PERFORMANCE ANALYSIS OF MEMBERSHIP FUNCTIONS IN FUZZY LOGIC CONTROL OF DC SERVO MOTORS

DISSERTATION/THESIS

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR
THE AWARD OF THE DEGREE
OF

MASTER OF TECHNOLOGY
IN
CONTROL AND INSTRUMENTATION

Submitted by:

Pradeep Sharma

Roll No. 2K11/C&I/20

Under the supervision of

Dr. Bharat Bhushan

DEPARTMENT OF ELECTRICAL ENGINEERING
DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering)
Bawana Road, Delhi-110042

2017

DEPARTMENT OF ELECTRICAL ENGINEERING

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)
Bawana Road, Delhi-110042

CERTIFICATE

I, **Pradeep Sharma, Roll No. 2K11/C&I/20** student of M.Tech. (Control and Instrumentation), hereby declare that the dissertation/project titled "**Performance Analysis of Membership Functions in Fuzzy Logic Control of DC Servo Motors**" under the supervision of **Dr. Bharat Bhushan** of Electrical Engineering Department, Delhi Technological University in partial fulfillment of the requirement for the award of the degree of Master Technology has not been submitted elsewhere for the award of any Degree.

Place: Delhi

Date: 2017 **PRADEEP SHARMA**

Dr. Bharat Bhushan
Professor
EED, DTU

ACKNOWLEDGEMENT

I Take this opportunity to express my sincere gratitude to all those who have been instrumental in the successful completion of this dissertation.

Dr. Bharat Bhushan, Professor, Dept. of Electrical Engineering, Delhi Technological University, My project guide, has guided me for the successful completion of this dissertation. It is worth mentioning that he always provided the necessary guidance and support. I sincerely tank him for his wholehearted guidance.

I am grateful for the help and cooperation of Prof**. Madhusudan Singh**, Head of the Department of Electrical Engineering, Delhi Technological University, for providing the necessary lab facilities and cooperation and I wish to thank all faculty members whoever helped to finish my project in all aspects.

I would also like to thank my beloved parents, who always give me strong inspirations, moral supports, and helpful suggestions. Without them, my study career would never have begun. It is only because of them; my life has always been full of abundant blessing. To all the named and many unnamed, my sincere thanks. Surely it is Almighty's grace to get things done fruitfully.

Pradeep Sharma
2K11/C&I/20

ABSTRACT

Most of the industrial applications use the traditional PID controllers because they are simple, easy to use, have robust modeling, lower in cost and quite effective for resolving the issues in linear systems but the PID controller is observed not to be suitable for solving the high order issues. Therefore, Fuzzy controllers come to the existence in order to resolve the high order issues in linear systems. This study provides a FLC to control the angular position of the servo motor. The implacability of FLC's to the servo motors produces the quality results. PID controllers are not capable of generating reliable output if the input parameters get changed, but fuzzy logic controller has this ability as it works upon the basis of the set of rules which are defined as per the input membership functions.

The objective of this study is to design and analyze the performance of the difference type of membership functions to control the angular position of the servo motors. Error and velocity is considered as the input membership function and control as an output membership functions in proposed work. The results have been driven by implementing the proposed FLC in MATLAB and depicts that the various types of membership functions have various effects on control of the servo meter. It is also observed that the range of the membership functions also leads to the variations in control output.

INDEX

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1 PREFACE

These days various control system based algorithms have been implemented for control system engineering and position control. In digital servomotor, the position control can be done by using the different kind of automatic process. Conversely, the nonlinearities like load effect which have a negative effect to the performance of the system, did not exist in system.

Electrical motors are used for generalized applications whereas other electrical motors have been developed for application specific tasks. Electric motors can also employ for fulfilling the dynamic need of the systems without increasing the temperature of electric motor. Hence it is an important to determine the load characteristics while electing the electric motor. Other factors such as mission goals, availability of power and cost can also be considered while selecting the electric motor for system. But when implemented practically, it becomes difficult to fetch the full information regarding the uncertainties. Lot of research had been conducted till now in order to perform position control in servo motors. And fuzzy logics are one of the prominent mechanism that is preferred over PID controllers specifically in servo motors for handling the uncertainties [1].

1.2 DC SERVO MOTOR

DC servo motors has various advantages over traditional motors therefore it is widely used for high performance drive systems. A DC servo motor can also be defined as DC motor with low power rating. It can be used as actuators for power systems to drive a load to the system. The DC motor has rapid dynamic respond and high level of initial torque to inertia. DC motors is widely acceptable in industrial applications, robotics, domestic appliances and other applications where the speed and position control is the major task to perform because it has various properties such as leading reliability, low in cost and quite robust.

It can have said to be an actuator in system control which facilitates transitional and rotary movements of the system. If such is done by applying the permanent magnet DC motors then the configuration is; the permanent magnet is used to perform the field excitation and behaves static under every kind of operating states. DC servo motor can perform bidirectional speed management by using the 4 quadrant based operation along with re-created braking facilities.

DC motors that are used in industrial application work on two principles i.e. open loop and closed loop DC servo motor. But the closed loop servo motor is highly preferable over open loop motor. In this system an input is transmitted to the employed error detector and perform a comparison among feedback and error signal, as the result an error signal has been generated which is further used for given as an input to the controller in order to reduce the error of the system. In DC motor the speed and position of servo can have fluctuations by managing the flux, terminal voltage, armature resistance that is implemented to the circuit. There are various methods that can be used for position or speed control but the most common methods are field resistance control, armature voltage control and armature resistance control [2].

1.3 OPEN LOOP AND CLOSED LOOP RESPONSE OF DC SERVO MOTORS

There are mainly two types of response in DC motors. One is closed loop and other is open loop. This section discusses these two in brief.

1.3.1 Open loop use of DC servo motor:

An open loop response of a DC Servo motor comprises of a no feedback module which can be forwarded while motor operations. Thus, open loop DC servo motor is not capable to detect the error in order to perform a performance check. Thus, it is named as open loop DC servo motor because it did not have any feedback reference which comes out from motor. On the other hand, open loop servo motors are not totally open loop in nature because there is an internal feedback system that is deployed on it. The function of this internal feedback system is to generate an output in the form of position feedback to the servo controller in the motor system. The position feedback is generated by the potentiometer which is linked to the servo gear system. After then, the reading of potentiometer is converted to the values which are directly proportional to the location of the servo shaft. The internal feedback system is the pitfall of open loop DC servo motor because it generates feedback only to the internal servo control. The figure .11 represents the block diagram of open loop servo motor with internal feedback system.

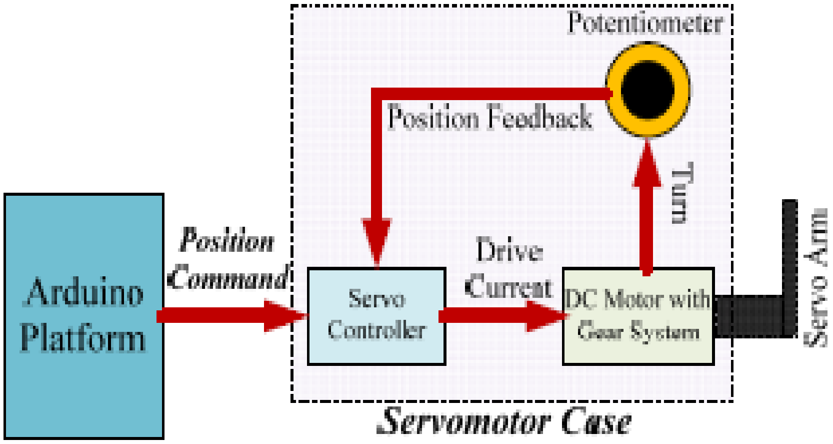

Figure 1.1: Block diagram of open loop servo motor with "internal feedback" mechanism [14].

1.3.2 Closed-loop use of DC servo motor

The closed loop DC servomotor operates opposite to the open loop servo motor. The closed loop servo motor comprises of a feedback reference which is done in order to evaluate that whether the system is performing as per expectations or not. It is feedback reference which serves as the advantage for the closed loop servo motor, as the main controller have all the status and track of errors. It is notified that there are various kinds of models of closed loop DC motor available in the market in the form of potentiometer to support voltage, encoder to support pulses etc. The interesting point is that these feedback systems of closed loop DC servo also works as the internal feedback systems but the difference is that it is directly proportion to the motor shaft. The figure 1.2 represents the block diagram of closed loop DC servo motor [3].

As a matter of fact, these feedbacks work similarly to the "internal" feedback where it is proportional to the angular position of the motor shaft. This paper focuses on the use of DC servo motor with voltage position feedback as shown in Figure-1.2.

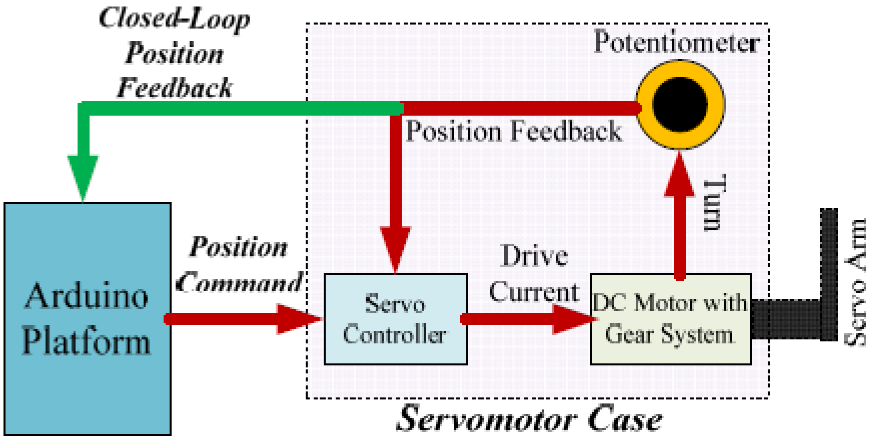

Figure 1.2: Block diagram of closed loop DC servo motor [14].

1.4 APPLICATIONS OF SERVO MOTOR

DC servo motors become popular among various fields where the starting and stopping functions are needed to be performed quick and rapidly such as robotics control, machinery control, automation etc. Such kind of applications need a high-speed control reliability and good response. In robotics, the servo motor is used for controlling the motion of robotic arms as per the relevancy of position. The rotor in the servo motor is made up of special lightweight material and used for reducing the inertia of amateur and to develop the required magnetic flux. The inertia with low rotor is used for enhancing the caliber of immediate start and stop operation during on-off of the system. Servo motor is quite expensive thus; it is not suitable for small scale applications.

DC servo motor is a device which is specifically used for converting the electrical energy to mechanical energy. The movement of output shaft to an angular position is one advantage of DC servo motor. The small motors that are particularly used in laboratory purpose can be defined as a electromechanical device which is used for determining the position of the motor amateur which is generated from an outer DC voltage which generates the motor torque and speed.

The good control response and wider speed control range can be obtained by using the DC motor [4]. Therefore, it is also applicable to the high control equipment such as rolling mill, double hulled tankers, high precision digital tools. Mainly, the hardware of the digital servo motor comprises of DC brush type servo motor, motor shaft encoder, belt drives for the movement of the platform, motor shaft encoder and rail encoder, friction break, flywheel and pulleys. It also comprises of magnetic field and electric field that intercommunicate with each other in order to produce a mechanical force. An error sensing feedback reference is used for correcting the performance of the system. The major objective of error sensing feedback reference is to manage the mechanical position and speed of the other parameters. The image in figure 1.3 depicts the DC servo motor which is connected to an encoder to supply the position and speed feedback. In the simplest application it will only measure the position which is further compared to the command position. In case if there is a difference between output position and required signals then an error signal is generated correspondingly. Thus, the motor shaft rotates in either direction and then brings the output shaft to the exact position. If the approached position is reached then the error signals is reduced to zero and the motor stops working.

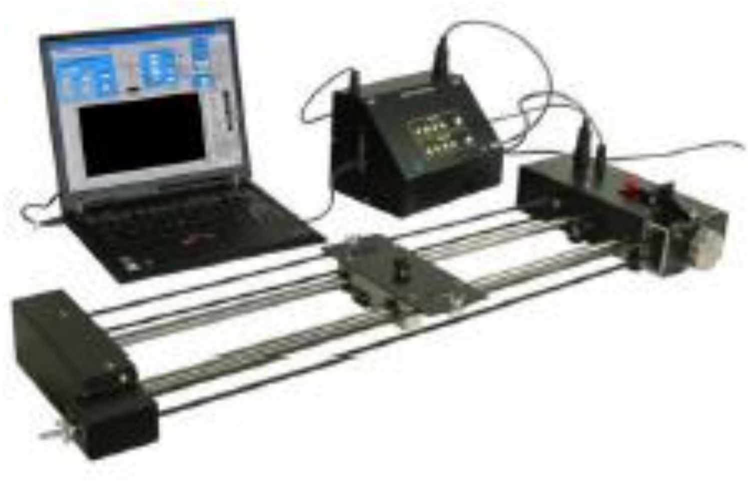

Figure 1.3: Digital DC servo Motor [19]

A DC motor acts as a common actuator in control systems. It directly leads to the rotary motion and connectes with the wheels or cables and drums so that it can generate the transitional motion [5].

1.5 Overview of positional rotation servo motor

Positional rotation servo motor can be further classified into two types that are as follow: first one is DC servo motor and the second one is AC servo motor. DC power is required to control the DC servo motor. On the other hand AC servo motor is controlled by AC power signal. Types of feedbacks used in Servo motors are Encoder or Voltage feedback. The encoder installed in the servo motor generates the pulse shaped signal, which will be further processed by the Servo controller to produce the positional feedback. On the other hand, in voltage feedback system the potentiometer is associated and which is further linked with servo motor axis in order to obtain the relative angular position of motor. The relative value of angular position of motor is derived from value of potentiometer. Usually, DC servo meter finds the application in the field of robotics, automation system and electronics. These motors are specially used where precise angular movement is required like robotic arm, gripper, etc. In the ordinary DC motor only two voltage control signals are there: OFF and ON [18]. Pulsed shape signal that is directly proportional to the rotation angle of motor is used to control the movement of Servo motor. At present various types of servo motor can be obtained in the market. The servo motor is also associated with the physical stop mechanism in order to stop the rotation within the limits and also helps in protecting the rotational sensor. The positional rotation servo motor system can be classified in two categories that are as follow: one is closed-loop and other one is open loop.

1.6 Modeling

This section of the work is organized to represent a generalized mathematical model for motor. For this purpose the ironless rotor DC servo motor has been used and rated the torque and speed are 1.5 N.cm and 2500 rpm.

1.6.1 Identification of DC motor for speed control

The electric part is avoided in order to design a simple and less complex model. The identification process for LTI model i.e. linear time invariant is elected in order to evaluate the grey model in discrete format. The following subspace identification method is used for evaluating the A,B,C and D state space model.

$$x(k+1) = Ax(k) + Bu(k) \ldots \ldots \ldots (1.12)$$

$$y(k) = Cx(k) + Du(k) \ldots \ldots \ldots . (1.13)$$

In above equation u(k) represents the system input and output respectively.

x(k) denotes a atate vector coresponding to the system at time interval of k.The multi sine signal data is used for the purpose of identification with 6000 samples , 1 Hz frequency range and 0.01

sampling time [9]. The open loop DC servo motor has been used for simulation. The utlized discrete identificationmodel is represneted in section below:

$$x(k+1) = 0.9605x(k) + 0.0001014u(k) \dots \dots (1.14)$$

$$y(k) = 340.3x(k)$$

The BFR (Best Fit Rate) is a term which is used measured to evaluate the outcome of the model.

$$BFR\% = 100\%.\max\left(1 - \frac{\|y-\hat{y}\|_2}{\|y-\overline{y}\|_2}, 0\right) \dots \dots .. (1.15)$$

Here in BFR,$\overline{y}$ refeers to the y and $\|.\|$ representes the 2nd normalizationof a sequence.

1.6.2 Identification of DC motor for position control

This section shows the representation of DC motor for position control by using 3rd order LTI system. Whereas it can also be represented by using the combination of second order and first order systems with integrator. It can be achieved by using the dynamics of electrical parts which are quite suitable in comparison to the mechanical parts. The following formulation represents the discrete time model for DC servo motor.

$$G(z) = \frac{a}{z+b} . \frac{T}{z-1} \dots \dots \dots .. (1.16)$$

In this equation, T_s shows the sampling time of the signals which is considered as 0.01 sec, a and b are constant parameters [10]. A closed loop based grey box modeling can be used whose configuration (figure 1.4) is also given in this section.

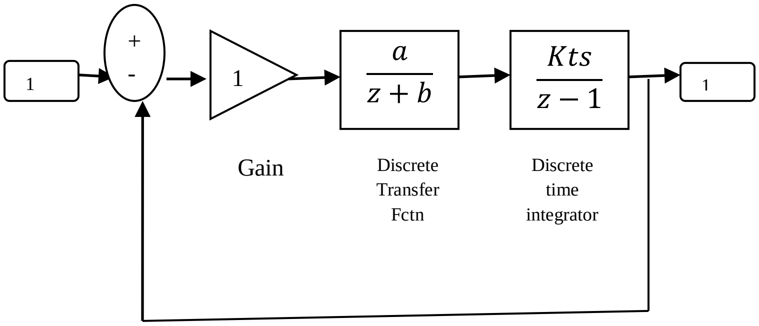

Figure 1.4: Block Diagram for identification of position control model [15]

1.7 PID CONTROLLERS

PID controllers are known as Proportional-Integral-Derivative controllers. There are mainly three control parameters that need to be adjusting in obtaining an output. Output i.e. obtained is the combination of Proportional, Integral and Derivative parameters. The main areas of using this controller are the industry as well as education.

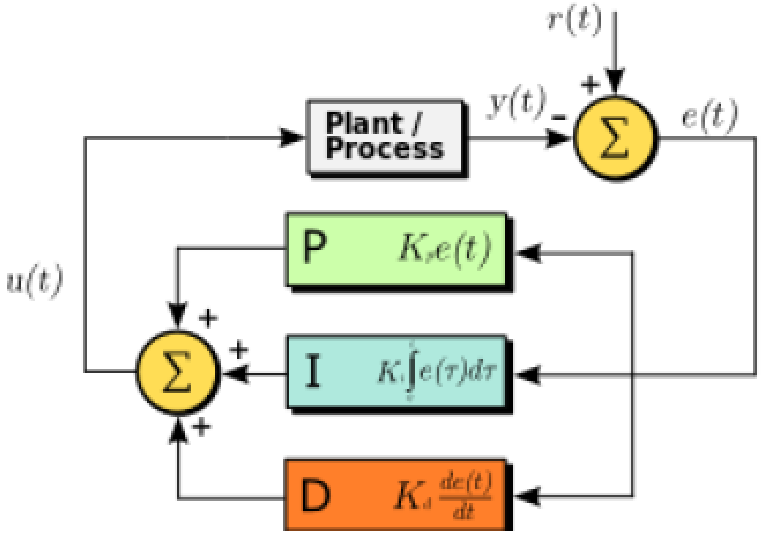

Figure 1.5: Conventional PID Controller system design [19]

Figure 1.5 shows a block diagram for a PID controller. PID controllers are used in almost every area but these two fields are basic building as most of the controllers are used in these areas. PID controllers must be tuned to provide the installation of the controllers. Thus tuning is performed on the dynamic behavior of the system [19].

