INTELLIGENT CONTROL OF ROBOTIC SYSTEMS

by

Mehmet CELLEK

July 2006

INTELLIGENT CONTROL OF ROBOTIC SYSTEMS

by

Mehmet CELLEK

A thesis submitted to

the Graduate Institute of Sciences and Engineering

of

Fatih University

in partial fulfillment of the requirements for the degree of

Master of Science

in

Electronics Engineering

July 2006
Istanbul, Turkey

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science.

Prof. Dr. Muhammet KÖKSAL
Head of Department

This is to certify that I have read this thesis and that in my opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master of Science.

Assoc. Prof. Dr. Erkan İMAL
Supervisor

Examining Committee Members:

1. Prof. Dr. Muhammet KÖKSAL

2. Assoc. Prof. Dr. Erkan İMAL

3. Assist. Prof. Dr. M. Atakan KURT

It is approved that this thesis has been written in compliance with the formatting rules laid down by the Graduate Institute of Sciences and Engineering.

Assist. Prof. Dr. Nurullah ARSLAN
Director

Date
July 2006

INTELLIGENT CONTROL OF ROBOTIC SYSTEMS

Mehmet CELLEK

M. S. Thesis - Electronics Engineering
July 2006

Supervisor: Assoc. Prof. Dr. Erkan İMAL

ABSTRACT

In recent years, intelligent control techniques such as fuzzy inference control or fuzzy logic control has received attention of a number of researchers in the area of power electronics, robotic systems, and motion control. This thesis investigates MATLAB simulations of a position control system, where fuzzy logic, PID, and Fuzzy-PID controllers are used. For each type of controllers, various cases are considered. Maximum overshoot, settling time, steady-state error parameters for each case are calculated. Then, the performance characteristics are compared with each other. The simulation results clearly indicate that, Fuzzy Logic technique has been proven to be an excellent solution method for control problems where the number of rules for a system is finite. The FLC has also proven to be a robust controller. Therefore, FLC can be successfully implemented in systems which need robust performance. The total error of the control loop can be reduced according to the requirements of a system by tuning the FLC. The tuning process is the modification of fuzzy sets and fuzzy rule base. The combination of Fuzzy Logic and PID controller has yielded successful results which are worth considering.

Keywords: Intelligent Control, Fuzzy Logic Controller, Fuzzy-PID Controller, PID Controller

ROBOTİK SİSTEMLERİN AKILLI KONTROLU

Mehmet CELLEK

Yüksek Lisans Tezi – Elektronik Mühendisliği
Temmuz 2006

Tez Yöneticisi: Doç. Dr. Erkan İMAL

ÖZ

Son yıllarda, Bulanık Çıkarım Denetimi veya Bulanık Mantık Denetimi gibi akıllı denetim teknikleri, güç elektroniği, robotik sistemler ve hareket denetimi gibi alanlarda çalışan birçok araştırmacının dikkatini çekmiştir. Bu tezde, bulanık mantık, PID ve bulanık-PID denetleyicilerini kullanan bir konum denetim sistemi Matlab simülasyonları ile incelenmektedir. Denetleyicilerin her biri için birçok durum göz önüne getirilmiştir. Her bir durum için, Maksimum Geçiş, Durulma Zamanı ve Kararlı-Durum Hata parametreleri hesaplanmıştır. Daha sonra, her birinin performans karakteristikleri karşılaştırılmıştır. Simülasyon sonuçları, sınırlı sayıda kural gerektiren sistemlerin denetleme problemlerinin çözümünde, bulanık mantık tekniğinin mükemmel olduğunu göstermiştir. Bulanık mantık denetleyicisinin, aynı zamanda, dayanıklı bir denetleyici olduğu ispatlanmıştır. Bundan dolayı, bulanık mantık denetleyicisi, dayanıklı performans gerektiren sistemlerde başarı ile uygulanabilir. Denetleme döngüsünün toplam hatası, sistem gereksinimlerine bağlı olarak, bulanık mantık denetleyicisinin ayarlanmasıyla azaltılabilir. Ayarlama işlemi, bulanık kümelerin ve bulanık kural tabanının değiştirilmesiyle yapılır. Bulanık mantık ve PID denetleyicilerinin kombinasyonu ile dikkate değer başarılı sonuçlar alınmıştır.

Anahtar Kelimeler: Akıllı Denetim, Bulanım Mantık Denetleyici, Bulanık-PID Denetleyici, PID Denetleyici.

DEDICATION

To my parents

&

To affection, reconciliation, and friendship of the WORLD

ACKNOWLEDGEMENT

I express sincere appreciation to my advisor Assoc. Prof. Dr. Erkan İMAL for his guidance and insight throughout this study.

Thanks go to the other faculty members, Prof. Dr. Muhammet KOKSAL, Assist. Prof. Dr. Onur TOKER, and Research Assist. Cihan ULAŞ for their valuable suggestions and comments.

I express my thanks and appreciation to my family for their understanding, motivation and patience. Lastly, but in no sense the least, I am thankful to all colleagues and friends who made my stay at the university a memorable and valuable experience.

TABLE OF CONTENTS

LIST OF TABLES

TABLE **PAGE**

LIST OF FIGURES

FIGURE **PAGE**

FIGURE **PAGE**

FIGURE **PAGE**

FIGURE **PAGE**

LIST OF SYMBOLS AND ABBREVIATIONS

SYMBOL/ABBREVIATION

FC	Fuzzy Logic
FLC	Fuzzy Logic Control
FIS	Fuzzy Inference System
PID	Proportional, Integrated, Derivative
Fuzzy-PID	Fuzzy- Proportional, Integrated, Derivative
RPM	Revolutions Per Minute
AI	Artificial Intelligence
GA	Genetic Algorithm
A/D	Analog-to Digital
FIS	Fuzzy Inference System
GUI	Graphical User Interface
CHR	Chein, Hrones and Reswick

CHAPTER 1

INTRODUCTION

Intelligent systems have capability of understanding, learning, reasoning, and making inferences or decisions from unfinished information. An intelligent system needs a system in which the representation and processing of knowledge are central functions. For calculative intelligence, soft computing is an important branch where fuzzy logic, probability theory, neural networks, and genetic algorithms are synergistically used in decision making. In this thesis, the use of fuzzy logic in a servo system will be shown.

Controller design for a system requires some knowledge about it. This usually involves a mathematical description of the relation among inputs to the process, its state variables, and its output. This description is called the model of the system. The model can be represented by a set of transfer functions for time invariant linear systems, or other relationships for non-linear or time-varying systems.

Designing of complex systems can be a very difficult task. In a complex system such as a multiple input and multiple output system, inexact models can lead to unstable systems, or unsuitable system performance. Fuzzy Logic Control (FLC) is an impressive alternative approach for systems, which are difficult to model. The FLC uses the qualitative aspects of the human decision process to construct the control algorithm. This can lead to a robust controller design. The modeling of a mobile robot is a very complex task and a direct application of FLC is used in this area (Kelkar, 1997).

1.1 OBJECTIVE

The main aspect addressed in this thesis is the design of an intelligent controller combined with a well-known classical controller for a robotic system of the Cartesian type using Fuzzy Logic, PID and Fuzzy-PID. MATLAB Toolboxes are utilized for the design and simulation of the controllers for the selected mechatronic system.

1.2 THESIS OBJECTIVE

The main goal of this thesis is to design a Fuzzy-PID Controller through MATLAB simulations for a Cartesian Robot System.

The design of the controller is performed in three stages.

In the *first stage,* the universe of discourse is identified, and fuzzy sets are defined. The rule base (Fuzzy Control Rules) for the control is constructed through the operator's expert knowledge. The membership functions and their intervals are defined. Aggregation and defuzzification methods are selected. Then, the performances of the controllers are tested through a series of simulations, and finally the rule base and the membership functions are revised to improve the performance of the system.

In the *second stage,* the PID Controller is designed, and tuned by using the Chien, Hrones and Reswick method, and the Trial and Error approach for getting the best controller for the system. Then, the performance of the controller is tested through a series of simulations of the system.

In the *third* and *final stage*, the first and second stages are combined with each other. All cases are simulated for the combined model. Parameters are tuned to improve the performance of the system.

1.3 OUTLINE OF THE THESIS

Chapter 2 gives a definition of Mechatronic Systems, components of a Robotic System, and also a brief literature survey on robotics is presented.

An overview of the intelligent control, fuzzy set theory in particular, is presented in Chapter 3.

Chapter 4 has a brief introduction about Matlab Fuzzy Logic Toolbox. The designs of Fuzzy Logic Controller, PID Controller, and Fuzzy-PID Controller are given extensively.

Chapter 5 presents the results of the simulation study obtained from MATLAB in detail for the selected cases.

Finally, the conclusions for this thesis are drawn in Chapter 6.

CHAPTER 2

MECHATRONIC SYSTEMS

2.1 MECHATRONICS

2.1.1 Definition

Mechatronic is an interdisciplinary field, or engineering, dealing with the design of products whose function relies on the integration of mechanical and electronic components coordinated by a control architecture. Thus, the primary disciplines in mechatronics are mechanics, electronics, controls and computer engineering as seen in Figure 2.1.

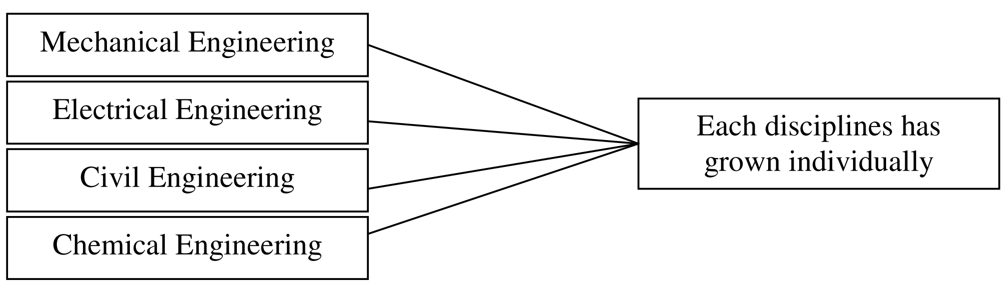

Figure 2.1: Disciplines in Mechatronics.

The field of Mechatronic Engineering has shown a strong development since about 1980. Initially the name for this field was introduced in the Japanese industry involved with the development and manufacturing of Hard-Disc drives. The intense co-operation between engineers from different technical disciplines enabled the creating of new and exciting products, instruments and equipment. Some examples of mechatronic systems are

- Aircraft flight control and navigation system.
- Automobile electronic fuel injection and antilock brake systems.
- Numerically controlled (NC) machine tools.
- Robots.
- Smart kitchen.
- Toys, etc.

A typical mechatronic system consists of the following components:

- Actuators - produce motion or cause some action.
- Sensors - detect the state of the system parameters, inputs and outputs.
- Digital devices - control the system.
- Conditioning and interfacing circuits - provide connections between the control circuits and the input/output devices.
- Graphical displays - provide visual feedback to users (Furukawa, 2004).

2.2 ROBOTS

A robot is a very good example of a mechatronic system. Robots alone are hardly ever useful. They are used together with other devices, peripherals, and other manufacturing machines. They are generally integrated into a system, which as a whole is designed to perform a task or do an operation. Although there are different standards for what are considered to be robots, a definition will be given in the next section.

2.2.1 Definition

A reprogrammable, multifunctional manipulator designed to move material, parts, tools, or specialized devices through various programmed motions for the performance of a variety of tasks.

A more stimulating definition can be found in Webster. According to Webster a robot is: An automatic device that performs functions normally ascribed to humans, or a machine in the form of a human.

2.2.2 History

The acclaimed Czech playwright Karel Capek (1890-1938) made the first use of the word 'robot', from the Czech word for forced labor or serf.

In R.U.R. (Rossum's Universal Robots), Capek raises a paradise, where the machines initially bring so many benefits, but, in the end, an equal amount of disease in the form of social unrest and unemployment.

2.2.3 Robot Components

A robot, as a system, consists of the following components, which are integrated together to form a whole.

2.2.3.1 Manipulator

This is the main body of the robot, and consists of links, joints, and other structural elements of the robot.

2.2.3.2 End Effector

This part is connected to the last joint (hand) of a manipulator, which generally handles objects, makes connection to the other machines, or performs the required tasks.

2.2.3.3 Actuators

Actuators are the "muscles" of the manipulators. Common types of actuators are servomotors, stepper motors, pneumatic cylinders, and hydraulic cylinders. There are also other actuators that are more novel, and are used in specific situations.

2.2.3.4 Sensors

Sensors are used to collect information about the internal conditions of a robot, or to communicate with outside environment. As in humans, the robot controller needs to know where each link of the robot is in order to know the robot's configuration. Robots are often equipped with external sensory devices such as vision system, touch and tactile sensors, speech synthesizers, etc., which enable a robot to communicate with outside world.

2.2.3.5 Controller

The controller is rather similar to cerebellum, and although it does not have the power of brain, it still controls motions. The controller receives its data from the computer, controls the motions of the actuators, and coordinates the motions with sensory feedback information.

2.2.3.6 Processor

The processor is the brain of a robot. It calculates the motions of the robot's joints, determines how much and how fast each joint must move to achieve the desired location and speeds, and oversees the coordinated actions of the controller and the sensors.

2.2.3.7 Software

There are perhaps three groups of software that are used in a robot. One is the operating system, which operates the computer. The second one is the robotic software, which calculates the necessary motion of each joint based on the kinematic equations of the robot. The third group is the collection of routines and application programs that are developed in order to use the peripheral devices of the robots, such as vision routines, or to perform specific tasks.

2.2.4 Types of Robots

Robot configurations generally follow the coordinate frames with which they are defined, and the following configurations are common for positioning the hand of the robot.

2.2.4.1 Cartesian/Rectangular Robot

These robots are made of three linear prismatic joints that position the end effector, which are usually followed by additional revolute joints that orientate the end effector. In this thesis, the Cartesian Robotic System is opted for its simplicity in Matlab simulations. It has basically three axes, and rigid structure. It is also easy to be visualized, and to program.

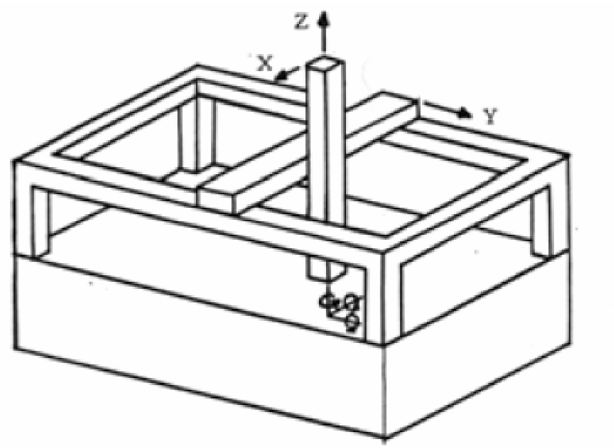

Figure 2.2: Cartesian Robot Configuration

2.2.4.2 Cylindrical Robot

Cylindrical coordinate robots have two prismatic joints and one revolute joint for positioning the part, plus revolute joints for orienting the part.

2.2.4.3 Spherical Robot

Spherical coordinate robots follow a spherical coordinate system, which has one prismatic and two revolute joints for positioning the part, plus additional revolute joints for orientation.

2.2.4.4 Articulated Robot

An articulated robot's joints are all revolute, similar to a human's arm. It is perhaps the most common configuration in industrial robots.

2.2.4.5 SCARA Robot

SCARA robots have two revolute joints that are parallel, and allow the robot to move in a horizontal plane, plus an additional prismatic joint that move vertically. SCARA robots are very common in assembly operations. Their specific characteristic is that they move compliant in the *x-y* plane.

2. 3 LITERATURE SURVEY ON ROBOTICS

In recent years, computer networks have reproduced quickly. In our daily lives, they have become an important tool. In this context, a paper on "Controlling Mobile Robots in Distributed Intelligent Sensor Network" has been published by Lee and Hoshimoto in 2003. They suggest that mobile robots need sufficient sensory information on the environment in order to navigate. In their study, they controlled a mobile robotic system with distributed intelligent sensor network. They monitored all areas by sensor devices, and connected them with other sensor devices throughout the network. As a result, the mobile robots are able to complete tasks simply by following orders from the sensor devices in the networked environment (Lee and Hoshimoto, 2003).

Recently, it has been observed that many researchers are interested in internet-based robotics because of the wide-use of internet. The TeleGarden system robot arm control system through a Web browser was designed and Mars Pathfinder were developed. With these developments, Han, S. Kim, Lee, and J. Kim published a paper. This paper describes the implementation of an internet-based personal robot with new direct internet control architecture. The system is insensitive to the inherent internet time delay. Simulations and experimental results in the real internet environment exhibit the effectiveness and applicability of the internet-based personal robot (Han et al, 2001).

One of the chances to exploit advanced communications is in the context of human/machine interfacing with intelligent systems in both on Earth and in space. A paper about Remote Human/Machine Control was prepared by Liberatore and Newman for NASA. They presented that "Intelligent systems, such as robots, are rapidly becoming more competent. However, fully autonomous capability is not on the foreseeable horizon. The simultaneous explosive growth in communications technology offers the opportunity to gain immediate benefits from semi-autonomous systems through shared human/machine control." Obtaining effective network-based human/machine control would yield benefits in manufacturing, in domestic automation, and in space exploration, (Liberatore and Newman, 2002).

