Addis Ababa University

Addis Ababa Institution of Technology

Department of Electrical and Computer Engineering

Industrial control engineering

DESIGN AND COMPARE ADAPTIVE NEURO-FUZZY INFERENCE SYSTEM (ANFIS) WITH GENETIC ALGORITHM (GA) TUNNED PI CONTROLLER FOR SPEED CONTROL OF VECTOR CONTROLLED INDUCTION MOTOR DRIVE.

BY

DANIEL AREGA

A thesis submitted to the school of Electrical and Computer Engineering of Addis Ababa University in partial fulfillment of the requirements for the degree of Master of Science in Control Engineering.

Advisor: Dereje Shiferaw (Ph.D.)

November, 2018

Addis Ababa, Ethiopia

Addis Ababa University

Addis Ababa Institution of Technology

School of Electrical and Computer Engineering

DESIGN AND COMPARE ADAPTIVE NEURO-FUZZY INFERENCE SYSTEM (ANFIS) WITH GENETIC ALGORITHM (GA) TUNNED PI CONTROLLER FOR SPEED CONTROL OF VECTOR CONTROLLED INDUCTION MOTOR DRIVE.

By

Daniel Arega

Approval by Board of Examiners

_____	_____
Chairman, Dept. Graduate Committee	Signature
Dr. Dereje Shiferaw	_____
Advisor's Name	Advisor Signature
_____	_____
Internal Examiner	Signature
_____	_____
External Examiner	Signature

DECLARATION

I, the undersigned, declare that this thesis is my original work, has not been presented for a degree in this or other universities, all sources of materials used for this thesis work have been fully acknowledged.

Name: Daniel Arega

Signature: _____

Place: Addis Ababa Institute of Technology, Addis Ababa University, Addis Ababa, Ethiopia

This thesis has been submitted for examination with my approval as a university advisor

Dereje Shiferaw (Ph.D.) _____

Advisor's Name Signature

ACKNOLODGEMENT

First of all, with the deepest gratitude I wish to thank my advisor Dr. Dereje Shiferaw for his boundless effort, Golden time, insightful comments, being patient for correcting every movement errors throughout the accomplishment of my work.

I would also thank my mother who is responsible for the beginning of my life and its true journey and who illuminates every breath I take through her existence.

Finally, my gratitude also goes to the many of my courageous friends, in particular for Mr. Getenew Dessalegn and Sefinew Getnet who are always by my side in editing and transforming my words brilliantly with unlimited support from the beginning of my work to end.

Table of Contents

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS AND SYMBOLS

AC	Alternating Current
AI	artificial Intelligence
ANFIS	Adaptive Neuro-Fuzzy Inference System
ANN	Artificial Neural Network
BPM	Back Propagation Method
DC	Direct Current
DTC	Direct Torque
DFOC	Direct Field-Oriented Control
FOC	Field-Oriented Control
FLC	Fuzzy Logic Control
GUI	Graphical User Interface
GA	Genetic Algorithm
Hz	Hertz
IFOC	Indirect Field-Oriented Control
IM	Induction Motor
IVCIM	Indirect Vector controlled Induction Motor
KW	Kilo Watts
LSE	Least Square Error
NFC	Neuro-Fuzzy control
PI	Proportional Integral
SVPWM	Space Vector Pulse Width Modulation
SCIM	squirrel cage induction motor

*	indicates reference or command value
Z-N	Ziegler-Nicholas
Tr	rotor time constant (Lr/Rr)
Ls, Lr	stator and rotor inductance
Lm	mutual inductance
Lls, Llr	leakage inductances of stator and rotor
Rs, Rr	stator and rotor resistance

ABSTRACT

Nowadays, vector controlled induction motor drives with variable speed applications are widely used in order to achieve good dynamic performance and wide speed control. The conventional speed controllers for vector control of induction motor drive suffer from the problem of stability; besides, these controllers such as PI/PID controllers show either steady state error or sluggish response to the agitation in reference setting or during load perturbation. In this thesis a new method of controlling technique based on the combination of Artificial Neural Network (ANN) and fuzzy logic (FL) is proposed to improve the speed control of indirect vector controlled induction motor drive. Indirect vector controlled induction motor with genetic algorithm (GA) optimized PI controller is developed and is replaced with adaptive neuro-fuzzy controller (ANFIS) to overcome the problem of overshoot occurred in PI controller and to obtain quick steady state response and better speed control. The proposed technique is implemented using MATLAB/Simulink. In this thesis, the speed, torque and stator current responses with GA based PI controller and proposed adaptive neuro-fuzzy controller are compared and found that the proposed ANFIS based controller showed increased dynamic performance. The proposed adaptive neuro-fuzzy controller is better in overshot which is 0.475% and that of PI controller is 14.368%, raise time and settling time.

Keywords: Induction Motor (IM), Vector Control, Genetic Algorithm (GA), ANFIS, FLC

CHAPTER ONE

1. INTRODUCTION

1.1 Background of Induction Motor

Electrical machines are a general term for electric motors, and electric generators. They are electromechanical energy converters i.e. an electric motor converts electricity to mechanical power while an electric generator converts mechanical power to electricity. So, any machine can be used as either a generator or a motor. These machines classified as direct current (DC) and alternating current (AC) machines based on the current type[1]. Electrical motors in the home run refrigerators, freezers, vacuum cleaners, blenders, air conditioners, fan and many similar appliances. In the workplace motors provide the motive power for almost all tools[2].

An induction motor (IM) is a type of asynchronous AC motor where power is supplied to the rotating device by means of electromagnetic induction. IM can be classified as cage type and wound type based on their rotor construction. The induction motor with a cage was invented by Mikhail Dolivo-Dobrovolsky about a year later in Europe[3]. These motors are now the preferred choice for industrial application due to their rugged construction, lower cost, absence of brushes (which are required in most DC motors). When operated directly from the ac line voltage, induction motor operates nearly at constant speed. However, by means of power electronic converters, it is possible to change the speed of an induction motor[4][5].

Even though induction motor (IM) is widely used in industry because of its advantages, unfortunately, the speed of an induction motor cannot be continuously varied without additional expensive equipment. High-performance control of an induction motor is more difficult than DC motors, because the induction motor is inherently nonlinear, multivariable time varying, and have internal coupling effect and so its stability analysis is very complex system [3] [23]. Also the control task is further complicated by the fact that the induction motor is subject to unpredictable disturbances (such as load changes) and there are uncertainties in machine parameters. So its speed control is not as straight forward as that of a dc motor [2][6]. With the fastest developments in power electronics, microprocessors and digital electronics, electric drives are now became more reliable, compact and efficient.

1.2 Statement of Problem

In many application areas, the most common mechanism to control the speed of IM is PI controller. But it has performance limitations. Now a days, there is also controlling mechanisms using artificial intelligence controllers like artificial neural network (ANN), fuzzy logic control (FLC) and also other controlling methods. But the use of FLC needs a lot of trial and error has to be carried out to obtain the desired response which is time consuming. On the other hand, ANN alone is insufficient if the training data are not enough to take care of all the operating modes.

Therefore, in order to overcome these problems and achieve high performance, it is better to use the combination of ANN and FLC which is called (ANFIS) for controlling speed of vector controlled IM. In this proposed thesis, GA based PI controller is replaced by adaptive neuro-fuzzy controller and comparing performances.

1.3 Literature Review

The research work carried out by various researchers in the field of modeling, control and implementation of speed control of IMs using various control strategies is presented in this section. Various researchers have worked on the speed control of IMs using various control techniques. Some of the techniques are the space vector pulse width modulation (SVPWM) method, the PI method, the sliding mode control method, the fuzzy logic control (FLC) method, the adaptive Neuro-Fuzzy inference system (ANFIS) method, and artificial neural network (ANN).

The classical or conventional type of control (i.e. Using PI controller) is used in most of the electrical motor drives. It requires exact mathematical model to control the system. When there are system parametric variations, the behavior of the system is unsatisfactory and it deviates from the desired performance[7]. This control method is not satisfactory because the PI gains are selected by using trial and error.

Design of FLC based self-tuning proportional integral controller for control of speed in IMs was addressed by Ambarisha Mishra,Vasundhara Mahajan,P. Agarwal,in [8]. The tuning of the conventional proportional integral controller was obtained using the fuzzy rules obtained from tests. A number of operating conditions were considered in the controller design. Some of the operating conditions were steep change in load torque, speed reversion, decrease or increase in rotor resistance, change in the inertia of the system or self-inductance of the system.

Gauri V. Deshpande and S.S.Sankeshwari [9] presented a comparative study of PI, FL, Fuzzy PI and hybrid speed controllers for vector control of IM drives in their research paper. They used an indirect vector-controlled strategy for the control of current-controlled voltage source inverters. They studied the responses for starting, speed reversal and load perturbations.

However, the fuzzy method suffers from certain disadvantages. To design FLC, it needs many trial and errors to gate suitable rules, number and type of membership functions. The rules are written on the basis of experience of the observer and are random in nature. These rules may not be adequate in those circumstances. Hence, the result obtained from the fuzzy controller may not be optimal. Furthermore, there are no specific design tools to develop the control strategy due to the non-availability of numerical and analytical methods to tackle the control problem. A simple FLC has a narrow speed operation and needs much more manual adjustment by trial and error if high performance is needed. On the other hand, it is extremely tough to create a serial of training data for artificial neural network (ANN).

ANN controller is one of the intelligent controller, which is usually utilized for two purposes: for constructing non-linear controllers and for adding human intelligence to controllers, such as perception (sensory information process), understanding, recognition, inference, learning, diagnosis and others. Kung and Liaw[10] developed an adaptive speed control of drives using neural networks in their paper. The effectiveness of the proposed controller was confirmed by some simulated and experimental results.

However, NN does not have any knowledge of the system (adaptability) though it has the learning capability. ANN is not enough if wide speed range is required.

Rajesh Kumar R. A., Gupta Rajesh S., and Surguse [11] [12]studied about the control of speed of vector controlled IM using ANFIS. The authors selected the PI parameters by trial and error method. This leads to get less performance controller and consuming time.

In this thesis, the PI parameters are selected by using Genetic algorithm tuning method to get optimized parameters. And then the conventional PI controller is replaced by Adaptive Neuro-Fuzzy Inference System (ANFIS), which tunes the fuzzy inference system with hybrid learning algorithm. This makes fuzzy system to learn. The performance of the proposed neuro-fuzzy based vector controlled induction motor drives are investigated at different operating conditions.

1.4 Objectives of the study

1.4.1 General objective

The main objective is to control the speed of vector controlled induction motor by using adaptive neuro-fuzzy inference system (ANFIS) and compare the result with GA tuned PI controller.

1.4.2 Specific objectives

The specific objectives are:

- To understand and analyze the dynamic model of IM.
- To design and implement GA based PI controller.
- To design and implement ANFIS for IM speed control using MATLAB/SIMULINK.

1.5 Methodology

The overall block of speed control of vector controlled induction motor using adaptive neuro-fuzzy inference system (ANFIS) is shown in Figure 1.2. In this thesis work, indirect field oriented control scheme is used for speed control of induction motor. Due to effect of parameters variation on IFOC speed control of induction motor, the artificial intelligent controller (ANFIS) is designed properly. To do this work, the first task is describing the statement of the problem and then literature survey then define the objectives of the research. Driving IFOC models and decoupling of the induction motor model was the next task, the simulation carried out for MATLAB/Simulink and analysis the performance of Artificial Intelligence (AI) for speed control of induction motor was next task. Finally, formulate the conclusion and recommendation of the thesis based on the Simulink result. The overall methodology that we followed is described by using the flow chart shown in figure 1.1 below.

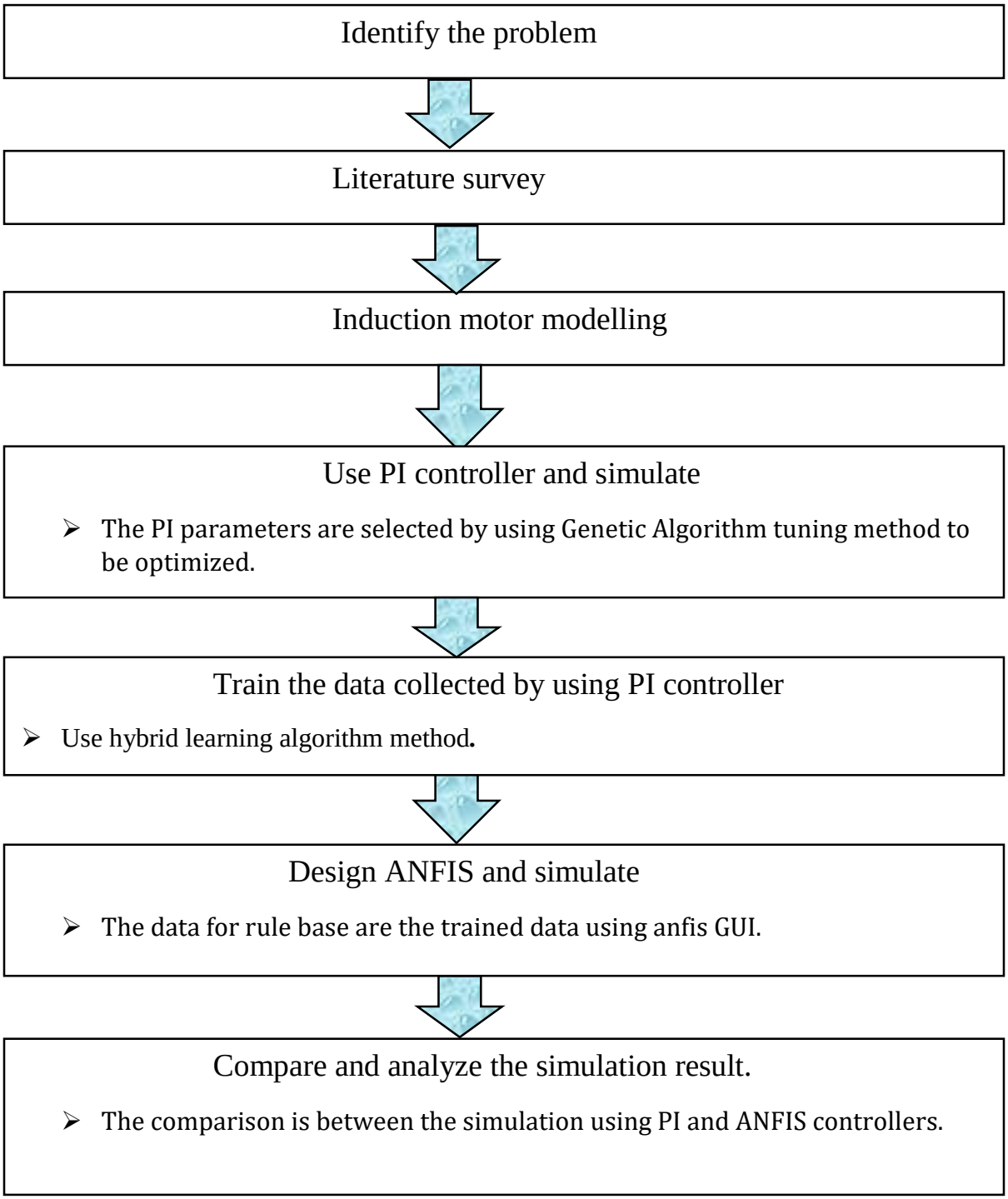


Figure 1-1: General Methodology Flow Chart

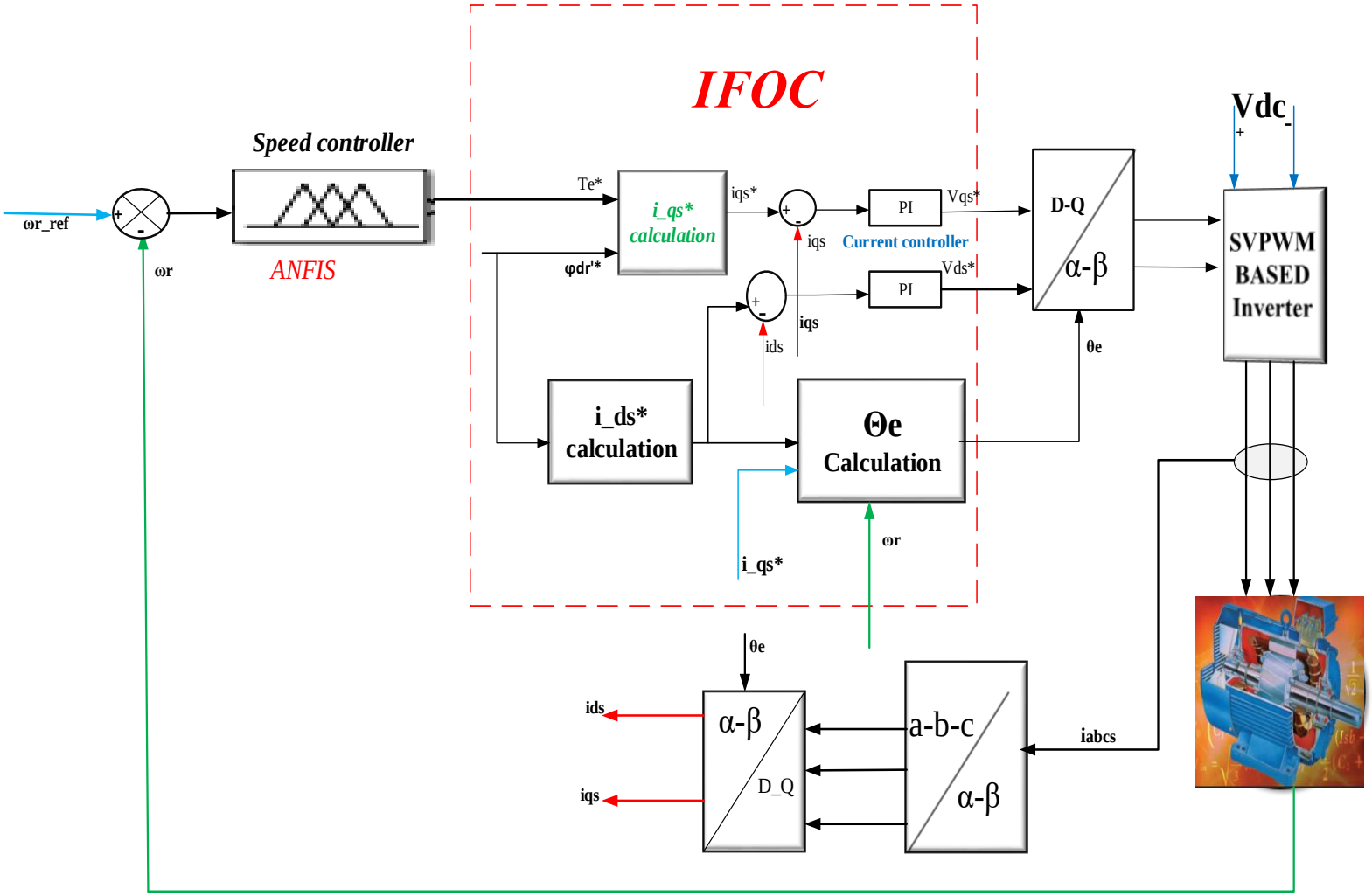


Figure 1-2: General schematic of ANFIS based speed control of IVCIM drive system

1.6 Scope of the study

The scope is limited to comparing the performance of the GA based PI controller and the intelligence ANFIS controller by using MATLAB/ SIMULINK. The simulation included the transient and steady state performances at different operating conditions.

1.7 Organization of Thesis

This thesis has been organized into five chapters. Chapter one presents about background of three phase induction motor, statement of the problem, objectives of the study, the methodology and literature review leading towards the completion of the thesis. The second chapter discusses about dynamic modeling of induction motor which includes reference frames and axis transformations, stator and rotor based reference frame model of induction machine. Chapter three presents the control of on induction machine (scalar control, vector control (direct and indirect control)), design of GA based PI controller and the principle of artificial intelligent controllers specifically about ANFIS. The fourth chapter is about simulation results obtained using MATLAB/SIMULINK and discussions of the results. The final chapter is talking about conclusions and recommendations.

CHAPTER TWO

2. THREE-PHASE INDUCTION MOTOR MODELLING

2.1 Introduction

In this chapter, the induction motor modeling is discussed. The chapter starts with the introduction of squirrel cage induction motor and then the reference frames. Then the co-ordinate transformations (Clarke and Park Transformations) are introduced. Finally, induction machine model based on an induction machine equivalent circuit in a synchronously rotating reference frame d-q axis is presented.

2.1.1 The squirrel cage induction motor

Induction motors[4] with squirrel-cage rotors are the workhorses of industry because of their low cost and rugged construction. There are many varieties of induction motors. Single-phase induction motors are used in low power ratings (fractional kW to a few kW) in applications where their speed does not have to be controlled in a continuous manner. Wound-rotor induction generators are used in large power ratings (300 kW and higher) for wind-electric generation. However, our focus for this thesis is on three-phase, squirrel-cage induction motors, which are commonly used in adjustable-speed applications, has a simple construction, low-cost, and rugged nature.

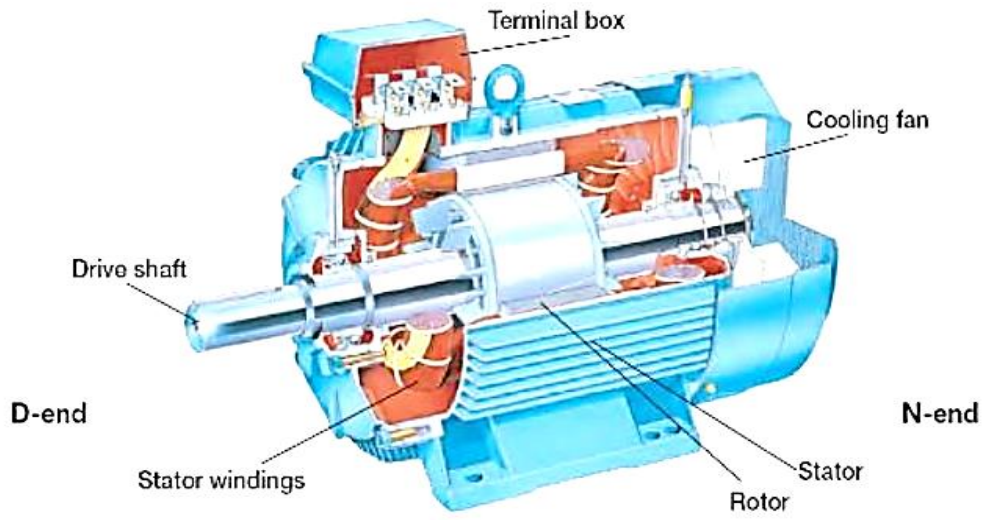


Figure 2-1: Squirrel cage induction motor [10]

The stator of an induction motor consists of three-phase windings, sinusoidally distributed in the stator slots. These three windings are displaced by 120° in space with respect to each other, as shown in fig 2.2a below.

The rotor consisting of a stack of insulated laminations, has electrically conducting bars of copper or aluminum inserted (molded) through it, close to the periphery in the axial direction. These bars are electrically shorted at each end of the rotor by electrically conducting end-rings, thus producing a cage-like structure, as shown in Fig 2-2(b) below.

