Realization of Speed Control for a Separately Excited DC Motor Using Fuzzy Logic

Dawit Mekonnen Gebremichael

A Thesis Submitted to the
Department of Electrical Power and Control Engineering
School of Electrical Engineering and Computing

Presented in Partial Fulfillment of the Requirement for the Degree of Master of Science in Electrical Power and Control Engineering
(Specialization in Power Electronics)

Office of Graduate Studies
Adama Science and Technology University

Adama, Ethiopia

October, 2019

Realization of Speed Control for a Separately Excited DC Motor Using Fuzzy Logic

Dawit Mekonnen Gebremichael

Advisor: Tefera T.Yetayew(PhD)

A Thesis Submitted to the
Department of Electrical Power and Control Engineering
School of Electrical Engineering and Computing

Presented in Partial Fulfillment of the Requirement for the Degree of
Master of Science in Electrical Power and Control Engineering
(Specialization in Power Electronics)

Office of Graduate Studies
Adama Science and Technology University

Adama, Ethiopia

October, 2019

APPROVAL OF BOARD OF EXAMINERS

We, the undersigned, members of the Board of Examiners of the final open defense by Dawit Mekonnen Gebremichael have read and evaluated his thesis entitled “Realization of Speed Control for a Separately Excited DC Motor Using Fuzzy Logic ” and examined the candidate. This is, therefore, to certify that the thesis has been accepted in partial fulfillment of the requirement of the Degree of Master’s in Power Electronics.

Dr. Tefera T.Yetayew	____________	____________
Advisor	Signature	Date
____________________	____________	____________
Chairperson	Signature	Date
____________________	____________	____________
Internal Examiner	Signature	Date
____________________	____________	____________
External Examiner	Signature	Date

DECLARATION

I hereby declare that this MSc Thesis is my original work and has not been presented for a degree in any other university, and all sources of material used for this thesis have been duly acknowledged.

Name: Dawit Mekonnen Gebremichael

Signature: ____________________

This MSc Thesis has been submitted for examination with my approval as thesis advisor.

Name: Dr. Tefera T.Yetayew

Signature: ____________________

Date of submission: ____________________

ADVISOR'S APPROVAL SHEET

To: Electrical Power and Control Engineering Department
Subject: Thesis Submission

This is to certify that the thesis entitled "Realization of Speed Control for a Separately Excited DC Motor Using Fuzzy Logic" submitted in partial fulfillment of the requirements for the degree of Master's in Power Electronics, the Graduate program of the department of Electrical Power and Control Engineering, and has been carried out by Dawit Mekonnen Gebremichael Id. No. GSR/0049/09, under my supervision. Therefore, I recommend that the student has fulfilled the requirements and hence hereby he can submit the thesis to the department.

Dr. Tefera T.Yetayew	____________________	____________________
Advisor	Signature	Date

ACKNOWLEDGEMENTS

Firstly and more importantly, my thanks and praise goes to the Almighty God and Saint Merry for their grace and blessings throughout the entire work. Next to that, I would like to express my sincere gratitude to my advisor Dr. Tefera T.Yetayew for his continuous support and guidance for the successful completion of this thesis.

My thanks are extended to Mr. Desalegn Abebaw, Dr. Mesfin Abebe, Prof. Jayant Shekhar, Mr. Esayas Meshesha, Mr. Samuel Managdew, Smart Software and Systems (3S) SIG members, and academic and research assistants of Electrical Engineering and Computing School for their assistance and encouragement throughout my work.

I do want to thank my friends, Power System and Power Electronics students for their support and time we had together. Last but not least, I would like to thank my family for their great support during my study.

Adama, October, 2019

Dawit Mekonnen

TABLE OF CONTENTS

CONTENTS **PAGE**

LIST OF TABLES

LIST OF FIGURES

FIGURE **PAGE**

SYMBOLS

B_m	Viscose Friction Coefficient
ce	Change in Error
e	Error
E_b	Motor Back Electromotive Force
E_c	Controller Output Voltage
f	Frequency
I_a	Armature Current
$I_a(ref)$	Reference Armature Current
I_{astart}	Starting Current
I_f	Field Current
I_L	Load Current
I_o	Output Current
J	Moment of Inertia
K_1	Speed Filter Gain
K_2	Current Filter Gain
K_a	Armature Constant
K_c	Controller Gain
K_i	Integral Gain
K_n	Speed Controller Gain
K_m	Back EMF Constant
K_p	Proportional Gain
K_t	Converter Gain
K_v	Motor Voltage Constant
L_a	Armature Inductance
L_f	Field Inductance
M_p	Maximum Overshoot
P_d	Power Developed
R_a	Armature Resistance
R_{ext}	External Resistance
R_f	Field Resistance
T	Time Period
T_1	Speed Feedback Filter Time Constant
T_2	Current Filter Time Constant
T_a	Armature Time Constant
T_c	Current Controller Time Constant
T_d	DC Motor Torque
T_L	Load Torque
T_m	Electromechanical Time Constant

T_n	Speed Controller Time Constant
T_{off}	Off Time
T_{on}	On Time
t_d	Delay Time
t_p	Peak Time
t_r	Rise Time
t_s	Settling Time
τ_i	Integral Time
u	Control Signal to the Process
V	Voltage
V_a	Armature Voltage
$V_a(ref)$	Reference Armature Voltage
V_c	Comparator Voltage
V_{dc}	DC Voltage
V_f	Field Voltage
V_L	Load Voltage
V_T	Terminal Voltage
V_{tri}	Triangular Carrier Voltage
V_o	Output Voltage
ω	Angular Velocity
$\omega(fb)$	Feedback Angular Velocity
$\omega(ref)$	Reference Angular Velocity
Φ	Field Flux

ABBREVIATIONS

AC	Alternating Current
ANN	Artificial Neural Network
BEV	Battery Electric Vehicles
BJT	Bipolar Junction Transistor
BOA	Bisector of Area
COG	Center of Gravity
DC	Direct Current
Emf	Electromotive Force
FIS	Fuzzy Inference System
FL	Fuzzy Logic
FLC	Fuzzy Logic Controller
GA	Genetic Algorithm
GTO	Gate Turn Off Thyristor
IGBT	Insulated Gate Bipolar Junction Transistor
LCD	Liquid Crystal Display
MATLAB	Matrix Laboratory
mf	Membership Function
MIMO	Multiple Input Multiple Output
MOM	Mean of Maximum
MOSFET	Metal Oxide Semiconductor Field Effect Transistor
P	Proportional
PD	Proportional-Derivative
PI	Proportional-Integral
PID	Proportional-Integral-Derivative
PSO	Particle Swarm Optimization
PWM	Pulse Width Modulation
SISO	Single Input Single Output
TRC	Time Ratio Control

ABSTRACT

The DC motors are used as an adjustable speed machine for wide range of operation. The four main reasons for the usage of DC motors are reliability, simplicity, ease of application and favorable cost. At the same time, the DC motor drive is less complex compared to AC motor drives. Thus, the use of electric DC motor is increasing steadily. However, it faces efficiency problems on the dynamic response for high speed sensitive applications. This is usually due to the power electronics devices particularly converters and algorithm of inference used. To counteract these drawbacks, four quadrants operation of chopper based speed control of separately excited DC motor using fuzzy logic is introduced in the study. In this thesis, the effect of number of input error membership functions on the speed performance is studied, simulation is performed for both fuzzy and PI controllers using MATLAB/Simulink software and accordingly comparison is made on their speed performances. From the results obtained efficiency problems observed during speed control in dynamically changing environment has resolved using fuzzy logic controller and better speed responses are obtained as compared to PI controllers. In addition to this, as the number of error membership functions increases, an improvement on the speed tracking has noticed and hence, reduced steady state error and smooth dynamic response is obtained.

Keywords:- *Four quadrant chopper, Fuzzy logic control, IGBT, Separately excited DC motor, Speed control*

CHAPTER ONE

INTRODUCTION

1.1 Background

Nowadays, development of high performance motor drives is very essential for industrial applications. A high performance motor drive system must have good dynamic speed command tracking and load regulating response [1]. Compared to AC drives, DC drives are less complex [2]. Thus, DC drives because of their simplicity, ease of application, reliability and favorable cost have long been a backbone of industrial application [3]. In addition to this, DC motors are controllable over a wider range of speeds by proper adjustment of the terminal voltages and are well known by their excellent control of speed for acceleration and deceleration [4]. However, it faces efficiency problems in speed control of dynamically changing environment.

The speed of motor can be controlled in different methods such as armature voltage control, field flux control method and armature resistance control method [5]. Thus the goal of this thesis is to control the speed of DC motor drive for all four quadrants operation of chopper using fuzzy logic with armature voltage control method to achieve the speed for below the rated speed. And then simulate in MATLAB to analyze the observed result accordingly. Thus it will provide solution for efficiency problems observed in the speed control of DC motors operating in dynamically changing environment with the existing conventional controllers.

DC drives are widely used for adjustable speed control, frequent starting, better speed regulation, braking and reversing. Some examples of DC motor application includes e-bikes, e-bicycle, e-scooters, tread- mill, potable sewing machine, drill machine, railway engine (traction), paper mills, mine winders, hoists, elevators, printing presses, textile mills, machine tools, cranes and rolling mills etc. Thus in many different applications, it is required to control speed of DC motor [5] - [7].

There are conventional (P, PI, PID) controllers which are used for DC motor speed control but have drawbacks such as undesired high overshoot, needs system mathematical model, requires constant tuning of parameters and works on SISO systems [8]. Thus, these drawbacks are eliminated by using Fuzzy Logic Controller (FLC) which is used to apply the required control voltage that will be sent to DC motor based on fuzzy rule base of motor speed error and change in speed error. Fuzzy logic control is a control algorithm based on a linguistic control strategy, which is derived from expert knowledge into an automatic control strategy. The operation of a Fuzzy Logic Controller is based on qualitative knowledge about the system being controlled.

In this speed control scheme the power electronic converter, chopper is used as power controller. A chopper is a static power electronic device that converts fixed DC input voltage to a variable DC output voltage. In this thesis, the power semiconductor device, IGBT is used for speed control of separately excited DC motor when using four quadrants chopper. The power semiconductor devices used for a chopper circuit can be force-commutated thyristor, power BJT, MOSFET and IGBT. GTO based choppers can also be used. These devices are generally represented by a switch. When the switch is off, no current can flow. Current can flow through the load when switch is on [9].

1.2 Statement of the Problem

This thesis focuses on the idea of a research question “How to improve efficiency problems observed with PI controllers in the speed control of separately excited DC motor drives used for dynamically changing environments”. It is because,

- The use of conventional controllers (P, PI, PID) enable designers to improve system performance with the need to develop system mathematical models. Whereas, the use of fuzzy logic improves system performance without the need for system mathematical models. Thus, conventional controllers fail to provide the speed controlling for a non-linear load which is overcome by fuzzy logic controllers.

- Conventional controllers have very high overshoot of currents initially, which is undesired. This undesired overshoot leads to undesired results and negatively impacts the DC motor such as worn out brush of motors.

- The conventional controllers need constantly adjustment in order to achieve better control performance. Whereas fuzzy self tuning parameters controller is automatically adjusted parameters according to the speed error and the rate of change in speed error. So fuzzy logic controller requires no continuous tuning and human manipulation is reduced.

- Conventional controllers work only in SISO systems, while fuzzy controllers work on both SISO and MIMO systems and it provides better result than conventional controllers.

Thus, in order to overcome the problem discussed above, a fuzzy logic controller is used for the control purpose.

1.3 Objective of the Study

1.3.1 General Objective

The main objectives of this thesis is to control the speed of separately excited DC motor used in dynamically changing environment for four quadrants chopper operation by using fuzzy logic.

1.3.2 Specific Objectives

- To develop linear model for the separately excited DC motor using transfer function.
- To formulate fuzzy logic control algorithm and PI control system.
- To simulate the modeled system with fuzzy and PI controllers using MATLAB/Simulink software.
- To study the effect of number of error membership functions on the speed performance.
- To compare fuzzy and PI controllers on the speed performance.

1.4 Significance of the Study

Over the past decade, the DC machine have gained high popularity in the industry, especially where high performance is required owing to higher efficiency, simple and adjustable speed control with braking and reversing, ease of application, reliability and favorable cost. Even though, with recent advancement in power converters the AC motor drives are becoming high competitive with DC motor drives, the future trend is towards DC drives. In addition to armature voltage control, providing regenerative braking of motors and capability of returning energy back to supply, makes DC chopper play a significant role in the newly emerging technologies such as battery electric vehicles (BEV), marine hoists, trolley cars, forklift trucks etc.

Fuzzy logic reasoning is used for the purpose of reliability and rapid response of leveling power. Recent developments in artificial-intelligence-based control have brought into focus a possibility of replacing a PI speed controller with a fuzzy logic (FL) equivalent. That is why, in this research work fuzzy logic controller is used to improve the efficiency problems observed in the speed control of separately excited DC motor drives used for dynamically changing environments when using PI controllers. Fuzzy logic speed control is sometimes seen as the ultimate solution for high-performance drives of the next generation. Thus nowadays, FLC applications are successfully used in many fields including automatic focus cameras, household materials such as dishwashers, automobile industry etc [10].

1.5 Delimitation

This research work focuses on separately excited DC motor drives that are used in dynamically changing environment for all four quadrant chopper with simulation using MATLAB/Simulink software. This is because DC drives are widely used than AC drives, in applications, such as, good speed regulation, adjustable speed control, braking and reversing frequent starting, and position control systems.The techniques of speed control for DC motors are less expensive and simpler than AC motors. The significant feature of the separately excited DC motor configuration is its ability to produce high starting torque at low operation speed [11].

An advantage to a separately excited shunt field is the ability of a variable speed to provide independent control of the armature and field. Where as Permanent magnet DC motors are limited to the amount of load they can drive and torque is usually limited to 150 percent of rated torque to prevent demagnetization of the permanent magnets, Series DC motors cannot be used where a constant speed is required under varying loads and its speed with no load increases to the point where the motor can become damaged. Thus, series connected DC motors generally are not suitable for use on most variable speed drive application. In Compound DC motors the series field can cause control problems in variable speed drive applications and is generally not used in four quadrant drives.

1.6 Limitations

While dealing with this paper there are some limitations experienced.These includes,

- Dealing with fuzzy logic control needs human experience and also
- It includes high computations, so there is a high computation burden.

1.7 Organization of the Thesis

The organization of the thesis is as follows:

- Chapter 1 has presented an introduction on the research background, focus idea of research question, objective and significance of the research work with its scope.
- Chapter 2 discusses on literature reviews of different papers related to this research and theory about separately excited DC motor with its speed control, modeling of the DC motor, four quadrant chopper, fuzzy logic and PI controllers.
- Chapter 3 provides the procedure of the whole research work and designs of both fuzzy and PI controllers.

- Chapter 4 shows the results obtained and discusses by categorizing into three parts as effect of number of membership functions on the speed performance, simulation results of FL and PI controllers and comparison of controllers on the speed performance.
- Chapter 5 summarizes all the research work done with its obtained results and hence, draws conclusions and recommendations for further studies are also given.

CHAPTER TWO

LITERATURE REVIEW

2.1 Empirical Review

This section introduces related and most recent scientific researches with a focus on fuzzy and conventional controllers. In the papers different controllers for DC motor speed control methods have used. It has been discussed the gap and has identified issues to improve all these works.

A speed control of DC Motor using Fuzzy Logic Controller in MATLAB has been presented by Rajagiri et al. [12]. This speed control scheme comprises of two parts; software and hardware implementation. The software part aims to design and develop a fuzzy logic controller in MATLAB/Simulink whereas the hardware part consists of DC motor driver for converting AC voltage into variable DC voltage; and data acquisition card (NI PCI-6221 with 68 Pin) which is used as the hardware interface between hardware and software. The designed Fuzzy Logic Controller for the running of DC motor, results in a better response compared to the normal response of DC motor. It also describes from the experimental results on the real plant, as the proposed Fuzzy Logic Controller's response is very good and it is robust, faster and flexible. This shows control of the speed and hence with the help of FLC shorter settling time can be achieved by tuning the control rules, membership functions and universe of discourse of the output variable. The paper states as this all shows an idea of DC motor speed control with simple software application with good hardware results.

Mahajan [13] has made modelling and simulation of a DC motor driven geared system using Simulink. The equations have evaluated for a combination of electric and geared system with 240 V input voltage which is provided in the form of a step input and transfer function has formulated for both parts with a reference speed of 2500 rpm from no load condition to full load condition. The simulation has done for an open and closed loop response with unstable results, that is, rise time and settling time values are too high. With the application of a PID controller uncontrolled and controlled simulation are performed under load conditions from no load to full load cases and hence, the settling time is reduced by 70% with a decrease in speed by 200 rpm and then again settles. The author has also figured out as this simulation can be used as robotic gripper actuation or drive system of electric vehicle.

Hamoodi et al. [14] have discussed speed control of separately excited DC motor employing PID controller which is implemented by MATLAB/Simulink and practically. The DC motor here used is of Terco Company type, and its speed below and up to the rated speed has been achieved by changing the armature voltage. A comparison is made by Simulink modeling circuit and practical connection, and the experimental relationship between voltage-speed and torque-speed characteristics has been investigated for both as well. The output performance which is obtained by normalized value in PID is very close and near to accuracy which shows as the PID constants (Kp, Ki, Kd) have made an improvement on the speed response curves. From the dynamic behavior of the speed response with closed loop circuit more improvements have seen as compared with that of open loop circuit using Ziegler method.

The paper presented by Gokulkannan et al. [15] mainly deals with speed control method of DC motor by varying armature voltage using chopper as power converter and PI as speed and current controller on MATLAB/Simulink. The MATLAB simulation modelling has used IGBT due to its best performance of speed control, fast switching and low losses. Here 5HP, 240V separately excited DC motor and additionally 300V DC supply are given to the field with constant circuit load of 20Kg. For the discrete PI controller, the proportional gain (Kp) and integral gain (Ki) are chosen to be 0.05 and 1.5 respectively using Ziegler-Nichols tuning method. Using MATLAB/Simulink the overall chopper Simulink model is done and analysed. Following this, a comparison is made with different types of controllers (P, PD, PI and PID); the speed and performance of the DC motor is measured. Hence, it is found Proportional plus Integral (PI) as the most preferred controller, which is designed to eliminate the need for continuous operator attention thus provides automatic control to the system. Finally, the simulation is done successfully for the chopper fed speed control of separately excited DC motor at below the rated speed using PI controller.

An automatic closed loop speed control of DC motor is presented by Kosare et al. [16], using IGBT which provides a low impedance adjustable “DC” voltage for the motor armature, thereby providing speed control. It has been discussed a systematic development of a closed loop DC motor linear speed control scheme with armature voltage control full bridge converter, identifies and describes the choices with current and speed controller, related to P and PI controllers. The MATLAB simulation for speed control of 0.5-hp 1450 rpm separately excited DC motor with speed feed back through a tacho-generator is successfully implemented using Pic 16 micro-controller with LCD showing the reference speed and feedback, and from different cases taken performance of DC motor is more better in case of PI than P controller approach.

Shantanu and Bind [17] have demonstrated the comparative analysis of PI and FLC schemes for speed control of separately excited DC motor by varying armature voltage, using chopper circuit. Authors discussed the results of speed controllers PI, FLC and without controller graphically using MATLAB/simulink. With obtained results, PI controller found very much useful as it improves steady state response of the system, and its time of response was high but disadvantageous in tuning of PI controllers, high ripples in torque and current, and high peak overshoot very much drop of speed when load is applied over it. These drawbacks were highly improved by using FLC, as tuning is not required in FLC it reduced the ripples in torque and current very much and there was no peak overshoot in the system, but little bit sluggish as compared to PI controller. By comparing both controllers, FLC has proved to be much superior than PI controller.

Hiware and Deshmukh [18] have demonstrated a speed control of separately excited DC motor by using the IGBT based chopper employing PI-type controller as speed and current controller with the aim to eliminate the delay and providing fast response to sudden changes in motor speed. The motor speed below and up to the rated speed has been achieved by changing the armature voltage with an advantage that speed varies proportionally with armature voltage and varies inversely with field voltage by keeping field and armature voltage constant respectively. The simulation is done in MATLAB under condition of changing load, changing reference speed and changing input voltage, initially with a simplified open loop model, resulting with uncontrolled output speed. And finally successful DC motor speed control is analyzed using closed loop model of PI-type controller.

Nagarajan et al. [19] have described a closed loop speed control scheme of separately excited DC motor using chopper (buck converter) as power converter and Proportional-Integral type controller as a speed and current controller has been described in this paper. MOSFET switch is used in order to have best performance of voltage control, fast switching and low losses. PI controller parameters are selected using Ziegler-Nichols tuning rules for educational guess and also a comparison is made in PI against P and PID. Obtained result showed a better response in speed control of separately excited DC motor under below the rated speed in PI than P and PID controllers.

Suman and Giri [20] have presented an approach to investigate the speed control of separately excited DC motor using conventional controllers (such as P, PI, PID) and FLC. The use of FLC is investigated about mamdani type fuzzy inference system (FIS) that has been five membership functions where both are same for input and output parameters, and compared with conventional controllers. The FLC has successfully controlled the speed of DC motor with least amount of overshoot, lower rise time, and bare minimum transient and steady state parameters than other conventional controllers.

Ozdemir et al. [21] have described a speed control scheme for a separately excited DC motor traction motor in light metro vehicle which fed from a PWM-DC-DC converter using fuzzy logic controller. In fuzzy control the motor speed range has well covered seven membership functions using non linear model, and the designed drive system has been simulated and results are compared with those obtained by a conventional PID controller. With the fuzzy controller, the overshoot is dramatically reduced, oscillation is effectively eliminated and its robustness is proved. From simulation results, it has been concluded that as the fuzzy logic controller was superior to conventional controller (PID) among several key parameters, including accuracy in stopping at plat forms, rider comfort (jerkiness of acceleration and braking) and energy conservation.

From the related research works discussed it has observed that, different controllers have used for the speed control of DC motor. Simulations and hardware implementations have performed in some of these related research works which is appreciated. But the four quadrants operation of the DC motor and/or comparison with fuzzy logic controller in the dynamically changing environment has not been covered. Hence, the gaps observed in these related works are solved in this research work.

2.2 DC Motors

Electrical machines are energy converting devices which convert one form of energy to another form and vice verse. Thus motors take an input of electrical energy in order to produce an output of mechanical energy. Electrical motors can be broadly classified into two different categories called DC (Direct Current) and AC (Alternating Current) [22].

DC motors are motors which run on Direct Current from battery or DC power supply in order to convert electrical energy into mechanical energy. Direct Current is the term used to describe electricity supplied to DC motors at a constant voltage. When DC power supply or a battery is connected between DC motor and electric loads, the motor convert electrical energy to mechanical work as the output shafts rotate [23].

DC motors can also be classified based on the electrical connections of the field winding and the armature winding. These different connections of winding leads to machines operating with different characteristics. The field winding can be either self-excited or separately-excited. That is, in self-excited DC motors terminals of the winding can be connected across the input voltage terminals whereas in separately-excited case it fed from a separate voltage source. Further, in self-excited motors, the field winding can be connected either in series or in parallel with the armature winding. These different types of connections give rise to very different types of machines. There are different kinds of DC motors, but their principle of operation is the same. This paper focus on speed control of a separately excited DC motor.

2.3 Separately Excited DC Motor

2.3.1 Construction and Operation of Separately Excited DC Motor

A DC motor is comprised of three main parts, a current carrying conductor called armature, a circuit for magnetic field provided by magnets of poles called field system and a commutator that switches the direction of current in the armature as it passes a fixed point in space. Separate excited DC motor has two important parts, one is armature winding and the other is field winding which are electrically separate from each other [24]. That means the I_f is independent of the I_a and thus, any change in the armature current has no effect on the field current.

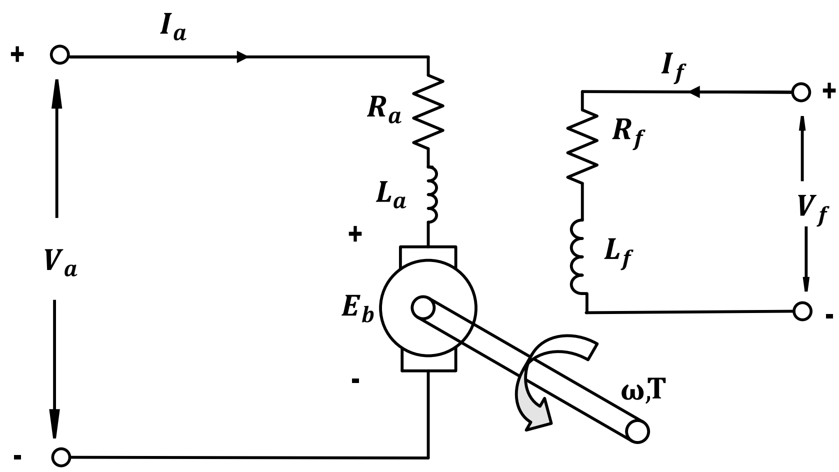


Figure 2.1: Schematic diagram of separately excited DC motor [25]

When a separately excited motor is excited by a field current of I_f and an armature current of I_a flows in the circuit, armature flux and field flux are created. Armature current is supplied to the rotor via brush and commutator setup to reduce friction. And due to magnetic interaction between the armature flux and field flux, the motor develops a back emf and a torque on the rotor to balance the load torque at a particular speed. Field winding is generally provided on the stator while the armature is provided on the rotor.

2.3.2 Basic Field, Armature and Torque Equations

Instantaneous field voltage is given in Equation (2.1).

$$V_f = I_f R_f + L_f \frac{dI_f}{dt} \tag{2.1}$$

Where R_f and L_f are the field resistor and inductor, respectively. Instantaneous armature voltage is given in Equation (2.2).

$$V_a = I_a R_a + L_a \frac{dI_a}{dt} + E_b \tag{2.2}$$

Where R_a and L_a are the armature resistor and inductor, respectively. The motor back emf, which is also known as speed voltage is expressed in Equation (2.3).

$$E_b = K_v \omega I_f \tag{2.3}$$

Where K_v is the motor voltage constant (in V/A-rad/s) and ω is the motor speed (in rad/s). The torque developed by the motor is given in Equation (2.4).

$$T_d = K_t I_f I_a \tag{2.4}$$

Where ($K_t = K_v = K$) is the torque constant (V/A-rad/s). Sometimes it is written as shown in Equation (2.5).

$$T_d = K_t \Phi I_a \tag{2.5}$$

For normal operation, the developed torque must be equal to the load torque plus the friction and inertia as shown in Equation (2.6).

$$T_d = J\frac{d\omega}{dt} + B_m\omega + T_L \tag{2.6}$$

Where B_m is viscous friction constant (N.m/rad/s), T_L is load torque (N.m) and J is inertia of the motor (Kg.m^2).