CHAPTER 3

INTELLIGENT CONTROL

AN OVERVIEW OF TECHNIQUES

3.1 INTRODUCTION

Intelligent control depicts the training where control methods are developed that try to copy important characteristics of human intelligence. These characteristics include adaptation and learning, planning under large uncertainty and deal with large amounts of data. Today, the area of intelligent control inclined to surround everything that is not characterized as conventional control; it has, however, shifting boundaries and what is called “intelligent control” today will apparently be called “control” tomorrow. The main trouble in specifying what is meant by the term Intelligent control comes from the fact that there is no agreement about the definition of human intelligence and intelligent behaviour. The centuries old discussion of what constitutes intelligence is still continuing amid educators, psychologists, computer scientists and engineers.

Intelligent control has a number of areas having a relationship with. Intelligent control is dealing with or combining different branches of learning as it combines and extends theories and methods from areas such as control, computer science and operations research. Intelligent control methodologies are being applied to robotics and automation, communications, manufacturing, traffic control, to indicate but a few application areas. It uses theories from mathematics and search for motivation and ideas from biological systems. Neural networks, fuzzy control, genetic algorithms, planning systems, expert systems, hybrid systems are all areas where related work is taking place. The areas of computer science, and in particular artificial intelligence, provide knowledge representation of ideas, methodologies and tools such as semantic networks, frames, reasoning techniques and computer languages such as prolog. Concepts and algorithms developed in the areas of adaptive control and machine learning help intelligent controllers to adapt and learn. Progresses in sensors, actuators, computation technology and communication networks help provide the necessary infrastructure for carrying out intelligent control hardware (Antsaklis, 1997).

3.2 INTELLIGENT CONTROL TECHNIQUES

In this section, a brief overview of fuzzy logic control as an example of intelligent control techniques is provided. The objective here is not to present a comprehensive treatment. We only consider some studies and basics to give an insight.

3.2.1 Fuzzy Logic

3.2.1.1 Introduction

Fuzzy logic is a set of mathematical principles for knowledge representation based on degrees of membership. In a world increasingly controlled by computers with their absolute “1” or “0” and “on” or “off” concepts, a term like fuzzy logic proposes inaccuracy or imprecision. Even Webster’s dictionary defines “fuzzy” as:

fuzz·y (–e) adj. 2. not clear, distinct, or precise; blurred

Fuzzy logic can address complex control problems, such as robotic arm movement, chemical or manufacturing process control, antiskid braking systems, or automobile transmission control with more precision and accuracy, in many cases, than traditional control techniques have (Adcock, 1993).

3.2.1.2 History, Research and Applications

Fuzzy logic was devised in 1965 by Professor Lotfi Zadeh. He published his famous paper “Fuzzy Sets”. In this publication, a new concept for applying natural language terms was introduced, and this term named as Fuzzy Logic. And also in 1973, Lotfi Zadeh published his second most powerful paper. This paper outlined a new approach to the analysis of complex systems, in which Zadeh suggested capturing human knowledge in fuzzy rules.

Japanese engineers immediately realized that fuzzy controllers were very easy to design, and worked very well for many problems, with their ability to perceive sensation to new technology. Because fuzzy control does not need a mathematical model of the process, it could be applied to many systems which are conventional control theory could not be used due to absence of mathematical models. Sugeno began to create Japan's first fuzzy application control of a Fuji Electric water purification plant in 1980. And also in 1983, he began to pioneer a work on a fuzzy robot, a self parking car that was controlled by calling out commands (Sugeno and Nishida, 1985).

Research regarding suitability of various implications in fuzzy model has been managed by Cao and Kandel. They constructed fuzzy model using 36 fuzzy implications and two different interpretations of connective for using linguistic descriptions of a DC series motor, and then developed a root-mean-square error (Cao and Kandel, 1989).

In July 1987, the Second Annual International Fuzzy Systems Association Conference was held in Japan. At this conference, Hirota displayed a fuzzy robot arm that played two-dimensional Ping-Pong in real time (Hirota et al, 1989).

For the last 20 years, Fuzzy Control Theory has emerged as a fruitful approach to a wide variety of control problems.

During the Second AIFAC meeting in Tokyo, Takeshi Yamakawa demonstrated a fuzzy system that balanced an inverted pendulum. Spectators were influenced with this demonstration. In his later experiments, Yamakawa mounted a wine glass containing water, or a live mouse on top of the pendulum; in both cases, the system maintained stability (Yamakawa, 1990).

Coleman and Godbole made a comparison between Fuzzy Logic, Proportional Integrated Derivative (PID) and Sliding Mode Control. They concluded that “Fuzzy

Logic is a very robust control tool for a control engineer" (Coleman and Godbole, 1996).

The success of fuzzy theory systems in Japan surprised the researchers in Europe and in the United States. Some still criticize fuzzy theory, but many others have been changing their minds, and giving fuzzy theory a chance to be taken critically.

In the late 1980s, and early 1990s, from a theoretical point of view, fuzzy systems and control advanced rapidly. Solid progress was made on some fundamental problems in fuzzy systems and control. For example, in order for determining membership functions in a systematic manner, neural network techniques were used, and rigorous stability analysis of fuzzy control systems has appeared.

Fuzzy systems have been applied to a wide variety of fields, the most significant applications have concentrated on control problems. Now, some milestones of how fuzzy systems were used in a number of consumer products and industrial systems will be given next.

Fuzzy Control of a Cement Kiln: Holmblad and ∅stergaard of Denmark developed a fuzzy system to control the cement kiln in the late 1970s. In their system, fuzzy system had four inputs and two outputs.

Fuzzy Washing Machines: The fuzzy washing machines were the first major buyer products to use fuzzy systems. They were produced by Matsushita Electric Industrial Company in Japan around 1990.

Fuzzy Control of Subway Train: The Sendai subway in Japan is the most significant application of fuzzy systems. The train runs along very smoothly , and the fuzzy control system considers four performance criteria, which are safety, rigid comfort, traceability to target speed, and accuracy of stopping gap.

Digital Image Stabilizer: Matsushita introduced what is called a digital image stabilizer, which is based on fuzzy systems, and stabilizes the picture when the hand is shaking.

Fuzzy Systems in Car: Nissan has patented a fuzzy automatic transmission that saves fuel by 12 to 17 percent.

3.2.1.3 The Traditional Approach

To illustrate the difference between fuzzy logic and the traditional approach, here is an example. Fuzzy logic uses the continuum of values between 0 (completely false) and 1 (completely true). Instead of just black and white, it employs the spectrum of colors, accepting that things can be partly true and partly false at the same time as seen in Figure 3.1.

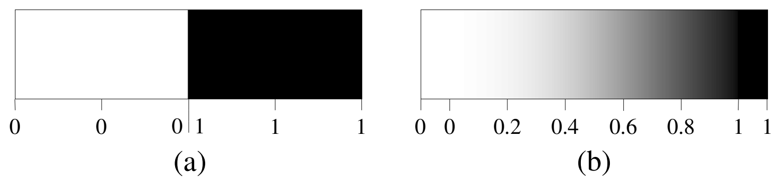

Figure 3.1: Difference between Boolean (crisp) and Fuzzy Logic: (a) Boolean Logic, (b) Fuzzy Logic

3.2.1.4 Fuzzy Logic Control

The term controller is generally defined as a mechanism used to guide or regulate the operation of a machine, apparatus or constellations of machines and apparatus.

Fuzzy Logic Control is a control law that is described by a knowledge-base consisting of IF...THEN rules. It implies that the control algorithm is a knowledge-based algorithm.

The design of a fuzzy logic controller is carried out in three stages (see Figure 3.2):

1. Fuzzification (transformation of crisp input to fuzzy value).
2. Inference engine.
3. Defuzzification (transformation of fuzzy output to crisp output).

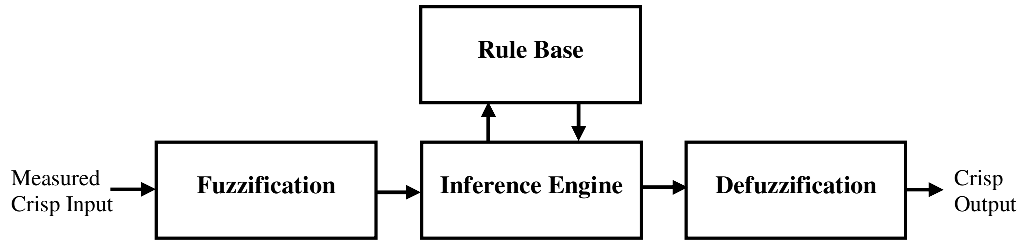

Figure 3.2: Fuzzy Logic Phases

Fuzzification

Fuzzification is the process of decomposing a system input and/or output into one or more fuzzy sets. Many types of curves can be used, but triangular or trapezoidal shaped membership functions are the most common ones because they are easier to represent in embedded controllers.

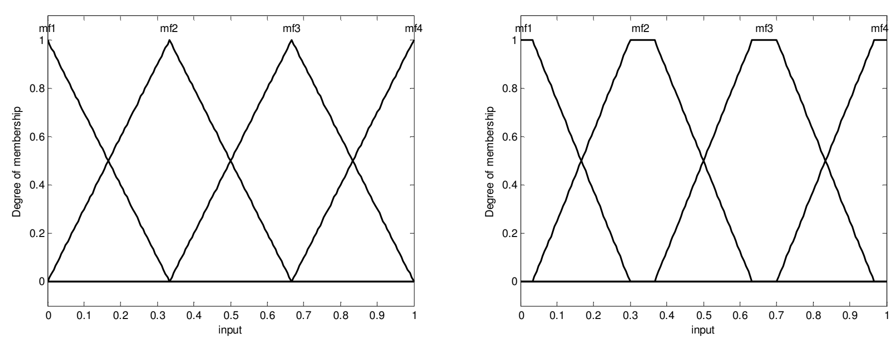

Figure 3.3: Triangular and Trapezoidal membership functions.

The Figure 3.3 shows a system of fuzzy sets for an input with trapezoidal and triangular membership functions. Each fuzzy set spans a region of input (or output) value graphed with the membership. Any particular input is interpreted from this fuzzy set and a degree of membership is interpreted. The membership functions should overlap to allow smooth mapping of the system. The process of fuzzification allows the system inputs and outputs to be expressed in linguistic terms so that rules can be applied in a simple manner to express a complex system.

Inference Engine

The impact of the rules in the rule base under the current input is computed by the inference engine. Aggregation of rules, making the choice of fuzzy reasoning method, and implication is performed by the inference engine. Mamdani implications are the most widely used implications in fuzzy systems and fuzzy control, and IF-THEN rule is interpreted as given in Equation 3.1.

$$\mu_{Q_{MM}}(x, y) = \min\left[\mu_{FP_1}(x), \mu_{FP_2}(y)\right] \tag{3.1}$$

Rule base design includes the choice of the source and the contents of the set of rules. That is

- Modeling Knowledge of the control engineer
- Design of human operator's actions and experience
- Fuzzy design of the controlled process
- Modeling rule base and data learning

Defuzzification

Defuzzification is defined as a mapping from fuzzy set which is the output of the fuzzy inference engine to crisp point. There are several types of defuzzification methods. The most common methods are

- maximum defuzzification method, and
- centroid calculation defuzzification method.

Centroid Calculation Defuzzification Method

The centroid method is the most popular defuzzification technique. Unlike maximum defuzzification method, the centroid method takes into account the entire possibility distribution in calculating its representative point. The defuzzification method is similar to the formula for calculating center of gravity in physics, if we view $\mu_A(x)$ as the density of mass at x. Alternatively, we can view the maximum defuzzification method as calculating a weighted average, where $\mu_A(x)$ serves as the weight for value x. If x is discrete, the defuzzification equation using the centroid calculation method is given in Equation 3.2.

$$A = \frac{\sum_x \mu_A(x) \times x}{\sum_x \mu_A(x)} \tag{3.2}$$

Similarly, if x is continuous, the equation is

$$A = \frac{\int \mu_A(x) x dx}{\int \mu_A(x) dx} \tag{3.3}$$

3.2.1.5 Application of Fuzzy Logic in Robotics

Fuzzy Logic Control systems can be used for both controlling robots, as well as for adding intelligence to applications where other systems may be inadequate, or difficult to use. For example, in one application, fuzzy logic was used to directly control the torque output of a switched reluctance motor by a current modulation scheme. Although fuzzy logic can be used for controlling robots in lieu of classical control systems, there are many other applications where fuzzy logic may perhaps be more appropriate.

During the last four decades, many “model based” control strategies have been proposed by the researchers. Zilouchain and Howard designed a Fuzzy Inference system for the control of a robot manipulator. In their study, Adaptive Neuro Fuzzy Inference System (ANFIS) is utilized for the solution of inverse kinematics as well as the control of a commercial robot manipulator. The special ANFIS controller for each joint generates the demanded control signals to a DC servo motor to move the associated link to the new position (Zilouchain and Howard, 2000).

During the last years, mobile robot navigation has been the foremost study. In 1998 Saffiotti introduced “Fuzzy Logic in Autonomous Robot Navigation: A Case Study”. The development of techniques for autonomous navigation comprises one of the more important courses in the current research on mobile robotics. Saffiotti discusses how fuzzy computation techniques have be used in the SRI International mobile robot Flakey to address some of the difficult issues posed by autonomous navigation. Through experimental works, his techniques have been confirmed in both in-house experiments and public events. The use of fuzzy logic has resulted in smooth motion control, and robust performance control (Saffiotti, 1998).

And also, fuzzy logic or fuzzy set theory has received attention of a number of researchers in the area of power electronics and motion control in recent years. “Robust Position Control of Induction Motor Using Fuzzy Logic Control” has been conducted by Won, Kim and Bose. Their paper describes a vector-controlled induction motor position servo drive where fuzzy control is used to get robustness against parameter variation and load torque disturbance effects (Won et al., 1992).

Through these applications, a robot can become unique, more intelligent, or more useful. For example, consider a mobile robot that is designed for draft tract of land. A fuzzy logic control system can be used in deciding what action to take, depending on the speed of the robot, the terrain, robot’s power, etc. A robot whose end effector must exert a force proportional to two other inputs, say, the size of a part and its weight is another example. In yet another example, a robot maybe used to sort a bag of objects based on their colors according to the colors of the rainbow. In these, and countless other similar examples, fuzzy logic may be the best choice to incorporate the intelligence needed to accomplish the task. In addition, many peripheral devices are added to robots or work with a robot through their own controller. In these cases, a separate microprocessor may be used to control the function of the device independent of the robot controller. When appropriate, fuzzy logic may be incorporated into the microprocessor for its own functions.

CHAPTER 4

DESIGN OF CONTROLLERS

4.1 INTRODUCTION

Three modular controllers will be designed for a mechatronic system given in block diagram in Figure 4.1, which shows actuators, sensors, controllers, mechanical system, and the relationship amongst them.

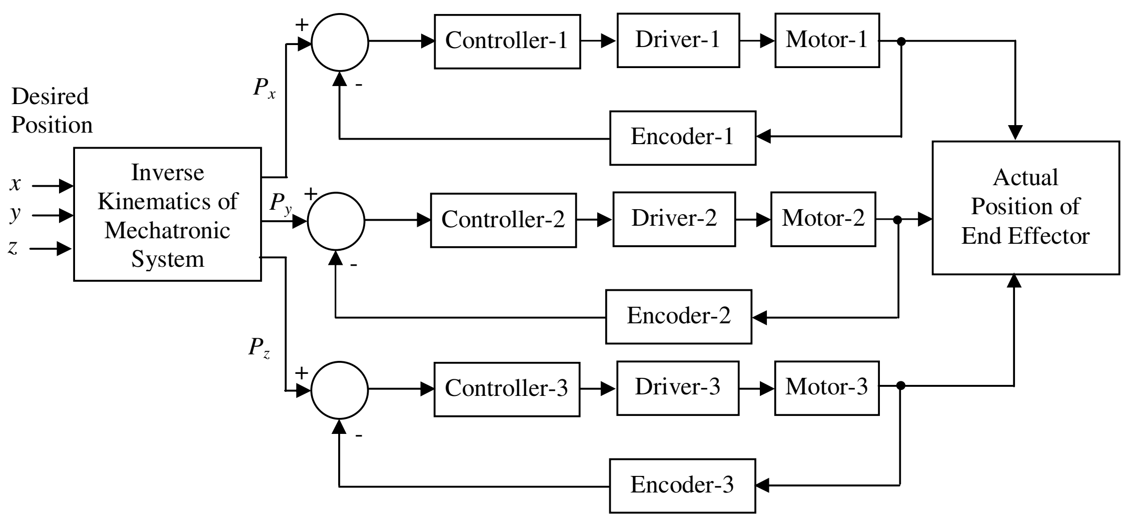

Figure 4.1: Cartesian Robot Control System

The design of a mechatronic system is a specific example of an engineering design. The goal of control engineering is to obtain the configuration, specifications, and identification of the key parameters of a suggested system to meet an actual need.

The first step in the design process consists of establishing the system goals. For example, one may state that our goal is to control the position of a servomotor exactly. The second step is to identify the variables that are desired to be control. The third step is to write the specifications in terms of the accuracy that must reached. This required accuracy of control will then lead to the identification of a sensor to measure the controlled variable.