In the above figure three control parameters are used to obtain the output. The first parameter is Proportional control which increases gain. The second parameter is integral control which reduces steady state error. And the last one is Derivative control which improves transient response [11]. Input is taken from the world and produces output to the world.

PID controller as an equation:-.

$$u\ (t)= K_p[e(t) + 1/T_i\int_0^t e(t')dt` + T_d(de(t)/dt)] +b\ldots\ldots\ldots\ldots\ldots\ldots..(1.17)$$

Where, u defines the control signal, e shows the difference between the current value and the set point, k_p is the gain for a proportional controller, t_i integral controller scales through this parameter, t_d derivative controller can be scale through this parameter, t defined the total time taken by the error measurement, b defines the set point of the given or taken signal which is also considered as bias or offset value.

Collaborated with three parameters given, PID can be referred as transfer function in MATLAB defined below:

Transfer function of the controller can be expressed as:

$$C(s) = K_p(1 + \tau_d s + 1/\tau_i s)\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots..(1.18)$$

Where K_p shows proportional gain, τ_d describes Derivative time and τ_i defines integral time.

Proportional term of the controller defines proportional to the generated error given as:

$$P_{out} = K_p\ e\ (t)\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots(1.19)$$

Proportional control value depends upon the error rate which means that higher the error higher the proportional control explained in the equation above. Thus it can be concluded that proportional control brings the system into fast set point [12]. On the other hand, it produces steady state error which leads to an overshoot when the system gets to the set point. There is another possibility of increasing the proportional gain but that can lead to a system unstable.

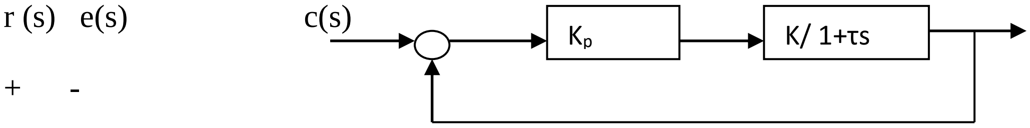

Figure 1.6 : Proportional control

Proportional control as shown in figure 1.6, can expressed as closed loop transfer function like:

$$\frac{c(s)}{r(s)} = \frac{\frac{KK_p}{1+\tau s}}{1+\frac{KK_p}{1+\tau s}} = \frac{KK_p}{1+KK_p+\tau s} = \frac{KK_p}{1+KK_p}\frac{1}{1+\tau' s}\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots(1.20)$$

Where $\tau' = \frac{\tau}{1+KK_p}$

Step input $r(s) = \frac{A}{s}$

$$c(s) = \frac{KK_p}{1+KK_p}\frac{A}{s(1+\tau' s)} \text{ or } c(t)= \frac{AKK_p}{1+KK_p}(1-e^{-st/\tau')} \ldots\ldots\ldots..(1.21)$$

From above given equation (1.21), it is declared that:

(1) Factor $\frac{1}{1+KK_p}$ can be used for improving time which means that time constant can be decreased by this.

(2) Difference between desired response and output response steady state offset is

$$A\ (1-\frac{KK_p}{1+KK_p}) = \frac{A}{1+KK_p} \ldots\ldots\ldots\ldots..(1.22)$$

Integral part of the controller influence with the variations of the error on time given as:

$$I_{out}= K_i \int_0^t e(T)\,d\text{T}\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots.(4)$$

It helps in eliminating the problem of proportional term which was steady state error but it has a disadvantage too i.e. it affects the stability of the system. Thus it can be concluded that integral term depends on pass values of the error [13].

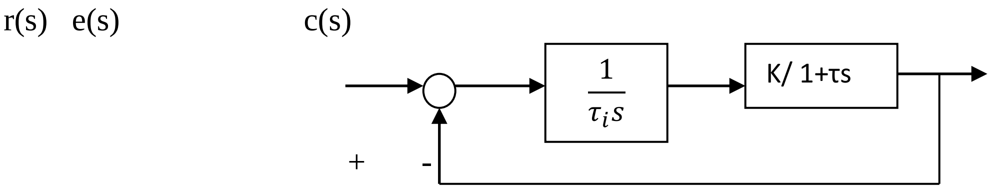

Figure 1.7: Integral control action

Above figure 1.7 obtain following equation as:

$$\frac{c(s)}{r(s)} = \frac{\frac{KK_p}{\tau_i s(1+\tau s)}}{1+\frac{K}{\tau_i s(1+\tau s)}} = \frac{K}{K+\tau_i s+\tau\tau_i s^2} \ldots\ldots\ldots.(1.23)$$

Where step input r(s) $=\frac{A}{S}$,

$$e(s) = \frac{1}{1+\frac{k}{\tau_i(1+\tau s)}}\frac{A}{S} = \frac{\tau_i s(1+\tau s)}{\tau_i s(1+\tau s)+K}\frac{A}{S} \ldots\ldots\ldots..(1.24)$$

$$e_{ss} = \overset{Lt}{s\rightarrow 0}\ se(s) = 0\ldots\ldots\ldots1.25$$

Unlike other terms **Derivative control** is relative to the rate of changes of the error defined as per the equation below:

$$D_{out}= K_d d/dt\ e\ (t)\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ (1.26)$$

The above equation helps in estimating the future error which leads to increase or decrease the speed of correction. This term helps in taking decision soon which provides detection of any changes on the error and system remains stable. This term is very sensitive to disturbances. If there is no change in the error then derivative influence will not be generated.

Transfer function of PD controller expressed as:

$$C(s) = K_p\ (1+\tau_d s)\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots(1.27)$$

Derivative controller cannot be used alone due to its some drawbacks thus Proportional Derivative controller has been used to provide stability to the closed loop system. To prove this fact process transfer function can be explained as [14]

$P(s)= \frac{1}{js^2}$

Therefore resultant closed loop transfer function is:

$$\frac{c(s)}{r(s)}= \frac{\frac{K_p(1+\tau_d s)}{js^2}}{1+\frac{K_p(1+\tau_d s)}{js^2}} = \frac{K_p(1+\tau_d s)}{js^2+K_p(1+\tau_d s)} \ldots\ldots..(1.28)$$

Where $js^2 + K_p\tau_d s + K_p=0$ represents characteristics equation which will provide stable closed loop response.

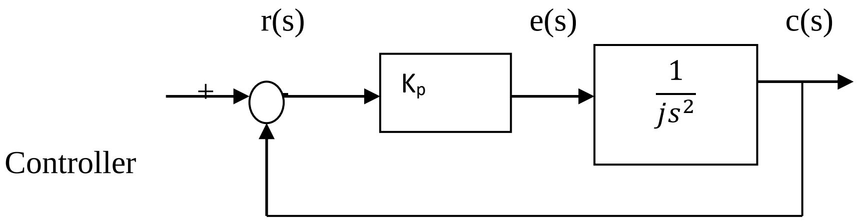

Figure 1.8: Control action with a higher order process [20]

With the use of P and P-D controller step response will be compared represent in figure 1.8.

In this research paper, fuzzy logic controller is used to control a plant with the help of human knowledge with linguistic variables. The main advantages of using this system are good popularization as well as high fault tolerance. It can also apply to non linear systems due to which it is famous for years [20].

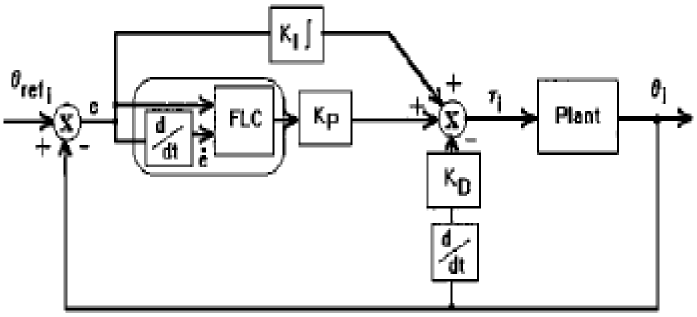

Figure 1.9: Fuzzy based PID controller system design

Performance of the controller can be enhanced by the hybrid fuzzy controller i.e. combination of P and ID controller shown in figure 1.9 where proportional term is replaced with incremental FL controller and integral and derivative term will remain same [15].

$$\Delta Ti\ (Ki)= KPi\Delta ui(k) + KIiTei(k)- KDi\frac{\Theta\mathbf{i}\ (\mathbf{K})-\mathbf{2}\boldsymbol{\theta}\mathbf{i}(\mathbf{k}-\mathbf{1})+\boldsymbol{\theta}\mathbf{i}(\mathbf{k}-\mathbf{2}}{\mathbf{T}}\ldots\ldots..(1.29)$$

From above equation,Δu_i (k) shows output of incremental FL controller.

CHAPTER 2

LITERATURE REVIEW

Nader Jamali et al in[1]represented a mathematical formulation based method which worked upon the basis of the variations in parameters of PD controllers particularly for fuzzy based servo motor. The operations performed by the proposed technique utilized the approximation of nonlinear fuzzy function which was comprised of error and error derivation. After then the rule of fuzzy control had been applied to techniques which had been applied to the microcontroller such as AVR for controlling the angular position of angular motor. This study had also analyzed the performance and robustness of the enhanced fuzzy controller with respect to the PID controllers in terms of no load, outer noise corresponding to various magnitudes levels. The simulation results had been obtained and had been proved that the proposed fuzzy based controller had been better than the traditional in terms of rise time, overshoot, settling time etc specifically in case of saturation and friction. **In [2],Thwin Thu Lynna**developed a fuzzy control based servo motor for controlling the position in flat bed printing machine. It was depicted in this work that the DC servo motors are highly preferred because it provides high rating and speed of the motor. The proposed system works by using the closed loop servo motor and perform position control via position sensor which is connected with motor shaft in order to generate the feedback reference. The simulation had been done in order to evaluate the performance of the proposed fuzzy logic based DC servo motor drive and illustrates that it provides a dynamicity in performance which makes it more suitable for industrial applications. **Meghana Singh in [3],** presented a review to the various types of controllers that can be used for compensating the various issues in network control communication systems. In network communication, the data travels from one node to another node in order to reach the destination. While covering such a long route, sometimes data packets suffers from various loses such as delay in communication, misplacements of data packets etc. Network control system is geographically dispersed network in which the actuators and controllers are deployed and shared by the network. **Xudong Liu et al in [4]** addressed the issue of time delay in closed loop servo motor which is managed via network. A smith predictor based network servo system had been developed by the author in this study. In this work a control strategy had been developed by combining the adaptive least mean square and smith compensator to compensate the network delay to a certain level. The simulation of the proposed work had been done by implementing it to the real world applications. After observing the results it was shown that the smith predictor has reduced time delay in comparison to the PID controller. Hence it was proved that the proposed method was more reliable, robust and self-adaptive. Further smith predictor had been used by **Honge Ren in [5],**developed a robust fixed order H_∞ control design in order to overcome the uncertainties such as time delay in system. The smith predictor structure that was utilized by the author depicted that the H_∞ robust performance can be obtained by using the convex constraints as per the parameters of the linearly parameterized model. The proposed work considered the model with the tentative dead time, frequency domain and multi-modularity etc. The working of the proposed model was also capable to extend in order to use it as a gain-scheduled dead time compensator. **Cui Hao et al in [6]** adopted the fuzzy logics and smith predictor and developed a single mechanism which was named as fuzzy adaptive smith predictive control system to compensate the problem of three tank system. The PID controller was used for setting the PID parameters in order to enhance the resistance capability of random disturbance and smith control so that the time delay can be reduced. **Amirul Syafiq et al. in [14]** represented that the angular position of DC servo motor can be controlled by using the controller to manage the specific angle. It was observed that the Arduino controller was quite popular in the market on those days specifically for closed loop servo motors. The various methods for controlling the servo motor by using arduino microcontroller had been represented in this study. These methods were IDE (Arduino Integrated Drive Electronics), Support Package (Support target for simulink), IO package (Input\output package). A comparative analysis had been provided in this work over these three position control methods in servo motors. **Mohamed A. Darwish in [15]** this study depicted the outline and test execution of a discrete-time settled request H∞ controller for a DC engine speed and position control. To gave a model to the DC engine, twosystem identification methods were utilized. In the first a model for DC engine speed control was recognized in open-circle in light of discovery demonstrating though in the other one a model for position control was distinguished in shut circle in light of dim box displaying. An expansion of HIFOO tool stash to discrete-time controller configuration grew and was utilized to synthesize the controller. The execution of the outlined controller in correlation with different control procedures was illustrated. The paper goes for showing straightforward displaying and control combination strategies with the assistance of accessible programming devices to configuration low-unpredictability controllers as far as outline and usage. Thus, modest equipment can be used for a few applications. **Pradnya Pathade et al. in [16]** The aim of this study was to draw a contrast among conventional controller and fuzzy logics based controllers in terms of time specification for DC servo motors. It is comprised of design and development of a controller for managing the angular position of the by using MATLAB as a simulink platform. The scope of his study was to apply fuzzy and PID controllers collectively in order to develop a model for controlling the angular position of DC servo motor. **Y.Y. Yao in [17]** reviewed and compared the concepts of fuzzy sets and rough sets. The formulation of fuzzy sets had been reviewed on the basis of the two various approaches. The rough sets was represented with respect to the two different aspects such as set oriented view and operator oriented view. It was observed that the rough sets which falls under the category of set oriented views were closely related to the fuzzy sets. Both of the sets are capable to consider for classical algebra whereas the rough sets related to the operator oriented category were quite different from fuzzy sets.

CHAPTER 3

PROBLEM FORMULATION

3.1 PROBLEM STATEMENT

In this Era of advanced technology, the use of controlling process in industries and engineering field has been increased rapidly. Various energy converters are used and electric motors are one of the most prominent energy converter preferred for machine tools and robotics. These kinds of electric motor are required to implement an automatic control to its parameters such as position control, speed management, acceleration control and current control. As the technology is getting advanced day by day and giving birth to the new applications that require position control in order to perform in an efficient way. Various kinds of methods are available in the market for controlling the angular position of the electrical motors. In this study, we have considered the servo motor because it is in high demand nowadays. In addition to various control mechanisms there is a requirement to design an effective model for servo motor. On the basis of the review that has been conducted in this study, it is concluded that various types of controllers have been used by various authors in previous work in order to control the angular position of the servo motor but some of them proved advantageous and some of them had no positive effect on the performance of the system. Most of the work had been conducted by using the PID controllers and fuzzy logic controllers. But it was observed that the singular position of the motor had been controlled to a specific angle only. Therefore, there is a requirement to develop an enhanced controller to control the angular position of the servo motors.

3.2 OBJECTIVES

1. To implement the Fuzzy logic based controller to control the angular position of servo motor.
2. To perform the analysis to the various fuzzy membership functions.
3. To implement the Fuzzy Analysis using MATLAB software.

3.3 PHYSICAL STREAM

Let us consider a DC servo motor to study the above objectives whose electric circuit of the amateur and free-body diagram of the rotor shown in figure 3.1.

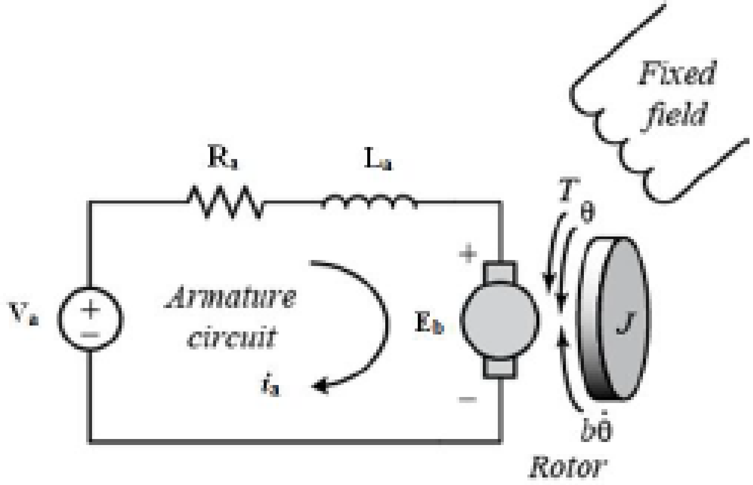

Figure 3.1: Schematic representation of DC motor

In above figure of DC motor, the input is considered as the amateur voltage which is depicted by V in volts. The evaluation variables are the angular velocity of the shaft which is depicted by ω in radian per second and shaft θ in radian.

