

Sudan University of Science and Technology

College of Graduate Studies

Electrical Engineering

**The position control of servo motor by using PID
controller and fuzzy logic controller**

التحكم الموضعي لمحرك الخدمة باستخدام المتحكم التناسبي
التفاضلي التكاملي والمتحكم الغامض المنطقي

A thesis Submitted in Partial Fulfillment for the Requirements

For the Degree of M.Sc. In Electrical Engineering (Power)

Prepared by:

ELhabeeb Babiker ELhabeeb Babiker

Supervised by:

Dr: Mansour Babiker Idris Hussien

May 2014

الاية

خَالِدُونَ ﴿٢٥٧﴾ أَلَمْ تَرَ إِلَى الَّذِي حَاجَّ إِبْرَاهِيمَ فِي رَبِّهِ
أَنَآءَ اتَّهَمَ اللَّهُ الْمَلَكَ إِذْ قَالَ إِبْرَاهِيمُ رَبِّيَ الَّذِي يُحْيِي
وَيُمِيتُ قَالَ أَنَا أُحْيِي وَأُمِيتُ قَالَ إِبْرَاهِيمُ فَإِنَّ اللَّهَ يَأْتِي
بِالشَّمْسِ مِنَ الْمَشْرِقِ فَأْتِ بِهَا مِنَ الْمَغْرِبِ فَبُهِتَ الَّذِي
كَفَرَ وَاللَّهُ لَا يَهْدِي الْقَوْمَ الظَّالِمِينَ ﴿٢٥٨﴾ أَوْ كَالَّذِي مَرَّ

(صدق الله العظيم)

سورة البقرة الاية (257)

Dedication

This thesis is dedicated to my parents and family

ACKNOWLEDGEMENT

I would like to express my sincere gratitude to my supervisor Dr: Mansour Babiker Idris Hussien for his continued and encouragement throughout the course of this research .

My family and friends for all their support Finally, I Thank

ABSTRACT

The position control system one of the interesting term in control system Engineering. Nowadays, several control system algorithm have been applied in that application. PID controller is a well-known controller and widely used in feedback control in industrial processes. In position control system, PID controller sometimes cannot accurate for this application because of nonlinear properties. Therefore in this research, the fuzzy logic controller is proposed to overcome the problem of PID controller. Fuzzy logic controller has ability to control the nonlinear system because of the algorithm is concentrate by simulating the expert and implemented in language. Based on the experimental result, the fuzzy logic controller designed able to improve the performance of the position control system compared to the PID controller in term of reduced settling time, overshoot and oscillation can be reduced.

مستخلص البحث

التحكم في الوضع للانظمة الهندسية باستخدام المحركات الكهربائية من الشائع استخدامها وبصورة كبيرة وتوجد طرق مختلفة تطبق للتحكم ويستخدم متحكم الPID بصورة عريضة في المنظومات الصناعية عن طريق التغذية العكسية ومع كثرة استخدامه الا انه قليل الدقة بسبب عدم الخطية في خصائصه في هذا المشروع وللحصول علي دقة عالية في التحكم استخدم متحكم المنطق الغامض لموثوقيته وسهولة خوارزميته ولغته في الحصول علي النتائج وتم تصميم لتحسين اداء متحكم الPID من حيث تقليل تقليل زمن الاستقرار وكذلك تقليل التذبذب واعلي قيمة للاشارة .

Table of Contents

Description	Page No
Quran Verse	I
Dedication	II
Acknowledgement	III
Abstract	IV
Table of contents	VI
List of Figures	VIII
List of Tables	X
List of Abbreviations	XI
Chapter One: Introduction	1
1.1 introduction	1
1.2 Problem Statement	2
1.3 Objective	2
1.4 project Scope	2
1.5 Thesis outline	3
Chapter two: Servo motor	4
2.1 Introduction	4
2.1.1 The Encoder	5
2.1.2 The Motors	5
2.1.3 The control system	6
2.2 Mechanism	6
2.3 Servo Motor Modeling	7
2.4 The control system of the DC motor	9

CHAPTER THREE PID AND FUZZY LOGIC CONTROLLERS	11
3.1 PID Controller	11
3.1.1 Overview of methods	13
3.1.2 Manual tuning	13
3.1.3 PID tuning software	14
3.2 Fuzzy Logic Controller	14
3.2.1 Structure of Fuzzy Logic	20
3.2.1.1 Fuzzification	21
3.2.1.3 Defuzzification	32
3.2.1.3.1 Center of gravity(COG)	33
3.2.1.3.2 Center of gravity method for singleton (COGS)	33
3.1.2.3.3 Mean of maxima (MOM)	34
CHAPTER FOUR: SIMULATION RESULTS AND DISCUSSION	35
4.1 Case study	35
4.2 Simulation Results of PID	35
4.3 Simulation Results of the fuzzy logic controller	37
4.4 Discussion of the results	38
Chapter five: Conclusion and Recommendations	40
5.1 Conclusion	40
5.2 Recommendations	41
Appendix	42
References	45

List of Figures

Figure NO	Description	Page NO
2.1	Servomotor	4
2.2	construction of a standard servo motor	7
2.3	Schematic diagram Block diagram	8
2.4	control system of DC motor	9
3.1	Block diagram of the closed loop servo motor with PID controller	12
3.2	direct control of fuzzy logic control	16
3.3	feed forward control	18
3.4	Fuzzy parameter adaptive control	18
3.5	Fuzzy parameter adaptive control	20
3.6	Linear membership functions	23
3.7	Linear-trapezoidal membership function	24
3.8	Gaussian membership functions	25
3.9	Gaussian-trapezoidal membership functions	25
3.10	Membership functions comprised of fuzzy singletons	26
3.11	sigmoid membership function	26
3.12	Fuzzy logic statement evaluation.	27
3.13	Binary logic statement evaluation	28
3.14	Fuzzy logic statement evaluation	28
3.15	logical operation used in fuzzy logic	30
3.16	one input one output rule base with non-singleton output sets	34

4.1	Simulink diagram of PID controller for position control of servo motor.	36
4.2	unit step response of PID controller for position control of servo motor	36
4.4	Simulink diagram of Fuzzy for position control of servo motor.	37
4.5	unit step response of FLC for position control of servo motor	37
4.5	unit step system response for the PID and fuzzy logic controllers	38

List of tables

Table NO	Description	Page NO
3.1	Standard logical operations truth table	29
3.2	logical operations truth table	29
4.1	Overshoots of the PID and fuzzy logic controllers	39
4.2	Beck Time of the PID and fuzzy logic controllers.	39
4.3	Settling Time of the PID and fuzzy logic controllers	39

List of Abbreviation

Symbols	Descriptions
PMSM	Permanent magnet synchronous motor
FLC	Fuzzy logic controller
MMF	Magnetic motive force
COG	Center of gravity
MOM	Mean of maxima
EMF	Electro motive force
PID	Proportional and integral and derivative
GTO	Gate turn off
IGBT	Insulated gate bipolar transistor
MOSFET	Metal oxide semiconductor field effect transistor

CHAPTER ONE

INTRODUCTION

1.1 Introduction

Several control system algorithms have been applied in control system engineering. The one of the interesting term in that application is position control. Position control for servo motor exists in a great variety of automatic processes. However, the performance of servo motor is influenced by uncertainties such as nonlinear properties, mechanical parameter variation, external disturbance, unstructured uncertainty due to nonideal field orientation in the transient state and unmodelled dynamics. From a practical point of view, complete information about uncertainties is difficult to acquire in advance. To deal with these uncertainties, much research has been carried out in recent years to apply various approaches in the position control. {1}

When it occurs, the fuzzy control method begins to be used instead of the PID control method, and the basic continuous feedback control is PID controller. The PID controller possesses good performance but is not adaptive enough. This is appealing when the load is changed, where the original controller generally cannot maintain the design performance and thus should be re-designed for the new system conditions. {2}

As a very good method, a fuzzy controller method can adopt expert knowledge into regulations of control system and the regulations can be used to determine the output value by logical inference. So, it is also does not need precision system model and has high robust ability. In recent years, fuzzy controllers have been widely developed and a variety of methods have been proposed to improve the performance of fuzzy controller. {3}

1.2 Problem Statement

There are wide of control techniques that can be applied to meet the control objectives of the system that depend on the factors of which the proposed design objective might on. These factors are behaviors in terms of nonlinearity system, time response and engineering goals such as cost and reliability. The factors that motivate many researchers to use conventional control theory and techniques are system nonlinearity that may contain unknown parameters. That unknown parameters may not be estimated accurately if reliable experimental data is absent and also the delays present in the process of system might complicate achieving high performance control. Fuzzy logic controller has been use in control system when the mathematical model of the interested process is vague of exhibits uncertainties. One of The advantages of using the fuzzy logic control that the developed controller can deal with increasingly complex system. It can also be implemented without precise knowledge of the model structure of dynamic system.

1.3 Objectives

The main objectives of the project are:

1. To design the PID and fuzzy logic controller for position control to control the servo motor system by using MATLAB simulation
2. To investigate the application of PID and fuzzy logic controller in servo motor control.
3. To test the performances of PID and fuzzy logic controller by experiment

1.4 Project Scope

The scope of this project is:

1. Understanding of the servo motor control background, analyzing the problem and investigated fuzzy logic controller theory which has been applied to the servo motor system.

2. Design and implementation of the control algorithm that is carried by PID and fuzzy logic controller.
3. Implementation PID and fuzzy logic controller on the servo motor control by using MATLAB simulation.

1.5 Thesis outline

This thesis is presented in five chapters. This introductory chapter, introduction, gives introductory notes to the thesis and chapters contains. Chapter Two, servo motor, provides background on the servo motor system and modeling. Chapter three provides background on the PID and fuzzy logic controllers. Chapter four deals with the study case, simulation and results. Chapter five deals with the conclusion and recommendations.

CHAPTER TWO

SERVO MOTOR

2.1 Introduction

Servo motor is an automatic device that uses error sensing feedback to correct the performance of a mechanism. The term correctly applies only to the systems where the feedback or error correction signals help to control mechanical position or other parameters. A common type of servo provides position control. Servos are commonly electrical or partially electronic in nature, using an electric motor as the primary means of creating mechanical force. Other types of servos use hydraulics, pneumatics, or magnetic principles. Usually, servos operate on the principle of negative feedback, where the control input is compared to the actual position of the mechanical system as measured by some sort of transducer at the output. Any difference between the actual and wanted values (error signal) is amplified and used to drive the system in the direction necessary to reduce or eliminate the error. Servomotors are available as AC or DC motors. Today, servo motor are used in automatic machine tools, satellite tracking antennas, remote control airplanes, automatic navigation systems on boats and planes, and anti-aircraft gun control systems. Figure (2.1) shows the construction of a standard servo motor. {4}



Figure (2.1): construction of a standard servo motor

2.1.1 The Encoder

The first servo motors were developed with synchronous as their encoders. Much work was done with these systems in the development of radar and anti-aircraft artillery during World War II.

