e, I_q^e are the direct and quadrature currents respectively in rotating reference frame [11].

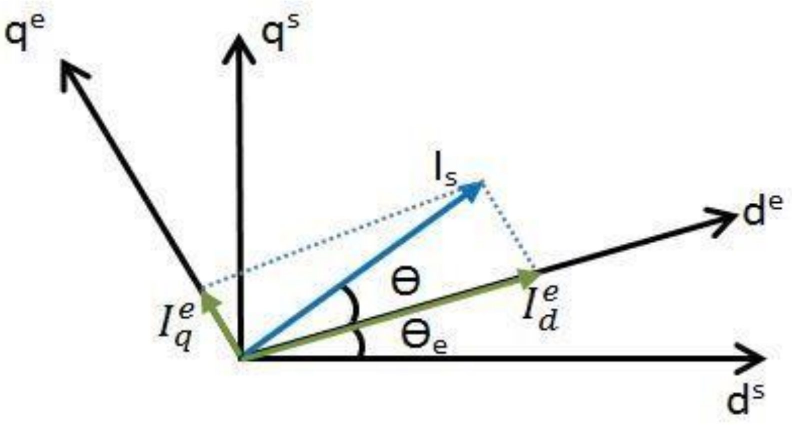


Figure 2.4: Stator current space vector and its component in the rotating reference frame [11]

While the inverse transformation from rotating reference frame to stationary reference frame is expressed as

$$\begin{bmatrix} I_d^S \\ I_q^S \end{bmatrix} = \begin{bmatrix} \cos\Theta_e & -\sin\Theta_e \\ \sin\Theta_e & \cos\Theta_e \end{bmatrix} \begin{bmatrix} I_d^e \\ I_q^e \end{bmatrix} \tag{2.9}$$

2.4. Modeling equations

The induction motor modeling usually contains three parts: flux equations, voltage equations and torque equations [9].

2.4.1. Flux equations

There are three kinds of fluxes: stator flux, rotor flux and mutual flux. The stator flux and rotor flux vectors can be expressed in terms of stator current i_s and rotor current i_r, given below:

$$\Psi_s = L_s I_s + L_m I_r \tag{2.10}$$

$$\Psi_r = L_r I_r + L_m I_s \tag{2.11}$$

where L_s and L_r are the stator inductance and the rotor inductance, respectively; L_m is the mutual inductance between the stator and rotor windings.

2.4.2. Voltage equations

As we mentioned above, the voltage equations of induction motors can be described in different coordinate frames. Three kinds of coordinate frames are commonly used to describe the induction motors [9]. They are:

- Stationary frame;
- Synchronous rotating frame;
- Field orientation synchronous frame.

2.4.2.1. Voltage equations in stationary frame

The induction motor usually has three sets of stator windings, and the rotor can be considered as three sets of windings as well [9]. Both stator and rotor can be represented by inductance and resistance in an equivalent-circuit. Using the Clarke transformation, the three-phase induction motor model can be expressed as an equivalent two-phase model [9]. The equivalent circuits in the stationary frame of the induction motor are shown in Figure 2.5 and 2.6:

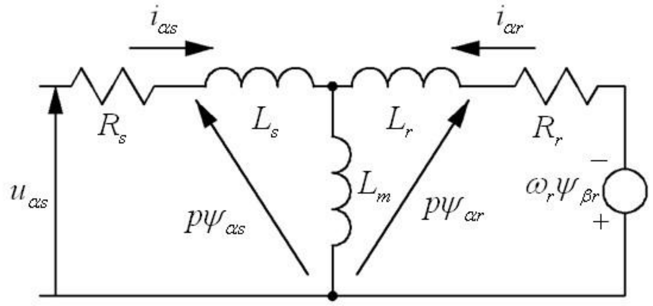


Figure 2.5: Alpha component of the induction motor equivalent-circuit

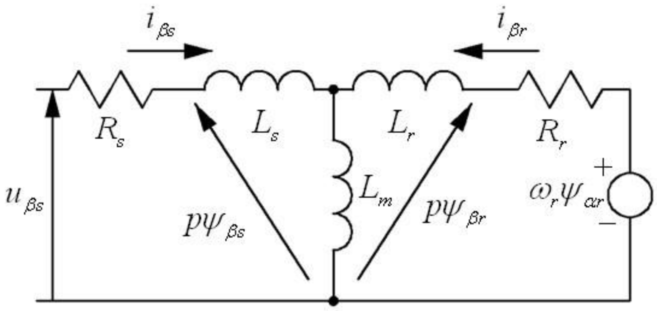


Figure 2.6: Beta component of the induction motor equivalent-circuit

Based on the Kirchhoff's voltage law (KVL), the stator and rotor voltage equations can be expressed as:

$$u_{\alpha s} = i_{\alpha s}R_s + p\psi_{\alpha s} \tag{2.12}$$

$$u_{\beta s} = i_{\beta s}R_s + p\psi_{\beta s} \tag{2.13}$$

$$0 = i_{\alpha r}R_r + p\psi_{\alpha r} + \omega_r\psi_{\beta r} \tag{2.14}$$

$$0 = i_{\beta r}R_r + p\psi_{\beta r} - \omega_r\psi_{\alpha r} \tag{2.15}$$

where ω_r $\psi_{\beta r}$ and ω_r $\psi_{\alpha r}$ are rotating velocity emf and p is the differential operator.

Substituting eq. 2.10 and 2.11 into eq. 2.12 and 2.13 respectively, the stator voltage equations can be rewritten:

$$u_{\alpha s} = (R_s + L_s p)i_{\alpha s} + L_m p i_{\alpha r} \tag{2.16}$$

$$u_{\beta s} = (R_s + L_s p)i_{\beta s} + L_m p i_{\beta r} \tag{2.17}$$

Also, the rotor voltage equations can be rewritten as:

$$0 = L_m p i_{\alpha s} + (R_r + i_r p)i_{\alpha r} + \omega_r L_m i_{\beta s} + \omega_r L_r i_{\beta r} \tag{2.18}$$

$$0 = L_m p i_{\beta s} + (R_r + i_r p)i_{\beta r} - \omega_r L_m i_{\alpha s} - \omega_r L_r i_{\alpha r} \tag{2.19}$$

The relationship of voltages and currents can be given in matrix form from the four voltage equations:

$$\begin{pmatrix} u_{\alpha s} \\ u_{\beta s} \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} (R_s + L_s p) & 0 & L_m p & 0 \\ 0 & (R_s + L_s p) & 0 & L_m p \\ L_m p & \omega_r L_m & (R_r + L_r p) & L_r \omega_r \\ -L_m \omega_r & L_m p & -L_r \omega_r & (R_r + L_r p) \end{pmatrix} \cdot \begin{pmatrix} i_{\alpha s} \\ i_{\beta s} \\ i_{\alpha r} \\ i_{\beta r} \end{pmatrix} \tag{2.20}$$

where ω_r is rotor speed.

2.4.2.2. Voltage equations in synchronous rotating frame

This is a rotary frame and the rotating speed is ω_e . Using the Park transformation, eq. 3.12, and 3.13 can be expressed in the d-q synchronous rotating reference frame, so the stator voltage equations can be rewritten as:

$$u_{ds} = i_{ds}R_s + p\psi_{ds} - \omega_e\psi_{qs} \tag{2.21}$$

$$u_{qs} = i_{qs}R_s + p\psi_{qs} + \omega_e\psi_{ds} \tag{2.22}$$

The last terms in eq. 2.21 and eq. 2.22 can be defined as speed emf as they are directly related with the synchronous speed of the motor. When $\omega_e = 0$, these equations turn back to stationary forms.

Similarly, the rotor voltage equations can be derived from eq. 2.14 and eq. 2.15. At this time, the rotor speed is ω_r, and since the d-q axes are fixed with the rotor, the relative speed with respect of synchronously rotating frame is $\omega_e - \omega_r$. Therefore, in the synchronous rotating frame, the rotor voltage equations should be rewritten as:

$$0 = i_{dr}R_r + p\psi_{dr} + (\omega_r - \omega_e)\psi_{qr} \tag{2.23}$$

$$0 = i_{qr}R_r + p\psi_{qr} - (\omega_r - \omega_e)\psi_{dr} \tag{2.24}$$

The stator voltage equations can be rewritten as:

$$u_{ds} = i_{ds}(R_s + L_s p) - L_s\omega_e i_{qs} - L_m\omega_e i_{dr} + L_m p i_{qr} \tag{2.25}$$

$$u_{qs} = i_{qs}(R_s + L_s p) + L_s\omega_e i_{ds} + L_m\omega_e i_{dr} + L_m p i_{qr} \tag{2.26}$$

The rotor voltage equations can be rewritten as:

$$0 = L_m p i_{ds} - (\omega_e - \omega_r)L_m i_{qs} + (R_r + L_r p)i_{dr} - L_r(\omega_e - \omega_r)i_{qr} \tag{2.27}$$

$$0 = L_m p i_{qs} + (\omega_e - \omega_r)L_m i_{ds} + (R_r + L_r p)i_{qr} + L_r(\omega_e - \omega_r)i_{dr} \tag{2.28}$$

Equations 2.25 – 2.28 also can be expressed in a matrix form as:

$$\begin{pmatrix} u_{ds} \\ u_{qs} \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} (R_s + L_s p) & -L_s\omega_e & L_m p & -L_m\omega_e \\ L_s\omega_e & (R_s + L_s p) & L_m\omega_e & L_m p \\ L_m p & -L_m(\omega_e - \omega_r) & (R_r + L_r p) & -L_r(\omega_e - \omega_r) \\ L_m(\omega_e - \omega_r) & L_m p & L_r(\omega_e - \omega_r) & (R_r + L_r p) \end{pmatrix} \cdot \begin{pmatrix} i_{ds} \\ i_{qs} \\ i_{dr} \\ i_{qr} \end{pmatrix} \tag{2.29}$$

2.4.3. Torque equations

A torque is produced as a result of the interaction of stator magnetic field and rotor magnetic field, so one can have:

$$T_e = K_0 B_r \times B_s \tag{2.30}$$

where T_e is the induced torque and B_r and B_s are the magnetic flux densities of the rotor and the stator respectively.

Also, the electromagnetic torque is produced by the interaction of current and magnetic field. Using two current quantities (stator current and rotor current) and three fluxes (stator flux, mutual flux and rotor flux), the torque can be expressed in six different forms [13]:

$$T_e = K_1 \Psi_s \times I_r \tag{2.31}$$

$$T_e = K_2 \Psi_m \times I_r \tag{2.32}$$

$$T_e = K_3 \Psi_r \times I_r \tag{2.33}$$

$$T_e = K_4 \Psi_s \times I_s \tag{2.34}$$

$$T_e = K_5 \Psi_m \times I_s \tag{2.35}$$

$$T_e = K_6 \Psi_r \times I_s \tag{2.36}$$

Where $K_1 \sim K_6$ are the torque coefficients. In fact, these six expressions are all identical.

Also, the system motion equation is given by:

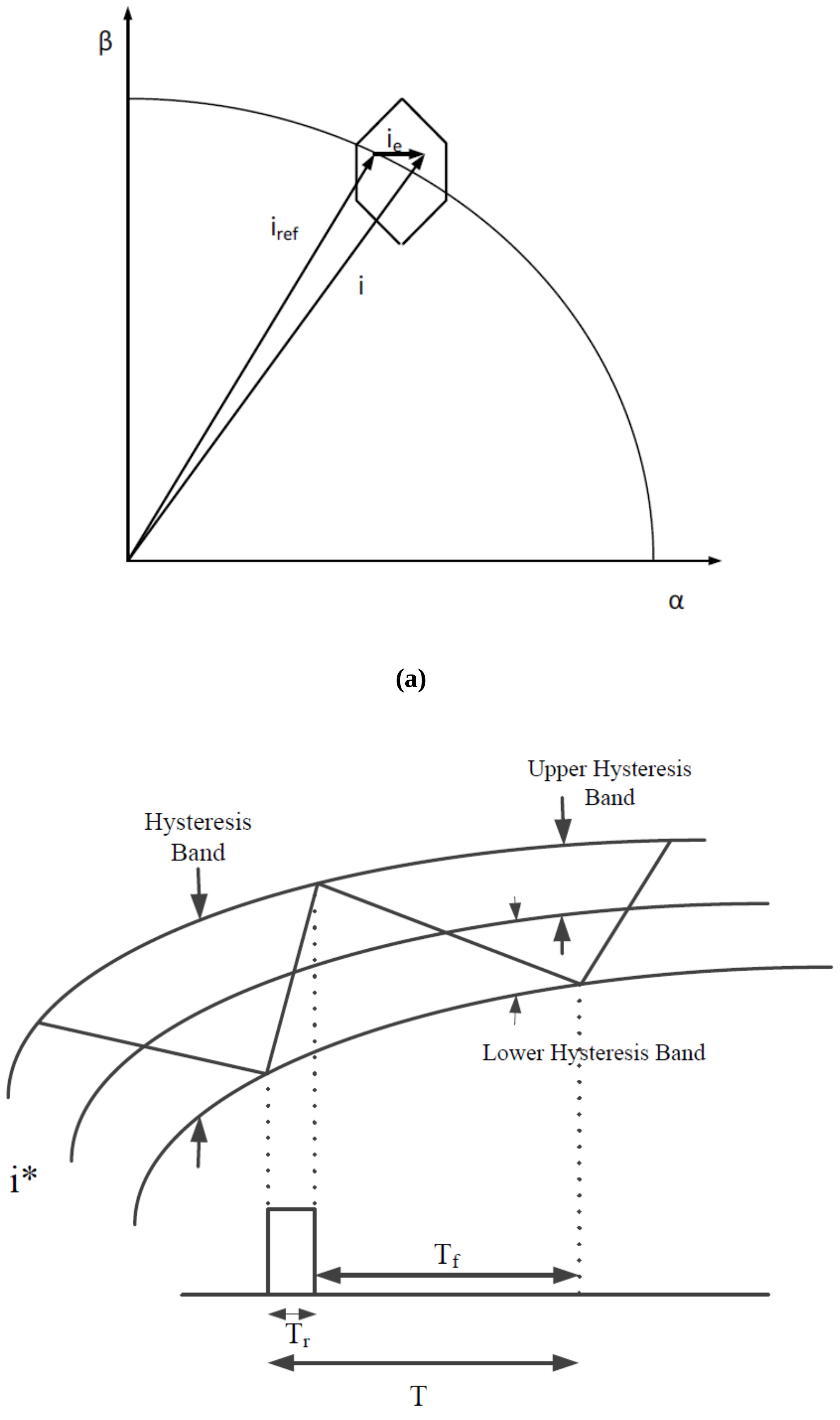
$$T_e = T_L + \frac{J}{n_p} \cdot \frac{d\omega}{dt} \tag{2.37}$$

where T_L is the load torque, ω is the rotor rotating speed, and n_p is the poles numbers, and J is rotor’s moment of inertia.

2.5. Hysteresis-band current controller

It is an instantaneous negative feedback current control method in which the actual current chases the reference current within the hysteresis band. This controller generates the reference current wave of desired frequency and magnitude, and it is compared with the actual wave. In this work, three phase voltage source inverter is used.

Hence, three hysteresis band of width ‘δ’ are defined around their respective phase currents. As these three currents are dependent from each other, hence, the system is transformed to (α,β) co-ordinate axis which is similar to stationary reference frame.



(a)

(b)

Figure 2.7(a),(b): Hysteresis current controller

When three hysteresis bands are transformed into this co-ordinate system, this will result in hysteresis hexagonal area. In steady state, the reference current i_{ref} rotates on circle round the origin as shown in Figure 2.7. The realistic value of current i has to be kept within the area of hexagon. Hence, each time when tip of i touches the border of the surface that is heading out of the hexagon, the inverter has to be switched accordingly in order to keep the current in the area of hexagon.

2.6. Space vector pulse width modulation (SVPWM) technology

Space Vector Pulse Width Modulation (SVPWM) is an algorithm which translates phase voltage references (i.e. phase to neutral) coming from controller into duty cycles that can be applied to PWM peripherals like IGBT bridge or MOSFET bridge.

The circuit model of a typical three-phase voltage source PWM inverter is shown in Figure 2.8. S_1 to S_6 are the six power switches. When an upper transistor is switched on, the corresponding lower transistor must be switched off; otherwise the DC power will be short-circuited. Therefore, the on and off states of the upper transistors S_1, S_3 and S_5 can be used to determine the output voltage.

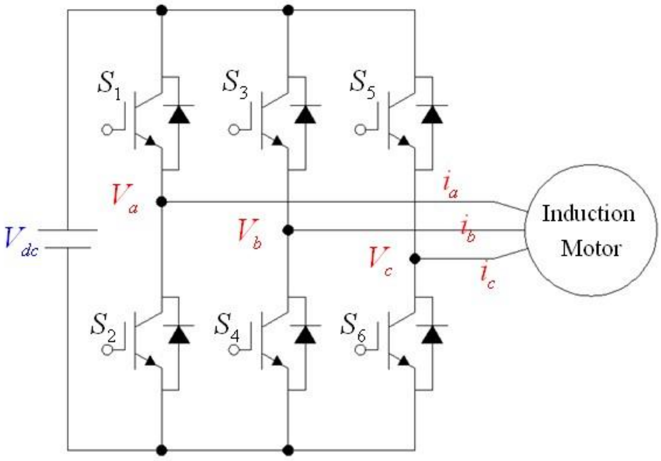


Figure 2.8: Three-phase voltage source inverter

The six transistors in the inverter can form 8 switch variables, 6 of them are nonzero vectors and 2 of them are zero vectors.