Here:

V_a=armature voltage (V)

R_a = armature resistance (Ω)

L_a= armature inductance (H)

i_a = armature current (A)

E_b= back emf (V)

T = Torque (Nm)

θ = angular position of rotor shaft (rad)

3.4 SYSTEM EQUATION

In armature control of separately excited DC motor, the voltage applied to the armature can be molded without molding the voltage of the field where the output voltage and motor [6] torque is related to the given formulation:

$$V_a(t) = R_a i_a(t) + L_a \frac{di_a(t)}{dt} + Eb(t) \ldots \ldots .(3.1)$$

Here the motor torque t is connected to the armature current which is depicted with i by constant factor K:

$$T = Ki \dots \dots \dots . (3.2)$$

The back electromotive force is connected to the angular velocity by:

$$E_b = K\omega = K\frac{d\theta}{dt} \dots \dots \dots . . (3.3)$$

On the basis of the figure 3.1, Newton's law and Kirchhoff's law can be combined to generate the equation as below:

$$j\frac{d^2\theta}{dt^2} + b\frac{d\theta}{dt} = ki \dots \dots \dots \dots \dots \dots (3.4)$$

$$L\frac{di}{dt} + Ri = V - K\frac{d\theta}{dt} \dots \dots \dots \dots . . (3.5)$$

3.4.1 Transfer Function

The transfer equation of servo motor can be explained by using the Laplace function in the basis of the 3.4 and 3.5 equations:

$$js^2\theta(s) + bs\theta(s) = KI(s) \dots \dots \dots . (3.6)$$

$$LsI(s) + RI(s) = V(s) - Ks\theta(s) \dots \dots \dots (3.7)$$

The Laplace operator from 3.7 equations can be defined as below:

$$I(s) = \frac{V(s) - Ks\theta(s)}{R + Ls} \dots \dots \dots . (3.8)$$

And substitute 3.8 to 3.6

$$js^2\theta(s) + bs\theta(s) = K\frac{V(s) - Ks\theta(s)}{R + Ls} \dots \dots . (3.9)$$

The equations of DC motor can be represented in the form of block diagram as follows:

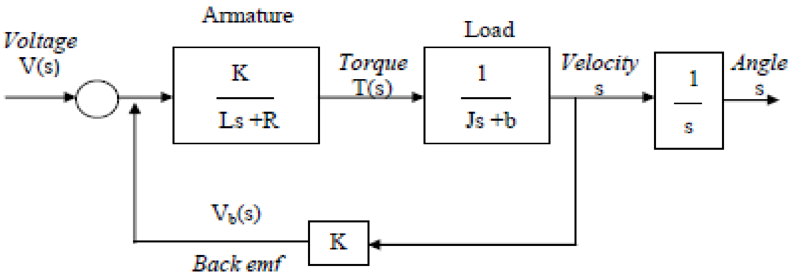

Figure 3.2 : Block Diagram of Servo Motor on the basis of the transfer function [19]

On the basis of the equation 3.9, the transfer function of input voltage , V(s) to the θ:

$$G_a(s) = \frac{\theta(s)}{V(s)} = \frac{K}{s[(R+Ls)(js+b)+K^2)]} \dots \dots . . (3.10)$$

On the basis of the block diagram in figure 3.2, the equation for the input voltage (V) to the angular velocity can be defined as below:

$$G_a(s) = \frac{\omega(s)}{V(s)} = \frac{K}{s[(R+Ls)(js+b)+K^2)]} \dots \dots . . (3.11)$$

3.5 PRINCIPLE OPERATION OF DC SERVO MOTOR CONTROLS

The operations in servo motors are controlled by a pulse which is of variable length. The input signal to this pulse is categorized on the basis of the various characters such as minimum pulse, maximum pulse, highest and repetition rate and shown in figure 3.3 below [7]:

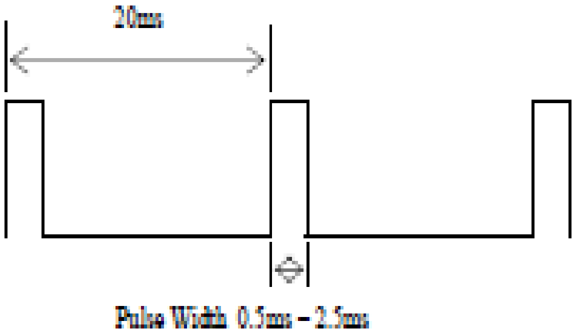

Figure 3.3 Input Pulse of servo motor [18].

According to the rotation constraint in servo motor, the neutral stage depicts the level or position in which the performance of the servomotor is similar to the potential rotation in clockwise direction. The angle is evaluated on the basis of time interval implemented to the pulse to the signal (Pulse Coded Modulation). The pulse is detected after a fixed time interval i.e. 20 ms. The operating lifetime of the motor is determined on the basis of the pulse length. The figure 3.4 represents the relationship of pulse and direction of servo motor.

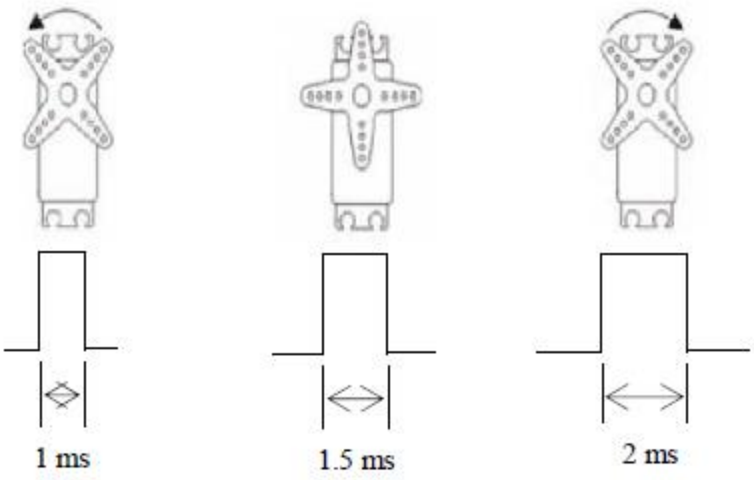

Figure 3.4 : Relationship between Pulse and direction of Servo motor [18]

The Pulse Coded Modulation is a mechanism which is used for managing the power to the electric devices. The current in load is managed by turning resetting the switches among supply and load. More and more power can supply to the load by simply setting the switch to ON status for long time. The modulation generators and sinusoidal signals are derived by the AC. The signals depicted as triangular shape in figure 3.4 represents the carrier and switching frequency corresponding to the inverter. The sine wave is generated by the modulation generator in order to establish the pulse width. In case of DC motor, the pulse coded modulation is generated on the basis of the comparison that is driven between triangular signals and signals of DC servo motor as represented by figure 3.5. The range of the DC signals lies between minimum and maximum voltage of triangle waves. The measurement of the modulated signals is done by using the percentage of duty cycle [8].

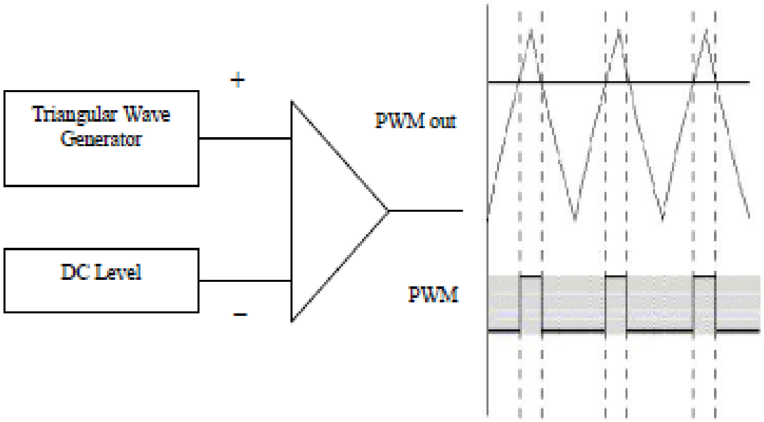

Figure 3.5 : Generation of PWN and DC Drives[18]

3.6 MATHEMATICAL MODEL

The circuit in figure 3.6 in this section depicts the mathematical model of DC servo motor. The electric portion shown by the armature and mechanical portion by T and J in the circuit. The field of excitation is static , the armature controller totally relies upon armature voltage.

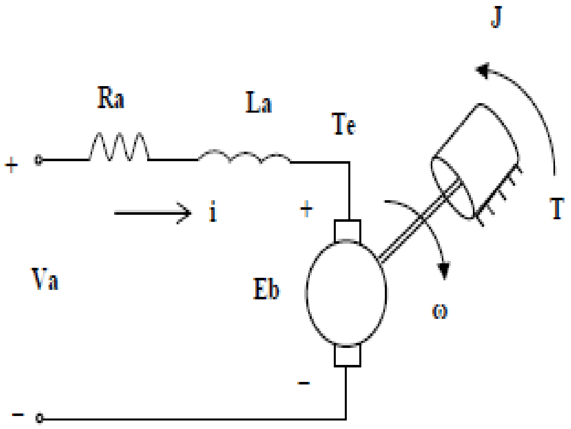

Figure 3.6 : Mathematical model of Servo Motor[18]

The mathematical model on the basis of the circuit below can be defined as below:

$$V_a = R_a i + L_a \frac{di}{dt} + E_b \dots\dots..(3.12)$$

$$j\frac{d\omega}{dt} = T_e - T \dots\dots\dots\dots.(3.13)$$

In equation 1.12, $E_b = d\omega, T_e(electromagnetic\ Torque) = Ki, i = armature\ current,\ V_a = armature\ voltage, R_a = armature\ resistance, L_a = armature\ inductation$ and K variable is used for depicting the torque and electromagnetic constant, ω stands for rotor angular, angular speed of the rotor is defined by using the ω,

CHAPTER 4

SPEED CONTOL OF DC MOTOR USING FUZZY LOGIC

4.1 PROPOSED WORK

As defined in above section that in order to control the angular position of the servo motor, various authors conducted studies by using different type of controllers such as PID controllers, arduino controllers and fuzzy logic controllers. The fuzzy logic controllers are evaluated as the most suitable controller to the servo motors as this is the less complex mechanism and did not include any of the mathematical models. The purpose of this study is to control the angular position of the servo motor by using FLC (Fuzzy Logic Controllers) on MATLAB application. The proposed work implements the position control algorithm with respect to various type of membership function. The reason behind choosing the fuzzy logics is that its design works on the basis of the knowledge driven rules and another advantage of the FLC is that it has a simple design topology. FLC is able to work in a human-like manner and also have characteristic of converting the linguistics based rules to control rules.

4.2 FUZZY LOGICS

Fuzzy is applied in various fields such as engineering, studies, medical etc., in order to derive a decision. Fuzzy system is easy as well as simple to understand and implements. Fuzzy system is a logical system which is in the form of many-valued logic. The truth table of these values lies between the range of 0 and 1, since Boolean logic supports the 0 and 1 only and considers the result either 0 or 1. It also supports the elements which are surrounded by the set may either have partial degree of membership means either element belongs to a set or not. These degrees are managed by any particular functions when applied with the linguistic variables. Fuzzy use linguistic variables in addition to quantitative variables in order to present vague concept. Membership function defines mapping of a membership value between 0 and 1 in the given input space. Universe of discourse is another term used for input space.

Following figure 4.1 explains the working process of fuzzy system in brief. Firstly, a crisp value is added to the fuzzy system as an input. Then Fuzzification process is applied to the crisp fuzzy values. Fuzzification is a process which converts the crisp values into fuzzy sets.

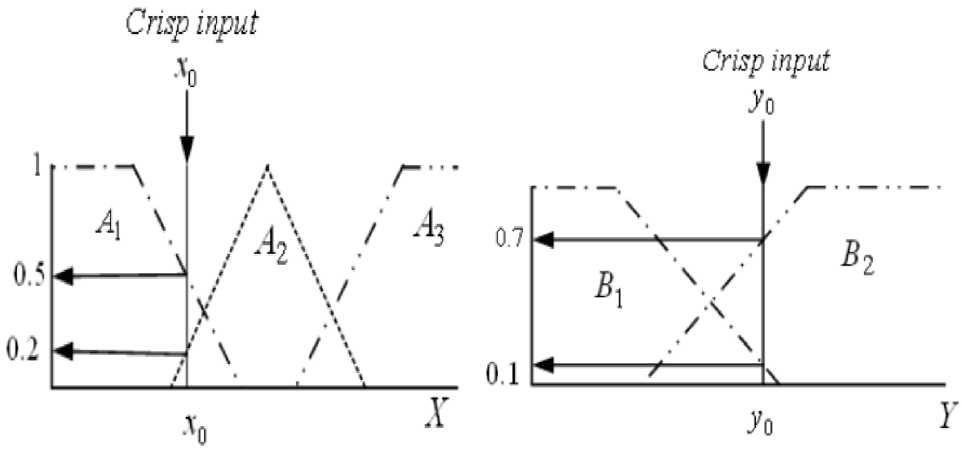

Figure 4.1: Fuzzification

Then defined rules are applied to the fuzzy input set driven by applying fuzzification. On the basis of rules an intelligent decision is taken and then the fuzzy sets are converted to the crisp values back by applying the Deffuzification as shown below in figure 4.2.

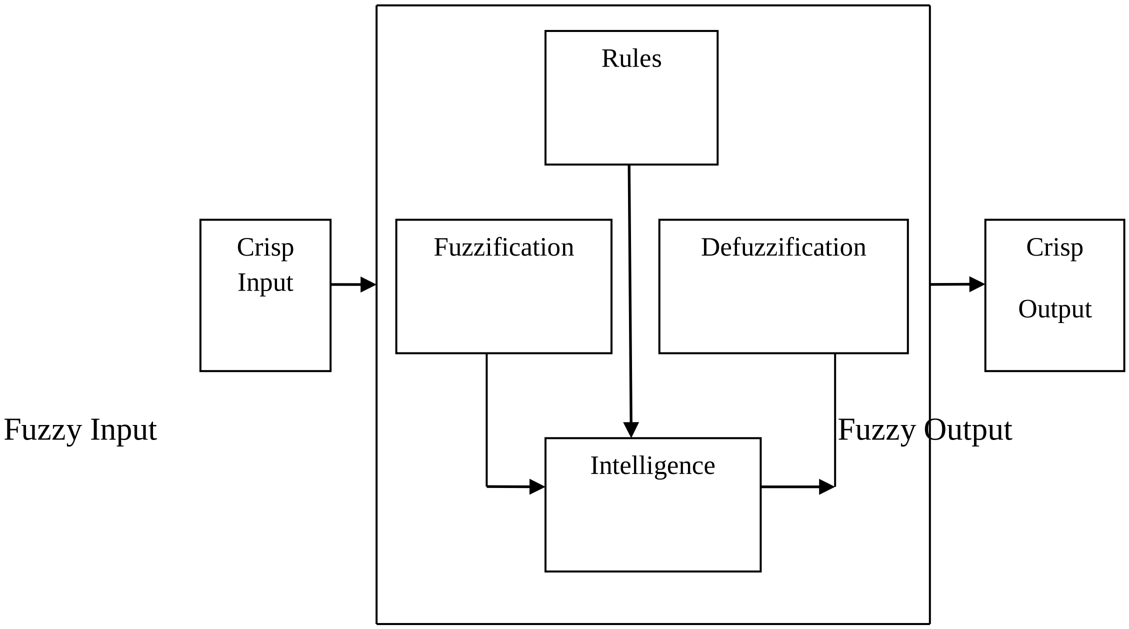

Figure 4.2: Working of Fuzzy Logic Based System

4.2.1 FUZZY RULES

Fuzzy logics have dual core nature on one hand it act as a rule based system and on other side it is also a non linear mapping. In order to understand the transformation of IF-THEN rules into non linear mapping let's consider the example that we have a collection of IF-THEN rules:

$$R^{(1)}: IF x_1 is F_1^l and x_n is F_n^l THEN y is G^l \ldots \ldots \ldots 4.1$$

In equation (1) $x = (x_1, \ldots., \ldots, x_n) \in R^n and y \in R$ belongs to the input and output of the fuzzy logics.

To understand the rule formation in fuzzy let's consider an Example of AND gate. In AND gate the input is always taken in the form of 0 or 1. The AND gate generates High output only if both of the inputs are high. The Truth table of AND logic gate is as follow (table1):

Table 4.1. Truth Table of AND Logic Gate

Input 1 (Inp1)	**Input 2 (Inp2)**	**Output**
0	0	0
0	1	0
1	0	0
1	1	1

Hence the IF-THEN rules in Fuzzy system for AND gate will be generated as follows:

$$IF Inp1\ is Low and Inp2\ is Low$$
$$THEN Output is Low$$

According to this rule, if Inp1 and Inp2 are 0 then the Output will be 0.

$$IF Inp1\ is Low and Inp2\ is High$$
$$THEN Output is Low$$

The above rule defines that if Inp1 is Low and Inp2 is high then the Output will be Low.

$$IF Inp1\ is High and Inp2\ is Low$$

$$THENOutputisLow$$

Accordingly IF Inp1 is high and Inp2 is low then the output in this situation will be low.

$$IFInp1\ isHighandInp2\ isHigh$$

$$THENOutputisHigh$$

This rule defines that IF Inp1 and Inp2 are high then the Output will be high.

The above example defines that the output of fizzy system depends upon the input parameters and rules that are defined by considering these input parameters.

Systems used in Fuzzy logics are as follows:

1. Sugeno
2. Mamdani

a) Sugeno:

Sugeno is a fuzzy model. It works as follows:

1. Let's consider that fuzzy inference two inputs x and y respectively and generate corresponding single output z.
2. A first-order Sugeno model will have following rules as shown in figure 4.3 below : -

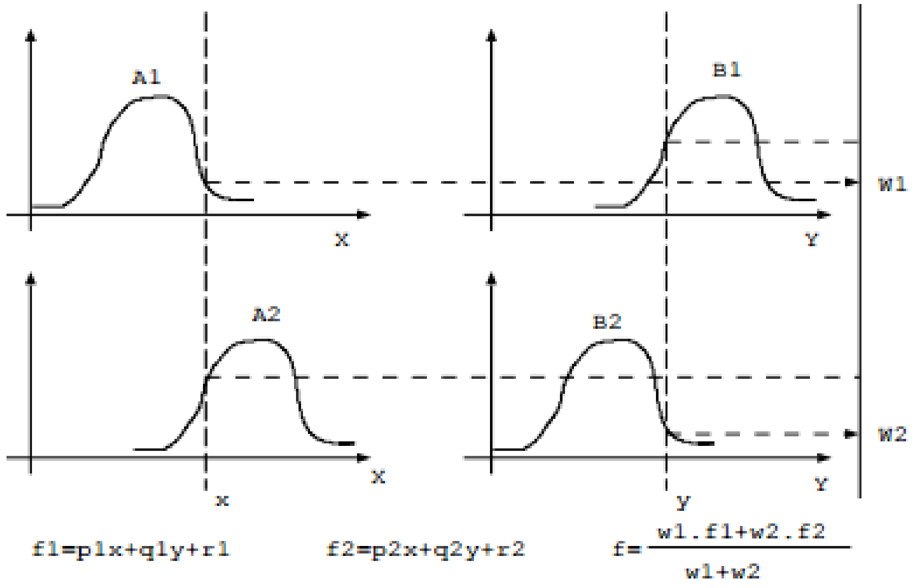

Figure 4.3: Graphical representation of Sugeno model

b) Mamdani:

It is the most commonly used fuzzy interference technique developed by Mamdani and Assilian hence named as Mamdani. In this model, as shown graphically in figure 4.4, the fuzzy implication is applied by mamdani's minimum operator. The steps of Mamdami fuzzy interference system are as follows:

1. Apply Fuzzification to the input variables.
2. Apply fuzzy operators to the fuzzify variables.
3. Then implement implication methods.
4. Implement Aggregation process.
5. Apply Defuzzification at last

Consider an example of fuzzy inference with two inputs x and y respectively and generate corresponding single output z.