Simple servo motors may use resistive potentiometers as position encoder. These are only used at the very simplest and cheapest level, and are in close competition with stepper motor. They suffer from wear and electrical noise in the potentiometer track. Although it would be possible to electrically differentiate their position signal to obtain a speed signal, PID controllers that can make use of such a speed generally warrant a more precise encoder. Modern servo motor use optical encoders, either absolute or incremental. Absolute encoder can determine their position at power-on, but are more complicated and expensive. Incremental system, like stepper motor, often combines their inherent ability to measure intervals of rotation with a simple zero-position sensor to set their position at start-up. Many servomotors are rotary, but are used for ultimate control of a linear motion. In some of these cases, a linear encoder is used. These servomotors avoid inaccuracies in the drive train between the motor and linear carriage, but their design is made more complicated as they are no longer a pre-packaged factory-made system. {4}

2.1.2 The Motors

The type of motor is not critical to servo motor and different types may be used. At the simplest, brushed magnet DC motors are used, owing to their simplicity and low cost. Small industrial servo motors are typically electronically-commutated brushless motors. For large industrial servo motors, AC induction motors are typically used, often with variable frequency drive to allow control of their speed. For ultimate performance in a compact package, brushless AC motor

with permanent magnet field is used effectively large versions of brushless DC electric motor. Drive modules for servomotors are a standard industrial component. Their design is a branch of power electronics, usually based on a three-phase MOSFET H bridge. These standard modules accept a single direction and pulse count (rotation distance) as input. They may also include over-temperature monitoring, over-torque and stall detection features. As the encoder type, gear head ratio and overall system dynamics are application specific, it is more difficult to produce the overall controller as an off-the-shelf module and so these are often implemented as part of the main controller. {4}

2.1.3 The Control system

Most modern servo motors are designed and supplied around a dedicated controller module from the same manufacturer. Controllers may also be developed around microcontrollers, but this is rarely worth the time and trouble, compared to buying off-the-shelf. {4}

2.2 Servo motor mechanism

As the name suggests, a servo motor is a servomechanism. More specifically, it is a closed –loop servomechanism that uses position feedback to control its motion and final position. The input to its control is some signal, either analogue or digital, representing the position commanded for output shaft. The motor is paired with some type of encoder to provide position and speed feedback. In the simplest case, only the position is measured. The measured position of output is compared to the command position, the external input to the controller. If the output position that required, an error signal is generated which then causes the motor to rotate in either direction, as needed to bring the output shaft to the appropriate position. As the position approach, the error signal reduces to zero and

the motor stops. The very simplest servo motor use position-only sensing via a potentiometer and bang-bang control of their motor, the motor always rotates at full speed (or stopped).this motor not widely used in industrial motion control, but they form the basis of simple and cheap servos for radio-controlled models. More sophisticated servo motors measure both the position and also the speed of the output shaft. They may also control the speed of their motor, rather than always running at full speed. Both of these enhancements, usually in combination with a PID control algorithm, allow the servo motor to be brought to its commanded position more quickly and more precisely, with less overshooting. {4}

2.3 Servo Motor Modeling

Servo motor is used for position or speed control in closed loop control systems. It has implemented proportional integral, fuzzy logic inference system respectively at the variable working situations to the simulation model which has prepared at the Matlab programmers for improvement the servomotor performance. The equivalent circuit diagram of servo motor is presented in Figure (2.2) The armature is modeled as a circuit with resistance, R_a connected in series within inductance, L_a and a voltage source, $V_b(t)$ representing the back emf in the armature when the rotor rotates.

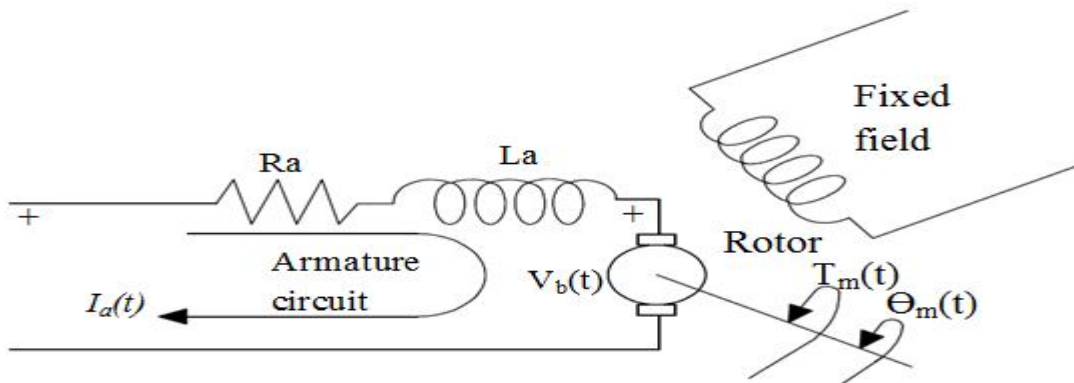


Figure (2.2): Servomotor system

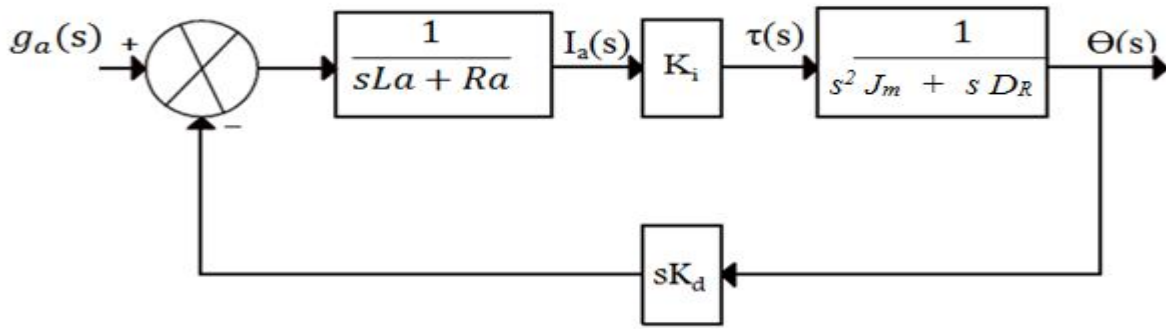


Figure (2.3): Schematic diagram Block diagram

Kirchhoff's Voltage Law is used to map the armature circuitry dynamic of the motor. Thus, assuming that the inductance L_a can be ignored, The supply voltage ($E_a(t)$) is product of motor input power by the armature current ($I_a(t)$).

$$E_a(t) = I_a(t)R_a + V_b(t) \dots \dots \dots (2.1)$$

Since the armature current is rotating in a magnetic field, its back electromotive force is proportional with speed. when $V_b(t)$ is the velocity of the conductor normal to the magnetic field. {4}

$$V_b(t) = K_B s \theta_m(t) \dots \dots \dots (2.2)$$

The typical equivalent mechanical loading on a motor that connected to the motor shaft including total moment of inertia, J_m and total viscous friction, B . Assume that $T(t)$ is the torque developed by the motor. which is given by

$$T(t) = J_m s^2 \theta_m + B s \theta_m(t) \dots \dots \dots (2.3)$$

The developed motor output torque for this servo motor can be given by,

$$T(t) = K_T I_a(t) \dots \dots \dots (2.4)$$

using Laplace transformation on the equation above, become,

$$E_a(t) = R_a I_a(t) + V_b(t) \dots \dots \dots (2.5)$$

The above equation can be written as

$$I_a(t) = \frac{E_a(t) - V_b}{R_a} \dots \dots \dots (2.6)$$

From equation (2.3) and equation (2.4), it is found that the motor output torque is given by equation (2.7).

$$K_T I_a(t) = J_m s^2 \theta_m + B s \theta_m(t) \dots \dots \dots (2.7)$$

Substituting equation (2.6) into equation (2.7) it is as found

$$K_T \left(\frac{E_a(t) - V_b}{R_a} \right) = J_m s^2 \theta_m + B s \theta_m(t) \dots \dots \dots (2.8)$$

Substituting equation (2.6) into equation (2.7) it is as found

$$K_T \left(\frac{E_a(t) - K_B s \theta_m(t)}{R_a} \right) = J_m s^2 \theta_m + B s \theta_m(t) \dots \dots \dots (2.9)$$

By simplifying equation (2.9), the final transfer function can be obtained as equation (2.10)

$$\frac{\theta_m(t)}{E_a(t)} = \frac{K_T}{J_m R_a s^2 + s(B R_a + K_T K_B)} \dots \dots \dots (2.10)$$

2.4 The control system of the DC motor

The control system of the dc motor represented the following circuit (Figure (2.4))

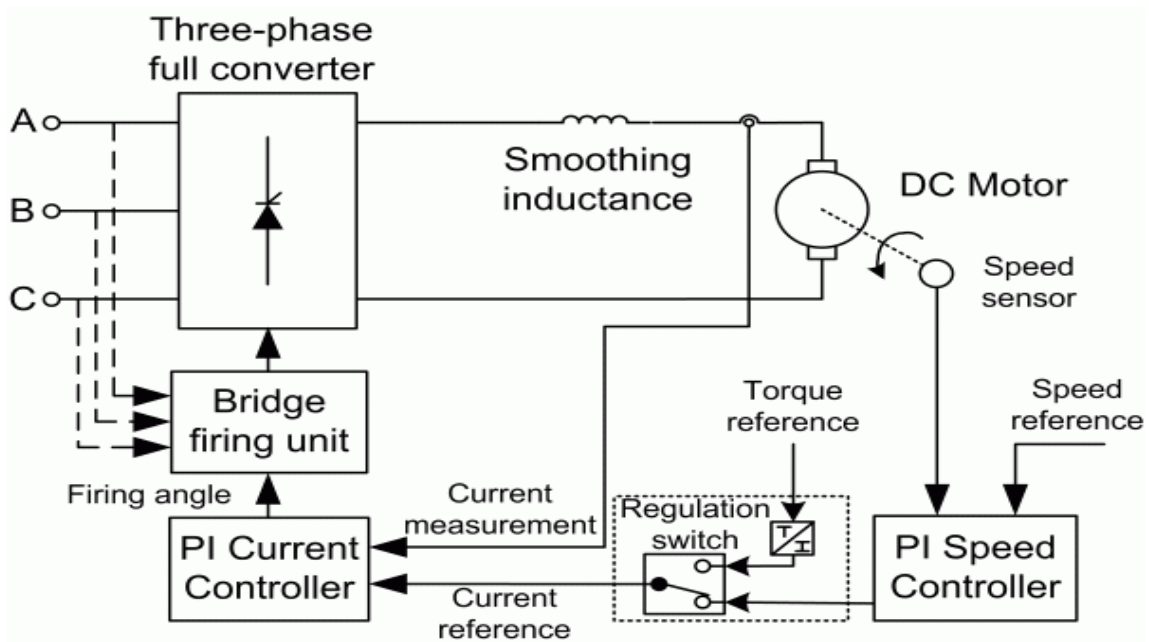


Figure (2.4): control system of DC motor

The components of this circuit are:

- Three- phase full converter:

The main function of this element is to convert AC voltage to dc voltage. The dc voltage must be passing through smoothing inductance for harmonics elimination from the rectified dc voltage. The rectifier consists of thyristor bridge which needs continues firing process.

- Firing circuit

It is an electronic circuit consists of transistors to give firing signal to the gate of thyristors.

- Speed sensor:

The main function of this sensor is the measuring of the speed that comes from motor rotation instantaneously.

- PI speed

The main function is comparing reference speed with motor speed, and giving a control signal depending on motor speed.

- PI current

The main function is comparing the current comes from the regulation switch and the motor input current, and giving a control signal depending on motot input current.

