Sudan University of Science and Technology
College of Post Graduate Studies

COMPARISON BETWEEN PROPORTIONAL INTEGRAL DERIVATIVE CONTROLLER AND FUZZY LOGIC CONTROLLER FOR DC SERVOMOTOR

مقارنة بين المتحكم التناسبي التكاملي التفاضلي والمتحكم الغامض في محرك الخدمة ذي التيار المستمر

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

Prepared By:

Mayssaa Awad Elkhalifa Subier

Supervised By:

Dr. Awadalla Taifour Ali

November 2015

بسم الله الرحمن الرحيم

Sudan University of Science & Technology

College of Graduate Studies

جامعة السودان للعلوم والتكنولوجيا

كلية الدراسات العليا

كلية الدراسات العليا

Approval Page

Name of Candidate: Mayssaa Awad Elkhalifa Subier

Thesis title: Comparison Between Proportional Integral Derivative Controller and Fuzzy Logic Controller for DC servomotor

مقارنة بين المتحكم التناسبي التكاملي التفاضلي والمتحكم المنطقي الغامض في محرك الخدمة ذو التيار المستمر

Approved by:

1. External Examiner

Name: Dr. ElZein Hassan Nasir

Signature: .... Date: 19/11/2015

2. Internal Examiner

Name: Dr. Aamir Hashim Obeid Ahmed

Signature: .... Date: 19/11/2015

3. Supervisor

Name: Dr. Awadalla Taifour Ali Ismail

Signature: .... Date: 22/11/2015

قال تعالى:

* لاَ يُكَلِّفُ اللهُ نَفْساً إِلاَّ وُسْعَهَا لَهَا مَا كَسَبَتْ وَعَلَيْهَا مَا اكْتَسَبَتْ رَبَّنَا لاَ تُؤَاخِذْنَا إِن نَّسِينَا أَوْ أَخْطَأْنَا رَبَّنَا وَلاَ تَحْمِلْ عَلَيْنَا إِصْراً كَمَا حَمَلْتَهُ عَلَى الَّذِينَ مِن قَبْلِنَا رَبَّنَا وَلاَ تُحَمِّلْنَا مَا لاَ طَاقَةَ لَنَا بِهِ وَاعْفُ عَنَّا وَاغْفِرْ لَنَا وَارْحَمْنَا أَنتَ مَوْلاَنَا فَانصُرْنَا عَلَى الْقَوْمِ الْكَافِرِينَ *

صدق الله العظيم

الاية 286, من سورة البقرة

DEDICATION

TO MY HUSBAND SAMIR, AND MY DAUGHTER HALA ,FOR THEIR UNDERSTANDING AND CONSIDERATION; TO MY PARENTS FOR THEIR LOVE OVER MANY YEARS AND TO MY FAMILY FOR THEIR SUPPORT.

I DEDICATE THIS EFFORT…

Acknowledgement

In preparing this thesis, I was in contact with many people, they have contributed towards my understanding and thoughts. First of all I wish to express my sincere appreciation to my thesis supervisor, Dr. Awadalla Taifour, for encouragement, guidance, critics and friendship. Without his continued support and interest, this thesis would not have been the same as presented here. I also thank Dr. Hussam Widaa, with whom I discussed the basics of the research subject, and who provides me much knowledge. Not to forget Eng. Aman Jacknoon who supported me all the way, finally I would like to thank Eng. Galal Abd Elrahman and all Electrical Engineering Department staff, their hospitality and support during the period of this research is greatly acknowledged.

I would like to give my sincere appreciation to all my friends and others who have provided assistance at various occasions. Their views and tips are useful indeed. Unfortunately, it is not possible to list all of them in this limited space. Finally to all my family members where without them I would not be here.

ABSTRACT

This study presents a comparative study of two controllers for position and speed control of DC servomotor. PID controller is the most usually used for the position control in DC servomotor. However, the PID controller has some disadvantages such as: the high starting overshoot, slow response for unexpected disturbance and high sensitivity to the controller gain. So, the reasonably design PID controller with computational optimization approach method is proposed to overcome the disadvantage of the conventional PID controller. In this study fuzzy logic controller is proposed and the performance of both controllers are compared. Simulation results are presented and analyzed for both controllers. In this study it is observed that the fuzzy login controller has a better response than a PID controller for position and speed control of DC servomotor.

مستخلص

تقدم هذه الدراسة مقارنة بين متحكمين للتحكم في وضع و سرعة محرك خدمة تيار مستمر.غالبا ما يستخدم المتحكم التناسبي التكاملي التفاضلي للتحكم في وضع محرك الخدمة. ولكن رغم ذلك نجد ان للمتحكم التناسبي التكاملي التفاضلي بعض العيوب مثل البدء بتجاوز عالي للهدف والاستجابة البطيئة للتغيرات الفجائية وحساسية عالية لكسب المتحكم وعليه فإن مقترح تصميم المتحكم التناسبي التكاملي التفاضلي بالطريقة الحسابية المثلى يتغلب على بعض عيوب المتحكم التناسبي التكاملي التفاضلي التقليدي. في هذه الدراسة تم اقتراح المتحكم الغامض وتم مقارنة الأداء بالنسبة للمتحكمبن. وتم عرض نتائج المحاكاة وتحليلها بالنسبة للمتحكمين. و تمت ملاحظة ان المتحكم الغامض له استجابة أفضل من المتحكم التناسبي التكاملي التفاضلي بالنسبة للوضع والسرعة لمحرك الخدمة ذو التيار المستمر.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

AC	Alternative current
CE	Change of error
D	Derivative term
DC	Direct current
E	Error
Emf	Electro motive force
FIS	Fuzzy interface system
FL	Fuzzy logic
FLC	Fuzzy logic controller
GUI	Graphical user interface
I	Integral term
KVL	Kirchhoff's voltage law
MF	Membership function
N	Negative
NL	Negative large
NM	Negative medium
P	Positive
PD	Proportional derivative
PI	Proportional integral
PID	Proportional integral derivative
PL	Positive large
PM	Positive medium
Z	Zero

LIST OF SYMBOLS

K_p	Proportional gain
K_i	Integral gain
K_d	Derivative gain
V_b	inverse electromotive force
I_a	Rotor Current
R_a	Armature Resistance
L_a	Armature inductance
J_m	Moment of inertia of the rotor
B_m	Coefficient of friction
K_b	Back electromotive force constant
E_b	Back emf in the armature
T_L	Load moment
Mp	Maximum overshoot
tr	Rise time
ts	Settling time
$\mu(x)$	Degree of membership function
ωm	Motor angular velocity
Θm	Angular position
I_f	Field current
Ψ	Magnetic flux
T_m	Torque
K_a	Motor torque constant
e(t)	Error
P_{out}	Proportional term
I_{out}	Integral term
D_{out}	Derivative term

CHAPTER ONE

INTRODUCTION

CHAPTER ONE

INTRODUCTION

1.1 General

Servomotor system consists of different electrical and mechanical components. The different components are combined together to perform the function of the servomotor. DC servomotors have ability to be controlled by changing voltage signal connected to the input. The characteristics made them powerful actuators used everywhere. The most widely used control strategy in industry is proportional integral derivative (PID) controller. The popularity of PID controllers can be attributed partly to their robust performance in a wide range of operating conditions and partly to their functional simplicity. PID controller has been designed for higher order system using automated PID tuning by using Matlab's SISO design tool. Fuzzy logic employing the logic of approximate reasoning continues to grow in importance, as it provides an inexpensive solution for controlling ill-known complex systems, a fuzzy logic controller has been proposed by using Fuzzy Logic Toolbox to design the Fuzzy Inference System (FIS). The simulation results for a higher order systems will be demonstrated both for PID and fuzzy logic control.

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 nonlinear characteristics that affect both output speed and position of a DC servomotor such as saturation and friction could degrade the performance of conventional controllers. Conventional PID controller doesn't achieve the optimum performance and has some disadvantages such as: the high starting overshoot, slow response for unexpected disturbance and high sensitivity to the controller gain.

1.3 Objectives

This study investigates the problem of replace a proportional-integral-derivative (PID) controller algorithm with a fuzzy controller, using initial guesses as to the fuzzy membership functions and rules; to tune the fuzzy controller for optimum performance; and to compare the performance results to those from a PID controller. When tuned for a given set of operating conditions, the fuzzy controller matches or exceeds the PID controller performance.

The main objectives of this study are to:

- ✓ Design position and speed control of DC servomotor system using PID controller.
- ✓ Design position and control of DC servomotor system using fuzzy logic controller.
- ✓ Comparison of the results of both proposed controllers.

1.4 Methodology

- Literature review of all previous studies.
- Development of mathematical and computer models of DC servomotor.
- Use of Mamdani inference in fuzzy logic toolbox.
- All controllers design and simulation results are carried out using MATLAB toolbox.

1.5 Lay out

This thesis consists of five chapters: Chapter One presents an introduction to the principles of the study. Chapter Two discusses a theoretical background of DC servomotor, PID controller and fuzzy system. Chapter Three presents the system control design of position and speed control of DC servo system. Chapter Four presents the simulation results. Chapter Five provides the conclusion and recommendations.

CHAPTER TWO

THEORETICAL BACKGROUND AND LITERATURE REVIEW

CHAPTER TWO

THEORETICAL BACKGROUND AND LITERATURE REVIEW

2.1 Introduction

For the most part electrical machines DC or AC are mostly used for continuous energy conversion. Servomotor sometimes called control motors are electric motors that are specially designed and constructed, mainly for use in feedback control systems, as output actuators. Their power rating can vary from a fraction of a watt up to a few hundred watts. They have a high speed of response, which requires low rotor inertia. These motors are therefore smaller in diameter and longer in length. They normally operate at low or zero speed and thus have a larger size for their torque or power rating than conventional motors of similar rating applications [1].

2.2 Servomotor

There are two types of servomotors--AC and DC. AC servos can handle higher current surges and tend to be used in industrial machinery. DC servos are not designed for high current surges. Generally speaking, DC motors are less expensive than their AC counterparts [15].

2.2.1 Types of servomotors

There are two types of servomotors: DC servomotors and AC servomotors.

- **DC servomotors**

Dc servomotors are normally used as prime movers in computers, numerically controlled machinery, or other applications where starts and stops are made quickly and accurately. Servomotors have lightweight, low-inertia armatures that respond quickly to excitation-voltage changes. In addition, very low armature inductance in these servomotors results in a low electrical time constant (typically 0.05 to 1.5 mess) that further sharpens servomotor response to command signals. Servomotors include permanent-magnetic, printed-circuit, and moving-coil (or shell) DC servomotors. The rotor of a shell DC servomotor consists of a cylindrical shell of copper or aluminum wire coils which rotate in a magnetic field in the angular space between magnetic pole pieces and a stationary iron core. Servomotors usually have two, four, or six poles [13]. DC servomotors are separately excited dc motors or permanent dc motors. A schematic diagram of separately excited DC servomotor is shown in Figure (2.1). These DC servomotors are normally controlled by the armature voltage. The armature is designed to have large resistance so that the torque –speed characteristics are linear and have a large negative slope shown in Figure (2.2). The negative slope provides viscous damping for the servo drive system. Recall that the armature mmf and excitation field mmf are in quadrature in a DC machine Figure (2.3), this provides a fast torque response because torque and flux are decoupled. Therefore, a step change in the armature voltage (or current) Result in a quick change in the position or speed of the rotor [1].

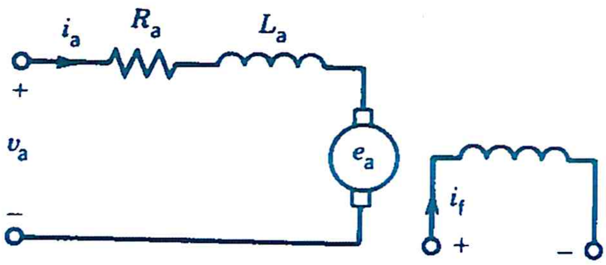

Figure 2.1: Separately excited DC servomotor

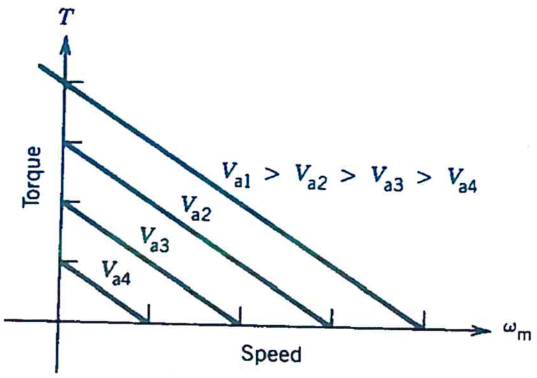

Figure 2.2: DC motor torque-speed characteristics

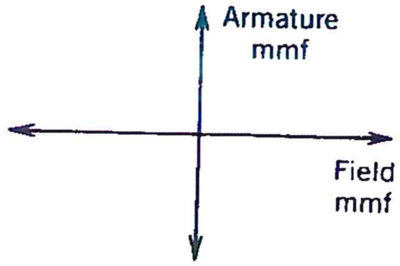

Figure 2.3: Armature mmf and excitation field mmf

- **AC servomotors**

Indeed most high power servomotors are DC servomotors the power rating from a few watts to several hundred watts. Presently AC servomotors are used for low power applications. AC servomotors are robust in construction and have a lower inertia. However in general they are nonlinear and highly coupled machines and their torque – speed characteristics are not as ideal as those of DC servo motors besides they are low-torque devices compared to DC servomotors of the same size. Most AC servomotors used in control systems are of the two phase squirrel cage induction type. The frequency is normally rated at 60 or 400 Hz the higher frequent is preferred in air bone systems [1].

