Addis Abeba university

Addis Abeba Institute of Technology

School of Ethiopia and Computer Engineering

Speed Control of Three Phase Induction Motor Using Adaptive Neuro Fuzzy Inference System(ANFIS) And FUZZY-PID Controllers

A Thesis Submitted to the School of graduate studies of Addis Abeba University in Partial Fulfillment of the requirement for the Degree of

Masters of Science in Control Engineering

By

Abemelek Getachew

Advisor

Chala Merga (Assistant Professor)

September , 2024

Addis Ababa, Ethiopia

Addis Abeba University

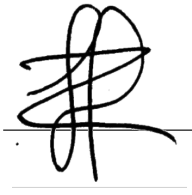
Addis Abeba Institute of Technology

School of Electrical and Computer Engineering

Speed Control of Three Phase Induction Motor Using Adaptive Neuro Fuzzy Inference System(ANFIS) And FUZZY-PID Controllers

By: Abemelek Getachew

By the Board of Examiners' approval

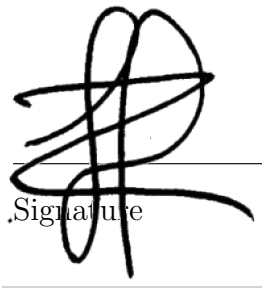
Name	**Date**	**Signature**
(Dean,School of Graduate Committee)		
Dr. Chala Merga Abdissa	26.Jan.2025	
(Advisor)		
(Internal Examiner)		
(External Examiner)		

Declaration

This certifies that the thesis, "Speed Control of Three Phase Induction Motor using ANFIS Controller and Compare with Fuzzy tuned PI Controller," written by Abemelek Getachew and submitted as a partial fulfillment of the requirements for the Master of Science in Control Engineering degree, compiles with university regulations and meets accepted standards for originality and quality.

Name of student	Date	Signature

This thesis has been submitted for evaluation with my endorsement as the university advisor.

Dr. Chala Merga Abdissa	26.Jan.2025	
Name of the Advisor	Date	Signature

Acknowledgment

Initially, I wish to express my appreciation to God, whose guidance has been instrumental in my success. He has directed me along the right path and endowed me with the wisdom and patience needed to complete this work.

This thesis is dedicated to my late sister Hermela Getachew (PhD Candidate) who unexpectedly passed away recently. This paper is one of the many accomplishments I couldn't have done without her.

I want to thank my advisor from the bottom of my heart, Assistant Professor Chala Merga, for his exceptional guidance, invaluable feedback, and continuous support throughout the completion of this work. His extensive knowledge and commitment to academic success have greatly influenced the direction of this study. I feel incredibly fortunate to have had the chance to study under him because he is not just an outstanding scholar but also an incredibly motivating advisor. I will forever be indebted to him for his generosity and mentorship, and it has been a privilege to study under his direction.

I am incredibly appreciative of my family's consistent support and understanding over the highs and lows of my academic career. Their encouragement has been truly priceless, and I consider myself incredibly fortunate to have them by my side. Last but certainly not least, I want to sincerely thank Mekdelawit Tadesse, my future wife, for her amazing help and support.

Without the combined assistance of all these people, this thesis would not have been feasible, and for that, I am truly grateful.

Abemelek Getachew

Abstract

This paper presents the application of a rule-based Artificial Neuro Fuzzy Inference System(ANFIS) controller for closed loop Volts/Hertz(V/f) induction motor speed control. ANFIS controllers provide several advantages over conventional controllers,including economic feasibility,broader operational range, and easier tuning using natural language. Additionally,self-organizing fuzzy controllers can automatically refine an initial set of fuzzy rules.

The proposed control architecture utilizes two normalized input signals-Speed error and its derivative-to generate the output frequency change. Membership functions and fuzzy rules are defined using the Fuzzy Inference System(FIS) editor in MATLAB. The control surface is analyzed to verify the relationship between inputs and outputs.

The system is modeled in MATLAB/SIMULINK,and the performance of the proposed Neuro-Fuzzy Logic Controller is compared with that of a conventional Proportional-Integral(PI) controller. The controller is fine tuned through trial-and-error,followed by auto-tuning simulations.Simulation results demonstrate the effectiveness and superior performance of the proposed control approach in achieving precise speed regulation for the induction motor.

Through simulation results produced with MATLAB/SIMULINK software, the efficacy of the suggested control approach is confirmed.

Key words: *V/f induction motor speed control,Artificial Neuro Fuzzy Inference System (ANFIS),proportional integral derivative controller,Fuzzy Logic(FLC), MATLAB/SIMULINK*

Contents

Declaration ii

Acknowledgment iii

Abstract iv

1 Introduction 1

1.1 Background of The Study 1

1.2 Statement of The Problem 4

1.3 Objective 6

1.3.1 General Objective 6

1.3.2 Specific Objective 6

1.4 Methodology 6

1.5 Study Significance 7

1.6 Study Limitations and Scope 7

1.7 Structure of the study 8

2 Literature Review 9

2.1 Construction and Operation of Induction Motors 9

2.1.1 The Rotating Magnetic Field Principle 10

2.2 Induction Motor Speed Regulation 11

2.2.1 Scalar Control 11

2.2.2 Vector Control 11

2.2.3 Regulating Speed as a Process Control Technique 11

2.2.4 Techniques for Controlling Speed 12

2.2.5 Controlling Variable Frequencies Using Voltage Sources 13

2.3 Volt/frequency Control Basics 14

2.4 Proportional Integral(PI) Controller 15

2.5 Adaptive Neuro-Fuzzy Inference System 15

2.6 Controllers for Neural Networks and Fuzzy Logic 17

2.7 Conventional Controllers 18

3 System Design and Modeling of The Induction Motor **19**
3.1 Transformation of Coordinate Frames . . . 19
3.1.1 The Transformation of Clarke . . . 19
3.1.2 Parks Transformation . . . 19
3.1.3 Clarke Park Transformations . . . 20
3.1.4 Volt/Hertz(Scalar Control) . . . 21
3.2 Mathematical Modeling . . . 24
3.2.1 Equivalent Circuit of Induction Motor . . . 24
3.3 Space Vector Model of Induction Machine(SI Units) . . . 28
3.3.1 Electrical System Equations . . . 28
3.3.2 Flux Linkage Current Relations . . . 29
3.3.3 Equations for Mechanical Systems . . . 30
3.4 Pulse Width Modulation(PWM) . . . 30
3.5 Three Phase Induction Motor Dynamic Model Verification . . . 31
3.5.1 Dynamic Modeling . . . 32
3.6 Simulation Results for Model Verification of Three Phase Induction Motor . . . 35
3.6.1 Simulation Results of IM with Full load . . . 35
3.6.2 Simulation Results of IM with NO load . . . 35
3.6.3 Simulation Results of IM at Breaking down torque . . . 37

4 Controller Designs **38**
4.1 Controller Tuning . . . 38
4.2 Reasons for choosing the ANFIS (Adaptive Neuro-Fuzzy Inference System) . . . 38
4.3 Adaptive Neuro-fuzzy Inference System . . . 39
4.3.1 Membership Function Design . . . 40
4.3.2 Learning algorithm of ANFIS . . . 42
4.3.3 Hybrid Learning Algorithm . . . 44
4.3.4 The creation of neuro fuzzy adaptive inference systems for speed control . . . 45
4.4 Design of Fuzzy Logic Controllers . . . 50
4.4.1 The fuzzy language of input and output variables . . . 51

5 Simulation Results **55**
5.1 Simulation Modeling . . . 55
5.2 Simulation results of IM with Controllers . . . 56

5.2.1 Three-phase stator current and estimated d-q current 56

6 Conclusion and Recommendations **65**

6.1 Conclusion 65

6.2 Recommendation 66

References **68**

A Appendix **69**

List of Figures

2.1 Rotor Squirrel Cage Construction 10
2.2 Construction of Wound Rotors 10
2.3 Variable Frequency Control 14
3.1 Field-Oriented Control Transformations 21
3.2 Simplified Induction Motor Steady-State Equivalent Circuit 21
3.3 Stator Voltage versus Frequency Profile under V/Hz Control 22
3.4 An Induction Motor with Constant Stator Flux: Torque against Slip Speed 23
3.5 Open Loop and Closed Loop 23
3.6 A circuit diagram of an induction motor with equivalent 25
3.7 Reference frames in induction machine analysis 29
3.8 Generation of Sinusoidal PWM signal 31
3.9 Open Loop 31
3.10 Closed Loop 32
3.11 d and q axis current transformations 33
3.12 MATLAB Simulink Model of Three Phase Induction Motor 34
3.13 Electromagnetic torque and speed at full load 35
3.14 Output Current at full load 36
3.15 Electromagnetic torque and speed at NO load 36
3.16 Output Current at NO load 36
3.17 Electromagnetic torque and speed at Breaking down Torque 37
3.18 Output Current at Breaking down Torque 37
4.1 The ANFIS Control scheme of Speed controlling IM 40
4.2 ANFIS Controller input and Output 40
4.3 Power Circuit Connection Diagram of Induction Motor 41
4.4 Basic ANFIS Structure 42
4.5 Learning procedure for ANFIS using a hybrid algorithm 43
4.6 ANFIS Controller design Flow chart 47
4.7 ANFIS Torque Regulator 48
4.8 Data sets are loaded into the ANFIS editor toolbox for training and verification. . 49
4.9 Testing FIS with checking Data set 49
4.10 Schematic of proposed ANFIS controller 50
4.11 Function of Membership for the Input Variables e and ce 52
4.12 FIS editor: MATLAB's rules window 53

4.13 The input error's membership function (Δe) 53
4.14 Membership function for the input Change of Error 53
4.15 Membership feature for the Change of Control output (ω_{Sl}) 54
4.16 A three-dimensional control surface plot 54
5.1 Over all Model of ANFIS based scalar Control of IM 57
5.2 Over all Model of scalar Control of IM with Controllers 58
5.3 Three phase sine wave generation Sub system 58
5.4 SPWM Sub-system 59
5.5 Stator Current in Three Phases Under Loading (5 Nm) 59
5.6 The d-q Current in Loaded Conditions, Both Real and Estimated (5 Nm) 59
5.7 ANFIS and PID Controller output @ load torque 2 Nm 60
5.8 ANFIS, Fuzzy Logic and PID Controller output @ load torque 2 Nm 60
5.9 ANFIS and PID Controller output @ Full load torque 5 Nm 60
5.10 ANFIS, Fuzzy Logic and PID Controller output @ Full load torque 5 Nm 61
5.11 ANFIS and PID Controller output @ load torque 10 Nm 61
5.12 ANFIS and PID Controller output @ NO load torque 62
5.13 ANFIS,Fuzzy Logic and PID Controller output @ NO load torque 62
5.14 ANFIS and PID Controller output @ NO load torque 63
5.15 PID Controller output 64
5.16 ANFIS Controller output 64

List of Tables

4.1 The developed Adaptive Neuro-Fuzzy Inference System's specifications 49
4.2 The set of rules used to determine values of K_p and K_i 52
5.1 **Calculated at Load Torque = 5 N.m** 61

List of Acronyms

PI	Proportional-Integral
EMF	Electromotive force
MMF	Magnetomotive force
FLC	Fuzzy Logic Controller
Id	Direct Axis Current
Iq	Quadrature Axis Current
ANFIS	Adaptive Neuro-Fuzzy Inference System
IM	Induction Motor

Chapter 1

Introduction

1.1 Background of The Study

Since their inception, induction motors have experienced a significant increase in use. They are frequently used as actuators in robotics, single-phase home appliances, a variety of industrial operations, and other related applications. Their cost-effectiveness, straightforward design, and sturdy construction are the main factors contributing to their rising popularity. Moreover, it has been demonstrated that induction motors are more dependable than DC motors. But in spite of these benefits, induction motors exhibit complex, time-varying dynamics that are highly non-linear, and certain states and outputs are not measurable. As a result, controlling them presents a considerable engineering challenge in the industrial sector. Researchers around the world have developed numerous advanced control strategies to address this issue.

One of the key constraints in motor selection for various applications is speed control. As a result, this topic has been extensively researched over the past few years, leading to the development of several approaches. A detailed understanding of these methods has been provided. Among these, the Volts/Hertz control system stands out as the most commonly used speed control technique, thanks to its ability to offer a wide speed range with good performance in both steady-state and transient conditions. As part of this initiative, the Volts/Hertz control system is explained in detail. We refer to this method as "scalar control mode." In contrast to the Vector control mode, where the input and output commands are torque/flux and reference current, respectively, the scalar control mode uses speed as both the input and output command. Despite providing better dynamic speed regulation, vector control drives are more difficult to build because they require real-time coordinate transformations to translate line currents into two-axis representations and vice versa.

Voltage-frequency converters, a significant advancement from the field of power electronics, enable a wide range of speed adjustments. However, due to the highly non-linear dynamics involved in controlling induction motors, sophisticated control algorithms are necessary for effective speed regulation. Traditional controllers, such as fuzzy, neural, or mathematical models, are commonly used for this purpose. The most frequently employed controller types include Fuzzy Logic Controllers (FLC), Proportional-Integral Proportional-Derivative (PD), Proportional-Integral-Derivative (PID), or a mix of these methods.

Many real-world problems can be effectively solved using the PID controller. PID controllers can offer a strong and dependable control if they are calibrated correctly. [1] PID controllers are particularly common in industrial applications because of this precise property. The non-linear dynamics of induction motors cause design complexity, which is the only drawback to using traditional PI, PD, and PID controllers for induction motor speed control. [2] To determine the parameters, the non-linear systems must be linearity by the traditional controllers. Since it is very impossible to create a flawless non-linear model, the parameter values that are derived from it are approximations. Again, independent of changes in load, Variable Speed Drives (VSD) for Induction Motors (IM) require a broad operating speed range and a fast torque response. [3] In order to meet requirement, we must employ incredibly challenging management strategies. Traditional control methods face the following challenges:

1. dependence on the accuracy of the system's mathematical representation.
2. Temperature fluctuations, motor saturation, and load disruptions prevent expected performance.
3. Using typical linear control only results in satisfactory performance at one operating speed.
4. Using the appropriate coefficients to achieve outcomes that are acceptable,

The aforementioned suggests that one needs to be familiar with the model of the system that needs to be controlled in order to use traditional control procedures. Because motor dynamics are non-linear, the conventional approach of computing the mathematical model of an induction motor is challenging. The system behaves in an inappropriate manner whenever environmental conditions or the system itself change. The cost and design complexity of conventional controllers that are intended to deliver great performance are increased. Lately, all other electrical drives, including induction motor drives, have been using soft computation techniques extensively. These strategies are as follows:

1. Artificial Neural Network (ANN).
2. Fuzzy Logic set(FLS)
3. Fuzzy-Neural Network(FNN)
4. Genetic Algorithm Based System(GAB)
5. Genetic Algorithm assisted System(GAA)

In order to overcome the issues with conventional controllers, fuzzy control has been applied in a variety of motor control applications. Fuzzy control has become more and more popular

over the last thirty years due to its knowledge-based algorithm, improved non-linearity handling capabilities, and independence in plant modeling [4]. The fuzzy logic controller's (FLC) prominence can be attributed to linguistic control. In this instance, a precise mathematical model of the system to be regulated is not required.Thus, the control algorithm of fuzzy logic basically aims to imitate human thought processes. [5] Thus, the FLC has shown to be highly helpful in the industries, since it is capable of giving even uncertain nonlinear systems complex non-linear control. A fuzzy logic controller allows for great performance in terms of precision, reliability, speed, and stability in addition to the previously listed features.

Using simple if-then statements, a fuzzy logic controller (FLC) is created with two inputs: the error and the rate of change of error. This can be implemented without complicated hardware. We'll go into more detail on how to model an FLC later. The following is a summary of the advantages that an FLC offers.

- The design is easy.
- For the controller, it gives a glimpse of human intelligence.
- It is economical.
- The system does not need to be mathematically modeled.
- Numerical variables are substituted with linguistic ones.
- The system's non-linearity is easily manageable.
- The system reacts quickly.
- The system is now more reliable.
- A high level of accuracy is attained. [5]

Fuzzy controllers can be used in systems where it is extremely difficult to accurately identify process parameters and describe the process because of these advantages. Consequently, it lends a fuzzy aspect to the control system.

Although the Fuzzy Logic Controller produces better and faster control, its primary design challenges are finding a comprehensive and consistent rule set, the right amount of linguistic variables, and the membership function's shape. To get the correct result, a lot of trial and error must be done, which takes time. However, an ANN is insufficient on its own if there is insufficient training data to cover all operational modes. The limitations of Fuzzy Logic Control and Artificial Neural Networks can both be addressed with the use of the Adaptive Neuro-Fuzzy Inference System. Basic idea behind a neuro-fuzzy network comes from the way humans learn:

fuzzy rules put up an initial understanding of a function, and the neural network's learning skills iterative enhance the degree of function approximation. Thus, the main objective of this thesis was to develop an ANFIS speed controller for a scalar-controlled drive that combines the learning power of neural networks with the fuzzy logic knowledge representation and PI controller speed.

Convolutions neural networks, or CNNs, are a specific type of artificial neural network used in image processing and recognition that is designed specifically to handle pixel input. In order to identify a pattern in the dataset, image classification entails removing the key characteristics from the image data. Convolution is essentially the process of multiplying two functions point-by-point to create a third function. The picture pixel matrix serves as the initial function, and after that, a filter and convolution operation are used to create a feature map or activation map. [6]

1.2 Statement of The Problem

In the majority of induction motor applications, the speed control is rather common. Traditionally, the induction motor's speed information is measured or computed using a rotor position/speed sensor. Installing speed sensors has the benefit of virtually removing the measurement's dependence on the machine control. In certain applications, space constraints and certain environmental factors make the installation of speed sensors challenging or impossible. Due to the shaft's elongation, the installation raises the risk of failures. A significant portion of the overall cost of the auxiliary equipment for induction motor control is attributed to the speed sensor, making it another important consideration.

Artificial neural networks offer several well-known benefits that make them ideal for use in IM (induction motor) control and estimation. These benefits include fault tolerance, robustness to input harmonic disturbances, learning and generalization capabilities, fast parallel processing, and the capacity to accurately estimate arbitrary nonlinear functions. Additionally, using artificial neural network techniques has the benefit of being parameter-independent and not requiring an exact mathematical model of the system. As a result, this method lessens the computational difficulties involved in using extra observers.

The ANFIS speed controller's function is to enhance system performance. Under various operating situations while maintaining cost-effectiveness, efficiency, dynamism, dependability, and the ability to handle non linearity.

ANFIS controller was used in the design and development of this three phase speed control

for induction motors, it will increase the induction motor speed control's efficiency [7]. The following question is thus addressed by the study.

- Three phase induction motor speed controlling method manually is very expensive and it is not chosen as best.
- Error occurs while controlling three phase induction motor speed.
- utilizing different controllers.
- Scalar control exhibits a slow transient response and significant torque reduction at lower speeds.
- Mostly require educating man to perform this manual controlling.
- Less safety of manual controlling.

1.3 Objective

1.3.1 General Objective

Designing and simulating an Artificial Neuro Fuzzy Inference System (ANFIS) based speed controller for a three-phase induction motor is the primary goal of this thesis. Using a scalar control paradigm, this system will be contrasted with a fuzzy-tuned PI controller. To get the desired speed response, the induction motor's voltage and frequency inputs will be regulated.

1.3.2 Specific Objective

The specific objectives of this thesis are as follows:

- for gaining a basic understanding of PID controllers and artificial neuro fuzzy inference systems (ANFIS).
- to discover how to use scalar control to regulate the speed of a three-phase induction motor.
- to understand the torque-speed relationship or characteristics of an induction motor clearly.
- to create the Artificial Neuro Fuzzy Inference System (ANFIS) and train it.
- To gain comprehensive understanding of induction motor dq axis modeling.
- To test and validate the proposed control topology on MATLAB®/SIMULINK® .

1.4 Methodology

This section focuses on utilizing a Fuzzy Logic-PI Controller and an Adaptive Neuro Fuzzy Inference System (ANFIS) for regulating the speed of three-phase induction motors. It involves conducting a comprehensive literature review to explore their application in motor speed control. A mathematical model of the motor's mechanical dynamics and electrical equations will be developed as a foundation for designing the controllers. The accuracy of the model will be validated by comparing it with experimental data and the known characteristics of three-phase induction motors.

The ANFIS Controller is tailored for scalar-controlled three-phase induction motors, considering system dynamics and parameter variations during the selection of the reference and adjustable

models. The adjustable model will be fine-tuned to ensure precise speed control across varying operating conditions. Additionally, a Fuzzy Logic-PI Controller will manage the motor's speed, with fuzzy logic rules and membership functions designed to address the system's non-linearities and uncertainties. The primary objective is to achieve accurate and stable speed control by optimizing the PI gains.

The proposed scalar speed control system is evaluated through computer-based simulations under varying load torque and operating conditions to determine its effectiveness. Based on the findings, practical recommendations will be provided for implementing and optimizing the ANFIS-based scalar speed control system with the Fuzzy Logic-PI Controller in industrial applications. These recommendations aim to address challenges, enhance productivity, and ensure reliable operation in real-world settings.

1.5 Study Significance

Businesses, factories, industries, and anyone that wish to safely manage the speed of an induction motor will find a good speed controlling system in the design and simulation of this controller model. The application of speed control using Neuro fuzzy logic controller is efficient and more convenient controlling system across the world thereby providing significant controlling impact. Moreover, the technology adopted also provides potential technology transfer to speed controlling into possible development of it.

1.6 Study Limitations and Scope

The thesis's scope is to develop an artificially intelligent speed controller based on the scalar control model and employing the Artificial Neuro Fuzzy Inference System (ANFIS) approach. Additionally, it aims to provide a learning package for the ANFIS Controller for scalar speed control of induction motors.

This thesis inhibits the conduct of the following in the entire limitation of the study:

- Implementation of a hardware design.
- Different Control techniques available for controlling of Induction Motor.
- This thesis work does not consider for other motors.
- The simulation does not work for other software other than MATLAB®/SIMULINK®

- Machine elements,computational design and other mathematical functionality model are not fully included in the study.
- Complexity of the MATLAB®/SIMULINK® constructing of the simulation properly.