Steady-State Operation

Under steady-state operations, time derivatives is zero. Assuming the motor is not saturated the field voltage in the circuit is shown in Equation (2.7).

$$V_f = I_f R_f \tag{2.7}$$

The back emf is given by Equation (2.8).

$$E_b = K_v \omega I_f \tag{2.8}$$

The armature voltage in the circuit is given by the Equation (2.9).

$$\begin{aligned} V_a &= I_a R_a + E_b \\ &= I_a R_a + K_v \omega I_f \end{aligned} \tag{2.9}$$

The motor speed can easily be derived as shown in the Equation (2.10).

$$\omega = \frac{(V_a - I_a R_a)}{K_v I_f} \tag{2.10}$$

If R_a is a small value (which is usual), or when the motor is lightly loaded. i.e. I_a is small. Then the motor speed becomes as shown Equation (2.11).

$$\omega = \frac{V_a}{K_v I_f} \tag{2.11}$$

That is I_f the field current is kept constant, the motor speed depends only on the supply voltage. The developed torque is then given in Equation (2.12).

$$\begin{aligned} T_d &= K_t I_f I_a \\ &= B_m \omega + T_L \end{aligned} \tag{2.12}$$

The required power is given by Equation (2.13).

$$P_d = T_d \omega \tag{2.13}$$

From these derived equations several important facts can be deduced for a steady state operation of DC motor. That is, for a fixed field current, or flux (I_f), the torque demand can be satisfied by varying the armature current (I_a). Also the motor speed can be varied by controlling either V_a (voltage control) or V_f (field control). Therefore, these observations lead to the application of variable DC voltage for controlling the speed and torque of DC motor.

2.3.3 Torque-Speed Characteristics

For an effective use of a DC motor in an application, it is necessary to understand its characteristic curves. Since every motor has a specific torque/speed curve and power curve, the relation between torque and speed is important in choosing a DC motor for a particular application.

The total emf induced in the DC motor by several coils wound on the rotor can be expressed in Equation (2.14).

$$E_b = K \Phi \omega \tag{2.14}$$

Where K is an armature constant, and is related to the geometry and magnetic properties of the motor, and ω is the speed of rotation. The electrical power generated by the machine is given by the Equation (2.15).

$$\begin{aligned} P_d &= E_b I_a \\ &= K \Phi \omega I_a \end{aligned} \tag{2.15}$$

Applying KVL in the armature circuit the armature voltage is given by Equation (2.16).

$$V_a = I_a R_a + E_b \tag{2.16}$$

Where V_a is voltage applied to the armature terminals of the motor and R_a is the resistance of the armature winding. From Equation (2.14) and Equation (2.16), we have two expressions for the induced voltage as give in Equation (2.17).

$$\begin{aligned} E_b &= K\Phi\omega \\ &= V_a - I_a R_a \end{aligned} \tag{2.17}$$

The torque developed in the rotor (armature) is given by the Equation (2.18).

$$T_d = K\Phi I_a \tag{2.18}$$

From Equation (2.18), the current in the armature winding can be found as given by the Equation (2.19).

$$I_a = \frac{T_d}{K\Phi} \tag{2.19}$$

Substituting for I_a in Equation (2.17) and rearranging the terms will results in Equation (2.20).

$$V_a - K\Phi\omega = R_a(\frac{T_d}{K\Phi}) \tag{2.20}$$

Therefore, the torque developed in the rotor can be expressed as given by Equation (2.21).

$$T_d = \frac{K\Phi}{R_a}(V_a - K\Phi\omega) \tag{2.21}$$

This equation shows the relationship between the torque and speed of a separately excited DC motor. If the terminal voltage V_a and flux Φ are kept constant, the torque-speed relationship is a straight drooping line.

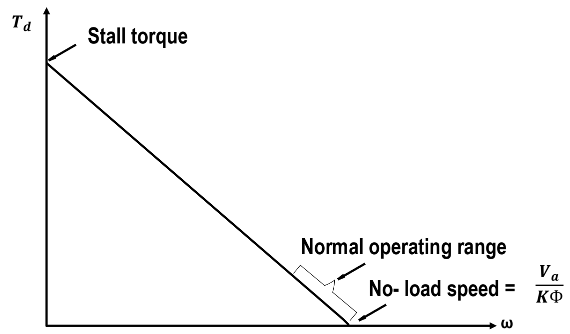


Figure 2.2: Torque-speed characteristics of separately excited DC motor [26]

The graph in the Figure 2.2 shows a torque/speed curve of a separately excited DC motor. Note that torque is inversely proportional to the speed of the output shaft. In other words, there is a trade off between how much torque a motor delivers, and how fast the output shaft spins. Motor characteristics are frequently given as two points on this graph:

- The stall torque, represents the point on the graph at which the torque is a maximum, but the shaft is not rotating.

- The no load speed, is the maximum output speed of the motor (when no torque is applied to the output shaft).

The motor load determines the final operating point on the torque curve. As illustrated in the Figure 2.3, when a motor is connected to drive a load, the interaction of torque demanded by the load and the torque produced by the motor determines the point of operation.

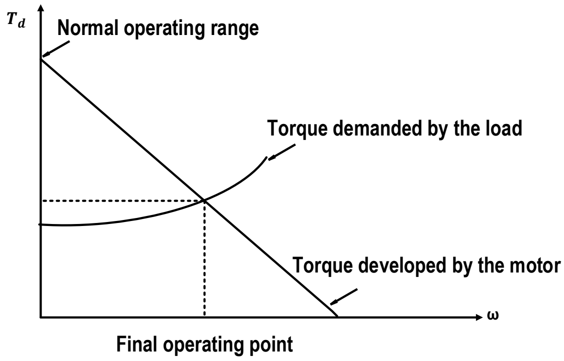


Figure 2.3: Interactions of the DC motor and mechanical load [27]

The graph in the Figure 2.3 shows the interaction of DC motor and mechanical load. The starting torque of the motor is higher than the load torque demanded by the load. The difference between these two torques forces the motor to rotate. As the motor starts to rotate and picks up speed, the developed torque decreases. The motor finally comes to a stable operating point when the two torques balance each other.

2.4 Model of a DC Motor

Use armature voltage equation as given in Equation (2.22)

$$V_a = I_a R_a + L_a \frac{dI_a}{dt} + E_b \tag{2.22}$$

Using Laplace transform of Equation (2.22)

$$\begin{aligned} V_a &= I_a(s)R_a + L_a S I_a(s) + E_b \\ &= (R_a + L_a S)I_a(s) + E_b \end{aligned} \tag{2.23}$$

$$V_a - E_b = (R_a + L_a S)I_a(s) \tag{2.24}$$

$$I_a = \frac{V_a - E_b}{R_a + L_a S} \tag{2.25}$$

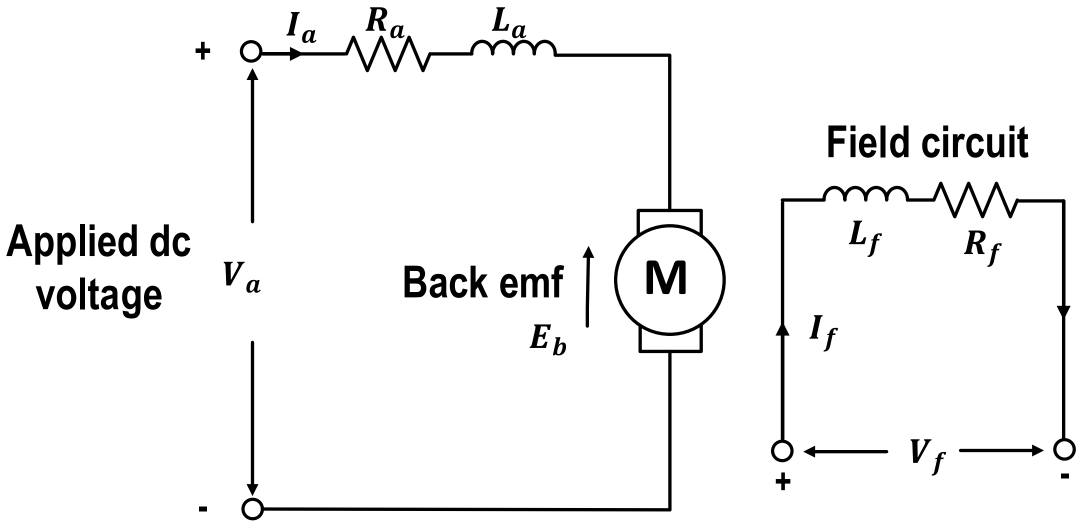


Figure 2.4: Model of DC motor [28]

where back electromotive force of the motor is given in Equation (2.26).

$$E_b = K\Phi\omega \tag{2.26}$$

$$I_a(s) = \frac{V_a - K\Phi\omega}{R_a(1 + \frac{L_a S}{R_a})} \tag{2.27}$$

The torque equation written as

$$T_d = J\frac{d\omega}{dt} + B_m\omega + T_L \tag{2.28}$$

Let neglect viscose friction coefficient, $B_m = 0$

$$\begin{aligned} T_d &= J\frac{d\omega}{dt} + T_L \\ &= K\Phi I_a \end{aligned} \tag{2.29}$$

where,

$$T_a = \frac{L_a}{R_a} \tag{2.30}$$

Using Laplace transform of Equation (2.29)

$$T_d = JS\omega(s) + T_L \tag{2.31}$$

$$\begin{aligned} \omega(s) &= \frac{T_d - T_L}{JS} \\ &= \frac{K\Phi I_a - T_L}{JS} \end{aligned} \tag{2.32}$$

Let load torque is neglected, $T_L = 0$

$$\omega(s) = \frac{K\Phi I_a}{JS} \tag{2.33}$$

and substituting Equation (2.27) in (2.33) results,

$$\begin{aligned}\omega(s) &= \frac{(K\Phi/R_a)[(V_a - K\Phi\omega)/(R_a + L_a S)]}{JS} \\ &= \frac{(K\Phi V_a - K^2\Phi^2\omega)/[R_a(\frac{R_a+L_aS}{R_a})]}{JS}\end{aligned} \tag{2.34}$$

$$JS\omega(s) = \frac{K\Phi V_a - K^2\Phi^2\omega}{R_a(\frac{R_a+L_aS}{R_a})} \tag{2.35}$$

$$JS\omega(s)R_a(\frac{R_a + L_aS}{R_a}) = K\Phi V_a - K^2\Phi^2\omega(s) \tag{2.36}$$

$$\begin{aligned}K\Phi V_a &= JS\omega(s)R_a(\frac{R_a + L_aS}{R_a}) + K^2\Phi^2\omega(s) \\ &= \omega(s)[JSR_a(\frac{R_a + L_aS}{R_a}) + K^2\Phi^2]\end{aligned} \tag{2.37}$$

$$\frac{\omega(s)}{V_a} = \frac{K\Phi}{JSR_a(\frac{R_a+L_aS}{R_a}) + K^2\Phi^2} \tag{2.38}$$

Take R_a as common factor for the denominator

$$\begin{aligned}\frac{\omega(s)}{V_a} &= \frac{K\Phi}{R_a[JS(R_a + L_aS)/R_a + \frac{1}{R_a}K^2\Phi^2]} \\ &= \frac{(K\Phi/R_a)}{JS(\frac{R_a+L_aS}{R_a}) + \frac{1}{R_a}K^2\Phi^2}\end{aligned} \tag{2.39}$$

Divide both numerator and denominator with $JS(\frac{R_a+L_aS}{R_a})$ and substitute Equation (2.30)

$$\frac{\omega(s)}{V_a} = \frac{(K\Phi/R_a)/[JS(\frac{R_a+L_aS}{R_a})]}{1 + [(K^2\Phi^2/R_a)/JS(\frac{R_a+L_aS}{R_a})]} \tag{2.40}$$

$$\frac{\omega(s)}{V_a} = \frac{(K\Phi/R_a)/[JS(1 + T_aS)]}{1 + [((K\Phi)^2/R_aJ)(1/S(1 + T_aS))]} \tag{2.41}$$

Since $T_m = R_aJ/(K\Phi)^2$

$$\frac{\omega(s)}{V_a} = \frac{(K\Phi/R_a)/[JS(1 + T_aS)]}{1 + [(1/T_m)(1/S(1 + T_aS))]}$$

Add the numerator

$$\frac{\omega(s)}{V_a} = \frac{(K\Phi/R_a)[1/JS(1+T_aS)]}{[T_mS(1+T_aS)+1]/[T_mS(1+T_aS)]}$$

Multiply the equation in the numerator by $\frac{(R_a/(K\Phi)^2)}{(R_a/(K\Phi)^2)}$

$$\frac{\omega(s)}{V_a} = \frac{(K\Phi/R_a)(R_a/(K\Phi)^2)[1/JS(1+T_aS)][1/(R_a/(K\Phi)^2)]}{[(1+T_mS(1+T_aS))/(T_mS(1+T_aS))]} \tag{2.42}$$

$$\frac{\omega(s)}{V_a} = \frac{(1/K\Phi)(1/[(JR_a/(K\Phi)^2)S(1+T_aS)])}{[(1+T_mS(1+T_aS))/(T_mS(1+T_aS))]} \tag{2.43}$$

$$\frac{\omega(s)}{V_a} = \frac{(1/K\Phi)[1/(T_mS(1+T_aS))]}{[(1+T_mS(1+T_aS))/(T_mS(1+T_aS))]} \tag{2.44}$$

$$\frac{\omega(s)}{V_a} = (\frac{1}{K\Phi})[\frac{1}{T_mS(1+T_aS)}][\frac{T_mS(1+T_aS)}{1+T_mS(1+T_aS)}] \tag{2.45}$$

$$\frac{\omega(s)}{V_a} = \frac{(1/K\Phi)}{ST_m(1+ST_a)+1} \tag{2.46}$$

Let load torque, $T_L = 0$. Apply full voltage, V_a by assuming armature inductance, $L_a = 0$. Substitute these values in the Equation (2.22) and (2.28). Assume $B_m = 0$.

$$V_a = I_aR_a + E_b \tag{2.47}$$

Since $E_b = K\Phi\omega(t)$

$$V_a = I_aR_a + K\Phi\omega(t) \tag{2.48}$$

and

$$\begin{aligned} T_d &= \frac{Jd\omega}{dt} \\ &= K\Phi I_a \end{aligned} \tag{2.49}$$

$$I_a = (\frac{Jd\omega}{dt})\frac{1}{K\Phi} \tag{2.50}$$

Substitute Equation (2.50) in (2.48)

$$V_a = K\Phi\omega(t) + R_a(\frac{Jd\omega}{dt}\frac{1}{K\Phi}) \tag{2.51}$$

Multiply both sides by $(\frac{1}{K\Phi})$

$$\frac{V_a}{K\Phi} = \omega(t) + \frac{JR_a}{(K\Phi)^2}(\frac{d\omega}{dt}) \tag{2.52}$$

Since $T_m = JR_a/(K\Phi)^2$. At no load, when the machine reaches full speed, i.e

$$\omega(noload) = \frac{V_a}{K\Phi} \tag{2.53}$$

Combining both Equation (2.52) and (2.53) gives

$$\begin{aligned} \omega(noload) &= \frac{V_a}{K\Phi} \\ &= \omega(t) + \frac{JR_a}{(K\Phi)^2}(\frac{d\omega}{dt}) \\ &= \omega(t) + T_m(\frac{d\omega}{dt}) \end{aligned} \tag{2.54}$$

Since $K\Phi = K_m$, then describe this in terms of T_m.

$$\begin{aligned} T_m &= \frac{JR_a}{(K\Phi)^2} \\ &= \frac{JR_a}{(K_m)^2} \end{aligned} \tag{2.55}$$

From Equation (2.55) J is obtained as

$$J = \frac{T_m(K_m)^2}{R_a} \tag{2.56}$$

Substitute Equation (2.56) and K_m value in (2.32)

$$\omega(s) = \frac{K\Phi I_a(s)}{JS} - \frac{T_L}{JS} \tag{2.57}$$

$$\omega(s) = \frac{K\Phi I_a(s)}{S(\frac{T_m(K_m)^2}{R_a})} - \frac{T_L}{S(\frac{T_m(K_m)^2}{R_a})} \tag{2.58}$$

$$\omega(s) = \frac{K_m}{(K_m)^2}\frac{R_a I_a(s)}{ST_m} - \frac{R_a T_L}{ST_m(K_m)^2} \tag{2.59}$$

$$\omega(s) = \frac{R_a}{(K_m)^2}\frac{I_a(s)}{JS} - \frac{T_L R_a}{(K_M)^2}\frac{1}{T_m S} \tag{2.60}$$

Take $\frac{1}{T_m S}$ as common factor

$$\omega(s) = [(\frac{R_a}{K_m})I_a(s) - \frac{T_L R_a}{(K_m)^2}][\frac{1}{T_m S}] \tag{2.61}$$

Further simplifying Equation (2.46) will results

$$\frac{\omega(s)}{V_a(s)} = \frac{(1/K\Phi)}{ST_m(1 + ST_a + 1)} \tag{2.62}$$

$$\frac{\omega(s)}{V_a(s)} = \frac{(1/K_m)}{1 + ST_m + S^2T_aT_m} \tag{2.63}$$

But, the armature time constant, T_a is much less than the electromechanical time constant, T_m. That is,

$$T_a << T_m \tag{2.64}$$

Now,

$$1 + ST_m + S^2T_aT_m \approx 1 + S(T_m + T_a) + S^2T_aT_m = (1 + ST_m)(1 + ST_a) \tag{2.65}$$

Note that this all process is to represent the transfer function in the time constant form. If we get the transfer function in time constant form know the largest time constant that will delay the system. So, if we use the PI controller, zero of the PI controller can be chosen so this large delay can be canceled. Note that delay means a lag and zero means a load, so this will try to compensate the whole system.

$$\frac{\omega(s)}{V_a(s)} = (\frac{1}{K_m})(\frac{1}{(1 + ST_m)(1 + ST_a)}) \tag{2.66}$$

These time constants of the system, T_m and T_a determines the response of the system.

2.5 DC Motor Speed Control

In many applications motor speed is required to vary over a wide range in order to accomplish certain task. The ease with this wide speed variation becomes one of the most attractive features of DC motors as compared with AC motors [29]. The back electromotive force for a separately excited DC motor is given in Equation (2.67).

$$\begin{aligned} E_b &= K\Phi\omega \\ &= V_a - I_aR_a \end{aligned} \tag{2.67}$$

After rearranging, speed of the motor is given by the Equation (2.68).

$$\omega = \frac{(V_a - I_aR_a)}{K\Phi} \tag{2.68}$$

From the equation given above it is observed that the speed can be varied by using any of the following methods [30].

- Armature voltage control (By varying V_a)
- Field Control (By Varying Φ)
- Armature resistance control (By varying R_a)

In armature voltage control the field is kept constant and it is applicable for speed values below rated speed whereas in field control the voltage is kept constant and is used for values above rated speed. In armature resistance control an external variable resistance is inserted in series with the armature circuit. In this paper speed is controlled using armature voltage control.

2.5.1 Armature Voltage Control

This speed control approach is usually applicable to separately excited DC motors with the armature resistance (R_a) and field flux (Φ) kept constant. Under normal operation of motor, since the drop across the armature resistance is small compared to E_b and therefore, the back emf is approximately equal to the terminal voltage ($V_T = V_a$). i.e. $E_b \approx V_a$. Since, $E_b = K\Phi\omega$ the angular speed can be gives in Equation (2.69).

$$\omega = \frac{V_a}{K\Phi} \tag{2.69}$$

From this equation it is observed that, if flux is kept constant, the speed changes linearly with armature voltage (V_a) and as the armature voltage is increased, the speed increases and vice verse. The relationship between speed and applied voltage is shown in the Figure 2.5. This method provides smooth variation of speed control.

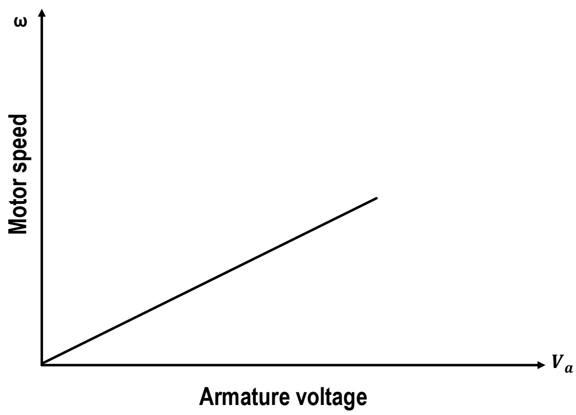


Figure 2.5: No-load speed of DC motor as a function of armature voltage [31]

2.5.2 DC Motor Starting

If the DC motor is connected directly across the supply, the starting current is dangerously high and it is given by the Equation (2.70).

$$I_a = \frac{(V_a - E_b)}{R_a} \tag{2.70}$$

At standstill, $E_b = 0$, therefore the starting current is also given in the Equation (2.71).

$$I_{astart} = \frac{V_a}{R_a} \tag{2.71}$$

Thus, the starting current can be reduced either by reducing armature voltage (V_a) during starting, or by increasing resistance in the armature circuit (R_a). Reducing V_a during starting requires a variable voltage supply whereas increasing resistance in the armature circuit requires an additional resistance R_{ext} to be connected in the armature circuit.

2.6 Four Quadrant Chopper

A chopper is a high speed on-off switch that converts fixed and unregulated DC input voltage into variable regulated, controllable DC output voltage and it is considered as a DC equivalent of an AC transformer as they behave in the same manner. Nowadays choppers are used all over the world for rapid transit systems such as in forklift trucks, mine haulers, trolley cars and marine hoist. Choppers provides smooth control, high efficiency, fast response and regeneration facility, and hence, the future electric automobiles are expected to use choppers for their speed control and braking purpose [32]. There are different power semiconductor devices used for a chopper circuit such as force commutated thyrister, power BJT, GTO, IGBT and MOSFET can be mentioned and these devices are represented by a switch.

As shown in the Figure 2.6, the chopper circuit can operate in all four quadrants of the Vo-Io plane. That is, the output voltage and current can be controlled both in magnitude and direction. Therefore, the power flow can be in any direction. In the first and third quadrants power flows from the source to the load and is assumed to be positive whereas in the second and fourth quadrants power is assumed negative and thus, power is fed back to the source. This can happen if the load is inductive or back emf source such as a dc motor.

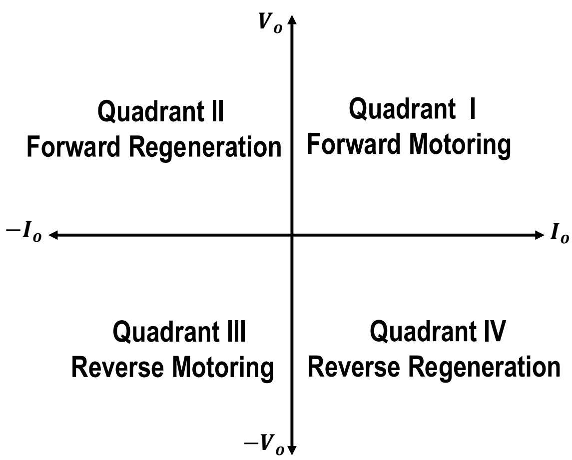


Figure 2.6: Four quadrant of V_o, I_o plane [33]

This four quadrant chopper is widely used in reversible dc motor drives since it requires power flow in either direction and hence, it is possible to implement regeneration and dynamic braking by means of which fast dynamic response is achieved.

The voltage signal can be step up or step down by varying the duty ratio of chopper. Basically the DC chopper compares the signal (voltage) coming from the controller with its own voltage and then makes an average output voltage signal to motor. Hence, the duty ratio of chopper is determined by the controller used. The average output voltage which is obtained by integrating chopped input voltage over the chopping period until the desired set point obtained, varies from 0 to 1.

2.6.1 Chopper Duty Cycle Control Methods

There are different control strategies of duty cycle in order to achieve an average value of output voltage V_o through opening and closing the semiconductor switch periodically. These are:

1. Time Ratio Control (TRC)
2. Current-Limit Control

Time Ratio Control (TRC)

In this method the duty ratio, on-time to chopping period is varied and this can be achieved in two different ways. These are:

- **Constant Frequency System:** In this technique the on-time is varied but chopping frequency f is kept constant and hence, it is called pulse width modulation scheme as it involves changing pulse change by varying the on time.
- **Variable Frequency System:** In this technique, the chopping frequency f is varied and either the on-time (T_{on}) or off-time (T_{off}) is kept constant and hence is called frequency modulation scheme.

Current-Limit Control

In this method, the previous set load current values decide when to turn on and turn off the chopper circuit. These set values are upper limit or maximum level of load current and lower limit or minimum of load current. The chopper, switched off when the load current reaches the maximum level and switched on when the load current falls below minimum level. Thus, by varying the maximum and minimum level of load current the switching frequency of chopper can be controlled.

2.6.2 Chopper Circuit Description

In four quadrant chopper circuit, four switching devices with anti-parallel connected diodes occurs and thus, these switching devices are termed as full bridge dc to dc converter. The input to the full bridge dc to dc converter is fixed magnitude dc voltage V_{dc} whereas the output can be a variable dc voltage with either polarity. Therefore, the circuit is termed as quadrant chopper or dc to dc converter as shown in the Figure 2.7. When a gating signal in a full bridge dc to dc converter is given to a switching device either the switching device or the diode only will conduct depending on the directions of the output load current.