The rule defined in Mamdami fuzzy interference is as follows:

$$R_i = IFxisA_iANDyisB_iTHENzisC_i, i = 1 \dots., n\text{...............}4.9$$

With Triangular Membership Function:

$$\mu_A(x) = \begin{cases} 1 - \dfrac{x-a}{\alpha} if|x-a| \leq \alpha \\ 0 \qquad\qquad Otherwise \end{cases} \dots\dots\dots\dots.4.10$$

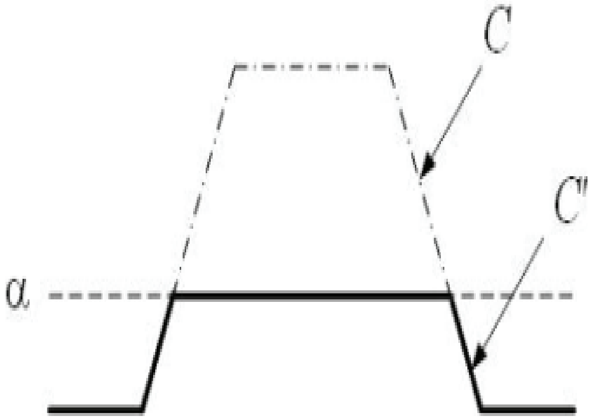

Figure 4.4: Graphical representation of Mamdani Model]

4.3 ADVANTAGE OF USING FUZZY LOGIC CONTROLLER

The major advantage of using fuzzy logic controller over traditional controller is that in fuzzy controller there is no requirement of mathematical modeling in order to design the controller. In fuzzy logics controller the control rule are created on the basis of the knowledge of the system behavior. Another main advantage of using the fuzzy logics controller is that it did not require any high speed processors.

4.4 METHODOLOGY

This section represents the strategy that is followed to implement the proposed work by using the fuzzy logic controllers. The step by step procedure for the proposed work is a below:

First step is to select the user defined membership functions. Then the fuzzy inference system is designed which performs the fuzzification, generating set rules by users and Deffuzification. After this the generated set of rules is applied to the input membership functions on the basis of the various kind membership functions i.e. trapezoidal, Gaussian and triangular. And at last the control output is generated by applying the proposed FLC with respect to the different type of membership functions. After generating the results, the comparison is performed among the generated output of three of the membership functions in order to find out the suitable membership function to control the angular position of servo motor by using the FLC instead of conventional controllers. The same is represented graphically in figure 4.5.

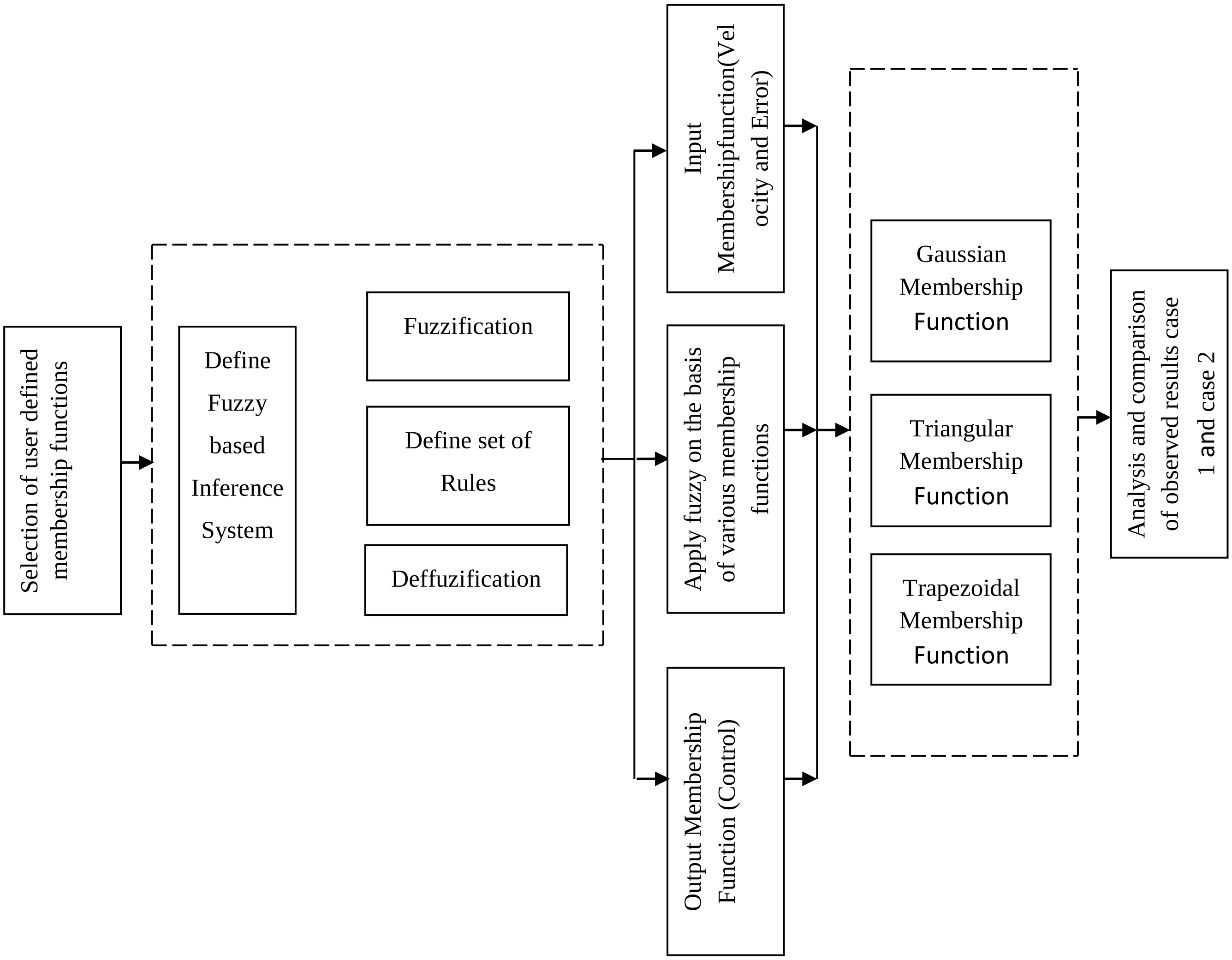
Selection of user defined membership functions
Define Fuzzy based Inference System
Fuzzification
Define set of Rules
Deffuzification
Input Membershipfunction(Velocity and Error)
Apply fuzzy on the basis of various membership functions
Output Membership Function (Control)
Gaussian Membership Function
Triangular Membership Function
Trapezoidal Membership Function
Analysis and comparison of observed results case 1 and case 2

Figure 4.5: Methodology of Proposed work

CHAPTER 5

SIMULATION RESULTS AND DISCUSSION

5.1 SIMULATION PLATFORM

MATLAB is a computing environment along with a high level, interpreted, dynamically typed language, aiding functional, object oriented, and event directed archetypes. It is relevant for arithmetic computations, specifically computations comprising matrix applications and linear algebra. Excellent support has been provided by MATLAB for data visualization and its concise and expressive syntax, as well as the plethora of predefined functions that result in a dynamic environment excellent for rapid prototyping with minimal overhead. However, MATLAB is not just a scripting language for rapid and dirty computations. Latest versions have examined a powerful increase for the support of large scale, highly structured code to rival the languages like C++, Java and many others. If you want the best of both worlds, MATLAB's integrated java support lets you create and manipulate instances of java class's right in your MATLAB programs. We can also call out to the code written in C, C++, Perl, FORTRAN, or execute dos or Unix system commands. Moreover, MATLAB code can be transported for use in Java, C, C++, and .Net programs, or as a standalone operation, alleviating its prime deprivation - that it is not free. As an interpreted language, some operations are slower as compared to compiled language, particularly those involving loops. This too has improved in recent versions. Loops can be vectorized however in many cases, resulting in very rapid code that appeals to low level, highly advanced, compiled operations from the MATLAB library.

When the most computationally important sections of your program can be written this way, the agility of MATLAB code rivals that of compiled languages. Finally, with the addition of various toolkits, e.g. for Statistics, Bioinformatics, Optimization, or Image Processing etc., the MATLAB software can be extended for more specialized requirements.

MATLAB is a programming language that works on mathematical calculations and brings the facility of high performance. This environment is widely used for technical computing. It incorporates distinct facilities on a particular platform like computations, programming, visualization etc. It has its own pre incorporated data structure and gadgets for editing and debugging. It also supports the concepts of OOPs. It is considered as a desirable environment or platform which is easy to teach, learn and for research purpose.

Comparing to traditional languages such as FORTRAN, MATLAB is more advantageous. Matlab provides more interactive GUI, through which the end users who does not have much knowledge about this software, can also access it easily. Array is the primary data aspect of Matlab, which does not have to define the dimensions also. This was accessible to end user in

1984 and till today, it is in use by numerous industries and research workers of distinct universities. Graphical commands are supported by it, hence by using some basics commands, one can operate it easily.

5.2 EXPERIMENTAL RESULTS

This section depicts the results that are obtained after implementing the proposed work in MATLAB by using the various membership function of fuzzy logic controller to the simplest fuzzy control scheme for a servo motor with the control model as under: -

$$Y(s)/ U(s) = G(s) = 1/s(s+3.6)$$

The discrete model for the plant can de depicted as under: -

$$Y(k)= 1.407y(k-1)-0.407y(k-2) +0.0237u(k-1) +0.0175u(k-2)$$

Following are the two types of input variables to the selected fuzzy controller: -

a. Error between the set point and the actual position of the shaft.
b. Rate of change of error.

And the output variable is

a. Incremental voltage signal to the driver circuit of the motor.

The case study pertains to the selection of the most suitable membership function in case of both the inputs and the output to produce best performance of the servo control. The study has been carried out in two parts as under: -

a. Case I. Implementation of common membership function to the inputs and different one to output.
b. Case II. Implementation of different membership function to the inputs and different one to output.

The membership functions that are considered in proposed work are as below:

a. Gaussian Membership function
b. Trapezoidal Membership function
c. Triangular Membership function
d. Pi Membership function.
e. Sigmoidal Membership function.

MATLAB code for performance of the selected servo motor control is given at Annexure I of the Appendix. Results with parameters are as under: -

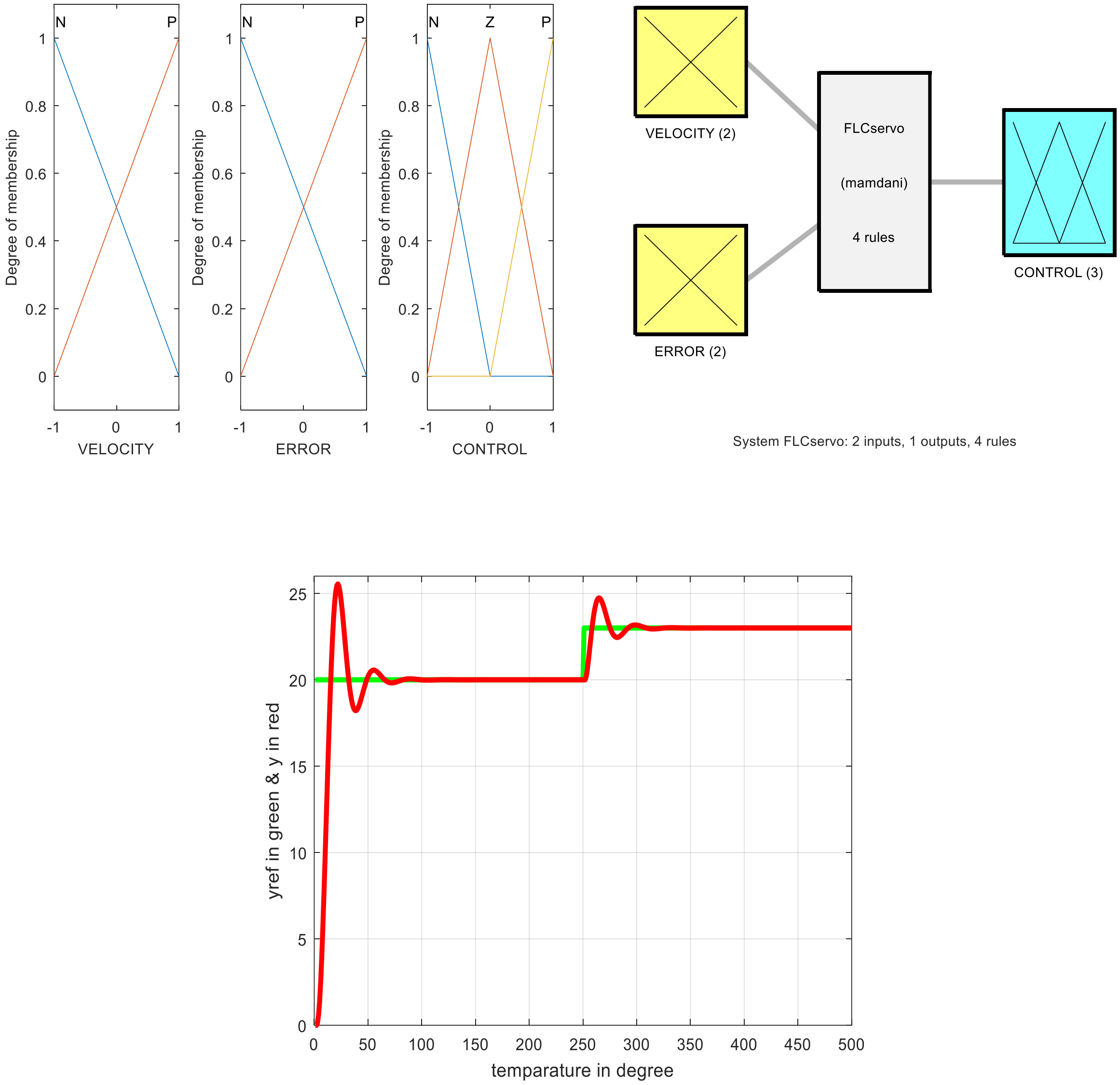

Figure 5.1: FLC for Triangular Membership Function

The parameters of the control output is given at the table below. The settling time for the control graph is 284.7395 ms with peak at 25.5464 and peak time at 22 ms.

Table 5.1 : Parameters of control graph for Triangular Membership Function

Rise Time	Settling Time	Overshoot	Undershoot	Peak	Peak Time
9.9937	284.7395	11.0715	0	25.5464	22

5.2.1 Case I :Implementation of Different Membership Function to the Inputs and the Output.

Script for the selection and implementation of membership function to inputs i.e., error and rate of change of error is at Anneure II to the Appendix. Membership Functions for all outputs in this case has been kept as triangular. Range for the inputs and the output has been selected as -1 to +1. In all the output/control graphs the reference output is shown as green and the control output is shown as red. Output results and the parameters of the control graphs are as appended below: -

a. Inputs :Trapezoidal Membership Function, Output : Triangular Membership Function

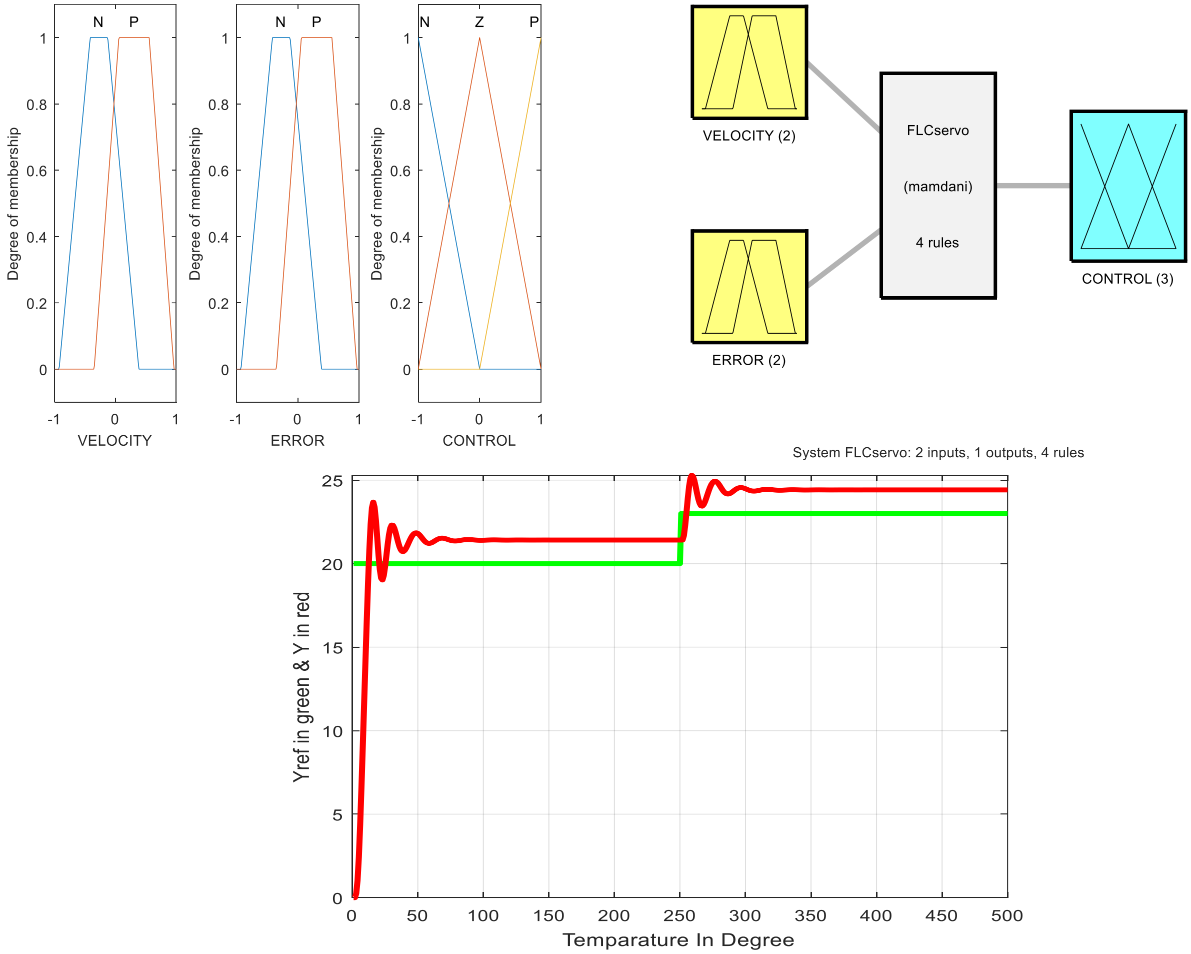

Figure 5.2: FLC for Trapezoidal Membership Function

The parameters of the control output is as below: -

Table 5.2 : Parameters for Control Graph of Trapezoidal Membership Function

Rise Time	Settling Time	Overshoot	Undershoot	Peak	Peak Time
8.7448	277.9806	3.5961	0	25.2905	259

