Sudan University of Science & Technology

College of Graduate Studies

A COMPARATIVE STUDY OF PID, FUZZY AND NEURO-FUZZY CONTROLLERS FOR POSITON CONTROL OF DC SEVOMOTOR

دراسة مقارنة بين المتحكم التناسبى التكاملى التفاضلى والمتحكمات الذكية للتحكم فى وضع محرك خدمة التيار المستمر

A Thesis Submitted in Partial Fulfillment to the Requirements for the Degree of M.Sc. in Electrical Engineering (Control and Microprocessor)

Prepared by:

Yassir Abdallah Alhameem Albager

Supervised by:

Dr. Awadalla Taifour Ali

May, 2014

الآية

بِسۡمِ ٱللَّهِ ٱلرَّحۡمَٰنِ ٱلرَّحِيمِ

ٱقۡرَأۡ بِٱسۡمِ رَبِّكَ ٱلَّذِي خَلَقَ ﴿١﴾ خَلَقَ ٱلۡإِنسَٰنَ مِنۡ عَلَقٍ ﴿٢﴾
ٱقۡرَأۡ وَرَبُّكَ ٱلۡأَكۡرَمُ ﴿٣﴾ ٱلَّذِي عَلَّمَ بِٱلۡقَلَمِ ﴿٤﴾ عَلَّمَ ٱلۡإِنسَٰنَ مَا
لَمۡ يَعۡلَمۡ ﴿٥﴾

صدق الله العظيم

الآيات (1 - 5) من سورة العلق

DEDICATIONS

This thesis is lovingly dedicated to my parent, my brothers, my sisters and my friends I love you both as you supported me all the way to this day. I miss you all. To they have given me the drive and discipline to tackle any task with enthusiasm and determination. Without their love and support this project would not have been made possible.

ACKNOWLEDGEMENT

I would like to thank all the people that supported me through my academic way, and helped me to understand the engineering concepts needed that were needed for our engineering life. My special thanks to my Supervisor social professor ***Awadallalla Taifor Ali,*** *as he was beside me in every step throughout my research, and for his great effort to show me the best techniques and sources to solve different problems .*

ABSTRACT

This thesis presents a comparative study of various controllers for the position control of DC servomotor. The most commonly used controller for the position control of DC servomotor is conventional Proportional- Integral- Derivative (PID) controller. However, the PID controller has some disadvantages such as: the high starting overshoot, sensitivity to controller gains and sluggish response due to sudden disturbance. So, the relatively design PID controller with computational optimization approach method is proposed to overcome the disadvantages of the conventional PID controller. Further, two fuzzy logic based controllers namely; fuzzy control and neuro-fuzzy control are proposed in this study and the performance of these controllers are compared with PID controller performance. Simulation results are presented and analyzed for all controllers. It is observed that neuro-fuzzy controller gives a better response than other controllers for the position control of DC servomotor drives.

مستخلص

يقدم هذا البحث مقارنة دراسية بين متحكمات مختلفة للتجكم فى وضع محرك خدمة التيار المستمر. أغلب المتحكمات المستخدمة للتحكم فى الوضع هى المتحكمات التقليدية المعروفة بالمتحكم التناسبى التكاملى التفاضلى.ولكن رغم ذلك نجد أن المتحكم التناسبى التكاملى التفاضلى لديه بعض العيوب مثل البدء بتجاوز عالي للهدف ،حساسيه لكسب المتحكمه والاستجابة البطيئة للتغيرات الفجائية .وعليه فإن مقترح تصميم المتحكم التناسبى التكاملى التفاضلى بالطريقة الحسابية المثلى هو يتغلب على بعض عيوب المتحكم التناسبى التكاملى التفاضلى التقليدى .وغضافة لذلك فإن إثنين متحكمان يعتمدان على المنطق الغامض وهما المتحكم الغامض والمتحكم العصبى الغامض تم إقتراحهما فى هذه الدراسة وتمت مقارنة أداء هذة المتحكمات مع المتحكم التناسبى التكاملى التفاضلى .تم عرض نتائج المحاكاة وتحليلها لكافة المتحكمات .لوحظ أن المتحكم العصبى الغامض له الاستجابة الأفضل من بين المتحكمات الأخرى للتحكم فى وضع محرك الخدمة ذو التيار المستمر.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

AC	Alternative Current
ANFIS	Adaptive Neural Fuzzy Interface System
DC	Direct Current
e.m.f	Electro Motive Force
FIS	Fuzzy Interface System
FL	Fuzzy Logic
FLC	Fuzzy Logic Control
GUI	Graphical User Interface
HP	Horse Power
MF	Membership Function
PID	Proportional Integral Derivative

LIST OF SYMBOLS

Vf	Field voltage
Ωm	Angular velocity
Φ	Magnetic flux
I*f*	Field current
G	Transfer function
KP	Proportional gain
KI	Integration gain
Kd	Derivative gain
V	Voltage across the coil of the armature
Eb	Back emf electrical motion force
Ia	Rotor’s current
Ra	Armature resistant
La	Armature inductance
KM	Velocity constant
Kb	back electromotive force constant
TM	The electromagnetic torque
J	Moment of inertia of the rotor
Bm	Coefficient of friction
Θm	Angular position
Gc	Transfer function of DC motor
e ,Δe	Error change
ωn	Undammed natural frequency
ζ	Damping ratio
MP	Maximum overshoot
Ts	Settling time
Tr	Rise time

CHAPTER ONE
INTRODCTION

CHAPTER ONE

INTRODCTION

1.1 General Overview

Servomotor system consists of different mechanical and electrical components. The different components are integrated together to perform the function of the servomotor. DC servomotors have good torque and speed characteristics; also, they have the ability to be controlled by changing the voltage signal connected to the input. These Characteristics made them powerful actuators used everywhere. The main concern about DC servomotors is how to eliminate the non-linear characteristics that affect both the output speed and position. Another important non-linear behavior in servomotors is the saturation effect, in which the output of the motor cannot reach the desired value. The goal here is to find a smart controller that is capable of eliminating as much as possible from these non-linearties, so that to obtain a better controllability of servomotor drivers.

1.2 Problem Statement

The control position of DC servomotor problems with a conventional control algorithm is due to the effects of non-linearity of a DC servomotor. The non-linear characteristics that affect both the output speed and position of a DC servomotor such as saturation and friction could degrade the performance of conventional controllers. Conventional control strategies are of a fixed structure, fixed parameters design, so the tuning and optimization of these controllers is a challenging and difficult task, particular under varying load conditions, parameters change and abnormal models of operation.

1.3 Objectives

The main objectives of this study are to:

- ✓ Design position control of DC servomotor system using PID controller.
- ✓ Design position control of DC servomotor system using fuzzy logic.
- ✓ Design position control of DC servomotor system using neuro-fuzzy controller.

Comparison of the results of all proposed controllers is one of important objectives of the study.

1.4 Methodology

Study of all previous works.

- ✓ Descriptive analysis of DC servomotor.
- ✓ Mathematical analysis and computer modeling of DC servo motor.
- ✓ Design of PID, fuzzy neuro-fuzzy controllers using MATLAB toolbox.

1.5 Layout

This thesis consists of five chapters: Chapter one presents an introduction to the principles of the study, the reasons and motivation and also discusses the objectives and outline methodologies of the study. Chapter two discuses a theoretical background of DC servomotor, PID controller, fuzzy system, neural network and neuro-fuzzy controller. Chapter three presents the system control design of positon control of DC servomotor system. Chapter four presents the simulation results. Finally, Chapter five provides the conclusion and recommendations.

CHAPTER TWO

THEORETICAL BACKGROUND AND LITERATURE REVIEW

CHAPTER TWO

THEORETICAL BACKGROUND AND LITERATURE REVIEW

2.1 Introduction

Automatic systems are common place in people daily life, they can be found in almost any electronic devices and appliances we use daily, starting from air conditioning systems, automatic doors, and automotive cruise control systems to more advanced technologies such as robotic arms, production lines and thousands of industrial and scientific applications. DC servomotors are one of the main components of automatic systems; any automatic system should have an actuator module that makes the system to actually perform its function. The most common actuator used to perform this task is the DC servomotor. Historically, DC servomotors also played a vital role in the development of the computer's disk drive system; which make them one of the most important components in people life that we cannot live without it. Due to their importance, the design of controllers for these systems has been an interesting area for researchers from all over the world. However, even with all of their useful applications and usage, servomotor systems still suffer from several non-linear behaviors and parameters affecting their performance, which may lead for the motor to require more complex controlling schemes, or having higher energy consumption and faulty functions in some cases [2] .

2.2 DC Servomotor Systems

DC servomotors are DC motors that incorporate encoders and are used with controllers for providing feedback and closed-loop control. Specifically, servomotors provide precise motion and position control, accommodating complex motion patterns and profiles more readily than other types of motors. Often servomotors have better bearings or higher tolerance designs than traditional DC motors. Some designs also use higher voltages in order to achieve greater torque The basic parts of any DC servomotor are shown in Figure (2.1). [1].

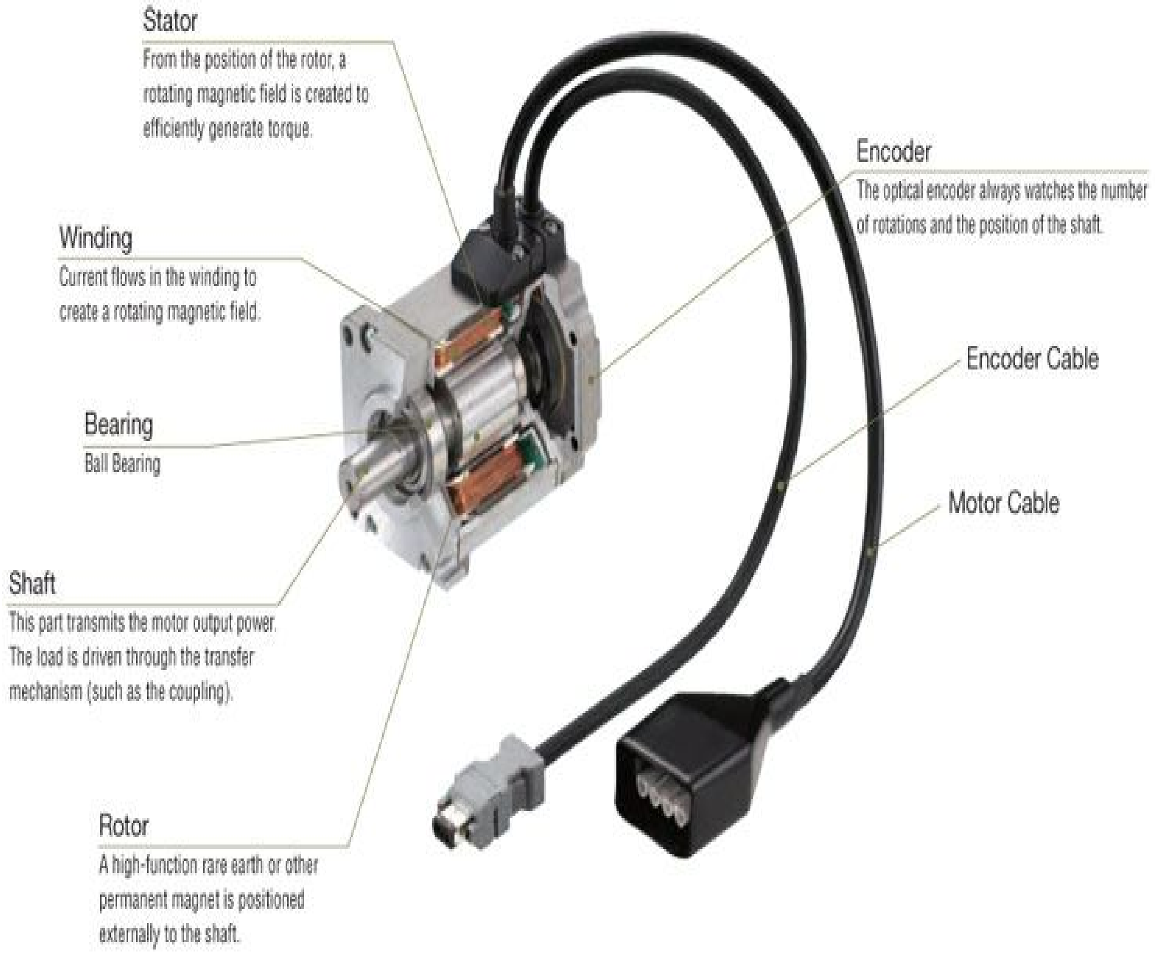

Stator
From the position of the rotor, a rotating magnetic field is created to efficiently generate torque.
Winding
Current flows in the winding to create a rotating magnetic field.
Bearing
Ball Bearing
Shaft
This part transmits the motor output power. The load is driven through the transfer mechanism (such as the coupling).
Rotor
A high-function rare earth or other permanent magnet is positioned externally to the shaft.
Encoder
The optical encoder always watches the number of rotations and the position of the shaft.
Encoder Cable
Motor Cable

Figure 2.1 DC servomotor

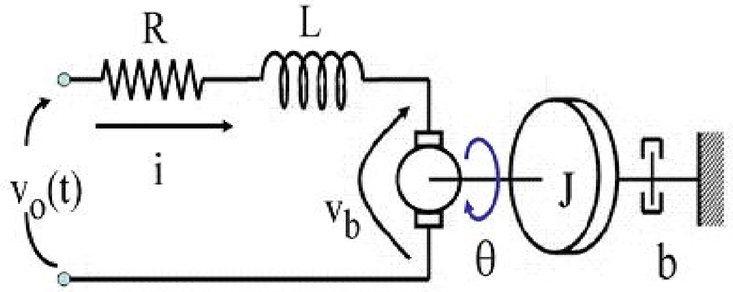
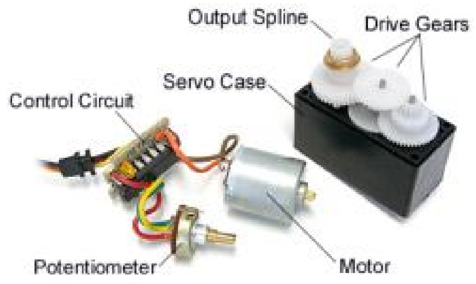


Figure 2.2 Servomotor circuit.

It's clear that the servomotor has two main components are shown in Figure (2.2). the first is the electrical component; which consists of resistance R, inductance L, input voltage $V_{in}(t)$ and the back electromotive force V_b. The second component of the servomotor is the mechanical part, from which we get the useful mechanical rotational movement is obtained at the shaft. The mechanical parts are the motor's shaft, inertia of the motor, and load inertia J and damping b. θ Refers to the angular position of the output shaft which can be used later to find the angular speed of the shaft ω DC Servomotors have good torque and speed characteristics; also they have ability to be controlled by changing the voltage signal connected to the input. These characteristics made them powerful actuators used everywhere. The main concern about DC servomotors is how to eliminate the non-linear characteristics that affect both the output speed and position. Another important non-linear behavior in servomotors is the saturation effect, in which the output of the motor cannot reach the desired value. The saturation effect is very common in almost all servomotor systems. Other non-linear effect is the dead zone; in which the motor will not start to rotate until the input voltage reaches a specific minimum value, which makes the response of the system slower and requires more controllability. A mathematical type of non-linear effect found in the servomotors is the backlash in the motor gears. Some of the servomotors use internal gears connections in order to improve their torque and speed characteristics, but this improvement comes over the effect in the output speed and position characteristics. The goal here is to find a smart controller that is capable of eliminating as much as possible from these non-linearties, so that we will have a better controllability of servomotor drives[3]

2.2.1 Construction of DC servo motor

To fully understand how the servo works, it need to take a look under the hood. Inside there is a pretty simple set-up: a small DC motor, potentiometer, and a control circuit. The motor is attached by gears to the control wheel. As the motor rotates, the potentiometer's resistance changes, so the control circuit can precisely regulate how much movement there is and in which direction. When the shaft of the motor is at the desired position, power supplied to the motor is stopped. If not, the motor is turned in the appropriate direction. The desired position is sent via electrical pulses through the wire signal as shown in Figure 2.3. The motor's speed is proportional to the difference between its actual position and desired position. So if the motor is near the desired position, it will turn slowly, otherwise it will turn fast. This is called proportional control. This means that the motor will only run as hard as necessary to accomplish the task at hand [3].



