Addis Ababa University

Addis Ababa Institute of Technology

School of Electrical and Computer Engineering

Design and Simulation of Speed Sensorless, FOC of Induction Motor Drive Using ANFIS Controller and ANN Estimator

---

By: Biniam Abera

Thesis Submitted to the School of Graduate Studies of Addis Ababa University in Partial Fulfillment of the Requirements for the Degree of Master of Science in Electrical and Computer Engineering (Industrial Control Stream)

Supervised by: - Dr. Mengsha Mamo

January, 2020

Addis Ababa, Ethiopia

Addis Ababa University

Addis Ababa Institute of Technology

School of Electrical and Computer Engineering

This is to certify that the thesis prepared by Biniam Abera, entitled 'Speed Sensorless, Field Oriented Control of Induction Motor Drive Using ANFIS Controller and ANN Estimator', submitted in partial fulfillment of the requirements for the degree of Master of Sciences in Industrial Control Engineering complies with the regulations of university and meets the accepted standards with respect to originality and quality.

Approved by board of Examiners

Dr. Mengesha Mamo		13/01/2020
Thesis Advisor	Signature	Date
Dr. Dereje Shiferaw		03/02/2020
External Examiner	Signature	Date
Ms. Yalemzerf Getnet		03/02/2020
External Examiner	Signature	Date

School Chair Person

Dr. Yalemzewed Negash

Declaration

I certify that research work titled 'Design and Simulation of Speed Sensorless, Field Oriented Control of Induction Motor Drive Using ANFIS Controller and ANN Estimator' is my own work. The work has not been presented elsewhere for assessment. Where material has been used from other sources it has been properly acknowledged / referred.

Signature of Student_____beny_____

Name of Student_____Biniam Abera_____

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

Dr. Mengesha Mamo _____

Advisor's Name Signature

Date 13/01/2020 G.C

Acknowledgement

First of all, I would like to offer my sincere gratitude to Almighty God, for his Grace and Blessings.

A special thanks to my supervisor, Dr. Mengesha Mamo for the shared wisdom and guidance offered through this endeavor, with whom I found inspiration and deep admiration. I want to acknowledge him as great teacher, great scholar and a man to whom I can never repay the debt. What a privilege to study under his guidance.

It is my great pleasure to thank Gash Elias, Mr. Patrick, Mr. Yibeltal and Mr. Getachew whose friendship enlivened both the research and my stay at AAiT. Particularly, the wonderful experience with Mr. Patrick in part taking on the discussions on different aspects of this research work is simply unforgettable. Further, my sincere thanks go to the entire Graduate fellowship members, their spiritual help and teachings was indispensable for my life.

Abstract

This thesis presents design and simulation of Artificial Neural Network solution for speed estimation including an ANFIS for control of IM drives. In the past years the research efforts in the field of IM control concentrated on the identification and observation of this highly nonlinear dynamic plant. Existing vector control methods require speed sensor for control and field orientation purpose, but their installation makes the drive system bulky, unreliable and expensive and installing them might not be feasible in some applications, such as motor drives in hostile environment or high-speed drives. In such cases speed is obtained from easily measurable stator quantities. Many speed sensorless techniques have been proposed to cope up with speed sensing problem. Developed speed estimation algorithms are more or less parameter dependent and/ or computationally time consuming. In this thesis, the proposed estimation method is based on ANN to obtain the speed signal. The conventional PI controller is replaced by an ANFIS which tunes the fuzzy inference system with hybrid learning algorithm. The ANN is used as estimator, trained by Levenberg- Marquardit algorithm. The data for training are obtained from conventional FOC simulations when the motor drive is working in closed loop at various values of speeds and loads for speed observation. The complete drive system is modeled using MATLAB®2019a. Finally, the drive results have been analyzed for both steady state and dynamic conditions such as of speed tracking capability, torque response quickness, low speed behavior, step response of drive with speed reversal and sensitivity to motor parameter uncertainty. The error of simulation result between actual and estimated speed have been less than 0.3% for transient response, 0.2% for speed tracking and 0.44% during low speed operation. It was observed from simulation results that by using PI and ANFIS controller, for the reference speed of 151rad/sec, the rise and settling time are improved by 0.0938 and 0.1289 seconds at full respectively also robust response is achieved with the latter controller.

Key Words: Field Oriented Control, SVPWM, Speed Sensorless Operation, ANN, ANFIS

Nomenclature

List of Abbreviation's

AC	Alternate Current
AI	Artificial Intelligence
ANFIS	Adaptive Neuro- Fuzzy Inference System
ANN	Artificial Neural Network
ASICS	Application Specific Integrated circuits
CRBW	Current Regulator Bandwidth
DC	Direct Current
DSP	Digital Signal Processors
DTC	Direct Torque control
DFO	Direct Field Orientation
FLC	Fuzzy Logic Controller
FOC	Field oriented control
GA	Genetic Algorithms
GUI	Graphical user Interface
IM	Induction Motor
IRFO	Indirect Rotor Field Orientation
MATLAB®	Matrix Laboratory
MLP	Multi-layer Perceptron
MRAS	Model Reference Adaptive System
NNs	Neural Networks
PI/D	Proportional Integral/ Derivative
RMSE	Root Mean Squared Error
SVPWM	Space Vector Pulse Width Modulation
SCIM	Squirrel Cage Induction Motor
VFD	Variable frequency Drive
VSD	Variable Speed Drive

List of Symbols

$\psi_{sd}^{e}, \psi_{sq}^{e}$	Direct and quadrature components of stator flux in synchronously rotating reference frame
$\psi_{rd}^{e}, \psi_{rq}^{e}$	Direct and quadrature components of rotor flux in synchronously rotating reference frame
i_{sd}^{e}, i_{sq}^{e}	Direct and quadrature components of stator current in synchronously rotating reference frame
v_{sd}^{e} , v_{sq}^{e}	Direct and quadrature components of stator voltage in synchronously rotating reference frame
R_s, R_r	Stator and Rotor resistance in Ohms
L_s , L_r	Stator and Rotor Inductance in Henry
L_m , σL_s	Mutual Inductance and Stator transient Inductance
L_{ls} , L_{lr}	Stator and Rotor leakage inductances
ω_{ref} , ω_{est}	Reference and Estimated rotor speed
J	Moment of Inertia in kg.m^2
$T_{em}\ and\ T_L$	Developed and Load torque in N.m
$e(k) and\ e(k-1)$	Current and previous samples of speed error
$\Delta e(k)$	Change in error signal
η	Learning rate in NN
p	Number of pole pairs

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1 INTRODUCTION

1.1 Back ground of the Study

Controlled induction motor drives without mechanical speed sensors at the motor shaft have attractions of low cost and high reliability. To replace the sensor, the information on the rotor speed is extracted from measured stator voltages and currents at the motor terminals. Vector controlled drives require estimating the magnitude and spatial orientation of the fundamental magnetic flux waves in the stator or in the rotor. Open loop estimators and closed loop observers are used for this purpose. They differ with respect to accuracy, robustness, and sensitivity against model parameter variations. Dynamic performance and steady-state speed accuracy in the low-speed range can be achieved by exploiting parasitic effects of the machine [1].

AC drives based on full digital control have reached the status of a mature technology. Ongoing research has concentrated on the elimination of the speed sensor at the machine shaft without deteriorating the dynamic performance of the drive control system [1]. Speed estimation is an issue of particular interest with induction motor drives where the mechanical speed of the rotor is generally different from the speed of the revolving magnetic field. The advantages of speed-sensorless induction motor drives are reduced hardware complexity and lower cost, reduced size of the drive machine, elimination of the sensor cable, better noise immunity, increased reliability, and less maintenance requirements. Operation in hostile environments mostly requires a motor without speed sensor.

Conventional vector control methods [1] require motor speed as a feedback signal. To obtain the speed information, transducers such as shaft-mounted tacho- generators, resolvers, or digital shaft position encoders are used, which degrade the system's reliability, especially in hostile environment. The common sensorless Induction motor control strategies are derived from the sensor-based Field Oriented control methods that have been extended to include speed estimation algorithms, without using mechanical speed sensors.

However, the estimated speed accuracy is generally sensitive to model parameter mismatch if the machine is loaded, especially in the field-weakening region and in the low speed range. The parameters contributing to this variation are, Rotor resistance variation with temperature, Stator resistance variation with temperature, Stator inductance variation due to saturation of the stator teeth [2]. Extensive research activities are performed around the academic and industrial world to design observers and controllers with good dynamic performance and robust for parameter variation.

Recently, Artificial Intelligence (AI) techniques have been widely adopted in electrical engineering field and none less in the electrical drive systems. A number of AI techniques are known in literature such as Expert systems, Fuzzy logic, and Artificial Neural Network (ANN). Amongst these, ANN is comparatively the most powerful known AI technique [2]. ANN as is indicative emulates the functioning of human nervous system and hence incorporates human thinking into computer program.

ANN basically can be viewed as a set of computing systems in parallel coordination with each other. One of the most important features of neural networks is the ability to learn and adapt to a certain scenario. The ANNs exhibit nonlinear behavior with automatic optimization and learning capabilities. The neural network is fed with the input and target data set and trained using different neural network structures that are available in literature [3] so that the system simply learns. Thus, ANNs find application in modeling, estimation, and control of systems where little knowledge of the system dynamics is known.

A variety of different solutions for sensorless ac drives have been proposed in the past few years. Other observers like Kalman filter, Extended Kalman filter, MRAS and Luenberger observers require the induction motor model for development of sensorless control strategy. These models involve the calculation of motor resistance and inductances online continuously so as to develop a perfect observer. But ANN does not depend on these parameters and hence it avoids the effect of change of these parameters on the design of speed observer for the induction motor model [4].

Vector control enables induction machines to be utilized in applications where DC machines used to be applied in the past. The advantages of induction machines over DC machines are lower cost of purchase and maintenance, better reliability in hostile environment, higher efficiency, simplicity, ruggedness, absence of brushes, etc.[4].

However, vector control method requires more sophisticated hardware and software than the control system of DC machines.

Vector control principles utilize reference frame transformation to refer three phase quantities to a rotating reference frame with fictitious direct and quadrature axes. The new set of differential equations in the new reference frame does not contain time-varying inductances and the decoupling of torque and flux control can be achieved by appropriate selection of the angular speed of rotation of the reference [5].Torque and flux of the machine can be then controlled independently with two different current commands, this being analogous to torque and flux control in DC machines that are used for high dynamic performance.

The controllability of speed/torque in an induction motor with good transient and steady state responses forms the main criteria in the designing of a controller. Though, PI controller is able to achieve these but with certain drawbacks. The gains cannot be increased beyond certain limit so as to have an improved response. Moreover, the non-linearity of the system making it more complex for analysis. Also, when ignoring the system non- linearity, it deteriorates the controller performance. With the advent of artificial intelligent techniques, these drawbacks can be mitigated. One such technique is the use of Fuzzy Logic in the design of controller either independently or in hybrid with PI controller or with other Intelligent controllers [6].

Fuzzy Logic Controller yields superior and faster control, but main design problem lies in the determination of consistent and complete rule set, optimal number of linguistic variables and shape of the membership functions. 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. The draw-backs of Fuzzy Logic Control and Artificial Neural Network can be overcome by the use of Adaptive Neuro-Fuzzy Inference System. The main concept of a neuro-fuzzy network is derived from the human learning process, where an initial knowledge of a function is first setup by fuzzy rules and then the degree of function approximation is iteratively improved by the learning capabilities of the neural network. Hence, in this thesis design of ANFIS speed controller which combines the learning power of neural network with knowledge representation of fuzzy logic and ANN speed observer for vector-controlled drive was the main target.

1.2 Problem Statement

The speed control is quite common in the most induction motor applications. Traditionally the speed information of the induction motor is measured or calculated using a rotor position / speed sensor. One of the advantages of installing speed sensors is that the measurement is almost independent of the machine control itself. The precision can be very high when the high-resolution sensor is used. So, it is regarded as an accurate method for the speed measurement. However, the speed sensors are difficult or not allowed to install due to the space limitation and several environment conditions in some applications. The installation increases the possibility of failures due to the extension of shaft. Another most important issue is the cost of the speed sensor, which takes a large portion of total cost for the auxiliary equipment of induction motor control. The speed estimation basically is the algorithm that can infer the measurement required from other more easily available measurements like voltages and currents.

Artificial neural networks are well suited for application in IM control and estimation, because of their known advantages as: ability to approximate arbitrary nonlinear functions to any desired degree of accuracy, learning and generalization, fast parallel computation, robustness to input harmonic ripples and fault tolerance.

Additionally, Artificial neural network techniques offer the advantage of parameter independency and non-requirement of the exact system mathematical model. As such it reduces the computational problems associated with other observers as is the case with Kalman and Luenbrerger observers to estimate the rotor speed. Designing ANN to observe rotor speed without deteriorating the dynamic performance of the drive from variables that are relatively easily measured like stator voltage and current is the main motivation behind this thesis.

Induction motors are characterized by highly non-linear, complex and time-varying dynamics and inaccessibility of some of the states and outputs for measurements. Hence it can be considered as a challenging engineering problem in the industrial sector. Various advanced control techniques have been devised by various researchers across the world.

Existing conventional controllers in Vector control methods have two primary disadvantages, sensitivity to motor parametric variations and flux errors at low speeds. These are problems conventional PI controllers do not deal with well, leading to

deterioration in dynamic performance. This is where artificial intelligent controllers have proven to be excellent alternatives to speed control. Recently, there has been observed an increasing interest in combining artificial intelligent control tools with classical control techniques.

The ANFIS speed controller is designed to obtain improvements in system performance, cost-effectiveness, efficiency, dynamism, reliability and handling nonlinearity during different operating conditions.

1.3 Objectives

1.3.1 General Objectives

- The general objective of this thesis is to design and simulate Artificial Neural Network based speed observer and ANFIS controller for speed sensorless, vector-controlled induction motor drive for high performance applications.

1.3.2 Specific Objectives

- To Design dynamic model of three phase induction motor, Field oriented control techniques and modulation techniques for Inverter in adjustable speed drive application.
- To Design an Artificial Neural Network as function approximator for rotor speed observation of three phase Induction Motor.
- To Design hybrid Neuro- Fuzzy controller to precisely control the speed of Induction motor Drive for high performance drive applications.
- To Generate Training, Test and Validation data from conventional PI controller using MATLAB® simulation and comparing performance of sensored and sensorless schemes depending on bandwidth, rise-time, settling time maximum overshoot and robustness to parameter variation.

1.4 Methodology

The vector control or field-oriented control (FOC) of AC machines makes it possible to control AC motor in a manner similar to the control of a separately excited DC motor. In AC machines, the torque is developed by the interaction of current and flux. In induction motor the power is given to the stator only, the current responsible for flux production, and

the current responsible for torque production are not easily separate. The main criterion of vector control is to separate the components of stator current responsible for flux production, and the also the torque. The vector control in an AC machines is obtained by controlling the magnitude, frequency, and stator current phase, by inverter control scheme. As, the control of the motor is obtained by controlling both magnitude and phase angle of the current, this control method is given a name i.e. vector control. In order to achieve independent control of flux and torque in induction machines, the stator (or rotor) flux linkages phasor is maintained constant in its magnitude and its phase is stationary with respect to current phasor.

To perform vector control of induction motor drive the following steps should be performed:

- Dynamic modelling of induction motor in dq frame.
- Measuring the motor quantities (phase voltages and currents).
- Transforming them to the 2-phase system (α, β) using a Clarke transformation.
- Calculating the rotor flux space vector magnitude and position angle
- Transforming stator currents to the d-q coordinate system using a Park transformation
- The stator current torque- (i_{sq}) and flux- (i_{sd}) producing components have to be separately controlled.
- The output stator voltage space vector is calculated using the decoupling block.
- An inverse Park transformation transforms the stator voltage space vector back from the d-q coordinate system to the 2-phase system fixed with the stator.
- Using the space vector modulation, the output 3-phase voltage is generated.

Knowledge of the rotor flux space vector magnitude and position is key information for AC induction motor vector control. With the rotor magnetic flux space vector, the rotational coordinate system (dq) can be established. There are several methods for obtaining the rotor magnetic flux space vector. The flux model utilizes monitored rotor speed and stator voltages and currents to obtain the magnitude and position of rotor flux. Indirect field-oriented control mechanism is chosen to obtain the obtain rotor flux position in this thesis.

A controller is a device which controls each and every operation in the system making decisions. From the control system point of view, it is bringing stability to the system when there is a disturbance, thus safeguarding the equipment from further damages. It may be hardware-based controller or a software-based controller or a combination of both.

Currently, Fuzzy Logic control (FLC) has proven effective for complex, nonlinear and imprecisely defined processes for which standard model-based control techniques are impractical or impossible. Fuzzy Logic, deals with problems that have vagueness, uncertainty and use membership functions with values varying between 0 and 1. If the data available is not reliable, or if the system is too complex to derive the required decision rules, then the development of a fuzzy logic controller become quite difficult. In such case, the expert knowledge can be used for framing the proper rules which can be further used to tune the controller for obtaining the better result. Artificial Neural Network (ANN) has the powerful capability for learning, adaptation, robustness and rapidity. Therefore, the advantages of ANN have been used for framing the proper rules of the fuzzy logic controller by adaptation and learning algorithm which we call ANFIS controller.

Design methodology used for ANFIS based speed controller is expert control, which mimic a human expert. The control objective is to design a controller function such that the plant state can follow a desired trajectory as closely as possible. The normalized speed error and the rate of change of actual speed error are the inputs of the proposed neuro-fuzzy controller.

To start the ANFIS learning; First, a training data set that contains the desired input/output data pairs of target systems to be modeled is to be required.

The design parameters required for any ANFIS controller are:

- Number of data pairs,
- Training data set,
- Checking data sets,
- Fuzzy inference systems for training and,
- Number of epochs to be chosen to start the training, learning results to be verified after mentioning the step size.

Back Propagation Algorithm is used to train the neural network for speed estimation and sigmoidal activation functions in hidden layers and linear functions in the output layer are utilized. For ANFIS controller scheme training hybrid learning algorithm (combination of both Back propagation and Least square estimate) to adapt the premise and consequent parameters were utilized. Generalized bell function is selected as activation functions in nodes for mathematical convenience.

This thesis presents a scheme of Indirect vector control of induction motor with ANN estimator and an ANFIS controller for improving the transient response, when subjected to parameter variations and load torque disturbances. The required data for training the ANN estimator and ANFIS controller is obtained by simulation of the closed loop system with PI and FLC controllers respectively.

The whole system overview is depicted below in diagrammatic form.

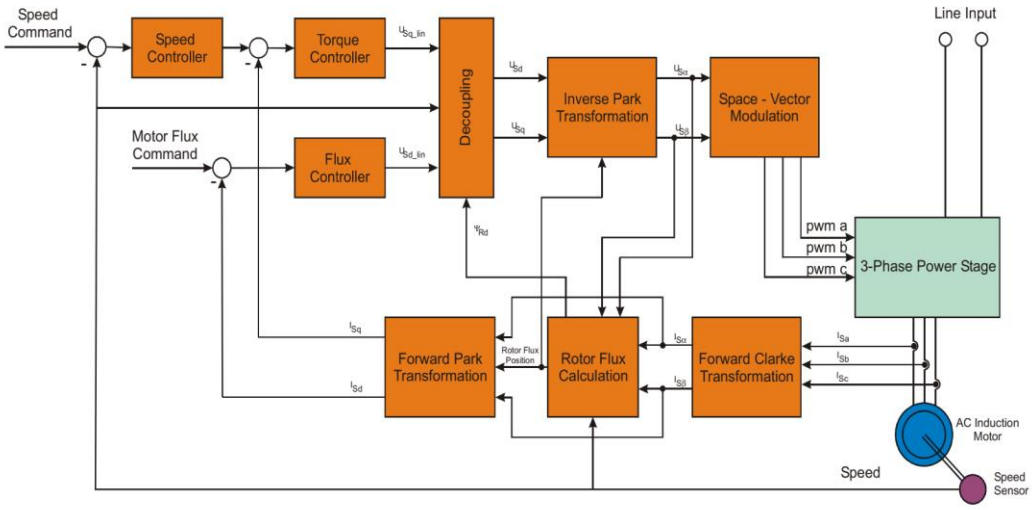


Figure 1.1: Conventional PI controller for Data Generation

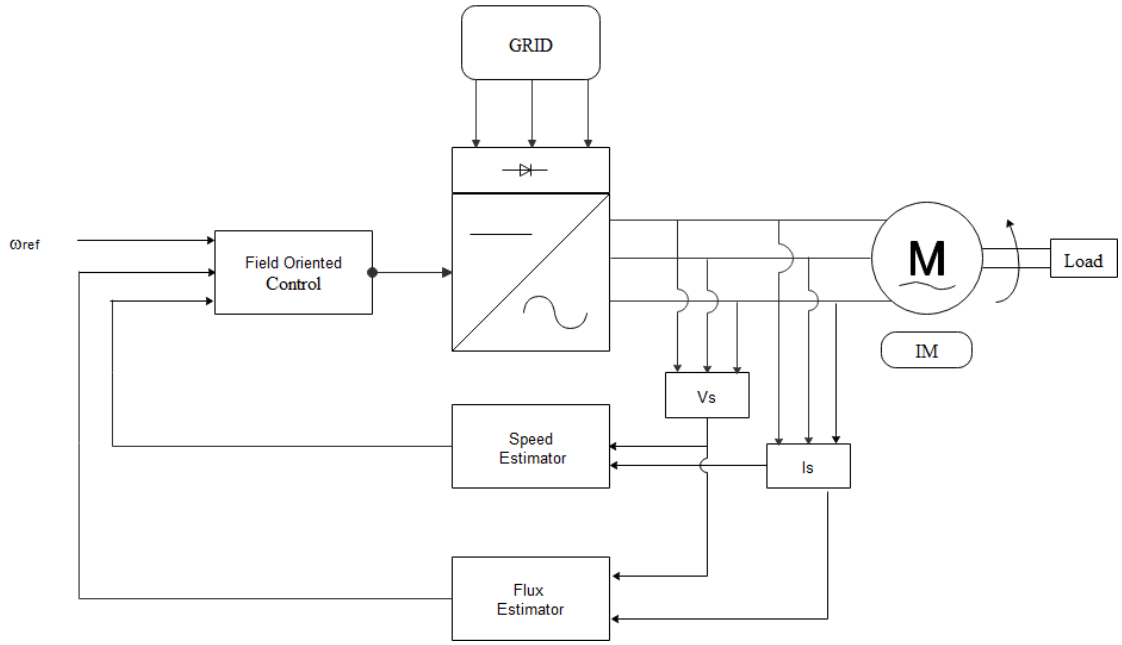


Figure 1.2: Schematic diagram of speed Sensorless Drive

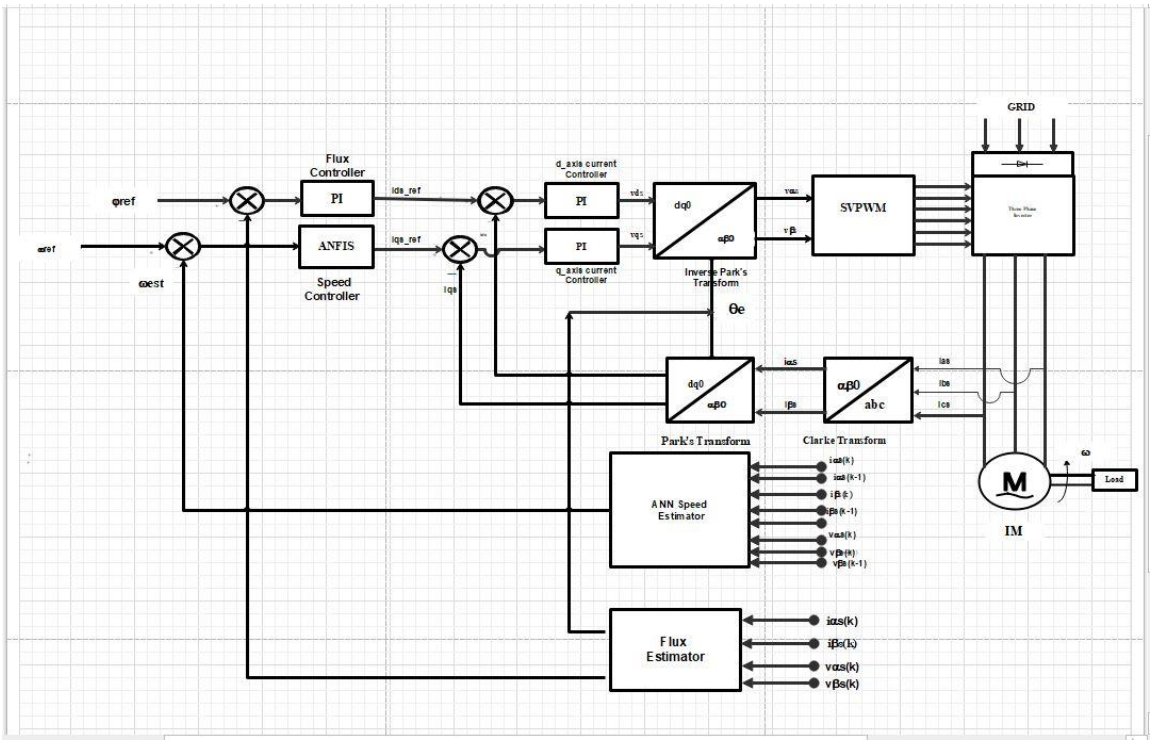


Figure 1.3: Proposed system using ANN estimator and ANFIS Controller

1.5 Thesis Organization

The thesis is organized in to six chapters including this introduction. **Chapter two** describes the Squirrel cage Induction Motor and its Mathematical model based on space vector and Reference frame theory with brief introduction of Field oriented control mechanisms. Conventional PI controllers for outer speed and inner current control loops are designed in this chapter for data generation purpose. The generated data is used for speed sensorless scheme simulation. In **Chapter three,** the whole structure of the proposed MLP artificial neural network for speed estimation of induction motor is shown**. Chapter four** describes the structure of neuro-fuzzy controller and its application in speed control of induction motor drive. The performance of the proposed methods under the whole range of operation conditions of induction motor is then shown in **Chapter five.** Finally, some conclusion is drawn from the findings of this thesis and further research gaps are recommended in **chapter Six.**

CHAPTER 2 FIELD ORIENTED CONTROL OF INDUCTION MACHINE

2.1 Introduction

Induction motors, particularly those of the squirrel-cage type, have been for almost a century industry's principal workhorse. Until the early seventies, they had been mostly operated in the constant-frequency, constant-voltage, uncontrolled mode which even today is still most common in practice. Adjustable-speed drives had been based on D.C motors, mainly in the classic Ward-Leonard arrangement.

The advent of thyristors, the first controlled semiconductor power switches and, consequently, the development of variable-frequency converters based on these switches, made possible wide-range speed control of induction motors. The most popular, so-called scalar control methods consist in simultaneous adjustments of the frequency and magnitude of the sinusoidal voltage or current supplied to the motor. This allows making steady-state operating characteristics of an induction motor similar to those of a D.C motor. Adjustable-speed A.C drive systems, employing various control principles have been replacing the D.C drives in numerous industrial applications, such as pumps, fans, compressors, and conveyor belts. Induction motors have here a clear competitive edge over D.C machines. They are significantly less expensive, more robust, and capable of reliable operation in harsh ambient conditions, even in an explosive environment.

The Field Orientation Principle was first formulated by Hasse, in 1968, and Blaschke, in 1970. At that time, their ideas seemed impractical because of the insufficient means of implementation. However, in the early eighties, technological advances in static power converters and microprocessor-based control systems made the high-performance A.C drive systems fully feasible. Since then, hundreds of papers dealing with various aspects of the Field Orientation Principle have appeared every year in the technical literature, and numerous commercial high-performance A.C drives based on this principle have been developed. The term "vector control" is often used with regard to these systems. Today, it seems certain that almost all D.C industrial drives will be overthrown in the foreseeable future, to be, in major part, superseded by A.C drive systems with vector-controlled induction motors. This transition has already been taking place in industries of developed

countries. Vector controlled A.C drives have been proven capable of even better dynamic performance than D.C drive systems, because of higher allowable speeds and shorter time constants of A.C motors.[7]

The v/fcontrol principle adjusts a constant V/Hz ratio of the stator voltage by feed-forward control. It serves to maintain the magnetic flux in the machine at a desired level. Its simplicity satisfies only moderate dynamic requirements. High dynamic performance is achieved by field orientation, also called vector control mechanism. The stator currents are injected at a well-defined phase angle with respect to the spatial orientation of the rotating magnetic field, thus overcoming the complex dynamic properties of the induction motor. The spatial location of the magnetic field, the field angle, is difficult to measure. There are various types of models and algorithms used for its estimation, discussion of them would be later on this chapter in detail. Control with field orientation may either refer to the rotor field or to the stator field, where each method has its own merits. This chapter mainly focuses on describing the most important and established control strategy for induction machines (i.e. the vector control).

2.1.1 Physical Construction of Induction Machine

Three phase induction motor is constructed from two main parts, namely the rotor and stator:

1. Stator: As its name indicates stator is a stationary part of induction motor. A stator winding is placed in the stator of induction motor and the three-phase supply is given to it.
2. Rotor: The rotor is a rotating part of induction motor. The rotor is connected to the mechanical load through the shaft. Squirrel-cage induction motor is the main focus of this thesis and most widely used in the industry is shown in the figure 2.1 below.