The relationship between the switching variable vector $[S_1, S_3, S_5]^T$ and the phase voltage $[V_{an}\ V_{bn}\ V_{cn}]^T$, and the line to line voltage $[V_{ab}\ V_{bc}\ V_{ca}]^T$ can be expressed below:

$$\begin{pmatrix} V_{an} \\ V_{bn} \\ V_{cn} \end{pmatrix} = \frac{V_{dc}}{3} \begin{pmatrix} 2 & -1 & -1 \\ -1 & 2 & -1 \\ -1 & -1 & 2 \end{pmatrix} \begin{pmatrix} S_1 \\ S_3 \\ S_5 \end{pmatrix} \tag{2.38}$$

$$\begin{pmatrix} V_{ab} \\ V_{bc} \\ V_{ca} \end{pmatrix} = V_{dc} \begin{pmatrix} 1 & -1 & 0 \\ 0 & 1 & -1 \\ -1 & 0 & 1 \end{pmatrix} \begin{pmatrix} S_1 \\ S_3 \\ S_5 \end{pmatrix} \tag{2.39}$$

According to eq. 2.38 and eq. 2.39, the eight voltage vectors, switching vectors and the output line to neutral voltage can be summarized in the given table:

Table 2.1: Voltage vectors table

Voltage vectors	Switching vectors			Line to neutral voltage (V_{dc})		
	S_1	S_3	S_5	V_{an}	V_{bn}	V_{cn}
V0	0	0	0	0	0	0
V1	1	0	0	2/3	-1/3	-1/3
V2	1	1	0	1/3	1/3	-2/3
V3	0	1	0	-1/3	2/3	-1/3
V4	0	1	1	-2/3	1/3	1/3
V5	0	0	1	-1/3	-1/3	2/3
V6	1	0	1	1/3	-2/3	1/3
V7	1	1	1	0	0	0

Also, these eight vectors could be plotted out in the Alfa-beta plane as shown below:

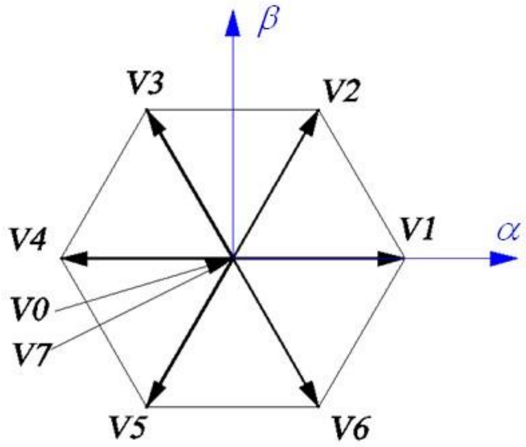


Figure 2.9: Voltage source inverter output vectors in the Alfa-beta plane

The SVPWM output voltage waveforms are shown below, the first one is a saddle-shaped wave which represents the phase voltage, and the second one is the line to line output voltage.

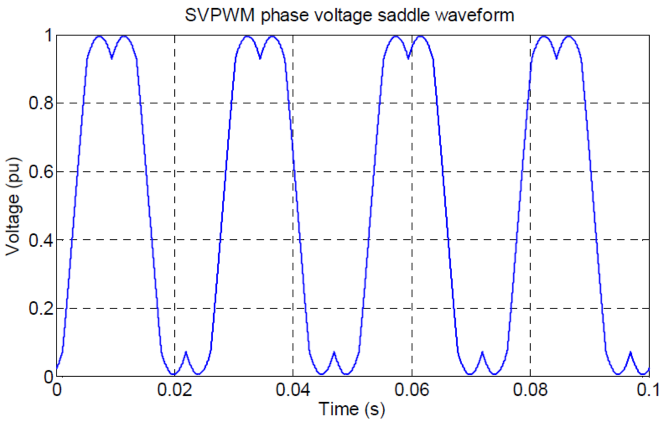


Figure 2.10: Phase voltage of SVPWM

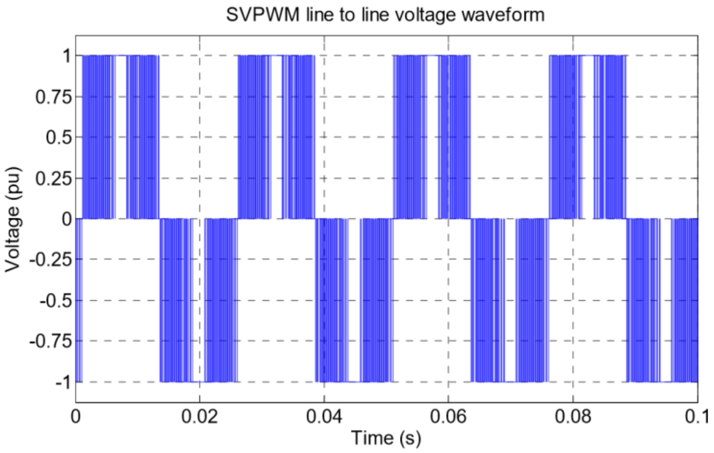


Figure 2.11: Line to line voltage of SVPWM

Chapter III: Field oriented control of induction motor

3.1. Introduction

Vector control or field oriented control has been invented by Blaschke to emulate DC motor characteristics in an induction motor [3] and [14]. In general, an electrical motor can be thought of as controlled torque source. The torque is produce in the motor by the interaction between the magnetic field of the stator field and the current in the rotor. The stator field should maintain at a certain level, sufficiently high to produce a high torque, but not too high to result in excessive saturation of the magnetic circuit of the motor. By fixed stator field, the torque is proportional to the rotor current [10].

The construction of a separately excited DC motor ensures that the stator field is always orthogonal to the rotor field. Being orthogonal, there is no interaction between these two fields. Therefore, independent control of the rotor current and stator field is feasible where the current in the stator determine the system field, while the current in the rotor can be used as a direct mean of torque control [3], [10] and [15].

In squirrel-cage induction motor, the rotor current is not fed directly by an externally source but it results from induced EMF in the rotor winding. In other words, the stator current is the source of magnetic field in the stator and rotor current. Therefore, the control of induction motor is not simple as DC motor due to the interaction between the stator field and rotor field whose orientation is not always held at 90° but it is varying depending on the operation conditions. We can obtain DC motor like performance in an induction motor by holding orthogonal orientation between the stator and rotor fields to achieve independently control of flux and torque. Such a scheme is called vector control or field oriented control [14] and [15]. First we will review the fundamentals of torque production and control in the DC motor.

3.2. Field oriented control principle

Field oriented control (FOC) consists of controlling the stator current components, represented by a vector, in a synchronously rotating reference frame d^e-q^e, in which the expression of the electromagnetic torque of the smooth-air-gap of the motor is similar to the expression of the torque in a separately excited DC motor. This technique is based on transformation of three phase time and speed dependent system into two coordinate time variant system, where the sinusoidal variable appears as DC quantities in steady state. Thus, this transformation leads to a structure similar of that in DC motor [16] and [12].

The motor terminal phase currents I_{as}, I_{bs}, and I_{cs} are converted by Clarke transformation from three-phase to two-phase sinusoidal signals, I_d^s and I_q^s components, as shown in figure 3.1a. The two components are then converted by Park transformation to I_d^e and I_q^e components. These components are the direct axis component and quadrature axis component, respectively, in the rotating reference frame as shown in Figure 3.1b.

In FOC the direct component of stator current space vector I_d^e is analogous to field current I_f, and I_q^e is analogous to armature current I_a of a DC motor. Therefore, the torque can be expressed as

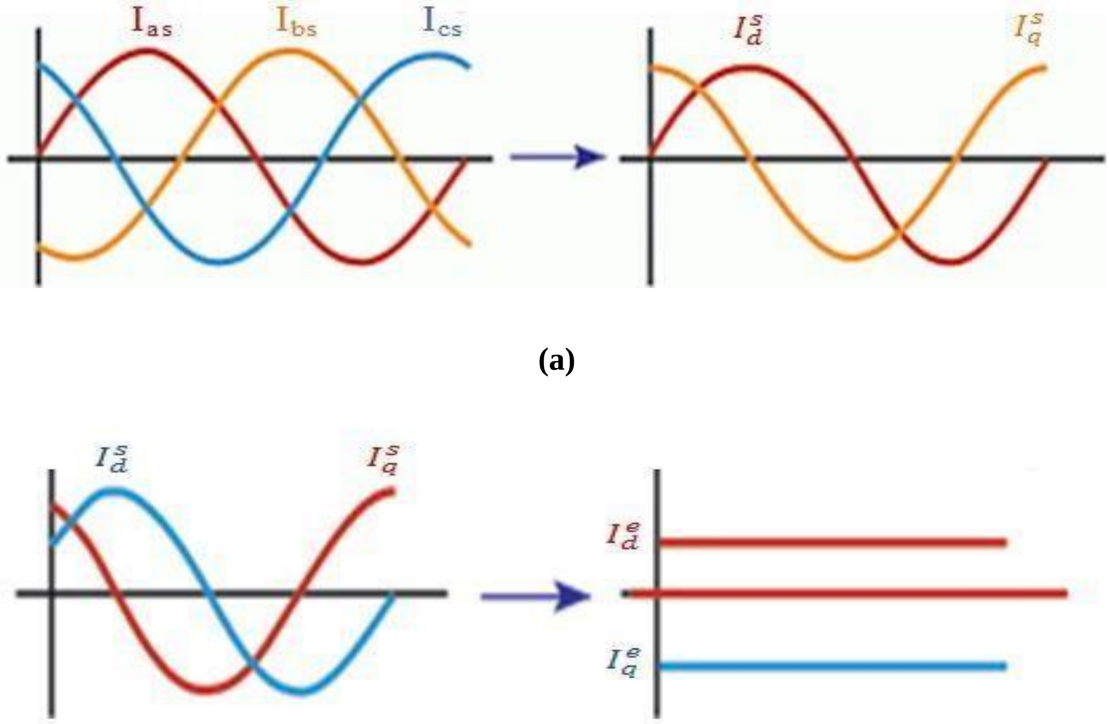
$$T_e = K_T\, I^e I_q^e \tag{3.1}$$

Or

$$T_e = \acute{K}_T\, \hat{\Psi}_r\, I_q^e \tag{3.2}$$

Where: $\hat{\Psi}_r$ is the peak value of sinusoidal space vector.

T_e is the motor torque.



(a)

(b)

Figure 3.1: (a) Clarke transformations, (b) Park transformations

The DC like machine performance can be achieved if the I_d^e is aligned in the direction of flux $\widehat{\Psi}_r$ and I_q^e is established perpendicular to it, as shown in figure 3.2. This mean that when quadrature reference current $I_q^{e^*}$ is controlled, it affects the actual I_q^e component only, but does not affect the flux $\widehat{\Psi}_r$. Similarly, when direct reference current $I_d^{e^*}$is controlled, it controls the flux only and does not affect the I_q^e component of current.

This vector or field orientation of currents is essential under all operating conditions in a vector control drive [16].

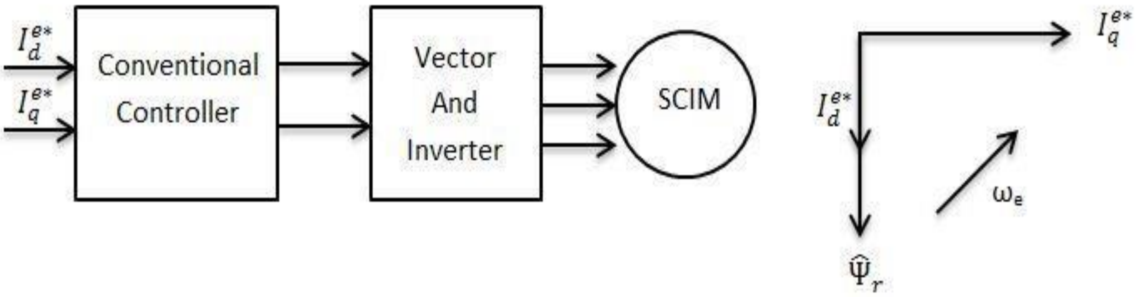


Figure 3.2: Vector control block diagram with two control current inputs

3.3. Space vector projection in stator reference frame

The three-phase currents, voltage, and fluxes of AC motor can be analyzed in terms of complex space vector. Figure 3.3 shows a cross section of a simple three-phase, two pole AC motor with three-phase current vectors.

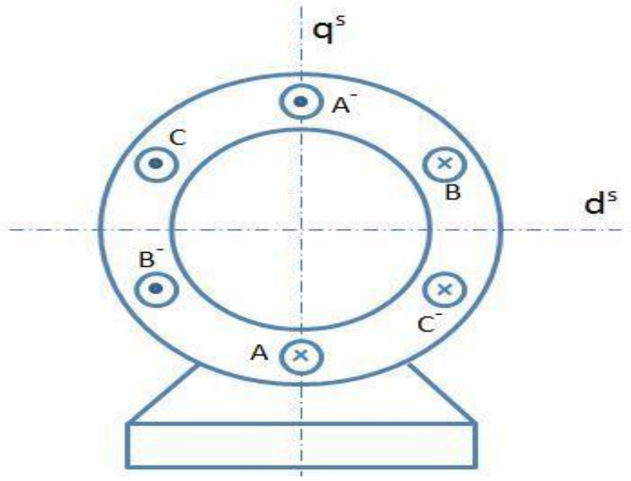


Figure 3.3: Schematic representation of a three-phase, 2-pole stator of an AC motor [10]

The horizontal and vertical geometrical axes of the stator are aligned with two axes, the direct axis, d^s, and the quadrature axis, q^s, respectively. They represent so-called stator or stationary reference frame.

The stator coils are supplied from a balanced three-phase AC source with a radian frequency ω. Figure 3.4 shows stator current vectors $I_{as,}$ I_{bs}, and I_{cs} at an instant of time, for example ωt=60°. The resultant stator current space vector, $I_{s,}$ constitutes a vector sum of the phase currents vectors. Since the three currents are 120° apart from each other, and since the phase-A coil was assumed to be in the vertical axis of the stator, then, in analytical form,

$$I_s = I_{as}e^{jo} + I_{bs}e^{j12o} + I_{cs}e^{j240} \tag{3.3}$$

It must be stressed that space vector has constant magnitude and rotate with angular velocity ω equal to the supply radian frequency [10].

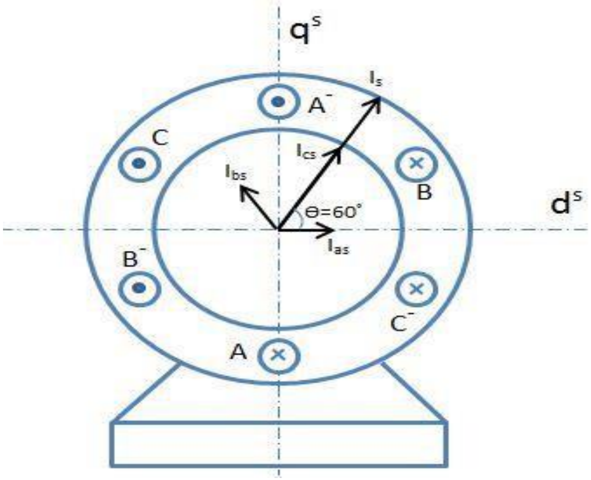


Figure 3.4: Stator current vectors at Θ = 60° [10]

Taking the d^s-axis as real and the q^s-axis as imaginary, vector $\overrightarrow{I_s}$ can be expressed as

$$\overrightarrow{I_s} = I_s\, e^{j\Theta} = I_d^s + jI_q^s \tag{3.4}$$

Where, in a P-pole stator of an AC motor the geometrical angle, Θ, can be expressed as

$$\Theta = \frac{2}{P}\,\omega t \tag{3.5}$$

And as illustrated in Figure 3.5, I_s donates the magnitude of vector $\overrightarrow{I_s}$, while I_d^s and I_q^s are its real part (d^s-axis) and imaginary (q^s-axis) components. These components have AC wave form under sinusoidal steady state conditions, this is because the current space vector is rotating vector with respect to the stator reference frame [10].

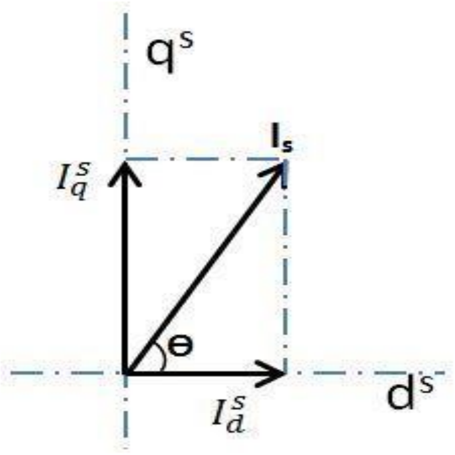


Figure 3.5: Stator current space vector and components in the stator reference frame [10]

3.4. Space vector projection in rotating reference frame

Control systems are usually represented by block diagrams in which the variables are time varying DC signals. Thus, Controlling AC motor is somewhat inconvenient because of AC quantities. Therefore, another transformation will be needed to convert the AC d^s-q^s components of the motor into DC variables. This conversion is so-called excitation or rotating reference frame d^e-q^e [10].

The excitation reference frame rotates with the angular velocity ω in the same direction as the rotor flux space vector rotates. As a result, in the steady states, the motor vectors coordinates in the new reference frame don’t vary in time. Figure 3.6 shows the stator current space vector in both stator reference frame and excitation reference frames [10].

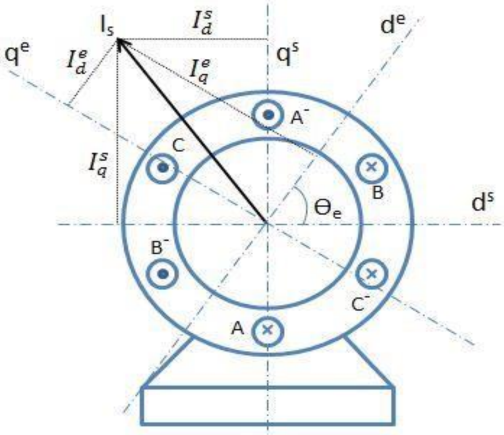


Figure 3.6: Stator current space vector and components in both the stator and excitation reference frame

3.5. Field oriented control methods

Knowledge of the instantaneous rotor flux position (angle), with which the revolving reference frame is aligned, constitutes the necessary requirement for proper transformation from stationary reference frame to rotating reference frame, or vice versa, in field orientation. In fact if there is an error in this variable, the d^e-axis is not aligned with the rotor flux vector. Thus, I_d^e and I_q^e are incorrect flux and torque components of the stator currents [12].

There are two general methods to measure the rotor flux vector angle. One, called the direct or feed-back method was invented by Blaschke, and the other, known as the indirect or feed-forward method was invented by Hasse. The two methods differ in the way the rotor angle is determined [14].