A schematic diagram of a two phase AC servomotor is shown in Figure (2.4). The stator has two distributed windings displaced 90 electrical degrees apart. One winding called reference or fixed phase is connected to a constant voltage source. The other winding called the control phase is supplied with a variable voltage of the same frequency as the reference phase but is phase –displaced by 90 electrical degrees. The control phase voltage is usually supplied from a servo amplifier. The direction of rotation of the motor depends on the phase relation leading or lagging of the control phase voltage with respect to the reference phase voltage. For balanced two phase voltage, the torque – speed characteristic of the motor is similar to that of the three phase induction motor [1]. For low rotor resistance, this characteristic is nonlinear, as shown in Figure (2.5). such a torque –speed characteristic is unacceptable in control systems, however, if the rotor resistance is high the torque –speed characteristic as shown in Figure(2.5) is essentially linear over a wide speed range ,particularly near zero speed . To control the machine it is operated with fixed voltage for the reference phase and variable voltage for the control phase. The torque – speed characteristic are essentially linear (high rotor resistance is assumed) for various control phase voltages.

In low power control applications (below a few watts) a special rotor construction is use d to reduce the inertia of the rotor [1].

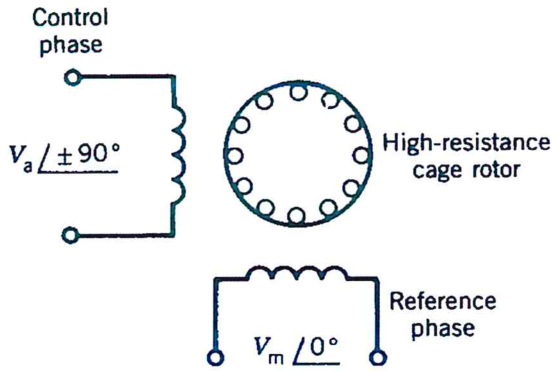

Figure 2.4: Two phase AC servomotor

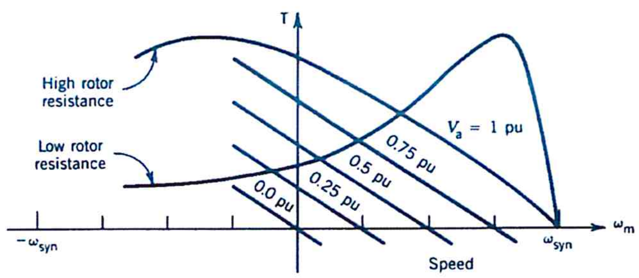

Figure 2.5: Torque speed characteristics

2.2.2 Operation of DC servomotor

The DC servomotor has some control circuits and a potentiometer (a variable resistor, aka pot) that is connected to the output shaft. The potentiometer allows the control circuitry to monitor the current angle of the servomotor. If the shaft is at the correct angle, then the motor shuts off. If the circuit finds that the angle is not correct, it will turn the motor the correct direction until the angle is correct. The output shaft of the servo is capable of travelling somewhere around 180 degrees. Usually, it's somewhere in the 210 degree range, but it varies by manufacturer. A normal servo is used to control an angular motion of between 0 and 180 degrees. A normal servo is mechanically not capable of turning any farther due to a mechanical stop built on to the main output gear. The amount of power applied to the motor is proportional to the distance it needs to travel. So, if the shaft needs to turn a large distance, the motor will run at full speed. If it needs to turn only a small amount, the motor will run at a slower speed. This is called proportional control [14].

2.2.3 Advantages and disadvantages of servomotor

Servomotors have many advantages and disadvantages:

- **Advantages**

i. If a heavy load is placed on the motor, the driver will increase the current to the motor coil as it attempts to rotate the motor. Basically, there is no out-of-step condition. (However, too heavy a load may cause an error).

ii. High-speed operation is possible.
iii. Higher efficiency.
iv. Reduce noise.
v. Reliability.
vi. Longer life time.
vii. Deduction of electromagnetic interface with number of winding on rotor.

- **Disadvantages**

i. Since the servomotor tries to rotate according to the command pulses, but lags behind, it is not suitable for precision control of rotation.
ii. Higher cost.
iii. When stopped, the motor's rotor continues to move back and forth one pulse, so that it is not suitable if you need to prevent vibration [7].

2.2.4 Applications

Servomotors have many applications in automotive market and industrial and customer markets:

- **In automotive market:**

i. Power mirror positioning.
ii. Power seats positioning motors.
iii. Power door and trunk lock mechanisms.
iv. Windshield wiper motors.
v. Heating, Ventilation, and Air Conditioning (HVAC) vent controls.
vi. Power sliding door, sunroof, and convertible top actuators.
vii. Headlight positioning and leveling actuators.

- **In industrial and consumer markets:**

i. Proportioning valves for gasses and liquids.
ii. Paper and materials handling equipment.
iii. HVAC ventilation control.
iv. Entertainment equipment (powered, remotely controlled volume controls for audio receivers and mixers).

2.3 PID Controller

A proportional–integral–derivative controller (PID controller) is a general control loop feedback mechanism controller commonly used in industrial control systems – a PID is the most usually used feedback controller. A PID controller calculates an "error" value as the difference between a measured process variable and a desired set point. The controller attempts to minimize the error by adjusting the process control inputs [8-9].

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: the proportional depends on the present error, the integral on the accumulation of past errors, and derivative 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 [8-9].

In the absence of knowledge of the underlying process, a PID controller is the best controller. By tuning the three parameters in the PID controller algorithm, the controller can provide control action designed for specific process requirements. The response of the controller can be described in terms of the responsiveness of the controller to an error, the degree to which the controller overshoots the set point and the degree of system oscillation. Note that the use of the PID algorithm for control does not guarantee optimal control of the system or system stability [8-9].

Some applications may require using only one or two actions to provide the appropriate system control. This is achieved by setting the other parameters to zero. A PID controller will be called a PI, PD, or I controller in the absence of the respective control actions. PI controllers are fairly common, since derivative action is sensitive to measurement noise, whereas the absence of an integral term may prevent the system from reaching its target value due to the control action.

The value of PID controls lies in their general applicability to most control systems. 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 provide optimal control.

2.3.1 PID controller theory

PID Control is by far the widest type of automatic control used in industry. Even though it has a relatively simple algorithm/structure, there are many subtle variations in how it is applied in industry. A PID controller is a generic control loop feedback mechanism widely used in industrial control systems. A PID controller will correct the error between the output and the desired input or set Point by calculating and give an output of correction that will adjust the process accordingly. A PID controller has the general form:

$$\mathrm{u(t)} = K_p e(t) + K_i \int_0^t e(\tau)\, d\tau + K_d \frac{de}{dt} \qquad (2.1)$$

Where:

K_p is proportional gain.

K_i is the integral gain.

K_d is the derivative gain.

The PID controller calculation (algorithm) involves three separate parameters; the Proportional, the Integral and Derivative values.

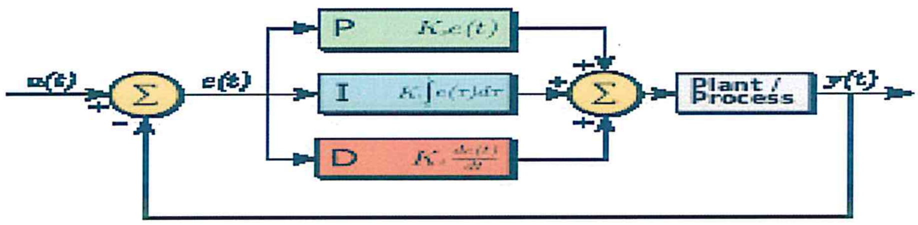

Figure 2.6 Control loop basics

2.3.2 Proportional control action

For a controller with proportional control action, the proportional term makes a change to the output that is proportional to the current error value .The proportional response can be adjusted by multiplying the error by a constant K_p, called the proportional gain

$$P_{\text{out}} = K_p e(t) \tag{2.2}$$

Whatever the actual mechanism may be and whatever the form of the operating power, the proportional controller is essentially an amplifier with an adjustable gain. A high proportional gain results in a large change in the output for a given change in the error. If the proportional gain is too high, the system can become unstable. In contrast, a small gain results in a small output response to a large input error, and a less responsive or less sensitive controller. If the proportional gain is too low, the control action may be too small when responding to system disturbances. Tuning theory and industrial practice indicate that the proportional term should contribute the size of the output change [9-10].

2.3.3 Integral control action

The contribution from the integral term is proportional to both the magnitude of the error and the duration of the error. The integral in a PID controller is the sum of the instantaneous error over time and gives the accumulated offset that should have been corrected previously. The accumulated error is then multiplied by the integral gain K_i and added to the controller output. The integral term is given by:

$$I_{\text{out}} = K_i \int_0^t e(\tau)\, d\tau \tag{2.3}$$

The integral term accelerates the movement of the process towards set point and eliminates the residual steady-state error that occurs with a pure proportional controller. However, since the integral term responds to accumulated errors from the past, it can cause the present value to overshoot the set point value [9-10].

2.3.4 Derivative control action

The derivative of the process error is calculated by determining the slope of the error over time and multiplying this rate of change by the derivative gain k_d . The magnitude of the contribution of the derivative term to the overall control action is termed the derivative gain K_d. The derivative term is given by:

$$D_{\text{out}} = K_d \frac{d}{dt} e(t) \tag{2.4}$$

The derivative term slows the rate of change of the controller output. Derivative control is used to reduce the magnitude of the overshoot produced by the integral component and improve the combined controller-process stability. However, the derivative term slows the transient response of the controller. Also, differentiation of a signal amplifies noise and thus this term in the controller is highly sensitive to noise in the error term, and can cause a process to become unstable if the noise and the derivative gain are sufficiently large [9-10].

2.3.5 Important of PID

PID control remains an important control tool for three reasons: past record of success, wide availability and simplicity in use. These reasons reinforce one another, thereby ensuring that the more general framework of digital control with higher order controllers has not really been able to displace PID control. It is really only when the process situation demands a more sophisticated controller or a more involved controller solution to control a complex process that the control engineer uses more advanced techniques. Even in the case where the complexity of the process demands a multi-loop or multivariable control solution, a network based on PID control building blocks is often used [10].

2.3.6 Stability

If the PID controller parameters (the gains of the proportional, integral and derivative terms) are chosen incorrectly, the controlled process input can be unstable [8]. Generally, stability of response is required and the process must not oscillate for any combination of process conditions and set points, though sometimes marginal stability is acceptable or desired.

2.3.7 PID Tuning

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 system. It is well know that the basic and modified PID control schemes have proved their usefulness in providing satisfactory control, although in many given situations they may not [16]. Tuning a control loop is the adjustment of its control parameters (gain/proportional band, integral gain/reset, derivative gain/rate) to the optimum values for the desired control response. Stability (bounded oscillation) is a basic requirement, but further than, different systems have different behavior, different applications have different requirements, and requirements may conflict with one another.

PID tuning is a difficult problem, even though there are only three parameters and in principle is simple to describe, because it must satisfy complex criteria within the limitations of PID control. There are accordingly various methods for loop tuning, and more sophisticated techniques are the subject of patents [9].

2.3.8 Tuning objectives and existing methods

Pre-selection of a controller structure can pose a challenge in applying PID control. As vendors often recommend their own designs of controller structures, their tuning rules for a specific controller structure does not necessarily perform well with other structures. One solution seen is to provide support for individual structures in software. Nonetheless; controller parameters are tuned such that the closed-loop control system would be stable and would meet given objectives associated with the following:

- Stability robustness.
- Set-point following and tracking performance at transient, including rise-time, overshoot, and settling time.
- Regulation performance at steady-state, including load disturbance rejection.

• Robustness against plant modeling uncertainty.

• Noise attenuation and robustness against environmental uncertainty.

Tuning methods for PID controllers can be grouped according to their nature and usage, as follow:

- ***Analytical methods***

PID parameters are calculated from analytical or algebraic relations between a plant model and an objective (such as internal model control or lambda tuning). These can lead to an easy-to-use formula and can be suitable for use with online tuning, but the objective needs to be in an analytical form and the model must be accurate.

- ***Heuristic methods***

These are evolved from practical experience in manual tuning (such as the Ziegler-Nichols tuning rule) and from artificial intelligence (including expert systems, fuzzy logic and neural networks). Again, these can serve in the form of a formula or a rule base for online use, often with tradeoff design objectives.

- ***Frequency response methods***

Frequency characteristics of the controlled process are used to tune the PID controller (Such as loop-shaping). These are often offline and academic methods, where the main concern of design is stability robustness.

- ***Optimization methods***

These can be regarded as a special type of optimal control, where PID parameters are obtained ad hoc using an offline numerical optimization method for a single composite objective or using computerized heuristics or an evolutionary algorithm for multiple design objectives. These are often time-domain methods and mostly applied offline.