Induction machines can be considered as the workhorse of industry, as they are rugged and cheap. Before the advent of electronic power converters, the use of induction machines was limited to applications operating at nearly constant speed. The advances in the field of power semiconductors, combined with the development of powerful and cheap digital signal processors and adequate control strategies, have resulted in the widespread industrial use of AC variable speed drives (VSDs).

VSDs using three-phase induction motors can be found nowadays in practically all industrial and transportation sectors, for low-voltage and medium-voltage applications, and in a wide power range from fractional horsepower to multi-megawatt rating.[8] Discussing the variety of different methods for speed control of induction motor drive requires an understanding of the dynamic modeling of the motor which is treated in a section below.

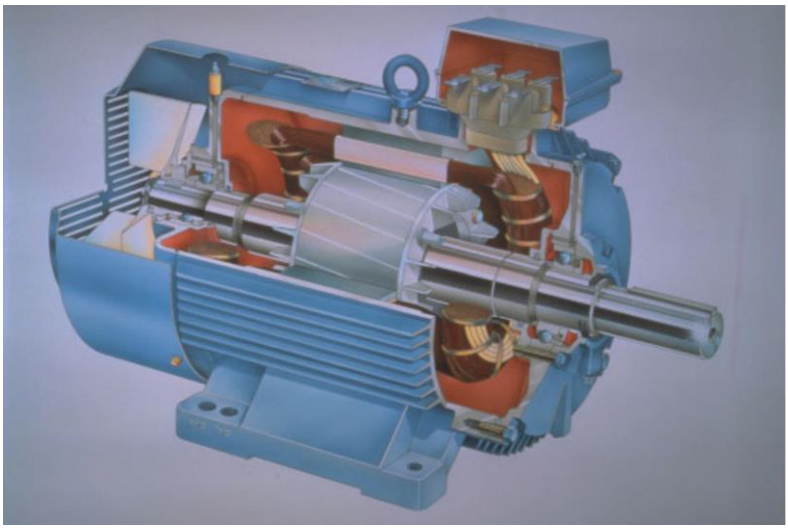


Figure 2.1: Squirrel cage Induction Motor 2012, ABB [7]

2.2 Mathematical Modeling of Induction Machine

2.2.1 On Modelling in General

A theory is often a general statement of principle abstracted from observation and a model is a representation of a theory that can be used for prediction and control. To be useful, a model must realistic and yet simple to understand and easy to manipulate. These are conflicting requirements, for realistic models are seldom simple and simple models are seldom realistic. Often, the scope of a model is defined by what considered relevant. Features or behavior that are permanent must be included in the model and those that are not can be ignored. Modelling here refers to the process of analysis and synthesis to arrive at a suitable mathematical description that encompasses the relevant dynamic characteristics of the component, preferably in terms of parameters that can be easily determined in practice.

In mathematical modelling tries to establish functional relationships between entities which are important. A model supposedly imitates or reproduces certain essential characteristics or conditions of the actual often on a different scale [9].

2.2.2 Dynamic model of the induction motor using three-phase variables

Analysis of induction machines is often realized using per-phase equivalent circuits. While such equivalent circuits are simple and useful for machines operating in steady state, they are inadequate for the analysis of machines operated from electronic power converters, which are frequently subject to changing operating conditions.

Complex vector representation provides an alternative means for the modeling and analysis of three phase induction machines, and three-phase systems in general. Compared to other notations, complex vectors provide a much simpler and insightful representation of the dynamic effects that physically occur in the machine, i.e. the relationships among voltages, currents and fluxes, as well as electromechanical power conversion.

Dynamic models of three-phase induction machines are presented in this section, and they are used in further sections for the development of high-performance control strategies.

The dynamic model is based on the concept of vector quantities of an A.C machine, introduced by Kovacs and Racz in 1959. The motor can be represented either in the form of an equivalent circuit or a set of equations. This procedure allows analysis of the dynamics of the motor which can then be supplied with any kind of voltage, not necessarily a sinusoidal one: [7].

The following assumptions are made when a complete equations system is written to describe the continuous-time linear model of the induction machine [10]:

- Geometrical and electrical machine configuration is symmetrical;
- Space harmonics of the stator and rotor magnetic flux are negligible;
- Infinitely permeable iron;
- Stator and rotor windings are sinusoidally distributed in space and replaced by an equivalent concentrated winding;
- Saliency effects, the slotting effects are neglected;

• Magnetic saturation, anisotropy effect, core loss and skin effect are negligible;

• Windings resistance and reactance do not vary with the temperature;

• Currents and voltages are sinusoidal terms.

• End and fringing effects are neglected

All these assumptions do not alter in a serious way the final result for a wide range of induction machines.

Fig. 2-2 shows a simplified winding model of a three-phase, two-pole induction machine, the stator and rotor phase winding being denoted by as-as', bs-bs', cs-cs', and ar-ar', br-br', cr-cr' respectively. In this figure, θm stands for the angle between the MMF (magneto-motive force) produced by phase a of the stator and the MMF produced by phase a of the rotor, ω_m being the angular speed in electrical units in rad/s. If the number of pole pairs of the machine, p, is different from 1, the relationship between the mechanical units, θ_{rm} and electrical units θ_m, is given by:

$$\theta_{rm} = \frac{\theta_m}{p} \tag{2.1}$$

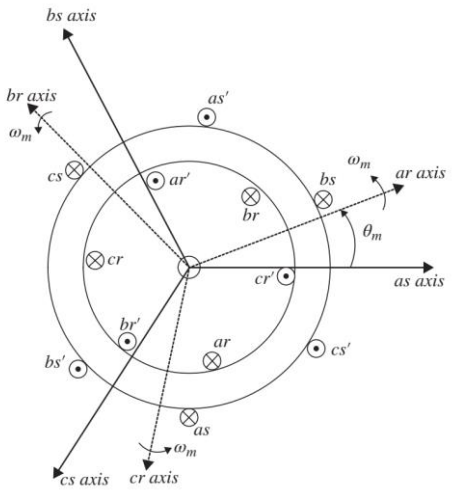


Figure 2.2: Idealized winding model of induction motor [10]

The stator voltage equation in matrix form is (2.2), R_s being the stator winding resistance and $v_{as}, v_{bs}, v_{cs}, i_{as}, i_{bs}, i_{cs}$ and $\psi_{as}, \psi_{bs}, \psi_{cs}$ the instantaneous stator voltages, currents and fluxes respectively.

$$\begin{bmatrix} v_{as} \\ v_{bs} \\ v_{cs} \end{bmatrix} = R_s \begin{bmatrix} i_{as} \\ i_{bs} \\ i_{cs} \end{bmatrix} + \frac{d}{dt} \begin{bmatrix} \psi_{as} \\ \psi_{bs} \\ \psi_{cs} \end{bmatrix} \tag{2.2}$$

The stator fluxes can be written in matrix form as function of the stator and rotor currents as (2.3), where L_h is the self-inductance of the stator, which is twice the mutual inductance between stator windings, and $L_{\sigma s}$ is the leakage inductance of the stator windings

$$\begin{bmatrix} \psi_{as} \\ \psi_{bs} \\ \psi_{cs} \end{bmatrix} = \begin{bmatrix} L_{\sigma s} + L_h & -\frac{1}{2}L_h & -\frac{1}{2}L_h \\ -\frac{1}{2}L_h & L_{\sigma s} + L_h & -\frac{1}{2}L_h \\ -\frac{1}{2}L_h & -\frac{1}{2}L_h & L_{\sigma s} + L_h \end{bmatrix} \begin{bmatrix} i_{as} \\ i_{bs} \\ i_{cs} \end{bmatrix} + L_h \begin{bmatrix} cos\theta_m & cos(\theta_m + \frac{2\pi}{3}) & cos(\theta_m - \frac{2\pi}{3}) \\ cos(\theta_m - \frac{2\pi}{3}) & cos\theta_m & cos(\theta_m + \frac{2\pi}{3}) \\ cos(\theta_m + \frac{2\pi}{3}) & cos(\theta_m - \frac{2\pi}{3}) & cos\theta_m \end{bmatrix} \begin{bmatrix} i_{ar} \\ i_{br} \\ i_{cr} \end{bmatrix} \tag{2.3}$$

The same procedure can be repeated to derive the rotor electromagnetic equation. The rotor voltage equation is given by (2.4), being R_r the rotor winding resistance and $v_{ar}, v_{br}, v_{cr}, i_{ar}, i_{br}, i_{cr}$ and $\psi_{ar}, \psi_{br}, \psi_{cr}$ the instantaneous rotor voltages, currents and fluxes respectively:

$$\begin{bmatrix} v_{ar} \\ v_{br} \\ v_{cr} \end{bmatrix} = R_r \begin{bmatrix} i_{ar} \\ i_{br} \\ i_{cr} \end{bmatrix} + \frac{d}{dt} \begin{bmatrix} \psi_{ar} \\ \psi_{br} \\ \psi_{cr} \end{bmatrix} \tag{2.4}$$

Similarly, as for the stator circuits, the rotor fluxes can be represented in matrix form as a function of the stator and rotor instantaneous currents as (2.5), where $L_{\sigma r}$ is the leakage inductance of the rotor windings.

$$\begin{bmatrix}\psi_{ar}\\ \psi_{br}\\ \psi_{cr}\end{bmatrix}=\begin{bmatrix}L_{\sigma r}+L_h & -\frac{1}{2}L_h & -\frac{1}{2}L_h\\ -\frac{1}{2}L_h & L_{\sigma r}+L_h & -\frac{1}{2}L_h\\ -\frac{1}{2}L_h & -\frac{1}{2}L_h & L_{\sigma r}+L_h\end{bmatrix}\begin{bmatrix}i_{ar}\\ i_{br}\\ i_{cr}\end{bmatrix}$$
$$+L_h\begin{bmatrix}cos\theta_m & cos(\theta_m-\frac{2\pi}{3}) & cos(\theta_m+\frac{2\pi}{3})\\ cos(\theta_m+\frac{2\pi}{3}) & cos\theta_m & cos(\theta_m-\frac{2\pi}{3})\\ cos(\theta_m-\frac{2\pi}{3}) & cos(\theta_m+\frac{2\pi}{3}) & cos\theta_m\end{bmatrix}\begin{bmatrix}i_{as}\\ i_{bs}\\ i_{cs}\end{bmatrix} \quad (2.5)$$

2.3 Basics of Space vector Theory

Given a generic set of three-phase variables$x_a, x_b and\ x_c$, the corresponding complex space vector, $x_{\alpha\beta}$,is defined as (2.6). The complex space vector is seen to consist of a real component x_α, and an imaginary component x_β .

$$\overrightarrow{x_s}=x_\alpha+jx_\beta=\frac{2}{3}(x_a+a.x_b+a^2.x_c) \quad (2.6)$$
$$a=e^{\frac{j2\pi}{3}} \quad (2.7)$$

Equation (2.6) can be written using a polar form (2.8), X and θ_s being the amplitude and the angle respectively. The constant $\frac{2}{3}$ in (2.6) is chosen for convenience to preserve the amplitude of the variables.

From (2.6) the relationship between the $\alpha\beta$ components of the complex vector and the abc quantities are:

$$\vec{x}_s=Xe^{j\theta s} \quad (2.8)$$
$$x_\alpha=Re\{x_{\alpha\beta}\}=\frac{2}{3}\left(x_a-\frac{1}{2}x_b-\frac{1}{2}x_c\right) \quad (2.9)$$
$$x_\beta=Im\{x_{\alpha\beta}\}=\frac{1}{\sqrt{3}}(x_b-x_c) \quad (2.10)$$

The above equations can be represented in a matrix form (2.11), with matrix T often being called the direct Clarke transformation. The term x_0 in (2.11) is the zero- sequence component and is needed for the matrix T to be invertible.

$$\begin{bmatrix} x_\alpha \\ x_\beta \\ x_0 \end{bmatrix} = T. \begin{bmatrix} x_a \\ x_b \\ x_c \end{bmatrix} \tag{2.11}$$

$$T = \frac{2}{3}\begin{bmatrix} 1 & -1/2 & -1/2 \\ 0 & \frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \end{bmatrix} \tag{2.12}$$

The inverse Clarke transformation is given by:

$$\begin{bmatrix} x_a \\ x_b \\ x_c \end{bmatrix} = T^{-1}. \begin{bmatrix} x_\alpha \\ x_\beta \\ x_0 \end{bmatrix} \tag{2.13}$$

The space complex vector defined by (2.8) is referred to a stationary reference frame. Complex space vectors that result from applying this transformation to the voltages, currents, and fluxes in induction machines and AC systems in general will rotate in steady state at the fundamental excitation frequency ω_s. It is useful for modeling, analysis, and control purposes to transform the complex vectors from the stationary reference frame to a frame rotating at the same angular frequency as the three-phase variables, which is denoted as the synchronous reference frame [8] . This is done using (2.14), the resulting complex vector in the synchronous reference frame being indicated by an e superscript.

$$\vec{x}^e = \vec{x}^s \, e^{-j\theta s} \tag{2.14}$$

The transformation between the stationary to the rotating reference frame when applied to the Cartesian coordinates is given by (2.14), which is known as the rotational transformation also known as Park's transformation. The components of the resulting complex vector in the synchronous reference frame are indicated by subscripts d and q respectively.

$$\begin{bmatrix} x_d \\ x_q \end{bmatrix} = \begin{bmatrix} \cos\theta_s & -\sin\theta_s \\ \sin\theta_s & \cos\theta_s \end{bmatrix}. \begin{bmatrix} x_\alpha \\ x_\beta \end{bmatrix} \tag{2.15}$$

The transformation is shown schematically in figure below.

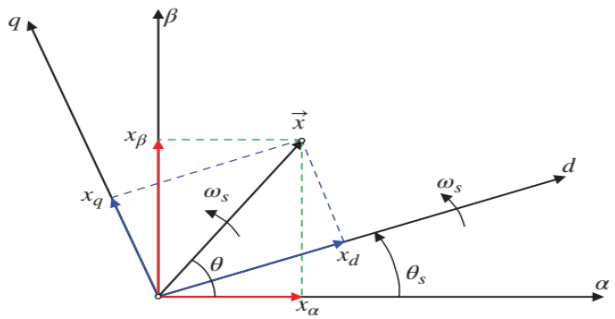


Figure 2.3: Transformation from αβ components to dq components

The transformation of a complex vector from the synchronous to the stationary reference frame is readily obtained from (2.16).

$$\vec{x}^s = \vec{x}^e\, e^{j\theta s} \tag{2.16}$$

2.3.1 Dynamic model of the induction motor using complex space vectors

Using the definition of the space complex vectors in (2.6), the corresponding stator voltage, current, and flux linkage space complex space vectors are obtained as (2.17), (2.18), and (2.19) respectively.

$$\overrightarrow{v_s}^s = \frac{2}{3}(v_{as} + a.v_{bs} + a^2.v_{cs}) \tag{2.17}$$

$$\vec{i_s}^{\,s} = \frac{2}{3}(i_{as} + a.i_{bs} + a^2.i_{cs}) \tag{2.18}$$

$$\overrightarrow{\psi_s}^s = \frac{2}{3}(\psi_{as} + a.\psi_{bs} + a^2.\psi_{cs}) \tag{2.19}$$

stator flux is expressed as (2.20):

$$\vec{\psi}_s^s = L_s\vec{i}_s^s + L_m\vec{i}_r^r\, e^{j\theta_m} \tag{2.20}$$

The stator voltage equation using complex vector notation is can be obtained applying

$$\vec{v}_s^s = R_s\vec{i}_s^s + \frac{d\vec{\psi}_s^s}{dt} \tag{2.21}$$

And combining the above two equations, the following expression for the stator voltage equation is reached.

$$\vec{v}_s^s = R_s\vec{i}_s^s + L_s\frac{d\vec{i}_s^s}{dt} + L_m\frac{d\left(\vec{i}_r^r\, e^{j\theta_m}\right)}{dt} \tag{2.22}$$

An identical procedure can be followed with the rotor voltage equation, which is reproduced here for convenience.

$$\begin{bmatrix} v_{ar} \\ v_{br} \\ v_{cr} \end{bmatrix} = R_r \begin{bmatrix} i_{ar} \\ i_{br} \\ i_{cr} \end{bmatrix} + \frac{d}{dt} \begin{bmatrix} \psi_{ar} \\ \psi_{br} \\ \psi_{cr} \end{bmatrix} \tag{2.23}$$

The corresponding voltage, current, and flux linkage complex space vectors are defined by (2.24), (2.25), and (2.26), where superscript *r* stands for a rotor reference frame:

$$\vec{v}_r^r = \frac{2}{3}(v_{ar} + a.v_{br} + a^2.v_{cr}) \tag{2.24}$$

$$\vec{i}_r^r = \frac{2}{3}(i_{ar} + a.i_{br} + a^2.i_{cr}) \tag{2.25}$$

$$\vec{\psi}_r^r = \frac{2}{3}(\psi_{ar} + a.\psi_{br} + a^2.\psi_{cr}) \tag{2.26}$$

Applying the transformations to the rotor circuit, the following expression of the rotor flux space vector is obtained:

$$\vec{\psi}_r^r = \left(\frac{3}{2}L_h + L_{\sigma r}\right)\vec{i}_r^r + \frac{3}{2}L_h\vec{i}_s^s e^{-j\theta_m} \tag{2.27}$$

Using the definition of the mutual inductance in (2.28) and defining the rotor inductance as below, the rotor flux can be written as (2.29)

$$L_r = L_m + L_{\sigma r} \tag{2.28}$$

$$\vec{\psi}_r^r = L_r\vec{i}_r^r + L_m\vec{i}_s^s e^{-j\theta_m} \tag{2.29}$$

The space vector voltage equation of the rotor circuit is obtained from (2.4) and (2.6)

$$\vec{v}_r^r = R_r\vec{i}_r^r + \frac{d\vec{\psi}_r^r}{dt} \tag{2.30}$$

Which combined with (2.29) results in;

$$\vec{v}_r^r = R_r\vec{i}_r^r + L_r\frac{d\vec{i}_r^r}{dt} + L_m\frac{d\left(\vec{i}_s^s e^{-j\theta_m}\right)}{dt} \tag{2.31}$$

In squirrel cage induction machines, there is no voltage supply to the rotor circuit, the rotor voltage equation being therefore:

$$0 = R_r\vec{i}_r^r + L_r\frac{d\vec{i}_r^r}{dt} + L_m\frac{d\left(\vec{i}_s^s e^{-j\theta_m}\right)}{dt} \tag{2.32}$$

2.3.2 Dynamic model of the induction motor in stationary reference frame

Transforming the rotor voltage equation (2.32) to the stationary reference frame and using (2.16), (2.33) is obtained. This equation, combined with (2.22), provides the induction machine voltage equations in the stationary reference frame.

$$0 = R_r \vec{i}_r^s + \frac{d\vec{\psi}_r^s}{dt} - j.\omega_m.\vec{\psi}_r^s \tag{2.33}$$

Separating the stator voltage equation (2.22) and the rotor voltage equation (2.33) into their $\alpha\beta$ components, (2.34) – (2.35) and (2.36) – (2.37) are obtained respectively.

$$v_{\alpha s} = R_s i_{\alpha s} + \frac{d\psi_{\alpha s}}{dt} \tag{2.34}$$

$$v_{\beta s} = R_s i_{\beta s} + \frac{d\psi_{\beta s}}{dt} \tag{2.35}$$

$$0 = R_r i_{\alpha r} + \frac{d\psi_{\alpha r}}{dt} + \omega_m \psi_{\beta r} \tag{2.36}$$

$$0 = R_r i_{\beta r} + \frac{d\psi_{\beta r}}{dt} - \omega_m \psi_{\alpha r} \tag{2.37}$$

On the other hand, from (2.20) and (2.29), the corresponding stator and rotor flux space complex vectors in stationary reference frame are given by

$$\vec{\psi}_s^s = L_s.\vec{i}_s^s + L_m \vec{i}_r^r \tag{2.38}$$

$$\vec{\psi}_r^s = L_m.\vec{i}_s^s + L_r \vec{i}_r^r \tag{2.39}$$

Taking the real and imaginary components the following equations are obtained

$$\psi_{\alpha s} = L_s i_{\alpha s} + L_m i_{\alpha r} \tag{2.40}$$

$$\psi_{\beta s} = L_s i_{\beta s} + L_m i_{\beta r} \tag{2.41}$$

$$\psi_{\alpha r} = L_m i_{\alpha s} + L_r i_{\alpha r} \tag{2.42}$$

$$\psi_{\beta r} = L_m i_{\beta s} + L_r i_{\beta r} \tag{2.43}$$

2.3.3 Dynamic model of the induction motor in synchronous reference frame

The dynamic equations of the stator and rotor circuits of the induction machine in the stationary reference frame (2.21) and (2.33) can be transformed to a reference frame synchronous with the fundamental excitation by multiplying them by $e^{-j\theta s}$. The resulting

stator and rotor voltage equations are (2.44) – (2.45), superscript e denoting space vectors referred to a synchronously rotating reference frame.

$$\vec{v}_s^e = R_s\vec{i}_s^e + \frac{d\vec{\psi}_s^e}{dt} + j\omega_s\vec{\psi}_s^e \tag{2.44}$$

$$0 = R_r\vec{i}_r^e + \frac{d\vec{\psi}_r^e}{dt} - j(\omega_s - \omega_m)\vec{\psi}_r^e \tag{2.45}$$

Taking the real and imaginary components of the stator voltage equation yields:

$$v_{ds}^e = R_s i_{ds}^e + \frac{d(\psi_{ds}^e)}{dt} - \omega_s\psi_{qs}^e \tag{2.46}$$

$$v_{qs}^e = R_s i_{qs}^e + \frac{d(\psi_{qs}^e)}{dt} - \omega_s\psi_{ds}^e \tag{2.47}$$

Similarly, for the rotor voltage equation:

$$0 = R_r i_{dr}^e + \frac{d\psi_{dr}^e}{dt} - (\omega_s - \omega_m)\psi_{dr}^e \tag{2.48}$$

$$0 = R_r i_{qr}^e + \frac{d\psi_{qr}^e}{dt} + (\omega_s - \omega_m)\psi_{dr}^e \tag{2.49}$$

Realizing the same transformation with the stator and rotor fluxes (2.38) and (2.39), the stator and rotor flux equations referenced to the synchronously rotating reference frame are obtained.

$$\vec{\psi}_s^e = L_s\vec{i}_s^e + L_m\vec{i}_r^e \tag{2.50}$$

$$\vec{\psi}_r^e = L_m\vec{i}_s^e + L_r\vec{i}_r^e \tag{2.51}$$

And decomposing into real and imaginary components:

$$\psi_{ds}^e = L_s i_{ds}^e + L_m i_{dr}^e \tag{2.52}$$

$$\psi_{qs}^e = L_s i_{qs}^e + L_m i_{qr}^e \tag{2.53}$$

$$\psi_{dr}^e = L_m i_{ds}^e + L_r i_{dr}^e \tag{2.54}$$

$$\psi_{qr}^e = L_m i_{qs}^e + L_r i_{qr}^e \tag{2.55}$$

2.3.4 Torque Equation

The torque equation of a three-phase induction machine can be represented as [8]:

$$T_{em} = [i_{as}\ i_{bs}\ i_{cs}].\frac{dL_{sr}}{d\theta_m}.\begin{bmatrix} i_{ar} \\ i_{br} \\ i_{cr} \end{bmatrix} \tag{2.56}$$

Being L_{sr} matrix inductance:

$$L_{sr} = L_h\begin{bmatrix} cos\theta_m & cos(\theta_m + \frac{2\pi}{3}) & cos(\theta_m - \frac{2\pi}{3}) \\ cos(\theta_m - \frac{2\pi}{3}) & cos\theta_m & cos(\theta_m + \frac{2\pi}{3}) \\ cos(\theta_m + \frac{2\pi}{3}) & cos(\theta_m - \frac{2\pi}{3}) & cos\theta_m \end{bmatrix} \tag{2.57}$$

The inductance matrix can be expressed as:

$$L_{sr} = L_h\begin{bmatrix} \frac{e^{j\theta m} + e^{-j\theta m}}{2} & \frac{e^{j\left(\theta_m + \frac{2\pi}{3}\right)} + e^{-j\left(\theta_m + \frac{2\pi}{3}\right)}}{2} & \frac{e^{j\left(\theta_m - \frac{2\pi}{3}\right)} + e^{-j\left(\theta_m - \frac{2\pi}{3}\right)}}{2} \\ \frac{e^{j\left(\theta_m - \frac{2\pi}{3}\right)} + e^{-j\left(\theta_m - \frac{2\pi}{3}\right)}}{2} & \frac{e^{j\theta m} + e^{-j\theta m}}{2} & \frac{e^{j\left(\theta_m + \frac{2\pi}{3}\right)} + e^{-j\left(\theta_m + \frac{2\pi}{3}\right)}}{2} \\ \frac{e^{j\left(\theta_m + \frac{2\pi}{3}\right)} + e^{-j\left(\theta_m + \frac{2\pi}{3}\right)}}{2} & \frac{e^{j\left(\theta_m - \frac{2\pi}{3}\right)} + e^{-j\left(\theta_m - \frac{2\pi}{3}\right)}}{2} & \frac{e^{j\theta m} + e^{-j\theta m}}{2} \end{bmatrix} \tag{2.58}$$

Using (2.7), the derivative of the inductance matrix with respect to θ_m is:

$$\frac{dL_{sr}}{d\theta_m} = j.e^{j\theta_m}.\frac{L_h}{2}.\begin{bmatrix} 1 & a & a^2 \\ a^2 & 1 & a \\ a & a^2 & 1 \end{bmatrix} - j.e^{-j\theta_m}.\frac{L_h}{2}.\begin{bmatrix} 1 & a^2 & a \\ a & 1 & a^2 \\ a^2 & a & 1 \end{bmatrix} \tag{2.59}$$

The torque equations is a function of three-phase variables being:

$$\begin{aligned} T_{em} = {} & j.e^{j\theta_m}.\frac{L_h}{2}.[i_{as}\ i_{bs}\ i_{cs}].\begin{bmatrix} 1 & a & a^2 \\ a^2 & 1 & a \\ a & a^2 & 1 \end{bmatrix}\begin{bmatrix} i_{ar} \\ i_{br} \\ i_{cr} \end{bmatrix} \\ & - j.e^{-j\theta_m}.\frac{L_h}{2}.[i_{as}\ i_{bs}\ i_{cs}].\begin{bmatrix} 1 & a^2 & a \\ a & 1 & a^2 \\ a^2 & a & 1 \end{bmatrix}\begin{bmatrix} i_{ar} \\ i_{br} \\ i_{cr} \end{bmatrix} \end{aligned} \tag{2.60}$$

Which after rearranging terms results in:

$$T_{em} = j.e^{j\theta_m}.\frac{L_h}{2}.[i_{as}\ i_{bs}\ i_{cs}].\begin{bmatrix}1\\a^2\\a\end{bmatrix}.[1\ a\ a^2].\begin{bmatrix}i_{ar}\\i_{br}\\i_{cr}\end{bmatrix} - j.e^{-j\theta_m}..\frac{L_h}{2}.[i_{as}\ i_{bs}\ i_{cs}].\begin{bmatrix}1\\a\\a^2\end{bmatrix}.[1\ a^2\ a].\begin{bmatrix}i_{ar}\\i_{br}\\i_{cr}\end{bmatrix} \quad (2.61)$$

Replacing the phase currents by the corresponding space vectors, the following torque equation is obtained, superscript * stating for the complex conjugate of the corresponding space vector.

$$T_{em} = \frac{9}{4}\frac{L_h}{2}.(j.\vec{i}_s^*.e^{j\theta_m}.\overrightarrow{i_r}i_r - j.\overrightarrow{i_s}.e^{-j\theta_m}\vec{i}_r^*) \quad (2.62)$$

By analyzing the last equation, it is possible to distinguish the difference between two conjugated space vectors:

$$\vec{x} = \vec{i}_s.e^{-j\theta_m}.\vec{i}_r^* \quad (2.63)$$

$$\vec{x}^* = \vec{i}_s^*.e^{j\theta_m}.\vec{i}_r \quad (2.64)$$

These resulting space vectors are complex conjugated; therefore :

$$\vec{x} - \vec{x}^* = (a + j.b) - (a - j.b) = 2.j.b \quad (2.65)$$

This means that the subtraction of a space vector and its complex conjugate results in two times the imaginary part. Therefore, by using this last result, the torque expression yields:

$$T_{em} = -\frac{9}{4}\frac{L_h}{2}.j.(2j).Im\{\vec{x}\} \quad (2.66)$$

Which is equal to:

$$T_{em} = \frac{3}{2}\left(\frac{3}{2}L_h\right).I_m\{\vec{x}\} \quad (2.67)$$

Or equivalently:

$$Tem = \frac{3}{2}.L_m.Im\{\vec{i}_s.e^{-j\theta_m}.\vec{i}_r^*\} \quad (2.65)$$

It can be noted that the torque is equal to a constant $\left(\frac{3}{2}L_m\right)$, multiplied by the imaginary part of a product of current space vectors.