Heavy Duty Servo a servo motor (L) and an assembled servo (R)

Figure 2.3

2.2.2 DC servomotor model

Recalling the DC servomotor diagram from Figure 1.1, the transfer function of the DC servomotor can be derived using Kirchhoff's voltage law and laplace transforms [1]**.** as the following:

$$\boldsymbol{V_o = V_m = R_m\, i_m + L_m \frac{di}{dt} + V_b} \tag{2.1}$$

The Back-Electromotive Force (EMF) can be found by using the equation:

$$\boldsymbol{V_b = K_m \frac{d\theta}{dt} = K_m} \tag{2.2}$$

Where Vb is the induced voltage, Km is the motor torque constant, and ωm is the angular rotating speed. It can be seen that ωm can be calculated by the Equation (2.3)

$$\boldsymbol{\omega_m = \theta^{\cdot}} \qquad (2.3)$$

And Using Laplace transform

$$\boldsymbol{\omega_{(s)} = s\theta(s)} \qquad (2.4)$$

the concern in this stage is to control the angular rotating speed ωm by controlling the input voltage Vm.

Where:

J = moment of inertia of the rotor

b = dampening ratio of the mechanical system

T = motor torque

I = Current

Vm = back emf

θ = shaft position

K = electromotive force constant

ωm = Measured angular Speed

R = Motor Armature Resistance

L = Inductance

V = Source Voltage

The transfer function of the output angular speed is derived using Laplace transform using the second order system equation:

$$G(S) = \frac{\omega_n}{S^2 + 2\xi\omega s + \omega^2} \qquad (2.5)$$

The resulting transfer function:

$$G(S) = \frac{K}{(JS+b).(Ls+R)+K^2} \qquad (2.6)$$

From Equation 2.4, the relationship between the angular position and the speed can be found by multiplying the angular position by$1/s$. Our major concern on this research is the proper control of the angular speed of the motor; since the angular speed is the part that suffers the most from the non-linearties. The angular non-linear effect on the angular position tends to be less due to the term used to derive it $1/s$, which adds an integral effect or filter effect to this part. Figure 2.4 shows the block diagram of servomotor system using MATLAB SIMULINK.

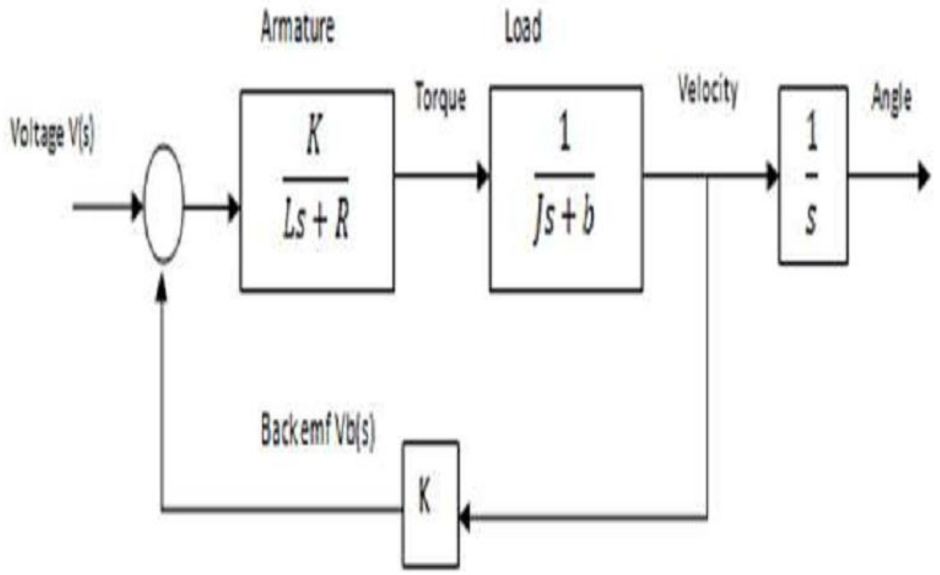


Figure 2.4 Servomotor SIMULINK block diagram.

2.2.3 Servo Motor Control

Servos are controlled by sending an electrical pulse of variable width, or pulse width modulation (PWM), through the control wire. There is a minimum pulse, a maximum pulse and a repetition rate. A servo motor can usually only turn 90 degrees in either direction for a total of 180 degree movement. The motor's neutral position is defined as the position where the servo has the same amount of potential rotation in the both the clockwise or counter-clockwise direction. The PWM sent to the motor determines position of the shaft, and based on the duration of the pulse sent via the control wire; the rotor will turn to the desired position. The servo motor expects to see a pulse every 20 milliseconds (ms) and the length of the pulse will determine how far the motor turns. For example, a 1.5ms pulse will make the motor turn to the 90-degree position. Shorter than 1.5 ms moves it to 0 degrees, and any longer than 1.5 ms will turn the servo to 180 degrees, as shown in Figure 2.5

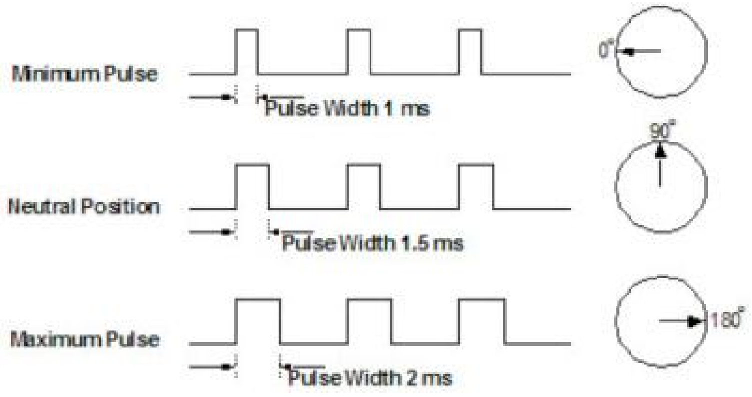


Figure (2.5)Variable Pulse width control servo position

When these servos are commanded to move, they will move to the position and hold that position. If an external force pushes against the servo while the servo is holding a position, the servo will resist from moving out of that position. The maximum amount of force the servo can exert is called the torque rating of the servo. Servos will not hold their position forever though; the position pulse must be repeated to instruct the servo to stay in position [4].

2.3 PID Controller Overview

PID is an acronym for the mathematical terms *Proportional, Integral,* and *Derivative.* Proportional means a constant multiple. A number is said to be a proportion to another if there exists a constant n such that $y = nx$. This n can be positive or negative, greater or less than one. To make the formula more accurate by PID controller standards, proportion is given by K_P and the x term is the control loop error e: y = K_P(e). The term *Integral* means the summation of a function over a given interval. In the case of controller PID that is the sum of error over time: $y = \int f\,(\mathrm{e})dt$. Finally, Derivative is the rate of change during a given interval. Interpreted by a PID controller: All three of these PID controller components create output based on measured error of the process being regulated as in Figure (2.6). If a control loop functions properly, any changes in error caused by set point changes or process disturbances are quickly eliminated by the combination of the three factors P, I, and D.

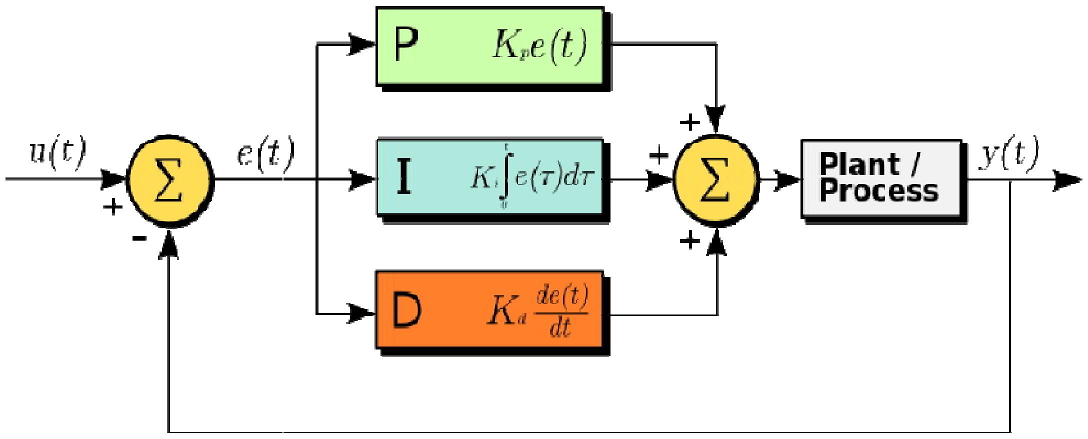


Figure (2.6). PID controller

In the field of process control systems, it is well known that the basic and modified PID control schemes have proved their usefulness in providing satisfactory control, although in many given situations they may not provide optimal control.first the design of a PID controlled system is presented using Ziegler - Nichols tuning rules.The PID controller calculation (algorithm) involves three separate constant parameters, and is accordingly sometimes called three-term control: the proportional, the integral and derivative values.

Heuristically, these values can be interpreted in terms of time: P depends on the present error, I on the accumulation of past errors, and D is a prediction of future errors, based on current rate of change. The weighted sum of these three actions is used to adjust the process via a control element such as the position of a control valve, or the power supplied to a heating element. If both the transient and steady-state response of the system must be improved, then neither a PI nor a PD controller may meet the desired specifications. Adding a zero PD May improve the transient response but does not increase the type number of the system. Adding a pole at the origin increases the type number but may yield an unsatisfactory time response even if one zero is also added. With PID controller, two zeros and a pole at the origin are added. This both increases the type number and allows satisfactory reshaping of the root locus.The transfer function of a PID controller is given by equation (2.7) and (2.8):

$$C(S) = K_{P+}\frac{K_i}{s} + K_{ds=K_d}\frac{\boldsymbol{S^2} + \mathbf{2}\boldsymbol{\xi}\omega s + \omega^{\mathbf{2}}}{\mathrm{s}} \qquad (2.7)$$

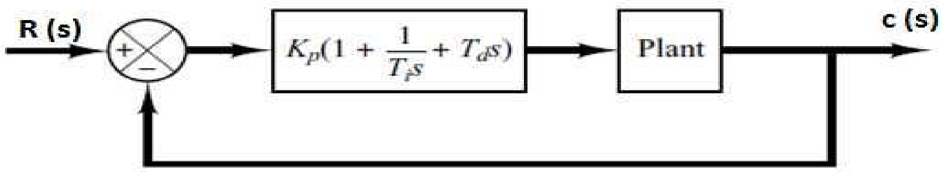
$$\mathbf{2}\boldsymbol{\xi}\omega_{n=}\frac{K_p}{K_d}, \omega_n{}^{\mathbf{2}} = \frac{K_i}{K_d} \qquad (2.8)$$

Where K_P, K_i, and K_d are the proportional, integral, and derivative gain, respectively [11]. Most PID controllers are adjusted on-site, many different types of tuning rules have been proposed in the literature. Using these tuning rules, delicate and fine tuning of PID controllers can be made on-site. Also, automatic tuning methods have been developed and some of the PID controllers may possess on-line automatic tuning capabilities. The usefulness of PID controls lies in their general applicability to most control systems [5].

2.3.1 Ziegler Nichols method for tuning PID controller

In particular, when the mathematical model of the plant is not known and therefore analytical design methods cannot be used, PID controls prove to be most useful. In the field of process control systems, it is well known that the basic and modified PID control schemes have proved their usefulness in providing satisfactory control, although in many given situations they may not. This section presents the design of a PID controlled system using Ziegler and Nichols tuning rules [6].

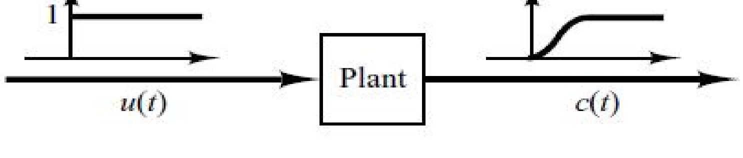
If a mathematical model of the plant can be derived, then it is possible to apply various design techniques for determining parameters of the controller that will meet the transient and steady-state specifications of the closed-loop system Figure 2.7. However, if the plant is so complicated that its mathematical model cannot be easily obtained, then an analytical or computational approach to the design of a PID controller is not possible. Then I must resort to experimental approaches to the tuning of PID controllers. The process of selecting the controller parameters to meet given performance specifications is known as controller tuning. Ziegler and Nichols suggested rules for tuning PID controllers (meaning to set values Kp, T_i, and T_d) based on experimental step responses or based on the value of Kp that results in marginal stability when only proportional control action issued. Ziegler–Nichols rules, which are briefly presented in the following, are useful when mathematical models of plants are not known. (These rules can, of course, be applied to the design of systems with known mathematical models.)Such rules suggest a set of values of Kp, Ti, and Td that will give a stable operation of the system. However, the resulting system may exhibit a large maximum overshoot in the step response, which is unacceptable. In such a case we need series of fine tuning sunlit an acceptable result is obtained. In fact, the Ziegler–Nichols tuning rules give an educated guess for the parameter values and provide a starting point for fine tuning, rather than giving the final settings for K_p, T_i, and T_d in a single shot.



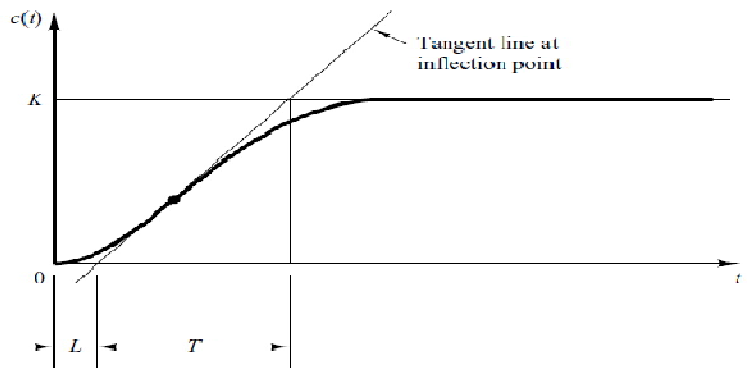
Figure(2.7):PID control of a plant

Ziegler and Nichols proposed rules for determining values of the proportional gain Kp integral time Ti and derivative time Kd based on the transient response characteristics of a given plant. Such determination of the parameters of PID controllers or tuning of PID controllers can be made by engineers on-site by experiments on the plant. (Numerous tuning rules for PID controllers have been proposed since the Ziegler– Nichols proposal. They are available in the literature and from the manufacturers of such controllers.)There are two methods called Ziegler–Nichols tuning rules: the first method and the second method. We shall give a brief presentation of these two methods. First method, the response is obtained experimentally of the plant to a unit-step input, as shown in figure (2.8). If the plant involves neither integrator (s) nor dominant complex-conjugate poles, then such a unit-step response curve may look S-shaped, as shown in figure (2.9). This method applies if the response to a step input exhibits an S-shaped curve. Such step response curves may began rated experimentally or from a dynamic simulation of the plant. The S-shaped curve may be characterized by two constants, delay time Land time constant T.The delay time and time constant are determined by drawing a tangent line at the inflection point of the S-shaped curve and determining the intersections of the tangent line with the time axis and line c(t)=K, as shown in Figure (2.8). The transfer Function C(s)/U(s) may then be approximated by a first-order system with a transport Lag as equation

$$\frac{C(S)}{U(S)} = \frac{Ke^{-LS}}{T_S+1} \tag{2.9}$$



Figure(2.8):Unit-step response of a plant



Figure(2.9):S-shaped response curve

Ziegler and Nichols suggested setting the values of Kp, Ti and T formula shown in Table (2-1).