Designers proceed to configure a system that will result in the desired control performance. This system configuration will normally consist of a sensor, the process under control, an actuator, and a controller. The next step consists of identifying an applicant for the actuator.

Another step is the selection of a controller, which often consists of a summing amplifier that will compare the desired response and the actual response and then forward this error-measurement signal to amplifier.

The final step in the design process is the adjustment of the parameters of the system in order to achieve performance.

4.2 A BRIEF INTRODUCTION TO PID AND FUZZY CONTROLLERS

Conventional proportional-integral-derivative (PID) controllers have been well developed and applied for about half a century. They are widely used for industrial automation and process control today. The main reason is due to their absence of complexity of operation, ease of design, inexpensive maintenance, low cost, and effectiveness for most linear systems. In recent years, motivated by the rapidly developed advanced microelectronics and digital processors, conventional PID controllers have gone through a technological evolution, from pneumatic controllers by means of analog electronics to microprocessors via digital circuits (Tang et al, 2001).

Most PID controllers used in industry are tuned by trial and error method. Even if a more methodical design based on classical control theory is used, it is still limited by the supposition that the plant is linear, or at least a plant that operates in a linear region, and its dynamics do not change over time. When nonlinear effects arise in the system to be controlled, the PID controller may have difficulty in controlling it. These nonlinearities include friction, saturation, backlash, hysteresis, etc. Eventually, all physical systems will display nonlinear, time-varying behavior as inputs grow unbounded and as the systems are operated over extended periods of time. The PID controller may have difficulty in controlling these types of plants whereas a more robust type of controller will provide better performance (Nonnenmarcher and Gao, 2001).

One such type of controller is fuzzy controller. Fuzzy logic control is an intelligent control technique that uses human expert knowledge of the system in a form of the so-called IF-THEN control rules. These rules act on linguistic variables that will have words as its values characterized by fuzzy sets depending on the operating region of the system. This method of control mimics that of human reasoning.

In the following sections, the design of fuzzy logic controller, PID controller and Fuzzy-PID controller will be given.

4.3 DESIGN OF FUZZY LOGIC CONTROLLER

Fuzzy Logic Controllers (FLC) use fuzzy logic as a process of mapping from a given input (crisp numerical value **e**) to an output (control signal, **u**). This process has a basic structure that includes a fuzzifier, an inference engine, a knowledge base (fuzzy rule base), and a deffuzzifier which transforms fuzzy sets into real-valued control signals.

Dealing with fuzzy logic in MATLAB is very easy due to new Fuzzy Logic Toolbox. The Toolbox supplies three categories of tools: Command line functions, Graphical or Interactive tools, and SIMULINK blocks.

- **Command line FIS functions are**

addmf - Add membership function to FIS

addrule - Add rule to FIS.

addvar - Add variable to FIS.

defuzz - Defuzzify membership function.

evalfis - Perform fuzzy inference calculation

evalmf - Generic membership function evaluation.

gensurf - Generate FIS output surface.

getfis - Get fuzzy system properties.

mf2mf - Translate parameters between functions.

mfstrtch - Stretch membership function.

newfis - Create new FIS. parsrule - Parse fuzzy rules

plotfis - Display FIS input-output diagram.

plotmf - Display all membership functions for one variable.

readfis - Load FIS from disk.
rmmf - Remove membership function from FIS
rmvar - Remove variable from FIS.
setfis - Set suzzy system properties.
showfis - Display annotated FIS.
showrule - Display FIS rules
writefis - Save FIS to disk.

- **Graphical User Interface editors (GUI tools) are**

fuzzy - Basic FIS editor.
mfedit - Membership function editor.
ruleedit - Rule editor and parser.
ruleview - Rule viewer and fuzzy inference diagram.
surfview - Output surface viewer.

- **SIMULINK blocks**

Once a fuzzy system is produced using GUI tools or some other methods, it can be directly embedded into SIMULINK using the Fuzzy Logic Controller block shown in Figure 4.2.

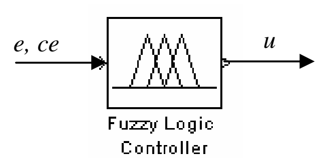

Figure 4.2: Fuzzy Logic Controller Block

Although it is possible to use the Fuzzy Logic Toolbox by working exactly from the command line, in general it is much easier to build up a system graphically, so that GUI tools are commonly used for building, editing, and monitoring Fuzzy Inference Systems.

The process of mapping from a given input to an output using fuzzy logic includes membership functions, fuzzy logic operators, and If-Then rules.

- **Membership Functions**

This Toolbox includes 11 built-in membership function types which are built from several basic functions: These linear functions *(triangular and trapezoidal*), the Gaussian distribution function (*gaussian curves and generalized bell)*, the sigmoid curve, and quadratic and cubic polynomial curves (*Z, S, and Pi curves*).

- **Fuzzy Logic Operators**

In the Fuzzy Logic Toolbox, two built-in **AND operations** are supported: ***Min*** (minimum) and ***prod*** (algebraic product). Two built-in **OR operations** are also supported: ***Max*** (maximum), and the ***probor*** (probabilistic OR, also known as algebraic sum).

Related to **implication method**, two built-in methods are supported, and they are the same functions as used by the AND method, so that, ***Min*** method shorten the output fuzzy set, and ***prod*** scales the output fuzzy set.

Related to **aggregation method**, three built-in methods are supported: ***Max*** (maximum), ***probor*** (probabilistic OR), and ***sum*** (simply the sum of each rule's output set).

Although centroid (center of gravity) calculation is the most popular **defuzzification method**, there are five built-in methods supported: *centroid, bisector, mean of maxima, largest of maxima, and smallest of maxima.*

- **If-Then Rules**

Since rules can be prepared in three different formats (*verbose, symbolic,* and *indexed*), the verbose format makes the system easier to explain.

Each rule has a *weight, i.e. a degree of reliability,* (a number determined between 0 and 1) which is applied to the number given by the antecedent. Generally, this weight is 1, so it has no effect at all on the implication process.

4.3.1 Simulink Model of the System

Figure 4.3 illustrates the simulink model for a Fuzzy Logic controlled position control system. The actual position is compared with the reference position to produce an error (**e**) signal, and to determine the change in error (**ce**). These two parameters are used as two-inputs to the FLC. The output of the FLC is the control action needed to bring the actual position to the desired value. The gain may be varied from 1 to 100 which characterizes the performance of the Fuzzy Logic Controller. The saturation block imposes upper and lower bounds on a signal. When the input signal is within the range specified by the lower limit and upper limit parameters, the input signal passes through unchanged. When the input signal is outside these bounds, the signal is clipped to the upper or lower bounds.

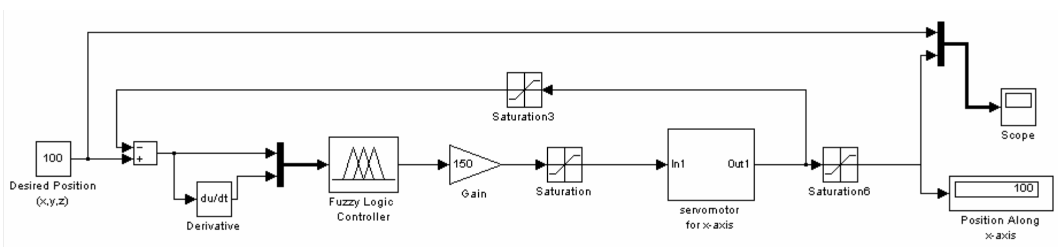

Figure 4.3: Simulink model of a position control system with Fuzzy Logic Controller.

The servomotor subsystem block contains the transfer function of the plant (i.e. the servomotor). This transfer function is the mathematical model of the servomotor, which is given in Equation 4.1, and is of the second-order. The display block is used to show the value of the input, and the scope block display the signals generated during a simulation.

$$G(s) = \frac{s+5}{s^2 + 14s + 40.02} \tag{4.1}$$

4.3.2 Simulation Studies of the Fuzzy Logic Controlled System

The Fuzzy Inference System (FIS) Editor of Figure 4.4 that handles the high-level issues for the system: For example, input and output variables, and their names. The Fuzzy Logic Toolbox does not limit the number of inputs. However, the number of inputs may be limited by the available memory of a computer. If the number of inputs is too large, or the number of membership functions is too big, then it may also be difficult to analyze the FIS using the other GUI tools.

The Membership Function Editor is used to define the shapes of the membership functions connected with each variable.

The Rule Editor is used for constituting of the list of the rules that represents the desired behavior of the system (The MatWorks, 2002).

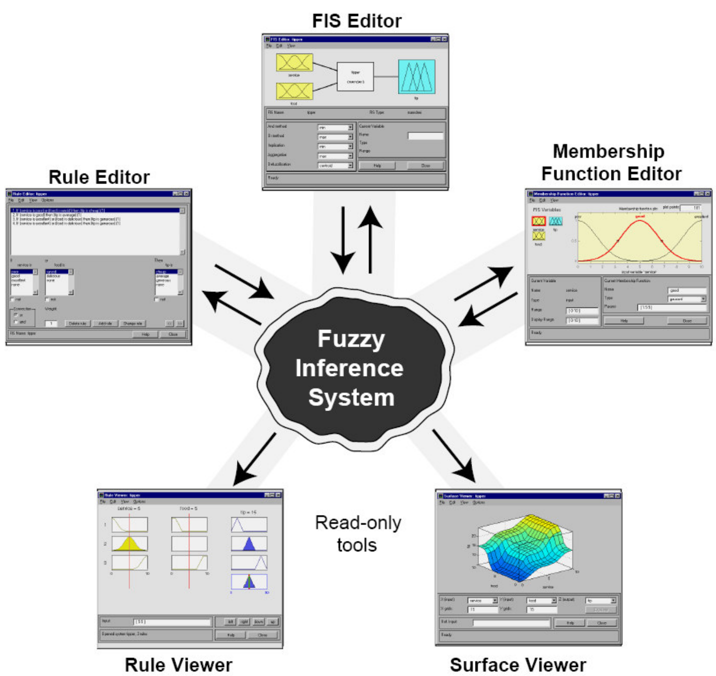

Figure 4.4: Fuzzy Inference System

The Rule Viewer and the Surface Viewer are used for looking at, as opposed to editing, the FIS. They are literally read-only tools. The Rule Viewer is a MATLAB based display of the fuzzy inference diagram shown at the bottom of Figure 4.4. Used as a diagnostic, it can show (for example) which rules are active, or how individual membership function shapes are affecting the results. The Surface Viewer is used to display the dependency of one of the outputs on any one or two of the inputs-that is, it generates and plots an output surface map for the system.

For the best results and observe the changes on the system distinctly, 10 different cases have been determined for the simulation of Fuzzy Logic Controller, and all of them are tested using the model given in Figure 4.3. The results will be given in Chapter 5. Now, these cases are to be examined as follows.

Case-1

For Case-1, FIS was designed as seen in Figures 4.5a, 4.5b, 4.5c, 4.5d, and 4.5e.

It may be seen in Figure 4.5a that Mamdani-type inference, two inputs labeled **e** (error) and **ce** (change in error), and one output labeled **output** were created using the FIS editor.

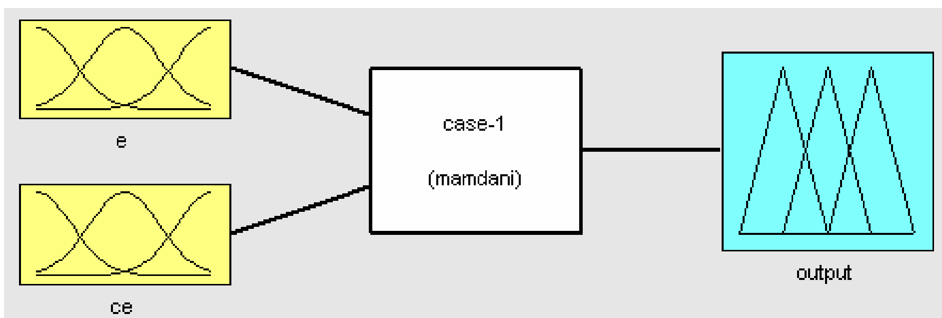

Figure 4.5a: FIS Editor

As mentioned in sub-section of 4.3, the following methods are used for the Fuzzy Logic operations in the fuzzy logic controller block.

“And” operation: *prod* (algebraic product)

Algebraic product is given by

$$t_{ap}(a,b) = ab \tag{4.1}$$

where $a = \mu_A(x)$ and $b = \mu_B(x)$.

“Or” operation: *probor* (probabilistic OR)

Probabilistic OR, also named as Algebraic Sum of the s-norm, is expressed as

$$S_{as}(a,b) = a + b - ab \tag{4.2}$$

Implication method: *prod* (algebraic product, Mamdani implication)

Mamdani implications are the most widely used implications in fuzzy systems and fuzzy control. One of them is given in Equ. 4.3.

$$\mu_{Q_{MP}}(x,y)=\mu_{FP_1}(x)\mu_{FP_2}(y) \tag{4.3}$$

where FP_1 and FP_2 are fuzzy propositions.

Aggregation method: *max* (product inference engine)

In product inference engine, individual-rule based inference with union combination, Mamdani's product implication, algebraic product of all t-norm operators, and max for all the s-norm operators are used. The product inference engine is given in Equation 4.4.

$$\mu_{B'}(y)=\max_{l=1}^{M}\left[\sup_{X\in U}(\mu_{A'}(X)\prod_{i=1}^{n}\mu_{A_i^l}(x_i)\mu_{B^l}(y))\right] \tag{4.4}$$

where A' is given input fuzzy set, $\mu_{A_i^l}(x_i)$ is the membership function of the antecedent part (IF-part), $\mu_{B^l}(y)$ is the membership function of the IF-THEN rule (implication) (Wang, 1997).

Defuzzification method: *centroid* (center of gravity)

The centroid method was explained in Chapter 3, and the formulas of the method were given in Equations 3.2 and 3.3.

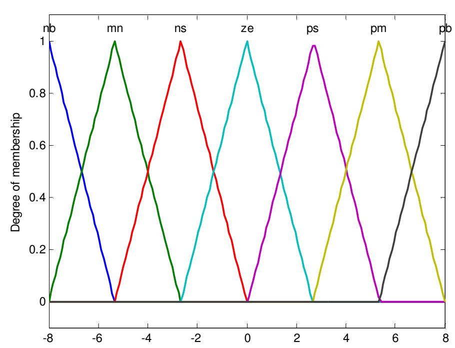

Figure 4.5b: Membership Functions for input-1, **e**

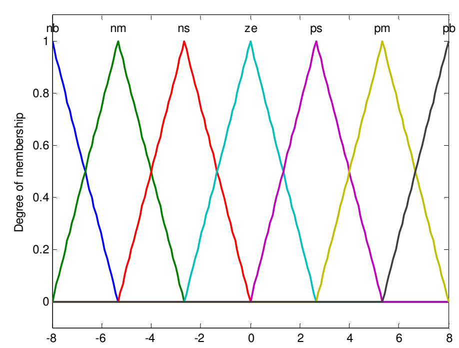

Figure 4.5c: Membership Functions for input-2, **ce**

Triangular membership functions (*trimf*) are used for input-1 (**e**) and input-2 (**ce**). For the sake of simplicity in expressing the control rules, both the error (**e**) and the change in error (c**e**) are scaled in the range of [-8, 8] before applying them to the controller. These variables are quantized into a number of points corresponding to the elements (fuzzy sets) of a universe of discourse, i.e. membership function and values are assigned to the linguistic variables using seven basic fuzzy subsets whose labels are given in the following:

nb : Negative Big

nm : Negative Medium

ns : Negative Small

ze : Zero

ps : Positive Small

pm : Positive Medium

pb : Positive Big

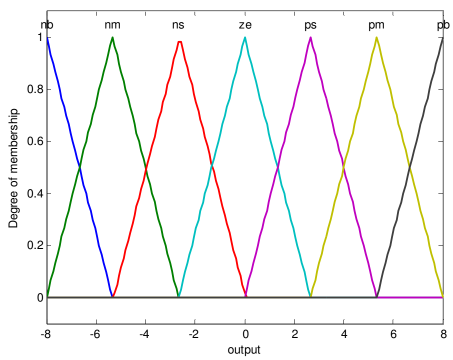

Figure 4.5d: Membership Functions for the output

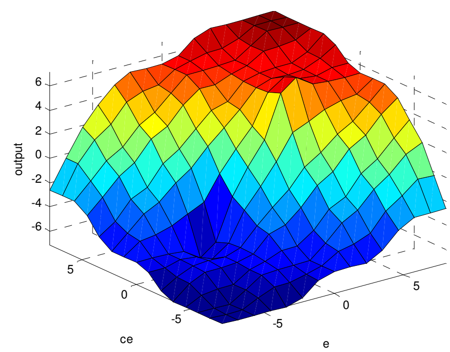

Figure 4.5e: Surface Viewer for case-1

Triangular membership functions (*trimf*) are also used, for the output and the output is scaled in the range of [-8, 8]. The same seven basic fuzzy subsets are used for the output, too.

For Case-1, the rules for the Fuzzy Logic controller are given in Table 4.1.