Combining the stator and rotor current and flux complex space vectors, several alternative expressions of the electromagnetic torque produced by the machine can be obtained.

$$T_{em} = \frac{3}{2}.p.Im\{\psi_{\alpha\beta r}.i_{\alpha\beta r}^*\} = \frac{3}{2}.p.(\psi_{\beta r}.i_{\alpha r} - \psi_{\alpha r}.i_{\beta r}) \quad (2.66)$$

$$T_{em} = \frac{3}{2}.p.\frac{L_m}{L_s}.Im\{\vec{\psi}_s^{s*}\vec{i}_r^{s}\} \quad (2.67)$$

$$T_{em} = \frac{3}{2}.p.Im\left\{\vec{\psi}_s^{s*}\vec{i}_s^{s}\right\} \quad (2.68)$$

$$T_{em} = \frac{3}{2}.p.\frac{L_m}{L_r}.Im\left\{\vec{\psi}_r^{s*}\vec{i}_s^{s}\right\} \quad (2.69)$$

$$T_{em} = \frac{3}{2}\frac{L_m}{\sigma L_r l_s}.p.Im\{\vec{\psi}_r^{s*}\vec{\psi}_s^{s}\} \quad (2.70)$$

$$T_{em} = \frac{3}{2}.L_m p.Im\{\vec{i}_r^{s}\vec{i}_s^{s}\} \quad (2.71)$$

The leakage coefficient in the above equation

$$\sigma = 1 - \frac{L_m^2}{L_s L_r} \quad (2.72)$$

The torque equation as a function of the rotor flux and the stator current in the synchronous reference frame (2.73) can be obtained directly from (2.69):

$$T_{em} = \frac{3}{2}\frac{L_m}{L_r}p\ Im\left\{\vec{\psi}_r^{e*}\vec{i}_s^{e}\right\} = \frac{3}{2}\frac{L_m}{L_r}\ p\left(\psi_{dr}^{e}.i_{qs}^{e} - \psi_{qr}^{e}.i_{ds}^{e}\right) \quad (2.73)$$

2.4 Field Oriented Control

Referring the voltage, current, and flux complex space vectors to a reference frame which rotates synchronously with the fundamental excitation has been shown to convert quantities which are AC in steady state into DC. Implementation in which the synchronous reference frame is aligned with the fundamental wave of one of the fluxes in the machine (rotor, stator, or airgap) is advantageous for analysis and control purposes [2]. Consequently, three different types of vector control schemes can be considered, namely:

- rotor flux-oriented vector control;
- stator flux-oriented vector control;
- airgap flux-oriented vector control.

In rotor flux-oriented control, also called field-oriented control, all the variables are referred to a reference frame aligned with the rotor flux. By doing this, the rotor flux and the torque produced by the machine can be controlled separately through the *d* and *q* axis components of the stator current, eventually resulting in simpler control structures. Rotor flux-oriented vector control follows therefore a philosophy similar to the control of separately excited DC machines. This equivalence is graphically shown in Fig. 2-6. Referring the stator currents to a reference frame aligned with the rotor flux allows an

independent control of flux and torque, similarly as for the DC machine, with ids controlling the flux, iqs being used to control the torque.

Conversely, decoupling between the *d* and *q* axis currents for the control of flux and torque is lost when stator or airgap fluxes are used as the reference frame. Owing to this, rotor flux-oriented control has become the most popular choice. Only this strategy is covered in this thesis. Initially, two different implementations were developed using rotor flux-oriented vector control: direct vector control (feedback method) and indirect vector control (feedforward method).

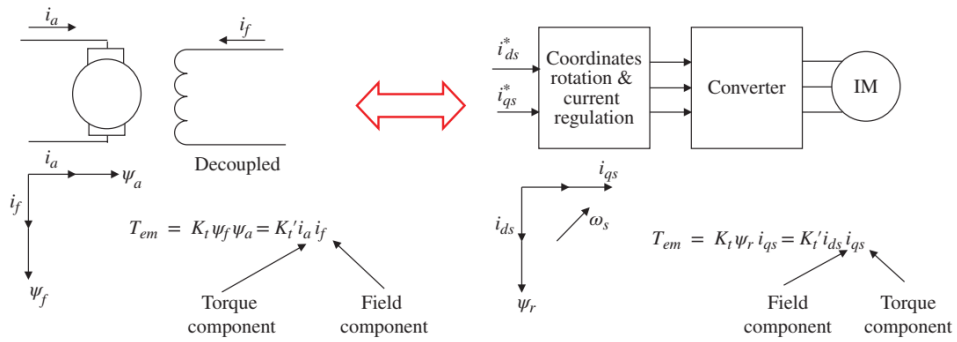


Figure 2.4: Separately excited DC motor versus vector-controlled induction motor [8]

In both direct and indirect field-oriented drives, the method to achieve the condition that the rotor flux and rotor current vectors are always perpendicular is twofold. The first part of the strategy is to ensure that

$$\psi^e_{qr} = 0 \tag{2.74}$$

And the second to is to ensure that

$$i^e_{dr} = 0 \tag{2.75}$$

Clearly, if the above two equations hold during transient conditions, the rotor flux linkage and rotor current vectors are perpendicular during those same conditions. By suitable choice of θ_e on an instantaneous basis, $\psi^e_{qr} = 0$ can always be satisfied by choosing the position of the synchronous reference frame to put all of the rotor flux linkage in the d-axis. Satisfying $i^e_{dr} = 0$ can be accomplished by forcing the d-axis stator current to remain constant [10].

To see this, consider the d-axis rotor voltage equation (with zero rotor voltage):

$$0 = R_r i_{dr}^e + \frac{d\psi_{dr}^e}{dt} - (\omega_s - \omega_m)\psi_{qr}^e \tag{2.76}$$

By suitable choice of reference frame, ψ_{qr}^e can be set to zero the above equation becomes

$$0 = R_r i_{dr}^e + \frac{d\psi_{dr}^e}{dt} \tag{2.77}$$

Substitution of the d-axis rotor flux linkage equation (2.54) into the above equation and rearranging yields:

$$0 = R_r i_{dr}^e + \frac{d}{dt}(L_r i_{dr}^e + L_M i_{ds}^e) \tag{2.78}$$

$$\frac{d(i_{dr}^e)}{dt} = -\frac{R_r}{L_r} i_{dr}^e - \frac{L_M}{L_{rr}} \frac{d(i_{ds}^e)}{dt} \tag{2.79}$$

The above equation can be viewed as stable first order differential equation in i_{dr}^e with $\frac{d(i_{ds}^e)}{dt}$ as input. Therefore, if i_{ds}^e is held constant, then i_{dr}^e will go to, and stay at, zero, regardless of other transients which may be taking place.

The indirect vector control method exploits the inherent characteristics of the induction machine as follows: By adjusting the slip angular frequency and the magnitude of the stator current the rotor flux and the torque component current can be controlled separately. In this control method, there is no need to identify the position of the rotor flux linkage. However, to control the slip angular frequency, the instantaneous rotor speed should be measured.

Although direct field-oriented control can be made fairly robust with respect to variation of machine parameters, the sensing of the air-gap flux linkage (typically) using hall- effect sensors is somewhat problematic (and expensive) in practice. This has led to considerable interest in indirect field-oriented control methods that are more sensitive to knowledge of the machine parameters but do not require direct sensing of the rotor flux linkages.

An algorithm for implementing field-oriented control without knowledge of the rotor flux linkages, from the q-axis rotor voltage equation

$$0 = R_r i_{dr}^e + (\omega_e - \omega_m)\psi_{dr}^e + p\psi_{qr}^e \tag{2.80}$$

Since,$\psi_{qr}^e = 0$

The above equation becomes

$$0 = R_r i_{qr}^e + (\omega_e - \omega_m)\psi_{dr}^e \quad (2.81)$$

$$\omega_m = \omega_e - \frac{R_r i_{qr}^e}{\psi_{dr}^e} \quad (2.82)$$

Expressing $i_{qr}^e \; interms \; of \; i_{qs}^e$

$$\psi_{qr} = L_m i_{qs} + L_r i_{qr} \quad (2.83)$$

$$L_m i_{qs} + L_r i_{qr} = 0 \quad (2.84)$$

$$i_{qr}^e = -\frac{L_m}{L_r} i_{qs}^e \quad (2.85)$$

$$\psi_{dr} = L_m i_{ds} + L_r i_{dr} \quad (2.86)$$

$$\psi_{dr}^e = \boldsymbol{L}_m i_{ds}^e \quad (2.87)$$

$$\omega_e = \omega_m + \frac{R_r}{L_m}\frac{i_{qs}^e}{i_{ds}^e} \quad (2.88)$$

Instead of mechanical speed sensor, ANN estimator is be applied in this thesis to extract ω_m. θ_e Is calculated by integrating the above equation

$$\theta_e = \int \left(\omega_m + \frac{R_r}{L_r}\frac{i_{qs}^e}{i_{ds}^e}\right) dt \quad (2.89)$$

It is considerably simpler than the direct field-oriented control—though it is much more susceptible to performance degradation as a result of error in estimating the effective machine parameters.

The torque equation in the rotor flux coordinates can be obtained particularizing (2.73) for the case of $\psi_{qr} = 0$.

Fig. 2-7 shows the rotor flux and stator current space vectors in a synchronous reference frame when the *d* axis is aligned with the rotor flux.

$$T_{em} = \frac{3}{2}\frac{Lm}{L_r} p\psi_{dr}.i_{qs} \quad (2.90)$$

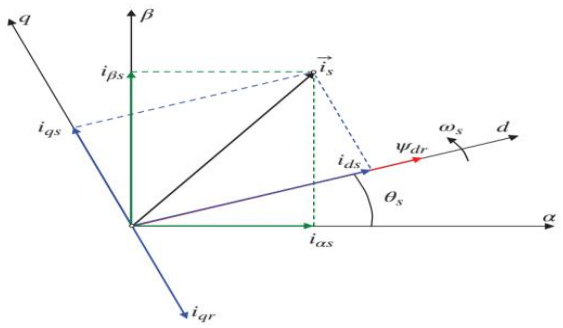


Figure 2.5: Stator and rotor current vector components in the stationary and rotor flux reference frames [8]

2.4.1 The stator voltage Equation

It has been shown in the preceding sections that the rotor flux and torque produced by the machine can be independently controlled through the *d* and *q* axis components of the stator current respectively. However, most of electronic power converters used in electric drives operate as a voltage source (i.e. they apply a voltage to the machine), the resulting current being a function of the machine parameters and of its operating point.

Combining (2.44), (2.50), and (2.51), it is possible to express the stator voltage as a function of the stator currents and the rotor flux in a rotor flux synchronous reference frame:

$$v_{ds}^{e} = R_s i_{ds}^{e} + \sigma L_s \frac{di_{ds}^{e}}{dt} - \omega_s \sigma L_s i_{qs}^{e} + \frac{L_m}{L_r}\frac{d\psi_{dr}^{e}}{dt} \tag{2.91}$$

$$v_{qs}^{e} = R_s i_{qs}^{e} + \sigma L_s \frac{di_{qs}^{e}}{dt} + \omega_s L_s i_{ds}^{e} - \omega_s \frac{L_m}{R_r}\frac{d\psi_{dr}^{e}}{dt} \tag{2.92}$$

Defining the stator transient resistance (2.93), these equations can be rearranged as (2.94) – (2.95):

$$R_s' = R_s + R_r \frac{L_m^2}{L_r^2} \tag{2.93}$$

$$v_{ds}^{e} = R_s' i_{ds}^{e} + \sigma L_s \frac{di_{ds}^{e}}{dt} - \omega_s \sigma L_s i_{qs}^{e} - R_r \frac{L_m}{L_r^2}\psi_{dr}^{e} \tag{2.94}$$

$$v_{qs}^{e} = R_s' i_{qs}^{e} + \sigma L_s \frac{di_{qs}^{e}}{dt} + \omega_s \sigma L_s i_{ds}^{e} + \omega_m \frac{L_m}{L_r}\psi_{dr}^{e} \tag{2.95}$$

These equations can be further simplified. The cross-coupling terms in the right side of (2.94) and (2.95), namely (2.96) – (2.97), are due to the coordinate transformation of the stator equivalent RL circuit to the synchronous reference frame.

$$v_{ccd} = -\omega_s \sigma L_s i_{qs}^{e} \tag{2.96}$$

$$v_{ccq} = \omega_s \sigma L_s i_{ds}^{e} \tag{2.97}$$

On the other hand, the rotor flux dependent terms in the right side of (2.94) and (2.95), namely (2.98) – (2.99) account for the effect of the rotor flux on the stator voltage (i.e. the back-emf).

$$e_d = -R_r \frac{L_m}{L_r^2}\psi_{dr}^{e} \tag{2.98}$$

$$e_q = \omega_m \frac{L_m}{L_r} \psi_{dr}^e \tag{2.99}$$

Combining (2.96) – (2.97) and (2.98) – (2.99) into the equivalent terms cds and cqs (2.100) – (2.101), the stator voltage equations can be written now as (2.102) – (2.103).

$$c_{ds} = e_d + v_{ccd} = -\omega_s \sigma L_s i_{qs}^e - R_r \frac{L_m}{L_r^2} \psi_{dr}^e \tag{2.100}$$

$$c_{qs} = e_q + v_{ccq} = \omega_s \sigma L_s i_{ds}^e + \omega_m \frac{L_m}{L_r} \psi_{dr}^e \tag{2.101}$$

$$v_{ds}^e = R_s' i_{ds}^e + \sigma L_s \frac{d i_{ds}^e}{dt} + c_{ds} \tag{2.102}$$

$$v_{qs}^e = R_s' i_{qs}^e + \sigma L_s \frac{d i_{qs}^e}{dt} + c_{qs} \tag{2.103}$$

It can be observed from (2.102)–(2.103) that if the terms c_{ds} and c_{qs} are neglected the relationship between the d and q components of the stator voltage and current corresponds to an RL load (i.e. a first-order system). Accurate, high bandwidth control of the current is therefore possible with proper current regulator designs. The terms c_{ds} and c_{qs} can be seen therefore as disturbances, which can be either explicitly decoupled or compensated by the current regulators. It is noted in this regard that both c_{ds} and c_{qs} consist of two different parts that have very different characteristics: cross-coupling terms and back-emf terms.

As already mentioned, the cross-coupling voltages v_{ccd} and v_{ccq} in (2.96) and (2.97) result from the transformation of the stator winding corresponding transient impedance to the synchronous reference frame. These voltages vary proportional to the d and q axis components of the stator current, and to the fundamental frequency. The stator current vector in current-regulated drives (especially the q axis component) can change very quickly (from zero to its rated value in ms or even less)[8]. This means that the voltages in v_{ccd} and v_{ccq} can change very quickly as well, especially v_{ccd}. These effects will be more relevant at high fundamental excitation frequencies, owing to the presence of ω_s in the equations. It is concluded that v_{ccd} and v_{ccq} can produce large disturbances with very fast dynamics, which can therefore have significant adverse effects on the current dynamics when the drive operates at high speeds.

On the other hand, the induced voltages due to the back-emf ed and eq, (2.98) and (2.99), account for the effect of the rotor flux on the stator voltage. Both terms are proportional to

the rotor flux, eq also being proportional to the rotor speed. Since the rotor flux in vector-controlled machines is kept constant, or changes relatively slowly (e.g. in the field weakening region), back-emf voltage will be characterized by slow dynamics. It is also noted that the rotor speed in electric drives change relatively slowly compared to the stator current dynamics. Consequently, *ed* and *eq* can be considered disturbances with relatively slow dynamics. Because of this, their effects are easier to compensate.

2.4.2 Current Control in synchronously Rotating Reference Frame

The voltage commands for vds and vqs required to obtain the desired currents ids and iqs could in principle be obtained in a feed-forward manner using (2.94) and (2.95). However, the sensitivity of these equations to machine parameters as well as the need to calculate the currents' derivatives present in the right hand of these equations make this solution highly inadvisable. Instead, feedback-based solutions using current regulators are preferred, owing to their simplicity and robustness.

Current regulation of AC drives presents two distinguishing characteristics:

- Though all the three-phase currents need to be controlled simultaneously, only two independent variables exist, as the phase currents add to zero. Any strategy trying to control the three-phase current using three independent current regulators is therefore intrinsically incorrect.
- Contrary to most systems, the electrical variables in AC machines are sinusoidal in steady state, with frequencies that can go from zero up to several hundred Hz or even higher. Because of this, the use of an integral action in the controller is no longer capable of providing zero steady-state errors, assuming the currents are controlled in the stationary reference frame.

A variety of current regulator designs have been proposed to realize the current control in AC drives, but a thorough discussion of this topic is beyond the scope of this chapter. Among all the proposed solutions, PI regulators implemented in a synchronous reference frame are widely accepted as the standard solution for current regulation in vector-controlled AC drives [8].

Controlling the *d* and *q* axis currents in a synchronous reference using a PI regulator provides an appropriate solution to the two aforementioned problems. Since in the synchronous reference frame the AC variables become DC in steady state, the integral

action of the PI design guarantees zero steady-state error. Also, only two independent regulators are used (one for the *d* and one for the *q* axis), which is consistent with the fact that only two independent currents exist.

2.4.2.1 Synchronous current regulator tuning using pole zero cancellation

Tuning of the synchronous PI current regulators can be addressed using well-known tools for linear systems. If ideal decoupling of the effects due to cross-coupling and the back-emf is assumed, the dynamic behavior of the *d* and *q* axis currents is given by (2.104).

$$\frac{i_{ds}^{e}(s)}{v_{ds}^{e}(s)} = \frac{i_{qs}^{e}(s)}{v_{qs}^{e}(s)} = \frac{1}{\sigma L_s s + R_s'} \tag{2.104}$$

On the other hand, the transfer function of the synchronous PI current regulators in the synchronous reference frame are given by (2.105), with ε_{ds} and ε_{qs} being the error between the corresponding current commands and the actual current for the *d* and *q* axis respectively.

$$\frac{v_{ds}^{*}(s)}{\varepsilon_{ds}(s)} = \frac{v_{qs}^{*}(s)}{\varepsilon_{ds}(s)} = k_{pi} + \frac{k_{ii}}{s} = k_{pi}\left(\frac{s + \frac{k_{ii}}{k_{pi}}}{s}\right) \tag{2.105}$$

A simple and effective way to select the gains for the current regulator is to use the zero of the regulators to cancel the dynamics of the load, as shown in (2.106).

$$\frac{k_{ii}}{k_{pi}} = \frac{R_s'}{\sigma L_s} \tag{2.106}$$

By doing this, the dynamics of the closed-loop system are (2.107), which corresponds to a first-order system with a gain equal to 1, and a bandwidth in rad/s of *CRBW* (2.108).

$$\frac{i_{ds}(s)}{i_{ds}^{*}(s)} = \frac{i_{qs}(s)}{i_{qs}^{*}(s)} = \frac{k_{pi}}{\sigma L_s s + k_{pi}} \tag{2.107}$$

$$CRBW = \frac{k_{pi}}{\sigma L_s} \tag{2.108}$$

Therefore, knowing the desired bandwidth of the current regulator *CRBW*, the proportional gain of the current regulator k_{pi} is obtained using (2.108), the integral gain needed to obtain pole-zero cancellation k_{pi} being given by (2.106).

2.4.3 The Flux Observer

Rotor flux oriented control of induction machines requires knowledge of the rotor flux magnitude and phase angle. Though the use of rotor flux sensors was investigated in the past, such sensors are never used in practice nowadays, owing to cost and reliability issues. Therefore, the rotor flux needs to be estimated from measurable variables. These include in principle the stator voltages and currents and the rotor speed/position.

It is possible to estimate the rotor flux using the stator voltage equations in the stationary reference frame. Using (2.34) and (2.35), the stator flux in the stationary reference frame can be estimated as:

$$\hat{\psi}_{\alpha s} = \int (v_{as} - R_s i_{\alpha s}) dt \tag{2.109}$$

$$\hat{\psi}_{\beta s} = \int (v_{\beta s} - R_s i_{\beta s}) dt \tag{2.110}$$

Using the known relationships among the stator and rotor fluxes and currents (2.52) – (2.55) the following equations linking the rotor fluxes with stator fluxes and currents obtained:

$$\hat{\psi}_{\alpha s} = \sigma L_s i_{\alpha s} + \frac{L_m}{L_r} \psi_{\alpha r} \tag{2.111}$$

$$\hat{\psi}_{\beta s} = \sigma L_s i_{\beta s} + \frac{L_m}{L_r} \psi_{\beta r} \tag{2.112}$$

Combining (2.109- 2.110) and (2.111- 2.112), the α and β components of the rotor flux in the stationary reference frame are obtained:

$$\hat{\psi}_{\alpha r} = \frac{Lr}{L_m} \int (v_{as} - R_s i_{\alpha s}) dt - \frac{\sigma L_s Lr}{L_m} i_{\alpha s} \tag{2.113}$$

$$\hat{\psi}_{\beta r} = \frac{Lr}{L_m} \int (v_{\beta s} - R_s i_{\beta s}) dt - \frac{\sigma L_s Lr}{L_m} i_{\beta s} \tag{2.114}$$

The amplitude and phase of the estimated rotor flux in the stator reference frame are given by:

$$|\vec{\psi}_r| = \sqrt{\psi_{\alpha r}^2 + \psi_{\beta r}^2} \tag{2.115}$$

$$\theta_s = \operatorname{atan}\left(\frac{\psi_{\beta r}}{\psi_{\alpha r}}\right) \tag{2.116}$$

2.4.4 Speed Control

In high-performance electric drives, the current loop is tuned to have a bandwidth much faster than the speed control loop. The current control dynamics can therefore be safely

neglected. Consequently, the actual torque can be assumed to be nearly equal to the commanded torque (2.117):

$$T_{em}^{*} \approx T_{em} = K\psi_{dr} i_{qs} \tag{2.117}$$

The overall mechanical equation will depend on the mechanical characteristics of the machine and load. The machine can be considered as nearly a pure mechanical, invariant inertia, friction being almost negligible. However, the mechanical characteristics of the load strongly depend on the application. Consequently, selection of the parameters for the speed regulator will depend on the mechanical load characteristic; no general tuning procedure can be given. Tuning of the speed controller for the case of a load including mechanical inertia and a viscous friction, as well as the load torque (2.118), is briefly discussed here.

$$T_{em} = T_L + J\frac{d\Omega}{dt} + B\Omega \tag{2.118}$$

The transfer function that links the electrical speed with the torque produced by the machine is:

$$\frac{\omega_m(s)}{T_{em}(s)} = \frac{p}{2\pi}\left(\frac{1}{Js + B}\right) \tag{2.119}$$

The dynamics of the speed control loop are normally selected to be significantly slower than those of the current control loop. This restriction comes first from the physical nature of the mechanical systems, whose time constant is significantly larger than that of the electrical subsystem. In addition, proper operation of cascaded control systems requires that the inner (current) control loop behaves significantly faster than the outer (speed) control loop, as this enables independent design and tuning of both control loops.

Being the transfer function of the PI speed controller (2.120), the transfer function linking the actual speed with the speed command in electrical units is given by (2.121), while the transfer function linking the speed with the disturbance torque is (2.122).

$$PI_f = k_{pf} + \frac{k_{if}}{s} \tag{2.120}$$

$$\frac{\omega_m(s)}{\omega_m^{*}(s)} = \frac{k_{pf}.s + k_{if}}{\frac{2\pi}{p} J\, s^2 + \left(\frac{2\pi}{p} B + k_{pf}\right)s + k_{if}} \tag{2.121}$$

$$\frac{\omega_m(s)}{T_L(s)} = \frac{s}{\frac{2\pi}{p} J\, s^2 + \left(\frac{2\pi}{p} B + k_{pf}\right)s + k_{if}} \tag{2.122}$$

The system dynamics are seen to correspond to those of a second-order system. Similar to that discussed for the tuning of the synchronous PI current regulator, zero-pole cancellation can be used to reduce the dynamics to those of a first-order system. To achieve this, the relationship between the mechanical parameters and the speed PI controller gains is given by (2.123). The proportional gain of the speed controller kpf is then selected to achieve the desired bandwidth.

$$\frac{k_{if}}{k_{pf}} = \frac{B}{J} \; and \; \frac{k_{pf}}{2\pi J} = SRBW \tag{2.123}$$

2.5 Space Vector Pulse Width Modulation

The SVPWM technique is one of the most popular PWM techniques due to a higher DC bus voltage use (higher output voltage when compared with the SPWM) and easy digital realization [8]. The concept of the SVPWM relies on the representation of the inverter output as space vectors or space phasors. Space vector representation of the output voltages of the inverter is realized for the implementation of SVPWM. Space vector simultaneously represents three-phase quantities as one rotating vector; hence each phase is not considered separately. The three phases are assumed as only one quantity. The space vector representation is valid for both transient and steady state conditions in contrast to phasor representation, which is valid only for steady state conditions. The concept of the space vector arises from the rotating air-gap MMF in a three-phase induction machine. By supplying balanced three-phase voltages to the three-phase balanced winding of a three-phase induction machine, rotating MMF is produced, which rotates at the same speed as that of individual voltages, with an amplitude of 1.5 times the individual voltage amplitude.

The space vector is defined as (Eq. 2.6) Where x_a,x_b and x_c are the three-phase quantities of voltages, currents, and fluxes.

In the inverter PWM, the voltage space vectors are considered. Since the inverter can attain either positive 0.5 Vdc or negative 0.5 Vdc (if the DC bus has mid-point) or Vdc, 0, i.e. only two states, the total possible outputs are $2^3 = 8$ (000, 001, 010, 011, 100, 101, 110, 111). Here 0 indicates the upper switch is 'OFF' and 1 represents the upper switch is 'ON.' Thus, there are six active switching states (power flows from the input/DC link side of the

inverter to the output/load side of the inverter) and two zero switching states (no power transfer from input/DC link to the load/load side). The operation of the lower switches is complimentary. The possible space vectors are computed and listed in Table 2.1. typical three phase Voltage source inverter is also shown below in fig (2.8).

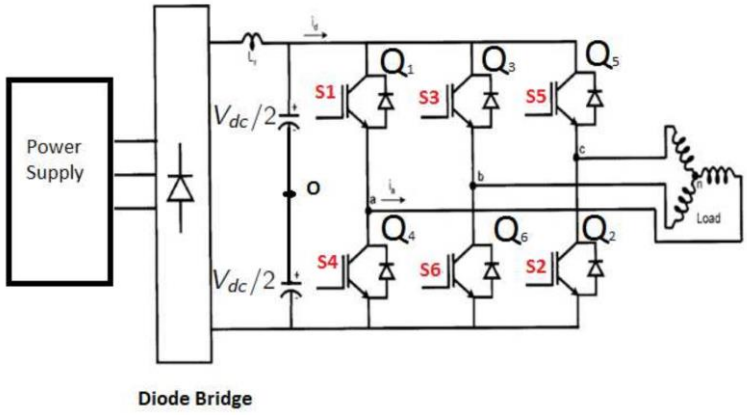


Figure 2.6: Three phase bridge Inverter [11]

The space vectors are shown graphically in Figure 2.9. The tips of the space vectors, when joined together, form a hexagon. The hexagon consists of six distinct sectors spanning over 360 degrees (one sinusoidal wave cycle corresponds to one rotation of the hexagon) with each sector of 60 degrees. Space vectors 1, 2. . ..6 are called active state vectors and 7, 8 are called zero state vectors. The zero state vectors are redundant vectors but they are used to minimize the switching frequency. The space vectors are stationary while the reference vector v_s^* is rotating at the speed of the fundamental frequency of the inverter output voltage. It circles once for one cycle of the fundamental frequency.

The reference voltage follows a circular trajectory in a linear modulation range and the output is sinusoidal. The reference trajectory will change in over-modulation and the trajectory will be a hexagon boundary when the inverter is operating in the six-step mode. In implementing the SVPWM, the reference voltage is synthesized by using the nearest two neighboring active vectors and zero vectors. The choice of the active vectors depends upon the sector number in which the reference is located. Hence, it is important to locate the position of the reference voltage. Once the reference vector is located, the vectors to be used for the SVPWM implementation to be identified. After identifying the vectors to be used, the next task is to find the time of application of each vector, called the 'dwell time'.

The output voltage frequency of the inverter is the same as the speed of the reference voltage and the output voltage magnitude is the same as the magnitude of the reference voltage [11].

