FUZZY LOGIC CONTROLLER FOR INVERTED PENDULUM WITH REDUCED LINGUISTIC VARIABLE

A DISSERTATION

Submitted in partial fulfillment of the
requirements for the award of the degree
of
MASTER OF TECHNOLOGY
in
ELECTRONICS AND COMMUNICATION ENGINEERING
(With Specialization in Control and Guidance)

By
KULDEEP KUMAR DHIMAN

DEPARTMENT OF ELECTRONICS AND COMPUTER ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY ROORKEE
ROORKEE - 247 667 (INDIA)

JUNE, 2011

CANDIDATE'S DECLARATION

I hereby declare that the work, which is presented in this dissertation report, titled **"Fuzzy Logic Controller for Inverted Pendulum with Reduced Linguistic Variable"** being submitted in partial fulfillment of the requirements for the award of the degree of **Master of Technology** with specialization in **Control and Guidance**, in the Department of Electronics and Computer Engineering, Indian Institute of Technology, Roorkee is an authentic record of my own work carried out from July 2010 to June 2011, under guidance and supervision of **Dr. R. MITRA**, Professor, Department of Electronics and Computer Engineering, Indian Institute of Technology, Roorkee.

The results embodied in this dissertation have not submitted for the award of any other Degree or Diploma.

Date: 28 **June 2011**
Place: Roorkee

(KULDEEP KUMAR DHIMAN)

CERTIFICATE

This is to certify that the statement made by the candidate is correct to the best of my knowledge and belief.

Date: 23 **June 2011**
Place: Roorkee

(Dr. R. MITRA)
Department of Electronics and Computer Engineering,
Indian Institute of Technology Roorkee
Roorkee – 247 667, (INDIA)

ACKNOWLEDGEMENTS

It is my privilege and pleasure to express my profound sense of respect, gratitude and indebtedness to my guide, **Dr. R. MITRA**, Professor, Department of Electronics and Computer Engineering , Indian Institute of Technology, Roorkee, for his inspiration, guidance, constructive criticisms and encouragement throughout this dissertation work.

Thanks are due to the Lab staff Control and Guidance Lab, Department of Electronics and Computer Engineering, IIT Roorkee for providing necessary facilities.

I gratefully acknowledge my sincere thanks to my family members for their inspirational words and moral support during course of this work.

I am greatly indebted to all my friends, who have graciously applied themselves to the task of helping me with ample morale support and valuable suggestions. Finally, I would like to extend my gratitude to all those persons who have directly or indirectly helped me in the process and contributed towards this work.

KULDEEP KUMAR DHIMAN

ABSTRACT

The inverted pendulum is a classical control problem, which involves developing a control system to balance a pendulum. The aim of this study is to stabilize the Inverted Pendulum such that the position of the cart on the track is controlled quickly and accurately so that the pendulum is always erected in its inverted position during such movements.

This thesis present a design methodology for stabilization of an IP (Inverted Pendulum) with reference tracking using ANFIS (Adaptive Neural Fuzzy Inference System) with a single linguistic variable and two membership function only hence two rules. The proposed FLC (Fuzzy Logic Controller) is the simplest FLC that retains all the merits of four input linguistic variable FIS tuned by ANFIS and is more robust than the conventional PD controller. Experiments are carried out in MATLAB Simulink to demonstrate the performance of the purposed controller. The design procedure is conceptually simple and natural.

In this design procedure firstly two independent PD controllers are tuned one for angle and another for cart position on the rail. Now training data is taken from these controllers and FIS is tuned with the help of ANFIS. Once FIS is tuned, then further work is carried out to reduce the numbers of input linguistic variables, by adding one variable to another and tuning the gain parameters using trial and error method.

Systems are simulated in the presence of disturbance. Overshoot and settling time are also kept in mind while comparing between two simulation results, because these two entities are mutually dependent on each other, if we reduce one the other will increase, and vice-versa.

Contents

LIST OF FIGURES

Chapter 1

INTRODUCTION

The Inverted Pendulum (IP) System offers a very good example for control engineers to verify modern control methods. This system is a highly nonlinear and open-loop unstable one. This means that standard linear techniques cannot model the nonlinear dynamics of the system. When the system is simulated the pendulum falls over quickly. The IP system has the property of being unstable, of higher order, multi variable and is highly coupled, which can be treated as a typical nonlinear problem [1, 2]. The characteristics of the inverted pendulum makes identification and control more challenging. The system's characteristics are an unstable equilibrium point at the upright position of the pendulum, a stable equilibrium point at the pendant position, as well as two uncontrollable points when the pendulum is at the horizontal position [4]. Inverted Pendulum is a model for the altitude control of a space booster rocket and a satellite, an automatic aircraft landing system, aircraft stabilization in the turbulent air-flow, stabilization of a cabin in a ship etc.

The inverted pendulum is an interesting subject from the control point of view due to its inherent nonlinearity. The problem is to balance a pole on a mobile platform that can move in only two directions, to the left or to the right. This control problem is fundamentally the same as those involved in rocket or missile propulsion. Common control approaches such as Proportional-Integral-Derivative (PID) control and Linear Quadratic control (LQ) require a good knowledge of the system and accurate tuning in order to obtain desired performances [5, 15]. However, it is often impossible to specify an accurate mathematical model of the process, or the description with differential equations is extremely complex.

In order to obtain control surface, the inverted pendulum dynamics should be locally linearized [15]. Moreover, application of these control techniques to a two or three stage inverted pendulum may result in a very critical design of control parameters and difficult stabilization. However, using artificial intelligence controllers such as artificial neural network and fuzzy logic controllers, the controller can be design without requiring the model to be linearized [3, 6, 9, 14]. The non-linearized model can be simulated directly using the Matlab application to see result. Therefore, in this thesis following controllers have been simulated. Which may be divided into two categories.

1. Conventional Controller
   Proportional Derivative (PD)
2. Artificial Intelligence Controller
   Fuzzy Logic Controller (FLC) tuned with the help of ANFIS (Adaptive Neuro Fuzzy Inference System)

1.1 Motivation

My educational experience, at IIT Roorkee has given me a broad background in designing different types of control strategy to control various systems available in control system laboratory like Ball-Beam System, Inverted Pendulum, Magnetic Levitation, Robot Arms etc. Since my current interests lie within Fuzzy Logic Controller (FLC) and Neural Network (NN), I decided to design a controller using FLC and NN to stabilize an unstable system.

The following reasons help explain why the inverted pendulum on a cart has been selected as the system on which the findings of this thesis will be implemented.

1. It is a non-linear system, yet can be approximated as a linear system if the operating range is small (i.e. slight variations of the angle from the norm).
2. Intuition plays a large part in the human understanding of the inverted pendulum model. When the control method is supplemented with a fuzzy logic and artificial neural network optimization techniques, the result will provide an insight into the measure of ability of the method to provide control.
3. The cart/pole system is a common test case for fuzzy logic, so any result can be compared to previous work in the field. A proportional, derivative (PD) will be used as a reference because it is one of the basic approaches for controlling the system performance.

1.2 Objective and Scope of Work

Although many investigations have been carried out on the inverted pendulum problem, like PD, hybrid PD + Fuzzy, FIS, ANFIS and so on, each investigation has its own merits [1-8]. One most popular method is the FIS tuned with the help of ANFIS. A common FIS controller controls both angle as well as reference position, consist four linguistic variables at the input and one at the output [1]. If each linguistic variable has two MFs (Membership Function) then rule base requires total 16 rules. If we takes three MFs in place of two, than number of rules are 81. Therefore, a huge number of rules for multi input FIS is a problem of FIS.

The aim of this work is to implement a FLC on inverted pendulum, tuned with the help of Adaptive Neuro Fuzzy Inference System (ANFIS). The implemented FLC must have minimum numbers of linguistic variables hence the minimum numbers of rules, without degrading the performance of FIS, tuned by ANFIS. To satisfy the intended objective, the following scope of work was carried out.

1. Determine the mathematical model for an inverted pendulum system.
2. Design a conventional PD controller, simulate it using Matlab and tune it for optimal performance controller.
3. Design an ANFIS controller and trained it using training data obtain from PD controller.
4. Reduced the numbers of linguistic variables of ANFIS controller.
5. Comparison of the simulation results of conventional PD controller and ANFIS controller with reduced number of linguistic variable.

1.3 Literature Review

The classic control problem of the inverted pendulum is interesting in that it can be solved using a wide variety of systems and solutions. Numbers of literature are available regarding suggestion to design a controller for inverted pendulum with some merits and demerits. Robustness and disturbance rejection properties are two important performance criterions for any control system and important parameters to compare various controlling schemes.

The PID controller and the general LQR, linear quadratic optimal controller, are designed mostly by use of the approximate linear model of the inverted pendulum system [5, 15]. Although the designed controller can stabilize the inverted pendulum system, the control object in the simulation test is still approximately linear model rather than the actual inverted pendulum system.

Many studies have been reported on fuzzy logic controllers which have enjoyed successful industrial applications and have demonstrated significant improvements in performance over conventional techniques [3, 5, 6, 9, 10, 14, 16]. In most applications, however, the design of the controllers is accomplished by "trial and error" methods using computer simulation.

Many of the literature are available on hybrid method to control the inverted pendulum system [4, 13]. The neuro-fuzzy controllers are the popular controllers of this century because of their remarkable effectiveness, involve human expertise, learning capability and broad applicability [2, 7, 8]. Some of the literature suggest tuning of fuzzy controller with the help of artificial neural network [1, 12].

1.4 Organization of the thesis

The report has been organized into seven chapters. **Chapter 1** gives an impression of the subject, basics, literature survey and objective of the study. **Chapter 2** briefly discussed the Inverted Pendulum and its classifications. **Chapter 3** describes the modeling of single stage linear inverted pendulum and various control strategy. **Chapter 4** contains the introduction of fuzzy logic systems. **Chapter 5** gives an overview of the artificial neural network. **Chapter 6** presents the simulation results for controller designed using PD and ANFIS (Adaptive Neuro Fuzzy Inference System). **Chapter 7** presents the conclusion of the study and suggestions are given for further study of this work.

Chapter 2

INVERTED PENDULUM

2.1 Introduction

Inverted pendulum (IP) problem is the combination of research area like robotics, control theory, computer control, etc. The IP system has the property of unstable, high order, multi-variable and highly coupled, which can be treated as a typical nonlinear control problem. IP system provides an excellent experimental platform for examining specific control theories or typical solutions and thus promoting the development of the new theories. They are widely applied in different fields such as semiconductors, delicate devices processing, robot control technology, artificial intelligence, missiles interception control systems, aviation docking control technology and general industrial applications.

2.2 Inverted Pendulum classification

There are many series of IP systems and are classified as follows:-

- Linear Inverted Pendulum
- Circular Inverted Pendulum
- Planar Inverted Pendulum
- Configurable Inverted Pendulum

Linear IP has pendulum plant on a linear motion module with one degree of freedom. The cart moves on the sliding shaft horizontally.

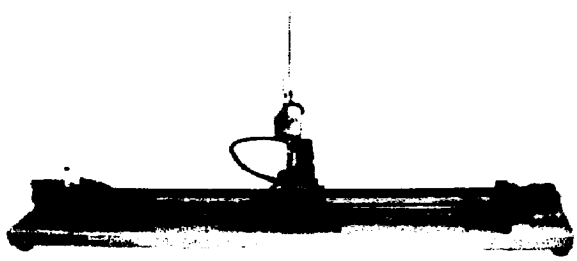

Fig. 2.2.1 Linear Inverted Pendulum

Circular IP system has the pendulum plant on a circular motion module with one degree of freedom. The pendulum is on the arm end and rotates around the center of the circle.

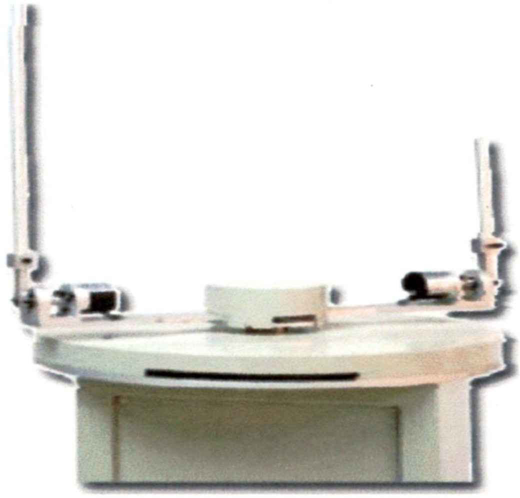

Fig. 2.2.2 Circular Inverted Pendulum

Planar IP system has the pendulum plant on the planar motion module with two degrees of freedom.

Fig. 2.2.3 Planar Inverted Pendulum

Configurable IP is a new class of IP systems whose pendulum plant is composed of pendulum rod and connection rod. The connection rod can be configured to three modes: level, vertical up and vertical down.

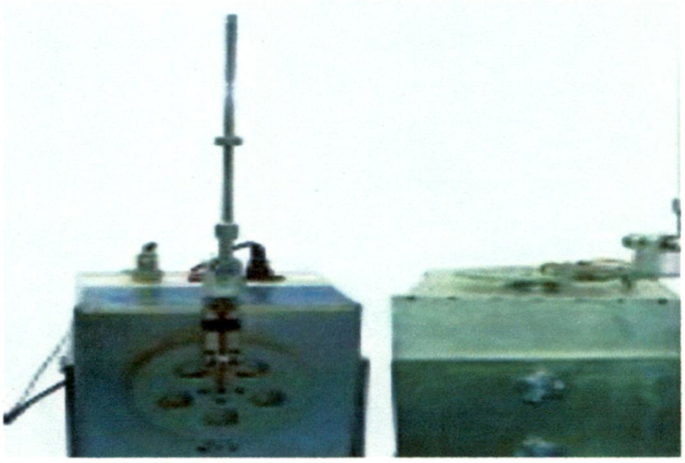

Fig. 2.2.4 Configurable Inverted Pendulum

2.3 Inverted Pendulum properties

All Inverted Pendulum (IP) systems have the following properties:-

1. Nonlinearity: IP is a typical nonlinear control system. In real control, the system model is usually linearized.
2. Uncertainty: Most uncertainties come from model uncertainty, mechanical transmission error and other resistances. In real control, uncertainties are reduced by controlling errors.
3. Open loop instability: There are two equilibrium states for IP systems, vertical up and vertical down, in which the vertical up is the unstable equilibrium point and vertical down is the stable equilibrium point.
4. Limitations: The IP system performance is limited by mechanisms like motion module travel distance, motor torque etc. The effect of travel distance to IP swing up is especially evident, short travel distance easily gets the cart exceed the limit switch.

Chapter 3

SINGLE STAGE LINEAR INVERTED PENDULUM SYSTEM MODEL AND ANALYSIS

The problem involves a cart, able to move backwards and forwards, and a pendulum, hinged to the cart at the bottom of its length such that the pendulum can move in the same plane as the cart. That is, the pendulum mounted on the cart is free to fall along the cart's axis of motion. The system is to be controlled so that the pendulum remains balanced and upright and is resistant to a disturbance. Free body diagram of the inverted pendulum system is shown in Fig. 3.1.

The mechanical modeling is based on the foundation of understanding the motion patterns through physical and mathematical means to set up the internal input/output relationship in the system. There are difficulties in IP system modeling because of its instability. The dynamic equation will be obtained in the inertial frame by classical mechanics theory.

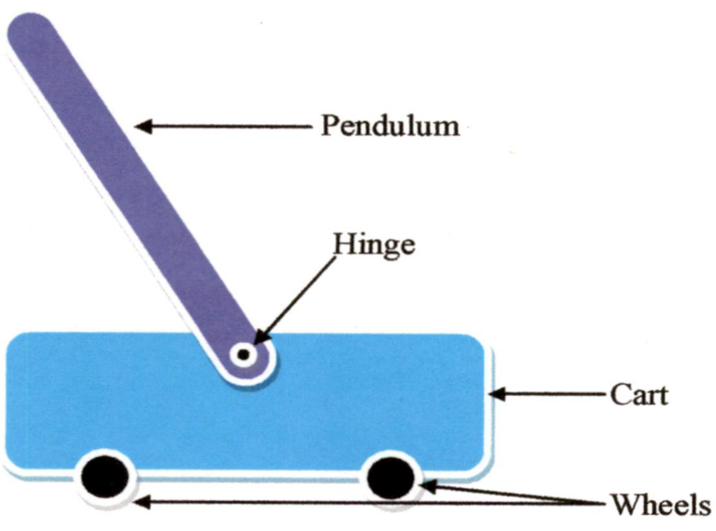

Fig. 3.1 Free body diagram of the inverted pendulum system

3.1 Newton's mechanics

Single stage linear IP can be simplified as a system of cart and even quality rod, as shown in Fig. 3.1.1

The Parameters and their symbols are as follows:

M - cart mass

m - rod mass

b - friction co-efficient of the cart

l -distance from the rod axis rotation center to the rod mass center

I - rod inertia

F - force acting on the cart

x - cart position

Φ - angle between the rod and vertically upward direction

θ - angle between the rod and vertically downward direction

Fig. 3.1.1 is the force analysis of cart and rod system. N and P denote the interactive force of cart and rod in the horizontal and vertical direction respectively.