b. Inputs :Guassian Membership Function, Output : Triangular Membership Function

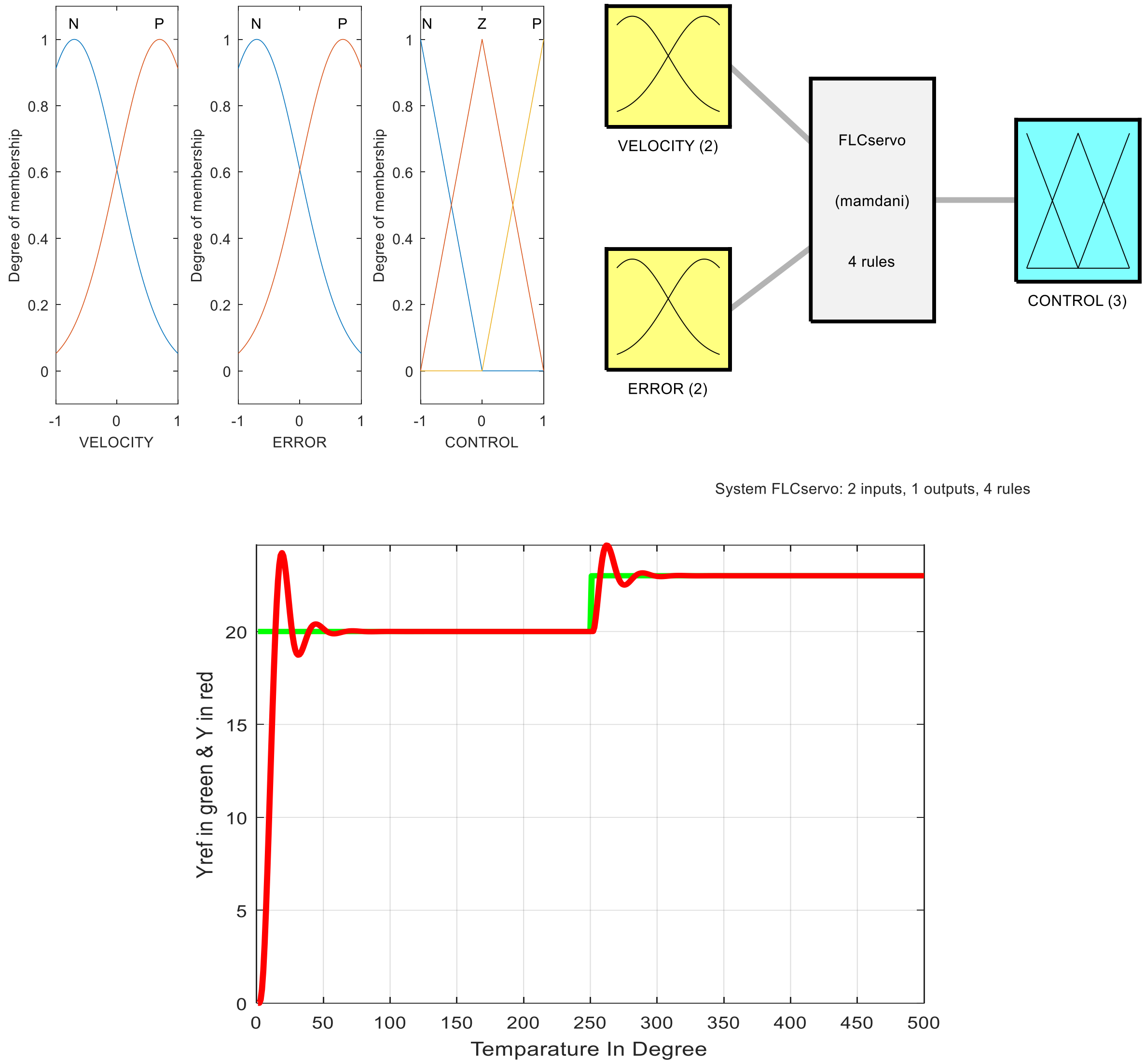

Figure 5.3 : FLC for Guassian Membership Function

The parameters of the control output is given at the table below. The settling time for the control graph is 284.7395 ms with peak at 25.5464 and peak time at 22 ms.

Table 5.3 : Parameters for Control Graph of Guassian Membership Function

Rise Time	Settling Time	Overshoot	Undershoot	Peak	Peak Time
9.1100	277.2099	7.1567	0	24.6460	262

c. <u>Inputs :Pi Membership Function, Output : Triangular Membership Function</u>

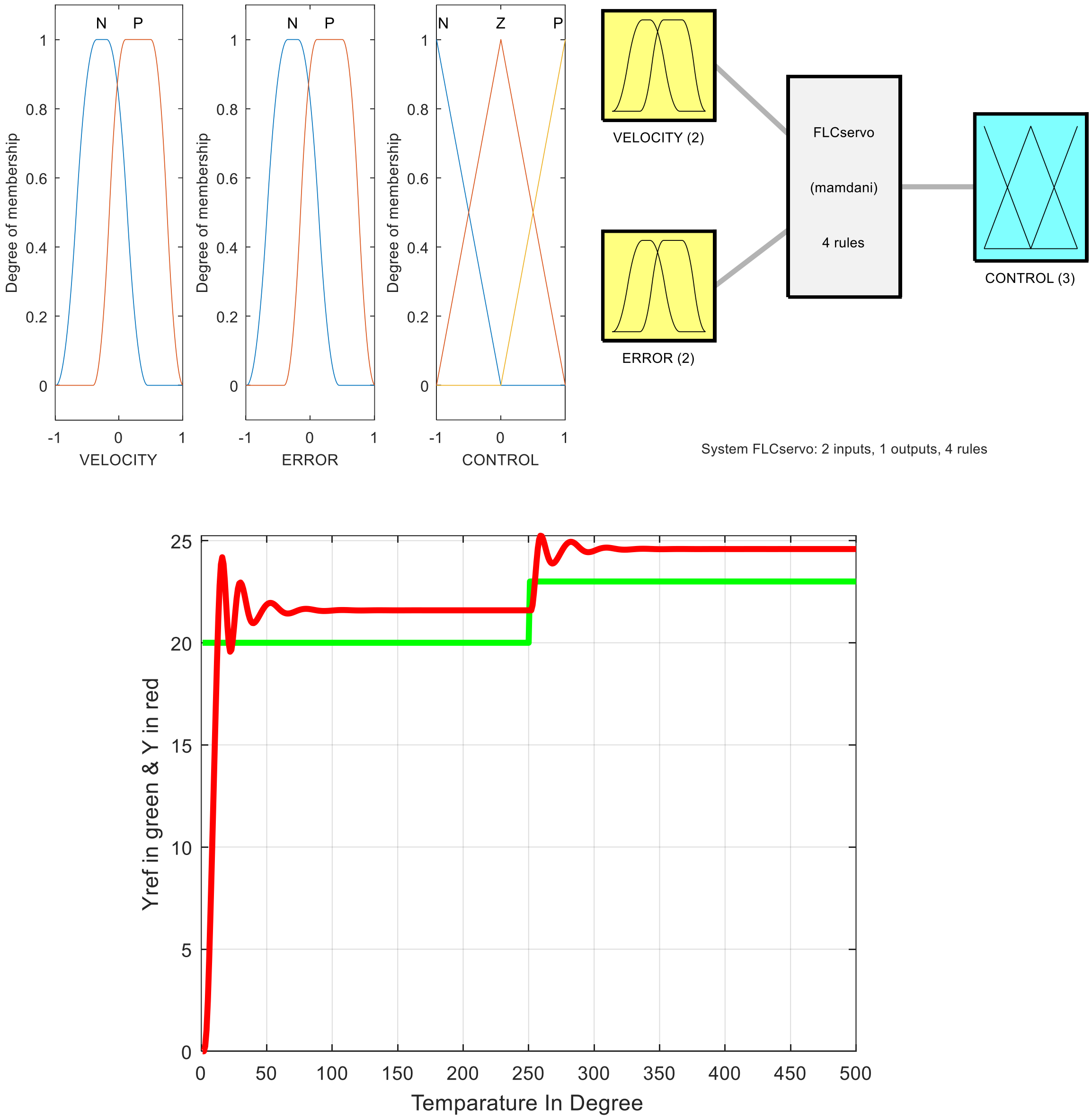

Figure 5.4 : FLC for Pi Membership Function

The parameters of the control output is given at the table below. The settling time for the control graph is 271.7294 ms with peak at 25.2421 and peak time at 259 ms.

Table 5.4 : Parameters for Control Graph of Pi Membership Function

Rise Time	Settling Time	Overshoot	Undershoot	Peak	Peak Time
8.5748	271.7294	2.6582	0	25.2421	259

d. Inputs : Sigmoidal Membership Function, Output : Triangular Membership Function

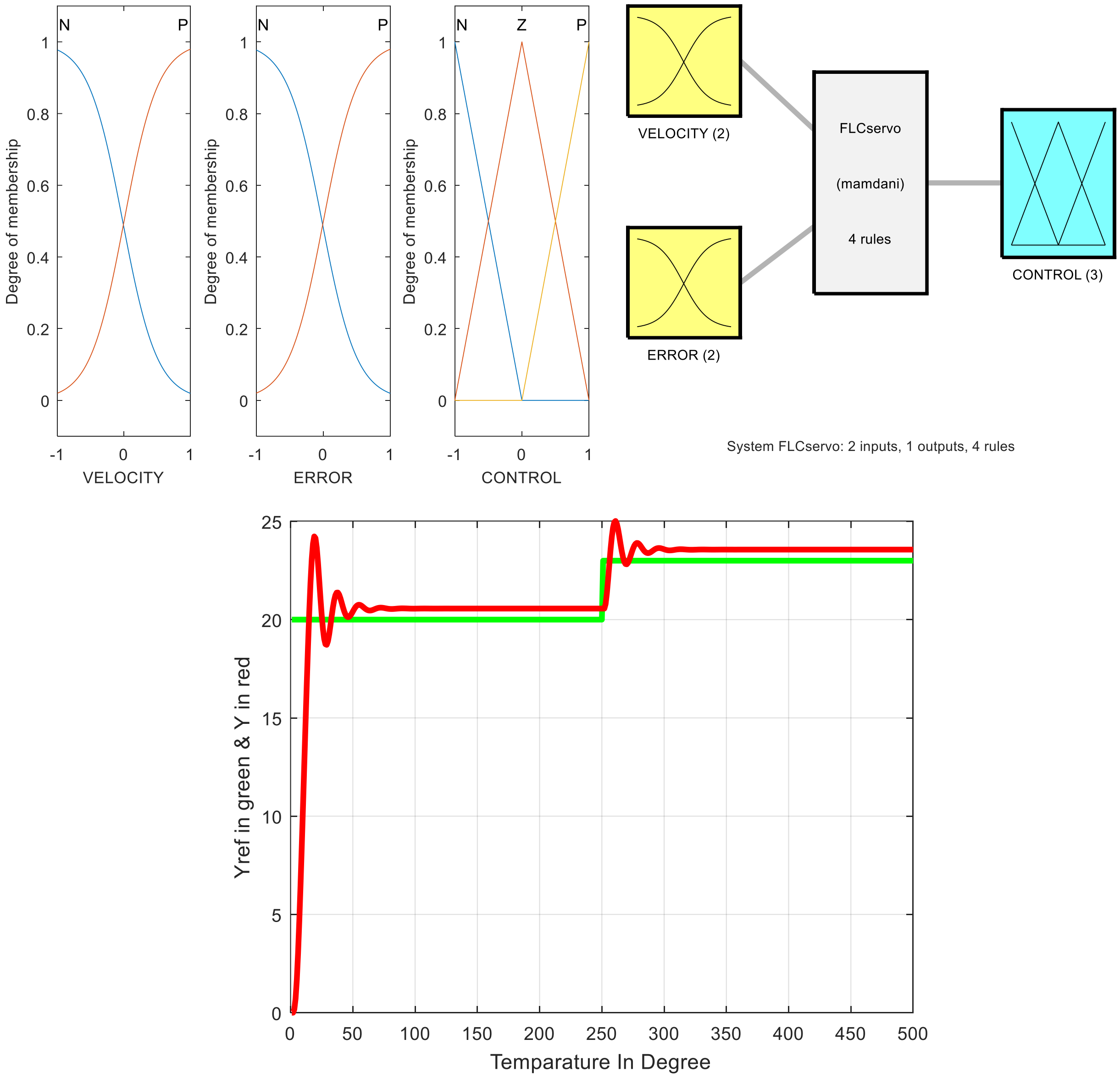

Figure 5.5 : FLC for Sigmoidal Membership Function

The parameters of the control output is given at the table below. The settling time for the control graph is 272.2708 ms with peak at 25.0182 and peak time at 261ms.

Table 5.5 : Parameters for Control Graph of Sigmoidal Membership Function

Rise Time	Settling Time	Overshoot	Undershoot	Peak	Peak Time
10.0365	272.2708	6.1734	0	25.0182	261

5.2.2. <u>Case II. Implementation of different membership function to the inputs and common one to output</u>.

a. <u>Inputs : Error-Gaussian MF, Velocity- Trapezoidal MF, Output- Triangular MF</u>

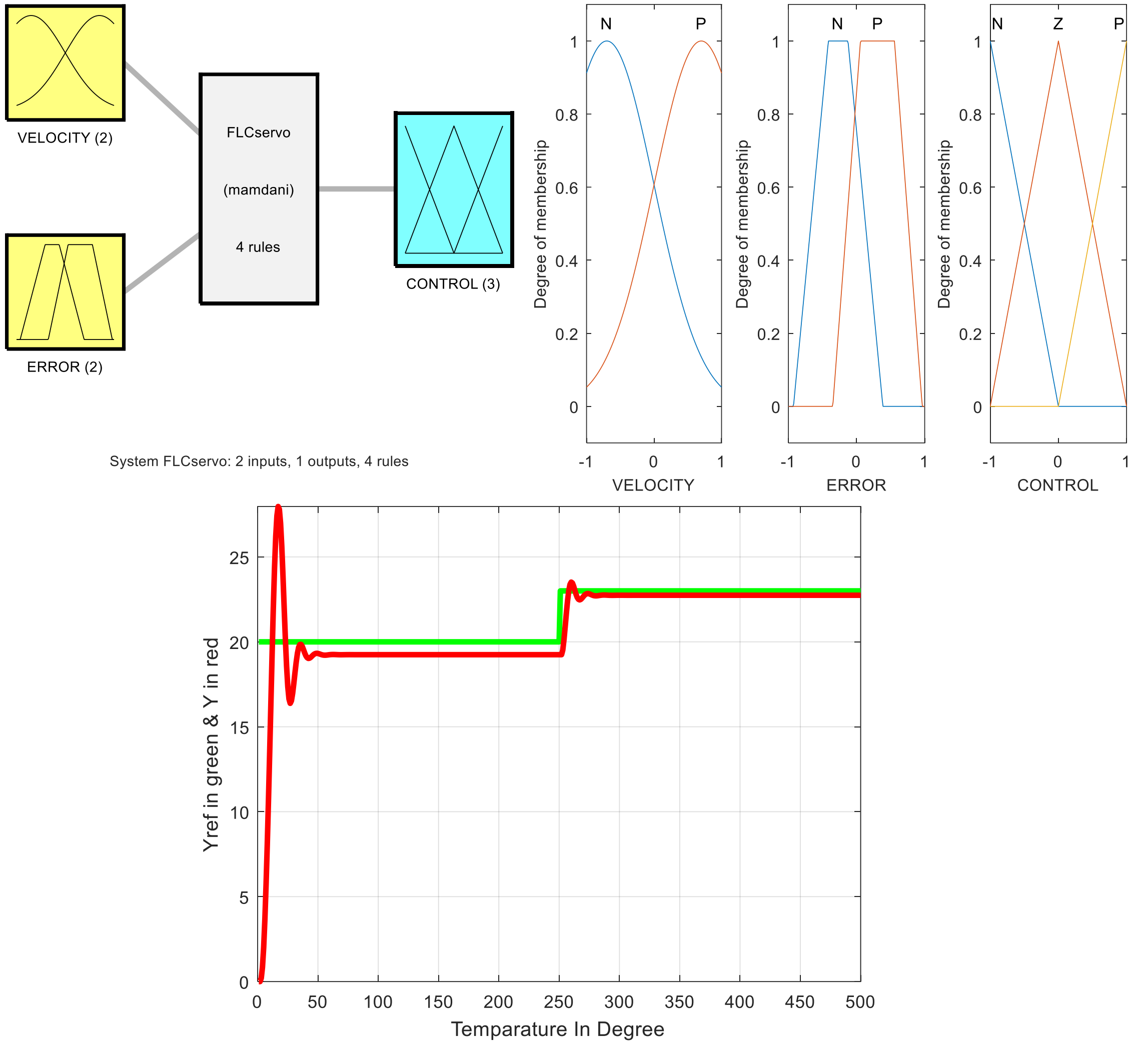

Figure 5.6 : FLC for Gaussian MF/ Trapezoidal MF as Input & Triangular MF as Output

The parameters of the control output is given at the table below. The settling time for the control graph is 262.3386 ms with peak at 27.9675and peak time at 17ms.