Table 4.1: Table of rules for Case-1

ce \ e	***nb***	***nm***	***ns***	***ze***	***ps***	***pm***	***pb***
nb	nb	nb	nb	nm	nm	ns	ns
nm	nb	nb	nm	nm	ns	ze	ps
ns	nb	nm	nm	nb	ze	ps	pm
ze	nm	nm	ns	ze	ps	pm	pm
ps	nm	ns	ze	pb	pm	pm	pb
pm	ns	ze	ps	pm	pm	pb	pb
pb	ns	ps	pm	pm	pb	pb	pb

Table 4.1 shows the fuzzy controller rule base in tabular form for case-1, where all the entries of the matrix are the fuzzy sets of error (e), change in error (ce), and control input to the plant (i.e. motor).

An example control rule from Table 4.1 is

IF e is nb AND ce is ze THEN output is nm

In the case of position servo, the above given control rule implies that if the motor position error (e) is Negative Big and change in error (ce) is about zero, then the motor current should be Negative Medium. This means that the present position of motor shaft is very far away from the reference position, and therefore it requires a medium torque to turn the motor shaft to the set point quickly.

In Figure 4.5e, the surface viewer for two-input, one-output system is given as three-dimensional plot. The surface viewer has a special capability that is very helpful in cases with two (or more) inputs and one output. One can actually grab the axes and reposition them to get a different three-dimensional view on the data.

Case-2

For Case-2, FIS was designed as seen in Figures 4.6a, 4.6b, 4.6c, 4.6d, and 4.6e.

It may be seen in Figure 4.6a that Mamdani-type inference, two inputs labeled **e** (error) and **ce** (change in error), and one output labeled **output** were created using the FIS editor.

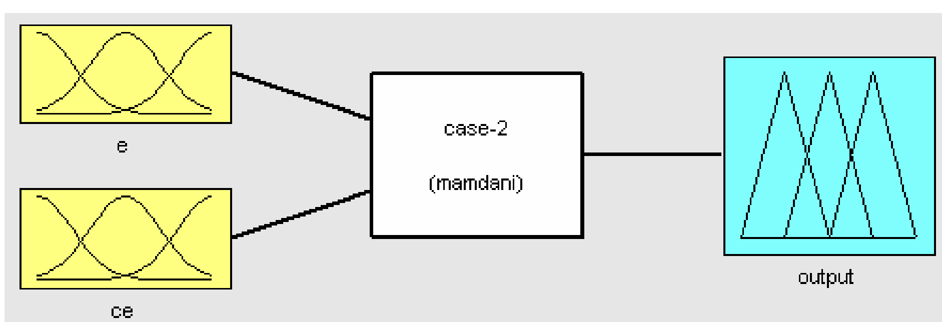

Figure 4.6a: FIS Editor

As mentioned in sub-section of 4.3, the following methods are used for the Fuzzy Logic operations in the fuzzy logic controller block.

"And" operation : *prod* (algebraic product)

"Or" operation : *probor* (probabilistic OR)

Implication method : *prod* (algebraic product, Mamdani implication)

Aggregation method : *max* (product inference engine)

Defuzzification method : *centroid* (center of gravity)

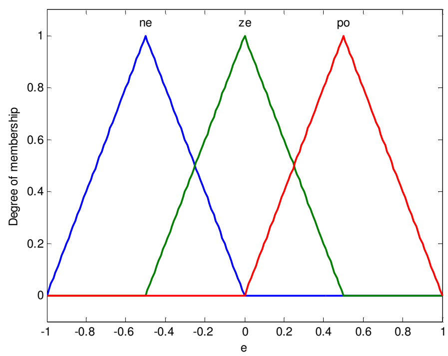

Figure 4.6b: Membership Functions for input-1, **e**

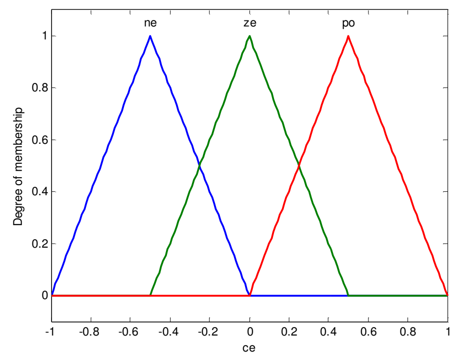

Figure 4.6c: Membership Functions for input-2, **ce**

Triangular membership functions (*trimf*) are used for input-1 (**e**) and input-2 (**ce**). For the sake of simplicity in expressing the control rules, both the error (**e**) and the change in error (**ce**) are scaled in the range of [-1, 1] before applying them to the controller. These variables are quantized into a number of points corresponding to the elements (fuzzy sets) of a universe of discourse, i.e. membership function and values are assigned to the linguistic variables using three basic fuzzy subsets whose labels are given in the following.

ne : Negative

ze : Zero

po : Positive

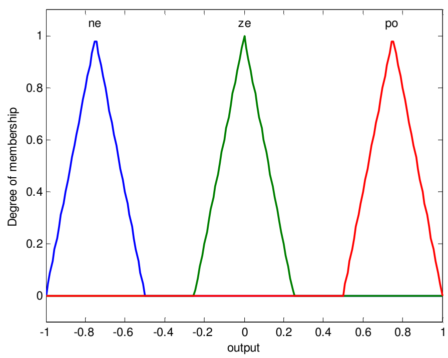

Figure 4.6d: Membership Functions for the output

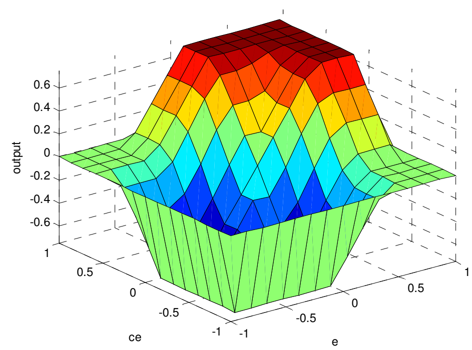

Figure 4.6e: Surface Viewer for case-2

Triangular membership function types (*trimf*) are also used, and the output is scaled in range of [-1, 1]. The same three basic fuzzy subsets are used for the output, too.

For Case-2, the rules for the Fuzzy Logic controller are given in Table 4.2;

Table 4.2: Table of rules for Case-2

ce \ e	***ne***	***ze***	***po***
ne	ne	ne	ze
ze	ne	ze	po
po	ze	po	po

Table 4.2 shows the fuzzy controller rule base in tabular form for case-2, where all the entries of the matrix are the fuzzy sets of error (e), change in error (ce), and control input to the plant (i.e. motor). In Figure 4.6e, the surface viewer for two-input, one-output systems is represented in three-dimensional plot.

Case-3

For Case-3, FIS was designed as seen in Figures 4.7a, 4.7b, 4.7c, 4.7d, and 4.7e.

It may be seen in Figure 4.7a that Mamdani-type inference, two inputs labeled **e** (error) and **ce** (change in error), and one output labeled **output** were created using the FIS editor.

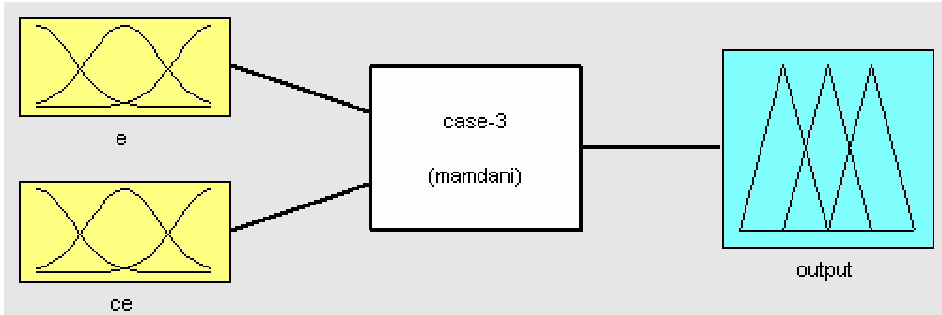

Figure 4.7a: FIS Editor

As mentioned in sub-section of 4.3, the following methods are used for the Fuzzy Logic operations in the fuzzy logic controller block.

"And" operation : *min* (minimum)

"Or" operation : *max* (maximum)

Implication method : *min* (minimum)

Aggregation method : *max* (product inference engine)

Defuzzification method : *centroid* (center of gravity)

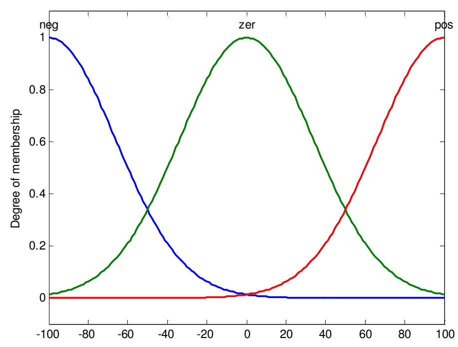

Figure 4.7b: Membership Functions for input-1, **e**

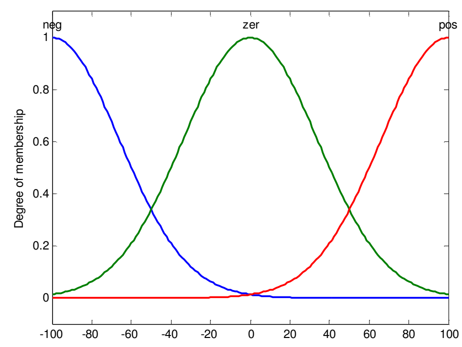

Figure 4.7c: Membership Functions for input-2, **ce**

Gaussian distribution functions (*gaussian curves and generalized bell)* (*gaussmf*) are used for input-1 (**e**) and input-2 (**ce**). For the sake of simplicity in expressing the control rules, both the error (**e**) and change in error (**ce**) are scaled in the range of [-100, 100] before applying them to the controller. These variables are quantized into a number of points corresponding to the elements (fuzzy sets) of a universe of discourse, i.e. membership function and values are assigned to the linguistic variables using three basic fuzzy subsets whose labels are given in the following.

neg : Negative

zer : Zero

pos **:** Positive

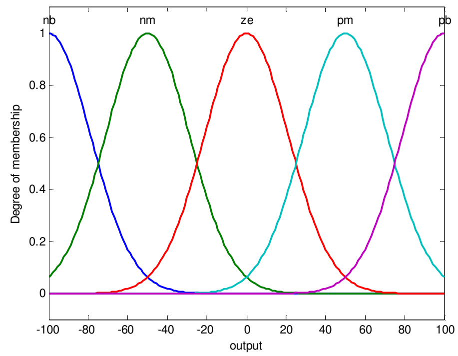

Figure 4.7d: Membership Functions for the output

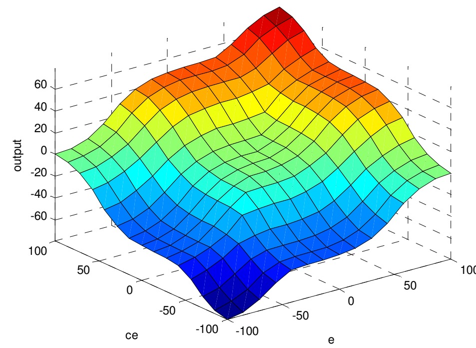

Figure 4.7e: Surface Viewer for case-3

Gaussian distribution functions (*gaussian curves and generalized bell)* (*gaussmf*) are also used, for the output, and the output is scaled in range of [-100, 100]. In the output two fuzzy membership functions **nm** (negative medium) and **ps** (positive medium) are added, so five basic fuzzy subsets are used.

For case-3, the rules for the Fuzzy Logic controller are given in Appendix A. Table A.1 shows the fuzzy controller rule base in tabular form for Case-3, where all the entries of the matrix are the fuzzy sets of error (e), change in error (ce), and the control input to the plant (i.e. motor). In Figure 4.7e, the surface viewer for two-input, one-output system is given as three-dimensional plot.

Case-4

For Case-4, FIS was designed as seen in Figures 4.8a, 4.8b, 4.8c, 4.8d, and 4.8e.

It may be seen in Figure 4.8a that Mamdani-type inference, two inputs labeled **e** (error) and **ce** (change in error), and one output labeled **output** were created using the FIS editor.

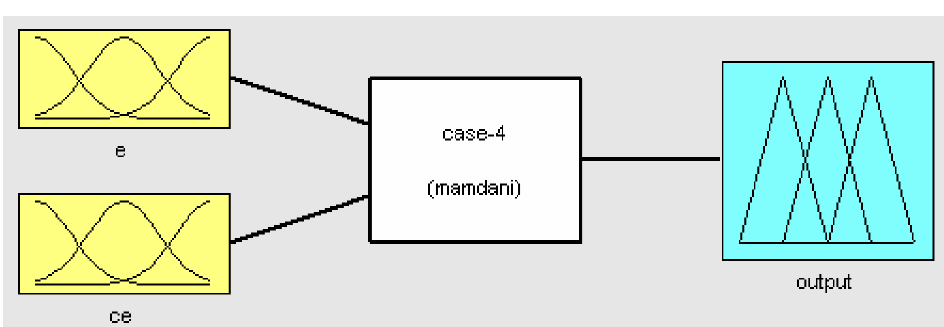

Figure 4.8a: FIS Editor

As mentioned before, the following methods are used for the Fuzzy Logic operations in the fuzzy logic controller block.

"And" operation : *prod* (algebraic product)
"Or" operation : *probor* (probabilistic OR)
Implication method : *prod* (algebraic product, Mamdani implication)
Aggregation method : *max* (product inference engine)
Defuzzification method : *centroid* (center of gravity)

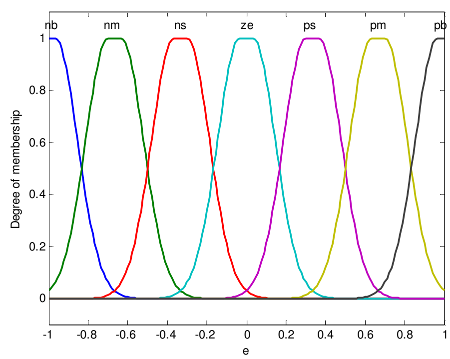

Figure 4.8b: Membership Functions for input-1, **e**

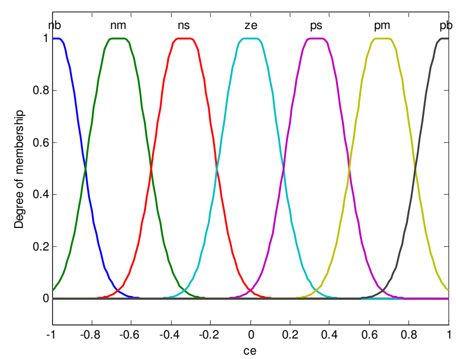

Figure 4.8c: Membership Functions for input-2, **ce**

Gaussian distribution functions (*gaussian curves and generalized bell)* (*gauss2mf*) are used for input-1 (**e**) and input-2 (**ce**). For the sake of simplicity in expressing the control rules, both the error (**e**) and the change in error (**ce**) are scaled in the range of [-1, 1] before applying them to the controller. These variables are quantized into a number of points corresponding to the elements (fuzzy sets) of a universe of discourse, i.e. membership function and values are assigned to the linguistic variables using seven basic fuzzy subsets whose labels are given in the following:

nb : Negative Big
nm : Negative Medium
ns : Negative Small
ze : Zero
ps : Positive Small
pm : Positive Medium

pb **:** Positive Big

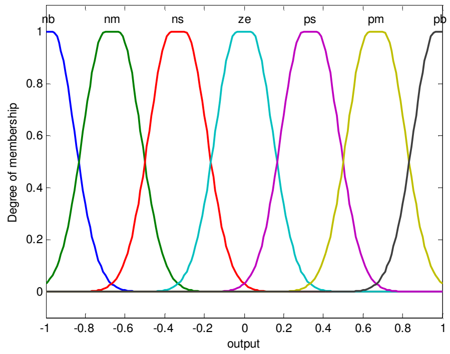

Figure 4.8d: Membership Functions for the output

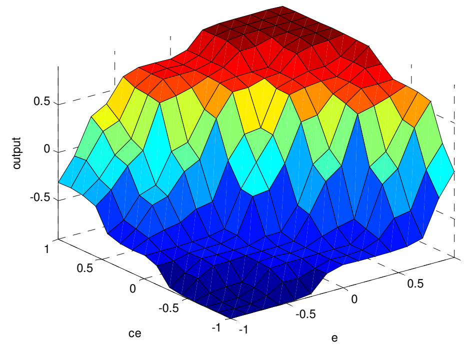

Figure 4.8e: Surface Viewer for case-4

Gaussian distribution functions (*gaussian curves and generalized bell)* (*gauss2mf*) are also used, for the output and output is scaled in range of [-1, 1]. The same seven basic fuzzy subsets are used for the output, too.

For case-4, the rules for the Fuzzy Logic controller are given in Appendix A. Table A.2 shows the fuzzy controller rule base in tabular form for Case-4, where all the entries of the matrix are the fuzzy sets of error (e), change in error (ce), and the control input to the plant (i.e. motor). In Figure 4.8e, the surface viewer for two-input, one-output system is given as three-dimensional plot.

Case-5

For Case-5, FIS was designed as seen in Figures 4.9a, 4.9b, 4.9c, 4.9d, and 4.9e.