CHAPTER THREE

PID AND FUZZY LOGIC CONTROLLERS

3.1 PID Controller

A proportional–integral–derivative controller (PID controller) is a generic control of feedback mechanism (controller) widely used in industrial control systems . a PID is the most commonly 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. 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, denoted P, I, and D. Simply , 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, a damper, or the power supplied to a heating element. In the absence of knowledge of the underlying process, a PID controller has historically been considered to be 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. A PID controller is one of the most commonly used controllers because it is simple and robust. This controller is extremely popular because it can

usually provide good closed loop response characteristics, can be tuned using relatively simple rules and easy to construct using either analogue or digital components. Figure (3.1) below illustrates the block diagram of PID controller. {5}

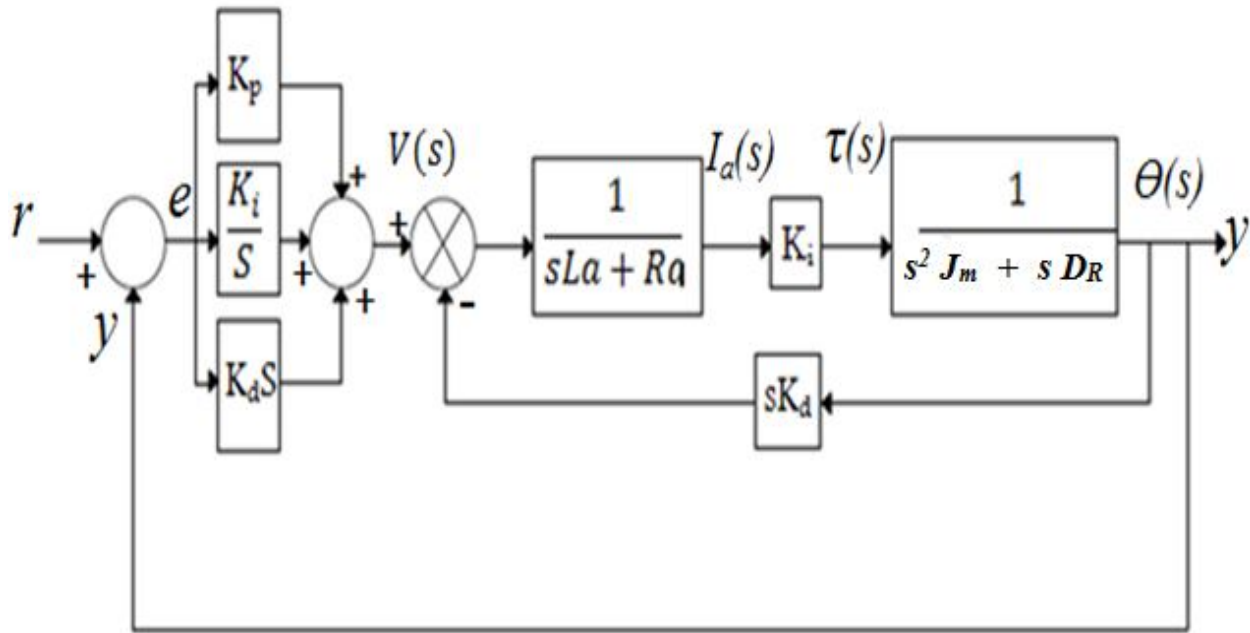


Figure (3.1): Block diagram of the closed loop servo motor with PID controller

The PID controller can be defined as in equation (3.1) by the relationship between controller input $e(t)$ and the controller output $V(t)$ that is applied to the motor armature

$$V(t) = k_p e(t) + k_i \int_0^T e(t) dt + (k_p/dt) de(t)$$

Using Laplace transformation, the transfer functions of PID controller is as follows

$$V(s)/E(s) = k_p + (k_i/s) + k_d s$$

Assuming that $K_p = K$, $K_i = \frac{K}{t_i}$ and $K_d = K t_d$ the transfer function of the PID controller can be as follows

$$K(s) = (k t_d / s) \{ s^2 + (1/T_d) s + (k/t_d t_i) \}$$

3.1.1 Overview of methods

There are several methods for tuning a PID loop. The most effective methods generally involve the development of some form of process model, then choosing P, I, and D based on the dynamic model parameters. Manual tuning methods can be relatively inefficient, particularly if the loops have response times on the order of minutes or longer. The choice of method will depend largely on whether or not the loop can be taken "offline" for tuning, and on the response time of the system. If the system can be taken offline, the best tuning method often involves subjecting the system to a step change in input, measuring the output as a function of time, and using this response to determine the control parameter. {5}

3.1.2 Manual tuning

If the system must remain online, one tuning method is to first set K_i and K_d values to zero. Increase the K_p until the output of the loop oscillates, then the K_p should be set to approximately half of that value for a "quarter amplitude decay" type response. Then increase K_i until any offset is corrected in sufficient time for the process. However, too much K_i will cause instability. Finally, increase K_d , if required, until the loop is acceptably quick to reach its reference after a load disturbance. However, too much K_d will cause excessive response and overshoot. A fast PID loop tuning usually overshoots slightly to reach the set point more quickly; however, some systems cannot accept overshoot, in which case an over-damped closed-loop system is required, which will require a K_p setting significantly less than half that of the K_p setting that was causing oscillation. {5}

3.1.3 PID tuning software

Most modern industrial facilities no longer tune loops using the manual calculation methods shown above. Instead, 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 changes. Mathematical PID loop tuning induces an impulse in the system, and then uses the controlled system's frequency response to design the PID loop values. In loops with response times of several minutes, mathematical loop tuning is recommended, because trial and error can take days just to find a stable set of loop values. . Some digital loop controllers offer a self-tuning feature in which very small set point changes are sent to the process, allowing the controller itself to calculate optimal tuning values. Other formulas are available to tune the loop according to different performance criteria. Many patented formulas are now embedded within PID tuning software and hardware modules. {4}

Advances in automated PID Loop Tuning software also deliver algorithms for tuning PID Loops in a dynamic or Non-Steady State (NSS) scenario. The software will model the dynamics of a process, through a disturbance, and calculate PID control parameters in response. This method gives automatic oscillation of the process to compute the proportional, integral and derivative gains. The PID controller has been simulated based on simulink. {5}

3.2 Fuzzy Logic Controller

Fuzzy Logic Controller (FLC) is based on fuzzy logic controller and constitutes a way of converting linguistic control strategy into an automatic by generating a rule base which controls the behavior of the system. Fuzzy control is control method based on fuzzy logic. Fuzzy logic provides a remarkably simple

way to draw definite conclusions from vague ambiguous or imprecise information. It suitable for applications such as the speed control of dc motor which is has non linearity's. {4}

FLC have some advantages compared to other classical controller such as simplicity of control, low cost and the possibility to design without knowing the exact mathematical model of the process. Fuzzy logic incorporates an alternative way of thinking which allows modeling complex systems using higher level of abstraction originating from the knowledge and experience. Fuzzy logic can be described simply as “computing words rather than numbers” or “control with sentence rather than equations.”The applications of fuzzy logic are usually for household appliance such as washing machine and rice cooker. Fuzzy also been used in industrial process such as cement kilns, underground trains and robots.

While it is relatively easy to design a PID controller, the inclusion of fuzzy rules creates many extra design problems, and although many introductory textbooks explain fuzzy control, there are few general guidelines for setting the parameters of a simple fuzzy controller. The approach here is based on a three step design procedure that builds on PID control:

1. Start with a PID controller.
2. Insert an equivalent, linear fuzzy controller.
3. Make it gradually nonlinear.

Fuzzy logic can be described simply as "computing with words rather than numbers", fuzzy control can be described simply as "control with sentences rather than equations". A fuzzy controller can include empirical rules, and that is especially useful in operator controlled plants

Take for instance a typical fuzzy controller

1. If error is Neg and change in error is Neg then output is NB
2. If error is Neg and change in error is Zero then output is NM

The collection of rules is called a rule base. The rules are in the familiar if-then format, and formally the if-side is called the antecedent and the then-side is called the consequent (more often, perhaps, the pair is called antecedent – consequent or premise – conclusion).

The input value "Neg" is a linguistic term short for the word negative. the output value "NB" stands for negative big and "NM" for negative medium .The computer is able to execute the rules and compute a control signal depending on the measured inputs error and change error. The objective here is to identify and explain the various design choices for engineers. {6}

In a rule based controller the control strategy is stored in a more or less natural language. The control strategy is isolated in a rule base opposed to an equation based description. Arul based controller is easy to understand and easy to maintain for a non-specialist end-user. {8}

An equivalent controller could be implemented using conventional techniques _ in fact, any rule based controller could be emulated in, say, Fortran - it is just that it is convenient. {7}

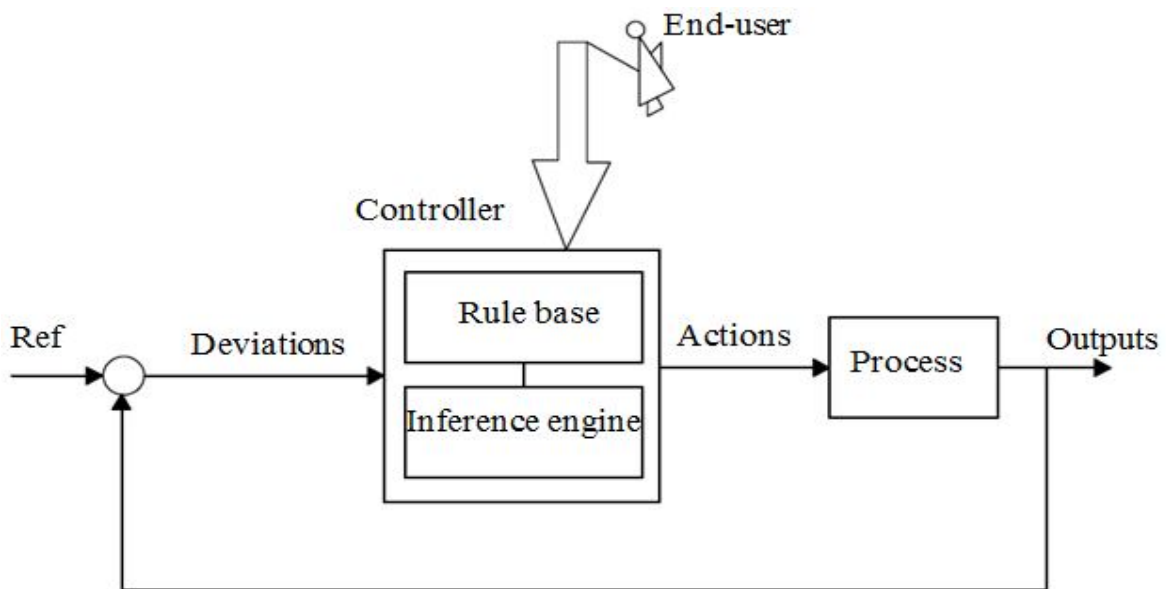


Figure (3.2) direct control of fuzzy logic control

To isolate the control strategy in a rule base for operator controlled systems.

Fuzzy controllers are being used in various control schemes (IEC, 1996). The most obvious one is direct control, where the fuzzy controller is in the forward path in a feedback control system Fig (3.2). The process output is compared with a reference, and if there is

a deviation, the controller takes action according to the control strategy. In the figure, the arrows may be understood as hyper-arrows containing several signals at a time for multi loop control. The sub-components in the figure will be explained shortly. The controller is here a fuzzy controller, and it replaces a conventional controller, say PID(proportional integral-derivative) controller.