Table 5.6 : Parameters for Control Graph of Gaussian MF/ Trapezoidal MF as Input & Triangular MF as Output

Rise Time	Settling Time	Overshoot	Undershoot	Peak	Peak Time
7.2673	262.3386	22.9682	0	27.9675	17

b. Inputs : Error-Trapezoidal MF, Velocity- Gaussian MF, Output- Triangular MF

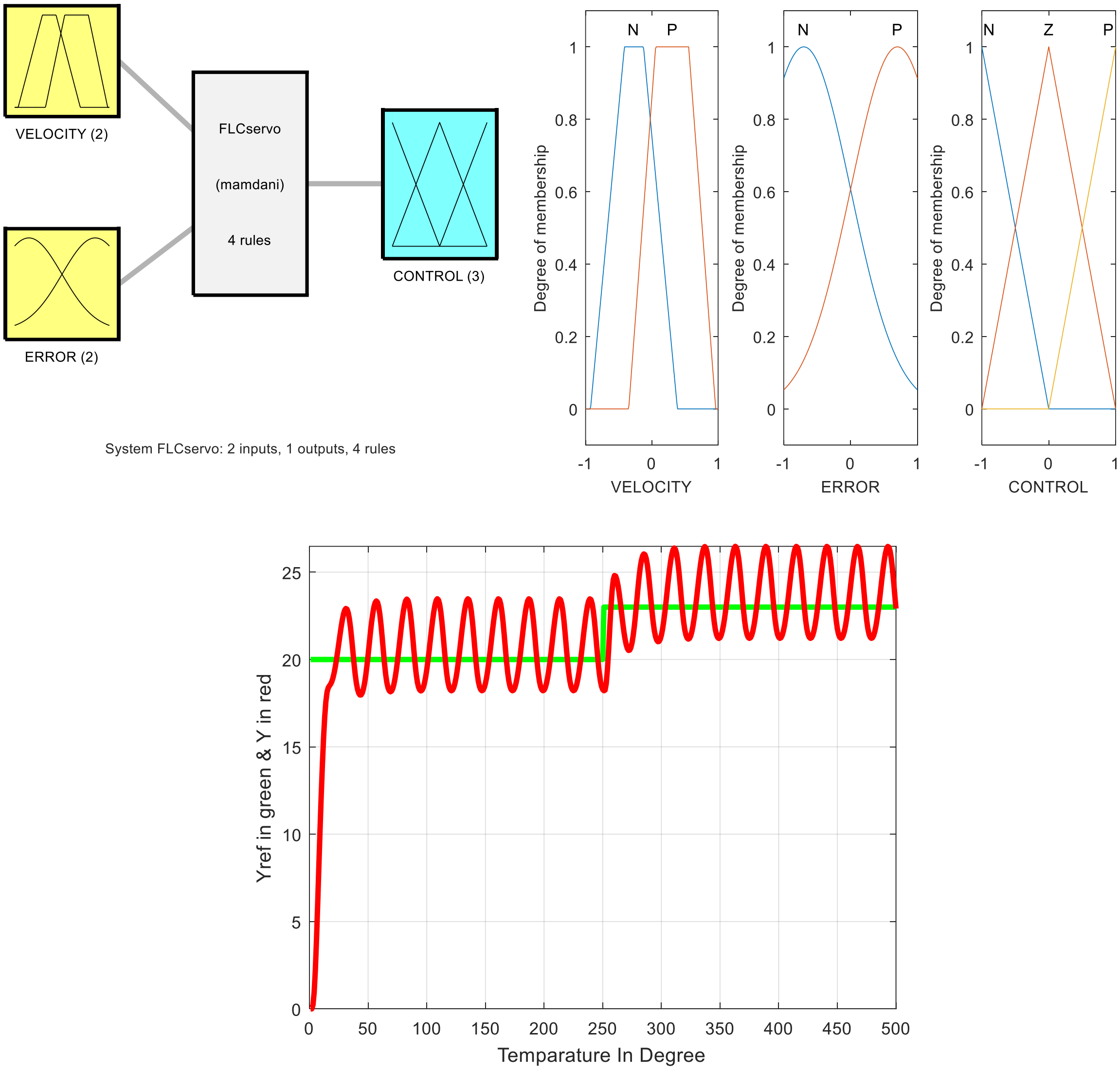

Figure 5.7: FLC for Trapezoidal MF/ Gaussian MF as Input & Triangular MF as Output

The parameters of the control output is given at the table below. The settling time for the control graph is 499.3974ms with peak at 26.4780and peak time at 363ms.

Table 5.7 : Parameters for Control Graph of Trapezoidal MF/ Gaussian MF as Input & Triangular MF as Output

Rise Time	Settling Time	Overshoot	Undershoot	Peak	Peak Time
19.3311	499.3974	15.5533	0	26.4780	363

c. Inputs : Error-Trapezoidal MF, Velocity- Triangular MF, Output- Gaussian MF

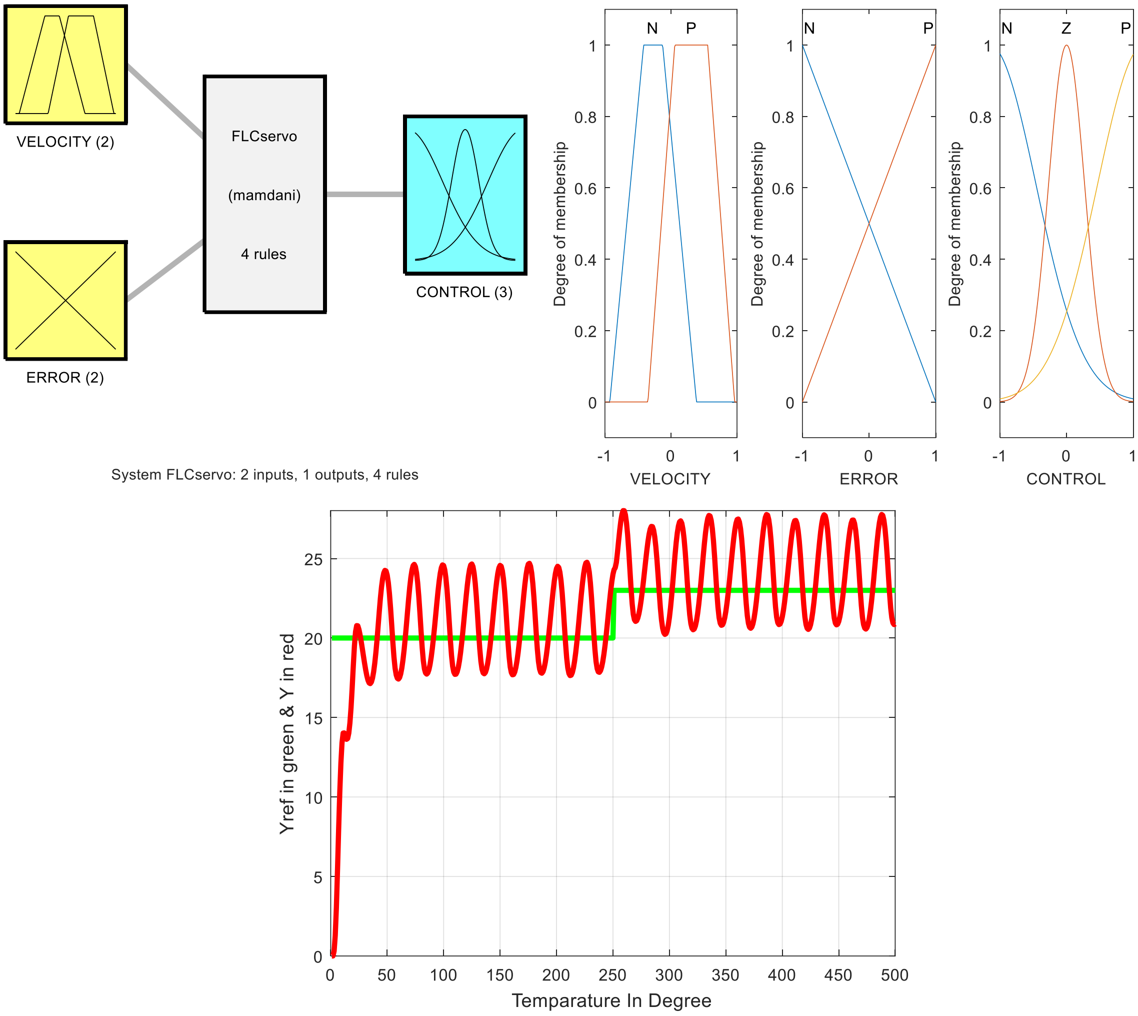

Figure 5.8: FLC for Trapezoidal MF/ Triangular MF as Input & Gaussian MF as Output

The parameters of the control output is given at the table below. The settling time for the control graph is 496.9356 ms with peak at 28.0139and peak time at 259 ms.

Table 5.8 : Parameters for Control Graph of Trapezoidal MF/ Triangular MF as Input & Gaussian MF as Output

Rise Time	Settling Time	Overshoot	Undershoot	Peak	Peak Time
15.9752	496.9356	34.0920	0	28.0139	259

d. Inputs : Error- Triangular MF, Velocity- Trapezoidal MF, Output : Gaussian MF

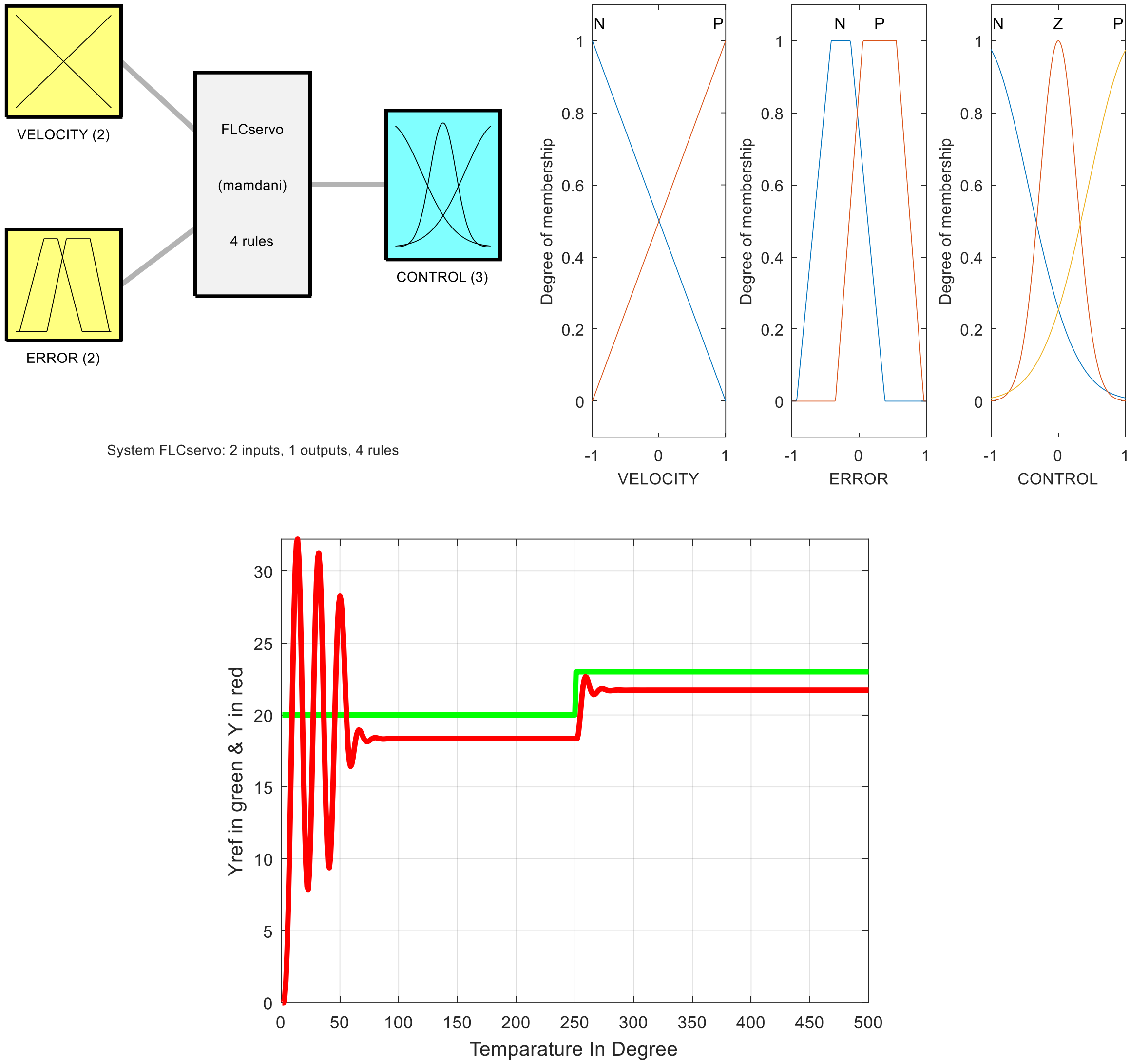

Figure 5.9: FLC for Triangular MF/ Trapezoidal MF as Input & Gaussian MF as Output

The parameters of the control output is given at the table below. The settling time for the control graph is 261.8834ms with peak at 32.2233and peak time at 14 ms.

Table 5.9 : Parameters for Control Graph of Triangular MF/ Trapezoidal MF as Input & Gaussian MF as Output

Rise Time	Settling Time	Overshoot	Undershoot	Peak	Peak Time
4.789	261.8834	48.3565	0	32.2233	14

e. Inputs : Error- Guassian MF, Velocity- Triangular MF, Output :Trapezoidal MF

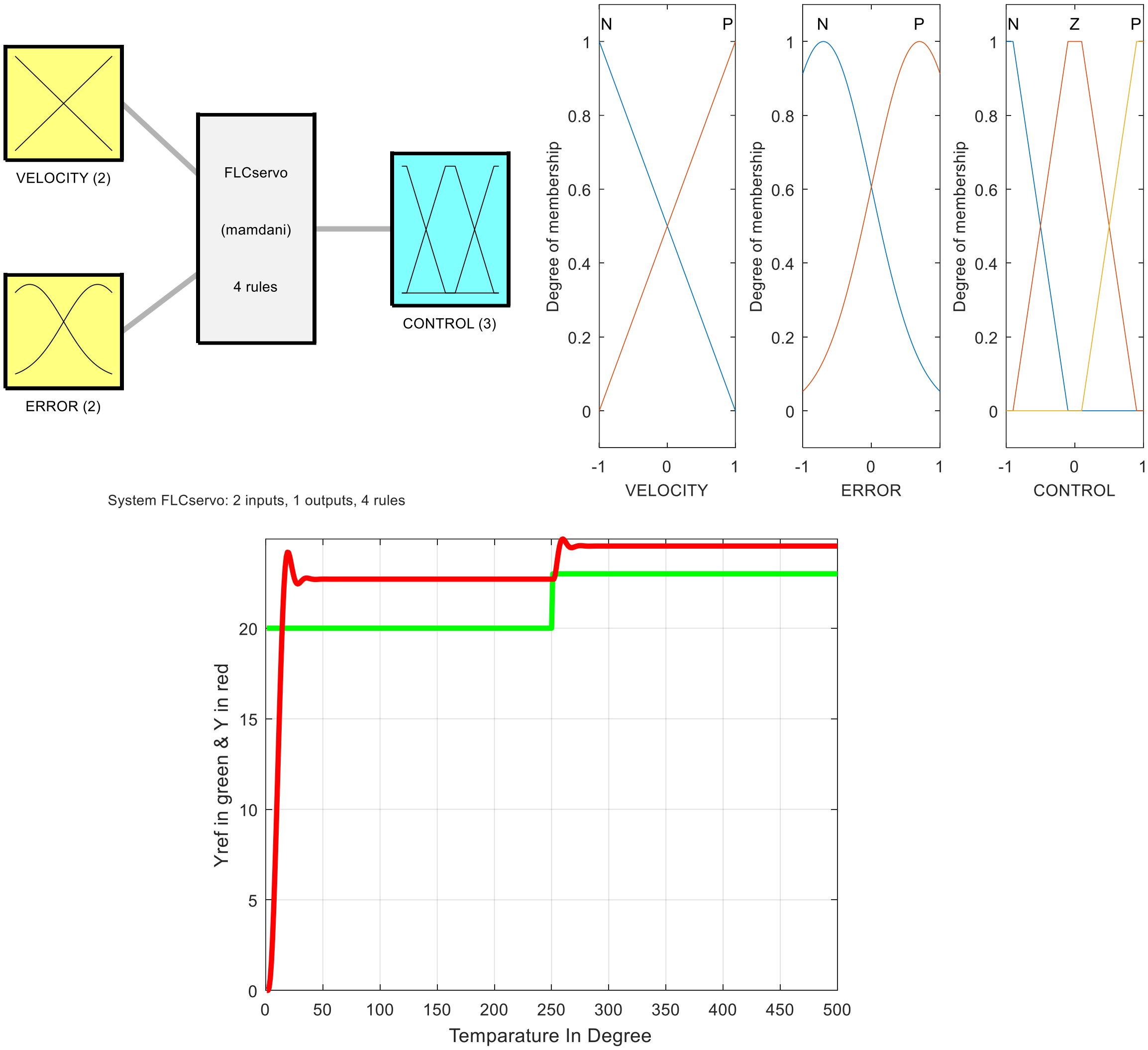

Figure 5.10: FLC for Gaussian MF/ Triangular MF as Input & Trapezoidal MF as Output

The parameters of the control output is given at the table below. The settling time for the control graph is 255.6478ms with peak at 24.9232and peak time at 260 ms.

Table 5.10 : Parameters for Control Graph of Gaussian MF/ Triangular MF as Input & Trapezoidal MF as Output

Rise Time	Settling Time	Overshoot	Undershoot	Peak	Peak Time
10.1390	255.6478	1.5831	0	24.9232	260

f. Inputs : Error- Triangular MF, Velocity- Guassian MF, Output :Trapezoidal MF

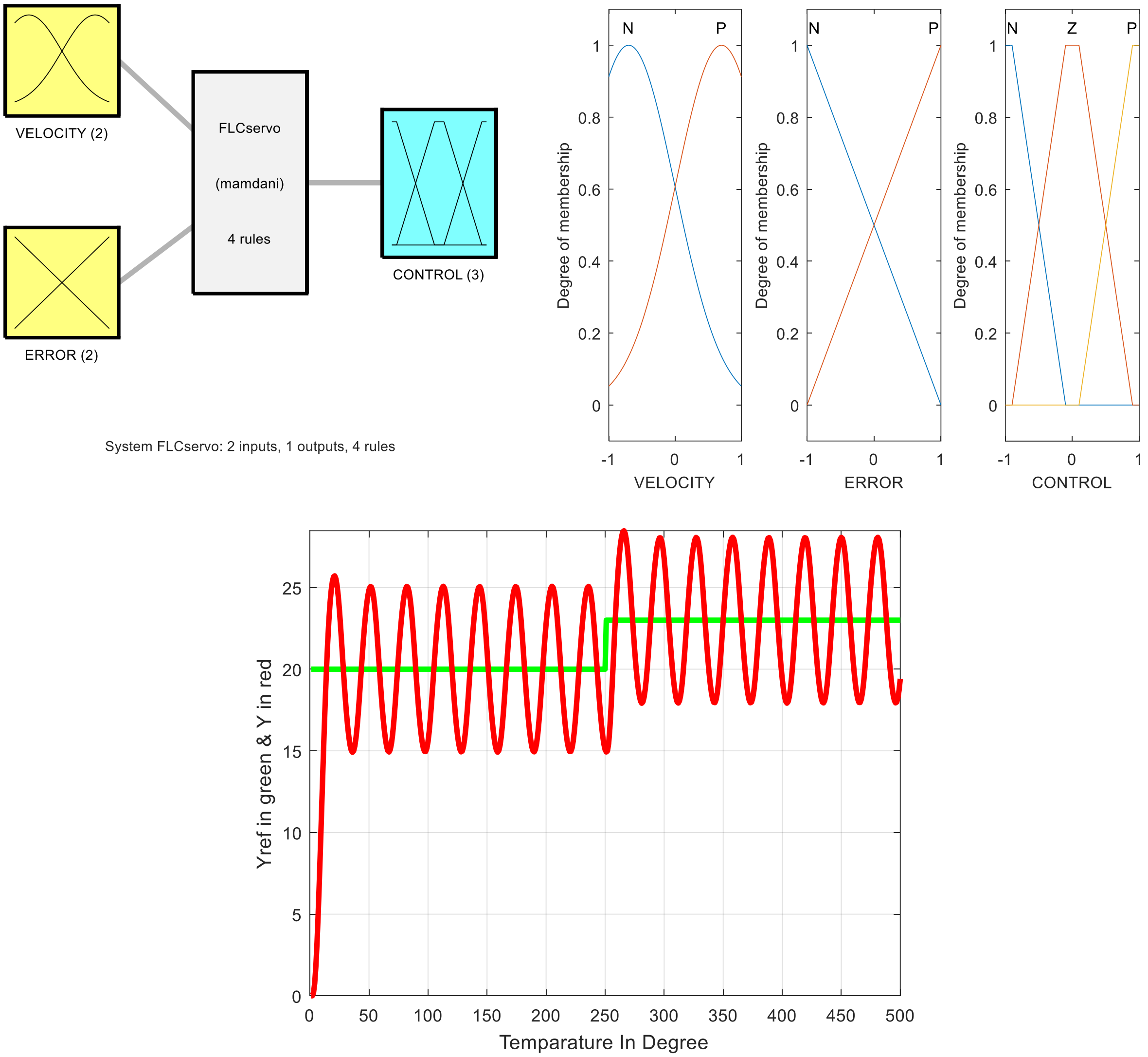

Figure 5.11: FLC for Triangular MF/ Gaussian MF as Input & Trapezoidal MF as Output

The parameters of the control output is given at the table below. The settling time for the control graph is 499.4302 ms with peak at 28.4737and peak time at 266 ms.