3.5.1. Direct field-oriented control

In direct field-oriented control (DFOC), the angle and magnitude of the reference flux vector are either measured or estimated from the stator currents and voltage using flux observers. Figure 3.7 shows a simple scheme for estimating the magnitude and phase (position) of the rotor flux vector, based on measuring the air gap flux and stator current. Placing two Hall sensors of magnetic field in the air gap of the motor, allows determination of the direct and quadrature components, λ_{dm} and λ_{qm}, of the air gap flux vector, λ_m. In analytical form, the rotor flux vector can be expressed as

$$\lambda_r = \frac{L_r}{L_m}\lambda_m - L_{1r}I_s \tag{3.6}$$

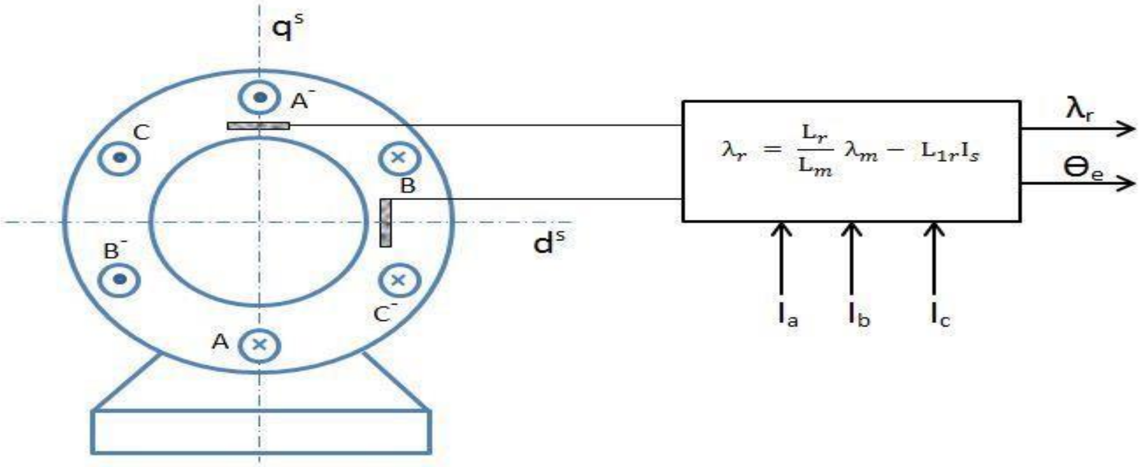


Figure 3.7: Determination of the magnitude and position of the rotor flux vector using Hall sensors and a rotor flux calculator [10]

Using equation (3.6), signals λ_{dr} and λ_{qr} are calculated as

$$\lambda_{dr} = \frac{L_r}{L_m} \lambda_{dm} - L_{1r} I_d^s \tag{3.7}$$

$$\lambda_{qr} = \frac{L_r}{L_m} \lambda_{qm} - L_{1r} I_q^s \tag{3.8}$$

Magnitude, λ_r, and phase, Θ_e, of the rotor flux vector are determined using the rectangular to polar coordinate transformation [10], [17].

$$\lambda_{dr} + j\lambda_{qr} \xrightarrow{\text{yiels}} \lambda_r \angle \Theta_e \tag{3.9}$$

Installing sensors in the air gap are inconvenient, and they spoil the ruggedness of the induction motor. Therefore, the stator current and voltage are used to compute the rotor flux vector. The stator flux vector can be estimated using following equation

$$\lambda_s = \int_0^t (\lambda_s - R_s I_s)\, dt + \lambda_s\,(0) \tag{3.10}$$

Then the air gap flux using the following equation

$$\lambda_m = \lambda_s - L_{1s} I_s \tag{3.11}$$

And finally the rotor flux vector, λ_r, can be estimated from equation (3.6) [17].

3.5.2. Indirect field oriented control

Indirect field orientation (IFOC) is an alternative approach to direct flux orientation. It is based on calculation of the slip speed ω_{sl}. The angular position, Θ_e, of the rotor flux vector is expressed as

$$\Theta_e = \int_0^t \omega_{sl}\, dt + \omega_{act}\, dt \tag{3.12}$$

Where ω_{act} is the rotor speed, which is easy to measure using a shaft position sensor. And ω_{sl} is the slip frequency speed. The required value of slip speed can be calculated by following equation

$$\omega_{sl} = \frac{L_m}{L_r} \cdot \frac{R_r}{\Psi_r} \cdot I_q^e \tag{3.13}$$

The IFOC is more popular than DFOC in industrial application due to implementation simplicity [10] and [17].

3.5.3. Indirect field-oriented control algorithm

The block diagram of indirect field-oriented control method is shown in figure 3.8. The induction motor is fed by a variable voltage, variable frequency PWM inverter, which operates in current control mode. The control scheme generates inverter switching commands to achieve the desired torque at the motor shaft. The algorithm of IFOC is given as:

1. Measure the stator phase currents I_a, I_b, and I_c. These currents are feed to Clarke transformation module that gives two components, I_d^s and I_q^s, in stationary reference frame.
2. Transform the set of these two currents, I_d^s and I_q^s, into rotating reference frame. This conversion called Park transformation, and provides I_d^e and I_q^e.
3. The rotor flux is computed by

$$\hat{\Psi}_r = \frac{\mathrm{L}_m \mathrm{I}_d^e}{1+\tau_r} \tag{3.14}$$

Where τ_r is the rotor time constant calculated by

$$\tau_\mathrm{r} = \frac{\mathrm{L_r}}{\mathrm{R_r}} \tag{3.15}$$

4. The rotor angle, Θ_e, required for coordinate transformation is computed by equation (3.12).
5. The motor speed, ω_{act}, is compared with the reference speed ω_{ref} and the error produced is fed to the speed controller. The output of the speed controller is electromagnetic torque $\mathrm{T_e^*}$.
6. The quadrature stator current component reference $I_q^{e^*}$ is calculated by

$$I_q^{e*} = \left(\frac{2}{3}\right)\left(\frac{2}{p}\right)\left(\frac{\mathrm{L}_r}{\mathrm{L}_m}\right)\left(\frac{T_e^*}{\hat{\Psi}_r}\right) \tag{3.16}$$

7. The direct stator current component reference I_d^{e*} is obtained by

$$I^{e*} = \frac{\hat{\Psi}_r^*}{I_m} \tag{3.17}$$

8. I_d^{e*} and I_q^{e*} current references are converted into $I_d^{s^*}$ and $I_q^{s^*}$, current references in stationary reference frame by using inverse Park transformation.
9. $I_d^{s^*}$ and $I_q^{s^*}$ current references are converted into phase current references I_a^*, I_b^*, and I_c^* by using inverse Clarke transformation and fed to the current controller. The controller processes the measured and reference currents to produce the inverter gating signals [15] and [18].

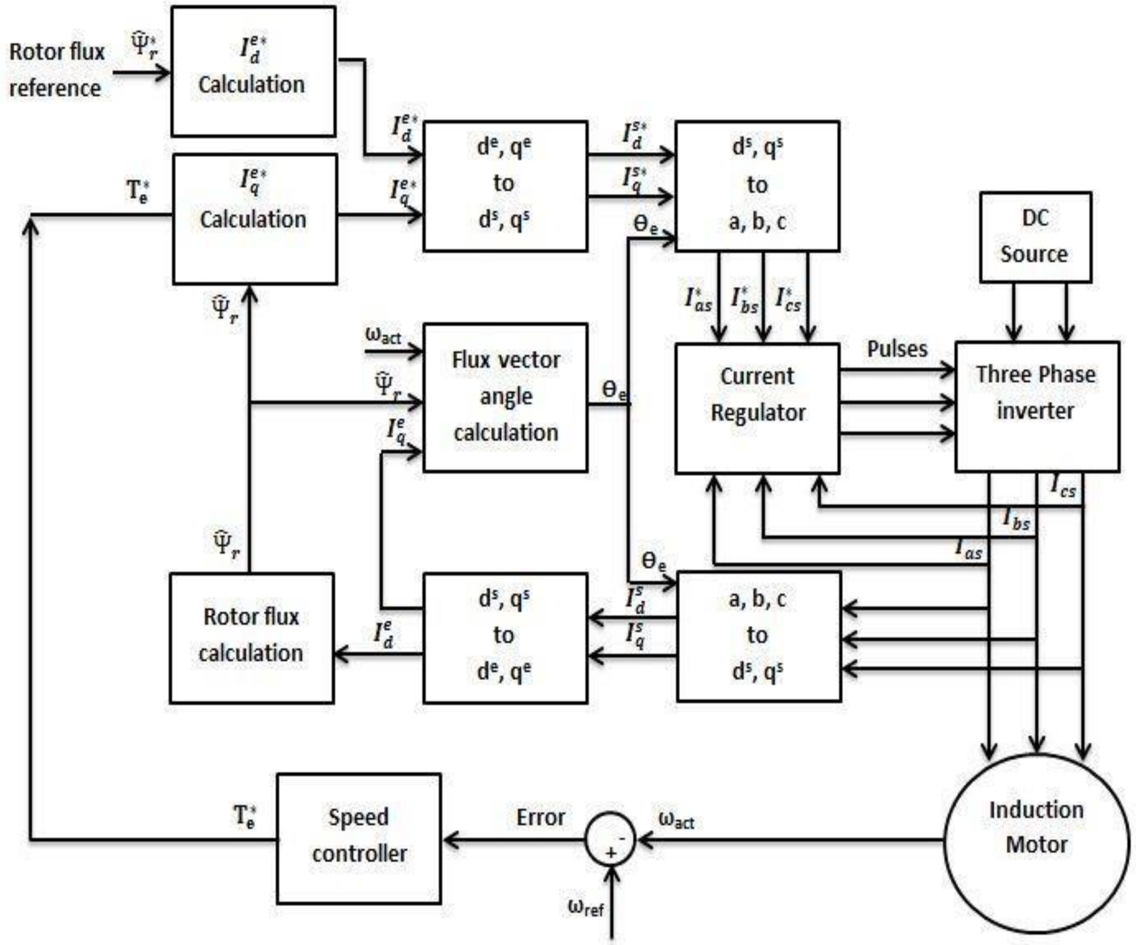


Figure 3.8: Indirect field-oriented control scheme.

Chapter IV: Control of the induction machine

The dynamic d-q model of an AC motor is complex, multivariable, and nonlinear. However, vector control or field-oriented control can overcome this problem, but accurate vector control is nearly impossible. To combat this problem, classical control, fuzzy logic controller, and fuzzy-PID controller are combined with indirect field-oriented control to solve this problem [16].

4.1. Conventional controller

Induction motor can be controlled with the help of conventional PI, PD, and PID controller with the use of indirect field-oriented control technique. The conventional controller is a feedback controller. It calculates an error value as the difference between the measured process value and the desired set point value and then drives the controlled plant to keep the steady state error equal to zero.

4.1.1. PI Controller

Proportional-Integral, PI, controller is most widely adopted in industrial application due to its simple structure, easy to design and low cost. PI controller produces an output signal consist of a sum of error and the integral of that error. The error represents the difference between the desired motor speed and the actual motor speed and it is expressed as

$$\mathrm{E} = \omega_{ref} - \omega_{act} \tag{4.1}$$

Figure 4.1 shows the block diagram of classical PI controller. The transfer function for PI controller is expressed as

$$\frac{\mathrm{U(s)}}{\mathrm{E(s)}} = \mathrm{K_P} + \frac{\mathrm{K_i}}{\mathrm{s}} \tag{4.2}$$

where K_P is the proportional gain, K_i is the integral gain, and U(s) is the output control signal which represent T_e^* torque reference in vector control drive.

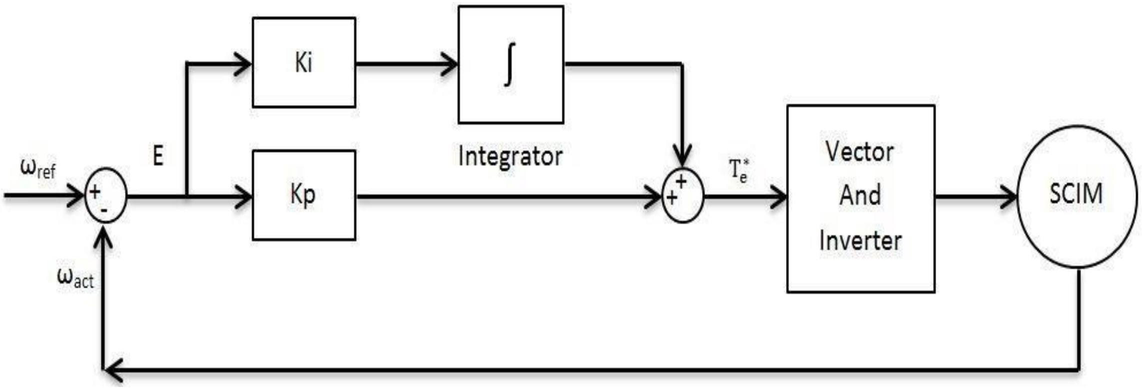


Figure 4.1: PI controller block diagram with vector control

The control signal is proportional to the error signal, the integral of error, the proportional gain K_p, and the integral gain K_i. The proportional controller will have the effect of reducing the rise time and steady state error. But, will never eliminate the error. The Integral control will have the effect of reduced the error near to zero value. But it has a negative effect on the speed of the response and overall stability of the system.

4.1.2. PD Controller

Proportional-Derivative, PD, controller has the ability to predict the future error of the system. Therefore, it uses to increase the stability of the system. The output of PD controller consists of a sum of two terms, the error signal and the derivative of that error.

The error signal is computed by equation (4.3). Figure 4.2 shows the block diagram for PD controller. The transfer function of PD controller is expressed as

$$\frac{\mathrm{U(s)}}{\mathrm{E(s)}} = K_P + K_d S \tag{4.3}$$

Where K_P is the proportional gain, K_d is the derivative gain, and U(s) is the output control signal which represent T_e^* torque reference in vector control drive.

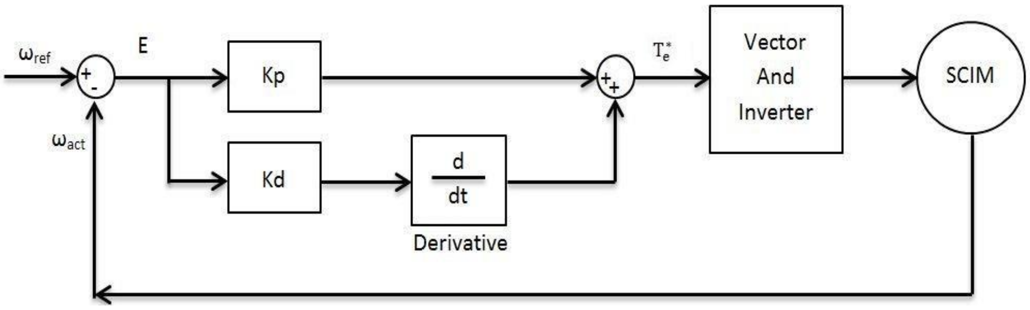


Figure 4.2: PD controller block diagram with vector control

The control signal is proportional to the error signal, the derivative of the error, the proportional gain K_i and the derivative gain K_d. Derivative control is used to anticipate the future behavior of the error signal by using corrective actions based on the rate of change in the error signal. Thus, it will have the effect of increasing the stability of the system, reducing the overshoot, and improving the transient response.

4.1.3. PID Controller

Proportional-Integral-Derivative, PID, controller is widely used in industrial control system. PID controller has all the necessary dynamics: fast reaction on change of the controller input (D controller), increase in control signal to lead error towards zero (I controller) and suitable action inside control error area to eliminate oscillations (P controller). Derivative mode improves stability of the system and enables increase in gain K_p, which increases speed of the controller response. The output of PID controller consists of three terms the error signal, the error integral and the error derivative. The error signal is computed by equation (4.1). Figure 4.3 shows the block diagram of PID controller. The transfer function of PID controller is expressed as

$$\frac{\mathrm{U(s)}}{\mathrm{E(s)}} = K_P + \frac{K_i}{s} + K \;\; S \tag{4.4}$$

Where K_P is the proportional gain, K_d is the derivative gain, k_i is the integral gain, and U(s) is the output control signal which represent T_e^* torque reference in vector control drive.

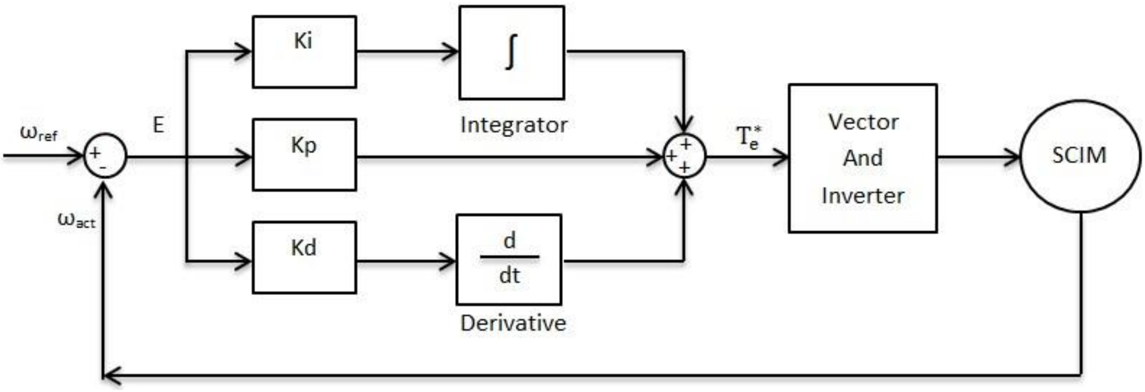


Figure 4.3: PID controller block diagram

PID controller combines the advantage of proportional, derivative and integral control action. Table 4.1 shows the effect of gain coefficient on the system performance.

Type	Rise time	Overshoot	Settling time	Steady state error
Kp	Decrease	Increase	Small change	Decrease
Ki	Decrease	Increase	Increase	Eliminate
Kd	Small change	Decrease	Decrease	Small change

Table 4.1: The effects of gain coefficients on the performance of PID controller system

4.2. Fuzzy logic controller

4.2.1. Introduction

Fuzzy logic, first introduced by Lotfi A. Zadeh [11] in 1965, embodies human-like thinking into a control system. A fuzzy controller employs a mode of approximate reasoning resembling the decision-making route of humans, that is, the process people use to infer conclusions from what they know. Fuzzy control has been primarily applied to the control of processes through fuzzy linguistic descriptions stipulated by membership functions.

The conventional Boolean logic has been extended to deal with the concept of partial truth - truth values which exist between “completely true" and "completely false", and what we shall be referring to as fuzzy logic [11]. This is achieved through the concept of degree of membership. The essence of fuzzy logic rests on a set of linguistic if-then rules, like a human operator. It has met a growing interest in many motor control applications due to its non-linearity handling features and independence of plant modeling. Moreover, the fuzzy logic concepts play a vital role in developing controllers for the plant since it isn’t needy of the much-complicated hardware and all it necessitates are only some set of rules.

4.2.2. Classical set and fuzzy set: a comparison

Let X be a space of objects (called universe of discourse or universal set) and x be a generic element of X.