• ***Adaptive tuning methods***

These are for automated online tuning, using one or a combination of the previous methods based on real-time identification [17].

2.3.9 Effects of PID controllers' parameters

Effects of PID controllers' parameters K_p, K_i and K_d on a closed loop system are summarized in the Table (2.1).

Table 2.1: Effects of PID controllers' parameters K_p, K_i and K_d

Closed loop Response	Rise Time(sec.)	Maximum Overshoot (%)	Settling Time(sec.)	Steady State Error	Stability
Increase of K_p	Decrease	Increase	Small change	Decrease	Degrade
Increase of K_i	Decrease	Increase	Increase	Eliminate	Degrade
Increase of K_d	Small change	Decrease	Decrease	Small change	Improve

2.4 Fuzzy Logic

Fuzzy logic was initiated in 1965, by Lotfi A. Zadeh, professor for computer science at the University of California in Berkeley [2]. The fuzziness comes from uncertainty about the domain being represented consequently There is nothing fuzzy at all about the logic, which is defined in mathematical terms that are precise, repeatable, and programmable. Essentially, Fuzzy logic (FL) is a multi-valued logic that allows intermediate values to be defined between conventional evaluations like true or false, Yes or no, high or low, etc. Fuzzy logic is a tool to help you control complex systems. Commercially, fuzzy logic has been used with great achievement to control machines and consumer products. To apply a fuzzy logic system, you need to set ideas to be presented through using of it; otherwise you are using the wrong tool. Additionally in case of regular controller, certain performance concerns should be addressed. Fuzzy logic controllers are being widely used to control nonlinear systems.

2.4.1 Fuzzy elements

Fuzzy control system is based on four main parts shown in Figure (2.7):

i. The fuzzification interface simply modifies the inputs so that they can be interpreted and compared to the rules in the rule-base.

ii. The "rule-base" holds the knowledge, in the form of a set of rules, of how best to control the system.

iii. The inference mechanism evaluates which control rules are relevant at the current time and then decides what the input to the plant should be.

iv. Defuzzification interface converts the conclusions reached by the inference mechanism into the inputs to the plant [4].

Basically, you should view the fuzzy controller as an artificial decision maker that operates in a closed-loop system in real time. It gathers plant output data, compares it to the reference input, and then decides what the plant input should be to ensure that the performance objectives will be met [4].

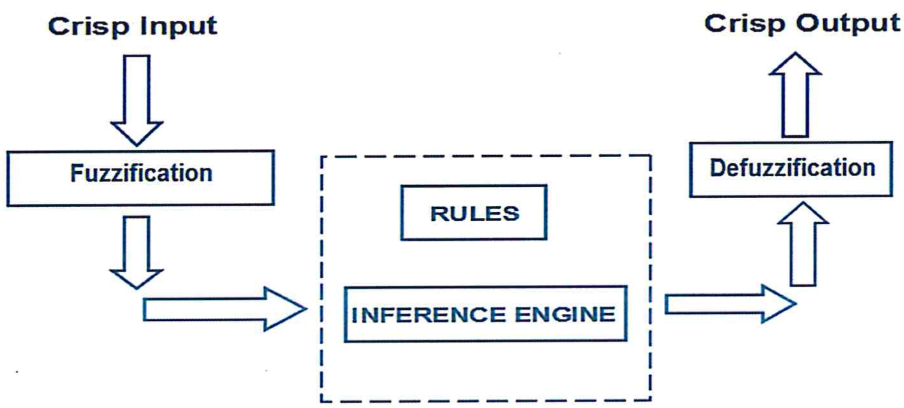

Figure 2.7: Fuzzy structure

2.4.2 Fuzzy sets theory

The word fuzzy means vagueness. Fuzziness takes place when the boundary of a piece of information is not clear-cut. Fuzzy set has been introduced as an extension of classical concept of set, where elements have degrees of membership [6].

Classical set theory allows the member ship of the element in the set in binary terms, a bivalent condition –an element either belongs or do not belong to the set.

Fuzzy set theory permits the gradual assessment of the membership of the element s in a set described of the aid of a membership function valued in the real unit interval [0, 1]. Fuzzy sets generalize classical sets, since the indicator functions of classical sets are special cases of the membership functions of fuzzy sets, if the latter only take values 0 or 1. In fuzzy set theory, classical bivalent sets are usually called crisp sets. The fuzzy set theory can be used in a wide range of domains in which information is incomplete or imprecise, such as bioinformatics [5].

Fuzzy set theory, on the other hand, guarantees a smooth transition between adjacent classes allowing a continuous degree of membership ranging from 0, element completely out of the set, to 1, element completely in the set [5].

For a given fuzzy set A defined over a space X, with x being a generic element, the membership function f A(x) is defined as a function assigning to each element x its degree of membership μA(x) to set A, so:

$$f\,A(x) : x \in X \rightarrow \mu A(x) \in [0; 1] \qquad (2.6)$$

Fuzzy set theory and fuzzy logic are not trying to offer an alternative to probability theory since they are modeling vagueness rather than Uncertainty [5].

2.4.3 Fuzzy subset

In fuzzy logic, a fuzzy subset A of a set B is defined by a "membership function" which gives the degree of membership of each element of B belonging to A. There is a strong relationship between Boolean logic and the concept of a subset. There is a similar strong relationship between fuzzy logic and fuzzy subset theory [5].

A subset U of a set S can be defined as a set of ordered pairs, each with a first element that is an element of the set S, and a second element that is an element of the set {0, 1 }, with exactly one ordered pair present for each element of S. This defines a mapping between elements of S and elements of the set {0, 1}. The value zero is used to represent non-membership, and the value one is used to represent membership. The truth or falsity of the statement x is in U is determined by finding the ordered pair whose first element is x. The statement is true if the second element of the ordered pair is 1, and the statement is false if it is 0.

2.4.4 Crisp sets

In most everyday kinds of thinking, and linguistic reflections of that, people use crisp sets to categorize things. Being a member of a crisp set is an all or nothing affair. Thinking with crisp sets makes everything simpler, because something either is or is not a member of a crisp set. They can be used to represent black and white conceptual thinking. Oftentimes too, when something is a member of a given crisp set it is then (at the same time) not a member of any other crisp set. Again this simplifies the logic used with this kind of thinking process. Linguistic constructions which reflect this kind of thinking can be quite useful. Especially when crisp categories are used. A set A, is well described by a function called characteristic function, this function defined on the universal space X, assumes:

A value of 1 for those elements x that belong to set A & value of 0 for those elements x that do not belong to set A the notations used to express these mathematically are:

$$\left.\begin{array}{l} A: X \rightarrow [0, 1] \\ A(x) = 1, \text{ x is a member of A} \\ A(x) = 0, \text{ x is not a member of A} \end{array}\right\} \quad (2.7)$$

Alternatively, the set A can be represented for all element s x € X by its characteristic function μ A(x) defined as:

$$\left\{\begin{array}{l} 1 \text{ if } \quad x \text{ €} X \\ \mu\, A(x) = 0 \text{ otherwise} \end{array}\right. \quad (2.8)$$

Thus in classical set theory μ A(x) has only the values 0" false" and 1 "true". Such sets are called crisp sets. In other words The Crisp sets are special cases of fuzzy sets.

2.4.5 Universal space

The universal space for fuzzy sets in fuzzy logic was defined only in integers, now the universal space of fuzzy sets and fuzzy relations is defined with three numbers, the first two numbers specify the start and end of the universal space and the third argument specifies the increment between elements, this gives the user more flexibility in choosing the universal space.

2.4.6 Linguistic descriptions

The linguistic description provided by the expert can generally be broken into several parts. There will be "linguistic variables" that describe each of the time varying fuzzy controller inputs and outputs [4].

2.4.7 Fuzzy membership functions

One of the key issues in all fuzzy sets is how to determine fuzzy membership functions it fully defines the fuzzy set provides a measure of the degree of similarity of an element to a fuzzy set it can take any form, but there are some common examples that appear in real applications [5].

2.4.8 Fuzzy operation

- ***Fuzzy Union***

The most commonly used method for fuzzy union is to take the maximum. That is, given two fuzzy sets A and B with membership functions μA(x) and μB(x):

$$\mu_{AUB}(x) = \max(\mu_A(x), \mu_B(x)) \tag{2.9}$$

- ***Fuzzy Intersection***

The most commonly adopted t-norm is the minimum. That is, given two fuzzy sets A and B with membership functions μA(x) and μB(x):

$$\mu_{A\cap B}(x) = \min(\mu_A(x), \mu_B(x)) \tag{2.10}$$

- ***Fuzzy Complement***

To be able to develop fuzzy systems we also have to deal with NOT or complement. This is the same in fuzzy logic as for Boolean logic. For a fuzzy set A, A denotes the fuzzy complement of a Membership function for fuzzy complement is:

$$\mu_{A}(x) = 1 - \mu_A(x) \tag{2.11}$$

2.4.9 Fuzzy logical operations

- ***Logical OR***

$$\mu A \vee B(u,v) = \max\{\mu A(u), \mu B(v)\}, u \in U \wedge v \in V \tag{2.12}$$

- ***Logical AND***

$$\mu A \wedge B(u,v) = \min\{\mu A(u), \mu B(v)\}, u \in U \wedge v \in V \tag{2.13}$$

- ***Logical NOT***

$$\mu A(u) = 1 - \mu A(u), u \in U \tag{2.14}$$

2.4.10 Advantage for using fuzzy controllers

1. Very robust.
2. Can be easily modified.
3. Can use multiple input and outputs sources.
4. Much simpler than its predecessors (linear algebraic equations).
5. Very quick and cheaper to implement [2, 4, 5].

2.4.11 Application areas

Fuzzy systems have been used in a wide variety of applications in engineering science, business, medicine, psychology, and other fields. For instance, in engineering some potential application areas include the following [4]:

- Aircraft/spacecraft: Flight control, engine control, avionics systems, failure diagnosis, navigation, and satellite attitude control.
- Automated highway systems: Automatic steering, braking, and throttle control for vehicles.
- Automobiles: Brakes, transmission, suspension, and engine control.
- Autonomous vehicles: Ground and underwater.
- Manufacturing systems: Scheduling and deposition process control.
- Power industry: Motor control, power control/distribution, and load estimation.
- Process control: Temperature, pressure, and level control, failure diagnosis, distillation column control, and desalination processes.
- Robotics: Position control and path planning.

CHAPTER THREE

SYSTEM CONTROL DESIGN

CHAPTER THREE

SYSTEM CONTROL DESIGN

3.1 System Model

Basically, a model has to be constructed from observed data; Graphical models are made up from certain measurements. To interact with a system, some concept of how its variables relate to each other is needed. With a broad definition, such an assumed relationship among observed signals can be called a model of the system. For certain systems it is appropriate to describe its properties using numerical tables and/or plots. These descriptions are called graphical models. Linear systems, for example, can be uniquely described by their impulse or step responses or by their frequency responses. Graphical representations of these are widely used for various design purposes [12]. The Block diagram of DC servomotor is shown in Figure (3.1) with zero value of load torque (T_L) [19].

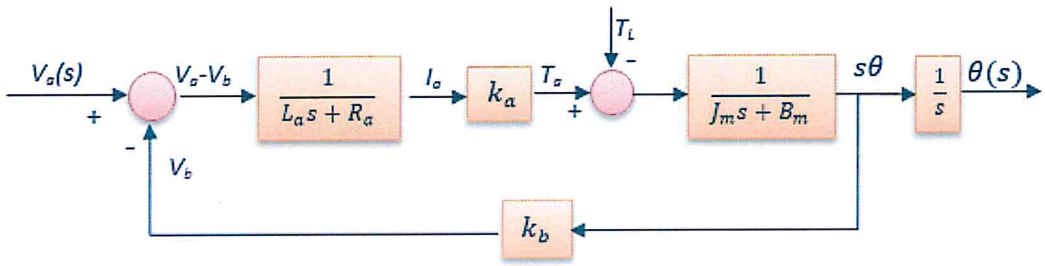

Figure3.1: Block diagram for DC servo motor

3.1.1 System mathematical models

For more advanced applications, it may be necessary to use models that describe the relationships among the system variables in terms of mathematical expressions like difference or differential equations. These models are called mathematical models (or analytical models). Mathematical models may be further characterized by a number of adjectives (time continuous or time discrete, lumped or distributed, linear or nonlinear, etc.) signifying the type of difference or differential equation used. The use of mathematical models is inherent in all fields of engineering and physics. In fact, a major part of the engineering field deals with how to make good designs based on mathematical models [12].