In feed forward control Fig (3.3) a measurable disturbance is being compensated. It requires good model, but if a mathematical model is difficult or expensive to obtain, a fuzzy model may be useful. Fig (3. 2) shows a controller and the fuzzy compensator, the process and the feedback loop are omitted for clarity. The scheme, disregarding the disturbance input, can be viewed as a collaboration of linear and nonlinear control actions; the controller may be a linear PID controller, while the fuzzy controller F is a supplementary nonlinear controller

Fuzzy rules are also used to correct tuning parameters in parameter adaptive control schemes Fig (3. 4). If a nonlinear plant changes operating point, it may be possible to change the parameters of the controller according to each operating point. This is called gain scheduling since it was originally used to change process gains. A gain scheduling controller contains a linear controller whose parameters are changed as a function of the operating point in a preprogrammed way. It requires thorough knowledge of the plant, but it is often a good way to compensate for nonlinearities and parameter variations. Sensor measurements are used as scheduling variable that govern the change of the controller parameters, often by means of a table look-up. {7}

Whether a fuzzy control design will be stable is a somewhat open question. Stability concerns the system's ability to converge or stay close to equilibrium. A stable linear system will converge to the equilibrium asymptotically no matter where the system state variables start from. It is relatively straight forward to check for stability in linear systems,

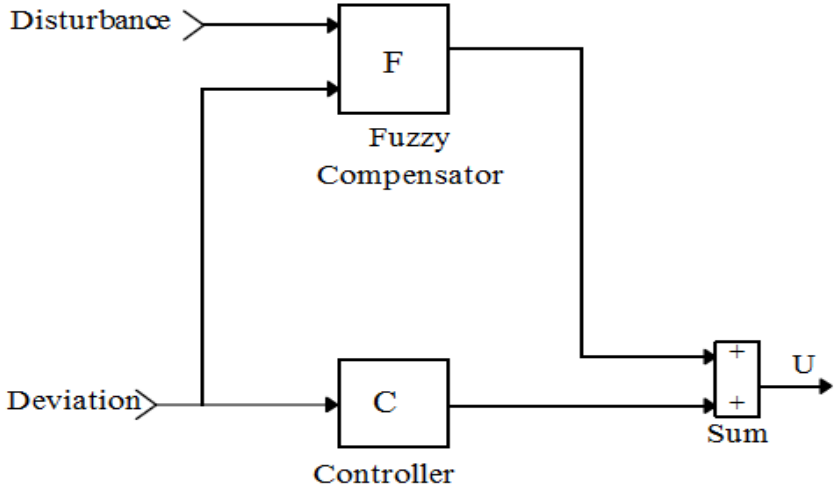


Figure (3.3) feed forward control

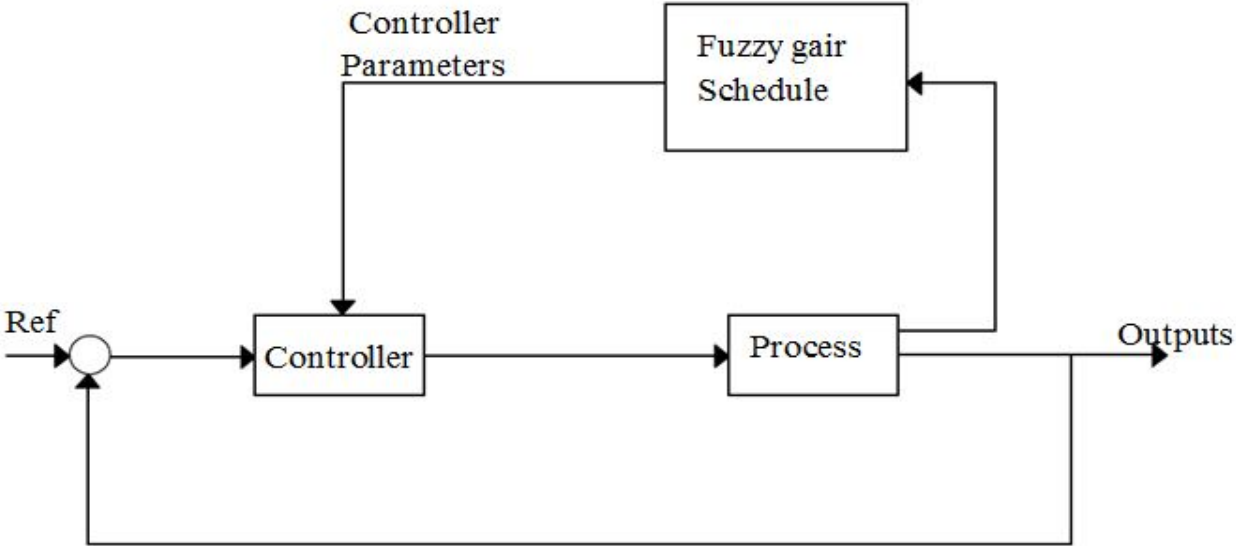


Figure (3.4) Fuzzy parameter adaptive control

for example by checking that all eigen values are in the left half of the complex plane. For nonlinear systems, and fuzzy systems are most often nonlinear, the stability concept is more complex. A nonlinear system is said to be asymptotically stable if, when it starts close to equilibrium, it will converge to it. Even if it just stays close to the equilibrium, without converging to it, it is said to be stable (in the sense of Lyapunov). To check conditions for stability is much more difficult with nonlinear systems, partly because the system behavior is also influenced by the signal amplitude apart from the frequencies. There are at least four main sources for finding control rules. {7}

- Expert experience and control engineering knowledge One classical example is the operator's handbook for a cement kiln The most common approach to establishing such a collection of rules of thumb, is to question experts or operators using a carefully organized questionnaire.
- Based on the operators control action fuzzy if - then rules can be deduced from observations of an operator's control actions or a log book. The rules express input-output relationships.
- Based on fuzzy model of the process. A linguistic rule base may be viewed as an inverse model of the controlled process. Thus the fuzzy control rules might be obtained by inverting a fuzzy model of the process. This method is restricted to relatively low order systems, but it provides an explicit solution assuming that fuzzy models of the open and closed loop systems are available. Another

approaches fuzzy identification or fuzzy model-based control (see later).

- Based on learning. The self-organizing controller is an example of a controller that finds the rules itself. Neural networks is another possibility.

There is no design procedure in fuzzy control such as root-locus design, frequency response design, pole placement design, or stability margins, because the rules are often nonlinear.

Therefore we will settle for describing the basic components and functions of fuzzy controllers, in order to recognize and understand the various options in commercial software packages for fuzzy controller design. There is much literature on fuzzy control and many commercial software tools, but there is no agreement on the terminology, which is confusing. There are efforts, however, to standardize the terminology, and the following makes use of a draft of a standard from the International Electro technical Committee. {7}

3.2.1 Structure of Fuzzy Logic

There are specific components characteristic of a fuzzy controller to support a design procedure. Figure (3.5) shows the controller between the preprocessing block and post processing block.

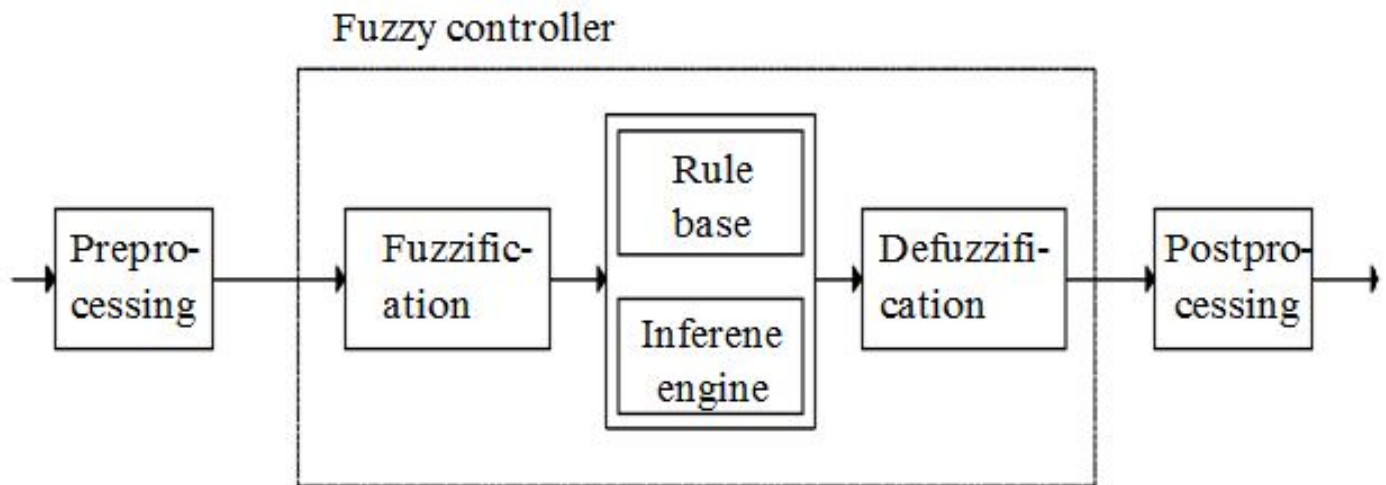


Figure (3.5): Fuzzy parameter adaptive control

Developing a Fuzzy logic control and applying it to a control problem involves several steps:

1. Fuzzification
2. Fuzzy rule evaluation (fuzzy inference engine)
3. Defuzzification

The total fuzzy inference system is a mechanism that relates the inputs to a specific output or set of outputs. First, the inputs are categorized linguistically (fuzzification), then the linguistic inputs are related to outputs (fuzzy inference) and, finally, all the different outputs are combined to produce a single output (defuzzification).{7}

3.2.1.1 Fuzzification

Fuzzification is the conversation of crisp numerical values into fuzzy linguistic quantifiers. Fuzzification is performed using membership functions. Each membership function evaluates how well the linguistic variable may be described by a particular fuzzy qualifier. In other words, the membership function derives a number that is representative of the suitability of the linguistic variable to be classified by the fuzzy variable (set). This suitability is often described as the degree of membership. In order to maintain a relationship to traditional binary logic, the membership values must range from 0 to 1 inclusive. Since each input has a number of membership functions (one for each fuzzy variable), the outputs of all the membership functions for a particular crisp numerical input are combined to form a fuzzy vector. Any number of normalizing expressions can perform fuzzification. Two of the more common functions are the linear and Gaussian .In both cases there is one parameter, m that indicates the midpoint of the region and another, s that defines the width of the membership functions. For the linear function, the width is specified by σ_L and the midpoint by μ_L . Similarly for the

Gaussian function, the width is specified by σ_G and the midpoint by μ_G . Equations (3.4(a)) and (3.4(b)) define the linear and Gaussian membership functions, respectively :{ 10}

$$\left\{ \begin{array}{l} 1 - \frac{|x - \mu_L|}{\sigma_L} \text{ if } x \in [(\mu_L - \sigma_L), (\mu_L + \sigma_L)] \\ 0 \text{ otherwise} \end{array} \right. \dots\dots\dots (3.4(a))$$

$$\exp\left(-\frac{(x - \mu_G)^2}{2(\sigma_G)^2}\right) \dots\dots\dots (3.4(a))$$

Where

$$\mu_L = \mu_G \dots\dots\dots (3.5(a))$$

$$\mu_L = 3\sigma_G \dots\dots\dots (3.5(b))$$

The relations expressed by Eq.(3.5(a)) and (3.5(b)) are made because of the characteristics of the Gaussian function. Because a Gaussian membership function may never have a membership value of 0, some appropriate value close to zero must be chosen as the cutoff point. At a distance of $3\sigma_G$ from the mean, the Gaussian membership function results in a membership value of 0.05. Thus the width of the Gaussian function is chosen as $3\sigma_G$.

As previously mentioned, fuzzification of the input has resulted in a fuzzy vector where each component of this vector represents the degree of membership of the linguistic variables into a specific fuzzy variable's category. The number of components of the fuzzy vector is equal to the number of fuzzy variables used to categorize specific linguistic variables. For illustrative purposes, we consider an example with linguistic variable x and three fuzzy variables positive (PV) zero (ZE) and negative (NV). If we describe the membership function (fuzzifier) as X, then we will have three membership functions: XPV, XZE and XNV. The fuzzy

linguistic universe for the input x can be described by the set U_x that is defined in Eq.

$$U_x = [X_{NV} \ X_{ZE} \ X_{PV}] \dots \dots \dots (3.6)$$

Where

X_{PV} : is the membership function for the Positive fuzzy variables

X_{ZE} : is the membership function for the Zero fuzzy variables

X_{NV} : is the membership function for the Negative fuzzy variables.