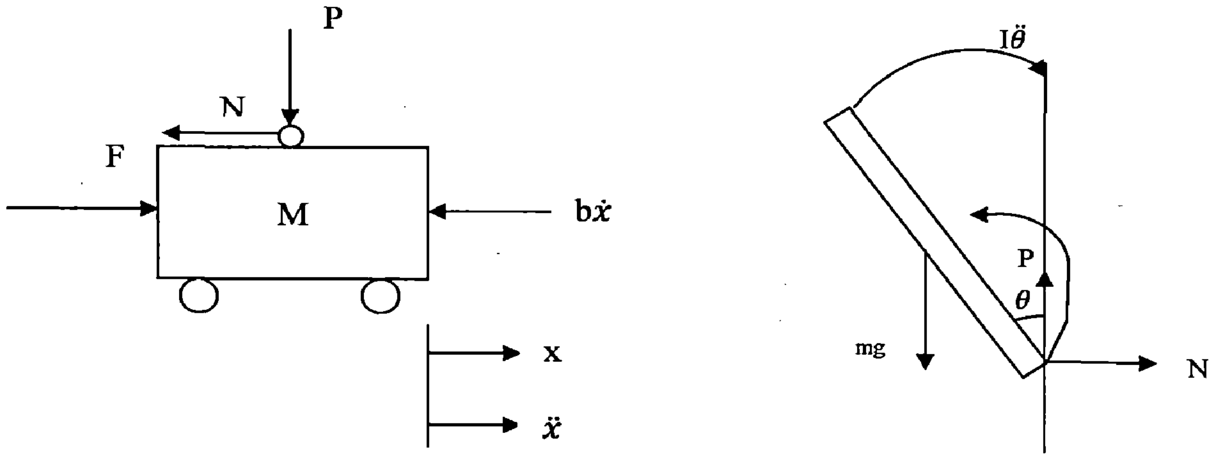

Fig. 3.1.1 Cart and Rod Force analysis

From the forces in the horizontal direction

$$M\ddot{x} = F - b\dot{x} - N \tag{3.1.1}$$

From the force acting on the rod in horizontal direction

$$N = m\frac{d^2}{dt^2}(x + l\sin\theta) \qquad (3.1.2)$$

That is

$$N = m\ddot{x} + ml\ddot{\theta}\cos\theta - ml\dot{\theta}^2\sin\theta \qquad (3.1.3)$$

Combining equation (3.1.3) with equation (3.1.1), the first dynamic equation is obtained as

$$(M + m)\ddot{x} + b\dot{x} + ml\ddot{\theta}\cos\theta - ml\dot{\theta}^2\sin\theta = F \qquad (3.1.4)$$

To get the second dynamic equation, analyze the force in the vertical direction

$$P - mg = m\frac{d^2}{dt^2}(l\cos\theta) \qquad (3.1.5)$$

$$P - mg = -ml\ddot{\theta}\sin\theta - ml\dot{\theta}^2\cos\theta \qquad (3.1.6)$$

By moment conservation

$$-Pl\sin\theta - Nl\cos\theta = I\ddot{\theta} \qquad (3.1.7)$$

Note: the direction of moment is negative because $\theta = \pi + \Phi$, $\cos\Phi = -\cos\theta$, $\sin\Phi = -\sin\theta$.

Combining the above two equation, the second dynamic equation is obtained as:

$$(I + ml^2)\ddot{\theta} + mgl\sin\theta = -ml\ddot{x}\cos\theta \qquad (3.1.8)$$

Let $\theta = \pi + \Phi$ (Φ is the angle between the rod and vertically upward direction), assume Φ is relatively small to 1 (unit in radian), which means $\Phi \ll 1$, then following approximation can be obtained: $\cos\theta = -1$, $\sin\theta = -\Phi$, $\left(\frac{d\theta}{dt}\right)^2 = 0$. Let u denote the input force of the object, linearize the two dynamics equations.

$$\begin{cases}(I+ml^2)\ddot{\phi}-mgl\phi=ml\ddot{x}\\(M+m)\ddot{x}+b\dot{x}-ml\ddot{\phi}=u\end{cases}\tag{3.1.9}$$

The Laplace transformation of equation (3.1.9) is as follows

$$\begin{cases}(I+ml^2)\Phi(s)s^2-mgl\Phi(s)=mlX(s)s^2\\(M+m)X(s)s^2+bX(s)s-ml\Phi(s)s^2=U(s)\end{cases}\tag{3.1.10}$$

Note: the initial condition is assumed to be 1 when deducing the transfer function.

The output angle is Φ, solving the first equation

$$X(s)=\left[\frac{(I+ml^2)}{ml}-\frac{g}{s^2}\right]\Phi(s)\tag{3.1.11}$$

Or

$$\frac{\Phi(s)}{X(s)}=\frac{mls^2}{(I+ml^2)s^2-mgl}\tag{3.1.12}$$

Let $v=\ddot{x}$, then

$$\frac{\Phi(s)}{V(s)}=\frac{ml}{(I+ml^2)s^2-mgl}\tag{3.1.13}$$

Substituting the above equation to the second one in equation (3.1.10)

$$(M+m)\left[\frac{(I+ml^2)}{ml}-\frac{g}{s}\right]\Phi(s)s^2+b\left[\frac{(I+ml^2)}{ml}+\frac{g}{s^2}\right]\Phi(s)s-ml\Phi(s)s^2=U(s)\tag{3.1.14}$$

The transfer function is obtained after simplification

$$\frac{\Phi(s)}{U(s)} = \frac{\frac{ml}{q}s^2}{s^4 + \frac{b(I+ml^2)}{q}s^3 - \frac{(M+m)mgl}{q}s^2 - \frac{bmgl}{q}s} \tag{3.1.15}$$

in which

$$q = (M+m)(l+ml^2) - (ml^2)$$

Assume the system state space equations are

$$\begin{aligned} \dot{X} &= AX + Bu \\ y &= CX + Du \end{aligned} \tag{3.1.16}$$

Solve the algebraic equation of $\ddot{x}, \ddot{\phi}$, the solutions are as follows

$$\begin{cases} \dot{x} = \dot{x} \\ \ddot{x} = \frac{-(I+ml^2)b}{I(M+m)+Mml^2}\dot{x} + \frac{m^2gl^2}{I(M+m)+Mml^2}\phi + \frac{(I+ml^2)}{I(M+m)+Mml^2}u \\ \dot{\phi} = \dot{\phi} \\ \ddot{\phi} = \frac{-mlb}{I(M+m)+Mml^2}\dot{x} + \frac{mgl(M+m)}{I(M+m)+Mml^2}\phi + \frac{ml}{I(M+m)+Mml^2}u \end{cases}$$

(3.1.17)

The state space equation can then be written as

$$\begin{bmatrix} \dot{x} \\ \ddot{x} \\ \dot{\phi} \\ \ddot{\phi} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & \frac{-(I+ml^2)b}{I(M+m)+Mml^2} & \frac{m^2gl^2}{I(M+m)+Mml^2} & 0 \\ 0 & 0 & 0 & 1 \\ 0 & \frac{-mlb}{I(M+m)+Mml^2} & \frac{mgl(M+m)}{I(M+m)+Mml^2} & 0 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ \phi \\ \dot{\phi} \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{(I+ml^2)}{I(M+m)+Mml^2} \\ 0 \\ \frac{ml}{I(M+m)+Mml^2} \end{bmatrix} u$$

$$y=\begin{bmatrix}x\\ \phi\end{bmatrix}=\begin{bmatrix}1&0&0&0\\0&0&1&0\end{bmatrix}\begin{bmatrix}x\\ \dot{x}\\ \phi\\ \dot{\phi}\end{bmatrix}+\begin{bmatrix}0\\0\end{bmatrix}u \tag{3.1.18}$$

The first equation of (3.1.9) is

$$(I+ml^2)\ddot{\phi}-mgl\phi=ml\ddot{x}$$

For a rod of even quality, the moment of inertia is

$$I=\frac{1}{3}ml^2 \tag{3.1.19}$$

Substituting (3.1.19) in (3.1.9),

$$(\frac{1}{3}ml^2+ml^2)\ddot{\phi}-mgl\phi=ml\ddot{x} \tag{3.1.20}$$

After simplification,

$$\ddot{\phi}=\frac{3g}{4l}\phi+\frac{3}{4l}\ddot{x} \tag{3.1.21}$$

Let $X=\{x,\dot{x},\phi,\dot{\phi}\},u'=\ddot{x}$, then

$$\begin{bmatrix}\dot{x}\\ \ddot{x}\\ \dot{\phi}\\ \ddot{\phi}\end{bmatrix}=\begin{bmatrix}0&1&0&0\\0&0&0&0\\0&0&0&1\\0&0&\frac{3g}{4l}&0\end{bmatrix}\begin{bmatrix}x\\ \dot{x}\\ \phi\\ \dot{\phi}\end{bmatrix}+\begin{bmatrix}0\\1\\0\\ \frac{3}{4l}\end{bmatrix}u' \tag{3.1.22}$$

$$y=\begin{bmatrix}x\\ \phi\end{bmatrix}=\begin{bmatrix}1&0&0&0\\0&0&1&0\end{bmatrix}\begin{bmatrix}x\\ \dot{x}\\ \phi\\ \dot{\phi}\end{bmatrix}+\begin{bmatrix}0\\0\end{bmatrix}u'$$

3.2 Real System Model

M=0.455 kg, m=0.21 kg, l=0.61/2 m, g=9.8 m/s^2, $I = \frac{ml^2}{3} = 0.026047$ kg.m^2

Considering the parameters as defined above, the transfer function of pendulum rod angle and cart displacement is given by:

$$\frac{\Phi(s)}{X(s)} = \frac{0.0366s^2}{0.03721s^2 - 0.62769} \quad (3.2.1)$$

The transfer function of pendulum rod and cart acceleration is:

$$\frac{\Phi(s)}{V(s)} = \frac{0.0366}{0.03721s^2 - 0.62769} \quad (3.2.2)$$

The transfer function of pendulum angle and external force acting on the cart is:

$$\frac{\Phi(s)}{U(s)} = \frac{0.13964s}{s^3 + 0.0014197s^2 - 1.59257s - 0.02395} \quad (3.2.3)$$

Analysis of the inverted pendulum with root locus, frequency response, cascade PID controller, swing up control and LQR controller were carried out.

3.3 Root Locus Analysis

The open loop transfer function of the inverted pendulum is

$$\frac{\Phi(s)}{V(s)} = \frac{0.0366}{0.03721s^2 - 0.62769}$$

Let the input be cart acceleration and output be IP system pendulum rod angle.

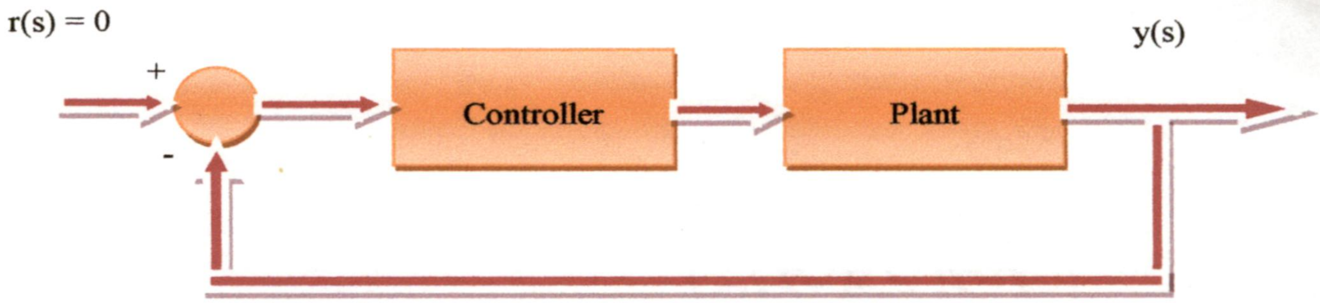

Fig. 3.3.1 Single Stage Linear IP Close Loop Diagram

The closed loop transfer function root locus is as shown below:-

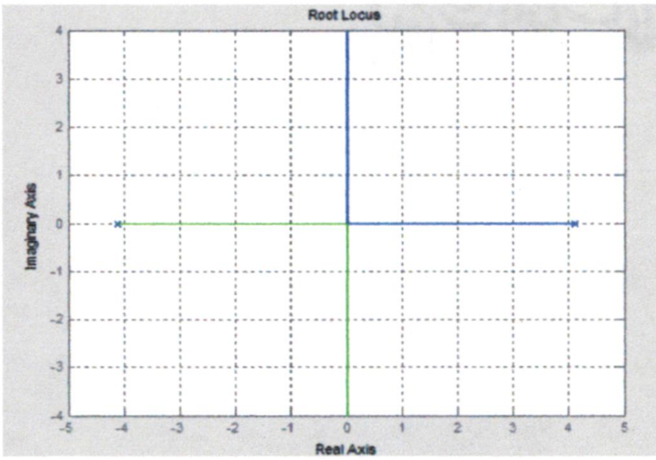

Fig. 3.3.2 Root Locus of Closed Loop Transfer Function of 1-Stage Linear IP

There are two symmetrically located closed loop poles on the real axis, one on the right half plane and one on the left half plane. The root loci start from these points and intersect at the origin and terminate at infinity. This means however the system gain changes, the locus will stay on the right half plane. Hence the system is always unstable.

A controller is to be designed to make the system satisfying the following performance criteria

Response time $t_s = 0.5$ seconds (2%)

Peak Overshoot $M_p \leq 10\%$

With the above specifications, the system controller can be obtained as

$$G_c(s) = \frac{299(s+5.27)}{s+22.70}$$

The root locus of the compensated system and step response of the closed loop is as shown below

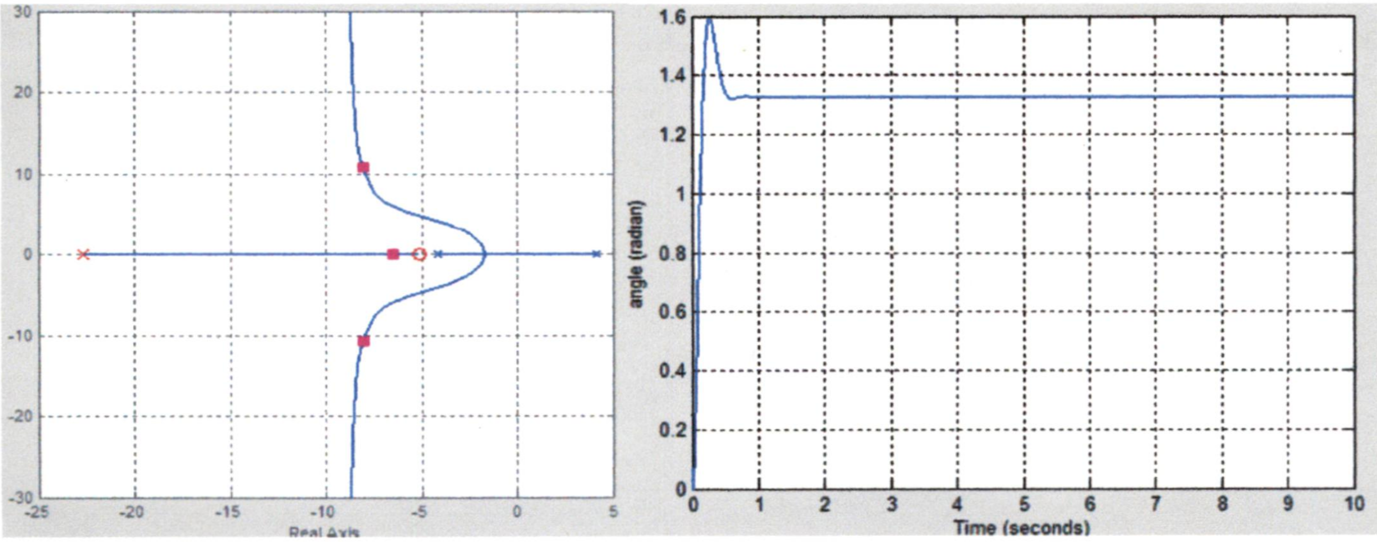

Fig. 3.3.3 Root Locus of the Compensated System
and Step Response of the Closed Loop

Though the system has good stability but there is certain steady state error and large overshoot. In order to reduce the peak overshoot to less than 1.5%, the damping ratio is increased from 0.6 to 0.8, and the system controller is obtained

$$G_c(s) = \frac{254(s+5.02)}{s+24.4}$$

The root locus of the compensated system and step response of the closed loop is as shown below;

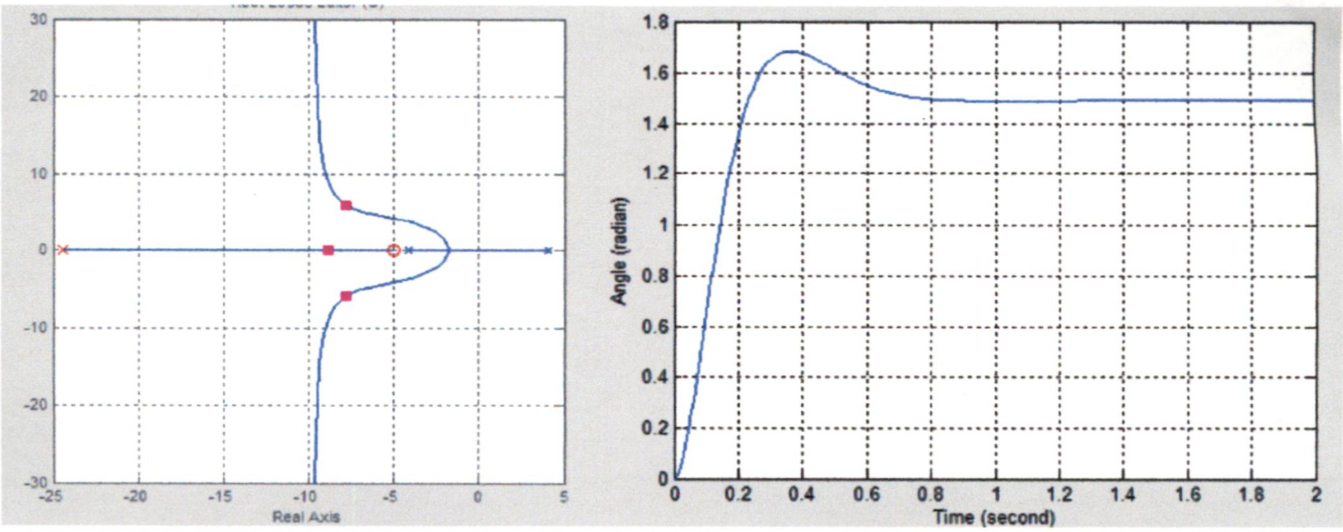

Fig. 3.3.4 Root Locus of the Compensated System

and Step Response of the Closed Loop

An attempt was made to stabilize the inverted pendulum using a double PD controller in cascade configuration. The compensator transfer function worked out as

$$G_c(s) = \frac{338(s+4.8)(s+1.18)}{(s+0.01)(s+28.6)}$$

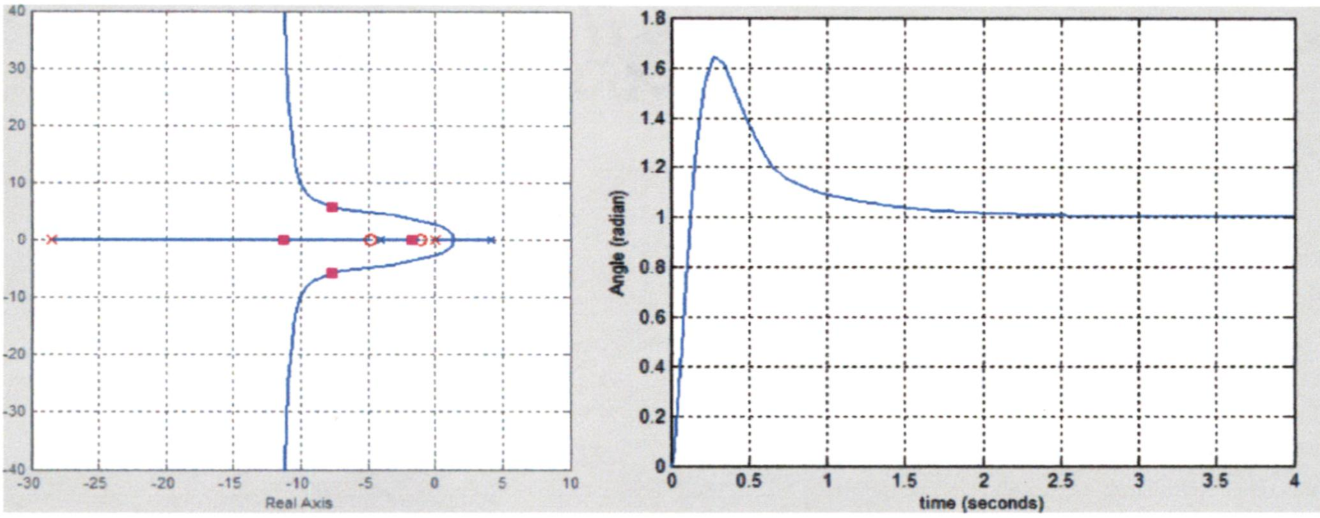

Fig. 3.3.5 Root Locus of the Compensated System (Double PD Controller)

and Step Response of the Closed Loop

As seen from Fig. 3.3.5, the peak overshoot has increased to 62% and the settling time has increased to 2 seconds. Thus it is not meeting both the performance criteria.