Table 2.1: The possible space vectors using SVM

Switching States	Space Vector Number	Phase to neural Voltage Space Vectors
000	7	0
001	5	$\frac{2}{3}V_{dc}e^{\frac{j4\pi}{3}}$
010	3	$\frac{2}{3}V_{dc}e^{\frac{j2\pi}{3}}$
011	4	$\frac{2}{3}V_{dc}e^{j\pi}$
100	1	$\frac{2}{3}V_{dc}e^{j0}$
101	6	$\frac{2}{3}V_{dc}e^{\frac{j5\pi}{3}}$
110	2	$\frac{2}{3}V_{dc}e^{\frac{j\pi}{3}}$
111	8	0

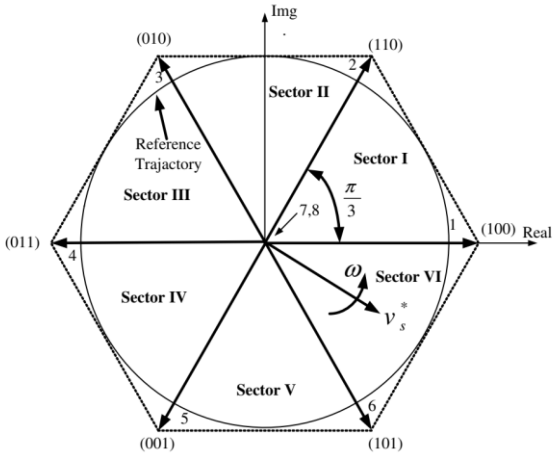


Figure 2.7: Principle of SVM [11]

The time of application of the different space vectors are calculated using the 'equal volts-second principle'. According to this principle, the product of the reference voltage and the sampling/switching time (Ts) must be equal to the product of the applied voltage vectors and their time of applications, assuming that the reference voltage remains fixed during the switching interval. When the reference voltage is in sector I, the reference voltage can

be synthesized by using the vectors V1, V2, and V_0 (zero vector), applied for time t_a,t_b andt_0, respectively. Hence, using the equal volt-second principle, for sector I:

$$V_s^* T_s = V_1 t_a + V_2 t_b + V_0 t_0 \tag{2.124}$$

$$where\ \ Ts = t_a + t_b + t_0 \tag{2.125}$$

The space vectors are given as

$$V_s^* = |V_s^*| e^{j\alpha} \tag{2.126}$$

$$V_1 = \frac{2}{3} V_{dc} e^{j0} \tag{2.127}$$

$$V_2 = \frac{2}{3} V_{dc} e^{\frac{j\pi}{3}}\ and\ \ V_0 = 0 \tag{2.128}$$

Substituting equation (2.126) into equation (2.124) and separating the real (α -axis) and imaginary (β-axis) components:

$$|V_s^*| \cos(\alpha)\, T_s = \frac{2}{3}\, V_{dc} t_a + \frac{2}{3}\, V_{dc} \cos\left(\frac{\pi}{3}\right) t_b \tag{2.129}$$

$$|V_s^*| \sin(\alpha)\, T_s = \frac{2}{3}\, V_{dc} \sin\left(\frac{\pi}{3}\right) t_b \tag{2.130}$$

When $\alpha = \frac{\pi}{3}$

Solving equations (2.129) and (2.130) for the time of applications $t_a and\ t_b$:

$$t_a = \frac{\sqrt{3}|V_s^*|}{V_{dc}} \sin\left(\frac{\pi}{3} - \alpha\right) T_s \tag{2.131}$$

$$t_b = \frac{\sqrt{3}|V_s^*|}{V_{dc}} \sin(\alpha)\, T_s \tag{2.132}$$

$$t_0 = T_s - t_a - t_b \tag{2.133}$$

Generalizing the above equation for six sectors gives the following, where $k = 1,2, \ldots . 6$ is the sector number:

$$t_a = \frac{\sqrt{3}|V_s^*|}{V_{dc}} \sin\left(k\frac{\pi}{3} - \alpha\right) T_s \tag{2.134}$$

$$t_b = \frac{\sqrt{3}|V_s^*|}{V_{dc}} \sin\left(\alpha - \frac{(k-1)\pi}{3}\right) T_s \tag{2.135}$$

$$t_0 = T_s - t_a - t_b \tag{2.136}$$

After locating the reference location and calculating the dwell time, the next step in SVPWM implementation is the determination of the switching sequence. The requirement is the minimum number of switching's to reduce switching loss, ideally one power switch should turn 'ON' and turn 'OFF' in one switching period.

In order to obtain a fixed switching frequency and optimum harmonic performance from SVPWM, each leg should change its state only once in each switching period.

This is achieved by applying the zero-state vector followed by two adjacent active state vectors in a half switching period. The next half of the switching period is the mirror image of the first half. The total switching period is divided into seven parts, the zero vector (000) is applied for 1/4th of the total zero vector time, followed by the application of active vectors for half of their application time and then again zero vector (111) is applied for $1/4^{th}$ of the zero-vector time. This is then repeated in the next half of the switching period. This is how symmetrical SVPWM is obtained.

Average leg voltages can be solved for sector I as shown:

$$V_A, avg = \frac{\left(\frac{V_{dc}}{2}\right)}{Ts}[t_0 + t_a + t_b - t_0)] \tag{2.137}$$

$$V_B, avg = \frac{\left(\frac{V_{dc}}{2}\right)}{T_s}[t_0 - t_a + t_b - t_0] \tag{2.138}$$

$$V_C, avg = \frac{(\frac{V_{dc}}{2})}{T_s}[t_0 - t_a - t_b - t_0] \tag{2.139}$$

Substituting the equation of time of applications from equation (2.134) – (2.136) for other sectors can be derived for other sectors:

$$V_A, avg = \frac{\sqrt{3}}{2}|v_s^*| \sin\left(\alpha + \frac{\pi}{3}\right) \tag{2.140}$$

$$V_B, avg = \frac{\sqrt{3}}{2}|v_s^*| \sin\left(\alpha - \frac{\pi}{6}\right) \tag{2.141}$$

$$V_C, avg = -\frac{\sqrt{3}}{2}|v_s^*| \sin\left(\alpha + \frac{\pi}{3}\right) \tag{2.142}$$

2.6 Literature Review

Induction motors are basically characterized by their dynamics in currents and fluxes under the effect of the applied voltages at the stator. Those electrical dynamics generate the mechanical torque and rotating speed of the machine. Estimation issues then arise from the fact that among all those quantities of interest, only stator currents can easily be measured online, while they are needed for control, or monitoring. From this, a lot of efforts have been dedicated to methods of reconstruction of the unmeasured variables, which amounts to a problem of observer design. The fact that a solution can exist is characterized by the so-called observability property, and in the case of induction motors, due to the nonlinear nature of the dynamics, this property depends on the operating conditions. In addition, the problem is made even more difficult by the fact that model parameters themselves can change during operation, or just not be very well known. From this, the problem of state variable reconstruction extends to that of parameter identification [24].

Sensorless control can be carried out on Induction motor drives using different methods that are available in literature. Amongst these the Observer based approach has been the most successful. Observer based models are state estimators that through the knowledge of input and output over a finite interval of time are able to determine the states of the system. The Observers have been designed for sensorless control of the Induction motor drive in order to estimate the speed of IM drives. Amongst these the Luenberger Observer and the Kalman Filter variations (i.e. Extended Kalman Filter, Unscented Kalman Filter, researched upon for sensorless IM drives. Even though these methods have been successfully implemented for sensorless operation but they offer huge computational burden and tuning of Kalman filter is particularly tedious and time consuming. Moreover, the performance of Unscented Kalman filter is unsatisfactory for load changes [4].

One of the major sources of problems with the EKF comes from the fact that the covariance of the estimation is propagated through a linear model, which leads to rather important inaccuracies. The unscented Kalman filter (UKF) adopts a superior technique to propagate this quantity, which leads to improved accuracy (although some empirical tuning is involved) [24].

Various MRAS observers have been introduced in the literature based on rotor flux, back EMF and reactive power. Closed-loop speed estimators based on machine models can give satisfactory performance in sensorless induction machine drives. Speed estimators using model reference adaptive system (MRAS) are one category of the closed-loop machine-model based estimators, which can give reliable estimation and good performance for speed sensorless induction motor drives

Rotor flux MRAS is the most popular MRAS strategy and significant attempts have been made to improve its performance [12].

Stator current MRAS speed estimator is used for closed loop speed control of induction motor without mechanical speed sensor in [13]. These schemes suffer from parameter sensitivity and pure integration problems which may limit the performance at low and zero speed region of operation.

ANNs are a family of intelligent algorithms which can be used for time series prediction, classification, and control and identification purposes. Neural networks have an ability to train with various parameter of induction motor. The main features of ANN are the self-learning and self -organizing capabilities. This makes it useful in designing an Observer based on ANN. Such a system offers advantages as it does not require the exact mathematical model of system and is mathematically less burdensome [4].

In many industrial drives advanced digital control strategies for the control of field-oriented induction motor drives with a conventional speed PI controller, have gained the widest acceptance in high performance AC servo system [12]. However, in certain applications such as steel mills and paper mills robotics, machine tools, the drive operates under a wide range of load change characteristics. Under such conditions, the system parameters vary substantially leading in most of the cases to load disturbance. The PI controller is very easy to be implemented; although it cannot lead to good tracking and regulating performance simultaneously. Further; its control performance parameters are sensitive to the system parameters variations and load disturbances.

This thesis focused on design of an ANN based rotor speed observer to handle the short comings of sensorless control schemes such as parameter sensitivity, high computational efforts and Instability at low and zero speed and ANFIS controller to control the speed of induction motor drive to minimize the problems related to conventional PI controllers.

CHAPTER 3 SPEED OBSERVER USING ARTIFICIAL NEURAL NETWORK (ANN/MLP)

3.1 Introduction

An ANN consists of a number of artificial neurons that are interconnected together. The structure of artificial neuron is inspired by the concept of biological neuron shown in Fig. 3.1. Basically, it is the processing element in the nervous system of the brain that receives and combines signals from other similar neurons through thousands of input paths called dendrites. Each input signal (electrical in nature), flowing through dendrite, passes through a synapse or synaptic junction as shown. The junction is an infinitesimal gap in the dendrite which is filled with neurotransmitter fluid that either accelerates or retards the flow of the signal. These signals are accumulated in the nucleus (or soma), nonlinearly modified at the output before flowing to other neurons through the branches of axon as shown. The adjustment of the impedance or conductance of the synaptic gap by the neurotransmitter fluid contributes to the "memory" or "intelligence" of the brain. According to the theory of the neuron, we are led to believe that our brain has distributed associative memory or intelligence characteristics which are contributed by the synaptic junctions of the cells. It may be interesting to note here that when a human baby is born, it has around 100 billion neurons in the brain.

The model of an artificial neuron closely matches the biological neuron. Basically, it has op-amp summer-like structure. The artificial neuron (or simply neuron) is also called processing element, neuron, node, or cell. Each input signal (continuous variable or discrete pulses) flows through a gain or weight (called synaptic weight or connection strength) which can be positive (excitory) or negative (inhibitory), integer or non -integer. The summing node accumulates all the input-weighted signals, adds to the weighted bias signal, and then passes to the output through the nonlinear (or linear) activation or transfer function.

This Chapter is intended to provide a general introduction to neural networks. Concepts of Artificial Neural Networks (ANN) are introduced, and their attributes are described. Many types of ANN exist and the configuration that are most applicable to the context of control problems is overviewed. The special interest from this point of view has Multilayer Neural

Network (MNN) with backpropagation learning algorithm. Application of ANN for estimation of the motor speed on the base of phase current and voltage measurement is presented for speed sensorless field-oriented control of induction motor drive.

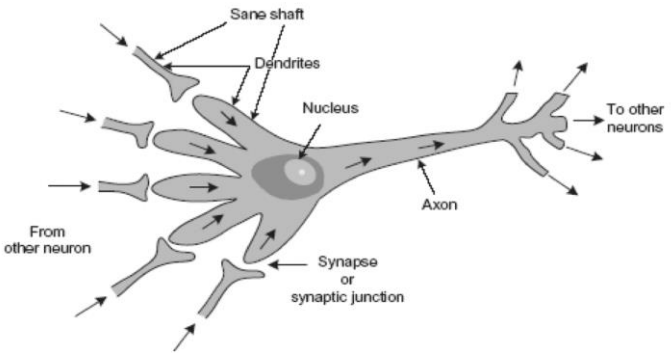


Figure 3.1: Structure of biological Neuron [2]

3.2 Basics of Artificial Neural Networks

Recently parallel distributed processing such as Artificial Neural Networks (ANN) have received wide attentions in processing a complex data in very short time. ANN models are composed of many linear or nonlinear computational elements (neurons or nodes) operating in parallel. Parallelism, robustness, and learning ability are among the main features, which determined wide applications for ANN to control of industrial processes.

Artificial Neural Networks have several important characteristics, which are of interest to control engineers [14]:

- Modeling, because of their ability to be trained using data records for the particular system of interest.
- Handling Nonlinear systems, the artificial neural networks have the ability to learn nonlinear relationship.
- Multivariable systems. Artificial Neural Networks, by their nature, have many inputs and many outputs and so can be easy applied to multivariable systems.
- Parallel structure. This feature implies very fast parallel processing, fault tolerance and robustness.

The above features are the main reasons for the interest, which is currently being concentrated in this field.

3.3 Artificial Neuron Model

The elementary computational elements, which create neural network, have many inputs and only one output. These elements are inspired by biological neurons systems and, therefore, they are call neurons (or by analogy with directed graphs - nodes).

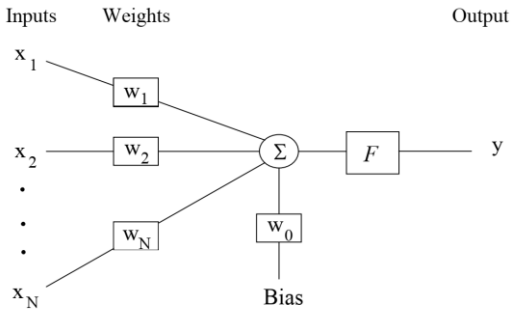


Figure 3.2: Neuron Model [14]

The individual inputs x_j weighted by elements w_j are summed to form the weighted output signal:

$$e = \sum_{j=0}^{N} w_j x_j \tag{3.1}$$

And $x_0 = 1$

where elements w_j, are called synapse weights, can be modified during the learning process.

The output of the neuron unit is defined as follows:

$$y = F(e) \tag{3.2}$$

Note, that w_o - is adjustable bias and F – is activation function (also called transfer function). Thus, the output, y, is obtained by summing the weighted inputs and passing the results through a nonlinear (or linear) activation function F.

The activation function F map, a weighted sum's e (possibly) infinite domain to a prespecified range. Although the number of F functions is possibly infinite, six types are regularly applied in the majority of ANN [14]: linear, step, bipolar, sigmoid, hyperbolic tangent. With the exception of the linear F function, all of these functions introduce a nonlinearity in the network by bounding the output within a fixed range.

In the next subsection some examples of commonly used activation functions are briefly presented.

3.3.1 Activation Functions

The linear F function Fig. 3.3 produces a linearly modulated output from the input e as described by equation:

$$F(e) = \xi e \tag{3.3}$$

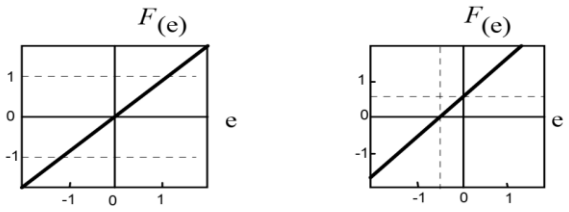


Figure 3.3: Linear Activations functions [14]

where e ranges over the real numbers and ξ is a positive scalar. If ξ = 1, it is equivalent to removing the F function completely. In this case:

$$y = \sum_{j=0}^{N} w_j x_j \tag{3.3}$$

The step F function (Fig. 3.3a) produces only two, typically, a binary value in response to the sign of the input, emitting +1 if e is positive and 0 if it is not. This function can be described as:

$$F(e) = \begin{cases} 1, if\ e \geq 0 \\ 0, otherwise \end{cases} \tag{3.4}$$

One small variation of Eq. 3.4 is the bipolar F function (see Fig. 3.4b) which replaces the 0 output value with $a = -1$.

$$F(e) = \begin{cases} 1, if\ e \geq 0 \\ -1, otherwise \end{cases} \tag{3.5}$$

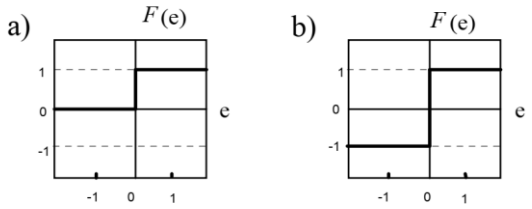


Figure3.4: Step Activation functions [14]

The sigmoid F function is a continuous, bounded, monotonic, nondecreasing function that provides a graded, nonlinear response within a prespecified range. The most common function is the logistic function:

$$F(e) = \frac{1}{1 + \exp(-\beta e)} \tag{3.6}$$

where $\beta > 0$(usually $\beta = 1$), which provides an output value from 0 to 1. The alternative to the logistic sigmoid function is the hyperbolic tangent:

$$F(e) = \tanh(\beta e) = \frac{\exp(\beta e) - \exp(-\beta e)}{\exp(\beta e) + \exp(-\beta e)} \tag{3.7}$$

Which ranges from $-1\ to + 1$

Both the hyperbolic tangent and logistic functions approximate the signum and step function, respectively, and yet provide smooth, non-zero derivatives with respect to input signals. Sometimes these two activation functions are referred to as squashing functions since the inputs to these functions are squashed to the range [0,1] or [-1,1]. They are also called sigmoidal functions because their S-shaped curves exhibit smoothness and asymptotic properties. Sometimes the hyperbolic tangent functions are referred to as bipolar sigmoidal function and the logistic function are referred to as binary sigmoidal. Both of these activations are used often on regression and classification problems.

For neural networks to approximate a continuous valued function not limited to the interval [0,1] or [-1,1], usually the node function for the output layer to be a weighted sum with no squashing functions. This is equivalent to a situation in which the activations function is an identity, and output nodes of this type are often called linear nodes.

3.4 ANN Topologies

In the biological brain, a large number of neurons are interconnected to form the network and perform advanced intelligent activities. Artificial Neural Network is built by neuron models and in most cases consists of neurons layers interconnected by weighted connections. The arrangement of the neurons, connections, and patterns into a neural network is referred to as a topology (or architecture).

3.4.1 The Layers of Neurons

Neural networks are organized into layers of neurons. Within a layer, neurons are similar in two respects:

- the connection that feed the layer of neurons are from the same source;
- the neurons in each layer utilize the same type of connections and activation F function.

A one-layer network with N inputs and M neurons is shown in Fig. 3.5.

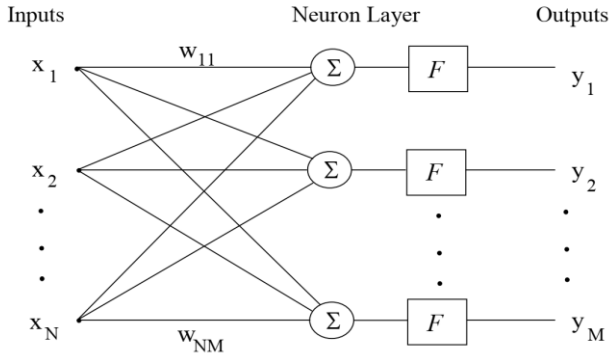


Figure 3.5: One-layer Network [14]

In this topology, each element of the input vector X is connected to each neuron input through the weight matrix **W**. The sum of the appropriate weighted network inputs **W*X** is the argument of the activation **F** function. Finally, the neuron layer outputs form a column vector **Y**. It is common for the number of inputs to be different from the number of neurons, i.e. N $\neq M$.

3.4.2 Multi-Layer Neural Networks

A neural network can have several layers. There are two types of connections applied in MNN:

- Intralayer connections are connections between neurons in the same layer.
- Interlayer connections are connections between neurons in different layers.

It is possible to build ANN that consist of one, or both, types of connections. Organization of the MNN is classified largely into two types:

- A feedforward network,
- A feedback (also called recurrent) network.

When the MNN has connections that feed information in only one direction (e.g., input to output) without any feedback pathways in the network, it is a feedforward MNN. Most widely used in Power Electronics and drive applications. But if the network has any feedback paths, where feedback is defined as any path through the network that would allow the same neurons to be visited twice, then it is call a feedback MNN.

An example of multilayer feedforward network is shown in Fig. 3.6. Each layer has a weight matrix $\boldsymbol{W}_k^{(l)}$ a weighted input $\boldsymbol{E}^{(l)}$, and an output vector $\boldsymbol{Y}^{(l)}$, where l is the layer number. The layers of a multilayer ANN play different roles. Layers whose output is the network output are called output layers. All other layers are called hidden layers. In many literatures additional layer so called input layer is introduced. This layer consists of input vector to the whole MNN (in this layer input vector is equal to output vector).

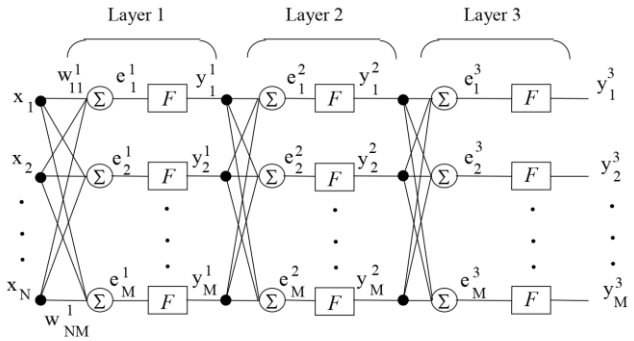


Fig 3.6 Multi-layer feed forward Neural network [15]

Feedback ANN has all possible connections between neurons. Some of the weight can be set to zero to create layers within the feedback network if that is desired. The feedback network are quite powerful because they are sequential rather than

combinational like the feedforward networks. The output of such networks, because of the existing feedback, can either oscillate or converge.

3.5 Learning and Training of ANN

One of the most important qualities of ANN is their ability to learn. Learning is defined as a change of connection weight values that result in the capture of information that can later be recalled. Several algorithms are available for a learning process. Generally, the learning methods can be classified into two categories: supervised and unsupervised learning.

Supervised learning is a process that incorporates an external teacher and (or) global information. The supervised learning algorithms include: error correction learning, reinforcement learning, stochastic learning, etc.

Unsupervised learning, also referred to as self-organization, is a process that incorporates no external teacher and relies upon only local information during the entire learning process. Examples of unsupervised learning include: Hebbian learning, principal component learning, differential Hebbian learning, min-max learning and competitive learning.

Most learning techniques utilizes off-line learning. When the entire pattern set is used to condition the connections prior to the use of the network, it is called off-line learning. For example, the backpropagation training algorithm is used to adjust connections in multilayer feedforward ANN, but it requires thousands of cycles through all the pattern pairs until the desired performance of the network is achieved. Once the network is performing adequately, the weight is stored and the resulting network is used in recall mode thereafter. Off-line learning systems have the inherent requirement that all the patterns have to be resident for training.

Not all networks perform off-line learning. Some networks can add new information "on the fly" nondestructively. If a new pattern needs to be incorporated into the network's connections, it can be done immediately without any loss of prior stored information. The advantage of off-line learning networks is that they usually provide superior solutions in difficult problems such as nonlinear classification, but on-line

learning allows ANN to learn during the system operation. In this Section only, the supervised learning algorithms based on error correction for feedforward ANN will be described. This is because in drive control systems mostly the feedforward ANN are applied.

3.5.1 Learning algorithms for feedforward Neural Networks

The back-propagation is a generalization of the Least Mean Squares (LMS) algorithm. In this algorithm, an error function is defined and equal to the mean square difference between the desired output and the actual output of the feedforward ANN. In order to minimize this error function, the backpropagation algorithm uses a gradient search technique.

3.5.1.1 The Widrow-Hoff (Standard delta) learning rule

Learning rule for one linear neuron

Let us consider the simplest case of ANN. It means, that the Neural Network consists of one linear neuron with N inputs. We will study supervised learning process of this network. So, it is convenient to introduce so-called teaching sequence. We can define this sequence as follows:

$$T = \left\{\left\{X^{(1)}, z^{(1)}\right\}, \left\{X^{(2)}, z^{(2)}\right\}, \ldots\ldots \left\{X^{(P)}, z^{(P)}\right\}\right\} \tag{3.8}$$

where each element $\{X^{(j)}, z^{(j)}\}$ consists of input vector X in the j^{th} step of learning process, and appropriate desired output signal z.

In order to show the learning algorithm, we define the error function as:

$$Q = \frac{1}{2}\sum_{j=1}^{P}\left(z^{(j)} - y^{(j)}\right)^2 \quad z^j and\ y^j\ are\ desired\ and\ estimated\ outputs \tag{3.9}$$

We can rewrite this equation in the following form:

$$Q = \sum_{j=1}^{P} Q^{(j)} \tag{3.10}$$

Where:

$$Q^{(j)} = \frac{1}{2}\left(z^{(j)} - y^{(j)}\right)^2 \tag{3.11}$$

Since Q is a function of W, the minimum of Q can be found by using the gradient descent method:

$$\Delta w_i = -\eta \frac{\partial Q}{\partial W_i} \tag{3.12}$$

Where η is proportionality constant called learning rate.

For the j^{th} step of learning process we can obtain:

$$w_i^{(j+1)} - w_i^{(j)} = \Delta w_i = -\eta \frac{\partial Q}{\partial W_i} \tag{3.13}$$

And using chain rule:

$$\frac{\partial Q^{(j)}}{\partial W_i} = \frac{\partial Q^{(j)}}{\partial y^{(j)}} \frac{\partial y^{(j)}}{\partial w_i} \tag{3.14}$$

The first part shows the error changes in the j^{th} step of the learning process with the output of the neuron and the second parts how much changing w_i changes that output. From (Eq. 3.12) is:

$$\frac{\partial Q^{(j)}}{\partial y^{(j)}} = -\left(z^{(j)} - y^{(j)}\right) = -\delta^{(j)} \tag{3.15}$$

Since the linear network output is defined as:

$$y^{(j)} = \sum_{k=1}^{N} w_k^{(j)} . x_k^{(j)} \tag{3.16}$$

Then:

$$\frac{\partial y^{(j)}}{\partial w_i} = x^{(j)} \tag{3.17}$$

Substituting (Eq.3.15) and (Eq. 3.17) back into (Eq. 3.14) we obtain:

$$-\frac{\partial Q^{(j)}}{\partial W_i} = \delta^{(j)} x^{(j)} \tag{3.18}$$

Thus, the rule for changing weights (Eq. 3.12) is given by:

$$\Delta w^{(j)} = \eta \delta^{(j)} x^{(j)} \tag{3.19}$$

Or in vector form:

$$\Delta \boldsymbol{W}^{(j)} = \eta \delta^{(j)} \boldsymbol{X}^{(j)} \tag{3.20}$$

Finally, the algorithm for new values of the weight vector W can be written as:

$$\boldsymbol{W}^{(j+1)} = \boldsymbol{W}^{(j)} + \eta \delta^{(j)} \boldsymbol{X}^{(j)} \tag{3.21}$$

The delta rule is the basis for most applied learning algorithms. The standard delta rule essentially implements gradient descent in a sum-squared error for linear functions. In this case, without hidden layers, the error surface is shaped like a bowl with only one minimum, so that the gradient descent is guaranteed to find the best set of weights with hidden layers, however, it is not so obvious how to compute the derivatives, and the error surface is not concave upward, so there is the danger of getting stuck in local minima.

3.6 The Back-Propagation Training Algorithm

Backpropagation is the generalization of the Widrow-Hoff learning rule to multiple-layer networks and nonlinear differentiable transfer functions. Input vectors and the corresponding target vectors are used to train a network until it can approximate a function, associate input vectors with specific output vectors, or classify input vectors in an appropriate way as defined by you. Networks with biases, a sigmoid layer, and a linear output layer are capable of approximating any function with a finite number of discontinuities.

Standard backpropagation is a gradient descent algorithm, as is the Widrow-Hoff learning rule, in which the network weights are moved along the negative of the gradient of the performance function. The term backpropagation refers to the manner in which the gradient is computed for nonlinear multilayer networks.

The ANN has analogy with biological neural network, as mentioned before. Like a biological network, where the memory or intelligence is contributed in a distributed manner by the synaptic junctions of neurons, the ANN synaptic weights contribute similar distributed intelligence. This intelligence permits the basic input–output mapping or pattern recognition property of NN. This is also defined as associative memory by which

when one signal pattern is impressed at the input, the corresponding pattern is generated at the output. This pattern generation or pattern recognition is possible by adequate training of the NN.

Properly trained backpropagation networks tend to give reasonable answers when presented with inputs that they have never seen. Typically, a new input leads to an output similar to the correct output for input vectors used in training that are similar to the new input being presented. This generalization property makes it possible to train a network on a representative set of input/target pairs and get good results without training the network on all possible input/output pairs [15].

3.7 Speed Estimation Mechanisms for Induction Motor Drives

The typical IRFO induction motor drive requires the use of an accurate shaft encoder for correct operation. The use of this encoder implies additional electronics, extra wiring, extra space and careful mounting which detracts from the inherent robustness of cage induction motors. Moreover, at low powers (2 to 5 kW) the cost of the sensor is about the same as the motor. Even at 50 kW, it can still be between 20 to 30% of the machine cost [16]. Therefore, there has been great interest in the research community in developing a high-performance induction motor drive that does not require a speed or position transducer for its operation.

This section discusses issues related to speed estimation of three phase Induction motor for its closed loop operation as speed-controlled drive. A drive in which the speed/position sensor is absent is usually called 'sensorless' drive, where 'sensorless' symbolizes absence of the shaft sensor. However, the sensors required for stator current measurement (and, in many cases, stator voltage measurement as well) remain to be present so that the term 'sensorless' is somewhat misleading. Sensorless vector control of an induction machine has attracted wide attention in recent years. Many attempts have been made in the past to extract the speed signal of the induction machine from measured stator currents and voltages. The first attempts have been restricted to techniques which are only valid in the steady-state and can only be used in low cost drive applications, not requiring high dynamic performance.