Table (2–1): Ziegler–Nichols Tuning Rule Based

Type of Controller	K_p	T_i	T_d
P	$\frac{T}{L}$	∞	0
PI	$0.9\frac{T}{L}$	$\frac{L}{0.3}$	0
PID	$1.2\frac{T}{L}$	$2L$	$0.5L$

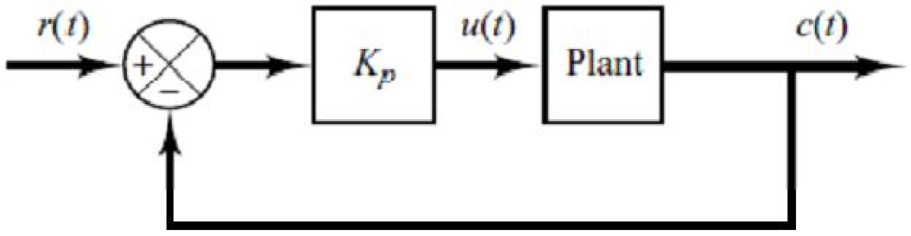
Notice that the PID controller tuned by the first method of Ziegler–Nichols rules gives (2.3)

$$G_C(S) = K_P\left(1 + \frac{\mathbf{1}}{T_{iS}} T_{ds}\right)$$

$$1.2\frac{\boldsymbol{T}}{\mathrm{L}}\left(\mathbf{1} + \frac{\mathbf{1}}{\mathbf{2}L_S} + .\mathbf{5}L_S\right) \qquad (2.10)$$

$$0.6T\frac{(S+\frac{\mathbf{1}}{L})^{\mathbf{2}}}{S}$$

Thus, the PID controller has a pole at the origin and double zeros at s= –1/L. Second method, first set Ti = ∞ and Td= 0. Using the proportional control action only (see figure(2.10), increase Kp from 0 to a critical value Kcr at which the output first exhibits sustained oscillations. If the output doesnot exhibit sustained oscillations for whatever value Kp may take, then this method does not apply. Thus, the critical gain Kcr and the corresponding period Pcr are experimentally determined as shown in Figure 2.11



Figure(2.10) closed-loop system with a proportional controller

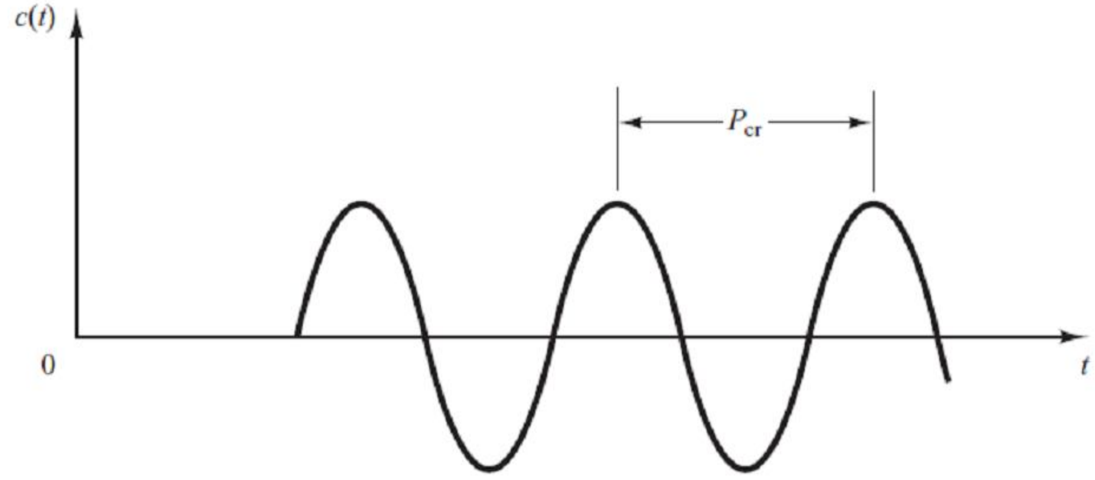


Figure (2.11) Sustained oscillation with period pcr (pcr is measured in sec.)

Ziegler and Nichols suggested that set the values of the parameters Kp, Ti, and Td according to the formula shown in Table (2.2).

• Notice that the PID controller tuned by the second method of Ziegler Nichols rules gives(2.4)

$$Gc(s) = K_p \left(1 + \frac{1}{T_i s} + T_d s\right)$$

$$Gc(s) = .075\ K_{cr}\ P_{cr} \frac{\left(S + \frac{4}{P_{cr}}\right)^2}{S} \qquad (2.11)$$

$$Gc(s) = 0.6K_{cr} \left(1 + \frac{1}{0.5P_{cr}S} + 0.125P_{cr}s\right)$$

Table (2.2): Ziegler–Nichols tuning rule based on critical gain K_{cr} and critical period Pcr (Second method)

Type of Controller	K_p	T_i	T_d
P	$0.5K_{cr}$	∞	0
PI	$0.45K_{cr}$	$\frac{1}{1.2}P_{cr}$	0
PID	$0.6K_{cr}$	$0.5P_{cr}$	$0.125P_{cr}$

Thus, the PID controller has a pole at the origin and double zeros at s=(-4)/$\boldsymbol{Pcr}$, Note that if the system has a known mathematical model (such as the transfer function), then we can use the root frequency of the sustained oscillations w found from the crossing points of the rootif the root-locus branches do not cross the Nichols tuning rules (and other tuning rules presented in the literature) have been widely used to tune PID controllers in process control systems where the plant dynamics are not precisely known. Over many years, such tuning rules proved to bvery useful. Ziegler–Nichols tuning rules can, of course, be applied to plants whose dynamics are known. (If the plant dynamics are known, many analytical and graphical approaches to the design of PID controllers are available, in addition to Ziegler - Nichols tuning rules [6].

2.4 Fuzzy Logic

One of the more popular new technologies is “intelligent control,”which is defined as a combination of control theory, operations research, and Artificial Intelligence (AI). Judging by the billions of dollars worth of sales and thousands of patents issued worldwide, led by Japan since the an noun cement of the first fuzzy chips in 1987,fuzzy logics still perhaps the most popular area in AI.

To understand fuzzy logic, it is important to discuss fuzzy sets. In1965 ,Zadeh wrote a seminal paper in which he introduced fuzzy sets, that is, sets with un sharp boundaries. These sets are generally in better agreement with the human mind and reasoning that works with shades of gray, rather than with just black or white. Fuzzy sets are typically able to represent linguistic terms, for example, warm, hot, high, low, close, far, etc. Nearly 10 years later (in 1974), Mamdani succeeded in applying fuzzy logic for control in practice. Today, in Japan, United States, Europe, Asia, and many other parts of the world, fuzzy control is widely accepted and applied.

Anew logic system based on the premises of fuzzy sets is known as fuzzy logic. The need and use of multi level logic can be traced from the ancient works of Aristotle ,who is quoted as saying“ There will be a sea battle tomorrow.”Such a statement is not yet true or false, but is potentially either. Much later, around AD 1285 – 1340 William of Occam supported two-valued logic but speculated on what the truth value of “if *p* then *q* ”might be if one of the two components ,*p* or *q* ,as neither true nor false .During the period of 1878–1956, Luck as irenics proposed three–level logic as a “true” (1), a “false” (0), and a “neuter”(1/2),which represented half-true or half-false.In subsequent times, logician sin China and other part soft he world continued on the notion of multi level logic. Zadeh ,in his seminal 1965 paper ,finished the task by following through with the speculation of previous logicians and showing that what he called “fuzzy sets ”was the foundation of any logic, regardless of the number of truth levels assumed .He chose the in no cent word “fuzz” for the continuum of logical values between 0 (completely false) and 1 (completely true).The theory of fuzzy logic deals with two problems:

- The fuzzy set theory, which deals with the vagueness found in semantics, and
- The fuzzy measure theory ,which deals with the ambiguous nature of judgments and evaluations.

The primary motivation and “banner” of fuzzy logics the possibility of exploiting to lerance for some inexactness and imprecision. Precision is often very costly, so if a problem does not warrant great precision, one should not have to pay for it. The traditional example of parking a car is a note worthy illustration. If the driver is not required to park the car with in an exact distance from the curb, why spend any more time than necessary on the task as long as it is a legal parking operation? Fuzzy logic and classical logic differ in the sense that the former can handle both symbolic and numerical manipulation, where as the latter can handle symbolic manipulation only.

In a broad sense, fuzzy logic is a union of fuzzy (fuzzified) crisp logics. To quote Zadah, “Fuzzy logic’s primary aim is to provide a formal, computationally-oriented system of concepts and techniques for dealing with modes of reasoning which are approximate rather than exact.”Thus, in fuzzy logic, exact(crisp) reasoning is considered to be the limiting case of approximate reasoning. In fuzzy logic, one can see that everything is a matter of degrees [7].

2.4.1 Fuzzy sets and conventional sets

First proposed at 1965 and based on the concept of fuzzy Sets, fuzzy set theory Provides means for representing uncertainty. Probability theory is the primary tool for analyzing uncertainty and assumes that the uncertainty is a random process. However Uncertainty is not always random though and fuzzy set theory is used to model the kind Of   uncertainty associated with imprecision, and lack of information. Conventional set theory distinguishes between those elements that are members of a set and those are not, there being very, clear or crisp boundaries. Figure (2.12) shows the crisp set“medium temperature”. Temperatures   between 20 and 30 Clie with in crisp set, and have a member ship value of one.

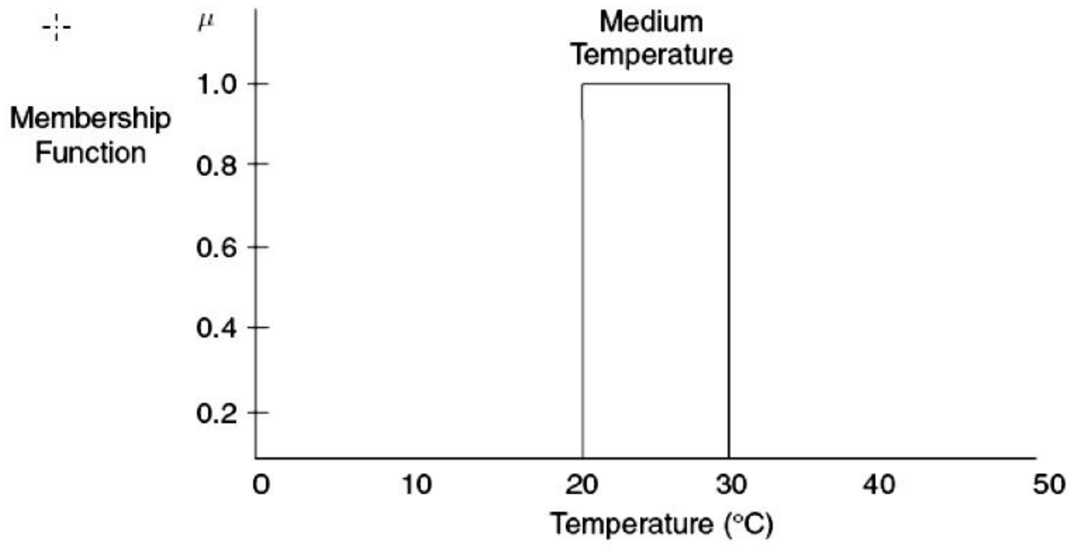


Figure (2.12) crisp set

The central concept of fuzzy set theory is that the membership function probability theory, can have value of between 0 and1. In function μ has linear relationship with the x This produces a triangular shaped fuzzy set.triangles are commonly used because they give good results and computation is simple. Other arrangement includes non -symmetrical triangles, trapezoids, and Gaussian.Let the fuzzy set (see Figure(2.13)) "medium temperature" be called fuzzy set M. If an element u of the universe of discourse U lies within fuzzy set M, it will have a value of between 0 and1

1. This is expressed as

$$\mu M \varepsilon [0, 1] \qquad (2.12)$$

When the universe of discourse is discrete and finite, fuzzy set M may be expressed as equation 2.12

$$M = \sum_{i=1}^{N} \frac{\mu M(ui)}{ui} \qquad (2.13)$$

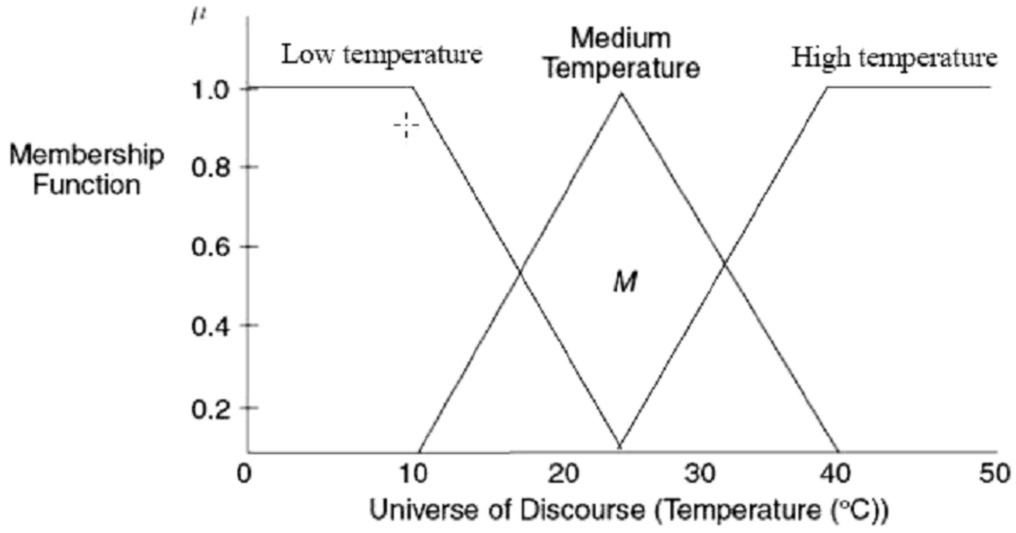


Figure (2.13): Fuzzy set.

2.4.2 Operations on fuzzy sets

Let A and B be two fuzzy sets within a universe of discourse function μ A ε [0,1] and μB ε [0,1] respectively. The following fuzzy set operations can be defined

2.4.2.1 Equality

Two fuzzy sets A and B are equal if they have the same membership function within a universe of discourse Uμ A(u) =μB u (u),∀ u ε U (2.14)

2.4.2.2Unio

The union of two fuzzy sets A and B corresponds to Boolean OR function and is given

by:

μ A∪B(u) = max[μ A(u), μ B(u)],∀ ulU (2.15)

2.4.2.3 Intersection

The intersection of two fuzzy sets A and B corresponds to the Boolean AND function and is given by:

$xA \cap B(u) = \min[xA(u), xB(u)], \forall$ ulU (2.16)

2.4.2.4 Complement

The complement of fuzzy set A corresponds to the Boolean NOT function and is given by[13]:

$x\ \text{-}A(u) = 1\text{-}xA(u), \forall$ ulU (2.17)

2.4.3 Fuzzy relations

Many application problem descriptions include fuzzy relations. For example, to describe a plant or a control system one determines, how an output(s) depends on inputs, or the relationship between outputs and inputs. If one constructs a database and an information system, one determines the relations between different at tributes. To model a fuzzy system one uses rules like if speed is slow then pressure should be high If the speed is denoted as variable A and pressure as variable B then one will have in a general case the rule: if A then B [8].

2.4.4 Linguistic variables

To specify rules for the rule-base, the expert will use a "linguistic description"; hence, linguistic expressions are needed for the inputs and outputs and the characteristics of the inputs and outputs. "linguistic variables" is used (constant symbolic descriptions of what are in general time-varying quantities) to describe fuzzy system inputs and outputs. For our fuzzy system, linguistic variables denoted by ˜ui are used to describe the inputs ui. Similarly, linguistic variables denoted by ˜yi are used to describe outputs yi. For instance, an input to the fuzzy system may be described as ˜u1 ="position error" or ˜u2="velocity error," and an output from the fuzzy system may be ˜y1="voltage in" [9].

2.4.5 Fuzzy control system design

Fuzzy control provides a formal methodology for representing, manipulating, and implementing a human's heuristic knowledge about how to control a system. In this section we seek to provide a philosophy of how to approach the design of fuzzy controllers.

The Fuzzy controller block diagram is given in Figure (2.14), where a fuzzy controller embedded in a closed-loop control system. The plant outputs are denoted by y(t), its inputs are denoted by u(t), and the reference input to the fuzzy controller is denoted by r(t).

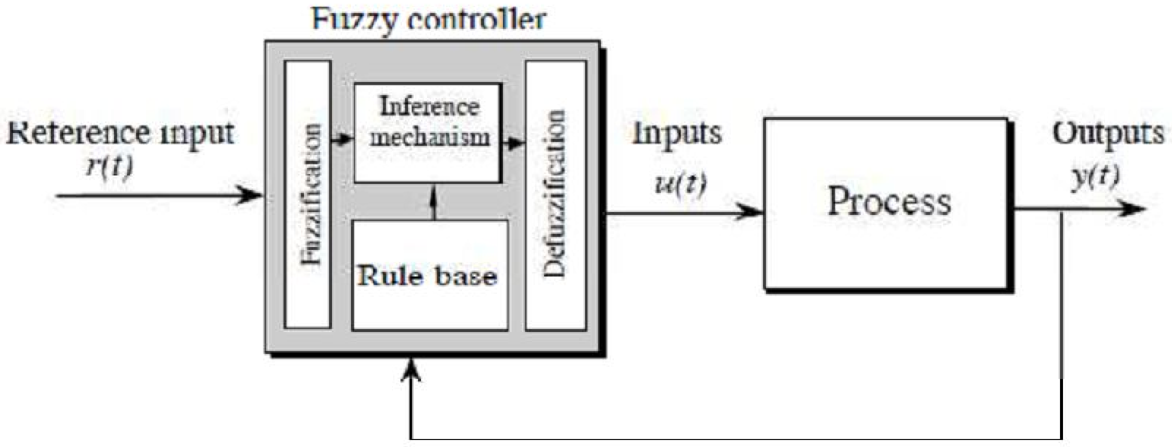


Figure (2.14): Fuzzy controller architecture

The fuzzy controller has four main components:

- The "rule-base" holds the knowledge, in the form of a set of rules, of how best to control the system.
- The inference mechanism evaluates which control rules are relevant at the current time and then decides what the input to the plant should be.
- The fuzzification interface simply modifies the inputs so that they can be interpreted and compared to the rules in the rule-base.