Conclusion of Root Locus Analysis

The inverted pendulum being a highly unstable system needs theoretically a PD type controller to compensate. Since one of the poles of the open loop system is on the right half plane, by selecting proper amount of gain with compensation will stabilize the rod in the unstable equilibrium position. The transfer function considered was of the pendulum rod angle and not both rod angle as well as cart position.

3.4 Frequency response analysis

The open loop transfer function of the inverted pendulum is

$$\frac{\Phi(s)}{V(s)} = \frac{0.0366}{0.03721s^2 - 0.62769}$$

Bode plot of the uncompensated system is as shown below

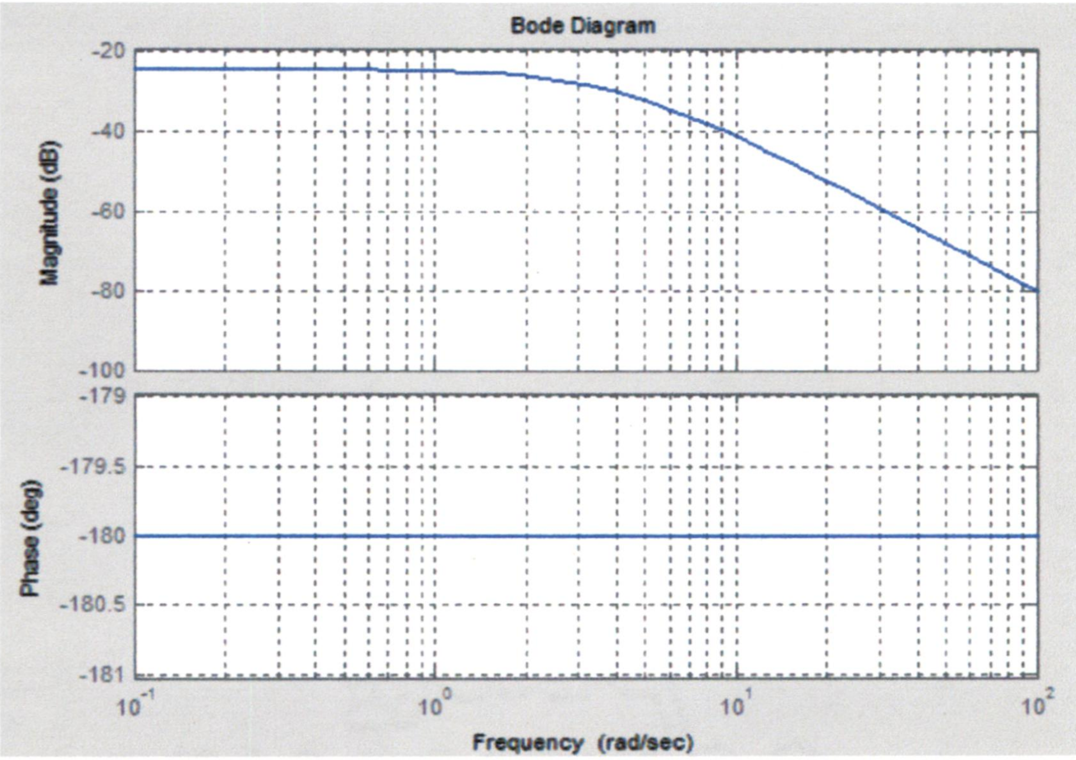

Fig. 3.4.1 Bode Plot of the Uncompensated 1-Stage Linear IP

The system has no zero but two poles, one of which is on the right half of s-plane. According to Nyquist stability criterion, the sufficient and necessary condition of closed loop system to be stable is: when ω change from -∞ to +∞, the open loop transfer function G(j ω) encircle the point -1+j0 p times , in which p is the number of poles of open loop transfer function on the right half of s plane. The open loop transfer function has a pole on right half of s plane, so G(j ω) needs to encircle the point -1+j0 once. From the Fig. 3.4.2, the Nyquist plot does not encircle the point -1+j0 once. So the system is unstable. Further, a controller design is required to stabilize the system.

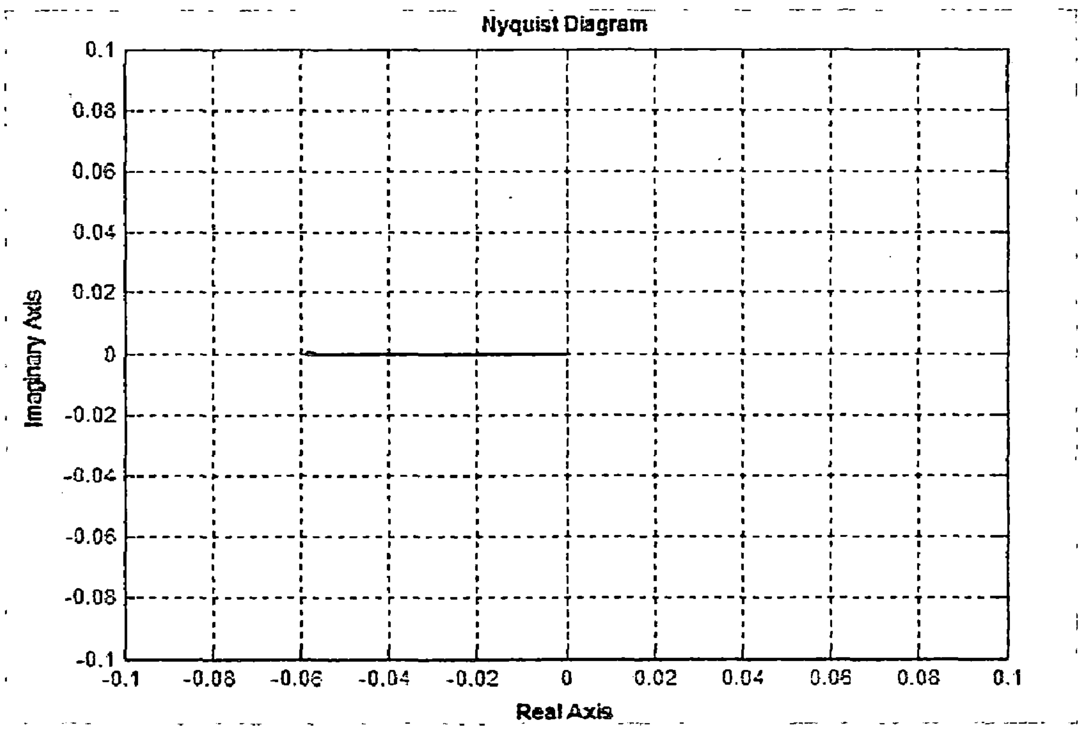

Fig. 3.4.2 Nyquist Plot of The Uncompensated System

The performance requirements are static position constant be 10, phase margin 50°, gain margin larger than or equal to 10 dB.

After going through the controller, it can be established as

$$G_c(s) = \frac{617(s+6.03)}{(s+55.3)}$$

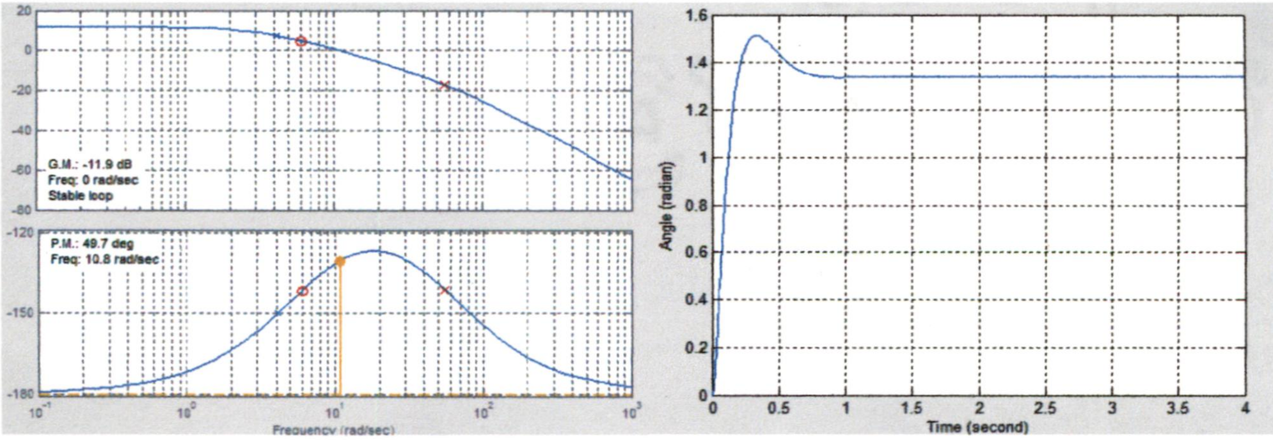

Fig. 3.4.3 Bode Plot of the Compensated System

and Step Response of the Closed Loop

It is observed that there is certain steady state error. To ensure small steady state error, lag-lead controller can be used. This ensures increase in low frequency gain, decrease in steady state error and the system bandwidth as well as stability margins will both increase. The lag-lead controller transfer function can be worked out as

$$G_c(s) = \frac{489(s+0.247)(s+5.22)}{(s+0.000347)(s+44.2)}$$

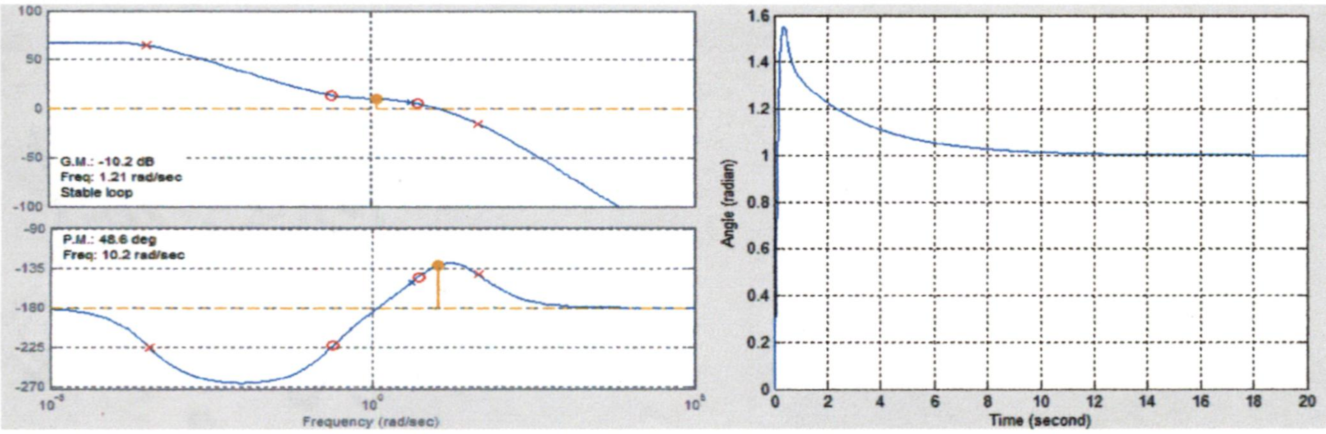

Fig. 3.4.4 Bode Plot of the Compensated System (Double PD Controller)

and step response of the closed loop

Conclusion of Frequency Response Analysis

From the step response Fig. 3.4.4, it is obvious that steady state error has reduced considerably but this is at the cost of increase in peak overshoot, settling time. The controller transfer function so derived cannot stabilize the physical inverted pendulum system as the gain is very high and the given transfer function takes into account the stabilization of pendulum rod angle only and not the cart position.

3.5 PID control analysis

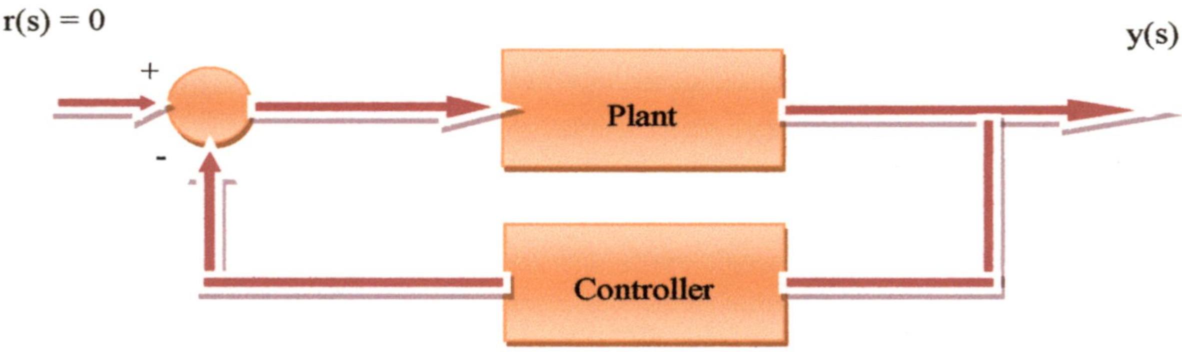

Fig. 3.5.1 Single Stage Linear IP Closed Loop Control System

PID controller transfer function will be

$$G_c(s) = \frac{K_D s^2 + K_p s + K_I}{s}$$

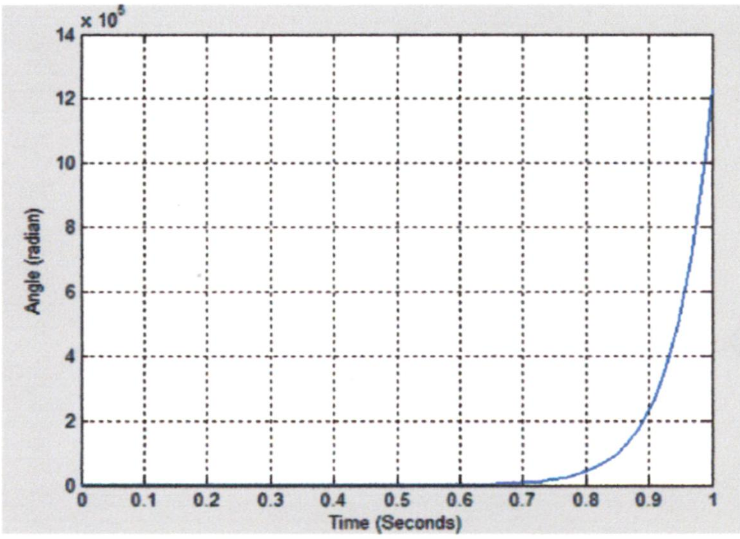

Fig. 3.5.2 Step Response of Open Loop System

The step response of the open loop system alone (without any controller) is depicted above. With increase in proportional gain alone, the step response can be modified to oscillations. By including the derivative gain, the oscillations in the step response of the open loop system dies out and the system exhibits non-zero steady state error. By increasing the integral gain, the steady state error can be reduced considerably but results in increase in settling time. A set of values for proportional, derivative and integral is worked out as 70, 12 and 80 respectively. The step response of the system with PID controller in feedback configuration is as shown below.

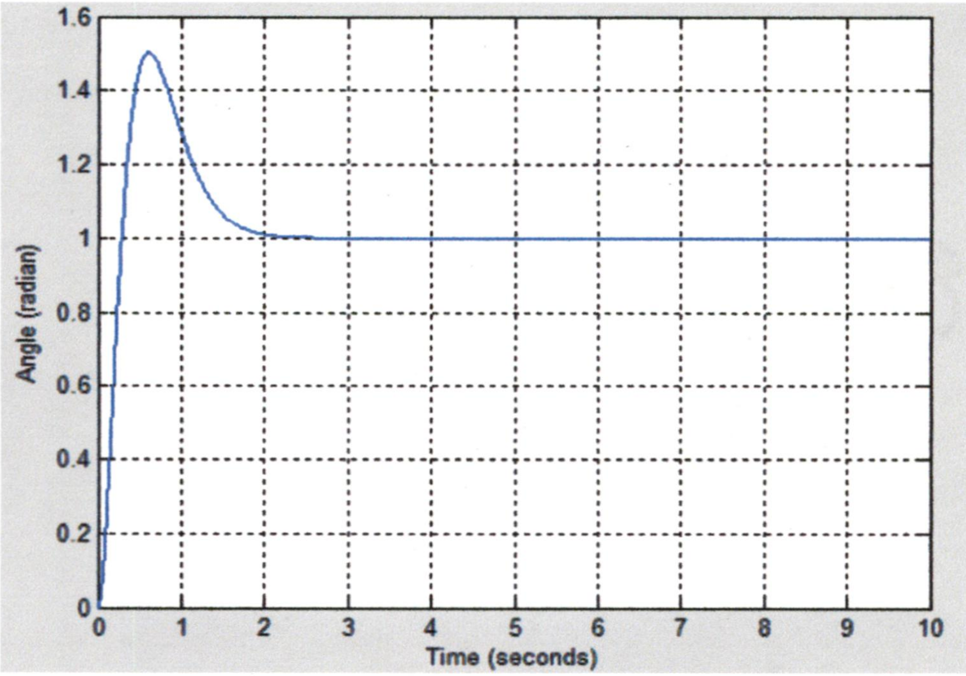

Fig. 3.5.3 Step Response of the System with PID Controller in Feedback Configuration

Conclusion of PID control analysis

The PID controller is SISO (Single Input Single Output) system. It only controls the pendulum rod angle not the cart position. The cart might move towards one direction only. For cart position control we need to design one more controller working independently from the angle controller.