More sophisticated techniques are required for high performance applications in vector-controlled drives. In a sensorless drive, speed information and control should be provided with an accuracy of 0.5% [17] or better, from zero to the highest speed, for all operating conditions and independent of saturation levels and parameter variations. In order to achieve good performance of sensorless vector control, different speed estimation schemes have been proposed, so that a variety of speed estimators exist nowadays. In general, all the existing speed estimation algorithms belong to one of the following three groups:

1. Speed estimation from the stator current spectrum;

2. Speed estimation based on the application of an induction machine model;

3. Speed estimation by means of artificial intelligence techniques (artificial neural networks). Various schemes of sensorless speed control techniques are shown below in Fig (3.7). Discussing all the mechanisms for speed estimation is beyond this thesis. One of the contributions of this thesis is designing an ANN to estimate the speed of Induction motor drive with high accuracy.

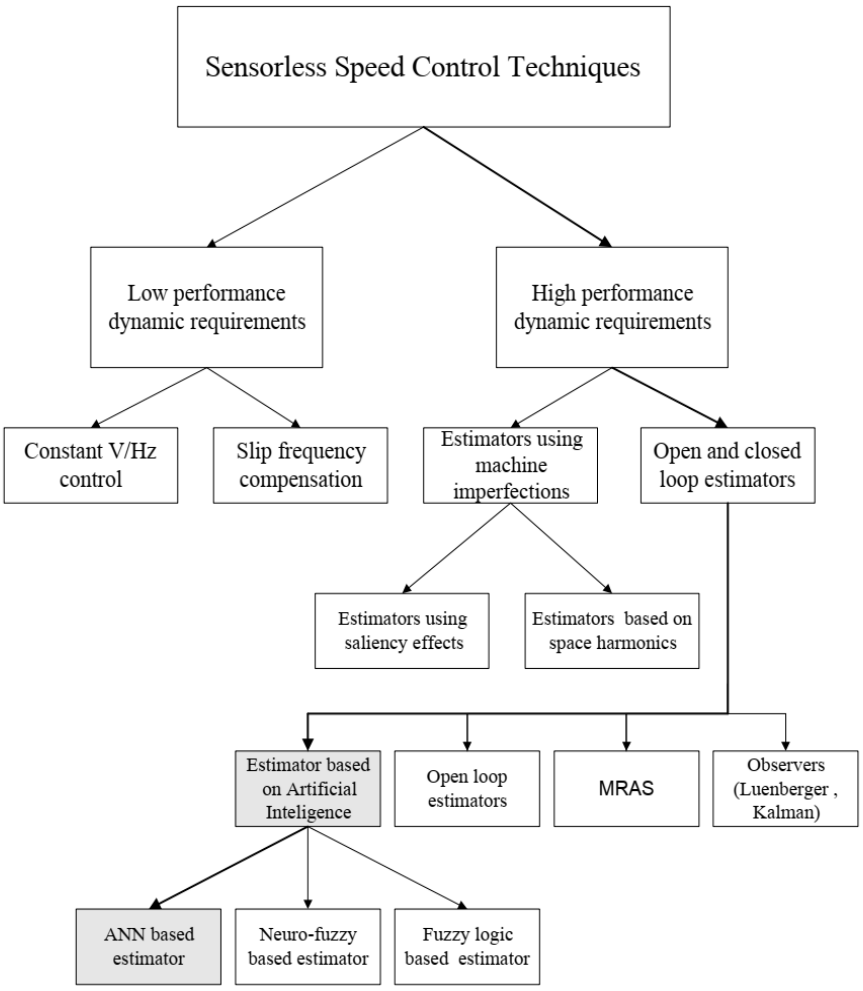


Figure 3.7: Schemes of speed Estimation for sensorless operation [17]

3.8 Speed Estimation using Multi-layer Neural Networks

Artificial neural networks (NN) are well suited for application in IM control and estimation, because of their known advantages as: ability to approximate arbitrary nonlinear functions to any desired degree of accuracy, learning and generalization, fast parallel computation, robustness to input harmonic ripples and fault tolerance. These aspects are important in the case of nonlinear systems, like converter-fed drives, where linear control theory cannot be directly applied. Additionally, high efficiency power electronic converters used for IM supply operate in switch mode, which results in very noisy signals. For these reasons the NN are attractive for signal processing and control of IM drives.

Neural networks (NN) offer an alternative way to handling this problem. Because of their ability to mapping different kinds of nonlinearities and generalization properties, NN can be applied in the situation when system is not exactly identified. So, its parameters could be not exactly known, but should be learned.

For the speed estimation of the induction motor, there is necessary to find out the proper neural network architecture. Up to now there does not exist any widely used rules to choose the proper network architecture, neither the number of neural units. There are just generally accepted rules as a simplest network with highest accuracy. First it is necessary to design right structure of the artificial neural network and it is also important to determine such inputs to ANN, which are available in structure of vector control and from which is able to estimate a rotor speed of the induction motor. A recommended method for determination of ANN structure does not exist, so the final ANN was designed by means of trial and error. The main goal was to find the simplest neural network with good accuracy of speed estimation. This is the key for industry use of ANNs. The flow chart that describes the procedure for backpropagation algorithm to estimate speed is shown in figure below.

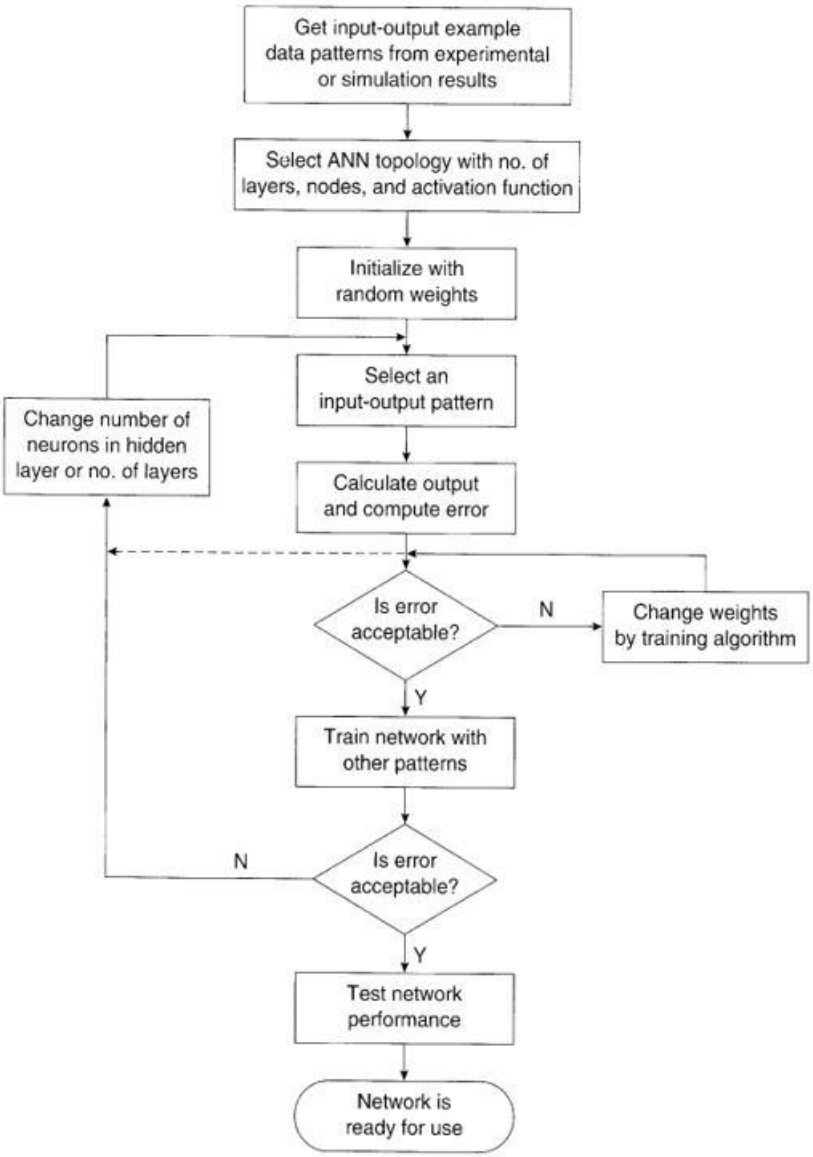


Figure 3.8: Flow chart for backpropagation training of Feed forward Neural network

From the voltage equation of rotor side in stationary reference (Eq 2.36- 2.37) frame we can express the speed with voltage and current as shown below.

$$\omega_m = -[(L_m - \frac{L_r L_s}{L_m})pi_{\alpha s} + \frac{L_r}{L_m}(v_{\alpha s} - r_s i_{\alpha s}) + \frac{R_r}{L_m}(\int (v_{\alpha s} - R_s i_{\alpha s})dt) - R_r \frac{L_s}{L_m} i_{\alpha s})]/\psi_{\beta r} \quad (3.22)$$

$$\omega_m = \frac{-[(L_m - \frac{L_r L_s}{L_m})pi_{\alpha s} + \frac{L_r}{L_m}(v_{\alpha s} - R_s i_{\alpha s}) + \frac{R_r}{L_m}(\int (v_{\alpha s} - R_s i_{\alpha s})dt) - R_r \frac{L_s}{L_m} i_{\alpha s})]}{(\boldsymbol{L}_m - \frac{L_r L_s}{L_m})\, i_{\beta s} + \frac{L_r}{L_m}\,(\int (v_{\beta s} - R_s i_{\beta s})dt} \quad (3.23)$$

$$\omega_m = \frac{L_r(v_{\alpha s} - R_s i_{\alpha s}) - p(L_s L_r - L_m^2) i_{\alpha s}}{(L_m 2 - L_s L_r)\, i_{\beta s} + L_r \int (v_{\beta s} - R_s i_{\beta s})dt} + \frac{R_r \int (v_{\alpha s} - R_s i_{\alpha s})dt - R_r L_s i_{\alpha s}}{(L_m 2 - L_s L_r)\, i_{\beta s} + L_r \int (v_{\beta s} - R_s i_{\beta s})dt} \quad (3.24)$$

The above equation reflects a kind of nonlinear mapping relation between the rotor speed and the stator voltage and current which can be described as:

$$\omega_m(k) = f(v_{\alpha s}, v_{\beta s}, i_{\alpha s}, i_{\beta s}) \tag{3.25}$$

This nonlinear relation can be realized by using artificial neural network.

We have selected a feed forward neural network structure with three layers. The first input layer consists of 8 inputs, such as current and delayed samples of stator voltage and current in the stationary reference frame.

$$v_{\alpha s}(k), v_{\alpha s}(k-1), v_{\beta s}(k), v_{\beta s}(k-1), i_{\alpha s}(k), i_{\alpha s}(k-1), i_{\beta s}(k), i_{\beta s}(k-1).$$

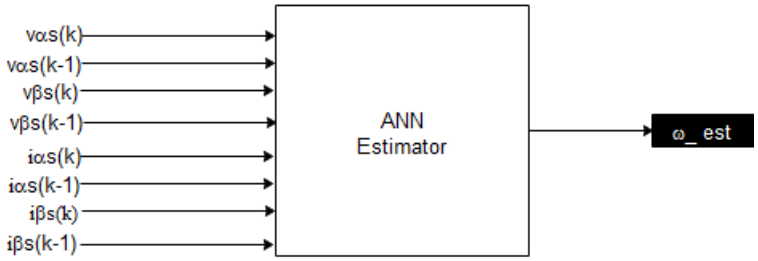


Figure 3.9: Speed estimation block using ANN

Sigmoidal tangent activation function utilized in the hidden layer, since both the inputs and outputs have bipolar values and linear activation function in the output layer since the output range might be greater than 1. During training Levenberg Marquardt algorithm is used. Number of epochs, learning rate and momentum updates are shown in Appendix as MATLAB® script. Training and validation data are obtained from simulation of conventional PI controlled drive simulation in MATLAB®/SIMULINK environment. Results during different operating conditions are discussed in chapter five.

One of the problems that occur during Neural Network training is called overfitting. The error on the training set is driven to a very small value, but when new data is presented to the network the error is large. The network has memorized the training examples but it has not learned to generalize to new situations. The default method for improving generalization is early stopping. This method is automatically provided for all of supervised network creation functions including the back-propagation network creation function such as feed forward net. The available data is divided into three subsets. The first

subset is training set which is used for computing the gradient and updating the network weight and biases. The second subset is the validation set. The error on the validation set is monitored during the training process.

The validation error normally decreases during the initial phase of training, as does the training set error. However, when the network begins to overfit the data, the error on the validation set typically begins to rise. When the validation error increases for a specified number of iterations (`net.trainParam.max_fail`), the training is stopped, and the weights and biases at the minimum of the validation error are returned.

The test set error is not used during training, but it is used to compare different models. It is also useful to plot the test set error during the training process. If the error in the test set reaches a minimum at a significantly different iteration number than the validation set error, this might indicate a poor division of the data set. To raise the generalization ability of the designed network we have collected simulation data during different operation scenarios like parameter and reference speed variation of the drive.

CHAPTER 4 NEURO- FUZZY SPEED CONTROLLER DESIGN FOR IM DRIVE

4.1 Introduction

In the modern-day world, the AC motors play a vital role in the industrial sector especially in the field of electric drives. Without proper controlling of the speed, it is virtually impossible to achieve the desired task for any industrial application. AC motors, particularly the squirrel-cage induction motors (SCIM), enjoy several inherent advantages like simplicity, reliability, low cost and virtually maintenance-free electrical drives. However, for high dynamic performance industrial applications, their control remains a challenging problem because they exhibit significant non-linearities and many of the parameters, mainly the rotor resistance, vary with the operating conditions. Different control techniques to control the speed of the IMs have been developed by various researchers across the world. The classical or the conventional control, PI/ PID controllers are widely used in high performance drives, which had lot of advantages as well as dis-advantages. In recent years Intelligent controllers are becoming the part of drive control due to the development of fast Digital signal processors, which might yield excellent results. Fuzzy logic and Neural networks are getting wide acceptance in drive control due to their inherent advantages. The neural network has the inherent advantage of being able to adapt itself and also in its learning capabilities.

Similarly, the salient feature that is associated with the fuzzy logic is the distinct ability to take into account the prevailing uncertainty and imprecision of real systems with the help of the fuzzy if-then rules. In order to exploit the advantage of the self-adaptability and learning capability of the neural network and the capability of the fuzzy system to take into account of the prevailing uncertainty and imprecision of real systems with the help of the fuzzy if-then rules, an integrated control approach comprising of both the fuzzy logic and the neural network has been considered. This hybrid system is called the Adaptive network based fuzzy inference system (ANFIS).

4.2 Adaptive Neuro Fuzzy Inference Systems

The fuzzy based controllers develop a control signal which yields on the firing of the rule base, which is written on the previous experiences & are then fired to get the final output. As a result of which, the outcome of the controller is also random & optimal results may not be obtained. To avoid this, selection of the proper rule base depending upon the situation can be achieved by the use of an ANFIS controller, which becomes an integrated method of approach for the control purposes & yields excellent results. In the designed ANFIS scheme for the speed control of IM, Neural network techniques are used to select a proper rule base, which is achieved using the hybrid algorithm. This integrated approach improves the system performance, cost-effectiveness, efficiency, dynamism, reliability of the designed controller.

The basic structure of Mamdani fuzzy inference system is a model that maps input characteristics to input membership functions, input membership functions to rules, rules to a set of output characteristics, output characteristics to output membership functions, and the output membership functions to a single-valued output or a decision associated with the output. Such a system uses fixed membership functions that are chosen arbitrarily and rule structure that is essentially predetermined by the user's interpretation of the characteristics of the variables in the model.[18]

Neuro fuzzy control combines the mapping and learning ability of an artificial neural network with the linguistic and fuzzy inference advantages of fuzzy logic that have the ability to self-modify their membership function to achieve a desired performance. Thus, a neuro fuzzy controller has the potential to outperform conventional ANN or fuzzy logic controller. The adaptive network based fuzzy inference system (ANFIS) controller employs a Tagaki Sugno Kang (TSK) fuzzy inference system [19].

4.2.1 ANFIS Architecture

The basic ANFIS architecture is shown in figure 4.2. Square nodes are adaptive in the ANFIS structure denote parameter sets of the membership functions of the TSK fuzzy system. Circular nodes are static (non-modifiable) and perform operations such as product or max/min calculations. A hybrid learning rule is used to accelerate parameter adaptation. This uses sequential least squares in the forward pass to identify consequent parameters, and backward pass to establish the premise parameters [19].

Assuming that the fuzzy inference system under consideration has two inputs *x* and *y* and one output *z*. For a first-order Sugeno fuzzy model with two if-then rules:

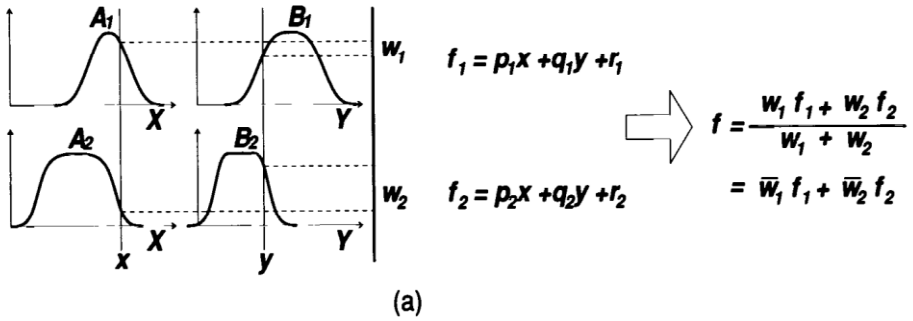


Figure 4. 1: A two-input first-order Sugeno fuzzy model with two rules [19]

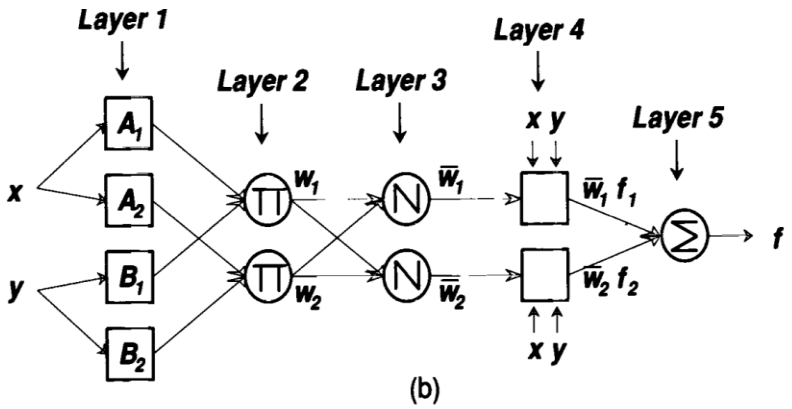


Figure 4.2: Equivalent ANFIS structure [19]

For simplicity, it is assumed that the fuzzy inference system under consideration has two inputs and one output. The rule base contains the fuzzy if-then rules of Takagi and Sugeno's type as follows: *If x is A and y is B then z is* $f(x, y)$ where A and B are the fuzzy sets in the antecedents and $z = f(x, y)$ is a crisp function in the consequent. Usually $f(x, y)$ is a polynomial for the input variables x and y. But it can also be any other function that can approximately describe the output of the system within the fuzzy region as specified by the antecedent. When $f(x, y)$ is a constant, a zero order Sugeno fuzzy model is formed which may be considered to be a special case of Mamdani fuzzy inference system [20] where each rule consequent is specified by a fuzzy singleton. If $f(x, y)$ is taken to be a first order polynomial a first order Sugeno fuzzy model is formed. For a first order two rule Sugeno fuzzy inference system, the two rules may be stated as:

Rule 1: if x is A_1 and y is B_1, then $f_1 = p_1 x + q_1 y + r_1$

Rule 2: If x is A_2 and y is B_2, then $f_2 = p_2x + q_2y + r_2$

Here type-3 fuzzy inference system proposed by Takagi and Sugeno is used. In this inference system the output of each rule is a linear combination of the input variables added by a constant term. The final output is the weighted average of each rule's output. The individual layers of this ANFIS structure are described below:

Input Node (Layer 1): Nodes in this layer contains membership functions. Every node i in this layer is square and adaptive with a node function:

$$O_1^i = \mu_{Ai}(x), for\ i = 1,2\ or \tag{4.1}$$

$$O_1^i = \mu_{Bi-2}(y), for\ i = 3,4 \tag{4.2}$$

where, x is the input to node i, A_i is the linguistic variable associated with this node function and μ_{Ai} is the membership function of A_i.

Similarly, y is the input to node i, B_{i-2} is the linguistic variable associated with this node function and μ_{Bi-2} is the membership function of B_{i-2}.

Here the membership function can be any appropriate parameterized membership function, such as the generalized bell function:

$$\mu_{Ai}(x) = \frac{1}{1 + \left|\left(\frac{x - c_i}{a_i}\right)^2\right|^{b_i}} \tag{4.3}$$

Where x is the input and $\{a_i, b_i, c_i\}$ are parameter sets. Parameters in this layer are called premise parameters.

Rule Nodes (Layer 2): Every node in this layer is fixed and circle node labeled Π, whose output is the product of all incoming signals:

$$O_i^2 = w_i = \mu_{Ai}(x) * \mu_{Bi}(y), i = 1,2 \tag{4.4}$$

Each node's output represents the firing strength of a rule. This layer chooses the minimum value of two input weights. Any other T-norm operators that perform fuzzy AND (e.g. min) can be used as the node function in this layer.

Layer 3: Every node in this layer is a fixed node. Each i^{th} node calculates the ratio of the i^{th} rule's firing strength to the sum of firing strengths of all the rules. The output from the i^{th} node is the normalized firing strength given by,

$$O_i^3 = \bar{w}_i = \frac{w_i}{w_1 + w_2}, \qquad i = 1,2 \tag{4.5}$$

For convenience, outputs of this layer are called normalized firing strengths.

Layer 4: Every node i in this layer is an adaptive node with a node function

$$O_i^4 = \bar{w}_i f_i = \bar{w}_i(p_i x + q_i y + r_i) \tag{4.6}$$

Where w_i is a normalized firing strength from layer 3 and $\{p_i, q_i, r_i\}$ are the parameter sets of this node. Parameters in this layer are referred to as consequent parameters.

Layer 5: The single node in this layer is fixed node labeled Σ, which computes the overall output as the summation of all incoming signals:

$$overall\ output = O_i^5 = \sum_i \bar{w}_i f_i = \frac{\sum_i w_i f_i}{\sum_i w_i} \tag{4.7}$$

4.2.2 Hybrid Learning Algorithm

In the ANFIS structure, it is observed that given the values of premise parameters (fixed premise parameters), the final output can be expressed as a linear combination of the consequent parameters.

In symbols, the output f can be written as:

$$f = \frac{w_1}{w_1 + w_2} f_1 + \frac{w_2}{w_1 + w_2} f_2 \tag{4.8}$$

$$= \bar{w}_1(p_1 x + q_1 y + r_1) + \bar{w}_2(p_2 x + q_2 y + r_2) \tag{4.9}$$

$$= (\bar{w}_1 x)p_1 + (\bar{w}_1 y)q_1 + (\bar{w}_1)r_1 + (\bar{w}_2 x)p_2 + (\bar{w}_2 y)q_2 + (\bar{w}_2)r_2 \tag{4.10}$$

where f is linear in the consequent parameters $(p_1, q_1, r_1, p_2, q_2, r_2)$.

In the forward pass of the learning algorithm, consequent parameters are identified by the least squares estimate. In the backward pass, the error signals, which are the derivatives of the squared error with respect to each node output, propagate backward from the output layer to the input layer. In this backward pass, the premise parameters are updated by the gradient descent algorithm.

ANFIS is functionally equivalent to a Takagi Sugeno fuzzy inference system. By using stipulated input output training data pairs, ANFIS tunes the membership functions and other associated parameters by backpropagation gradient descent and least square type method. Such methodologies make the ANFIS modeling more systematic and less reliant on expert knowledge. Every node in layer one and layer four have an adaptive node with updated parameters in order to achieve a desired input output mapping. These parameters are updated according to given training data and gradient based learning procedure, while, the nodes in layers two and three are fixed with no parameters.

In learning process of ANFIS, the premise parameters (on layer one) and consequent parameters (on layer four) should be tuned to optimum mathematical relation between inputs and outputs. ANFIS uses a two-pass learning cycle. In the forward pass the algorithm uses least squares method to identify the consequent parameters. In the backward pass the errors are propagated backward and the premise parameters are updated by gradient descent. An initial fuzzy inference system is first created and improved through the learning process. ANFIS tunes the initial fuzzy inference system by a hybrid algorithm which, combining the gradient descent back propagation and least square optimization techniques. At each epoch the error is calculated. The stopping criteria might be when the maximum epoch reached or the minimum error goal achieved. i.e. the same as neural network [21].

4.3 Design of Adaptive Neuro Fuzzy Inference Systems as Speed controller

In this section, a brief review of the ANFIS concepts to control speed of induction motor in sensorless vector-controlled drive is presented. The concept of neural networks started in the late-1800s as an effort to describe how the human mind performed in the olden days. As years rolled by, these neural networks started playing a very important role in the various engineering applications. Neural networks have been applied successfully to speech recognition, image analysis and adaptive control, in order to construct software agents or autonomous robots & in the control of machines.

ANNs are a family of intelligent algorithms which can be used for time series prediction, classification, and control and identification purposes. Neural networks have an ability to train with various parameter of induction motor. As a non-linear function, they can be used

for identifying the extremely nonlinear system parameters with high accuracy. Recently, the use of neural networks, to identify and control nonlinear dynamic systems has been proposed because they can approximate a wide range of non-linear functions to any desired degree of accuracy.

Moreover, they have the advantages of extremely fast parallel computation and fault tolerance characteristics. Also, there have been some investigations into the application of NNs to power electronics and ac drives, including speed estimation. This technique gives a fairly good estimate of the speed and is robust to parameter variation. However, the neural network speed estimator should be trained sufficiently with various patterns to get good performance.

Fuzzy logic is one of the successful applications in the control engineering field which can be used to control various parameters of the real time systems. This logic combined with neural networks yields very significant results. Neural networks can learn from data. However, understanding the knowledge learned by neural networks has been difficult. To be more specific, it is usually difficult to develop an insight about the meaning associated with each neuron and each weight. In contrast, fuzzy rule-based models are easy to be understood because it uses linguistic terms and the structure of IF-THEN rules. Unlike neural networks, however, fuzzy logic by itself cannot learn. The learning and identification of fuzzy logic systems need to adopt techniques from other areas, such as statistics, system identification. Since neural networks can learn, it is natural to merge these two techniques.

4.3.1 Controller Design

A controller is a device which controls each and every operation in the system making decisions. From the control system point of view, it is bringing stability to the system when there is a disturbance, thus safeguarding the equipment from further damages. It may be hardware based controller or a software based controller or a combination of both. In this section, the development of the control strategy for control of induction motor speed is presented using the concepts of ANFIS control scheme.

To start the ANFIS learning; first, a training data set that contains the desired input / output data pairs of target systems to be modeled is to be required. The design parameters required for any ANFIS controller are viz., Number of data pairs, Training data set and checking

data sets, Fuzzy inference systems for training, Number of epochs to be chosen to start the training, learning results to be verified after mentioning the step size.

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 as a result; 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 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 [22].

The inputs to the ANFIS controller, i.e., the error & the change in error is modeled using the Eq. (4.11) – (4.12) as:

$$e(k) = \omega_{ref} - \omega_{est} \tag{4.11}$$

$$\Delta e(k) = e(k) - e(k-1) \tag{4.12}$$

where ω_{ref} is the reference speed, ω_{est} is the estimated rotor speed, is the $e(k)$ error and $\Delta e(k)$ is the change in error.

In order to generate adaptive neuro-fuzzy system, conventional PI controller is designed and simulated using MATLAB®/2019a to collect training and checking data for 3.5 seconds. After collecting the input/output data, the data divided in to training and checking data sets randomly. 3.5 million data points are collected and out of these data, 2,450,000 (70% of total data) have been considered as training data set and remaining 1,050,000 (30% of total data) have been considered as checking data. Since 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 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 7 and type of input membership functions used are generalized bell functions shown in Eq. 4.3. Linear membership function type is selected for output, totally 49 rules are generated, and the number of iterations selected is 100. The plot of membership functions for inputs, surface and rule views are shown in appendix C. The performances of 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) using Grid Partitioning algorithm whose membership function parameters are tuned using hybrid learning algorithm method.

The training data set was used to train the ANFIS, whereas the checking data set was used to verify the generalization capability of the trained ANFIS model for the adaptation of learning content. Once the input/output data pairs have been collected, 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 membership functions and also number of epoch numbers). After initializing FIS and specifying the number of epochs the next step was to 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. After loading the training data and checking data sets in Neuro-fuzzy Designer, the following results have been observed. Figure 4.3. shows the training and checking data sets after loading in ANFIS editor toolbox.

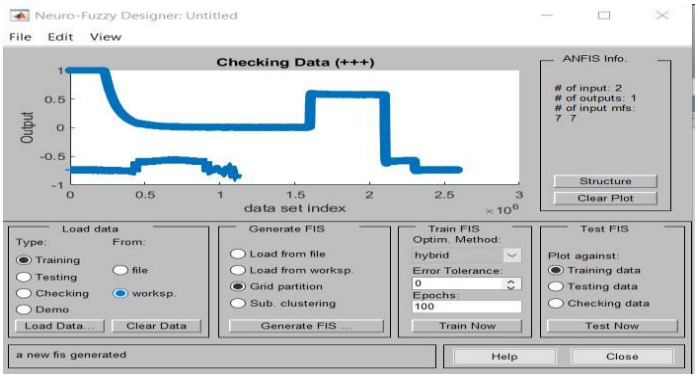


Figure4.3: Training and Checking data sets after loading in ANFIS editor toolbox.