Table 5.11 : Parameters for Control Graph of Triangular MF/ Gaussian MF as Input & Trapezoidal MF as Output

Rise Time	Settling Time	Overshoot	Undershoot	Peak	Peak Time
7.5608	499.4302	46.7058	0	28.4737	266

CHAPTER 6

CONCLUSION AND FUTURE SCOPE

In this work the fuzzy logic controller is applied to the servo motor for controlling the angular position of the servo motor. The main objective of the work can be concluded as the analysis of the effects of types of membership functions on the control output of the servo motor. The analysis has been done on the basis of two different cases where different membership functions were applied to the inputs i.e. error and rate of change error and control output. The range in all cases was restricted to – 1 to +1. The comparison of the parameters obtained in case I is as shown below : -

Table 5.12: Parameter comparison for Similar Inputs & different Output Membership Function

Ser No	Membership Function	Parameters						
		Rise Time	Settling Time	Settling Min	Settling Max	Overshoot	Undershoot	Peak
1.	Triangular	9.9937	284.7395	18.2238	25.5464	11.0715	0	25.5464
2.	Trapezoidal	8.7448	277.9806	19.0201	25.2905	3.5961	0	25.2905
3.	Guassian	9.1100	277.2099	18.7281	24.6460	7.1567	0	24.6460
4.	Pi	8.5748	**271.7294**	19.5617	25.2421	2.6582	0	25.2421
5.	Sigmoidal	10.0365	272.2708	18.7184	25.0182	6.1734	0	25.0182

The comparison of parameters in Case II is as below: -

Table 5.13: Parameter comparison for different Input and Output Membership Function

Ser No	Membership Function			Parameters							
	Error	Velocity	Output	Rise Time	Settling Time	Settling Min	Settling Max	Overs-hoot	Unders-hoot	Peak	Peak Time
1.	Guassian	Trapezoidal	Triangular	7.2673	262.3386	16.3795	27.9675	22.9682	0	27.9675	17
2.	Trapezoidal	Guassian	Triangular	19.3311	499.3974	17.9693	26.4780	15.5533	0	26.4780	363
3.	Trapezoidal	Triangular	Guassian	15.9752	496.9356	17.1322	28.0139	34.0920	0	28.0139	259
4.	Triangular	Trapezoidal	Guassian	4.789	261.8834	7.8644	32.2233	48.3565	0	32.2233	14
5.	Guassian	Triangular	Trapezoidal	10.1390	255.6478	22.3284	24.9232	1.5831	0	24.9232	260
6.	Triangular	Guassian	Trapezoidal	7.5608	499.4302	14.9089	28.4737	46.7058	0	28.4737	266

In all the Case I, it can be seen that the settling time varies from 271.7294 to 284.7395 in Pi & Triangular membership function respectively. In terms of Y following up & meeting the reference parameter is concern, the control graft in case of inputs with Guassian membership function shows better results. The peak achieved in case of Sigmoidal membership function for inputs is least. The control response in case of different selection of membership function for both the inputs and the outputs is complicated in case of combination of Ser No 2, 3 & 6 in table 5.2 , the settling time is very high ranging from 496.9356 to 499.4302. Thus, the control would take a longer time to implement and the response of the servo motor would be slower. However, the combination of membership function selected in case of Ser No 5 i.e., error with Guassain membership function, velocity with triangular and output with Trapezoidal membership function yields better results than the others. It settles down test with minimum overshoot. On the basis of the results that are define in this work, it can be concluded that the best results are achieved when a combination of Guassain membership function is used for error, Triangular membership function for rate of change of error and Trapezoidal membership function for control output is applied. Therefore, the fuzzy logic controller produces the good and dynamic results for controlling the angular position of the servo motor with above combination. In future system can be enhanced by introducing the improvements in the fuzzy logic controllers with respect to the effective result provider membership function, implementation of optimization algorithms like Graphical algorithms, FF and different types of Neural Network Systems.

.

APPENDIX

Annexure I

```
% FLC Servo
% Refer Review Example 12.1
clc;
close all;
clear all
% Create new FIS with filename "FISservo.fis"
sys=newfis('FLCservo')
% Define membership functions for the
% Input variable "Rate of change of error"
sys=addvar(sys,'input','VELOCITY',[-1 1]);
sys=addmf(sys,'input',1,'N','trimf',[-1 -1 1]);
sys=addmf(sys,'input',1,'P','trimf',[-1 1 1]);
% Define membership functions for the input variable "Error"
sys=addvar(sys,'input','ERROR',[-1 1]);
sys=addmf(sys,'input',2,'N','trimf',[-1 -1 1]);
sys=addmf(sys,'input',2,'P','trimf',[-1 1 1]);
% Define membership functions for the
% Output variable "Incremental change in input to the motor"
sys=addvar(sys,'output','CONTROL',[-1 1]);
sys=addmf(sys,'output',1,'N','trimf',[-1 -1 0]);
sys=addmf(sys,'output',1,'Z','trimf',[-1 0 1]);
sys=addmf(sys,'output',1,'P','trimf',[0 1 1]);
% Define fuzzy rules
rulelist=[2 2 3 1 1;1 2 2 1 1;2 1 2 1 1;1 1 1 1 1];
sys=addrule(sys,rulelist);
% Define gain constants
GE=0.03;
GV=0.05;
GU=60;
% Define reference signal
for k=1:500
if (k<=250)
yref(k)=20;
else
yref(k)=23;
end
end
% Command line simulation
y(1)=0; y(2)=0; u(1)=0;
for k=1:499
    e(k)=yref(k)-y(k);
if(k==1)
v(1)=0;
else
        v(k)=(e(k)-e(k-1))/0.25;
end
```

```
einp(k)=e(k)*GE;
vinp(k)=v(k)*GV;
delu(k)=evalfis([vinp(k) einp(k)],sys);
   u(k+1)=u(k)+delu(k)*GU;
if(k==1)
y(2)=0;
else
      y(k+1)=0.0237*u(k)+0.0175*u(k-1)+1.407*y(k)-0.407*y(k-1);
end
end
k=1:500;
plot(k,yref,'linewidth',3,'color',[0 1 0]);
grid
axis([0 500 0 26]);
xlabel('temparature in degree');ylabel('yref in green & y in red');
hold on
plot(k,y,'linewidth',3,'color',[1 0 0]);
```

Annexure II

1. Main Script

```
clc;
close all;
clear all

warning off;

% Create new FIS with filename "FISservo.fis"
sys=newfis('FLCservo');

% Selection of membership functions for the
disp('Select Any MemberShip Function')
disp('1- Trapezoidal MF')
disp('2- Gaussian MF')
disp('3- Pi MF')
disp('4- Sigmoidal MF')
disp('5- Triangular  MF')

system=0;
while system<1
system=input('Enter Your Choice Of MF: ');
if system<=0 || system >5
   system=0;
end
end

disp('Enter Option For MF range')
disp('1- Range -1 to 1')
disp('2- Range -2 to 2')
disp('3- Range -3 to 3')
disp('4- Range -4 to 4')

Rangetype=0;
whileRangetype<1
Rangetype=input('Enter your Choice For Range: ');
ifRangetype<=0 || Rangetype>5
Rangetype=0;
end
end
ifRangetype==1
   sys=System_Selection(sys,system);
elseifRangetype==2
   sys=DiffRSystemSele(sys,system);
elseifRangetype==3
   sys=DiffRSystemSeleA(sys,system);
elseifRangetype==4
   sys=DiffRSystemSeleB(sys,system);
end
```

```
% Define membership functions for the
% Output variable "Incremental change in input to the motor"
sys=addvar(sys,'output','CONTROL',[-1 1]);
sys=addmf(sys,'output',1,'N','trimf',[-1 -1 0]);
sys=addmf(sys,'output',1,'Z','trimf',[-1 0 1]);
sys=addmf(sys,'output',1,'P','trimf',[0 1 1]);

% Define fuzzy rules
rulelist=[2 2 3 1 1;1 2 2 1 1;2 1 2 1 1;1 1 1 1 1];

sys=addrule(sys,rulelist);

figure
subplot(131)
plotmf(sys,'input',1)
subplot(132)
plotmf(sys,'input',2)
subplot(133)
plotmf(sys,'output',1)

figure
plotfis(sys)

% Define gain constants
GE=0.03;
GV=0.05;
GU=60;
% Define reference signal
for k=1:500
if (k<=250)
yref(k)=20;
else
yref(k)=23;
end
end
% Command line simulation
y(1)=0; y(2)=0; u(1)=0;
for k=1:499
    e(k)=yref(k)-y(k);
if(k==1)
v(1)=0;
else
        v(k)=(e(k)-e(k-1))/0.25;
end
```

```
einp(k)=e(k)*GE;
vinp(k)=v(k)*GV;
delu(k)=evalfis([vinp(k) einp(k)],sys);
   u(k+1)=u(k)+delu(k)*GU;
if(k==1)
y(2)=0;
else
      y(k+1)=0.0237*u(k)+0.0175*u(k-1)+1.407*y(k)-0.407*y(k-1);
end
end
k=1:500;
figure
plot(k,yref,'linewidth',3,'color',[0 1 0]);
grid
axis([0 500 0 max(y)]);
xlabel('Temparature In Degree');ylabel('Yref in green & Y in red');
hold on
plot(k,y,'linewidth',3,'color',[1 0 0]);

S = stepinfo(y)
```

2. Sub Scripts

(a)

```
function sys=System_Selection(sys,system)
% Create new FIS with filename "FISservo.fis"
% Define membership functions for the
if system==1
% Input variable "Rate of change of error"
sys=addvar(sys,'input','VELOCITY',[-1 1]);
sys=addmf(sys,'input',1,'N','trapmf',  [-0.9258 -0.4138 -0.1258 0.3862]);
sys=addmf(sys,'input',1,'P','trapmf', [-0.349 0.0575 0.556 0.964]);
% Define membership functions for the input variable "Error"
sys=addvar(sys,'input','ERROR',[-1 1]);
sys=addmf(sys,'input',2,'N','trapmf',[-0.9258 -0.4138 -0.1258 0.3862]);
sys=addmf(sys,'input',2,'P','trapmf',[-0.349 0.0575 0.556 0.964]);
elseif system==2
% Input variable "Rate of change of error"
sys=addvar(sys,'input','VELOCITY',[-1 1]);
sys=addmf(sys,'input',1,'N','gaussmf',  [0.7 -0.7]);
sys=addmf(sys,'input',1,'P','gaussmf',  [0.7 0.7]);
% Define membership functions for the input variable "Error"
sys=addvar(sys,'input','ERROR',[-1 1]);
sys=addmf(sys,'input',2,'N','gaussmf',  [0.7 -0.7]);
sys=addmf(sys,'input',2,'P','gaussmf',  [0.7 0.7]);
elseif system==3
% Input variable "Rate of change of error"
sys=addvar(sys,'input','VELOCITY',[-1 1]);
```

```
sys=addmf(sys,'input',1,'N','pimf', [-0.9898 -0.3498 -0.1898 0.4502]);
sys=addmf(sys,'input',1,'P','pimf', [-0.405 0.103 0.5 1.01]);
% Define membership functions for the input variable "Error"
sys=addvar(sys,'input','ERROR',[-1 1]);
sys=addmf(sys,'input',2,'N','pimf', [-0.9898 -0.3498 -0.1898 0.4502]);
sys=addmf(sys,'input',2,'P','pimf', [-0.405 0.103 0.5 1.01]);
elseif system==4
% Input variable "Rate of change of error"
sys=addvar(sys,'input','VELOCITY',[-1 1]);
sys=addmf(sys,'input',1,'N','sigmf',   [-3.841 -0.0192]);
sys=addmf(sys,'input',1,'P','sigmf',   [3.88 0.0021]);
% Define membership functions for the input variable "Error"
sys=addvar(sys,'input','ERROR',[-1 1]);
sys=addmf(sys,'input',2,'N','sigmf',    [-3.841 -0.0192]);
sys=addmf(sys,'input',2,'P','sigmf',  [3.88 0.0021]);
elseif system==5
disp('Default Case Selected')
% Input variable "Rate of change of error"
sys=addvar(sys,'input','VELOCITY',[-1 1]);
sys=addmf(sys,'input',1,'N','trimf',[-1 -1 1]);
sys=addmf(sys,'input',1,'P','trimf',[-1 1 1]);
% Define membership functions for the input variable "Error"
sys=addvar(sys,'input','ERROR',[-1 1]);
sys=addmf(sys,'input',2,'N','trimf',[-1 -1 1]);
sys=addmf(sys,'input',2,'P','trimf',[-1 1 1]);
end
```

(b)

```
function sys=DiffRSystemSele(sys,system)
% Create new FIS with filename "FISservo.fis"
% Define membership functions for the
if system==1
% Input variable "Rate of change of error"
sys=addvar(sys,'input','VELOCITY',[-2 2]);
sys=addmf(sys,'input',1,'N','trapmf',   [-1.957 -0.9334 -0.3574 0.6666]);
sys=addmf(sys,'input',1,'P','trapmf',  [-0.698 0.115 1.112 1.928]);
% Define membership functions for the input variable "Error"
sys=addvar(sys,'input','ERROR',[-2 2]);
sys=addmf(sys,'input',2,'N','trapmf',[-1.957 -0.9334 -0.3574 0.6666]);
sys=addmf(sys,'input',2,'P','trapmf', [-0.698 0.115 1.112 1.928]);
elseif system==2
% Input variable "Rate of change of error"
sys=addvar(sys,'input','VELOCITY',[-2 2]);
sys=addmf(sys,'input',1,'N','gaussmf',[1.4 -1.4]);
sys=addmf(sys,'input',1,'P','gaussmf',[1.4 1.4]);
% Define membership functions for the input variable "Error"
sys=addvar(sys,'input','ERROR',[-2 2]);
sys=addmf(sys,'input',2,'N','gaussmf',[1.4 -1.4]);
sys=addmf(sys,'input',2,'P','gaussmf',[1.4 1.4]);
```

```
elseif system==3
% Input variable "Rate of change of error"
sys=addvar(sys,'input','VELOCITY',[-2 2]);
sys=addmf(sys,'input',1,'N','pimf',  [-1.98 -0.6996 -0.3796 0.9004]);
sys=addmf(sys,'input',1,'P','pimf',  [-0.81 0.206 1 2.02]);
% Define membership functions for the input variable "Error"
sys=addvar(sys,'input','ERROR',[-2 2]);
sys=addmf(sys,'input',2,'N','pimf', [-1.98 -0.6996 -0.3796 0.9004]);
sys=addmf(sys,'input',2,'P','pimf', [-0.81 0.206 1 2.02]);
elseif system==4
% Input variable "Rate of change of error"
sys=addvar(sys,'input','VELOCITY',[-2 2]);
sys=addmf(sys,'input',1,'N','sigmf', [-1.921 -0.0384]);
sys=addmf(sys,'input',1,'P','sigmf',  [1.94 0.0042]);
% Define membership functions for the input variable "Error"
sys=addvar(sys,'input','ERROR',[-2 2]);
sys=addmf(sys,'input',2,'N','sigmf', [-1.921 -0.0384]);
sys=addmf(sys,'input',2,'P','sigmf', [1.94 0.0042]);
elseif system==5
disp('Default Case Selected')
% Input variable "Rate of change of error"
sys=addvar(sys,'input','VELOCITY',[-2 2]);
sys=addmf(sys,'input',1,'N','trimf',[-2 -2 2]);
sys=addmf(sys,'input',1,'P','trimf',[-2 2 2]);
% Define membership functions for the input variable "Error"
sys=addvar(sys,'input','ERROR',[-2 2]);
sys=addmf(sys,'input',2,'N','trimf',[-2 -2 2]);
sys=addmf(sys,'input',2,'P','trimf',[-2 2 2]);
end
```