A classical set A (A is a subset of X), is defined as a collection of elements or objects$x \in X$, such that each x can either belong or not belong to the set A. By defining a *characteristic function* for each element x in X, we can represent the classical set A by a set of ordered pairs

$(x, 0)$ or $(x, 1)$ which indicates $x \notin A$ or $x \in A$, respectively.

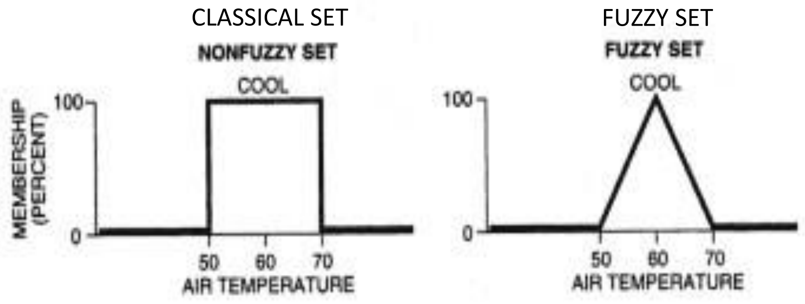


Figure 4.4: Example of classical set and fuzzy set

In spite of being an important tool for the engineering sciences, classical sets fail to replicate the nature of human conceptions, which tend to be abstract and vague.

A fuzzy set [11] conveys the degree to which an element belongs to a set. In other words, if X is a collection of objects denoted generically by x, then a fuzzy set A in X is defined as a set of ordered pairs:

$$A = \{(x, \mu_A(x)) | x \in X\} \tag{4.5}$$

Where $\mu_A(x)$ is known as the *membership function* for the fuzzy set A. MF serves the purpose of mapping each element of X to a membership grade (or membership value) between 0 and 1. Clearly, if the value of $\mu_A(x)$ is restricted to either 0 or 1, then A is reduced to a classical set and $\mu_A(x)$ is the characteristic function of A.

4.2.3. Fuzzy sets with a continuous universe

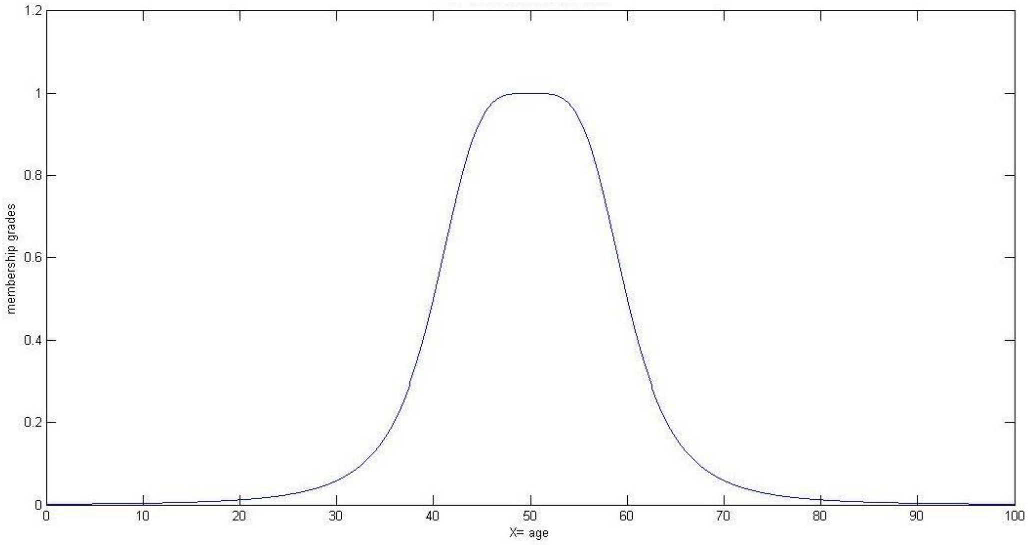


Figure 4.5: Membership function on a continuous universe

Let X be the set of possible ages for human beings. Then the fuzzy set A = "about 50 years old" may be expressed as

$$A = \{(x, \mu_A(x)) | x \in X\}$$

where,

$$\mu_A(x) = \frac{1}{1+(\frac{x-50}{10})^4} \qquad (4.6)$$

The aforementioned example clearly expresses the dependence of the construction of a fuzzy set on two things:

- Identifying a suitable universe of discourse.
- Laying down a suitable membership function.

At this point, it is imperative to state that the specification of membership functions is *subjective*, meaning that membership functions stated for the same notion by different persons will tend to vary noticeably. Subjectivity and non-randomness differentiate the study of fuzzy sets from probability theory. Latter deals with tangible handling of random phenomena.

Crisp variable: A crisp variable is a physical variable that can be measured through instruments and can be assigned a crisp or discrete value, such as a temperature of 30 ^{0}C, an output voltage of 8.55 V etc.

Linguistic variable: When the universe of discourse is a continuous space, the common practice is to partition X into several fuzzy sets whose MFs cover X in a more or less uniform manner. These fuzzy sets, which usually carry names that conform to adjectives appearing in our daily linguistic usage, such as "large", "medium" or "small", are called linguistic values. Consequently, the universe of discourse X is often called the linguistic variable.

4.2.4. Fuzzy set-theoretic operations

The most elementary operations on classical sets include union, intersection and complement. Analogous to these operations, fuzzy sets also have similar operations [11] which are explained below.

4.2.4.1. Containment or subset

Fuzzy set A is contained in fuzzy set B (or, equivalently, A is a subset of B) iff $\mu_A(x) \leq \mu_B(x)$ for all x. The following figure clarifies this concept.

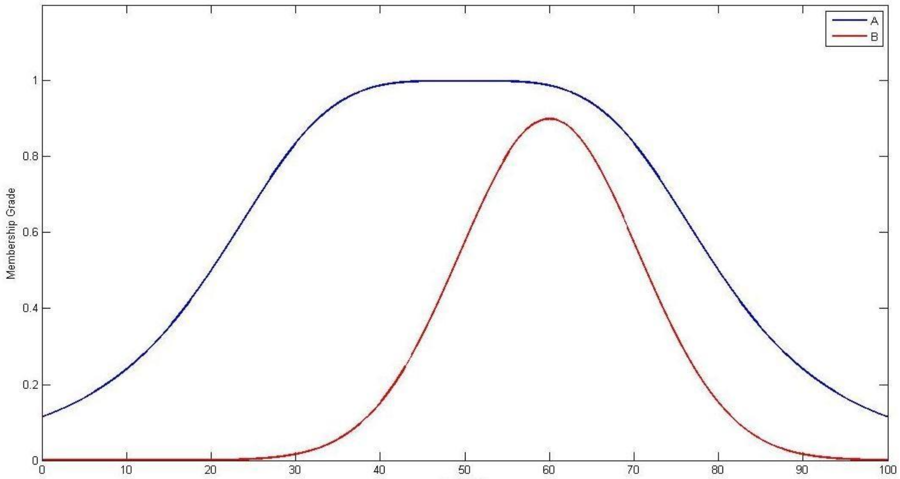


Figure 4.6: The concept of containment or subset

4.2.4.2. Union (Disjunction)

The union of two fuzzy sets A and B is a fuzzy set C, written as C = A ∪ B or C = A OR B, whose MF is related to those of A and B by

$$\mu_C(x) = \max\big(\mu_A(x), \mu_B(x)\big) = \mu_A(x) \ \vee \ \mu_B(x) \tag{4.7}$$

Equivalently, union is the *smallest* fuzzy set containing both A and B. Then again, if D is any fuzzy set encompassing both A and B, then it also contains A ∪ B. A union of two fuzzy sets A and B is shown in Fig 4.7 (b).

4.2.4.3. Intersection (Conjunction)

The intersection of two fuzzy sets A and B is a fuzzy set C, written as C = A ∩ B or C = A AND

B, whose MF is related to those of A and B by

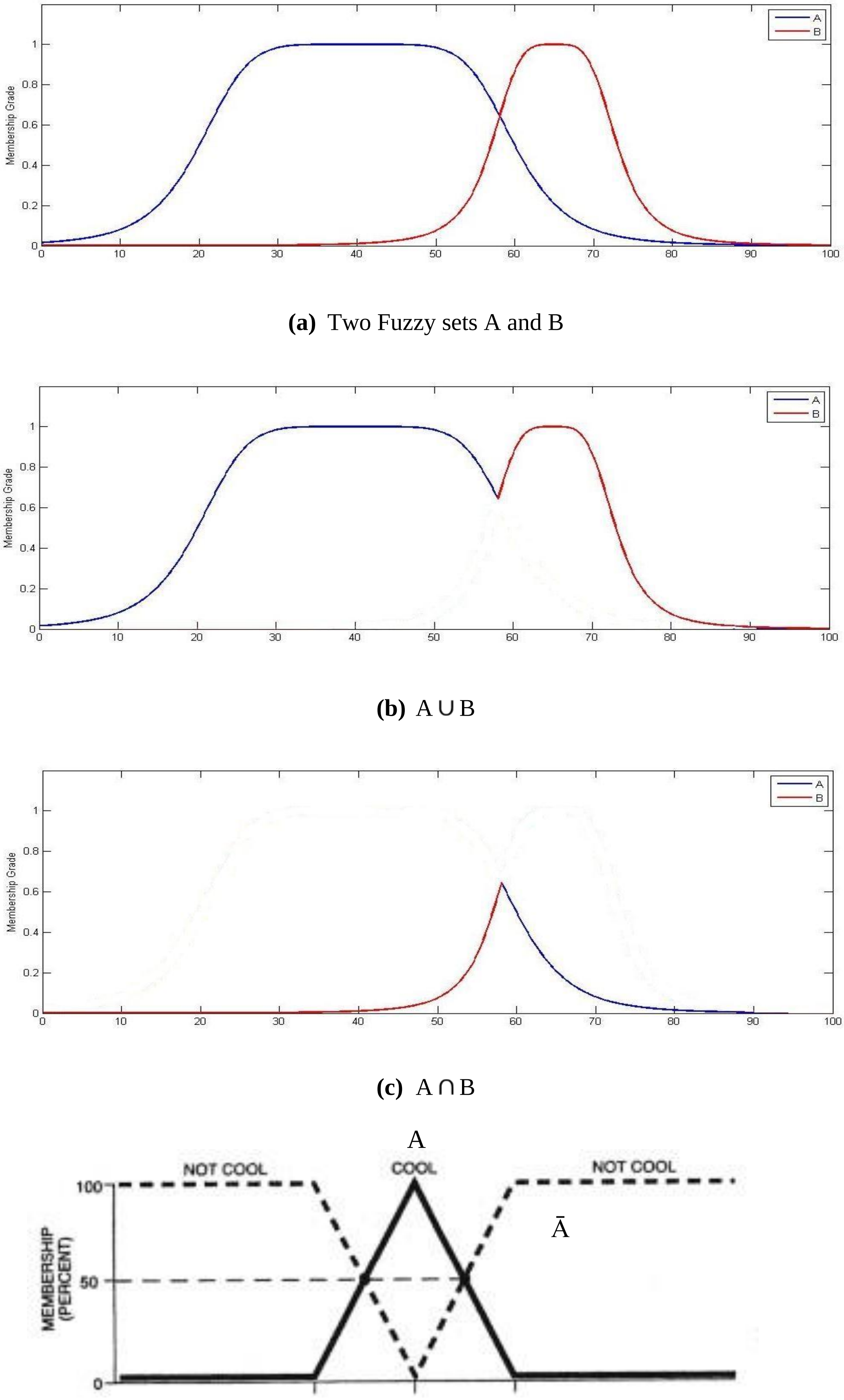
$$\mu_C(x) = \min\big(\mu_A(x), \mu_B(x)\big) = \mu_A(x) \ \wedge \ \mu_B(x) \tag{4.8}$$

Analogous to the definition of union, intersection of A and B is the *largest* fuzzy set which is contained in both A and B. An intersection of two fuzzy sets A and B is shown in Fig 4.7 (c).

4.2.4.4. Complement (Negation)

The complement of fuzzy set A, designated by Ā (NOT A), is defined as

$$\mu_{\bar{A}}(x) = 1 - \mu_A(x) \tag{4.9}$$



(a) Two Fuzzy sets A and B

(b) A ∪ B

(c) A ∩ B

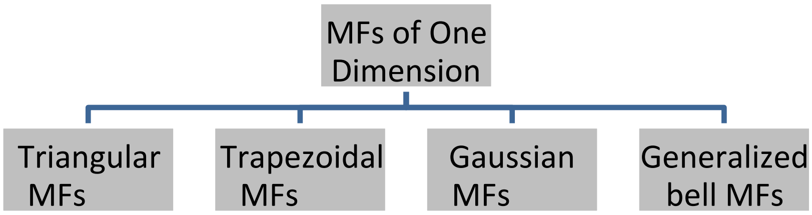
(d) Fuzzy set A and its Complement Ā

Figure 4.7: Operations on Fuzzy sets

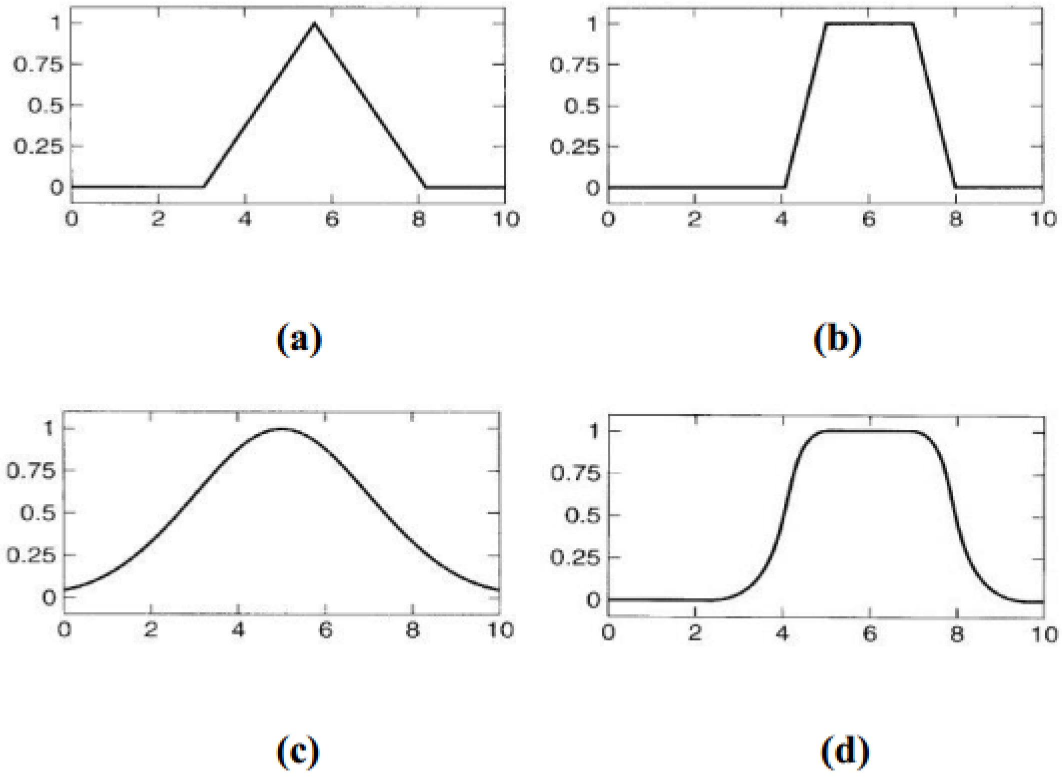
4.2.5. Formulating membership functions

Any membership function completely characterizes the fuzzy set that it belongs to. A convenient and succinct way to define an MF is to express it as a mathematical function. In order to define fuzzy membership function, designers choose many different shapes based on their preference and know-how.

Different classes of parameterized membership functions [19] commonly used are:



Among the alternatives just mentioned, the most popularly used MFs in real-time implementations are triangular and trapezoidal because of the fact that these are easy to represent the designer’s idea and require low computation time.



(a)

(b)

(c)

(d)

Figure 4.8: Different types of membership functions (a) triangular (b) trapezoidal (c) Gaussian (d) generalized bell

The aforementioned classes of parameterized MFs of one dimension are defined for MFs with a single input. MFs of higher dimensions can be defined similarly as per the increase in the number of inputs.

4.3. Fuzzy logic controller design

4.3.1. Components of FLC

The inputs to a Fuzzy Logic Controller are the processed with the help of linguistic variables which in turn are defined with the aid of membership functions. The membership functions are chosen in such a manner that they cover the whole of the universe of discourse. To avoid any discontinuity with respect to minor changes in the inputs, the adjacent fuzzy sets must overlap each other [20]. Because of a small time, constant in Fuzzy Logic Controllers, this criterion is very important in the design of the same.

There are basically three essential segments in Fuzzy Logic Controller:

1- Fuzzification block or Fuzzifier.
2- Inference System.
3- Defuzzification block or Defuzzufier.

4.3.1.1. Fuzzification block or fuzzifier

The first step towards designing a Fuzzy Logic Controller is choosing appropriate inputs which will be fed to the same. These input variables should be such that, they represent the dynamical system completely. Then the function of the Fuzzifier comes into picture. As discussed before, instead of using numerical variables, fuzzy logic uses linguistic variables for processing information. But since the inputs to the FLC are in the form of numerical variables (or in other words, crisp sets), they need to be converted into linguistic variables. This function of converting these crisp sets into fuzzy sets (linguistic variables) is performed by the Fuzzifier.

The fuzzification technique involves outlining the membership functions for the inputs. These membership functions should cover the whole universe of discourse and each one represents a fuzzy set or a linguistic variable. The crisp inputs are thus transformed into fuzzy sets. Triangular MF, Trapezoidal MF, Bell MF, Generalized Bell MF or Sigmoidal MF [19] can be used. Even a hybrid of any of the above Membership Functions can be used for fuzzification.