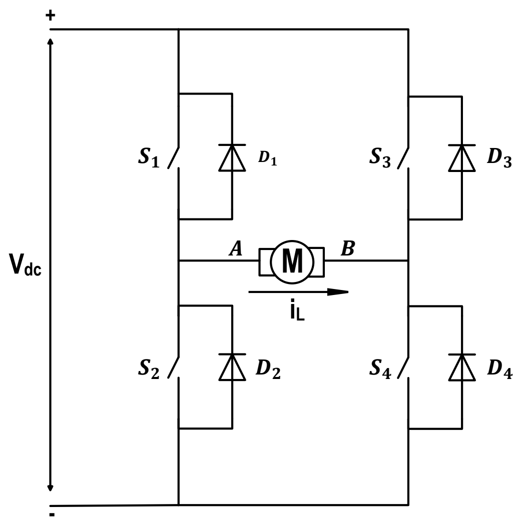


Figure 2.7: Four quadrant chopper circuit [34]

2.6.3 Switching Modes of Four Quadrant Chopper

There are two different modes in which the switches of four quadrant chopper can be switched. These are:

- **PWM with Bipolar Voltage Switching:** In this mode of switching the output voltage swings in both direction that is, from $+V_{dc}$ to $-V_{dc}$.
- **PWM with Unipolar Voltage Switching:** In this mode of switching the output voltage swings either from –zero to $+V_{dc}$ or zero to $-V_{dc}$.

2.6.4 Operation of the Four Quadrant Chopper with Bipolar Voltage Switching

When applying gating signals simultaneously to the switches S1 and S4 are turned ON, the load voltage V_{dc} with terminal 'A' positive and the load current I_L flows in the direction from A to B and because of the load inductance, the current cannot change instantaneously.

The conduction of the diode D3 will connect the load terminal B to the positive terminal of the source hence, the load voltage V_L will now become $-V_{dc}$. As the load voltage is negative and the current is still positive, the power is negative and it now flows from the load to the source. This operation of chopper circuit brings to the fourth quadrant and its operation will continues as long as the current is positive. When S1 and S4 are off, S3 and S2 can be turned ON.

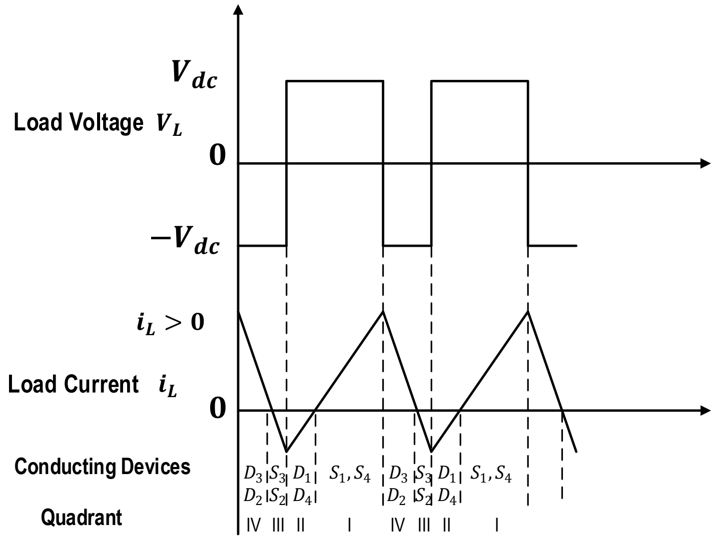


Figure 2.8: Load voltage and current with inductive load and $i_L > 0$ (positive) [35, p. 748]

When the current passes through zero, the switches S1 and S4 are turned off and the switches S3 and S2 can be turned on. The load current now becomes negative and passes through S3 and S2 with current direction in the load as from B to A. This operation of chopper circuit corresponds to the third quadrant. When switches S3 and S2 are tuned off, the diode D1 and D4 start to conduct and this bring the operation of the chopper circuit in the second quadrant.

In the first and third quadrant operation of chopper, power flows from the source to the load that is, power is positive and thus it is considered as forward power flow whereas reverse power flow in second and forth quadrant. The relevant wave forms showing the operation of full bridge converter in all the four quadrant is shown in the Figure 2.8 and Figure 2.9.

2.6.5 Generation of Gating Signals

In the four quadrant chopper, the gating signals are obtained by comparing a triangular carrier waveform with a positive and negative control voltage level. When a pulse is generated by comparing + V with the triangular carrier is used to turn on S1 and its complement is used to

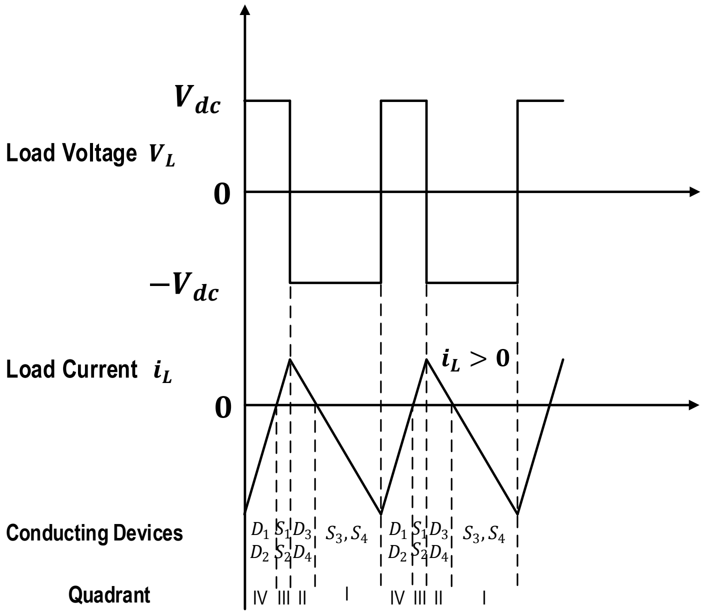


Figure 2.9: Load voltage and current with inductive load and $i_L < 0$ (negative) [35, p. 749]

turn on S2 whereas the pulse generated by comparing – V with triangular carrier is used to turn on S3 and its complement is used to turn S4 and the voltage varies from $-V_{tri}$ to $+V_{tri}$. In the figure the generator of gating signal for the four quadrant chopper with unipolar voltage switching is shown and a triangular carrier wave is generated. This generated triangular wave is compared with $+V_c$ and $-V_c$ in the comparator-1 and comparator-2 respectively.

The comparator-1 gives an output of gating signal to S1 and its compliment gives an output of gating signal to S2 whereas comparator-2 gives an output of gating signal to S3 and its compliment gives an output of gating signal to S4.

2.7 Fuzzy Theory

2.7.1 Definitions and Basics

When we compare both classical and fuzzy theory it is found that the classical theory concentrates much on precision rather than easy and efficient controlling mechanism. That is hundred percent either true or false (1 or 0). Unlike classical sets, the fuzzy sets have a certain degree of membership for each element. That is, fuzzy is all about the relative importance of precision.

If-Then Rules:

Fuzzy sets depend on certain rules called the rule base which is the most important requirement for the fuzzy logic. The rule base generally consists of various rules of the form If-Then, in which the fuzzy sets and the membership functions are declared first. Then the If-Then rules for the membership functions are decided for the particular control. The output is controlled by these rules on input. For each particular control, the If-Then rules for the membership functions are decided accordingly and hence, the output is controlled by these rules on input.

In a typical If-Then rule of fuzzy there consists of two parts named the antecedent and consequence or conclusion where the If statement is the antecedent and the Then statement is the consequence. That is, If - (Antecedent) and Then - (Consequence).

2.7.2 Sets and Operations

For a set in a universe of discourse if the elements belonging to the set must satisfy the rules specified by the set, then we call this as a classical set. It is denoted as given in Equation (2.72).

$$A = \Big\{ x \ \epsilon \ U \Big| x \ \ meets \ \ some \ \ conditions \Big\} \tag{2.72}$$

It can also be represented by Equation (2.73).

$$\mu A(x) = \{1 \ \ if \ \ x \ \epsilon \ A, 0 \ \ if \ \ x \ \ not \ \epsilon \ A\} \tag{2.73}$$

Unlike the classical set, in a fuzzy set each element has a certain degree of membership with which it belongs to the particular fuzzy set and it is given in the Equation (2.74).

$$A = \Big\{ (x, \mu A(x)) \Big| x \ \epsilon \ U \Big\} \tag{2.74}$$

There are three main fuzzy set operations. These are, Union, Intersection and Complement. Let us consider two fuzzy A and B sets such that $A, B \ \epsilon \ U$ where, U is the universe of discourse.

Union

It can be defined as the smallest set which contains all the elements in ‘A’ *or* ‘B’. It can be written as ‘A + B’ or ‘A ∪ B’. The membership function for Union is given in Equation (2.75).

$$\mu A \cup B(x) = \mu A(x) \cup \mu B(x) \tag{2.75}$$

Intersection

It can be defined as the biggest fuzzy set in both A and B which contains both the elements in A and B. It can also written as 'A ∩ B'. The membership function for Intersection is defined as given in Equation (2.76).

$$\mu A \cap B(x) = \mu A(x) \cap \mu B(x) \tag{2.76}$$

Complement

The Complement of a fuzzy set can defined as a fuzzy set with the membership function shown in Equation (2.77).

$$\mu(x) = 1 - \mu(x) \tag{2.77}$$

2.8 Fuzzy Logic Control

Fuzzy logic control is one of the control algorithm that mainly depends up on the rules formed by the linguistic variables which are being derived from expert knowledge into an automatic control strategy. Fuzzy logic does not need for hard mathematical modeling in order to provide exact the solution for the controlled system and hence, fuzzy uses only simple mathematical calculation to simulate the expert knowledge, unlike other conventional controllers. Even though it doesn't need any complex mathematical calculation, but it gives good performance in a control system. Thus, this control method is one of the best available solution for recent broad class of challenging control problems.

Fuzzy logic control is based on the fuzzy set theory such that each element has a degree of membership with which it belongs to any particular set. Fuzzy sets can be imagined as classical sets without much sharper boundaries. Mostly, it is taken as an attractive choice when precise mathematical formulations are not possible. That is, it is used when the precision required is moderate.

Fuzzy theory is most successfully applied to practical problem by using fuzzy control systems. If fuzzy systems are used in order to model the process and also if the controllers are designed based on the model, then we call the resulting controllers as fuzzy logic controllers (FLC). There are four main components in a fuzzy logic controller as shown in the Figure 2.10. These are,

1. Fuzzification,
2. Inference engine,
3. Fuzzy rule base,
4. Defuzzification

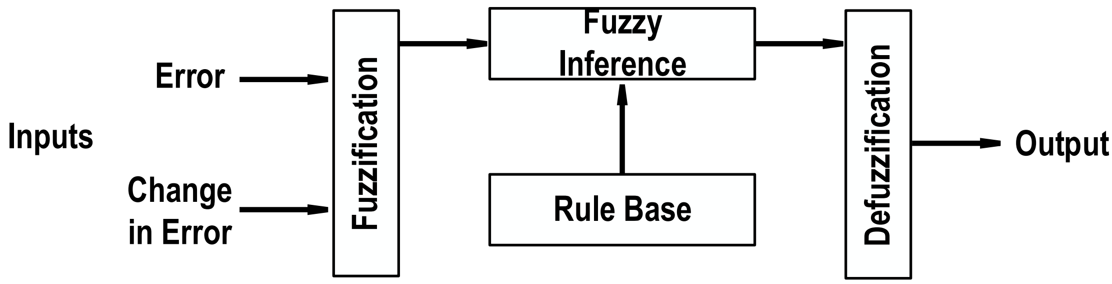


Figure 2.10: Structure of fuzzy logic controller [36]

2.8.1 Fuzzification

In designing of fuzzy controller the most important step to be considered is identifying the state variables that efficiently control the plant and followed by fuzzification of the inputs since fuzzy logic controller works with only the fuzzy inputs. Since, fuzzy rule base block employs rules on only linguistic variables, the measured numerical inputs called crisp inputs have to be converted into fuzzy linguistic variables by a block called fuzzifier. The fuzzification process is taken place in this fuzzifier block. Thus, fuzzification is the process of converting a numerical state variable (real number or crisp variable) into a fuzzy input linguistic variable (fuzzy number). System variables, which are generally used are comprises of state error, the rate of change of state error (state error derivative), the area of state error (state error integral) or etc. In this speed control system, speed error and rate of change of speed error are chosen as the input signals.

In fuzzification process the measured input data is converted into degree of membership functions. That is, the measured data is compared with conditions of rules and determined how well data is matched with rule at particular instance. Thus, here developing a degree of membership function is observed. This fuzzification process is achieved with the different types of fuzzifiers. These are generally three types of fuzzifiers. They are;

1. Singleton fuzzifier
2. Gaussian fuzzifier
3. Trapezoidal or Triangular fuzzifier

2.8.2 Fuzzy Rule Base

It is a decision making logic that is, simulating a human decision process, inters fuzzy control action from the knowledge of the control rules and linguistic variable definitions. A

Rule-base is the collection of rules and these rules are written in Rule-base block according to the control system requirement. The rules are usually in "If-then" form where the If side is called the conditions and the Then side is called the conclusion.

In this process fuzzy logic operations are more engaged and then results in control action to take place by the inference engine according to the fuzzy inputs on the current process. In this DC motor speed control system, the computer is able to execute the rules and compute a control signal based on the measured input speed error and change in speed error. In such a rule based controller the control strategy is stored in a more or less natural language (human language) and thus it is easy to understand and easy to maintain for non-specialist end user.

2.8.3 Inference Engine

Inference engine is defined as the software code which processes the rules, cases, objects or other type of knowledge and expertise based on the facts of a given situation. In inference engine aggregation is done in which degree of fulfillment is calculated according to the condition specified by a rule. When a problem that involves logic rather than fencing skills is going to be solved, we take a series of inference steps that may include deduction, association, recognition, and decision making. An inference engine can be considered as an information processing system (such as a computer program) that systematically employs inference steps similar to that of a human brain. There are two methods of fuzzy inference system. They are Mamdani and Sugeno and it will be discussed in detail later.

2.8.4 Defuzzification

While using Fuzzy Logic Controller (FLC) it produces an output required in a linguistic variable (fuzzy number) form and as per the real world requirements, the linguistic variables should to be transformed to a crisp output (real number). Hence, this is done in defuzzification block. Defuzzification block converted resulting fuzzy set into a number that is sent to the system and this number is actually the control signal.

Defuzzification is the process of delivering a quantifiable result in the fuzzy form. It is the methodology of converting a fuzzy quantity into a precise quantity. That is why it is said that defuzzification is the reverse of fuzzification. That is, defuzzification is the transformation of a fuzzy linguistic variable into a precise numerical state variable (real number or crisp variable), whereas fuzzification is the conversion of a precise numerical state variable (real number) to a fuzzy linguistic variable.

Generally, we can say a defuzzification process is when all the actions that have been activated are combined and converted into a single non-fuzzy output signal which is the control signal of the system. There are many defuzzification methods but the most common methods are as follows:

1. Centre of Gravity Method (COG),
2. Bisector of Area Method (BOA),
3. Mean of Maximum Method (MOM).

In center of gravity (COG) method the crisp control value is the abscissa of the center of gravity of the fuzzy set. That is, it obtains the center of area occupied by the fuzzy set. This method is also best known defuzzification method and utilized as part of this thesis work.

In bisection of area (BOA) defuzzification method, the abscissa of the vertical line is calculated in such a way that divides the area of the resulting membership function into two equal areas.

In mean of maximum (MOM) method the crisp value is chosen as the point with the highest membership and since there may be several point with maximum membership value in the overall implied fuzzy set. Thus, it is common method to calculate the mean value of these maximum points called mean of maximum.

A fuzzy control system has a number of rules that change various variables into a "fuzzy" form, that is, the outcome is shown as membership functions and their degree of membership in fuzzy sets. When dealing with fuzzy set selecting the highest membership belonging may be easiest but not much useful since it results in loss of some data in the process.

2.8.5 Membership Function

In the design of fuzzy logic control system the most important things are the process design of membership function for inputs and outputs, and design of fuzzy if-then rule knowledge base. The membership function is the graphical representation of the degree of belonging (magnitude of participation) of an element to the fuzzy set and illustrates how completely a crisp variable belongs to a fuzzy set (degree of membership, μ). The degree of membership is the degree to which a crisp variable belongs to a fuzzy set. It is expressed either as a fractional value ranging from 0.0 to 1.0 or percentage ranging from 0% to 100%.

The membership functions are determined by using the knowledge of human experts and data collected from various sensors. Designers choose many different shapes based on their preference and experience in order to define fuzzy membership function. There can be different memberships functions associated with each input and output response. There are generally four types of membership functions used as shown in Figure 2.11.

1. Trapezoidal membership function
2. Triangular membership function
3. Gaussian membership function

4. Generalized bell membership function

Among them the most popular shapes are triangular and trapezoidal membership functions because these shapes are easy to represent designer's idea and require low computation time. The quality of control which can be achieved using fuzzy controller is determined by the number of membership function .With increase in the number of membership function, the quality (accuracy) of control improves. When the number of linguistic variables increases, the complexity, computational time and required memory also increases. Therefore, a compromise has to be considered between the quality of control and computational time to choose the number of linguistic variables.

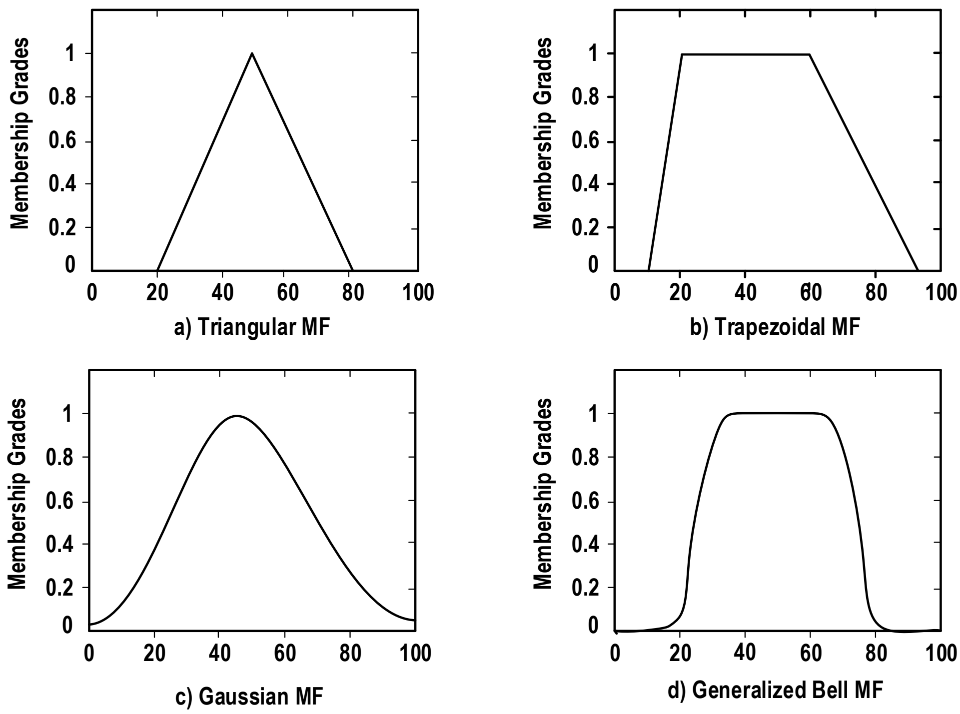


Figure 2.11: Shapes of different membership functions [37]

2.8.6 Different Fuzzy Control Methods

We can design fuzzy interface systems using two different methods. These are:

- Mamdani fuzzy inference
- Sugeno fuzzy inference

Their outputs vary somewhat in these two methods.

Mamdani Fuzzy Inference

This method of fuzzy inference is currently the most commonly used one. It was among the first control systems built using fuzzy set theory. In Mamdani inference method the output variable is expected to be fuzzy sets. Using a number of fuzzy sets can be tedious in some cases hence it is advantageous to use single membership function of a linguistic variable. This method of using a single linguistic variable in output is known as Singleton output mechanism and it greatly simplifies the computation required by the more general Mamdani method which finds the centroid of the two dimensional function. Thus, Singleton output mechanism enhances the defuzzification process. But in the case of Sugeno type of inference can be used to model any inference system where the output function is either linear or constant.

Sugeno Fuzzy Inference

Sugeno fuzzy inference is similar to the Mamdani fuzzy inference in many aspects. The first two parts of the fuzzy inference process, fuzzifying the inputs and applying the fuzzy operator, are exactly the same. The main difference between Mamdani and Sugeno is that, the Sugeno output membership functions are either linear or constant whereas it is fuzzy in Mamdani.

If we assume the first input is x and the second input is y, then the output (O_i) is of the linear form $O = Ax + By + C$ and for a zero-order Sugeno model, the output will be a constant $(A = B = 0)$. The output level of each rule (O_i) is only weighted by the weightage (W_i) of the rule. That is, the final output of the system is the weighted average of all rule outputs.

2.8.7 Comparison of Mamdani and Sugeno Methods

Both inference systems are the same in many aspects and their main difference arises from the output membership functions which is either linear or constant for Sugeno type and fuzzy for Mamdani one. Sugeno system works well with optimization and adaptive techniques and hence, they are more compact and computationally efficient than a Mamdani system. When comparing the performance of both Mamdani and Sugeno models the highest rank is obtained from Sugeno fuzzy inference mode. The following are some final considerations about the two different methods.

Advantages of the Mamdani Method

- It is instinctual. It can be trained in human intuition.
- It has widespread acceptance.

- It is well suited to human input.

Advantages of the Sugeno Method

- It is very effective in calculations and controlling.
- It is generally used to enhance the linear techniques.
- It works well with optimization and adaptive techniques.
- It is well suited to mathematical analysis.
- It has guaranteed continuity of the output surface.

In this thesis, Mamdani fuzzy inference type is selected because of its wide acceptance in capturing expert knowledge in more effective way although it sometime slows due to computational burden in defuzzification process. In this speed control case, since mathematical inputs are not available, Sugeno method cannot be used.

2.8.8 Advantages of Fuzzy Logic Control

- Fuzzy controller does not require mathematical model of a system and calculation is easy, hence very easily designed.
- It can operates very well even in highly non-linear systems that would be difficult or impossible to model mathematically.
- It is inherently robust since it does not require precise, noise-free inputs and if sensors fail to work properly it can be programmed to turn off safely, instead of keep on working which could be dangerous to the plant.
- Cheaper because they are easier to design and increased robustness than other non-linear controllers.
- It can be worked with multiple input multiple output systems, although defining the rule base quickly for MIMO systems becomes complex. Creating the rule base for fuzzy control is easier than optimizing the parameters for PID control.
- Very simple user interface. Easier end-user interpretation when the final user is not an expert.
- Fuzzy logic is flexible. Since the rule base is user defined it can be modified and hence new sensors can easily be incorporated into the system.
- Fuzzy logic is conceptually easy to understand since human knowledge and experience can be implemented to relate input to output in linguistic term.

- Fuzzy logic is a more intuitive approach like a human which gives it the strength to adapt to the difficulties in the control.
- Can achieve less overshoot and oscillation and doesn't require fast processors and it can also control fast processes.
- It requires less data storage in the form of membership functions and rules than conventional look up table for non-linear controllers.

2.8.9 Disadvantages of Fuzzy Logic Control

- Human knowledge is often incomplete and episodic as compared to systematic way.
- If the model is not known then it is impossible to achieve the stability of the controller system.
- Sometime rules are mismatched and non-coherent.
- In complex operation fuzzification and defuzzification take long time.

2.8.10 Implementation of Fuzzy Logic Control

It maps the input to an output in a very efficient way and the mapping can be very easily controlled without much complex knowledge about the process. Steps to be performed in fuzzy control:

- First, study how the rule base system and the operator can be applied to this control.
- Understand the concept of membership functions and linguistic variables.
- Analyse the power system to be controlled and decide the state variables to be considered as inputs to the system.
- Form an understanding as how to control the state variables to get the required control in the plant.
- Now form the Rule base for the linguistic variables of inputs and outputs.
- Try to optimize the membership functions to make the control more efficient.
- Integrate the fuzzy controller into the plant and check the result.

2.9 PI Controller

A proportional-integral (PI) controller is a special case of the PID controller in which the derivative (D) term of the error is not used and it is widely used in industrial control systems. PI controller works by calculating error (the speed error in our case), from the processed measured value and the desired reference point. If the system is not clearly known, best results are achieved applying PI controller provided it is tuned properly by keeping parameters of the system according to the nature of the system [38].

The PI controller depends upon two parameters which are called the proportional (P) and the integral (I) term:

- P determines the reaction to current error. That is, P term is proportional to current value of the error.
- I determines reaction to the sum of recently appeared errors. That is, it accounts for past values of the error and integrates overtime to produce the integral term.

The relation between the control signal u and the error is stated in the Equation (2.80). (where u is the control signal to the process, e is the error, τ_i is the integral time, K_p and K_i are the proportional, and integral gains, s is the laplace variable).

$$u(t) = K_p e(t) + K_i \int_0^t e(t)dt \qquad (2.78)$$

$$u(s) = K_p e(s) + \frac{K_i e(s)}{s} \qquad (2.79)$$

since $K_i = K_p / \tau_i$

$$TransferFunction = \frac{u(s)}{e(s)} = K_p(1 + \frac{1}{\tau_i s}) \qquad (2.80)$$

In controlling a speed of motor drives a simple proportional gain in the speed loop may not be sufficient enough since it may results in a high overshoot and also an undesirable steady state error in the speed. Hence, another compensation technique is required to improve the performance of the drive. That is, PI (proportional plus integral) control which is most commonly used compensation method for DC motor drives.

A proportional controller will reduce the rise time and will reduce, but never eliminate, the steady-state error. The integral term contribution is proportional to both the magnitude of the error and the duration of the error. Thus, it accelerates the movement of the process towards the set point and eliminates the residual steady-state error that occurs with a pure proportional controller [39].

The PI controller can be summarized as follows:

1. The integrative action will eliminate also the disturbance effects to the process output.

2. More complex in control design than P-controller and require complex tuning (selecting of the K_p and K_i gain).

3. Complexity in algorithm of implementation when using microcontrollers (embedded system).

4. Increasing the K_i coefficient makes the system more responsive and hence increasing performance. Though increasing the K_i to high values may cause the system to oscillate and become unstable.

5. The integrative action has a negative effect on the speed of the response.

2.9.1 Controller Basics

In control system of closed loop, control is required to modify the behavior of a system so that it behaves in a specific desirable manner over time. Thus, the controller plays an important role in providing a very easy and common technique of keeping motor speed at any desirable set-point speed under variable load conditions. Not only this but it can also be used to maintain the speed at the desired set-point value when set-point is ramping up or down at a specified rate. The essential addition for this condition to the previous system is a reason for the present speed to be measured.