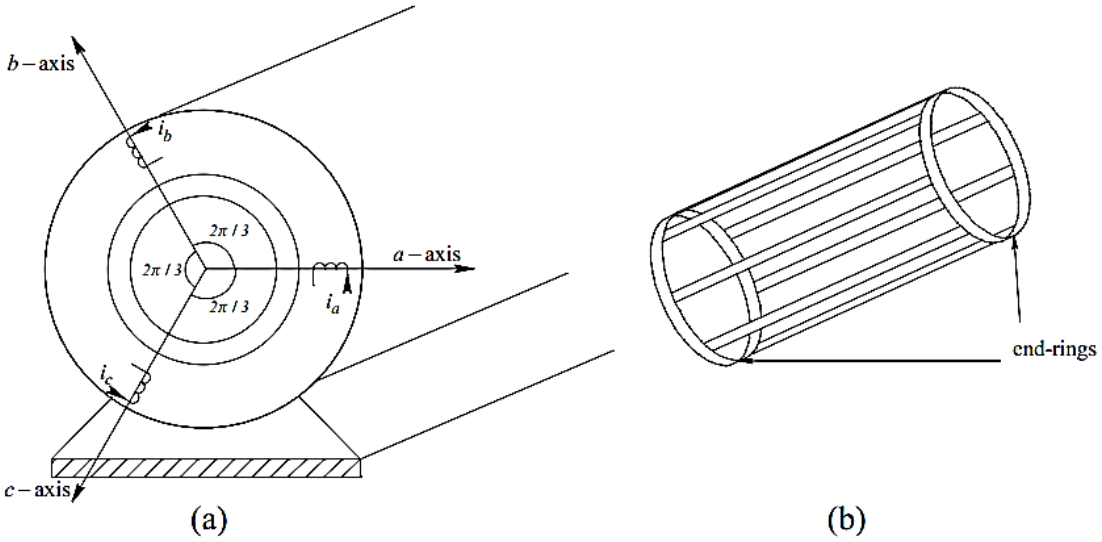


Figure 2-2: a) stator b) rotor of squirrel IM [3]

The rotor of the squirrel cage IM is composed of the laminated core and rotor bars. The rotor bars are embedded in slots inside the rotor laminations and are shorted on both ends by end rings. When the stator winding is connected to a three-phase supply, a rotating magnetic field is generated in the air gap. The rotating field induces a three-phase voltage in the rotor bars. Since the rotor bars are shorted, the induced rotor voltage produces a rotor current, which interacts with the rotating field to produce the electromagnetic torque[13]. As a sinusoidal excitation is applied to the windings of the stator, a sinusoidal flux is established in the air gap rotating at a speed given by

$$Ns = \frac{120f}{p} \qquad (2.1)$$

Where Ns is the speed in rpm, f is the frequency of the applied excitation in Hz, and P is the number of poles. The rotor attempts to turn at this speed, also known as the synchronous speed, but never quite reaches this speed, rotating instead at a speed Nr due to slip.

2.2 Reference frames

Since the voltages describing the machine are time variant and some of the machine inductances are function of rotor speed, the model of IM is complex compared to that of DC motor. A change of variables is often used to reduce the complexity of machine differential equations and inductances[14]. During start-up and other severe motor operations, the induction motor draws large currents, produces voltage drops, oscillatory torque and can even generate harmonics in the power system. It is therefore important to be able to model the induction motor in order to predict these phenomena. Various models have been developed, and the d-q or two-axis model for the study of transient behavior has been well tested and proven to be reliable and accurate. There are three preferred reference frames as follows[15].

- The stationary reference frame when the d-q axes don't rotate.
- The synchronously rotating reference frame when the d-q axes rotate at synchronous speed.
- The rotor reference frame when the d-q axes rotate at rotor speed.

For this thesis, synchronously rotating reference frame is selected for modeling of induction motor because it is simple for simulation using computer.

2.2.1 $\alpha - \beta$ Transformation

This transformation technique is called Clarke transformation. The transformation of (a- b- c) axis to alpha-beta axis is derived using the transformation axis shown in the figure 2.4 below. The following fig.2-3 shows the invers and forward transformation of Clarke and Parke transformation.

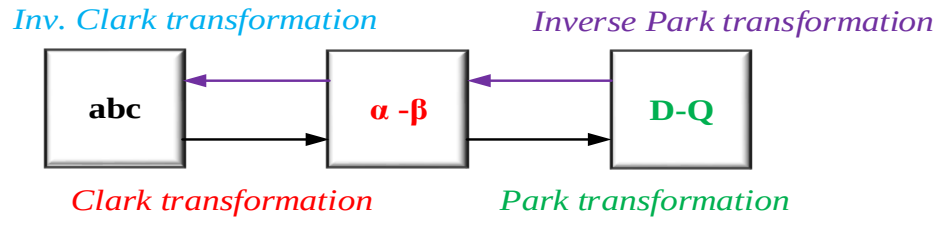


Figure 2-3: Clark and Park transformations

$$\begin{bmatrix} f\alpha \\ f\beta \end{bmatrix} = \begin{bmatrix} 1 & -1/2 & -1/2 \\ 0 & \sqrt{3}/2 & -\sqrt{3}/2 \end{bmatrix} \begin{bmatrix} fa \\ fb \\ fc \end{bmatrix} = \begin{bmatrix} cos\theta & -sin\theta \\ sin\theta & cos\theta \end{bmatrix} \begin{bmatrix} fd \\ fq \end{bmatrix} \quad (2.2)$$

Where f can be voltage, current, etc. and θ is the rotor flux position.

2.2.2 D_Q transformation

Direct Quadrature (d-q) transformation is a mathematical transformation used to simplify the analysis of three phase circuit. In the case of balanced three phase circuits, application of d-q transformation reduces the three AC quantities to two quantities[16]**.**

In order to transform the three-phase current/voltage system (a- b-c) to an arbitrarily chosen two-phase reference frame involving two orthogonal windings, the so-called Park transformation is used. A Change of variables that formulates a transformation of 3-phase variables of stationary circuit elements to two D-Q axis may be expressed as[14]:

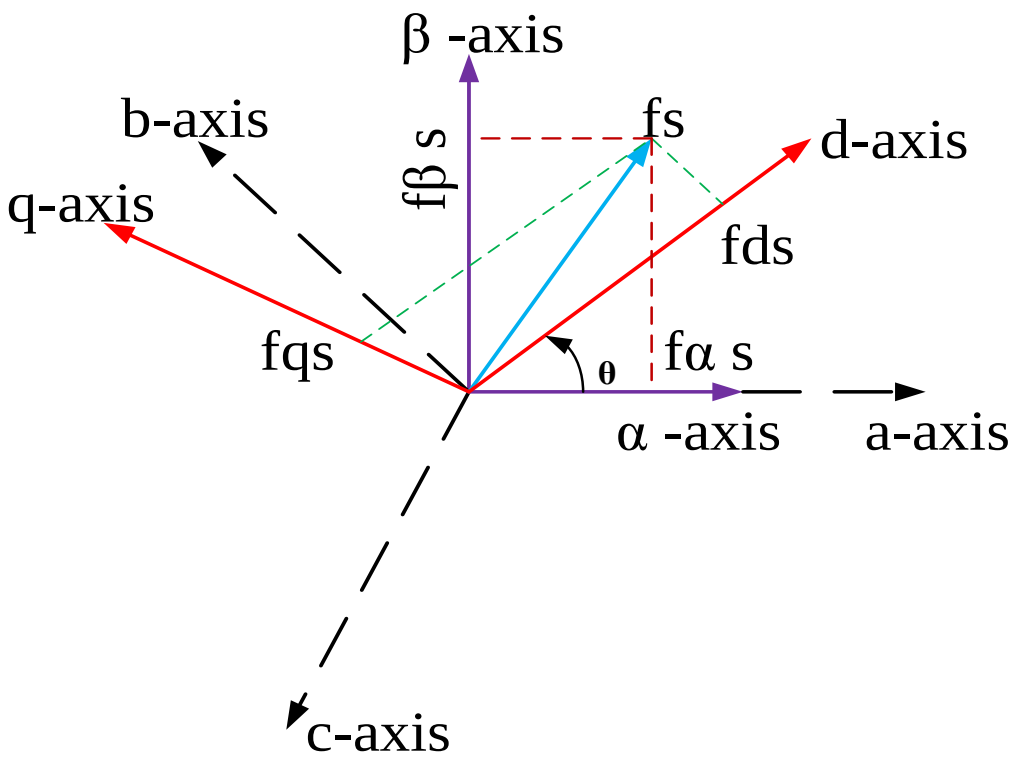


Figure 2-4: Transformation by trigonometric relationships [12]

$$fdqos = Ks * fabcs \quad (2.3)$$

Where: $fqdos = [fqs\ fds\ fos]'$ and $fabcs = [fas\ fbs\ fcs]'$ where $'$ indicates the transpose of the matrix and f can be voltage, current, flux etc.

$$Ks = \frac{2}{3} * \begin{bmatrix} \cos(\theta) & \cos\left(\theta - 2 * \frac{pi}{3}\right) & \cos\left(\theta + 2 * \frac{pi}{3}\right) \\ \sin(\theta) & \sin\left(\theta_2 - 2 * \frac{pi}{3}\right) & \sin\left(\theta + 2 * \frac{pi}{3}\right) \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \end{bmatrix} \tag{2.4}$$

When determining the equivalent two phase machine field components from the three phase fields, we must reduce the overall magnitude by a factor of 2/3.Using the transformation matrix, Ks the voltage and the current equations in the D-Q axis becomes:

$$\begin{bmatrix} vds \\ vqs \\ vos \end{bmatrix} = \frac{2}{3} * \begin{bmatrix} \cos(\theta) & \cos\left(\theta - 2 * \frac{pi}{3}\right) & \cos\left(\theta + 2 * \frac{pi}{3}\right) \\ \sin(\theta) & \sin\left(\theta_2 - 2 * \frac{pi}{3}\right) & \sin\left(\theta + 2 * \frac{pi}{3}\right) \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \end{bmatrix} * \begin{bmatrix} vas \\ vbs \\ vcs \end{bmatrix} \tag{2.5}$$

$$\omega = \frac{d\theta}{dt} \tag{2.6}$$

In balanced three phase operation, the zero component of both the current and voltage is zero. i.e. i0=V0=0.

$$\begin{bmatrix} ids \\ iqs \end{bmatrix} = \frac{2}{3} * \begin{bmatrix} \cos(\theta) & \cos\left(\theta - 2 * \frac{pi}{3}\right) & \cos\left(\theta + \frac{pi}{3}\right) \\ \sin(\theta) & \sin\left(\theta - 2 * \frac{pi}{3}\right) & \sin\left(\theta + 2 * \frac{pi}{3}\right) \end{bmatrix} * \begin{bmatrix} ias \\ ibs \\ ics \end{bmatrix} \tag{2.7}$$

If the direct axis is override on the a-axis, then the angle θ is zero. Therefore the current equation becomes.

$$\begin{bmatrix} ids \\ iqs \end{bmatrix} = \frac{2}{3} * \begin{bmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & -\frac{\sqrt{3}}{2} & \frac{\sqrt{3}}{2} \end{bmatrix} \begin{bmatrix} ias \\ ibs \\ ics \end{bmatrix} \tag{2.8}$$

This can be checked by using the MATLAB/SIMULINK as shown in figure 2.5 below. The Simulink block which represents the transformation is shown in Appendix B.

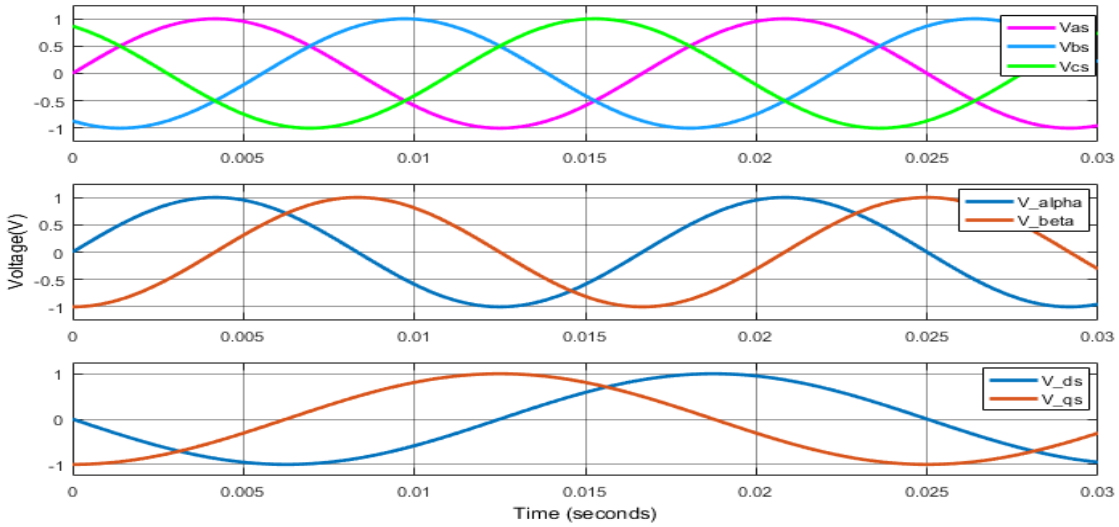


Figure 2-5: Simulation diagram of Clark and parks transformation of Vabcs.

The following values are used for the Simulink model. For all voltages (Vas, Vbs, Vcs), the amplitude is one and the frequency is 50Hz and also the phase is 0,-2*pi/3 and 2*pi/3 respectively. The angle output value is between [0 2*pi] and the time value for the angle is [0 0.01]. Transformation procedure is similar as the voltage for both current and flux.

The inverse parks transform is

$$fabcs = Ks^{-1} fdq0s \tag{2.9}$$

Where: $Ks^{-1} = \frac{2}{3} * \begin{bmatrix} \cos(\theta) & \sin(\theta) & 1 \\ \cos\left(\theta - 2 * \frac{pi}{3}\right) & \sin\left(\theta - 2 * \frac{pi}{3}\right) & 1 \\ \cos\left(\theta + 2 * \frac{pi}{3}\right) & \sin\left(\theta + 2 * \frac{pi}{3}\right) & 1 \end{bmatrix}$

2.3 Space vector representation of IM

In developing the IM space-vector model, it is assumed that (1) the induction generator is symmetrical in structure and three-phase balanced, and (2) the magnetic core of the stator and rotor is linear with negligible core losses. The IM space-vector model is generally composed of three sets of equations: voltage equations, flux linkage equations, and motion equation [3, 4].

$$\begin{cases} \overrightarrow{vs} = Rs\overrightarrow{is} + p\overrightarrow{\varphi s} + j\omega e\overrightarrow{\varphi s} \\ \overrightarrow{vr} = Rr'\overrightarrow{ir} + p\overrightarrow{\varphi r} + j(\omega e - \omega r)\overrightarrow{\varphi r} \end{cases} \tag{2.10}$$

$$\begin{cases} \overrightarrow{\varphi s} = (Lls + Lm)\overrightarrow{\imath s} + Lm\overrightarrow{\imath r} = Ls\overrightarrow{\imath s} + Lm\overrightarrow{\imath r} \\ \overrightarrow{\varphi r} = (Llr + Lm)\overrightarrow{\imath r} + Lm\overrightarrow{\imath s} = Lr\overrightarrow{\imath r} + Lm\overrightarrow{\imath s} \end{cases} \quad (2.11)$$

$$\begin{cases} J\dfrac{d\omega r}{dt} + F\omega r = Te - Tl \\ Te = \dfrac{3P}{2} Re\{j\overrightarrow{\varphi s}\overrightarrow{\imath s}\} = -\dfrac{3P}{2} Re\{j\overrightarrow{\varphi r}\overrightarrow{\imath r}\} \end{cases} \quad (2.12)$$

The above equations are voltage, flux and motion equations respectively. Where s and r subscripts indicate stator and rotor axis, and p=d/dt, P is number of poles, Re indicates the real part. Rs and Rr are stator and rotor resistances respectively, vs and vr represents stator and rotor voltage , $\varphi s, \varphi r$ stands for stator and rotor fluxes, Lls and Llr represents leakage inductamces of stator and rotor and Lm is mutual inductance, is and ir are rotor and stator currents, Te and Tl represents electromechanical and Load torques respectively, ωr is rotor speed, J, F indicates the inertia and coefficient of static friction respectively, j stands for complex number. The arrow is for indication the vector.

2.4 Two axis dynamic model of three phase induction motor

Dynamic modelling and simulation of induction motor drives are of great importance to both industry due to the prevalence of these types of drives in various industrial settings as well as in the validation of design process of the motor drive systems. The dynamic model of the induction motor is derived by transferring the three-phase quantities into two phase direct and quadrature axis quantities. In IM modeling, the following assumptions have been taken.

- 3-phase squirrel cage induction motor
- Core loss and skin effects are negligible
- Currents and voltages are sinusoidal
- Balanced and symmetrical operating conditions
- Synchronously rotating reference frame
- No magnetic saturation

Using the transformation methods and the assumptions taken, the model of IM is given as follows[14].

‘Stator voltage equations in d-q axis’.

$$Vqs = Rs.iqs + \omega e.\varphi ds + P\varphi qs$$

$$Vds = Rs.ids - \omega e.\varphi qs + P\varphi ds \quad (2.13)$$

$$Vos = Rs.i0s + P\omega 0s$$

'Rotor voltage equations'

$$Vdr' = 0 = Rr'idr' - (\omega e - \omega r)\varphi qr' + P\varphi dr'$$

$$Vqr' = 0 = Rr'iqr' + (\omega e - \omega r)\varphi dr' + P\varphi qr' \quad (2.14)$$

$$V0r' = Rr'i0r + P\varphi 0r'$$

In the case of a squirrel-cage induction motor, the rotor voltages, Vqr' and Vdr', are zero because of the rotor bars are shorted.

'Stator flux linkage equations'

$$\varphi qs = Lls\ iqs + Lm(iqs + iqr') = Lsiqs + Lmiqr'$$

$$\varphi ds = Lls\ ids + Lm(ids + idr') = Lsids + Lmidr' \quad (2.15)$$

$$\varphi 0s = Lls\ i0s$$

'Rotor flux linkage equations'

$$\varphi qr' = Llr'iqr' + Lm(iqs + iqr') = Lriqr' + Lmiqs$$

$$\varphi dr' = Llr'idr' + Lm(ids + idr') = Lridr' + Lmids \quad (2.16)$$

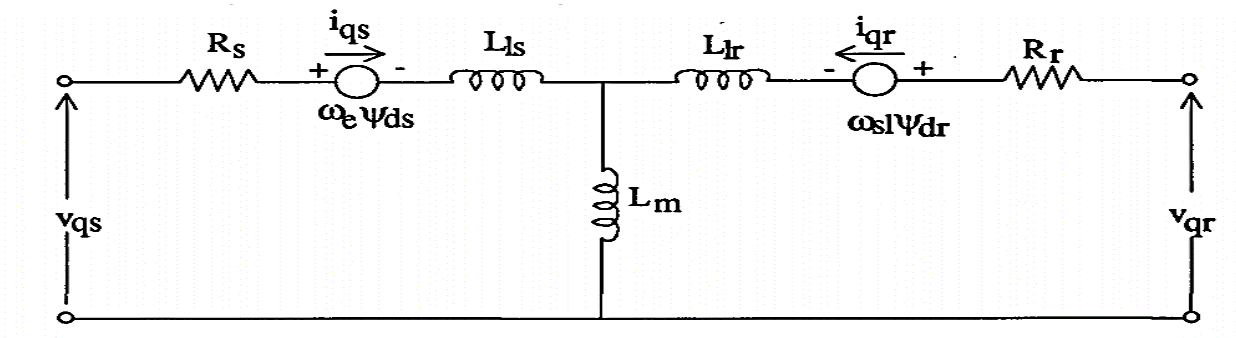
$$\varphi 0r' = Llr'i0r'$$

'Torque equations'

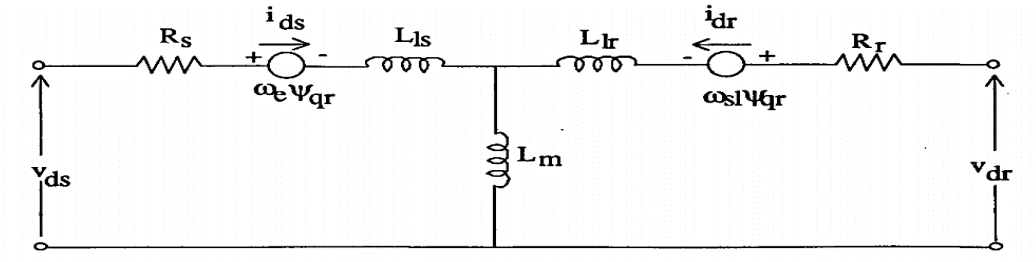
$$\mathrm{Te} = \frac{3}{2}\frac{\mathrm{p}}{2}(\varphi \mathrm{qr}'\mathrm{idr}' - \varphi \mathrm{dr}'\mathrm{iqr}') \quad (2.17)$$

$$Te = Tl + J\frac{d\omega r}{dt} + B\omega r$$

For the balanced three phase operation, the zero components of both current and voltage are zero.



(a) Quadrature component equivalent circuit



(b) Direct component equivalent circuit

Figure 2-6: D-Q dynamic equivalent circuit [15]

If the above voltage and flux linkage equations are combined and expressed in matrix form, it will result in the form as shown below.

$$\begin{bmatrix} \text{vqs} \\ \text{vds} \\ \text{vqr}' \\ \text{vdr}' \end{bmatrix} = \begin{bmatrix} \text{Rs pLs} & \omega\text{e Ls} & \text{pLm} & \omega\text{e LM} \\ -\omega\text{e Ls} & \text{Rs} + \text{pLs} & -\omega\text{e Lm} & \text{pLm} \\ \text{pLm} & \omega\text{sl Lm} & \text{Rr}' + \text{pLr}' & \omega\text{sl Lr}' \\ -\omega\text{sl Lm} & \text{pLm} & -\omega\text{sl Lr}' & \text{Rr}' + \text{pLr}' \end{bmatrix} * \begin{bmatrix} iqs \\ ids \\ iqr' \\ idr' \end{bmatrix} \tag{2.18}$$

Modeling of the voltage-input model is based on the fifth-order equation, which can be expressed in matrix form as follows.

$$\frac{d}{dt}\begin{bmatrix} ids \\ iqs \\ idr \\ iqr \\ \omega r \end{bmatrix} = \begin{bmatrix} -RsLr & \frac{P}{2}\omega rLm^2 & RrLm & \frac{P}{2} * \omega rLrLm & 0 \\ -\frac{P}{2}\omega rLm^2 & -RsLr & -\frac{P}{2} * \omega rLrLm & RrLm & 0 \\ RsLm & -\frac{P}{2} * \omega rLsLm & -RrLs & -\frac{P}{2} * \omega rLsLm & 0 \\ \frac{P}{2} * \omega rLsLm & RsLm & -\frac{P}{2} * \omega rLsLr & -RrLs & 0 \\ \frac{PLmLd^2}{3J} iqs & -\frac{PLmLd^2}{3J} ids & 0 & 0 & -\frac{F}{J} \end{bmatrix} \begin{bmatrix} ids \\ iqs \\ idr \\ iqr \\ \omega r \end{bmatrix}$$

$$+ \frac{1}{Ld^2}\begin{bmatrix} Lr & 0 & -Lm & 0 & 0 \\ 0 & Lr & 0 & -Lm & 0 \\ -Lm & 0 & Ls & 0 & 0 \\ 0 & -Lm & 0 & Ls & 0 \\ 0 & 0 & 0 & 0 & -\frac{Ld^2}{J} \end{bmatrix} \begin{bmatrix} vds \\ vqs \\ 0 \\ 0 \\ Tl \end{bmatrix} \tag{2.19}$$

Where $Ld = \sqrt{LsLr - Lm\text{^}2}$

CHAPTER THREE

3. CONTROL OF THREE PHASE INDUCTION MOTOR

3.1 Introduction

The control of an induction motor is considerably more complex than that of a DC motor. This is partly due to the complex dynamics of an AC machine [23]. Various scalar and vector control methods exist, and the nature of the application generally determines which control strategy is applied. Induction machine occupies an important position in industry due to its various advantages over other machines as regards to price, size, robustness etc. However, it was traditionally used in fixed speed applications because it represents a highly nonlinear, interacting, multivariable control plant requiring complex control algorithms. It is essentially a constant-speed machine as long as it is connected to a constant voltage and constant frequency power supply. In this case the operating speed is very close to the synchronous speed, and changes in the torque makes small changes in the speed. It is suitable for use in constant speed drive systems, while for variable speed applications the controllers used are more complicated than the traditional dc motor speed controllers[6][3]. However, with the advancement of power electronics and vector control techniques, it is possible to vary the speed as required. There are various methods for controlling the speed of an induction machine. Some of the important methods are briefly discussed in this chapter.

3.2 Induction motor control techniques

In many applications, the capability to vary speed efficiently can lead to large savings in energy. Adjustable-speed induction-motor drives are also used for electric traction, and for motion control to automate factories[4]. Controlling the speed of an induction motor is far more difficult than controlling the speed of a DC motor since there is no linear relationship between the motor current and the resulting torque as there is for a DC motor. To control the induction motor there are different types of control strategies, these are:

- Variable supply voltage control
- Variable rotor resistance control
- Constant Volts/Hz control (scalar control)
- direct torque control
- Vector Control

3.2.1 Scalar control

Scalar control[10][4], as the name indicates is due to magnitude variation of the control variables only, and disregards any coupling effect in the machine. For example, the voltage of the machine can be controlled to control the flux, and the frequency or slip can be controlled to control torque. Scalar controlled drives give somewhat inferior performance than the other control schemes but they are easy to implement. Scalar control is based on the steady state model of the machine. The control is due to the magnitude variation of the control variables only, and disregards the coupling effect in the machine. Variable supply voltage control, Variable rotor resistance control and Constant Volts/Hz control (scalar control) are common scalar controls of IM. For this thesis, the vector control technique is selected because of many advantages which will discussed in the following sections.