3.6 State Space Analysis

Pole placement method places the closed loop pole of MIMO system to the expected place by designing state feedback controller, thus to satisfy system transient and steady state performance requirement. Designing of controller by pole placement method and its application to 1-stage linear IP system is presented below.

For control system $\dot{X} = AX + Bu$

X	State vector (n dimension)
u	Control vector
A	n x n constant matrix
B	n x 1 constant matrix

Select control signal to be u = -KX and the state feedback gain matrix K is decided by the equation: $K = [\alpha_n - a_n : \alpha_{n-1} - a_{n-1} : \ldots\ldots \alpha_2 - a_2 : \alpha_1 - a_1]T^{-1}$, where, $\alpha_1, \alpha_2, \ldots\ \alpha_n$ are the coefficients of the expected polynomial and a_1, $a_2 \ldots\ a_n$ are coefficients of the characteristic equation. T is a matrix that transforms the state space equations to controllable canonical form. T=MW, where M is the controllability matrix $M = [B: AB: \ldots A^{n-1}B]$ and

$$W = \begin{bmatrix} a_{n-1} & a_{n-2} & \cdots & a_1 & 1 \\ a_{n-2} & a_{n-3} & 1 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ a_1 & 1 & \cdots & 0 & 0 \\ 1 & 0 & \cdots & 0 & 0 \end{bmatrix}.$$

The state space equations using cart acceleration as input are:

$$\begin{bmatrix} \dot{x} \\ \ddot{x} \\ \dot{\phi} \\ \ddot{\phi} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 24.1 & 0 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ \phi \\ \dot{\phi} \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \\ 0 \\ 2.5 \end{bmatrix} u'$$

$$y=\begin{bmatrix}x\\ \phi\end{bmatrix}=\begin{bmatrix}1&0&0&0\\0&0&1&0\end{bmatrix}\begin{bmatrix}x\\ \dot{x}\\ \phi\\ \dot{\phi}\end{bmatrix}+\begin{bmatrix}0\\0\end{bmatrix}u$$

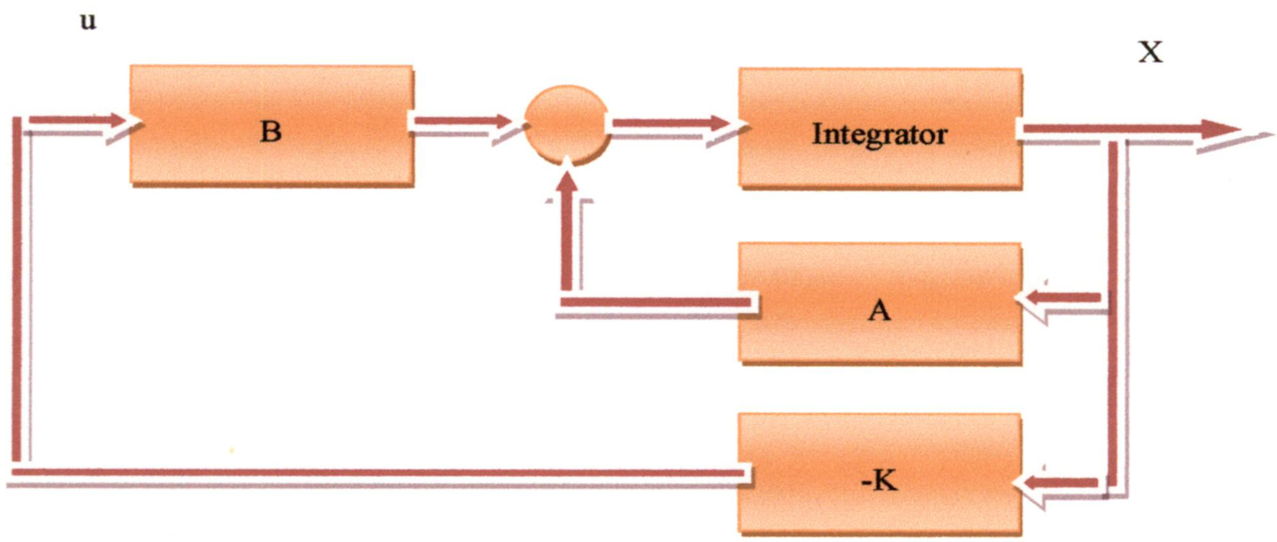

Fig. 3.6.1 State Feedback Close Loop Control Diagram

The controller is to be designed for a system settling time to be about 3 seconds and the damping ratio is about 0.5. In the case of inverted pendulum, cart position, cart displacement, angular position and angular displacement are considered as state variables as is evident in the above matrix-differential equations. Using pole placement technique, the state feedback gain matrix K can be computed as K = [-66.3485 -29.8755 95.3394 21.5302] and the control variable becomes $\mu = -KX = 66.3485x + 29.8755\dot{x} - 95.3394\phi - 21.5302\dot{\phi}$

The impulse response of the system using state feedback pole placement technique is as shown below

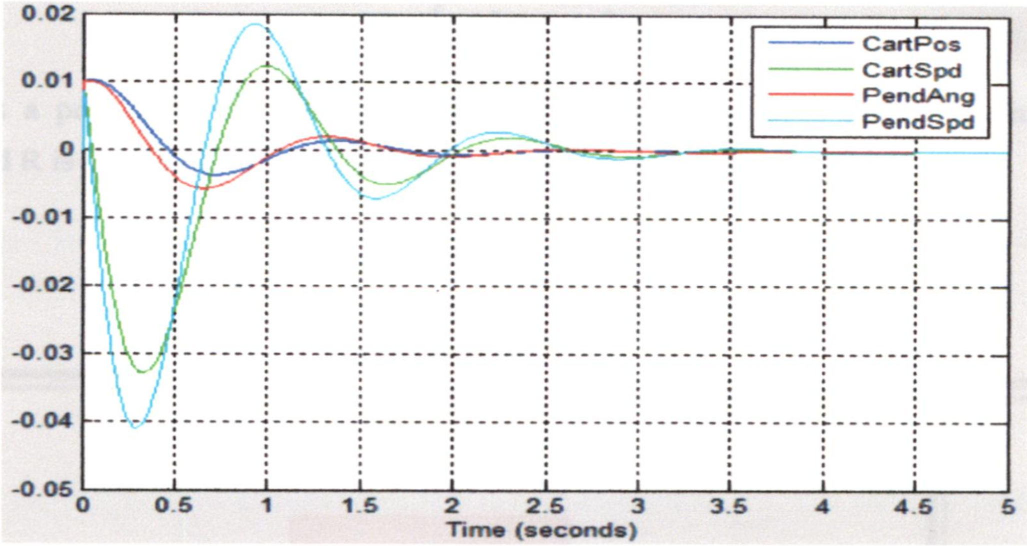

Fig. 3.6.2 Impulse of the 1-Stage Linear IP System
Using State Feedback Pole Placement Technique

Conclusion of state feedback analysis

The state feedback pole placement techniques takes the state value of cart position, cart speed, pendulum rod angle and pendulum angular speed and provides the compensation. In real time control, the 1-stage linear IP system is adjusted manually to make the pendulum rod at π and then the compensation is applied. As shown in the impulse response of the system, the 1-stage linear IP system stabilizes with a marginal cart displacement.

3.7 Linear Quadratic Regulator Optimal control

The LQR optimal control principle is defined by the system equations:

$$\dot{X} = AX + Bu$$

Determine a matrix K that gives the optimal control vector

$$u(t) = -K*x(t)$$

such that the performance index is minimized:

$$J = \int_0^\infty (X^* Q X + u^* R u) dt$$

Q is a positive definite (or semi-positive definite) hermitian or real symmetric matrix and R is a positive definite hermitian or real symmetric matrix.

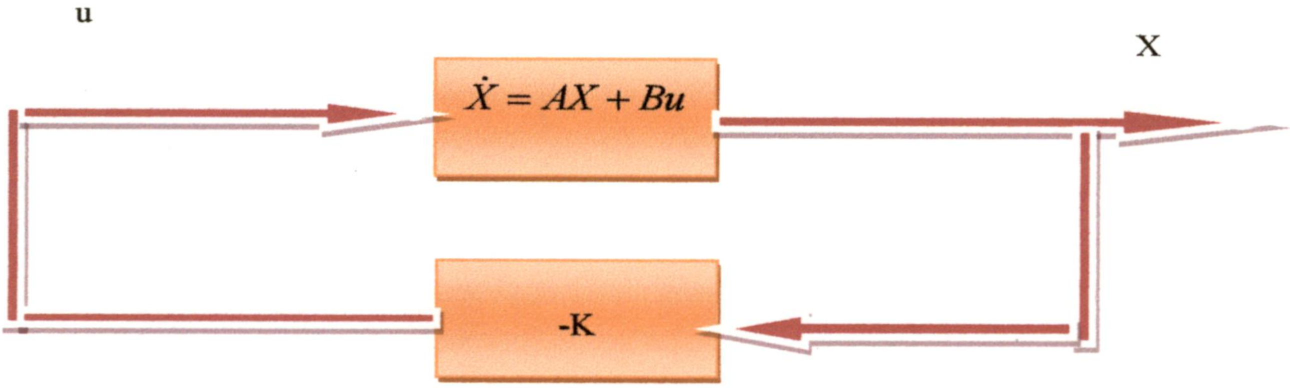

Fig. 3.7.1 Optimal LQR Control Diagram

The 1-stage linear IP system state space equations are:

$$\begin{bmatrix} \dot{x} \\ \ddot{x} \\ \dot{\phi} \\ \ddot{\phi} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 24.1 & 0 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ \phi \\ \dot{\phi} \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \\ 0 \\ 2.5 \end{bmatrix} u'$$

$$y = \begin{bmatrix} x \\ \phi \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ \phi \\ \dot{\phi} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \end{bmatrix} u'$$

Applying linear feedback controller, suppose R is a impulse input acting on the cart and the four state variables $x, \dot{x}, \phi, \dot{\phi}$ represents cart position, cart velocity, pendulum rod angle and pendulum rod velocity respectively. The output $y = [x, \varphi]$ includes cart position and pendulum angle. The objective is to design a controller such that, when acting on an impulse input signal to the system, the pendulum rod will be back to vertically up after oscillation.

$$Q = C' * C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$Q_{1,1}$ is the cart position coefficient and $Q_{3,3}$ is the pendulum rod angle coefficient and the input coefficient R is 1. Using MATLAB command l LQR, the value of K is

K= [-1 -1.8241 25.6773 5.2265]

The LQR control impulse response is depicted in Fig. 3.7.2. It can be observed from the Fig. 3.7.2, closed loop system response overshoot is small, but the settling time is long. Control gain can be increased to decrease the settling time.

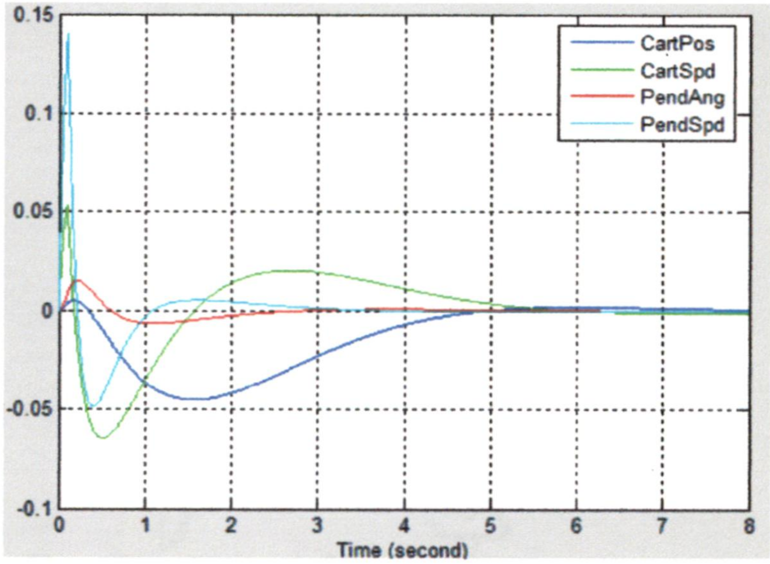

Fig. 3.7.2 LQR Control impulse Response

It can also be observed that in Q matrix, both the settling time and pendulum rod angle movement will decrease as $Q_{1,1}$ increase, Let $Q_{1,1}$ =1000 and $Q_{3,3}$ = 200, then

K=[-31.623 -21.3 78.846 15.80]

The system response will be as follows.

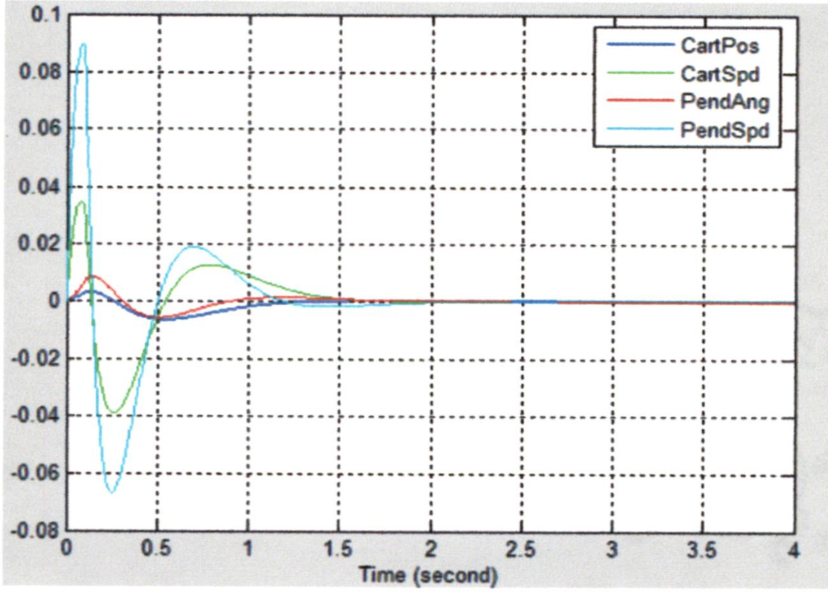

Fig. 3.7.3 LQR Control Impulse Response with Reduced Settling Time and Pendulum Rod Angle Movement

From Fig. 3.6.6, it is obvious that system response time improved dramatically.

Conclusion of LQR Optimal control analysis

In the LQR Optimal control, stabilization of the system is achieved through a definition of performance index. The response can be made faster by increasing $Q_{1,1}$ and $Q_{3,3}$. But for real discrete time control system, large control may result in oscillation or saturation.

Chapter 4

FUZZY LOGIC AND FUZZY CONTROLLER

Conventional control system design depends upon the development of a mathematical description of the system's behavior. This usually involves assumptions being made in relation to the system dynamics and any non-linear behavior that may occur. In cases where assumptions in respect of non-linear behavior cannot be made, the need to describe mathematically, ever increasing complexity becomes difficult and perhaps infeasible.

Fuzzy logic is the application of logic to imprecision and has found application in control system design in the form of Fuzzy Logic Controllers (FLCs). Fuzzy logic controllers facilitate the application of human expert knowledge, gained through experience, intuition or experimentation, to a control problem. Such expert knowledge of a system's behavior and the necessary intervention required to adequately control that behavior is described using imprecise term known as "linguistic variables". The imprecision of linguistic variables reflects the nature of human observation and judgment of objects and events within our environment, and there use in FLCs thus allows the mapping of heuristic, system-related information to actions observed to provide adequate system control. In this way, FLCs obviate the need for complex mathematical descriptions of non-linear behavior to the nth degree and thus offer an alternative method of system control.

4.1 Fuzzy Logic Controller

Fig. 4.1.1 shows the structure of Fuzzy controller. It consists of a preprocessing, fuzzification interface, knowledge base, fuzzy inference system, defuzzification interface and a post processing unit. The preprocessing block transforms the input (e and ė) on the actual universe of discourse (UOD) to the normalized universe of discourse, using the input scaling factors KP, KD and KC for computational simplicity. The fuzzification block converts crisp inputs to appropriate fuzzy sets using the membership functions.

Linguistic Variable, Rule Bases and Membership Functions

Linguistic variables are descriptive terms that might be used, and best understood, by an expert of the system under consideration, which describe the behavior of a system and the applied actions required to control that system. For the FLC in this study, the linguistic variables are based upon the error e(t), and the rate-of-change of error, de/dt.

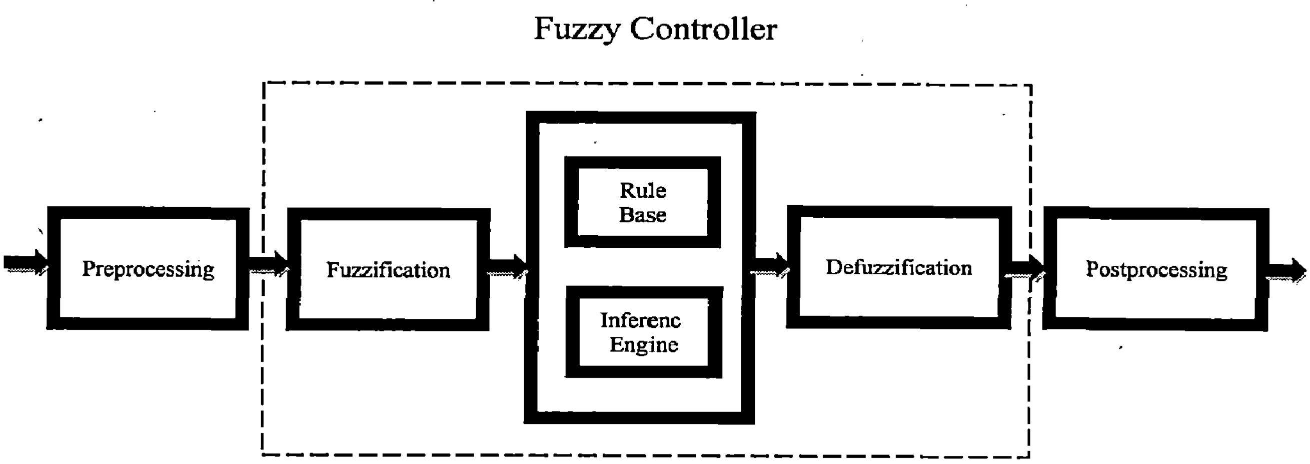

Fig. 4.1.1 Structure of Fuzzy Controller

The rule base of a FLC consists of a set of behavior/action constructs that describe the action to be taken on the occurrence of particular observed/measured system behavior or state. The constructs consist of a premise (i.e. system behavior/state) and the associated consequent (i.e. the action to be taken in order to achieve adequate system control under the observed system behavior/state) used in an 'if premise then consequent' form. Combinations of multiple premises and consequents are possible which enhance the precision of the rule-base. The rule base of a FLC must adequately cover all possible system behavior in respect of applied actions, in order for the FLC to provide reliable system control.