In this closed loop speed controller, a tacho-generator is attached to motor rotor and generates a voltage signal which is proportional to the motor speed. This voltage signal is fed back to the input where signal is subtracted from the set-point speed to produce a speed error signal. This speed error signal is then fed to the controller in order to find out what the magnitude of the controller output will be to make the motor run at the desired set-point speed (i.e. reference speed). For example, if the error speed is negative, this means the motor is running fast so that the controller output should be decreased and in another way if the error speed is positive, this means the motor is running slow so that the controller output should be increased.

2.9.2 Controller Type Selection

To design a controller that makes a system behave in a desirable manner, we need a way to predict the behavior of the quantities of interest over time, specifically how they change in response to different inputs. Hence, the control action can be considered as simple like if the speed error is found negative (i.e. the motor is running above set-point), then multiply it by some scale factor which is known as gain and set the output drive to the desired level. But this method is only partially satisfied because of the following reason: that is, if the motor runs at the set-point under no load condition there will be no speed error obtained (speed error is zero). So the motor runs freely. If a load is applied, the motor speed slows down and a positive speed error is obtained. Then the output is increased by a proportional amount in order to try and restore the desired speed as compared to the no load condition. However, while the motor speed is recovering, the error reduces drastically and so does the drive level. Then it results the motor speed to stabilize at a speed below the set-point speed at which the load is balanced by the product of error speed and the gain. This approach discussed is what we call proportional control and it is works with the current value of the speed error. But this proportional controller has a drawback as it can never force the motor to run exactly at the set point speed.

From the above discussion it is observed as an improvement is required. Thus, correction is made by adding or subtracting a small amount to the output. This will continues until the motor reaches the set-point. This can be done by adding the instantaneous speed error observed for instant at regular interval of time and multiplying this by another gain (integral gain) before adding the result to the proportional correction made earlier. This approach focuses on the integration of the speed error.

Up to now we have seen two mechanisms constituting a PI (proportional-integral) controller which works simultaneously trying to correct the motor speed. The proportional term works on fast-acting correction which will produce a change in the output as quickly as the error arises. The integral action takes a finite time to act but has the capability to eliminate the residual steady-state error that occurs with a pure proportional controller.

When an additional correction to the output drive is required a further refinement which uses the rate of change of speed error known as derivative control is used. Since the derivative term is a best estimate of the future trend of speed error based on its current rate of change, it can be used to give a very fast response to sudden changes in motor speed. It is difficult to generate a derivative term in the output of simple PID controllers that has any significant effect on motor speed. Even though the derivative control reduces the rapid speed oscillation caused by high proportional gain, it is not used in many controllers due to its sensitivity to measurement noise (random error). This noise affects the main signal to be amplified and reflected in the controller output. Hence the most suitable controller for speed control as compared to conventional controllers is PI type controller.

2.9.3 Importance of Current Controller in a DC Drives System

The importance of current controller in a DC drive system is to control the rated current limit in motor armature when the motor is made to run from zero speed to high speed. That is, when the motor is allowed to run from zero speed to a high speed, the motor will delay some time to speed up due to electromechanical time constant. But still the speed controller used for controlling speed acts very fast. Initially, the speed feedback is zero which will result in full controller output voltage E_c (reference voltage) and hence the converter gives maximum voltage output. Since back Emf is zero at this starting time, a very large current flow which sometime exceeds the motor maximum current limit and this can damage the motor winding. Therefore, to solve the problem stated above we need to control current in motor armature using current controller which will take care of motor rated current limit. Note that, the applied voltage V_a will now not dependent on the speed error only but also on the current error. Hence, we should ensure that V_a is applied in such a way that does not draw more than the rated current during positive and negative torque operation of the motor. So, current controller is required hence an inner current loop is used.

2.9.4 Advantages of PI Controller

- Due to presence of the integral term, steady state error in speed is zero, making the system quite accurate.
- It is not necessary to use such high gains as required in proportional gain controller.
- The scheme has found to be quite robust and satisfactory.

2.9.5 Disadvantages of PI Controller

- If a very fast response is desired, the penalty paid is a higher overshoot in speed, which is not desirable.
- This can be designed without any overshoot but the response to a load change becomes very slow.
- A very high gain cannot be used, because that will resulting higher overshoots.

CHAPTER THREE

METHODOLOGY

3.1 Chapter Overview

In this chapter the methods and alternatives that have been used from the beginning until the end of this thesis work are stated. It includes mathematical modeling of DC motor, Fuzzy logic controller design, PI controller design, block models of four quadrant DC chopper design and Simulink model for speed control of separately excited DC motor using four quadrant DC chopper.

3.2 Procedures of the Study

To attain the objective of the study, the following procedures are undertaken:

- **Literature Survey:** Different literatures which are related to this thesis work have studied and different concepts are adopted.
- **System Modeling and Design:** modeling of the system and designing an appropriate fuzzy logic and PI controller for the system.
- **Code Writing:** In order to simulate the overall control system on fuzzy logic controller, algorithms are developed and programs are written.
- **Simulation:** the modeled and designed systems will be tested and simulated using Matlab/Simulink tools.
- **Validation:** Validation of results obtained from simulation of both Fuzzy and PI controllers with that of expected one.

3.2.1 Research Methodology

In this section, the work flow for the whole thesis work is described graphically using flow chart as shown in the Figure 3.1.

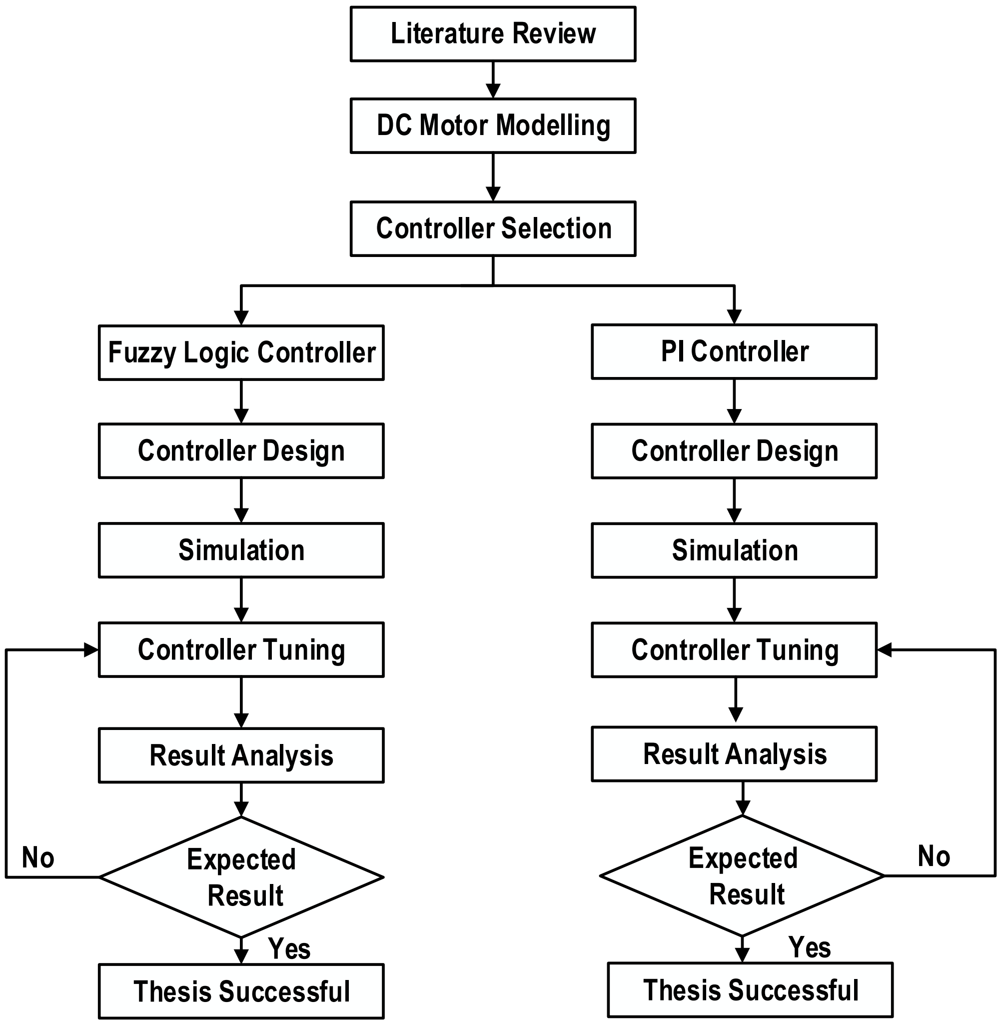


Figure 3.1: Flow chart of the work flow for the whole thesis

3.3 PI Controller Design

3.3.1 Design of the PI Current Controller

When motor drive starts to run the back emf is zero and a large current flows through the machine. Thus, we need to design current controller for this extreme condition as shown in Figure 3.2.

From the model given in the Figure 3.2 the transfer function becomes,

$$\frac{I_a(s)(f)}{I_a(s)(ref.)} = \frac{K_c[(1+T_cS)/(T_cS)](K_t)[(1/R_a)/(1+T_aS)]}{1+K_c[(1+T_cS)/(T_cS)](K_t)[(1/R_a)/(1+T_aS)][K_2/(1+T_2S]} \tag{3.1}$$

Now, T_c (current controller parameter) is in our control so it can be varied as when

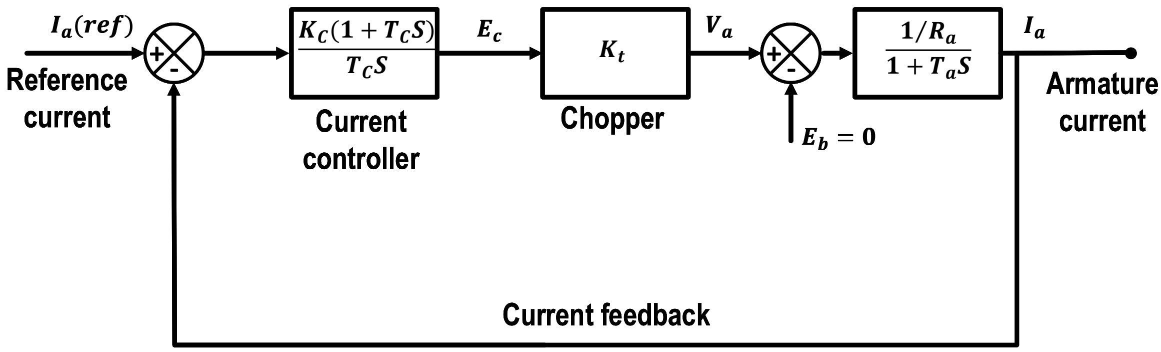


Figure 3.2: Block model for PI current controller design

required. Note that, we should choose T_c such that it cancel the largest time constant in the transfer function in order to reduce order of the system. Hence, the system response will be much faster. So, let us make

$$T_c = T_a \tag{3.2}$$

Putting Equation (3.2) in Equation (3.1)

$$\frac{I_a(s)(f)}{I_a(s)(ref.)} = \frac{K_c[(1+T_aS)/(T_aS)](K_t)[(1/R_a)/(1+T_aS)]}{1+K_c[(1+T_aS)/(T_aS)](K_t)[(1/R_a)/(1+T_aS)][K_2/(1+T_2S]} \tag{3.3}$$

$$\frac{I_a(s)(f)}{I_a(s)(ref.)} = \frac{[(K_cK_t)/(R_a)][(1+T_aS)/(T_aS)][1/(1+T_aS)](T_aS)(1+T_aS)(1+T_2S)}{(T_aS)(1+T_aS)(1+T_2S)+[(K_cK_tK_2)/(R_a)](1+T_aS)} \tag{3.4}$$

$$\frac{I_a(s)(f)}{I_a(s)(ref.)} = \frac{[(K_cK_t)/(R_a)](1+T_2S)(1+T_aS)}{(T_aS)(1+T_aS)(1+T_2S)+[(K_cK_tK_2)/(R_a)](1+T_aS)} \tag{3.5}$$

Take T_a as common factor for the denominator and divide both numerator and denominator by $(1+T_aS)$.

$$\frac{I_a(s)(f)}{I_a(s)(ref)} = \frac{[1/(1+T_aS)][(K_cK_t)/(R_a)](1+T_2S)(1+T_aS)}{[1/(1+T_aS)](T_a)[S(1+T_aS)(1+T_2S)+[(K_cK_tK_2)/(R_a)](1+T_aS)/(T_a)]} \tag{3.6}$$

$$\frac{I_a(s)(f)}{I_a(s)(ref)} = \frac{[(K_cK_t)/(T_aR_a)](1+T_2S)}{S(1+T_2S)+[(K_cK_tK_2)/(T_aR_a)]} \tag{3.7}$$

Let as assume

$$K_o = \frac{K_cK_t}{T_aR_a} \tag{3.8}$$

$$\frac{I_a(s)(f)}{I_a(s)(ref.)} = \frac{K_o(1+T_2S)}{S^2T_2+S+K_oK_2} \tag{3.9}$$

Dividing T_2 on the right hand side where T_2 is filtering lag.

$$\frac{I_a(s)(f)}{I_a(s)(ref.)} = \frac{(K_o/T_2)(1+T_2S)}{S^2+(S/T_2)+(K_oK_2)/(T_2)} \tag{3.10}$$

The characteristic equation is given in Equation (3.11).

$$S^2 + \frac{S}{T_2} + \frac{K_o K_2}{T_2} \approx S^2 + 2\zeta\omega + \omega^2 \tag{3.11}$$

From Equation (3.11)

$$\omega^2 = \frac{K_o K_2}{T_2} \tag{3.12}$$

$$\omega = \sqrt{\frac{K_o K_2}{T_2}} \tag{3.13}$$

and

$$2\zeta\omega = \frac{1}{T_2} \tag{3.14}$$

$$\begin{aligned} \zeta &= \frac{1}{2T_2\omega} \\ &= \frac{1}{2T_2\sqrt{(K_o K_2)/(T_2)}} \\ &= \frac{1}{2\sqrt{T_2 K_o K_2}} \end{aligned} \tag{3.15}$$

The characteristics equation is a second order, to get a proper response ζ should be 0.707 (i.e. $\zeta = \frac{1}{\sqrt{2}}$). So,

$$\zeta = 0.707 = \frac{1}{\sqrt{2}} = \frac{1}{2\sqrt{T_2 K_o K_2}} \tag{3.16}$$

After squaring both sides and crisscrossing Equation (3.16), we get

$$\begin{aligned} K_o &= \frac{1}{2K_2T_2} \\ &= \frac{K_c K_t}{R_a T_a} \end{aligned} \tag{3.17}$$

$$K_c = \frac{T_a R_a}{2K_2 K_t T_2} \tag{3.18}$$

From the Equation (3.17)

$$K_o = \frac{1}{2K_2T_2} \tag{3.19}$$

Now, substitute K_o in Equation (3.10) results

$$\frac{I_a(s)(f)}{I_a(s)(ref.)} = \frac{(1/K_2)(1+T_2S)}{2S^2T_2^2 + 2ST_2 + 1} \tag{3.20}$$

We observe that the zero in the Equation (3.20) may result in an overshoot. Therefore, we will use a time lag filter between the $I_a(ref.)$ and controller to cancel the effect of the current filter. Since the current loop time constant is much higher than filter time constant, a slight delay at $I_a(ref.)$ will not affect much. So we place a filter here. Hence,

$$\frac{I_a(s)(f)}{I_a(s)(ref.)(1/(1+T_2S))} = \frac{(1/K_2)(1+T_2S)}{2S^2T_2^2+2ST_2+1} \tag{3.21}$$

$$\frac{I_a(s)(f)}{I_a(s)(ref.)} = (\frac{1}{K_2})\frac{1}{2S^2T_2^2+2ST_2+1} \tag{3.22}$$

3.3.2 Design of the PI Speed Controller

The speed controller has designed for PI controller as shown in Figure 3.3.

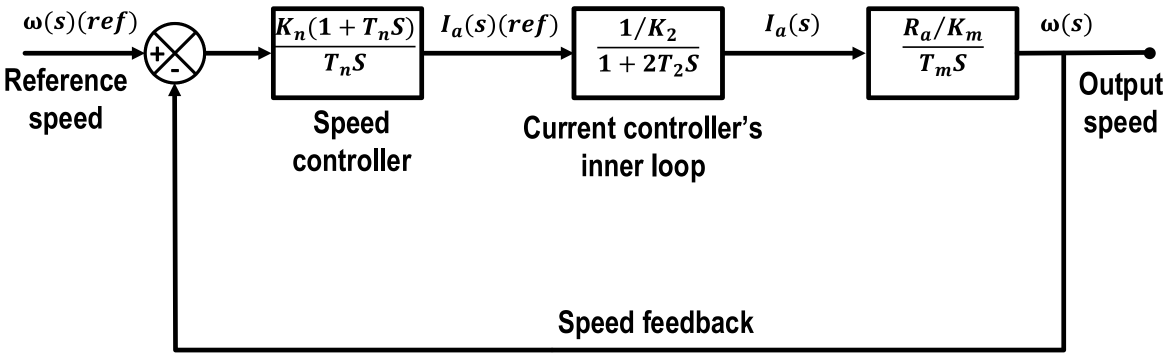


Figure 3.3: Block model for PI speed controller design

From the model given in the Figure 3.3 transfer function becomes,

$$\frac{\omega(s)}{\omega(s)(ref)} = \frac{(K_n/K_2)(R_a)/(K_mT_mT_n)(1+T_nS)/(1+2T_2S)(1/S^2)}{1+(K_nR_a)/(K_mK_2T_mT_n)(1+T_nS)/(1+2T_2s)(1/S^2)(K_1)/(1+T_1S)} \tag{3.23}$$

Here, we have the choice to T_n such that it cancels the largest time constant of the transfer function. So, let us make

$$T_n = 2T_2 \tag{3.24}$$

Putting this value in Equation (3.23) and we can write it as:

$$\frac{\omega(s)}{\omega(s)(ref)} = \frac{(K_nR_a)/(K_mK_2T_mT_n)(1+T_nS)/(1+T_ns)(1/S^2)}{1+(K_nR_a)/(K_mK_2T_mT_n)(1+T_nS)/(1+T_ns)(1/S^2)(K_1)/(1+T_1S)} \tag{3.25}$$

$$\frac{\omega(s)}{\omega(s)(ref)} = \frac{(K_nR_a)/(K_mK_2T_mT_n)}{S^2[1+(K_nR_a)/(K_mK_2T_mT_n)(1/S^2)(K_1/(1+T_1S)]} \tag{3.26}$$

$$\frac{\omega(s)}{\omega(s)(ref)} = \frac{(K_nR_a)/(K_mK_2T_mT_n)(T_mT_n)(1+T_1S))}{[S^2(K_mK_2T_mT_n)(1+T_1S]+(K_nR_aK_1))/(K_2K_m)} \tag{3.27}$$

$$\frac{\omega(s)}{\omega(s)(ref)} = \frac{(K_nR_a)/(K_2K_m)(1+T_1S)}{S^2T_mT_n(1+T_1S)+(K_nR_a)/(K_2K_m)(K_1)} \tag{3.28}$$

Ideally, $\omega(s) = \frac{1}{S(S^2+\alpha S+\beta)}$ In the above transfer function due to the absence of S term, the damping constant becomes zero and hence it results in oscillatory and unstable system. Thus, optimization is required. In order to optimize this we must get transfer function whose gain is close to unity.

3.3.3 Optimization of Speed Controller Transfer Function

Optimization technique aims at bringing output close to input variable so as to obtain unity gain for wide frequency range (i.e. modulus hugging).

$$\left|G(j\omega)\right| \approx 1 \tag{3.29}$$

For achieving unity gain the output should track the input variable instantaneously for any frequency variation with in bandwidth of the input variable.

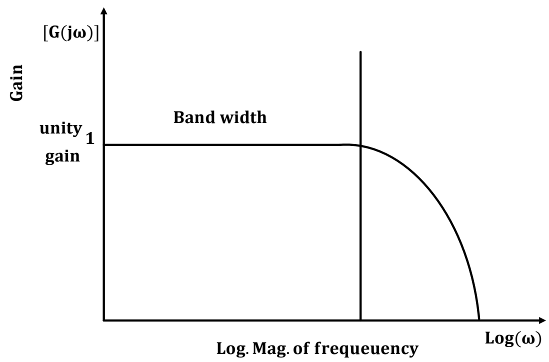


Figure 3.4: Gain versus frequency waveform

3.3.4 General Consideration for Optimization

Multiply both numerator and denominator of Equation (3.23) by $(1 + T_n S)$.

$$\frac{\omega(s)}{\omega(s)(ref)} = \frac{(K_n R_a)(1 + T_1 S)(1 + T_n S)}{S^2 T_m T_n K_2 K_m (1 + 2T_2 S)(1 + T_1 S) + (K_n R_a K_1)(1 + T_n S)} \tag{3.30}$$

From the denominator of Equation (3.30),

$$(1 + 2T_2 S)(1 + T_1 S) = 1 + S(2T_2 + T_1) + 2T_2 T_1 S^2 \approx 1 + S(2T_2 + T_1) \tag{3.31}$$

Since, T_1 and T_2 are smaller time constants, their product can be approximated to zero (i.e. $T_1 T_2 \approx 0$). Assuming, $\delta = 2T_2 + T_1$ and $K_o = \frac{K_n R_a}{K_2 K_m}$. So,$1 + S(2T_2 + T_1) = 1 + \delta S$.

Then,

$$\frac{\omega(s)}{\omega(s)(ref)} = \frac{(K_n R_a)/(K_2 K_m)(1 + T_n S)(1 + T_1 S)}{S^3 T_m T_n \delta + S^2 T_m T_n + (K_o K_1 T_n) S + K_o K_1} \tag{3.32}$$

The transfer function in the Equation (3.32) is of third order and the terms $(1 + T_nS)$ and $(1 + T_1S)$ in the denominator will be canceled by using filters.

3.3.5 For a standard third order system

$$G(j\omega) = \frac{b_o + j\omega b_1}{[a_0 + j\omega a_1 + (j\omega)^2 a_2 + (j\omega)^3 a_3]} \tag{3.33}$$

For low frequency $b_0 = a_0$ and $b_1 = a_1$. The above can be rewrite as:

$$G(j\omega) = \frac{a_o + j\omega a_1}{[a_0 + j\omega a_1 + (-1)(\omega)^2 a_2 + (-1)j(\omega)^3 a_3]} \tag{3.34}$$

$$G(j\omega) = \frac{a_o + j\omega a_1}{(a_0 - \omega^2 a_2) + j(\omega a_1 - \omega^3 a_3)} \tag{3.35}$$

The modulus of $G(j\omega)$ becomes,

$$\Big|G(j\omega)\Big| = \sqrt{\frac{a_o^2 + \omega^2 a_1^2}{a_o^2 + \omega^2(a_1^2 - 2a_oa_2) + \omega^4(a_2^2 - 2a_1a_3) + \omega^6 a_3^2}} \tag{3.36}$$

Now, to satisfy modulus hugging principle, (i.e. $\Big|G(j\omega)\Big| = 1$, coefficients of ω^2 and ω^4 are made equal to zero. So,

$$a_1^2 = 2a_oa_2 \tag{3.37}$$

and

$$a_2^2 = 2a_1a_3 \tag{3.38}$$

We need to use filters on $\omega(s)(ref)$ side to cancel $(1 + T_nS)(1 + T_1S)$ term:

$$\frac{\omega(s)}{\omega(s)(ref)(\frac{1}{1+T_nS})(\frac{1}{1+T_1S})} = \frac{(K_nR_a)/(K_2K_m)(1 + T_nS)(1 + T_1S)}{S^3T_mT_n\delta + S^2T_mT_n + (K_oK_1T_n)S + K_oK_1} \tag{3.39}$$

From optimization condition in Equation (3.38), we get

$$(K_oK_1T_n)^2 = 2 \times K_oK_1T_n \times T_mT_n \tag{3.40}$$

$$K_oK_1T_n = 2T_m \tag{3.41}$$

$$T_m = \frac{K_oK_1T_n}{2} \tag{3.42}$$

$$(T_mT_n)^2 = 2 \times K_oK_1T_n \times T_mT_n\delta \tag{3.43}$$

$$T_m = 2\delta K_oK_1 \tag{3.44}$$

And substitute Equation (3.42) in Equation (3.44),

$$\begin{aligned} T_m &= \frac{K_o K_1 T_n}{2} \\ &= 2\delta K_o K_1 \end{aligned} \tag{3.45}$$

$$\begin{aligned} T_n &= 4\delta \\ &= 4(2T_2 + T_1) \\ &= 8T_2 + 4T_1 \end{aligned} \tag{3.46}$$

From Equation (3.42) and (3.46):

$$\begin{aligned} T_m &= 2K_o K_1 \delta \\ &= 2(\frac{K_n R_a}{K_2 K_m}) K_1 \delta \end{aligned} \tag{3.47}$$

$$K_n = \frac{T_m K_m K_2}{2K_1 R_a \delta} \tag{3.48}$$

$$\begin{aligned} K_o &= \frac{K_n R_a}{K_2 K_m} \\ &= (\frac{T_m K_m K_2}{2K_1 R_a \delta})(\frac{R_a}{K_2 K_m}) \\ &= \frac{T_m}{2K_1 \delta} \end{aligned} \tag{3.49}$$

Putting the values of Km and Kn in the main transfer function, we get:

$$\frac{\omega(s)}{\omega(s)(ref)} = \frac{K_o}{S^3 T_m T_n \delta + S^2 T_m T_n + (K_o K_1 T_n) S + K_o K_1} \tag{3.50}$$

$$\frac{\omega(s)}{\omega(s)(ref)} = \frac{(T_m/2K_1\delta)}{S^3 T_m T_n \delta + S^2 T_m T_n + (T_m/2K_1\delta)(K_1 T_n) S + (T_m/2K_1\delta) K_1} \tag{3.51}$$

$$\frac{\omega(s)}{\omega(s)(ref)} = \frac{(T_m/2K_1\delta)}{T_m[S^3 4\delta^2 + S^2 4\delta + (1/2K_1\delta)K_1(4\delta)S + (1/2K_1\delta)K_1]} \tag{3.52}$$

$$\frac{\omega(s)}{\omega(s)(ref)} = (\frac{1}{2K_1\delta})[\frac{1}{4\delta^2 S^3 + 4\delta S^2 + 2S + (1/2\delta)}] \tag{3.53}$$

$$\frac{\omega(s)}{\omega(s)(ref)} = \frac{1}{K_1 + 4\delta S K_1 + 8\delta^2 S^2 K_1 + 8\delta^3 S^3 K_1} \tag{3.54}$$

3.3.6 Representation of Chopper in Transfer Function Form

A chopper is an “on” or “off” semiconductor switch which takes a fixed input voltage and gives variable DC output voltage and it works on the principle known as Pulse Width Modulation technique. So there is no time delay in its operation and hence, it can be represented by a simple constant gain K_t. The complete layout for separately excited DC motor speed control using PI controller is shown in the Figure 3.5.