Thus, the fuzzy vector, which is the output of the fuzzification step of the inference system, can be denoted by U_x :

$$U_x = [X_{NV} \ X_{ZE} \ X_{PV}] = [X_{NV} \ X_{ZE} \ \dots \ X_{PV}] \dots \dots \dots (3.7)$$

Where

X_{NV} : the membership value of x into the fuzzy region denoted by Negative.

X_{ZE} :the membership value of x into the fuzzy region denoted by Zero .

X_{PV} : the membership value of x into the fuzzy region denoted by Positive.

Equation (3.4(a)) represents the linear membership functions, which are illustrated in Fig (3.2). The linear function can be modified to form the linear-trapezoidal function. Under this modification, if the input x falls between zero and the mean, μ_L , of the respective region, then Eq. (3.5(a)) is used;

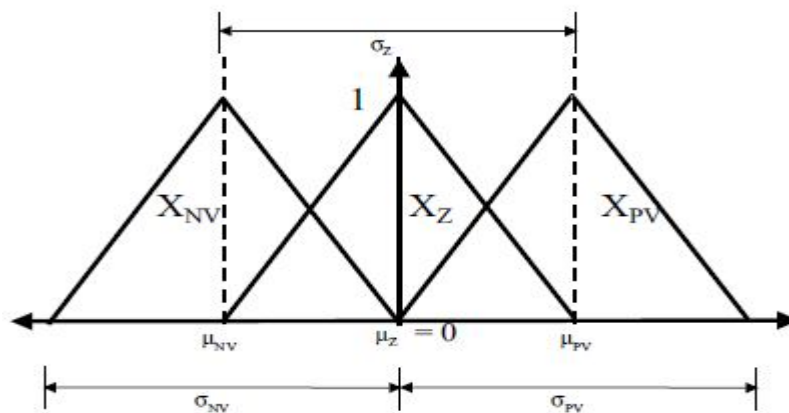


Figure (3.6):Linear membership functions

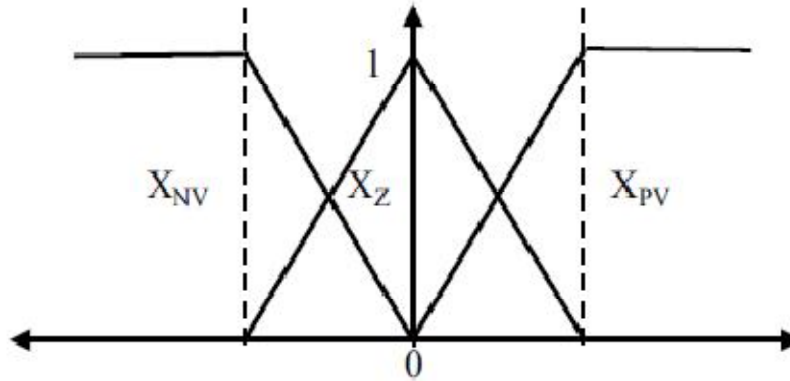


Figure (3.7): Linear-trapezoidal membership function

Otherwise, the membership value is equal to 1. Thus, it is arrived at the membership functions shown in Fig.3. Each region of the linear and trapezoidal-linear membership functions is distinguished from another by the different values of σ_L and μ_L . One important criterion that should be taken into consideration is that the union of the domain of all membership functions for a given input must cover the entire range of the input. Thus the trapezoidal modification is often employed to ensure coverage of the entire input space. {8}

The Gaussian membership function is characterized by Eq.(3.5(b)). The Gaussian function can also be modified to form the trapezoidal-Gaussian function. In this case, if the input falls between the mean μ_G and zero, Eq. (3.5(b)) is used to find the membership value. Otherwise the membership value becomes 1. The Gaussian function is shown in Fig (3.4) and the modified version is shown in Fig (3.5).

A third type of membership function, known as the fuzzy singleton, is also considered. The fuzzy singleton is a special function in which the membership value is 1 for only one particular value of the linguistic input variable and zero otherwise. Thus, the fuzzy singleton is a special case of the membership function with a width, σ , of zero. Therefore, the only parameter that needs to be defined is the mean, μ_s , of the singleton. Thus, if the input is equal to μ_s , then the membership value is 1. Otherwise it is zero. The singleton membership function as $S(\mu_s)$.

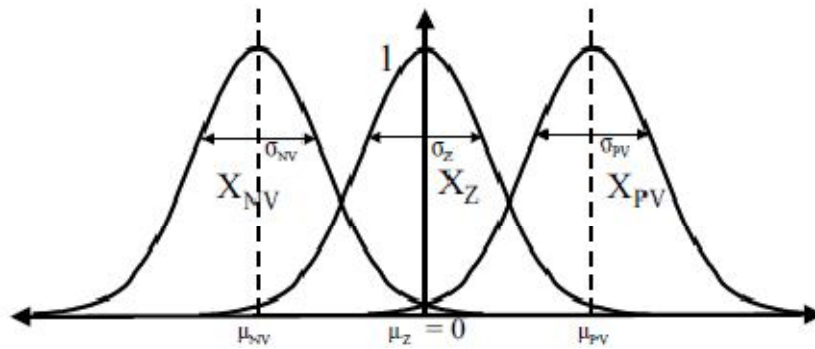
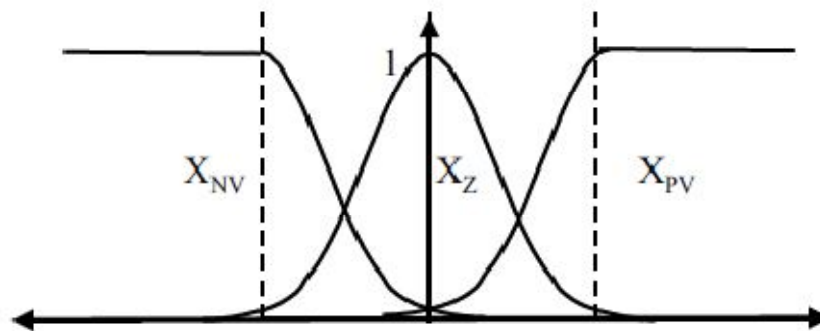


Figure (3.8):Gaussian membership functions.



Figure(3.9): Gaussian-trapezoidal membership functions.

The fuzzy singleton function is quite useful in defining some special membership functions. If we would like to dispense with the need for a continuous degree of membership and prefer a binary valued function, the fuzzy singleton is an ideal candidate. We can form the membership function representing the fuzzy variable as a collection of fuzzy singletons ranging within the regions denoted by $[\mu + \sigma /2, \mu - \sigma /2]$.

A graphical representation using three fuzzy variables (membership functions) is shown in Fig (3.6) (In Figure the corners are only slanted so that the regions are easier to distinguish from each other.) Thus, the membership functions shown in Fig.6 would be best defined as an integral of the fuzzy singleton with respect to the mean over the width of the function. Consequently, the membership function $X(\sigma, \mu)$ would be defined as follows:

$$X(\sigma, \mu) = \int_{\mu-6/2}^{\mu+6/2} S(\mu_s) d\mu \dots\dots\dots(3.8)$$

Where

$S(\mu_s)$ is the singleton function.

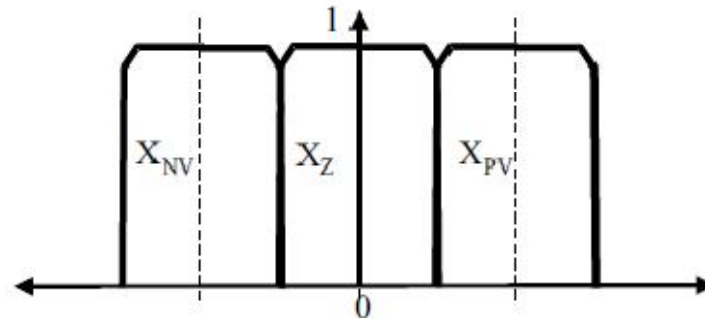


Figure (3.10): Membership functions comprised of fuzzy singletons.

Other function used in fuzzification

- The sigmoid curve
- Quadratic and cubic polynomial curves

You define the sigmoid membership function, which is either open left or right. Asymmetric and closed (i.e. not open to the left or right) membership functions can be synthesized using two sigmoid functions, so in addition to the basic sigmf, you also have the difference between two sigmoid functions, dsigmf, and the product of two sigmoid functions psigmf.

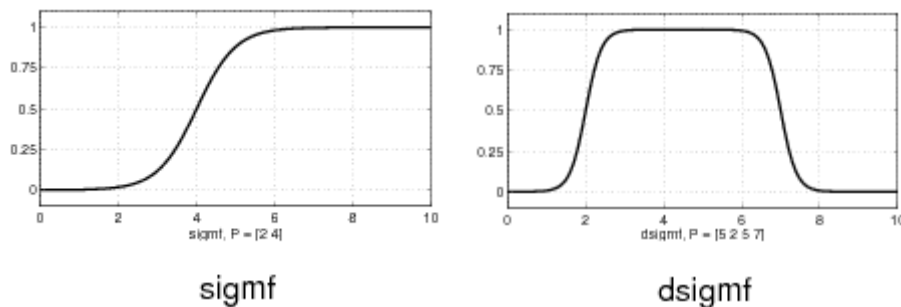


Figure (3.11): sigmoid membership function

Polynomial based curves account for several of the membership functions in the toolbox. Three related membership functions are the Z, S, and P curves, all named because of their shape. The function zmf is the asymmetrical polynomial curve open to the left, smf is the mirror-image function that opens to the right, and pmf is zero on both extremes with a rise in the middle

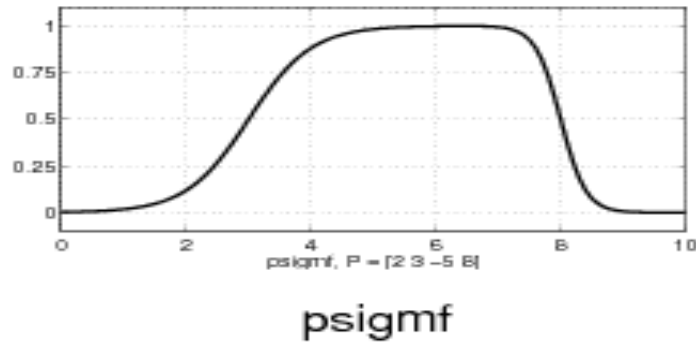


Figure (3.12): middle sigmoid membership function

3.2.1.2 The Fuzzy Inference Engine

The fuzzy inference engine uses the fuzzy vectors to evaluate the fuzzy rules and produce an output for each rule. Figure 1 shows a block diagram of the fuzzy inference engine. Note that the rule-based system takes the form found in Eq. This form could be applied to traditional logic as well as fuzzy logic albeit with some modification. A typical rule R would be:

$$R_i: \text{ IF } x_i \text{ THEN } y = C_i \dots \dots \dots (3.9)$$

Where

X_i is the result of some logic expression.

The logical expression used in the case of fuzzy inference Eq (3.) is of the form

$$x \in X_i \quad (3.7)$$

Where

x is the input and X_i is the linguistic variable.