- The defuzzification interface converts the conclusions reached by the inference mechanism into the inputs to the plant.

Basically, the fuzzy controller is viewed as an artificial decision maker that operates in a closed-loop system in real time. It gathers plant output data y (t), compares it to the reference input r (t), and then decides what the plant input u(t) should be to ensure that the performance objectives will be met.

To design the fuzzy controller, the control engineer must gather information on how the artificial decision maker should act in the closed-loop system. Sometimes this information can come from a human decision maker who performs the control task, while at other times the control engineer can come to understand the plant dynamics and write down a set of rules about how to control the system without outside help. These "rules" basically say, "If the plant output and reference input are behaving in a certain manner, then the plant input should be some value". A whole set of such "If-Then" rules is loaded into the rule-base, and an inference strategy is chosen, then the system is ready to be tested to see if the closed-loop specifications are met [8].

2.5 Neural Network

It is well known that biological systems can perform complex tasks without recourse to explicit quantitative operations. In particular, biological organisms are capable of learning gradually over time. This learning capability reflects the ability of biological neurons to learn through exposure to external stimuli and to generalize. Such properties of nervous systems make them attractive as computation models that can be designed to process complex data. For example, the learning capability of biological organisms from examples suggests possibilities for machine learning [9].

2.5.1 Neuron model

The multilayer perception neural network is built up of simple components. The single input neuron beginner, then it extends to multiple inputs. Next stack these neurons together to produce layers. Finally, The layers has been cascaded together to form the network.

2.5.2 Single-input neuron

A single-input neuron is shown in Figure (2.15) the scalar input P is multiplied by the scalar weight w to form wp, one of the terms that is sent to the summer. The other bias input, 1, is multiplied by a bias b and then passed to the summer. The summer output n, often referred to as the net input, goes into a transfer function f, which produces the scalar neuron output a. (Some authors use the term "activation function" rather than transfer function and "offset" rather than bias.) The neuron output is calculated as:

$$a = f(wp + b) \tag{2.18}$$

Note that w and b are both adjustable scalar parameters of the neuron. Typically the transfer function is chosen by the designer and then the parameters w and b will be adjusted by some learning rule so that the neuron input/output relationship meets some specific goal.

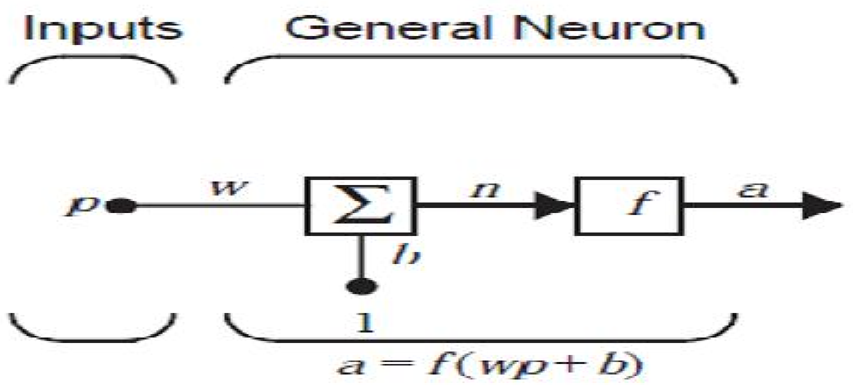


Figure (2.15) Single - input neuron

The transfer function may be a linear or a nonlinear function. A particular transfer function is chosen to satisfy some specification of the problem that the neuron is attempting to solve [10].

2.5.3 Neural control

Neural control refers both to a methodology in which the controller itself is a neural network, and to a methodology in which controllers are designed based on a neural network model of the plant. These two basically different approaches for implementing neural networks in control are referred to as direct and indirect design methods.

Fuzzy control is a control method relying on perception- based information expressed in fuzzy logic. This is the case where the available data is in the form of a collection of linguistic .If. . . then. . . . rules. In other words, fuzzy control is amathematical method for implementing control strategies expressed in a natural language. This situation arises mostly in the control of complex systems, a situation that human operators handle well and for which natural language is an appropriate means for describing control strategies. As its name indicates, neural control refers to another control method when available data are in the form of measurements (observed numerical data) of the plant's behavior. This is the case where information is only in the form of system behavior, either of the real plant or of its simulated model, expressed as input-output measurements. In view of the generality of neural networks as function approximation devices, it is natural to use neural networks in control situations such as this. Specifically, when mathematical models of the plant dynamics are not available, neural networks can provide a useful method for designing controllers, provided we have numerical information about the system behavior in the form of input-output data. In other words, a neural network can be used as a "black box" model for a plant. Also, controllers based on neural networks will benefit from neural networks' learning capability that is suitable for adaptive control where controllers need to adapt to changing environment, such as for time-variant systems. In practice, neural network controllers have proved to be most useful for time-invariant systems. Basically, to build a neural network-based controller that can force a plant to behave in some desirable way, it needs to adjust parameters from the observed errors that are the difference between the plant's outputs and the desired outputs. Adjustment of the controller's parameters will be done by propagating back these errors across the neural network structure. This is possible if the mathematical model of the plant is known. When the mathematical model of the plant is not known, it needs to know at least an approximate model of the plant in order to do the above. An approximate (known) model of the plant is called an identified model. When we use input-output data from the plant to train a neural network to provide an approximate model to the plant, the neural network identified model of the plant obtaina. Neural network identified models are used in indirect neural control designs. After a general discussion of inverse dynamics, we will first discuss direct neural control designs and then indirect control.

2.5.4 Neural networks in direct neural control

Direct design means that a neural network directly implements the controller that is, the controller is a neural network as in Figure 2.16. The network must be trained as the controller according to some criteria, using either numerical input-output data or a mathematical model of the system.

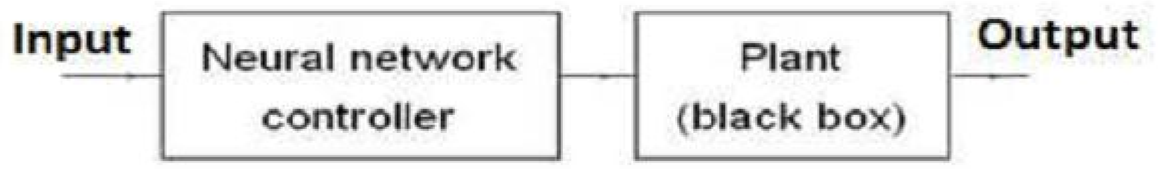


Figure 2.16: Direct design

A natural question that arises in this type of neural control is the selection of the type of neural network needed for the controller. We have seen from the previous chapter on neural networks that there are several types of neural network architectures. Multi-Layered Perceptron (MLP) neural networks are composed of configurations of simple perceptrons in a hierarchical structure forming a feed forward network. They have one or more hidden layers of perceptrons between the input and output layers. It is permissible to have any prior layer nodes connected to subsequent layer nodes via a corresponding set of weights. Different learning algorithms can be used for MLPs, but the most common ones have been the delta rule and error-back propagation algorithms. These algorithms do work fairly well but they tend to be slow. Faster and more efficient algorithms have been developed [8, 20, 32, 37], and ongoing research is continually discovering further improvements.

2.5.6 Neural networks in indirect neural control

Indirect neural control design is based on a neural network model of the system to be controlled. In this case, the controller itself may not be a neural network, but it is derived from a plant that is modeled by a neural network. This is similar to standard control in that a mathematical model is needed, but here the mathematical model is a neural network. Indirect neural control designs involve two phases. The first phase consists of identifying the plant dynamics by a neural network from training data that is, system identification. In the second phase, the control design can be rather conventional even though the controller is derived, not from a standard mathematical model of a plant, but from its neural network identified model. Since the identified neural network model of the plant is nonlinear, one way to design the controller is to linearize its identified neural network model and apply standard linear controller designs. Another way is through .instantaneous linearization [9].

2.6 Neuro-Fuzzy Systems

Both neural networks and fuzzy system are motivated by imitating human reasoning process. It utilizes human expertise. In fuzzy systems, relationships are represented explicitly in the form of the if-then rules. In neural networks, the relations are not explicitly given, but are encoded in the networks and parameters designed. Neurofuzzy systems combine semantic transparency of rule-based fuzzy systems with a learning capability of neural networks [11].

2.6.1 Adaptive network fuzzy inference systems

To illustrate the use of neural networks for fuzzy inference, present some successful Adaptive Neural Network Fuzzy Inference Systems (ANFIS), along with training algorithms known as ANFIS. These structures, also known as adaptive neuro-fuzzy inference systems or adaptive network fuzzy inference systems, were proposed by Jang. It should be noted that similar structures were also proposed independently by Lin and Lee and Wang and Mendel. These structures are useful for control and for many other applications [9].

2.6.2 Neuro – fuzzy controller

The neural predictive controller can be extended with Neuro-fuzzy controller, connected in parallel as shown in Figure 2.17. Neuro-fuzzy systems, which combine neural networks and fuzzy logic, have recently gained a lot of interest in research and application. A specific approach in neuro-fuzzy development is the Adaptive Neural Network Fuzzy Inference Systems (ANFIS).

ANFIS uses a feed forward network to search for fuzzy decision rules that perform well on a given task. Using a given input output data set, ANFIS creates an Fuzzy inference system for which membership function parameters are adjusted using a combination of a back propagation and least square method [12].

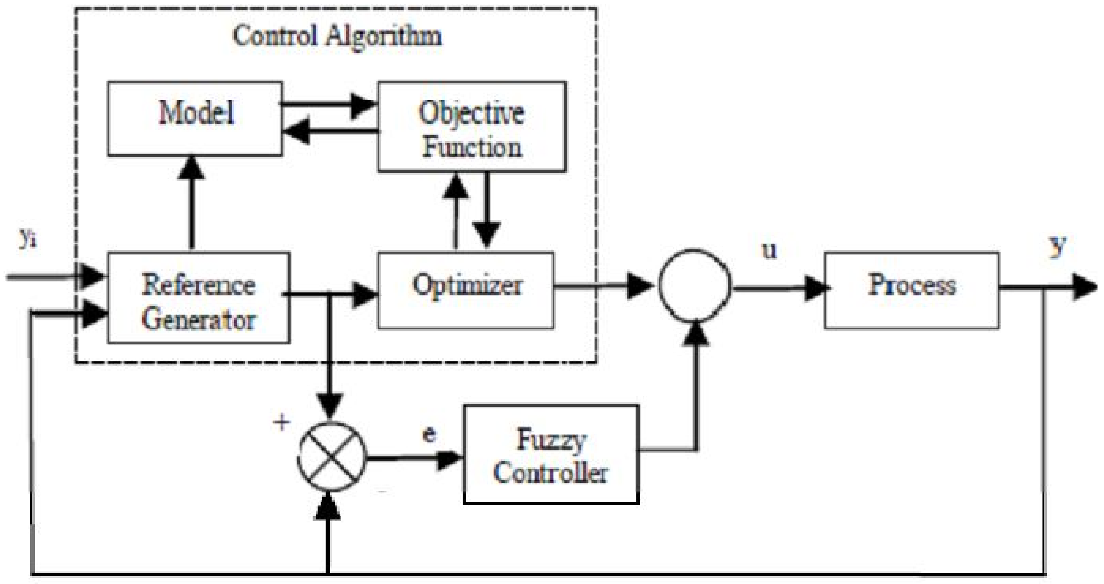


figure 2.17 Neuro – fuzzy control scheme

2.6.3 ANFIS as an estimator

ANFIS can be used for the estimation of some dependent variables in chemical process. The designed ANFIS estimator is used to infer the compositions from measurable tray temperatures distillation column. In estimator design process, different ANFIS structure are constructed and trained to find the architecture that gives the best performance as an estimator. As a first step to design an estimator, training data sets should be generated to train the estimator networks. These data sets consist of estimator inputs and desired output values. They are produced from the process input - output data. Since, ANFIS is a data processing method, it is important that the input - output data must be within the sufficient operational range including the maximum and minimum values for both input and output variables of the system. If this is not provided, estimator performance cannot be guaranteed and thus the designed estimator will not be accurate. Having generated the training data, estimators that have different architectures are trained with the obtained data sets. Performances of the trained estimators are evaluated with model simulations and best estimator architecture is obtained. These simulations are made to verify and to generalize the ANFIS structures. Verification is done to show how good the estimator structure learned the given training data. This is carried out by simulating the column models with specific initial process inputs used in obtaining training data sets. Generalization capabilities of the estimators are found with other simulations in which input process variables are in operational range but not used in training data formation. ANFIS estimator design consists of two parts: constructing and training. In constructing part, structure parameters are determined. These are type and number of input Membership Functions (MFs), and type of output MF. Any of several MFs such as Triangular, Trapezoidal and Gaussian can be used as an input MF. Frequently used MFs in literature are the Triangular and Gaussian. For this reason, they are chosen as input MF type in this study. Number of MFs on each input can be chosen as 3, 5, and 7 to define the linguistic labels significantly. Effective partition of the input space is important and it can decrease the rule number and thus increase the speed in both learning and application phase. Output MFs can be either a constant or in linear form. Both of these two forms are used for the output MF in this study. Having described the number and type of input MFs, the estimator rule base is constituted. Since, there is no standard method to utilize the expert knowledge; automatic rule generation method is usually preferred. According to this method, for instance, an ANFIS model with two inputs and five MFs on each input would result in $5^2= 25$ Takagi-Sugeno fuzzy if-then rules automatically. Although this method can require much computational knowledge especially in systems that have to be defined with many inputs, it is used in this study due to advantage of MATLAB software. Therefore, rule bases of the estimators are formed automatically with the number of inputs and number of MFs. After the ANFIS structure is constructed, learning algorithm and training parameters are chosen. As mentioned in the earlier in this chapter, back propagation or hybrid learning can be used as a learning algorithm. The hybrid learning algorithm is used in this study. Parameters in the algorithm are epoch size (presentation of the entire data set), error tolerance, initial step size, step size decrease rate, and step size increase rate. Since there is no exact method in literature to find the optimum of these parameters a trial and error procedure is used. MATLAB fuzzy logic toolbox is used to design ANFIS estimators' structures. Using the given training data set, the toolbox constructs an ANFIS structure using either a back propagation algorithm alone, or in combination with least squares type of method (hybrid algorithm). ANFIS model can be generated either from the command line, or through the ANFIS editor GUI. In this study, ANFIS Editor GUI is used to generate the ANFIS models with the chosen design parameters in construction phase. Written MATLAB code is used to train the ANFIS structure in the training step. The steps in ANFIS estimator design in this study utilizing the MATLAB fuzzy logic toolbox are as follows:

- Generated training data is loaded to the editor GUI.
- Design parameters, number of input MF, type of input and output MF, are chosen. Thus, initial ANFIS structure is formed.
- The code for the training is run with the initial structure.
- ANFIS structure constituted after training is saved to use as an estimator [11].

CHAPTER THREE

CONTROL SYSTEM DESIGN OF DC SERVOMOTOR

CHAPTER THREE

CONTROL SYSTEM DESIGN OF DC SERVOMOTOR

3.1 Introduction

Many textbooks have been written on control engineering, describing new techniques for controlling systems, or new and better ways of mathematically formulating existing methods to solve the ever-increasing complex problems faced bypracticing engineers. However, few of these books fully address the applicationsaspects of control engineering. It is the intention of this new series to redress this Situation.Theposition of DC servomotor can be adjusted to a great extent as to provide controllabilityeasy and high performance. The controllers of the position that are conceived for goal tocontrol the position of DC servomotor to execute one variety of tasks, is of several conventionaland numeric controller types, the controllers can be: PID controller, fuzzy logiccontroller; or the combination between them: Fuzzy-neural networks, fuzzy-geneticalgorithm, fuzzy-ants colony, fuzzy-swarm [12].