1.7 Structure of the study

The five chapters that make up this thesis are given in a logical order to preserve the coherence of the information. The thesis's general idea is presented in the first chapter and is developed in later chapters. In the second chapter, scalar control, the dq axis transformation process, artificial neural network types and elements, applications, and particularly control application on the Speed Control process-which is the subject of this section—are reviewed theoretically and empirically.A brief description of the thesis methodology, the software used for this thesis, the procedures for acquiring data, the structuring and building of models using the data, and the approach to data analysis are all included in the third chapter, which is titled Materials and methodology. The Artificial Neuro Fuzzy Inference System (ANFIS) training, validation, and testing process are described in this study along with the data classification ratio. Chapter 4 covers the simulation, results, and discussion section, which provides a graph-based summary of the simulation results. The discussion section extracts, interprets, and summarizes significant material.The Artificial Neuro Fuzzy Inference System (ANFIS) controller's performance is contrasted with the PI controller's by altering the conditions and references. Chapter five, in its entirety, concludes the research findings and offers pertinent recommendations based on them. It also suggests future research areas that this study does not explore.

Chapter 2

Literature Review

The reasons supporting the theories and assumptions are supported by the linked literature and comparable studies that this study mentions. The benchmark data in this chapter, which represents noteworthy works and academic studies that are pertinent to speed controlling systems, offers strong encouragement to continue the investigation.

2.1 Construction and Operation of Induction Motors

Over 85% of industrial motors and single-phase versions are induction motors, and are also used in a variety of domestic applications. If a constant speed motor is significantly shunted, the speed will drop by only a small percentage from idle to full load. Consequently, the majority of applications for induction motors have historically involved constant speed. Traditional speed control techniques have been either inefficient or extremely costly, in contrast to DC motors. However, DC motor drives are not appropriate for usage in filthy or hazardous environments since they need frequent maintenance and feature commutators and brushes. However, despite the fierce competition with DC training in these applications, the induction motor training -especially the type of squirrel cage is a simple and durable, cheaper, such as fan, blower, crane, traction, conveyor. For that reason, it is made for the same purpose. , Cheaper, smaller, and lighter structure.

A three-phase induction motor has a stator and a rotor, much like any other electric motor. An induction motor's stator, which is wound for three phases and a specific number of poles, is composed of several stampings with slots that are uniformly spaced to support the three phase winding. The motor's speed decreases as the number of poles increases and vice versa. A spinning magnetic field of constant strength is created when the stator winding is energized, and the rotor winding receives currents from the peripheral stator via electromagnetic induction, hence the name. The rotor is constructed in two different ways: squirrel-cage and wound-rotor. [7]. It is a hollow laminated core with slots on the outside that is fixed to a shaft.

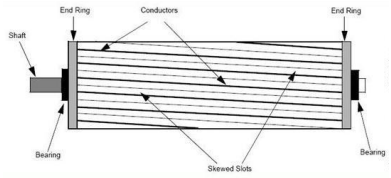


Figure 2.1: Rotor Squirrel Cage Construction

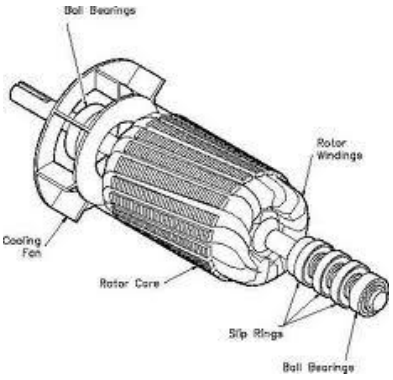


Figure 2.2: Construction of Wound Rotors

2.1.1 The Rotating Magnetic Field Principle

When a three phase voltage is applied to the stator winding, a spinning magnetic field is produced. Because its poles continuously change locations around the stator rather than being in one spot on the stator, this type of field is known as rotating. The magnitude of this field is always 1.5 ϕm, where ϕm is the maximum flux that can come from any phase.

The three-phase stator rotates at a synchronous speed of n_s and is surrounded by a spinning magnetic field when driven by a three-phase supply. This field passes through the air gap and cuts the stationary rotor conductors. The relative speed differential between the static rotor and the revolving flux induces electromotive forces in the rotor conductors. The rotor circuit is shorted, which causes current to start flowing in the rotor conductors. Again, these conductors are located in the magnetic field produced by the stator. The rotor conductors are thus subjected to mechanical force.This force tends to drive the rotor in the same direction as the revolving field by producing a torque. This is supported by Lenz's law, which states that rotor currents will tend to flow in the opposite direction from the force that is generating them. The rotor currents are now being generated by the relative speed of the motionless rotor conductors and the spinning field. The rotor thus begins running in the same direction as the stator field and attempts to catch it in order to decrease this speed. It is obvious that the stator field speed n_s is always greater than the rotor speed n. [7]

2.2 Induction Motor Speed Regulation

Induction motor speed control is more challenging than DC motor speed control, particularly when the same level of precision is needed. The complex power converters that supply this motor and the mathematical complexity of the induction machine's model are the main reasons behind this. Induction motor drivers with varying speeds employ various control strategies.

2.2.1 Scalar Control

Only the magnitude variation of the control variables is involved in the scalar control approach; the coupling impact between these variables is not taken into consideration. While the frequency or slip of a machine can be regulated to control the torque, the machine's voltage can be controlled to control the flux. Due to the coupling effect of torque and flux, which are functions of voltage or current and frequency, scalar controlled drives are simple to design but perform slowly. Because of the large drop in the stator impedance, there is also a significant torque drop at lower speeds. [8]

One simple scalar control technique is variable voltage variable frequency (v/f). The machine's performance under v/f control is enhanced by a speed control with closed loop flux, which also removes flux drift brought on by line voltage change. Current and speed at the machine terminal can be used to estimate the flux feedback signal. [9]

2.2.2 Vector Control

The vector control approach takes into account the control variables' phase and magnitude variations. Similar to an independently stimulated dc motor, it is utilized to achieve decoupled control of flux and torque. This control technique enhances the motor's dynamic performance by overcoming the slow response that occurs with scalar control. Accurate flux estimation (for Direct Field Oriented Control) or slip frequency estimation (for Indirect Field Oriented Control) are necessary for Field Oriented Control to function. For precise assessment of these quantities, rotor speed is necessary, which calls for a speed sensor. However, this has an impact on the motor drive's complexity, cost, and size. [10]

2.2.3 Regulating Speed as a Process Control Technique

Let's think about commuting to work. An accident is likely to occur when traveling at full speed. Furthermore, if you just drive at a safe speed for the duration of the trip, it will take

a long time to reach your destination. As a result, adjusting the speed to match the route effectively lowers the time required to accomplish the process's objective while maintaining reliable operation.

The following are some advantages of process control that a changeable speed drive may offer:

- Enhanced Efficiency in Operations
- Controlling acceleration as an extra motivator.
- operational speed variations for every process.
- adjusts for changes in process parameters.
- allows for sluggish functioning for setup.
- enables precise placement.
- gives control over torque.

2.2.4 Techniques for Controlling Speed

The relationship between the synchronous speed—that is, the speed at which the rotating flux rotates—and the speed of an induction motor can be expressed mathematically as follows:

$$n = (1 - s)n_s \tag{2.1}$$

Also,

$$n_s = \frac{120f}{P} \tag{2.2}$$

This suggests that there are two fundamental methods for controlling speed, namely

(i) With slip-control, the synchronous speed is fixed.

(ii) Synchronous speed control.

The following strategies are produced by more stringent sorting:

- Changing Poles
- Voltage Control for the Startor
- Control of Supply Frequency
- Eddy Current Coupled
- Control of Rotor Resistance
- Recovery of Slip Power

2.2.5 Controlling Variable Frequencies Using Voltage Sources

By altering the supply frequency, synchronous speed (and consequently the motor speed) can be managed.induced voltage in stator $E_1 \alpha \phi f$, where f is the supply frequency and ϕ is the air-gap flux.

Ignoring the terminal voltage and the stator voltage loss, which is only about 10% of the supply voltage $V_1 \alpha \phi f$

Evidently, reducing the supply frequency without correspondingly reducing the terminal voltage causes the air-gap flux to increase, bringing the motor's operating point closer to saturation. The disadvantages associated with this growth are as follows:

- An increase in the magnetizing current that is noticeable.
- distortion of stator copper loss and line current.
- stator copper loss and core loss increase.
- Acoustic noise is introduced.

To avoid the aforementioned effects, variable frequency control below the rated frequency is frequently applied at rated air-gap flux. This is achieved by varying the frequency of the terminal voltage in a manner that maintains the (V/f) ratio at the rated value.

It is crucial to keep in mind that when the (V/f) ratio is constant, a motor produces a constant maximum torque. An ideal motor would have a fully inductive stator magnetic circuit that maintains a constant flux by establishing a constant V/f ratio. On the other hand, the real motor has resistance and a magnetizing inductance coupled in series. Because the stator resistance drop (r_1) will become more substantial than the stator reactance reduction (x_1), the maximum torque will be smaller in this region at low speeds and frequencies. Because the same maximum torque can be maintained by increasing the (V/f) ratio at low frequencies, the resulting $V - f$ relation curve has a nonzero V for zero f.

The Volts/Hertz speed control was chosen over alternative methods mainly because of a few important characteristics. First of all, it makes it possible to adjust speed during both motoring and braking operations, from zero to above base speed. Second, because operations during transients can be performed at full torque with reduced current, a respectable dynamic response is obtained. Furthermore, at all frequencies, the functioning is obviously limited between the maximum torque point and the synchronous speed. This limitation helps to reduce copper losses, which raises the power factor and improves efficiency.

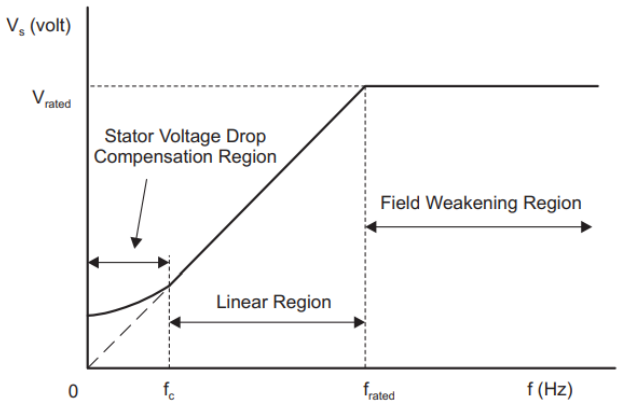


Figure 2.3: Variable Frequency Control

A cycloconverter or a voltage source inverter can be used as the variable frequency variable voltage source (VFVS) to accomplish this technique. The latter is only economical when utilized in high power drives because it uses a lot of thyristors. [11]

2.3 Volt/frequency Control Basics

The Volts/Hertz control method is essentially a scalar control approach, meaning that the only variables that are altered are the magnitudes of the control variables. Scalar control provides less dynamic performance than vector control, but it is easier to set up. The former cannot function at its peak when loads fluctuate dynamically. In variable-speed applications, scalar control systems can deliver satisfactory performance as long as a small variation in motor speed with loading is permitted. However, if fine control is required, a vector control system is essential.

Scalar control requires relatively little motor knowledge for frequency control. This control is hence commonly used. Nevertheless, the torque generated is load dependent as it is not directly controlled. The loophole is located here. Moreover, the delayed transient response of this kind of control is caused by the preset switching configuration of the inverter. The torque-speed characteristic curves make it clear that the torque at a given slip speed will remain constant when the air gap flux remains constant. Actually, this serves as the basis for the continuous Volts/Hertz regulation of an induction motor. This type of control can be implemented using either an open loop or a closed loop.

For real-world motor control applications, high dynamic performance is not usually required as long as the speed can be efficiently adjusted across the whole range. This enables the use of a sinusoidal steady state induction motor model, where the magnitude of the stator flux is proportional to the ratio of the magnitude of the stator voltage to its frequency. If this ratio is

maintained, the motor torque will only be reliant on the slip frequency because the stator flux will stay constant.

2.4 Proportional Integral(PI) Controller

The proportional-integral (PI) controller, a feedback control technique, is frequently used in three-phase induction motor control. As a component of a feedback loop, it regulates the motor's speed, current, and other properties in line with preset criteria.

Based on the current error—the discrepancy between the intended value (set point) and the actual value (measured value, like motor speed)—the proportional term of the PI controller produces an output. A more substantial correction step is required for a larger error. In the meantime, any steady-state error that can result from system biases, disturbances, or model mistakes is successfully eliminated by the PI controller's integral term (I), which takes into account the accumulation of previous errors over time. In the context of scalar control for induction motors, this integral term helps to gradually reduce any persistent deviations from the intended speed, ensuring precise and consistent speed management. The PI controller continuously assesses the difference between the desired and actual motor speeds by adjusting control signals using both proportional and integral terms. [12] This guarantees that, despite disruptions or uncertainties, the induction motor will precisely and effectively maintain the intended speed.

2.5 Adaptive Neuro-Fuzzy Inference System

An artificial neural network based on the Takagi–Sugeno fuzzy inference system is called an adaptive neuro-fuzzy inference system or adaptive network-based fuzzy inference system (ANFIS). The method was created around the beginning of the 1990s. [13] [3] It has the ability to combine the advantages of neural networks and fuzzy logic concepts into a single framework. In order to approximate nonlinear functions, its inference system translates into a collection of fuzzy IF-THEN rules with learning capabilities. [4] ANFIS is therefore regarded as a universal estimator. [13]One can utilize the best parameters derived from a genetic algorithm to use the ANFIS more effectively and optimally. [6] [8] It can be applied to intelligent energy management systems that are situationally aware. [10]

Fuzzy systems can execute reasoning using rules and represent extensive language knowledge, such as that provided by a human expert. Fuzzy systems don't, however, offer a way to automatically learn or adjust those rules. [2] Neural networks, on the other hand, are adaptable

systems that can be adjusted and trained using a collection of samples. After training, neural networks are able to generalize the learned information to handle new incoming data. However, extracting and comprehending such knowledge is extremely challenging. Stated differently, neural networks and fuzzy systems are complimentary paradigms. [13] Both static and dynamic neural networks and fuzzy logic controllers are examples of contemporary system analyses that have been effectively used in several real-world applications.When the system being studied is partially unknown and was previously thought to be nonlinear, these two methods are particularly helpful. A potent identification and control mechanism is created by combining the two approaches (neuro-fuzzy control systems). [6] Due to their many advantages over traditional computational systems, Fuzzy Inference Systems (FISs) and Artificial Neural Networks (ANNs) have garnered a lot of interest recently as potential contenders for revolutionary computational systems. In contrast to other traditional control techniques, ANNs and fuzzy logic control (FLC) are more model-free controllers, meaning they don't need a precise mathematical model of the system. [8] Neural networks and neuro-fuzzy modeling techniques have drawn a lot of interest for non-linear modeling. The intricacy and obscurity of their structures are the disadvantages, particularly when simulating intricate nonlinear systems. It has been shown that hybrid models may effectively represent a variety of industrial processes with nonlinearity, including fermentation, solid drying, heating and cooling, and continuous stirred tank reactors (CSTRs). [14] The temperature system is uncertain, nonlinear, and has a long delay time and time constant. Industries currently control temperature using the PID technology. PID control is a precise control; yet, it is challenging to self-tune the P, I, and D parameters, and the resulting control has significant time constants and overshoot. [15] In an expert system, fuzzy logic is utilized to convey uncertainty. The fuzzy controller problem is boiled down to finding the right set of IF-THEN rules, which a human expert can do. The ANFIS controller can create expert systems on its own by combining fuzzy logic and artificial neural networks. [16] Both compensating for plant parameter variations and adapting to changes in the process environment are beyond the capabilities of the fixed gain feedback controllers (PID). New resources for more effective control have been inspired by the introduction of fuzzy logic control, or FLC. For implementation, an a priori model of the process is not necessary. The ideal membership numbers in the conventional FLC are determined through time-consuming trial and error. Additionally, it takes a lot of time to transfer the range of input variables into the matching discourse universe using the traditional way. Artificial Neural Networks can be used to transfer the range of input variables into the matching discourse world. [5] Neural network applications for control systems have grown in significance. A popular gradient descent search method is the

back propagation neural network based on the extended delta learning algorithm. Although the back propagation algorithm has a number of drawbacks, including the inability to ensure convergence, its straightforward yet effective mathematical formula has made it the cornerstone of neuro-computing. The neural network must first learn the plant model before it can be utilized as a controller. The neural network can be trained using a variety of suggested learning designs. [3]

2.6 Controllers for Neural Networks and Fuzzy Logic

The fuzzy character of human thought serves as the foundation for the Artificial Intelligence (AI) class known as fuzzy logic. According to fuzzy set theory, it is commonly described as multi-valued control; an object's degree of membership in a given set might range from 0 (totally not in the set) to 1 (fully in the set). [6] Fuzzy logic control uses fuzzy rules or relations to linguistically describe human competence in process control. An inference mechanism uses this knowledge base to determine the control actions, together with some understanding of the process states. [17] In [8] For the three-phase induction motor, a fuzzy logic speed controller has been designed. A fuzzy logic controller's (FLC) primary benefits are its ability to manage nonlinearity and its independence from the plant model. [18].

Nevertheless, it is important computational cost associated with converting the verbal rules into control actions. By integrating Artificial Neural Networks (ANN) into the controller, this limitation is addressed.

Artificial neural networks (ANNs) are computer programs or electronic circuits that can mimic human thought processes. It performs well on image processing and pattern recognition tasks that are difficult for a conventional digital computer to handle. ANNs are frequently taught to learn and remember by giving them examples of input/output connection patterns, just like the human brain does. The plant's control action is formulated by ANN using nonlinear methods, which are easily comprehensible due to their mathematical structure. Additionally, it can learn any control relationship of practical importance through training, approximate any control relationship closely, and produce an output even in cases when the inputs do not come from the taught set. In [17], a FLC is implemented using ANN.

2.7 Conventional Controllers

Since 250 B.C., the Greeks have employed feedback control systems to regulate water level; these systems functioned similarly to the level regulator found in contemporary flush toilets. Steam power was developed in large part because of the fly-ball governor, which James Watt introduced to the steam engine for the first time in 1788.

The invention of high-gain, operational amplifiers in the 1930s, which are widely utilized in electronic equipment, required feedback regulation. Three-mode controllers with proportional, integral, and derivative (PID) feedback control action were first made available for purchase in the 1930s.Pneumatic PID controllers gained popularity in industry in the 1940s, and electronic equivalents appeared in the 1950s. In the late 1950s and early 1960s, reports of the first computer control applications in the process industries appeared. Digital hardware has been widely employed since the 1980s and has significantly improved process control. In essence, there are three modes for continuous controllers: derivative, integral, and proportional. Each has benefits and drawbacks of their own. Composite conventional controllers also exist, including proportional plus integral and derivative and the well-known controller proportional integral derivative (PID).

PI or PID controllers are used in numerous sectors to regulate the speed of induction motors. PID controllers are used in the real world to control the speed of three-phase induction motors for practical reasons. As more loads applied to the system the controller must be too sharp to track the desired output. This means when the load is larger the controller must not lose its operating efficiency. [11] By evaluating the error and its rate of change, the Fuzzy Logic Controller (FLC) operates with a particular focus on the tuning of the PI controller gains. After that, it uses a set of language rules to evaluate these inputs and provide appropriate controller gain modifications. The Permanent Magnet Synchronous Motor's (PMSM) control accuracy, responsiveness, and stability are all enhanced by this adaptive process. Integrating FLC into PMSM control systems for tuning PI controller gains offers a sophisticated and efficient approach to achieving optimal motor performance. [5] FLC improves the control system's precision, stability, and adaptability by utilizing language rules and real-time error analysis. This ultimately leads to better PMSM functioning and control under a range of circumstances. [2]

Chapter 3

System Design and Modeling of The Induction Motor

3.1 Transformation of Coordinate Frames

One of the most important steps in mathematically simulating the dynamics of a three-phase induction motor is the transformation of the coordinate frame. The equations describing the behavior of the motor are greatly simplified by the choice of the coordinate frame for analysis. In order to improve the performance of the control strategies, cross couplings must be eliminated during the design of the scalar control method. [19] The two most popular methods for transforming coordinates—Park transformation and Clarke transformation—will be discussed in this subsection.

3.1.1 The Transformation of Clarke

The three-phase stator currents are converted from the stationary reference frame $(\alpha - \beta)$ to the two-phase stationary reference frame (d-q) using the Clarke transformation. The following are the transformation equations:

$$\begin{bmatrix} I_d \\ I_q \end{bmatrix} = \begin{bmatrix} Cos(\theta) & Sin(\theta) \\ -Sin(\theta) & Cos(\theta) \end{bmatrix} \begin{bmatrix} I_\alpha \\ I_\beta \end{bmatrix} \tag{3.1}$$

where I_d and I_q are the two-phase stator currents in the stationary reference frame, $I\alpha$ and $I\beta$ are the three-phase stator currents, and θ is the electrical angle of the rotor.

3.1.2 Parks Transformation

The Park transformation is used to convert the stationary frame (d-q) into the rotating reference frame $(\alpha - \beta)$ such that the two-phase Stator currents match the rotor flux. The transformation equations are expressed as follows:

$$\begin{bmatrix} I_\alpha \\ I_\beta \end{bmatrix} = \begin{bmatrix} Cos(\theta) & -Sin(\theta) \\ Sin(\theta) & Cos(\theta) \end{bmatrix} \begin{bmatrix} I_d \\ I_q \end{bmatrix} \tag{3.2}$$

where $I\alpha$ and $I\beta$ are the two-phase stator currents in the stationary reference frame, $I-d$ and I_q are the three-phase stator currents, and θ is the electrical rotor angle.

3.1.3 Clarke Park Transformations

Different reference frames can be used to understand a three-phase motor's dynamics from different angles. [20] The three-phase stationary reference frame, often known as the ABC-frame, is the most widely used frame. The three axes of this frame, a, b, and c, correspond to the three electrical phases in the stator. The frame is physically aligned with the motor. Within this reference frame, voltage and current measurements are also made. It can be challenging to determine the three-phase frame's size and angle, so you must convert it to a two-phase system. The inverses of the Clarke and Park Transformations have been proposed as a solution to this issue.