3.2.2 Vector control

The technique called vector control[18] can be used to vary the speed of an induction motor over a wide range. It was initially developed by Blaschke (1971-1973). In the vector control scheme, a complex current is synthesized from two quadrature components, one of which is responsible for the flux level in the motor, and the other which controls the torque production in the motor. Essentially, the control problem is reformulated to resemble the control of a DC motor. Vector control offers a number of benefits including speed control over a wide range, precise speed regulation, fast dynamic response, and operation above base speed.

Vector control also known as Field Oriented Control (FOC) which allows torque and flux to be decoupled and controlled independently. FOC describes the way in which the control of torque and speed are directly based on the electromagnetic state of the motor, similar to a DC motor. It is the first technology to control the real motor control variables of torque and flux. With decoupling between the stator current components (magnetizing flux and torque), the torque producing component of the stator flux can be controlled independently. Decoupled control, at low speeds, the magnetization state of motor can be maintained at the appropriate level, and the torque can be controlled to regulate the speed. FOC has been solely developed for high-performance motor applications which can operate smoothly over the wide speed range, can produce full torque at zero speed, and is capable of quick acceleration and deceleration[29][30]. There are basically two schemes under vector control.

A. Direct vector control: Determines the rotor flux position from measurements, using field angle sensors. The direct vector control depends on the generation of unit vector signals from the stator or rotor flux signals. The air-gap flux signals can be measured directly or estimated from the stator voltage or current signals. In these systems, rotor speed is not required for obtaining rotor field angle information[31]. Here, the actual motor currents are converted to synchronously rotating frame currents using park transformation. The resulting dc quantities are compared with the reference d-axis and q-axis components. The outputs of the controllers are used to generate the pulse width modulated signals for switching the devices in the inverter bridge feeding the motor[33]. Since, these method has disadvantages due to cost (the sensor costs 10% of the drive system). Hence, due to these disadvantages, normally indirect method of vector control is preferred. The block diagram of direct vector control is shown in the fig.3-1 below.

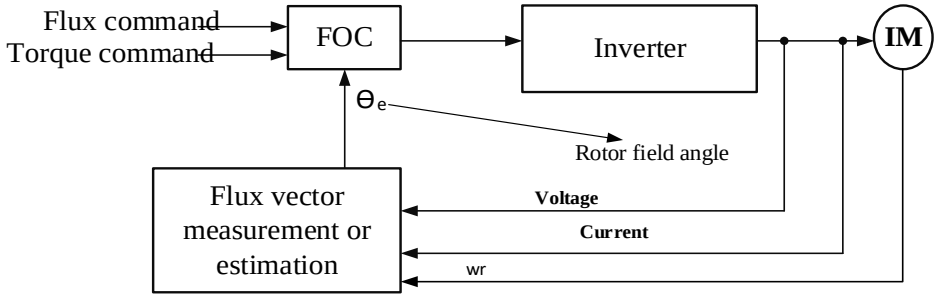


Figure 3-1: Vector control direct method

B. Indirect vector control: which measures the rotor position and in turn utilizes the slip relation to compute the angle of the rotor flux relative to the rotor axis as shown in the fig.3-2 below. In this method, the unit vector signal that transforms the synchronously rotating stator voltages into stationary frame signals has been generated from the speed signal and slip signal[33]. Here, we have used the indirect scheme to determine the rotor flux position due to the following advantages. Some of the advantages are sensors are eliminated, the dynamic performance of the indirect vector control is better than the direct vector control and the cost factor is decreased. Indirect vector control using intelligent controller approach avoids the use of flux and speed sensor which increases the installation cost and mechanical robustness. This method

has also disadvantage of the machine parameter variation affects the slip gain, and Correspondingly, both static and dynamic performances of the drive are affected[32].

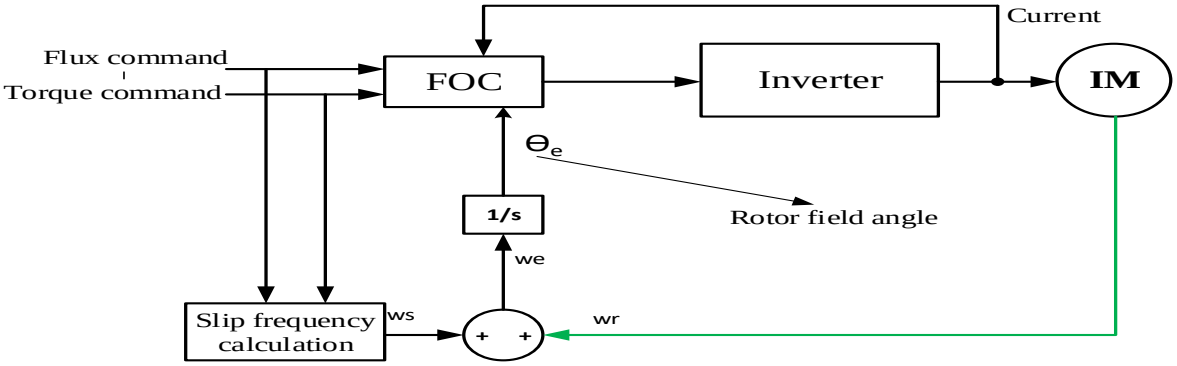


Figure 3-2: Indirect vector control block diagram.

3.3 Decoupling

The drawback of the coupling effect in the control of SCIMs is that it gives sluggish response and the system is easily prone to instability because of a high order system effect. This problem can be solved by making use of either vector control or field-oriented control. When this type of control strategy is adopted, it can make an induction motor to be controlled like a separately excited DC motor. To decouple and control the torque and flux independently, the rotor current and rotor flux are kept perpendicular. To achieve this, two cases must be considered for both direct and indirect FOC.

Case 1: Keep the q-axis rotor flux linkage zero; $\varphi qr' = 0$.

Case 2: Force to keep the d-axis current zero; *i*dr'=0. This can be satisfying by forcing the d-axis stator current to remain constant.

From equation (2.13) above substitute case 1 for d-axis rotor voltage equation. Then the result is [14]

$$Rr'idr' + P\varphi dr = 0 \qquad (3.1)$$

Substitute equation (2.15) of d-axis rotor flux linkage to equation (3.1), the result is

$$Pidr' = -\frac{Rr'}{Lr} idr' - \frac{Lm}{Lr} Pids \qquad (3.2)$$

Where: $Lr = Llr + Lm \; and \; P = \frac{d}{dt}$

From equation (3.2) if ids is keeping constant, then idr' goes to, and, stays at, zero. Then equation (2.16) of d-axis flux linkage becomes as follows when substituting the d-axis rotor current zero.

$$\varphi dr' = Lm \; ids \tag{3.3}$$

$$Te = \frac{3}{2}\frac{p}{2}\frac{Lm}{Lr}\varphi dr' iqs \tag{3.4}$$

Equation (3.3) shows that the d-axis flux linkage determined stator current alone. And equation (3.4) shows if the rotating reference frame with flux linkage as its d-axis is known we can produce desired torque. Equations (3.3) and (3.4) are the generic equations of rotor flux-oriented control.

Note that the torque is proportional to the product of the rotor flux linkages and the stator q - axis current. This resembles the torque expression of dc motor, which is proportional to the product of the field flux linkages and the armature current. If the rotor flux linkage is kept constant, then the torque is simply proportional to the torque producing component of the stator current (iqs), as in the case of the separately excited dc machine, where the torque is proportional to the armature current when the field current is constant.

The difference between the direct and indirect FOC is in how to determine the d-axis rotor flux ($\varphi dr'$) and the position angle (θe). In the case of DFOC the determination of θe is using hall- effect sensor which costs 10% of the total drive. Whereas using IFOC is not need of sensor and knowledge of rotor flux linkages to determine θe. The IFOC method is discussed as follows.

By keeping the two cases above, from equation (2.16) of the d-axis rotor flux linkage, we can get

$$iqr' = -\frac{Lm}{Lr} iqs \tag{3.5}$$

Also from equation (3.3) we can get the slip equation.

$$\omega e = \omega r - Rr\frac{iqr'}{idr'} \tag{3.6}$$

Substituting equations (3.3) and (3.5) in to equation (3.6), the result becomes as

$$\omega e = \omega r + \frac{Rr'}{Lr'}\frac{iqs}{ids} \tag{3.7}$$

Once the speed is calculated, we can get the position by integrating the speed.

$$\theta e = \int \omega e dt \tag{3.8}$$

Generally indirect vector control can be implemented using the following equations.

$$ids^* = \frac{1}{Lm}\varphi dr'^* \tag{3.9}$$

$$\varphi dr'^* = Lm * ids^* \tag{3.10}$$

$$iqs^* = \frac{4}{3p} * \frac{\frac{Lr'}{Lm} * Te^*}{\varphi dr'^*} \tag{3.11}$$

$$\omega sl = \frac{Rr'}{Lr'} * \frac{iqs*}{ids*} \tag{3.12}$$

$$\theta e = \int \omega e dt \tag{3.13}$$

Where θe is rotor field angle, ωe is speed, ωsl represents the slip speed and iqs, ids, iqr, idr represents stator and rotor current in q-axis and d-axis respectively.

The rotor circuit equation becomes[16]

$$p\varphi dr' + \frac{Rr}{Lr} * \varphi dr' = \frac{Lm}{Lr} * Rr\ ids \tag{3.14}$$

$$p\varphi qr' + \frac{Rr}{Lr} * \varphi qr' = \frac{Lm}{Lr} * Rr\ iqs \tag{3.15}$$

Where $p = d/dt$

For decoupling control, the stator flux component of current ids should be aligned on the d axis, and the torque component of current iqs should be on the q axis, that leads the rotor q-axis flux linkage to zero ($\varphi qr' = 0 \quad thus \quad \varphi r' = \varphi dr' + j\varphi qr' = \varphi dr', \tau r = \frac{Lr}{Rr}$) , Where τr is

rotor time constant, $\varphi dr', \varphi qr'$ represents rotor flux in the d-axis and q-axis respectively. From equation (3.14 and 3.15) we can get the following relationship.

$$\tau r * p\varphi dr' + \varphi dr' = Lm * ids \tag{3.16}$$

$$\varphi dr' = \varphi r' = \frac{Lm}{1 + \tau rs} * ids \tag{3.17}$$

3.4. Design of GA based PI controller

The use of PI controllers for speed control of induction machine drives is characterized by an overshoot during tracking mode and a poor load disturbance rejection. This is mainly caused by the fact that the complexity of the system does not allow the gains of the PI controller to exceed a certain low value. At starting mode, the high value of the error is amplified across the PI controller provoking high variations in the command torque. If the gains of the controller exceed a certain value, the variations in the command torque become too high and will destabilize the system. To overcome this problem, we propose the use of intelligent controllers instead of the conventional PI controller.

The transfer function of PI controller is

$$Gc(s) = Kp + \frac{Ki}{s} \tag{3.18}$$

Where Kp and Ki are controller gains and Gc(s) is the closed loop transfer function.

The speed control loop of induction motor is simplified as the following block diagram which is derived from the torque equations (2.17) and (3.4).

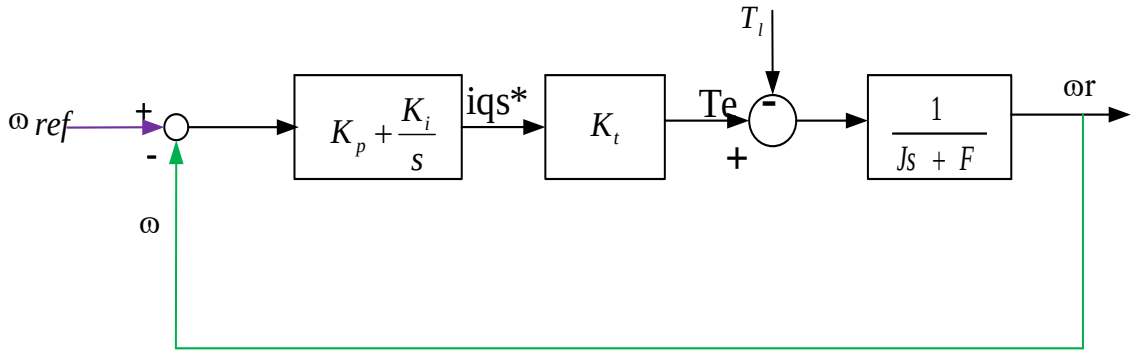


Figure 3-3: Speed control loop of induction motor

Where: $$Te = kt * iqs \quad (3.19)$$

$$kt = \frac{3}{2} * \frac{p}{2} * \frac{Lm}{Lr} * \varphi dr \quad (3.20)$$

$$\varphi dr = Lm * ids \quad (3.21)$$

Therefore $$kt = \frac{3}{2} * \frac{p}{2} * \frac{Lm^2}{Lr} * ids \quad (3.22)$$

The major inconvenience of Z-N is the necessity of a priori knowledge of the various parameters of the induction motor. To surmount this inconvenience, we can use a procedure of optimization to better design this type of controller.

The Genetic Algorithm[19] is a method for solving optimization problem that is based on natural selection, the process that drives biological evolution. The GAs repeatedly modifies a population of individual solutions. At each step, the GAs select individual at random from the current population to be parents and uses them to produce the children for the next generation. Over successive generation, the population evolves toward an optimal solution. Three main steps of GAs are:

Step1 Selection: Selects the individuals, called parents that contribute to the next generation.

Step 2 Crossover: Combines two parents to form children for the next generation.

Step 3 Mutation: Apply random changes to individual children.

The implementation of the tuning procedure through genetic algorithms starts with the definition of the chromosome representation [34]. As shown in the fig below the chromosome is formed by two values that correspond to the two gains to be adjusted in order to achieve a satisfactory behavior. The gains Kp and Ki are real numbers and characterize the individual to be evaluated.

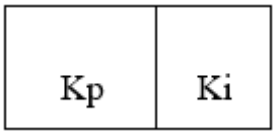


Figure 3-4: Chromosome definition

GA can be applied in the tuning of the PI speed controller's gains (KP, Ki) to ensure optimal control performance for the induction motor.

3.4.1 Fitness function

In PI controller design methods, the most common performance criteria are integrated absolute error (IA E), the integrated of time weight square error (ITSE), integrated of squared error (ISE) and integrated of time weight absolute error (ITAE) that can be evaluated analytically in the frequency domain [26]. In this paper, a time domain criterion is used for evaluating the PI controller. A set of good control parameters P and I can yield a good response that will result in performance criteria minimization in the time domain. Since the objective is to minimize the error between the set point (desired output) and the plant output (actual output), the fitness function we selected is an integral square error ISE, defined as:

$$J=\int_0^t e(t)^2 dt = \int_0^t (\omega^*(t) - \omega(t))^2) dt \qquad (3.23)$$

Where: e (t) is the error,$\omega^*(t)$ and $\omega(t)$ are reference and actual speeds. The block of the objective function is used to estimate the performances of the PI controller by minimizing this function. GA may be employed to optimize a model or a function. The model or a function is called a fitness function in a GA optimization problem. The following flowchart shows the basic steps of genetic algorithm.

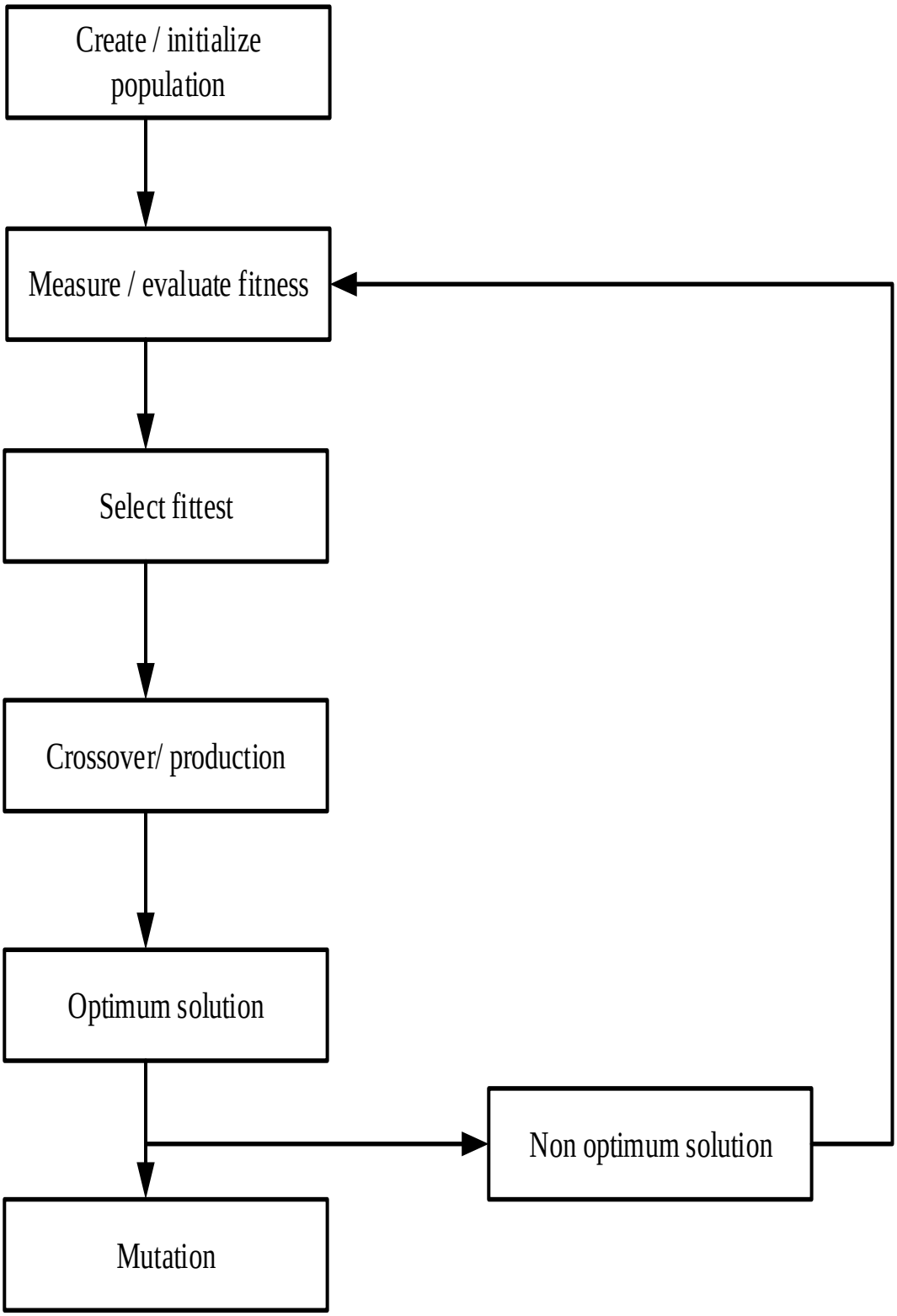


Figure 3-5: General flow chart for GA

The following Table 3-1 summarizes the genetic algorithm parameters chosen for the tuning of PI controller parameters (Ki and Kp).

Table 3-1: parameters of Genetic Algorithm

GA property	**Values**
Population size	60
Selection function	Tournament selection
Mutation function	Uniform mutation
Crossover function	Intermediate
Number of generation	150
Cross over probability	0.8
Mutation probability	0.1
Tolerance	10e-5

After giving the above parameters to GA, the PI controller can be easily tuned and thus system performance can be improved. The parameters of the PI speed controller obtained according to the procedure of optimization by the technique of the GA are tuned by the MATLAB command:

[x,fval] = ga(@Call_PI,2,[],[],[],[],lb, ub,[],options);

Where: x is the pi parameters (Kp and Ki), fval is fitness function value and @Call_PI is the function name. Because initial chromosomes in the genetic algorithm are randomly created, the program may need to be run several times in order to obtain satisfactory results. The following figure shows the best and mean fitness obtained during optimization.

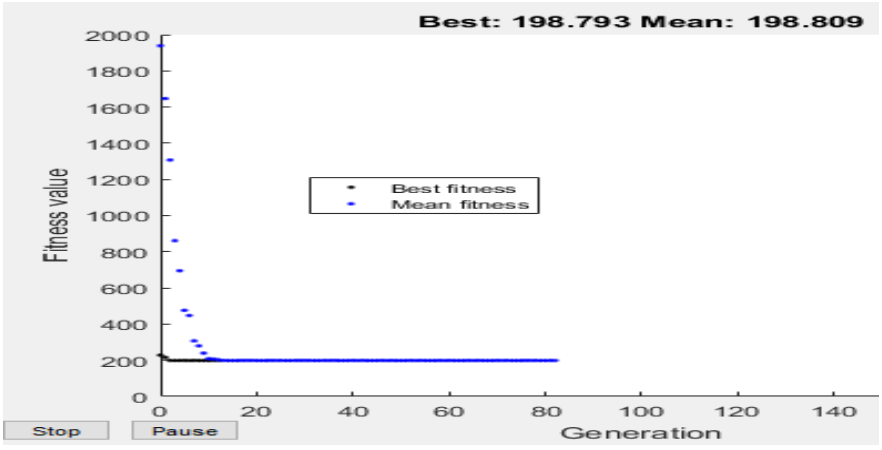


Figure 3-6: Best evaluated outputs of the fitness function at each generation.

The PI parameters for the current controller are tuned by trial and error. The tuned parameters for the current controller are KP=30 and Ki=5 for both direct and quadrature axis components.

Table 3-2: GA based PI parameters for speed controller

Parameters	**Controllers**		
	Speed	**Current (i_ds)**	**Current (i_qs)**
Kp	10.51	30	30
Ki	30.667	5	5

In the above optimization MATLAB command, the fitness function is the Call_PI.m which calls Simulink model 'toData.mdl'. The chromosome is the PI controller parameters [x1, x2]. Parameters of GA performance use values of the 'ga' function which are listed in Table 3.1.When the performance needs to be improved, the parameters may be changed by '*gaoptimset'* function.

3.4 Intelligent Controllers

The design of traditional PI controller depends on exact machine model and accurate model parameters. However, the difficulties of obtaining the exact parameters of the Induction motor lead to cumbersome design approach for these controllers. Moreover, the fixed gain PI controller is very sensitive to disturbance. On the other hand, the design of intelligent controllers such as fuzzy logic, neural network, neuro-fuzzy controllers, and etc. do not need exact mathematical model of the system. Simplicity, tolerance to uncertainty and less intensive mathematical design requirements are the main features of intelligent controllers, which are suitable to deal with nonlinearities and uncertainties of electric motors. Therefore, the intelligent controllers demand particular attention for high performance of drives. However, a simple FLC has a narrow speed operation and needs much more manual adjustment by trial and error if high performance is needed. On the other hand, it is extremely tough to create a serial of training data for ANN [20].

The difficulties that arise in induction motor control are complex computations, model nonlinearity, and uncertainties in machine parameters. Recently, intelligent techniques like FL, ANN and ANFIS are introduced in order to overcome these difficulties.

3.4.1 Fuzzy Logic Controller:

Fuzzy logic is one type of artificial intelligent which applies human-like thinking into a control system. A fuzzy system can be designed in the same manner as that of human thinking that is the method people employ to obtain conclusions from what they know[18].

Fuzzy Logic Controller (FLC) is a rule-based controller. The structure of the FLC resembles that of a knowledge-based controller except that the FLC utilizes the principles of fuzzy set theory in its data representation and its logic. The basic configuration of FLC is simply represented in four parts, as shown in Fig.3-8 below.