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 corresponding results and Simulink blocks are discussed in chapter five under performance of ANFIS speed controller. Also, the training results shown in Appendix C

Table 4.1: Specifications of the developed adaptive neuro-fuzzy inference system

Parameters	Description/Value	Parameters	Description/ Value
Optimization method	Hybrid	Number of outputs	1
Structure of FIS	Sugeno first order	Output MF type	linear
Or method	Max	No. of rules	49
And method	Min	No. of epochs /iterations	100
Implication method	Prod	No. of training data pairs	2,450,000
Aggregation method	Max	No. of checking data pairs	1,050,000
No. of inputs	2	No. linear parameters	147
No. of input MF	7	No. of nonlinear parameters	42
Type of input MF	Generalized bell	Total No. of parameters	189
No. of output MF	49	No. of nodes	131

CHAPTER 5 SIMULATIONS, RESULTS AND DISCUSSIONS

5.1 Simulink Modeling

MATLAB/Simulink is a software package which is used to model, simulate, and analyze dynamic systems. Simulink has the advantage of being capable of complex dynamic system simulations, graphical environment with visual real time programming and broad selections of toolboxes.

The mathematical equations presented in chapter two are used to model the three-phase induction motor in MATLAB/Simulink R2019a environment. Figure 5.1 shows the complete Simulink model of indirect vector speed control of three phase induction motor. The overall system Simulink block includes different sub functional blocks such as IM physical model block, voltage and current sensors, PI controller blocks, coordinate transformation blocks, IFOC block, speed estimator and SVPWM inverter block. Once the block diagram has been developed in MATLAB/Simulink it can be simulated using any number of different solvers. These solvers can compute the internal state variables of the blocks by solving their respective ordinary differential equations in MATLAB modeling configuration parameters. The solver is significant to decrease the computation time and improve the accuracy of the simulation.

A Simulink model of sensorless speed control of induction motor drive was developed using components from the Power System's Block set. The inverter and asynchronous motor configuration are used from an existing Simulink file. The Indirect Field Oriented Control (IFOC), as well as the speed estimation structure, was implemented using the theory outlined in Chapter 2 and Chapter 3. The motor used in this thesis is three-phase SCIM with a rated power of 3.3kw, rated speed 1450 rpm parameters are given in Appendix A.

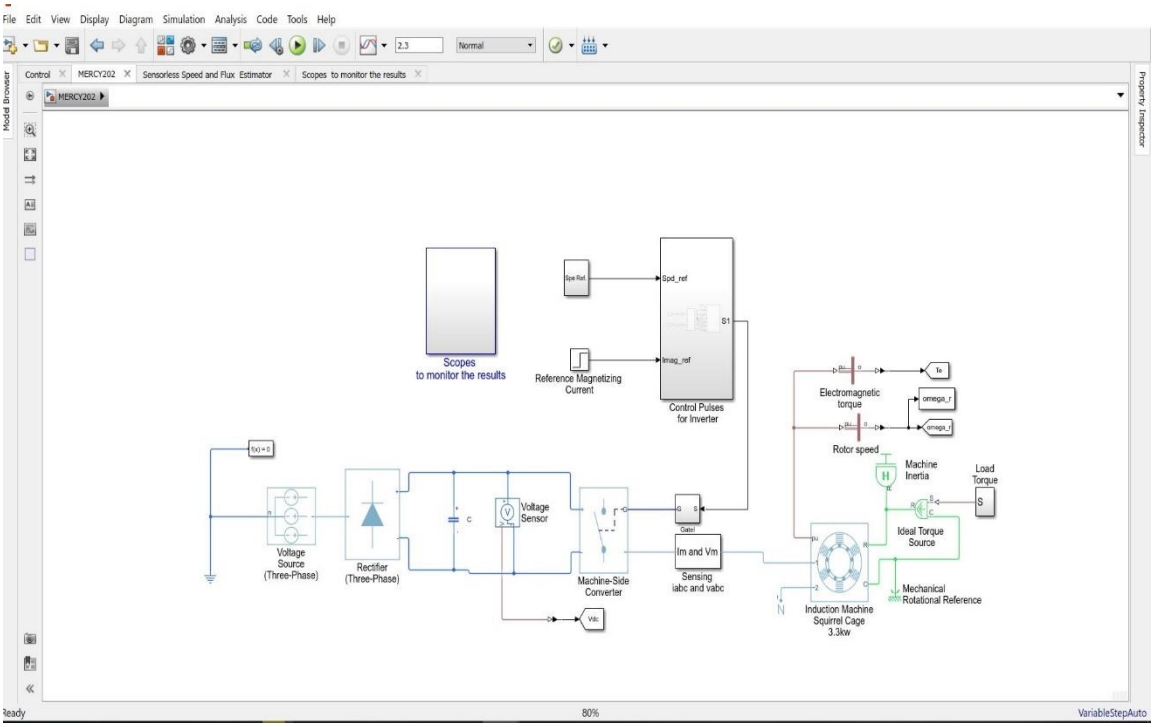


Figure 5.1: Overall block diagram for data generation

5.2 Field Oriented control of IM drive for Data Generation

The controller gains are obtained by following the procedure in section 2.4.2.1

Table 5.1: Controller Parameters for PI Controllers

Speed controller parameters	Proportional gain	Integral Gain
d-axis Current controller parameters	0.8	22.02
q-axis Current controller parameters	0.8	22.02
Speed Controller parameters	70.47	150.424

Simulation for different operating scenarios using PI conventional controller for 3.5 seconds to collect data for speed estimation. The four cases are shown below, rated speed at no load and full load condition, reference speed variation with speed reversal and parameter variation is also included. The performance measures are shown in table 5.2.

- Constant speed and No-load Condition with PI Controller

For the speed control of induction motor, the three-phase stator current which is generated by the three-phase voltage source inverter should be sinusoidal. This three-phase voltage source inverter is controlled by SVPWM blocks for appropriate stator current generation. These three phases current should be the equal magnitude and 120° phase shift with each other for appropriate rotating flux generation as shown in Fig. 5.2.

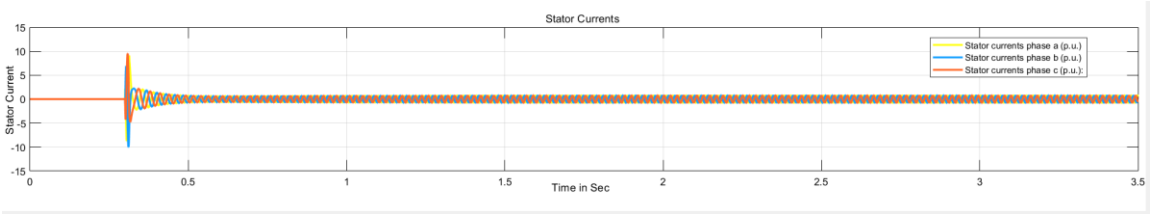
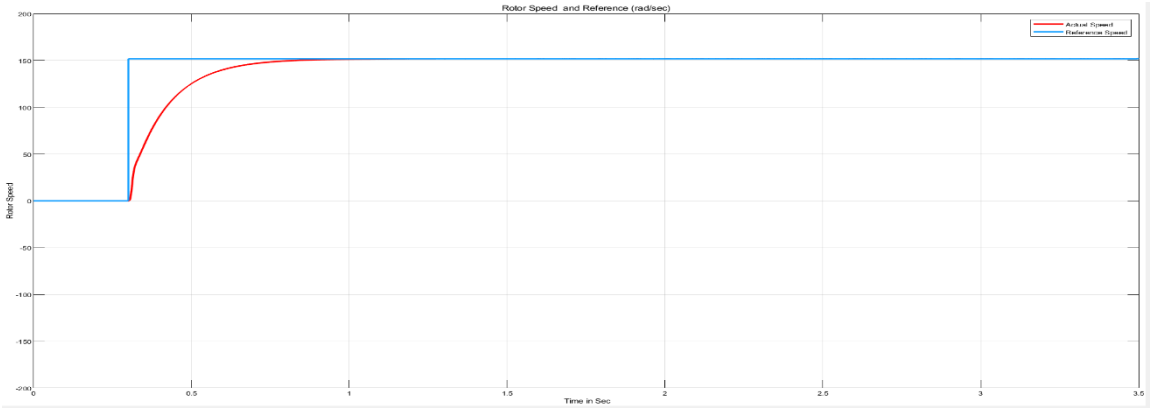
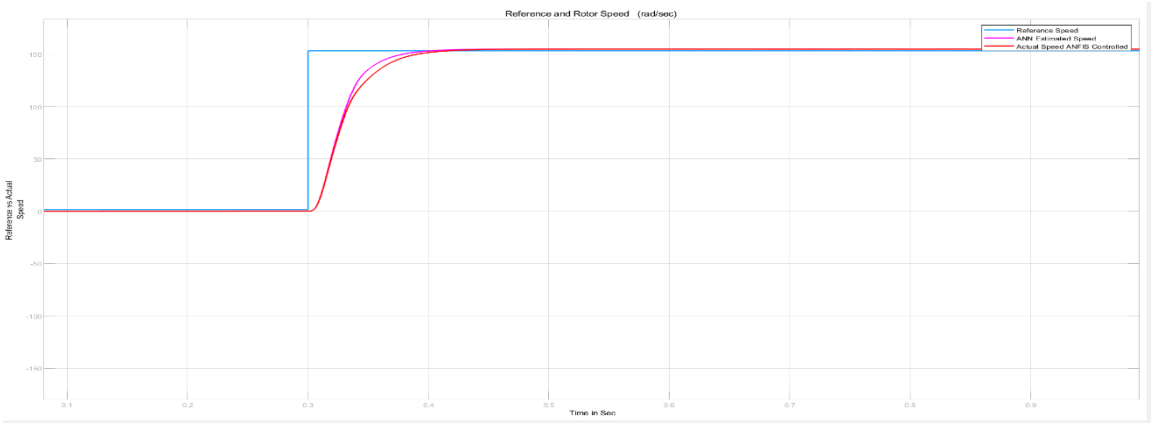


Figure 5.2: Current waveform with rated speed and no load

Rated speed is applied at 0.3 sec as reference speed and due to Inertia of the system and Physical limitation of Inverter



a). The rotor speed response at 151 rad/s at 0.3sec, step input signal with PI



b). The rotor speed response at 151 rad/s at 0.3sec, step input signal with ANN estimator And ANFIS Controller

Figure 5.3: Constant Speed response at zero Load Torque with PI and ANFIS controller

The speed control is on no load condition and full load conditions. The speed error result and performance measures show that the proposed scheme is far better than PI controller.

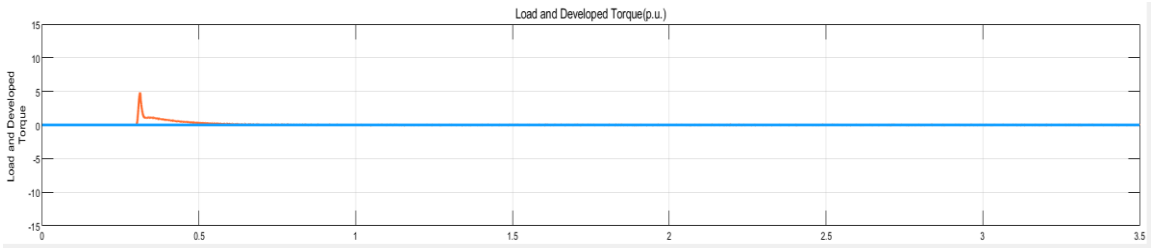


Figure 5.4: Load Torque response at Rated speed

- Response wave shapes, during rated speed and rated Torque

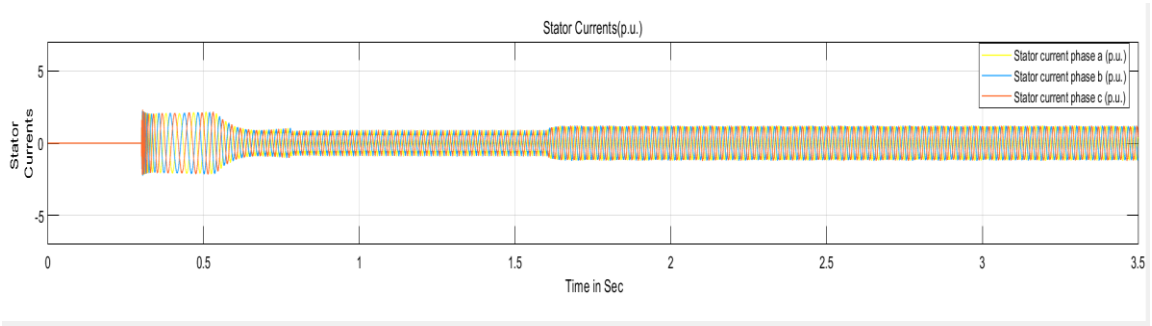


Figure 5.5: Current wave shape during rated speed and full load torque

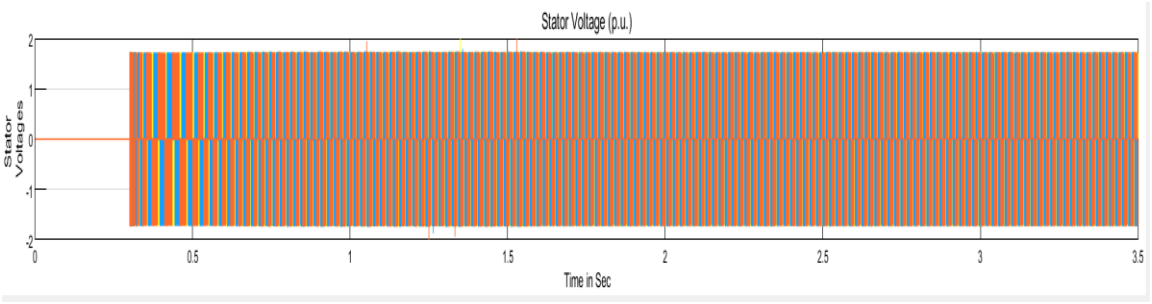
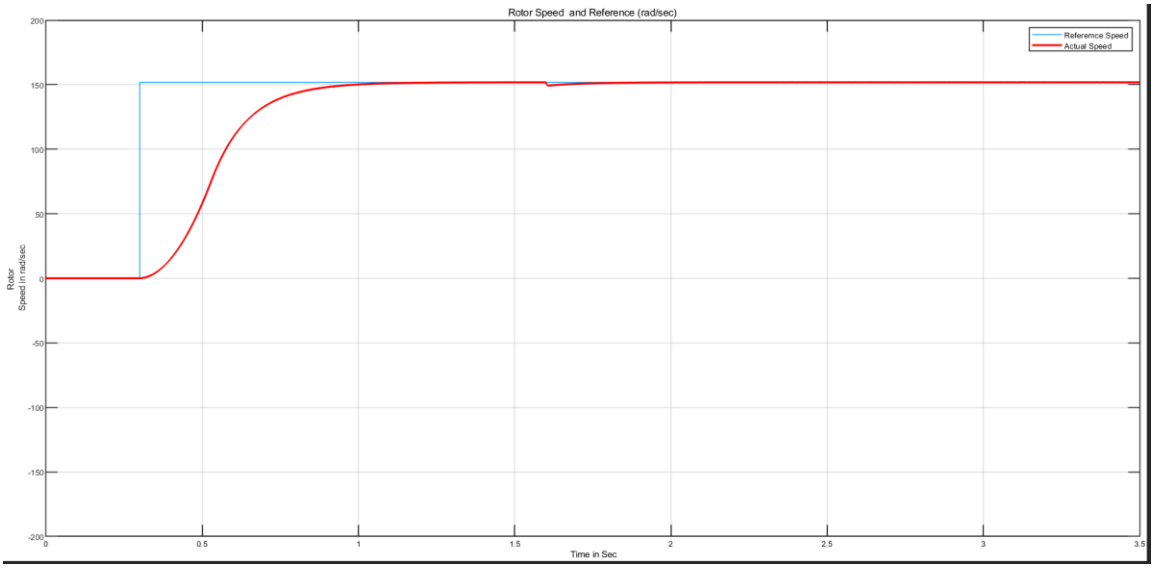
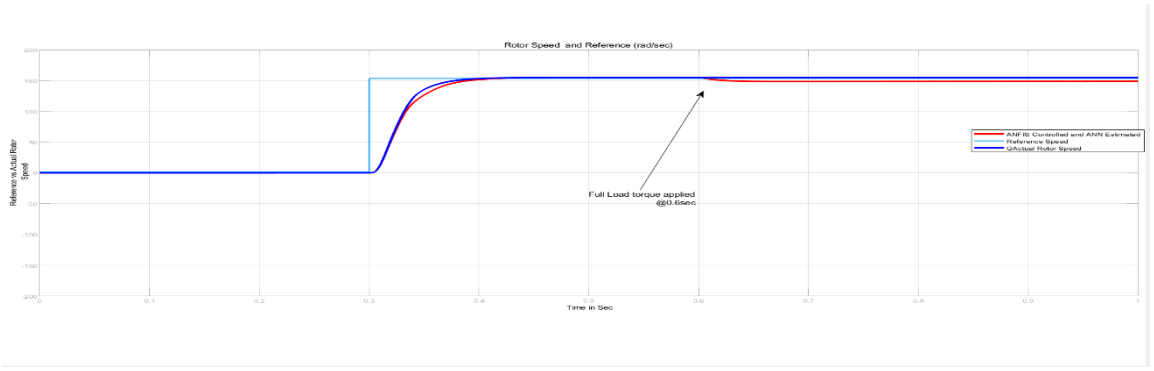


Figure 5.6: Voltage waveform during rated speed and full load torque



a). The rotor speed response at 151 rad/s at 0.3sec, step input signal with PI at Full Load



b). The rotor speed response at 151 rad/s at 0.3sec, step input signal with ANN estimator and ANFIS controller at full load

Figure 5.7: Speed response during Full load applied at Rated Speed operation

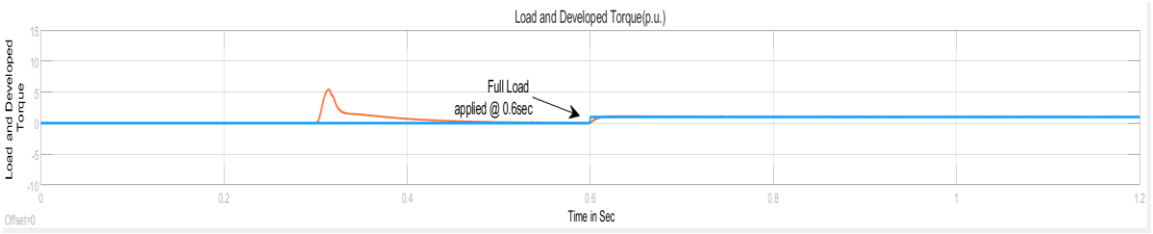


Figure 5.8: Torque response during full load applied

- Response wave shapes, during reference speed step change tracking

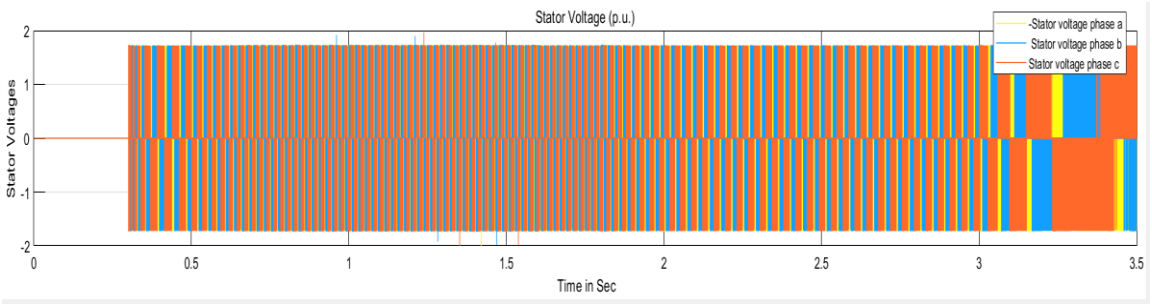


Figure 5.9: Stator Voltage waveform during reference step changes

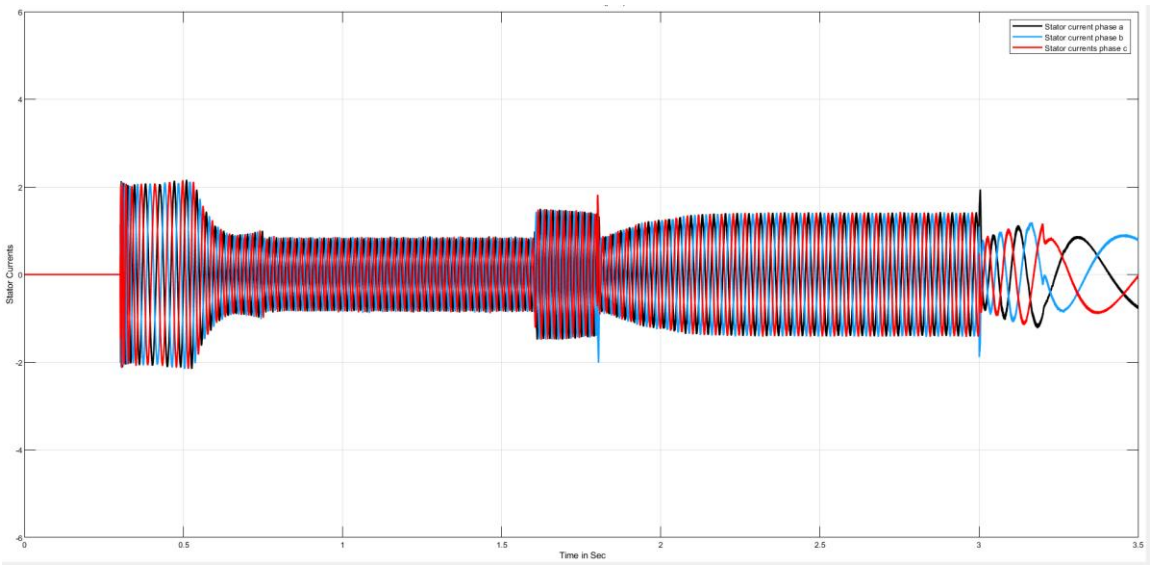


Figure 5.10: Stator Current wave shape during reference speed step changes

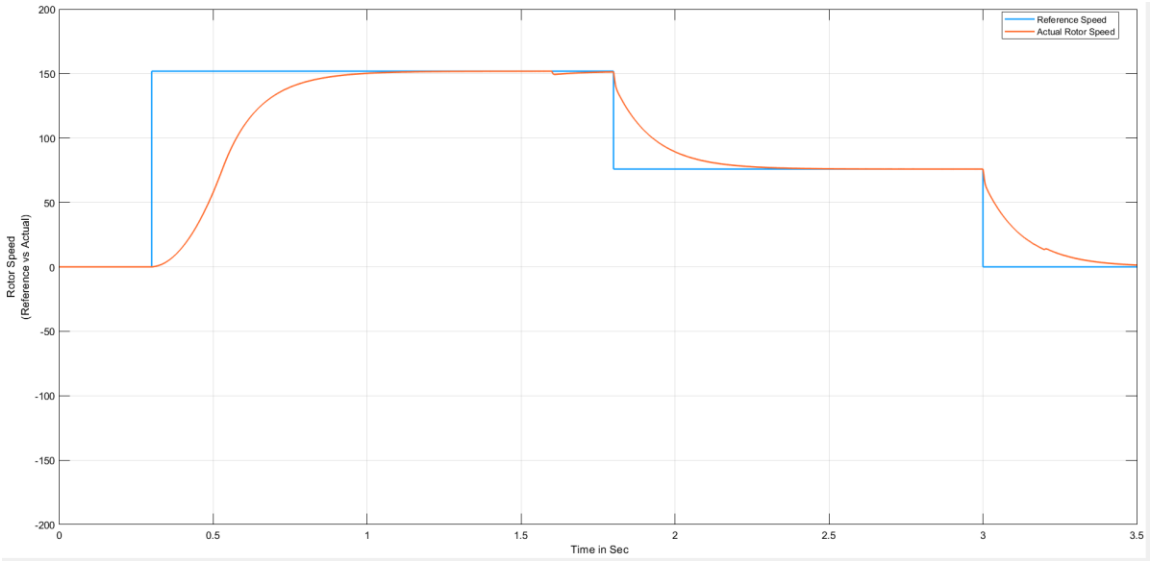


Figure 5.11: Speed response with PI Controller during step change in reference speed

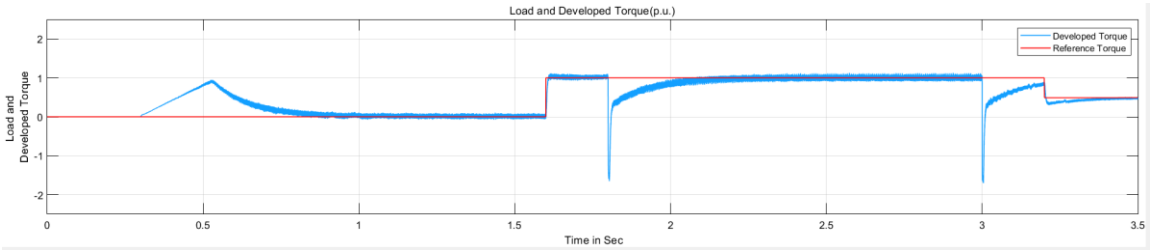


Figure 5.12: Torque response with PI Controller during reference speed step change