These modifications make it possible to transform the stator's three-phase currents into the spinning frame of the rotor. [15] A two-axis orthogonal stationary reference frame is obtained by converting three-phase data using the Clarke Transformation. Even when these quantities remain in a fixed frame, the rotor's reference frame continues to rotate.The Park Transformation then transforms these values into an orthogonal reference frame that include the quadrature and direct axes. [21] The combined transformations of Park and Clarke are expressed as follows:

$$\begin{bmatrix} I_d \\ I_q \end{bmatrix} = 2/3 \begin{bmatrix} Cos(\theta) & Cos(\theta - 2\pi/3) & Cos(\theta + 2\pi/3) \\ Sin(\theta) & Sin(\theta - 2\pi/3) & Sin(\theta + 2\pi/3) \end{bmatrix} \begin{bmatrix} I_a \\ I_b \\ I_c \end{bmatrix} \tag{3.3}$$

where I_d and I_q are the direct axis and quadrature axis components, respectively, and θ represents the rotor's angular position. Inverse conversion:

$$\begin{bmatrix} I_a \\ I_b \\ I_c \end{bmatrix} = \begin{bmatrix} Cos(\theta) & -Sin(\theta) \\ Cos(\theta - 2\pi/3) & -Sin(\theta - 2\pi/3) \\ Cos(\theta + 2\pi/3) & -Sin(\theta + 2\pi/3) \end{bmatrix} \begin{bmatrix} I_d \\ I_q \end{bmatrix} \tag{3.4}$$

The diagram below shows the specifics of these reference frame modifications.

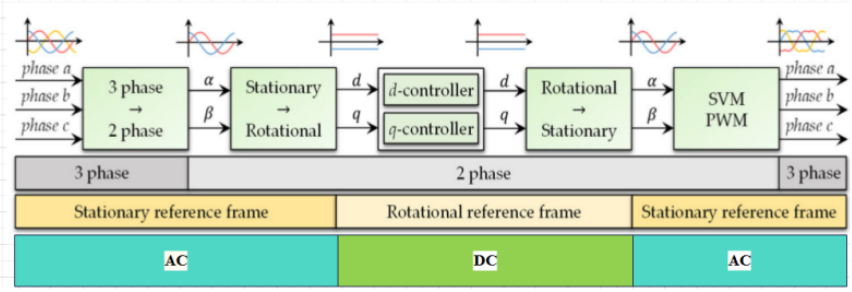


Figure 3.1: Field-Oriented Control Transformations

3.1.4 Volt/Hertz(Scalar Control)

The V/Hz control controls the speed of the induction motor by altering the frequency and stator voltages in a way that keeps the air gap flux at the required level in a steady-state. Since this approach just considers the steady state dynamic, it is commonly referred to as scalar control.By examining the analogous simplified steady state circuit shown in figure 3.1, the workings of this approach can be observed. This figure places the magnetizing inductance, which indicates the amount of air gap flux, in front of the total leakage inductance ($Ll = Ll_s + Ll_r$) and assumes that the stator resistance (R_s) is zero. It also embeds the stator leakage inductance (Ll_s) into the (also known as stator) rotor leakage inductance (Ll_r). Consequently, magnetizing current roughly equivalent to the stator voltage to frequency ratio can be used to generate the air gap flux. Its steady-state analytical phasor equation is as follows:

$$\bar{I_m} = \frac{\bar{V_s}}{j\omega L_m} \tag{3.5}$$

When the induction motor operates in the linear magnetic zone, the L_m stays constant. The following is an expression of the equation in terms of magnitude:

$$I_m = \frac{\Lambda_m}{L_m} \simeq \frac{V_s}{(2\pi f)L_m} \tag{3.6}$$

where the phasor representation is $\tilde{V}$," and the stator voltage and flux magnitudes are V and Λ, respectively. According to the last equation, torque becomes unaffected by the frequency of the

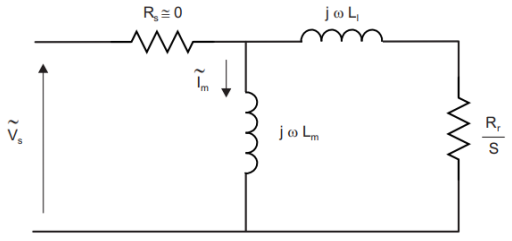


Figure 3.2: Simplified Induction Motor Steady-State Equivalent Circuit

supply and flux stays constant if any change in f results in the ratio V/f remaining constant. At each variable speed, the ratio of Vs/f would likewise be constant in order to maintain Λ_M. The stator voltages must therefore be raised in accordance with the speed in order to maintain the constant Vs/f ratio. But the true speed is not the frequency (or synchronous speed) because of a slide in relation to the driving force. The slip is negligible and the speed is almost equal to the synchronous speed at no-load torque. Consequently, the simple open-loop Vs/f (or V/Hz) system cannot effectively control the speed when load torque is present. Using the speed data, adding slip compensation to the system is simple. Figure 3.3 depicts the closed-loop V/Hz system with a speed sensor.

Actually, the rated values of these variables are typically used to determine the stator voltage

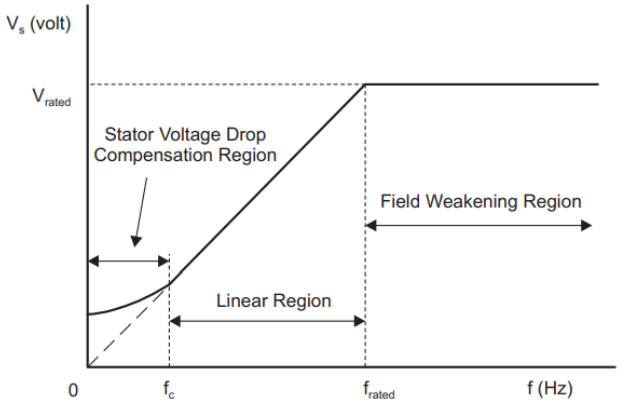


Figure 3.3: Stator Voltage versus Frequency Profile under V/Hz Control

to frequency ratio. The figure below shows the usual V/Hz profile. The V/Hz profile essentially has three speed ranges, which are as follows:

- It is impossible to overlook the voltage loss over the stator resistance at $0 - f_c$ Hz, and The V_s must be raised to account for it. The V/Hz profile is not linear, as this shows. The steady state equivalent circuit with $R_s \neq 0$ can be used to analytically estimate the proper stator voltages and the cutoff frequency (f_c).
- The constant ratio connection is followed at f_c - f_{rated} Hz. The air gap flow quantity is truly represented by the slope, as shown in Equation.
- The constant Vs/f ratio cannot be met at higher f_{rated} Hz because the stator voltages would be limited at the rated value to prevent insulation breakdown at stator windings. The air gap flux would thus diminish, which would unavoidably cause the generated torque to decrease. The term "field weakening region" is typically used to describe this area. This is avoided by violating the constant V/Hz concept at such frequencies.

Due to the constant maintenance Considering the stator flux (without regard to changes in the

frequency of the supply), Only the slip speed affects the torque produced. The figure 3.4 shows this. An AC induction motor's torque and speed can be controlled using the constant V/Hz concept by varying the slip speed. An AC induction motor's speed can be controlled both

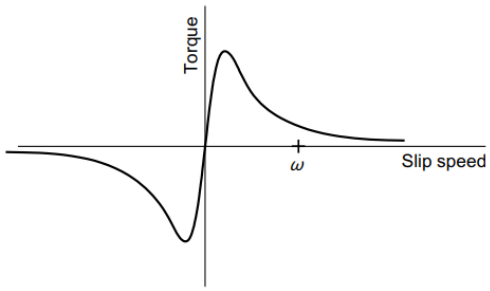


Figure 3.4: An Induction Motor with Constant Stator Flux: Torque against Slip Speed

open-loop and closed-loop using the constant V/Hz method. The application of open-loop speed control in fan, blower, and applications for HVAC (heating, ventilation, and air conditioning) where precise speed response is not an issue. Here, the goal speed and the presumption that the motor will nearly exactly match its synchronous speed are used to compute the supply frequency. It is considered acceptable that the motor slide resulted in the speed error.

Let's conclude that the voltage/frequency method is the most straightforward controller approach. Volts per Hz is the most widely used technique for controlling the speed of an induction motor. The Volt/Hz ratio is precisely equal to the torque generated by the IM. When the voltage and frequency are allowed to fluctuate while maintaining a constant percentage, the torque produced by IM will remain constant across the whole speed range. There are always two factors that the scalar control method depends on. Altering the input frequency will change the impedance, but it will also change the speed. A change in the impedance is what causes the current to increase or decrease.A low current causes the motor's torque to drop. A decrease in frequency or an increase in voltage may cause the coils to burn or get saturated. To prevent these issues, it is essential to alter the frequency and voltage at the same time.

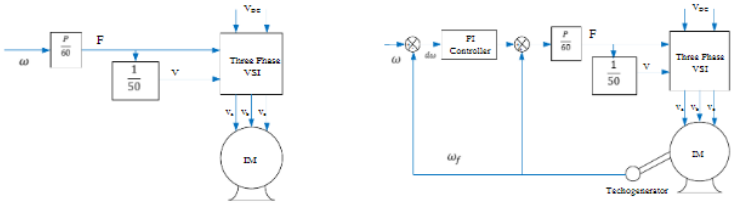


Figure 3.5: Open Loop and Closed Loop

3.2 Mathematical Modeling

Assumptions A few presumptions are necessary in order to derive the mathematical model of a three-phase induction motor. Below is a list of their names.

1. consistent air gap
2. Construction in the style of a squirrel cage
3. balanced stator and rotor windings that are spread sinusoidally.
4. Parameter change and saturation are disregarded.

The induction motor's steady state concept and analogous circuit can be used to study the machine's operation in a stable state. This suggests that when the load or stator frequency varies, all electrical transients are ignored. Applications utilizing variable-speed drives exhibit these variations. Because switch ratings and filter diameters are constrained, instead of utility sources, variable-speed drives are usually fed from finite sources. They are therefore unable to supply a sizable amount of intermittent electricity. To ascertain the excursions of currents and torque in the converter and motor, as well as the suitability of the converter switches and converters for a given number of motors, we must thus investigate the dynamics of converter fed variable speed drives.

3.2.1 Equivalent Circuit of Induction Motor

The dynamic behavior of an induction motor is described by time-varying equations for voltage and torque. That might be a little complicated, but it can successfully solve such differential equations. The machine's voltage equations can be simplified by changing a few variables and removing every inductance that changes over time as a result of electric circuits moving relative to one another.

Generalized Concept in Any Frame of Reference

Observer platforms and reference frames are similar in that they both greatly simplify the system equation and offer a unique viewpoint on the system being studied. For example, the system variables are sinusoidal, but for control purposes, it is better to have them as dc numbers. This might be accomplished, for example, by using a reference frame that revolves at the sinusoidal variable's angular speed. By moving at an angle equal to the sinusoidal supply's angle frequency, the reference frames reduce the differential speed between them to zero, causing sinusoidal to behave like a DC signal from them. After changing that lane, DC

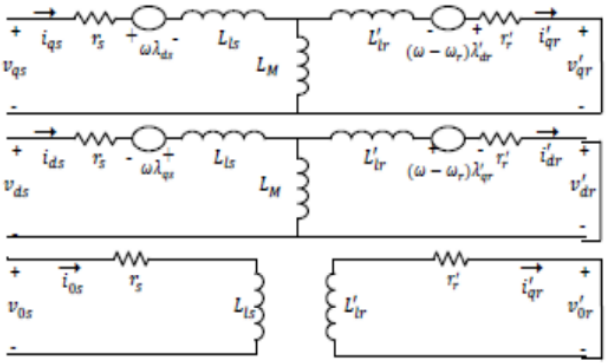


Figure 3.6: A circuit diagram of an induction motor with equivalent

values are the only way to specify the operating point, making it simpler to turn a nonlinear equation into a tiny signal equation that produces a system that is linearized around the place of operation. The use of reference frames has numerous benefits. It is preferable to derive a general transformation for each rotating reference frame rather than deriving a distinct transformation for each reference frame. Any unique reference frame model can be created by altering the frame position and speed in the general reference model.

Transformation from Three to Two Phases

The induction motor's dynamic model can be created by analyzing the equivalency between two- and three-phase machines. To illustrate the equivalency, comparable current magnitudes and the amount of MMF produced in two- and three-phase windings are employed. With equal current magnitudes and N_s turns per phase for each three-phase winding, the two-phase winding will have $3N_s/2$ turns per phase for MMF equivalence. To find the d and q axis MMF, the three-phase MMF can be solved along those axes. The common term maintains present equality in equations by removing the number of turns on both sides. When an induction machine is in a balanced state, its three-phase stator voltages look like this:

$$V_a = 2V_{rms}sin(\omega t); V_b = 2V_{rms}sin(\omega t - 2\pi/3); V_c = 2V_{rms}sin(\omega t + 2\pi/3) \tag{3.7}$$

The three line voltages in this case are V_a, V_b, and V_c. α, β and abc have the following connection.

$$\begin{bmatrix} V_\alpha \\ V_\beta \end{bmatrix} = 2/3 \begin{bmatrix} 1 & 1/2 & -1/2 \\ 0 & \sqrt{3}/2 & -\sqrt{3}/2 \end{bmatrix} \begin{bmatrix} V_a \\ V_b \\ V_c \end{bmatrix} \tag{3.8}$$

Next, the direct and quadrature axes' voltages:

$$\begin{bmatrix} V_d \\ V_q \end{bmatrix} = \begin{bmatrix} Cos(\theta) & sin(\theta) \\ -Sin(\theta) & Cos(\theta) \end{bmatrix} \begin{bmatrix} V_\alpha \\ V_\beta \end{bmatrix} \tag{3.9}$$

The following transformation is finally used to calculate the current in the rotor and stator instantaneous values on three-phase system:

$$\begin{bmatrix} i_\alpha \\ i_\beta \end{bmatrix} = \begin{bmatrix} Cos(\theta) & -sin(\theta) \\ Sin(\theta) & Cos(\theta) \end{bmatrix} \begin{bmatrix} i_d \\ i_q \end{bmatrix} \tag{3.10}$$

$$\begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} = 2/3 \begin{bmatrix} -1/2 & -\sqrt{3}/2 \\ -1/2 & -\sqrt{3}/2 \end{bmatrix} \begin{bmatrix} i_\alpha \\ i_\beta \end{bmatrix} \tag{3.11}$$

Simulink Model and Flux Linkage Equations

Flux connections can be used as variables to represent the dynamic model of an induction motor in any reference frame. Dynamic equations can be solved considerably more easily with analog and hybrid computers by lowering a number of variables. The flux connections remain continuous even when the voltages and currents are irregular. Thus, benefit of numerically stable differentiation of these variables is offered. Additionally, the Motor drives employ the flux linkages depiction to illustrate the decoupling of the torque and flux channels in synchronous and induction machines. The following is an expression for the different flux linkage equations:

$$\frac{d\psi_{qs}}{dt} = \omega_b \left[V_{qs} - (\omega_e/\omega_b)\psi_{ds} - (R_s/X_{ls})(\psi_{mq} - \psi_{qs})\right] \tag{3.12}$$

$$\frac{d\psi_{ds}}{dt} = \omega_b \left[V_{ds} - (\omega_e/\omega_b)\psi_{qs} - (R_s/X_{ls})(\psi_{md} - \psi_{ds})\right] \tag{3.13}$$

$$\frac{d\psi_{qr}}{dt} = \omega_b \left[V_{qr} - ((\omega_e - \omega_r)/\omega_b)\psi_{dr} - (R_r/X_{lr})(\psi_{mq} - \psi_{qr})\right] \tag{3.14}$$

$$\frac{d\psi_{dr}}{dt} = \omega_b \left[V_{dr} - ((\omega_e - \omega_r)/\omega_b)\psi_{qr} - (R_r/X_{lr})(\psi_{md} - \psi_{dr})\right] \tag{3.15}$$

where

$$\psi_{mq} = X_{ml}\left[\psi_{qs}/X_{ls} + \psi_{qr}/X_{lr}\right]\right] \psi_{md} = X_{ml}\left[\psi_{ds}/X_{ls} + \psi_{dr}/X_{lr}\right] \tag{3.16}$$

After that, by changing the flux linkage values to determine the currents,

$$i_{qs} = 1/X_{ls}(\psi_{qs} - \psi_{mq})X_{ml} = 1/\left[1/X_m + 1/X_{ls} + 1/X_{lr}\right] \tag{3.17}$$

$$i_{qr} = 1/X_{lr}(\psi_{qr} - \psi_{mq}) \tag{3.18}$$

$$i_{ds} = 1/X_{ls}(\psi_{ds} - \psi_{md}) \tag{3.19}$$

$$i_{dr} = 1/X_{lr}(\psi_{dr} - \psi_{md}) \tag{3.20}$$

The rotor speed and electromagnetic torque are calculated utilizing the previously mentioned formulas.

$$T_e = (3/2)(P/2)(1/\omega_b)(\psi_{ds}i_{qs} - \psi_{qs}i_{ds}) \tag{3.21}$$

$$\omega_b = \int (P/2J)(T_e - T_l) \tag{3.22}$$

with P = number of the machine's poles.

T_e = torque generated by electromagnetic forces.

Using the so-called phase variables—the stator currents i_{as}, i_{bs}, and i_{cs}, the angular displacement θ between the stator and rotor windings, and the rotor speed ω_m—the first step in developing a mathematical model of an induction machine is to describe it as a coupled stator and rotor polyphase circuit. The magnetic coupling is represented by the inductance matrix as a function of position. The matrix expression for the machine equations is developed using the Simulink language.

After converting the initial The additional variables for voltages, currents, and fluxes can then be seen as 2-D space vectors in the common k or dq frame, which corresponds to the stator and rotor abc frames of reference. Regardless of position, inductance becomes constant on common frame. The figure 3.7, which displays multiple coordinate systems or reference frames, makes this clear. The triplet [A_s B_s C_s] indicates a stator-connected three-phase system., In contrast, a corresponding two phase system is connected to the pair [a_s b_s]. You can choose from the following dq frames:

A) The stator frame in which $\omega_k = 0$

B) In the $\omega_k = \omega_m$ for the rotor frame

C) The synchronous frame is linked to stator excitation's frequency ω_s, which may fluctuate over time.

D) Rotor flux frame where the direction of the rotor flux vector is aligned with the d-axis.

The Common dq frame selection is typically determined by the machine's architecture and the symmetry restrictions placed by the excitation. Since all signals look as continuous dc in steady state, the synchronous frame is more practical, however any frames can be used because of

balanced sinusoidal stimulation and the precise symmetry seen in a three-phase induction motor. When asymmetry occurs, the asymmetrical component is attached to the common frame; in this case, the stator frame will resemble an induction motor with asymmetric stator windings or an uneven excitation. The machine dynamic equations appear as differential equations in the common dq frame, with constant coefficients (i.e., independent of rotor location) and nonlinearity limited to products of variables related to speed voltages and torque components. ω represents a frame's rotational speed or angular frequency (in electrical rad/s) relative to the stationary rotor. By integrating speed over time, the angular position determined, that is:

$$\theta = \int \omega dt \tag{3.23}$$

By defining a steady-state operating condition, initial conditions are set. In the simulation of a motor starting, we shall encounter the most basic scenario in which there are no initial circumstances. Generally speaking, obtaining a given operating state necessitates running the simulation for a duration determined by the initial circumstances' starting configuration. To ensure that the values are equivalent, power electronic input signals are often repeated at the start and finish of each switching cycle until a steady-state condition is achieved. Ultimate condition can be saved and employed in a refurbished simulation as the beginning state, where time sequencing has been applied to the input, when a steady-state condition is achieved.

3.3 Space Vector Model of Induction Machine(SI Units)

Figure 3.7 shows the analysis of induction motor reference frame.

3.3.1 Electrical System Equations

$$\bar{V_s} = R_s \bar{i_s} + \frac{d\bar{\lambda}_s}{dt} + \omega_k M \bar{\lambda_s} \tag{3.24}$$

$$\bar{V_r} = R_r \bar{i_r} + \frac{d\bar{\lambda}_r}{dt} + (\omega_k - \omega_m) M \bar{\lambda_r} \tag{3.25}$$

in which the space vector $\bar{f} = \begin{bmatrix} f_d & f_q \end{bmatrix}$ and the $\pi/2$ rotation operator $M = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$

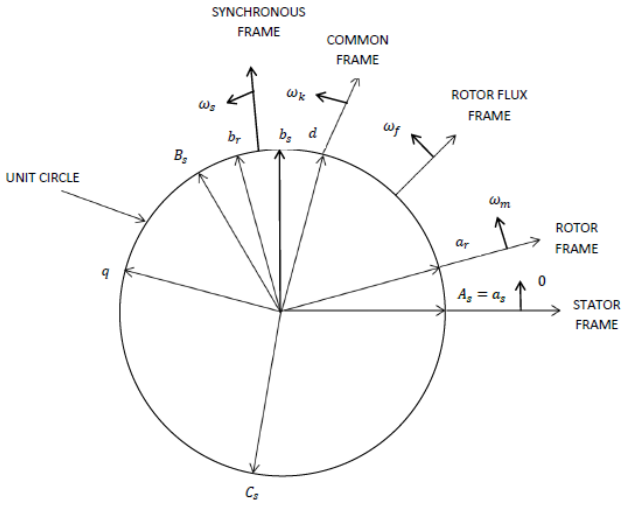


Figure 3.7: Reference frames in induction machine analysis

3.3.2 Flux Linkage Current Relations

$$\bar{\lambda_s} = L_s\bar{i_s} + L_m\bar{i_r} \tag{3.26}$$

Or

$$\bar{\imath_s} = \Gamma_s\bar{\lambda_s} - \Gamma_m\bar{\lambda_r} \tag{3.27}$$

Also

$$\bar{\lambda_r} = L_m\bar{i_s} + L_r\bar{i_r} \tag{3.28}$$

Or

$$\bar{\imath_s} = \Gamma_s\bar{\lambda_s} - \Gamma_m\bar{\lambda_r} \tag{3.29}$$

Where

$$L_s = L_m + L_{sl} \tag{3.30}$$

$$L_r = L_m + L_{rl} \tag{3.31}$$

$$\Gamma_s = L_r/\Delta \tag{3.32}$$

$$\Gamma_r = L_s/\Delta \tag{3.33}$$

$$\Gamma_m = L_m/\Delta \tag{3.34}$$

Where

$$\Delta = L_mL_{sl} + L_mL_{rl} + L_{sl}L_{rl} \tag{3.35}$$

3.3.3 Equations for Mechanical Systems

$$T_e = J \int \frac{d\omega_m ec}{dt} + B_m \omega_m ec + T_L \tag{3.36}$$

Where

$$T_e = k(\bar{\lambda}_s \otimes \bar{i}_s) = k(M\bar{\lambda}_s.\bar{i}_s) = k(\bar{\lambda}_{ds}\bar{i}_{qs} - \lambda_{qs}\bar{i}_{ds}) \tag{3.37}$$

or

$$T_e = k(\bar{i}_r \otimes \bar{\lambda}_r) = kL_m(\bar{i}_r \otimes \bar{i}_s) = kL_m/L_r(\bar{\lambda}_r \otimes \bar{i}_s) = k\Gamma_m(\lambda_r \otimes \lambda_s) \tag{3.38}$$

And $\omega_m ec = 2/\mathrm{P}\ \omega_m$, k = (3/2)(P/2)

3.4 Pulse Width Modulation(PWM)

DC-AC converters are called inverters. The function of an inverter is to convert a DC input voltage into a symmetrical AC output voltage with the proper amplitude and frequency. Voltage and current aberrations caused by low order harmonics are a common problem in the field of power electronics. Rapid advancement of semiconductor power devices and associated controllers has led induction motor drives can be used in a variety of variable speed drive applications, almost entirely replacing DC drives. Using PWM control techniques is one way to lessen low level harmonics in high power converters.The two most popular PWM strategies, SPWM and SVPWM, have been thoroughly explored in this study. MATLAB Simulink is used to model the six pulse inverter circuit, and harmonics in the controls that these two ways govern are checked. A PWM pulse can be produced by comparing a sinusoidal reference signal with a triangle wave and varying the pulse width in proportion to the sinusoidal reference signal's amplitude as measured at the pulse's center.