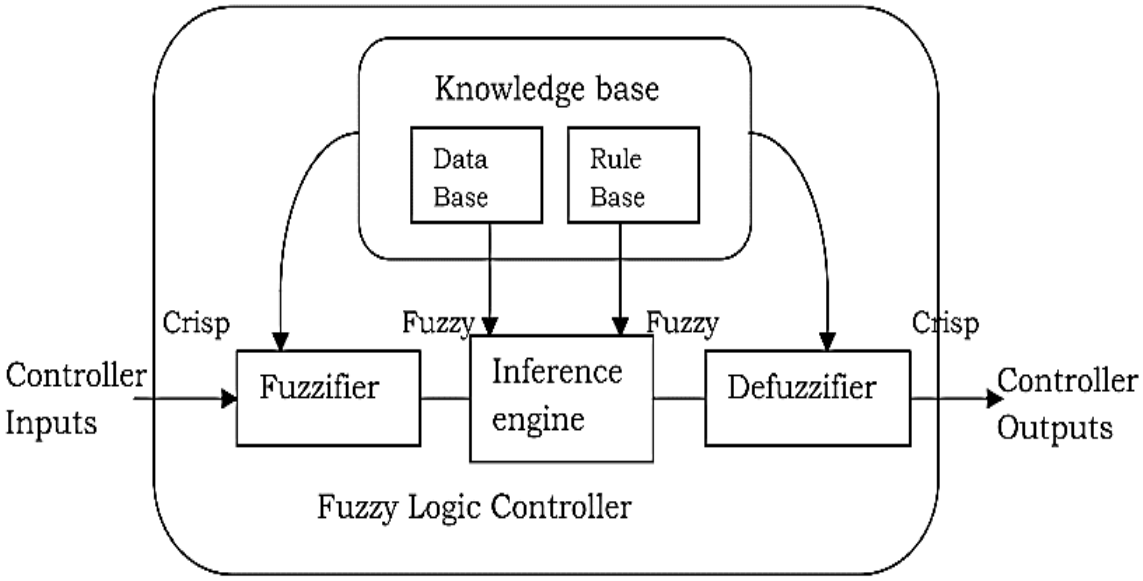


Figure 3-7: Schematic diagram of the FLC building blocks [18]

i. The fuzzifier

The various actions done by a fuzzifier are: the values of input parameters can be measured, transform the input parameters into corresponding universe of discourse through scale mapping and performs the process of fuzzification which converts crisp inputs to fuzzy values

ii. Knowledge base

Data base and linguistic rule base are present here. The necessary definitions in order to define the control rules and perform fuzzy manipulation are provided by the database and the linguistic control rules monitors the control goals and the control policy of domain experts.

iii. The decision making logic

It is the main key of FLC and it has the ability to perform the process of decision making based on concept of fuzzy and giving output actions using the application of fuzzy and rule base.

iv. Defuzzifier

A type of mapping which converts the fuzzy variables into crisp value. It gives a non-fuzzy actions from a control action which is in fuzzified form.

But the main disadvantage of the Fuzzy logic controller is that it has a narrow speed operation and needs much more manual adjusting by trial and error if high performance is needed. Other disadvantage of fuzzy logic is that system time response slows down with the increase in number of rules [28]. If the system does not perform satisfactorily, then the rules are reset again to obtain efficient results i.e. it is not adaptable according to the variation in data. The accuracy of the system is dependent on the knowledge and experience of human experts. The rules should be updated and weighting factors in the fuzzy sets should be refined with time. Neural networks, genetic algorithms, swarm optimization techniques, etc. can be used to for fine tuning of fuzzy logic control systems. So, new technique employing the combined properties of artificial neural networks and fuzzy logic known as Adaptive Neuro Fuzzy Inference System (ANFIS) has been developed.

3.4.2 Adaptive Neuro-fuzzy inference system (ANFIS)

ANFIS is a hybrid system incorporating the learning abilities of ANN and excellent knowledge representation and inference capabilities of fuzzy logic[21] that have the ability to self-modify their membership function to achieve a desired performance.

Literature survey shows that a new design technique, that is, Adaptive Neuro-Fuzzy Inference System (ANFIS), is used for design of fuzzy logic controller. The tuning of fuzzy inference system is carried out by hybrid algorithm based on some collection of input-output data. ANFIS method[22] is used as a teaching method for Sugeno-type fuzzy systems. Usually the number and type of fuzzy system membership functions are defined by user when applying ANFIS. The ANFIS employs the property of neural networks in order to develop a fuzzy inference system.

ANFIS is a simple data learning technique that uses Fuzzy Logic to transform given inputs into a desired output through highly interconnected Neural Network processing elements and

information connections, which are weighted to map the numerical inputs into an output. It combines the benefits of the two machine learning techniques (Fuzzy Logic and Neural Network) into a single technique. An ANFIS works by applying Neural Network learning methods to tune the parameters of a Fuzzy Inference System (FIS). ANFIS achieves great success because it offers desired data set, greater choice of member-ship functions to use, strong generalization abilities, excellent explanation facilities through fuzzy rules and it is easy to incorporate both linguistic and numeric knowledge for problem solving[23].

Most of Neuro-Fuzzy systems are either based on Takagi-Sugen (T-S) or Mamdani type fuzzy logic controller. T-S type based Nuro-Fuzzy models are widely used for model-based applications. It combines the advantages of being general approximators that can reach high accuracy and being easy to interpret. And also have high performance, but often requires complicated learning procedures and computational expensive. However, Mamdani type based Neuro-Fuzzy systems can be modeled using faster heuristics but with a low performance. Because of the above said merits of Mamdani type, T-S type neuro-fuzzy system is considered for this thesis.

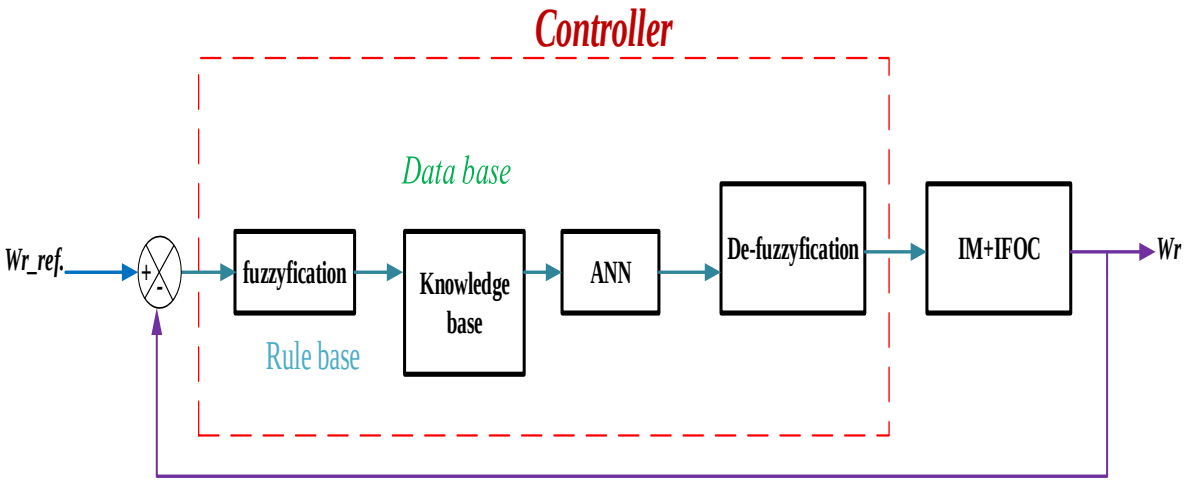


Figure 3-8: Figure 3-8: The ANFIS control scheme for speed control of IVCIM [22]

For this work, the proposed neuro-fuzzy controller incorporates fuzzy logic algorithm with a five layer artificial neural network (ANN) structure as shown in the figure 3.10b. The speed error and the rate of change of actual speed error are the inputs of the neuro-fuzzy controller and the output

is the reference torque. Sugeno fuzzy model with five- layer ANN structure is used in proposed controller. The following figure 3.9 shows how the logic of ANFIS controller looks like.

Input1=error=$(\omega ref - wr)$

Input2=change in error=$\Delta error = \Delta(\omega ref - \omega r)$

Where: $\omega ref \; and \; wr$ are the reference and actual rotor speeds respectively and Δ indicates the change.

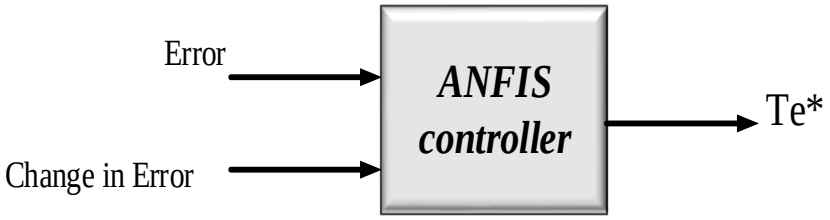


Figure 3-9: Logic of ANFIS controller

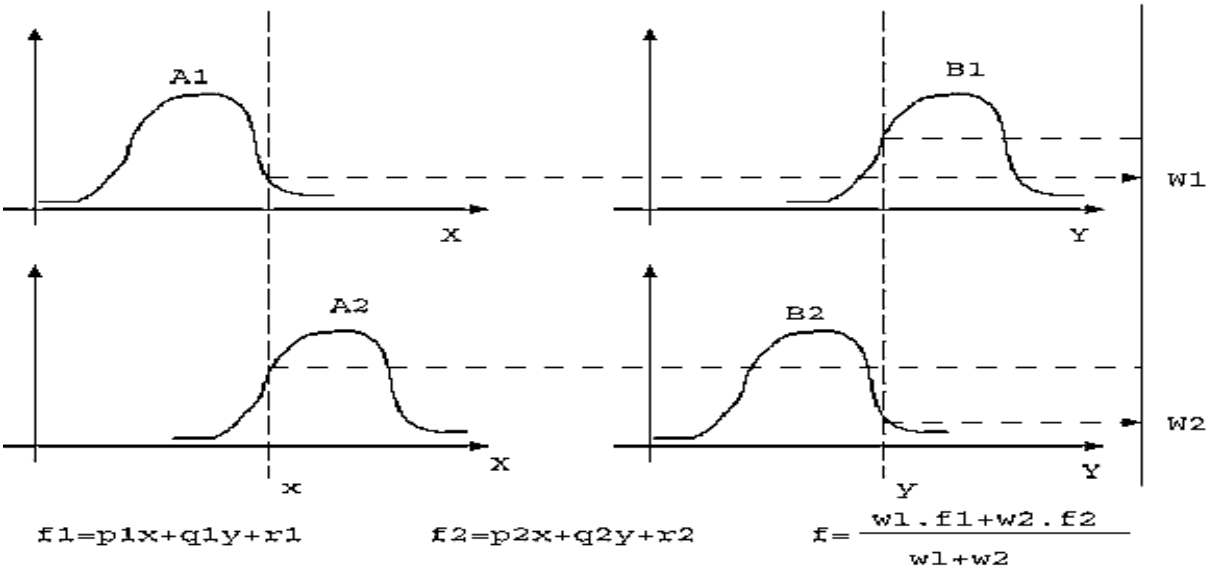
3.4.3 ANFIS Architecture

ANFIS is a simple data learning technique that uses a fuzzy inference system model to transform a given input into a target output. This prediction involves membership functions, fuzzy logic operators and if-then rules. There are two types of fuzzy system, commonly known as the Mamdani and Sugeno models. There are five main processing stages in the ANFIS operation, including input fuzzification, application of fuzzy operators, application method, output aggregation, and defuzzification which are discussed in the previous sections. For the fuzzy inference system that has two inputs x and y and one output f, a first-order Sugeno fuzzy model has rules as the following:

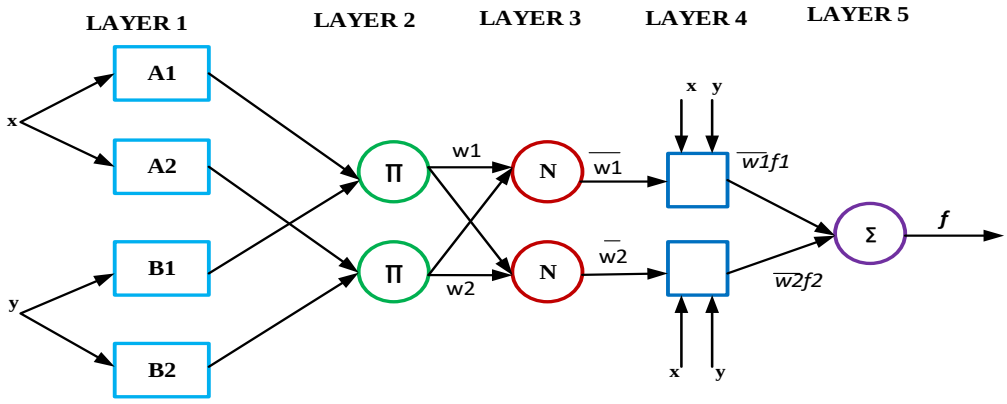
Rule1: If x is A1 and y is B1, then f1 = p1x+q1y +r1

Rule 2: if x is A2 and y is B2, then f2 = p2x+q2y +r2

Where Ai and Bi are the fuzzy sets in the antecedent, and pi, qi and ri are the design parameters which are called consequent parameters that are determined during the training process.



(a)



(b)

Figure 3-10: (a) fuzzy reasoning, (b) basic ANFIS architecture [23]

To reflect different adaptive capabilities, we use both circle and square nodes in an adaptive network. A square node (adaptive node) has parameters while a circle node (fixed node) has none. The parameter set of an adaptive network is the union of the parameter sets of each adaptive node. In order to achieve a desired input-output mapping, these parameters are updated according to given training data and a hybrid learning procedure is used. In this five-layer ANN structure, the first layer represents for inputs, the second layer represents for fuzzification, the third and fourth layers represents for fuzzy rule evaluation and the fifth layer represents for defuzzification.

Layer 1

Ol, i is the output of the i[th] node of the layer l. Every node i in this layer is an adaptive node with a node function

$$\begin{cases} O1,i = \mu A_i(x), & i = 1,2, or \\ O1,i = \mu B_{i-2}(y), for\ i = 3,4. \end{cases} \quad (3.24)$$

x and y are the inputs node i and A_i and B_{i-2} is a linguistic label associated with this node. Therefore O1,i is the membership grade of a fuzzy set (A1 A2 B1 B2). The membership type can be any type of member ship function which are available in fuzzy toolbox. For this thesis, typical membership function is triangular membership function which is given by:

$$\mu A(x) = \begin{cases} 0, & x \le a \\ \dfrac{x-a}{b-a}, & a \le x \le b \\ \dfrac{c-x}{c-b}, & b \le x \le c \\ 0, & x \ge c \end{cases} \quad (3.25)$$

Where ai, bi and ci are the parameter sets that are referred to as premise parameters which are optimized during training process.

Layer 2

Every node in this layer is a fixed node labeled Prod (π). The output is the product of all the incoming signals.

$$O2, i = wi = \mu A_i(x) * \mu B_i(y), i = 1,2. \quad (3.26)$$

Each node represents the fire strength of the rule. Any other T-norm operator that perform the AND operator can be used.

Layer 3

Every node in this layer is a fixed node labeled Norm. The i[th] node calculates the ratio of the i[th] rulet's firing strength to the sum of all rulet's firing strengths. Outputs are called normalized firing strength.

$$O3, i = \overline{wi} = \frac{wi}{w1 + w2}, \qquad i = 1,2. \quad (3.27)$$

Layer 4

Every node i in this layer is an adaptive node with a node function:

$$O4,i = \overline{wi}fi = wi(p_i x + q_i y + ri) \tag{3.28}$$

Where $\overline{wi}$ is normalized firing strength from layer 3.

Layer 5

The single node in this layer is a fixed node labeled sum (∑), which computes the overall output as the summation of all signals.

$$\text{Over all output(f)} = 05,1 = \sum_{\text{i}} \overline{\text{wi}}\text{fi} = \frac{\sum_{\text{i}} \text{wifi}}{\sum_{\text{i}} \text{wi}} \tag{3.29}$$

The parameters to be trained are ai, bi and ci of the premise parameters and pi, qi, and ri of the consequent parameters.

3.4.4 Learning algorithm of ANFIS

In the ANFIS architecture, the first layer and the fourth layer contain the parameters that can be modified over time. In the first layer, it contains a nonlinear of the premises parameters while the fourth layer contains linear consequent parameters[24]. To update both of these parameters, required a learning method that can train both of these parameters and to adapt to its environment. From fig.3.11b: the overall output is given by.

$$f = \frac{w1}{w1+_{w2}} f1 + \frac{w2}{w1 + w2} f2$$

$$f = \overline{w1}f1 + \overline{w2}f2$$

$$f = \overline{w1}(p1x + q1y + r1) + \overline{w2}(p2x + q2y + r2)$$

$$f = (\overline{w1}x)p1 + (\overline{w1}y)q1 + (\overline{w1})r1 + (\overline{w2}y)p2 + (\overline{w2y})q2 + (\overline{w2})r2 \tag{3.30}$$

When the values of premise parameters at layer two are fixed, the overall output f can be expressed in terms of linear combination of the consequent parameters p1, q1, r1,p2,q2and r2 which is of the form

$$S = S1 \oplus S2 \tag{3.31}$$

The above equation signifies that the set of total parameters (S) can be divided into a set of premise parameters (S1) and consequent parameters (S2) where $\oplus$ represents direct sum. The adaptive network's output can be written as

$$O = F(i, S) \tag{3.32}$$

Where i is the vector of input variables and F is the function of the fuzzy inference system. The total parameter set S is divided such that HoF is linear in the elements of S2 where H is the identity function. The hybrid learning algorithm of applying back propagation to find premise parameters S1 and Least Squares Estimates (LSE) to determine consequent parameters S2 is made use of in this offline learning[21]. Each learning step consists of two passes namely, forward and backward. In the forward pass, node outputs are propagated and the optimal consequent parameters are estimated by LSE while the premise parameters are assumed to be fixed for the current cycle. In the backward pass, back propagation modifies the premise parameters by gradient descent and the consequent parameters remain fixed. This iterative procedure is carried on till the error criterion is satisfied[23] [25]. Typically, this error criterion is the sum of the squared difference between the actual and desired outputs or in some situations, a user defined performance measure.

The output error is used to adapt the premise parameters by means of a standard back propagation algorithm. It has been proven that this hybrid algorithm is highly efficient in training the ANFIS systems. Table 3.3 summarizes these activities in each pass. The mathematical description of hybrid learning algorithm employed in ANFIS is given in Appendix A. This hybrid approach converges much faster since it reduces the search space dimensions of the original pure back propagation algorithm. This is because of the optimal estimation of consequent parameters while assuming premise parameters as fixed[21].

Table 3-3: Hybrid learning method

Type	**Forward path**	**Backward path**
Premise parameter	Fixed	Gradient descent
Consequent parameter	LSE	Fixed
Signal	Node output	Error rate

The reason of combination of back propagation (BPM) and least square method (LSM) is that when the premise parameters are not fixed, the search space becomes larger and the convergence

of the training becomes slower. Thus the training process that results in the learning has two steps in each iteration[26].

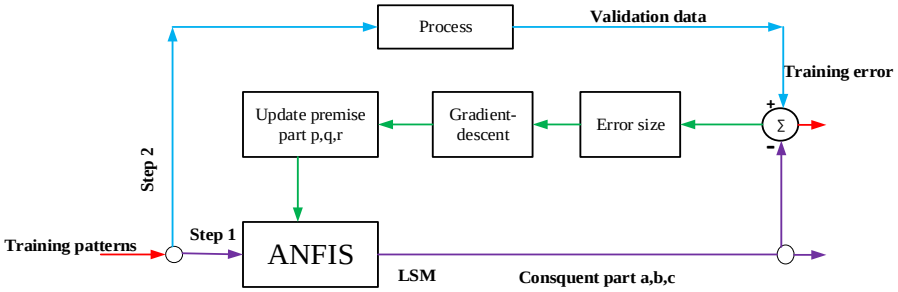


Figure 3-11: ANFIS learning process by hybrid algorithm

Step 1: In the first epoch, the input patterns are propagated and the optimal consequent parameters are identified by the LSM, while the premise parameters are assumed to be fixed for the current cycle through the training set.

Step 2: In the second epoch, the patterns are propagated again; the error signals propagate backward to modify or update the premise parameters, by gradient descent (a back propagation gradient descent method).

Since the hybrid learning approach converges much faster by reducing search space dimensions than the original back propagation method, it is more desirable. In the forward pass of the hybrid learning, node outputs go forward until layer 4 and the consequent parameters are identified with the least square method. In the backward pass, the error rates propagate backward and the premise parameters are updated by gradient descent.

3.5 Design of Adaptive Neuro - Fuzzy Inference System:

ANFIS based modelling combines the transparent linguistic representation of fuzzy systems with the learning ability of neural networks so that they can be trained to perform an input/output mapping. In designing, we cannot decide what the membership function must be, just by merely looking at the data. ANFIS permits the parameters to be automatically adjusted so that the membership functions capture the dynamics of data. The set of rules ANFIS provides is indicative of the underlying system and hence is valuable information to gain further insight into the process model. When a FIS is launched as a controller, the special requirement is that the refining of parameters of membership functions should be done in such a way that the best performance of the plant is guaranteed.

In order to generate fuzzy system, One hundred thousand one (100,000) data have been considered. Out of these data, seventy thousand (70% of total data) has been considered as training data set and remaining data has been considered as checking data. Since the ANFIS involves the combined properties of neural network and fuzzy logic, it provides a method for the fuzzy modeling procedure to learn information about a data set, in order to compute the membership function parameters that allow the associated fuzzy inference system (FIS) to track the given input/output data. In order to develop the FIS structure, a network type structure similar to that of neural network which maps inputs through input membership function and associated parameters and then through output membership functions and associated parameters to outputs can be considered.

The parameters associated with the membership functions will change through the learning process. The computation of these parameters is done by a hybrid learning algorithm. In this work, number of membership functions considered during the development of ANFIS is seven and type of input membership functions used is triangular membership functions. Linear membership function type is selected for output, totally 49 rules are generated, and the number of epochs considered is 60. The plot of type of membership functions for inputs, the rules, surface and rule views are shown in Appendix D.

The performances of the ANFIS models of both training and checking data were evaluated and the best training/checking data set was selected according to RMSE. In order to find the ideal ANFIS system, we trained the system with a variety of settings for items such as data set sample, epoch number, membership function type and number, and number of inputs to achieve the best

performance. Using a given input/output data set, the MATLAB toolbox function *anfis* constructs a fuzzy inference system (FIS) whose membership function parameters are tuned (adjusted) using hybrid learning algorithm method.

The data were divided into two separate sets: the training data set and the checking data set. The training data set was used to train the ANFIS, whereas the checking data set was used to verify the accuracy and the effective-ness of the trained ANFIS model for the adaptation of learning content. Two pair data sets were made with different combinations of 70% and 30% of the samples to improve the generalization properties of the adopted ANFIS as follows:

- Data set pair 1: Training set first 70%; Test set last 30%
- Data set pair 2: Training set last 70%; Test set first 30%

The best weights (giving minimum mean-squared error) of two different training sessions over each input/output training set were chosen as the final ANFIS models. The performances of the ANFIS models both training and testing data are evaluated and the best training/testing data set is selected according to training, checking and testing errors. So for this work, the first data set pair is selected to model the ANFIS.

The training and checking data pairs were gathered from the simulation result of the block shown in figure 4.1 using the PI controller. After we collect the input output data, we divided the total data in to training and checking data sets randomly. Once we collect the I/O data pairs, the first step was loading training and checking data pairs to the anfis GUI of MATLAB toolbox, then initialize the FIS was the second step (the initialization step includes the number and type of input and output member ship functions and also number of epoch numbers). After initializing FIS and specifying the number of epochs the next step was train the training data pairs until the minimum training and checking errors are recorded. If the errors are acceptable the final step was testing the training data using checking data to validate the model. Fig.3.12 below summarizes the overall flow chart of ANFIS model.