The DC servomotor is used extensively in control systems, for analytical purpose, it is necessary to establish mathematical models for DC motors for applications. Using the equivalent circuit diagram in Figure (3.3) to represent a DC servomotor. The armature is modeled as a circuit with resistance R_a Connected in series with an inductance L_a and a voltage source E_b representing the back emf in the armature when the rotor rotates. The Figure (3.2) shows the schematic representation for DC servo motor:

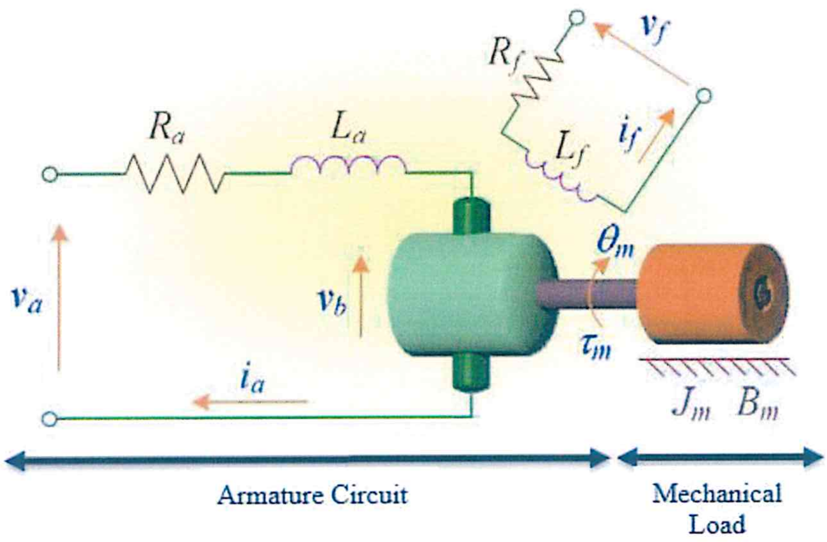

Figure 3.2: Schematic representation of the DC servo motor

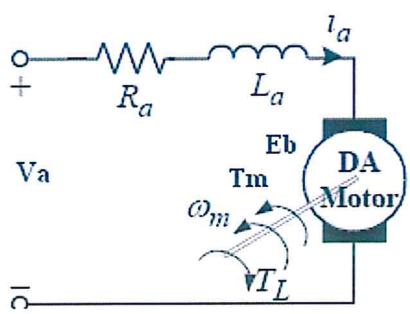

Figure 3.3: DC servomotor equivalence circuit

The magnetic flux ψ generated by the motor field circuit is proportional (K_f) to the field current (i_f)

$$\psi = k_f i_f \tag{3.1}$$

The torque τ_m developed by the motor is proportional (K_1) to the magnetic flux and the armature current i_a:

$$\tau_m = k_1 i_a \psi \tag{3.2}$$

If the field current is constant, the magnetic flux becomes a constant (K_2), so Equation (3.2) can be written as:

$$\tau_m = (k_1 k_2) i_a \tag{3.3}$$

And considering a $K_a = K_1 K_2$

Where

K_a: motor torque constant

The torque developed by the motor can be written as:

$$\tau_m = k_a i_a \tag{3.4}$$

When the armature rotates, a voltage is induced on it, called inverse electromotive force (V_b) that is proportional (K) to the product between the magnetic flux and angular speed $(\dot{\theta}_m)$ in the form:

$$v_b = k \psi \dot{\theta}_m \tag{3.5}$$

For a constant magnetic flux, the inverse electromotive force is directly proportional to the angular speed, considering $k_b = k\psi$, the inverse electromotive force can be written as:

$$v_b = k\psi\dot{\theta}_m \tag{3.6}$$

Where

K_b: Constant of inverse electromotive force.

Applying Kirchhoff laws (KVL) to the armature circuit, we obtain:

$$v_a = L_a\frac{di_a}{dt} + R_a i_a + v_b \tag{3.7}$$

In the domain S and clearing the armature current in Equation (3.7), we have:

$$I_a = \frac{V_a - V_b}{L_a s + R_a} \tag{3.8}$$

Else:

$$I_a = \frac{V_a - k_b s\Theta_m}{L_a s + R_a} \tag{3.9}$$

The motor torque equation is:

$$\tau_m = J_m\ddot{\theta}_m + B_m\dot{\theta}_m \tag{3.10}$$

Where:

J_m : Motor inertial momentum.

B_m : Motor viscous friction.

When matching Equations (3.4) and (3.10), we have:

$$k_a i_a = J_m \ddot{\theta}_m + B_m \dot{\theta}_m \tag{3.11}$$

In the domain S and clearing the armature current in Equation (3.11) we have:

$$I_a = \frac{J_m s^2 \Theta_m + B_m s \Theta_m}{k_a} \tag{3.12}$$

Matching Equations (3.8) and (3.12), we obtain:

$$k_a V_a - k_a k_b s \Theta_m = (L_a s + R_a)(J_m s + B_m) s \Theta_m \tag{3.13}$$

$$V_a = \frac{(L_a s + R_a)(J_m s + B_m) s \Theta_m + k_a k_b s \Theta_m}{k_a} \tag{3.14}$$

$$\frac{\Theta_m}{V_a} = \frac{k_a}{\left[(L_a s + R_a)(J_m s + B_m) + k_a k_b\right] s} \tag{3.15}$$

Equation (3.15) points the relationship between the DC motor spindle position and the supply voltage applied on its armature [18].

3.1.2 System simulink model

Simulink is a software package that enables modeling, simulating, and analyzing systems whose outputs change over time. Such systems are often referred to as dynamic systems. Simulink can be used to investigate the behavior of a wide range of real-world dynamic systems, including electrical circuits, shock absorbers, braking systems, and many other electrical, mechanical, and thermodynamic systems. A Simulink block diagram model is a graphical representation of a mathematical model of a dynamic system. Here explain of how Simulink works. Simulating a dynamic system is a two-step process with Simulink. First, a user creates a block diagram, via the Simulink model editor, that graphically depicts time-dependent mathematical relationships between the system's inputs, states, and outputs. The user then commands Simulink to simulate the system represented by the model from a specified start time to a specified stop time [11].

Servomotors parameters are defined as follows:

J_m = 0.000052 kg.m²

B_m =0.01 N.ms

K_b =0.0235 V /rad S^{-1}

K_a = 0.0235 Nm/A

R_a =2 ohm

R_a = 0.23 H

The purpose of modeling DC servomotor is to approach the actual Dc servomotor. By including the parameters, the transfer function of DC servomotor for controlling position will be [19]:

$$G\ \text{position}\ (s) = \frac{\theta(s)}{Va(s)} = \frac{19649}{S^3+201S^2+6290S} \qquad (3.16)$$

The transfer function for controlling speed is:

$$G\ \text{speed}\ (s) = \frac{\omega(s)}{Va(s)} = \frac{19649}{S^2+201S+6290} \qquad (3.17)$$

The following block diagram shows the Simulink model of uncontrolled system in Figure (3.4).

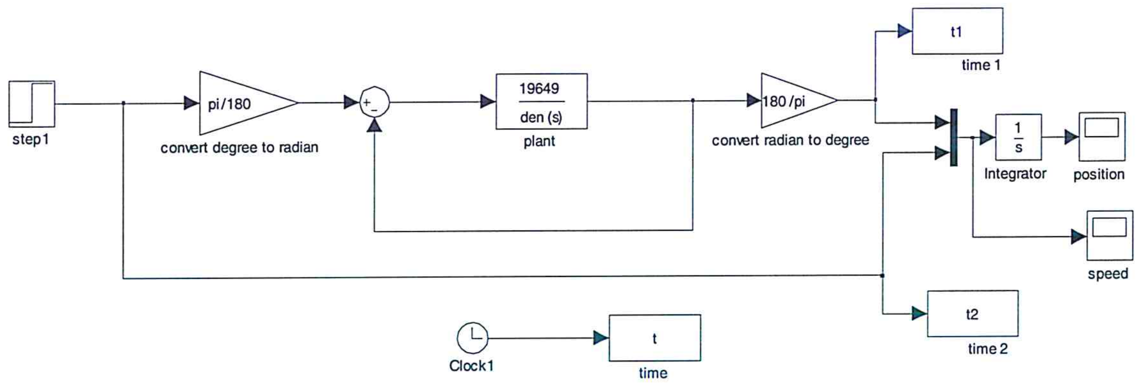

Figure 3.4: Simulink model for uncontrolled system

3.2 PID Controller Design

PID controller design by substituting the Equation (3.17) in MATLAB SIMULINK techniques to obtain three term control: the integral proportional and derivative values of PID controller that will meet the transient and steady state specifications of the closed loop system. The Block diagram shows the Simulink model of controlled system with PID controller in Figure (3.5).

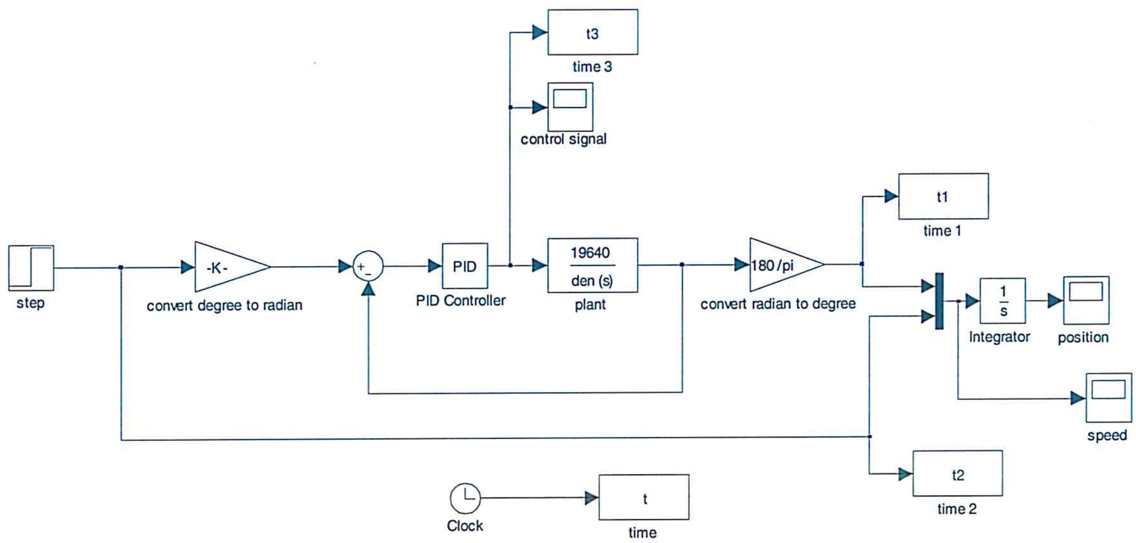

Figure 3.5: Simulink model for system with PID controller

PID tuning and loop optimization software are used to ensure consistent results. These software packages will gather the data, develop process models, and suggest optimal tuning. Some software packages can even develop tuning by gathering data from reference change.

A PID controller is being designed for a higher order system with transfer function in Equation (3.17). Before starting the simulation and implementation PID controller, constructing the design of PID controller by using Matlab's SISO design tool. The design architecture of Mat lab's SISO design tool for automated PID tuning is shown in Figure (3.6).

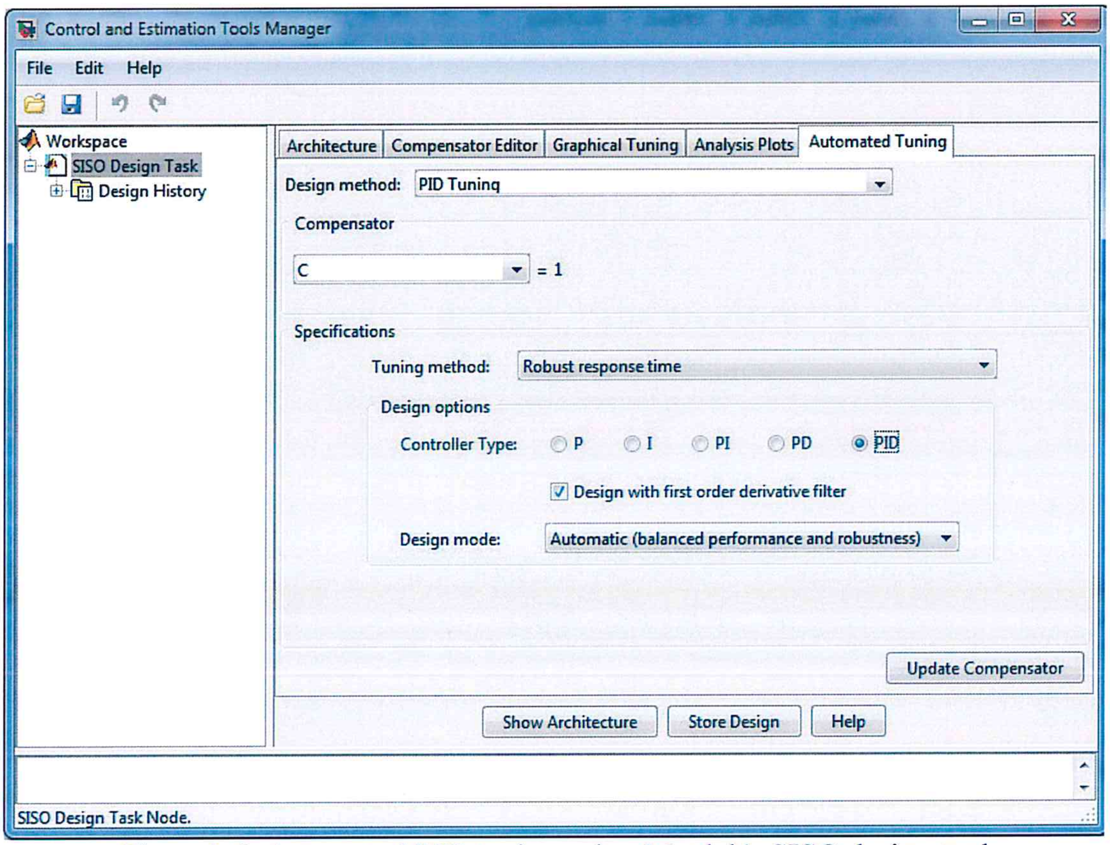

Figure3.6: Automated PID tuning using Mat lab's SISO design tool

The values of parameters optimization are k_p =5, k_d = -0.4156 and k_i =zero

3.3 Fuzzy Controller Design

Fuzzy control system design essentially amounts to: Choosing the fuzzy controller inputs and outputs. Choosing the preprocessing that is needed for the controller inputs and possibly post processing that is needed for the output. Designing each of four components of fuzzy controller.