Sinusodial Pulse Width Modulation(SPWM)

The amplitude modulation index of the fundamental frequency component,

$$M_a = V_m/V_{cr} \tag{3.39}$$

Typically, V_m is varied while V_{cr} remains constant in order to modify M_a is the amplitude modulation index. The frequency modulation index,

$$M_f = f_{cr}/f_m \tag{3.40}$$

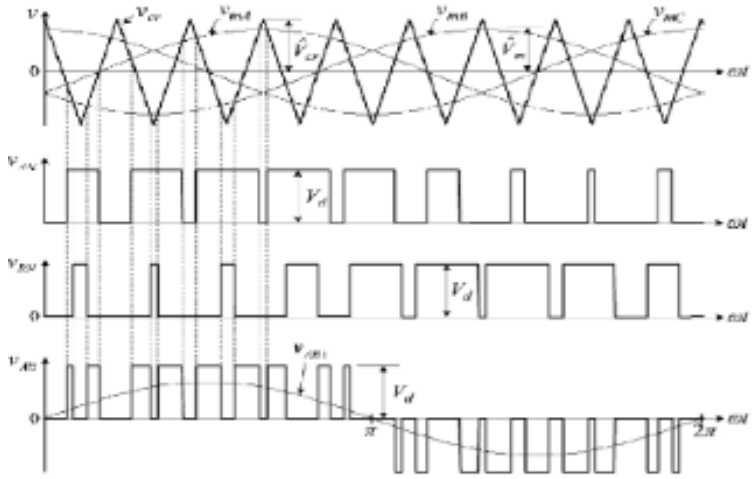


Figure 3.8: Generation of Sinusoidal PWM signal

The inverter's output frequency can be found using the sinusoidal reference signal's frequency. By regulating the inverter output's RMS voltage, which is established by the sinusoidal reference signal's peak amplitude, the modulation index M can be changed.

3.5 Three Phase Induction Motor Dynamic Model Verification

Simulation(Open loop)

V/f control circuit diagram for a three-phase induction motor that uses the SPWM approach.

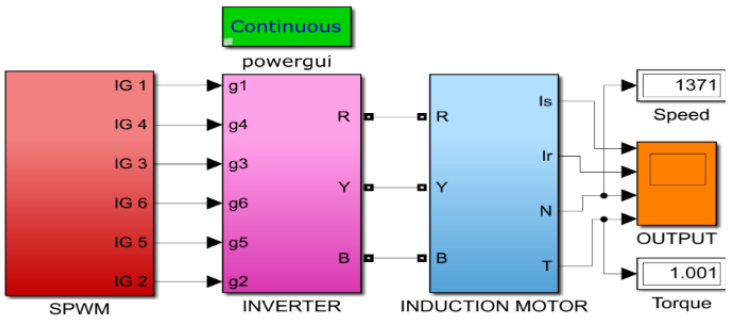


Figure 3.9: Open Loop

Simulation(Closed loop)

An induction machine's mathematical model can be obtained by first characterizing it in terms of the so-called phase variables, V_{is}, as a polyphase circuit with a coupled stator and rotor. the stator currents i_{as}, i_{bs}, and i_{cs}, the rotor speed ω_m, the angular displacement θ between the stator and rotor windings, and the rotor currents i_{ar}, i_{br}, and i_{cr}. Magnetic coupling is repre-

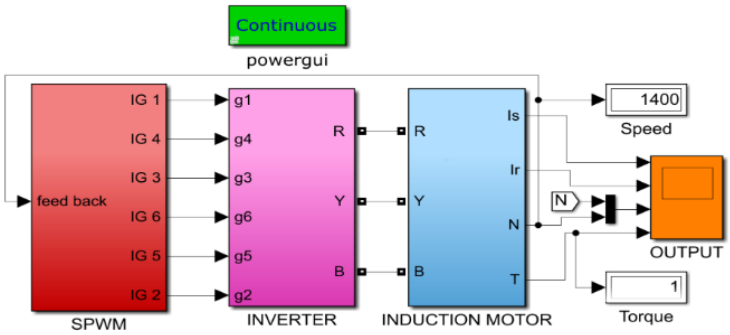


Figure 3.10: Closed Loop

sented by the inductance matrix, which varies with position.

After converting the original ABC frames of reference for the stator and rotor, the new variables for voltages, currents, and fluxes can then be viewed as 2 D space vectors in the common k or dq frame. Regardless of position, the inductances become constant in this common frame. According to the two axis theory of electric machines, the induction machine can be represented mathematically. It is common practice to use the depiction of the two-phase signal to decrease the induction machine's differential equations are complicated. Additionally, these equations can be made simpler and less difficult by eliminating from the machines voltage calculations all time varying inductance's caused by electric circuits in relative motion.An induction motor's dynamic behavior is described by the time-varying voltage and torque equations.

3.5.1 Dynamic Modeling

An induction motor with three phases uses many currents when it is started and during other brief activities. This can cause power systems to experience harmonics, oscillatory torques, and voltage drops.

- It is more precise and dependable to use the dq axis model to study such issues.
- The dynamic model is constructed using a two-phase motor in both direct and quadrature rotation.
- This approach has the advantage of being conceptually simple since it uses two sets of windings—one on the rotor and one on the stator.

$$V_{qs} = i_{qs} r_s + \frac{d\lambda_{qs}}{dt} + \omega \lambda_{ds} \tag{3.41}$$

$$V_{qs} = i_{qs} r_s + \frac{d(i_{qs} L_s + L_m i_{qr})}{dt} + \omega_e (i_{ds} L_s + l_m i_{dr} \tag{3.42}$$

$$\frac{d(i_{qs} L_s)}{dt} = V_{qs} - i_{qs} r_s - L_m \frac{di_{qr}}{dt} - \omega_e i_{ds} L_s - \omega_e L_m i_{dr} \tag{3.43}$$

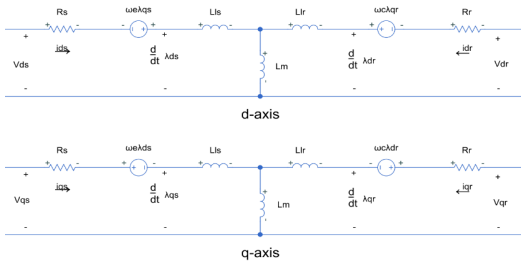


Figure 3.11: d and q axis current transformations

- $$i_{qs} = \int (\frac{V_{qs}}{L_s} - \frac{i_{qs}r_s}{L_s} - \frac{L_m P i_{qr}}{L_s} - \omega_e i_{ds} - \frac{\omega_e L_m i_{dr}}{L_s}) \tag{3.44}$$

$$V_{ds} = r_s i_{ds} + \frac{d\lambda_{ds}}{dt} - \omega\lambda_{qs} \tag{3.45}$$

$$V_{ds} = r_s i_{ds} + \frac{d(i_{ds}L_s + L_m i_{dr})}{dt} - \omega_e(i_{qs}L_s + l_m i_{qr} \tag{3.46}$$

$$\frac{d(i_{ds}L_s)}{dt} = V_{ds} - r_s i_{ds} - L_m \frac{di_{dr}}{dt} + \omega_e i_{qs} L_s + \omega_e L_m i_{qr} \tag{3.47}$$

- $$i_{ds} = \int (\frac{V_{ds}}{L_s} - \frac{i_{ds}r_s}{L_s} - \frac{L_m P i_{dr}}{L_s} + \omega_e i_{ds} - \frac{\omega_e L_m i_{qr}}{L_s}) \tag{3.48}$$

$$V_{qr} = r_r i_{qr} + (\omega_e - \omega_r)\lambda_{dr} + \frac{d(\lambda_{qr}dt}{} \tag{3.49}$$

$$V_{qr} = r_r i_{qr} + \frac{(i_{qr}L_r + L_m i_{qs}}{dt} + \omega_c(i_{dr}L_r + L_m i_{ds} \tag{3.50}$$

Where

$$(\omega_e - \omega_r) = \omega_c \tag{3.51}$$

$$\frac{d(i_{qr}L_r)}{dt} = V_{qr} - r_r i_{qr} - L_m \frac{di_{qs}}{dt} - \omega_c i_{dr} L_r - \omega_c L_m i_{ds} \tag{3.52}$$

- $$i_{qr} = \int (\frac{V_{qr}}{L_r} - \frac{i_{qr}r_r}{L_r} - \omega_c i_{dr} - \frac{\omega_c L_m i_{ds}}{L_r}) \tag{3.53}$$

$$V_{dr} = r_r i_{dr} + \frac{d(\lambda_{dr}dt}{-}(\omega_e - \omega_r)\lambda_{qr} \tag{3.54}$$

$$V_{dr} = r_r i_{dr} + \frac{(i_{dr}L_r + L_m i_{ds}}{dt} - \omega_c(i_{qr}L_r + L_m i_{qs}) \tag{3.55}$$

*

$$i_{dr} = \int (\frac{V_{qr}}{L_r} - \frac{i_{dr}r_r}{L_r} - \frac{L_m P i_{ds}}{L_r} + \omega_c i_{qr} + \frac{\omega_c L_m i_{qs}}{L_r}) \tag{3.56}$$

*

$$\omega_r = \int (T_e - T_l)\frac{P}{2J} \tag{3.57}$$

*

$$T_e = 3/2P/2L_m 9 i_{qs} i_{dr} - i_{ds} i_{qr}) \tag{3.58}$$

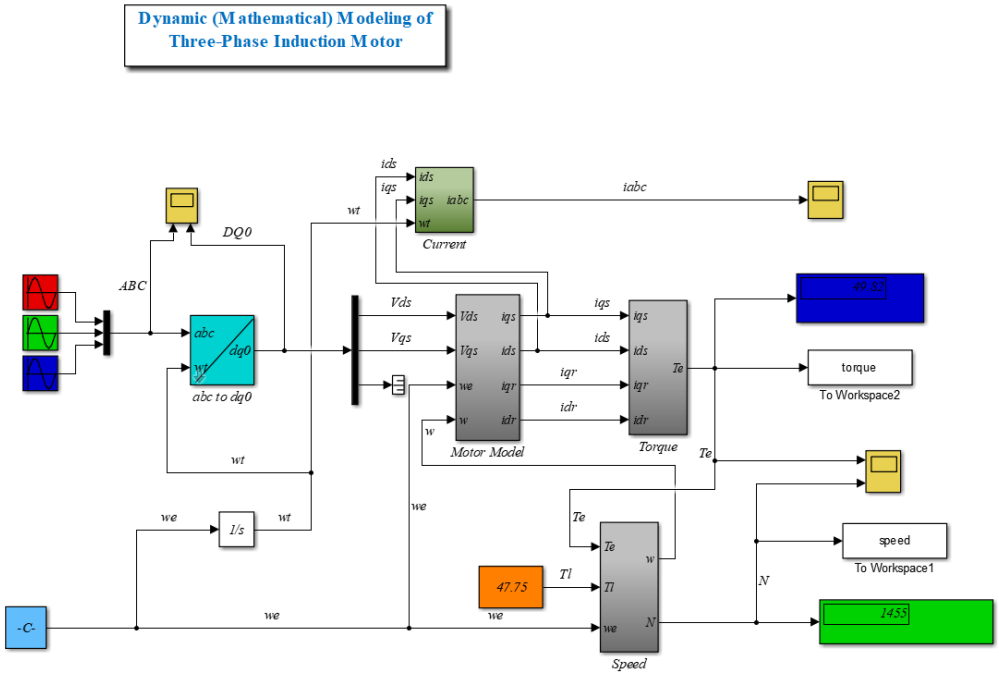


Figure 3.12: MATLAB Simulink Model of Three Phase Induction Motor

Model Verification was done in MATLAB/SIMULINK,by using this parameters

- Power = 5.4 hp (4 KW)
- Frequency = 50 Hz
- $R_s = 0.288\ \Omega$: Stator Resistor
- $R_r = 0.158\ \Omega$: Rotor Resistor
- $L_s = 0.0425$ H : Stator Inductance
- $L_r = 0.0418$ H : Rotor Inductance
- $L_m = 0.0412$ H : Mutual Inductance
- J = 0.4 : Motor Inertia $\text{kg} \cdot \text{m}^2$
- P = 4 : Pole
- $T_l = 47.75$: Load Torque $(N.m)$

- $\omega = 2\ \pi\ \times 50$
- $L_s = L_{ls} + L_m$
- $L_r = L_{lr} + L_m$

3.6 Simulation Results for Model Verification of Three Phase Induction Motor

3.6.1 Simulation Results of IM with Full load

Upon starting an induction motor and during other brief activities Because of the massive currents it draws when it starts and during other transient activities, an induction motor can result in oscillatory torques, voltage dips, and even power system harmonics. Observations have shown that the machine experiences short transients prior to reaching its full potential.

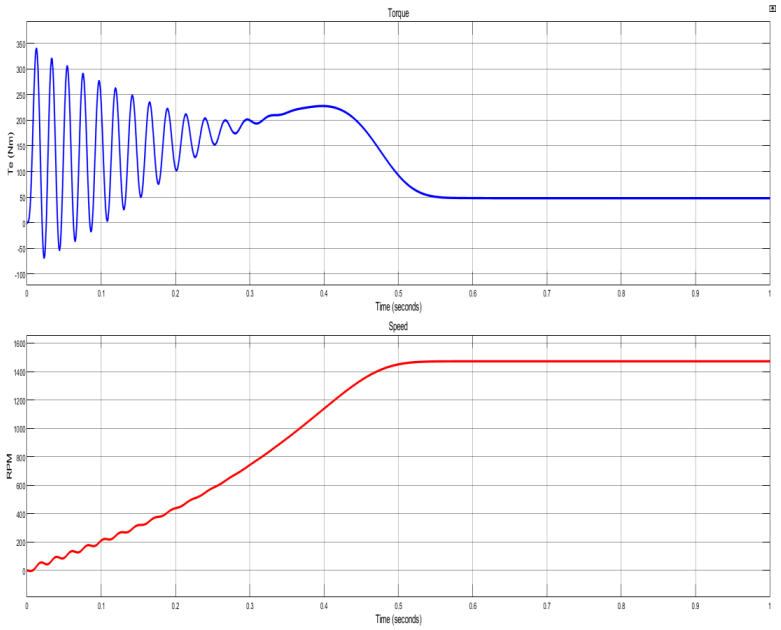


Figure 3.13: Electromagnetic torque and speed at full load

3.6.2 Simulation Results of IM with NO load

When there is no load on a three-phase induction motor, the rotor speed is close to, but not equal to, the synchronous speed. The rotor speed will decrease as the load increases.

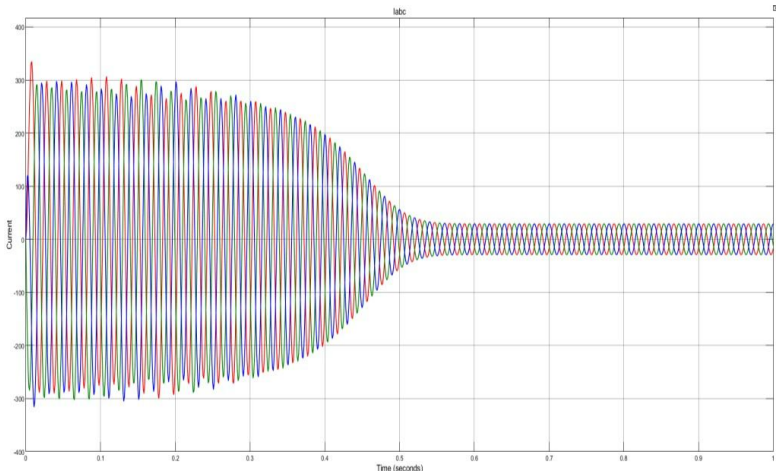


Figure 3.14: Output Current at full load

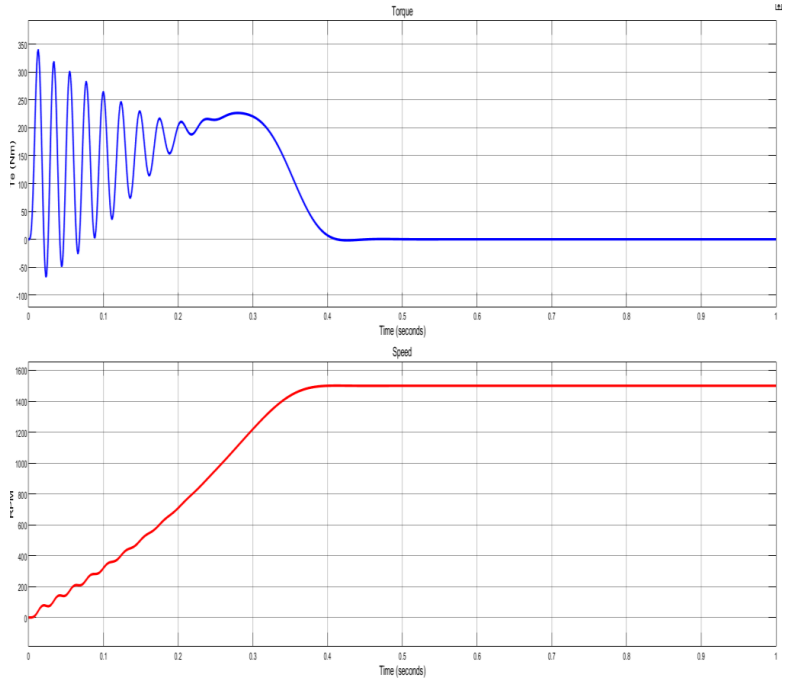


Figure 3.15: Electromagnetic torque and speed at NO load

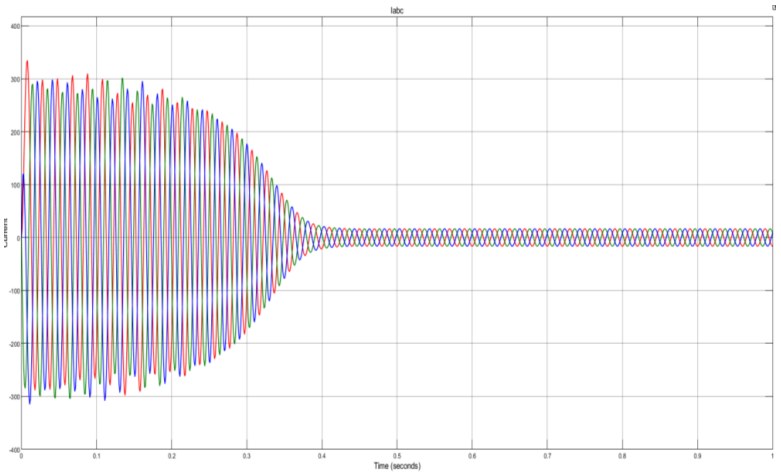


Figure 3.16: Output Current at NO load

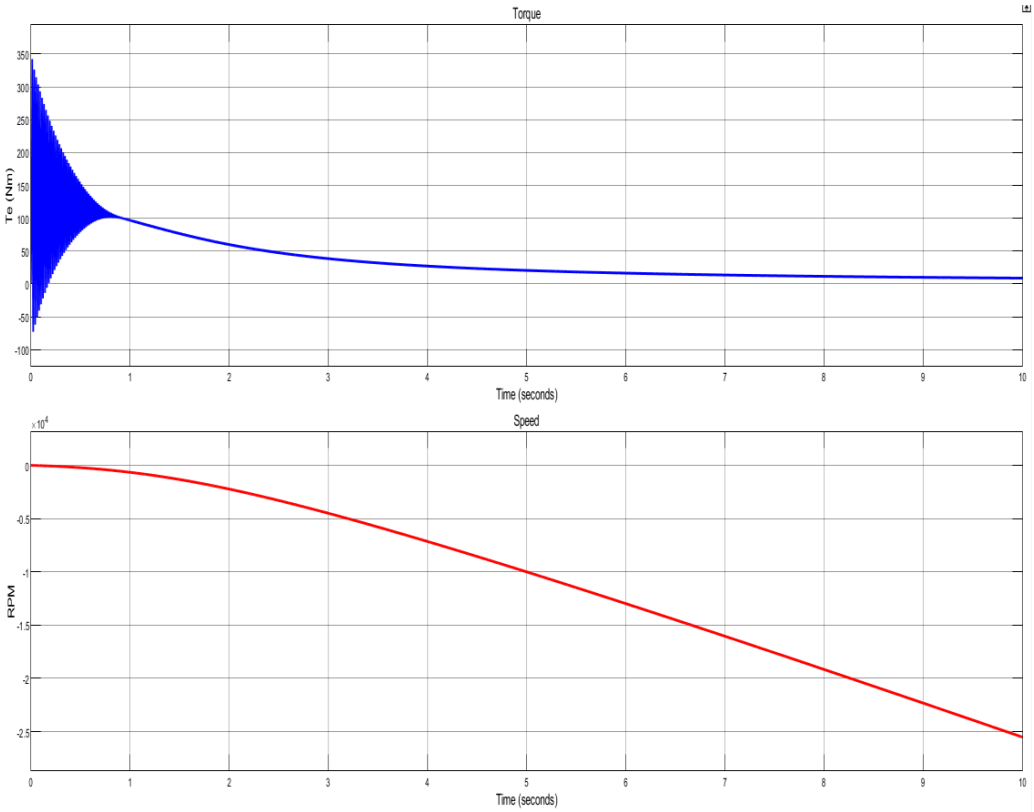


Figure 3.17: Electromagnetic torque and speed at Breaking down Torque

3.6.3 Simulation Results of IM at Breaking down torque

The amount of time needed to reach steady state speed will grow with the load. When the load is greater than the maximum electromagnetic torque generated for the applied terminal voltage, the motor stalls. When a motor's beginning is being simulated, the most basic scenario is when

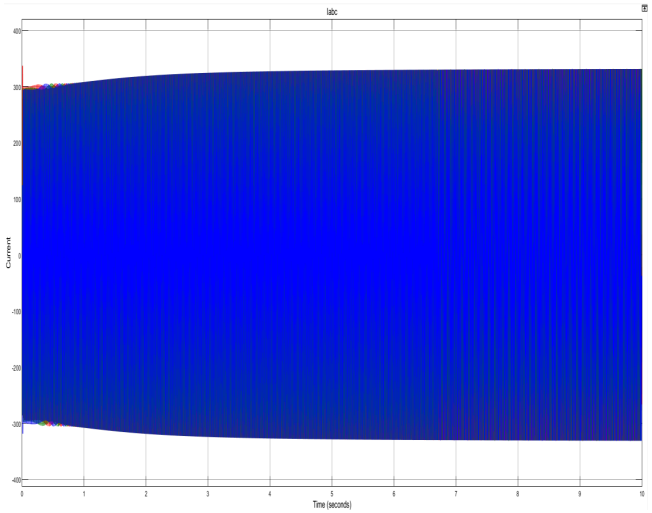


Figure 3.18: Output Current at Breaking down Torque

there are no initial circumstances. A specific operating condition is typically only achievable after a certain amount of simulation time, which is determined by the initial conditions.