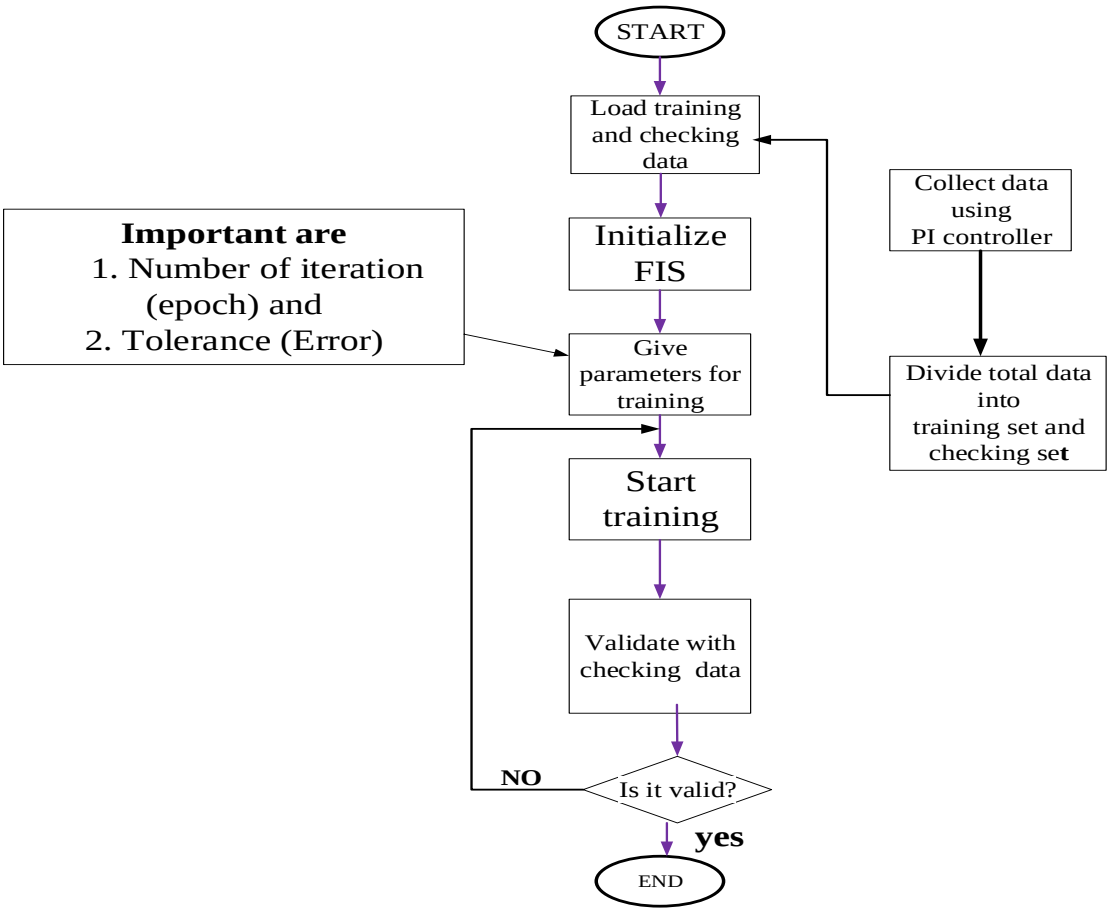


Figure 3-12: Flow chart for design of ANFIS model

After loading the training data and checking data sets in Neuro-Fuzzy Designer, the following results have been observed. Fig.3.13 shows the training and checking data sets after loading in ANFIS editor tool box.

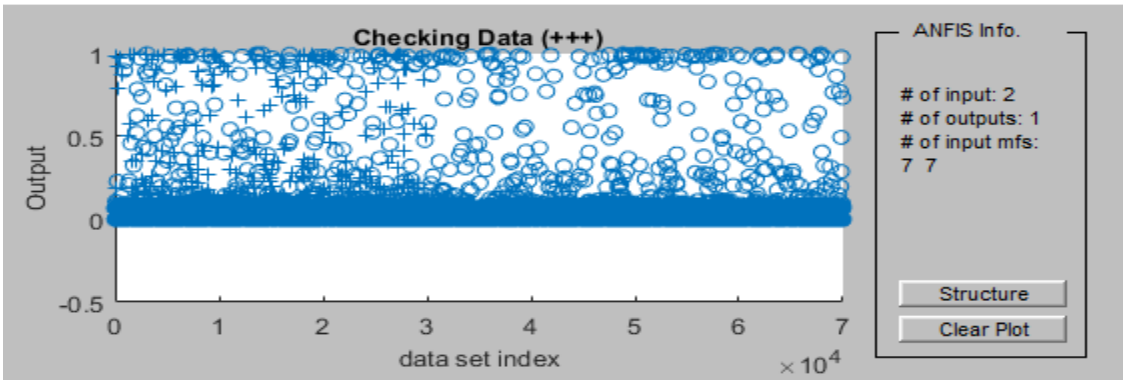
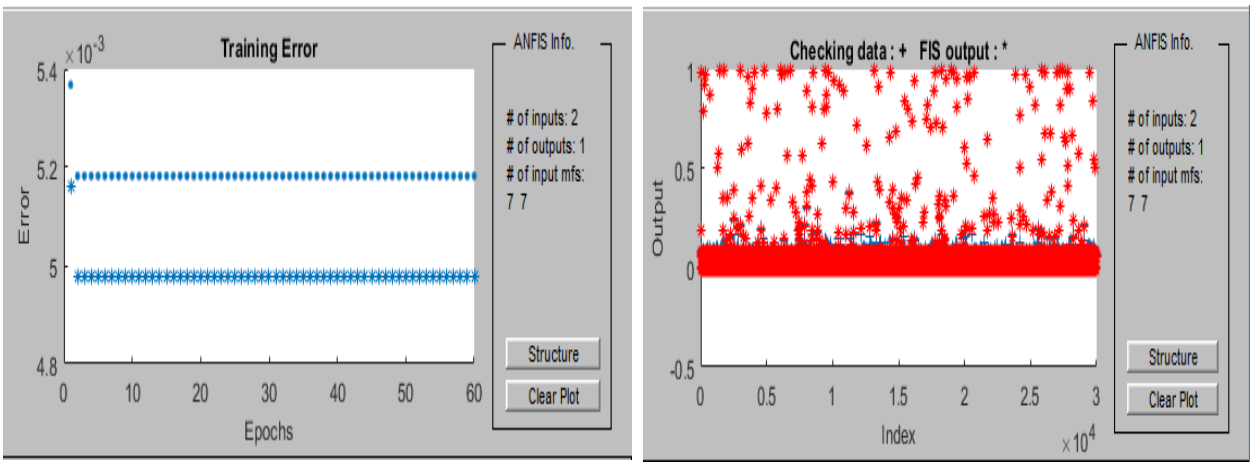


Figure 3-13: Loading I/O data sets

Checking data is used for testing the generalization capability of the fuzzy inference system at each epoch. The checking data has the same format as that of the training data. This data set is used to validate the fuzzy inference model. This validation is done by applying the checking data to the model and then seeing how well the model responds to this data. When the checking data option is used using the ANFIS Editor GUI, the checking data is applied to the model at each training epoch. The FIS membership function parameters are computed using the ANFIS Editor GUI when both training and checking data are loaded. The checking data is similar enough to the training data that the checking data error decreases as the training begins. The checking error is the difference between the checking data output value, and the output of the fuzzy inference system corresponding to the same checking data input value, which is the one associated with that checking data output value. The checking error records the RMSE for the checking data at each epoch. The ANFIS Editor GUI plots the checking error versus epochs curve as the system is trained. The average testing error for the checking data set is 0.03382 and training and cheking error are 0.0049 and 0.0052 respectively. When checking data is tested against the trained FIS, it looks satisfactory and it shown in figure 3.14b.



(a) (b)

Figure 3-14: a) Training and checking errors b) Testing the FIS with checking data set

As shown from figure 3-14a the training and checking errors decrease as the training starts and became constant after the number of epoch reaches about 5. The training error is about 0.0049 and the checking error is about 0.00518 which are acceptable. The RMS errors are negligibly small, and hence the ANFIS captured the essential dynamics of underlying system.

And figure 3-14b shows that how much the training and checking data are related and they are superimposed each other. The following figure shows the structure of the proposed ANFIS controller.

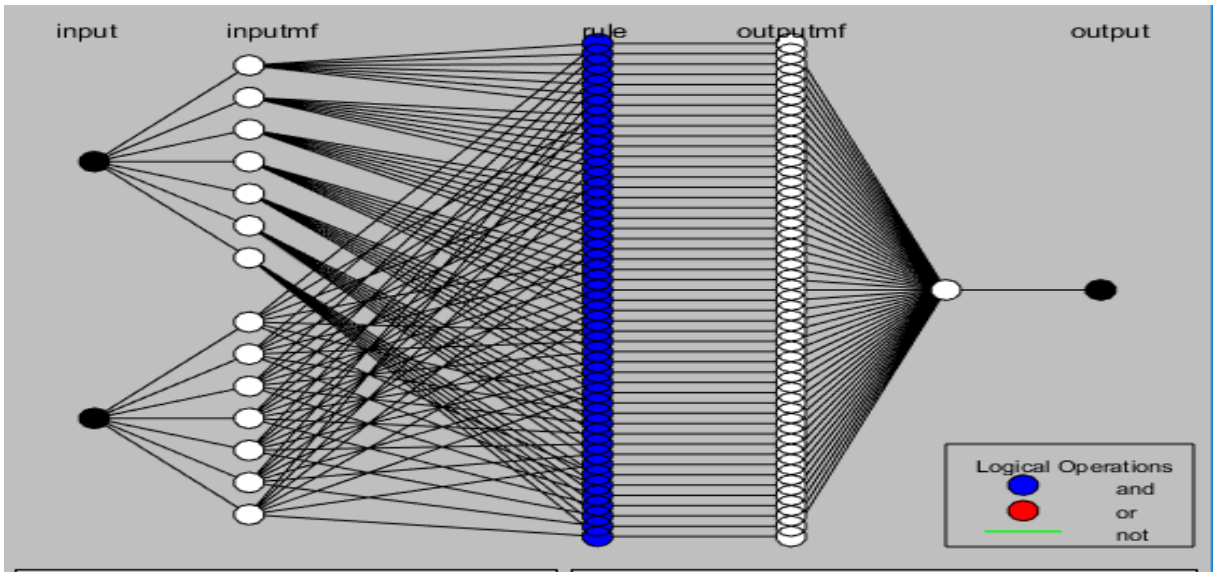


Figure 3-15: Schematic of proposed ANFIS controller

From the ANFIS structure shown in Figure 3-15, it has been observed that when the values of the premise parameters are fixed, the overall output can be expressed as a linear combination of the consequent parameters. Table 3.4 shows the propose ANFIS features.

Table 3-4: Proposed ANFIS features

Parameters	**Options chosen**	**Parameters**	**Options chosen**
Type	Sugeno first order	Output Mf type	linear
And method	Min	No. of rules	49
Or method	Max	No. of epochs	60
Implication method	Prod	No. of training data pairs	70,000
Aggregation method	Max	No. of checking data pairs	30,000
No. of inputs	2	No. of nodes	131
No. of input Mf type	7	No. linear parameters	147
Type of input mf type	Triangular Mfs	No. of nonlinear parameters	42
No. of output Mf	49	Total No. of parameters	189

CHAPTER FOUR

4. SIMULATION RESULTS AND DISCUSSIONS

4.1 Introduction

This chapter states about the overall SIMULINK blocks of IVCIM, all simulation results with brief discussion and analysis. The simulation result and discussion include the result of the induction motor without controller, with GA optimized PI controller, and with neuro-fuzzy controller. The result using PI and ANFIS was checked at different operating conditions such as, without load and with load, with step change and variable reference speed, reversal of speed, stair case reference speed , and finally we tried to check the parameter variation effect on both PI and Neuro-Fuzzy controllers.

Induction motor parameters used for simulation are shown in the table 4-1.below.

Table 4.1: Induction motor parameters.

Parameters	**Symbol**	**Numerical values**	**Unit**
Rated voltage	V	380	V
Rated power	P	3.5	hp
Rated speed	ω	2970	rpm
Stator resistance	Rs	0.288	Ohm(Ω)
Rotor resistance	Rr	0.158	Ohm(Ω)
Stator inductance	Ls	0.0425	H
Rotor inductance	Lr	0.0412	H
Mutual inductance	Lm	0.0418	H
Moment of Inertia	J	0.4	Kg.m^2
Viscous Friction coefficient	F	0.001	Nms^{-1}
No of Poles	p	2	-
Frequency	f	50	Hz

The overall Simulink model of speed control of vector controlled IM is shown in the following figure 4.1. The Simulink model includes indirect field oriented controller (IFOC), axis transformations (a-b-c to d-q and vice versa), space vector pulse width modulation (SVPWM) based 3-phase inverter and the induction motor sub- models. The sub blocks inside each models are shown in Appendix-B.

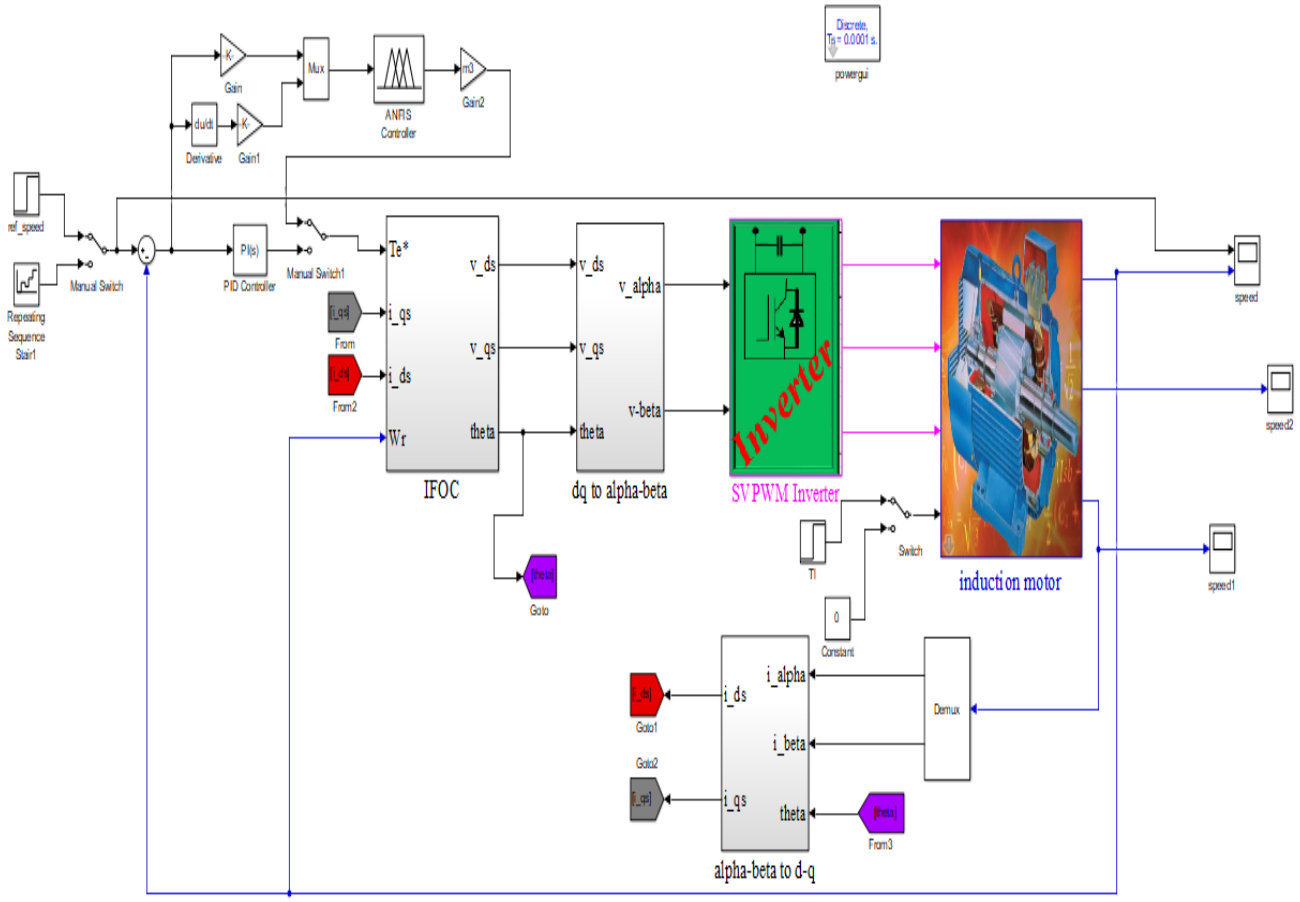


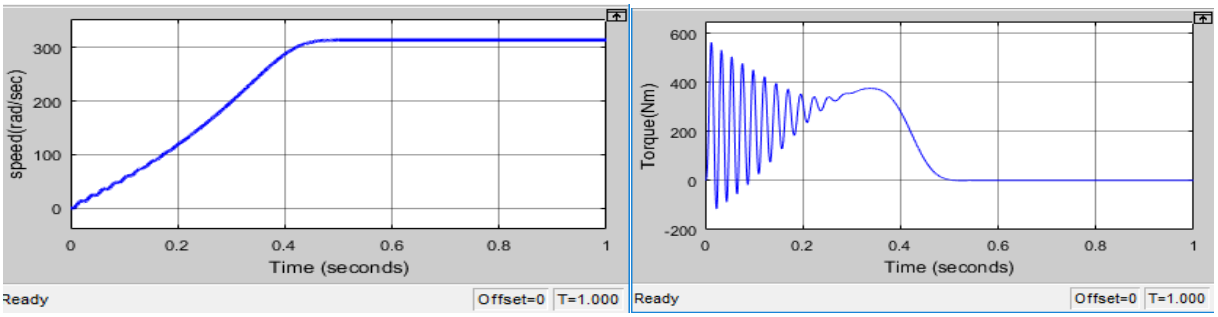
Figure 4-1: Over all Simulink block of speed control of IVCIM.

4.2 Result and discussion

To analyze the performance of the IVC induction motor drive, a SIMULINK model was created. A 3.5hp, 380V, 50Hz squirrel cage induction motor was used to study the dynamics of the drive. The parameters of the induction motor considered in this study are summarized in Table 4-1 above. Result and discussion in general carried out without controller and with controller.

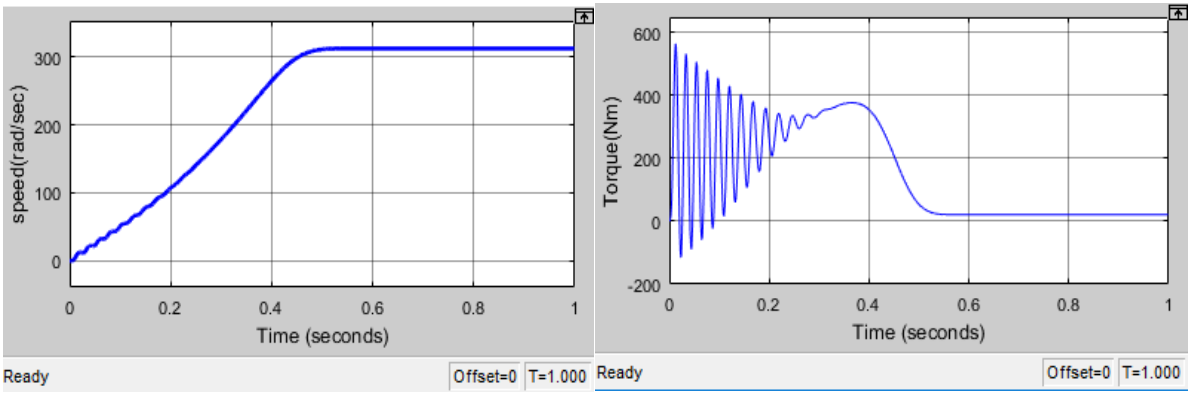
4.2.1 Speed and torque responses of 3-phase induction motor without controller

The following figures (4.2-4.4) show the responses of speed and torque of IM without controller. For the simulation 380V voltage magnitude and 50Hz frequency is used for each phase.



(a) (b)

Figure 4-2: a) Speed and b) torque response of IM at no load and speed of 3000 rpm (314 rad/sec)



(a) (b)

Figure 4-3: a) Speed and b) torque response of IM at load Torque of 5Nm.

As shown figure (4.2a and 4.3a) the motor runs around the synchronous speed which is 314 rad/second (3000 rpm) and it decreases when load is applied. And figure (4.2b and 4.3b) show that

the torque is going to be zero without load and to 5 with a load torque of 5Nm as the speed reaches to the synchronous speed which is expected.

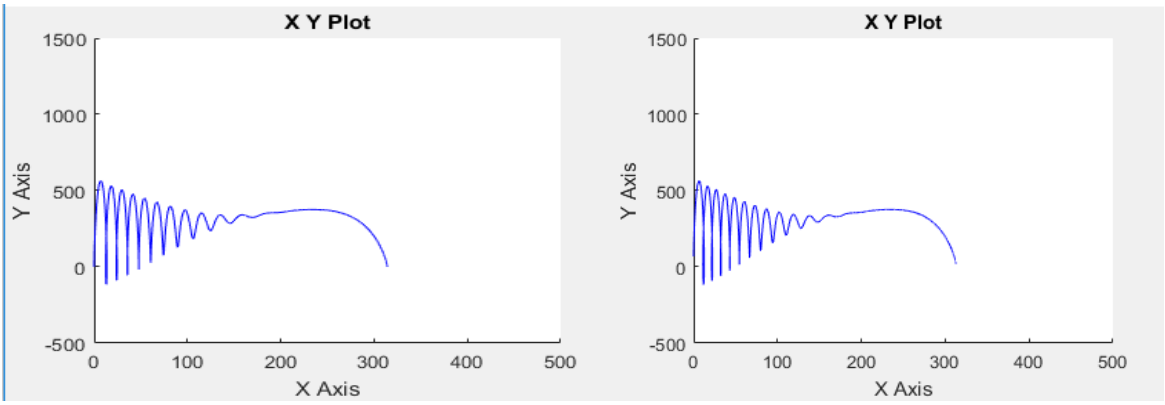


Figure 4-4: Torque speed characteristics of IM without load and with load of 5Nm.

We know the fact that, 'If rotor runs at the synchronous speed, then it will appear stationary to the rotating magnetic field and the rotating magnetic field will not cut the rotor. So, no induced current will flow in the rotor and no rotor magnetic flux will be produced, so no torque is generated'. So, using this fact, the result that we have shown in the torque simulation result is zero for no load and 5Nm where load torque is 5Nm which is an acceptable. To run the motor to the speed below and above rated speed, we need to design the controller. The responses of the motor using PI and Neuro- Fuzzy controllers at different operating conditions are shown in the next sections.

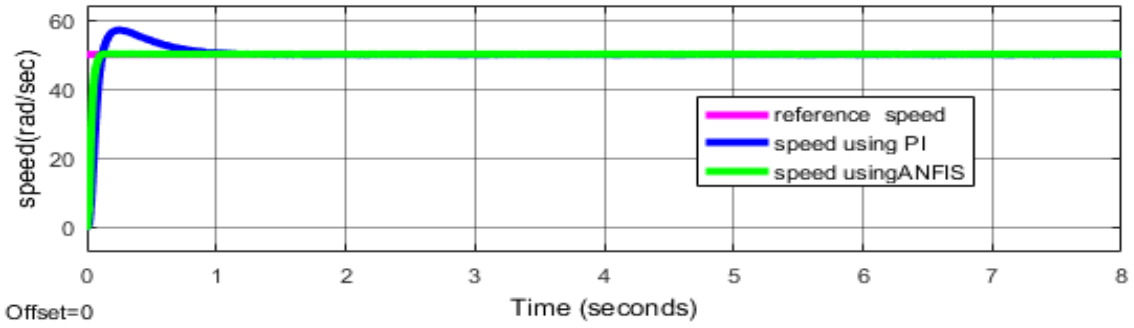
4.2.3 Simulation results using PI and Neuro-Fuzzy controllers

In order to prove the superiority of the proposed NFC, a comparison is made with the response of GA based PI speed controller based IM drive. The PI controller was replaced by an ANFIS after the simulation using PI controller is carried out. The performance of the proposed NFC based IM drive is investigated at different operating conditions. Some of the conditions that we have been considered are:

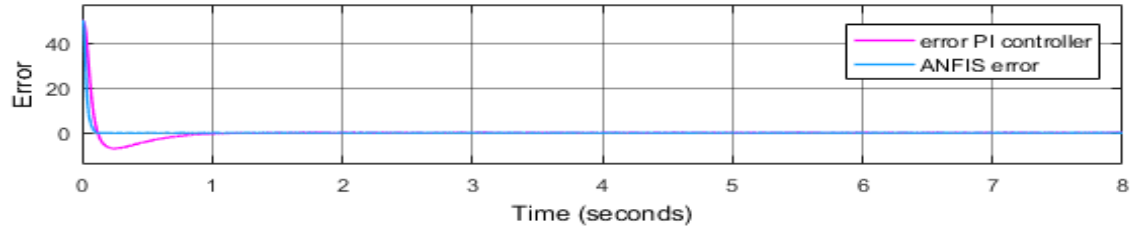
- Responses without load
- Responses with load disturbance
- Input parameter variation (speed and torque) and
- Motor parameter variation, specifically rotor resistance, Rr.

Simulation results and analysis without load

The dynamic behavior of the motor at no load and reference speed of 50 rad/sec. is illustrated in the next simulation results of figure 4.5 and 4.6.The simulation results for speed, torque and current are discussed in the following figures (4.5-4.6).