3.2 Motor's Parameters

The motor used in this study is DC separately excited and the motor's parameters are as follows [2]:

Armature resistance (Ra) = 0.6 Ω

Armature inductance (La) = 8 mH

Back e.m.f constant (Kb) = torque constant (Ki) = 0.55 V/rad/s

Mechanical inertia (J) = 0.0465 kg.m2

Friction coefficient (B) = 0.004 N.m/rad/s

$$G(S) = \frac{\theta(s)}{v(s)} = \frac{K}{\boldsymbol{s}((JS + b).(Ls + R) + K^2)} \frac{rad}{v}$$

By substituting above parameters in equation (3.1), then

$$Gc(s) = \frac{\theta(s)}{v(s)} = \frac{.55}{0.000372S3 + .02793S2 + .3049S} 3.1$$

3.3 PID Controller Design

By substituting the parametersof equation 3.1in MATLAB Simulinktechniques toobtainthree-term control:the integral - proportional, and derivative values, of PIDcontroller that will meet the transient and steady-state specifications of the closed-loop system.

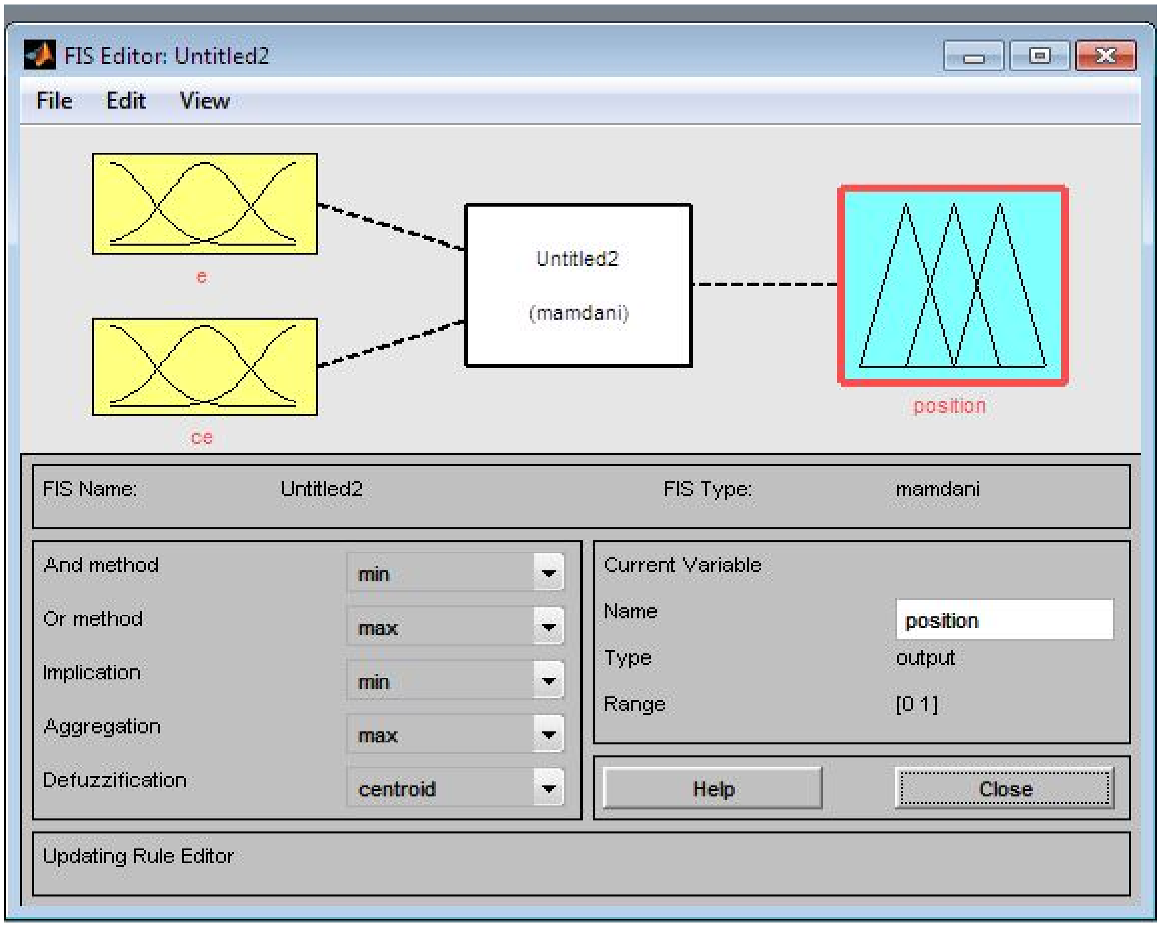
3.4 Fuzzy Controller Design

Fuzzy control system design essentially amounts to: (i) Choosing the fuzzy controllerinputs and outputs, (ii) Choosing the preprocessing that is needed for the controllerinputs and possibly post processing that is needed for the outputs, and (iii)Designingeach of the four components of the fuzzy controller as shown in Figure (2.14). Thereare standard choices for the fuzzification and defuzzification interfaces. Moreover,most often the designer settles on an inference mechanism and may use this for manydifferent processes. Hence, the main part of the fuzzy controller that we focus on fordesign is the rule-base.The rule-base is constructed so that it represents a human expert "in-the-loop."Hence, the information that we load into the rules in the rule-base may come from anactual human expert who has spent a long time learning how best to control theprocess. In other situations there is no such human expert, and the control engineer willsimply study the plant dynamics (perhaps using modeling and simulation) and writedown a set of control rules that makes sense. As an example, in the cruise control problem discussed above it is clear that anyonewho has experience driving a car can practice regulating the speed about a desired set-point and load this information into a rule-base. For instance, one rule that a humandriver may use is “If the speed is lower than the set-point, then press down further on the acceleratorpedal.” A rule that would represent even more detailed information about how to regulate the speed would be “If the speed is lower than the set-point AND the speed isapproaching the set-point very fast, then release the accelerator pedal by a smallamount.” This second rule characterizes our knowledge about how to make sure thatwe do not overshoot our desired goal (the set-point speed). Generally speaking, if weload very detailed expertise into the rule-base, we enhance our chances of obtainingbetter performance [13].

By editing "fuzzy" in workspace a window of FIS editor appears as in Figure (3.1).

3.4.1 Fuzzy basic FIS editor

The FIS editor displays high-level information about a fuzzy inference system shown as inFigure (3.1). At the top is a diagram of the system with each input and output clearlylabeled. Themembership function editorcan be brought By double-clicking on the input or output boxes, alsothe rule editor will be brought by Double-clicking on the fuzzy rule box in the center of thediagram. Just below the diagram is a text field that displaysthe name of the current FIS. The various functions used in the fuzzy implication process wasallowed by series of popup menus in the lower left of the window. Inthe lower right are fields that provide information about the current variable.Thecurrent variable is determined by clicking once on one of the input or output boxes.



FIS Name: Untitled2
FIS Type: mamdani

And method: min
Or method: max
Implication: min
Aggregation: max
Defuzzification: centroid

Current Variable
Name: position
Type: output
Range: [0 1]

Updating Rule Editor

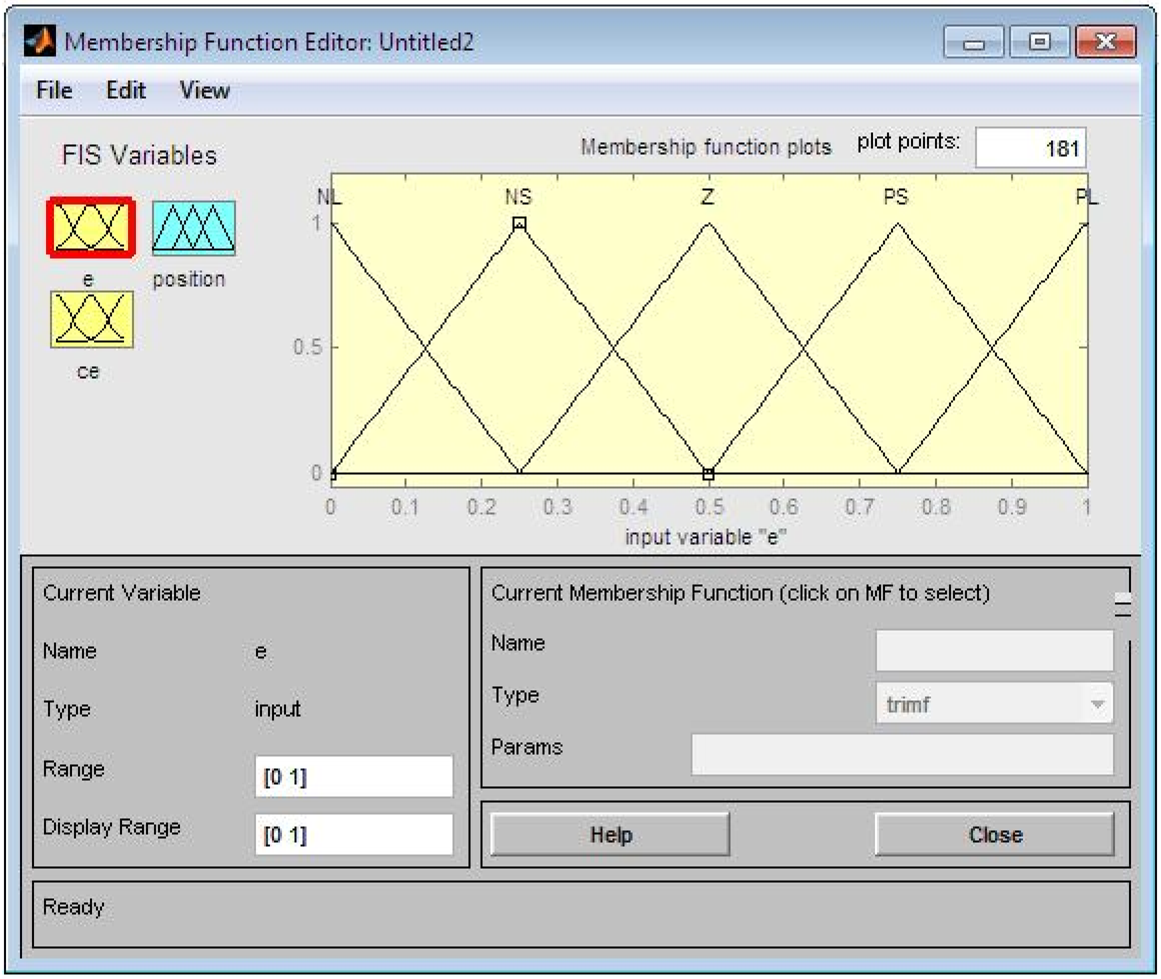
Figure (3.1): FIS editor

By adding INPUT variable from edit menu then FIS give two inputs variable whichone is error (e) and another is change of error (ce) etch one consist of five membershipfunctions to give 25 rules.

3.4.2 Membership function editor

all the membership functions for the FIS stored in the file a.FISweremodified byThe mfedit('a')that generates a membership function editor shown in Figure (3.2). mf edit(a) operates on a MATLAB workspace variable for a FIS structure a. mfeditalone opens the membership function editor with no FIS loaded.The Membership Function (MF) Editor is used to create, remove, and modify the MFsfor a given fuzzy systemthe currentvariable was selected by clicking once on one of the displayedboxesOn the left side of the diagram "variable palette". Information about the current variable is displayed in the text region below thepalette area. To the right is a plot of all the MFs for the current variable. It could select any of theseby clicking once on the line or name of the MF. Once selected, It could modify theproperties of the MF using the controls in the lower right.

MFs are added and removed using the edit menu.



Current Variable

Name e

Type input

Range [0 1]

Display Range [0 1]

Current Membership Function (click on MF to select)

Name

Type trimf

Params

Ready

Figure (3.2): Membership function

3.4.3 Rule editor

The rule editor shown in Figure (3.3), when invoked using ruled it ('a'), is used tomodifythe rules of a FIS structure stored in a file, a.fis. It can also be used to inspectthe rules being used by a fuzzy inference system.To use this editor to create rules, must befirst have all of the input and outputvariables will be wanted to use defined with the FIS editor. listbox and check box choices for input and output variables, connections, and weightscan be used to create the rules. When to operate on a workspace variable for a FIS structure called a. On the rule editor The syntax ruleedit(a) is used .there is a menu bar that was allowed to openrelated GUI tools open and save systems, and so on. The file menu for the rule editoris the same as the one found on the FIS editor. Refer to the reference entry fuzzy formore information.By using the edit menu items:

Undo: to undo the most recent change.

By using the View menu items:

Edit FIS properties... to invoke the FIS editor.

Edit membership functions... to invoke the membership function editor.

View rules... to invoke the rule viewer.

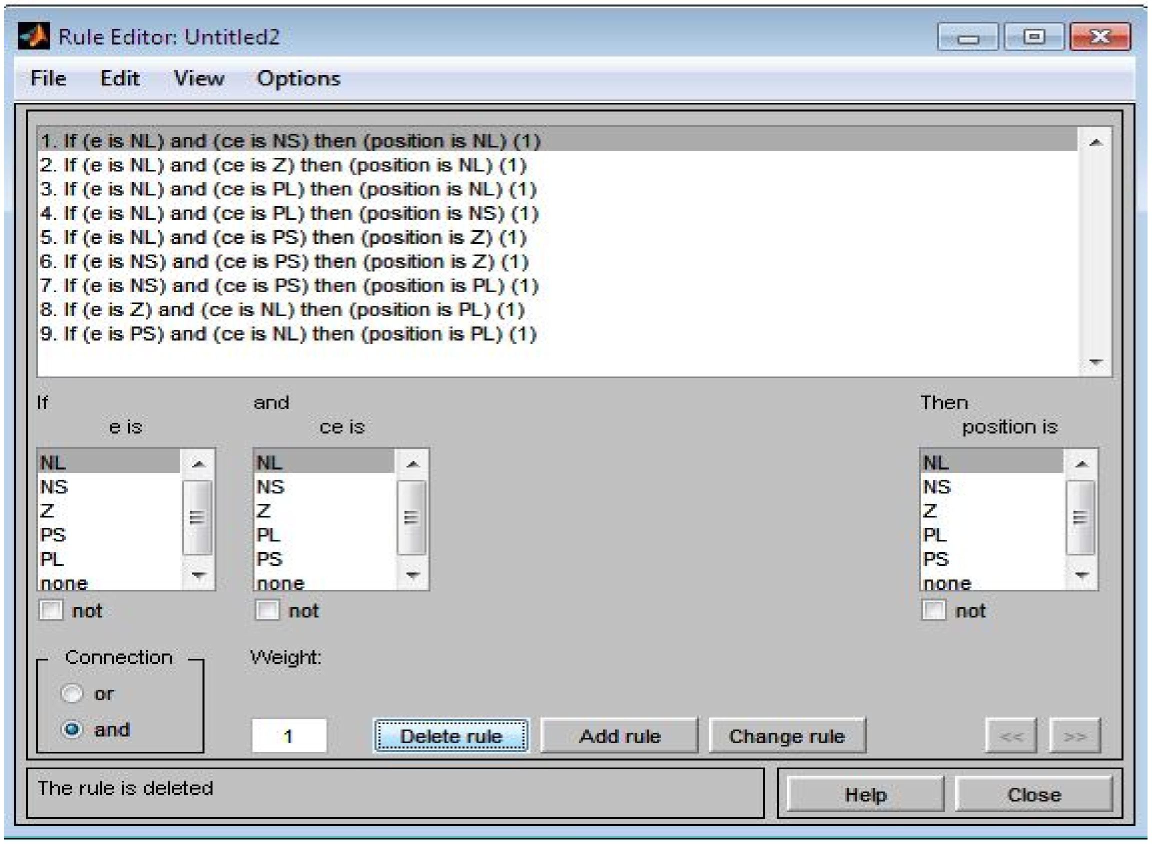
View surface... to invoke the surface viewer.

By using the options menu items:

Language to select the language: English, Deutsch, and Francais

Format: to select the format: verbose uses the words *if* and *then* and so on to

Create actual sentences.