There are standard choices of the fuzzification and Defuzzification interfaces moreover, most often the designer settles on an inference mechanism and may use this for many different processes. Hence, the main part of the fuzzy controller that we focus on for design is the rule base. 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 an actual human expert who has spent a long time learning how best to control the process. In other situation there is no such human expert, and the control engineer will simply study the plant dynamics (perhaps using modeling and simulation) and write down a set of control rules that makes sense [20].

The error (E) and change in error (CE) are taken as the two inputs for the fuzzy controller and one output control input.

The block diagram shows the system simulink model for controlled system with fuzzy logic controller in Figure (3.7).

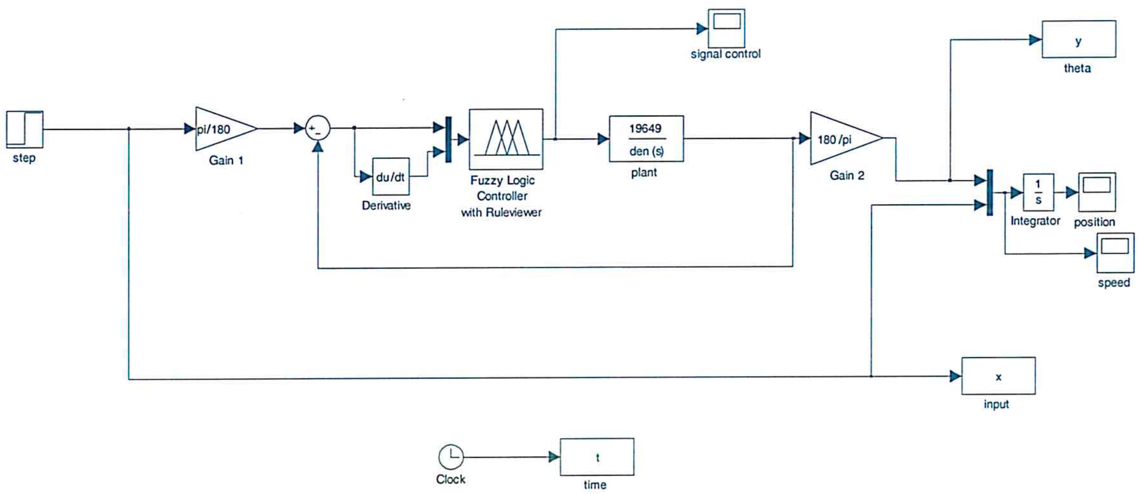

Figure 3.7: Simulink model for simulating fuzzy logic controller

3.3.1 Fuzzy basis FIS editor

The FIS editor shows high information about a fuzzy interface system shown in Figure (3.8) at the top is a diagram of the system with each input and output clearly labeled. The membership function editor can be gotten by double clicking on the input or output boxes, also the rule editor will be gotten by double clicking on the fuzzy rule box in the center of diagram. Just below the diagram is a text field that displays the name of the current FIS. The various functions used in the fuzzy implication process was allowed by series of popup menus in the lower left of the window. In the lower right are fields that provide information about the current variable. The current variable is determined by clicking once on the input or output boxes.

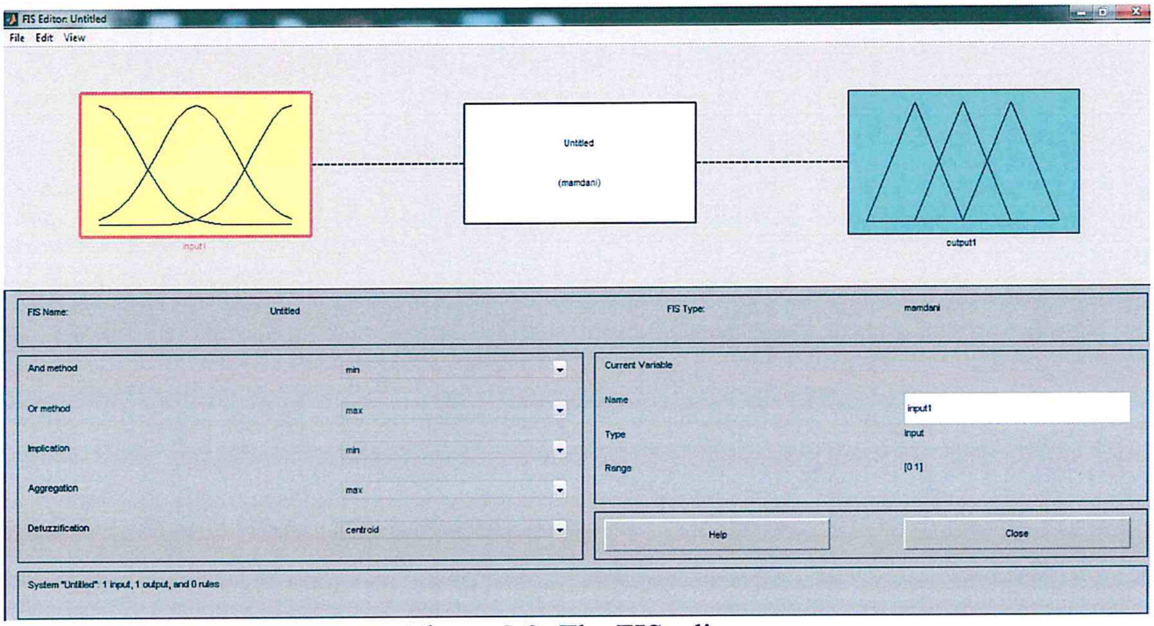

Figure 3.8: The FIS editor

The system needs two inputs, by adding second input from edit menu then FIS give two inputs variables which one is error (E) and another is change of error (CE). Each one consists of number of membership functions and seven rules. The Figure (3.9) shows the FIS editor after adding the second input.

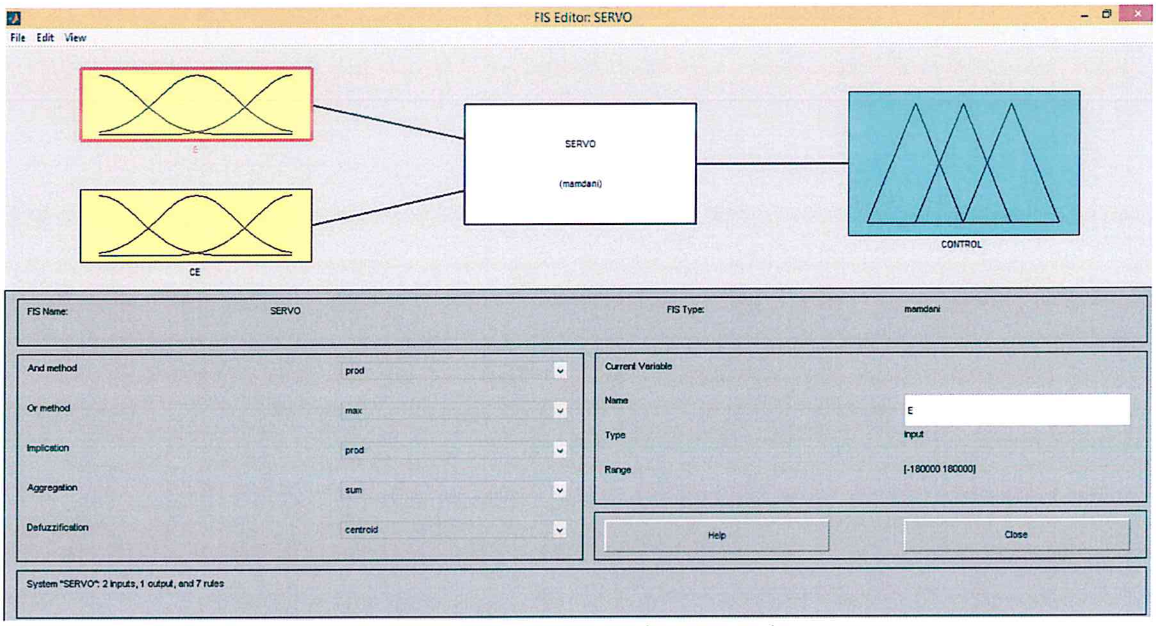

Figure 3.9: FIS editor with two inputs and one output

3.3.2 The membership function editor

The membership function editor shares some features with the FIS editor. The membership function editor is the tool that make the programmer displays and edits all of the membership function related with all inputs and output variables for entire fuzzy interface system [21].

When opening the membership functions editor to work on a fuzzy interface system that does not exist in the work space, there is no membership function associated with variables .The membership function (MF) editor is used to create, remove and modify the MF's for a given fuzzy system current variable was selected by clicking once on one of the displayed boxes on the left side of the diagram (Variable palette). The information for the selected variable id displayed in the text area blow the palette zone. After adjusting the shape of the membership function, click on the Membership function then the desired parameters will appear, then editing the name and type and range for the membership function, the will look as in Figure (3.10) showing the error (E) and by repeating the steps the change of error (CE) and the output are shown in Figure (3.11) and (3.12) respectively.

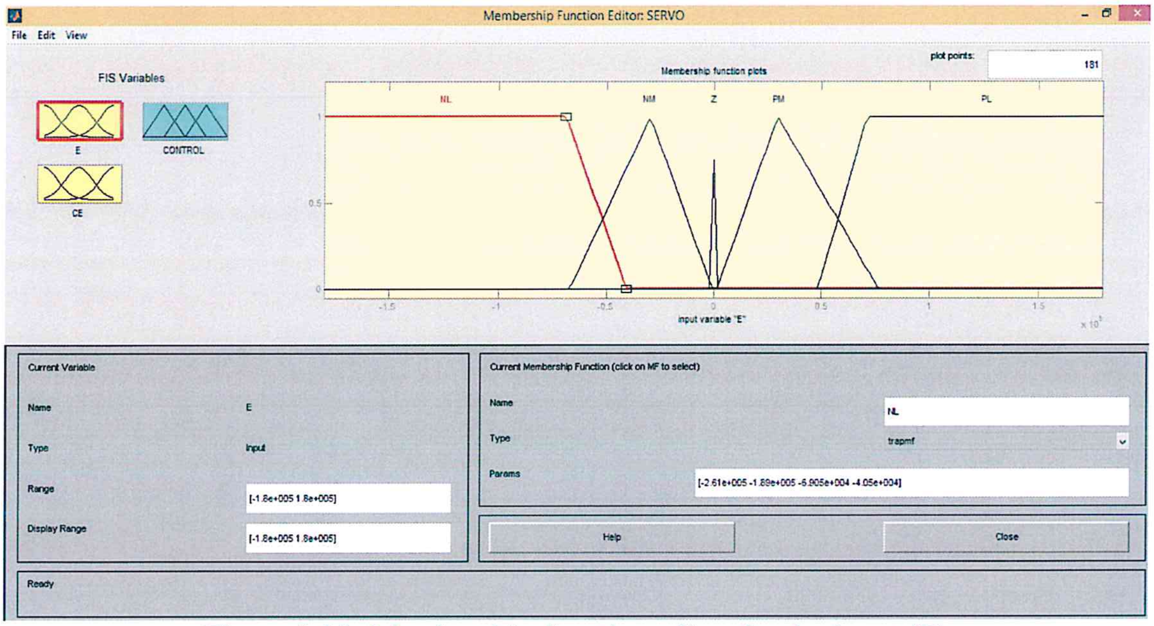

Figure 3.10: Membership function editor for the input (E)