The above descriptions of linguistic variables and rule-bases do not in themselves render the controller 'fuzzy', since, as defined, they could be adequately used in a boolean-based system. What makes the controller 'fuzzy' is the use of membership functions (MFs) to quantify to what degree of certainty each rule is true (i.e. fired) in respect of the system state at any particular time. The 'shapes' and relative spacing of the

MFs form a critical element of the FLC and describe expert understanding of the meaning of the linguistic variables. Typical MF shapes are triangular, trapezoidal, sigmoid or custom-based, with several MFs used to partition the domain of the numeric value under consideration (i.e. the universe of discourse UOD).

The use of MFs ensures that certainty, as defined within a FLC, is based upon the subjective interpretation of an expert rather than upon a probability distribution. Degrees of certainty (i.e. degree of membership of a fuzzy set) range from 0 to 1 in value and hence partial membership is possible. The FLC aggregates the levels of certainty for the entire rule -base to obtain an aggregate fuzzy output set, which is subsequently used to obtain a crisp (i.e. numerically valued), control action. The combination of the rule –base (RB), and associated membership functions (MF), constitute the controller knowledge base (KB), which in effect represents the embedded expert system knowledge. In general, two forms of FLC are defined,

- Mamdani
- Sugeno

Both of these architectures are similar in all respects except for the formulation of the output crisp value. In the Mamdani FLC, the output is formulated using fuzzy sets whereas the Sugeno type FLC uses single -spike output MFs (i.e. singletons) rather than distributed functions.

Fuzzification

This is the process of transforming numeric inputs to fuzzy values. The premise(s) of each rule is evaluated in respect of its degree of membership of the fuzzy sets defined across the range of possible values that the input may assume (i.e. the universe of discourse). For example, Fig. 4.1.2 below shows the MFs for the error input as generated using the MATLAB fuzzy GUI. An error input value of 0.4375 for the position controller, corresponds to a degree of membership of approximately 0.75 for the ZERO fuzzy set and a degree of membership of approximately of 0.25 for the PS (positive small) fuzzy set (i.e. μZERO[e(t)]=0.75 μNS[e(t)]=0.25). Degree of membership of all other fuzzy sets in the universe of discourse for the error, where e(t) = 0. 4375, is zero.

Inference

Having fuzzified the controller inputs, the inference process consists of two phases; **Rule Matching**: The controller evaluates the applicability of each of the rules with respect to the current system state using fuzzy operators (e.g. min). Where a rule contains only a single premise then this stage will return the value obtained from the fuzzification process. FLCs commonly use multiple premises within each rule and therefore the certainty as to what degree the rule as a whole applies to the current system state must be evaluated. To perform the evaluation, the controller applies a logic operator to the fuzzified values of the inputs. Two operators commonly used for the AND conjunction are the minimum and product operators (for OR conjunction, the max operator is commonly used). For the position controller, the min operator was used.

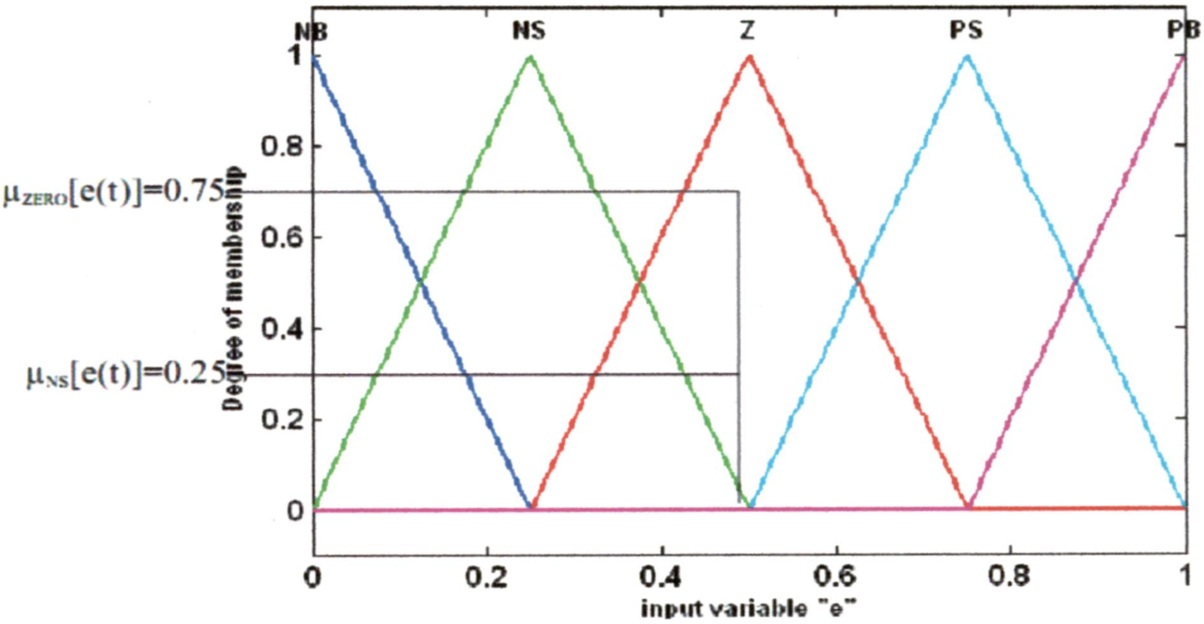

Fig. 4.1.2 Degree of Membership of Z and NS for Input, e = 0. 4375

Implied Conclusions: The consequent of each rule is a fuzzy set, which is truncated in accordance with the degree of certainty that the premise or conjunction of premises, of the rule applies to the current system state. The degree of certainty for the rule is evaluated by matching rules to the current system state using the FLC inputs as is outlined in the previous section. For all rules therefore deemed to be 'fired' (i.e. that apply) an implication operator is applied to the consequent fuzzy set in order to truncate the set relative to the μZERO[e(t)]=0.75 μNS[e(t)]=0.25 degree of firing for the rule. So example above where the min operator was used to evaluate a degree of certainty for the rule to be 0.25, then accordingly the consequent fuzzy set is truncated by this amount.

Rule Base

As stated, the rule-base consists of a set of linguistic variable constructs in the form of;

if premise_1 and/or.... premise_n then consequent1 and/or..... consequent_m which describes the system behavior or states to a level of resolution considered to adequately cover all expected states or behavior and the required actions. The number of rules is dependent upon the number of controller inputs and the number of linguistic variables used to describe those variables. For the position controller in this study, 2 inputs are used with 5 linguistic values to describe the nature of those inputs relative to their universe of discourse, which results in at most $5^2 = 25$ rules. Although in this case, every scenario has an associated entry, it is possible to leave a particular space blank, which would infer that the controller takes no action (i.e. output remains the same as previously).

For systems with 1, 2 or 3 inputs, a tabular form of the rule-base can be constructed. Fig. 4.1.3 illustrates the rule-base used for the heuristic position controller in tabular form.

$\dot{e}$

e		**NB**	**NS**	**Z**	**PB**	**PS**
	NB	NS	NS	NB	NB	Z
	NS	NS	NS	NB	Z	PB
e	**Z**	NB	NB	Z	PB	PB
	PB	NB	Z	PB	PS	PS
	PS	Z	PB	PB	PS	PS

Fig. 4.1.3 Heuristically-Tuned FLC Rule-Base

The rule -base above was arrived at through intuition and trial, using Simulink, and is not necessarily optimal for the system. A feature of the rule-base used is the symmetry across the diagonal. This feature occurs in systems where the physical behavior of the system exhibits symmetry, which is consistent in the case of the cart positioning model used in this study where the surface upon which it travels is even and considered identical in both possible directions of travel. Where systems display such symmetry, obtaining, or optimizing a rule-base may prove quicker if the symmetrical feature can be exploited to some extent.

Defuzzification

The final process of the FLC is to aggregate the fuzzy sets resulting from the inference mechanism to produce a decision (i.e. crisp output), which is the "most certain" in respect of the current system behavior.

A number of methods can be used for defuzzification (e.g. center-average, meanof-maxima), however the most commonly used method is the equation for computation of center-of-gravity (COG), or centroid, which ensures a smooth control action but which requires more complex calculations particularly for non-linear MFs.

4.2 Takagi-Sugeno Fuzzy Systems

This section defines a "functional fuzzy system," of which the Takagi-Sugeno fuzzy system is a special case. For the functional fuzzy system, we use singleton fuzzification, and the i^{th} MISO (Multi Input Single Output) rule has the form

If u_1 is A_1^j and u_2 is A_2^k and,...., and u_n is A_n^l Then $b_i = g_i(.)$

where "." simply represents the argument of the function g_i and the b_i are not output membership function centers.

The consequents of the rules are different, however. Instead of a linguistic term with an associated membership function, in the consequent we use a function $b_i = g_i(.)$ (Hence the name "functional fuzzy system") that does not have an associated membership function. Notice that often the argument of g_i contains the terms u_i, $i = 1, 2, \ldots, n$. but other variables may also be used. The choice of the function depends on the application being considered. Below, discussed linear and affine functions but many others are possible. For instance, may be to choose

$$b_i = g_i(.) = a_{i,0} + a_{i,1}(u_i)^2 + \ldots\ldots + a_{i,n}(u_n)^2$$

Virtually any function can be used (e.g., a neural network mapping or another fuzzy system), which makes the functional fuzzy system very general. For the functional fuzzy system we can use an appropriate operation for representing the premise (e.g., minimum or product), and defuzzification may be obtained using

$$y = \frac{\sum_{i=1}^{R} bi\mu i}{\sum_{i=1}^{R} \mu i}$$

It is assumed that the functional fuzzy system is defined so that no matter what its inputs are, we have

$$\sum_{i=1}^{R} \mu i \neq 0$$

One way to view the functional fuzzy system is as a nonlinear interpolator between the mappings that are defined by the functions in the consequents of the rules.

Chapter 5

NEURAL NETWORK

The key element of this control is the novel structure of the information processing system. It is composed of a large number of highly interconnected processing element (neurons) working in unison to solve specific problems. An artificial neural network (ANN) is configured for a specific application, such as pattern recognition or data classification, through a learning process. Learning is biological systems involves adjustment to synaptic connections that exists between the neurons. This is true of ANN's as well. Neural networks, with their remarkable ability to drive meaning from complicated or imprecise data, can be used to extract patterns and detect trends that are too complex to notice by either human or computer techniques. A trained neural network can be thought of as an "expert" in the category of information it has been given to analyze.

5.1 Biological Neural Network

A biological neuron or a nerve cell consists of synapses, dendrites, the cell body (or hillock), and the axon. The "building blocks" are discussed as follows:

1. The synapses are elementary signal processing devices

   A synapse is a biochemical device, which convert a pre-synaptic electrical signal into a chemical signal and then back into a post-synaptic electrical signal.

   The input pulse train has its amplitude modified by parameters stored in the synapse. The nature of this modification depends on the type of synapse, which can be either inhibitory or excitatory.
2. The postsynaptic signal are aggregated and transferred along the dendrites along the nerve cell body.
3. The cell body generates the output neural signal, which is transferred along the axon to synaptic terminal of other neurons.
4. The frequency of firing of a neuron is proportional to the total synaptic activities and a controlled by the synaptic parameters (weights).

5. The pyramidal cell can receive 104 synaptic input and it can fan-out the output signal to thousands of target cells—a connectivity difficult to achieve in the artificial neural network.

In general the function of the main elements can be given as,

Dendrites	-Receive signal from the other neuron
Soma	-Sums all incoming signals
Axon	-When a particular amount of input is received, then the cell fires. It transmits the signal through axon to other cell.

The fundamental processing element of a neural network is a neuron. This building blocks of human awareness encompasses a few general capabilities. Basically, a biological neuron receives input from other sources, combines them in some way, performs a generally nonlinear operation on the result, and then outputs the final result. Fig. 5.1.1 shows the relationship of these four parts.

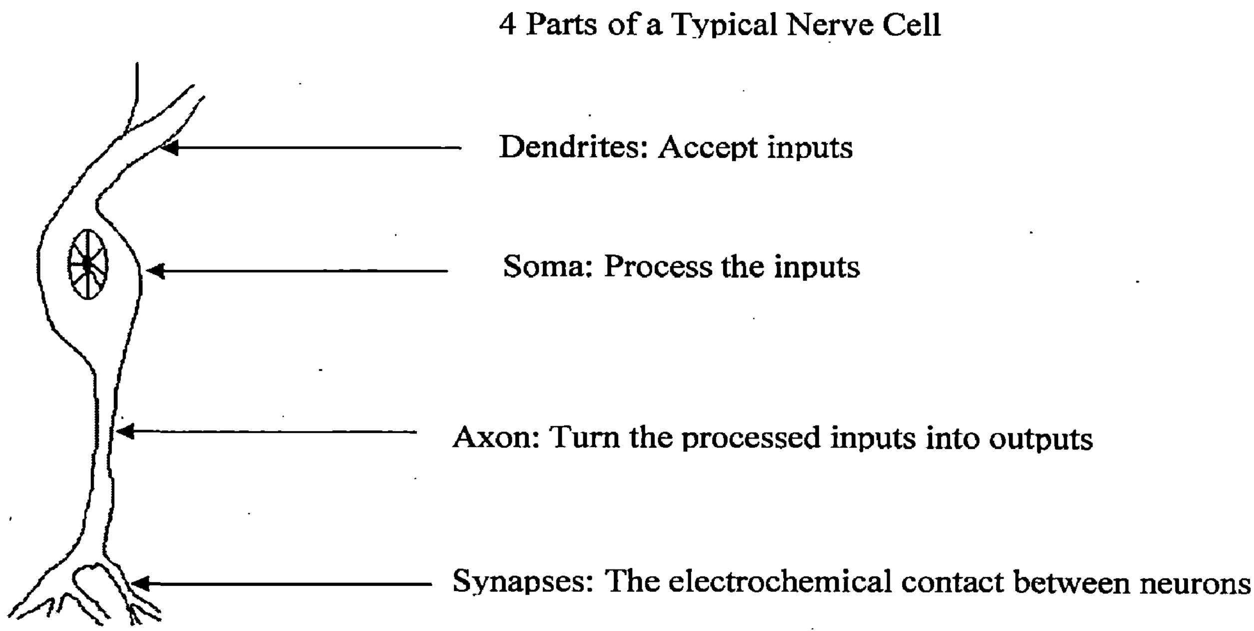

Fig. 5.1.1 A Biological Neuron

The properties of the biological neuron pose some features on the artificial neuron, they are;

1. Signals are received by the processing elements. This element sums the weighted input.
2. The weight at the receiving end has a capability to modify the incoming signal.
3. The neuron fires (transmitted output), when sufficient input is obtained.
4. The output produced from one neuron may be transmitted to other neurons.
5. The processing of information is found to be local.
6. The weight can be modified by experience.
7. Neurotransmitters for the synapse may be excitatory or inhibitory.
8. Both artificial and biological neuron have inbuilt fault tolerance.

Fig. 5.1.2 indicates how the biological neural net is associated with the artificial neural net.

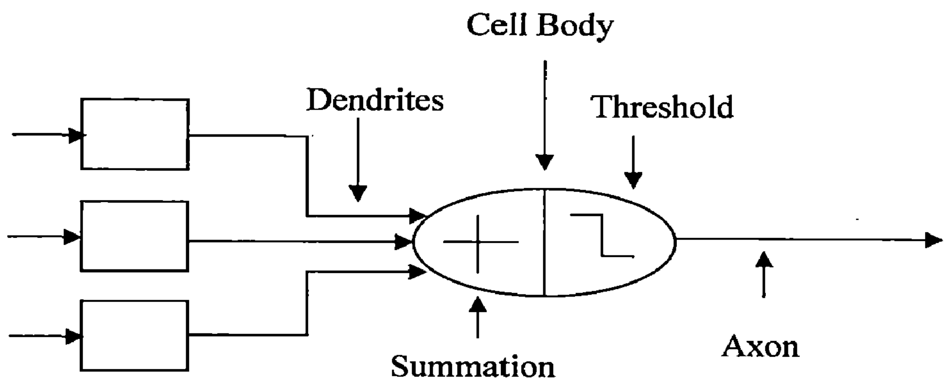

Fig. 5.1.2 Association of Biological Neural Network

5.2 Artificial Neural Network

Artificial neural networks are nonlinear information (signal) processing devices, which are built from interconnected elementary processing device called neurons.

An Artificial Neural Network (ANN) is an information-processing paradigm that is inspired by the way biological nervous systems, such as the brain, process information. The key element of this paradigm is the novel structure of the information processing system. It is composed of a large number of highly interconnected processing elements (neurons) working in union to solve specific problems. ANNs, like people, learn by example. An ANN is configure for a specific application, such as pattern recognition or data classification, through a learning process. Learning in biological systems involves adjustment to eh synaptic connections that exit between the neurons. This is true of ANNs as well.

ANNs are a type of artificial intelligence that attempt to imitate the way a human brain works. Rather than using a digital modal, in which all computations manipulate zeros and ones, a neural network works by creating connections between processing elements, the computer equivalent of neurons. The organization and weight of the connections determine the output.

A neural network is massively parallel-distributed processor that has a natural propensity for storing experimental knowledge and making it available for use. It resembles the brain in two respects:

1. Knowledge is acquired by the network through a learning process, and,
2. Inter-neuron connection strengths known as synaptic weights are used to store the knowledge.

Neural network can also be defined as parameterized computational nonlinear algorithms for (numerical) data/signal/image processing. These algorithms are either implemented on general-purpose computer or are built into a dedicated hardware.