- Response wave shapes, during reference speed step change and reversal

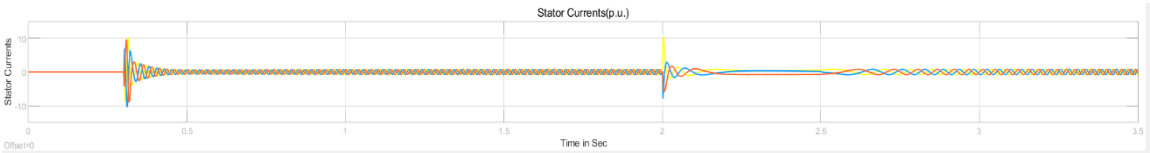


Figure 5.13: Current waveform during step speed change and reversal

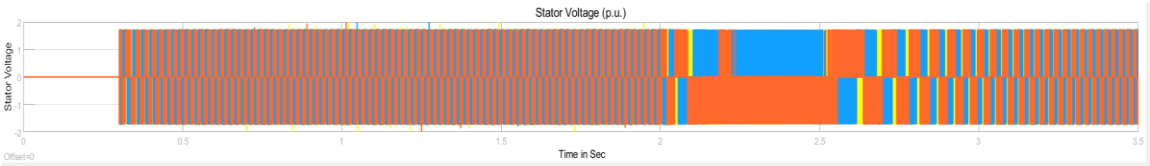


Figure 5.14: Voltage waveform during step speed change and reversal

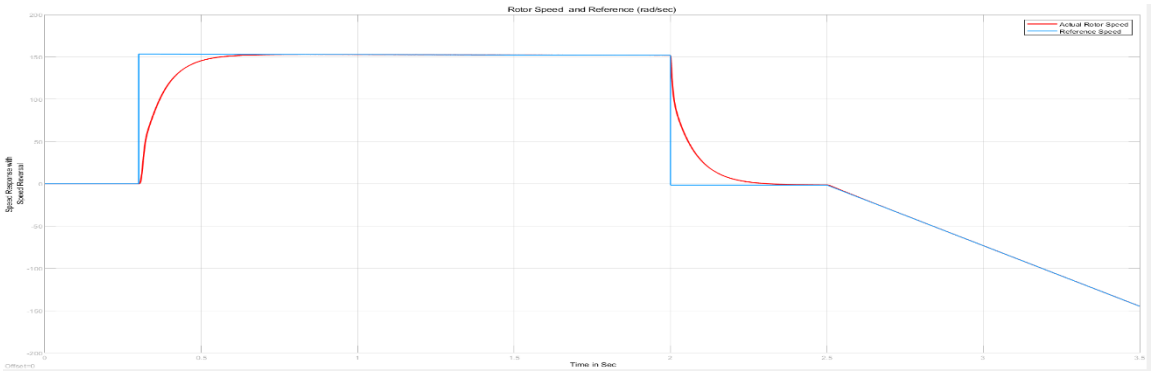


Figure 5.15: Speed tracking during step and speed reversal

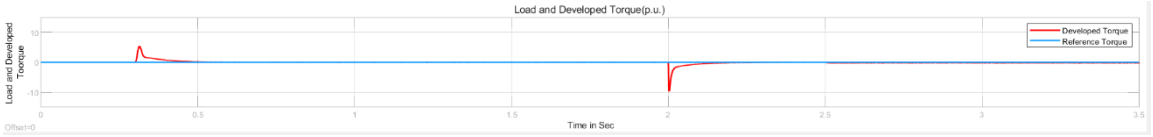


Figure 5.16: Torque Response during step Speed change and Reversal

- Response wave shapes, during rotor resistance change

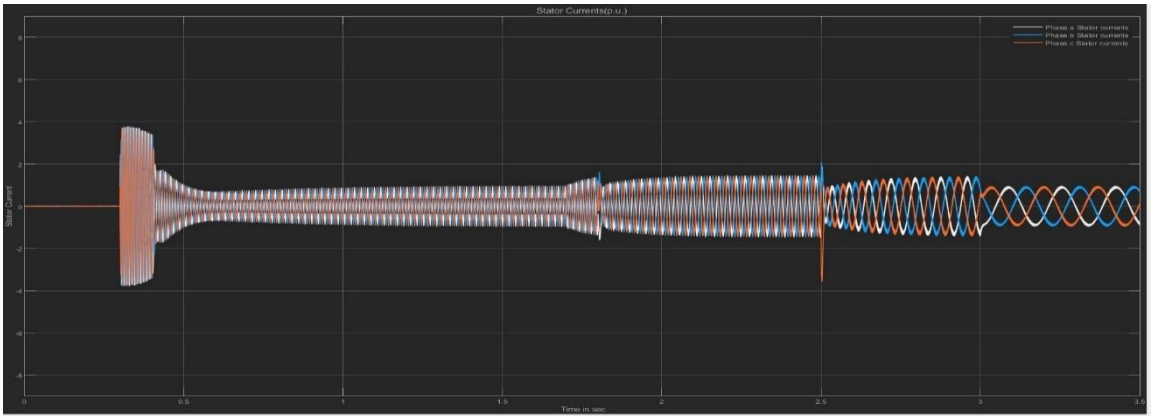


Figure 5.17: Stator Current wave form during Rotor resistance change by 100%

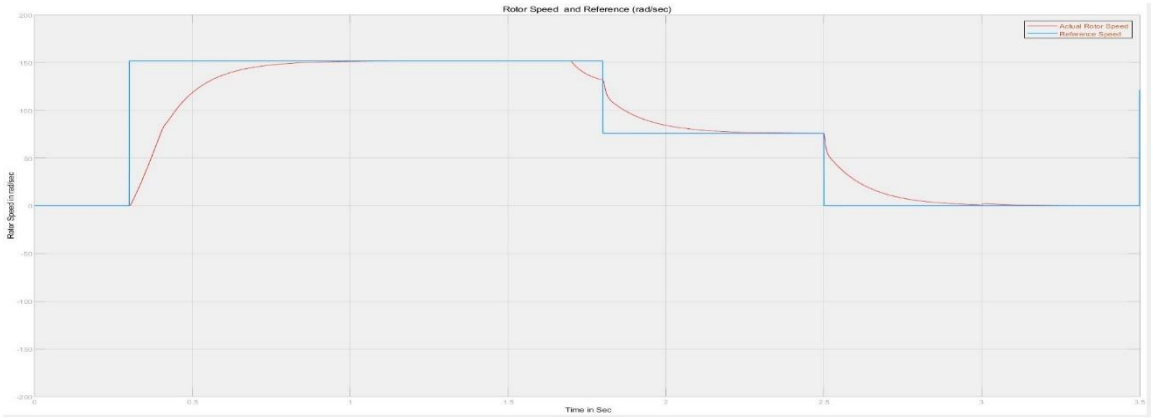


Figure 5.18: Speed response during Rotor resistance change by 100%

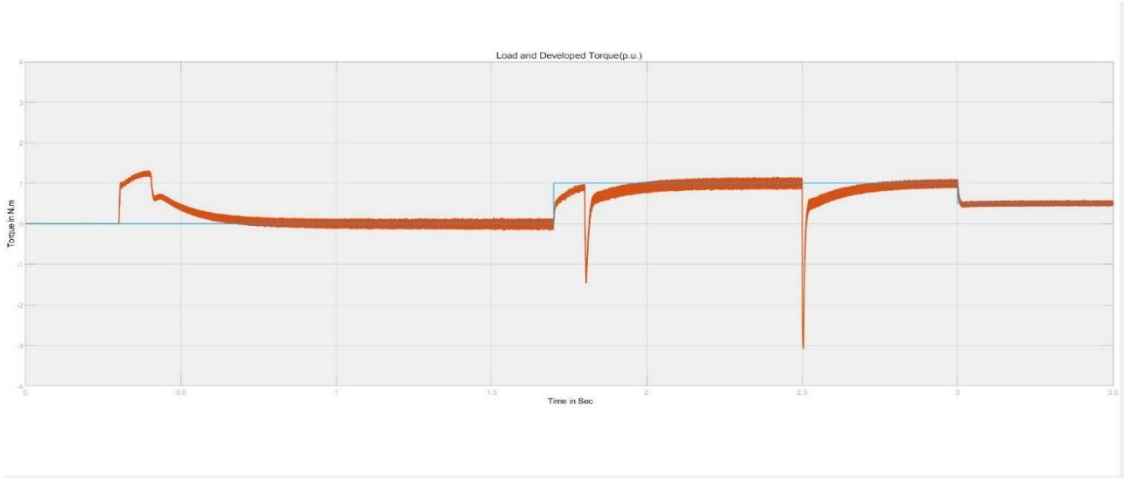


Figure 5.19: Torque response During Rotor Resistance change by 100%

Performance measurement summary during No load and Full Load operation

Table 5.2: Summary of PI and ANFIS controlled drive during different operating conditions

Performance measures	Speed response with PI Controller		With ANFIS controller	
	During No Load	During Full Load	During No Load	During Full Load
Rise time	0.1412	0.1498	0.0420	0.056
Settling time	0.2009	0.2119	0.0655	0.083
Steady state error	0	0.0037	0	0
Maximum overshoot	0.0445	0.0693	1.532	1.0647
Torque ripple at steady state	0.1	0.0	0	0

5.3 Speed estimation and control results discussion theoretically

The design and training of a neural network for satisfactory performance requires very time-consuming iterative procedure with large training data table. Fortunately, different software's highly automates the training procedure. The selection of hidden layer neurons normally requires several stages of iterations because there is no unique way to determine the optimum number of hidden layer neurons. If the number is small, the estimation error will not converge to the satisfactory level. Again, if the number is too large, the network will tend to memorize (look-up table function) rather than learning.

The main goal in this thesis was to obtain the estimation of the rotor flux vector (its magnitude and position) and rotor speed in various operation condition with help of neural network of the simplest structure.

The neural model was trained on the base of training patterns, prepared as input/output data table obtained by simulations of the induction motor transients during reference speed variation of the motor and load torque step changes. The backpropagation algorithm with Levenberg-Marquardt's modification was used for training procedures.

Input output patterns required for the training of ANN estimator are obtained by simulations of Induction motor model in stationary reference frames with the help of MATLAB. The neural network model developed which estimates the speed from the eight inputs namely $\alpha\beta$ axes stator voltages and currents and their delayed versions. The modelling equations for the speed estimation in terms of voltage and current is derived in chapter three. In this chapter the results during different dynamic and steady state conditions are discussed.

5.4 Speed Sensorless operation of three phase Induction Motor

A complete simulation model for vector-controlled Induction motor drive incorporating the proposed scheme is developed. It is simulated with PI controller and required data for training the ANFIS controller is obtained. The ANFIS controller is designed with two inputs, the speed deviation and its derivative and one control output. Seven linguistic variables for each input variable were used to get the desired performance. ANFIS uses neural networks to tune a fuzzy logic Sugeno type controller. To obtain the membership functions and the rules it presents hybrid learning with back propagation algorithm to tune them and least square methods to identify them. This thesis uses the ANFIS editor toolbox to train the ANFIS. The ANFIS inputs are used with seven generalized bell type membership functions, looking for a linear membership function at the output with an error tolerance equal to zero.

To assess the proposed scheme, various simulation tests are carried out with PI controller and proposed scheme with ANFIS controller. The motor parameters are in Table A.1 in Appendix.

In this thesis, a sincere attempt is made to improve the dynamic responses and make the speed of response very fast by designing an efficient controller using ANFIS control strategy, which is the one of the contributions of the thesis. The proper fuzzy inferencing system was generated by using Grid partitioning method. Here, we have formulated this complex control strategy for the speed control of IM, which has yielded excellent results compared to the other conventional controllers. The results of the simulation have showed a good transient response and a non-oscillating steady state response with excellent stabilization and we have tried to improve the dynamic performance of the developed controller by developing a sophisticated adaptive neuro fuzzy algorithm.

Simulink model was developed in MATLAB® 2019a with the ANFIS controller put in closed loop with the plant. The control strategy was developed by writing a set of 49 adaptive fuzzy rules coupled with the Hybrid learning algorithm. Simulations were run for a period of 3.5 second and the characteristic curves of speed, torque, current, load, etc. vs time were observed on the corresponding scopes in the developed model. From the simulation results, it can be observed that the characteristic curves of the IM take less time to stabilize and due to the incorporation of the ANFIS controller, it was observed that the motor reaches the rated speed very quickly compared to the other methods. The main advantage of the developed ANFIS coordination scheme is to increase the dynamic performance and to provide good stabilization.

Problems associated with sensorless control systems have mainly included parameter sensitivity, integrator drift, and problems at low frequencies. Some have tried to solve these problems by redesigning the induction machine. As it is most unfavorable using anything but standard machines, redesigned motors are not considered the best solution. The questions raised in this work are: what is the best possible solution using standard motors? To what extent can the problems at low frequencies, and the parameter sensitivity problems be reduced?

The simulation results of the proposed ANN speed estimator for sensorless speed control of induction motor drive is discussed in terms of:

- Set point tracking capability,
- Torque response quickness,
- Low speed behavior,

- Sensitivity to motor parameter uncertainty.

a) Set point Tracking capability

It is always crucial to assess the performance of an estimator based on the ability of the estimated speed to converge to the actual value, especially during transient state. This criterion has been well accepted as a primary indicator when benchmarking the performance of a sensorless speed estimator. It shows the convergence of the estimated rotor speed to the actual speed. Using the same parameters in the IM and the ANN estimator, the tracking performance of the estimator can be examined by changing the speed reference of the system. As can be seen from Fig.5.20 the proposed estimator tracks the step variation reference input. The estimated and the actual speed follow the reference speed with good accuracy and it takes 0.09second to track the reference speed at different level of speed including low speed region.

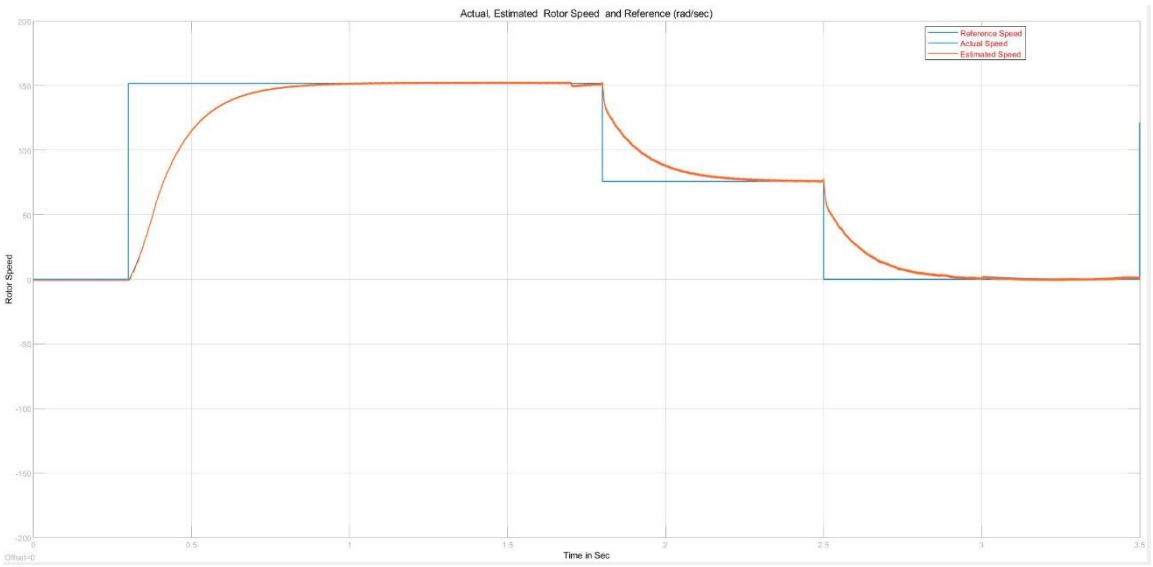


Figure 5.20: Speed response for Step change in reference speed with ANN estimator

b) Torque response quickness

To find the torque response quickness the motor is started with a zero torque and this value is increased to 21N.m after 1.6 seconds causing a drop-in motor speed half rated torque loaded at 3seconds. This happens because of the mismatch in the torques, i.e.; the developed torque is less than the load torque. To compensate for this mismatch, the controller increases the developed torque by increasing thus in effect the motor speed increases and comes back to the set point as shown in Figure 5.21.

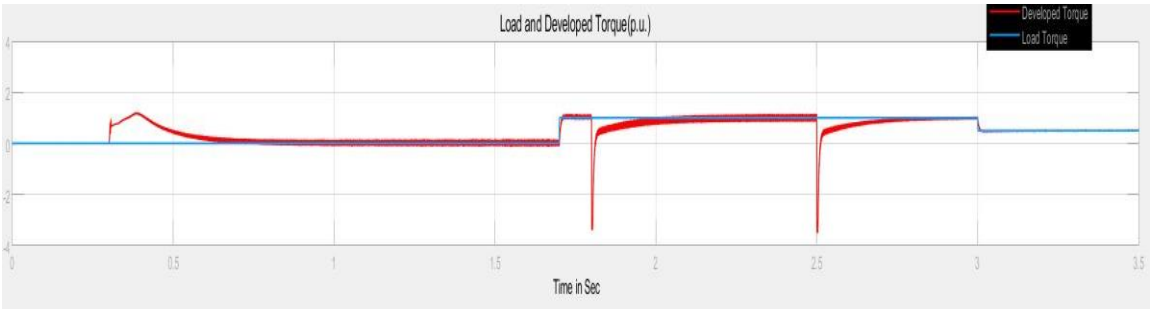


Figure 5.21: Torque Response during Load and reference Speed change

c) Low speed behavior

The aim of this test is to evaluate the performance of the ANN estimator at low speed. Fig.5.22 show that the estimated speed follows the actual speed very closely. There is also good field orientation down to zero speed. This means the system is stable at zero speed and continuous operation is possible. There is a very short period during settling when the isq response presents some oscillation due to the relatively poor speed estimate (this is large for the full load case). However, after a short period speed and current settle to their respective steady state values.

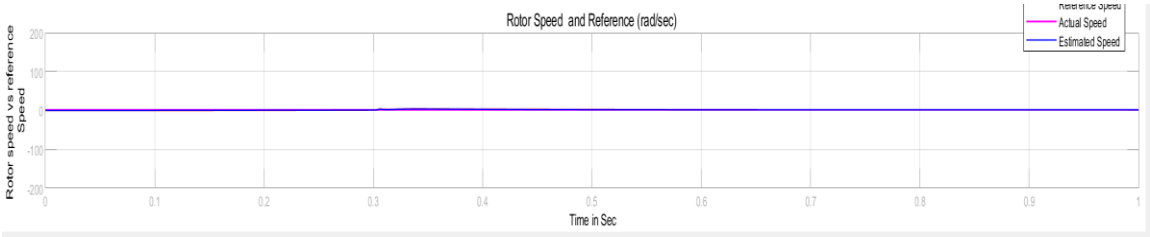


Figure 5.22: Speed response at zero speed without Load torque applied

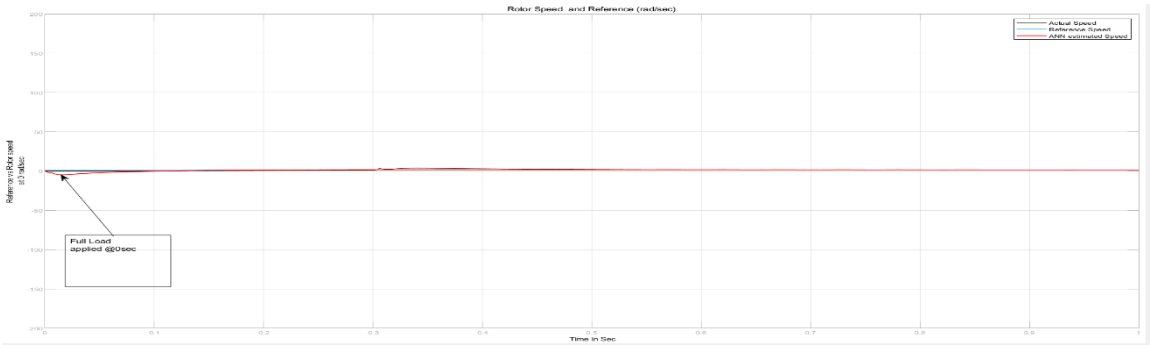


Figure 5.23: Speed response at zero speed, but full load torque applied

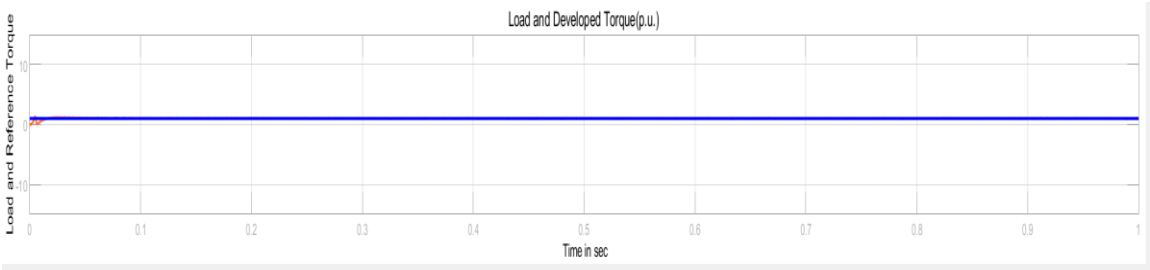


Figure 5.24: Torque response when Full load torque applied at zero speed

d) Parameter sensitivity

It is understood that the most of estimator's performance like MRAS, EKF and Luenberger Observers are highly dependent on the IM parameters since it structures realization is directly extracted from the IM dynamic equations. The IM parameters are affected by variations in the temperature and the saturations levels of the machine.

Incorrect setting of parameters in the motor and that instrumented in the vector controller and estimators will results in the deterioration of performance in terms of steady state error and transient oscillations of rotor flux and torque. As a consequence, parameter sensitivity has been treated as a secondary issue in a vector-controlled IM drives system. Some of the parameters detuning effect being studied are the stator resistance, rotor resistance, stator self-inductance, rotor self-inductance and motor moment of inertia. Amongst these parameters, stator and rotor resistances variation has been reported to have large influence on the estimator's performance. Others parameters has minimum effects but as the variations becomes larger, the effect to the estimator's performance also becomes significant.

Unlike, model-based estimators' neural network-based estimators are robust for parameter variation when they are trained with different operating points with enough training data. We have included training data that from simulation with 100% variation of stator and rotor resistance to make the network more general for parameter variations. The speed response result with PI controller is shown in Fig. (5.18) and it diverges from reference value, because when resistance changes occur subsequently current changes and that makes Torque to change from their set value. But ANN estimator and ANFIS Controller are more robust to parameter variations and they handle this properly.

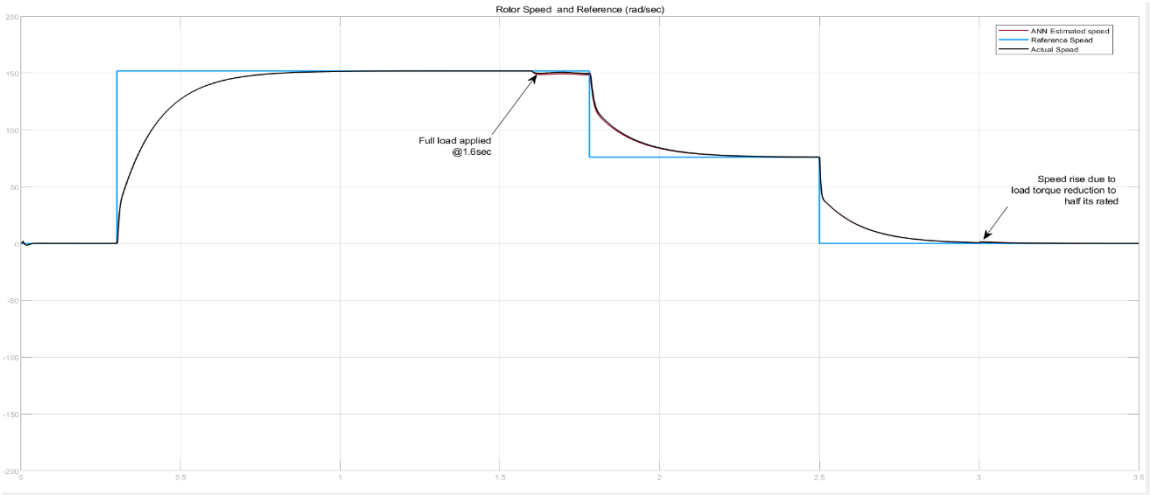


Figure 5.25: Speed response during Rotor resistance variation and step change in reference speed

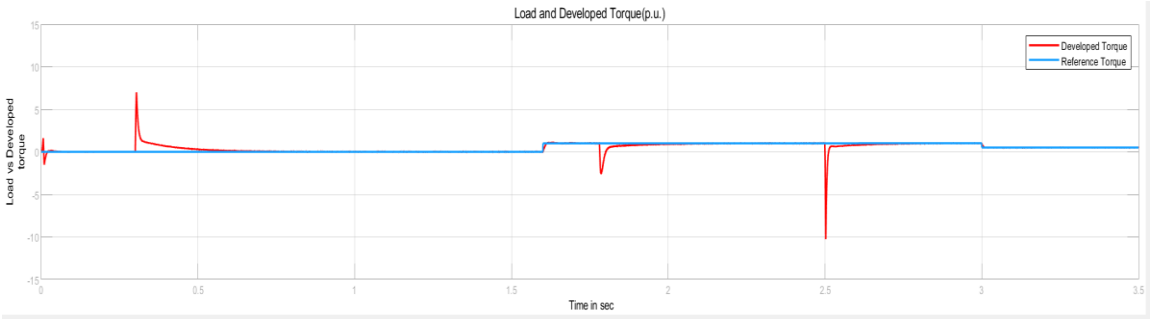


Figure 5.26: Torque response during rotor resistance change and step change in speed

CHAPTER 6 CONCLUSIONS AND FUTURE WORKS

6.1 Conclusions

In this thesis, mainly four tasks have been performed. These are design and simulation of conventional PI Controller for Inverter fed SCIM using FOC principles, design of feed forward neural network for speed estimation, design of ANFIS controller for indirect vector control of induction motor drive and finally, comparative analysis based on MATLAB/Simulink results of the above two controllers have been performed.

This thesis outlines techniques for speed estimation using ANN. The mathematical model of induction machine is considered and expressions for the rotor speed are obtained. A model free method is proposed in this thesis, in which ANN is trained to approximate the equation for the speed from input/output data. So, by training ANN the speed can be recovered with a high degree of accuracy. The performance of this speed estimator is evaluated by performing simulation using MATLAB/SIMULINK software. The simulation results show that, the system has robust enough in a wide range of speed and machine load for steady state in different case, such as no-load, load condition, the variation of the input parameters, change in rotor resistance. It was found out that the estimator has been smooth speed tracking and torque response in both forward, reversal operation and low speed operation. There is an error less than 0.25% between actual and estimator speed for input parameter variations (reference speed and load torque), motor parameter variation (stator and rotor resistance change), and also the speed variation error between estimated and actual speed have been less than 1% during different scenarios.

Due to the incorporation of the ANFIS controller in IM drive control, it was observed that the speed reaches the desired value quickly in a lesser time as compared with conventional PI Controller. Another advantage of the ANFIS is that its speed of operation is much faster and more robust to parameter variation; the tedious task of training of membership functions is done in ANFIS. Collectively, these results show that the ANFIS controller provides faster settling times, has very good dynamic response and good stabilization. It is concluded that the proposed intelligent estimator and controller has shown superior performance than that of conventional PI Controlled and sensored drives.

6.2 Future Research Directions

Operation below base speed is assumed throughout the work and the analysis and implementation of the different sensorless methods for field weakening operation is considered as a topic for further study.

It is proposed for future work that derivative free optimization methods such as Genetic Algorithms, Particle Swarm Optimization, Sequential Quadratic Programming can be used to find the optimal settings for the premise and consequent parameter's optimization in layer one and layer four of ANFIS speed controller structure instead of Hybrid learning Algorithms.

The thesis can be also extended for hard ware implementation using Application specific Integrated chips (ASICs), Digital signal processors (DSPs) and Field programmable gate array (FPGA).

At present, sensor-based controls are still widely used in industrial and transportation applications. But in the near future, speed sensorless-based drive systems will become a practical reality and will be used in many industrial applications. Sensorless-based systems will provide higher reliability and operation in adverse environmental conditions at a lower cost.

REFERENCES

[1] P.D. Marcetic, S.N. Slobodan, "Speed Sensorless AC Drives with the Rotor Time Constant Parameter Update", IEEE Transactions on Industrial Electronics, vol 54, n.5, October 2007, pp. 2618-2625.

[2] Bimal K. Bose," Modern Power Electronics & AC Drives", 1st Ed. Upper Saddle River, USA, Prentice Hall PTR, 2002

[3] Emanuele Cerruto, Alfio Consoli, Angelo Raciti, and Antonio Testa (1997), 'Fuzzy Adaptive Vector Control Of Induction Motor Drives', IEEE Trans. Power Elec.,vol. 12, No. 6, Nov 1997, pp. 1028 – 1040.

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

[5] D.W. Novtony and T.A. Lipo "Vector Control and Dynamics of AC drives." ,1st Edition, Wisconsn U.S.A, Oxford Science Publication,1996.

[6] M.N. Cirstea, A. Dinu, J.G. Khor, M. McCormick, "Neural and Fuzzy Logic Control of Drives and Power Systems" 1st Edition, Oxford, Great Brtain, Newnes publications,2002.

[7] A.M. Trzynadlowski, "The Field Orientation Principle in Control of Induction Motors", 1st Ed, Nevada, U.S.A, The Kluwer International Series,1994.

[8] Gonzalo Abad, "Power Electronics and Electric Drives for Traction Applications",1st Edition,Spain, John WILEY,2017 .

[9] Chee-Mun Ong "Dynamic Simulations of Electric Machinery Using MATLAB®/Simulink" 1st Edition,Indiana,USA,Prentice Hall PTR,1998 .

[10] Paul Krause, Oleg Wasynczuk Scott Sudhoff Steven Pekarek, "Analysis Of Electric Machinery And Drive Systems ", 3rd Ed, New Jersey, U.S.A, John Wiley and Sons Inc, 2013 .

[11] P. H. Tran, "MATLAB® / Simulink Implementation And Analysis Of Three Pulse-Width-Modulation (PWM)," M.Sc Thesis, FEE, Boise State University U.S.A, 2012.

[12] Teshome Hambissa, "Analysis and DSP Implementation of Sensor-less Direct FOC of Three-Phase Induction Motor Using Open-Loop Speed Estimator," M.Sc Thesis, SECE ,AAU, Addis Ababa, 2017.

[13] Werkagegn Tatek, "Model Reference Adaptive Control Based Sensorless Speed Control of Induction Motor," M.Sc Thesis, SECE ,AAU, Addis Ababa, 2017.

[14] D. L. Sobczuk, "Application of ANN for control of PWM Inverter fed Induction Motor Drives," Ph.D. dissertation, FEE, WARSAW University of technology, Poland, 1999.

[15] Howard Demuth, Mark Beale, Martin Hagan, "Neural Network Toolbox 5" User's Guide, The MathWorks, Inc. 3 Apple Hill Drive,2007.

[16] R.B. Gimenez, "High Performance Sensorless Vector Control of Induction Motor Drives", Ph.D. dissertation, FE, University of Nottingham, UK, 1995.

[17] J. Holtz, "Sensorless Control of Induction Motor Drives," Proceedings of the IEEE ,vol. 90, no. 8, Germany, pp 1359-1394, 2002.

[18] "Fuzzy Logic Toolbox, User's Guide", The MathWorks, Inc., 3 Apple Hill Drive Natick, MA 01760-2098, 2015.

[19] "Jyh Shing ,Roger Jang,"Neuro Fuzzy and Soft Computing, Acomputational Approach to Learning and Machine Intelligence" 1st Ed, Upper Saddle River,U.S.A, Prentice Hall, 1997 .

[20] Mrinal Buragohain "Adaptive Network based Fuzzy Inference System (ANFIS) as a Tool for System Identification with Special Emphasis on Training Data Minimization." Ph.D. dissertation, DECE, IIT Guwahati, India 2008

[21] "Fuzzy Logic Toolbox, User's Guide", The MathWorks, Inc., 3 Apple Hill Drive Natick, MA 01760-2098, 2017

[22] Girma Kassa, "Design and Comparative analysis of Genetic algorithm tuned Fractional and Integer order PI controllers with Adaptive neurofuzzy controller for speed control of indirect vector controlled Induction motor," M.Sc Thesis, SECE ,AAU, Addis Ababa, 2019.

[23] Seung-Ki Sul " Control of Electric Machine Drive Systems", 1st Ed, New Jersey, U.S.A, John Wiley and Sons,Inc, 2011 .

[24] Fouad Giri " AC Electric Motors Control, Advanced Design Techniques and Applications ", 1st Ed, UK, John Wiley and Sons,Inc, 2013 .