It may be seen in Figure 4.9a that Mamdani-type inference, two inputs labeled **e** (error) and **ce** (change in error), and one output labeled **output** were created using the FIS editor.

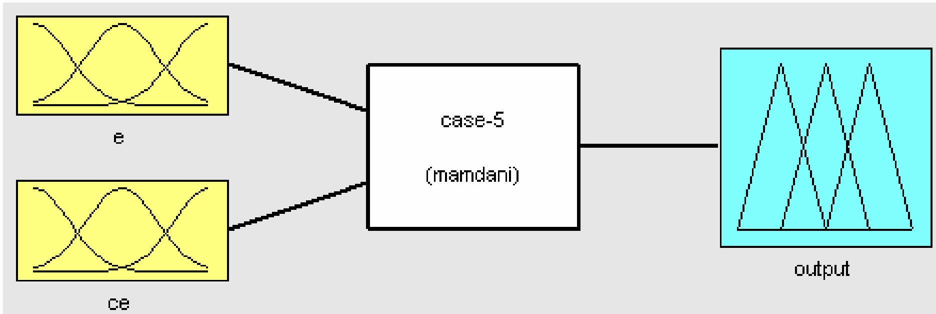

Figure 4.9a: FIS Editor

As mentioned in sub-section of 4.3, the following fuzzy logic operations are used for in the fuzzy logic controller block.

"And" operation	**:** *prod* (algebraic product)
"Or" operation	**:** *probor* (probabilistic OR)
Implication method	**:** *prod* (algebraic product, Mamdani implication)
Aggregation method	**:** *max* product inference engine)
Defuzzification method	**:** *centroid* (center of gravity)

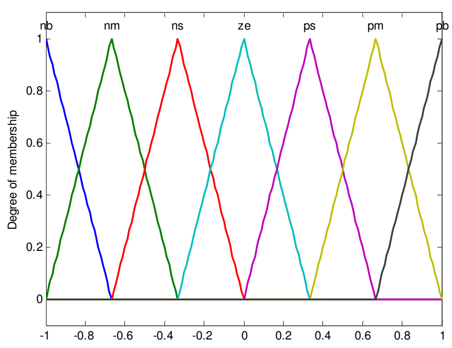

Figure 4.9b: Membership Functions for input-1, **e**

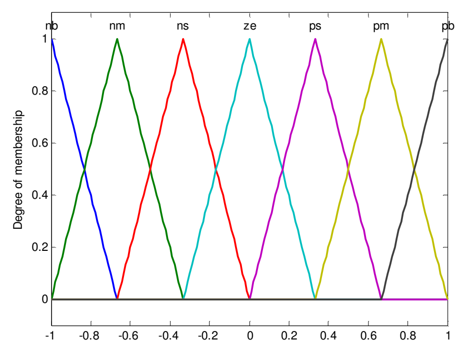

Figure 4.9c: Membership Functions for input-2, **ce**

Triangular membership functions (*trimf*) are used for input-1 (**e**), and input-2 (**ce**). For the sake of simplicity in expressing the control rules, both the error (**e**) and the change in error (**ce**) are scaled in the range of [-1, 1] before applying them to the controller. These variables are quantized into a number of points corresponding to the elements (fuzzy sets) of a universe of discourse, i.e. membership function and values are assigned to the linguistic variables using seven basic fuzzy subsets whose labels are given in the following:

nb : Negative Big
nm : Negative Medium
ns : Negative Small
ze : Zero
ps : Positive Small
pm : Positive Medium
pb : Positive Big

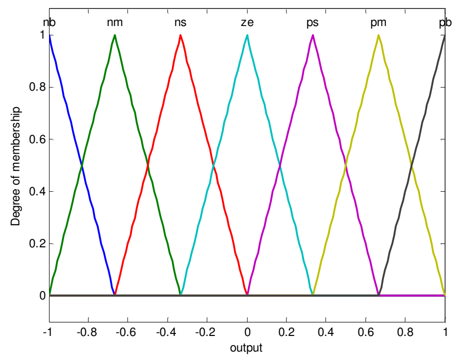

Figure 4.9d: Membership Functions for the output

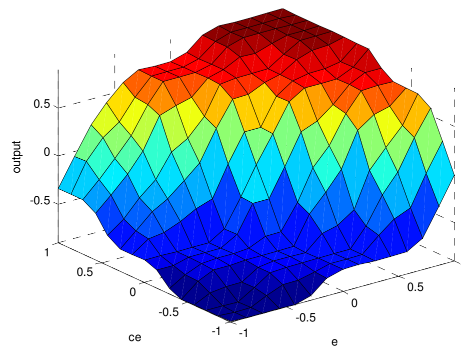

Figure 4.9e: Surface Viewer for case-5

Triangular membership functions (*trimf*) are also used, for the output, and the output is scaled in range of [-1, 1]. The same seven basic fuzzy subsets are used for the output, too.

For Case-5, the rules for the Fuzzy Logic controller are given in Appendix A.

Table A.3 shows the fuzzy controller rule base in tabular form for Case-5, where all the entries of the matrix are the fuzzy sets of error (e), change in error (ce), and control input to the plant (i.e. motor) In Figure 4.9e, the surface viewer for two-input, one-output system is given as three-dimensional plot.

Case-6

For Case-6, FIS was designed as seen in Figures 4.10a, 4.10b, 4.10c, 4.10d, and 4.10e.

It may be seen in Figure 4.10a that Mamdani-type inference, two inputs labeled **e** (error) and **ce** (change in error), and one output labeled **output** were created using the FIS editor.

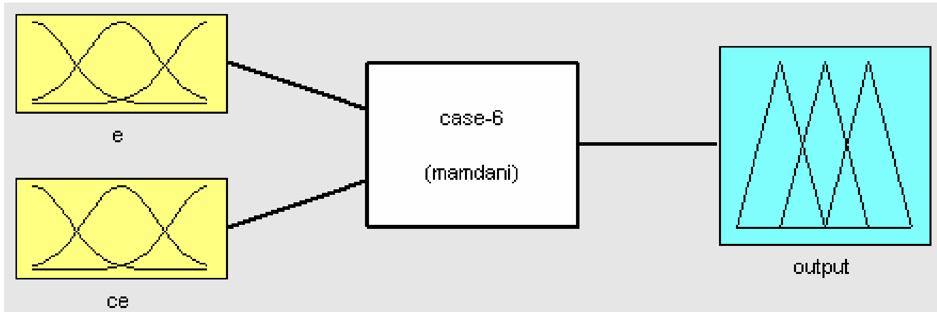

Figure 4.10a: FIS Editor

As mentioned in sub-section of 4.3, the following methods are used for the Fuzzy Logic operations in the fuzzy logic controller block.

"And" operation	**:** *prod* (algebraic product)
"Or" operation	**:** *probor* (probabilistic OR)
Implication method	**:** *prod* (algebraic product, Mamdani implication)
Aggregation method	**:** *max* (product inference engine)
Defuzzification method	**:** *centroid* (center of gravity)

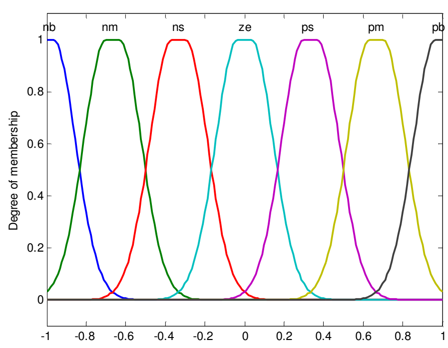

Figure 4.10b: Membership Functions for input-1, **e**

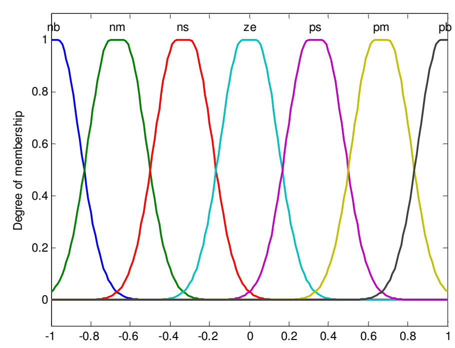

Figure 4.10c: Membership Functions for input-2, **ce**

Gaussian distribution functions (*gaussian curves and generalized bell)* (*gauss2mf*) are used for input-1 (**e**) and input-2 (**ce**). For the sake of simplicity in expressing the control rules, both the error (**e**) and the change in error (**ce**) are scaled in the range of [-1, 1] before applying them to the controller. These variables are quantized into a number of points corresponding to the elements (fuzzy sets) of a universe of discourse, i.e. membership function and values are assigned to the linguistic variables using seven basic fuzzy subsets whose labels are given in the following:

nb : Negative Big
nm : Negative Medium
ns : Negative Small
ze : Zero
ps : Positive Small
pm : Positive Medium
pb : Positive Big

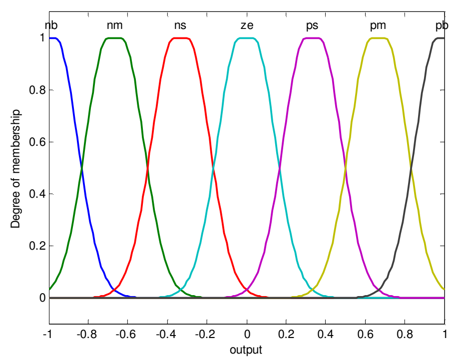

Figure 4.10d: Membership Functions for the output

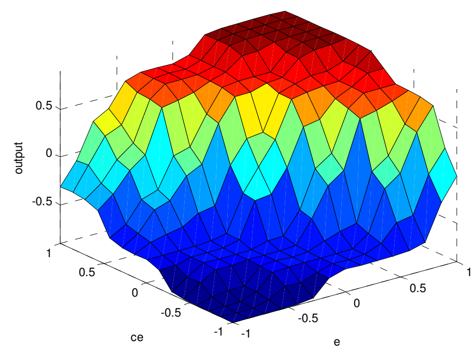

Figure 4.10e: Surface Viewer for case-6

Gaussian distribution functions (*gaussian curves and generalized bell)* (*gauss2mf*) are also used for the output, and the output is scaled in range of [-1, 1]. The same seven basic fuzzy subsets are used for the output, too.

For Case-6, the rules for the Fuzzy Logic controller are given in Appendix A.

Table A.4 shows the fuzzy controller rule base in tabular form for Case-6, where all the entries of the matrix are the fuzzy sets of error (e), change in error (ce), and control input to the plant (i.e. motor) In Figure 4.10e, the surface viewer for two-input, one-output system is given as three-dimensional plot.

Case-7

For Case-7, FIS was designed as seen in Figures 4.11a, 4.11b, 4.11c, 4.11d, and 4.11e.

It may be seen in Figure 4.11a that Mamdani-type inference, two inputs labeled **e** (error) and **ce** (change in error), and one output labeled **output** were created using the FIS editor.

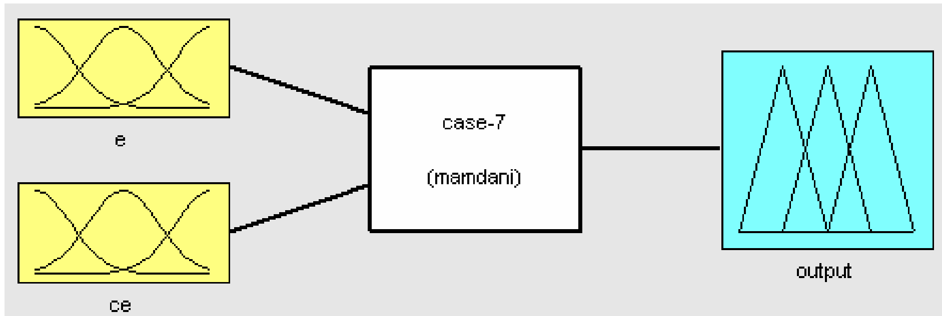

Figure 4.11a: FIS Editor

As mentioned at the beginning of this Chapter, the following fuzzy logic operations are used in the fuzzy logic controller block.

"And" operation	**:** *prod* (algebraic product)
"Or" operation	**:** *probor* (probabilistic OR)
Implication method	**:** *prod* (algebraic product, Mamdani implication)
Aggregation method	**:** *max* (product inference engine)
Defuzzification method	**:** *centroid* (center of gravity)

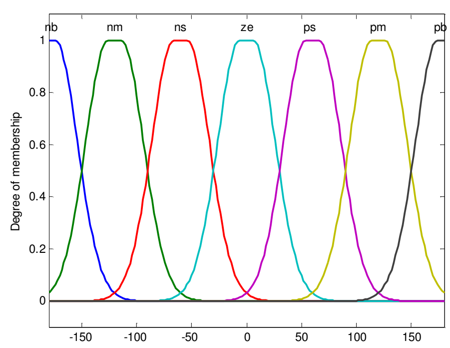

Figure 4.11b: Membership Functions for input-1, **e**

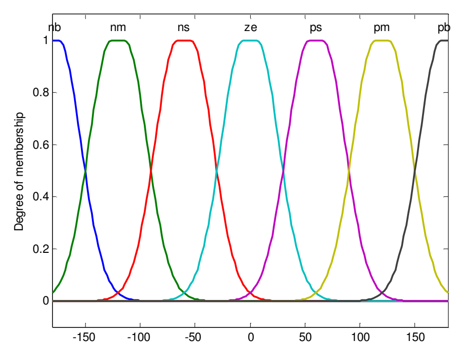

Figure 4.11c: Membership Functions for input-2, **ce**

Gaussian distribution functions (*gaussian curves and generalized bell)* (*gauss2mf*) are used for input-1 (**e**) and input-2 (**ce**). For the sake of simplicity in expressing the control rules, both the error (**e**) and the change in error (**ce**) are scaled in the range of [-180, 180] before applying them to the controller. These variables are quantized into a number of points corresponding to the elements (fuzzy sets) of a universe of discourse, i.e. membership function and values are assigned to the linguistic variables using seven basic fuzzy subsets whose labels are given in the following:

nb : Negative Big
nm : Negative Medium
ns : Negative Small
ze : Zero
ps : Positive Small
pm : Positive Medium
pb : Positive Big

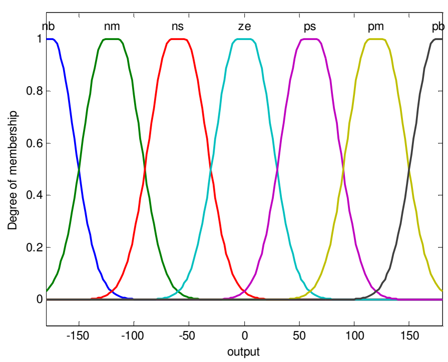

Figure 4.11d: Membership Functions for the output

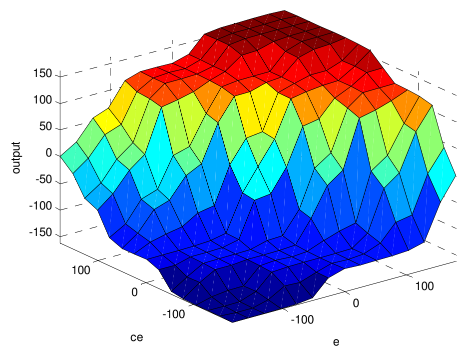

Figure 4.11e: Surface Viewer for case-7

Gaussian distribution functions (*gaussian curves and generalized bell)* (*gauss2mf*) are also used for the output, and the output is scaled in range of [-180, 180]. The same seven basic fuzzy subsets are used for the output, too.

For Case-7, the rules for the Fuzzy Logic controller are given in Appendix A.

Table A.5 expresses the robust control commands for Case-7. The surface viewer for the two-input, one-output system is given in three-dimensional plot in Figure 4.11e.

Case-8

For Case-1, FIS was designed as seen in Figures 4.12a, 4.12b, 4.12c, 4.12d, and 4.12e.

It may be seen in Figure 4.12a that Mamdani-type inference, two inputs labeled **e** (error) and **ce** (change in error), and one output labeled **output** were created using the FIS editor.

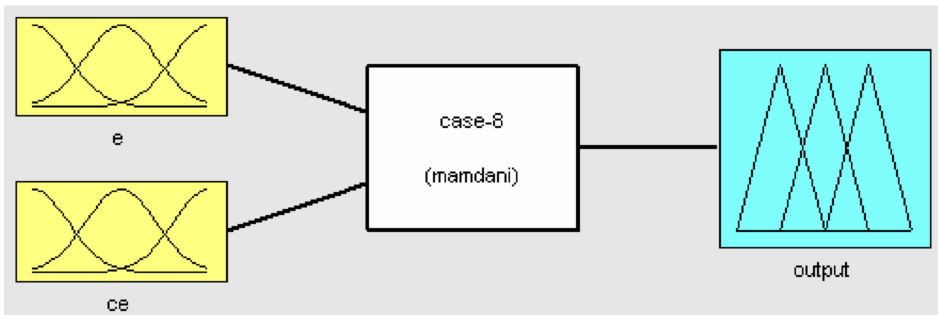

Figure 4.12a: FIS Editor

As mentioned in sub-section of 4.3, the following methods are used for the Fuzzy Logic operations in the fuzzy logic controller block.