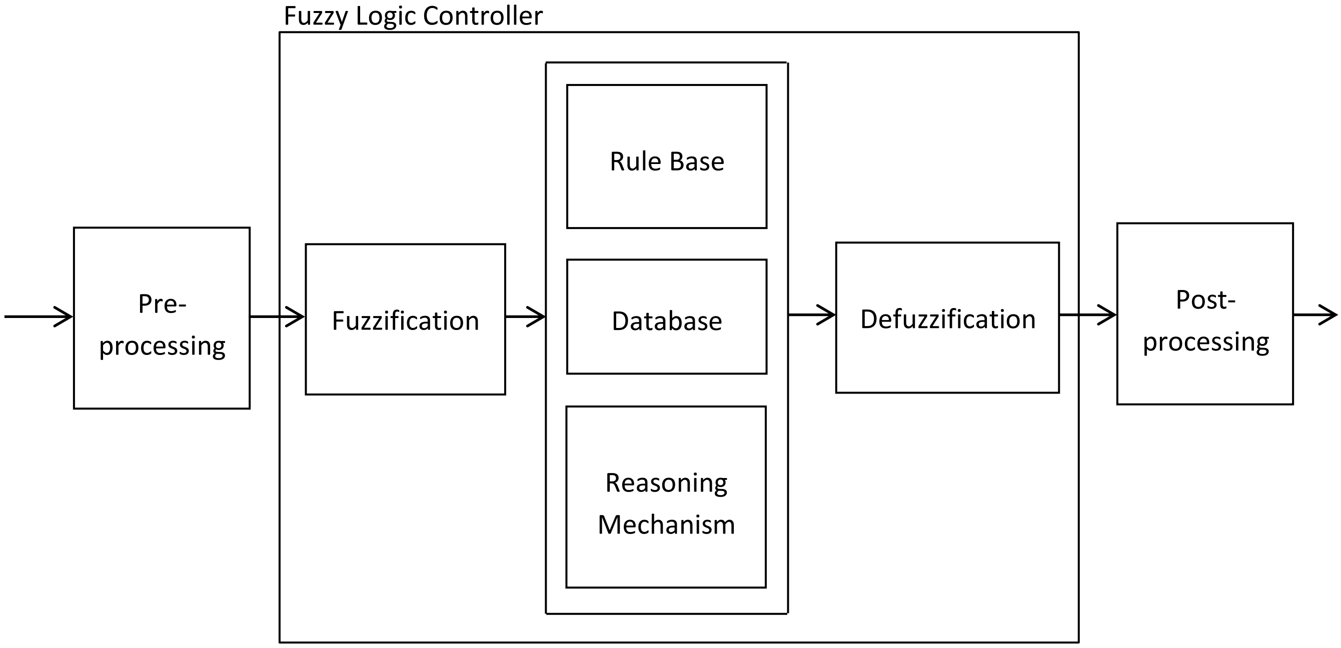


Figure 4.9: Fuzzy logic controller structure

4.3.1.2. Inference system

The inference system of a Fuzzy Logic Controller consists of the following three paradigms [19]:

1- **Rule Base:** It consists of a number of If-Then rules. The If side of the rule is called the antecedent and the Then side is called the consequence. These rules are very much similar to the Human thought process and the computer uses the linguistic variables, derived after fuzzification for execution of the rules. They very simple to understand and write and hence the programming for the fuzzy logic controller becomes very simple. The control strategy is stored in more or less the normal language.

2- **Database:** It consists of the all the defined membership functions that are to be used by the rules.

3- **Reasoning mechanism:** It performs the inference procedure on the rules and the data given to provide a reasonable output. It is basically the codes of the software which are process the rules and the all the knowledge based on a particular situation. It exercises a human brain type of attribute to methodically carry out the inference steps for processing the information.

4.3.1.3. Defuzzification block or defuzzifier

A defuzzifier performs the exact opposite function of a fuzzifier. It transforms the fuzzy variables (which are obtained as output after processing of the inputs) to crisp sets. The defuzzifier is necessary because in the real world the crisp values can only be taken as inputs to the other systems. Even though the fuzzy sets resemble the human thought process, their functionality is limited only to the above processes.

A defuzzifier is generally required only when the Mamdani Fuzzy Model is used for designing a controller. There are other types of architectures that can be used are:

- Tagaki-Sugeno Fuzzy Model [19].
- Tsukamoto Fuzzy Model [19].

Mamdani model is preferred here because it follows the Compositional Rule of Inference [19] strictly in its fuzzy reasoning mechanism. Unlike the Mamdani model, the outputs are defined with the help of a specific function for the other two models (first order polynomial in the input variables) and hence the output is crisp instead of fuzzy. This is counterintuitive since a fuzzy model should be able to propagate the fuzziness from inputs to outputs in an appropriate manner [19].

4.3.2. Block diagram for fuzzy speed control of induction motor

The Block diagram employing speed control of an induction motor is shown in Figure 4.10. It can be seen that the Indirect Field Oriented Control method is employed in the given block diagram. The frequency and supply voltage of the induction motor are varied such that it operates at steady state, at the desired speed. In the Vector control method, the inputs are torque/flux and output commands are reference current. Vector control drives provide excellent performance in terms of dynamic speed regulation, implementation of the same is tedious owing to on-line coordinate transformations that convert line currents into two axis representation and vice versa.

As shown in the block diagram, ω_m^* is chosen as the reference signal. The use of speed as reference signal is justified as the output of the system is speed and our aim is to control the speed of the induction motor. A tacho-generator, attached to the shaft of the induction motor, provides the current speed of the motor (ω_m) which is compared with the reference speed (ω_m^*), thus providing us with the speed error (E). This mechanism is called the feedback mechanism. A feedback mechanism is used to provide the quality of automation to the control system. The information about the instantaneous state of the output is fed back to the input which in turn is used to revise the same in order to achieve a desired output.

The Error Ration (RE), that is, the ratio of speed error over the reference error is computed and both E and RE are fed to the fuzzifier for fuzzification. The inference system then processes these two fuzzy inputs using the fuzzy control rules and the database, which are defined by the programmer based on the chosen membership function and fuzzy rule table, to give an output fuzzy variable. The fuzzy output thus obtained is defuzzified by the defuzzifier to give a crisp value, i.e. the Torque Reference (T_e^*). The speed error and change in error are expressed as follow

$$E = \omega_m^* - \omega_m \tag{4.10}$$

$$RE = \frac{E}{\omega_m^*} \tag{4.11}$$

The IFOC Block receives the Torque Reference from the FLC and generates the control Pulses to the inverter which Inverter uses these inputs to generate a three-phase voltage whose frequency and amplitude can be varied by the Fuzzy Logic Controller itself via the above-mentioned process. The three-phase voltage is fed to the induction motor which then runs with a speed which tends to follow the desired speed (reference speed, ω_m^*)

Before running the simulation in MATLAB/SIMULINK®, the Fuzzy Logic Controller is to be designed. This is done using the FIS editor. The membership functions and the rules have to be designed by the programmer so as to achieve the desired results. The FIS program thus generated is to be fed to the FLC before proceeding with the simulation.

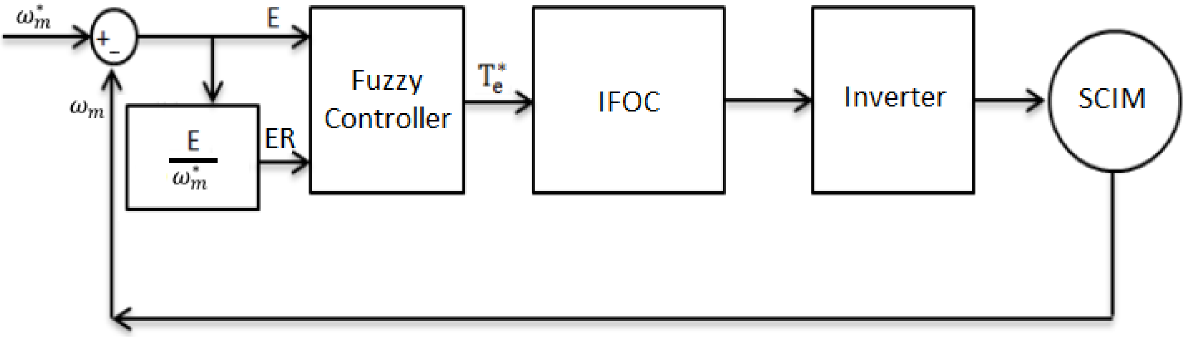


Figure 4.10: Block Diagram of the IFOC of IM with FLC

4.3.3. Membership functions design

The design of a Fuzzy Logic Controller requires the choice of Membership Functions. The membership functions should be chosen such that they cover the whole universe of discourse. It should be taken care that the membership functions overlap each other. This is done in order to avoid any kind of discontinuity with respect to the minor changes in the inputs. To achieve finer control, the membership functions near the zero region should be made narrow. Wider membership functions away from the zero region provides faster response to the system. Hence, the membership functions should be adjusted accordingly. After the appropriate membership functions are chosen, a rule base should be created. It consists of a number of Fuzzy If-Then rules that completely define the behavior of the system. These rules very much resemble the human thought process, thereby providing artificial intelligence to the system.

The inputs to the Fuzzy Logic Controller are:

1) Speed Error (E).
2) Error Ratio (RE).

4.3.3.1 Fuzzy sets and MFs for input variable speed error (E)

Table 4.2: Fuzzy sets and the respective membership functions for speed error (E)

Fuzzy set or label	Set Description	Range	Membership Function
NL (Negative Large)	Speed error is high in the negative direction.	-120 to -120 -120 to -50	Triangular
NM (Negative Medium)	Speed error is medium in the negative direction.	-120 to -50 -50 to -20	Triangular
NS (Negative Small)	Speed error is small in the negative direction.	-50 to -20 -20 to 0	Triangular
ZE (Zero)	Speed error is around zero.	-20 to 0 0 to 20	Triangular
PS (Positive Small)	Speed error is small in the positive direction.	0 to 20 20 to 50	Triangular
PM (Positive Medium)	Speed error is medium in the positive direction.	20 to 50 50 to 120	Triangular
PL (Positive Large)	Speed error is high in the positive direction.	50 to 120 120 to 120	Triangular

4.3.3.2. Fuzzy sets and MFs for input variable error ratio (ER)

Table 4.3: Fuzzy sets and the respective membership functions for error ratio (ER)

Fuzzy set or label	Set Description	Range	Membership Function
NL (Negative Large)	Speed error is high in the negative direction.	-1.0 to -1.0 -1.0 to -0.5	Triangular
NM (Negative Medium)	Speed error is medium in the negative direction.	-1.0 to -0.5 -0.5 to -0.2	Triangular
NS (Negative Small)	Speed error is small in the negative direction.	-0.5 to -0.2 -0.2 to 0	Triangular
ZE (Zero)	Speed error is around zero.	-0.2 to 0 0 to 0.2	Triangular
PS (Positive Small)	Speed error is small in the positive direction.	to 0.2 0.2 to 0.5	Triangular
PM (Positive Medium)	Speed error is medium in the positive direction.	0.2 to 0.5 0.5 to 1.0	Triangular
PL (Positive Large)	Speed error is high in the positive direction.	0.5 to 1.0 1.0 to 1.0	Triangular

These inputs are processed to obtain an output known as the change of control (T_e^*).

4.3.3.3.Fuzzy sets and MFs for output variable change of control (T_e^*)

Table 4.4: Fuzzy sets and the respective membership functions for change of control (T_e^*)

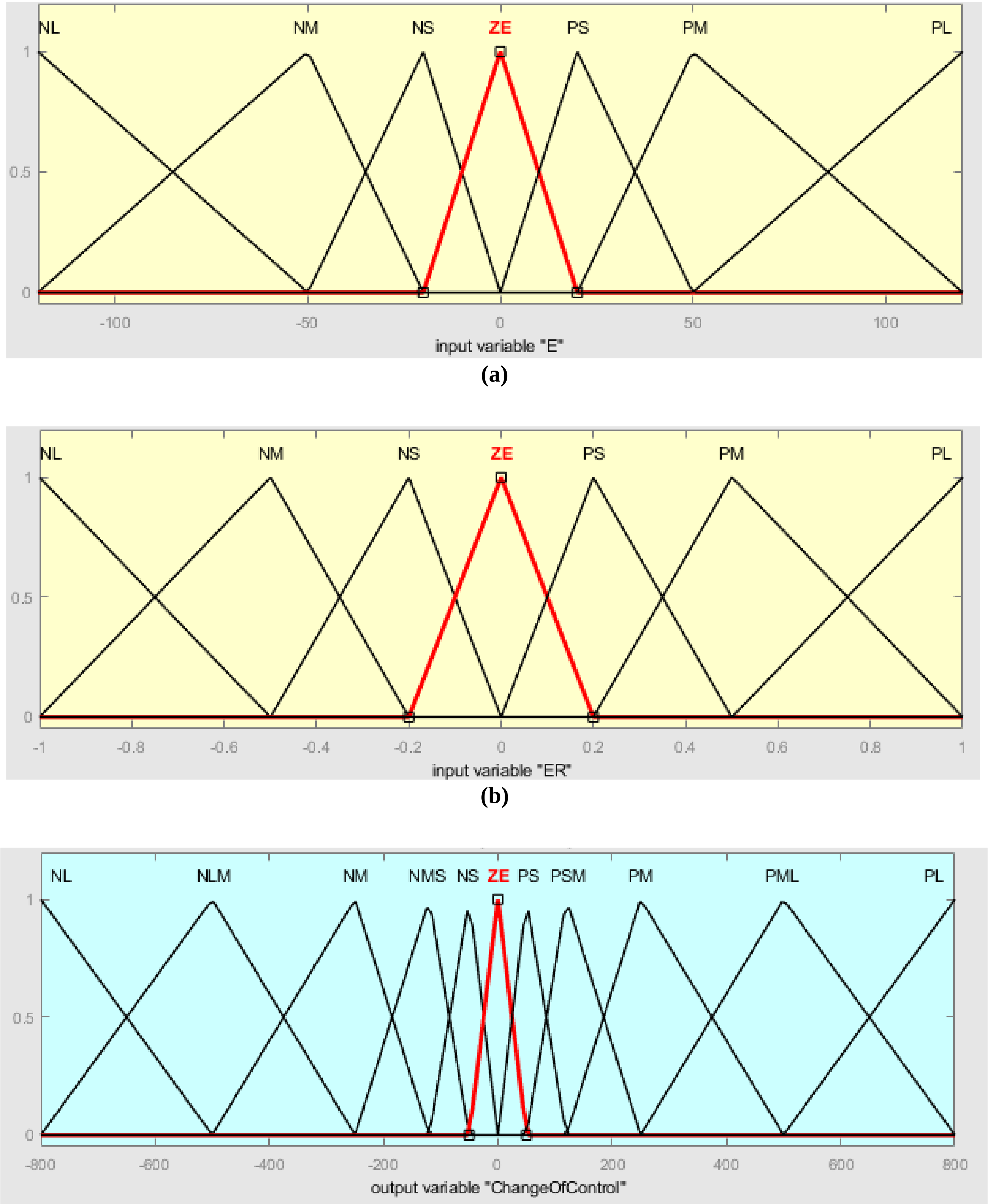
Fuzzy set or Label	Range	Membership Function
NL (Negative Large)	-800 to -800 -800 to -500	Triangular
NLM (Negative Large Medium)	-800 to -500 -500 to -250	Triangular
NM (Negative Medium)	-500 to -250 -250 to -120	Triangular
NMS (Negative Medium Small)	-250 to -120 -120 to -50	Triangular
NS (Negative Small)	-120 to -50 -50 to 0	Triangular
ZE (Zero)	-50 to 0 0 to 50	Triangular
PS (Positive Small)	0 to 50 50 to 120	Triangular
PMS (Positive Medium Small)	50 to 120 120 to 250	Triangular
PM (Positive Medium)	120 to 250 250 to 500	Triangular
PLM (Positive Large Medium)	250 to 500 500 to 800	Triangular
PL (Positive Large)	500 to 800 800 to 800	Triangular

4.3.3.4. Rule base design for the output (T_e^*)

The Rule Base for deciding the output of the inference system consists of 49 If-Then rules in this case since there are 7 fuzzy sets in each of the inputs. The table representing the rule base is as follows:

Table 4.5: Fuzzy rule table for output (T_e^*)

RE E	NL	NM	NS	ZE	PS	PM	PL
NL	NL	NL	NLM	NM	NMS	NS	ZE
NM	NL	NLM	NM	NMS	NS	ZE	PS
NS	NLM	NM	NMS	NS	ZE	PS	PMS
ZE	NM	NMS	NS	ZE	PS	PMS	PM
PS	NMS	NS	ZE	PS	PMS	PM	PLM
PM	NS	ZE	PS	PMS	PM	PLM	PL
PL	ZE	PS	PMS	PM	PLM	PL	PL



(a)

(b)

(c)

Figure 4.11: Membership functions for fuzzy speed controller (a) Error, (b) Error ratio, (c) Output control signal

Chapter V: SIMULINK model and results

Computer modeling and simulation is widely used to study the behavior of proposed system and to decide whether the new control design processes are valid in order to avoid the mistakes early in simulations before actual real implementation. Among several simulation software packages, SIMULINK is one of the most powerful techniques for simulating dynamic systems due to its graphical interface and simplicity. SIMULINK uses MATLAB as a tool for mathematical purposes which further enhance the modeling process. In the next few sections, the control scheme will be presented.

5.1. SIMULNIK model of controller scheme

A complete SIMULINK diagram of proposed control system for squirrel cage induction motor is shown in figure 5.1. The induction motor used in this simulation is a 50Hp, 460 V, 60 Hz, squirrel cage having the parameters listed in table 5.1. The induction motor stator is fed by a current controlled three-phase inverter bridge. The stator currents are regulated by hysteresis regulator which generates inverter drive signals for the inverter switches to control the induction motor. The motor torque is controlled by the quadrature-axis current component $I_q^{e^*}$ and the motor flux is controlled by direct-axis current component $I_d^{e^*}$. The motor speed is regulated by a control block which produces the required torque current component signal $I_q^{e^*}$.

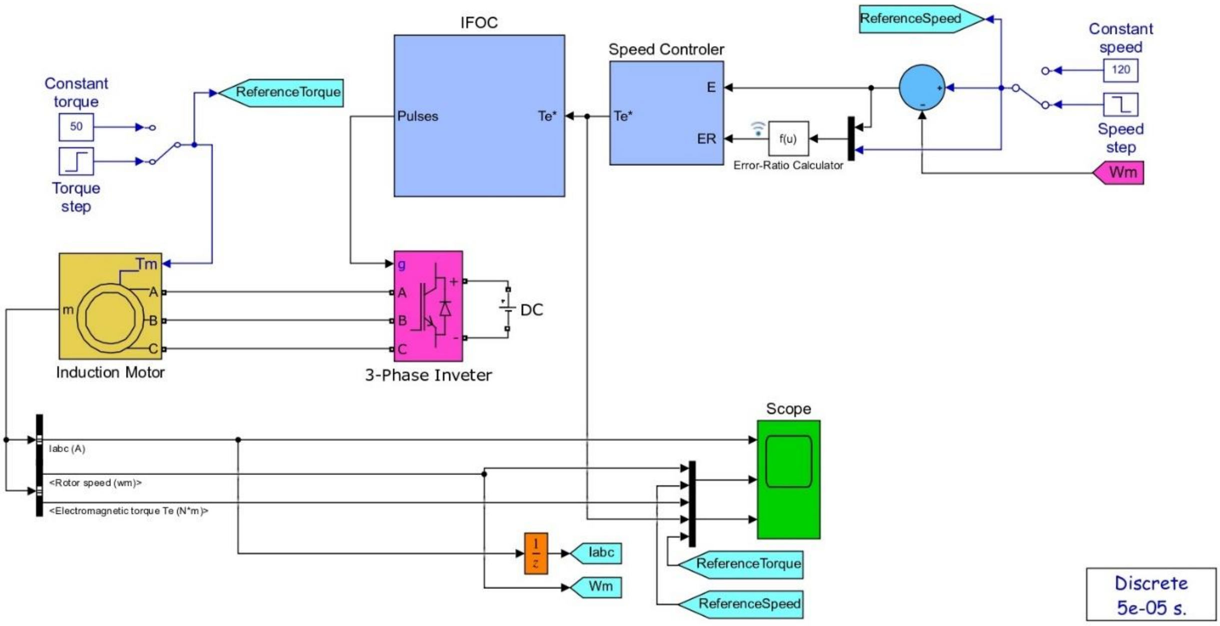


Figure 5.1: Complete SIMULINK model of speed controller system for three-phase squirrel cage induction motor

Table 5.1: A list of induction motor parameters with values based on predefined model in MATLAB.

