

**Analysis of Control of Inverted Pendulum using Adaptive
Neuro Fuzzy System**

A

Thesis

submitted in the partial fulfillment of requirements for the award of the degree of

Master of Engineering

in

Electronic Instrumentation and Control Engineering

to

THAPAR UNIVERSITY

PATIALA (PUNJAB)-147004



Submitted by

MOHIT KR SRIVASTAVA

Regn.No-80751013

Under the esteemed guidance of

MRS. GAGANDEEP KAUR

Lecturer(SG)

Department of Electrical and Instrumentation Engineering

THAPAR UNIVERSITY

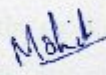
PATIALA (PUNJAB)-147004

June-2009

CERTIFICATE

This is to certify that the thesis titled "Analysis of Control of Inverted Pendulum using Adaptive Neuro Fuzzy System" being submitted by Mohit Kr Srivastava, Regn. No.: 80751013, in partial fulfillment for the requirements of the award of the degree of Master of Engineering in Thapar University, Patiala, is a record of student own work carried out under my supervision and guidance and this thesis has not been submitted to any other university or institute for award of any degree.

Date: 14/07/09


MOHIT KR SRIVASTAVA


Regn No. 80751013

It is certified that the above statement made by the student is correct to the best of our knowledge and belief.

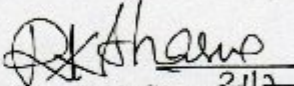

Mrs. GAGANDEEP KAUR

*Lecturer (SG), EIED,
Thapar University
Patiala*

Countersigned By:


Dr. Smarajit Ghosh

*Head of Department
EIED, Thapar University
Patiala*


Dr. R.K. Sharma

*Dean of Academic Affairs
Thapar University
Patiala*

ACKNOWLEDGEMENT

Words are often too less to reveals ones deep regards. An understanding of the work like this is never the outcome of the efforts of a single person. I take this opportunity to express my profound sense of gratitude and respect to all those who helped me through the duration of this thesis. First of all I would like to thank the **all mighty god**, one who has always guided me to work on the right path of the life. Without His grace this would never come to be todays reality. This work would not have been possible without the encouragement and able guidance of my supervisor, **Mrs Gagandeep Kaur** and their enthusiasm and optimism which made this experience both rewarding and enjoyable. Most of the novel ideas and solutions found in this thesis are the result of our numerous stimulating discussions. Their feedback and editorial comments were also invaluable for the writing of this thesis.

No words of thanks are enough for my **dear parents** whose support and care makes me stay on earth. Thanks to be with me.

At the end, I would like to thank all the faculty members of the department and my friends who directly or indirectly helped me in completion of my thesis.

Mohit Kr Srivastava
(Regn No. 80751013)

ABSTRACT

In this thesis modeling and simulation study of basically two control strategies of an inverted pendulum system are presented. The goal here is to determine which control strategy delivers better performance with respect to pendulum's angle and cart's position. The inverted pendulum represents a challenging control problem, which continually moves toward an uncontrolled state. Certain modern techniques are now available because of the development of artificial intelligence and soft computing methods. These methods make the system user friendly and precise control becomes an additional feature. Fuzzy control based learning makes the system more reliable and taking the advantage of precise control at different levels of existence as fuzzy is applicable to non value zero to full value as one, rest of the interim values are also considered. ANFIS is one of an example of fused neuro fuzzy system in which the fuzzy system is incorporated in such a framework which is adaptive in nature. The fuzzy controllers are used to sort out the problem of learning and when the ANFIS is used the adaptability of the system get improves which is based on that learning feature for non-linear system of inverted pendulum model. Learning, training and consideration of the interim levels of the fractional values of variables makes the system sophisticated. Facility of modelling makes the system workable. Modelling is done using Matlab and Simulation study has been done in Simulink in the inverted pendulum case study undertaken. It is found that ANFIS produces better modelling strategy as compared to fuzzy logical controllers (FLC).