APPENDICIES

Appendix A. Induction motor parameters

Table A.1 Parameters of Squirrel-cage induction motor used for simulation

Parameters	**Motor**
Power (KW)	3.3
Supply voltage (phase), Vs	220
Number of poles, P	4
Frequency (Hz)	50
Stator Resistance, Rs ($p.u.$)	0.0201
Stator Leakage inductance, $Lls(\ p.u)$	0.0349
Rotor Resistance, Rr(ohm)	0.0377
Rotor Leakage inductance, $Llr(p.u)$	0.0349
Mutual inductance, Lm ($p.u$)	1.527
Inertia ($Kg.m^2$)	0.001
Friction factor, $F(Nms^{-1})$	0.0106
Rated rotor speed(rad/sec)	151.84

Appendix B. Speed Estimation Using Artificial Neural Network

```
Speed Estimation in Stationary Reference Frame
vs_alpha1  =   out.vs_alpha(:,1)    ;                  % vs_alpha(k), To
Workspace block writes data to workspace and we call this data by using
out. function
vs_alphad   =   out.vs_alpha_d(:,1)  ;            % vs_alpha(k-1)
vs_beta1    =   out.vs_beta(:,1)     ;            % vs_alpha(k)
vs_betad    =   out.vs_beta_d(:,1)   ;            % vs_beta(k-1)
is_alpha1   =   out.is_alpha (:,1)   ;            % is_alpha(k)
is_alphad   =   out.is_alpha_d(:,1)  ;            % is_alpha(k-1)
is_beta1    =   out.is_beta(:,1)     ;            % is_beta(k)
is_betad    =   out.is_beta_d(:,1)   ;            % is_beta(k-1)
omegar      =   out.omega_r(:,1)     ;            % measured rotor speed
P=[vs_alpha1  vs_alphad vs_beta1 vs_betad is_alpha1  is_alphad is_beta1
is_betad ]'; % Input Vectors
Targets=omegar' ;                                    % Target output
trainFcn = 'trainlm';             % Levenberg Marquardt backpropagation.

hiddenLayer1Size = 10;
hiddenLayer2Size = 15;
net = fitnet([hiddenLayer1Size hiddenLayer2Size],  trainFcn);
net.trainParam.goal=1e-6;
net.layers{1}.transferFcn = 'tansig';
net.layers{2}.transferFcn = 'tansig';
net.layers{3}.transferFcn = 'purelin';
net.trainParam.epochs=100;
net.trainParam.trainRatio=60/100; % 60 percent of data used for training
net.trainParam.valRatio=20/100;   % 20 percent of data used for Validation
net.trainParam.testRatio=20/100;  % 20 percent of data used for Test
[net,tr]=train(net,P,Targets)
omega_est=net(P)
t=0:Tso:3.5;
plot(t,omega_est,t,Targets)
grid on
errors=gsubtract(omega_est,Targets);
plot(t,errors)
xlabel('Time in seconds')
ylabel('error in NN estimation')
grid on
performance = perform(net,Targets,omega_est)
tInd = tr.testInd;
tstOutputs = net(P(:, tInd));
tstPerform = perform(net,Targets(tInd), tstOutputs)
% View the Network
view(net)
gensim(net,Tso)
```

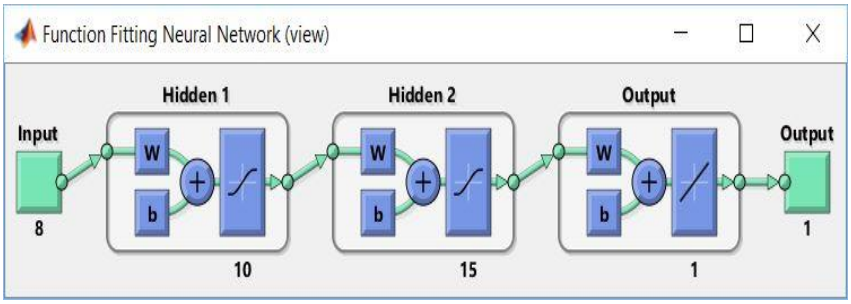


Figure B.1: Function fitting Neural Network view

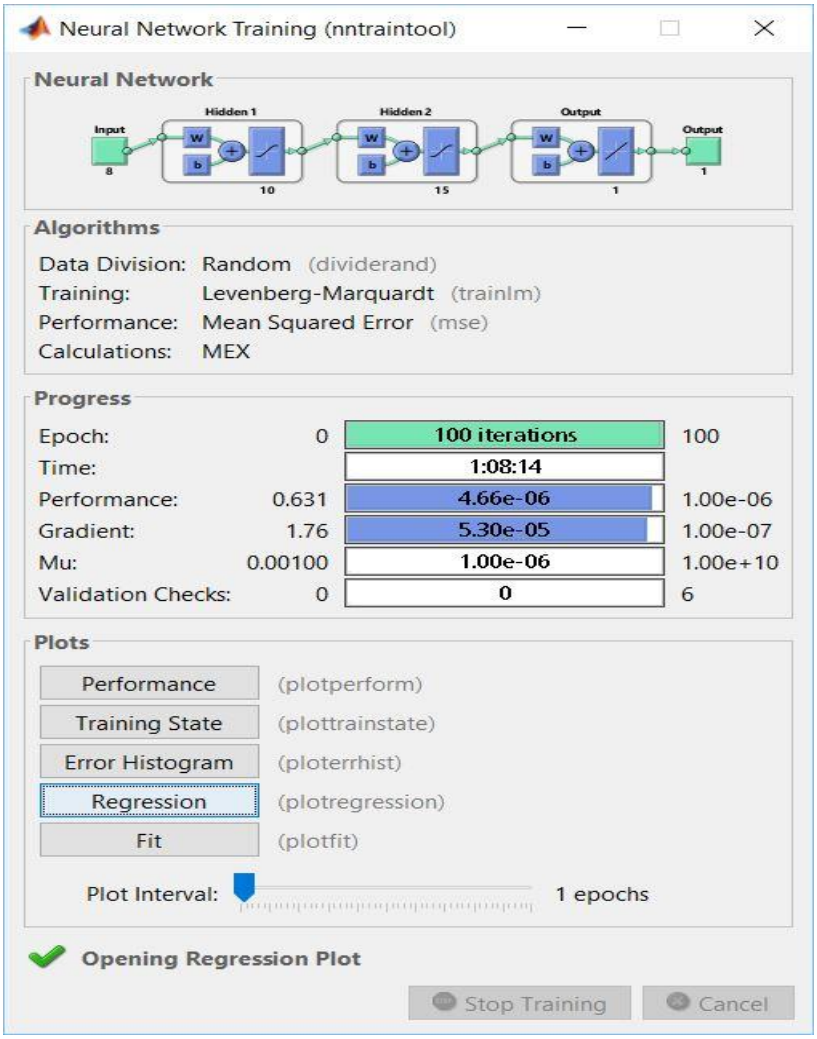


Figure B.2: Neural Network Training Tool

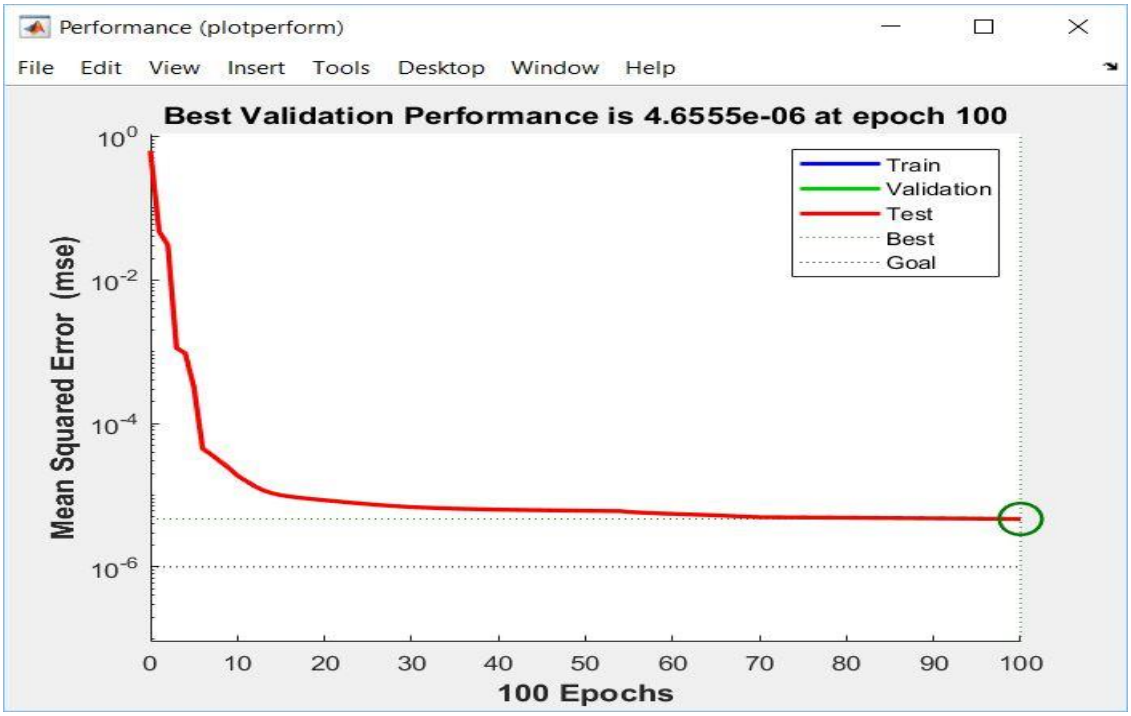


Figure B.3: Performance plot of trained Neural Network

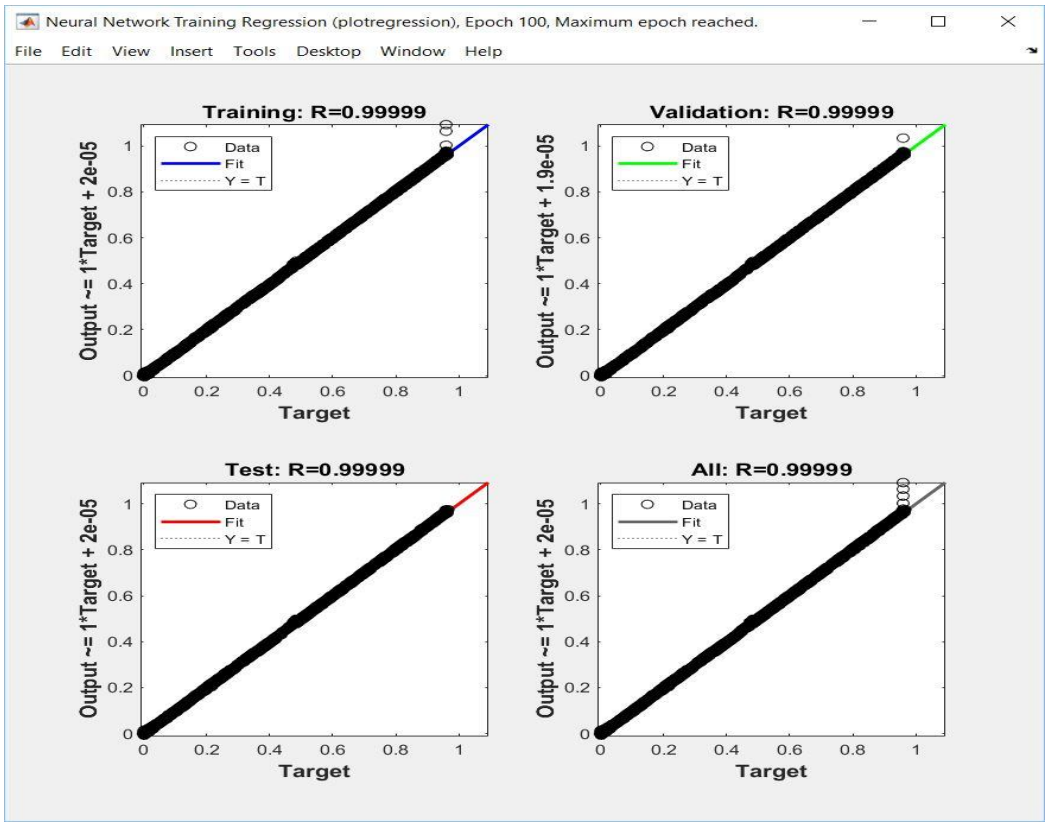


Figure B.4: Neural Network Training Regression plot

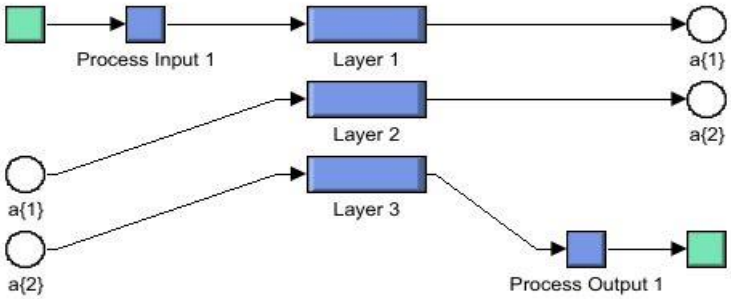


Figure B.5: Inside the Simulink block of Neural Network Estimator

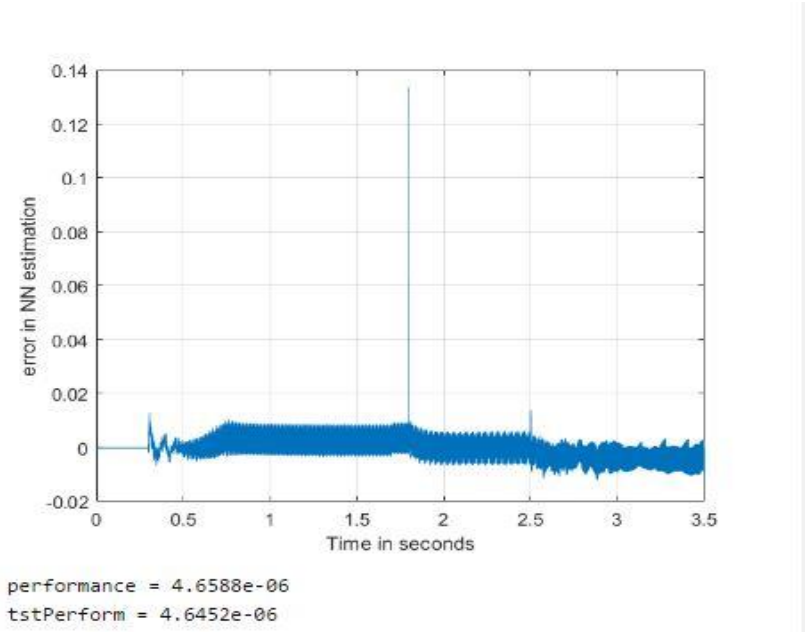


Figure B.6: Estimation Error during estimation

Appendix C. Speed Control using Adaptive Neuro Fuzzy Inference System using MATLAB procedure

We have defined seven Linguistic Variables for both inputs and output as Negative Big (NB), Negative Medium (NM), Negative Small (NS), Zero (Z), Positive Small (PS), Positive Medium (PM) and Positive Big (PB)). They are normalized and their range belongs to [-1 1]. Their respective plots are shown below.

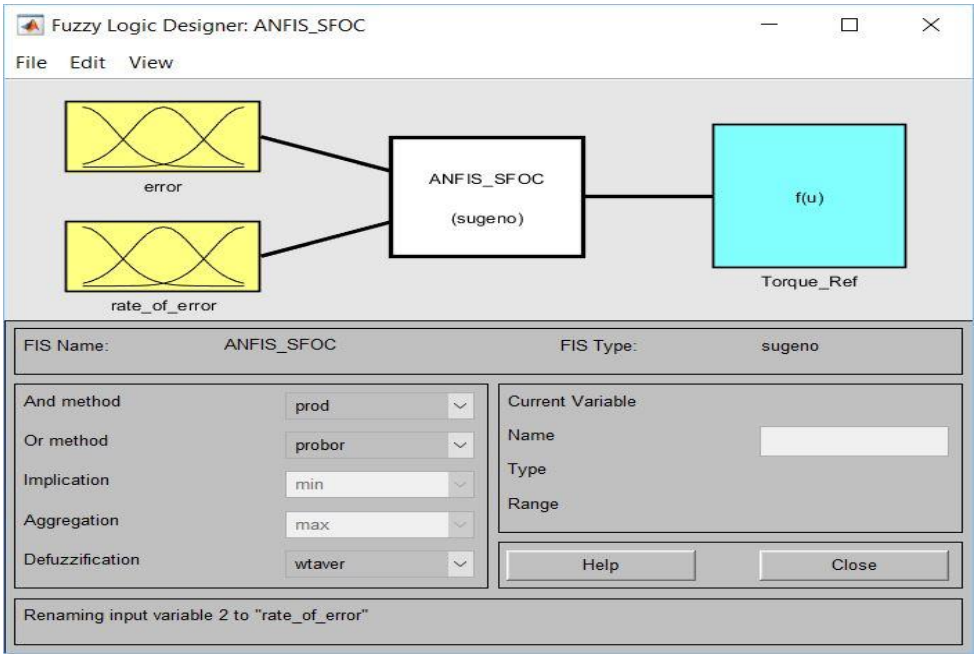


Figure C.1: FIS Property Editor for Sugeno type Fuzzy Model with two inputs and one output

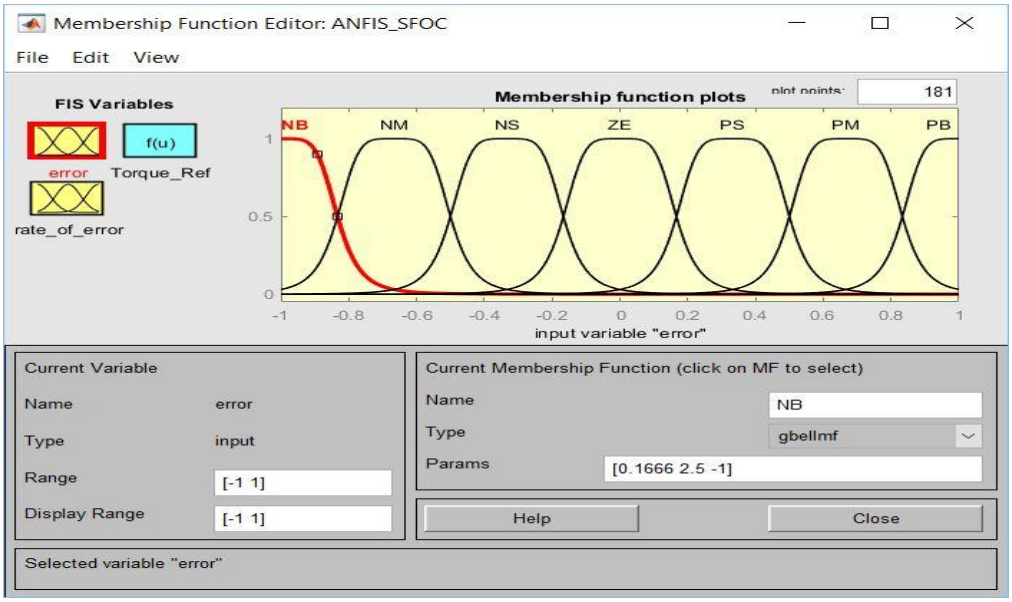


Figure C.2: Membership functions of input 1, error

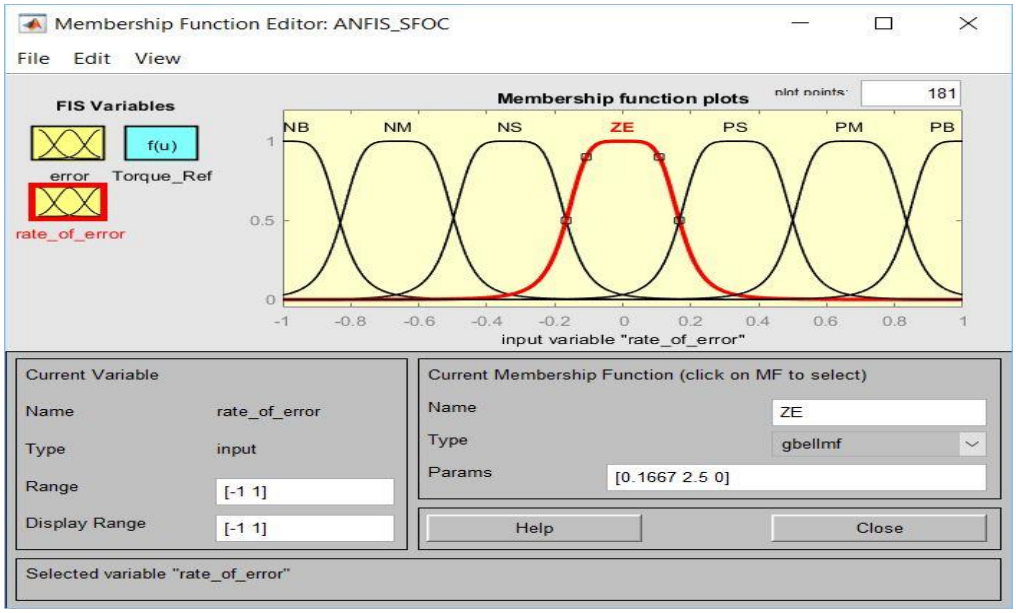


Figure C.3: Membership functions for input 2, rate of error

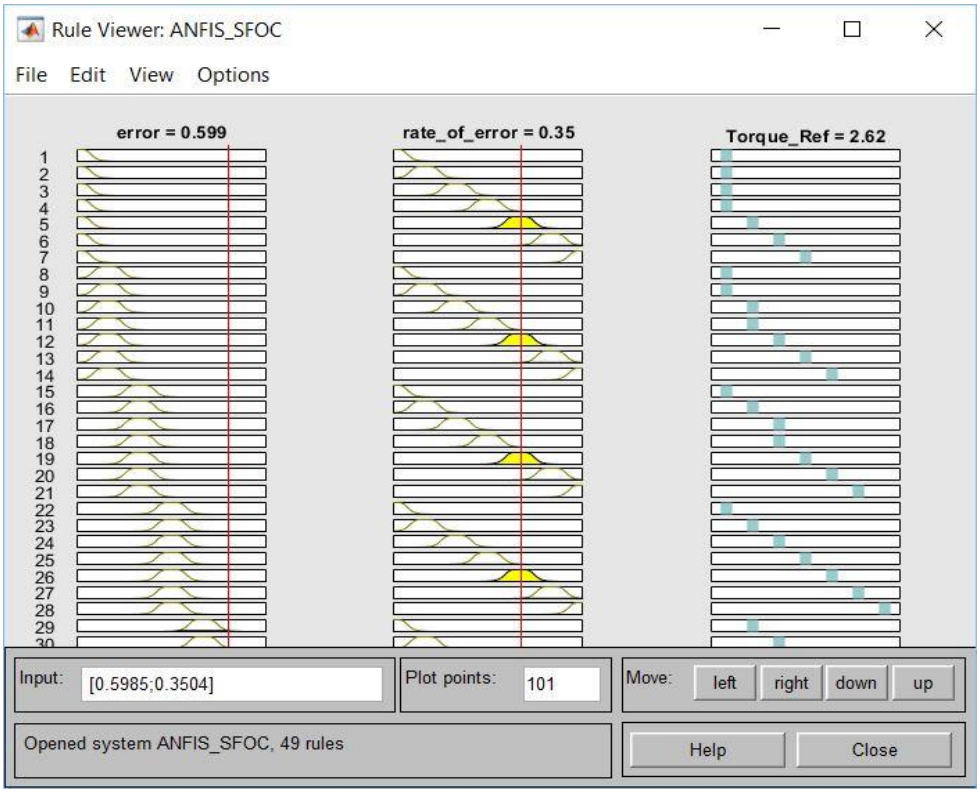


Figure C.4: Rule Viewer

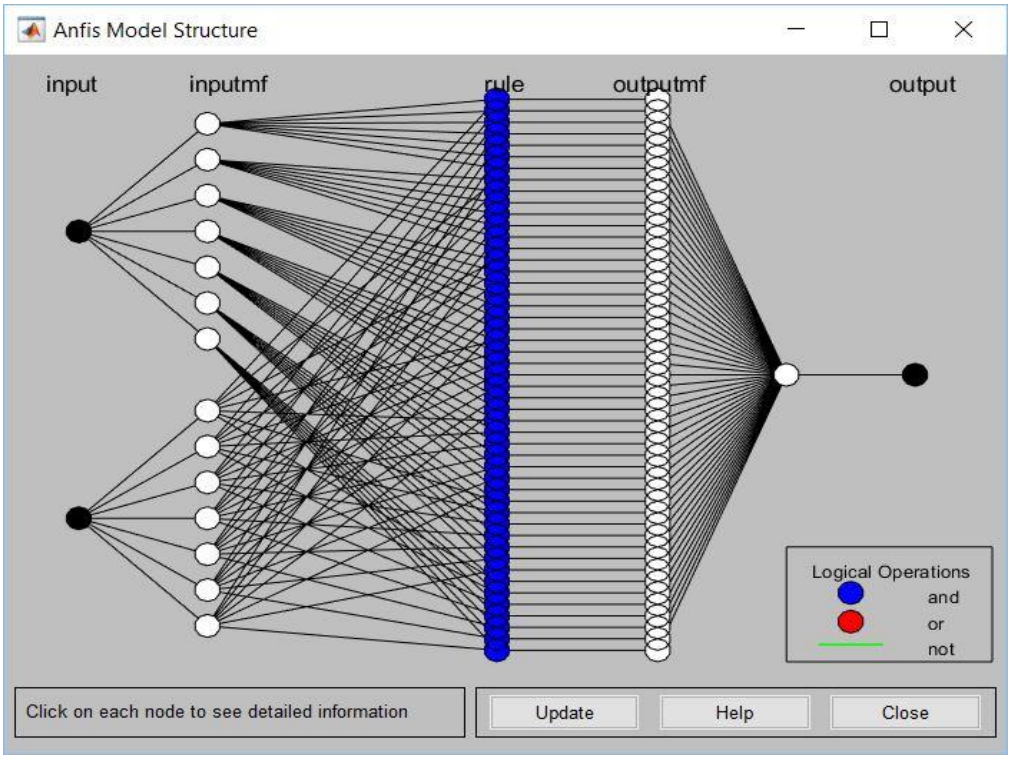


Figure C.5: ANFIS Model structure

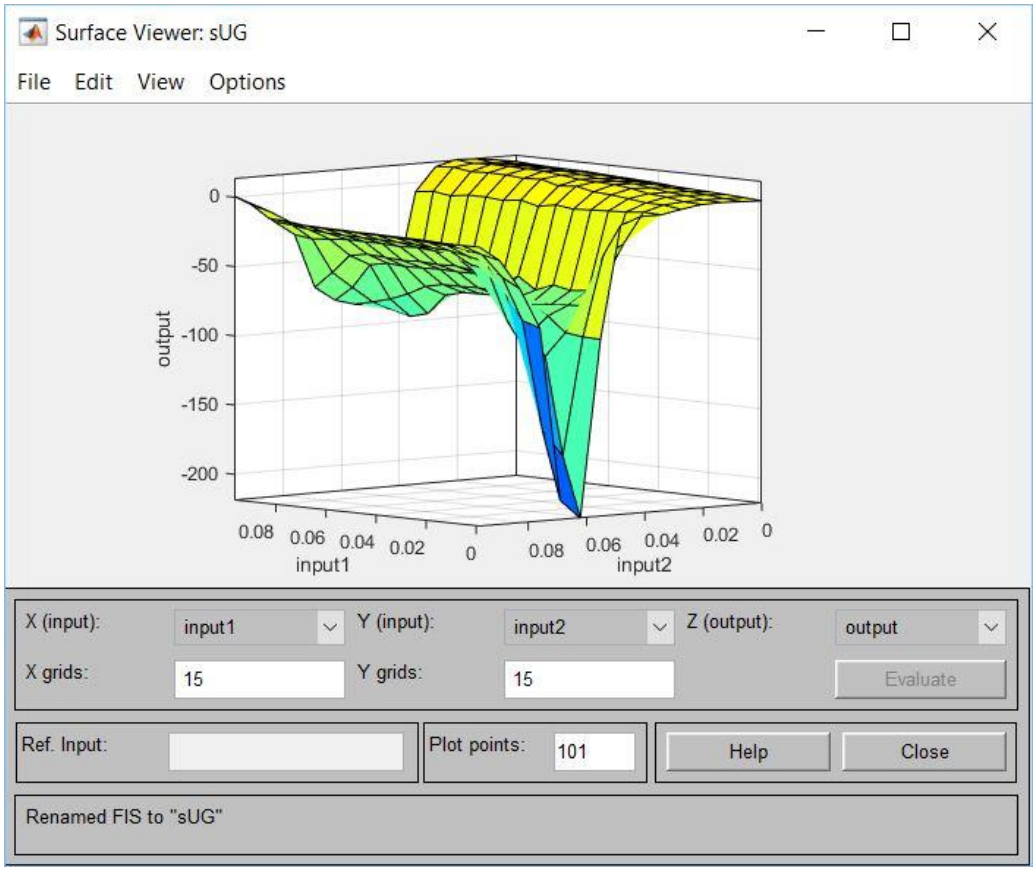


Figure C.6: Surface viewer after ANFIS Training

Minimal training RMSE is 0.000182 and minimal checking RMSE is 0.000270794 is achieved for ANFIS, as a controller. 100 epochs training conducted with hybrid training and zero error tolerance. Satisfactory performance is obtained with the above RMSE.