Artificial Neural Networks thus is an information-processing system. In this information-processing system, the elements are called as neurons, process information. The signals are transmitted by means of connection links. The links posses an associated weight, which is multiplied along with the incoming signal (net input) for typical neural net. The output signal is obtained by applying activation to the net input.

An artificial neuron is characterized by:

1. Architecture (connection between neurons)
2. Training or learning (determining weight on the connection)
3. Activation function

The structure of the simple artificial neural network is shown in Fig. 5.2.1.

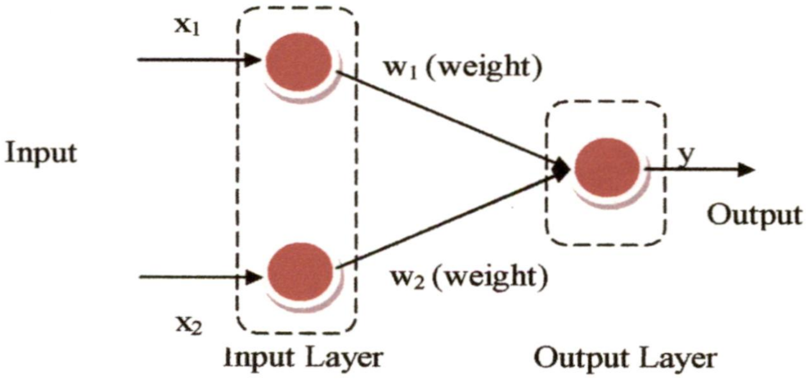

Fig. 5.2.1 A Simple Artificial Neural Net

Fig. 5.2.1 shows a simple artificial neural network with two input neurons (x_1 and x_2) and one output neuron (y). The interconnected weights are given by w_1 and w_2. An artificial neuron is a p-input single-output signal-processing element, which can be thought as a simple modal of a non-branching biological neuron. In Fig. 5.2.1, various input to the network are represented by the mathematical symbol, x(n). Each of these input are multiplied by a connection weight. These weight are represented by w(n). In the simplest case, these products are simply summed, fed through a transfer function to generate a result, and then delivered as output. This process land itself to physical implementation on a large scale in a small package. This electronic implementation is still possible with other network structures, which utilize different summing functions as well as different transfer functions.

5.3 Why Artificial Neural Network

The long course of evolution has given the human brain many desirable characteristics not present in Von Neumann or parallel computers. These include

- Massive parallelism,
- Distributed representation and computation,
- Learning ability,
- Adaptivity,
- Inherent contextual information processing,
- Fault tolerance and
- Low energy consumption

It is hoped that devices based on biological neural network posses some of these desirable characteristics. Modern computers outperform human in the domain of numeric computation and related symbol manipulation. However human can effortlessly solve complex perceptual problems (like recognizing a man in a crowd from a mere glimpse of his face) at such high speed and extent as to dwarf the world's fastest computer. Why is there such a remarkable difference in their performance? The biological neural system architecture is completely different from the Von Neumann architecture. The difference significantly affects the type of functions each computational modal can best perform.

Numerous efforts to develop "intelligent" programs based on Von Neumann's centralized architecture have not resulted in any general-purpose intelligent programs. Inspired by biological neuron networks, ANNs are massively parallel computing systems consisting of an extremely large number of simple processors with many interconnections. ANN modals attempts to use some "organization" principles believed to be used in human brain.

Either human or other computer technique can use neural networks, with their remarkable ability to drive meaning from complicated or imprecise data, to extract patterns and detect tends that are too complex to notice. A trained neural network can be thought as an "expert" in the category of information it hab been given to analyze. This expert than can be used to provide projections given new situations of interest and answer "what if" questions.

Other advantages include:

1. Adaptive learning: An ability to learn how to do task based on the data given for training or initial experience.
2. Self- organization: An ANN can create its own organization or representation of the information it receive during learning rule.
3. Real-time operation: ANN computation may be carried out in parallel, using special hardware devices designed and manufacture to take advantage of this capability.
4. Fault tolerance via redundant information coding: Partial distribution of a network leads to a corresponding degradation of performance. However, some network capabilities may be retained even after major network damage due to this feature.

5.4 Network Architecture

The arrangement of neurons into layers and the patterns of connection within and in-between layer are generally called as the architecture of the net. The neurons within a layer are found to be fully interconnected or not interconnected. The number of layer in the net can be defined to be the number of layers of weighted interconnected links between the particular slabs of neurons. If two layers of interconnected weights are present, then it is found to have hidden layers. There are various type of network architectures: Feed forward, feedback, fully interconnected net, competitive net, etc.

Artificial neural networks (and real neural network for that matter) come in many different shapes and size (see Fig. 5.4.1). In feed forward architectures, the activations of the units are set and then propagated through the network until the values of the output units are determined. The network acts as a vector-valued function taking one vector at the input and returning another vector on the output. For instance, the input vector might represent the characteristic of a bank customer and the output might be a prediction of whether that customer is likely to default on a loan. Or the inputs might represent the characteristic of a gang member and the output might be a prediction of the gang to which that person belongs.

A discussion on some commonly used nets follows.

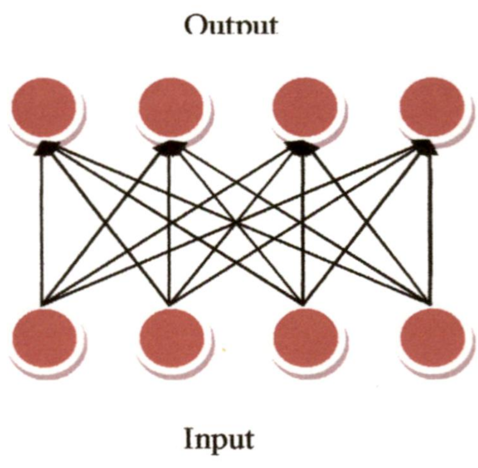

Fig. 5.4.1(a) Single Layer Feedforward

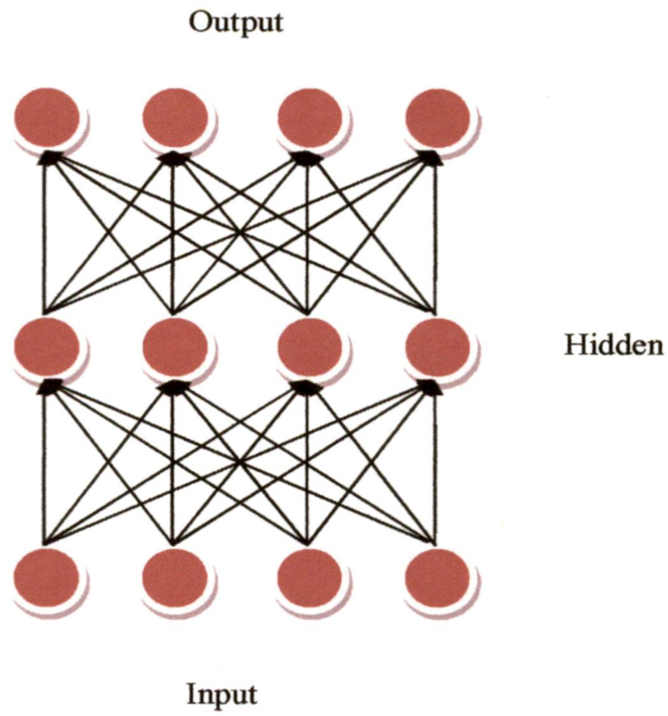

Fig. 5.4.1(b) Multi Layer Feedforward

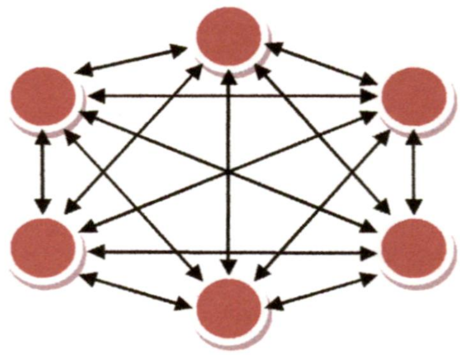

Fig. 5.4.1(c) Fully Recurrent Network

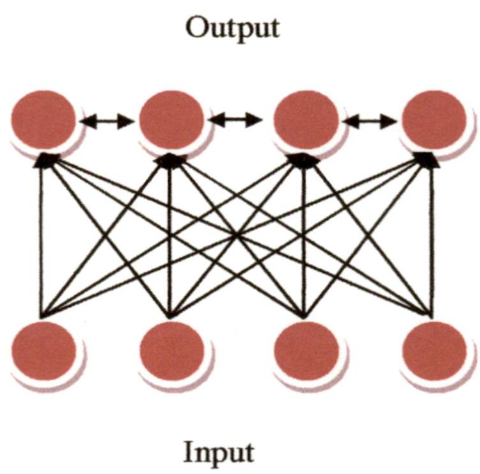

Fig. 5.4.1(d) Competitive Network

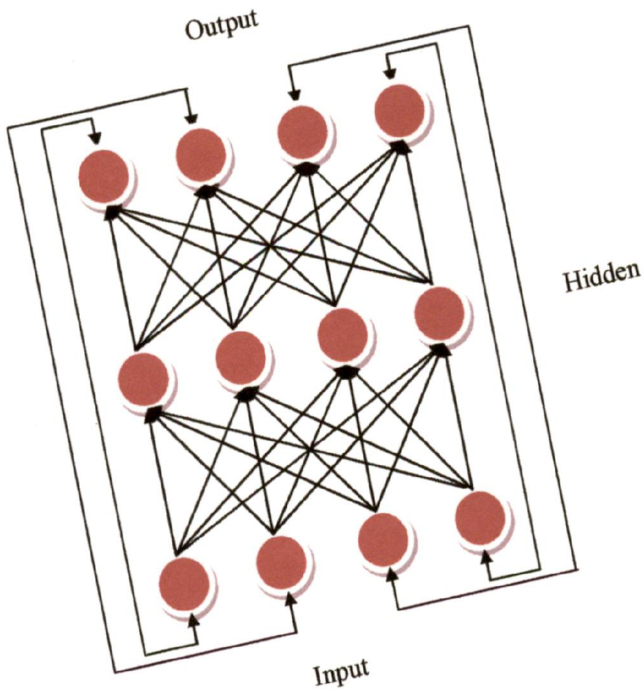

Fig. 5.4.1(e) Jordan Network

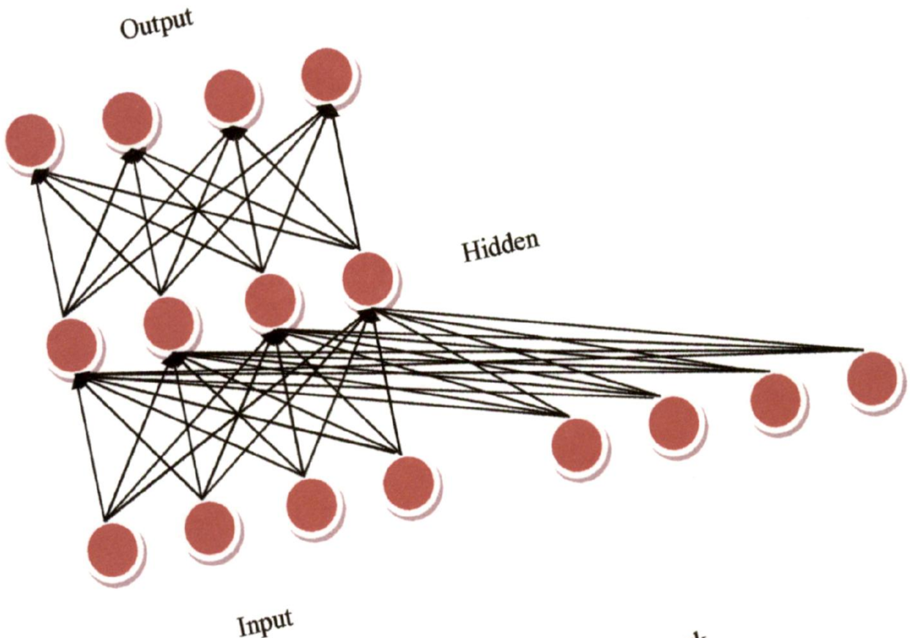

Fig. 5.4.1(f) Simple Recurrent Network

Feed Forward Net

Feed forward networks may have a single layer of weights where the inputs are directly connected to the outputs, or multiple layers with intervening sets of hidden units (see Fig. 5.4.1). Neural network use hidden units to create internal representations of the internal patterns. In fact, it has been shown that given enough hidden units, it is possible to approximate arbitrarily any function with a simple feed forward network. This result has encouraged people to use neural network to solve many kinds of problems.

1. Single Layer net: It is a feed forward net. It has only one layer of weighted interconnections. The input may be connected fully to the output units. But there is a chance that none of the input units respectively. There is also a case where, the input units are connected with other input units and output units with other output units. In a single layer net, the weights from one output unit do not influence the weight for other units.

2. Multi Layer Net: It is also feed forward net i.e., the net where the signals flow from the input units to the output units in a forward direction. The multi-layer net pose one or more layers of nodes between the input and output units. This is advantageous over single layer net in the sense that, it can be used to solve more complicated problems.

Competitive Net

The competitive net is similar to a single-layered feed forward network except that there are connections, usually negative, between the output nodes. Because of these connections the output nodes tend to complete to represent the current input pattern. Sometimes the output layer is completely connected and sometimes the connections are restricted to units that are close to each other (in some neighborhood). With an appropriate algorithm the latter type of network can be made to organize itself topologically. In a topological map, neurons near each other represent similar input patterns. Network of this kind have been used to explain the formation of topological maps that occur in many animal sensory systems including vision, audition, touch and smell.

Recurrent Net

The fully recurrent network is perhaps the simplest of neural network architectures. All units are connected to all other units and every unit is both an input and an output. Typically, a set of patterns is instantiated on all of the units, one at a time. As each patterns is instantiated the weight are modified. When a degraded version of one of the patterns is presented, the network attempts to reconstruct the pattern.

Recurrent networks are also useful in that they allow networks to process sequential information. Processing in recurrent networks depends on the state of the network at the last time step. Consequently, the response to the current input depends on previous inputs. Fig. 5.4.1 shows two such networks: the simple recurrent network and the Jordan network.

5.5 Neurofuzzy function approximation

Consider a standard rule base for a fuzzy proportional controller with the error e as input and a control signal x with singleton membership functions as the output,

If e is Pos then x is 100
If e is Zero then x is 0
If e is Neg then x is -100

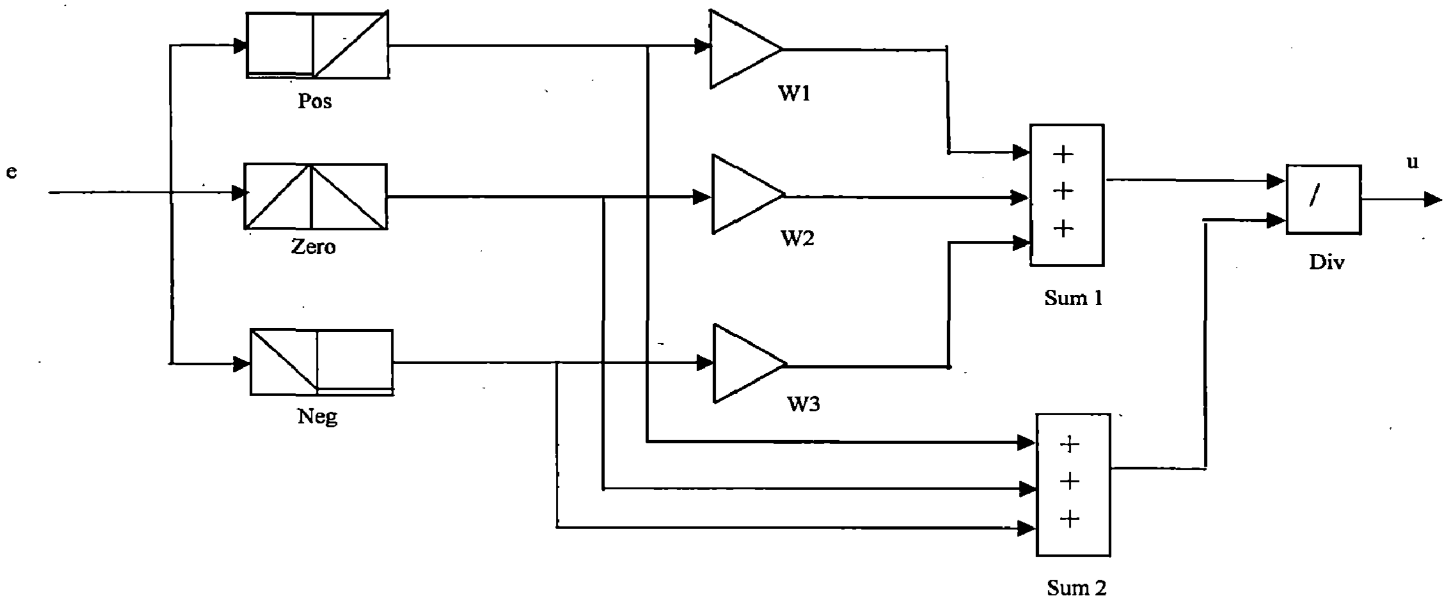

Fig. 5.5.1 Three Rules Perceived as a Network

The inference mechanism can be drawn in a block diagram somewhat like a neural network (Fig. 5.5.1). The network has an input layer, one hidden layer, and one output layer. The input node connects to the neurons in the hidden layer, this corresponds to the if-part of the rules. Each neuron only consists of an activation functions, there is no summation, because each neuron has only one input. The singleton control signals appear as weights on the outputs from the neurons. The one neuron in the output layer, with a rather odd appearance, calculates the weighted average corresponding to the centre of gravity defuzzification in the rule base. The network can be generalized to multi-input-multi-output control, but then the diagram becomes very busy.

Backpropagation applies to this network since all layers are differentiable. Two possibilities for learning are apparent. One is to adjust the weights in the output layer, i.e. all the singletons w_i until the error is minimized. The other is to adjust the shape of the membership functions, provided they are parametric.

The network can be described as a feedforward network with an input layer, a single hidden layer, and an output layer consisting of a single unit. The network performs a nonlinear mapping from the input layer to the hidden layer, followed by a linear mapping from the hidden layer to the output layer.