In binary logic, the expression in Eq (3.7) results in either true or false. However, in fuzzy logic we often require continuum of truth values. Figures (3.7) and (3. 8) illustrate the difference between binary logic and fuzzy logic. In traditional logic, there is a single point representing the boundary between true and false. While in fuzzy logic, there is an entire

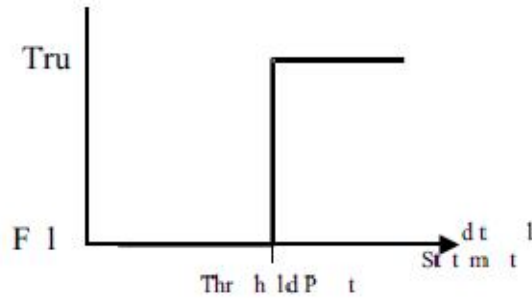


Figure (3.13):Binary logic statement evaluation.

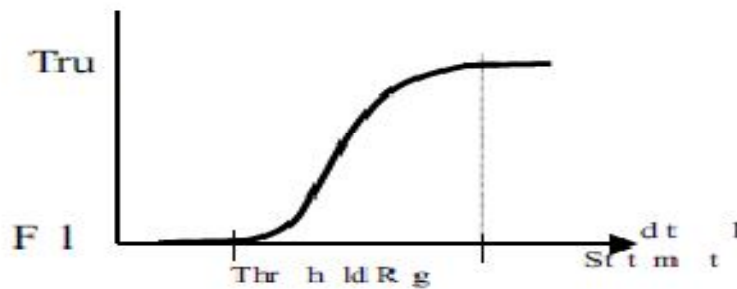


Figure (3.14):Fuzzy logic statement evaluation.

region over which there is a continuous variation between truth and falsehood. The second part of Eq(3.9), $y=C_i$, is the action prescribed by the particular rule. This portion indicates what value will be assigned to the output. This value could be either a fuzzy linguistic description or a crisp numerical value.

Logical Operations in fuzzy logic

Now that you understand the fuzzy inference, you need to see how fuzzy inference connects with logical operations.

The most important thing to realize about fuzzy logical reasoning is the fact that it is a superset of standard Boolean logic. In other words, if you keep the fuzzy

values at their extremes of 1 (completely true), and 0 (completely false), standard logical operations will hold. As an example, consider the following standard truth tables.

Table (3.1) standard logical operations truth table

A	B	A and B
0	0	0
0	1	0
1	0	0
1	1	1

A	B	A or B
0	0	0
0	1	1
1	0	1
1	1	1

A	not A
0	1
1	0

AND

OR

NOT

Now, because in fuzzy logic the truth of any statement is a matter of degree, can these truth tables be altered? The input values can be real numbers between 0 and 1. What function preserves the results of the AND truth table (for example) and also extend to all real numbers between 0 and 1?

One answer is the min operation. That is, resolve the statement A AND B, where A and B are limited to the range (0, 1), by using the function $\min(A,B)$. Using the same reasoning, you can replace the OR operation with the max function, so that A OR B becomes equivalent to $\max(A, B)$. Finally, the operation NOT A becomes equivalent to the operation. Notice how the previous truth table is completely unchanged by this substitution.

Table (3.2) logical operations truth table

A	B	Min(A,B)
0	0	0
0	1	0
1	0	0
1	1	1

A	B	Max(A,B)
0	0	0
0	1	1
1	0	1
1	1	1

A	1-A
0	1
1	0

AND

OR

NOT

Moreover, because there is a function behind the truth table rather than just the truth table itself, you can now consider values other than 1 and 0. The next figure uses a graph to show the same information. In this figure, the truth table is converted to a plot of two fuzzy sets applied together to create one fuzzy set. The upper part of the figure displays plots corresponding to the preceding two-valued truth tables, while the lower part of the figure displays how the operations work over a continuously varying range of truth values A and B according to the fuzzy operations you have defined. {7}

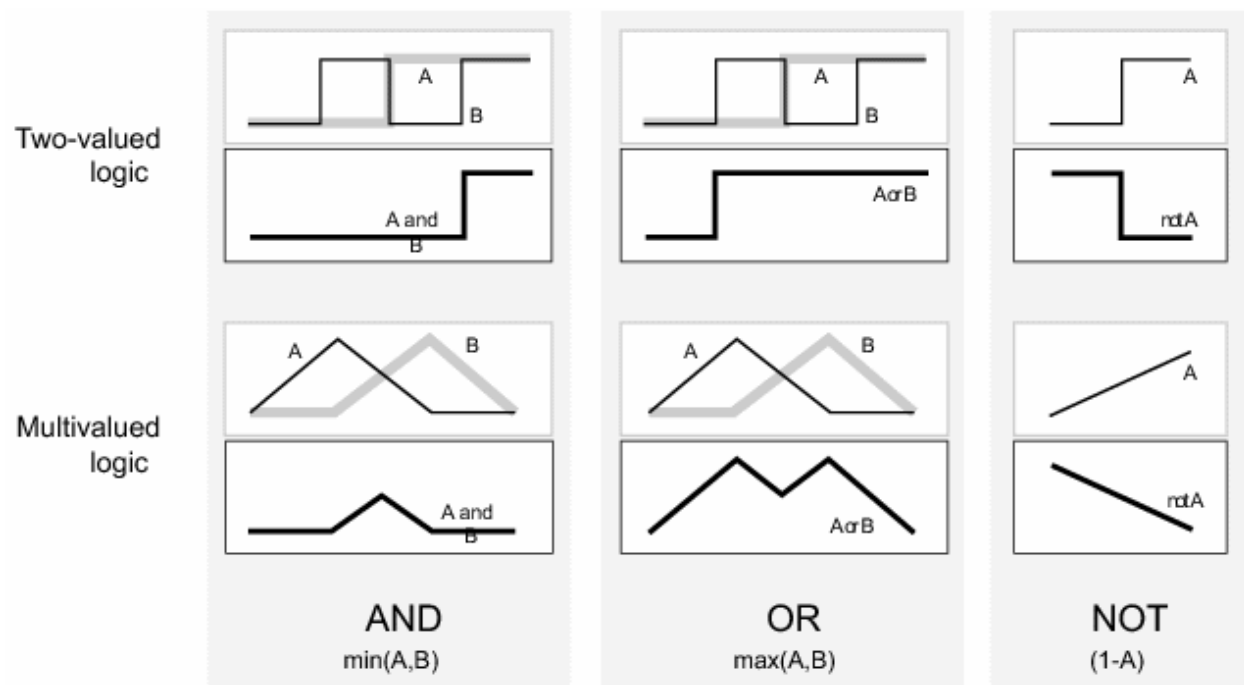


Figure (3.15) logical operation used in fuzzy logic

Given these three functions, you can resolve any construction using fuzzy sets and the fuzzy logical operation AND, OR, and NOT. The logical expression that dictates whether the result of particular rule is carried out could involve multiple criteria. Multiple conditions imply multiple inputs, as is most often the case in many applications of fuzzy logic to dynamic systems. Let us describe a system with two inputs x^1 and x^2 . For simplicity of explanation and without loss of

generality, we will use 3 fuzzy variables, namely PV, ZE and NV. Although each linguistic variable x^1 and x^2 uses the same fuzzy qualifiers, each input must have its own membership functions since they belong to different spaces. Thus, we will have two fuzzy vectors x^1 and x^2 , for the first and second input, respectively:

$$X^1 = \{X^1_{NV}(x^1_1) X^1_{ZE}(x^1_2) X^1_{PV}(x^1_3)\} = \{x^1_1 x^1_2 x^1_3\} \dots\dots\dots (3.10)$$

$$X^2 = \{X^2_{NV}(x^2_1) X^2_{ZE}(x^2_2) X^2_{PV}(x^2_3)\} = \{x^2_1 x^2_2 x^2_3\} \dots\dots\dots (3.11)$$

Where

X^n_{NV} membership function for the Negative fuzzy variable for input n

X^n_{ZE} membership function for the Zero fuzzy variable for input n

X^n_{PV} membership function for the Positive fuzzy variable for input n

x^1 first linguistic variable (input 1)

x^2 second linguistic variable (input 2)

X_{1i} the degree of membership of input 1 into the i th fuzzy variable's category

X_{2j} the degree of membership of input 2 into the j th fuzzy variable's category.

The inference mechanism in this case would be specified by the rule R_{ij} :

$$R_{ij} : \text{IF } \dots x^1_i \text{ AND } x^2_j \text{ THEN } y = C_i: \dots\dots\dots (3.12)$$

The specification R_{ij} is made so as to emphasize that all combinations of the Components of the fuzzy vectors should be used in separate rules.

A specific Example of one of the rule scan be described as follows:

$$R_{13}: \text{IF } x^1_1 \text{ AND } x^2_3 \text{ THEN } y=C_{13}:$$

This rule can be written linguistically as

$$R_{13}: \text{IF } x^1 \text{ is Negative AND } x^2 \text{ is Positive THEN } y = C_{13}$$

Where

X^1 is the first linguistic variable

X^2 is the second linguistic variable

C_{13} is the output action to be defined by the system designer.

The AND in Eq.(3.12) can be interpreted and evaluated in two different ways. First, the AND could be evaluated as the product of x^1_i and x^2_j . Thus

$$x^1_i \text{ AND } x^2_j = x^1_i x^2_j$$

The second method is by taking the minimum of the terms. In this case the result is the minimum value of the membership values. Therefore

$$x^1_i \text{ AND } x^2_j = \min(x^1_i, x^2_j) \dots \dots \dots (3.13)$$

The most commonly used method in the literature is the product method. Therefore, the product method is used here to evaluate the AND function. Equation (3.10) can be expanded to multiple inputs with multiple fuzzy variables. In the most general case of n inputs and k linguistic qualifiers, we would have the rule R_i :

$$R_i: \text{ IF } [x^1_i \text{ AND } x^2_i \text{ AND } \dots \text{ AND } x^n_{ij}] \text{ THEN } y = C_i (3.12) \dots \dots \dots (3.14)$$

Recalling that all combinations of input vector components must be taken between fuzzy vector components the system designer could have up to n_k rules, where n is the number of fuzzy variables used to describe the inputs and k is the number of inputs. The number of rules used by the fuzzy inference engine could be reduced if the designer could eliminate some combinations of input conditions. {10}

3.2.1.3 Defuzzification

The fuzzy inference engine as described previously often has multiple rules, each with possibly a different output. Defuzzification refers to the method employed to combine these many outputs into a single output. Using Eq (3.12) where multiple inputs $x^1 x^2 \dots x^n$ should be evaluated, the product due to the evaluation of the premise conditions (defined by the components of the fuzzy vectors) determines the strength of the overall rule evaluation, w_i :

$$w_i = x^1_i \text{ AND } x^2_i \text{ AND } \dots \text{ AND } x^n_i \dots \dots \dots (3.15)$$

$$w_i = x^1_i x^2_i \dots x^n_i$$

Where, x_{ki} is the membership value of the kth input into the ith fuzzy variable's category. This value, w_i , becomes extremely important in defuzzification. Ultimately, defuzzification involves both the set of outputs C_i and the corresponding rule strength w_i .

There are a number of methods used for defuzzification, including the center of gravity (COG) and mean of maxima (MOM). The COG method otherwise known as the fuzzy centroid is denoted by y_{COG} .

3.2.1.3.1 Center of gravity(COG)

The crisp output value x (white line in Fig(3.8)) is the abscissa under the center of gravity of the fuzzy set,

$$U = \frac{\sum_i \mu(x_i)x_i}{\sum_i \mu(x_i)} \dots \dots \dots (3.16)$$

Here x_i is a running point in a discrete universe, and $\mu(x_i)$, is its membership value in the membership function. The expression can be interpreted as the weighted average of the elements in the support set. For the continuous case, replace the summations by integrals.