Chapter 4

Controller Designs

Within the framework of the ANFIS Estimator for speed control, the three-phase induction motor's (IM) speed is controlled by the proportional-integral (PI) controller. The controller design must modify the proportional and integral gains (Kp and Ki) in order to attain the system's intended performance. Reducing the speed discrepancy between the reference speed and the motor's approximate speed. Formulas for Controllers: proportional-integral (PI) controller's control signal u(t) can be expressed:

$$u(t) = K_p e(t) + K_i \int_0^t e(t)\, dt \tag{4.1}$$

where u(t) is the control signal sent to the motor, K_p is the proportional gain of the controller, and K_i is its integral gain. Speed error, denoted by e(t), is difference between the reference speed (ω_{ref}) and the estimated speed (ω_r) of the IM.

4.1 Controller Tuning

For dependable and effective motor control, the Gains for the PI controller (K_p and K_i) must be tuned. A variety of tuning techniques may be used, including trial-and-error, sophisticated optimization techniques, and the Ziegler-Nichols method. This thesis uses a fuzzy logic-based approach and determine the best controller benefits. The Controller for Fuzzy Logic (FLC) dynamically adjusts the gains based on predetermined fuzzy rules, taking into account factors like inertia, torque supplied to the load, and motor operating circumstances. We select the ANFIS controller in order to compare its functionality in comparison to the PI controller's. The Auto tune mechanism for the PI controller is employed in the simulation for control the three-phase induction motor's speed.

4.2 Reasons for choosing the ANFIS (Adaptive Neuro-Fuzzy Inference System)

A number of reasons drive Adaptive Neuro-Fuzzy Inference System (ANFIS) use in diverse applications:

Handling Uncertainty: ANFIS are made to deal with ambiguous and imprecise data. There are frequently ambiguities and uncertainties in real-world systems that make it difficult to model or depict them using exact mathematical formulas. A framework for accounting for this uncertainty and basing choices on linguistic variables and fuzzy rules is offered by ANFIS.

Non linearity: Numerous systems display nonlinear behavior, which makes it difficult to adequately model with traditional control methods. Since ANFIS don't need explicit mathematical models, they are ideal for handling nonlinear systems. They are capable of capturing and representing complex nonlinear interactions through the use of membership functions and fuzzy rules.

Expertise: FLCs can be used to integrate thorough understanding of the control system. Although defining Heuristics and language norms combined with conventional control techniques can be difficult, experts can provide them according to what they know. This expertise can be translated to a real-time control approach with the use of FLCs.

Adaptability: ANFIS are able to adjust and make choices according to the system's present condition. Based on dynamically changing operating conditions and real-time inputs, they can modify their control actions. ANFIS are appropriate for systems that experience fluctuations, uncertainties, or changing environments because of their adaptability.

Robustness: ANFIS typically show strong resilience to perturbations and modeling uncertainty. ANFIS are able to accommodate uncertainties and fluctuations in the dynamics of the system because of their fuzzy rule-based structure. Because of their resilience, ANFIS are appropriate for systems with difficult-to-precise modeling or ones whose characteristics could vary over time.

The ANFIS was chosen because it provides resilience, interpretability, and flexibility while managing uncertainty, concurrently handling many inputs and outputs, nonlinearity, and expert knowledge. These characteristics make FLCs an effective control method in many different fields, including as process control, automotive systems, and robotics.

4.3 Adaptive Neuro-fuzzy Inference System

The hybrid system known as ANFIS combines artificial neural network learning capabilities with fuzzy logic's better knowledge representation and inference abilities. In order to achieve the intended performance, it might self-modify its membership function.

Based on a set of input-output data, a hybrid technique is used to optimize the fuzzy inference system. The Sugeno-type fuzzy systems are taught using the ANFIS technique. When using

ANFIS, the user usually specifies the type and quantity of fuzzy system membership functions. The ANFIS uses a neural network feature to generate a fuzzy inference system.
The reasons behind ANFIS's success include providing the necessary data set, a wider range of membership functions, strong generalization abilities, superior fuzzy rule explanation capabilities, and the simplicity with which language and numerical data can be combined to solve problems.

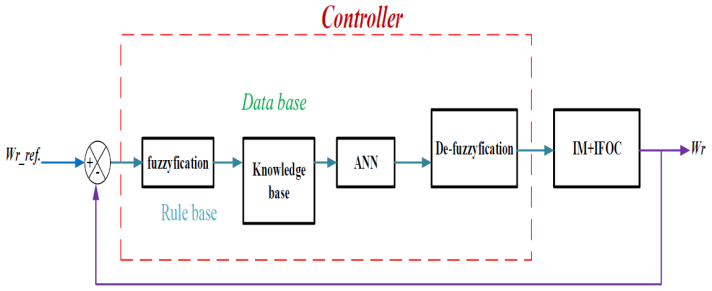


Figure 4.1: The ANFIS Control scheme of Speed controlling IM

4.3.1 Membership Function Design

- **Input Linguistic Variables**
  - Error in Speed (**e**)
  - Variation in Error (Δ**e**) or speed error derivative
- **Linguistic Variables Output**
  - Change of Control(ω_{sl})
  - Variation in Error (Δ**e**) or speed error derivative

A fuzzy inference system model is used in ANFIS, a straightforward data learning technique, to convert an input into the intended output. This prediction uses if-then rules, fuzzy logic operators, and membership functions. The five main processing stages of the ANFIS operation: the aforementioned sections have covered input fuzzification, fuzzy operator application, application technique, output aggregation, and defuzzification.

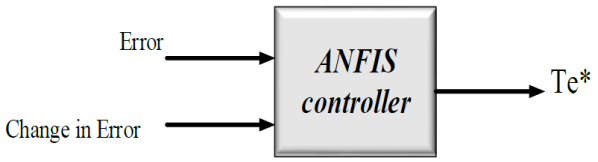


Figure 4.2: ANFIS Controller input and Output

This implementation of ANFIS uses a back propagation approach to adapt the system for fuzzy inference based on input-output data gathering. Mihoub et al. proposed the ANFIS controller. [3] in order for AC equipment to operate with excellent dynamic performance. They employed neuro-fuzzy controller following their use of the fuzzy controller. The latter is more dynamic than the former, they ultimately demonstrated. A sensor-less speed estimator system was developed by estimating motor speed using an artificial neural network (ANN). It was trained to evaluate a range of operating situations, including startup, step changes in reference speeds, and unknown load torque with variable parameters.

In order to suggest an IM spindle motor drive for sophisticated spindle motor applications that employs Both dead time and synchronous PWM compensation methods utilizing an ANFIS controller,carried out a real-time experiment. Because of motor specifications and control characteristics of the high speed IM drive changed over time, to control the rotor speed, they suggested using an ANFIS system. The plant was identified using a fuzzy neural network (Fuzzy Neural Network) identifier to train the network online and provide the ANFIS controller with the drive system's sensitivity information.

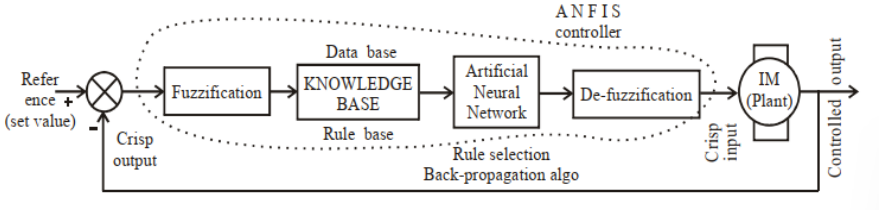


Figure 4.3: Power Circuit Connection Diagram of Induction Motor

- A first-order Sugeno fuzzy model for the fuzzy inference system with two inputs (x and y) and a single output (f) can be described using the following guidelines:
- Rule 1: f1 = p1x + q1y + r1 if x is A1 and y is B1.
- Rule 2: f2= p2x + q2y + r2 if x is A2 and y is B2.

where pi, qi, and ri are the design parameters (also known as resulting parameters) that are established during the training phase, and Ai and Bi are the fuzzy sets in the antecedent. The first layer in this five-layer artificial neural network structure stands for inputs, the second layer for fuzzification, the third and fourth layers for fuzzy rule assessment, and the fifth layer for defuzzification.

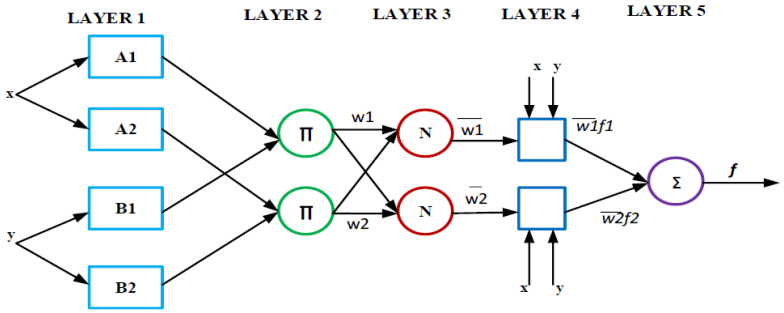


Figure 4.4: Basic ANFIS Structure

4.3.2 Learning algorithm of ANFIS

The parameters that are subject to change over time are found in the first and fourth layers of the ANFIS architecture. While linear consequent parameters are found in the fourth layer, nonlinear premises parameters are found in the first layer. In order to update these two parameters, a learning technique that can both train and adjust to its surroundings is needed.

This offline learning employs a hybrid learning methodology that uses using Least Squares Estimates (LSE) to determine consequent parameters S2 and back propagation to obtain premise parameters S1. Two passes are made in each learning step: forward and backward. For the current cycle, the forward pass propagates node outputs and utilizes LSE to estimate the best consequent parameters, while the premise parameters are assumed to be fixed. Back propagation uses gradient descent to Modify the backward pass's premise parameters, while resulting parameters stay constant. Until the error threshold is met, this iterative process is continued. This error criterion is often the total of the squared deviations between the intended and actual outputs, or in some situations, a user-defined performance metric.

In the absence of predetermined premise parameters, the search space grows and training convergence slows down, which is why both systems are used. As a result, there are two components to every training cycle that results in learning.

Step 1: In order for the LSM ascertain optimal consequent parameters, the input patterns are sent, and for the current cycle through the training set, it is assumed that the premise parameters are fixed.

Step 2: The patterns are replicated using gradient descent, a back propagation gradient descent method, but this time, error signals are used to update or modify premise parameters.

The hybrid learning strategy is more desired compared to the initial back propagation technique since comes together faster by reducing search space dimensions.

In particular, the type-3 method developed by Takagi and Sugeno is used. In each rule of

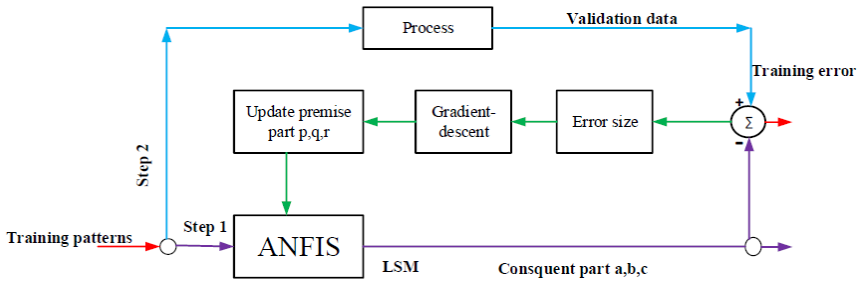


Figure 4.5: Learning procedure for ANFIS using a hybrid algorithm

this inference system, the input variables are combined linearly by adding a constant term. The weighted average is the final product of all the rules taken together. Each of layers that comprise ANFIS structure is described as follows:

Input Node (Layer 1): Membership functions are contained in nodes in this tier. Each node in this layer has a node function and is square and adaptable.

$$O_1^i = \mu_{Ai}(x), for i = 1, 2 or \tag{4.2}$$

$$O_1^i = \mu_{Bi-2}(y), for i = 3, 4 or \tag{4.3}$$

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

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

In this situation, Any suitable parameterized membership function, like the generalized bell function, can be used as the membership function:

$$\mu_{Ai}(x) = \frac{1}{1 + \left[\frac{(x-c_i)}{a_i}^2\right]_i^b} \tag{4.4}$$

where x is the input and a_i, b_i, and c_i are groupings of variables. The variables of this are known as premise parameters.

Rule Nodes (Layer 2): In this layer, every node has a circle labeled Π and is stationary. This node's output is the sum of all incoming signals.

$$O_i^2 = W_i = \mu_{Ai}(x) * \mu_{Bi}(y), i = 1, 2 \tag{4.5}$$

Each node's output demonstrates how strongly it operates. This layer chooses the lowest value between the two input weights. Any extra T-norm operators that perform fuzzy and (like min)

operations are permitted as the node function in this layer.

Layer 3: Every node in this tier is fixed. The ratio of the ith rule's firing strength to the sum of the firing strengths of all the rules is calculated by each ith node. The normalized firing strength provided by the ith node is its output.

$$O_i^3 = \overline{W_i} = \frac{W_i}{W_1 + W_2}, i = 1, 2 \tag{4.6}$$

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

Layer 4: Every node in this layer is flexible and has a node function.

$$O_i^4 = \overline{W_i} f_i = \overline{W_i}(p_i x + q_i y + r_i) \tag{4.7}$$

Where W_i is a normalized firing strength from layer 3 and p_i, q_i, and r_i are the parameter settings for this node. The parameters of this layer are referred to as subsequent parameters.

Layer 5: This layer's lone node, fixed node k, calculates the total output by adding up all of the incoming signals:

$$Overalloutput = O_i^5 = \sum_i = \overline{W_i} f_i = \frac{\sum_i W_i f_i}{\sum_i W_i} \tag{4.8}$$

4.3.3 Hybrid Learning Algorithm

If the presumption parameters (fixed presumptions) have been established, the final output of the ANFIS structure can be expressed as a linear combination of the following parameters. The following symbols can be used to represent the output f:

$$f = \frac{W_1}{W_1 + W_2} f_1 + \frac{W_2}{W_1 + W_2} f_2 \tag{4.9}$$

$$= \overline{W_1}(p_1 x + q_1 y + r_1) + \overline{W_2}(p_2 x + q_2 y + r_2) \tag{4.10}$$

$$= (\overline{W_1} x) p_1 + (\overline{W_1} y) q_1 + (\overline{W_2} x) p_2 + (\overline{W_2} y) q_2 + (\overline{W_2}) r_2 \tag{4.11}$$

where the parameters show that f is linear. that follow. (p_1,q_1,r_1,p_2,q_2,r_2).

The learning process uses the least squares approximation to calculate following parameters in its forward iteration. The derivatives of the squared error with respect to each node output are the error signals that go backward from the output layer to the input layer during the backward pass. The gradient descent approach is used in this backward step to update the premises parameters.

An ANFIS is comparable to a Takagi Sugeno fuzzy inference system in terms of functionality. Using predefined ANFIS uses back propagation gradient descent and the least squares type technique to fine-tune the membership functions and other pertinent parameters based on input-output training data pairs. By using these techniques, the ANFIS modeling becomes less dependent on expert knowledge and more methodical. To achieve the required input output mapping, each layer one and layer four node has an adaptive node with updated parameters. Although the nodes on layers two and three are fixed and parameterless, the gradient-based learning technique and the supplied training data are used to modify these parameters.
Adjusting the layer one premise parameters and the resulting layer four parameters is crucial during the ANFIS learning process in order to attain the optimal mathematical relationship between the inputs and outputs. ANFIS uses a learning cycle with two passes. Using the least squares method, the algorithm calculates the next variable on forward pass. In backward pass, as gradient descent is used to update the premise parameters, errors are propagated backward. Building a basic FIS is the number one step in the learning process, which continues to improve it. ANFIS uses a hybrid approach that combines least squares optimization and descending gradients back propagation used enhance original FIS. The computation of error occurs in every epoch. One way to determine when to give up could be to reach the maximum epoch or the least error target. In other words, comparable to a neural network.

4.3.4 The creation of neuro fuzzy adaptive inference systems for speed control

This section gives a summary of the ANFIS concepts for sensor-less vector-controlled drives' induction motor speed control. The idea of neural networks was first proposed in the late 1800s in an attempt to explain the functioning of the human mind in those earlier times. These neural networks began to become increasingly significant in the many technical applications as the years passed. Neural networks have been successfully used in voice recognition, image analysis, and adaptive control to build software agents or autonomous robots and to operate machinery.
Artificial neural networks (ANNs) are a class of intelligent algorithms that can be used for time series prediction, identification, and control. Various induction motor characteristics can be used to train neural networks. Since they are a non-linear function, they can be utilized to precisely identify the highly nonlinear system parameters. Neural networks have lately been presented as a tool for detecting and managing nonlinear dynamic systems due to their ability

to accurately simulate a wide range of non-linear functions to any desired level.
Additionally, it provide fault tolerance and lightning-fast parallel processing. Studies on the application of NNs for speed estimates in ac drives and power electronics have also been conducted. This approach is robust to parameter changes and yields a speed estimate that is fairly accurate. The neural network speed estimator needs to be properly trained using a range of patterns in order to function well.
Controlling various real-time system properties is among the effective uses of fuzzy logic in control engineering. Very important outcomes are achieved when neural networks are integrated with this logic. Data can teach neural networks new ideas. Nevertheless, it has been challenging to comprehend the data acquired by neural networks. To put it another way, it is typically challenging to comprehend the significance of each weight and neuron. On the other hand, because fuzzy rule-based models make use of language words and the IF-THEN rule structure, they are simple to comprehend. Fuzzy logic, in contrast to neural networks, is incapable of self-learning. Techniques from other fields, such as statistics and system identification, must be incorporated in order to learn and recognize fuzzy logic systems. Given the learning capabilities of neural networks, it makes sense to integrate these two approaches.

Flow chart for design of ANFIS Controller

The following outcomes were seen in Neuro-Fuzzy Designer following the loading of the training data and data set checks on figure 4.8.