(a)

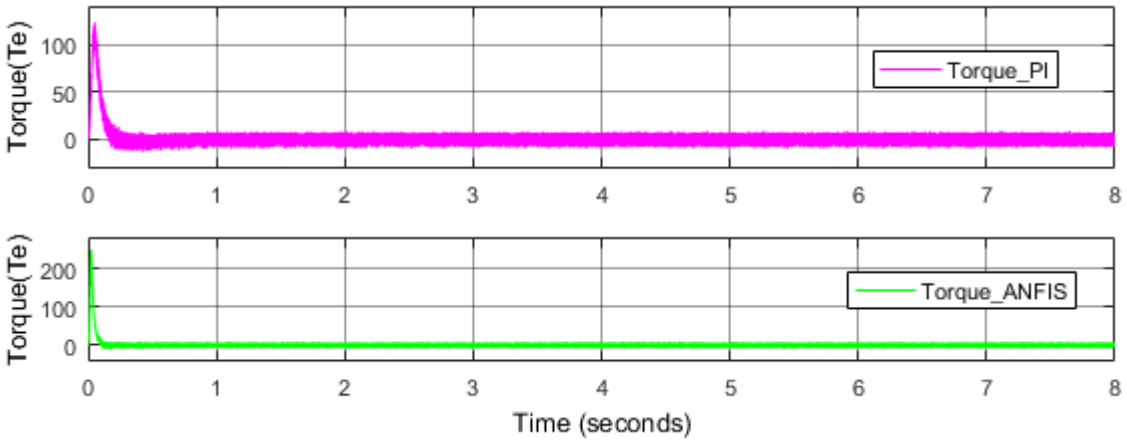


(b)

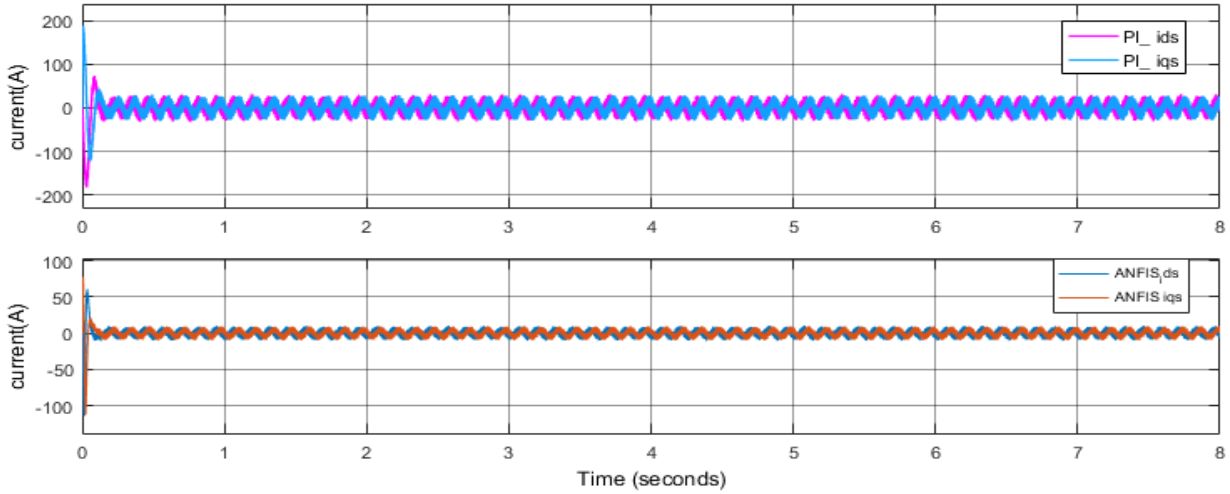
Figure 4-5: Performance Comparison of IVC with PI and ANFIS Speed controller a) speed (rad/sec.) and b) error at reference speed 50 rad/sec and no load (0 Nm).

The simulation results indicate that the ANFIS controller has the ability to follow the speed of the induction motor drive effectively. The ANFIS controller shows a faster response in comparison to the PI speed controller. The conventional PI speed controller based scheme shows considerable overshoot in the speed. With the use of ANFIS, the speed overshoot is reduced as shown in figure 4.5. It was observed from the simulation results that by using the neuro-fuzzy (ANFIS) control, for the set speed of 50 rad / s, the speed reaches its desired set value at 0.201 seconds and the overshoot is also very small compared to that of PI controller. This shows the effectiveness of the designed neuro-fuzzy controller & the designed neuro-fuzzy controller tries to speed up the performance of the drive. Further, it can also be observed that using the ANFIS control, the system stabilizes in a very less time compared to the PI controller because of the training process of the

ANN involved & the proper selection of the rule base. From the simulation results shown in the Figure 4.5, the responses of speed using ANFIS takes lesser time to settle & reach the desired value compared to that of PI controller. And figure 4.6 shows the torque and current responses.



(a)



(b)

Figure 4-6: Performance Comparison of IVC with PI and ANFIS Speed (a) Torque (Nm), (b) current (amp.) at reference speed 50 rad/sec and no load (0 Nm).

Torque characteristics for a set reference speed of 50 rad/sec. and at no load is shown in the Fig. 4.6a. The torque is settle at less time and its oscillation peak is low using ANFIS compared to that of PI controller. From the simulation result of figure 4.6a, we arrived at a conclusion that when the motor is operating at lower speeds, the torque is more. Hence, the machine requires more torque

to attain the set speed. Once the machine reaches the set speed of 50 rad/s, the average torque of the machine becomes nearly zero.

The current and Torque initial peak values of ANFIS are high compared to that of PI controller. But this is due to its settling time is very small. Since the peak values are high for a fraction of microseconds it doesn't affect the motor operation. Even though the peak torque at initial is high, it oscillates around zero fast compared to PI controller. So, the result shows the better performance is gained using ANFIS than PI. The following table summarizes the comparison of the two controllers for speed response. The current oscillates constant and it oscillates around zero and the starting current of ANFIS is less than that of PI controller.

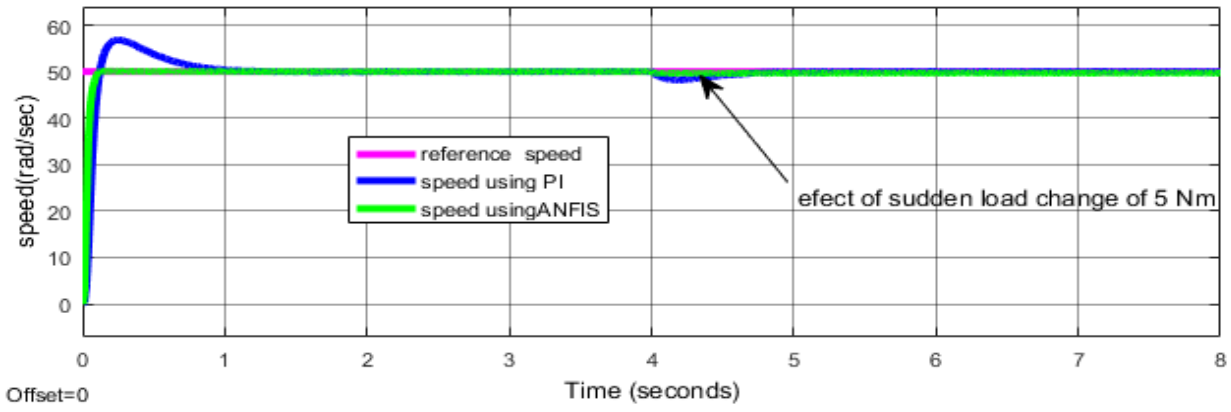
Table 4-2: Performance comparisons of controllers

Controller	**Peak Overshoot**	**Rise time (sec.)**	**Settling time (sec.)**	**Steady state error**
PI	14.368%	0.0697	1.252	0.04
ANFIS	0.475%	0.0458	0.201	0.03

The result shows better performance of NFC based IM drive as compare to conventional GA tuned PI controller based IM drive by overshoot, rise and settling time and also in steady state error.

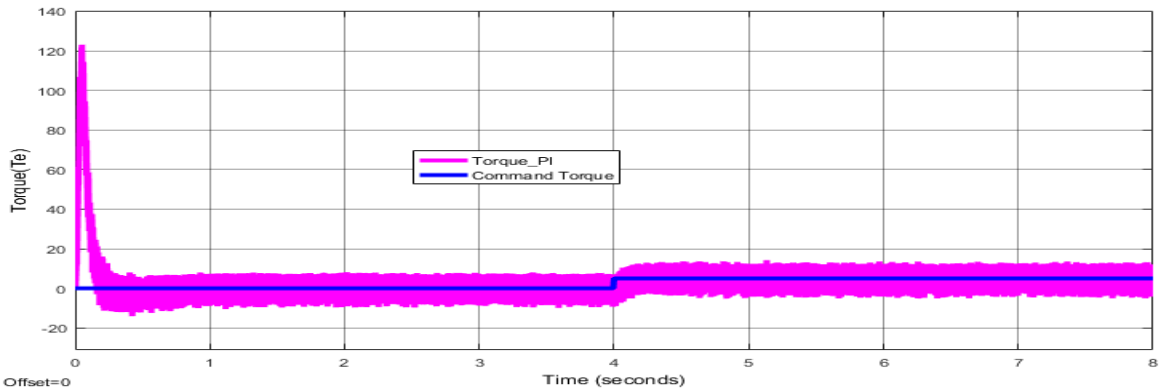
Simulation results and analysis with load.

The simulation result with include for a sudden load change at instant of time and sequential load commands. Fig. 4-7(a-c) shows the speed and torque responses for a sudden load change from no load to 5Nm at 4 seconds.

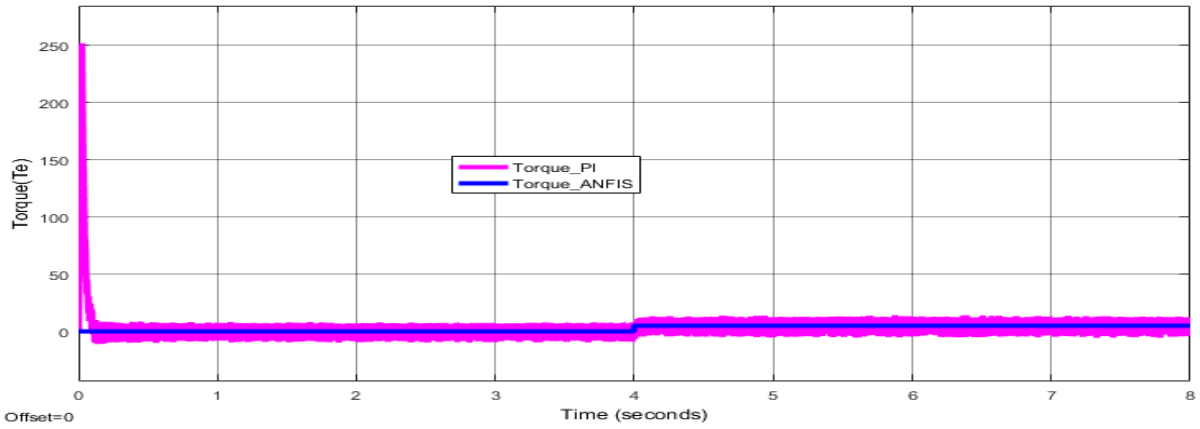


(a)

Figure 4.7a below shows the load disturbance rejection capabilities of each controller when using a step load change from 0 to 5Nm at 4 seconds. The ANFIS controller at that moment returns quickly to command speed, whereas the PI controller maintains a steady state error, was affected by change in load, but ANFIS is less affected by the sudden change in load.



(b)



(c)

Figure 4-7: a) speed and b) Torque response PI and c) Torque response ANFIS at sudden load change of 0Nm to 5 Nm and reference speed of 50rad/sec.

Fig. 4.7b shows the electromagnetic torque response, that the proposed neuro-fuzzy controller is more robust to load disturbance as compared to PI controller. The torque oscillates around zero until the time is reached to 5 seconds in both controllers. After 4 seconds, the electro mechanical torque settles to 5Nm which is load torque as expected. But the torque response using ANFIS is

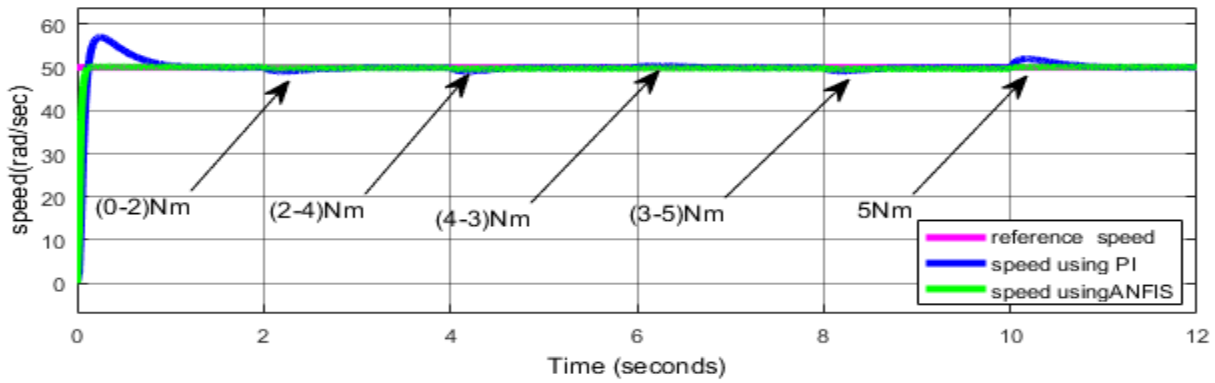
better in settling time, overshoot, and oscillation peak is low compared to that of PI controller. The Load torque was initially 0 Nm and it changed to 5Nm after 4 seconds.

Here, the PI controller was more affected by a sudden load change than ANFIS. So, the result shows that the proposed ANFIS is more robust to load disturbance as compared to PI controller. Figure 4.7b shows that the electromechanical torque response of PI and Neuro-Fuzzy controller. As shown from the figure the peak torque for ANFIS is greater than that of PI controller. This is due to that it settles fast which is below 0.1 second rather greater than 0.3 seconds for PI. The torque is high for negligible time which is in microseconds. So it doesn't affect the motor operation.

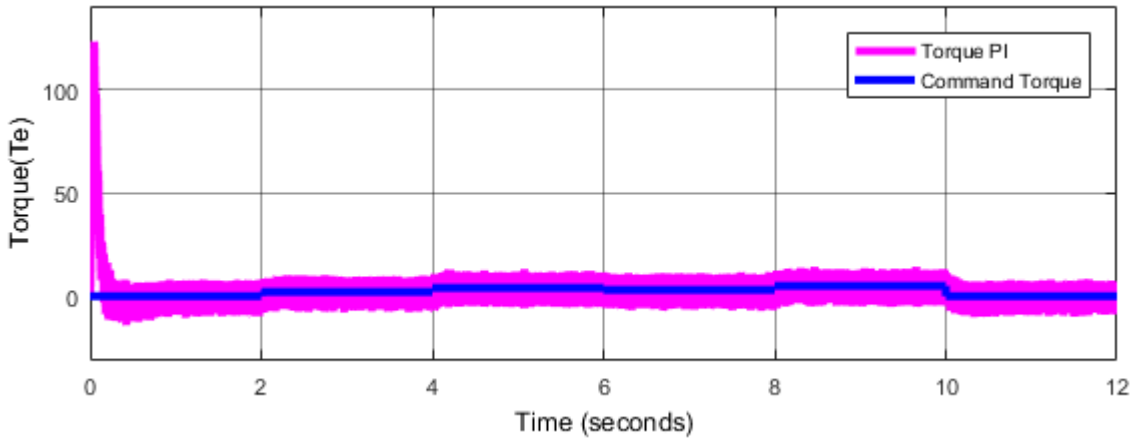
Input parameter variation (Stair case reference speed and load torque)

This is used to check whether the motor operates for wide range of speed i.e.; for lower speed and high speed ranges and to check the effect of variation load torque at each time on the motor. Figure 4.8 and 4.9 show the response of speed, torque and current for reference speed and load torque variation using both PI and ANFIS controllers. The reference speed is a sequential stair case input (0 40 30 20 80 100) rad/sec at sampling time of two seconds and the load torque variation is (0 2 4 3 5 0) at sampling time of two seconds. The result shows that the speed is tracking the reference speed as expected and the speed using ANFIS is better as discussed previously.

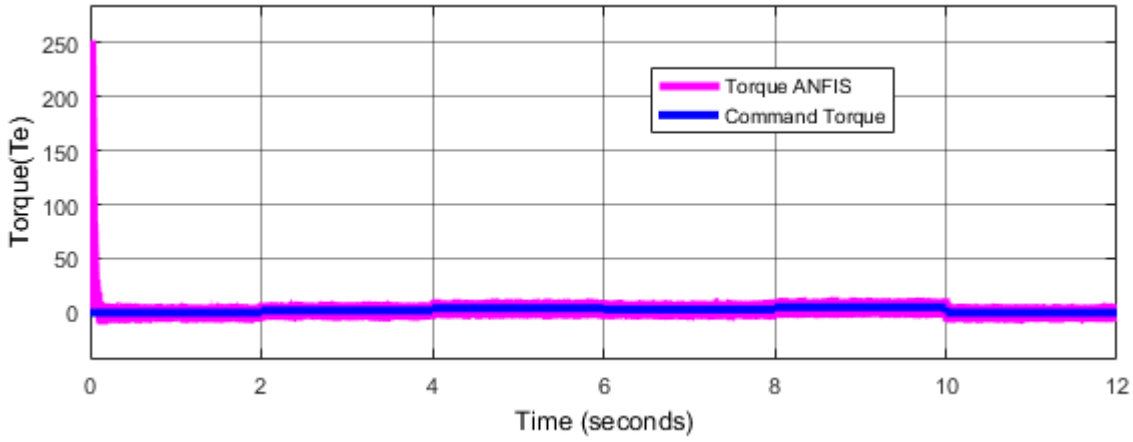
Load Torque variation (0 2 4 3 5 0) Nm.



(a)



(b)



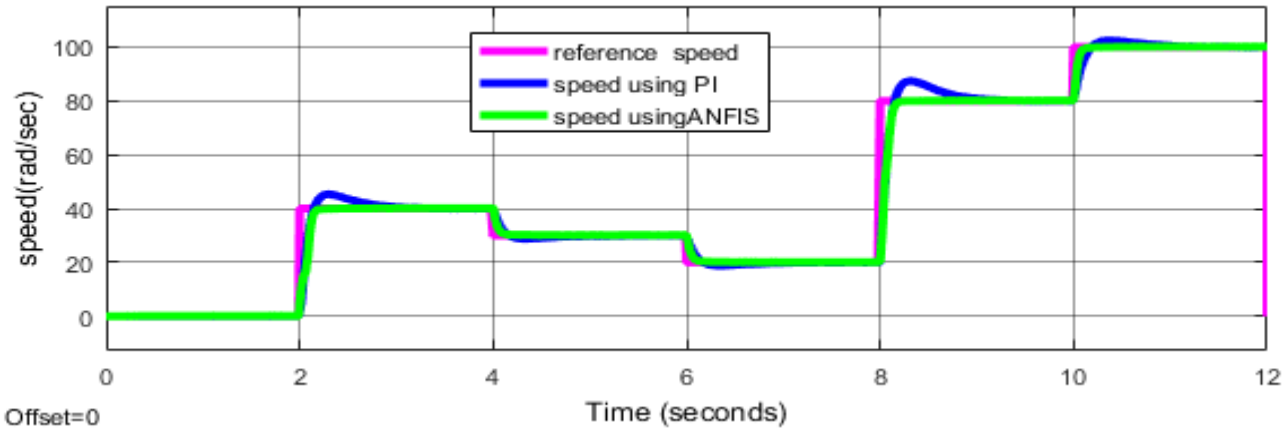
(c)

Figure 4-8: a) speed b) torque response of PI and c) torque response ANFIS controller for load torque variation of (0 2 4 3 5 0) Nm.

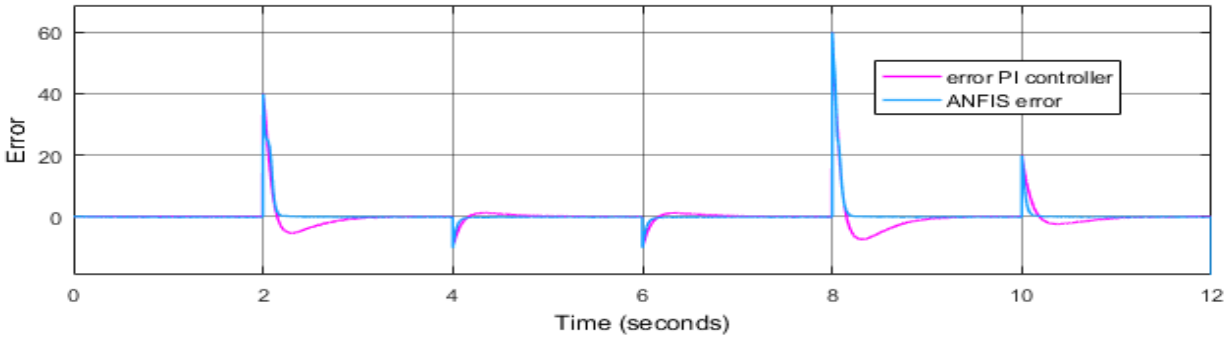
Here, figure 4.8a shows the speed response using ANFIS is less affected by load variation compared to PI controller. Figure 4.ba is the response of electro mechanical torque response. The result shows the torque is 0 until the time reached to 2 second, after 2 second it changes to 2 Nm until time is 4 second, again it changes to 3 Nm from time is equal to 4 second to 6 seconds, after 6 second the torque is changed to 5Nm until 8 seconds and finally it is returned back to 0Nm.

The torque using the ANFIS follows the command torque as fast as possible compared to that of using PI controller. There for the ANFIS controller has good speed and torque responses with and without load when compared with PI controller.

Reference speed variation (0 40 30 20 80 100) rad/sec.



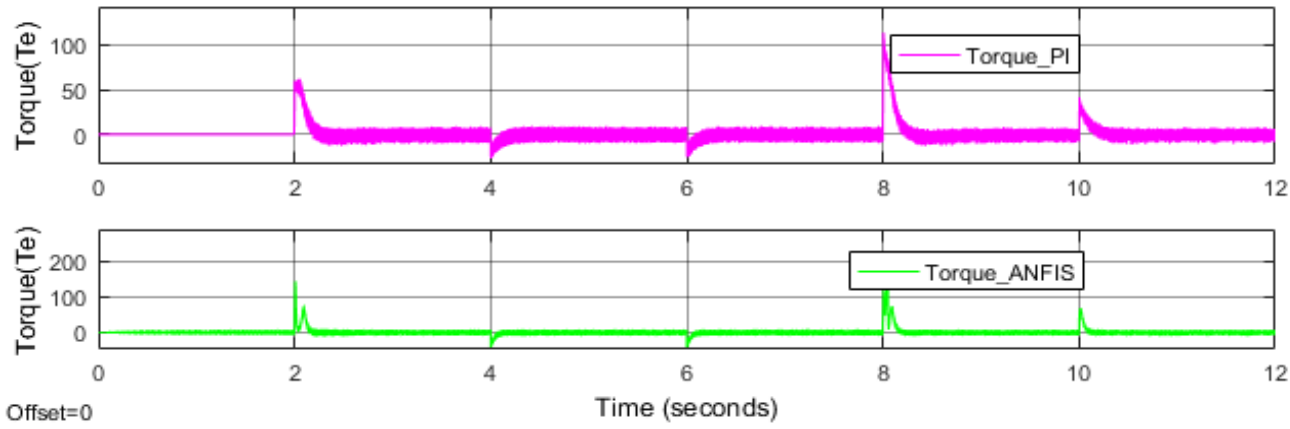
(a)



(b)

Figure 4-9: a) speed response and b) error of PI and ANFIS for sequential stair reference speed of (0 40 30 20 80 100) rad/sec and at no load.

Here, figure 4.9 shows the speed and error response of both PI and ANFIS controllers at variable speed of (0 40 30 20 80 100) rad/sec and at no load. The result shows that the speed using ANFIS is better in overshoot, settling time, rise time and steady state error as before. The figure also shows that we can operate the motor in wide range of motor (low range and high range). Figure 4.9b shows the error response of PI and ANFIS controllers. And as we have seen from the figure the error of ANFIS settles fast to zero compared to that of PI controller.



(a)

Figure 4-10: a) Torque responses of PI and ANFIS for sequential stair reference speed of (0 40 30 20 80 100) rad/sec at no load.

Figure 4.10 shows the torque and current responses of PI and ANFIS controllers for variable speed input. Figure 4.10a shows the torque response and the response using ANFIS is better in settling time and its oscillation peak is low though its peak starting at each turn is high compared to that of PI controller. Even if the starting torque is higher in ANFIS, but it is for a minimum microseconds. So, there is no negative impact on the operation of the motor.