It is a much used method although its computational complexity is relatively high. This method is also called centroid of area. {4}

3.2.1.3.2 Center of gravity method for singleton (COGS)

If the membership functions of the Conclusions are singletons Fig (3.8), the output value is

$$U = \frac{\sum_i \mu(s_i)s_i}{\sum_i \mu(s_i)} \dots \dots \dots (3.17)$$

Here s_i is the position of singleton i in the universe, and $\mu(s_i)$, is equal to the firing strength of rule i . This method has a relatively good computational complexity, and

x is differentiable with respect to the singletons s_i , which is useful in neurofuzzy systems. {4}

3.1.2.3.3 Mean of maxima (MOM)

An intuitive approach is to choose the point with the strongest possibility, i.e. maximal membership. It may happen, though, that several such points exist, and a common practice is to take the mean of maxima (MOM). This method disregards the shape of the fuzzy set, but the computational complexity is relatively good.

The defuzzifier must then choose one or the other, not something in between. These methods are indifferent to the shape of the fuzzy set, but the computational complexity is relatively small. {4}

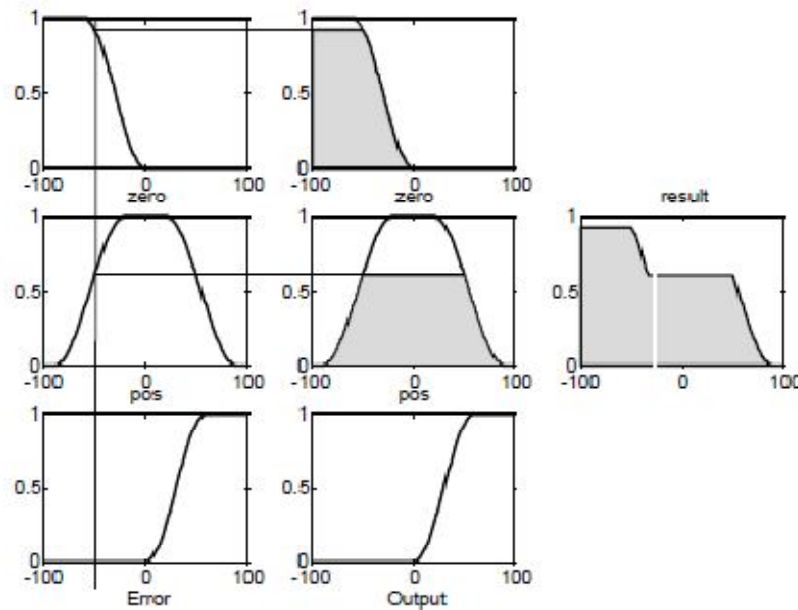


Figure (3.16) one input one output rule base with non-singleton output sets

CHAPTER FOUR
SIMULATION RESULTS AND DISCUSSION

4.1 Case study

The parameters for used servo motor are:

$$K_T \text{ (N.m/A)} = 0.121$$

$$K_B \text{ [V/(rad/s)]} = 0.121$$

$$R_a \text{ (\Omega)} = 2.23$$

$$B \text{ [N.m/(rad/s)]} = 0.0000708$$

$$J_m \text{ (kg.m}^2\text{)} = 0.00006286$$

$$B \text{ [N.m/(rad/s)]} = 0.0000708$$

$$K_p = 15$$

$$K_i = 5$$

$$K_d = 0.5$$

Thus, by substituting these values into equation (2.8), the transfer function of the servo motor is as follows.

$$\frac{\Theta_m(t)}{E_a(t)} = \frac{863.19}{s^2 + 105.58 s} \dots \dots \dots (4.1)$$

4.2 Simulation Results of PID

This section demonstrates the simulation results of speed control of servo motor (case study) the controllers are designed using computational optimization approach method. the result is obtained with unit step response using MATLAB The simulink diagram of the PID controller for speed control of the servo motor is shown in figure(4.1)

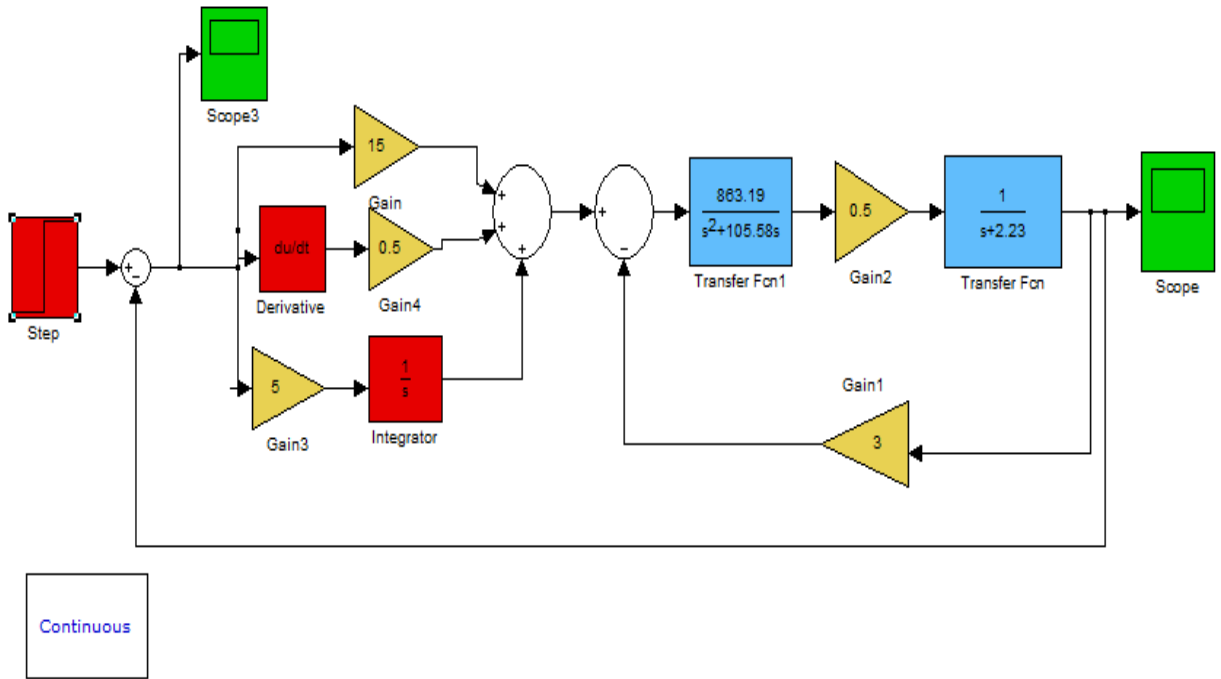


Figure (4.1) Simulink diagram of PID controller for speed control of servo motor.

Where the unit step response of the PID controller for position control of the servo motor is shown in figure (4.2).

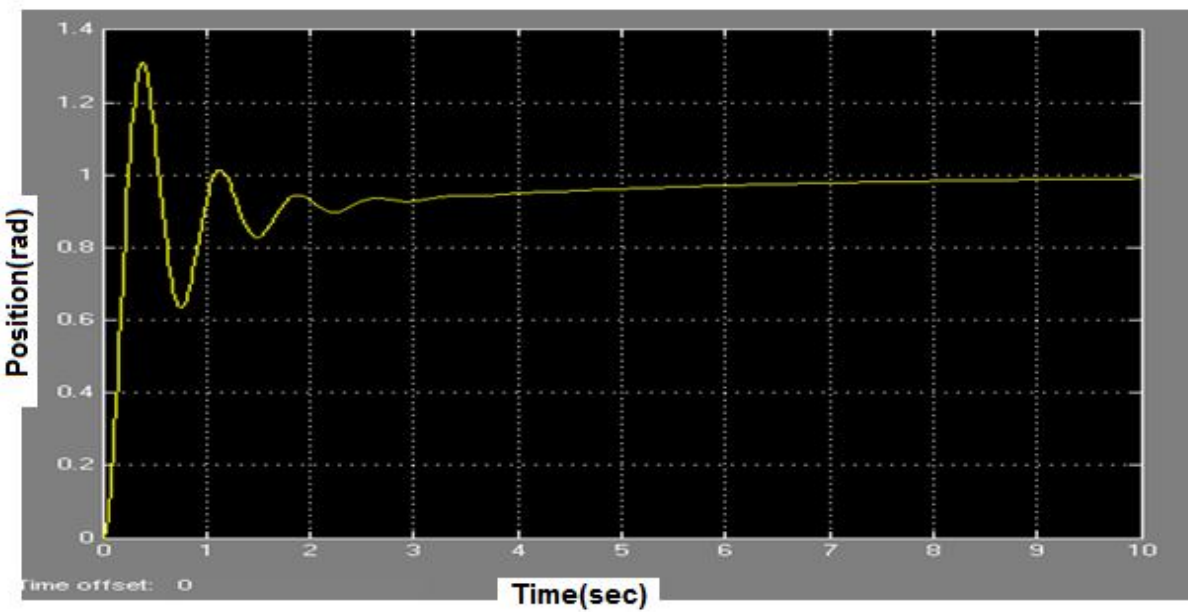


Figure (4.2): unit step response of PID controller for position control of servo motor

4.3 Simulation Results of the fuzzy logic controller

The simulink diagram of Fuzzy logic controller for position control of the servo motor is shown below in figure (4.3).

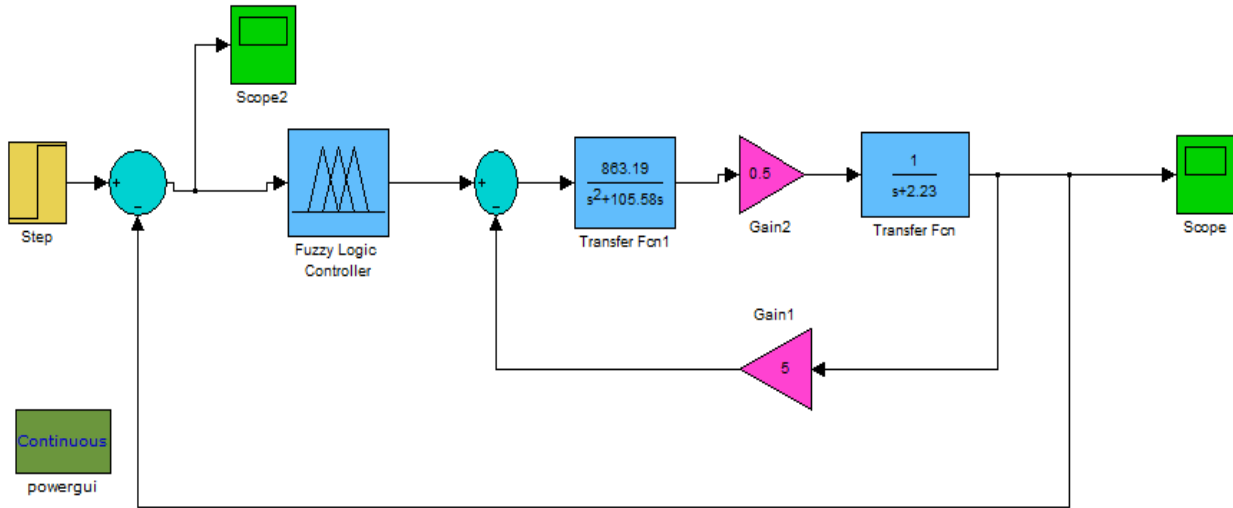


Figure (4.3): simulink diagram of Fuzzy for position control of servo motor.

Figures (4.4) show unit step response of the fuzzy logic controller for position control of servo motor.