5.6 Adaptive Neuro Fuzzy Inference System (ANFIS)

The network in Fig. 5.5.1 may be extended by assigning a linear function to the output weight of each neuron,

$$w_k = a_k^T u + b_k, \quad k = 1,2,\ldots,K$$

where $a_k \epsilon R^m$ is a parameter vector and b_k is a scalar parameter. The network is then equivalent to a first order Sugeno type fuzzy rule base (Takagi and Sugeno). The requirements for the radial basis function network to be equivalent to a fuzzy rule base is summarised in the following:-

- Both must use the same aggregation method (weighted average or weighted sum) to derive their overall outputs.
- The number of activation functions must be equal to the number of fuzzy if-then rules.

- When there are several inputs in the rule base, each activation function must be equal to a composite input membership function. One way to achieve this is to employ Gaussian membership functions with the same variance in the rule base, and apply product for the DQG operation. The multiplication of the Gaussian membership functions becomes a multi-dimensional Gaussian radial basis function.
- Corresponding activation functions and fuzzy rules should have the same functions on the output side of the neurons and rules respectively.

If the training data are contained in a small region of the input space, the centres of the neurons in the hidden layer can be concentrated within the region and sparsely cover the remaining area. Thus only a local model will be formed and if the test data lie outside the region, the performance of the network will be poor. On the other hand, if one distributes the basis function centres evenly throughout the input space, the number of neurons depends exponentially on the dimension of the input space.

ANFIS Architecture

Without loss of generality we assume two inputs, u_1 and u_2, and one output, y. Assume for now a first order Sugeno type of rule base with the following two rules

If u_1 is A_1 and u_2 is B_1 then $y_1 = c_{11}u_1 + c_{12}u_2 + c_{10}$

If u_1 is A_2 and u_2 is B_2 then $y_2 = c_{21}u_1 + c_{22}u_2 + c_{20}$

Incidentally, this fuzzy controller could interpolate between two linear controllers depending on the current state. If the firing strengths of the rules are α_1 and α_2 respectively, for two particular values of the inputs u_1 and u_2, then the output is computed as a weighted average

$$y = \frac{\alpha 1 y1 + \alpha 2 y2}{\alpha 1 + \alpha 2} = \bar{\alpha} 1 y1 + \bar{\alpha} 2 y2$$

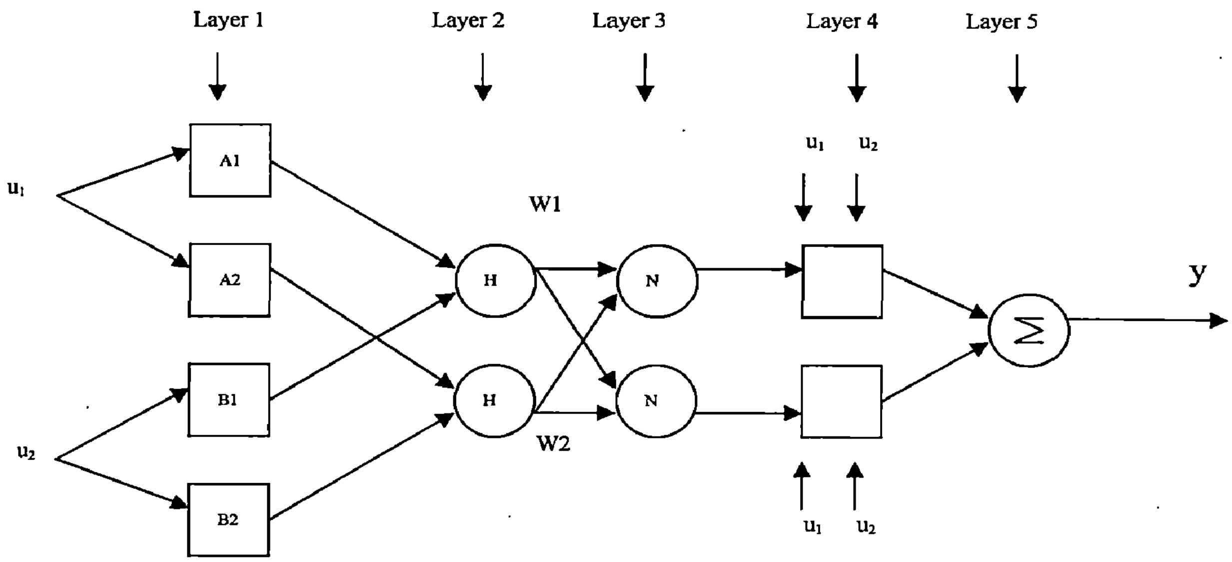

Fig. 5.6.1 Structure of the ANFIS Network

The corresponding ANFIS network is shown in Fig. 5.6.1. A description of the layers in the network follows:-

1. Each neuron i in layer 1 is adaptive with a parametric activation function. Its output is the grade of membership to which the given input satisfies the membership function, i.e., $\mu_{A1}(u_1)$, $\mu_{B1}(u_2)$, $\mu_{A2}(u_1)$ or $\mu_{B1}(u_2)$,An example of a membership function is the generalized Bell function

$$\mu(x) = \frac{1}{1+\left|\frac{x-c}{a}\right|^{2b}}$$

   where {a, b, c} are the parameter set. As the values of the parameters change, the shape of the bell-shaped function varies. Parameters in that layer are called premise parameters.

2. Every node in layer 2 is a fixed node, whose output is the product of all incoming signals. In general, any other fuzzy AND operation can be used. Each node output represents the firing strength α_i of the i^{th} rule.

3. Every node in layer 3 is a fixed node which calculates the ratio of the i^{th} rule's firing strength relative to the sum of all rule's firing strengths,

$$\bar{\alpha}_i = \frac{\alpha i}{\alpha 1 + \alpha 2}, \quad i = 1, 2$$

   The result is a normalized firing strength.

4. Every node in layer 4 is an adaptive node with a node output

$$\bar{\alpha}_i = \bar{\alpha}_i(c_{i1}u_1 + c_{i2}u_2 + c_{i0}), \qquad i = 1, 2$$

where $\bar{\alpha}_i$ is the normalized firing strength from layer 3 and $\{c_{i1}, c_{i2}u_2, c_{i0}\}$ is the parameter set of this node. Parameters in this layer are called consequent parameters.

5. Every node in layer 5 is a fixed node which sums all incoming signals.

It is straight forward to generalize the ANFIS architecture in Fig. 5.6.1 to a rule base with more than two rules.

The ANFIS learning Algorithm

When the premise parameters are fixed, the overall output is a linear combination of the consequent parameters. In symbols, the output y can be written as

$$\begin{aligned} y &= \frac{\alpha 1}{\alpha 1 + \alpha 2} y_1 + \frac{\alpha 2}{\alpha 1 + \alpha 2} y_2 \\ &= \bar{\alpha}_1(c_{11}u_1 + c_{12}u_2 + c_{10}) + \bar{\alpha}_2(c_{21}u_1 + c_{22}u_2 + c_{20}) \\ &= (\bar{\alpha}_1 u_1)c_{11} + (\bar{\alpha}_1 u_2)c_{12} + \bar{\alpha}_1 c_{10} + (\bar{\alpha}_2 u_2)c_{21} + (\bar{\alpha}_2 u_2)c_{22} + \bar{\alpha}_2 c_{20} \end{aligned}$$

which is linear in the consequent parameters c_{ij} ($i = 1, 2$; $j = 0, 1, 2$). A hybrid algorithm adjusts the consequent parameters c_{ij} in a forward pass and the premise parameters $\{a_i, b_i, c_i\}$ in a backward pass. In the forward pass the network inputs propagate forward until layer 4, where the consequent parameters are identified by the least-squares method. In the backward pass, the error signals propagate backwards and the premise parameters are updated by gradient descent.

Because the update rules for the premise and consequent parameters are decoupled in the hybrid learning rule, a computational speedup may be possible by using variants of the gradient method or other optimisation techniques on the premise parameters. Since ANFIS and radial basis function networks (RBFNs) are functionally equivalent under some minor conditions, a variety of learning methods can be used for both of them.

Chapter 6

SIMULATION WORK

The inverted pendulum and ANFIS controller systems modeled are implemented and simulated in the Matlab environment using Simulink and ANFIS editor.

If we control the IP with the help of a fuzzy controller, we need four linguistic variables. How to design a simplest fuzzy controller with reduced no of linguistic variable and hence reduced no of rules with the help of ANFIS without altering the performance of fuzzy controller is explained here.

Following steps required for reducing the linguistic variable:-

1. Design a PD controller for the modal of inverted pendulum; here we need two controllers one for angle and another for position as shown in Fig. 6.1.
2. Replace the PD controller 2 (angle controller) of Fig. 6.1 with a fuzzy controller as shown in Fig. 6.2. Fuzzy controller must have the linear control surface. For linear control surface we take the help of ANFIS because ANFIS tunes the FIS in such a way that the controller surface remains linear. Hence we got controller as below in Fig. 6.2.

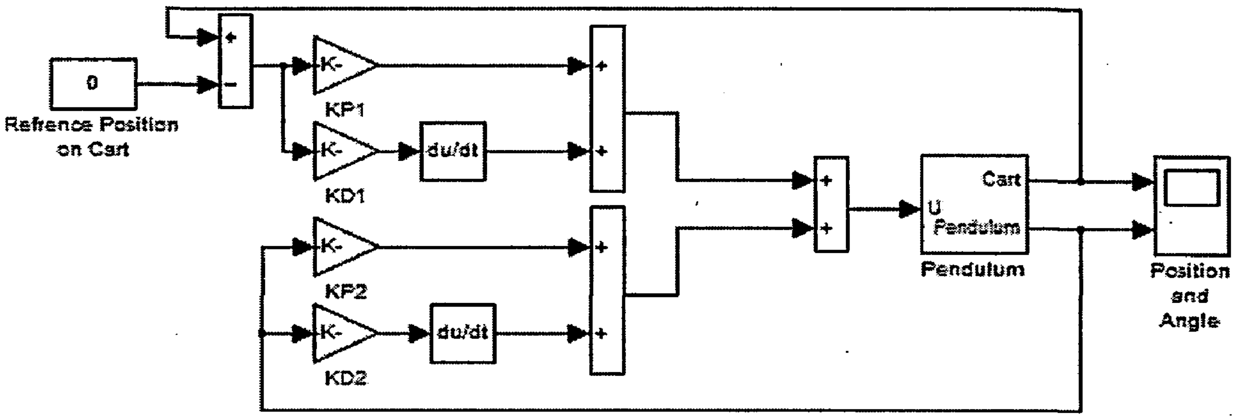

Fig. 6.1 Position and Angle of IP Controlled by Separate PD Controller

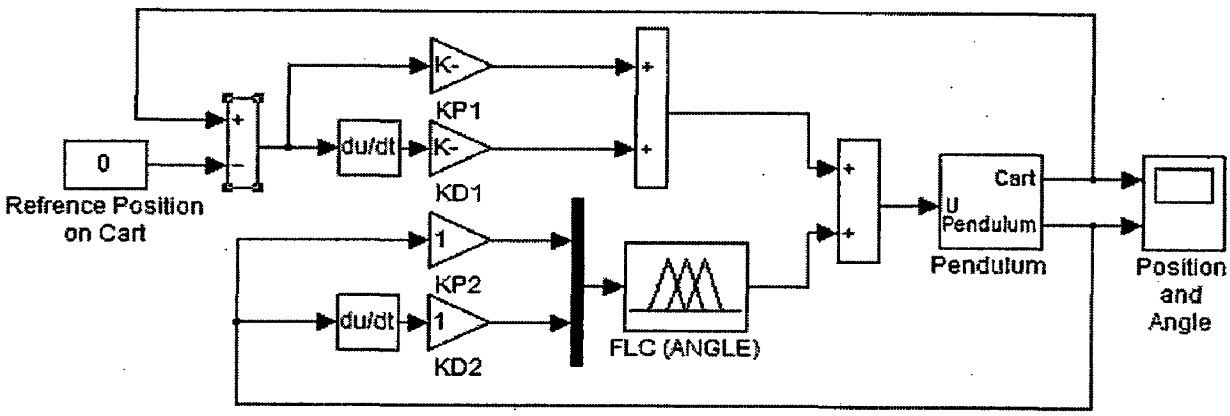

Fig. 6.2 Position Controlled by PD and Angle Controlled by FIS

3. Now replace the PD controller 1 (position controller) of Fig. 6.2 with the FIS tuned for angle in step 2. Tuned by adjusting the gain parameters KP1 and KD1 up to best performance. Now we have a FLC controller as in Fig. 6.3.

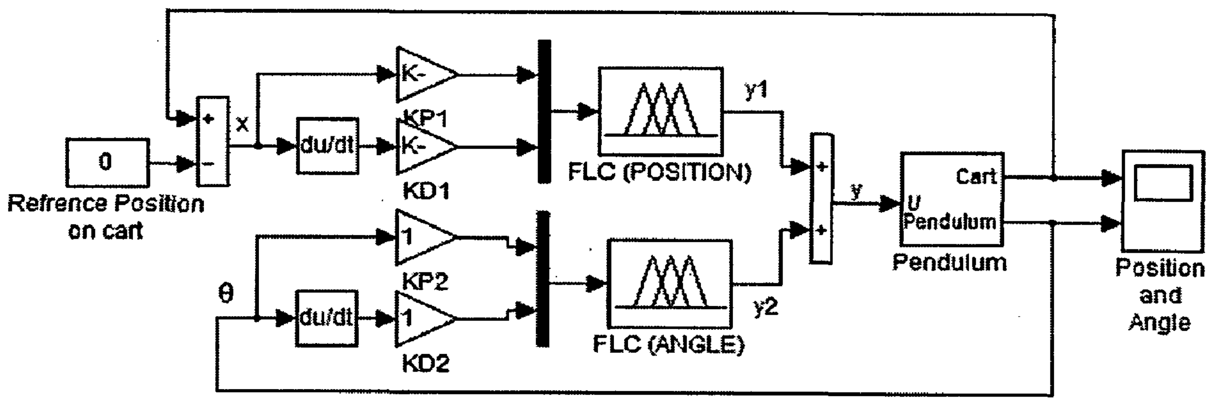

Fig. 6.3 Position and Angle of IP Controlled by Separate Fuzzy Controller

4. From Fig. 6.3 the output of fuzzy controller can be written as follows:-

$$y = y_1 + y_2$$

$$y = f(k_{p1}x, k_{D1}\dot{x}) + f(k_{p2}\theta, k_{d2}\dot{\theta})$$

Here FLC (ANGLE) and FLC (POSITION) both are same and have a linear control surface. So we can rewrite the above equation

$$y = f(k_{p1}x + k_{p2}\theta, k_{D1}\dot{x} + k_{d2}\dot{\theta})$$

As a result we can replace the FIS (ANGLE) and FIS (POSITION) of Fig. 6.3 with a single FIS as shown in Fig. 6.4.

5. With the help of same concept explain in step 3 we can further reduced the two linguistic variable of Fig. 6.4 in two one, as shown in Fig. 6.5.

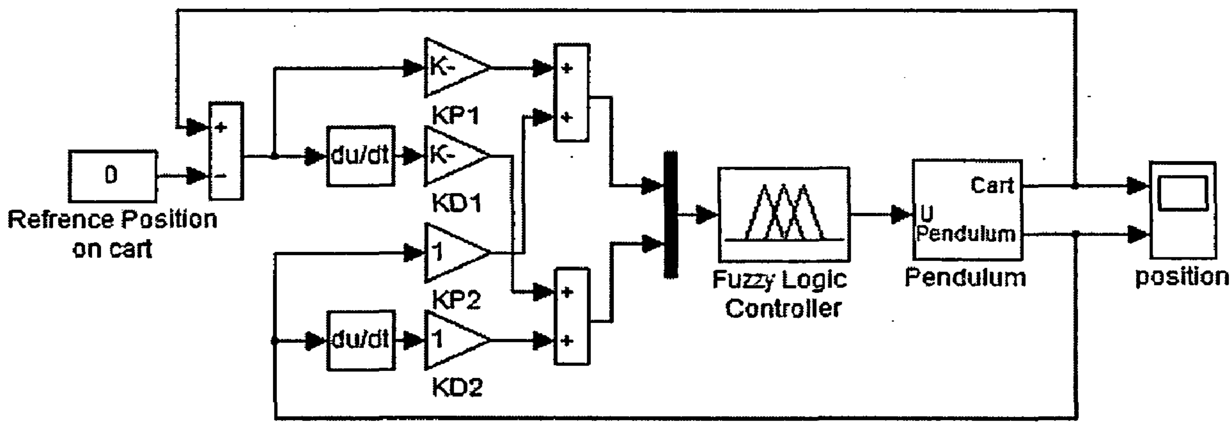

Fig. 6.4 Position and Angle both of IP Controlled by a Common tow Input Linguistic Variable Fuzzy Controller

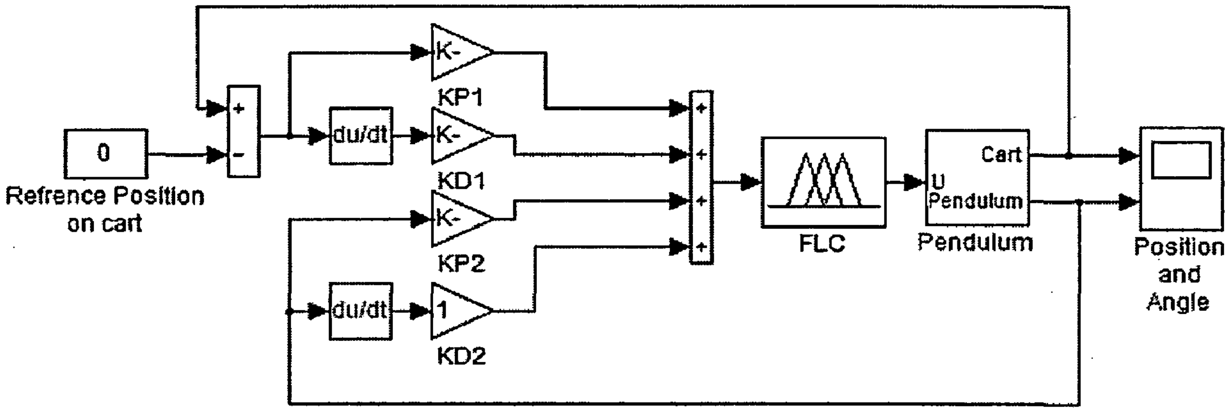

Fig. 6.5 Position and Angle both of IP Controlled by Common Single Input Linguistic Variable Fuzzy Controller

RESULTS

Fig. 6.6(a) and 6.6(b) are the plots of the angle and position of IP with a disturbance as square wave of time period 20 seconds with 50% duty cycle. Fig. 6.7(a) and 6.7(b) are the same with disturbance as a random signal. The origin position of cart on rail is zero and pendulum is balanced upward with angle zero radian. The graph shows that the FLC Reduced Linguistic Variable controller gives smaller overshoot and shorter settling time for both angle as well as reference position.