Motor Parameter variations

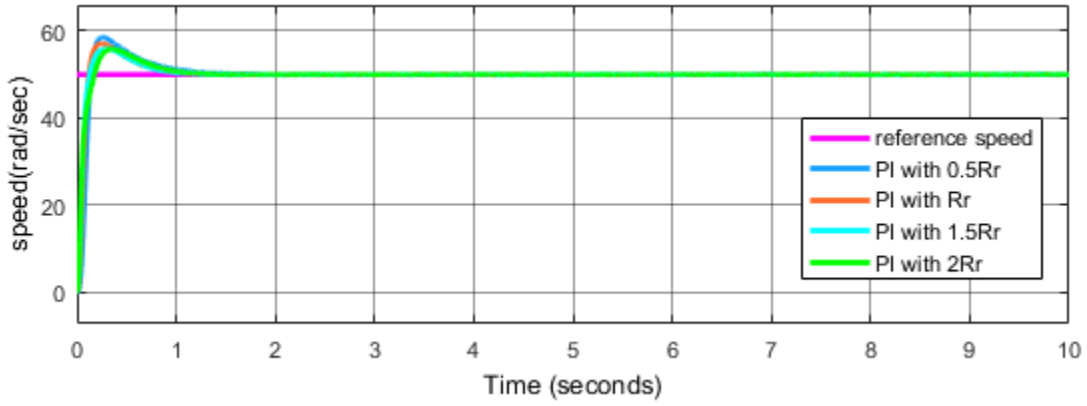
The performance of decoupling control methods based on machine models can be influenced by a mismatch between the parameters values being used in the controller and the actual machine parameters [5]. On the other hand, the parameters of the machine may change during the operation of the drive, causing deviation between the corresponding signals of the model and the machine[27]. The machine parameters (Rs, Rr, Lm, and Lr) will vary due to temperature, frequency, and saturation effects. The indirect vector controller dependent on the rotor resistance Rr, the mutual inductance Lm, and self -inductance of the rotor, Lr. In particular, the rotor resistance value changes 100% with temperature and frequency. This causes the actual flux and torque deviate from their set values. Also, the inductances change with the saturation of magnetic material.

The indirect vector controlled scheme depends heavily on motor parameters, especially on rotor time constant, τr which is the ratio of rotor inductance and rotor resistance. The rotor resistance variation is due to temperature and frequency and the inductance variation is due to magnetic

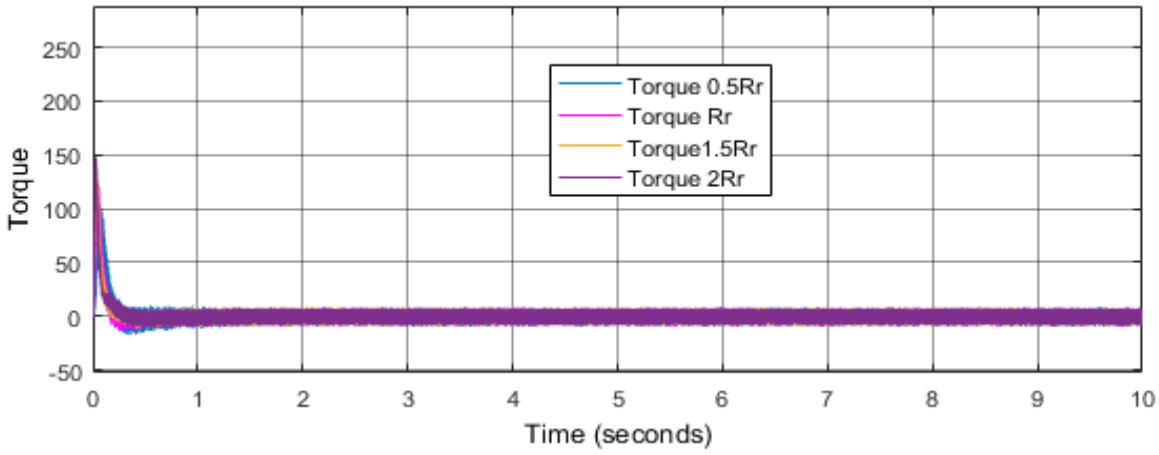
saturation. So, it is not surprising that IFOC drives are especially sensitive to parameters over a wide speed range. Since our assumption in modeling (chapter 2) is without magnetic saturation, this section focuses on the analysis of the effect of rotor resistance variation.

Hence, the drive performance can be related to purely slip calculation which is proportional to the rotor time constant, τr. The variation in rotor resistance affects τr and causes misalignment of ids, iqs pair with the rotor flux axis ψr. The misalignment causes a component in orthogonal axis to vary hence resulting in over fluxing or under fluxing. In an induction motor controlled by IFOC, the synchronous reference frame angle is indirectly calculated from the measured position and slip angle.

In high performance drives with induction motors, accurate information about motor parameters is crucial. Unfortunately, these parameters are not constant, depending on the temperature. The performance of the IM using PI controller and ANFIS controller for indirect vector control are studied under constant rotor resistance, ±50% and 100% change from nominal value in rotor resistance respectively, the results observed for speed and torque are shown in figure 4.11a and figure4.11b.



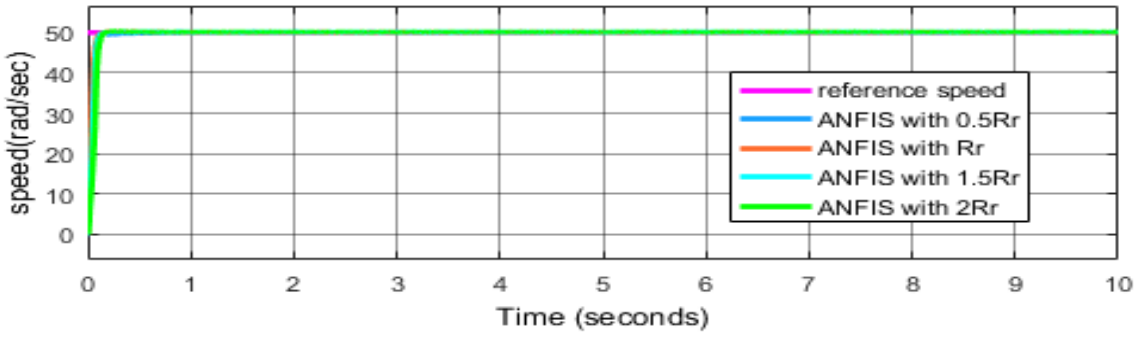
(a)



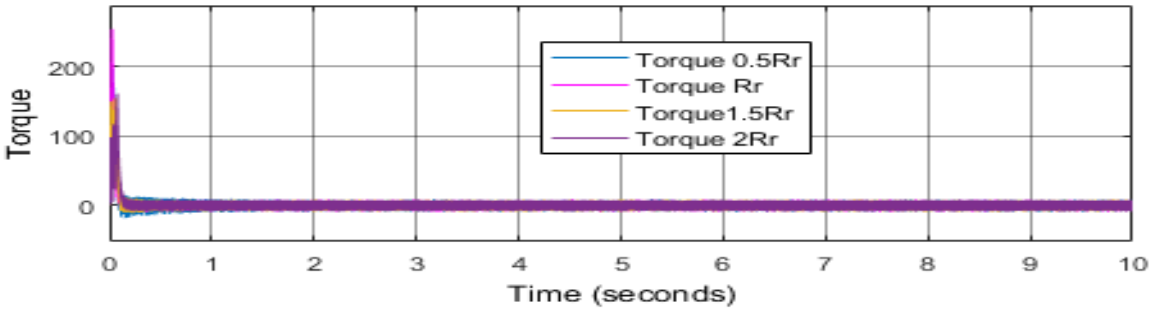
(b)

Figure 4-11: a) speed and b) torque response of PI and controller at Rr, 0.5Rr, 1.5Rr and 2Rr.

Figure 4.11 and 4.12 give the results of speed and torque responses using PI controller and ANFIS respectively which considering the changes of rotor resistance value, Rr. Figure 4.11a shows when Rr was reduced to half rated value, the results show the increasing of delay, rise time and overshoot of speed response compared to Rr, 1.5Rr and 2Rr. But, to achieve steady state ANFIS shows better performance. It is only taking 0.201 second for ANFIS compare to PI is 1.252 seconds even the value of Rr was changed. Figure 4.12 shows the speed and torque responses of ANFIS controller at nominal value of Rr, 0.5Rr, 1.5Rr and 2Rr changes. The result is at reference speed of 50 rad/sec and at no load. And the response is better compared to that of PI controller which is shown in figure 4.11 above.



(a)



(b)

Figure 4-12: a) Speed and b) torque response of ANFIS controller at Rr, 0.5Rr, 1.5Rr and 2Rr.

Here, figure 4.11(b) and 4.12(b) show the torque response using PI and ANFIS controllers. As shown from the results for PI controllers the torque with 2Rr has high peak value compared with 0.5Rr, Rr and 1.5Rr although it is going to settle fast compared to others. The torque with ANFIS is better in settling time, rising time, and has low oscillation peak compared to that of PI controller even if the rotor resistance is vary.

Here, it is seen that the performance of the motor is almost similar in both the cases of constant rotor resistance as in steady state conditions, as well as a ±50% and 100% change in rotor resistance that corresponds to the dynamic conditions of motor operation, which leads to parameter variations. The only downside of control strategy using PI controller is, the peak overshoot that occurs at the very start of the operation and high settling time. Such a sharp change in the motor speed at the start can lead to hazardous situations where the motor windings can burn, however, under ANFIS controller not only the peak overshoot is reduced to zero, but the amount of oscillations under steady state are reduced, thus giving promising results than PI controller.

CHAPTER FIVE

5. CONCLUSION AND RECOMMENDATION

5.1 CONCLUSION

The performance of ANFIS based intelligent controller for the speed control of indirect vector controlled, SVPWM voltage source inverter fed induction motor drive has been verified and compared with that of GA based conventional PI controller performance. The simulation results obtained have confirmed the very good dynamic performance and robustness of the Neuro-Fuzzy controller during the transient period and during the sudden loads. It is concluded that the proposed intelligent controller has shown superior performance than that of GA optimized PI controller. Generally, ANFIS shows better performance in the following performance results.

- A novel Neuro-fuzzy controller has advantages such as reduced number of rules, faster speed of operation and no need for modifications in membership function by conventional trial and error method for optimal response. This makes NFC an easy-build and robust controller. .
- The simulation results show that the designed neuro-fuzzy controller realizes a good dynamic behavior of the motor with a rapid settling time, no overshoot and has better performance than GA based PI controller.
- The performances of the proposed NFC based drive have been investigated at various operating conditions. A performance comparison between PI based drive, and the proposed NFC based drive has been presented. The proposed NFC based IM drive has been found to be more robust during change in load condition.
- Since fuzzy logic has tolerance for imprecision of data and neural network has tolerance for noisy data, their combination neuro-fuzzy is having good tolerance for the parameter variation particularly rotor resistances.

From MATLAB simulation results shown above, the characteristic curves of speed, torque, and current vs. time at different operating conditions were observed. The outputs take less time to stabilize, it was observed that the motor reaches the rated speed very quickly in 0.201 second for ANFIS and 1.252 second for the GA based PI controller. In addition to this the peak overshoot for the ANFIS is 0.475% which is acceptable unlike the PI controller which has 14.368% peak overshoot. The ANFIS controller also shows better in steady state error which is 0.03 and that of

PI controller is 0.04. Collectively, these results show that the ANFIS controller provides faster settling times, has very good dynamic response & good stabilization compared to PI controller.

The proposed controller is evaluated under simulations for a variety of operating conditions of the drive system and results demonstrate the effectiveness of these control structures to improve the performance of the drive system.

5.2. RECOMMENDATIONS

Research is endless. The work done in this thesis could be further researched upon and extended by hardware implementation. It is worth noting that the control approach proposed here can be effortlessly extended to any other high performance electric drives. In the future work, one will focus on the experimental validation of this proposed controller.

The developed control strategy is not only simple but also reliable and may be easy to implement in real-time applications by integrating ANFIS with dsp. So some body can implement easily using some interfacing cards after integrating ANFIS with dsp.

References

[1] M.G.SAY, "*Alternating Current Machine,*" UK, Pitman, 1976, pp 250-301.

[2] K. Bose, "*modern power electronics and AC drives,*"UK, Austin Hughes,2006,pp 1-46.

[3] N. Tesla, and M. Dolivo-dobrovolsky, "*Induction Motors,*" U. States, fourth ed,pp. 1–28, 1976.

[4] N. E. D. Mohan, "*Electric Drives an Integrative Approach,"* USA, MNPERE Minneapolis 2003,pp 337-364

[5] E. Y. Y. Ho and P. C. Sen, "*Decoupling control of induction motor drives*," *Ind. Electron. IEEE Trans.*, vol. 35, no. 2, pp. 253–262, 1988.

[6] G. V Deshpande, "*speed control of induction m otors using h ybrid pi*," vol. 6, no. 5, pp. 2253–2261, 2013.

[7] A. Hashim and O. Ahmed, "*Speed Control of Vector Controlled Induction Motors Using Integral-Proportional Controller*," vol. 15, no. 2, pp. 72–79, 2014.

[8] A. Mishra, V. Mahajan, P. Agarwal, M. Ieee, and S. P. Srivastava, "*Fuzzy Logic Based Speed and Current Control of Vector Controlled PMSM Drive,*" IEEE Trans. pp. 0–5, 2012.

[9] G. V Deshpande, "*SPeed Control of Induction Motors Using Hybrid Pi and ANN,"* vol. 6, *IEEE Trans* vol. 5, pp. 2253–2261, 2013.

[10] Y. S. Kung, C. M. Liaw, and M. S. Ouyang, "*Adaptive speed control for induction motor drives using neural networks," IEEE Trans. Ind. Electron.*, vol. 42, no. 1, pp. 25–32, 1995.

[11] P. G. Scholar and D. Eee, "*anfis based vector controlled induction motor,*" IEEE Trans.vol. 2, no. 3, pp. 152–156, 2014.

[12] R. Kumar, R. A. Gupta, and R. S. Surjuse, "*Adaptive Neuro-Fuzzy Speed Controller for Vector Controlled Induction Motor Drive,*" *Asian Power Electron. J.*, vol. 3, no. 1, pp. 8–14, 2009.

[13] W. Bin, L. Yongqiang, K. Samir, and Z. Navid, *Power Conversion and Control of Wind Energy Systems*, vol. 53, no. 9. 2011.

[14] Paul C. Krouse, "*Analysis of Electrical Machinery and Drive Systems,"*USA, IEEE power enginering socity, 2002,pp, 109-555.

[15] R. J. Lee, P. Pillay, R. G. Harley, and K. G. V Avenue, "*D, Q Reference Frames for the Simulation of Induction Motors," Electr. Power Syst. Res.*, vol. 8, no. 1, pp. 15–26, 1984.

[16] S. Shah, A. Rashid, and M. K. L. Bhatti, "*Direct Quadrate (D-Q) Modeling of 3-Phase Induction Motor Using MatLab / Simulink,"* vol. 3, no. 5, 2012.

[17] J.Westin,D. Bystrov and "*control scheme and controller design ,"* pp. 81–101.

[18] M. Popescu, :*induction motor modelling," Modeling of 3-Phase Induction Motor Using MatLab / Simulink,vol 2, no 3, pp. 30-55, 2000.*

[19] C. Singh, "*Genetic Algorithms Based PID controller Design," Int. J. Enginering Dev. Res.*, vol. 3, no. 3, pp. 2–5, 2015.

[20] Uchikawa, Y., Furushashi, "*Adaptive Neuro Fuzzy Inference System,"* IEEE Trans.Control System Technology, Vol-8, no-2, pp.270 278, March2000.pp. 168–192.

[21] J. R. Jang, "*Neuro-Fuzzy Modeling,*" IEEE Trans.Control System Technology, no-2, , March2000.vol. 83, no. 3, 1995.

[22] D. Bystrov and J. Westin, "*logic systems matlab toolbox gui*," A Summary. Part 1: PI Controller Tuning Rules.î , Proceedings of Irish Signals And Systems Conference, June 1999.

[23] A. Al-hmouz, J. Shen, S. Member, R. Al-hmouz, and J. Yan, "*Modeling and Simulation of an Adaptive Neuro-Fuzzy Inference System (ANFIS) for Mobile Learning,"* vol. 5, no. 3, pp. 226–237, 2012.

[24] A. O. Cruz and N. C. . Mestrado, "ANFIS : *Adaptive Neuro-Fuzzy Inference Systems,"* p. 53, 2009.

[25] R. Burns, "*Advanced control engineering,*" *Mar. Eng.*, USA, Butterworth, 2001,pp. 464,

[26] I. Ghajar, A. Najafi, S. A. Torabi, M. Khamehchiyan, and K. Boston, "*An Adaptive Network-based Fuzzy Inference System for Rock Share Estimation in Forest Road Construction,"* no 3. 2001, pp. 313–328, 2007.

[27] A. c. w. N. c. c. CHAN, w. s. LEUNG, "*Adaptive Decoupling Control of Induction Motor Drives,"* vol. 31, no. 1, pp. 41–47, 1990.

[28] L.Zhen and L.Xu, "*Fuzzy learning enhanced speed control of an indirected field-oriented induction machine drives,*"IEEE Trans.Control System Technology, Vol-8, no-2, pp.270 278, March2000.

[29] P. Vas, "*Sensorless Vector and Direct Torque Control,"* London, U.K.: Oxford Science Publication, 1998.

[30] R. Krishnan, '*Electric motor drives modeling, analysis and control'* (PHI Pvt. Ltd., New Delhi, 2003).

[31] G. Subba Reddy, "*Vector Controller based Speed Control of Induction Motor Drive with 3-Level SVPWM based Inverter"*, International Journal of Emerging Trends in Electrical and Electronics (IJETEE) Vol. 1, Issue. 4, pp. 1-11,March-2013

[32] S. D. Huang, Y. N. Wang, J. Gao, et al, "*The vector control based on MRAS speed sensorless induction motor drive,*" Conference of the 5th ICA, Hangzhou, China, pp. 4550-4553, 2004.

[33] C. Schauder, "*Adaptive speed identification for vector control of induction motors without rotational transducers,*" IEEE Transactions on Industry Applications, vol. 28, pp. 1054-1061, 1992.

[34] P. Vas, "*Artificial-Intelligence-Based Electrical Machines and Drives-Application of Fuzzy, Neural, Fuzzy-Neural and Genetic Algorithm Based Techniques*,"New York: Oxford University Press, 1999.

Appendix

A. Mathematical description of Hybrid Learning algorithms

A.1 Back propagation learning algorithm

The premise parameters {a, b, c} in Eqs.2.3and2.4are adaptive parameters that can be trained to get the parameters in accordance with its environments. Suppose to have an adaptive network and similar to the Fig.3-9b, where the network consists of five layers and has a total of N (L) node in layer-L, then the number of square error in the L layer to 'p' data is$1 \leq p \leq N$, and it can be defined as follows[21]:

$$E_p = \sum_{k=1}^{N(L)} d_k - X_{k,p}^L \qquad (A.1)$$

Where d_k is the k[th] component of the vector of the desired output, while $X_{k,p}^L$ is k[th] component of the vector of actual output generated by adaptive network with input from the input vector p. The main goal of adaptive learning system is to reduce errors that occur in the Eq.A.1. An early stage of learning begins by calculating the error rate of the output ith node and L layer, with derivation equation as follows:

$$\in_{L,i} = \frac{\partial E_p}{\partial_{i,p}^L} = -2\left(d_{i,p} - X_{k,p}^L\right) \qquad (A.2)$$

For internal nodes in the l layer at i position, the error rate can be calculated using the Chain Rule.

$$\frac{\partial E_p}{\partial X_{L,i}} = \sum_{m=1}^{N(l+1)} \left(\partial E_p / \partial X_{m,p}^{l+1}\right) \left(\partial X_{m,p}^{l+1} / \partial X_{m,p}^{l+1}\right) \qquad (A.3)$$

Where $0 \leq l \leq L-1$. Internal node error signal can be expressed as a linear combi-nation of the error rate in the layer node l (l + 1). Eqa.A.3 is used to calculate the error signal at i[th] layer node to l (l <L), while the use of Eq.A.3 to reach the final layer. Further, whenαis a parameter used in some node, and then the equation will be obtained as follows:

$$\frac{\partial E_p}{\partial \alpha} = \sum_{X^* \in S} \left(\frac{\partial E_p}{\partial X^*}\right)\left(\frac{\partial X^*}{\partial \alpha}\right) \qquad (A.4)$$

Where S is the set of nodes containing the parameter α, so that the whole issue of measurement

error of α will produce Eq. (A.5).

$$\frac{\partial E}{\partial \alpha} = \sum_{p=1}^{P} \frac{\partial E_p}{\partial \alpha} \qquad (A.5)$$

with steepest gradient descent method, the equation for repairing parameter α is obtained:

$$\Delta\alpha = -\eta \frac{\delta E}{\delta \alpha} \qquad (A.6)$$

Where η is the learning rate process and stated as follows:

$$\eta = \frac{k}{\sum \alpha \left(\frac{\partial E}{\partial \alpha}\right)^2} \qquad (A.7)$$

Where k is the step size, which can be changed in order to accelerate the convergence rate in adaptive networks.

A.2 least square error learning algorithm

During the premises parameter in a steady state, then all output derived from the consequent parameters can be specified in a combination linear equation (Jang 1993; Jang and Sun 1995) like equation 3.29. When N training data are given to Eq.3.29, then the equation will be obtained as follows:

$$(\overline{w1}x)1p1 + (\overline{w1}y)1q1 + \overline{(w1)}1r1 + (\overline{w2}y)1p2 + (\overline{w2y})1q2 + (\overline{w2})r2 = f1$$

.

.

.

$$(\overline{w1}x)np1 + (\overline{w1}y)nq1 + \overline{(w1)}nr1 + (\overline{w2}y)np2 + (\overline{w2}y)nq2 + (\overline{w2})nr2 = fn \qquad (A.8)$$

To simplify, Eq.A.8can be expressed in matrix form as shown in Eq.A.9:

$$A\theta = y \qquad (A.9)$$

Where θ is the vector M×1.M refers to the number of elements that are consequent parameter set. While A is the vector P×M, where P is the number of N data training provided to the adaptive network and y is the output vector P×1 whose elements are N number of output data of an adaptive network. Normally, the amount of training data is larger than the number of consequent

parameters, so the best solution for θ is minimizing the squared error$\|A\theta = y^2\|$. By the least squares estimator (LSE), the equation for θ is defined as

$$\theta^* = (A^T A)^{-1} A^T y \qquad (A.10)$$

Where A^T is the inverse of A and if not singular, $(A^T A)^{-1}$ is the pseudo-inverse of A. By using a recursive LSE method, then the Eq. A.10 becomes

$$\left.\begin{aligned} \theta_{i+1} &= \theta_i + P_{i+1}\alpha_{i+1}(y_{i+1}^T - \alpha_{i+1}^T\theta_i) \\ P_{i+1} &= P_i - (P_i + \alpha_{i+1}\,\alpha_{i+1}^T\,P_i)/(1 + \alpha_{i+1}^T\,P_i\,\alpha_{i+1}), i = 0,1 \ldots, P-1 \end{aligned}\right\} \qquad (A.11)$$

Where α_i^T is a row vector of the matrix A in Eq. A.9, y_i is i^{th} element of y. P_i sometimes called a covariance matrix and is defined by the following equation:

$$P_i == (A^T A)^{-1} \qquad (A.12)$$

B. Sub-blocks inside figure 4.1

To build the simulation model, the equations derived previously (see chapter 2) should be rearranged as follows. From the D-Q axis model of eq.2.15 and 2.16 the following equations can be derived.

$$\begin{cases} \varphi ds = (vds - Rsids + \omega e\, \varphi qs)/s \\ \varphi qs = (vqs - Rs\, iqs - \omega e\, \varphi ds)/s \\ \varphi dr' = (vdr' - Rr'idr + (\omega e - \omega r)\varphi qr')/s \\ \varphi qr' = vqr' - Rr'iqr - (\omega e - \omega r)\varphi dr')/s \end{cases} \quad (B.1)$$

Where the derivative operator p in Equation 2.13 is replaced by the Laplace operator S, and 1/S represents an integrator. The stator and rotor currents can be expressed in terms of stator and rotor flux linkages.