(c)

```
function sys=DiffRSystemSeleA(sys,system)
% Create new FIS with filename "FISservo.fis"
% Define membership functions for the
if system==1
% Input variable "Rate of change of error"
sys=addvar(sys,'input','VELOCITY',[-3 3]);
sys=addmf(sys,'input',1,'N','trapmf',    [-2.936 -1.4 -0.5361 0.9999]);
sys=addmf(sys,'input',1,'P','trapmf',   [-1.047 0.1725 1.668 2.892]);
% Define membership functions for the input variable "Error"
sys=addvar(sys,'input','ERROR',[-3 3]);
sys=addmf(sys,'input',2,'N','trapmf',[-2.936 -1.4 -0.5361 0.9999]);
sys=addmf(sys,'input',2,'P','trapmf', [-1.047 0.1725 1.668 2.892]);
elseif system==2
% Input variable "Rate of change of error"
sys=addvar(sys,'input','VELOCITY',[-3 3]);
sys=addmf(sys,'input',1,'N','gaussmf', [2.1 -2.1]);
sys=addmf(sys,'input',1,'P','gaussmf', [2.1 2.1]);
% Define membership functions for the input variable "Error"
```

```
sys=addvar(sys,'input','ERROR',[-3 3]);
sys=addmf(sys,'input',2,'N','gaussmf', [2.1 -2.1]);
sys=addmf(sys,'input',2,'P','gaussmf', [2.1 2.1]);
elseif system==3
% Input variable "Rate of change of error"
sys=addvar(sys,'input','VELOCITY',[-3 3]);
sys=addmf(sys,'input',1,'N','pimf',  [-2.97 -1.049 -0.5694 1.351]);
sys=addmf(sys,'input',1,'P','pimf',  [-1.215 0.309 1.5 3.03]);
% Define membership functions for the input variable "Error"
sys=addvar(sys,'input','ERROR',[-3 3]);
sys=addmf(sys,'input',2,'N','pimf',[-2.97 -1.049 -0.5694 1.351]);
sys=addmf(sys,'input',2,'P','pimf',  [-1.215 0.309 1.5 3.03]);
elseif system==4
% Input variable "Rate of change of error"
sys=addvar(sys,'input','VELOCITY',[-3 3]);
sys=addmf(sys,'input',1,'N','sigmf', [-1.281 -0.0576]);
sys=addmf(sys,'input',1,'P','sigmf',  [1.293 0.0063]);
% Define membership functions for the input variable "Error"
sys=addvar(sys,'input','ERROR',[-3 3]);
sys=addmf(sys,'input',2,'N','sigmf', [-1.281 -0.0576]);
sys=addmf(sys,'input',2,'P','sigmf', [1.293 0.0063]);
elseif system==5
disp('Default Case Selected')
% Input variable "Rate of change of error"
sys=addvar(sys,'input','VELOCITY',[-3 3]);
sys=addmf(sys,'input',1,'N','trimf',[-3 -3 3]);
sys=addmf(sys,'input',1,'P','trimf',[-3 3 3]);
% Define membership functions for the input variable "Error"
sys=addvar(sys,'input','ERROR',[-3 3]);
sys=addmf(sys,'input',2,'N','trimf',[-3 -3 3]);
sys=addmf(sys,'input',2,'P','trimf',[-3 3 3]);
end
```

(d)

```
unction sys=DiffRSystemSeleB(sys,system)
% Create new FIS with filename "FISservo.fis"
% Define membership functions for the
if system==1
% Input variable "Rate of change of error"
sys=addvar(sys,'input','VELOCITY',[-4 4]);
sys=addmf(sys,'input',1,'N','trapmf',[-3.915 -1.867 -0.7148 1.333]);
sys=addmf(sys,'input',1,'P','trapmf',[-1.396 0.23 2.224 3.856]);
% Define membership functions for the input variable "Error"
sys=addvar(sys,'input','ERROR',[-4 4]);
sys=addmf(sys,'input',2,'N','trapmf',[-3.915 -1.867 -0.7148 1.333]);
sys=addmf(sys,'input',2,'P','trapmf', [-1.396 0.23 2.224 3.856]);
elseif system==2
% Input variable "Rate of change of error"
sys=addvar(sys,'input','VELOCITY',[-4 4]);
```

```
sys=addmf(sys,'input',1,'N','gaussmf', [2.8 -2.8]);
sys=addmf(sys,'input',1,'P','gaussmf', [2.8 2.8]);
% Define membership functions for the input variable "Error"
sys=addvar(sys,'input','ERROR',[-4 4]);
sys=addmf(sys,'input',2,'N','gaussmf', [2.8 -2.8]);
sys=addmf(sys,'input',2,'P','gaussmf', [2.8 2.8]);
elseif system==3
% Input variable "Rate of change of error"
sys=addvar(sys,'input','VELOCITY',[-4 4]);
sys=addmf(sys,'input',1,'N','pimf',[-3.98 -1.42 -0.78 1.78]);
sys=addmf(sys,'input',1,'P','pimf', [-1.62 0.412 2 4.04]);
% Define membership functions for the input variable "Error"
sys=addvar(sys,'input','ERROR',[-4 4]);
sys=addmf(sys,'input',2,'N','pimf',[-3.98 -1.42 -0.78 1.78]);
sys=addmf(sys,'input',2,'P','pimf', [-1.62 0.412 2 4.04]);
elseif system==4
% Input variable "Rate of change of error"
sys=addvar(sys,'input','VELOCITY',[-1 1]);
sys=addmf(sys,'input',1,'N','sigmf', [-0.9607 -0.0768]);
sys=addmf(sys,'input',1,'P','sigmf', [0.9697 0.0084]);
% Define membership functions for the input variable "Error"
sys=addvar(sys,'input','ERROR',[-1 1]);
sys=addmf(sys,'input',2,'N','sigmf', [-0.9607 -0.0768]);
sys=addmf(sys,'input',2,'P','sigmf', [0.9697 0.0084]);
elseif system==5
disp('Default Case Selected')
% Input variable "Rate of change of error"
sys=addvar(sys,'input','VELOCITY',[-4 4]);
sys=addmf(sys,'input',1,'N','trimf',[-4 -4 4]);
sys=addmf(sys,'input',1,'P','trimf',[-4 4 4]);
% Define membership functions for the input variable "Error"
sys=addvar(sys,'input','ERROR',[-4 4]);
sys=addmf(sys,'input',2,'N','trimf',[-4 -4 4]);
sys=addmf(sys,'input',2,'P','trimf',[-4 4 4]);
end
```

Annexure III

1. Main Script

```
clc;
close all;
clear all

warning off;

sys=newfis('FLCservo');
disp('Select Any MemberShip Function')
disp('1- Gaussian MF')
disp('2- Triangular MF')
disp('3- Trapezoidal MF')

t1=input('Type Of Input MF for Velocity: ');
t2=input('Type Of Input MF for Error: ');
t3=input('Type Of Output MF for CONTROL: ');

sys=System_Selection_Combination(sys,t1,t2,t3);

% Define fuzzy rules
rulelist=[2 2 3 1 1;1 2 2 1 1;2 1 2 1 1;1 1 1 1 1];

sys=addrule(sys,rulelist);

figure
subplot(131)
plotmf(sys,'input',1)
subplot(132)
plotmf(sys,'input',2)
subplot(133)
plotmf(sys,'output',1)

figure
plotfis(sys)

% Define gain constants
GE=0.03;
GV=0.05;
GU=60;
% Define reference signal
for k=1:500
if (k<=250)
yref(k)=20;
else
yref(k)=23;
end
end
```

```
% Command line simulation
y(1)=0; y(2)=0; u(1)=0;
for k=1:499
   e(k)=yref(k)-y(k);
if(k==1)
v(1)=0;
else
      v(k)=(e(k)-e(k-1))/0.25;
end

einp(k)=e(k)*GE;
vinp(k)=v(k)*GV;
delu(k)=evalfis([vinp(k) einp(k)],sys);
   u(k+1)=u(k)+delu(k)*GU;
if(k==1)
y(2)=0;
else
      y(k+1)=0.0237*u(k)+0.0175*u(k-1)+1.407*y(k)-0.407*y(k-1);
end
end
k=1:500;
figure
plot(k,yref,'linewidth',3,'color',[0 1 0]);
grid
axis([0 500 0 max(y)]);
xlabel('Temparature In Degree');ylabel('Yref in green & Y in red');
hold on
plot(k,y,'linewidth',3,'color',[1 0 0]);

S = stepinfo(y)
```

2. <u>Sub Script</u>

```
if t1==1
sys=addvar(sys,'input','VELOCITY',[-1 1]);
sys=addmf(sys,'input',1,'N','gaussmf',  [0.7 -0.7]);
sys=addmf(sys,'input',1,'P','gaussmf',  [0.7 0.7]);

elseif t1==2
% Input variable "Rate of change of error"
sys=addvar(sys,'input','VELOCITY',[-1 1]);
sys=addmf(sys,'input',1,'N','trimf',[-1 -1 1]);
sys=addmf(sys,'input',1,'P','trimf',[-1 1 1]);

elseif t1==3
% Input variable "Rate of change of error"
sys=addvar(sys,'input','VELOCITY',[-1 1]);
sys=addmf(sys,'input',1,'N','trapmf',  [-0.9258 -0.4138 -0.1258 0.3862]);
sys=addmf(sys,'input',1,'P','trapmf', [-0.349 0.0575 0.556 0.964]);
end
```

```
if t2==1
sys=addvar(sys,'input','ERROR',[-1 1]);
sys=addmf(sys,'input',2,'N','gaussmf',  [0.7 -0.7]);
sys=addmf(sys,'input',2,'P','gaussmf',  [0.7 0.7]);
elseif t2==2
% Define membership functions for the input variable "Error"
sys=addvar(sys,'input','ERROR',[-1 1]);
sys=addmf(sys,'input',2,'N','trimf',[-1 -1 1]);
sys=addmf(sys,'input',2,'P','trimf',[-1 1 1]);

elseif t2==3
% Define membership functions for the input variable "Error"
sys=addvar(sys,'input','ERROR',[-1 1]);
sys=addmf(sys,'input',2,'N','trapmf',[-0.9258 -0.4138 -0.1258 0.3862]);
sys=addmf(sys,'input',2,'P','trapmf',[-0.349 0.0575 0.556 0.964]);
end

if t3==1
sys=addvar(sys,'output','CONTROL',[-1 1]);
sys=addmf(sys,'output',1,'N','gaussmf',  [0.7 -1.155]);
sys=addmf(sys,'output',1,'Z','gaussmf',   [0.2719 1.39e-17]);
sys=addmf(sys,'output',1,'P','gaussmf',   [0.7 1.161]);

elseif t3==2
sys=addvar(sys,'output','CONTROL',[-1 1]);
sys=addmf(sys,'output',1,'N','trimf',[-1 -1 0]);
sys=addmf(sys,'output',1,'Z','trimf',[-1 0 1]);
sys=addmf(sys,'output',1,'P','trimf',[0 1 1]);

elseif t3==3
sys=addvar(sys,'output','CONTROL',[-1 1]);
sys=addmf(sys,'output',1,'N','trapmf',  [-1 -1 -0.9 -0.1]);
sys=addmf(sys,'output',1,'Z','trapmf',  [-0.9 -0.1 0.1 0.9]);
sys=addmf(sys,'output',1,'P','trapmf', [0.1 0.9 1 1]);

end
```

Annexure IV

CASE 1:

```
clc
close all
clear all
warning off

% Create new FIS with filename "FISservo.fis"
sys=newfis('FLCservo');
disp('Select Any MemberShip Function')
disp('1- Gaussian MF')
disp('2- Triangular MF')
disp('3- Trapezoidal MF')

t1=input('Type Of Input MF for Velocity: ');
t2=input('Type Of Input MF for Error: ');
t3=input('Type Of Output MF for CONTROL: ');

sys=TypeSelection(sys,t1,t2,t3);

% Define fuzzy rules
rulelist=[ 1  1  1  1  1
         1  1  2  1  1
         1  2  3  1  1
         1  2  1  1  1
         2  3  2  1  1
         2  3  3  1  1
         2  1  1  1  1
         2  1  2  1  1
         3  2  3  1  1
         3  2  1  1  1
         3  3  2  1  1
         3  3  3  1  1];

sys=addrule(sys,rulelist);

figure
subplot(131)
plotmf(sys,'input',1)
subplot(132)
plotmf(sys,'input',2)
subplot(133)
plotmf(sys,'output',1)

figure
plotfis(sys)

% Define gain constants
```

```
GE=0.03;
GV=0.05;
GU=60;
% Define reference signal
for k=1:500
if (k<=250)
yref(k)=20;
else
yref(k)=23;
end
end
% Command line simulation
y(1)=0; y(2)=0; u(1)=0;
for k=1:499
    e(k)=yref(k)-y(k);
if(k==1)
v(1)=0;
else
        v(k)=(e(k)-e(k-1))/0.25;
end

einp(k)=e(k)*GE;
vinp(k)=v(k)*GV;
delu(k)=evalfis([vinp(k) einp(k)],sys);
    u(k+1)=u(k)+delu(k)*GU;
if(k==1)
y(2)=0;
else
        y(k+1)=0.0237*u(k)+0.0175*u(k-1)+1.407*y(k)-0.407*y(k-1);
end
end
k=1:500;
figure
plot(k,yref,'linewidth',3,'color',[0 1 0]);
grid
axis([0 500 0 max(y)]);
xlabel('Temparature In Degree');ylabel('Yref in green & Y in red');
hold on
plot(k,y,'linewidth',3,'color',[1 0 0]);

S = stepinfo(y)
```

CASE 2:

```
clc;
close all;
clear all
warning off;
```

```
sys=newfis('FLCservo');
disp('Select Any MemberShip Function')
disp('1- Gaussian MF')
disp('2- Triangular MF')
disp('3- Trapezoidal MF')

t1=input('Type Of Input MF for Velocity: ');
t2=input('Type Of Input MF for Error: ');
t3=input('Type Of Output MF for CONTROL: ');

sys=System_Selection_Combination(sys,t1,t2,t3);

% Define fuzzy rules
rulelist=[2 2 3 1 1;1 2 2 1 1;2 1 2 1 1;1 1 1 1 1];

sys=addrule(sys,rulelist);

figure
subplot(131)
plotmf(sys,'input',1)
subplot(132)
plotmf(sys,'input',2)
subplot(133)
plotmf(sys,'output',1)

figure
plotfis(sys)

% Define gain constants
GE=0.03;
GV=0.05;
GU=60;
% Define reference signal
for k=1:500
if (k<=250)
yref(k)=20;
else
yref(k)=23;
end
end
% Command line simulation
y(1)=0; y(2)=0; u(1)=0;
for k=1:499
   e(k)=yref(k)-y(k);
if(k==1)
v(1)=0;
else
        v(k)=(e(k)-e(k-1))/0.25;
end
```

```
einp(k)=e(k)*GE;
vinp(k)=v(k)*GV;
delu(k)=evalfis([vinp(k) einp(k)],sys);
   u(k+1)=u(k)+delu(k)*GU;
if(k==1)
y(2)=0;
else
      y(k+1)=0.0237*u(k)+0.0175*u(k-1)+1.407*y(k)-0.407*y(k-1);
end
end
k=1:500;
figure
plot(k,yref,'linewidth',3,'color',[0 1 0]);
grid
axis([0 500 0 max(y)]);
xlabel('Temparature In Degree');ylabel('Yref in green & Y in red');
hold on
plot(k,y,'linewidth',3,'color',[1 0 0]);

         S = stepinfo(y)
```

REFERENCES

[1]. Nader Jamali Soufi1, Mohsen Kabiri Moghaddam2, Saeed Sfandiarpour Boroujeni3, Alireza Vahidifar1, (2014) “A Parameter Varying PD Control for Fuzzy Servo Mechanism”, Scientific Research, Intelligent Control and Automation, Vol 5, Pp 156-169

[2]. Thwin Thu Lynna*, Eaintb, (2016) “Position Control of DC Servo Drive by Fuzzy Logic Controller in Flat-Bed Screen Printing Machine”, International Journal of Science, Engineering and Technology Research (IJSETR), Vol 5(10), Pp 2968-2973.

[3]. Meghanasingh, (2017), “EFFECTS OF TIME DELAY ON DIFFERENT TYPES OF CONTROLLER FOR NETWORKED CONTROL SYSTEMS: A REVIEW”, International Research Journal of Engineering and Technology, Vol 4 (4), Pp 993-1000.

[4]. Xudong Liu and Yong Gu Wang Yongliang Wang, (2014)“Control Research of Network Servo System Based on Smith Predictor “Yu IEEE”, Beij in. Proceeding of the 11th World Congress on Intelligent Control and Automation Shenyang, China, June 29 - July 4 2014

[5]. Honge Ren, Xuehai Cao and Jifeng Guo “Modified Smith Predictor Design and Its Applications to Long Time Delay Systems”. International Journal of Signal Processing, Image Processing and Pattern Recognition Vol. 8, No. 5 (2015)

[6]. Cui Hao,Denghua Li and Shuhua Peng “The Fuzzy Adaptive Smith-PID Control of Three-Tank-System” School of Automation Beijing Information Science & Technology University Beijing, ChinaE-mail:haocui043 3234@163.com IEEE 2010

[7]. D. Yue, Q. L. Han, and J. Lam, “Network-based robust H∞ control of systems with uncertainty,” *Automatica*, vol. 41, no. 6, pp. 999–1007, Jun. 2005.

[8]. S.Soucek, T. Sauter, and G. Koller, “Effect of delay jitter on quality of control in EIA-852-based networks,” in *Proc. IECON*, 2003, vol.2,pp. 1431–1436.

[9]. L. Zhang, C. Wang, and Y. Chen, “Stability and stabilization of a classof multimode linear discrete-time systems with polytypic uncertainties, ”*IEEE Trans. Ind. Electron.*, vol. 56, no. 9, pp. 3684–3692, Sep. 2009.

[10]. Q. Li, G. Yi, C. Wang, L. Wu, and C. Ma, “LMI-based stability analysis of networked control systems with large time-varying delays,” in Proc. IEEE Int. Conf. Mechatronics Autom., 2006, pp. 713–717.

[11]. Y. Xia, G. P. Liu, and D. H. Rees, “H∞ control for networked control systems in presence of random network delay and data dropout,” in Proc. Chin. Control Conf., 2006, pp. 2030–2034.

[12]. K. Natori and K. Ohnishi, “A design method of communication disturbance observer for time-delay compensation, taking the dynamic property of network disturbance into account,” IEEE Trans. Ind. Electron., vol. 55,no. 5, pp. 2152–2168, May 2008.

[13]. T. Richards and M.-Y. Chow, “Performance characterization of IP network-based control methodologies for DC motor applications. Part I,”in Proc. IECON, 2005, pp. 6–10.

[14]. Amirul Syafiq Sadun, Jamaludin Jalani and Jumadi Abdul Sukor, (2016), “a comparative study on the position control method of dc servo motor with osition feedback by using arduino”, ARPN Journal of Engineering and Applied Sciences, Vol 11(18), PP 10954-10958

[15]. Mohamed A. Darwish, 2Hossam S. Abbas, (2012), “DC Motor Speed and Position Control Using Discrete-Time Fixed-Order H∞ Controllers”, International Journal on Information Management, Vol 1 (1), Pp 1-14

[16]. Pradnya Pathade1 and S. D. Lokhand, (2012), Position Control of Servo Motor Using Fuzzy Logic Controller, IJCIRA, Vol 6(1), pp 47-52,

[17]. .Y. Yao, (1998), “A comparative study of fuzzy sets and rough sets”, Information Science, ELSEVIER, Vol 109, Pp 227-242

[18]. Ahmad N A Haider, Chellali Benachaiba, Mohammed Zahir, (2013), “Software Interfacing of Servo Motor with Microcontroller”, Journal of Electrical Systems, Pp 84-99.

[19]. Siti Nur Wahidah Binti Abdul Wahab, (2014), “motor control system development using microcontroller based on pid controller”, Pp 1-39

[20]. Eng Simon Sameul, (2017) “Neuro-Fuzzy Based Approach to Object Tracking and Motion Prediction”, IJSEA, Vol 6(10), Pp 1-15.