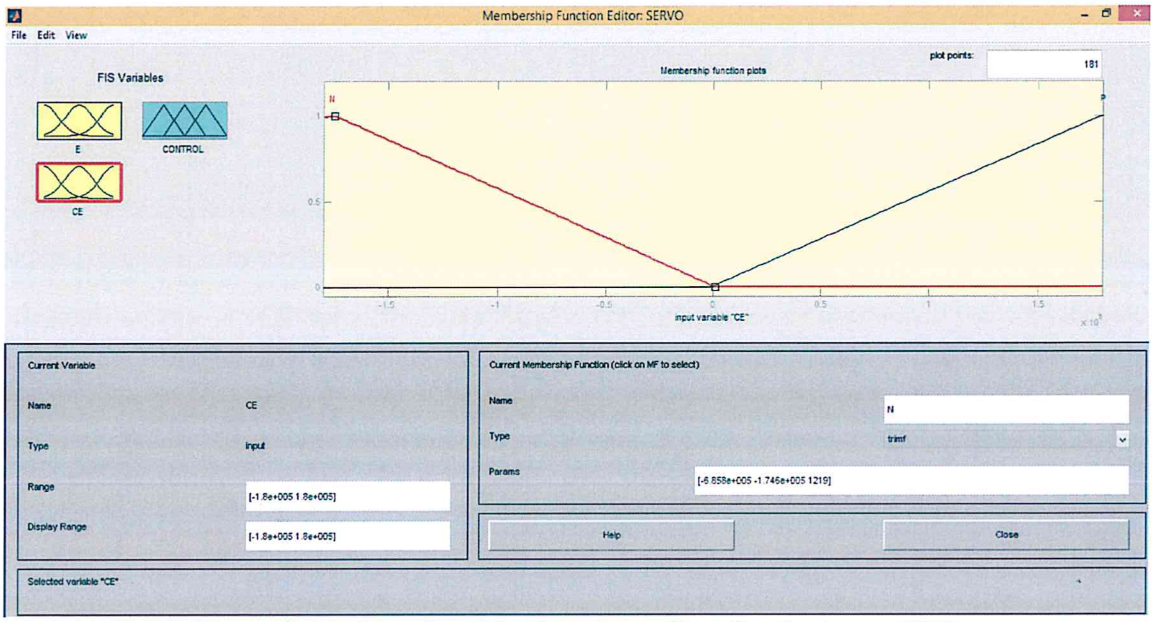

Figure 3.11: Membership function editor for the input (CE)

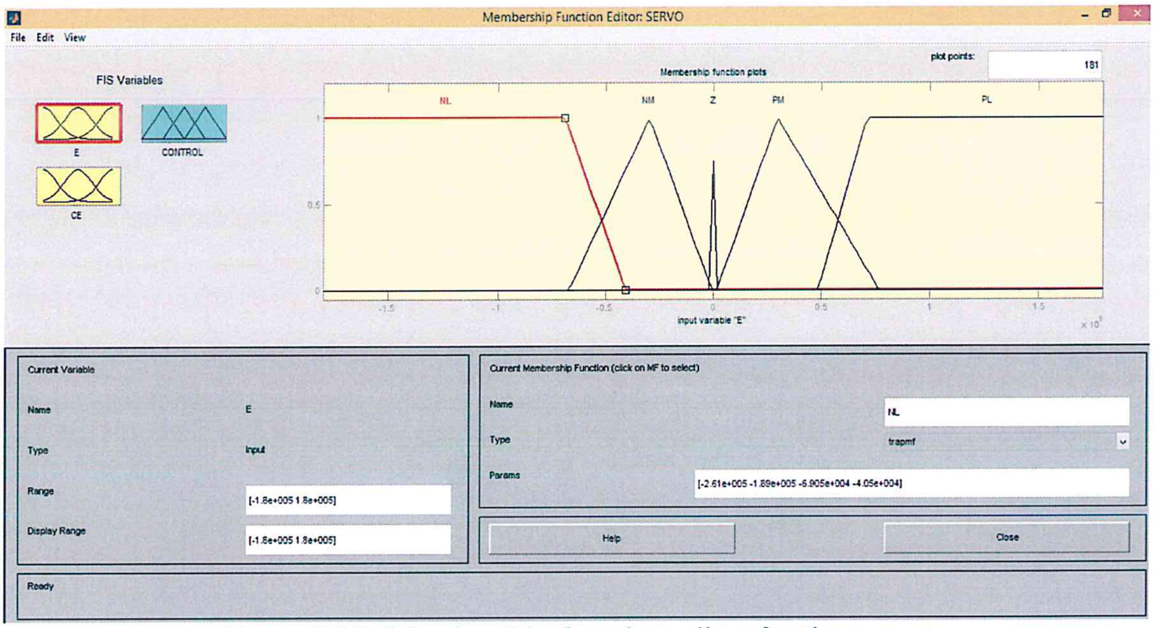

Figure 3.12: Membership function editor for the output

3.3.3 Rule base editor

Rule editor are shown by choosing edit menu rules edit selected.

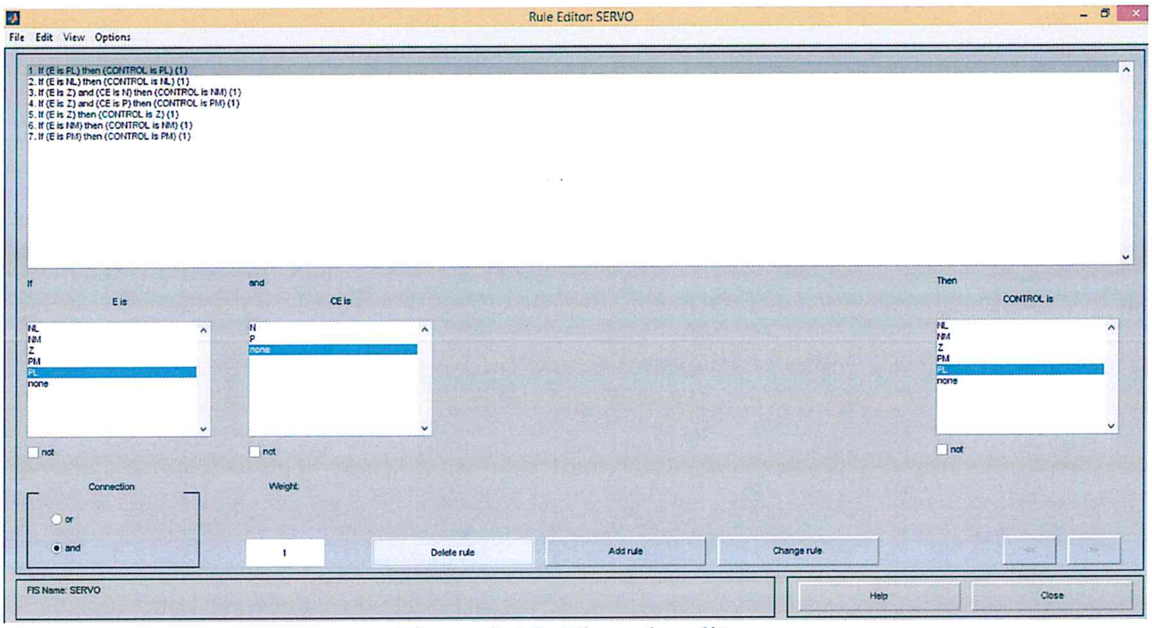

Figure 3.13: The rule editor

Based on description of the inputs and output variables defined with FIS editor, the rule editor allows to construct the rule statements automatically from GUI as shown in

Figure (3.13). For creating rules select the inputs and output linguistic value and connecting them by clicking add rule. There are 7 rules that used at the controllers which are based on human experience and information, here NL means Negative Large, NM means Negative Medium, Z means Zero, PM means Positive Medium, and PL means Positive Large, N means Negative and P means Positive.

1. If E is PL then CONTROL is PL
2. If E is NL then CONTROL is NL
3. If E is Z and CE is N then CONTROL is NM
4. If E is Z and CE is P then CONTROL is PM
5. If E is Z then CONTROL is Z
6. If E is NM then CONTROL is NM
7. If E is PM then CONTROL is PM

3.3.4 Rule viewer

The rule viewer shown in Figure (3.14) displays a whole fuzzy interface process. It based on fuzzy interface .each role is a row in plots and each column is a variable. The rule numbers are displayed on the left of each row.

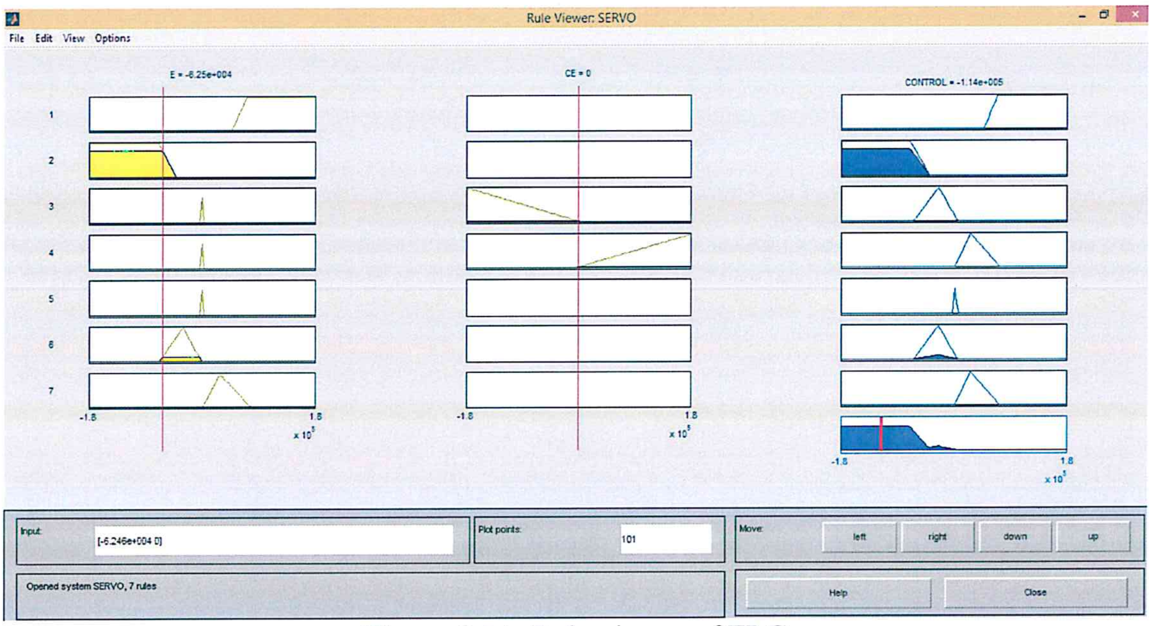

Figure 3.14: Rule viewer of FLC

3.3.5 Surface Viewer

The surface viewer of the rules is shown in Figure (3.15), presents the inputs and one output system generate a three dimensional plot that MATLAB can adaptively manage. When we move beyond three dimensional over all, we start to meeting trouble displaying the result. Consequently the surface viewer is equipped with pop- up menus that let you select any two inputs and any one input for plotting. Just below the pop-up menus are two text inputs fields that let you determine how many x – axis and y-axis grid lines you need to include. These allows to keep the calculation time reasonable for complex problems.

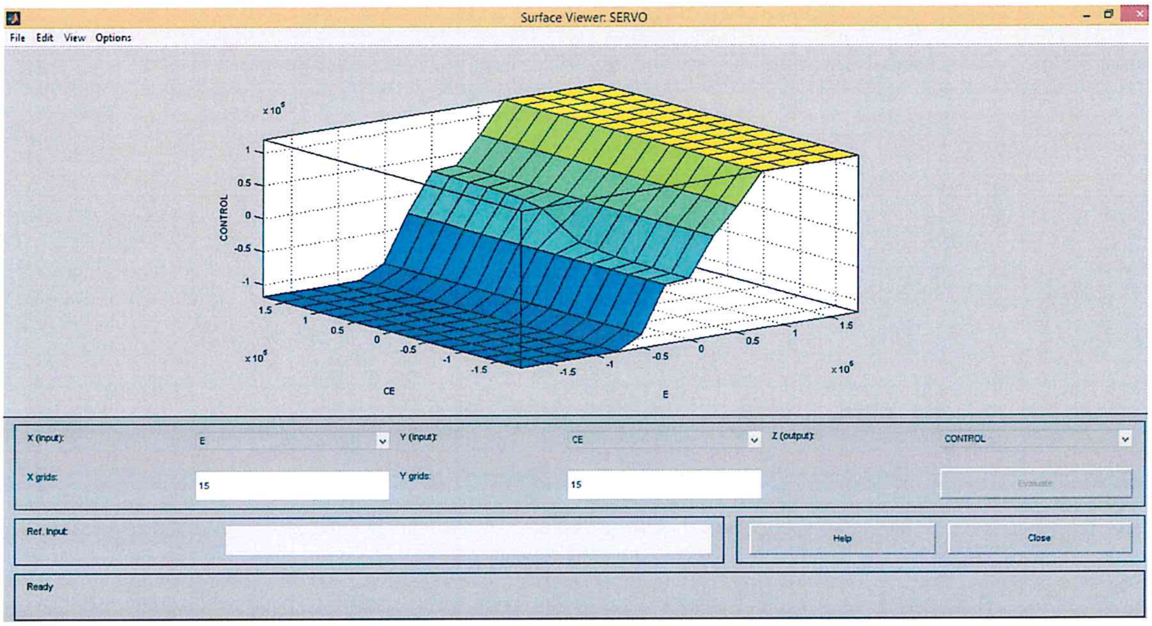

Figure 3.15: Surface view of FLC

CHAPTER FOUR

SYSTEM SIMULATION RESULTS

CHAPTER FOUR

SYSTEM SIMULATION RESULTS

4.1 Introduction

This chapter describes the simulation work as well as the experimental work. A DC servomotor system has been taken as a case study and its transfer function has been derived in Chapter three. The comparison between the two controller the PID and FLC as well as the original model is illustrated under MATLAB environment.