In ANFIS-based modeling, neural networks' capacity for learning is combined with the transparent language representation of fuzzy systems to enable training of the neural networks to carry out an input/output mapping. When designing, data alone is not enough to determine what the membership function should be. Consequently, ANFIS allows for automatic parameter changes; the membership functions accurately capture the data's dynamics.
The rules that ANFIS offers are representative of the underlying system, making them useful data for understanding the process model in greater detail. The unique requirement for the launch of FIS as a controller is that the settings of membership functions be refined in a way that ensures optimal plant performance. [21]
The following formulas are used to model the inputs to the ANFIS controller, namely the error and the change in error:

$$e(k) = \omega_{ref} - \omega_{est} \tag{4.12}$$

$$\Delta e(k) = e(k) - e(k-1) \tag{4.13}$$

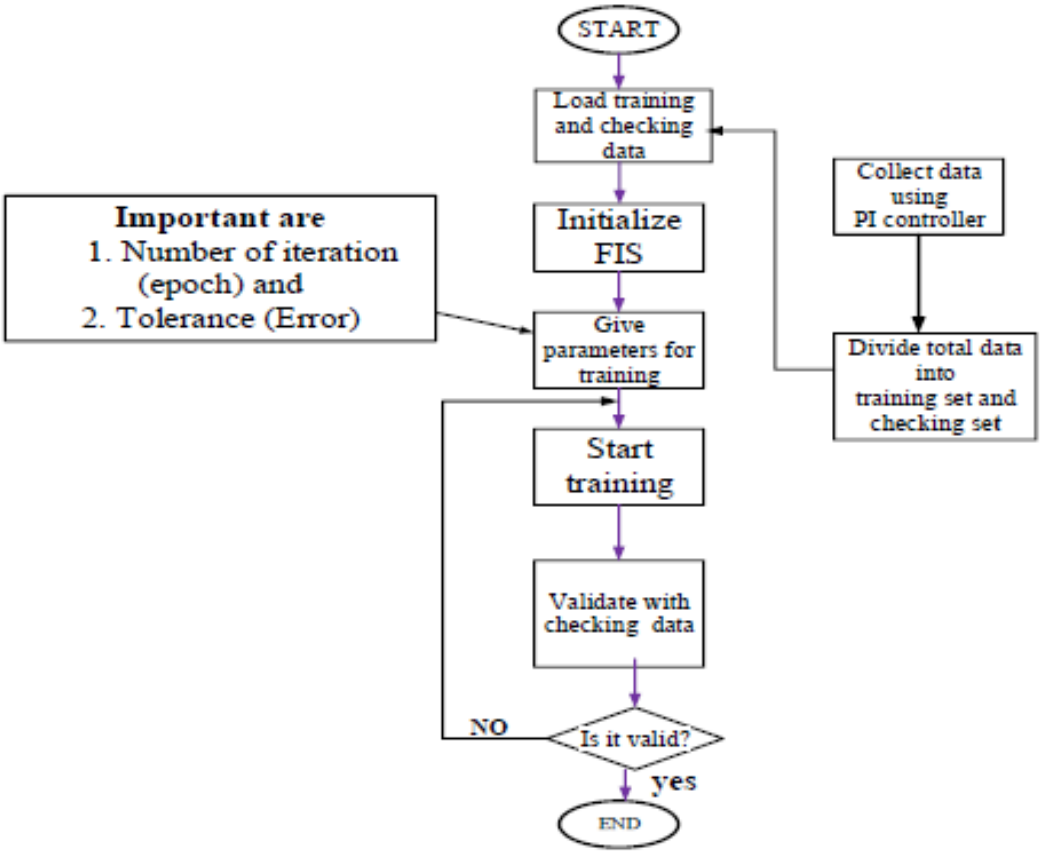


Figure 4.6: ANFIS Controller design Flow chart

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

To build ANFIS, a conventional PI controller that collects training or checking data for 3.5 seconds is built and simulated using MATLAB/SIMULINK. After the data was collected, the data was divided into training and checking data sets at random. Of the 3.5 million data points collected, three quarters were considered checking data, while the remaining 1,050,000 (30%) and 2,450,000 (70%) were classed as training data. To determine the membership function parameters that allow the associated fuzzy inference system (FIS) to monitor the input/output data, ANFIS combines the capabilities of fuzzy logic and neural networks, offering a method for the fuzzy modeling process to learn specifics about a data set. The FIS structure can be created by looking at a network type structure that maps inputs through input membership functions and related parameters and outputs through output membership functions and related parameters. A neural network's structure is comparable to this one.

In this thesis, generalized bell functions, as shown in the above equation, are the input membership functions that are utilized. A total of seven membership functions were considered during construction of ANFIS. A linear membership function type is chosen for output, resulting in

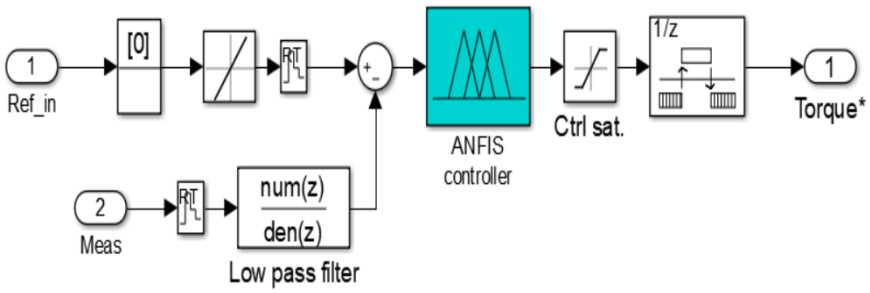


Figure 4.7: ANFIS Torque Regulator

the generation of 49 rules in total, and 100 iterations are chosen. The simulation result shows visualization of the inputs, surface, and rule views' membership functions. The parameters related to the membership functions will change as a result of learning. These parameters are computed via a mixed learning technique. The optimum training/checking data set was chosen using RMSE after the ANFIS models' performances on both training and checking data were assessed. We tested with several system parameters, including the data set sample, epoch number, membership function type and number, and number of inputs, in order to determine the optimal ANFIS system. The MATLAB® toolbox function *anfis* uses a hybrid learning approach to alter the membership function parameters of a fuzzy inference system (FIS) that is constructed using a specified input/output data set and the Grid Partitioning technique.
The system was trained using the training data set, while the checking data set was used to validate the learned ANFIS model's abcapacity for generalization in the context of adapting learned content. After the input/output data pairs were gathered, training and verifying data pairs were entered into the ANFIS GUI of the MATLAB toolbox. Setting the quantity and kind of input and output membership functions, along with the amount of epoch numbers, was the next step in initializing the FIS. Once FIS was configured and the number of epochs was specified, the training data pairs were trained until the minimum training and checking errors were observed. In order to validate the model, the last step was to compare the training data with checking data to see if the errors were acceptable. The Neuro-fuzzy Designer displayed the following outcomes following the loading of the training data and data set verification. The training and check data sets have been put into the ANFIS editor toolbox and are displayed in the figure 4.8. At each epoch, the fuzzy inference system's capacity for generalization is evaluated using checking data. The format of the training and verifying data is identical. This set of data is used to validate the fuzzy inference model. In order to perform this validation, the checking data is fed into the model, and the model's response is monitored. The checking data is added to the model at each training epoch when the ANFIS Editor GUI's checking data option is selected. After loading the training and validating data, the FIS membership function

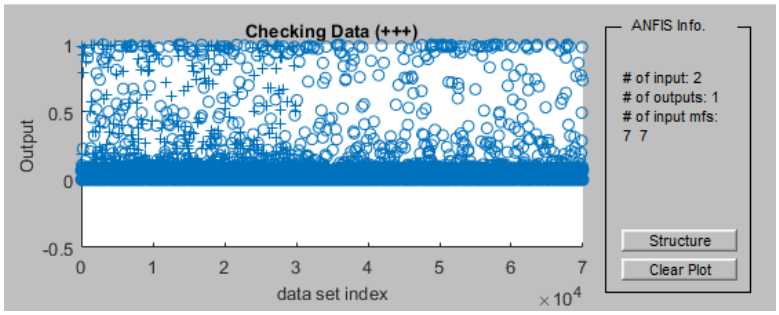


Figure 4.8: Data sets are loaded into the ANFIS editor toolbox for training and verification.

parameters are generated using the ANFIS editor GUI. As training increases, the checking data error decreases because it is sufficiently comparable to the training data. The checking error is the difference between the value of the checking data output and the output of the fuzzy inference system that is connected to that checking data output and corresponds to the same value of the checking data input. The RMSE for the checking data for every epoch is recorded by the checking error. Chapter 5 discusses the ANFIS speed controller's performance along with the pertinent findings and Simulink blocks.

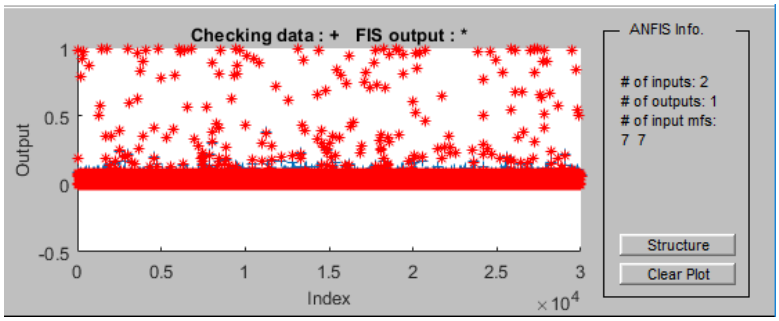


Figure 4.9: Testing FIS with checking Data set

Table 4.1: The developed Adaptive Neuro-Fuzzy Inference System's specifications

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. of inputs	147
Type of input MF	Generalized bell	Total No. of parameters	189
No. of output MF	49	No. of nodes	131

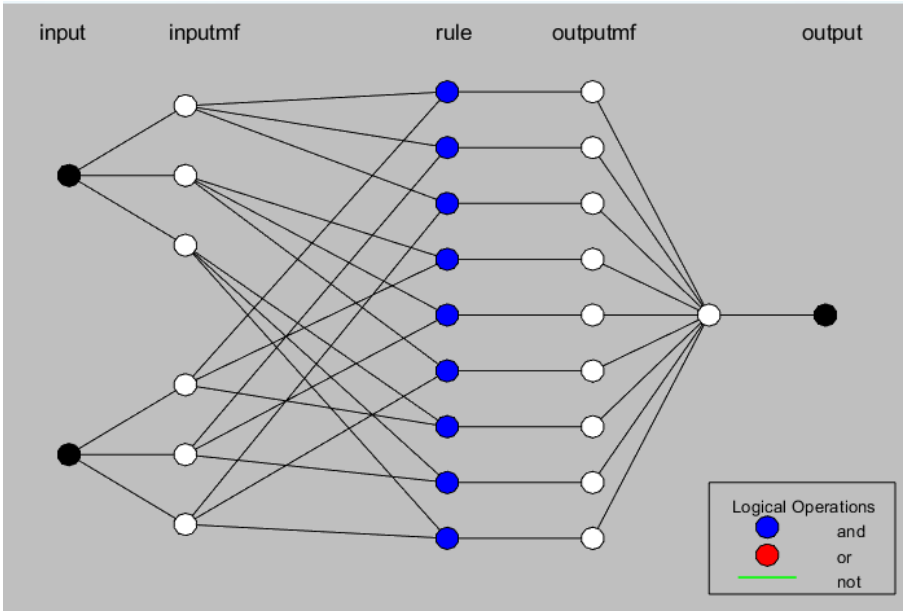


Figure 4.10: Schematic of proposed ANFIS controller

4.4 Design of Fuzzy Logic Controllers

The Fuzzy Logic Controller (FLC) is a control system that modifies the induction motor's behavior and makes judgments using fuzzy logic concepts. This rule-based control method uses fuzzy rules and language aspects to simulate human decision-making. Induction motors with unpredictable parameter variations are sometimes difficult to manage with traditional proportional-integral (PI) controllers. Using a fuzzy logic controller (FLC) to modify the PI controller's gains under different operating conditions is one possible way to address this issue. The flexibility and capacity of an FLC to dynamically modify the controller gains in response to the state of the system are what make its use so alluring. The FLC can intelligently adjust the PI controller's gains by using membership functions, fuzzy logic rules, and linguistic factors. The system can react faster, be more stable, and withstand changing situations better due to its flexibility. Critical system factors such as speed, load torque, and other pertinent variables are taken into account by the FLC in order to efficiently establish the proper gains for the PI controller. Fuzzy logic rules are used to modify the gains, and the linguistic variables offer a descriptive language to explain how the system functions.

For nonlinear systems (such as induction motors) that undergo unpredictable fluctuations, the control system is an alluring choice because it combines the flexibility of the FLC with the strength of the PI controller. It not only performs better but additionally displays efficiency and flexibility in the face of shifting operational areas.

In this thesis, the output signal u(t) is equivalent to the change in quadrature reference current

(I_q^*). Depending on the speed error (e(t)) and the rate at which the speed error changes (ce(t)), the tuning PI controller adjusts the proportional gain (Kp) and integral gain (Ki) parameters.

$$e(t) = \omega_e^*(t) + \omega_e(t) \tag{4.14}$$

$$ce(t) = \frac{d}{dt}e(t) \tag{4.15}$$

Where: $\omega_e^*(t)$ is reference speed and $\omega_e(t)$ actual speed of the motor.
Following self-tunning, the PI controller's control action can be characterized as follows:

$$u(t) = K_{p_{new}} \cdot e(t) + K_{i_{new}} \cdot \int e(t)\,dt + K_{p_{fuzzy}} \cdot e_{fuzzy}(t) + K_{i_{fuzzy}} \cdot \int e_{fuzzy}(t)\,dt \tag{4.16}$$

Where:

- The PI controller's output, or control signal, is denoted by u(t).
- The new gains that the PI controller acquired upon self-tuning are $K_{p_{new}}$ and $K_{i_{new}}$.
- The fuzzy control's gains, known as $K_{p_{fuzzy}}$ and $K_{i_{fuzzy}}$, change in real time according to the system being controlled.
- The value e(t) represents the speed mistake
- The integral of the speed error over time is denoted by $\int e(t)\,dt$.
- Determined by using the fuzzy logic control, the fuzzy error is represented by $e_{fuzzy}(t)$.
- $\int e_{fuzzy}(t)\,dt$ is the integral of the fuzzy error over time.

4.4.1 The fuzzy language of input and output variables

In a fuzzy logic system, the different levels or states of the variables are represented by linguistic phrases or labels known as the "fuzzy language of input and output variables." [15]. These linguistic phrases are frequently described using fuzzy sets. These sets give different values or ranges of the variables membership degrees.

Fuzzy language is crucial for transferring sharp values entered into fuzzy sets and selecting the optimal control strategy based on fuzzy rules. The fuzzy sets of outputs in the fuzzy structure described above is generated by mixing the inputs from the fuzzy sets and applying the fuzzy rules that were developed. The max-min approach is employed for aggregation, and the centroid method is used for defuzzification.

For each PI controller parameter, represented as and, the FIS has two outputs. The derivative

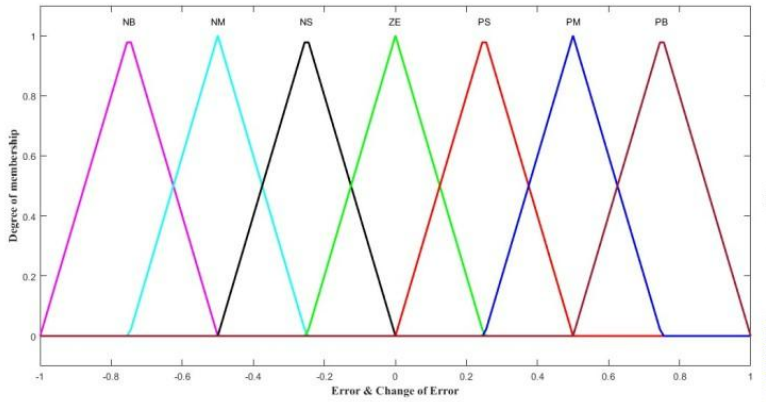


Figure 4.11: Function of Membership for the Input Variables e and ce

of error de(t) and error e(t) are the two inputs. The Mamdani model is used as the framework for fuzzy inference, modified to optimize the values of and, as demonstrated in figure 4.11.

Table 4.2: The set of rules used to determine values of K_p and K_i

e/ $\dot{e}$	NB	NM	NS	Z	PS	PM	PB
NB	VB	VB	MB	MB	B	B	M
NM	PB	MB	MB	B	B	M	S
NS	MB	MB	B	B	M	S	S
ZE	MB	B	B	M	S	S	MS
PS	B	B	M	S	S	MS	MS
PS	B	B	M	S	S	MS	MS
PM	B	M	S	S	MS	MS	ZE
PB	M	S	S	MS	MS	ZE	ZE

Triangular membership functions were chosen because of their capacity to provide excellent simplicity and control performance. Figure 4.9 displays the membership functions for speed error and the rate of change of speed error. In this system, Every variable in the seven layers is subjected to fuzzy membership functions. The 7 by 7 rule basis table used by the system is shown in Table 4.2.

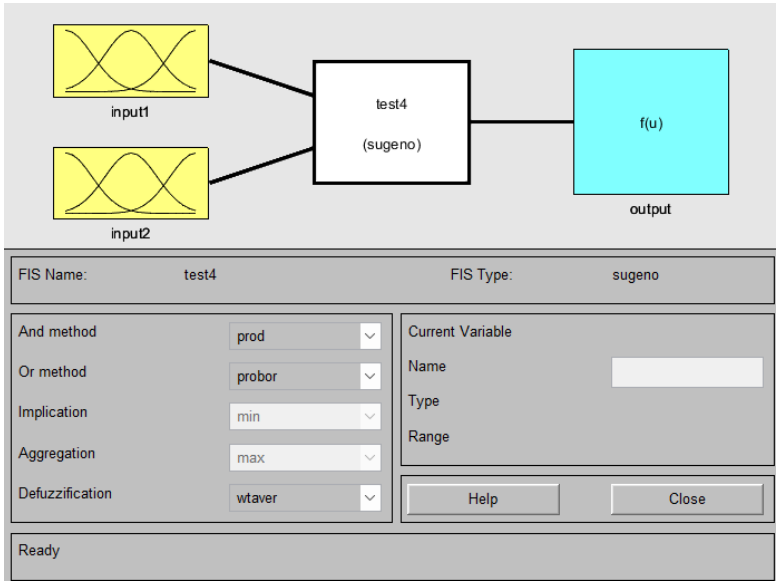


Figure 4.12: FIS editor: MATLAB's rules window

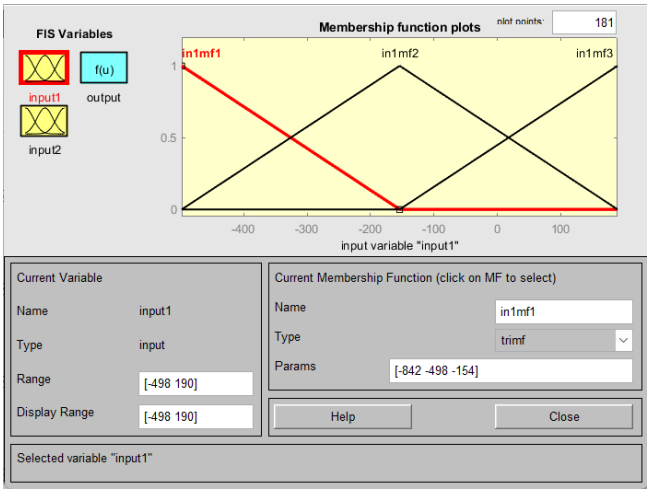


Figure 4.13: The input error's membership function (Δe)

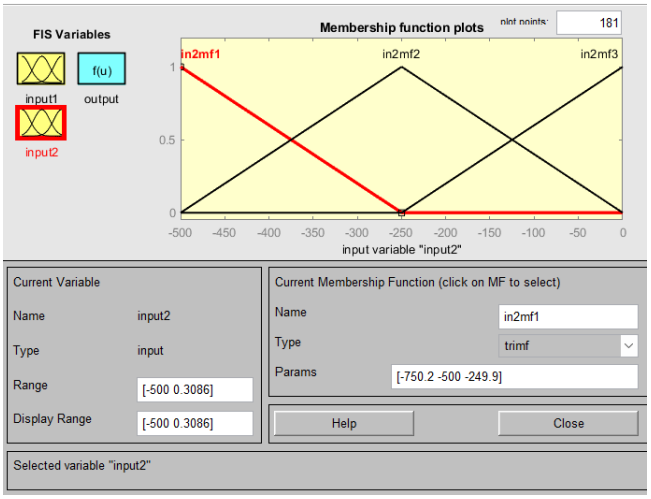


Figure 4.14: Membership function for the input Change of Error

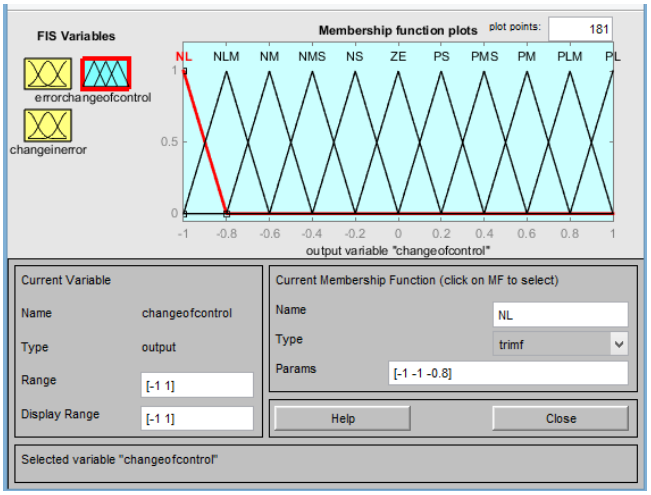


Figure 4.15: Membership feature for the Change of Control output (ω_{Sl})

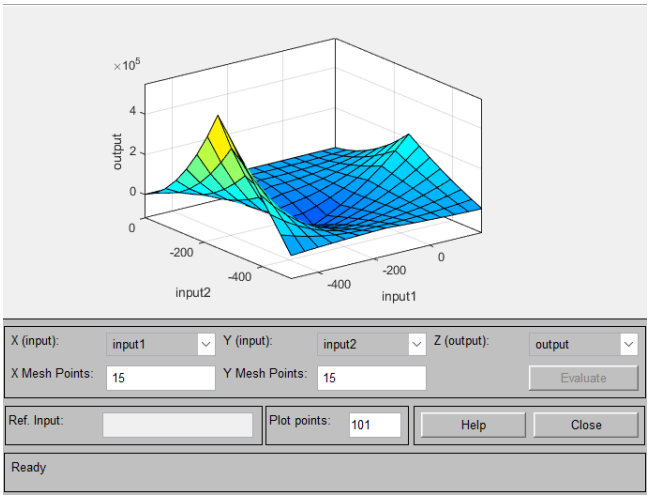


Figure 4.16: A three-dimensional control surface plot

Chapter 5

Simulation Results

5.1 Simulation Modeling

The MATLAB/Simulink software is employed in the analysis, modeling, and simulation of dynamic systems. A variety of tools, a graphical environment with visual real-time programming, and the capacity to model intricate dynamic systems are some benefits of using Simulink.
The MATLAB/Simulink environment is used to simulate the three-phase induction motor utilizing the mathematical equations that were introduced in the second chapter. The complete Simulink model for scalar speed control of a three-phase induction motor using PI and ANFIS controllers is displayed in Figure 5.1. The subfunctional blocks that comprise the overall system Simulink block comprise the voltage and current sensors, PI controller blocks, coordinate transformation blocks, ANFIS block, speed estimator, SVPWM inverter block, IM physical model block, and so on. A variety of solvers can be used to simulate the block diagram after it has been created in MATLAB/Simulink. By solving the ordinary differential equation for every block in terms of the MATLAB modeling setup parameters, these solvers are able to ascertain the internal state variables of the blocks. Reducing computation time and improving simulation accuracy are largely dependent on the solver.
This chapter presents and discusses the simulation findings for an induction motor's speed control. The assessment consists of a PI controller, a fuzzy controller, and an ANFIS controller under various conditions, such as fluctuating reference speeds, forward and reverse operation, external disturbances, load variations, and parameter fluctuations. The analysis includes step responses for rotor speed, position, stator current, electromagnetic torque, and handling load disturbances. The primary objective is to demonstrate the ANFIS controller's resistance to external disturbances, varying torque, varying speeds, and parameter changes. Furthermore, Examined is how load fluctuations affect motor performance, also controller performance is quantified using Controller Performance Indexes. The simulation's results show how the scalar control and tuned PI controller may achieve precise speed regulation with minimal steady-state error, exhibiting robustness against disruptions.