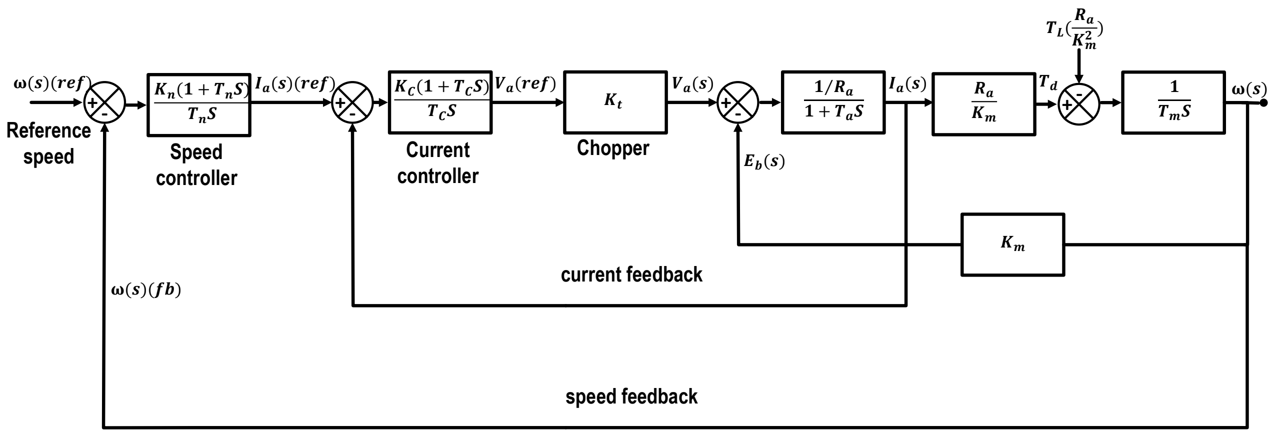


Figure 3.5: Complete layout for separately excited DC motor speed control using PI controller

As shown in the Figure (3.5), the speed control of separately excited DC motor using PI controller is accompanied by using feedback loops for both speed and current controllers. The PI controller operates with the concept of adjusting the deviation to the reference set value. Hence, it uses the feedback loops to calculate the error.

3.4 Fuzzy Controller Design

In the design of fuzzy logic controller the three main linguistic variables: Error, Change in Error and Control (Output) has to be defined based on the problem statement specified. Hence, for this research work these linguistic variables are defined as shown in the Table 3.1, 3.2 and 3.3, and also these variables are shown graphically in Figure 3.6, 3.7 and 3.8.

Table 3.1: Fuzzy sets and the respective membership functions for Error

Linguistic Value	Notation	Numerical Range	Membership Function
Negative	N	[-1 -1 -0.5 0]	Trapezoidal
Zero	Z	[-0.01 0 0.01]	Triangular
Positive	P	[0 0.5 1 1]	Trapezoidal

Table 3.1 shows the linguistic values with their respective numerical ranges and also membership functions selected for the input Error. One triangular and two trapezoidal membership functions with ranges from 0.01 and 1.0 respectively are used based on the maximum rated speed of the motor which is 1500 rpm and hence, this range is suitable for DC motor speed control.

Table 3.2: Fuzzy sets and the respective membership functions for Change in Error

Linguistic Value	Notation	Numerical Range	Membership Function
Negative	N	[-1 -1 0]	Triangular
Positive	P	[0 1 1]	Triangular

Table 3.2 shows the ranges of membership functions of linguistic variable Change in Error with their respective membership functions and linguistic values. Two triangular membership functions are used with ranges of 1.0.

Table 3.3: Fuzzy sets and the respective membership functions for Control

Linguistic Value	Notation	Numerical Range	Membership Function
Negative Large	NL	[-30 -25 -20]	Triangular
Negative Small	NS	[-15 -10 -5]	Triangular
Zero	Z	[-0.1 0 0.1]	Triangular
Positive Small	PS	[5 10 15]	Triangular
Positive Large	PL	[20 25 30]	Triangular

Table 3.3 shows five membership functions for output Control with all triangular type of membership function. And the notations of these five Control membership functions with their respective ranges of 10 and 0.1 are also described.

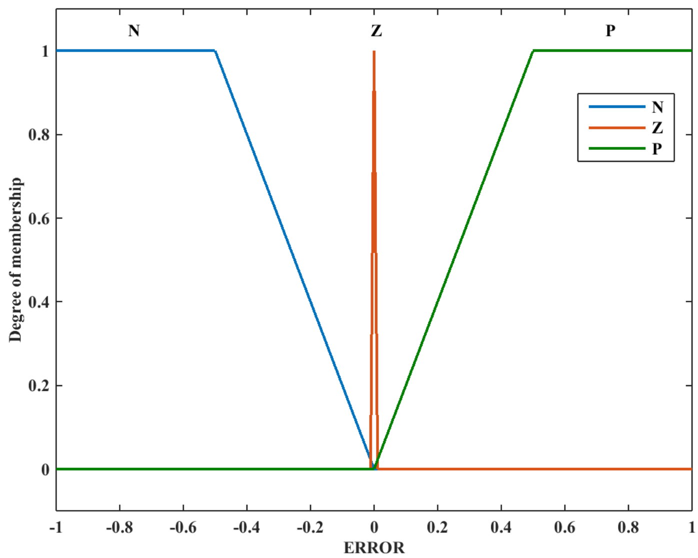


Figure 3.6: Membership function plots for input variable Error

Figure 3.6 shows the different types of membership functions for input Error with their respective ranges. In input Error three linguistic values are considered and of these the two inputs are with trapezoidal membership functions whereas the other one is with triangular membership function.

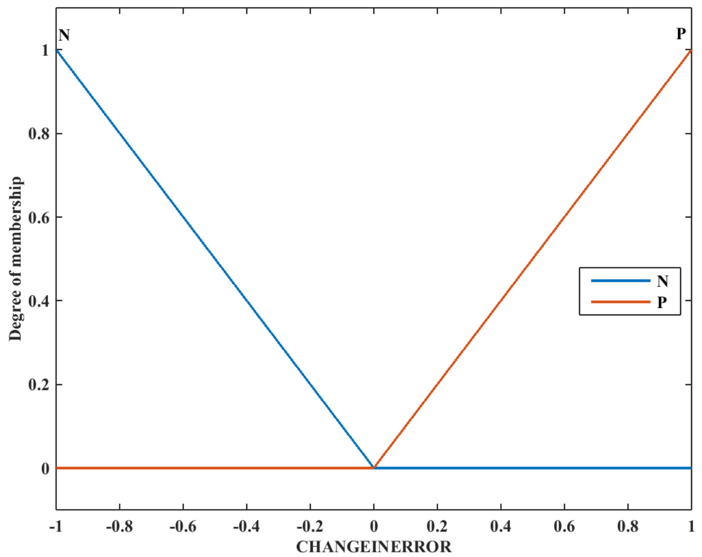


Figure 3.7: Membership function plots for input variable Change in Error

Figure 3.7 shows the same types of membership functions in different colors for input Change in Error with their respective ranges and notations. In input Change in Error two linguistic values are considered and all of them are with triangular membership functions.

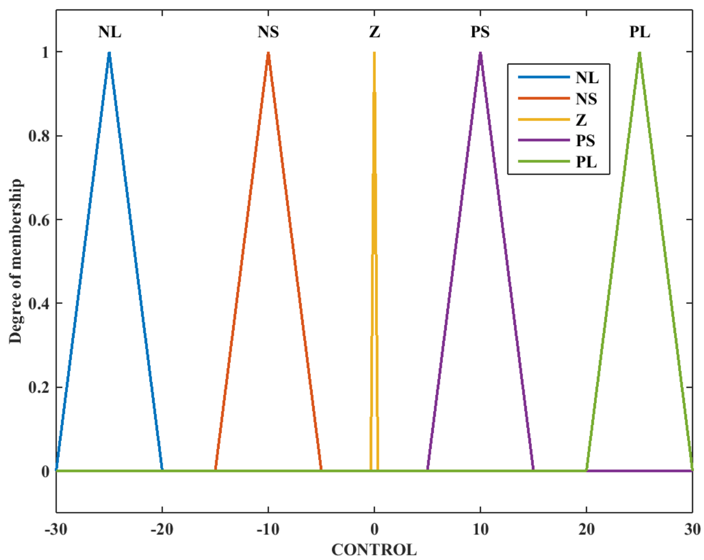


Figure 3.8: Membership function plots for output variable Control

Figure 3.8 shows the same types of membership functions for output Control in different colors with their respective ranges and notations. In output Control five linguistic values are considered and all of them are with triangular membership functions.

3.4.1 Rules

When designing fuzzy logic speed controller, the general rules are defined as, if the measured motor speed is less than the desired reference speed then speed up the motor and if the measured motor speed is more than the reference speed then slow down its speed. There are three linguistic variables of Error, two linguistic variables of Change in Error and five linguistic variables of Control from which 5 rules in fuzzy controller are written. In order to generate the necessary reference armature voltage signal ($V_a(ref)$) that will control the speed of the motor using FLC, the input speed error should be minimized. This is because as the input speed error is getting bigger, i.e. the controller input is becoming larger and hence it will results in poor motor speed performances. In addition to this changing of the input error plays an important role to define controller output. The 5 corresponding fuzzy rules are defined as follows.

1. If (ERROR is N) and (CHANGEINERROR is N) then (CONTROL is NL) (1)
2. If (ERROR is P) and (CHANGEINERROR is N) then (CONTROL is PS) (1)
3. If (ERROR is N) and (CHANGEINERROR is P) then (CONTROL is NS) (1)
4. If (ERROR is P) and (CHANGEINERROR is P) then (CONTROL is PL) (1)
5. If (ERROR is Z) then (CONTROL is Z) (1)

Fuzzy MATLAB code for 3 membership functions of Error, 2 membership functions of Change in Error and 5 membership functions of Control is described in the appendices. And also the fuzzy rules are tuned and the number of Error membership functions are updated to 131 with 133 fuzzy rules keeping other input and output membership functions as it was. The MATLAB code and fuzzy rules for the tuned fuzzy logic controller is described in the appendices.

Different linguistic value notations in combine with numbers are used for naming of membership functions. These notations (Z, S, M, L, B, H and G) are abbreviated form of zero, small, medium, large, big, huge and giant respectively. The arrangement of the notations together with numbers shows the proximity of the ranges from zero in either positive (P) or negative (N) direction. As farness increases its value also increases in their respective direction and vice verse.

As shown in the Figure 3.9 and Figure 3.10, the operation for the speed control of separately excited DC motor using four quadrants chopper with an armature voltage control method is discussed as follows. According to armature voltage control method used in this research work, reference speed is set with five different cases considered in the motoring and regenerative modes. Then, this reference speed ($\omega_n(ref)$) in rpm is converted to rad/sec.

The motor speed value measured (ω_n) is fed back and subtracted from the reference speed set to generate an error of speed. Initially, the motor speed feedback is zero. The speed error generated and its change are fed into the fuzzy speed controller as the two inputs.

From this fuzzy speed controller, a reference armature current signal ($I_a(ref)$) is generated. The feedback motor armature current measured is subtracted from the reference armature current generated and results in an error of armature current signal. The armature current signal error and its change are given to the fuzzy current controller as inputs. In the same way to that of initial speed, armature current feedback initially is zero and hence, in order to avoid high overshoot of current, armature current is limited to its maximum and minimum limits using current saturation.

From the fuzzy current controller, a reference armature voltage signal is generated and it will be multiplied by the chopper gain. Then, this reference armature voltage signal is sent to the comparator in order to identify the voltage sign (as positive or negative) in comparison with comparator triangular carrier wave and thus, it turns the respective switches on/off accordingly.

It is known that, the reference armature voltage sign depends on the operation modes selected (i.e. dependent on the required reference speed set). As the reference armature voltage signal is passing through the selected chopper switches, the required armature voltage will be generated and this is given to the motor as input in order to control the motor speed according to the reference speed set.

In this thesis work, the simulations results obtained for both fuzzy logic controller and PI controller are performed using separately excited DC motor and its specification is shown in the Table 3.4.

Table 3.4: Separately excited DC motor specification

Parameter	**Value**
Armature Resistance (R_a)	1 Ω
Armature Inductance (L_a)	0.5 H
Inertia (J)	0.01 Kg-m^2
Damping(B)	0.1 N-m-s
Motor Base Speed	1500 rpm

The overall speed control of separately excited DC motor with four quadrant chopper using fuzzy logic is shown in the Simulink models of Figure 3.9 and Figure 3.10.

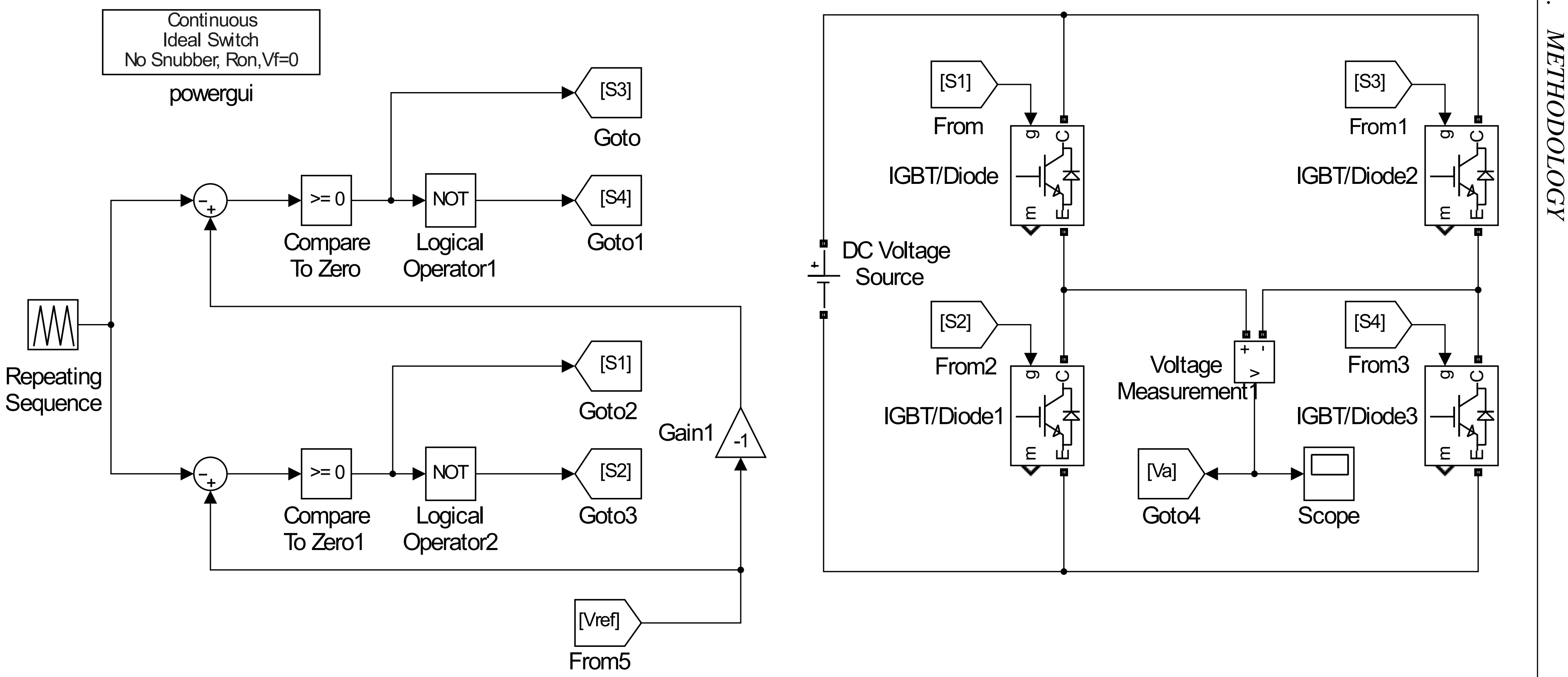


Figure 3.9: Block model for four quadrant chopper design

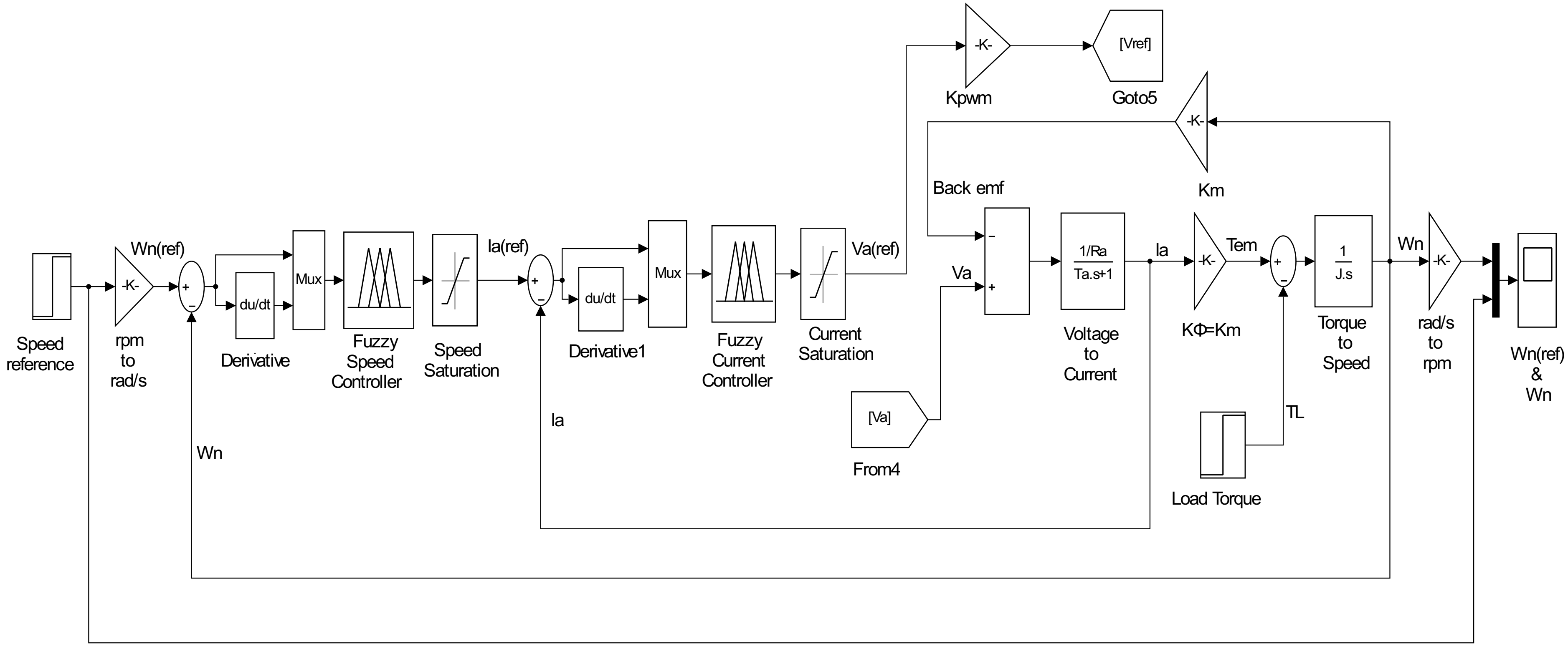


Figure 3.10: Simulink model for speed control of separately excited DC motor using four quadrant chopper

CHAPTER FOUR

RESULT ANALYSIS AND DISCUSSION

In this thesis, the speed control of separately excited DC motor is performed using fuzzy logic controller for all four quadrants of operations. In order to show its comparative advantages over PI controller, simulations are performed and evaluations are made. In addition to this, the effect of number of input Error membership functions on the speed performance is discussed. In general, in this chapter the results are stated and discussions are made under the following categories.

- Effect of number of Error membership functions on the speed performance
- Simulation results of FL and PI controllers
- Comparison of controllers on speed performance

4.1 Effect of Number of Error Membership Functions on the Speed Performance

As shown in the Table 4.1, in order to observe the effects of number of input Error membership functions on the speed performance of the motor different time-domain specifications are used. These specifications are defined as follows.

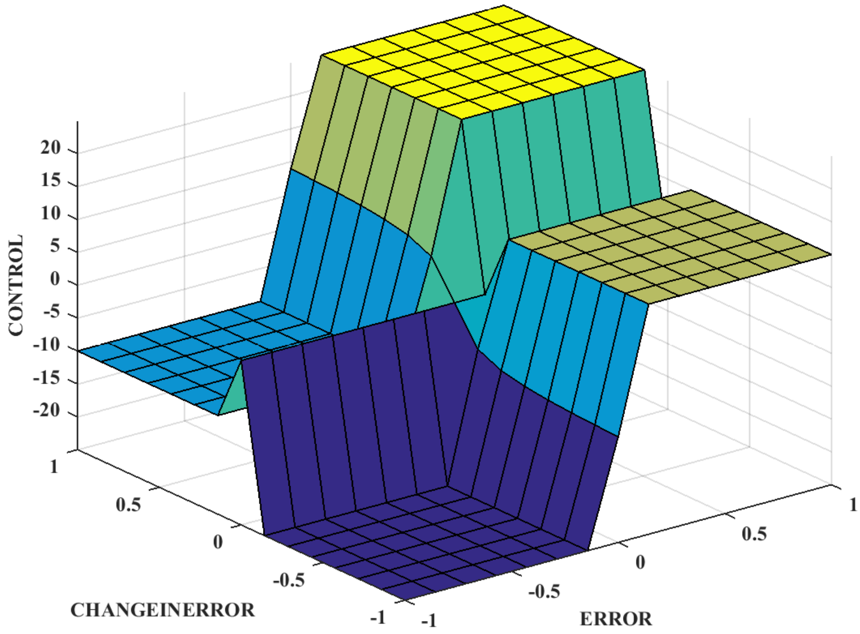
- **Delay time:** It is the time required for the response to reach half the final value the very first time.
- **Rise time:** It is the time required for the response to rise from 10% to 90%, 5% to 95%, or 0% to 100% of its final value.
- **Peak time:** It is the time required for the response to reach the first peak of the overshoot.
- **Maximum overshoot:** It is the maximum peak value of the response curve measured from reference input. The amount of the maximum (percent) overshoot directly indicates the relative stability of the system.
- **Settling time:** It is the time required for the response curve to reach and stay within a range about the final value of size specified by absolute percentage of the final value (usually 2% or 5%).

Table 4.1: Speed transient response for different number of membership functions

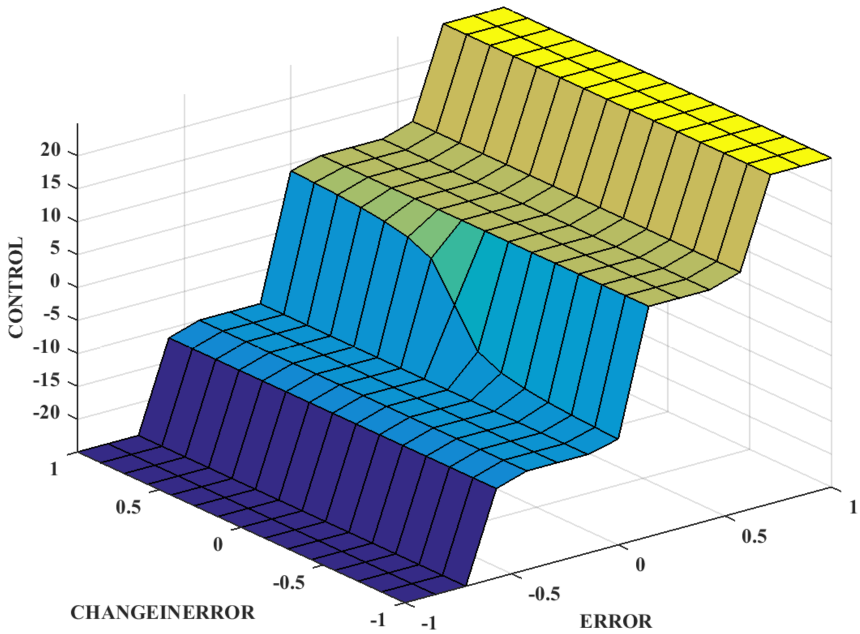
No. of input error mf	**Maximum overshoot M_p (%)**	**Peak time t_p(sec)**	**Rise time t_r(sec)**	**Settling time t_s(sec)**	**Delay time t_d(sec)**	**Measured speed (RPM)**
3	3.352142	0.596	0.421	0.624	0.296	786.6910
5	2.224445824	0.514	0.344	0.532	0.251	788.1109
7	2.140593256	0.514	0.344	0.527	0.251	789.0663
9	2.061941019	0.514	0.345	0.522	0.252	789.9644
11	2.03227283	0.514	0.345	0.519	0.252	790.1941
13	2.017378754	0.516	0.345	0.520	0.252	790.7241
15	1.969806033	0.514	0.346	0.514	0.252	791.0322
17	1.940608472	0.514	0.346	0.514	0.252	791.3755
19	1.908329361	0.514	0.346	0.514	0.252	791.7606
21	1.872302513	0.514	0.346	0.514	0.252	792.1957
23	1.8418017	0.479	0.333	0.479	0.219	792.6912
25	1.8418017	0.479	0.333	0.479	0.219	792.6912
27	1.861391752	0.435	0.303	0.435	0.183	792.9658
29	1.899098297	0.417	0.286	0.417	0.171	792.9658
31	1.973656431	0.388	0.258	0.388	0.155	793.3853
33	2.2140074	0.371	0.242	0.381	0.148	793.5791
35	2.713174632	0.355	0.227	0.372	0.142	793.7823
37	3.008326246	0.346	0.218	0.366	0.139	793.9232
39	3.339837985	0.336	0.208	0.359	0.137	794.0984
41	3.620653751	0.327	0.2	0.352	0.135	794.2654
43	3.880604605	0.319	0.193	0.346	0.133	794.4226
45	4.119366378	0.313	0.187	0.412	0.133	794.5688
47	4.343297185	0.308	0.183	0.420	0.132	794.7027
49	4.643670028	0.302	0.177	0.426	0.132	794.8756
51	4.804693141	0.299	0.175	0.429	0.132	794.9644
53	5.028120988	0.295	0.172	0.432	0.132	795.0823

Table 4.1 shows the speed transient responses of the motor as the number of input Error membership functions is increased. In order to observe its effects on the speed performance, the number and range of membership functions for input Change in Error and output Control are kept the same throughout the different number of Error membership functions considered. Hence, membership functions of 2 and 5 are used for Change in Error and Control respectively.