4.2 System Simulation Results without Controller

To create the simulation for DC servomotor without controller, the transfer function of Equation (3.17) in simulink model Figure (3.4) was taken as the speed system. The unit step applied to the system as input signal and the position output was displayed in Figure (4.1).

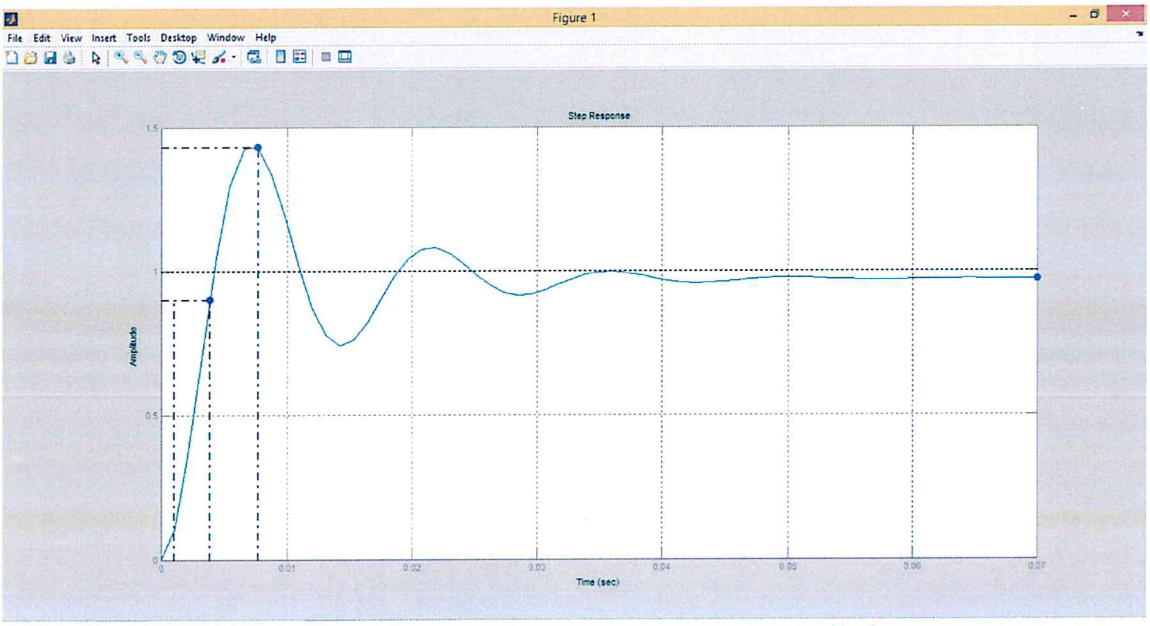

Figure 4.1: Position unit step response of uncontrolled system

Based on Figure (4.1), the final value is 1, the rise time (tr) is 0.00286 second, the maximum overshoot (Mp) is 43.1 % at 0.00766 second, peak amplitude is 1.43 and non-settling time (ts), the system is unstable.

By using the same simulink model in Figure (3.4) the simulation result for speed can be seen in Figure (4.2).

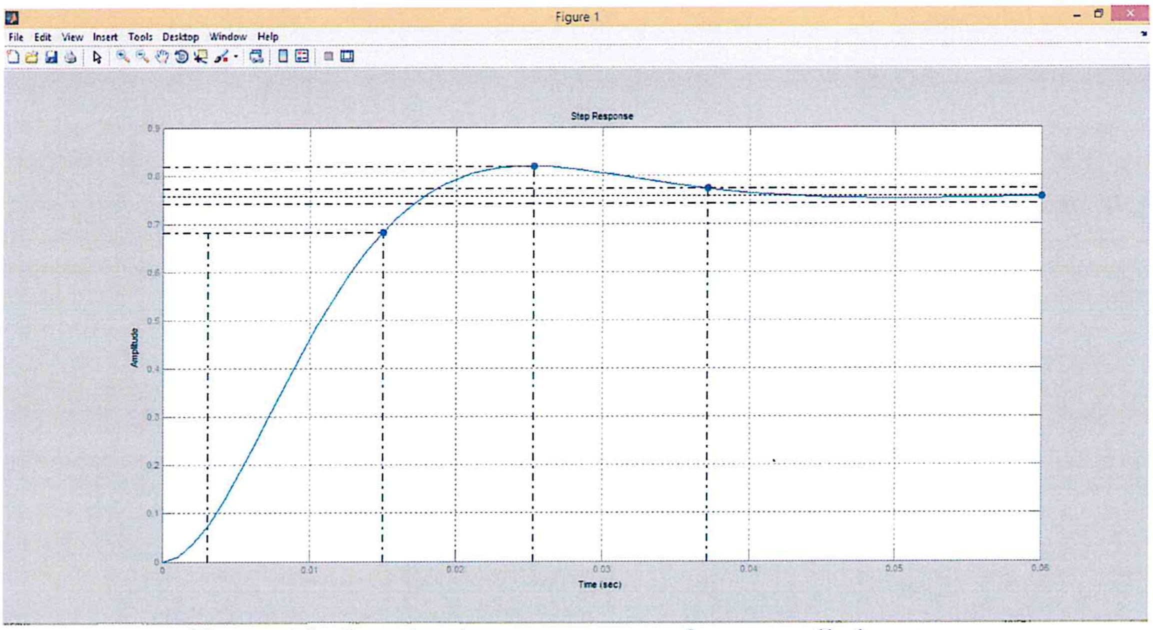

Figure 4.2: Speed unit step response of uncontrolled system

Based on Figure (4.2), the final value is 0.758, the rise time (tr) is 0.0119 second, the maximum overshoot (Mp) is 8.12 % for at 0.0253second, peak amplitude is 0.819 and the settling time (ts) is 0.0372 second.

4.3 System Simulation Results with PID Controller

The system simulated by adding the PID controller in Figure (3.5) and the simulation result for position can be seen in Figure (4.3).

For the simulation in Figure (4.3), the final value is 1, the rise time (tr) is 0.167, the maximum overshoot (Mp) is 2.85 % for at 0.734second, peak amplitude is 1.03 and the settling time (ts) is 1.42.

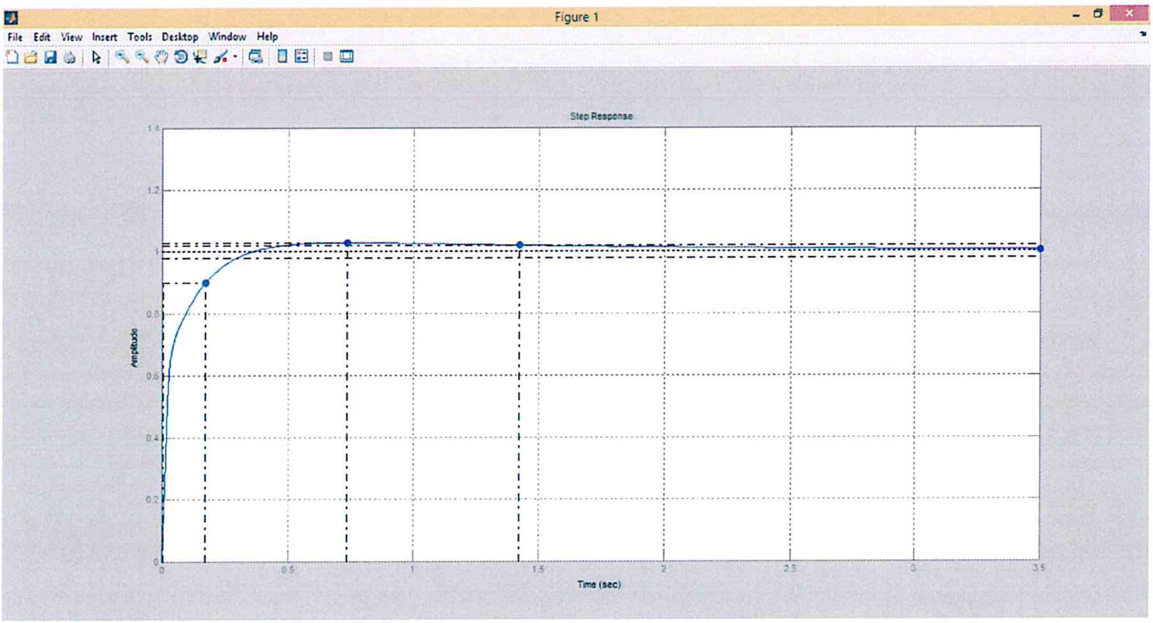

Figure 4.3: Position unit step response with PID controller

By using the same simulink model in figure (3.5) the simulation result for speed can be seen in figure (4.4).

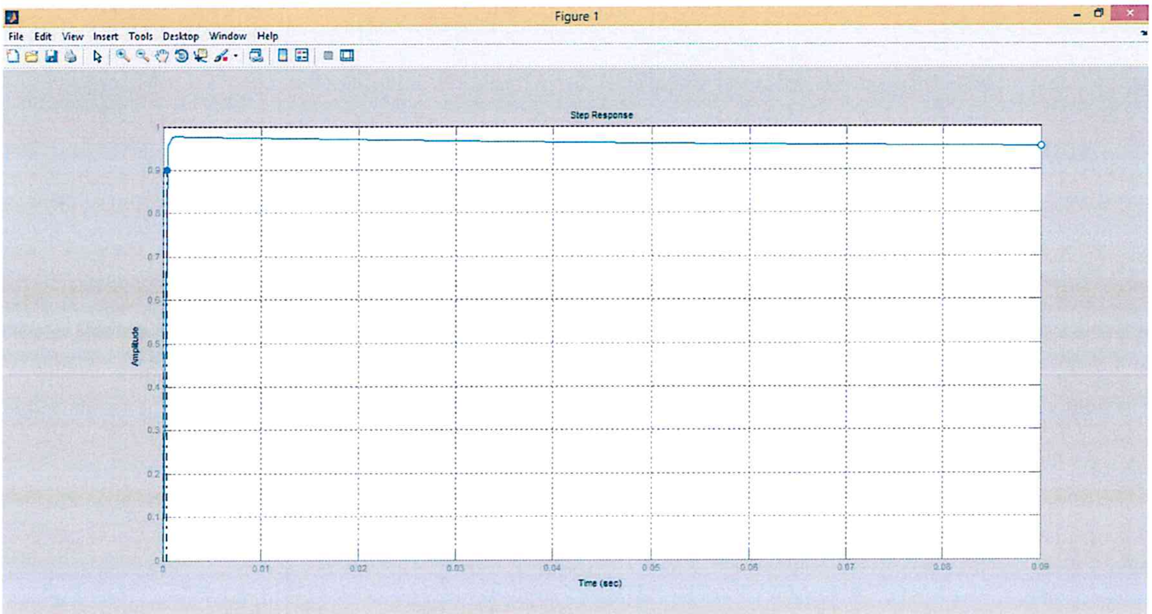

Figure 4.4: Speed unit step response with PID controller

Based on figure (4.4), the final value is 0.95, 0 % for overshoot (Mp) at 0.09second, the steady state error 1, 0.953second for the time (Tp), non-rise time and non-settling time.

The control signal for PID controller is shown in Figure (4.5):

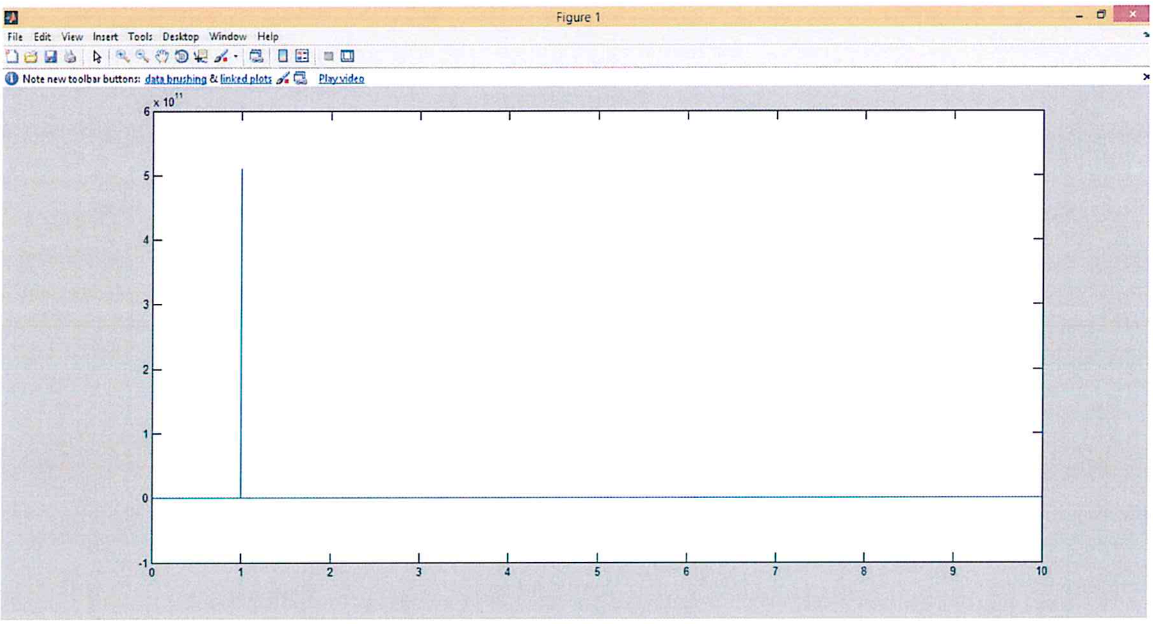

Figure 4.5: Control signal of PID controller

4.4 System Simulation Results with Fuzzy Controller

By adding the FLC in Figure (3.7) the simulation result for position is shown in Figure (4.6).