1. If (e is NL) and (ce is NS) then (position is NL) (1)
2. If (e is NL) and (ce is Z) then (position is NL) (1)
3. If (e is NL) and (ce is PL) then (position is NL) (1)
4. If (e is NL) and (ce is PL) then (position is NS) (1)
5. If (e is NL) and (ce is PS) then (position is Z) (1)
6. If (e is NS) and (ce is PS) then (position is Z) (1)
7. If (e is NS) and (ce is PS) then (position is PL) (1)
8. If (e is Z) and (ce is NL) then (position is PL) (1)
9. If (e is PS) and (ce is NL) then (position is PL) (1)

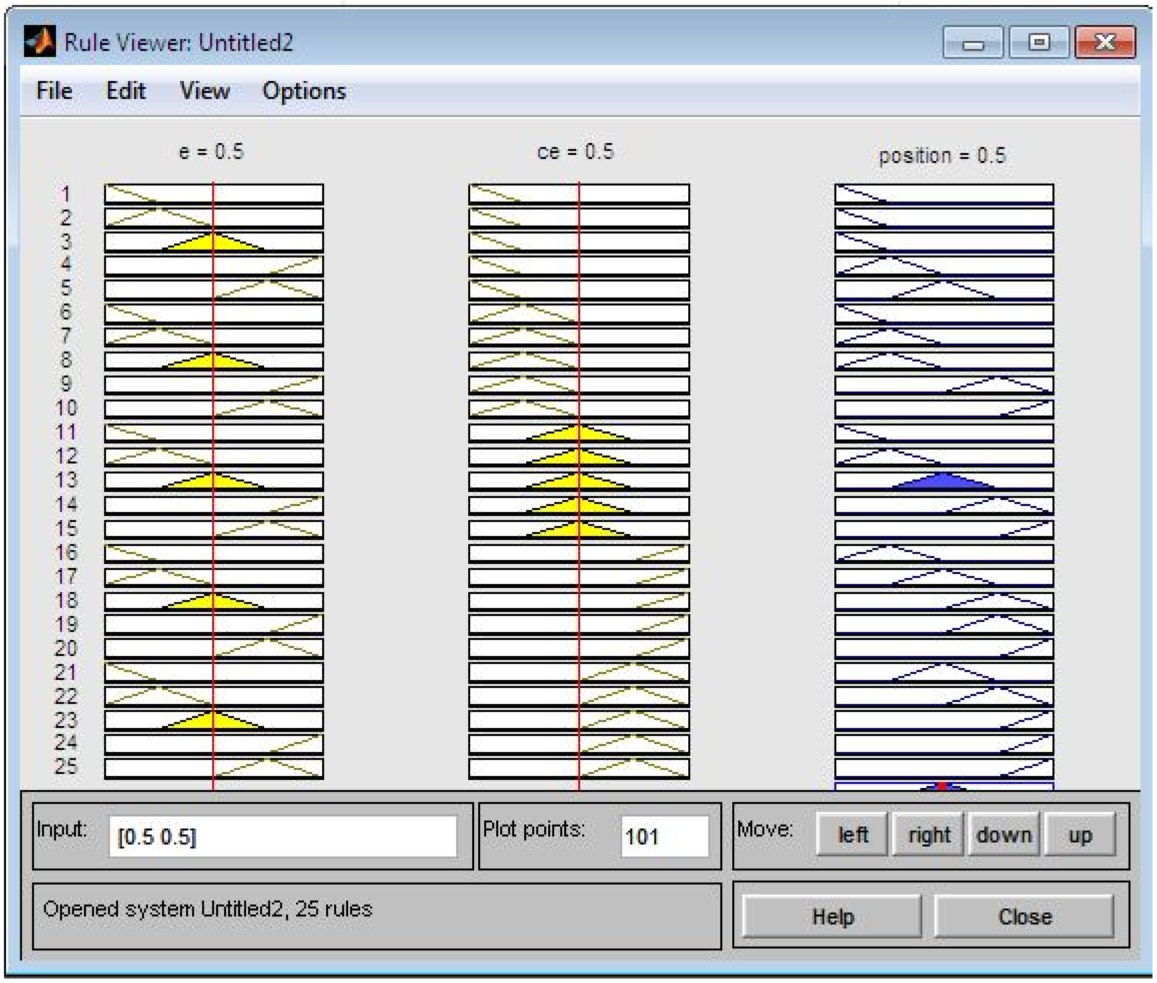
Figure (3.3): Rule editor

Table (3.1): Rule base for FIVE membership functions

E / DE	NL	NS	Z	PL	PS
NL	NL	NL	NL	NS	ZE
NS	NL	NS	NS	ZE	PS
Z	NL	NS	ZE	PS	PL
PL	NS	ZE	PS	PS	PL
PS	ZE	PS	PL	PL	PL

3.4.4 Rule viewer

The rule viewer displays shown in figure (3.4), in one screen, all parts of the fuzzyinference process from inputs to outputs. Each row of plots corresponds to one rule,and each column of plots corresponds to either an input variable (yellow, on the left) oran output variable (blue, on the right). It could change the system input either by typinga specific value into the Input window or by moving the long yellow index lines that godown each input variable's column of plots.



Input: [0.5 0.5]

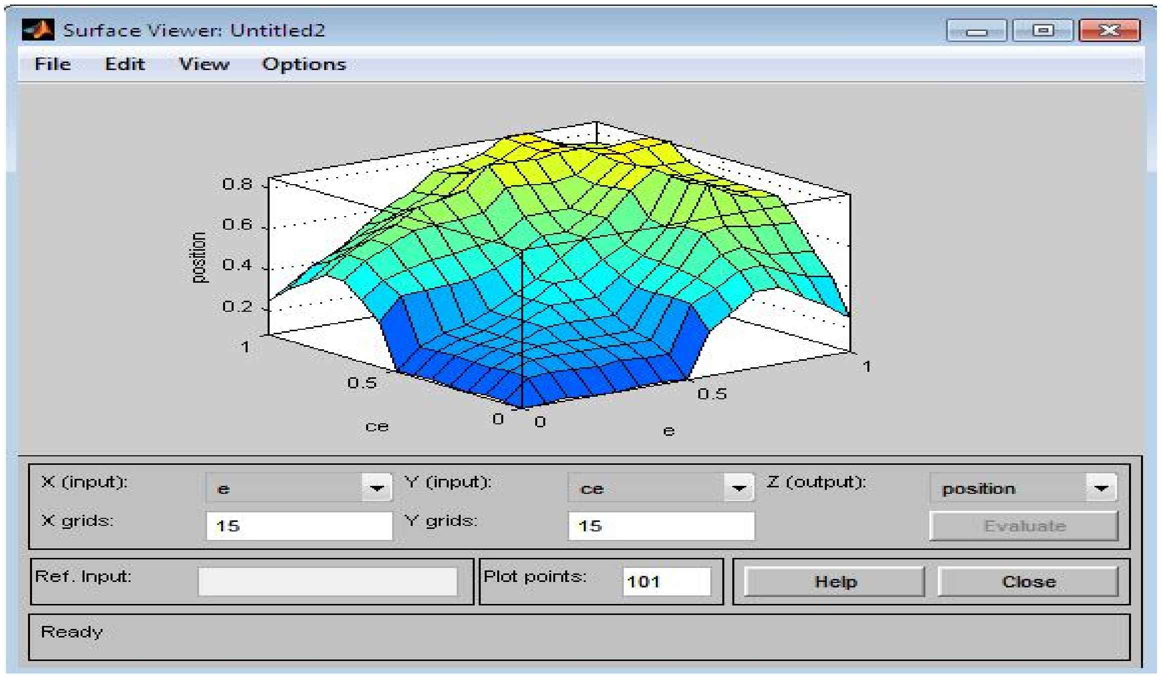
Plot points: 101

Opened system Untitled2, 25 rules

Figure (3.4): Rule viewer

3.4.5 Output surface viewer

the output surface of a FIS, a.fis, for any one or two inputs wasexamined by The surface viewer shown in Figure (3.5) invoked using surfview('a') is a GUI tool. Since itdoes not alter the fuzzy system or its associated FIS matrix in any way, it is a read-onlyeditor. the two input variables you want assigned to the two input axes (X and Y), as well the output variable you want assigned to the output (or Z) axis was seledby Using the pop-up menus,. Select the Evaluate button to perform the calculation and plot theoutput surface.Actually the surface was manipulatedBy clicking on the plot axes and dragging the mouse, so that can be viewed it from different angles. If there are more than two inputsto the system, they must be supplied, in the reference input section, the constant valuesassociated with any unspecified inputs.



X (input): e
X grids: 15
Y (input): ce
Y grids: 15
Z (output): position
Ref. Input:
Plot points: 101
Ready

Figure (3.5): Surface viewer

3.5Neuro-Fuzzy Controller Design

ANFIS uses a hybrid learning algorithm to identify the membership functionparameters of single-output, Sugeno type fuzzy inference systems . A combination ofleast-squares and back propagation gradient descent methods are used for training FISmembership function parameters to model a given set of input/output data.

[FIS, ERROR] = ANFIS (TRNDATA) tunes the FIS parameters using the input/outputtraining data stored in TRNDATA. For an FIS with Ninputs, TRNDATA is a matrix with N+1columns where the first N columns contain data for each FIS input and thelast column contains the output data. ERROR is the array of root mean square trainingerrors (difference between the FIS output and the training data output) at each epoch.ANFIS uses GENFIS1 to create a default FIS that is used as the starting point forANFIS training.Note: in this research training data was taken from PID controllers with computationaloptimization approach method.

3.5.1 ANFIS editor

The ANFIS editor shown in Figure (3.6) GUI from which the data set and train anfis were loaded was brought by Using anfisedit TheANFIS Editor GUI invoked usinganfisedit('a'), a FIS structure was used toopens the ANFIS editor GUI from which implement ANFISusing stored as a file a.FISanfisedit(a) operates the same way for a FISstructure a, stored as a variable in the MATLAB workspace, Figure (3.7) showingANFIS model structure . the related GUI tools On the ANFIS editor GUI, were being opened bythe menu bar, open and save systems, and so on. The file menu is the sameas the one found on the FIS editor

• By using the following edit menu item:

Undo to undo the most recent change.

FIS properties to invoke the FIS editor.

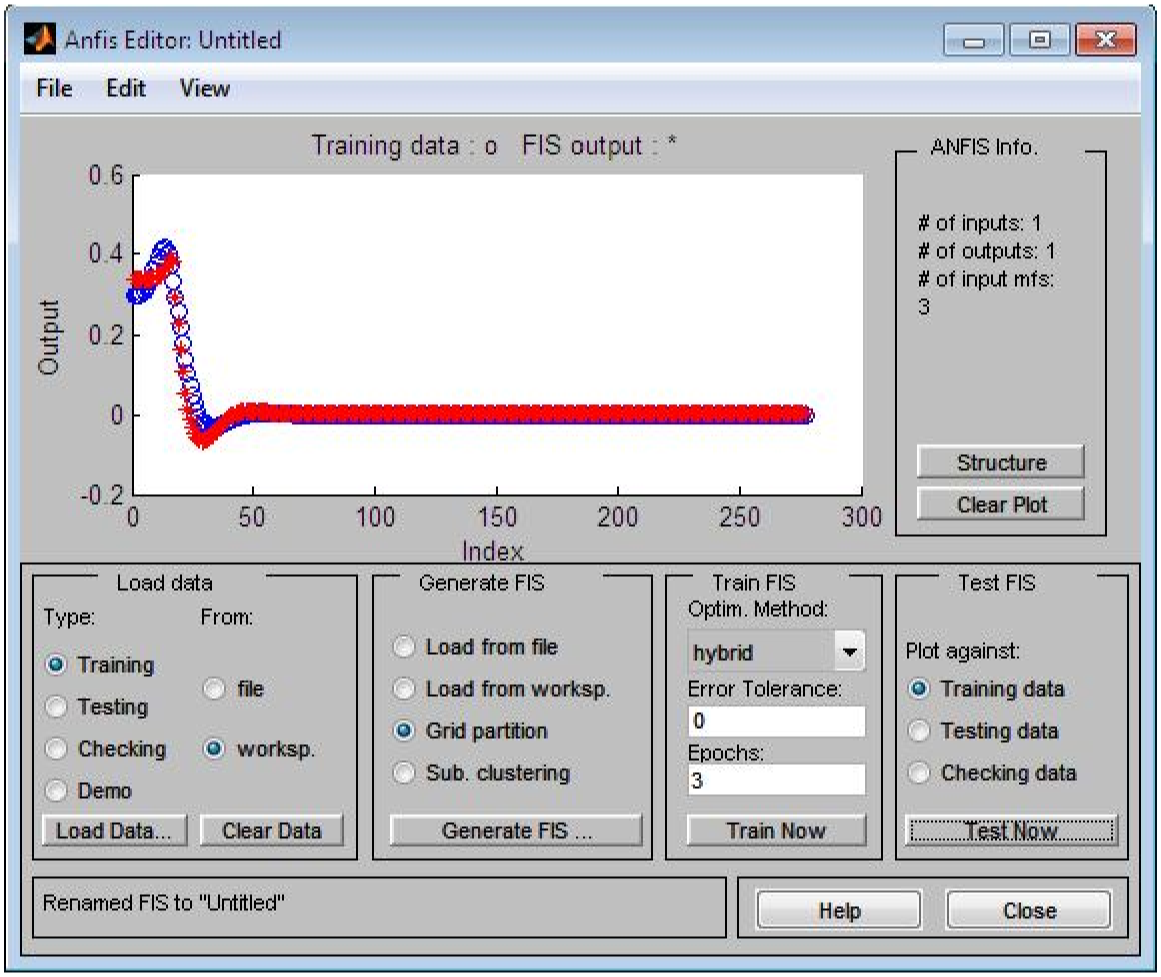
Membership functions to invoke the membership function editor.

Rules to invoke the rule editor.

- By using the following view menu items:

Rulesto invoke the rule viewer.

Surface to invoke the surface viewer.



# of inputs: 1
# of outputs: 1
# of input mfs:
3

Type: [✓ Training] Testing Checking Demo
From: file [✓ worksp.]

Load from file
Load from worksp.
[✓ Grid partition]
Sub. clustering

Optim. Method: hybrid
Error Tolerance: 0
Epochs: 3

Plot against: [✓ Training data] Testing data Checking data

Renamed FIS to "Untitled"

Figure (3.6): ANFIS editor

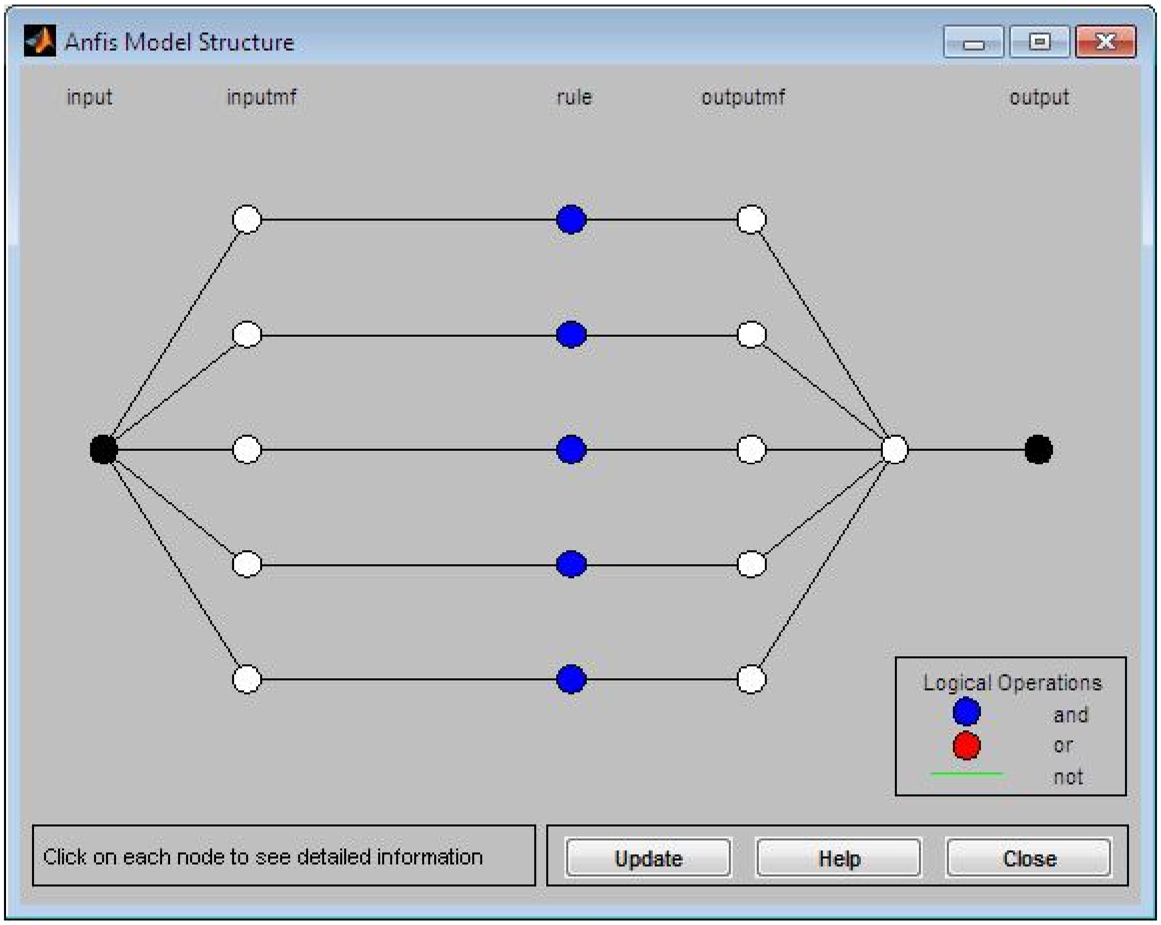


Figure (3.7): ANFIS model structure

CHAPTER FOUR

SIMULATION RESULTS AND DISCUSSION

CHAPTER FOUR

SIMULATION RESULTS AND DISCUSSION

4.1 Simulation Results of PID CONTROLLER

This section demonstrates the simulation results of a position control of DC servomotor by using design of controllers with MATLAB, by substituting the value of parameters: KP = 2.475, KI = 0.0805and KD = 0.7909 and by using the unit step response with MATLAB simulation as shown in figure (4.1) below:

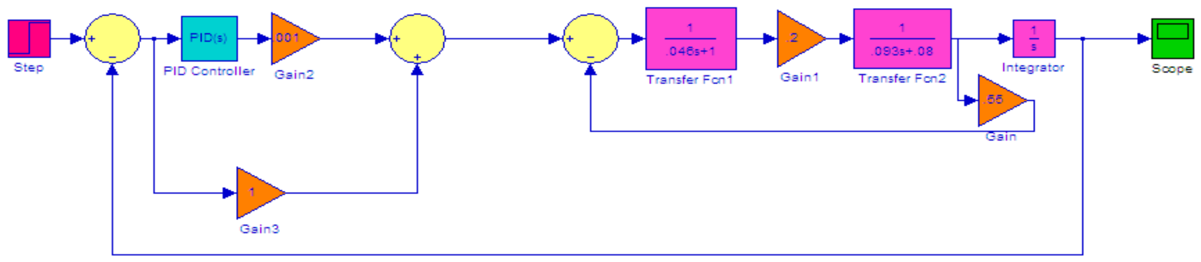


Figure (4.1): PID controller for position control of DC servomotor

Figure (4.2) illustrates the unit step response of PID controller for position control of DC servomotor

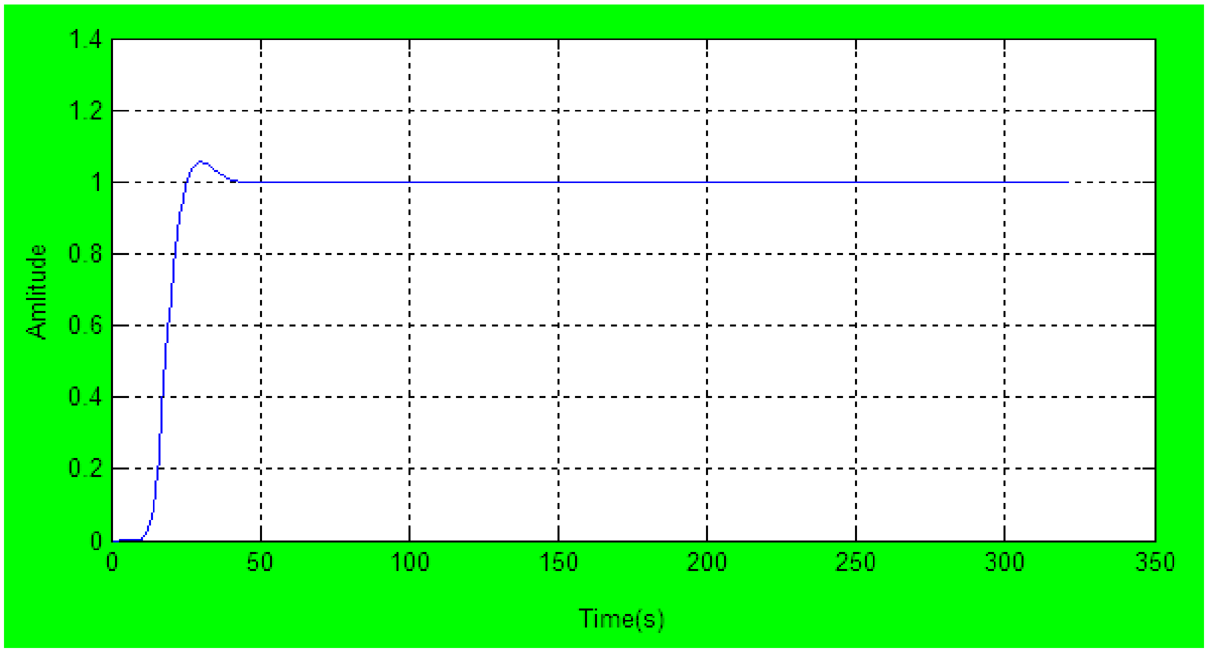


Figure (4.2): Unit step response of PID controller for position control of DC servomotor

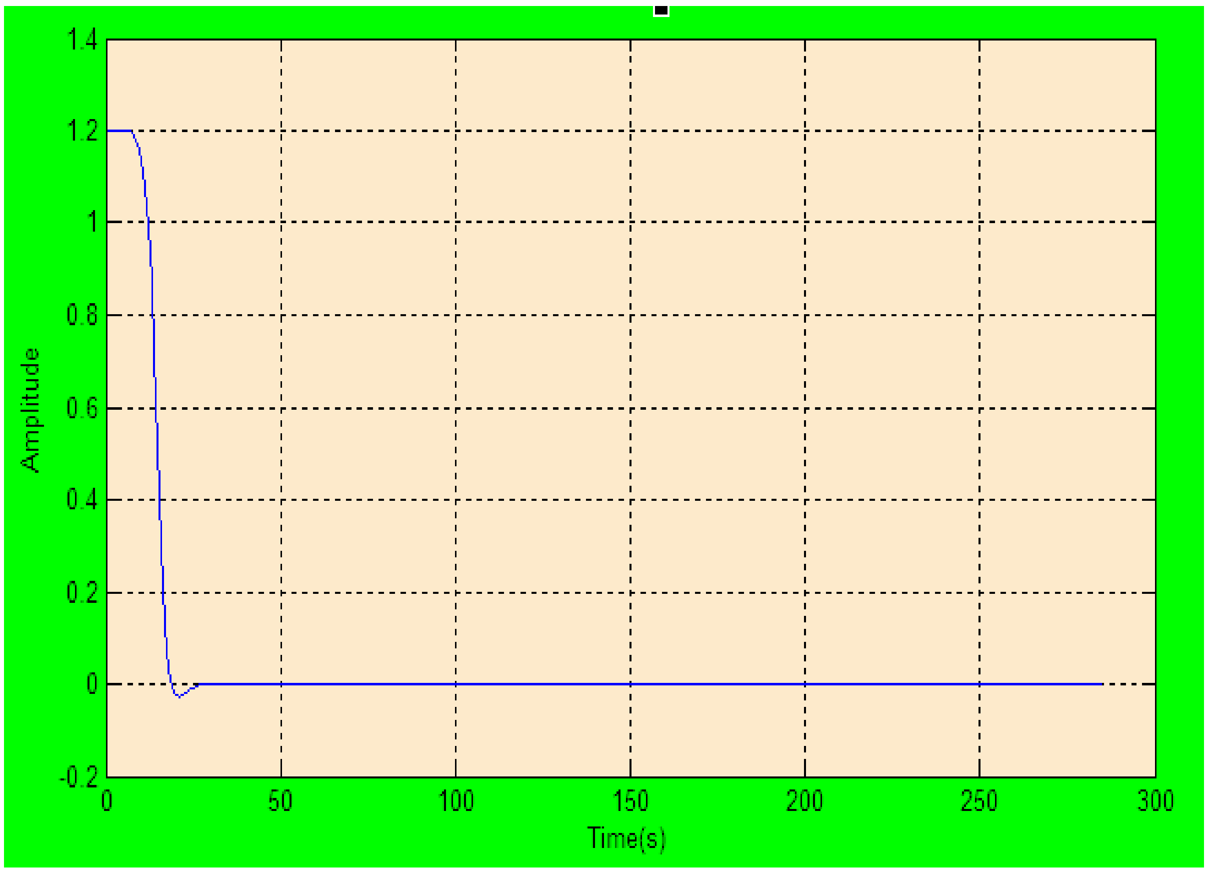


Figure (4.3) Control signal of PID controller

4.2 Simulation Results of Fuzzy

The system with a unit step input is simulated with MATLAB as shown in figure (4.3).By using trial and error method of tuning to calculate the FLC parameters. The values of FLC parameters are Kp = 2.8036and KI= .5, KD= .6351 figure (4.4) illustrate the response of system.

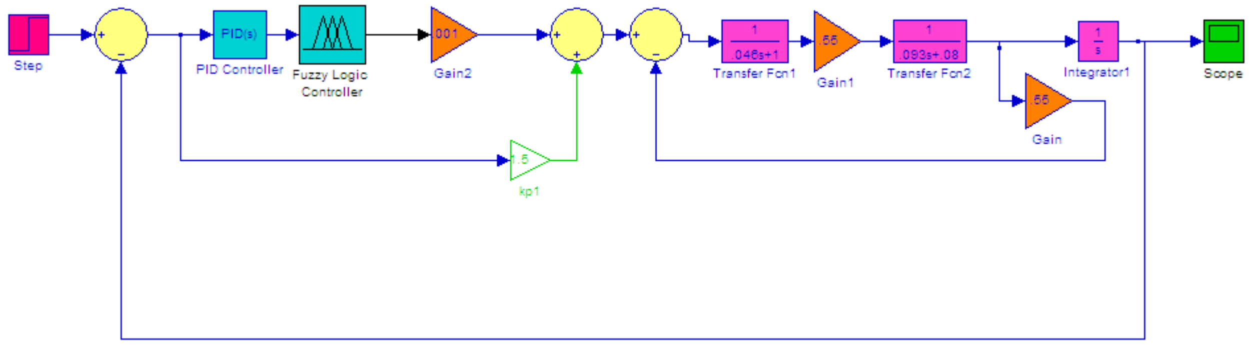


Figure (4.4): PID - like fuzzy for position control of DC Servomotor

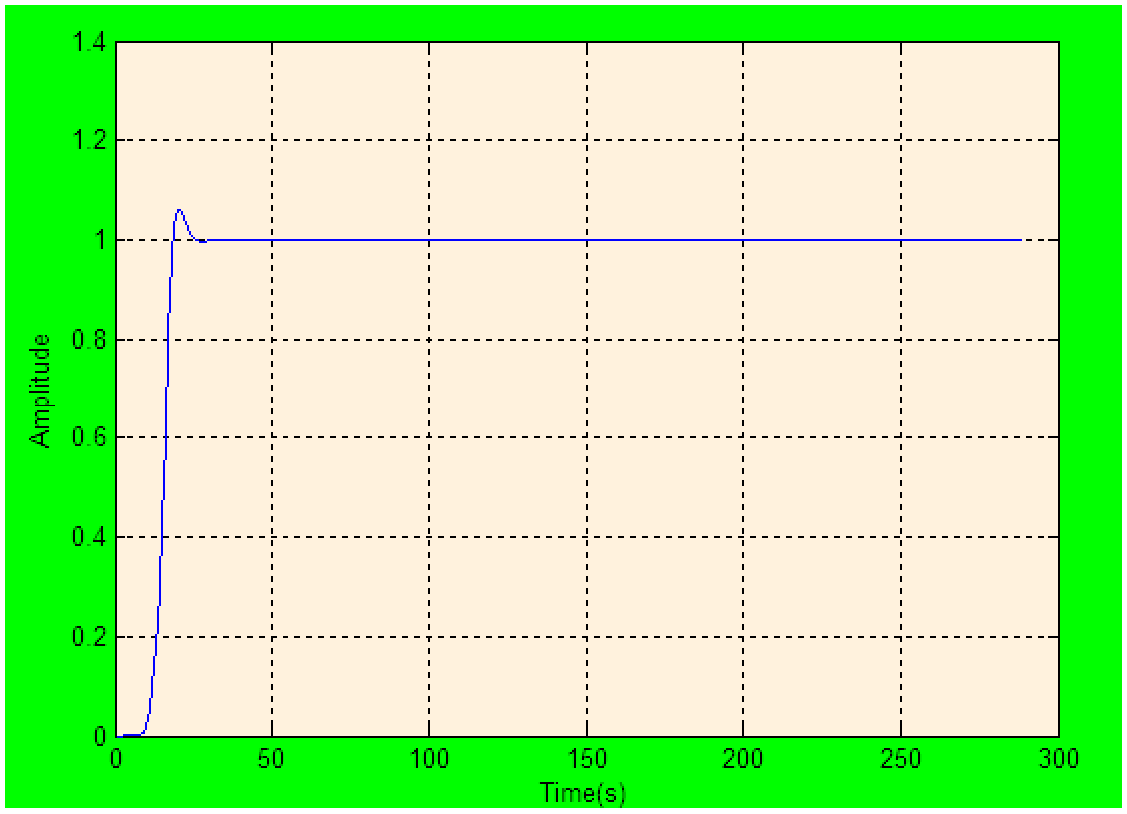


Figure (4.5): Unit step response of FLC for position control of DC servomotor

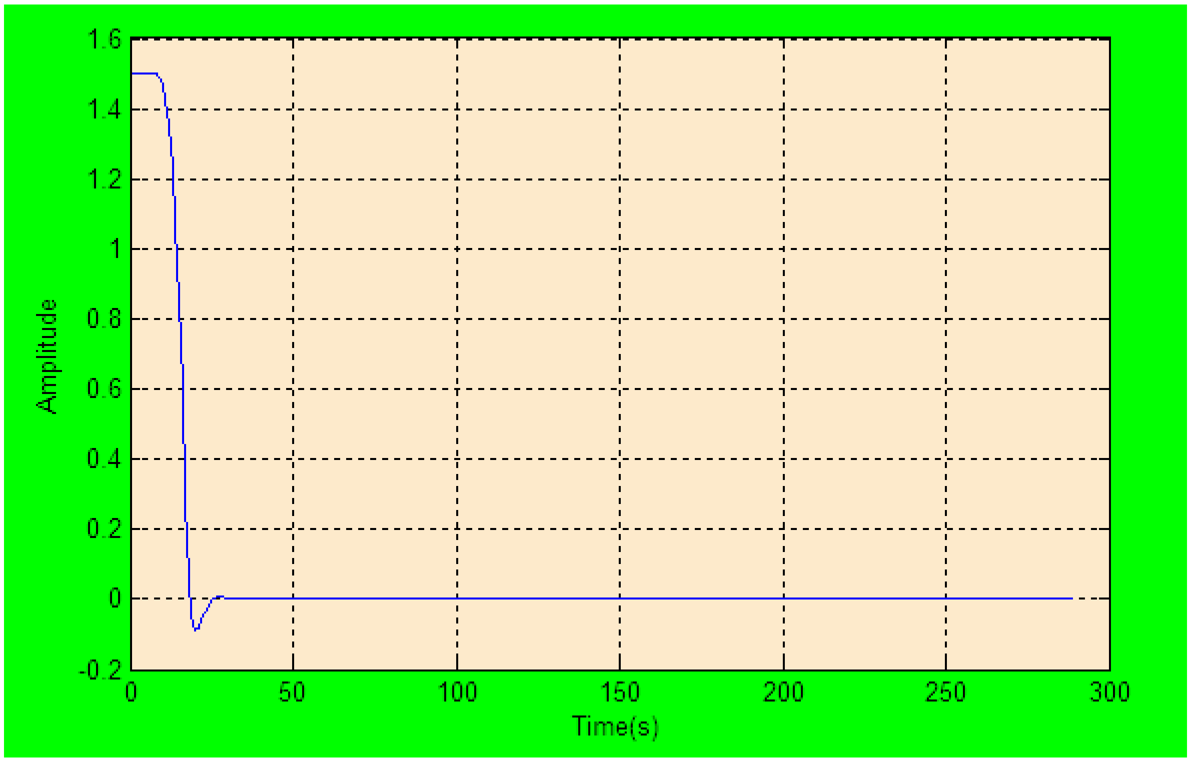


Figure (4.6): Control signal of fuzzy controller

4.3 Simulation Results of Neuro-Fuzzy Controller

By using train data from PID controllers using MATLAB SIMULINK approach to design Neuro-Fuzzy controller, Figure (4.7) illustrates block diagram of PID controller which train data come from it.