The effect of number of input Error membership functions on the speed performance using surface views is also shown in the Figure 4.1. To show its effect clearly, number of input Error membership functions are taken as 3 and 131 respectively.



(a) $mf = 3$



(b) $mf = 131$

Figure 4.1: Surface views (a) for 3 Error membership functions and (b) for 131 Error membership functions

The surface views in Figure 4.1 show the effect of number of Error membership functions on the speed performance of the separately excited DC motor clearly. Hence, in order to observe the effects of increasing number of input Error membership function on the motor speed performance clearly, the input Error has taken with the lowest and highest number of membership functions considered in this research work. Accordingly, the number of input Error membership functions has increased from 3 to 131 with other number of input membership function (Change in Error) and output membership function (Control) kept constant. Hence, the surface view with 131 input Error membership functions shows that the motor touches the reference speed in smoother way and can reach to the maximum reference speed set with better speed performance as compared to the surface view with 3 input Error membership functions.

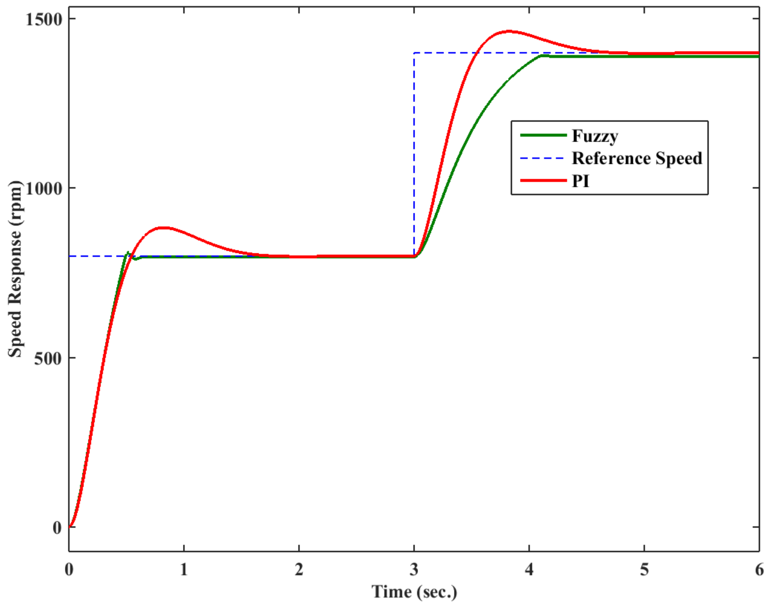
From the Table 4.1 and Figure 4.1 it has observed that the transient performance of the motor is not only dependent on the number of membership functions but also dependent on the type of membership function selected and the range of values set for these selected membership functions. But in general, from the simulation results obtained an improvement of reference speed tracking is observed as the number of input Error membership functions increases. This is because as the number of input Error membership functions increases, each speed errors are easily characterized in order to apply the respective control action. Hence, in order to obtain the required transient and steady state responses, selection of the type of membership function and the range of values should be made with respect to the specific application areas of the motor to be used.

4.2 Simulation Results of FL and PI Controllers

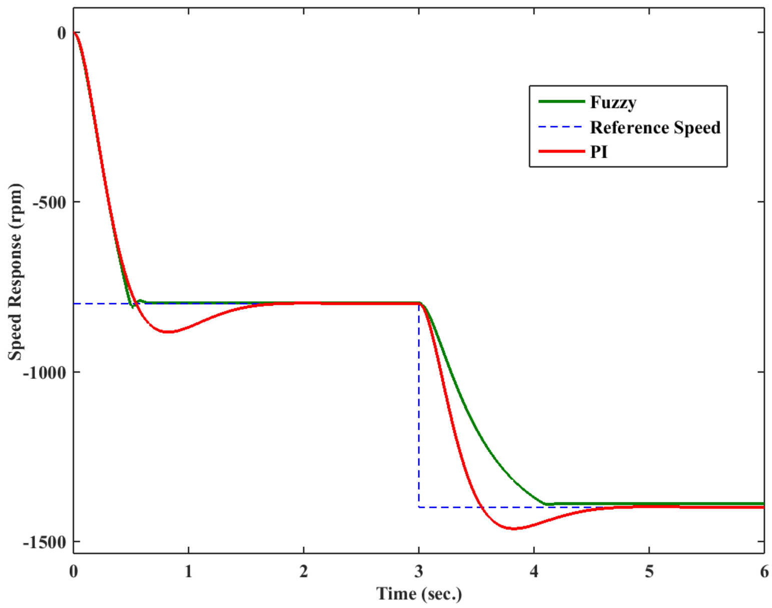
The simulation results show the speed response for separately excited DC motor using both Fuzzy and PI controller. Different circumstances of speed and load torque under motoring and regenerative modes are investigated with these controllers. These considered cases are,

- Different speed and load torque values in the motoring and regenerative modes
- Constant speed and torque in the motoring and regenerative modes
- Different speed values in the motoring and regenerative modes
- Different load torque values in the motoring and regenerative modes
- During changing in the forward and reverse motoring and regenerative modes

In the Figure 4.2 speed responses of the motor have performed at different speed and load torque values in both forward and reverse motoring modes.



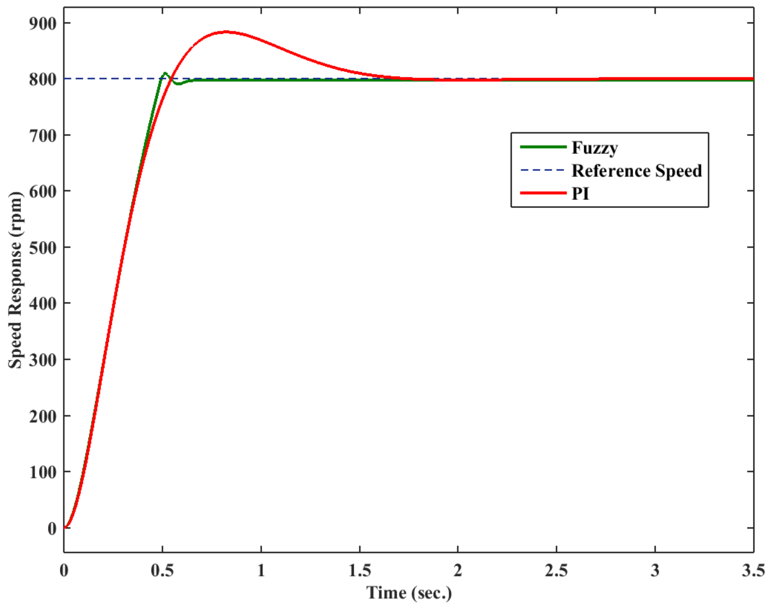
(a) forward motoring mode



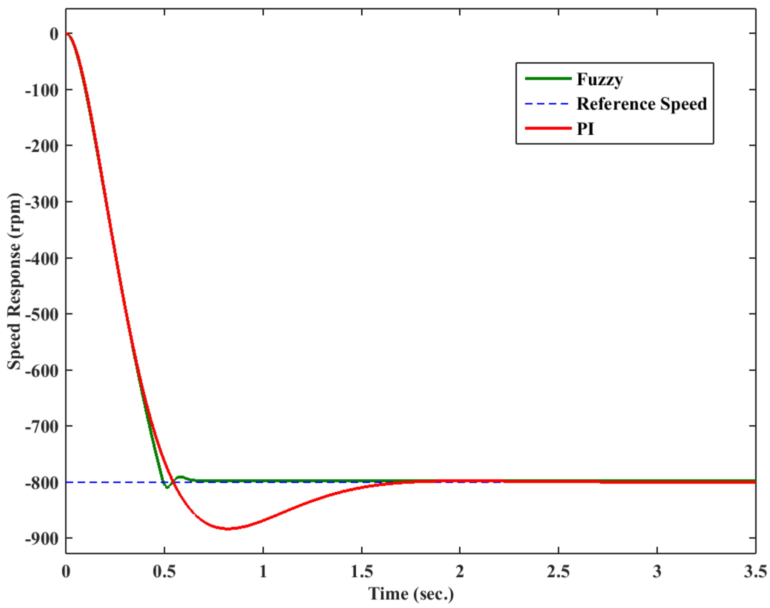
(b) reverse motoring mode

Figure 4.2: Speed response at different speed and load torque values in the motoring modes

Figure 4.2 shows the simulations of speed response at different speed and load torque values. In this case a change of speed and load torque values are made. These different speed values are 800 rpm and 1400 rpm for forward motoring, and -800 rpm and -1400 rpm for reverse motoring modes respectively. Whereas, the applied load torque values are 0.0002 N. m and 0.0006 N. m for forward and reverse motoring modes respectively. The speed and load torque value variations are made at an interval of 3 seconds. In the Figure 4.3 speed responses of the motor have performed at constant speed and load torque values in both forward and reverse motoring modes as shown.



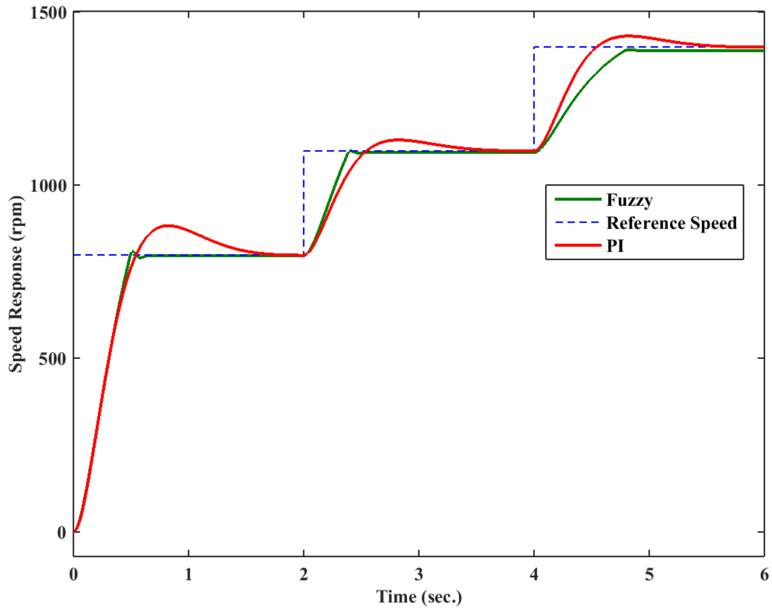
(a) forward motoring mode



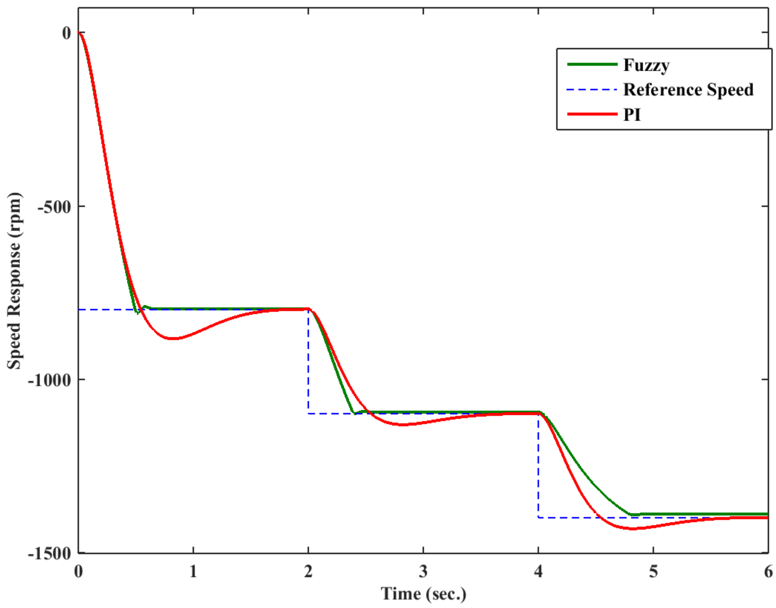
(b) reverse motoring mode

Figure 4.3: Speed response at constant speed and torque in the motoring modes

In the Figure 4.3 the simulation is carried out with constant speed and constant load torque of 800 rpm and 0.0002 N. m for forward motoring, and -800 rpm and -0.0002 N. m for reverse motoring mode respectively. In the Figure 4.4 speed responses of the motor have performed at different speed and constant load torque values in both forward and reverse motoring modes.



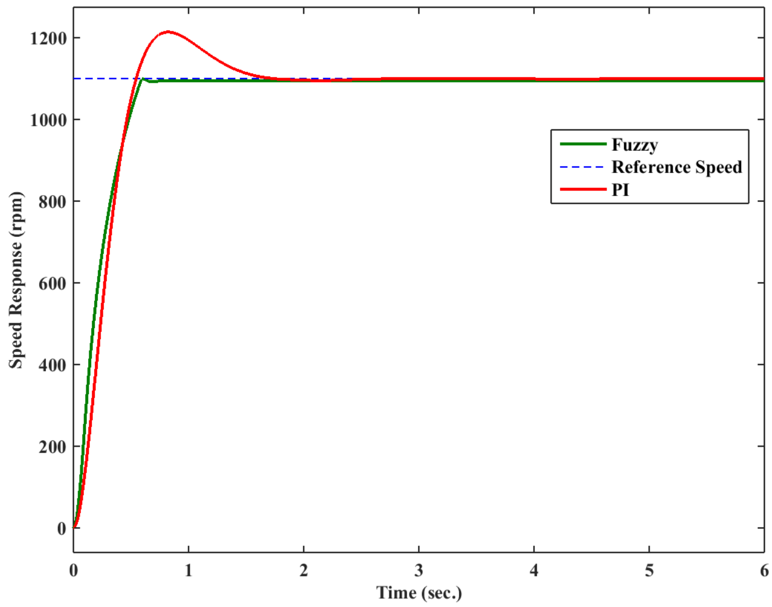
(a) forward motoring mode



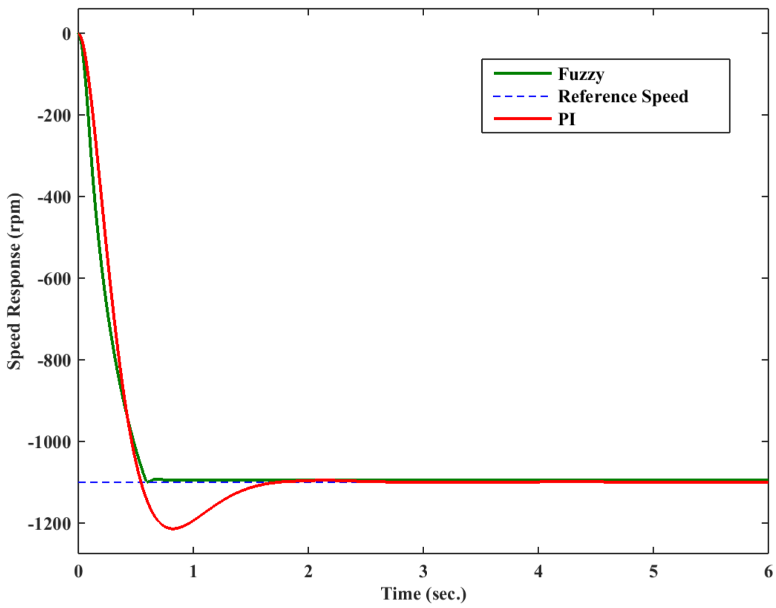
(b) reverse motoring mode

Figure 4.4: Speed response at different speed values in the motoring modes

Figure 4.4 shows the simulations of speed response at different speed values keeping load torque constant. In this case a stair types of speed values are taken. These different speed values are 800 rpm, 1100 rpm and 1400 rpm for forward motoring and -800 rpm, -1100 rpm and -1400 rpm for reverse motoring modes respectively. The speed variations are made at an interval of 2 seconds. A constant load torque of 0.0004 N. m and -0.0004 N. m is applied throughout the simulation for forward motoring and reverse motoring modes respectively. In the Figure 4.5 speed responses of the motor have performed at different load torque and constant speed values in both forward and reverse motoring modes.



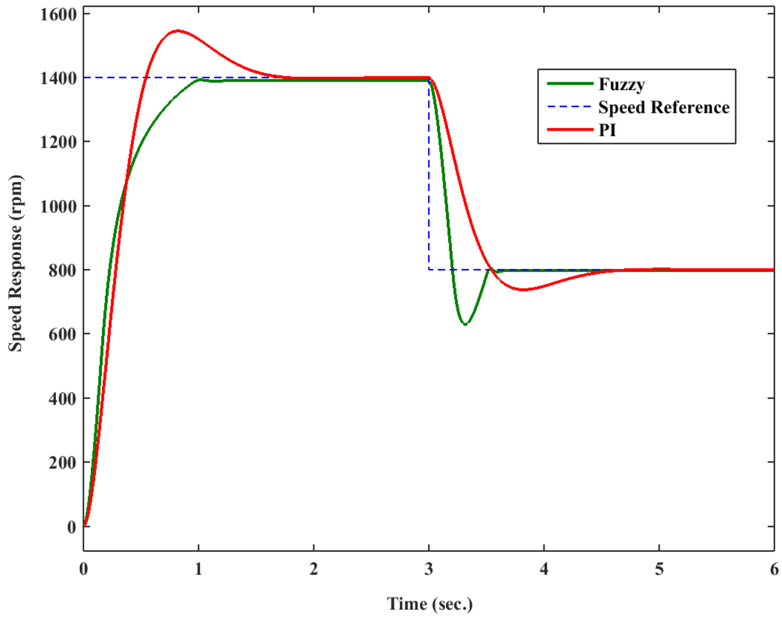
(a) forward motoring mode



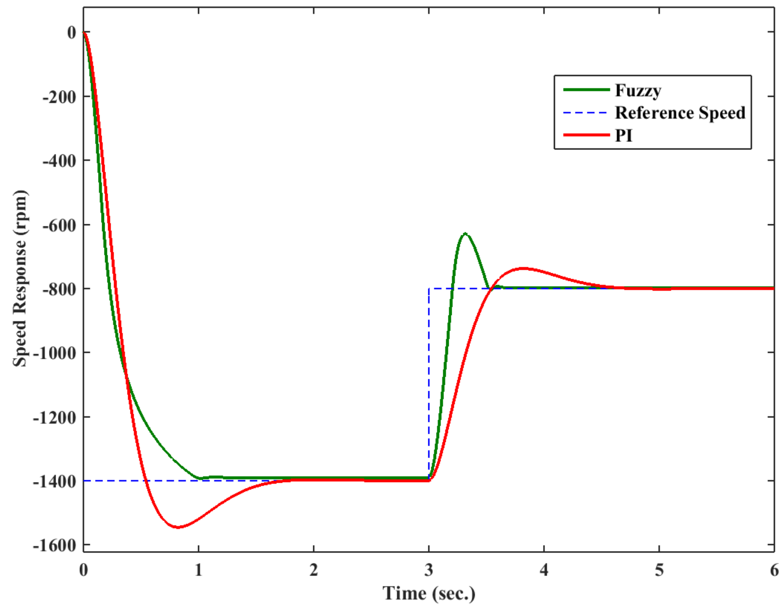
(b) reverse motoring mode

Figure 4.5: Speed response at different load torque values in the motoring modes

Figure 4.5 shows the speed responses at different load torque values keeping speed values constant. The load torque values are set to 0.0002 N. m, 0.0004 N. m and 0.0006 N. m for forward motoring and -0.0002 N. m, -0.0004 N. m and -0.0006 N. m for reverse motoring respectively with an interval of 2 seconds. Throughout the simulation the speed of the motor is adjusted with 1100 rpm and -1100 rpm for forward motoring and reverse motoring modes respectively. In the Figure 4.6 speed responses of the motor have performed at different speed and load torque values in both forward and reverse regenerative modes.



(a) forward regenerative mode

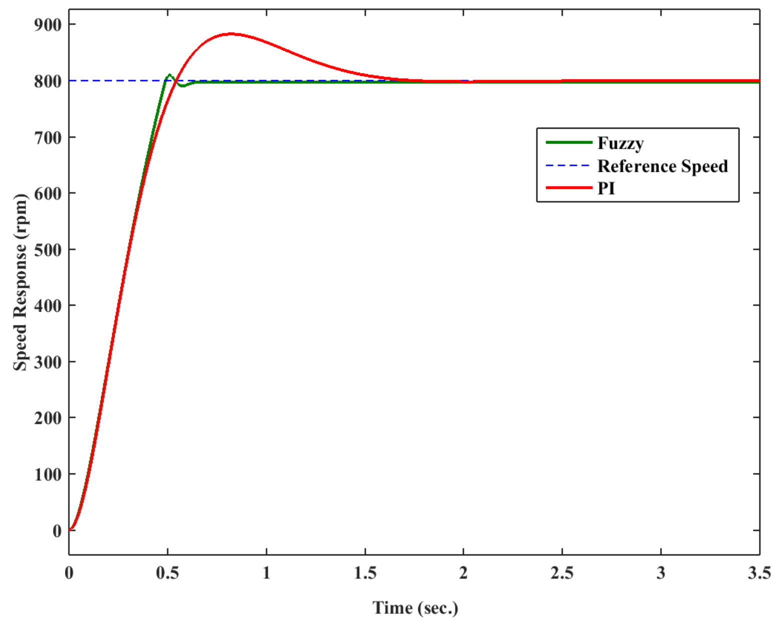


(b) reverse regenerative mode

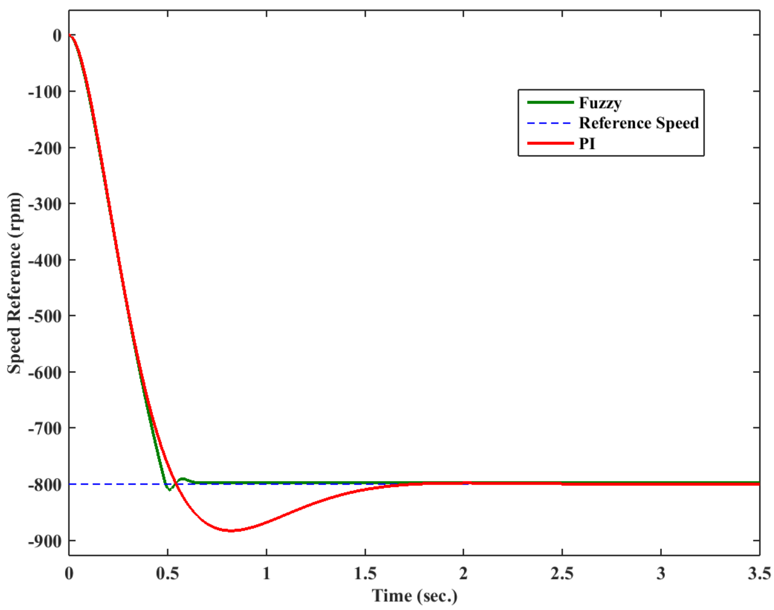
Figure 4.6: Speed response at different speed and load torque values in the regenerative modes

Figure 4.6 shows the simulations of speed response at different speed and load torque values. In this case a change of speed and load torque values are made. These different speed values are 1400 rpm and 800 rpm for forward regenerative, and -1400 rpm and -800 rpm for reverse regenerative modes respectively. Whereas the applied load torque values are

-0.0006 N. m and -0.0002 N. m for forward and reverse regenerative modes respectively. The speed and load torque value variations are made at an interval of 3 seconds. In the Figure 4.7 speed responses of the motor have performed at constant speed and load torque values in both forward and reverse regenerative modes.



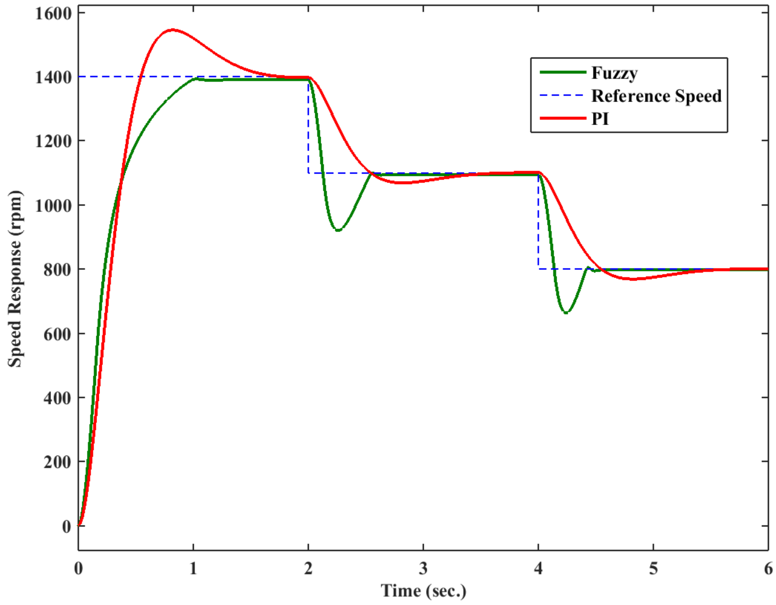
(a) forward regenerative mode



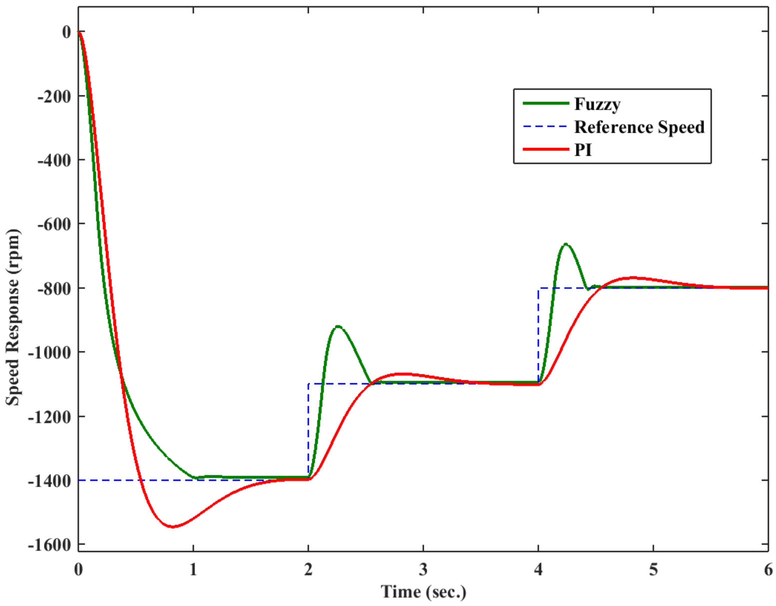
(b) reverse regenerative mode

Figure 4.7: Speed response at constant speed and torque in the regenerative modes

In the Figure 4.7 a constant speed and load torque is applied throughout the simulation. These speed and load torque values are of 800 rpm, -0.0002 N.m, and -800 rpm, 0.0002 N.m for forward regenerative and reverse regenerative modes respectively. In the Figure 4.8 speed responses of the motor have performed at different speed and constant load torque values in both forward and reverse regenerative modes.