5.2 Simulation and outcomes of induction motor speed control using fuzzy logic, PID controllers, and ANFIS Controllers

5.2.1 Three-phase stator current and estimated d-q current

The behavior of a motor's stator current under a 5 Nm load torque at a constant 1500 RPM is depicted in Figure 5.5, which is the simulation result. The stator current, which is initially non-sinusoidal at starting, progressively turns sinusoidal in the steady state, signifying steady motor operation. As seen in Figure 5.6, the d-axis current, or Id, stays at zero throughout the experiment. Additionally, the motor's reaction to load conditions throughout time is displayed via the Iq (q-axis current). This demonstrates how Iq changes take load torque into account. It also shows the degree to which the estimated current (Id and Iq) coincides with the real current (Id and Iq) or estimated currents as instructed by the motor controller.

* Repeating Sequence input - [1500 1400 1200 1000 1000] rpm
* Reference Final Speed - 1000 rpm
* Step time - 2
* Load Torque - 2

As illustrated in figure 5.7, the simulation was conducted with a range of speed inputs and a load torque of 2 Nm in order to assess the speed responsiveness of the motor when instructed to operate at 1000 RPM reference rates. Two was the step time. The speed response utilizing the PI controller initially displays a steady-state inaccuracy of 0.5 and an overshoot of 5.8 % for the references provided. The outcomes of the simulation show that the suggested estimator correctly follows a range of reference step inputs between 1500 RPM and 1000 RPM as we sequence 1500, 1400, 1200, 1000, and 1000 RPM. When operating at the same reference speed, the motor and load torque of 2 Nm, the speed response is shown in figure 5.8. It shows the three Controllers in operation at the same time. The fuzzy logic controller works, as can be seen, but it cannot keep up with the reference speed.

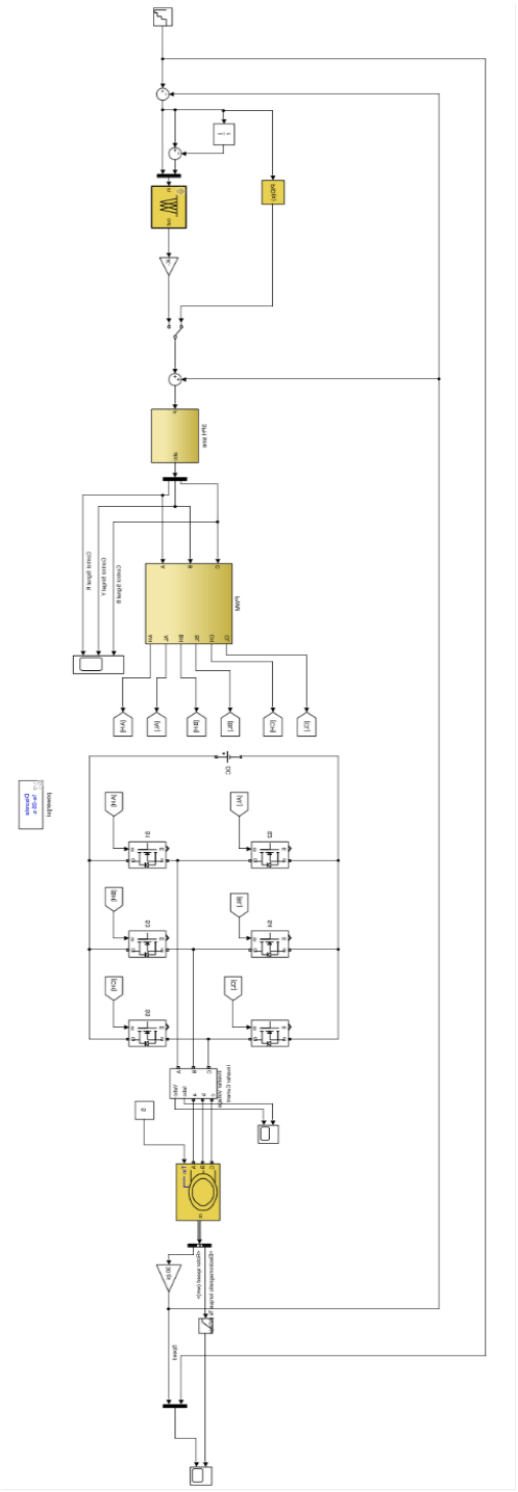


Figure 5.1: The overall model of a three-phase induction motor with PI controller that uses ANFIS-based scalar control

* Repeating Sequence input - 1500 rpm
* Reference Final Speed - 1000 rpm
* Step time - 5
* At full Load Torque - 5

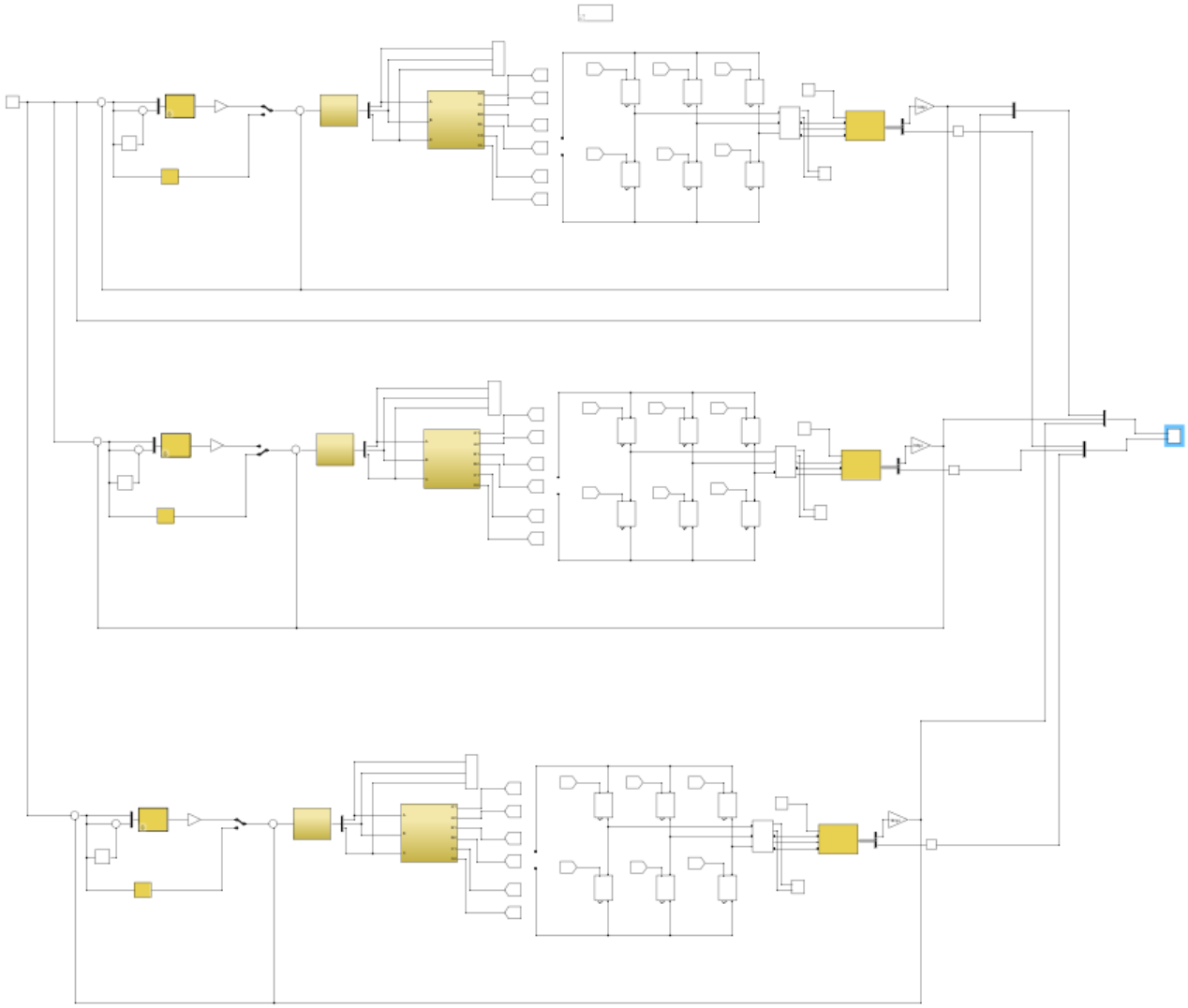


Figure 5.2: ANFIS, fuzzy logic, and PI controllers are used in the overall model of scalar control of a three-phase induction motor

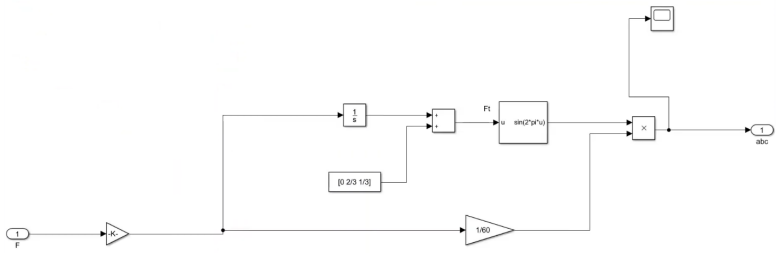


Figure 5.3: Three phase sine wave generation Sub system

The simulation's output, which is depicted in figure 5.9, was run with a step time of 5 and a variable speed input to assess the motor's speed response at 1000 RPM reference rates. The motor's full load torque was 5 Nm. For the given references, how the PI controller responds to speed first shows a 3.8 % overshoot and a 0.1 steady state inaccuracy.The simulation results show that when we put in step, the suggested estimator properly tracks the reference step input at 1500 RPM and 1000 RPM. While motor is instructed to run at the same reference

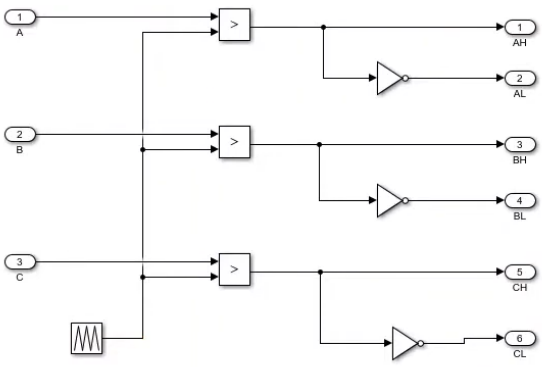


Figure 5.4: SPWM Sub-system

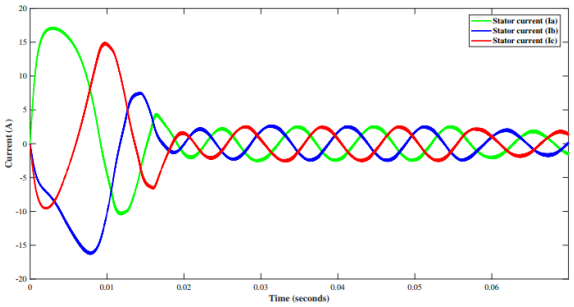


Figure 5.5: Stator Current in Three Phases Under Loading (5 Nm)

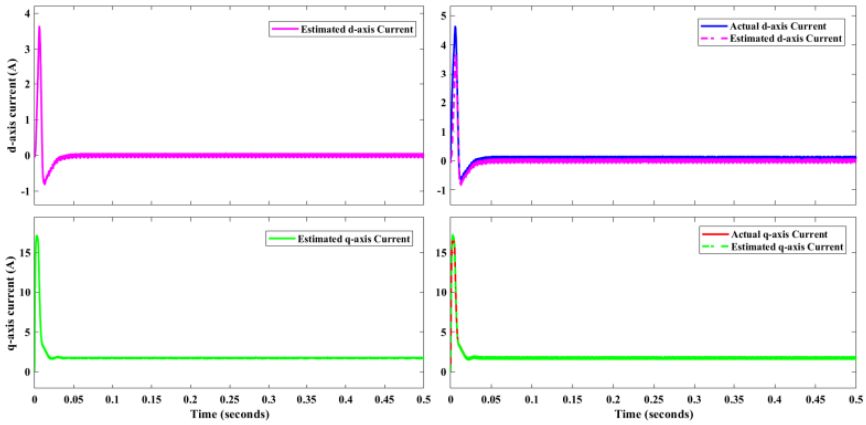


Figure 5.6: The d-q Current in Loaded Conditions, Both Real and Estimated (5 Nm)

speed and with the same load torque of 5 Nm, the speed response is displayed in figure 5.10. It demonstrates the parallel use of the three Controllers. The fuzzy logic controller functions as we can see, but it is unable to track the reference speed.

* Repeating Sequence input - [1500 1400 1200 1000 1000] rpm
* Reference Final Speed - 1000 rpm
* Step time - 2

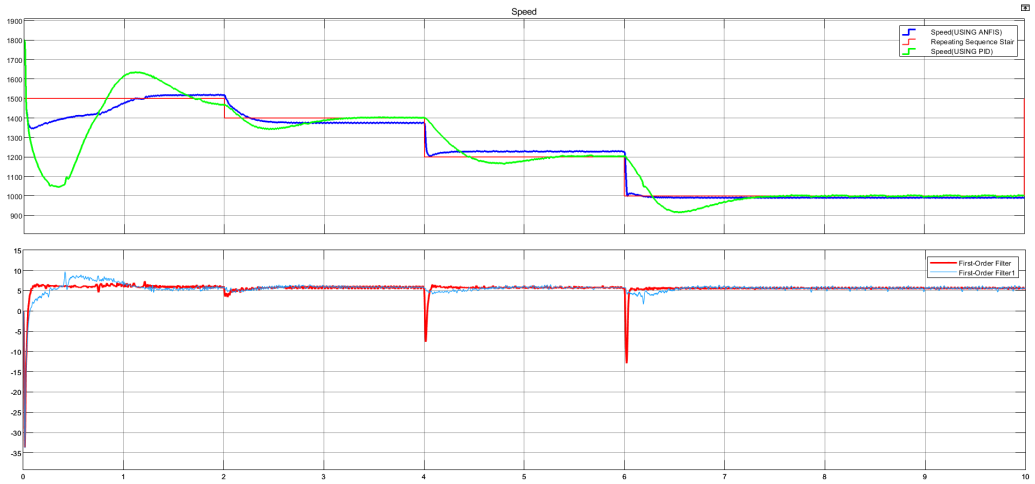


Figure 5.7: ANFIS and PID Controller output @ load torque 2 Nm

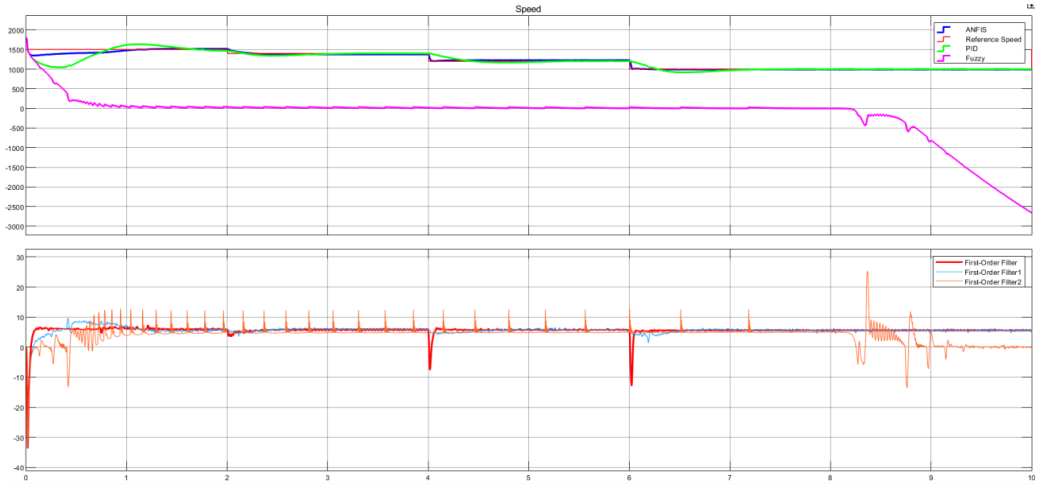


Figure 5.8: ANFIS, Fuzzy Logic and PID Controller output @ load torque 2 Nm

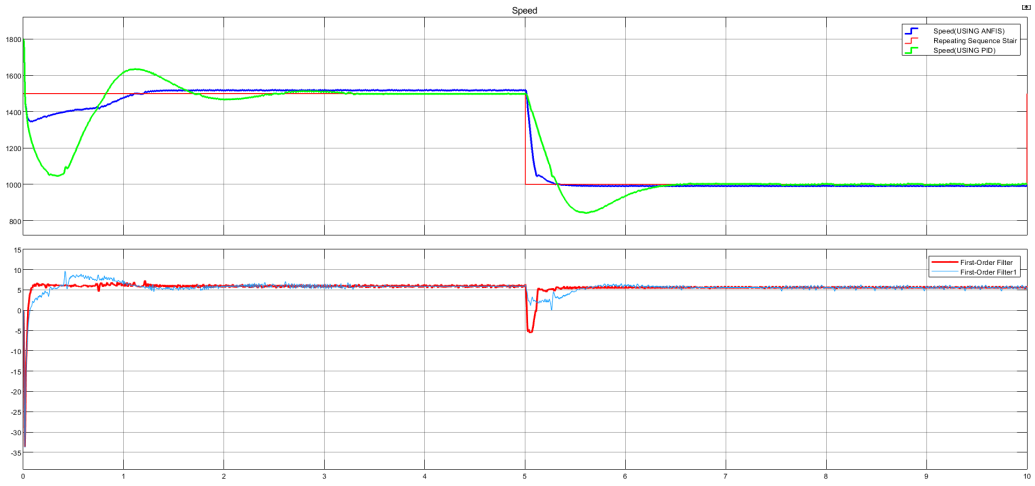


Figure 5.9: ANFIS and PID Controller output @ Full load torque 5 Nm

∗ With Load Torque - 10

The outcome of the simulation utilizing a range of input speed and a load torque of 10 Nm with step time 2 to evaluate the motor's speed response when running at 1000 RPM The reference

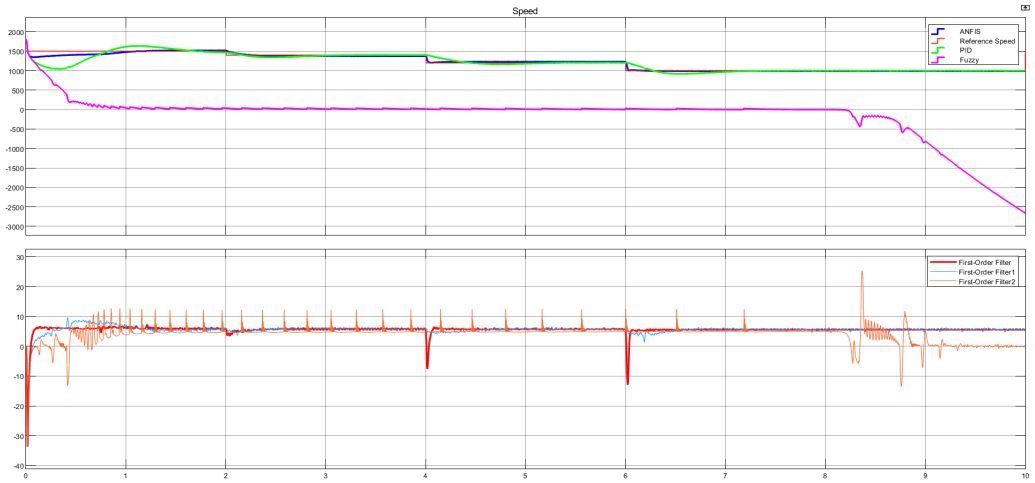


Figure 5.10: ANFIS, Fuzzy Logic and PID Controller output @ Full load torque 5 Nm

Table 5.1: **Calculated at Load Torque = 5 N.m**

Type of Controller	Rise Time	Settling Time	Peal Overshoot
ANFIS	0.22 sec	1.24 sec	1680 rpm
PID	0.46 sec	3.01 sec	1530 rpm

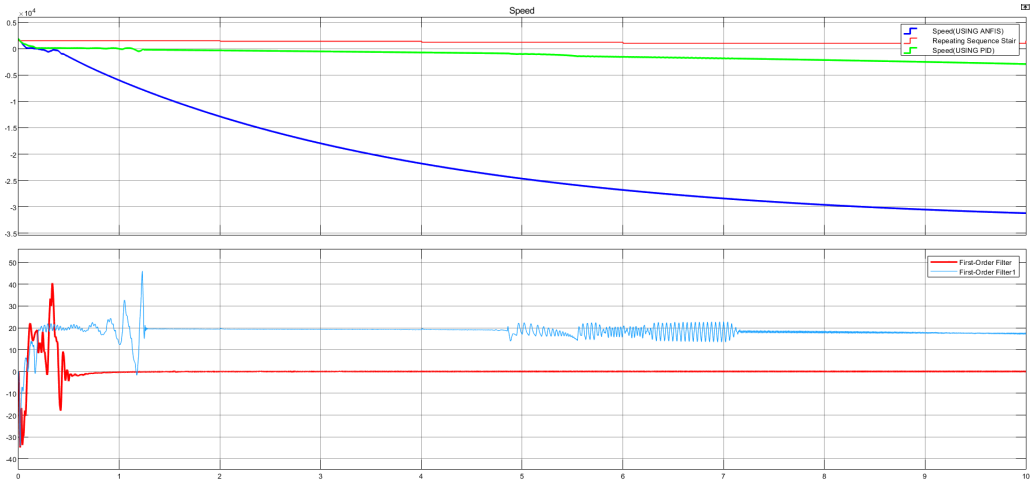


Figure 5.11: ANFIS and PID Controller output @ load torque 10 Nm

speeds are displayed in Figure 5.11. According to the simulation results, the suggested estimates failed to monitor a range of reference step inputs, from 1500 RPM to 1000 RPM, as we entered the following sequence: 1500, 1400, 1200, 1000, and 1000.When the motor is configured to operate at the same benchmark speed and 10 Nm load torque, the speed response is shown in figure 5.12. It shows the three Controllers in operation at the same time. The PI and fuzzy logic controllers were unable to track and trace the reference speed, despite the ANFIS Controller's best efforts.