Parameter	Symbol	value	Unit
Stator resistance	Rs	0.087	Ω
Rotor resistance	Rr	0.228	Ω
Mutual inductance	Lm	34.7	mH
Stator inductance	Ls	0.8	mH
Rotor inductance	Lr	0.8	mH
Inertia	J	1.662	kg.m^2
Friction factor	F	0.01	N.m.s
Pole pairs	P	2	()

5.1.1. Field-oriented control sub-blocks

The indirect field-oriented control algorithm has been simulated in MATLAB as it is shown in the figure 5.2. All sub blocks will be reviewed in the next coming sections.

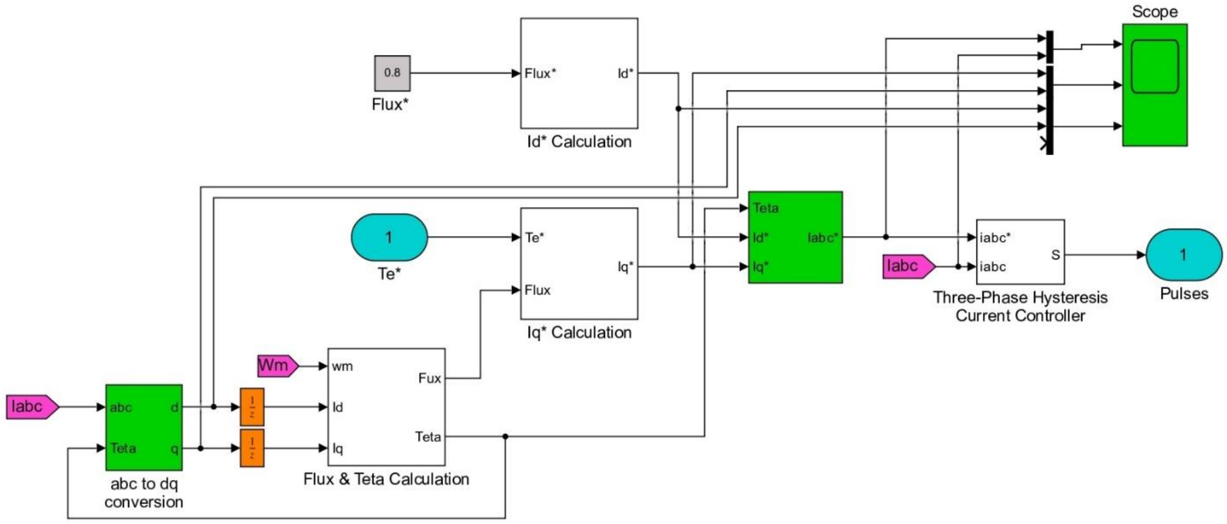


Figure 5.2: Field oriented control SIMULINK model

5.1.1.1. Forward Clarke and Park transformation

Figure 5.1 shows the SIMULINK model that converts the I_{as}, I_{bs}, and I_{cs} phase variables into I_d^e and I_q^e current components in rotating reference frame d^e-q^e. The equations (2.7) and (2.9) represent the mathematical relationship between the three-phase variables (I_{as}, I_{bs}, I_{cs}) and the current components (I_d^e , I_q^e). The simulation is done by entering stator currents to the max and applying the mathematical operation using function block parameters provided by MATLAB.

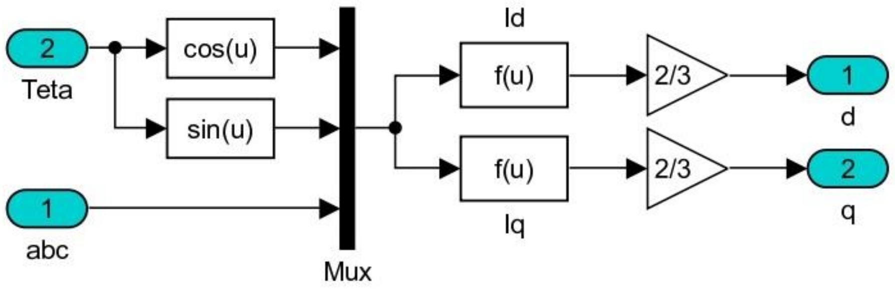


Figure 5.3: Forward Clarke and Park transformations SIMULINK model

In the Figure 5.3, we define the following:

Teta represents the angle between d^e-axis and rotor flux vector.

I_{abc} represents the three-phase stator currents I_{as}, I_{bs}, and I_{cs}.

I_d^e and I_q^e represents the stator current components in rotating reference frame.

5.1.1.2. Rotor flux and angle calculation

The knowledge of the rotor flux vector magnitude and position is a key information for field-oriented control of the three-phase induction motor. The SIMULINK model defined by the equations (3.14) and (3.12) is given by the Figure 5.4, and the Figure 5.5 respectively.

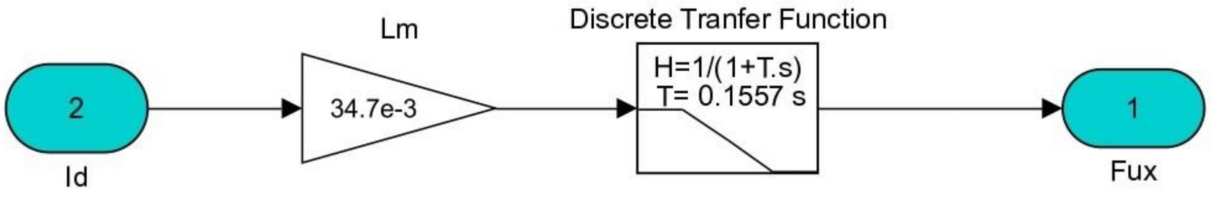


Figure 5.4: Rotor flux magnitude calculation

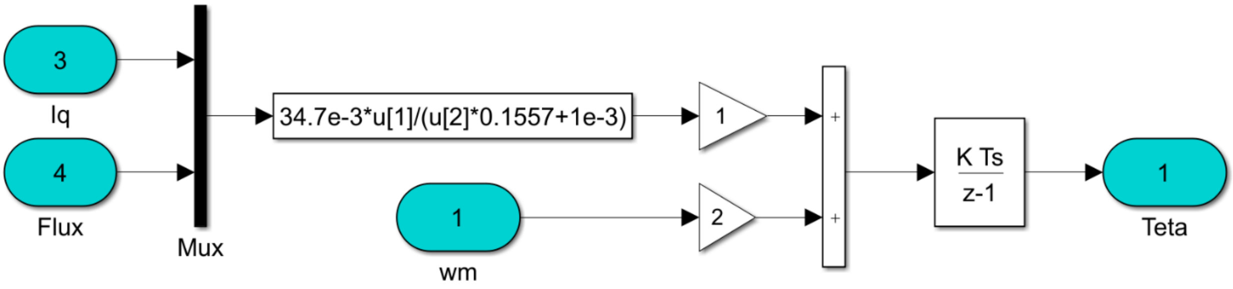


Figure 5.5: SIMULINK model to calculate rotor flux position (angle) that is necessary for transformation from stationary to rotating reference frame or vice versa

5.1.1.3. Direct reference current calculation

The stator direct-axis current reference $I_d^{e^*}$ can be obtained using equation (3.17) as it is shown in the Figure 5.6.

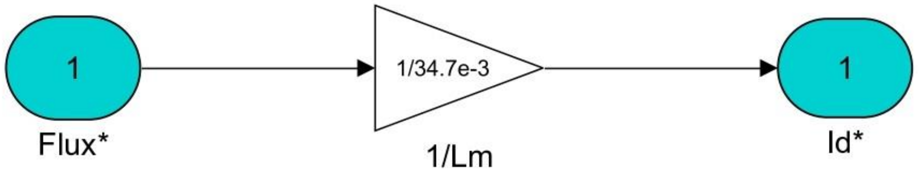


Figure 5.6: SIMULINK model to calculate the direct reference current component I_d^{e*}

Flux*: represents the reference flux value.

$I_d^{e^*}$: represents the direct current reference component.

5.1.1.4. Quadrature reference current calculation

The SIMULINK model shown in Figure 4.6 is used to obtain the stator quadrature reference current $I_q^{e^*}$ by using equations (3.14) and (3.16).

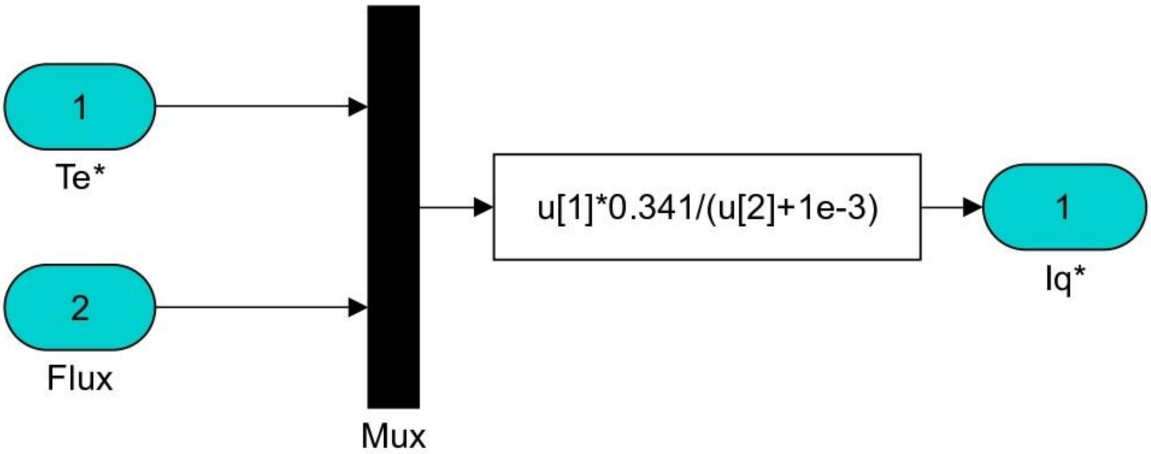


Figure 5.7: The used SIMULINK model to calculate quadrature current reference component $I_q^{e^*}$

T_e^* : represents the reference torque obtained from speed controller.

$I_q^{e^*}$: represents the quadrature reference current component.

5.1.1.5. Inverse Clarke and Park transformation

The inverse transformation SIMULINK model is shown in the Figure 5.8. This block uses equations (2.7) and (2.9) to perform the conversion of the $I_d^{e^*}$ and $I_q^{e^*}$stator current components in rotating reference frame back to I$_{as}$, I$_{bs}$, and I$_{cs}$ in the stationary reference frame.

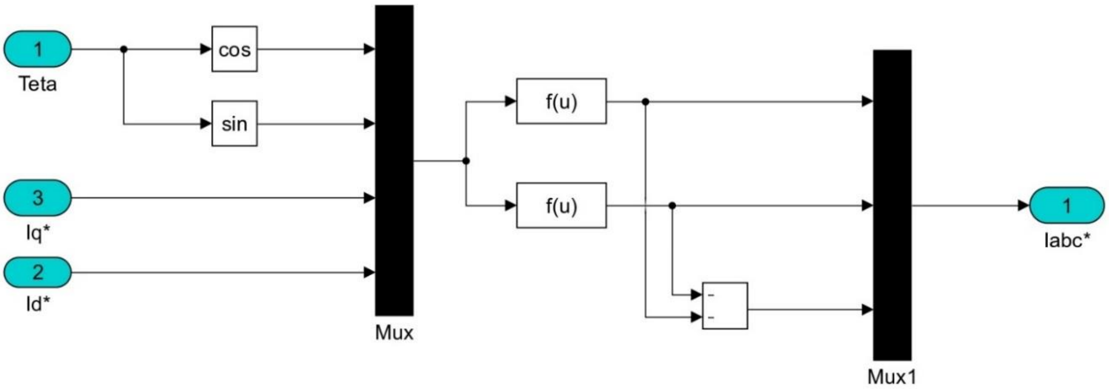


Figure 5.8: The used SIMULINK model to convert current components in rotating frame to three-phase currents in stationary frame

5.1.1.6. Current regulator

The current regulator shown in figure 5.9 is a bang-bang current controller with adjustable hysteresis band width. The hysteresis modulation is a feedback current control method where the motor current tracks the reference current within a hysteresis band. The figure below elaborates the operation principle of the hysteresis modulation. The controller generates sinusoidal reference current of desired magnitude and frequency, which is then compared to the actual motor line current. If the current exceeds the upper limit of the hysteresis band, the upper switch of the inverter arm is turned off and the lower switch is turned on. As a result, the current starts to decay. If the current passes the lower limit of the hysteresis band, the lower switch of the inverter arm is turned off and the upper switch is turned on. As a result, the current gets back into the hysteresis band. Hence, the actual current is forced to track the reference current within the hysteresis band.

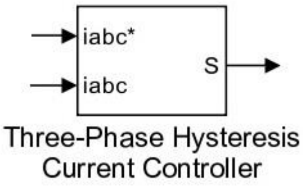


Figure 5.9: SIMULINK model of the current regulator

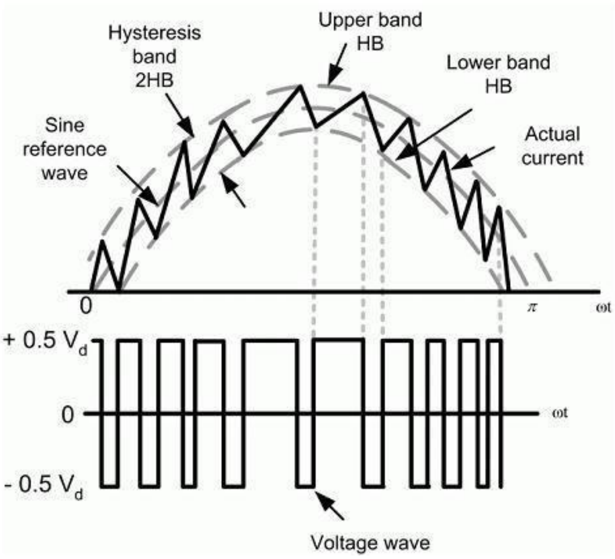


Fig 5.10: Operation principle of the hysteresis current regulator

5.1.2. Speed controller sub-block

The speed controller block contains a PID controller, a Fuzzy controller, and a Fuzzy-PID controller. The controllers developed in MATLAB using fuzzy logic toolbox and PID regulator respectively. The SIMULINK model for speed controller is shown in Figure 5.11. This model has the speed error (E) and the error ratio (ER) as inputs. The output signal from the model is the torque reference value which is then provided to IFOC block as input.

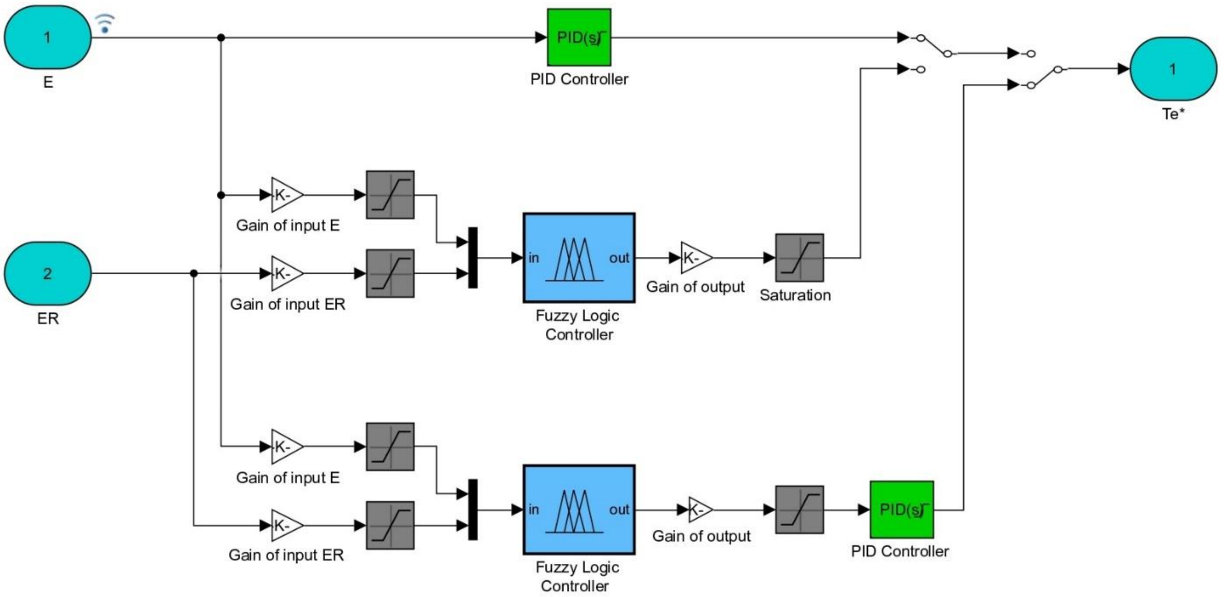


Figure 5.11: SIMULINK model of speed controller

E and ER: represent the error and the error ratio.

T_e^* : represents the reference torque.

5.2. Results and discussion

In this section, we present simulation results of the proposed speed controllers using MATLAB/SIMULINK and analyze them against the performance of three-phase squirrel cage induction motor. The effectiveness and robustness of the developed conventional PI speed controller, PID speed controller, Fuzzy speed controller, and Fuzzy-PID speed controller are evaluated under various operating conditions. The transient performances of all controllers are investigated while varying the reference speed and load torque values.