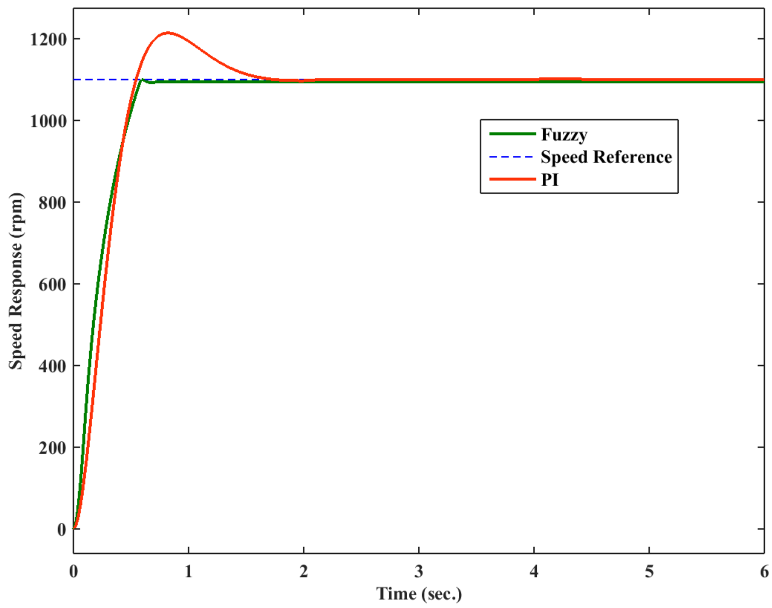
(a) forward regenerative mode



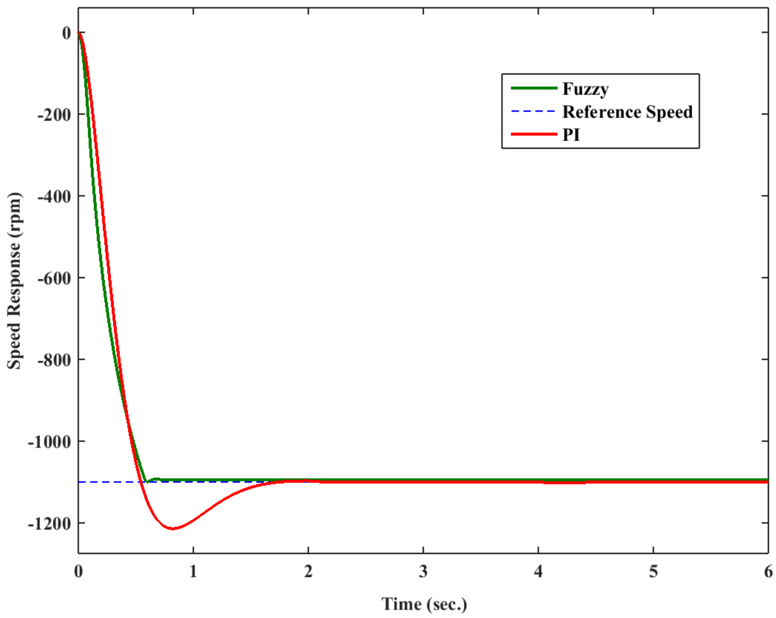
(b) reverse regenerative mode

Figure 4.8: Speed response at different speed values in the regenerative modes

The simulations of speed responses at different speed values with constant load torque are shown in Figure 4.8. A stair types of speed with values of 1400 rpm, 1100 rpm, 800 rpm, and -1400 rpm, -1100 rpm, -800 rpm are taken for forward and reverse regenerative modes respectively. The speed variations are kept at an interval of 2 seconds. A constant load torque of -0.0004 N. m and 0.0004 N. m is applied throughout the simulation for forward and reverse regenerative modes respectively. In the Figure 4.9 speed responses of the motor have performed at different load torque and constant speed values in both forward and reverse regenerative modes.



(a) forward regenerative mode



(b) reverse regenerative mode

Figure 4.9: Speed response at different load torque values in the regenerative modes

Figure 4.9 shows the simulation speed responses of the motor at different load torque values with speed value kept constant. These different load torque values are set to -0.0002 N. m, -0.0004 N. m, -0.0006 N. m, and 0.0002 N. m, 0.0004 N. m, 0.0006 N. m for forward and reverse regenerative modes respectively with a time interval of 2 seconds. The motor speed throughout the simulation is adjusted with 1100 rpm and -1100 rpm for forward regenerative and reverse regenerative modes respectively. In the Figure 4.10 speed responses of the motor have performed during changing in the forward and reverse motoring modes with different speed and constant load torque values.

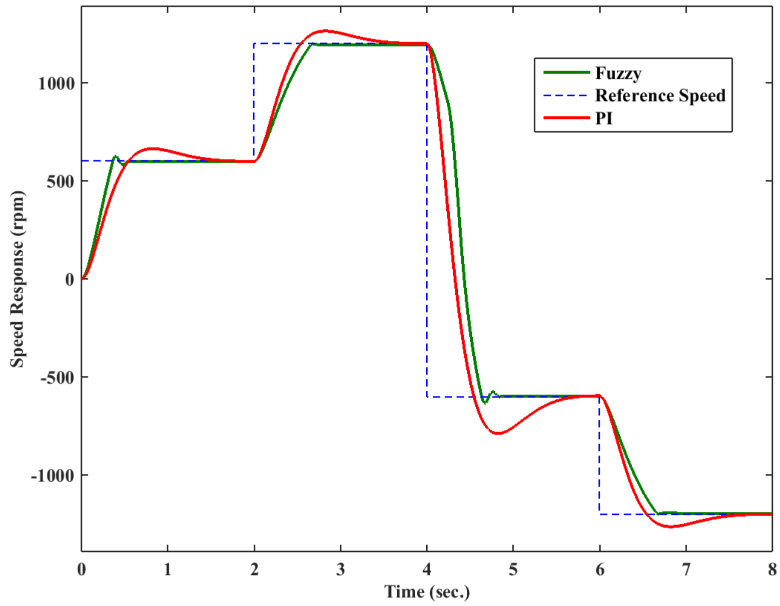


Figure 4.10: Speed response during changing in the forward and reverse motoring modes

In the Figure 4.10 it is shown that the speed response of the motor during changing of the speed in both forward and reverse motoring modes. These speed values are 600 N. m, 1200 N. m, -600 N. m and -1200 N. m and varies with an interval of 2 seconds. The load torque is kept constant at 0.0002 N. m for forward motoring with in the first 4 seconds and then a load torque of -0.0002 N. m is applied for reverse motoring modes within the next 4 seconds. In the Figure 4.11 speed responses of the motor have performed during changing in the forward and reverse regenerative modes with constant speed and load torque values.

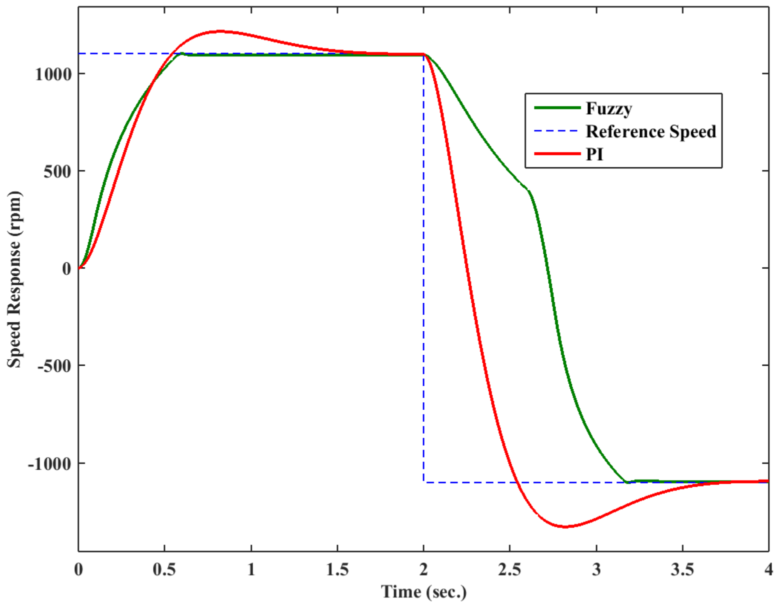


Figure 4.11: Speed response during changing in the forward and reverse regenerative modes

In the Figure 4.11 it has shown that the speed response of the motor during changing of the speed values from 1100 N. m to -1100 N. m in forward and reverse regenerative modes respectively. The load torque is kept constant at -0.0004 N. m and 0.0004 N. m for forward and reverse regenerative modes respectively. Changing of either the speed or load torque value is made at an interval of 2 seconds.

In all the simulation results of speed control of separately excited DC motor using fuzzy logic and PI controllers it has been observed the following. Generally, high overshoot with an increased peak time, increased settling time and zero steady state is experienced in the PI controllers whereas in fuzzy logic controller less overshoot with a decreased peak time, decreased settling time and non-zero steady state error is obtained. But when the speed tries to change into another speed level in the regenerative modes, high overshoot is experienced to attain fast settling time in fuzzy logic as compared to PI controller.

4.3 Comparison of Controllers on the Speed Performance

In the simulation results obtained, it is observed the speed responses of the motor using both fuzzy and PI controllers. As shown in Table 4.2 comparison of transient responses of both fuzzy and PI controllers is made. Comparison of these controllers is done under two different conditions that is before and after tuning of controller parameters.

Table 4.2: Comparison of controllers transient response

Transient parameters	Untuned controller		Tuned controller	
	PI	FL	PI	FL
M_p (%)	8.8465	3.352142	10.396875	1.550934602
t_p (sec.)	0.953	0.596	0.823	0.514
t_r (sec.)	0.429	0.421	0.368	0.349
t_s (sec.)	1.597	0.624	1.416	0.514
t_d (sec.)	0.285	0.296	0.254	0.254

The input parameters which are used for simulating both fuzzy logic and PI controllers are shown in the Table 4.3.

Table 4.3: Controller parameters before and after tuning

Controller type	Parameters	Before tuning	After tuning
PI	K_p	14	16.5792
	K_i	41.53311378	49.1847
FLC	No. of error mf	3	131
	No. of change in error mf	2	2
	No. of control mf	5	5
	No. of rules	5	133
	Range of mf	1	0.03

As shown in Table 4.3, before the fuzzy controller is tuned its number of error membership function was 3 and hence the number of fuzzy rules set are 5. But after tuning its membership functions are increased to 131 with the respective 133 fuzzy rules keeping other input and output parameters constant. The range of error membership functions is also reduced from 1 to 0.03 in order to share equally with the newly added error membership functions.

After tuning of both controllers, it has experienced a decrease in the values of settling time, rise time, peak time and delay time for the respective fuzzy and PI controllers and the percentage of maximum overshoot is reduced in fuzzy controller whereas an increment is observed in the PI controller. In general, when comparing both fuzzy and PI controllers better speed response has experienced with fuzzy logic controller as compared to PI controller.

CHAPTER FIVE

CONCLUSIONS AND RECOMMENDATIONS

5.1 Conclusion

This thesis work has presented the speed control of a separately excited DC motor with armature voltage control method using fuzzy logic controller. Up on doing this research work linear model for the separately excited DC motor is developed, four quadrants operation of DC chopper is discussed, both fuzzy and PI control systems are formulated, and simulations are performed for different motor speed and load torque values using MATLAB/Simulink software.

In this research work it has noticed/observed that even though transient responses of the motor does not depend only on the number of Error membership functions, increasing number of input Error membership functions has a direct improvement on the speed tracking and hence decreases the steady state errors.

From the simulation results of both controllers, better responses are obtained with fuzzy logic controller as compared to PI controller. And hence efficiency problems observed in the speed control of separately excited DC motor drives used for dynamically changing environments are solved by using fuzzy logic controller with four quadrants DC chopper.

5.2 Recommendation

In this thesis, simulations of speed control of separately excited DC motor is performed using fuzzy logic controller and effects of increasing number of Error membership functions on the speed performance are studied. Hence, from this research work the following points are recommended for further studies.

- Implement this research work in hardware to observe the actual feasibility.
- Evaluate the effect of number of membership functions of input Change in Error and output Control on the performance of the speed control.
- Enhance motor speed performance using various soft computing techniques such as GA, ANN, PSO and/or in combination with conventional controllers.
- Extend this control technique to other types of motors.

REFERENCES

[1] S. Tharani et al., “Speed control of a separately excited DC motor using optimization techniques,” *International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)*, vol. 2, no. 3, pp. 3924-3934, Mar. 2014.

[2] S. Tarei and S. Arora, “Comparison of speed control of DC motor using PID controller and various optimization techniques a review,” *International Journal of Engineering Science and Computing (IJESC)*, vol. 7, no. 7, pp. 13859-13864, Jul. 2017.

[3] V. Jambulingam, “Speed control of DC motor using MOSFET based chopper,” *International Journal of Engineering Development and Research (IJEDR)*, vol. 4, no. 2, pp. 1337-1341, 2016.

[4] V. R. Pote et al., “MATLAB simulation of speed control of DC motor by using chopper,” *International Journal of Emerging Technologies in Engineering Research (IJETER)*, vol. 5, no. 1, pp. 42-46, Jan. 2017.

[5] H. Rakholiya et al., “Design and simulation of speed control of DC motor using chopper,” *International Journal of Engineering Science and Computing (IJESC)*, vol. 7, no. 2, pp. 4523-4525, Feb. 2017.

[6] V. R. Vadapalli et al., “Speed control of DC motor using chopper,” *International Journal of Electrical and Electronics Research (IJEER)*, vol. 3, no. 1, pp. 289-295, Mar. 2015.

[7] S. Killedar et al., “Speed control of DC motor using chopper (MATLAB simulation),” *International Journal on Recent Technologies in Mechanical and Electrical Engineering (IJRMEE)*, vol. 3, no. 5, pp. 25-28, May 2016.

[8] A. Sadiq et al., “Field current speed control of direct current motor using fuzzy logic technique,” *International Journal of Information and Computation Technology (IJICT)*, vol. 3, no. 8, pp. 751-756, 2013.

[9] M. Suryawanshi et al., “MATLAB simulation on chopper based speed control of DC motor: A review,” *International Research Journal of Engineering and Technology (IRJET)*, vol. 4, no. 2, pp. 719-721, Feb. 2017.

[10] T. Saravanan and G. Saritha, “Analysis of chopper fed DC drive system with fuzzy logic speed control,” *Middle-East Journal of Scientific Research*, vol. 20, no. 11, pp. 1566-1571, 2014, doi: 10.5829/idosi.mejsr.2014.20.11.114169.

[11] T. C. Burg et al., “Velocity tracking control for a separately excited DC motor without velocity measurement,” in *Proc. American Control Conf. (ACC)*, Baltimore, MD, USA, Jun. 29 - Jul. 1, 1994, pp. 1051-1056, doi: 10.1109/ACC1994. 751908.

[12] A. K. Rajagiri et al., “Speed control of DC motor using fuzzy logic controller by PCI 6221 with MATLAB,” in *Proc. 1st Int. Conf. Sustain. Energy future Elect. Transp. (SeFet)*, Feb. 2019, pp. 1-6, doi: 10.1051/e3sconf/20198701004.

[13] B. Mahajan, “Modeling and simulation of DC motor driven geared system using Simulink,” *International Journal of Innovations in Engineering Research and Technology (IJIERT)*, IME 2019, pp. 71-75, Mar. 2019.

[14] S. A. Hamoodi et al., “DC motor speed control using PID controller implementation by Simulink and practical,” *International Journal of Electrical Engineering (IJEE)*, vol. 11, no. 1, pp. 39-49, 2018.

[15] M. Gokulkannan et al., “Performance improvement in separately excited DC motor using PI controller,” in *Proc. Int. Conf. Newer Eng. Concept Technol.(ICONNECT)*, Apr. 2018, pp. 1-6.

[16] A. A. Kosare et al., “Automatic closed loop speed control of DC motor using IGBT,” *International Journal of Recent Innovation in Engineering and Research (IJRIER)*, vol. 2, no. 5, pp. 57-64, May 2017.

[17] K. Shantanu and U. C. Bind, “Performance analysis of separately excited DC motor using PI and fuzzy logic controller,” *International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering (IJAREEIE)*, vol. 6, no. 5, pp. 4109-4119, May 2017, doi: 10.15662/IJAREEIE.2017.0605079.

[18] R. S. Hiware and K. S. Deshmukh, “Speed control of separately excited DC motor using chopper,” *International Research Journal of Engineering and Technology (IRJET)*, vol. 4, no. 1, pp. 799-803, Jan. 2017.

[19] R. Nagarajan et al., “Implementation of chopper fed speed control of separately excited DC motor using PI controller,” *International Journal of Engineering and Computer Science(IJECS)*, vol. 6, no. 3, pp. 20629-20633, Mar. 2017, doi: 10.18535/ijecs/v6i3.41.

[20] S. K. Suman and V. K. Giri, “Investigation and implementation of speed control of DC motor,” *Journal of Electrical Engineering (JEE)*, vol. 16, no. 2, pp. 315-322, Jan. 2016.

[21] E. Ozdemir et al., “Design and application of fuzzy controller for traction metro in metro vehicle,” *IFAC Transportation Systems*, vol. 30, no. 8, pp. 687-692, Jun. 1997.

[22] Z. M. Tun and L. Naing, “Double loop control of H-Bridge DC chopper fed permanent magnet DC drives using low cost hardware,” *World Academy of Science, Engineering and Technology: International journal of Electrical and Computer Engineering*, vol. 12, no. 11, pp. 857-866, Oct. 2018, doi: 10.5281/zenodo.2021977.

[23] D. Kerutagi et al., “Closed loop control of DC motor using firing angle,” *International Journal of Innovations in Engineering Research and Technology (IJIERT)*, vol. 5, no. 4, pp. 40-46, Apr. 2018.

[24] E. Hendawi et al., “Theoretical analysis and real time implementation of a classical controller with intelligent properties,” *Journal of Electrical Systems and Information Technology (JESIT)*, vol. 5, no. 1, pp. 23-34, May. 2018, doi: 10.1016/j.jesit.2017.12.001.

[25] S. Kumari and S. K. Swain, “Optimal control based PID tuning for control of four quadrant chopper fed DC motor,” in *Proc. Int. Conf. Tech. Smart-City Energy and Power (ICSESP)*, Bhubaneswar, India, Mar. 28-30, 2018, pp. 173-178, doi: 10.1109/icsesp.2018.8376695.

[26] A. R. Hambley, “DC machines,” in *Electrical Engineering Principles and Application*, A. Gilfillan, Ed., Upper Saddle River, NJ, USA: Prentice-Hall, 2014, Ch. 16, pp. 754-802.

[27] G. Rizzoni, “Introduction to electrical machines,” in *Principles and Applications of Electrical Engineering*, 5th ed. New York, NY, USA: McGraw-Hill, 2004, Ch. 17, pp. 827-888.

[28] A. Narendra et al., “PV fed separately excited DC motor with a closed loop speed control,” in *Proc. IEEE 8th Power India Int. Conf. (PIICON)*, Kurukshetra, India, Dec. 10-12, 2018, pp. 1-4, doi:10.1109/poweri.2018.8704449.

[29] M. A. Ali et al., “DC motor speed control using magnetic amplifier,” *International Journal of Recent Trends in Engineering and Research (IJRTER)*, vol. 4, no. 4, pp. 549-556, Apr. 2018, doi: 10.23883/IJRTER.2018.4273.TN5B3.

[30] J. R. Rashed, “Comparative study between conventional and intelligent methods for controlling on speed of DC shunt motor,” *Misan Journal for Academic Studies*, vol. 17, no. 33, pp. 33-49, Mar. 2018.

[31] A. Hughes and B. Drury,” DC motors,” in *Electric Motors and Drives: Fundamentals, Types and Applications*, 4th ed. New York, NY, USA: Elsevier, 2013, Ch. 4, pp. 73-111.

[32] B. Singh et al., “Modelling and designing of MATLAB based speed control of DC motor using PWM techniques,” *International Journal of Scientific Research in Science and Technology (IJSRST)*, vol. 3, no. 7, pp. 681-686, Sep./Oct. 2017.

[33] K. M. Saraswathy et al., “Four quadrant operation of BLDC motor suitable for DC micro grid for elevator application,” in *Proc. Int. Conf. Power, Control, Signals*

Computation (EPSCICON), Thrissur, India, Jan. 6-10, 2018, pp. 1-5, doi: 10.1109/EPSCICON.2018.8379599.

[34] S. Nadweh et al., "Using four-quadrant chopper with variable speed drive system DC-link to improve the quality of supplied power industrial facilities," *Majlesi Journal of Electrical Engineering (MJEE)*, vol. 13, no. 1, pp. 85-93, Mar. 2019.

[35] S. S. Metkar et al., "Four quadrant chopper drive use for speed control of DC motor," *International Journal for Scientific Research and Development (IJSRD)*," vol. 2, no. 12, pp. 747-749, Mar. 1, 2015.

[36] M. Faraz and D. - H. Kim, "An efficient actuator control mechanism using fuzzy logic on embedded systems," *International Journal of Control and Automation (IJCA)*," vol. 11, no. 6, pp. 35-44, Jan. 2018, doi: 10.14257/ijca.2018.11.6.04.

[37] N. Talpur et al., "An investigation of membership functions on performance of ANFIS for solving classification problems," in *Proc. IOP Conf. Ser.: Mater. Sci. Eng., in International Research and Innovation Submit (IRIS2017)*, 1st ed., Melaka, Malaysia, May 6-7, 2017, pp. 1-7.

[38] M. Irshad and A. Ali, "Optimal tuning rules for PI/PID controllers for inverse response processes," *IFAC-Papers Online*, vol. 51, no. 1, pp. 413-418, 2018.

[39] V. V. Kulkarni et al., "PID controller based DC motor speed control," *International Journal on Recent and Innovative Trends in Computing and Communication*, vol. 5, no. 9, pp. 35-38, Sep. 2017.

[40] Zain-Aldeen S. A. Rahman, "Design a fuzzy logic controller for controlling position of DC motor," *International Journal of Computer Engineering In Research Trends (IJCERT)*, vol. 4, no. 7, pp. 285-289, Jul. 2017, doi: 10.13140/RG.2.2.15206.88646.

[41] C. C. Lee, "Fuzzy logic in control systems: Fuzzy logic controller-part II," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 20, no. 2, pp. 419-435, Mar./Apr. 1990, doi: 10.1109/21.52552.

[42] P. Khanke and S. Jain, "Speed control of separately excited DC motor using various conventional controllers," *International Journal of Engineering Research and Applications (IJERA)*, vol. 5, no. 4, pp. 67-72, Apr. 2015.

[43] O. O. Adam Mohammed and A. T. Ali, "Comparative study of PID and fuzzy controllers for speed control of DC motor," *International Journal of Innovative Research in Science, Engineering and Technology (IJIRSET)*, vol. 3, no. 9, pp. 16104-16110, Sep. 2014, doi: 10.15680/IJIRSET.2014.0309045.

[44] W. Gubara et al. "Comparative study on the speed of DC motor using PID and FLC," in *Proc. Basic Sci. and Eng. Stud. Conf. (SGCAC)*, Khartoum, Sudan, Feb. 20-23, 2016, pp. 24-29, doi: 10.1109/SGCAC.2016.7458001.

[45] H. R. Hatem, "Control of pulse width modulation on direction and speed of DC motor using Arduino," *Journal of Engineering and Applied Sciences (JEAS)*, vol. 14, no. 5, pp. 1493-1497, Mar. 2019, doi: 10.3923/jeasci.2019.1493.1497.

[46] U. Hussain et al., "Speed and direction control of a DC motor using dual converter technique," *International Journal of Engineering Works (IJEW)*, vol. 6, no. 4, pp. 120-125, Apr. 2019.

[47] H. Jian, "Research on DC motor speed control system based on PID," in *Proc. 8th Int. Conf. Manuf. Sci. Eng. (ICMSE)*, Z. Y. Jiang, Ed. May 2018, pp. 671-674, doi: 10.2991/icmse-18.2018.123.

[48] A. A. Hassan et al., "Comparative study for DC motor speed control using PID controller," *International Journal of Engineering and Technology (IJET)*, vol. 9, no. 6, pp. 4181-4192, Dec. 2017, doi: 10.21817/ijet/2017/v9i6/170906069.

[49] K. Priyanka and A. Mariyammal, "DC motor speed control using PWM," *International Journal of Innovative Science and Research Technology (IJISRT)*, vol. 3, no. 2, pp. 584-587, Feb. 2018.