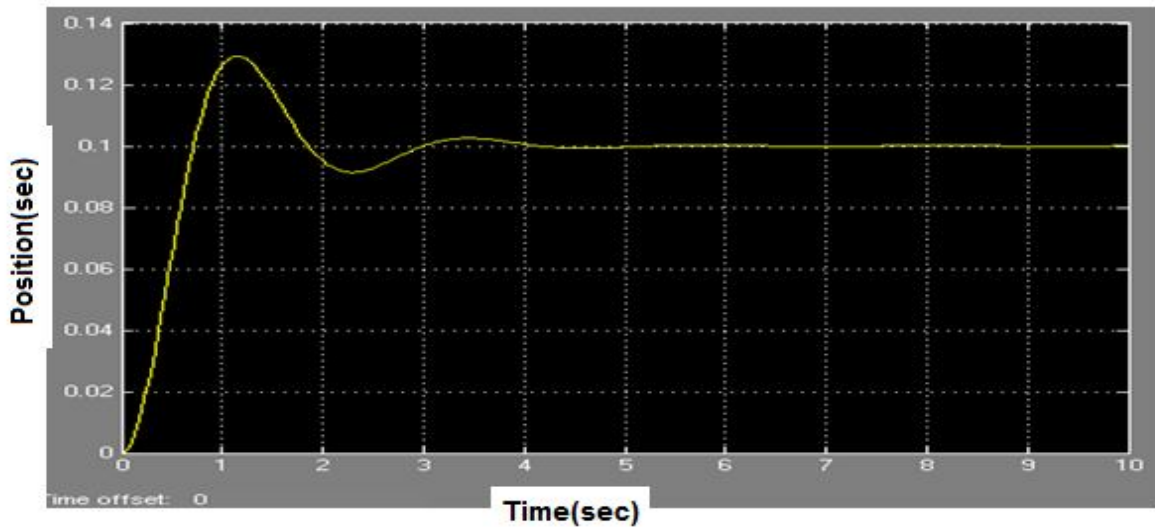


Figure (4.4): unit step response of FLC for position control of servo motor

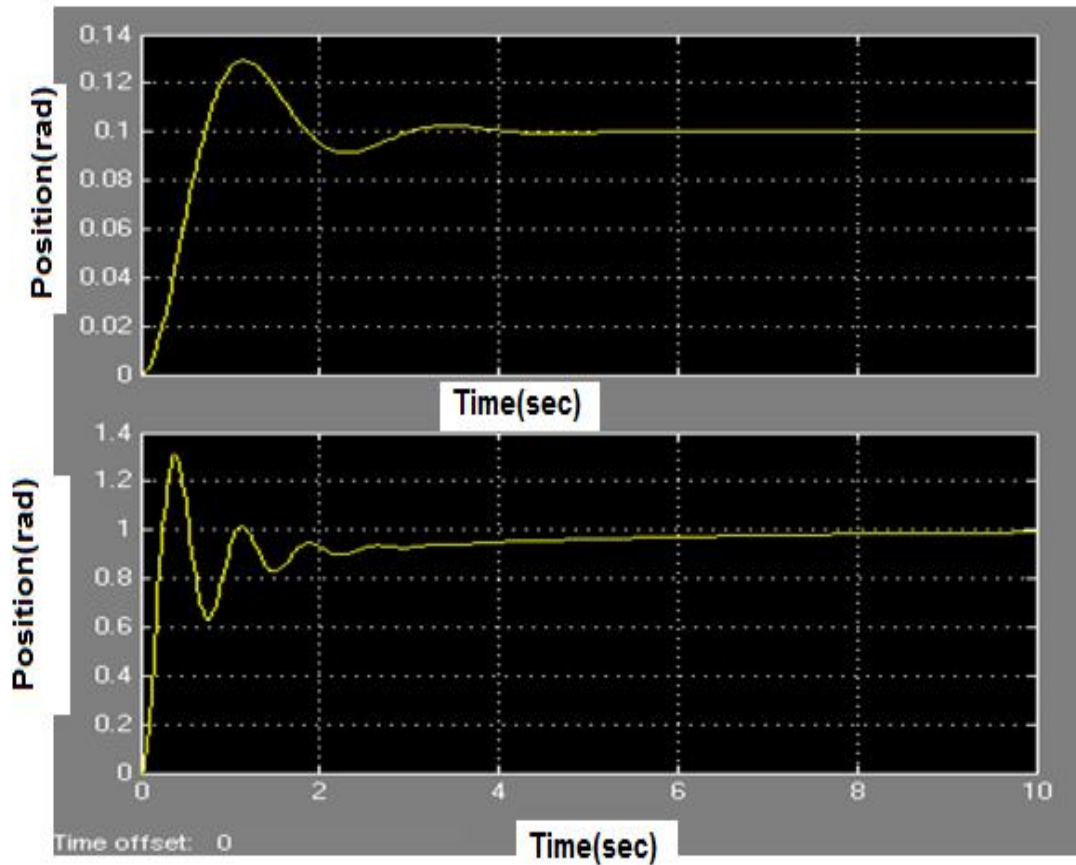


Figure (4.5): unit step system response for the PID and fuzzy logic controllers.

4.4 Discussion of the results

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 chapter. The MATLAB/SIMULINK model of system under study with the both controllers is shown in Figures (4.1), (4.3).

The results show that the overshoot for PID controller is higher than the fuzzy logic controller as table (4.1)

Table (4.1): overshoots of the PID and fuzzy logic controllers

Controller	Overshoot MP
PID	1.28
Fuzzy	1.24

Never the less the results show also the peak time for PID controller is smaller than fuzzy logic controller. And setting time for PID controller is larger than the fuzzy logic controllers, as illustrated in tables (4.2) and (4.3)

Table (4.2): peak time of the PID and fuzzy logic controllers.

Controller	T_p (sec)
PID	0.4
Fuzzy	1.2

Table (4.3): settling time of the PID and fuzzy logic controllers.

Controller	T_s (sec)
PID	9.8
Fuzzy	6.4

CHAPTER FIVE

CONCLUSION AND RECOMMENDATIONS

5.1 Conclusion

In this thesis, the servo motor has been reviewed from control theory perspective. PID controller with computational optimization approach and intelligent techniques such as Fuzzy logic Controllers for servo motor position control are discussed in some details.

The study is based on the MATLAB Simulink software platform and mathematical models of the servo motor.

For the design using PID controller with computational optimization approach the term values is selected, which presents the need of a lot of attempts to choose the right term, which give a good response.

Design using fuzzy logic controller was carried out also, and it gave perfect results. The overshoot, oscillation and setting time are found to be better with the fuzzy logic controller compared with PID controller.

Although the results show that the fuzzy logic controller is better for position control of the servo motor, there are some limitation with the fuzzy logic controller, implemented in lack of a systematic design methodology and the difficulty in predicting the stability and robustness of the controlled system.

5.2 Recommendations

- Servo motor position control can be extended in the future using modern techniques such as state space method, space vector PWM and artificial neural networks.
- PID controller and fuzzy logic controller can be used together for control position of servo motor.
- PD controller and fuzzy logic controller can be used together for control position of servo motor.
- The system can be tested experimentally, and then a comparison can be made between the experimental results and the obtained results using simulink modeling, for servo motor position control to investigate the results of simulink.

Appendix

Fuzzy logic controller program

```
[System]
Name='basha'
Type='mamdani'
Version=2.0
Num Inputs=2
Num Outputs=1
Num Rules=25
And Method='min'
Or Method='max'
Imp Method='min'
Agg Method='max'
Defuzz Method='centroid'

[Input1]
Name='step'
Range = [0 1]
Num MFs=5
MF1='vs':'trimf',[0 0.2 0.3]
MF2='s':'trimf',[0.1 0.3 0.4]
MF3='m':'trimf',[0.2 0.4 0.6]
MF4='b':'trimf',[0.5 0.6 0.8]
MF5='vb':'trimf',[0.7 0.85 1]

[Input2]
Name='feed'
Range=[0 1]
```

```
Num MFs=5
MF1='vs':'trimf',[0 0.2 0.3]
MF2='s':'trimf',[0.2 0.35 0.5]
MF3='m':'trimf',[0.4 0.55 0.7]
MF4='b':'trimf',[0.6 0.8 0.9]
MF5='vb':'trimf',[0.8 0.9 1]
```

```
[Output1]
```

```
Name='output1'
```

```
Range=[0 1]
```

```
Num MFs=5
```

```
MF1='vs':'trimf',[0 0.2 0.35]
MF2='s':'trimf',[0.2 0.35 0.5]
MF3='m':'trimf',[0.4 0.6 0.7]
MF4='b':'trimf',[0.6 0.8 0.9]
MF5='vb':'trimf',[0.8 0.9 1]
```

```
[Rules]
```

```
1 1, 5 (1): 1
1 2, 5 (1): 1
1 3, 5 (1): 1
1 4, 4 (1): 1
1 5, 4 (1): 1
2 1, 4 (1): 1
2 2, 4 (1): 1
2 3, 4 (1): 1
2 4, 4 (1): 1
2 5, 4 (1): 1
3 1, 4 (1): 1
```

3 2, 4 (1) : 1
3 3, 4 (1) : 1
3 4, 4 (1) : 1
3 5, 4 (1) : 1
4 1, 3 (1) : 1
4 2, 3 (1) : 1
4 3, 3 (1) : 1
4 4, 2 (1) : 1
4 5, 2 (1) : 1
5 1, 2 (1) : 1
5 2, 2 (1) : 1
5 3, 2 (1) : 1
5 4, 1 (1) : 1
5 5, 1 (1) : 1

References

- [1] C.C.kung and.-H.Su “*Adaptive fuzzy position control for electrical servo drive*” via total –sliding –mode technique,IEE Proc.-Electr.Power Appl.,Vol.152, No.6,2005
- [2]Jin Z.,Shuyun W. and Jinbang “*Improving method fuzzy control AC speed drive*” IEEE Proceeding of the 5th World Congress on Intelligent Control and Automation ,PP.4500-4505,2004
- [3] Li-Xin Wang “*Analysis and design of hierarchical fuzzy*” IEEE Tans. On Fuzzy systems.Vol.7,no.5,617-624,1999
- [4] Chekunor ,Aziehailma, Bintiche,Chekumelor”*fuzzy controller design for position control* ”Tun Hussein On Malaysia Electrical and Electronic Engineering, January 2012.
- [5] Mohamed Farid Bin Mohamed Faruq “*PID controller design for controlling DC motor speed using matlab application*” University Malaysia Pahang November 2008
- [6] S.R. Khuntia, K.B. Mohanty and S. Panda and C. Ardi “*A Comparative Study of PI, I-P, Fuzzy and Neuro-Fuzzy Controllers for Speed Control of DC Motor Drive*” International Journal of Electrical and Computer Engineering 5:5 2010
- [7] Boumediene, Allaoua, Abdellah Laoufi, Gasbaoui,And Abdessalam Abderrahmani “*Fuzzy DC Motor speed Control Using Particle Swarm Optimization* "Leonardo Electronic Journal and Technologies ,15-December 2009
- [8] LEONID REZINK” *Fuzzy Controllers*” Victoria University of echnology, 1997.
- [9] Jan Jantzen” *Design of fuzzy logic controller*” Technical University of Denmark Department of Automation Bldg 326, DK -2800 Lyngby, Denmark. Tech report no 98-E 864(design),19 Aug 1998

- [10] Chee-Mun Ong, Dynamic simulation of Electric Machinery using Matlab/Simulink, 1998 Prentice Hall.
- [11] P.S. Bimbhra, Generalized “theory of Electrical Machines” Khanna Publishers.
- [12] A.E. Fitzgerald, Charles Kingsley Jr., Stephen D. Umans “Electric Machinery” fourth edition 1983 McGraw-Hill,
- [13] Kilian “Modern control technology – component and system” 2nd edition