5.2.1. Fixed reference speed and varying load torque

The reference speed for all controllers was fixed at 120 rad/sec and the reference load torque was varying from 0 to 200 by steps. The computational time interval for conventional PI and PID controller models was fixed at 6 seconds while for fuzzy controller model it was fixed at 1 second. The response curves for PI, PID speed controllers and Fuzzy, Fuzzy-PID speed controllers are extracted as shown in the figures 5.12 through 5.15 respectively. The values of the different performance parameters such as rise time (t_r), settling time (t_s), peak overshoot (M_p), and steady state error (e_{ss}) of each controller are shown in table 5.2. According the results, it can be said that Fuzzy-PID has fast transient response and less overshoot than other controllers.

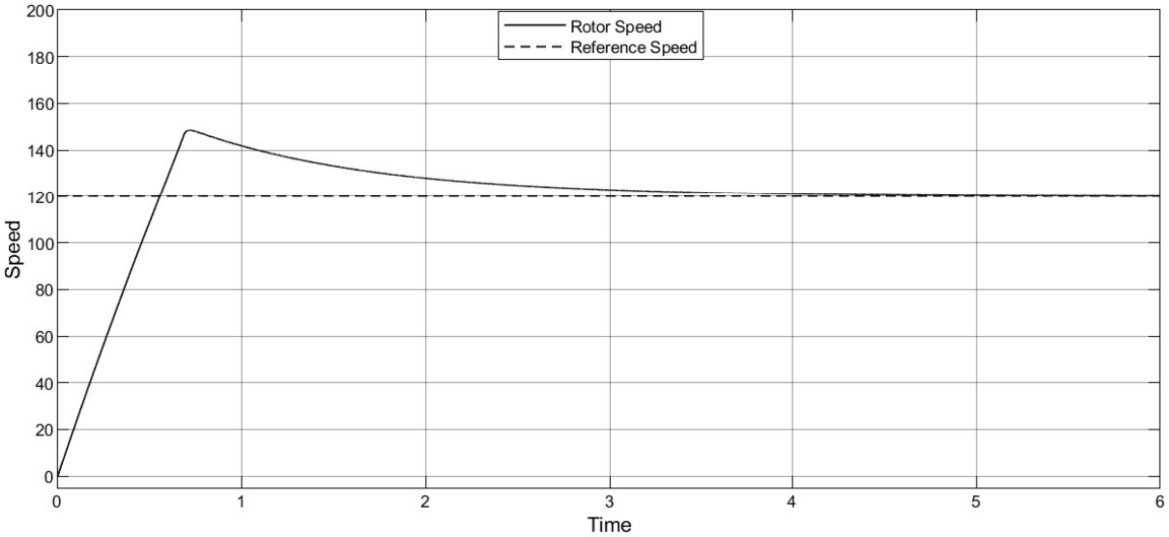


Figure 5.12: Speed response curve of IM at 100 N.m load torque and 120 rad/sec reference speed using PI controller

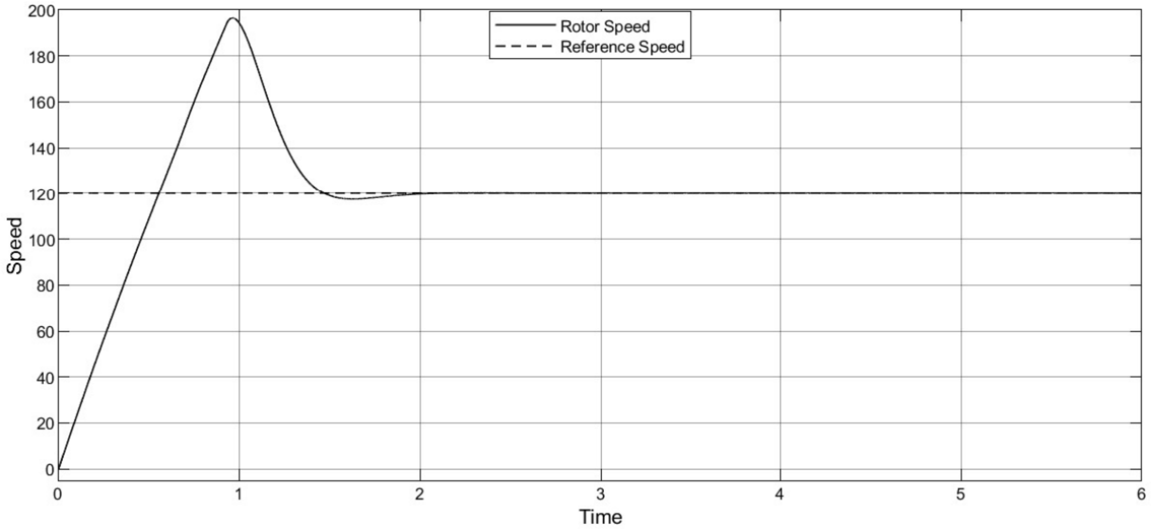


Figure 5.13: Speed response curve of IM at 100 N.m load torque and 120 rad/sec reference speed using PID controller

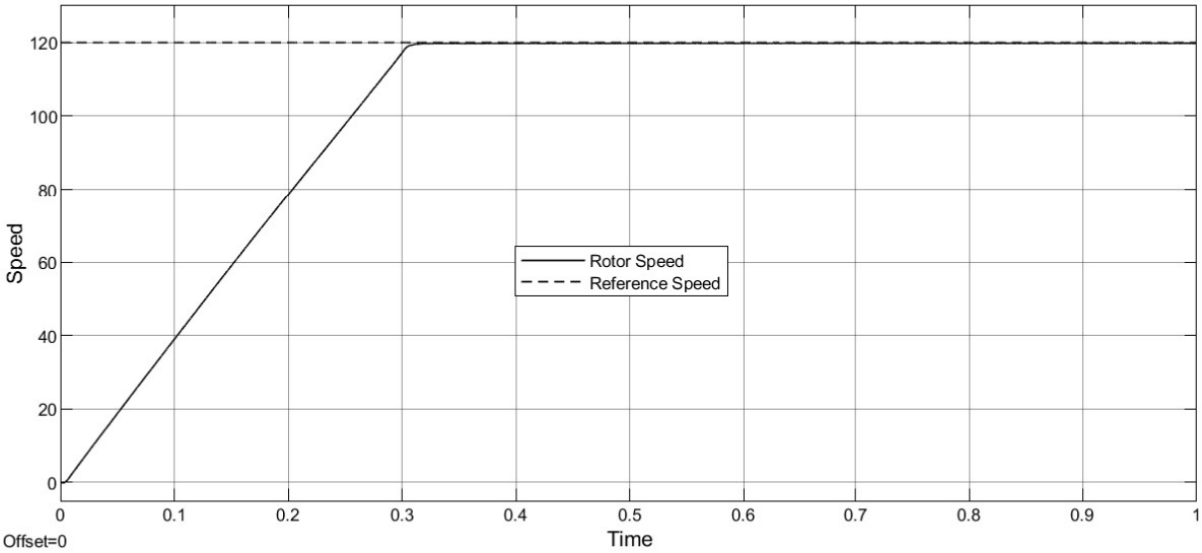


Figure 5.14: Speed response curve of IM at 100 N.m load torque and 120 rad/sec reference speed using Fuzzy controller

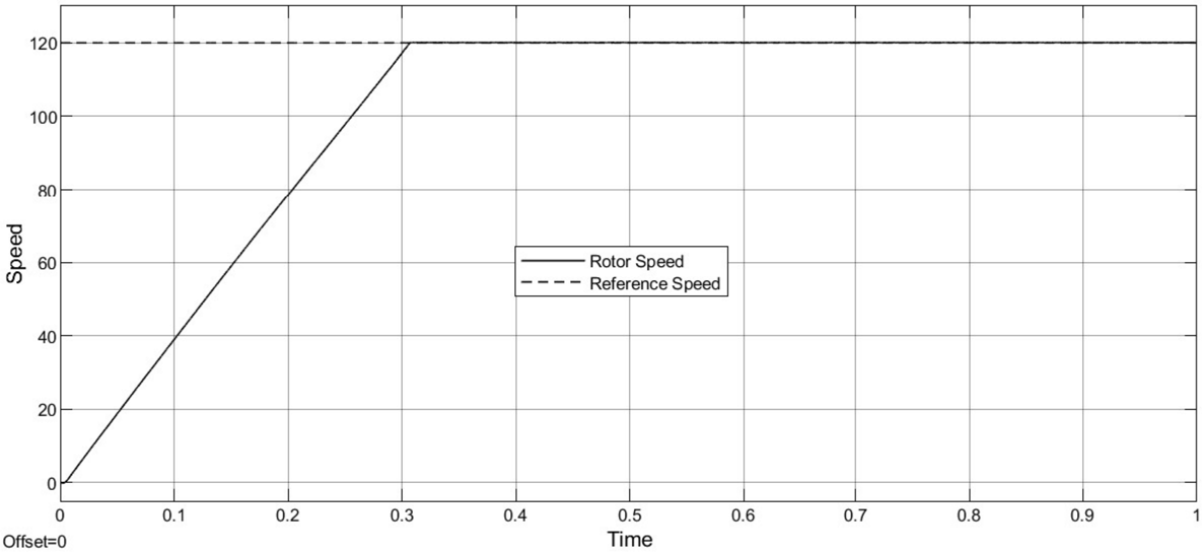


Figure 5.15: Speed response curve of IM at 100 N.m load torque and 120 rad/sec reference speed using Fuzzy-PID controller

Load (N.m)	Controller	t_r (sec)	t_s (sec)	Mp	ess
0	PI	0.3429	2.83	23.3	0.07
	PID	0.3434	1.185	69.1	0.02
	Fuzzy	0.2084	0.261	0	0.04
	Fuzzy-PID	0.2087	0.261	0.2	0.09
100	PI	0.4393	3.18	28.5	0.085
	PID	0.4356	1.677	76.5	0.021
	Fuzzy	0.2405	0.301	0	0.125
	Fuzzy-PID	0.2413	0.301	0.17	0.1
150	PI	0.5214	3.435	32.4	0.1
	PID	0.5114	1.885	80.4	0.0022
	Fuzzy	0.2607	0.327	0	0.38
	Fuzzy-PID	0.2618	0.327	0.16	0.1
200	PI	0.6221	3.78	37.6	0.14
	PID	0.6131	2.158	83.8	0.027
	Fuzzy	0.284	0.356	0	0.59
	Fuzzy-PID	0.2857	0.356	0.14	0.1

Table 5.2: Performance analyses of different speed controllers for IM at 120 rad/sec reference speed and different load torque.

5.2.2. Fixed load torque and step change in reference speed

The dynamic performances of the PI, PID, Fuzzy, and Fuzzy-PID controllers are analyzed by applying 100 N.m torque load and step change in reference speed, from 120 rad/sec to 40 rad/sec, at 3rd second interval time. Figures 5.16-5.19 show the simulation results. In these figures, the orange line represents the reference speed and blue line represents the response curve.

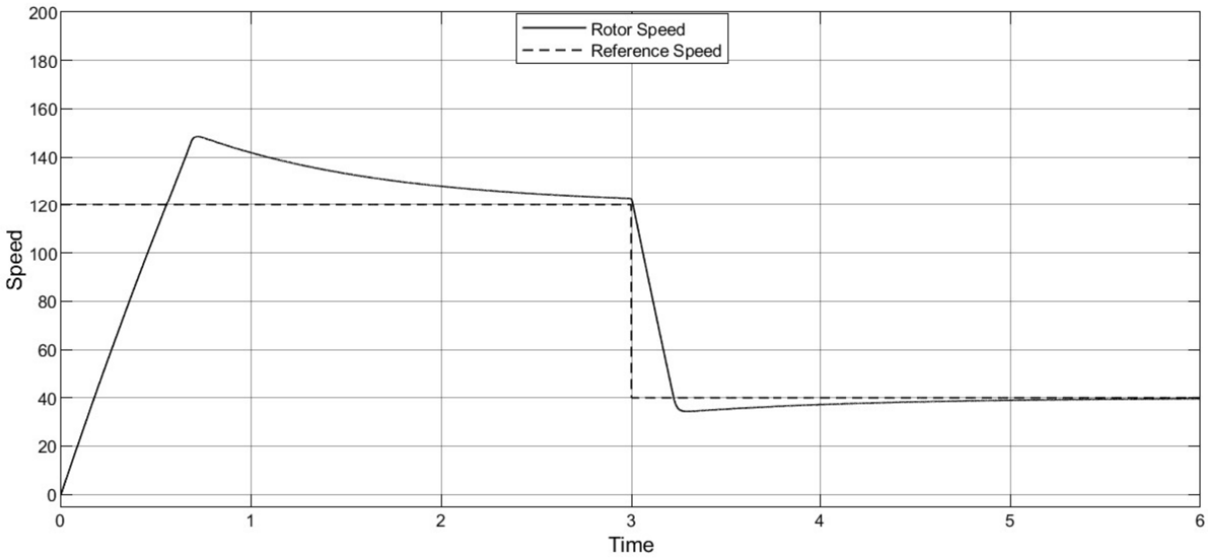


Figure 5.16: Speed response curve of IM at 100 N.m load torque and 120 to 40 rad/sec step change in reference speed using PI controller

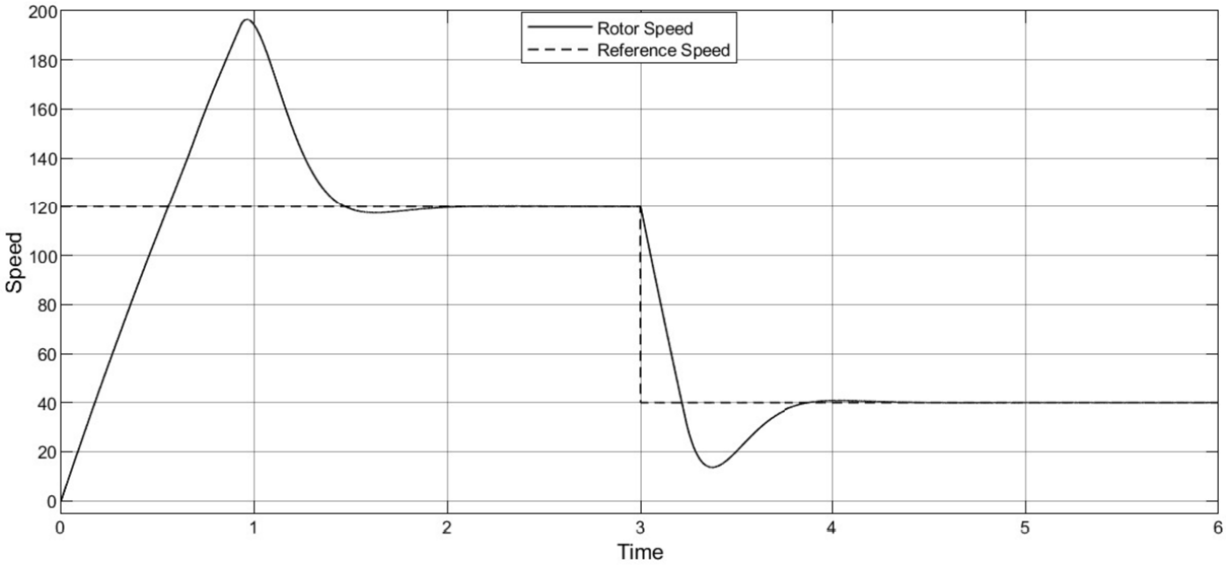


Figure 5.17: Speed response curve of IM at 100 N.m load torque and 120 to 40 rad/sec step change in reference speed using PID controller

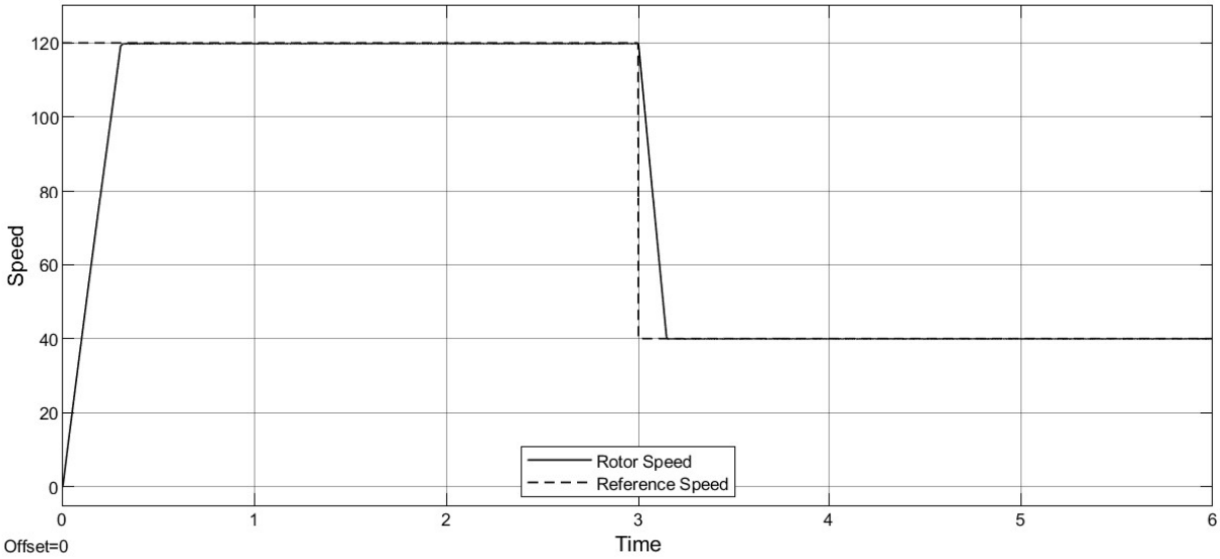


Figure 5.18: Speed response curve of IM at 100 N.m load torque and 120 to 40 rad/sec step change in reference speed using Fuzzy controller

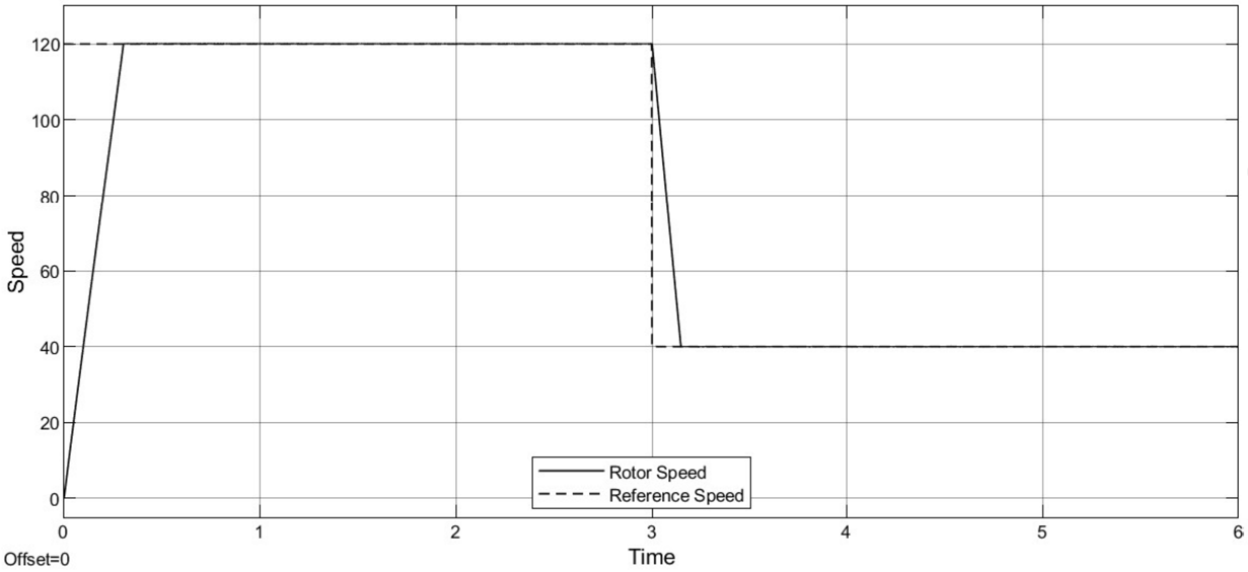


Figure 5.19: Speed response curve of IM at 100 N.m load torque and 120 to 40 rad/sec step change in reference speed using Fuzzy-PID controller

By analyzing figure 5.16, the motor speed in PI controller following the reference speed (120 rad/sec) until a step change in reference speed is applied to the motor at 3rd second interval time. We can easily notice that the motor speed response curve shows an overshoot = 28.5 and deviation from reference speed runs for +3 seconds to return to the desired speed.