The maximum torque the motor is capable of producing is known as its **breakdown torque**. Increasing the load above this point will cause the motor to stall and come to a sudden stop.

Induction motors usually have breakdown torques between 200 and 300 percent of their full load torque. According to the graph, the motor stalls because the load has surpassed the breakdown threshold.

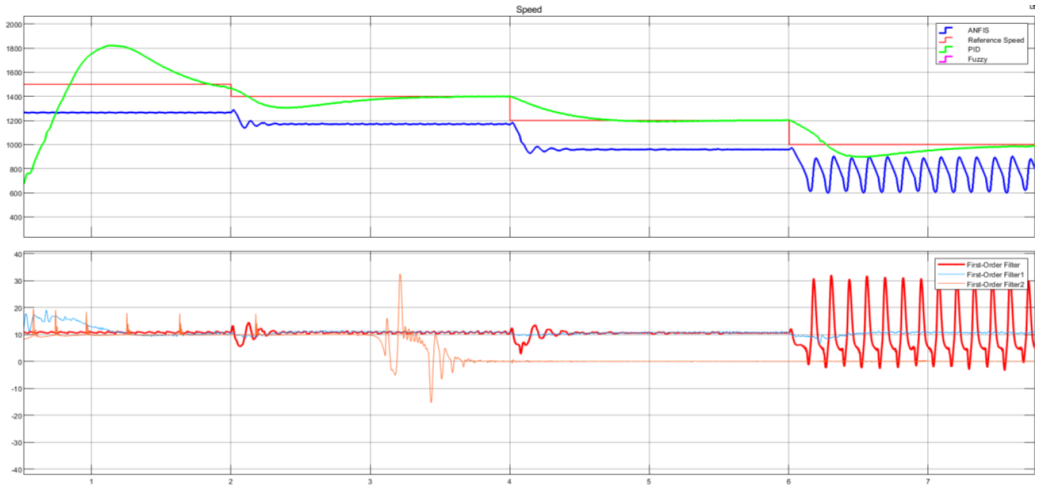


Figure 5.12: ANFIS and PID Controller output @ NO load torque

* Repeating Sequence input - [1500 1400 1200 1000 1000] rpm
* Reference Final Speed - 1000 rpm
* Step time - 2
* With No Load Torque

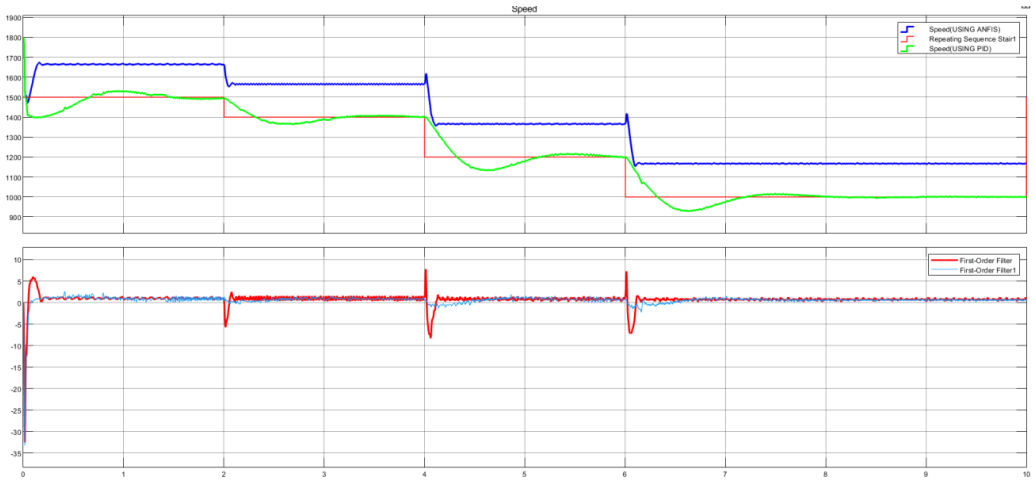


Figure 5.13: ANFIS,Fuzzy Logic and PID Controller output @ NO load torque

As shown in figure 5.13 and step time 2, a simulation was conducted using a range of speed inputs and a No load torque to assess the speed responsiveness of the motor when directed in order to run at 1000 RPM reference rates. Simulation results demonstrate that the proposed estimator failed to track a range of reference step inputs, from 1500 RPM to 1000 RPM, when we

entered the following sequence: 1500, 1400, 1200, 1000, and 1000. Figure 5.14 demonstrates the speed response when the motor is instructed to run without load torque at the same reference speed. It illustrates how the three Controllers can be used in parallel. It is evident that ANFIS Controller makes an effort on tracking reference speed, while fuzzy logic and PI controllers were unable to do so.

Let us see each controllers separately at:

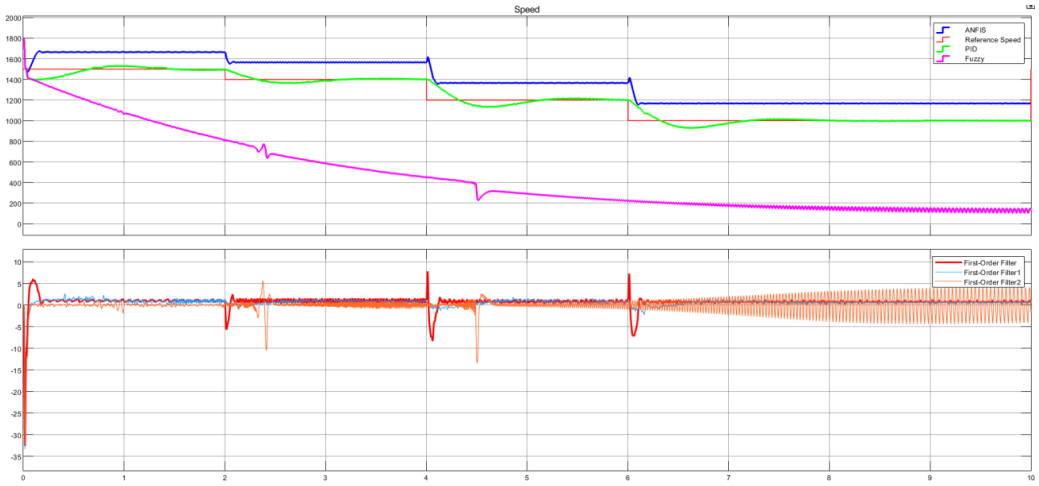


Figure 5.14: ANFIS and PID Controller output @ NO load torque

* Repeating Sequence input - [1500 1400 1200 1000 1000] rpm
* Reference Final Speed - 1000 rpm
* Step time - 2
* With Load Torque - 5

The auto-tuned PID controller has a 3.2 % overshoot and a peak overshoot of 1630 RPM, as shown in Figure 5.15. It clearly tracks the reference speeds that are considered.
Figure 5.16 shows that ANFIS Controller has a rising time of 0.2 seconds, a peak overshoot of 1530 RPM, and an overshoot of 0.2 %. Evidently, it nearly exactly matches the reference speeds.

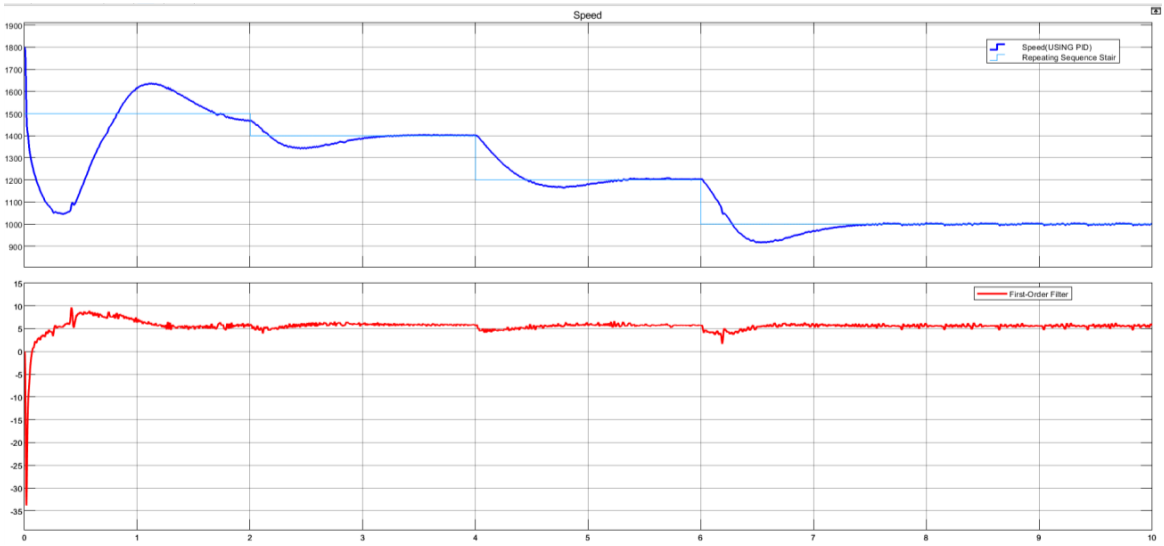


Figure 5.15: PID Controller output

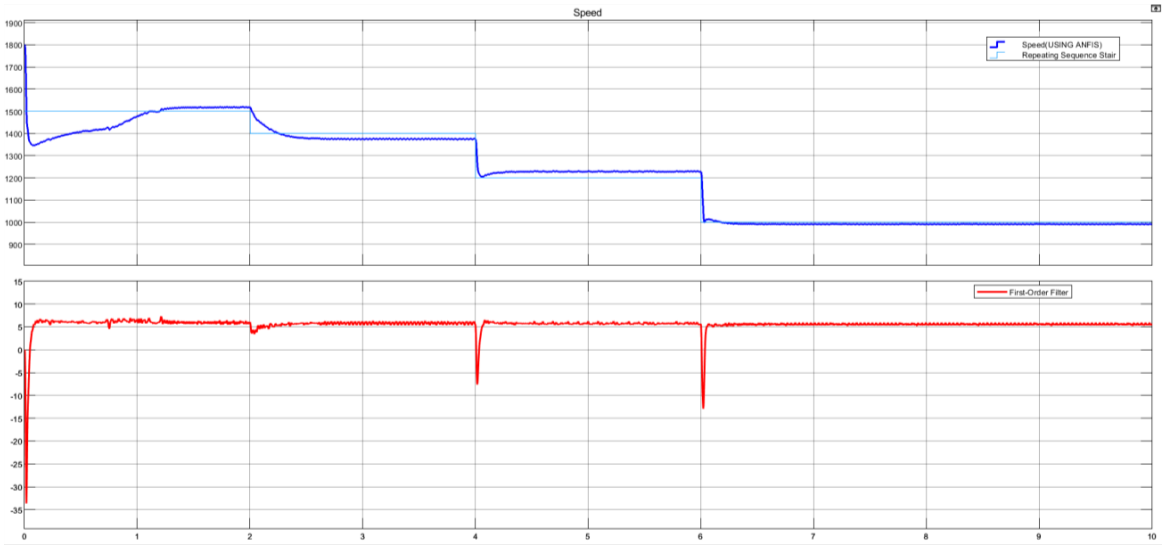


Figure 5.16: ANFIS Controller output

Chapter 6

Conclusion and Recommendations

6.1 Conclusion

Overshoots are less frequent when the ANFIS Controller is used than when both the fuzzy and PI controllers are used, as the foregoing figures illustrate. Additionally, the settling time is shortened in the ANFIS instance, but the rising time is longer. The ANFIS controller demonstrates a more precise response to changes in reference speed, whether above or below the base speed. It reaches the new reference speed more rapidly and exhibits significantly less overshoot. Figures illustrate that, unlike the ANFIS controller, the PI controller deviates from the new reference speed and struggles to maintain a steady state, especially when the speed is considerably higher or lower than the base speed. In contrast, the ANFIS controller achieves a stable state, with the resulting speed closely aligning with the new reference speed, even if it does not match it exactly.

In this simulation, the Fuzzy Logic Controller has certain disadvantages in addition to its advantages. Since a fuzzy logic controller had a shorter settling time and fewer system overshoots than a typical PI controller, the simulation benefited from its adoption. Even when the speed was higher than the base speed or much lower than the same, it tended to get closer to the benchmark pace, in contrast to the PI Controller, which produces 85% of the planned speed. Another advantage is that the design of the control mechanism was not very complex.

One of the limitations is that the rising time was slightly longer than with a conventional PI controller. When the real speed was adjusted from base speed, it was found to be 95 % of the intended output, or almost equal to the reference speed, rather than exactly matching it. Reaching a reference speed that is only marginally off after the reference speed has changed is an additional disadvantage. High rising times can be reduced by enhancing the membership functions.

The outcomes of the simulation have shown how well the Neuro-Fuzzy controller performs dynamically and how resilient it is to unforeseen loads and transient phases. The suggested intelligent controller is proven to be more effective than the fuzzy tuned PI controller.

With its modified design, the ANFIS Controller's performance was found to be reasonable. The speed of the induction machine decreased whenever it was loaded, but not much. The system's peak overshoot was found to have decreased in comparison to the previous design, but the rise

and settling times were mostly unaffected. This means that other programs can now make use of this controller.

ANFIS controllers generally provide better performance in terms of adaptability, robustness, and handling non-linearities compared to PI controllers.

The suggested controller is assessed using simulations for a range of drive system operating circumstances. The findings show how well these control structures work to enhance drive system performance.

6.2 Recommendation

There are numerous practical uses for the controller described in this thesis for induction motors. It is possible to examine its feasibility in the pertinent applications and make adjustments as necessary. For controller tuning, a variety of techniques, including genetic algorithms, can be applied. Furthermore, this thesis can be used to build other Controllers rather than just ANFIS. The thesis can also be expanded for hardware implementation using many technologies like digital signal processing(DSP), Application specific integrated chips(ASICs) and others which is suitable for it.

References

[1] J. G. Ziegler and N. B. Nichols. Optimum settings for automatic controllers. *Transactions of the ASME*, 64:759–768, 1942.

[2] G. Mallesham and A. Rajani. Automatic tuning of pid controller using fuzzy logic. In *Proceedings of the 8th International Conference on Development and Application Systems*, pages 120–126, 2006.

[3] Abdullah I. Al-Odienat and Ayman A. Al-Lawama. The advantages of pid fuzzy controllers over the conventional types. *American Journal of Applied Sciences*, 5(6):653–658, 2008.

[4] L. A. Zadeh. Fuzzy sets. *Information and Control*, 8:338–353, 1965.

[5] M. Chow, A. Menozzi, and F. Holcomb. On the comparison of emerging and conventional techniques for dc motor control. In *Proceedings of the IECON*, pages 1008–1013, 1992.

[6] Pavol Fedor and Daniela Perduková. A simple fuzzy controller structure. *Acta Electrotechnica et Informatica*, 5(4):1–4, 2005.

[7] D. P. Kothari and I. J. Nagrath. *Electric Machines*. Tata McGraw-Hill Education Private Limited, 2004.

[8] Ramón C. Oros, Guillermo O. Forte, and Luis Canali. Scalar speed control of a d-q induction motor model using fuzzy logic controller. In *Proceedings of the Conference*, 2023.

[9] I. J. Nagrath and M. Gopal. *Control Systems Engineering*. New Age International Publishers, 2007.

[10] R. Ouiguini, K. Djeffal, A. Oussedik, and R. Megartsi. Speed control of an induction motor using the fuzzy logic approach. In *Proceedings of the ISIE'97 Conference*, volume 3, pages 1168–1172. IEEE, 1997.

[11] Gopal K. Dubey. Fundamentals of electrical drives. In *Fundamentals of Electrical Drives*, chapter 6. Narosa Publishing House Pvt. Ltd., 2001.

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

[13] G. El-Saady, A. M. Sharaf, A. Makky, M. K. Sherriny, and G. Mohamed. A high performance induction motor system using fuzzy logic controller. In *Proceedings of the IEEE Conference*, pages 1058–1061. IEEE, 1994.

[14] 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. Master's thesis, School of Electrical and Computer Engineering (SECE), Addis Ababa University (AAU), Addis Ababa, 2019.

[15] Biniyam Abera. Speed sensorless, foc of imd using anfis controller and ann estimator. Master's thesis, School of Electrical and Computer Engineering (SECE), Addis Ababa Institute of Technology (AAiT), Addis Ababa, 2020.

[16] Yau-Tze Kao and Chang-Huan Liu. Analysis and design of microprocessor-based vector-controlled induction motor drives. *IEEE Transactions on Industrial Electronics*, 39:46–54, February 1992.

[17] J.-S. R. Jang, C.-T. Sun, and E. Mizutani. *Neuro-Fuzzy and Soft Computing*. Pearson Education Pte. Ltd., 1997.

[18] J. Martínez García and J. A. Domínguez. Comparison between fuzzy logic and pi controls in a speed scalar control of an induction machine. In *Proceedings of the CIRCE Conference*, 2023.

[19] Sujeet Kumar Soni, Manish Khemariya, and Anand Singh. Sensorless speed control of induction machine with adaptive-neuro fuzzy technique integrated mras module. *Journal of Electrical Engineering*, 2022.

[20] Mrinal Buragohain. *Adaptive Network Based Fuzzy Inference System (ANFIS) as a Tool for System Identification with Special Emphasis on Training Data Minimization*. PhD thesis, Department of Electronics and Communication Engineering (DECE), Indian Institute of Technology (IIT) Guwahati, Guwahati, India, 2008.

[21] Howard Demuth, Mark Beale, and Martin Hagan. *Neural Network Toolbox 5 User's Guide*. The MathWorks, Inc., 3 Apple Hill Drive, Natick, MA 01760-2098, USA, 2007.

Appendix
Fuzzy .fis Code

```
    [System]
Name='FSMC'
Type='Takagi-Sugeno (T-S)'
Version=2.0
NumInputs=2
NumOutputs=1
NumRules=49
AndMethod='min'
OrMethod='max'
ImpMethod='min'
AggMethod='max'
DefuzzMethod='centroid'

[Input1]
Name='input1'
Range=[-1.4 1.4]
NumMFs=8
MF1='NB':'trapmf',[-4.6 -1.4 -0.9667 -0.7003]
MF2='NM':'trimf',[-1 -0.667 -0.25]
MF3='NS':'trimf',[-0.667 -0.25 0]
MF4='ANZ':'trimf',[-0.25 0 0]
MF5='APZ':'trimf',[0 0 0.25]
MF6='PS':'trimf',[0 0.25 0.667]
MF7='PM':'trimf',[0.25 0.667 1]
MF8='PB':'trapmf',[0.7003 0.9667 1.4 4.6]

[Input2]
Name='input2'
Range=[-1.4 1.4]
NumMFs=8
```

```
MF1='NB':'trapmf',[-4.6 -1.4 -0.9667 -0.7003]
MF2='NM':'trimf',[-1 -0.667 -0.25]
MF3='NS':'trimf',[-0.667 -0.25 0]
MF4='ANZ':'trimf',[-0.25 0 0]
MF5='APZ':'trimf',[0 0 0.25]
MF6='PS':'trimf',[0 0.25 0.667]
MF7='PM':'trimf',[0.25 0.667 1]
MF8='PB':'trapmf',[0.7003 0.9667 1.4 4.6]


[Output1]
Name='output1'
Range=[-1 1]
NumMFs=7
MF1='NB':'trimf',[-1.333 -1 -0.6666]
MF2='NM':'trimf',[-1 -0.6666 -0.3334]
MF3='NS':'trimf',[-0.6666 -0.3334 0]
MF4='AZ':'trimf',[-0.3334 0 0.3334]
MF5='PS':'trimf',[0 0.3334 0.6666]
MF6='PM':'trimf',[0.3334 0.6666 1]
MF7='PB':'trimf',[0.6666 1 1.334]

[Rules]
1, 1 - 1 (1) : 1
2, 8 - 2 (1) : 1
3, 7 - 3 (1) : 1
4, 2 - 4 (1) : 1
5, 6 - 5 (1) : 1
6, 1 - 6 (1) : 1
7, 2 - 7 (1) : 1
8, 3 - 8 (1) : 1
9, 8 - 1 (1) : 1
10, 1 - 2 (1) : 1
11, 6 - 3 (1) : 1
```

12, 7 - 4 (1) : 1

13, 8 - 5 (1) : 1

14, 3 - 6 (1) : 1

15, 4 - 7 (1) : 1

16, 5 - 8 (1) : 1

17, 6 - 1 (1) : 1

18, 8 - 2 (1) : 1

19, 7 - 3 (1) : 1

20, 2 - 4 (1) : 1

21, 4 - 5 (1) : 1

22, 5 - 6 (1) : 1

23, 8 - 7 (1) : 1

24, 8 - 8 (1) : 1

25, 4 - 1 (1) : 1

26, 2 - 2 (1) : 1

27, 6 - 3 (1) : 1

28, 1 - 4 (1) : 1

29, 1 - 5 (1) : 1

30, 3 - 6 (1) : 1

31, 4 - 7 (1) : 1

32, 5 - 8 (1) : 1

33, 7 - 1 (1) : 1

34, 1 - 2 (1) : 1

35, 3 - 3 (1) : 1

36, 4 - 4 (1) : 1

37, 5 - 5 (1) : 1

38, 6 - 6 (1) : 1

39, 7 - 7 (1) : 1

40, 4 - 8 (1) : 1

41, 5 - 1 (1) : 1

42, 7 - 2 (1) : 1

43, 4 - 3 (1) : 1

44, 5 - 4 (1) : 1

45, 2 - 5 (1) : 1

46, 1 - 6 (1) : 1

47, 3 - 7 (1) : 1

48, 2 - 8 (1) : 1

49, 6 - 8 (1) : 1