"And" operation	**:** *prod* (algebraic product)
"Or" operation	**:** *probor* (probabilistic OR)
Implication method	**:** *prod* (algebraic product, Mamdani implication)
Aggregation method	**:** *max* (product inference engine)
Defuzzification method	**:** *centroid* (center of gravity)

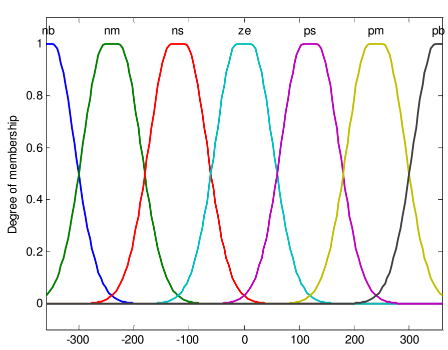

Figure 4.12b: Membership Functions for input-1, **e**

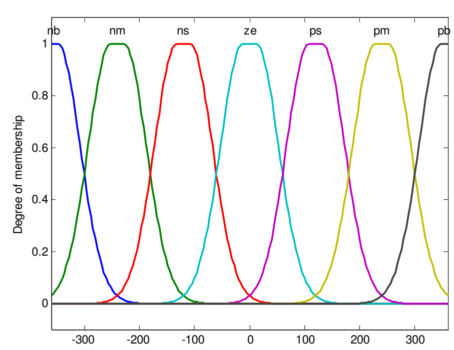

Figure 4.12c: Membership Functions for input-2, **ce**

Gaussian distribution functions (*gaussian curves and generalized bell)* (*gauss2mf*) are used for input-1 (**e**) and input-2 (**ce**). For the sake of simplicity in expressing the control rules, both the error (**e**) and the change in error (**ce**) are scaled in the range of [-360, 360] before applying them to the controller. These variables are quantized into a number of points corresponding to the elements (fuzzy sets) of a universe of discourse, i.e. membership function and values are assigned to the linguistic variables using seven basic fuzzy subsets whose labels are given in the following:

nb : Negative Big
nm : Negative Medium
ns : Negative Small
ze : Zero
ps : Positive Small
pm : Positive Medium
pb : Positive Big

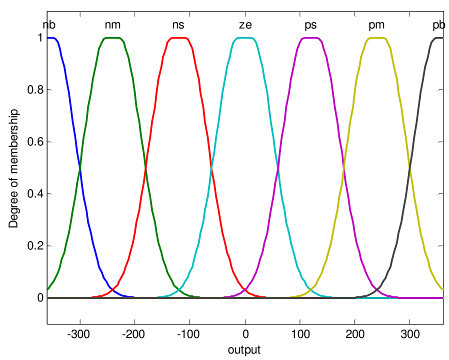

Figure 4.12d: Membership Functions for the output

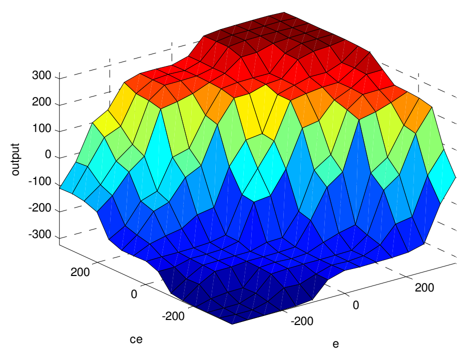

Figure 4.12e: Surface Viewer for case-8

Gaussian distribution functions (*gaussian curves and generalized bell)* (*gauss2mf*) are also used for the output, and the output is scaled in range of [-360, 360]. The same seven basic fuzzy subsets are used for the output, too.

For Case-8, the rules for the Fuzzy Logic controller are given in Appendix A; Table A.6 expresses the robust control commands for Case-8. The surface viewer for the two-input, one-output system is given in three-dimensional plot in Figure 4.12e.

Case-9

For Case-1, FIS was designed as seen in Figures 4.13a, 4.13b, 4.13c, 4.13d, and 4.13e.

It may be seen in Figure 4.13a that Mamdani-type inference, two inputs labeled **e** (error) and **ce** (change in error), and one output labeled **output** were created using the FIS editor.

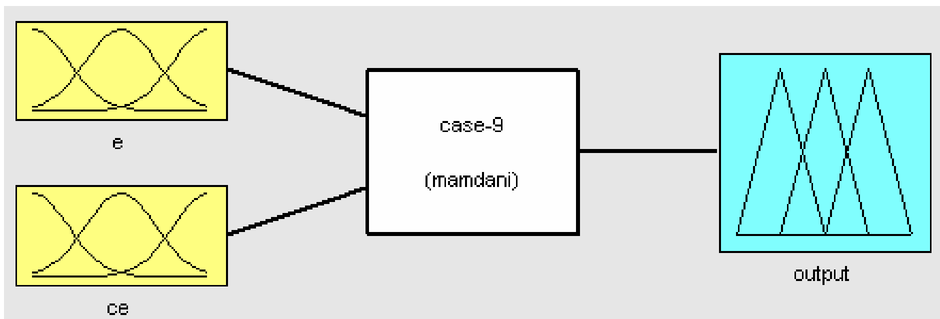

Figure 4.13a: FIS Editor

As mentioned in sub-section of 4.3, the following methods are used for the Fuzzy Logic operations in the fuzzy logic controller block.

"And" operation : *prod* (algebraic product)
"Or" operation : *probor* (probabilistic OR)
Implication method : *prod* (algebraic product, Mamdani implication)
Aggregation method : *max* (product inference engine)
Defuzzification method : *centroid* (center of gravity)

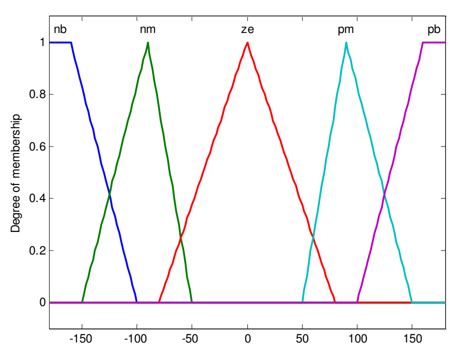

Figure 4.13b: Membership Functions for input-1, **e**

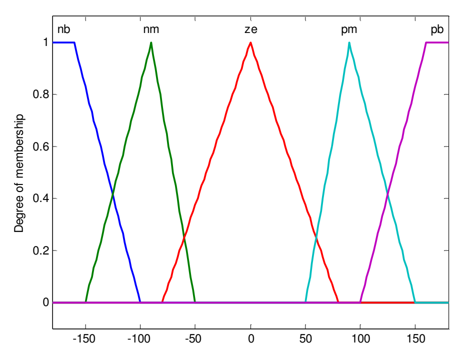

Figure 4.13c: Membership Functions for input-2, **ce**

Triangular membership functions (*trimf*) are used for input-1 (**e**) and input-2 (**ce**). For the sake of simplicity in expressing the control rules, both the error (**e**) and the change in error (**ce**) are scaled in the range of [-180, 180] before applying them to the controller. These variables are quantized into a number of points corresponding to the elements (fuzzy sets) of a universe of discourse, i.e. membership function and values are assigned to the linguistic variables using five basic fuzzy subsets whose labels are given in the following:

nb : Negative Big

nm : Negative Medium

ze : Zero

pm : Positive Medium

pb : Positive Big

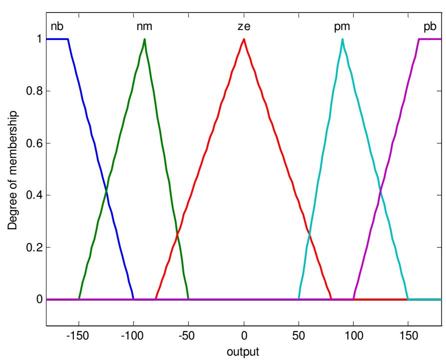

Figure 4.13d: Membership Functions for the output

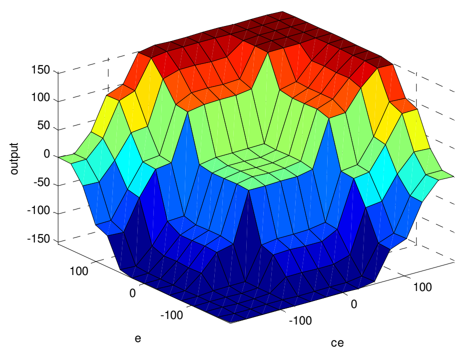

Figure 4.13e: Surface Viewer for case-9

Triangular membership functions (*trimf*) are also used for the output, and the output is scaled in range of [-180, 180] and also two membership functions **ns** (negative small), and **ps** (positive small) are added.

For Case-9, the rules for the Fuzzy Logic controller are given in Appendix A.

Table A.7 expresses the control rules for Case-9. The surface viewer for the two-input, one-output system is given in three-dimensional plot in Figure 4.13e.

Case-10

For Case-1, FIS was designed as seen in Figures 4.14a, 4.14b, 4.14c, 4.14d, and 4.14e.

It may be seen in Figure 4.14a that Mamdani-type inference, two inputs labeled **e** (error) and **ce** (change in error), and one output labeled **output** were created using the FIS editor.

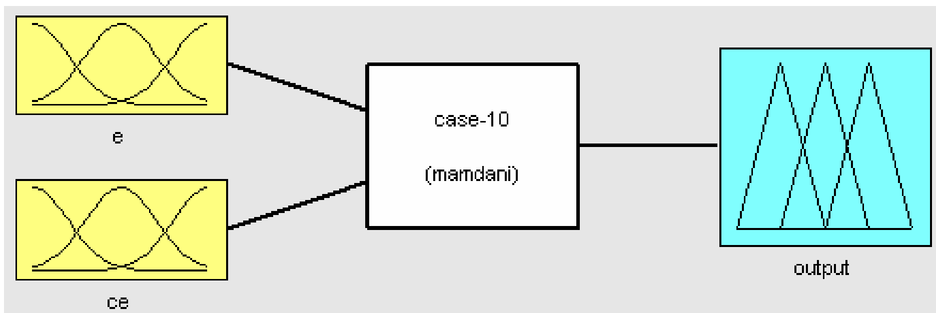

Figure 4.14a: FIS Editor

As mentioned in sub-section of 4.3, the following methods are used for the Fuzzy Logic operations in the fuzzy logic controller block.

"And" operation	**:** *prod* (algebraic product)
"Or" operation	**:** *probor* (probabilistic OR)
Implication method	**:** *prod* (algebraic product, Mamdani implication)
Aggregation method	**:** *max* (product inference engine)
Defuzzification method	**:** *centroid* (center of gravity)

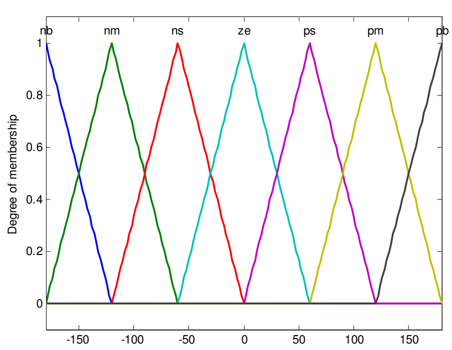

Figure 4.14b: Membership Functions for input-1, **e**

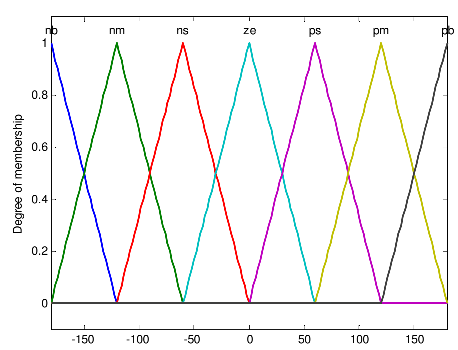

Figure 4.14c: Membership Functions for input-2, **ce**

Triangular membership functions (*trimf*) are used for input-1 (**e**) and input-2 (**ce**). For the sake of simplicity in expressing the control rules, both the error (**e**) and the change in error (**ce**) are scaled in the range of [-180, 180] before applying them to the controller. These variables are quantized into a number of points corresponding to the elements (fuzzy sets) of a universe of discourse, i.e. membership function and values are assigned to the linguistic variables using seven basic fuzzy subsets whose labels are given in the following:

nb : Negative Big
nm : Negative Medium
ns : Negative Small
ze : Zero
ps : Positive Small
pm : Positive Medium
pb : Positive Big

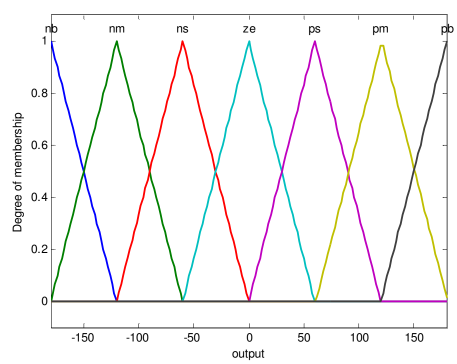

Figure 4.14d: Membership Functions for the output

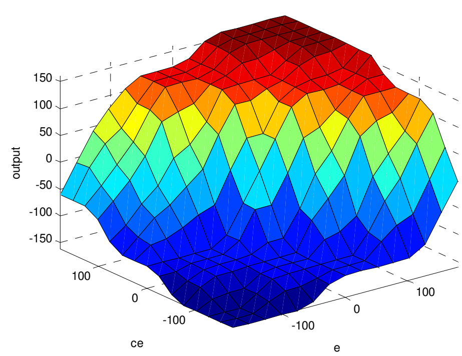

Figure 4.14e: Surface Viewer for case-10

Triangular membership functions (*trimf*) are also used for the output, and the output is scaled in range of [-180, 180]. The same seven basic fuzzy subsets are used for the output, too.

For Case-10, the rules for the Fuzzy Logic controller are given in Appendix A.

Table A.8 expresses the control rules for Case-10. The surface viewer for the TISO (two-input, single-output) system is given in three-dimensional plot in Figure 4.14e.

4.4 DESIGN OF PID CONTROLLER

Among all control design techniques, the PID controller is the most widely used control. Over 85% of all dynamic controllers are of the PID type. There is practically a great variety of types and design methods for the PID controller.

The PID stands for **P**roportiol-**I**ntegral-**D**ifferential control. Each of **P**, **I,** and **D** are terms in a control algorithm, and each has a special purpose. Sometimes certain of the terms are left out because they are not needed in the control design. It is possible to have a PI, PD or just a P control. It is very rare to have an ID control.

The standard PID control configuration is as shown in Figure 4.15. It is also sometimes called the "PID parameter form."

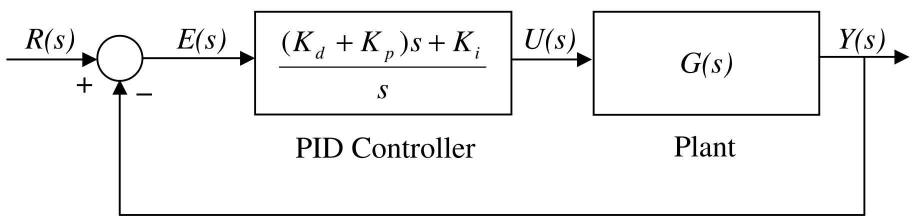

Figure 4.15: PID Controlled System

In this configuration, the control signal *U(s)* is the sum of three terms. Each of these terms is a function of the following error *E(s).* The term K_p shows that this term is proportional to the error. The term K_i/s is an integral term, and the term K_ds is a derivative term. Each term works "independently" of the other.

In this thesis, the PID (three-term) controller is incorporated into the system to provide for the best control.

4.4.1 Simulink Model of the System

Figure 4.16 illustrates the simulink model for the PID controlled position control system. After the tuning process, the best result has been recorded and the all input values have almost no overshoot and follows the input signal nicely.

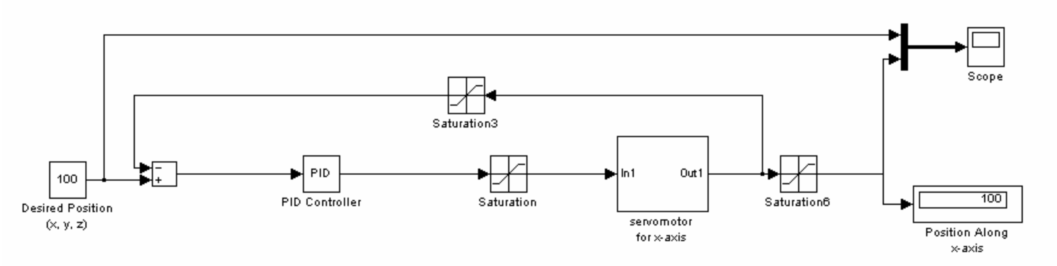

Figure 4.16: Simulink model of a position control system with PID Controller.

4.4.1.1 Tuning PID parameters

A well tuned PID controller has parameters which are adapted to the dynamic properties of the process, so that the control system becomes fast and stable. If the process dynamic properties varies without re-tuning the controller, the control system, gets reduced stability, or becomes more inactive.

The controller parameters may be tuned according to Chien, Hrones and Reswick (CHR) method. It is based on the time parameters of open loop step response. Chien, Hrones and Reswick also gives recommendation for the choice of the type of the controller.

Here, the CHR method is adopted in order to tune PID parameters, and finally trial and error approach is performed for getting the best response of the system.