In the case of PID controller as shown in figure 5.17, the motor speed response curve shows a big overshoot = 76.5 but the deviation from reference speed runs only for 2 seconds to return to the desired speed.

In the case of Fuzzy and Fuzzy-PID controllers as shown in the figures 5.18 and 5.19 respectively, the speed response doesn't show any noticeable overshoot and the deviation run for 0.24 second which is a short time compared to PI and PID controllers.

5.2.3. Fixed reference speed and step change in load torque

In the third step of simulation studies, the dynamic performances of the PI, PID, Fuzzy, and Fuzzy-PID controllers are analyzed by applying step change in load torque, 50 N.m to 350 N.m, at 3rd second interval time for the controllers. The figures 5.20 to 5.23 show the simulation results of the controllers.

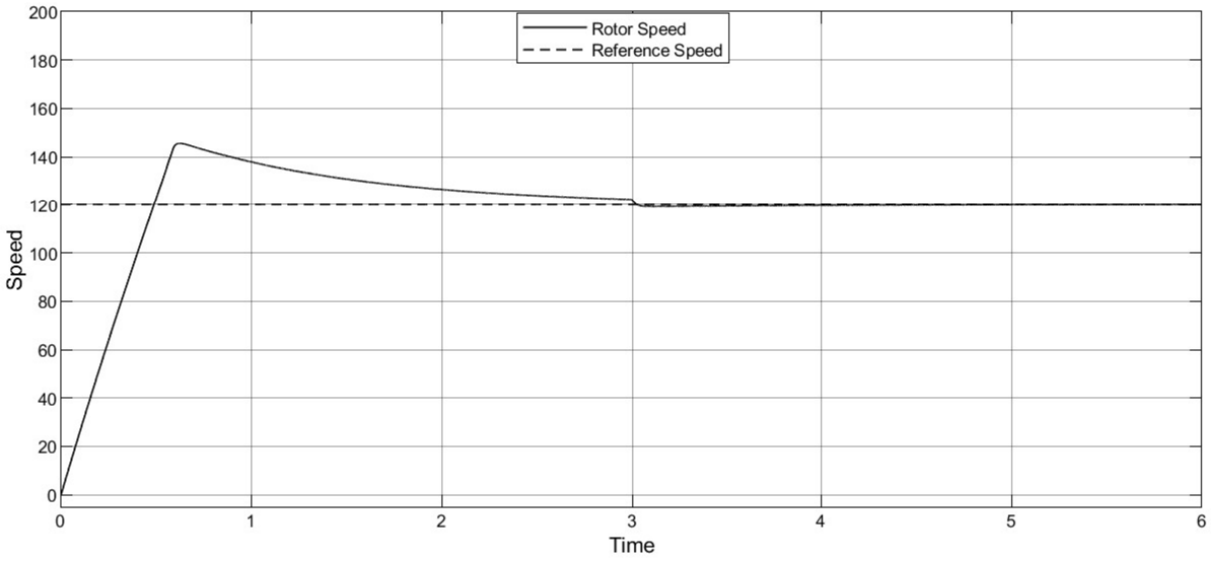


Figure 5.20: Speed response curve of IM at 120 rad/sec reference speed and 50 to 350 N.m step change in load torque using PI controller

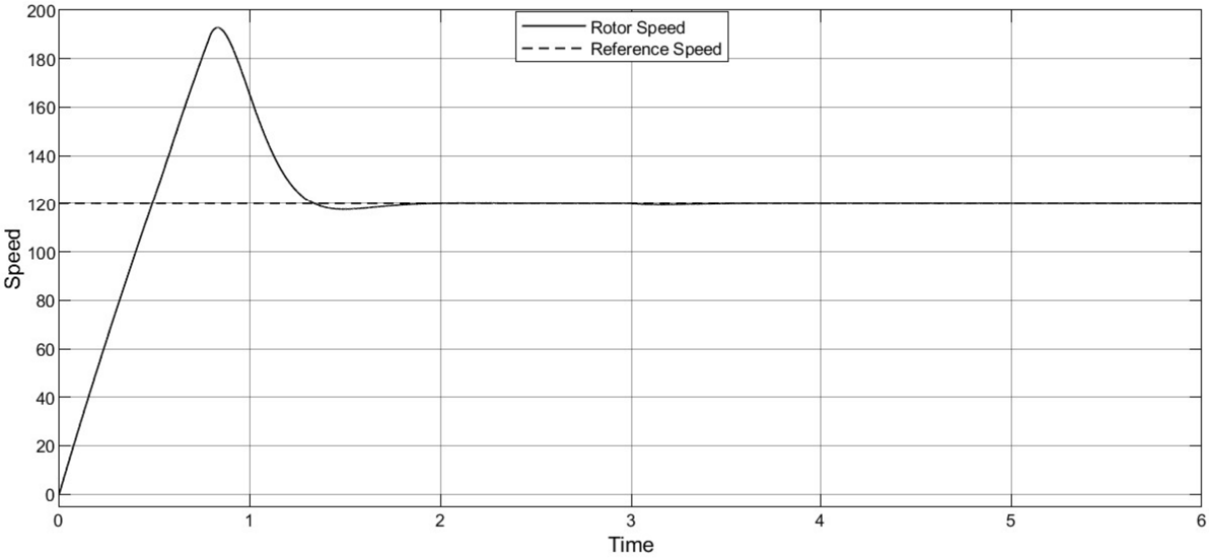


Figure 5.21: Speed response curve of IM at 120 rad/sec reference speed and 50 to 350 N.m step change in load torque using PID controller

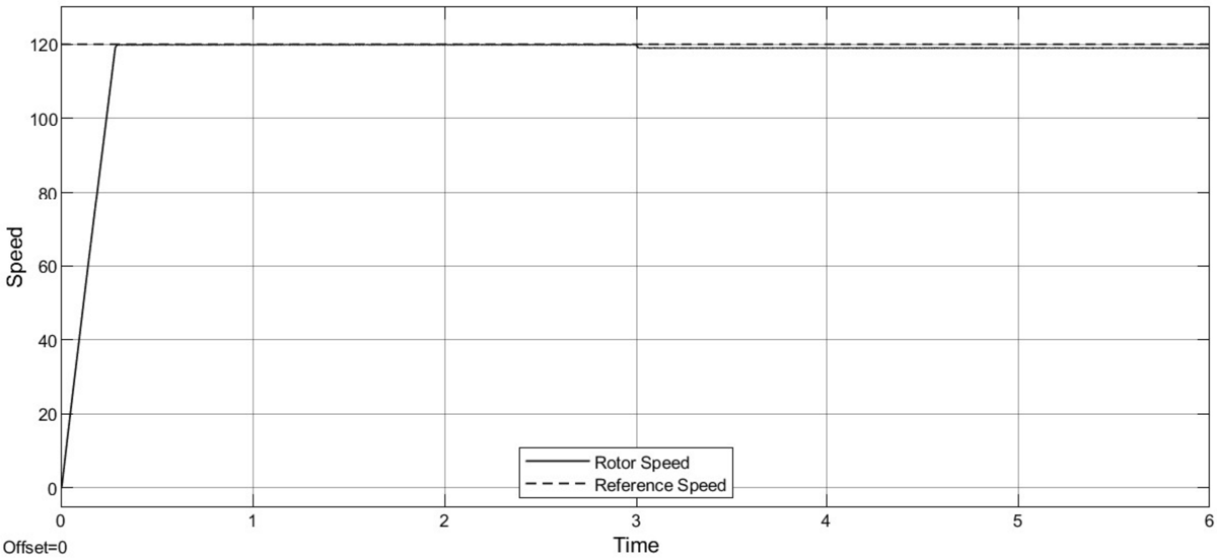


Figure 5.22: Speed response curve of IM at 120 rad/sec reference speed and 50 to 350 N.m step change in load torque using Fuzzy controller

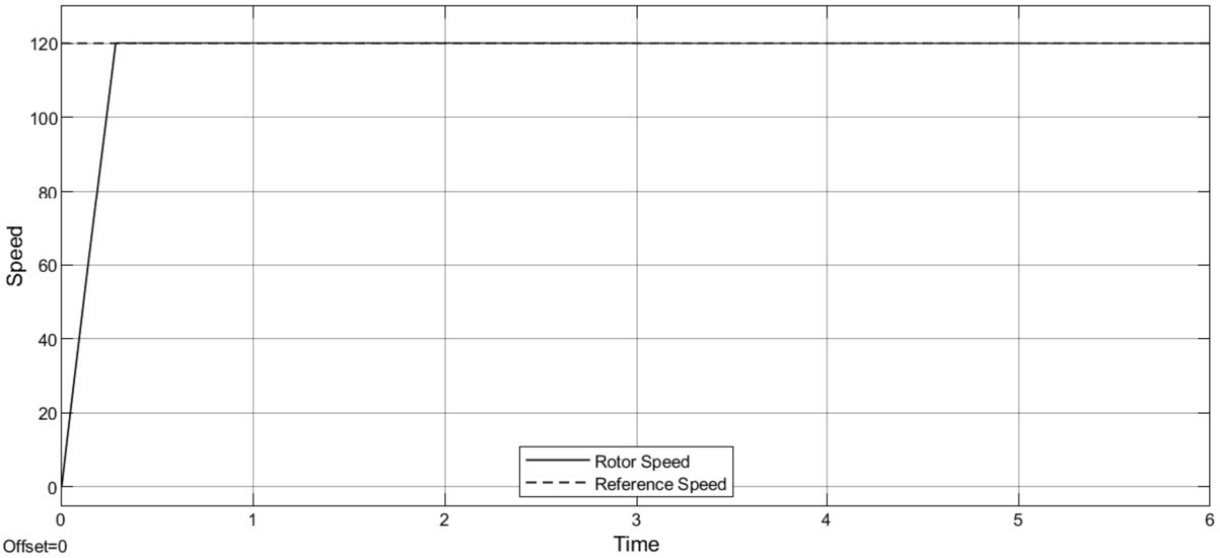


Figure 5.23: Speed response curve of IM at 120 rad/sec reference speed and 50 to 350 N.m step change in load torque using Fuzzy-PID controller

From figure 5.20, it is clear that the motor speed in PI controller is trying to follow the speed reference (120 rad/sec). It is also obvious that the response curve shows a noticeable undershoot and deviation when a step change in load torque is applied to the motor at the 3rd second. After that, the deviation runs for a couple of seconds until it returns to the desired speed reference.

From figure 5.21, it is clear that the motor speed in PID controller is following the speed reference (120 rad/sec). It is also obvious that the response curve shows a very small undershoot and deviation when a step change in load torque is applied to the motor at the 3rd second. After that, the deviation disappears in 0.3 seconds and it returns to the desired speed reference.

When using Fuzzy controller, the speed response as shown in figure 5.22 shows a noticeable undershoot and deviation when a step change in load torque is applied to the motor at the 3rd second. After that, the deviation doesn’t disappear it never returns to the desired speed reference which will increase the steady state error.

When using Fuzzy-PID controller, the speed response as shown in figure 5.23 doesn’t show any undershoot or deviation except small change in steady state error.

General Conclusion and future work

This project has successfully presented a Fuzzy-PID system for controlling a three-phase squirrel cage induction motor. Hybridization of fuzzy logic and conventional controllers is used as a single controller. Also, the indirect field-oriented control is used in the proposed system to solve the induction motor coupling effect problem that makes the system response slow and easily prone to instability.

PI, PID, Fuzzy, and Fuzzy-PID were designed and simulated in SIMULINK. The performance and robustness of all controllers have been evaluated under realistic operating conditions. Furthermore, a comparative study of the different control schemes has been completed using the performance significant measures such as rise time (tr), peak overshoot (Mp), settling time (ts), and steady state error (Ess).

Based on simulation results verification, it is concluded that dynamic response characteristics with the Fuzzy-PID controller take less time to settle and do reach the final steady state value especially when compared with the PI conventional controller. Also, it shows better robustness during the transient period when there is a step change and during the sudden load changes compared with PI controller.

In conclusion, we demonstrated that the proposed Fuzzy-PID speed controller for closed loop control of the induction motor drive system has an improved performance over the PI conventional controller, it gives better speed response, shows higher levels of robustness, and effectiveness.

Future work should include applying the proposed method to real time system and conduct full analysis of other power quality issues such as harmonic distortion, voltage imbalance, and power factor improvements. Additionally, system protection should take into consideration both under and over voltage conditions. Finally, it is essential to comment on the fact that induction motor working at rated flux will give optimum transient response, however, at lighter loads, excessive core loss can occur resulting in a very low efficiency. Therefore, future work should improve the motor efficiency by controlling the flux and thus allow for obtaining a balance between the copper and iron losses.

BIBLIOGRAPHY

[1] Ashok Kusagur, Shankarappa FakirappaKodad, and Sanker Ram, "Modelling & Simulation of an ANFIS controller for an AC drive," *World Journal of Modelling and Simulation*, vol. 8, no. 1, pp. 36-49, March 2011.

[2] Amit Mishra and Zaheeruddin, "Design of Speed Controller for Squirrel-cage Induction Motor using Fuzzy logic based Techniques," *International Journal of Computer Applications*, vol. 58, no. 22, pp. 10-18, November 2012.

[3] Pabitra Kumar Behera, Manoj Kumar Behera, and Amit Kumar Sahoo, "Comparative Analysis of scalar & vector control of Induction motor through Modeling & Simulation," *International Journal of Innovative Research in Electrical, Electronics, Instrumentation and Control Engineering*, vol. 2, no. 4, pp. 1340-1344, April 2014.

[4] ChChengaiah and Silva Prasad, "Performance of Inductoin Motor Drive by Indirect vector controlled method using PI and Duzzy Controllers," *International Journal of Science, Environment*, vol. 2, no. 3, pp. 475-469, 2013.

[5] ShardaPatwa, "Control of starting current in three phase Induction motor using Fuzzy logic controller," *International Journal of Advanced Technology in Engineering and Science*, vol. 1, no. 12, pp. 2732, December 2013.

[6] Ashok Kusagur, S. F. Kodad, and B. V. Sankarram, "Modelling of Induction Motor & Control of Speed Using Hybrid Controller Technology," *Journal of Theoretical and Applied Information Technology*, 2009.

[7] Amanulla and Manjunath Prasad, "Artificial Neural Network Based Speed and Torque Control of Three Phase Induction Motor," *International Journal of Science and Research (IJSR)*, vol. 2, no. 8, pp. 462-465, August 2013.

[8] http://en.wikipedia.org/wiki/Induction_motor. [Online].

[9] Novotny, Donald W. Lipo, and A. Thomas, *Vector control and dynamics of AC Drives*, Oxford University Press, New York, 1996.

[10] Andrzej M. Trzynadlowski, *Contorl of Induction Motors*, Elsevier Science Publishing Co Inc, San Diego, 2001.

[11] L. A. Zadeh, “Fuzzy sets,” *Information and Control*, vol. 8, pp. 338-353, 1965.

[12] Surajit Chattopadhyay, MadhuchhandaMitra, and Samarjit Sengupta, *Electric Power Quality*, Springer Science, New York, 2011.

[13] Frédéric Danion and Mark L. Latash, *Motor Control: Theories, Experiments, and Applications*, Oxford University Press US, 2010.

[14] Rakesh Singh Lodhi and Payal Thakur, "Performance & Comparison Analysis of Indirect Vector Control of Three Phase Induction Motor," *International Journal of Emerging Technology and Advanced Engineering*, vol. 3, no. 10, pp. 716-724, October 2013.

[15] Bemal Kumar Bose, *Modren Power Electronics and Ac Drives*, *1st ed*, Prentice Hall Inc, 2002.

[16] Texas Instruments Europe, *Field Orientated Control of 3-Phase AC-Motors*, Literature Number: BPRA 073, February 1998.

[17] Bilal Akin and Manish Bhardwaj, "Sensored Field Oriented Control of 3-Phase Induction Motors," *Texas Instruments Dallas*, Application Report SPRABP8, 2013.

[18] P. Fedor and D. Perduková, "A Simple Fuzzy Controller Structure," *ActaElectrotechnica et Informatica*, vol. 5, no. 4, pp. 1-4, 2005.

[19] Eiji Mizutani, Jyh-Shing Roger Jang, and Chuen-Tsai Sun, *Neuro-Fuzzy and Soft Computing*, Pearson Education Pte Ltd, chap. 2-3-4, September 1996.

[20] R. Ouiguini et al, "Speed Control of an Induction Motor using the Fuzzy logic approach," *International Symposium on Industrial Electronics (ISIE'97)*, Guimariies - Portugal, vol. 3, pg. 1168–1172, 1997.