APPENDICES

Appendix A: Fuzzy MATLAB Code

% For 3 Error Membership Functions

[System]

```
Name='Dawit3'
Type='mamdani'
Version=2.0
NumInputs=2
NumOutputs=1
NumRules=5
AndMethod='prod'
OrMethod='max'
ImpMethod='prod'
AggMethod='sum'
DefuzzMethod='centroid'
```

[Input1]

```
Name='ERROR'
Range=[-1 1]
NumMFs=3
MF1='N':'trapmf',[-1 -1 -0.5 0]
MF2='Z':'trimf',[-0.01 0 0.01]
MF3='P':'trapmf',[0 0.5 1 1]
```

[Input2]

```
Name='CHANGEINERROR'
Range=[-1 1]
NumMFs=2
MF1='N':'trimf',[-1 -1 0]
MF2='P':'trimf',[0 1 1]
```

[Output1]

```
Name='CONTROL'
Range=[-30 30]
NumMFs=5
MF1='NL':'trimf',[-30 -25 -20]
MF2='NS':'trimf',[-15 -10 -5]
MF3='Z':'trimf',[-0.1 0 0.1]
MF4='PS':'trimf',[5 10 15]
```

MF5='PL':'trimf',[20 25 30]

[Rules]	1 2, 2 (1) : 1
1 1, 1 (1) : 1	3 2, 5 (1) : 1
3 1, 4 (1) : 1	2 0, 3 (1) : 1

% For 131 Error Membership Functions

[System]

Name='Dawit131'

Type='mamdani'

Version=2.0

NumInputs=2

NumOutputs=1

NumRules=133

AndMethod='prod'

OrMethod='max'

ImpMethod='prod'

AggMethod='sum'

DefuzzMethod='centroid'

[Input1]

Name='ERROR'

Range=[-1 1]

NumMFs=131

MF1='NG10':'trapmf',[-1 -1 -0.99 -0.975]

MF2='NG9':'trimf',[-0.975 -0.96 -0.945]

MF3='NG8':'trimf',[-0.96 -0.945 -0.93]

MF4='NG7':'trimf',[-0.945 -0.93 -0.915]

MF5='NG6':'trimf',[-0.93 -0.915 -0.90]

MF6='NG5':'trimf',[-0.915 -0.90 -0.885]

MF7='NG4':'trimf',[-0.90 -0.885 -0.87]

MF8='NG3':'trimf',[-0.885 -0.87 -0.855]

MF9='NG2':'trimf',[-0.87 -0.855 -0.84]

MF10='NG1':'trimf',[-0.855 -0.84 -0.825]

MF11='NH11':'trimf',[-0.84 -0.825 -0.81]

MF12='NH10':'trimf',[-0.825 -0.81 -0.795]

MF13='NH9':'trimf',[-0.81 -0.795 -0.78]

MF14='NH8':'trimf',[-0.795 -0.78 -0.765]

MF15='NH7':'trimf',[-0.78 -0.765 -0.75]

MF16='NH6':'trimf',[-0.765 -0.75 -0.735]

MF17=‘NH5’:‘trimf’,[-0.75 -0.735 -0.72]
MF18=‘NH4’:‘trimf’,[-0.735 -0.72 -0.705]
MF19=‘NH3’:‘trimf’,[-0.72 -0.705 -0.69]
MF20=‘NH2’:‘trimf’,[-0.705 -0.69 -0.675]
MF21=‘NH1’:‘trimf’,[-0.69 -0.675 -0.66]
MF22=‘NB11’:‘trimf’,[-0.675 -0.66 -0.645]
MF23=‘NB10’:‘trimf’,[-0.66 -0.645 -0.63]
MF24=‘NB9’:‘trimf’,[-0.645 -0.63 -0.615]
MF25=‘NB8’:‘trimf’,[-0.63 -0.615 -0.6]
MF26=‘NB7’:‘trimf’,[-0.615 -0.6 -0.585]
MF27=‘NB6’:‘trimf’,[-0.6 -0.585 -0.57]
MF28=‘NB5’:‘trimf’,[-0.585 -0.57 -0.555]
MF29=‘NB4’:‘trimf’,[-0.57 -0.555 -0.54]
MF30=‘NB3’:‘trimf’,[-0.555 -0.54 -0.525]
MF31=‘NB2’:‘trimf’,[-0.54 -0.525 -0.51]
MF32=‘NB1’:‘trimf’,[-0.525 -0.51 -0.495]
MF33=‘NL11’:‘trimf’,[-0.51 -0.495 -0.48]
MF34=‘NL10’:‘trimf’,[-0.495 -0.48 -0.465]
MF35=‘NL9’:‘trimf’,[-0.48 -0.465 -0.45]
MF36=‘NL8’:‘trimf’,[-0.465 -0.45 -0.435]
MF37=‘NL7’:‘trimf’,[-0.45 -0.435 -0.42]
MF38=‘NL6’:‘trimf’,[-0.435 -0.42 -0.405]
MF39=‘NL5’:‘trimf’,[-0.42 -0.405 -0.39]
MF40=‘NL4’:‘trimf’,[-0.405 -0.39 -0.375]
MF41=‘NL3’:‘trimf’,[-0.39 -0.375 -0.36]
MF42=‘NL2’:‘trimf’,[-0.375 -0.36 -0.345]
MF43=‘NL1’:‘trimf’,[-0.36 -0.345 -0.33]
MF44=‘NM11’:‘trimf’,[-0.345 -0.33 -0.315]
MF45=‘NM10’:‘trimf’,[-0.33 -0.315 -0.3]
MF46=‘NM9’:trimf’,[-0.315 -0.3 -0.285]
MF47=‘NM8’:‘trimf’,[-0.3 -0.285 -0.27]
MF48=‘NM7’:‘trimf’,[-0.285 -0.27 -0.255]
MF49=‘NM6’:‘trimf’,[-0.27 -0.255 -0.24]
MF50=‘NM5’:‘trimf’,[-0.255 -0.24 -0.225]
MF51=‘NM4’:‘trimf’,[-0.24 -0.225 -0.21]
MF52=‘NM3’:‘trimf’,[-0.225 -0.21 -0.195]
MF53=‘NM2’:‘trimf’,[-0.21 -0.195 -0.18]
MF54=‘NM1’:‘trimf’,[-0.195 -0.18 -0.165]
MF55=‘NS11’:‘trimf’,[-0.18 -0.165 -0.15]

MF56=‘NS10’:‘trimf’,[-0.165 -0.15 -0.135]
MF57=‘NS9’:‘trimf’,[-0.15 -0.135 -0.12]
MF58=‘NS8’:‘trimf’,[-0.135 -0.12 -0.105]
MF59=‘NS7’:‘trimf’,[-0.12 -0.105 -0.09]
MF60=‘NS6’:‘trimf’,[-0.105 -0.09 -0.075]
MF61=‘NS5’:‘trimf’,[-0.09 -0.075 -0.06]
MF62=‘NS4’:‘trimf’,[-0.075 -0.06 -0.045]
MF63=‘NS3’:‘trimf’,[-0.06 -0.045 -0.03]
MF64=‘NS2’:‘trimf’,[-0.045 -0.03 -0.015]
MF65=‘NS1’:‘trimf’,[-0.03 -0.015 0]
MF66=‘Z’:‘trimf’,[-0.01 0 0.01]
MF67=‘PS1’:‘trimf’,[0 0.015 0.03]
MF68=‘PS2’:‘trimf’,[0.015 0.03 0.045]
MF69=‘PS3’:‘trimf’,[0.03 0.045 0.06]
MF70=‘PS4’:‘trimf’,[0.045 0.06 0.075]
MF71=‘PS5’:‘trimf’,[0.06 0.075 0.09]
MF72=‘PS6’:‘trimf’,[0.075 0.09 0.105]
MF73=‘PS7’:‘trimf’,[0.09 0.105 0.12]
MF74=‘PS8’:‘trimf’,[0.105 0.12 0.135]
MF75=‘PS9’:‘trimf’,[0.12 0.135 0.15]
MF76=‘PS10’:‘trimf’,[0.135 0.15 0.165]
MF77=‘PS11’:‘trimf’,[0.15 0.165 0.18]
MF78=‘PM1’:‘trimf’,[0.165 0.18 0.195]
MF79=‘PM2’:‘trimf’,[0.18 0.195 0.21]
MF80=‘PM3’:‘trimf’,[0.195 0.21 0.225]
MF81=‘PM4’:‘trimf’,[0.21 0.225 0.24]
MF82=‘PM5’:‘trimf’,[0.225 0.24 0.255]
MF83=‘PM6’:‘trimf’,[0.24 0.255 0.27]
MF84=‘PM7’:‘trimf’,[0.255 0.27 0.285]
MF85=‘PM8’:‘trimf’,[0.27 0.285 0.3]
MF86=‘PM9’:‘trimf’,[0.285 0.3 0.315]
MF87=‘PM10’:‘trimf’,[0.3 0.315 0.33]
MF88=‘PM11’:‘trimf’,[0.315 0.33 0.345]
MF89=‘PL1’:‘trimf’,[0.33 0.345 0.36]
MF90=‘PL2’:‘trimf’,[0.345 0.36 0.375]
MF91=‘PL3’:‘trimf’,[0.36 0.375 0.39]
MF92=‘PL4’:‘trimf’,[0.375 0.39 0.405]
MF93=‘PL5’:‘trimf’,[0.39 0.405 0.42]
MF94=‘PL6’:‘trimf’,[0.405 0.42 0.435]

MF95=‘PL7’:‘trimf’,[0.42 0.435 0.45]
MF96=‘PL8’:‘trimf’,[0.435 0.45 0.465]
MF97=‘PL9’:‘trimf’,[0.45 0.465 0.48]
MF98=‘PL10’:‘trimf’,[0.465 0.48 0.495]
MF99=‘PL11’:‘trimf’,[0.48 0.495 0.51]
MF100=‘PB1’:‘trimf’,[0.495 0.51 0.525]
MF101=‘PB2’:‘trimf’,[0.51 0.525 0.54]
MF102=‘PB3’:‘trimf’,[0.525 0.54 0.555]
MF103=‘PB4’:‘trimf’,[0.54 0.555 0.57]
MF104=‘PB5’:‘trimf’,[0.555 0.57 0.585]
MF105=‘PB6’:‘trimf’,[0.57 0.585 0.6]
MF106=‘PB7’:‘trimf’,[0.585 0.6 0.615]
MF107=‘PB8’:‘trimf’,[0.6 0.615 0.63]
MF108=‘PB9’:‘trimf’,[0.615 0.63 0.645]
MF109=‘PB10’:‘trimf’,[0.63 0.645 0.66]
MF110=‘PB11’:‘trimf’,[0.645 0.66 0.675]
MF111=‘PH1’:‘trimf’,[0.66 0.675 0.69]
MF112=‘PH2’:‘trimf’,[0.675 0.69 0.705]
MF113=‘PH3’:‘trimf’,[0.69 0.705 0.72]
MF114=‘PH4’:‘trimf’,[0.705 0.72 0.735]
MF115=‘PH5’:‘trimf’,[0.72 0.735 0.75]
MF116=‘PH6’:‘trimf’,[0.735 0.75 0.765]
MF117=‘PH7’:‘trimf’,[0.75 0.765 0.78]
MF118=‘PH8’:‘trimf’,[0.765 0.78 0.795]
MF119=‘PH9’:‘trimf’,[0.78 0.795 0.81]
MF120=‘PH10’:‘trimf’,[0.795 0.81 0.825]
MF121=‘PH11’:‘trimf’,[0.81 0.825 0.84]
MF122=‘PG1’:‘trimf’,[0.825 0.84 0.855]
MF123=‘PG2’:‘trimf’,[0.84 0.855 0.87]
MF124=‘PG3’:‘trimf’,[0.855 0.87 0.885]
MF125=‘PG4’:‘trimf’,[0.87 0.885 0.9]
MF126=‘PG5’:‘trimf’,[0.885 0.9 0.915]
MF127=‘PG6’:‘trimf’,[0.9 0.915 0.93]
MF128=‘PG7’:‘trimf’,[0.915 0.93 0.945]
MF129=‘PG8’:‘trimf’,[0.93 0.945 0.96]
MF130=‘PG9’:‘trimf’,[0.945 0.96 0.975]
MF131=‘PG10’:‘trapmf’,[0.975 0.99 1 1]

[Input2]

Name=‘CHANGEINERROR’

Range=[-1 1]
NumMFs=2
MF1='N':'trimf',[-1 -1 0]
MF2='P':'trimf',[0 1 1]

[Output1]

Name='CONTROL'
Range=[-30 30]
NumMFs=5
MF1='NL':'trimf',[-30 -25 -20]
MF2='NS':'trimf',[-15 -10 -5]
MF3='Z':'trimf',[-0.1 0 0.1]
MF4='PS':'trimf',[5 10 15]
MF5='PL':trimf',[20 25 30]

[Rules]

1 0, 1 (1) : 1
2 0, 1 (1) : 1
3 0, 1 (1) : 1
4 0, 1 (1) : 1
5 0, 1 (1) : 1
6 0, 1 (1) : 1
7 0, 1 (1) : 1
8 0, 1 (1) : 1
9 0, 1 (1) : 1
10 0, 1 (1) : 1
11 0, 1 (1) : 1
12 0, 1 (1) : 1
13 0, 1 (1) : 1
14 0, 1 (1) : 1
15 0, 1 (1) : 1
16 0, 1 (1) : 1
17 0, 1 (1) : 1
18 0, 1 (1) : 1
19 0, 1 (1) : 1
20 0, 1 (1) : 1
21 0, 1 (1) : 1
22 0, 1 (1) : 1
23 0, 1 (1) : 1
24 0, 1 (1) : 1
25 0, 1 (1) : 1
26 0, 1 (1) : 1
27 0, 1 (1) : 1
28 0, 2 (1) : 1
29 0, 2 (1) : 1
30 0, 2 (1) : 1
31 0, 2 (1) : 1
32 0, 2 (1) : 1
33 0, 2 (1) : 1
34 0, 2 (1) : 1
35 0, 2 (1) : 1
36 0, 2 (1) : 1
37 0, 2 (1) : 1
38 0, 2 (1) : 1
39 0, 2 (1) : 1
40 0, 2 (1) : 1
41 0, 2 (1) : 1
42 0, 2 (1) : 1
43 0, 2 (1) : 1
44 0, 2 (1) : 1
45 0, 2 (1) : 1
46 0, 2 (1) : 1
47 0, 2 (1) : 1
48 0, 2 (1) : 1
49 0, 2 (1) : 1
50 0, 2 (1) : 1
51 0, 2 (1) : 1
52 0, 2 (1) : 1
53 0, 2 (1) : 1
54 0, 2 (1) : 1
55 0, 2 (1) : 1
56 0, 2 (1) : 1
57 0, 2 (1) : 1
58 0, 2 (1) : 1
59 0, 2 (1) : 1
60 0, 2 (1) : 1
61 0, 2 (1) : 1
62 0, 2 (1) : 1
63 0, 2 (1) : 1
64 0, 2 (1) : 1
65 0, 2 (1) : 1
66 0, 3 (1) : 1
67 0, 4 (1) : 1
68 0, 4 (1) : 1
69 0, 4 (1) : 1
70 0, 4 (1) : 1
71 0, 4 (1) : 1
72 0, 4 (1) : 1
73 0, 4 (1) : 1
74 0, 4 (1) : 1

75 0, 4 (1) : 1
76 0, 4 (1) : 1
77 0, 4 (1) : 1
78 0, 4 (1) : 1
79 0, 4 (1) : 1
80 0, 4 (1) : 1
81 0, 4 (1) : 1
82 0, 4 (1) : 1
83 0, 4 (1) : 1
84 0, 4 (1) : 1
85 0, 4 (1) : 1
86 0, 4 (1) : 1
87 0, 4 (1) : 1
88 0, 4 (1) : 1
89 0, 4 (1) : 1
90 0, 4 (1) : 1
91 0, 4 (1) : 1
92 0, 4 (1) : 1
93 0, 4 (1) : 1
94 0, 4 (1) : 1
95 0, 4 (1) : 1
96 0, 4 (1) : 1
97 0, 4 (1) : 1
98 0, 4 (1) : 1
99 0, 4 (1) : 1
100 0, 4 (1) : 1
101 0, 4 (1) : 1
102 0, 4 (1) : 1
103 0, 4 (1) : 1
104 0, 4 (1) : 1
105 0, 5 (1) : 1
106 0, 5 (1) : 1
107 0, 5 (1) : 1
108 0, 5 (1) : 1
109 0, 5 (1) : 1
110 0, 5 (1) : 1
111 0, 5 (1) : 1
112 0, 5 (1) : 1
113 0, 5 (1) : 1
114 0, 5 (1) : 1
115 0, 5 (1) : 1
116 0, 5 (1) : 1
117 0, 5 (1) : 1
118 0, 5 (1) : 1
119 0, 5 (1) : 1
120 0, 5 (1) : 1
121 0, 5 (1) : 1
122 0, 5 (1) : 1
123 0, 5 (1) : 1
124 0, 5 (1) : 1
125 0, 5 (1) : 1
126 0, 5 (1) : 1
127 0, 5 (1) : 1
128 0, 5 (1) : 1
129 0, 5 (1) : 1
130 0, 5 (1) : 1
131 0, 5 (1) : 1
66 1, 2 (1) : 1
66 2, 4 (1) : 1

Appendix B: Fuzzy Rule Base

% For 3 Error Membership Functions

1. If (ERROR is N) and (CHANGEINERROR is N) then (CONTROL is NL) (1)
2. If (ERROR is P) and (CHANGEINERROR is N) then (CONTROL is PS) (1)
3. If (ERROR is N) and (CHANGEINERROR is P) then (CONTROL is NS) (1)
4. If (ERROR is P) and (CHANGEINERROR is P) then (CONTROL is PL) (1)
5. If (ERROR is Z) then (CONTROL is Z) (1)

% For 131 Error Membership Functions

1. If (ERROR is NG10) then (CONTROL is NL) (1)
2. If (ERROR is NG9) then (CONTROL is NL) (1)
3. If (ERROR is NG8) then (CONTROL is NL) (1)
4. If (ERROR is NG7) then (CONTROL is NL) (1)
5. If (ERROR is NG6) then (CONTROL is NL) (1)
6. If (ERROR is NG5) then (CONTROL is NL) (1)
7. If (ERROR is NG4) then (CONTROL is NL) (1)
8. If (ERROR is NG3) then (CONTROL is NL) (1)
9. If (ERROR is NG2) then (CONTROL is NL) (1)
10. If (ERROR is NG1) then (CONTROL is NL) (1)
11. If (ERROR is NH11) then (CONTROL is NL) (1)
12. If (ERROR is NH10) then (CONTROL is NL) (1)
13. If (ERROR is NH9) then (CONTROL is NL) (1)
14. If (ERROR is NH8) then (CONTROL is NL) (1)
15. If (ERROR is NH7) then (CONTROL is NL) (1)
16. If (ERROR is NH6) then (CONTROL is NL) (1)
17. If (ERROR is NH5) then (CONTROL is NL) (1)
18. If (ERROR is NH4) then (CONTROL is NL) (1)
19. If (ERROR is NH3) then (CONTROL is NL) (1)
20. If (ERROR is NH2) then (CONTROL is NL) (1)
21. If (ERROR is NH1) then (CONTROL is NL) (1)
22. If (ERROR is NB11) then (CONTROL is NL) (1)
23. If (ERROR is NB10) then (CONTROL is NL) (1)
24. If (ERROR is NB9) then (CONTROL is NL) (1)
25. If (ERROR is NB8) then (CONTROL is NL) (1)
26. If (ERROR is NB7) then (CONTROL is NL) (1)
27. If (ERROR is NB6) then (CONTROL is NL) (1)
28. If (ERROR is NB5) then (CONTROL is NS) (1)
29. If (ERROR is NB4) then (CONTROL is NS) (1)

30. If (ERROR is NB3) then (CONTROL is NS) (1)
31. If (ERROR is NB2) then (CONTROL is NS) (1)
32. If (ERROR is NB1) then (CONTROL is NS) (1)
33. If (ERROR is NL11) then (CONTROL is NS) (1)
34. If (ERROR is NL10) then (CONTROL is NS) (1)
35. If (ERROR is NL9) then (CONTROL is NS) (1)
36. If (ERROR is NL8) then (CONTROL is NS) (1)
37. If (ERROR is NL7) then (CONTROL is NS) (1)
38. If (ERROR is NL6) then (CONTROL is NS) (1)
39. If (ERROR is NL5) then (CONTROL is NS) (1)
40. If (ERROR is NL4) then (CONTROL is NS) (1)
41. If (ERROR is NL3) then (CONTROL is NS) (1)
42. If (ERROR is NL2) then (CONTROL is NS) (1)
43. If (ERROR is NL1) then (CONTROL is NS) (1)
44. If (ERROR is NM11) then (CONTROL is NS) (1)
45. If (ERROR is NM10) then (CONTROL is NS) (1)
46. If (ERROR is NM9) then (CONTROL is NS) (1)
47. If (ERROR is NM8) then (CONTROL is NS) (1)
48. If (ERROR is NM7) then (CONTROL is NS) (1)
49. If (ERROR is NM6) then (CONTROL is NS) (1)
50. If (ERROR is NM5) then (CONTROL is NS) (1)
51. If (ERROR is NM4) then (CONTROL is NS) (1)
52. If (ERROR is NM3) then (CONTROL is NS) (1)
53. If (ERROR is NM2) then (CONTROL is NS) (1)
54. If (ERROR is NM1) then (CONTROL is NS) (1)
55. If (ERROR is NS11) then (CONTROL is NS) (1)
56. If (ERROR is NS10) then (CONTROL is NS) (1)
57. If (ERROR is NS9) then (CONTROL is NS) (1)
58. If (ERROR is NS8) then (CONTROL is NS) (1)
59. If (ERROR is NS7) then (CONTROL is NS) (1)
60. If (ERROR is NS6) then (CONTROL is NS) (1)
61. If (ERROR is NS5) then (CONTROL is NS) (1)
62. If (ERROR is NS4) then (CONTROL is NS) (1)
63. If (ERROR is NS3) then (CONTROL is NS) (1)
64. If (ERROR is NS2) then (CONTROL is NS) (1)
65. If (ERROR is NS1) then (CONTROL is NS) (1)
66. If (ERROR is Z) then (CONTROL is Z) (1)
67. If (ERROR is PS1) then (CONTROL is PS) (1)
68. If (ERROR is PS2) then (CONTROL is PS) (1)

69. If (ERROR is PS3) then (CONTROL is PS) (1)
70. If (ERROR is PS4) then (CONTROL is PS) (1)
71. If (ERROR is PS5) then (CONTROL is PS) (1)
72. If (ERROR is PS6) then (CONTROL is PS) (1)
73. If (ERROR is PS7) then (CONTROL is PS) (1)
74. If (ERROR is PS8) then (CONTROL is PS) (1)
75. If (ERROR is PS9) then (CONTROL is PS) (1)
76. If (ERROR is PS10) then (CONTROL is PS) (1)
77. If (ERROR is PS11) then (CONTROL is PS) (1)
78. If (ERROR is PM1) then (CONTROL is PS) (1)
79. If (ERROR is PM2) then (CONTROL is PS) (1)
80. If (ERROR is PM3) then (CONTROL is PS) (1)
81. If (ERROR is PM4) then (CONTROL is PS) (1)
82. If (ERROR is PM5) then (CONTROL is PS) (1)
83. If (ERROR is PM6) then (CONTROL is PS) (1)
84. If (ERROR is PM7) then (CONTROL is PS) (1)
85. If (ERROR is PM8) then (CONTROL is PS) (1)
86. If (ERROR is PM9) then (CONTROL is PS) (1)
87. If (ERROR is PM10) then (CONTROL is PS) (1)
88. If (ERROR is PM11) then (CONTROL is PS) (1)
89. If (ERROR is PL1) then (CONTROL is PS) (1)
90. If (ERROR is PL2) then (CONTROL is PS) (1)
91. If (ERROR is PL3) then (CONTROL is PS) (1)
92. If (ERROR is PL4) then (CONTROL is PS) (1)
93. If (ERROR is PL5) then (CONTROL is PS) (1)
94. If (ERROR is PL6) then (CONTROL is PS) (1)
95. If (ERROR is PL7) then (CONTROL is PS) (1)
96. If (ERROR is PL8) then (CONTROL is PS) (1)
97. If (ERROR is PL9) then (CONTROL is PS) (1)
98. If (ERROR is PL10) then (CONTROL is PS) (1)
99. If (ERROR is PL11) then (CONTROL is PS) (1)
100. If (ERROR is PB1) then (CONTROL is PS) (1)
101. If (ERROR is PB2) then (CONTROL is PS) (1)
102. If (ERROR is PB3) then (CONTROL is PS) (1)
103. If (ERROR is PB4) then (CONTROL is PS) (1)
104. If (ERROR is PB5) then (CONTROL is PS) (1)
105. If (ERROR is PB6) then (CONTROL is PL) (1)
106. If (ERROR is PB7) then (CONTROL is PL) (1)
107. If (ERROR is PB8) then (CONTROL is PL) (1)

108. If (ERROR is PB9) then (CONTROL is PL) (1)
109. If (ERROR is PB10) then (CONTROL is PL) (1)
110. If (ERROR is PB11) then (CONTROL is PL) (1)
111. If (ERROR is PH1) then (CONTROL is PL) (1)
112. If (ERROR is PH2) then (CONTROL is PL) (1)
113. If (ERROR is PH3) then (CONTROL is PL) (1)
114. If (ERROR is PH4) then (CONTROL is PL) (1)
115. If (ERROR is PH5) then (CONTROL is PL) (1)
116. If (ERROR is PH6) then (CONTROL is PL) (1)
117. If (ERROR is PH7) then (CONTROL is PL) (1)
118. If (ERROR is PH8) then (CONTROL is PL) (1)
119. If (ERROR is PH9) then (CONTROL is PL) (1)
120. If (ERROR is PH10) then (CONTROL is PL) (1)
121. If (ERROR is PH11) then (CONTROL is PL) (1)
122. If (ERROR is PG1) then (CONTROL is PL) (1)
123. If (ERROR is PG2) then (CONTROL is PL) (1)
124. If (ERROR is PG3) then (CONTROL is PL) (1)
125. If (ERROR is PG4) then (CONTROL is PL) (1)
126. If (ERROR is PG5) then (CONTROL is PL) (1)
127. If (ERROR is PG6) then (CONTROL is PL) (1)
128. If (ERROR is PG7) then (CONTROL is PL) (1)
129. If (ERROR is PG8) then (CONTROL is PL) (1)
130. If (ERROR is PG9) then (CONTROL is PL) (1)
131. If (ERROR is PG10) then (CONTROL is PL) (1)
132. If (ERROR is Z) and (CHANGEINERROR is N) then (CONTROL is NS) (1)
133. If (ERROR is Z) and (CHANGEINERROR is P) then (CONTROL is PS) (1)