The formulas given in Eqns. 4.5-4.10 are used for the application of CHR method .

Without Overshoot:

$$K_p = 0.6\frac{T_g}{K_s.T_u} \tag{4.5}$$

$$T_i = T_g \tag{4.6}$$

$$T_d = 0.5T_u \tag{4.7}$$

With Overshoot:

$$K_p = 0.95\frac{T_g}{K_s.T_u} \tag{4.8}$$

$$T_i = 1.35T_g \tag{4.9}$$

$$T_d = 0.47T_u \tag{4.10}$$

where T_u is Time Delay, T_g is Compensation time, $K_s = \frac{x_o}{y_o}$, and T_i, K_p, T_d are the parameters of the PID controller.

A a result of simulations performed, T_u, T_g and K_s were determined as 5, 8, and 0,1249 respectively. The PID parameters of controller are calculated as in the following:

Without Overshoot: K_p=7.6861, T_i=8, and T_d=2.5

With Overshoot: K_p=12.1697, T_i=10.80, and T_d=2.35

These values are used in the simulation, and the results will be given in Chapter 5.

With the trial and error method carried out, various values for PID parameters have been determined to get the best results. The range of values are given below:

P Parameter : [20, 100]
I Parameters : [0, 5]
D Parameters : [0, 5]

4.5 DESIGN OF FUZZY-PID CONTROLLER

Figure 4.17 depicts the simulink model for the Fuzzy-PID controlled and the servomotor position control system. For that model, the parameters which were given in section 4.2 and section 4.3 have been used.

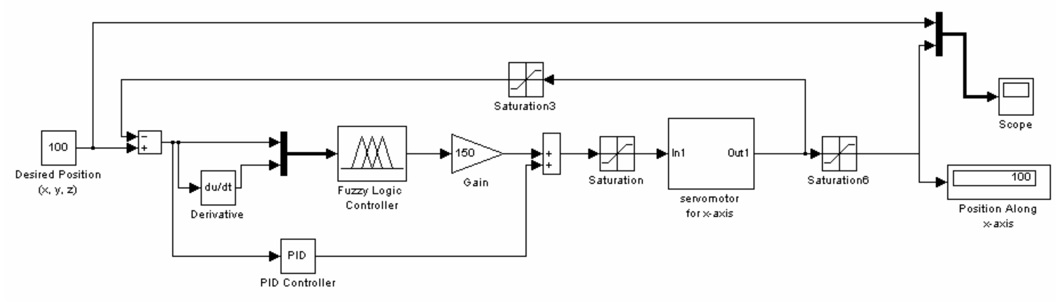

Figure 4.17: Simulink model of a position control system with Fuzzy-PID Controller

These parameters are summarized in Table 4.3. All of the parameters are simulated on the model, and the results are given in Chapter 5.

Table 4.3: Parameters used in Fuzzy-PID Controller.

Situations	Fuzzy Control	P	I	D
Situation-1	Case-7	40	0	2
Situation-2	Case-7	80	0	2
Situation-3	Case-2	40	0	2
Situation-4	Case-2	80	0	2
Situation-5	Case-5	40	0	2
Situation-6	Case-5	80	0	2
Situation-7	Case-9	40	0	2

The PID parameters are verified by trail and error method. The responses with these parameters are carefully examined, and graphically shown in the Chapter 5.

CHAPTER 5

SIMULATION RESULTS

The FLC was first implemented and tested in Simulink simulation, and so were PID, and Fuzzy-PID controllers. Then, all of them are compared with each other.

5.1 FUZZY LOGIC CONTROL

Some of the experimental results are provided in table and graphical form to further demonstrate the effectiveness of the Fuzzy Logic control method. For all 10 cases, the simulation results are given through Figures 5.1 and 5.14 graphically, and the numerical result are given in Table 5.1.

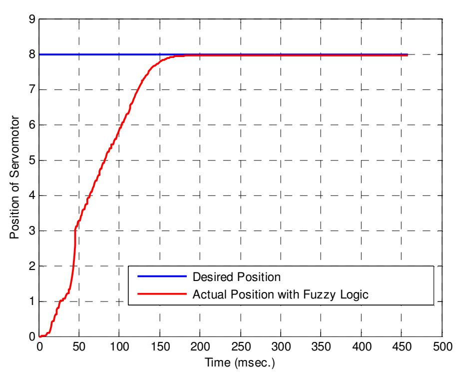

Figure 5.1: Position Control with Fuzzy Logic Controller, Case-1

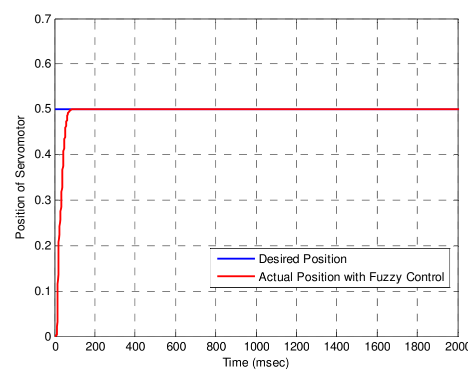

Figure 5.2: Position Control with Fuzzy Logic Controller, Case-2

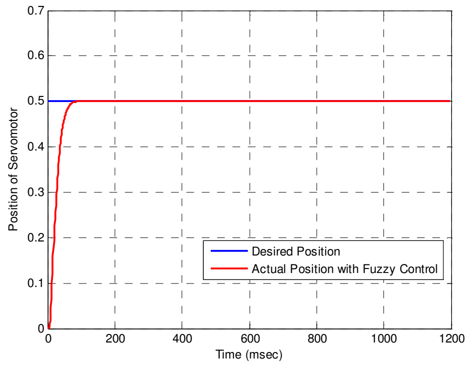

Figure 5.3: Position Control with Fuzzy Logic Controller, Case-3

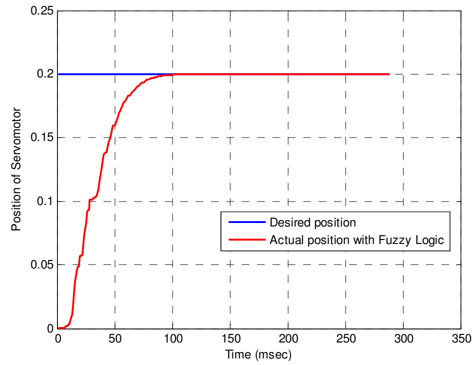

Figure 5.4: Position Control with Fuzzy Logic Controller, Case-4

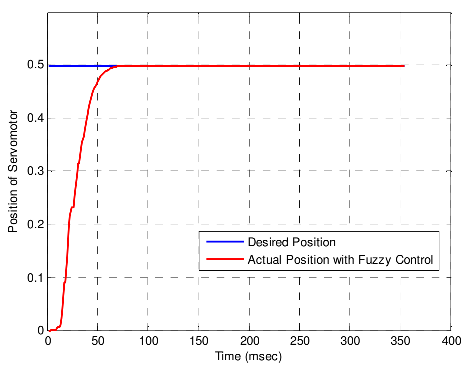

Figure 5.5: Position Control with Fuzzy Logic Controller, Case-5

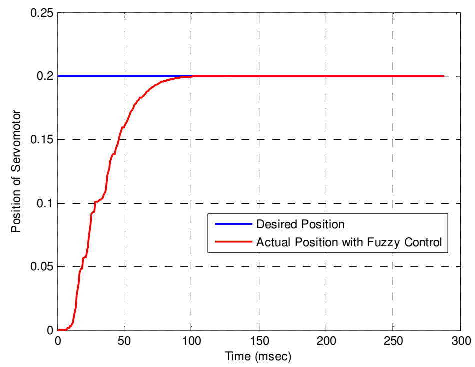

Figure 5.6: Position Control with Fuzzy Logic Controller, Case-6

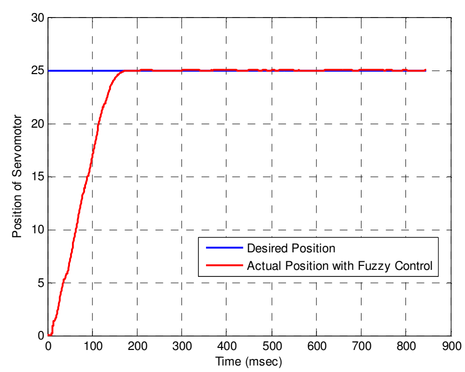

Figure 5.7: Position Control with Fuzzy Logic Controller, Case-7

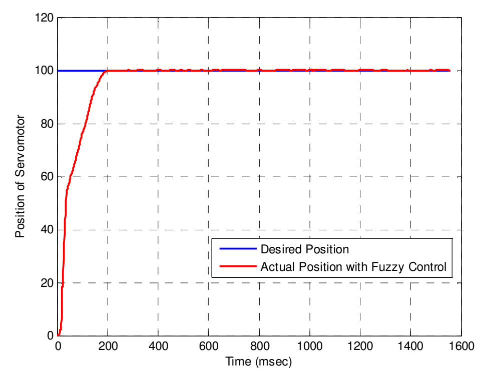

Figure 5.8: Position Control with Fuzzy Logic Controller, Case-8

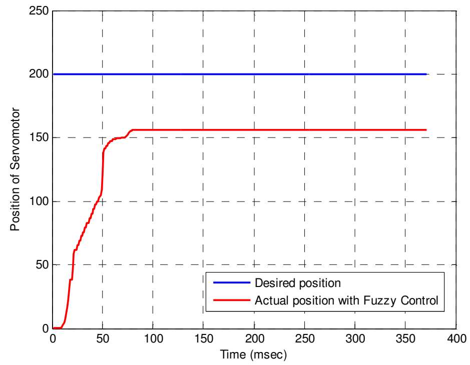

Figure 5.9: Position Control with Fuzzy Logic Controller, Case-9

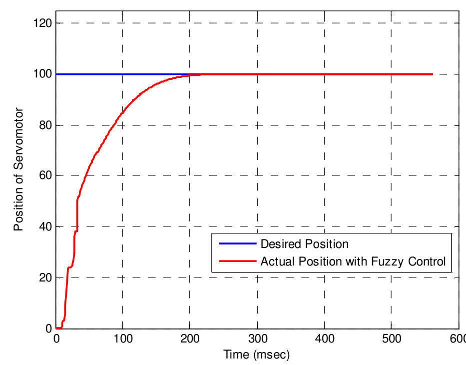

Figure 5.10: Position Control with Fuzzy Logic Controller, Case-10

It is seen from the Figures given so far that, the servomotor reaches to the desired position successfully.

Table 5.1: Fuzzy Logic Controller Performance Results

Cases	Max Overshoot (%)	Settling Time (ms)	Desired Position	Actual Position	Steady-State Error (%)
Case-1	No Overshoot	217	8.00	7.958	0.525
Case-2	No Overshoot	112	0.50	0.50	0
Case-3	No Overshoot	97	0.50	0.50	0
Case-4	No Overshoot	120	0.20	0.20	0
Case-5	No Overshoot	91	0.50	0.50	0
Case-6	No Overshoot	120	0.20	0.20	0
Case-7	No Overshoot	197	25.00	25.00	0
Case-8	No Overshoot	234	100.00	100.00	0
Case-9	No Overshoot	83	200.00	156.20	21.90
Case-10	No Overshoot	266	100.00	99.98	0.02

Next, the reference signals are applied in the form of square and sinusoidal waveform, and the results are given in Figures 5.11-5.12, and Figures 5.13-5.14, respectively.

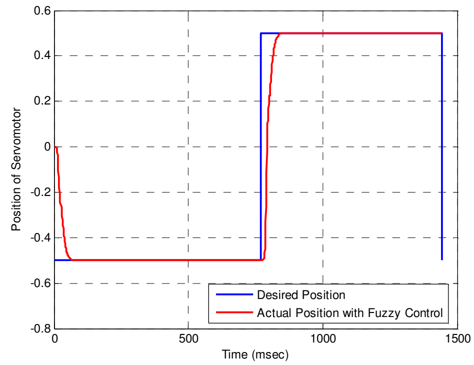

Figure 5.11: System Response to Square Wave Input Signal with FLC, Case-2

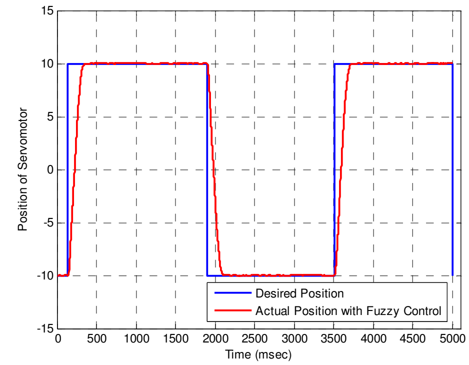

Figure 5.12: System Response to Square Wave Input Signal with FLC, Case-7

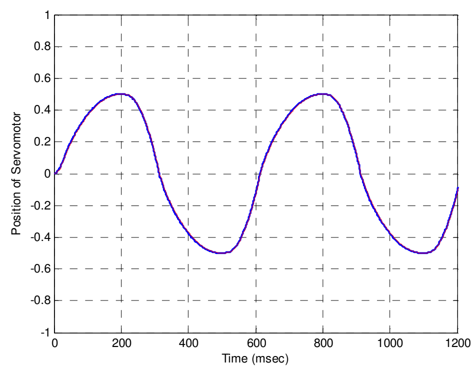

Figure 5.13: System Response to Sinusoidal Wave Input Signal with FLC, Case-2

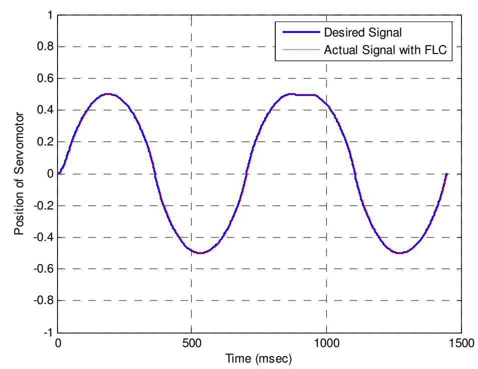

Figure 5.14: System Response to Sinusoidal Wave Input Signal with FLC, Case-7

5.2 PID CONTROL

In this section, the results of tuning the controller parameters with CHR, and Trial and Error methods are given.

Some of the simulation results are provided in table and graphical form to further demonstrate the success of the PID control method. For all cases, the simulation results are given through Figures 5.15 and 5.19 graphically, and the numerical result are given in Table 5.2.

Table 5.2: PID Controller Performance Results

Cases	Max Overshoot (%)	Settling Time (ms)	Desired Position	Actual Position	Steady-State Error (%)
CHR*	64.80	78	100	100	0
CHR**	39.52	46	50	50	0
Trial-1	0.50	97	100	100	0
Trial-2	1.30	35	100	100	0
Trial-3	8.20	289	100	100	0

**: CHR Parameters without overshoot case*

***: CHR Parameters with overshoot case*

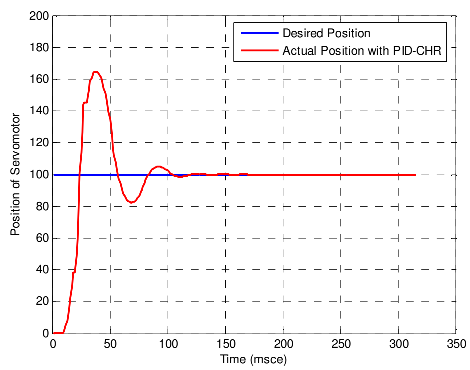

Figure 5.15: Position Control with PID Controller optimized using CHR Method without Overshoot

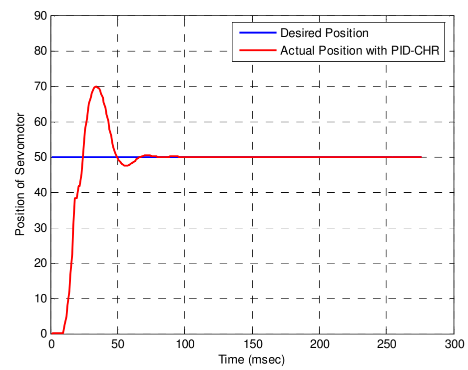

Figure 5.16: Position Control with PID Controller optimized using CHR Method with Overshoot

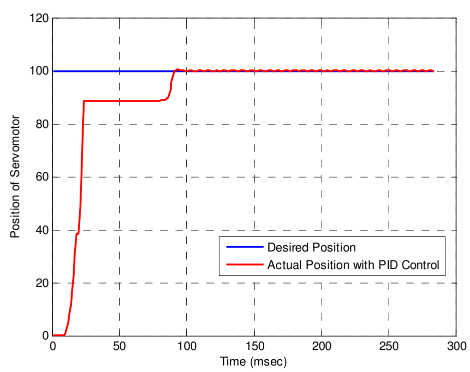

Figure 5.17: Position Control with PID Controller, Trial-1

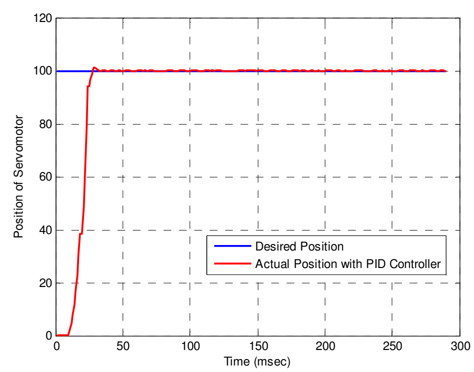

Figure 5.18: Position Control with PID Controller, Trial-2

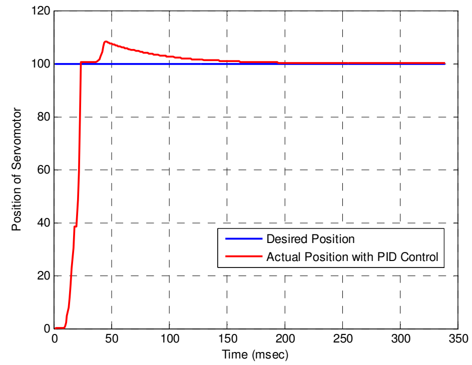

Figure 5.19: Position Control with PID Controller, Trial-3

It is seen from the Figures 5.15-5.19 that initial tuning using the CHR method has not given successful results on the system response. There happened overshoot problems. For the Trials 1 and 3, although the system's outputs reach the desired outputs, there is still an overshoot and smooth response problem. As for the trial 2, which yielded the best performance measures.