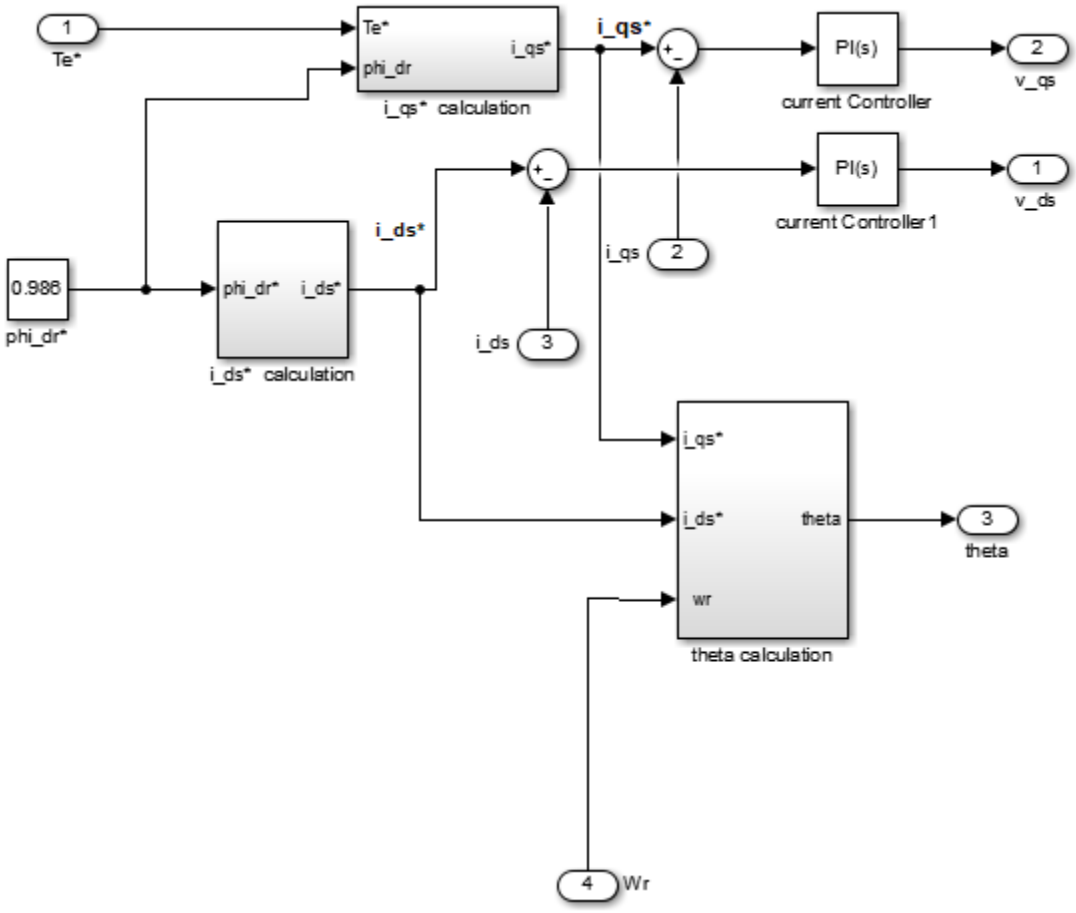
$$\begin{bmatrix} ids \\ iqs \\ idr' \\ iqr' \end{bmatrix} = \frac{1}{D1}\begin{bmatrix} Lr' & 0 & -Lm & 0 \\ 0 & Lr' & 0 & -Lm \\ -Lm & 0 & Ls & 0 \\ 0 & -Lm & 0 & Ls \end{bmatrix}\begin{bmatrix} \varphi ds \\ \varphi qs \\ \varphi dr' \\ \varphi qr' \end{bmatrix} \quad (B.2)$$

Where $D1 = Ls * Lr - Lm^2$

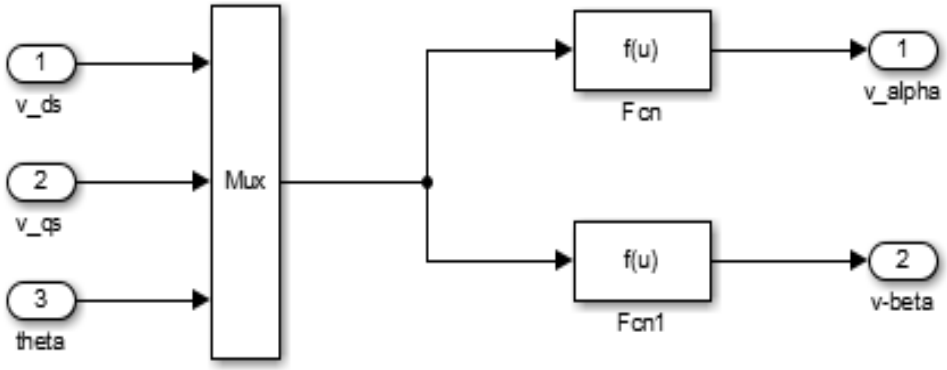
The motion and the torque equations for the simulation model are given by

$$\begin{cases} \omega r = \frac{P}{Js}(Te - Tl) \\ Te = \frac{3P}{4}(\varphi qr' idr' - \varphi dr' iqr') \end{cases} \quad (B.3)$$

Using the above equations the induction motor Simulink model is developed as shown in figure B-2b below. The following figures (B1-B3) are the sub blocks of IFOC, axis transformation, inverter and three-phase induction motor.

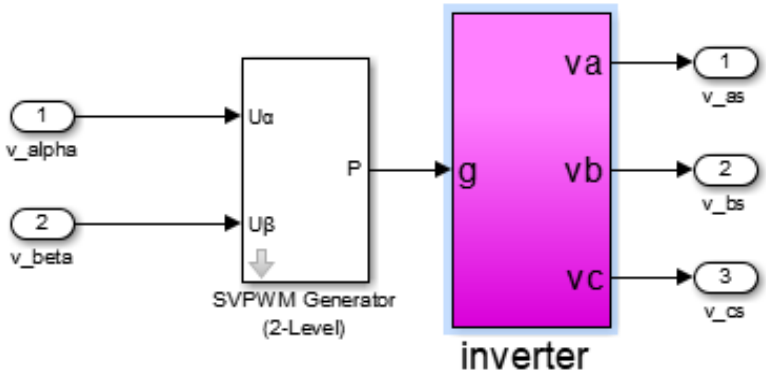


(a)

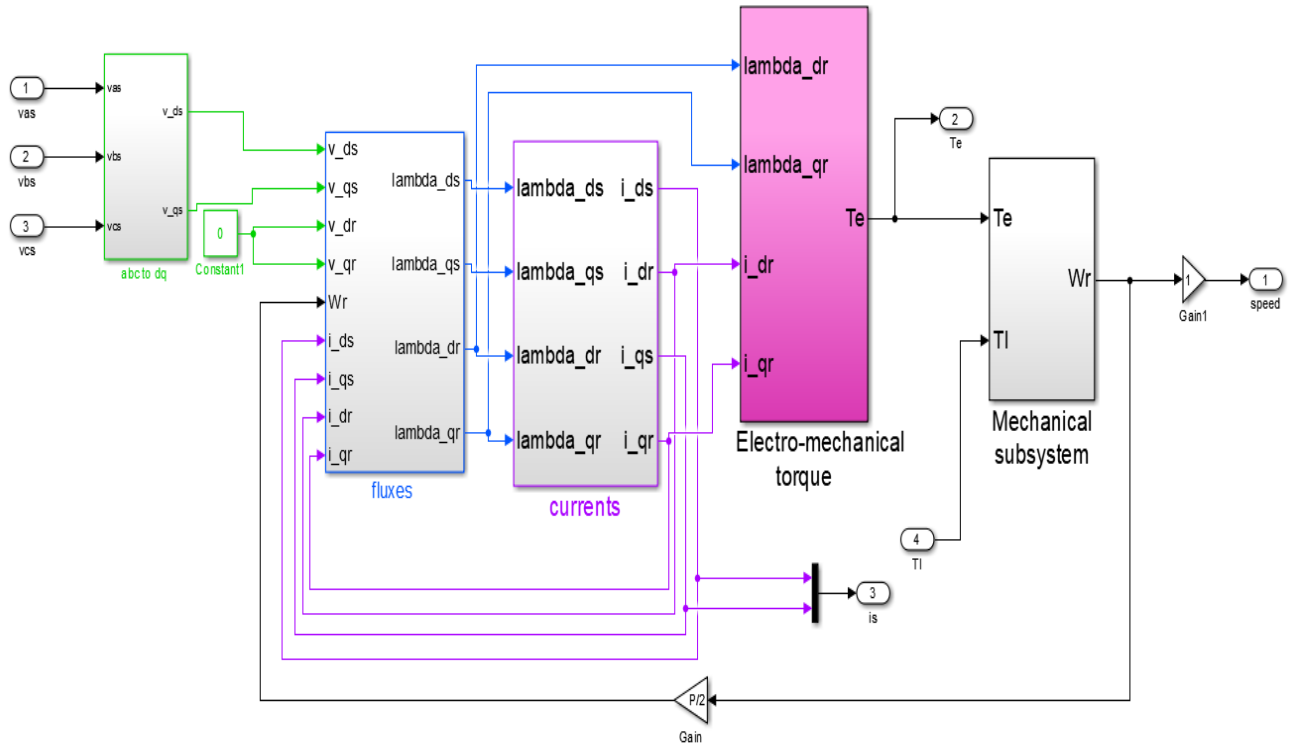


(b)

Figure B-1: Sub-blocks of a) IFOC and b) axis transformations



(a)



(b)

Figure B-2: Sub-blocks of a) SVPWM and b) IM.

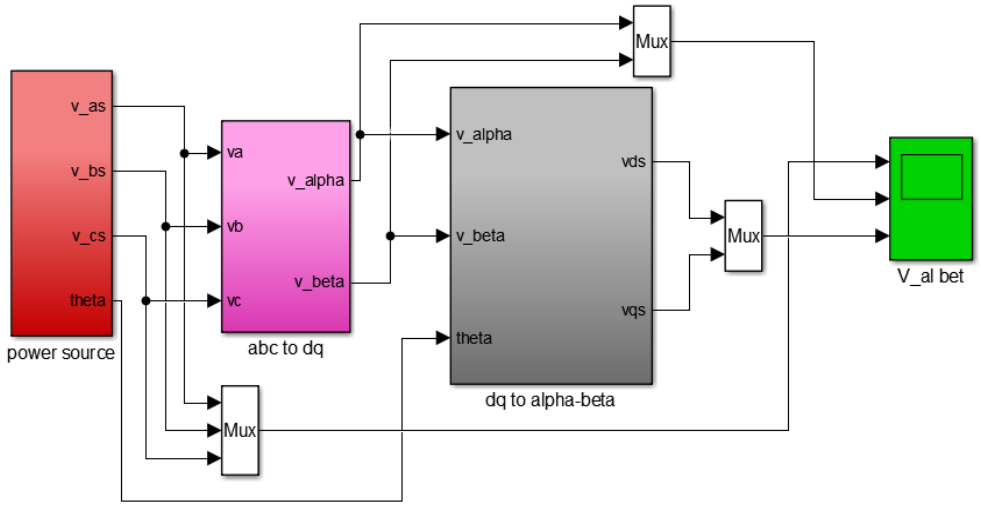
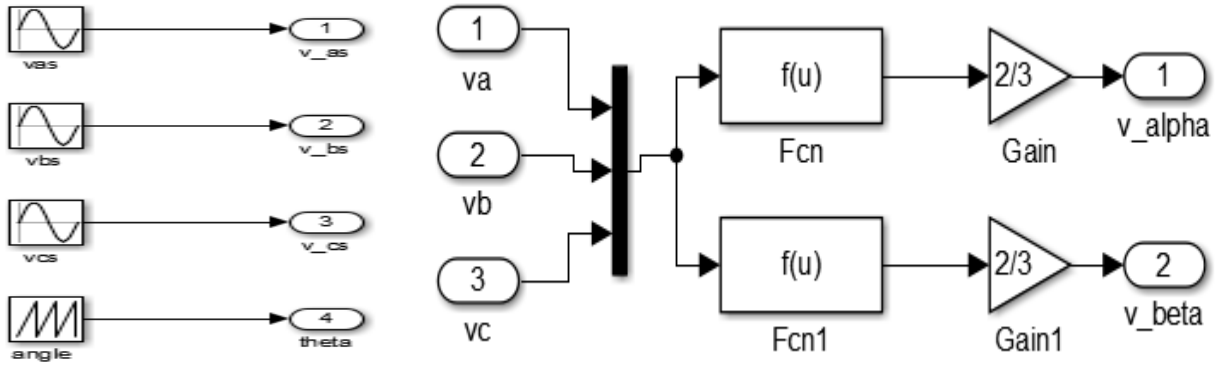
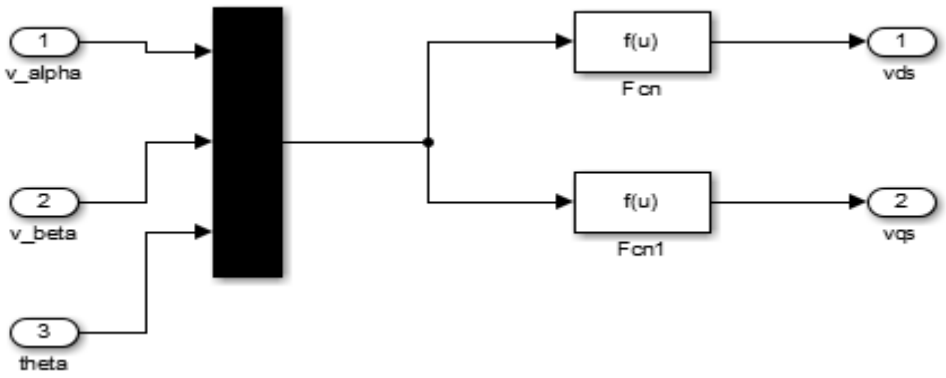


Figure B-3: Simulink diagram of a-b-c to d-q and d-q to alpha-beta axis transformation.



(a) (b)



(c)

Figure B-4: a) Power source b) a-b-c to alpha-beta and c) alpha-beta to D-Q transformation Sub blocks of figure B-3

C. GA optimization MATLAB code

```
function s=Call_PI(x)
assignin('base','x1',x(1));
assignin('base','x2',x(2));
[tout,xout,yout]=sim('toData',2);
z=yout;
[m,n] = size(z);
V=0;
Wref=50;  %reference
for i=1:size(z)
   V=V+(Wref-z(i))^2;
end
s=V; %sum of error square
end
% To run optimization
clc
clear
Rs=0.288; % stator resistance (ohm)
Rr=0.158; % rotor resistance (ohm)
Ls=0.0425; % stator inductance (H)
Lm=0.0412; %mutual inductance (H)
Lr=0.0418; %rotor inductance (H)
Tr=Lr/Rr; % rotor time constant
D1=Ls*Lr-Lm^2;
J=0.4; % moment of inertia (Kg.m2)
P=2; % number of pole
F=0.001; % viscos friction coefficient (Nms-1)
Llr=Lr-Lm; % rotor leakage inductance
Lls=Ls-Lm; % stator leakage inductance
Ts=0.0001; %sampling tome
m1=1.0e+03 *0.0508; %normalization gain for input1
```

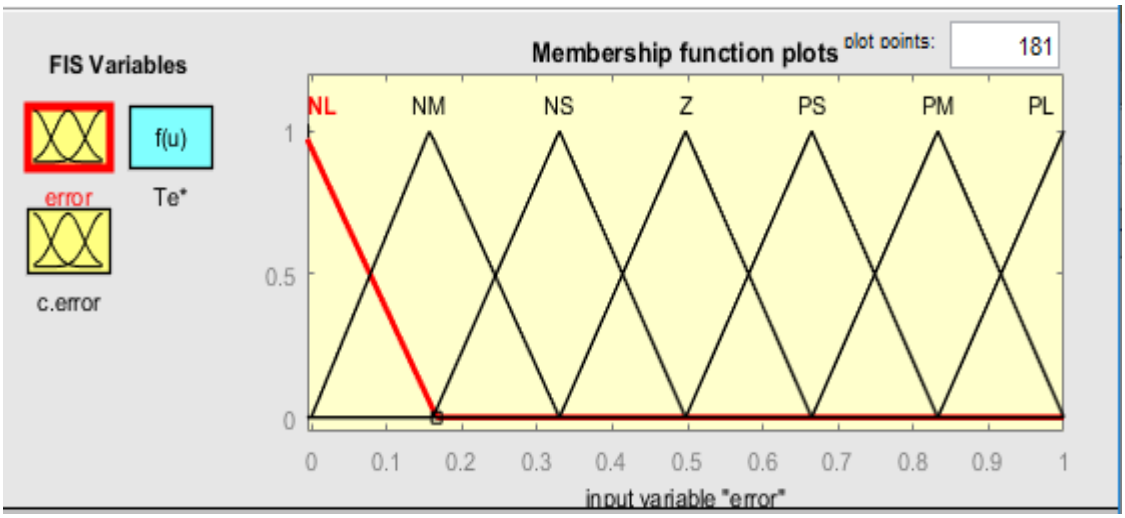
```
m2=1.0e+03 *0.1335; %normalization gain for input2
m3=1.0e+03 *1.5466; %normalization gain for output
options = gaoptimset(@ga);
options=gaoptimset(options,'PlotFcns',{@gaplotbestf},'Display','
iter');
options = gaoptimset(options,'PopulationSize',60);
options = gaoptimset(options, 'EliteCount',2);
options = gaoptimset(options, 'CrossoverFraction',0.8);
options = gaoptimset(options, 'Generations',200);
options =gaoptimset(options,'MutationFcn',@mutationadaptfeasible;
options= gaoptimset(options, 'SelectionFcn',@selectionstochunif);
lb = [0;0];
ub = [35;35];
[x,fval]= ga(@Call_PI,2,[],[],[],[],lb, ub,[],options);

% data sellection, normalization and division for training.
TotalData=(NL;L)% no load and with load
Data=TotalData(randsample(size(TotalData,1),100000),:);   %random
data selection
e=Data(:,1);% input 1 error
ce=Data(:,2);% input2 change in error
Te=Data(:,3);% output command torque(Te*)
m1=max(e);% m1 is normalization gain for i/p 1
m2=max(ce);%m2 is normalization gain for i/p 2
m3=max(Te);%m3 isnormalization gain for o/p
input1=e/m1;%normalize input 1
input2=ce/m2;%normalize input 2
out1=Te/m3;%normalize input 3
anfisData=[input1 input2 out1];% normalized data between -1 and 1
trndata=anfisData(1:70000,:); % training data set
chkdata=anfisData(70001:end,:); % checking data set
```

D. ANFIS rules and membership functions

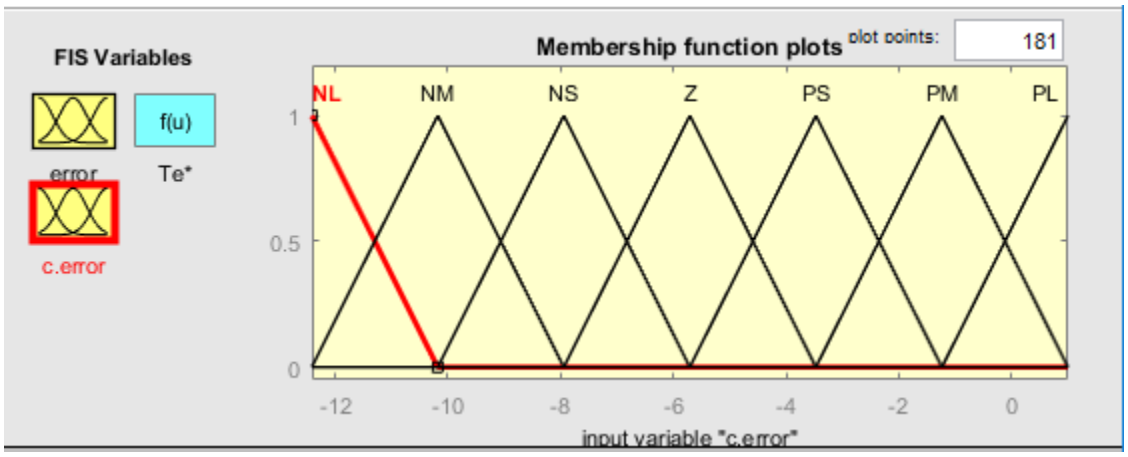
Membership functions for input and output variables have been chosen with triangular shapes and linear respectively as shown in Fig. D-1. Universe of discourse of input and output variables are divided in to seven fuzzy sets: NL (Negative Large), NM (Negative Medium), NS (Negative Small), ZE (Zero), PL (Positive Large), PM (Positive Medium), PS (Positive Small). Rule base for FLC speed controller is given in Table D-1 below.

Table D-1: fuzzy rules

E → **CE ↓**	**NL**	**NM**	**NS**	**Z**	**PS**	**PM**	**PL**
NL	NL	NL	NL	NL	NM	NS	Z
NM	NL	NL	NL	NM	NS	Z	PS
NS	NL	NL	NM	NS	Z	PS	PM
Z	NL	NM	NS	Z	PS	PM	PL
PS	NM	NS	Z	PS	PM	PL	PL
PM	NS	Z	PS	PM	PL	PL	PL
PL	Z	PS	PM	PL	PL	PL	PL

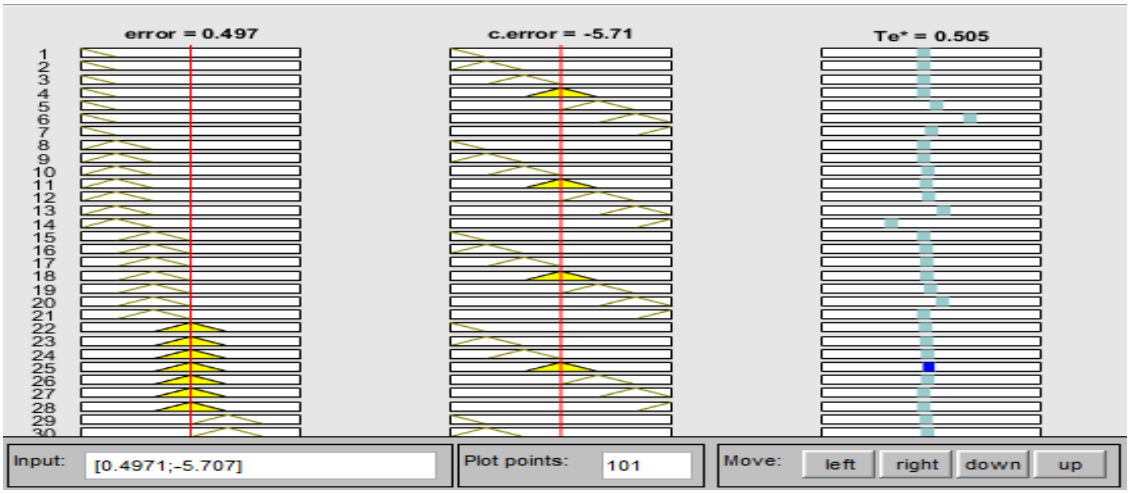


(a)

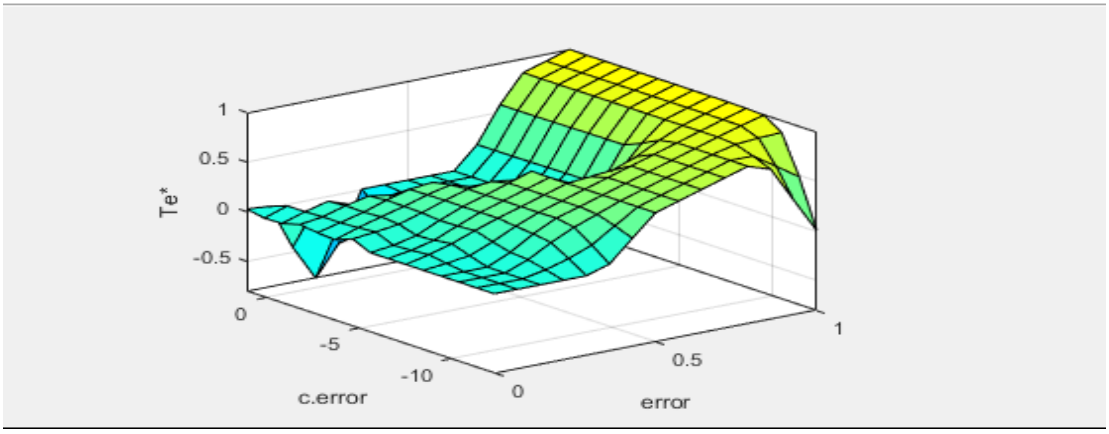


(b)

Figure D-1: a) Error and b) change in error MFs



(a)



(b)

Figure D-2: Views of a) Rules and b) surface

The Surface Viewer is used to display the dependency of the output on any one or two of the inputs. It generates and plots an output surface map for the system. The output is presented by a 3D surface model.