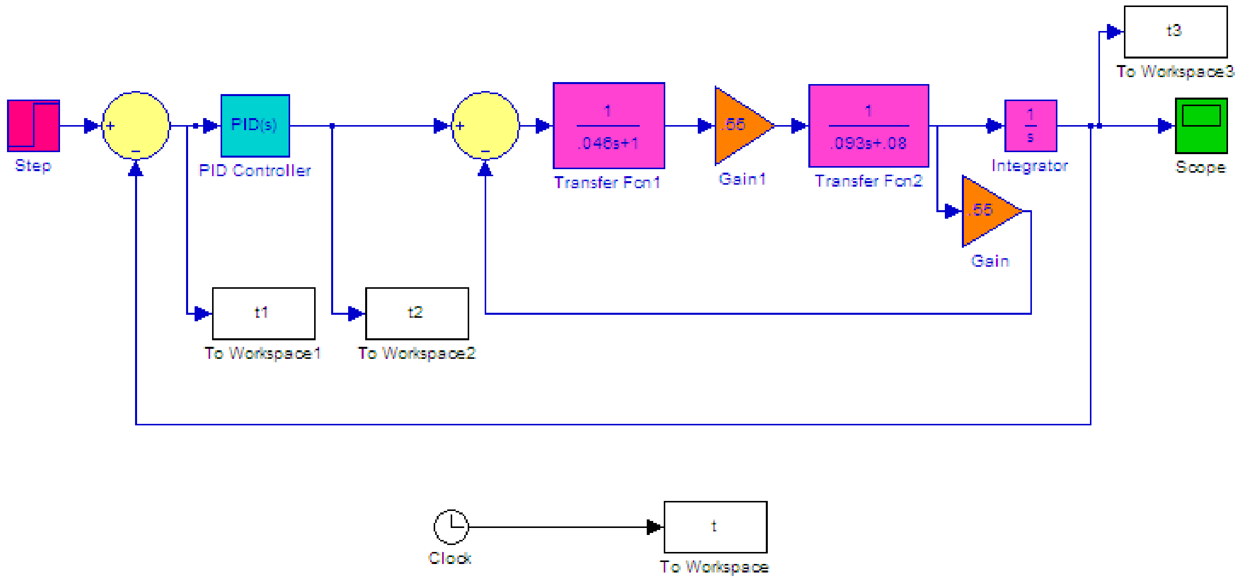


Figure (4.7): Block diagram of PID controller

The system with a unit step input is simulated with MATLAB to illustrate block diagram of Neuro-Fuzzy controller as shown in figure (4.8).

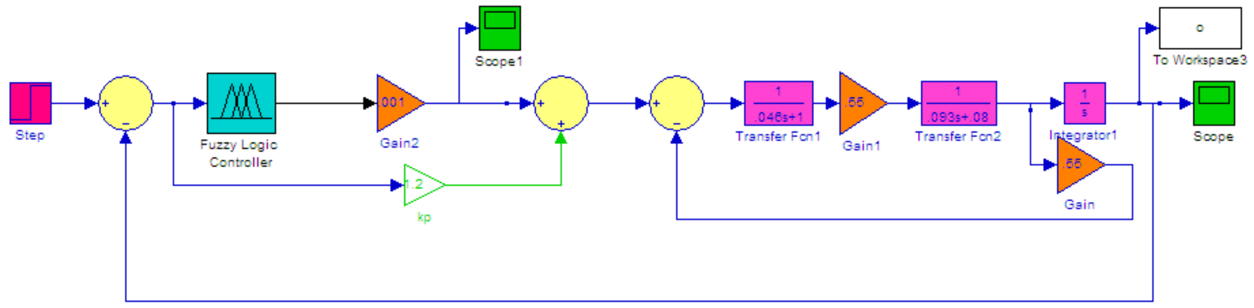


Figure (4.8): Block diagram of neuro-fuzzy controller

Figure (4.9) illustrate control signal of Neuro-Fuzzy controller and figure (4.10) illustrate the step response for position control of DC Servomotor.

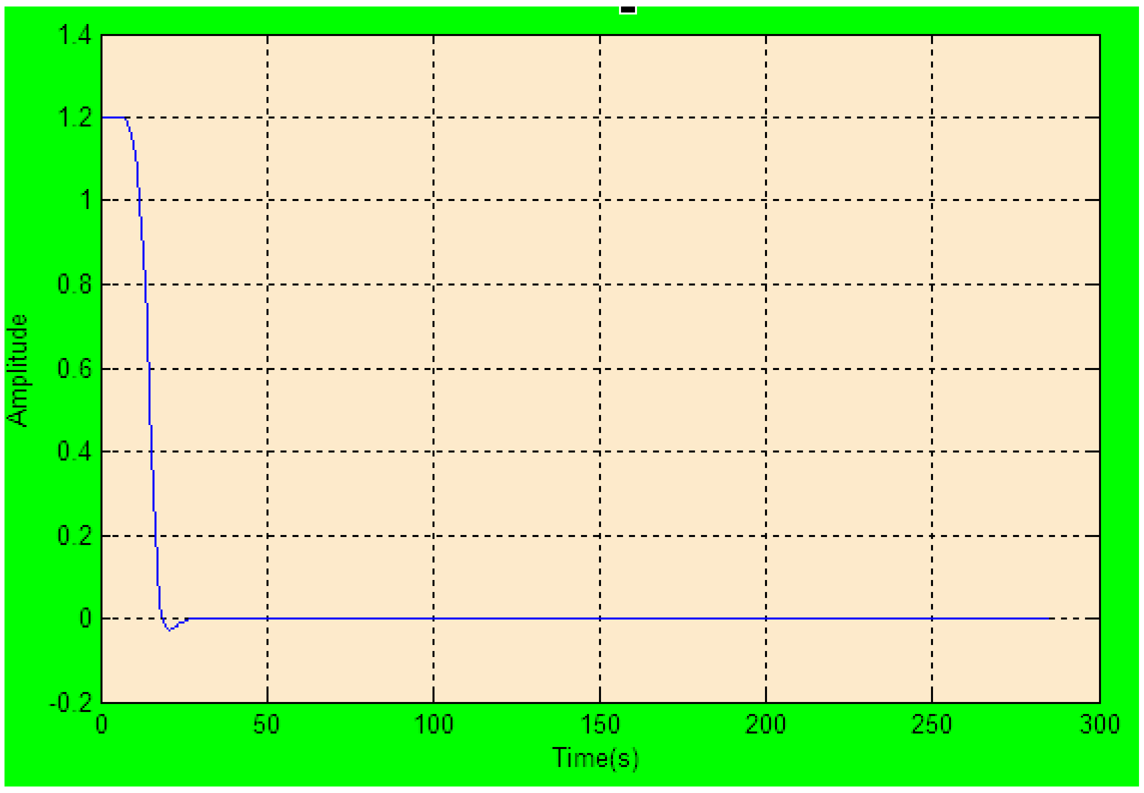


Figure (4.9): Control signal of neuro-fuzzy controller

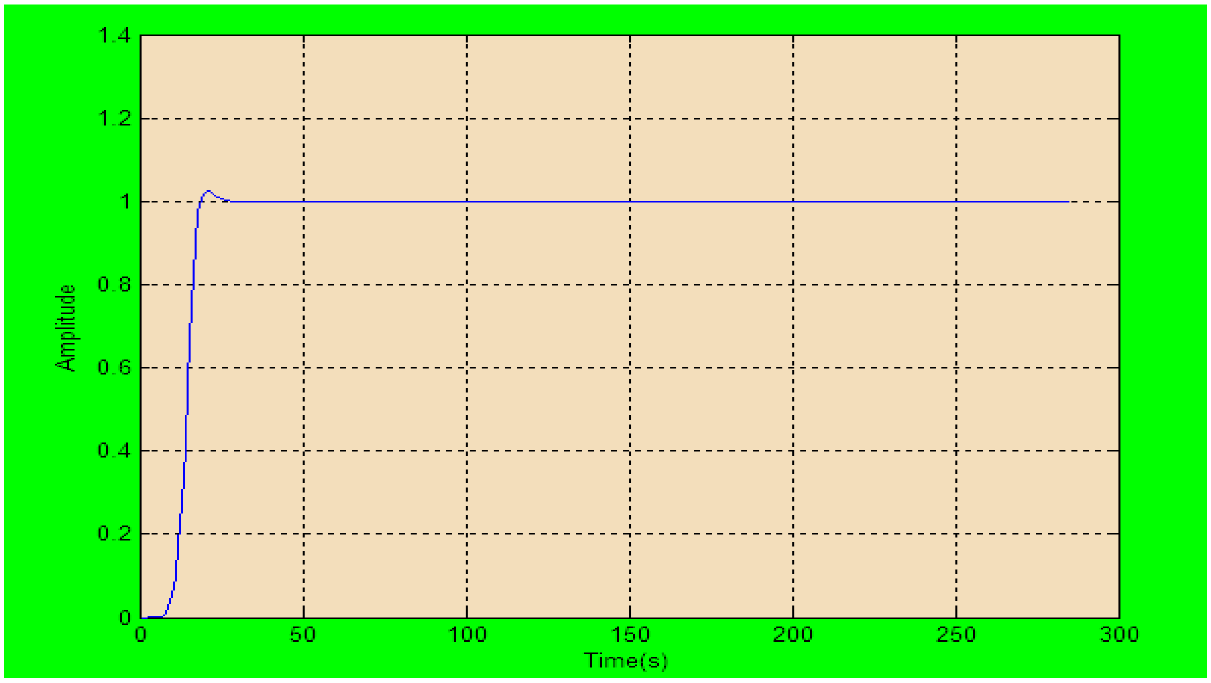


Figure (4.10): Unit step response of neuro-fuzzy controller for system

The unit step system response for all controllers PID, FLC and neuro-fuzzy are shown in figure (4.11).

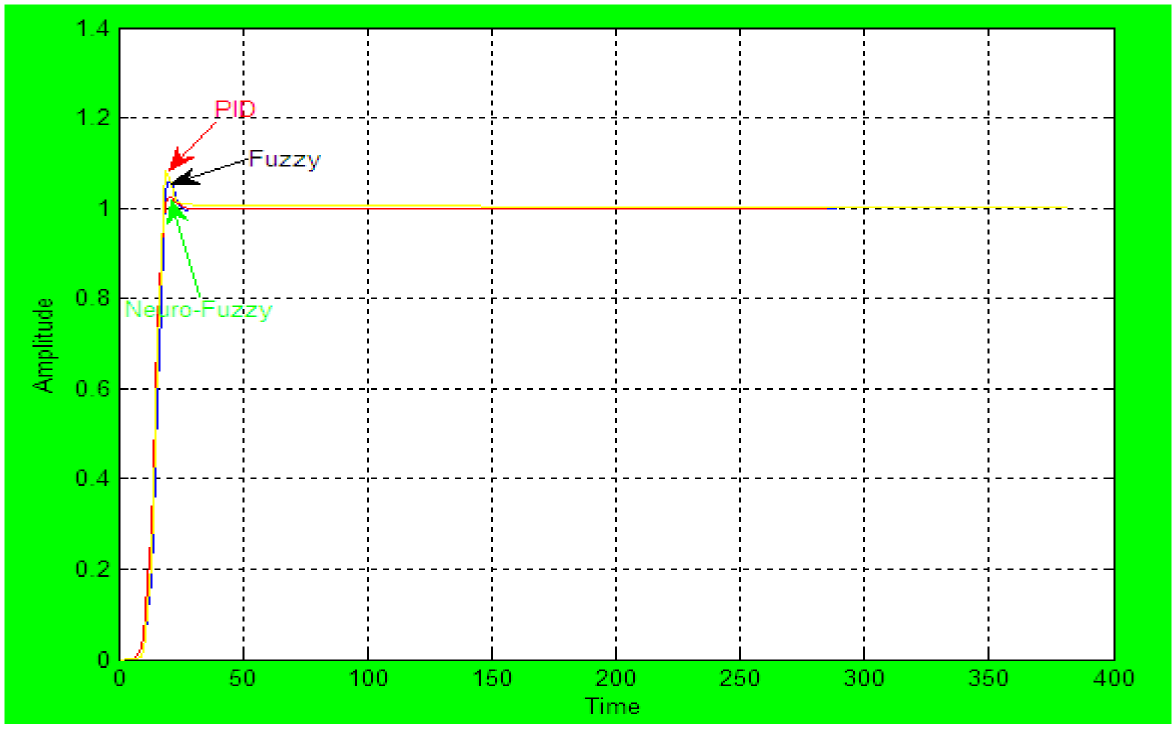


Figure (4.11): Unit step system response for all controllers

4.4 Comparison and Discussion

In order to validate the control strategies as described above, digital simulation were carried out on a converter DC motor drive system whose parameters are given in previous chapter. The MATLAB/SIMULINK models of system under study with all three controllers are shown in Figures (4.1), Figures (4.4) and (4.8). First a comparison has been made between the maximums overshoot, rise times and settling time illustrated in table (4.1):

Table (4.1): comparison between: maximums overshoot, rise times and settling time

Controller	Ts (Sec)	Tr (Sec)	Overshoot MP
PID	1.240	0.323	8.68
Fuzzy	2.321	1.025	5.8
Neuro-Fuzzy	2.640	1.37	2.37

CHAPTER FIVE

CONCLUSION AND RECOMMENDATIONS

CHAPTER FIVE

CONCLUSION AND RECOMMENDATIONS

5.1 Conclusion

In this study, the DC servomotor has been reviewed from control theory perspective. PID controller and intelligent techniques such as Fuzzy logic Controllers and their hybrid (ANFIS) are used for DC servomotor position control. This study was based on the MATLAB/Simulink software platform and mathematical models of the DC servomotor. Design with PID controller in MATLAB/Simulink, yield good response. Design with fuzzy controller had given perfect results, but also a trial-and-error method was needed to find the required parameters. Design with neuro-fuzzy controller results a very good response and very fast. The advantages of the neuro-fuzzy controller were that it was determined the number of rules automatically, reduces computational time, learns faster and produces lower errors than other method and no needed for tuning parameters. By proper design a neuro-fuzzy controllers can replace PID and Fuzzy controllers for the position control of DC servomotor drives. From simulation results, it was concluded that the used of ANFIS reduced design efforts

5.2 Recommendations

- Implementation of study, this could probably be achieved through use of DSP card or microcontroller.
- position control of DC servomotor with load and disturbance of DC servomotor.
- position control of DC servomotor using other AI techniques like neural network and compare results with PID, FLC and neuro-fuzzy controller.

References

1. www. oriental motor.com (4.1.2014)
2. Nise, Norman S.. Control Systems Engineering. Fifth edition 2008.John E. Traister “Electrical Design Details” 2nd edition-McGraw-hill Companies- copyright©2003.
3. John Bird “Electrical and Electronic Principles and Technology” 2ndedition - Newnes - copyright©2003.
4. www.jameca.com
5. Karl Johan “Control system design”Astrom-2002.
6. http://www.electrical4u.com/voltage-or-electric-potential-difference/#top(15march2013)
7. Thrishantha Nanayakkara and Mo Jamshidi “Intelligent Control Systems with an Introduction to System of Systems Engineering” Taylor & Francis Group-2010.
8. Leonid Rezink “Fuzzy Controllers” Victoria University of Technology, 1997.
9. Hung T. Nguyen - Nadipuram R.Prasad - Carol L. Walker - Elbert A. Walker “A First course in Fuzzy and Neural Control” CHAPMAN and HALL/CRC – 2003
10. Martin T. Hagan - Howard B. Demuth “Paper on Neural-Networks-for Control”Oklahoma State University, University of Idaho.
11. Oktavián Strádal, Radovan Soušek “Parameter Adaptation in a Simulation Model Using ANFIS” Mechanics Transport Communications, Academic journal-1, 2008.
12. Boumediene Allaoua, Abdellah Laoufi, Brahim Gasbaoui, And Abdessalam Abderrahmani “Paper on Neuro-Fuzzy DC Motor Speed Control Using Particle Swarm Optimization” Leonardo Electronic Journal of Practices and Technologies , 15- July-December 2009

13. Prof. Krishna Vasudevan “Paper on Application of DC motors” Indian Institute of Technology Madras.