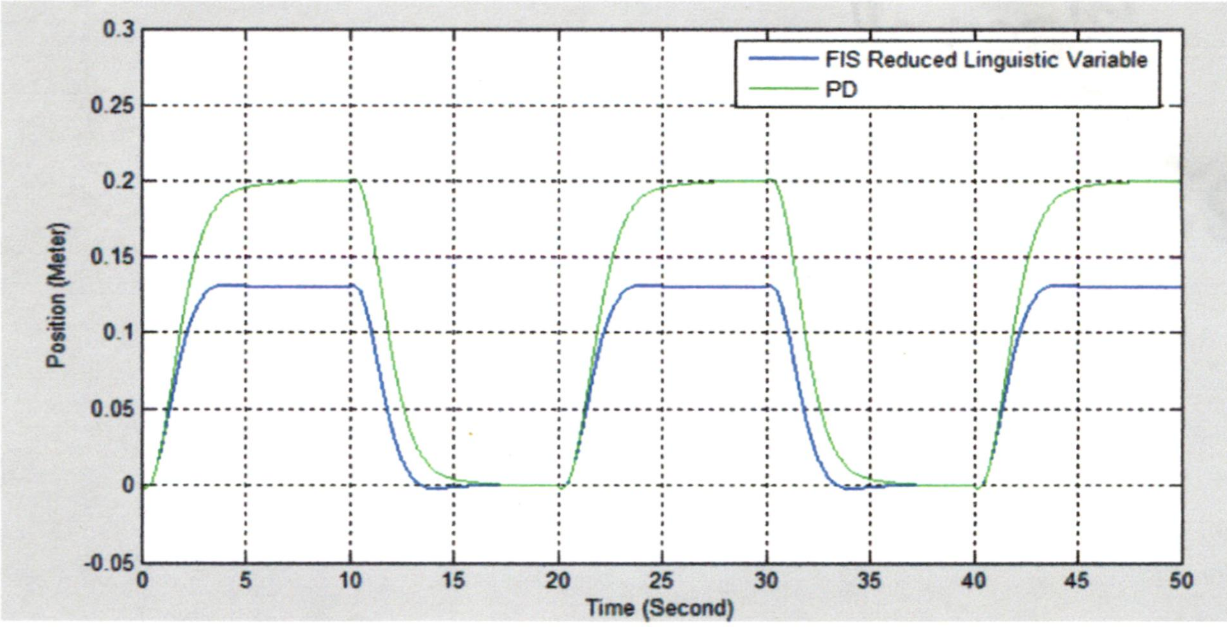

Fig. 6.6(a) Position with Disturbance as a Square Wave of Time Period 20 seconds (Duty Cycle of 10 seconds)

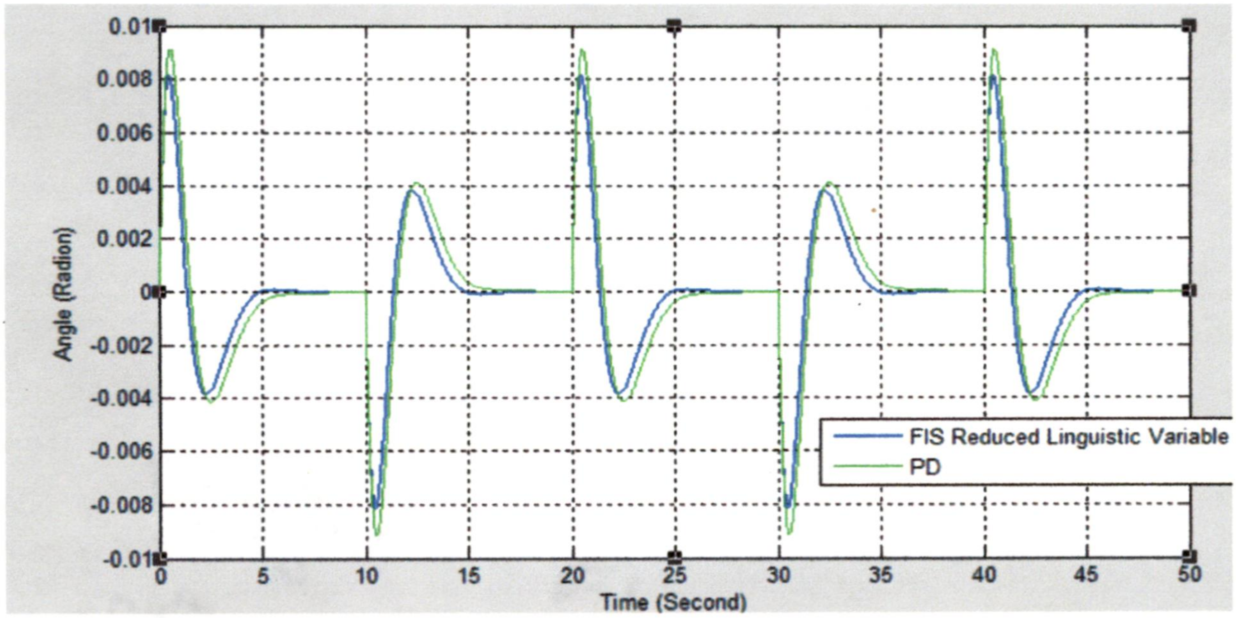

Fig. 6.6(b) Angle with Disturbance as a Square Wave of Time Period 20 seconds (Duty Cycle of 10 seconds)

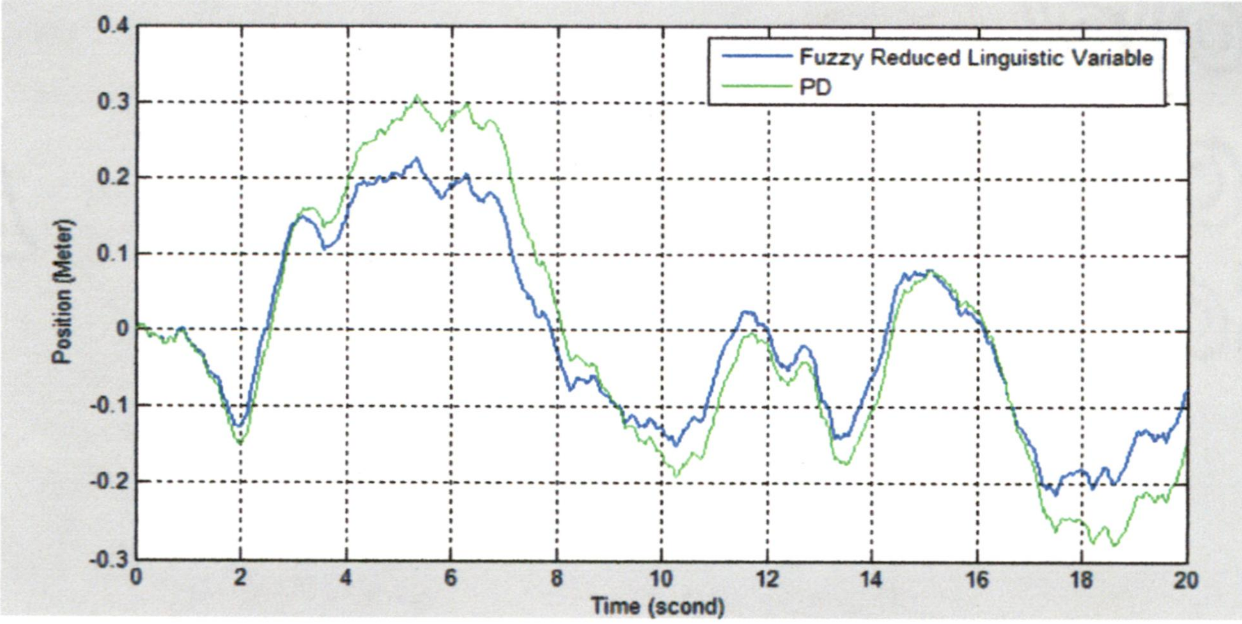

Fig. 6.7(a) Position with Disturbance as a Random Signal

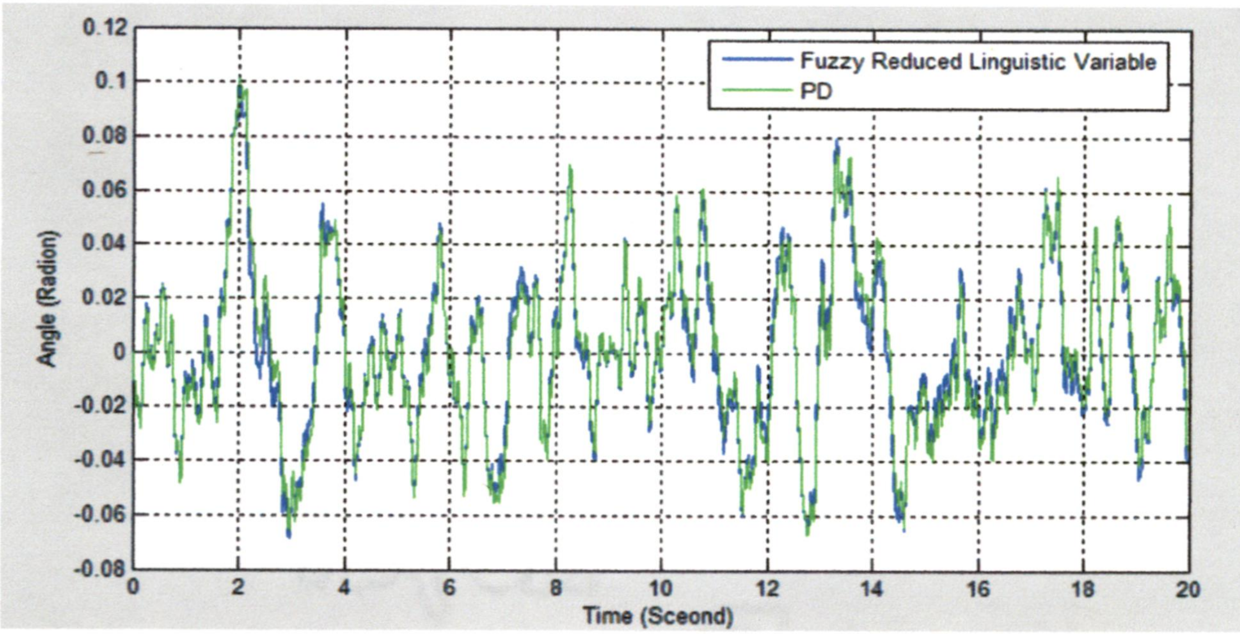

Fig. 6.7(b) Angle with Disturbance as a Random Signal

Fig. 6.8(a) and 6.8(b) are the plots when mass of IP changed to 1.00 kg and mass of cart to 0.95 kg. Initially the IP is in balanced position. After 5 seconds of simulation the reference position is changed from zero to 0.2 meter. The graphs show that the performance of conventional PD controller degrades rapidly, while FLC Reduced Linguistic Variable exhibits small performance degradation due to the parameters change.

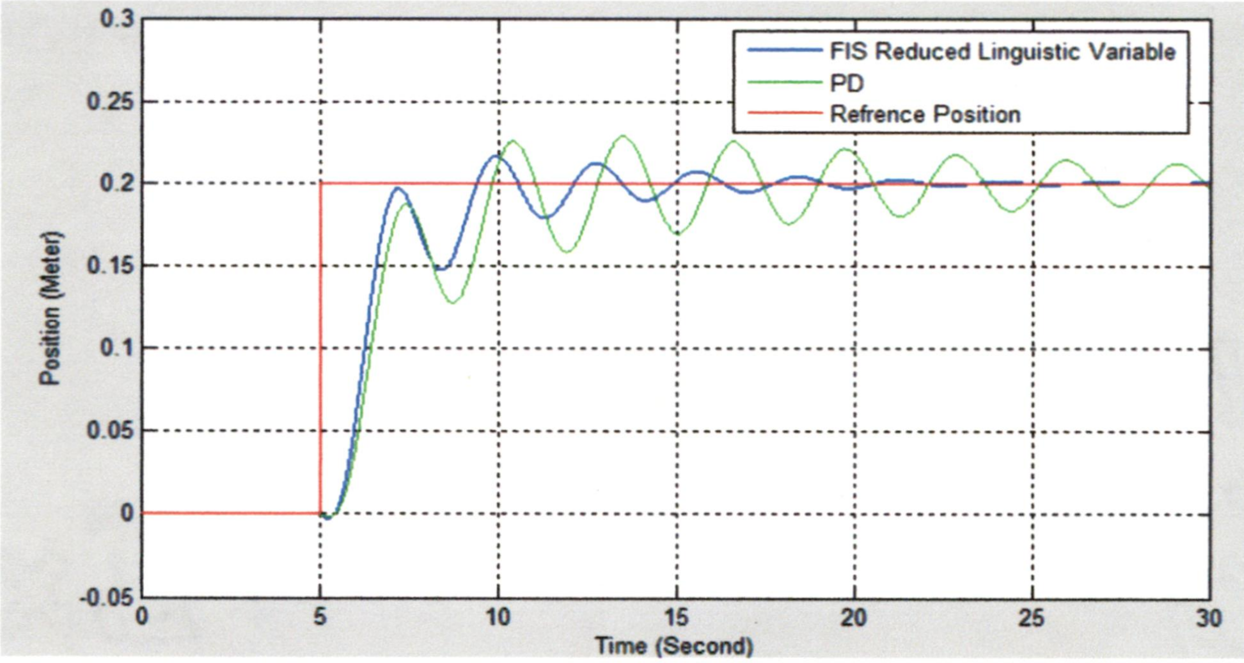

Fig. 6.8(a) Desired Position versus Cart Position Response (Mass of Cart and Pendulum Changed)

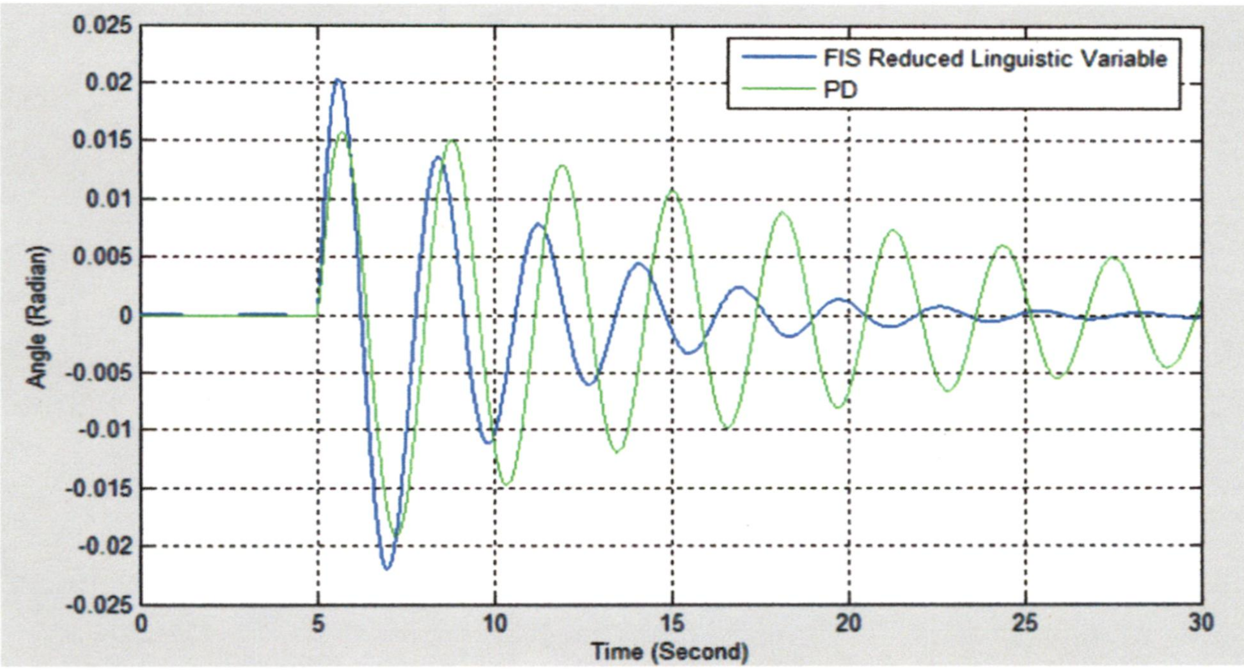

Fig. 6.8(b) Plot of angle (Mass of cart and Pendulum changed)

Chapter 7

CONCLUSION AND FUTURE SCOPE

In this thesis FLC Reduced Linguistic Variable has been implemented in the MATLAB environment, using the ANFIS editor and simulink. It has been used to control an Inverted Pendulum system. Experiments for its performance have been carried out and analyzed. Disturbance rejection and change in the parameters of inverted pendulum are considered. The results are compared with the conventional PD controller. It is shown that the implemented controller has better performance than the conventional PD controller in the presence of external disturbance and IP parameters variation. The implemented FLC controller is a controller with one linguistic variable and two rules, while the existing FLC has four linguistic variables and a numbers of rules, which depends on numbers of membership function. So we conclude that the purposed controller is the simplest FLC for Inverted Pendulum System without degrading the performance of existing FLC tuned by ANFIS.

Further future work is to improve the controller by way of producing the nonlinearity in the control surface, with the help of varying parameters of membership function or varying the shape or modifying the rule written in rule base. Due care must be taken while creating a nonlinear control surface because, created nonlinearity may improve one performance criteria and degrade another one. This method may not work successfully with one input linguistic variable, so two input linguistic variable may be chosen because, more number of input variable means more freedom of tampering with control surface.

The explained method can be implemented to the other multi input linguistic variable FLC also. A further research direction is to implement the reduced linguistic variable FLC in a controller which have more than four input linguistic variables (say 8 or more) and have more than one output.

Research paper published by the author:

[1] **K. K. Dhiman** and R. Mitra, "Fuzzy Logic Controller for Inverted Pendulum with Reduced Linguistic Variable" IFRSA International Journal of Computing (IIJC), vol. 1 (3), pp. 430-435, July 2011,

[2] S. Swarnkar, R. Mitra and **K. K. Dhiman**, "An ANFIS Controller for Ball and Beam System with Real Time Control Application." International Journal of Computer and Electrical Engineering (IJCEE), 2011. (Under review)