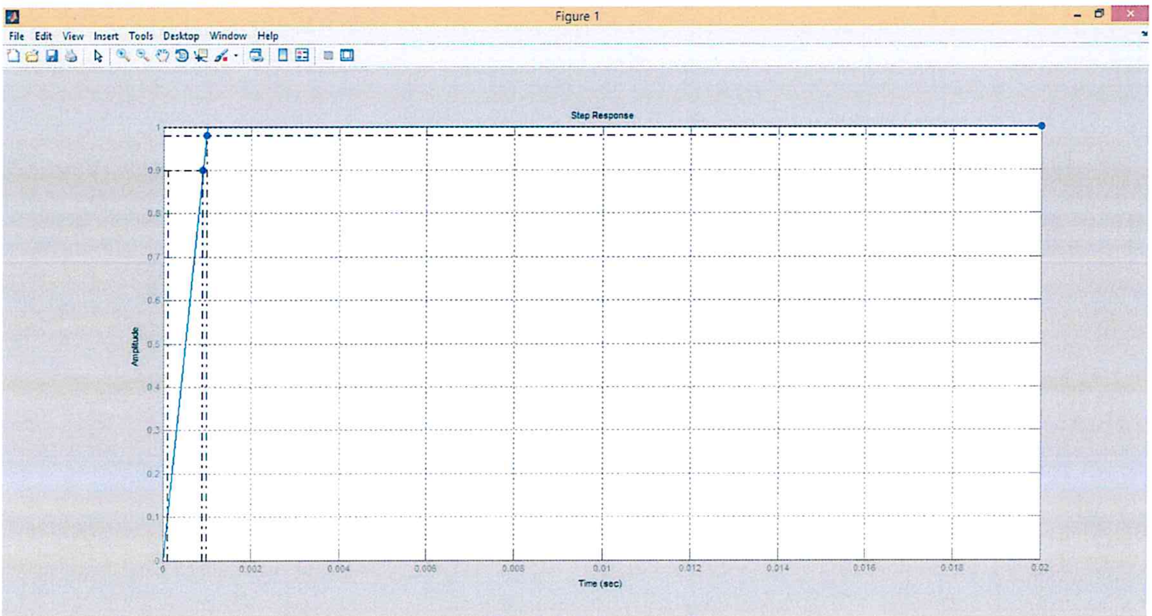

Figure 4.6: Position unit step response with Fuzzy controller

Based on the Simulink model for simulating the FLC Figure (4.6) shows the result of simulation for position. The final value is 1 for the system, 0.631 second for the rise time (tr), zero overshoot (Mp), 1.15second for settling time (ts) and zero steady state error.

By using the same simulink model in Figure (3.7) the simulation result for speed can be seen in Figure (4.7).

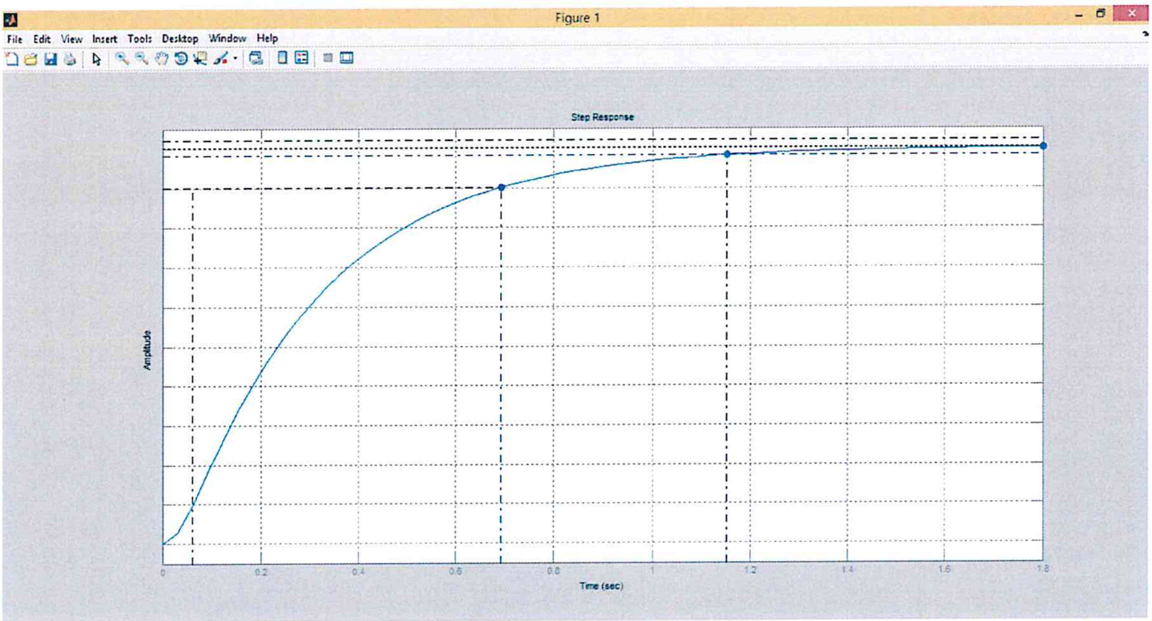

Figure 4.7: Speed unit step response with Fuzzy controller

The Figure (4.7) shows the result of simulation for speed. The final value is 1. For the system, also 0.000798 second for the rise time (tr), no overshoot (Mp) at less than 0.02 second, and 0.000978settling time (ts) and zero steady state error.

The Figure (4.8) shows the control signal for FLC controller:

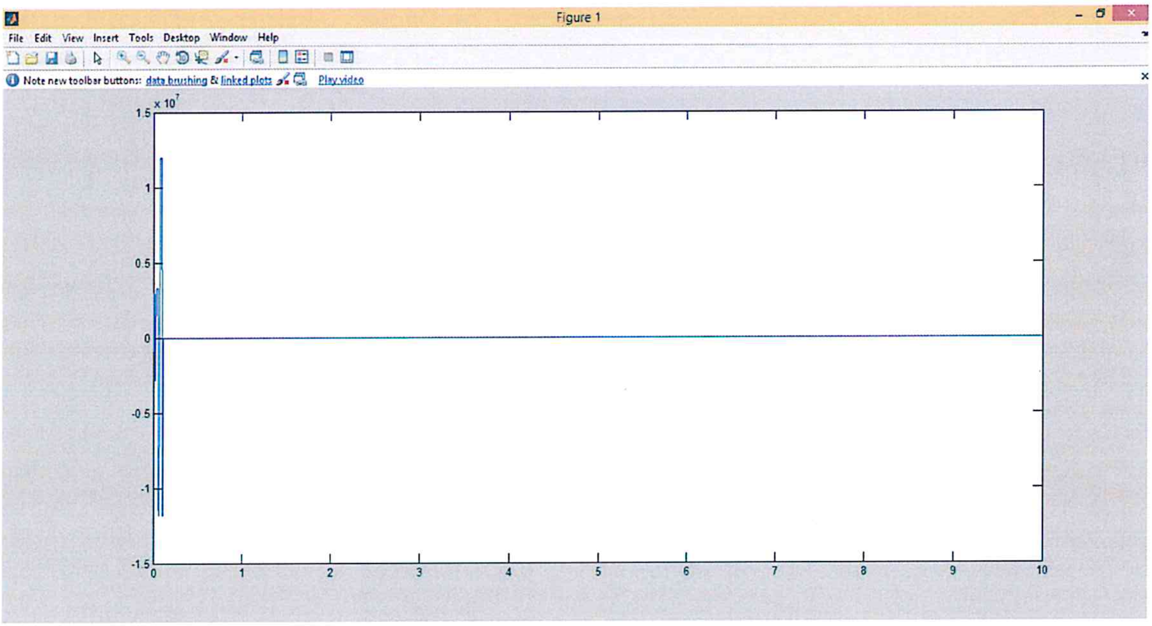

Figure 4.8: Control signal of FLC controller

4.5 Comparisons and Discussions

After implementation the DC servomotor model in MATLAB software, and design of both PID controller and FLC, the system response specifications with and without controllers for position and speed is obtained as in Table (4.1) and (4.2) respectively. Performance is evaluated on the basis of rise time, settling time and maximum overshoot.

Design of PID controller has been successfully simulated but it still has an overshoot of 2.8%in position response. Refer to the results of the fuzzy controller the system has better stability, zero overshoot and less settling time. Even though, the PID controller produces the response with lower rise time compared with the fuzzy logic controller, but it offers steady state error due to the oscillatory behavior in transient period. For the speed response the FLC reaches zero over shoot faster than PID, with better stability. The control signal for FLC in Figure (4.5) shows less amplitude than the control signal of PID in Figure (4.8) that illustrates the less consumption and cost of FLC. The results and observations show that the FLC achieves a better performance than PID controller.

Table 4.1: System position response specifications with and without controllers

Controller	ts (Second)	tr (Second)	Overshoot (MP) (%)
Without	Non	0.00286	43.1 at 0.00766
PID	1.42	0.167	2.85 at 0.734
FLC	1.15	0.631	0

Table 4.2: System Speed response specifications with and without controllers

Controller	ts (Second)	tr (Second)	Overshoot (MP) (%)
Without	0.0372	0.0119	8.12 at0.0253
PID	Non	Non	0 at 0.09
FLC	0.000978	0.000798	0 at 0.02

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 fuzzy logic controller are used for DC servomotor position and speed control. This study was based on mathematical models of DC servomotor and MATLAB SIMULINK software. Design with PID controller in MATLAB SIMULINK shows good response. The PID controller produces the response with lower rise time compared with the fuzzy logic controller, but it offers steady state error due to the oscillatory behavior in transient period. Design with fuzzy logic controller shows better results. Refer to the results of the fuzzy controller the system has better stability, zero overshoot and less settling time.

5.2 Recommendations

In this thesis Comparison between proportional integral derivative controller and fuzzy logic controller for DC servomotor is introduced and basic work has been done for the case study, so the following are some recommendations and Guides for further research:

- Using genetic algorithms.
- Using neural controller.
- Using neuro fuzzy.
- Implementation of the research.

REFERENCES

References

[1] P.C.Sen. (Parch Chandra), "Principle of Electric Machines and Power Electronics", second edition, John Wiley & Sons, 1997

[2] Timothy J.Ross, Fuzzy, "Logic with Engineering Applications'', second edition, John Wiley & Sons, 2004

[3] Kwang H. Lee, "First Course on Fuzzy Theory and Applications'', Springer-Verlag Berlin Heidelberg 2005

[4] Kevin M. Passino& Stephen Yurkovich, "Fuzzy Control'', Addison Wesley Longman. Inc, 1998

[5] George J. Klir/Bo YklSIt, "Fuzzy Sets and Fuzzy Logic Theory and Applications'', Prentice-Hall Inc.1995

[6] Didier Dubois and Henri Prade, "Fundamentals of Fuzzy Sets '', Kluwer academic publishers, 2000

[7] Asia/Europe: Nippon Pulse Motor Co., Ltd. "Basic of servomotor control'', North/South America: Nippon Pulse America, Inc.

[8] Katsuhiko Ogata, "Modern Control Engineering '', fourth edition, Prentiffi Hall, New Jmey, 2002

[9] KarlJ. Astrom and Tore Hagglund, "PID Controllers: Theory, Design and Tuning'', 2nd Edition, Instrument Society of America, 1995

[10] Michael A. Johnson Mohammad H. Moradi, "PID Control: New Identification and Design Methods'', Springer-Verlag London Limited, 2005

[11] The math work, "Simulink® Simulation and Model-Based Design'', The Math Works, Inc. 3 Apple Hill Drive, 2004

[12] Lennart Ljung, "System Identification: Theory for The User", second edition Prentice Hall, 1999

[13]http://mechatronics.mech.northwestern.edu/design_ref/actuators/servo_motor_intro.html

[14] http://www.ehow.com/about_4706437_principles-behind-servo-motors.html

[15] http://www.ehow.com/about_4706437_principles-behind-servo-motors.html#ixzz1vkkwUDVB
[16]www.electerical 4u.com/voltage –or-electric-potential –difference/#top (15march2013)
[17] IEEE transaction on control systems technology, vol. 13, NO. 4, July 2005 "PID Control System Analysis, Design, and Technology''
[18] C. Urrea and J Kern, "A New Model for Analog Servo Motors", Canadian Journal on Automation, Control and Intelligent Systems, Vol. 2, No. 2, March 2011.
[19] Munadi & M. Amirullah Akbar, " Simulation of Fuzzy Logic Control for DC Servo Motor using Arduino based on Matlab/Simulink'', International Conference on Intelligent Autonomous Agents, Networks and Systems Bandung, Indonesia, August 19-21, 2014
[20] Prof. Krishna vasudevan "Paper on application of dc motors ''indian institute of technology madras
[21] Math works, "Fuzzy logic user's guide'', Mathworks, Inc, 1999