5.3 FUZZY-PID CONTROL

For the Fuzzy-PID controller given in section 4.4 of Chapter 4, various situations were considered, and the results are given in Table 5.3, and also in Figures 5.20-5.26 graphically.

Table 5.3: Fuzzy-PID Controller Performance Results

Cases	Max Overshoot (%)	Settling Time (ms)	Desired Position	Actual Position	Steady-State Error (%)
Situation-1	No Overshoot	174	25	25	0
Situation-2	No Overshoot	163	25	25	0
Situation-3	No Overshoot	22	0.50	0.50	0
Situation-4	2.46	25	0.50	0.50	0
Situation-5	No Overshoot	68	0.50	0.50	0
Situation-6	4.82	58	0.50	0.50	0
Situation-7	No Overshoot	126	200	200	0

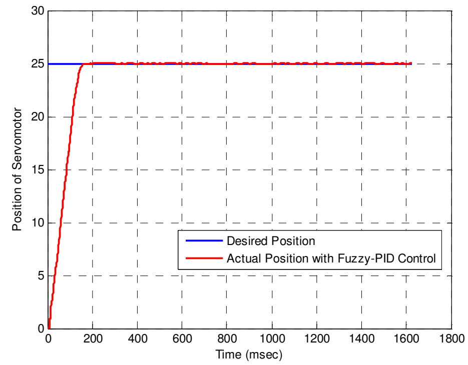

Figure 5.20: Position Control with Fuzzy-PID Controller, Situation-1

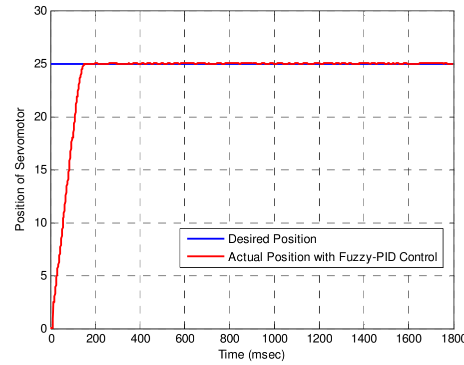

Figure 5.21: Position Control with Fuzzy-PID Controller, Situation-2

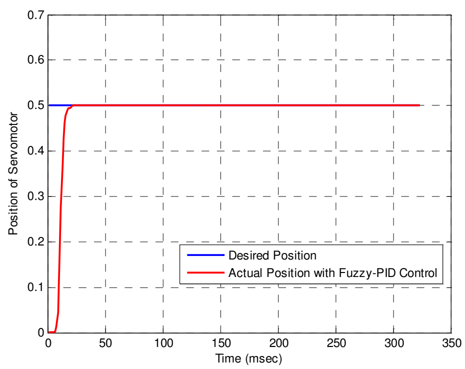

Figure 5.22: Position Control with Fuzzy-PID Controller, Situation-3

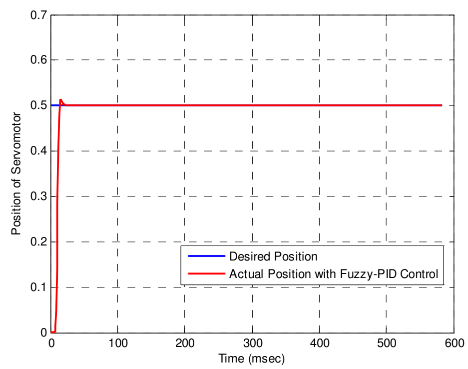

Figure 5.23: Position Control with Fuzzy-PID Controller, Situation-4

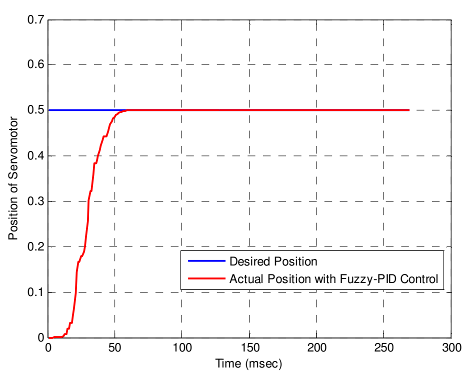

Figure 5.24: Position Control with Fuzzy-PID Controller, Situation-5

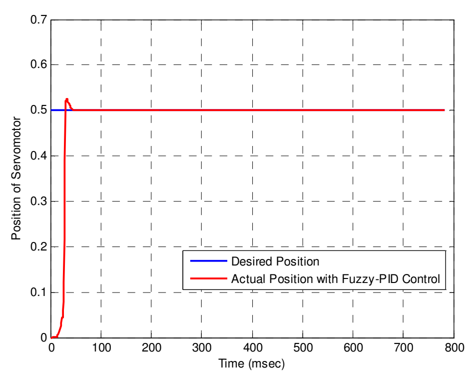

Figure 5.25: Position Control with Fuzzy-PID Controller, Situation-6

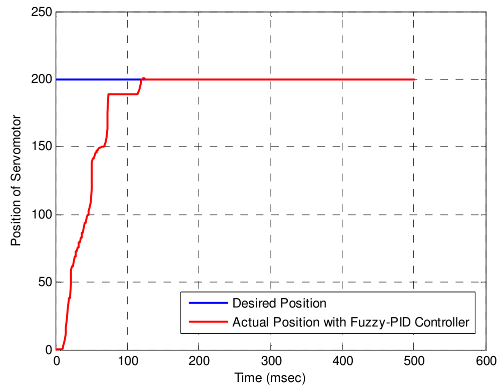

Figure 5.26: Position Control with Fuzzy-PID Controller, Situation-7

So, we can easily see that, Fuzzy-PID Controller is the best controller for the considered system except for the situation-7 of which the response is given in Figure 9.26. There is almost no overshoot (i.e. less than 5%), steady-state error, and the system's outputs reach the desired outputs with acceptable rise and short settling times.

CHAPTER 6

CONCLUSION

1. The design and simulation of a modular Fuzzy Logic, PID, and Fuzzy-PID based controllers for a position control system have been presented. Fuzzy rule base was generated by using the expert knowledge. Fuzzy membership functions and fuzzy sets were developed. The FLC method was tested with MATLAB fuzzy logic toolkit for a number of cases. Tuning of the controller in the form of adjusting the membership functions and the rules was performed in order to improve the performance of the system.

2. For control tasks, the fuzzy logic control is a very flexible and robust soft computing tool. The performance of the system was studied for ten different cases. For all cases of FLC, no overshoots did happen. The FLC exhibited good performance for nine cases. Only Case-9 has not a desired response.

3. Fuzzy logic control has been proven to be an excellent solution for control problems where the number of rules for a system is finite. The FLC in a way, acts as a learning system control, as it has the ability to learn from situations where it fails. This learning is possible by increasing the number of rules in the system. In this way the system can keep on learning until it becomes a perfect system.

4. For the considered system, Fuzzy Logic Controller has also been verified to be a robust controller. The systems, which need robust performance can be successfully controlled by FLC. By tuning the FLC, the total error of the control loop may be decreased according to the requirements of a system. The tuning is performed by modifying both fuzzy sets and fuzzy rule base.

5. The PID controller provided very rapid response in getting the steady-state value, but the main problem for a PID controller is the overshoot. Overshoot was observed for all cases of the PID controller.

6. In order to minimize the overshoot and settling time, a Fuzzy-PID Controller is proposed. For all situations there were no steady-state errors, and in all ones overshoot was reduced to the desired range (less than 10%). The situations 1, 2, 3, 5, and 7 did not exhibit any overshoot. The situations 4 and 6 had overshoots of 2.46% and 4.82%, respectively. These overshoots are less than 10% in magnitude, which are in the desired range. Thus, the combination of Fuzzy-PID controller can be recommended for the best performance.

APPENDIX A

TABLES OF RULES FOR DIFFERENT CASES

The tables of rules, which were used for the Design of Fuzzy Logic Controller for different Cases in Chapter 4, are given in the following.

Table A.1: Table of rules for Case-3

ce \ e	***ne***	***ze***	***po***
ne	nb	nm	ze
ze	nm	ze	pm
po	ze	pm	pb

Table A.2: Table of rules for Case-4

ce \ e	***nb***	***nm***	***ns***	***ze***	***ps***	***pm***	***pb***
nb	nb	nb	nb	nm	nm	ns	ze
nm	nb	nb	nm	nm	ns	ze	ps
ns	nb	nm	nm	nm	ze	ps	pm
ze	nm	nm	nm	ze	pm	pm	pm
ps	nm	nm	ze	pm	pm	pm	pb
pm	nm	ze	pm	pm	pm	pb	pb
pb	ze	pm	pm	pm	pb	pb	pb

Table A.3: Table of rules for Case-5

ce \ e	***nb***	***nm***	***ns***	***ze***	***ps***	***pm***	***pb***
nb	nb	nb	nb	nm	nm	ns	ze
nm	nb	nb	nm	nm	ns	ze	ps
ns	nb	nm	nm	nm	ze	ps	pm
ze	nm	nm	nm	ze	pm	pm	pm
ps	nm	nm	ze	pm	pm	pm	pb
pm	nm	ze	pm	pm	pm	pb	pb
pb	ze	pm	pm	pm	pb	pb	pb

Table A.4: Table of rules for Case-6

ce \ e	***nb***	***nm***	***ns***	***ze***	***ps***	***pm***	***pb***
nb	nb	nb	nb	nm	nm	ns	ze
nm	nb	nb	nm	nm	ns	ze	ps
ns	nb	nm	nm	nm	ze	ps	pm
ze	nm	nm	nm	ze	pm	pm	pm
ps	nm	nm	ze	pm	pm	pm	pb
pm	nm	ze	pm	pm	pm	pb	pb
pb	ze	pm	pm	pm	pb	pb	pb

Table A.5: Table of rules for Case-7

ce \ e	***nb***	***nm***	***ns***	***ze***	***ps***	***pm***	***pb***
nb	nb	nb	nb	nm	nm	ns	ze
nm	nb	nb	nm	nm	ns	ze	ps
ns	nb	nm	nm	nm	ze	ps	pm
ze	nm	nm	nm	ze	pm	pm	pm
ps	nm	nm	ze	pm	pm	pm	pb
pm	nm	ze	pm	pm	pm	pb	pb
pb	ze	pm	pm	pm	pb	pb	pb

Table A.6: Table of rules for Case-8

ce \ e	***nb***	***nm***	***ns***	***ze***	***ps***	***pm***	***pb***
nb	nb	nb	nb	nm	nm	ns	ze
nm	nb	nb	nm	nm	ns	ze	ps
ns	nb	nm	nm	nm	ze	ps	pm
ze	nm	nm	nm	ze	pm	pm	pm
ps	nm	nm	ze	pm	pm	pm	pb
pm	nm	ze	pm	pm	pm	pb	pb
pb	ze	pm	pm	pm	pb	pb	pb

Table A.7: Table of rules for Case-9

ce \ e	***nb***	***nm***	***ze***	***pm***	***pb***
nb	nb	nb	nm	nm	ze
nm	nb	nm	nm	ze	pm
ze	nb	nm	ze	pm	pb
pm	nm	ze	pm	pm	pb
pb	ze	pm	pm	pb	pb

Table A.8: Table of rules for Case-10

ce \ e	***nb***	***nm***	***ns***	***ze***	***ps***	***pm***	***pb***
nb	nb	nb	nb	nm	nm	ns	ze
nm	nb	nb	nm	nm	ns	ze	ps
ns	nb	nm	nm	nm	ze	ps	pm
ze	nm	nm	nm	ze	pm	pm	pm
ps	nm	nm	ze	pm	pm	pm	pb
pm	nm	ze	pm	pm	pm	pb	pb
pb	ze	pm	pm	pm	pb	pb	pb

REFERENCES

Adcock, Timothy A., *"What is Fuzzy Logic? An Overview of Latest Control Methodology"*, Digital Signal Processing Products, Texas Instruments, 1993.

Antsaklis, Panos J., *"Intelligent Control"*, Encyclopedia of Electrical and Electronics Engineering, John Wiley & Sons, Inc., USA, 1997.

Antsaklis, Panos J. and Kevin M. Passino, *An Introduction Intelligent and Autonomous Control*, Kluwer Academic Publishers, London, 1997.

Baldwin, J. F. and Guild, N. C. F., "Feasible Algorithms For Approximate Reasoning Using Fuzzy Logic", *Fuzzy Sets and Systems*, v3, pp. 225-251, 1980.

Baldwin, J. F. and Pilsworth, B. W., "Axiomatic Approach For Approximate Reasoning with Fuzzy Logic", *Fuzzy Sets and Systems*, v3, pp. 193-219, 1980.

Coleman, Charles P. and Godbole, Datta, University of California at Berkeley "A Comparison of Robustness: Fuzzy Logic, PID, & Sliding Mode Control", pp.06-08, 1996.

Furukawa, Tomonari, *"MTRN9224 Robot Design Lectures*, School of Mechanical and Manufacturing Engineering, University of New South, Wales, 2004.

Yamakawa, T., "Stabilization of an inverted pendulum by a high-speed Fuzzy Logic Controller Hardware System", *Fuzzy Sets and Systems*, Vol. 32, pp. 161-180, 1990.

Cao, Z. and Kandel, A., "Applicability of some Fuzzy Implication Operators", *Fuzzy Sets and Systems*, v31, pp. 151-186, 1989.

Nonnenmarcher, Wilfred and Gao, Zhiqiong, *"Fuzzy Logic Control of an Industrial Indexing Motion Application"*, ISA Conference September 10-13, 2001.

Liberatore, Vincenzo and Newman, Wyatt S. , "*Remote Human/Machine Control*", EECS Dept. Case Western Reserve University Cleveland, OH 44106, 2002.

Han, K. Kim, H., S., Kim, Y. J. and Kim, J. H., "Internet Control Architecture for Internet-Based Personal Robot," *Autonomous Robots Journal*, Kluwer Academic Publishers, vol. 10, no. 2, pp. 135-147, Mar. 2001.

Hirota, K., Arai, A., and Hachiso, S., "*Fuzzy controlled robot arm playing two-dimensional ping-pong game*", *Fuzzy Sets and Systems*, Vol. 32, No.2, pp. 149-159, 1989.

Kelkar, Nikhil D., *A Fuzzy Controller for three-dimensioned Line Following of an Unmanned Autonomous Mobile Robot*, M.S. Thesis, University of Cincinnati, 1997.

Lee, Joo-Ho and Hashimoto, Hideki, "*Controlling Mobile Robots in Distributed Intelligent Sensor Network", IEEE Transactions on Industrial Electronics*, Vol. 50, No. 5, October 2003.

Rose, Timothy J., *Fuzzy Logic with Engineering Applications*, John Wiley, West Sussex, 2005.

Saffiotti, Alessandro , *Fuzzy Logic in Autonomous Robot Navigation a case study,* Center for Applied Autonomous Sensor Systems Dept. of Technology, University of Örebro, Sweden, 1998.

Sugeno M. and Nishida, M., "*Fuzzy control of model car*," *Fuzzy Sets and Systems*, pp. 103-113, 1985.

Tang, K. S., Man, Kim Fung, Chen, Guanrong, and Kwang, Sam, *"An Optimal Fuzzy PID Controller"*, IEEE Transactions on Industrial Electronics, Vol. 48, No. 4, 2001.

The MathWorks, *Fuzzy Logic Toolbox User's Guide*, Version 2, The MathWorks Inc., 2002.

Wang, Li-Xin, *A Course in Fuzzy Systems and Control*, Prentice Hall, New Jersey, 1997.

Won, Chung-Yuen, Kim, Sei-Chan, and Bose, Bimal K., "*Robust Position Control of induction motor using fuzzy logic control",* Department of Electrical Engineering, Sung Kyun Kawn University, Seoul, Korea, 1992.

Yen, John and Langari, Reza, *Fuzzy Logic, Intelligence, Control, and Information*, Prentice Hall, New Jersey, 1999.

Zilouchian, Ali and Howard, David W., *"Design of a Fuzzy Inference System for Control of Robot Manipulators"*, Department of Electrical Engineering, Florida Atlantic University, Boca Raton, 2000.