TABLE OF CONTENTS

Certificate.....	i
Acknowledgement.....	ii
Abstract.....	iii
Table of Contents.....	iv
List of Figures.....	vii
List of Tables.....	x
List of Abbreviations an Constant Used.....	xi
Literature Survey.....	xiii
Introduction.....	xviii

Chapter 1

Elements of Artificial Intelligent System

1.1	Introduction.....	
1.2	Neural Networks (NN).....	1
1.2.1	Models of Neuron.....	1
1.2.2	Architectures of Neural Networks.....	4
1.2.3	Learning in Neural Networks.....	5
1.2.4	Learning algorithms.....	6
1.2.5	Adaptive networks.....	7
1.2.5.1	Back propagation of Adaptive Networks.....	8
1.2.5.2	Hybrid Learning Rule.....	10
1.3	Fuzzy Logic.....	11
1.3.1	The Fuzzy set.....	11
1.3.2	Linguistic Variables.....	14
1.3.3	Fuzzy if then Rules.....	14
1.3.4	Fuzzy Reasonin.....	15
1.3.5	Fuzzy Systems.....	15

Chapter 2

ANFIS (Adaptive Neuro Fuzzy Inference System)

2.1	Introduction.....	18
2.2	ANFIS architecture.....	18
2.3	ANFIS Learning algorithms.....	21
2.3.1	Hybrid Learning Algorithm – I.....	21
2.3.2	Hybrid Learning Algorithm - II.....	21
2.4	ANFIS as a Universal Aproximator.....	21

Chapter 3

Inverted Pendulum

3.1	Introduction.....	24
3.2	Modeling of Inverted Pendulum.....	24
3.3	Force Analysis and System Equations.....	26
3.4	Transfer Function of the Inverted Pendulum.....	27
3.5	State-Space analysis of Inverted Pendulum.....	28
3.6	Programming tools which are used for control of Inverted Pendulum.....	29
3.6.1	Matlab.....	29
3.6.2	What is a... M-File.....	30
3.6.3	What is Simulink.....	30

Chapter 4

Fuzzy Logic Controller for Inverted Pendulum

4.1	Introduction.....	32
4.2	Planning Of Fuzzy Controllers.....	33
4.2.1	How is FLC different from conventional control methods.....	34
4.2.2	How Does FLC Work.....	34
4.3	Fuzzy of Inverted Pendulum system.....	35
4.4	Inputs and Outputs of Fuzzy Logic Controllers.....	36
4.5	Membership functions of the Inverted Pendulum's FLC.....	37
4.6	Rule base.....	40
4.7	Conclusion.....	44

Chapter 5

Inverted Pendulum with ANFIS

5.1	Introduction.....	45
5.2	Applying ANFIS to Inverted Pendulum.....	46
5.3	Control Strategy.....	49
5.4	Conclusion.....	53

Chapter 6

Results and Discussion

6.1	Open Loop Results.....	54
6.2	Fuzzy Logic Controller Control Method.....	54
	6.2.1 Pendulum's Angle and Cart's Position.....	56
6.3	ANFIS training parameters.....	58
6.4	Statistical Results.....	66
6.5	Graphical Results and Comparison.....	67

Conclusion and Future Work	70
----------------------------------	----

References.....	71
-----------------	----

List of Figure

Figure No.		Page no.
Figure 1.1	Biological Neuron.....	1
Figure 1.2	Mathematical neuron.....	2
Figure 1.3	Basic structure of neural network.....	4
Figure 1.4	Adaptive network.....	7
Figure 1.5	Triangular Membership function.....	12
Figure 1.6	Trapezoidal Membership function.....	13
Figure 1.7	Gaussian Membership function.....	13
Figure 1.8	Membership functions of the term set temperature.....	14
Figure 1.9	Fuzzy Inference Engine.....	16
Figure 1.10	Mamdani Fuzzy Inference System.....	16
Figure 1.11	Takagi-Sugeno Fuzzy Inference System.....	17
Figure 2.1	Basic structure of ANFIS.....	19
Figure 2.2	ANFIS Architecture with nine rules.....	20
Figure 3.1	Free body diagram of the inverted pendulum system.....	25
Figure 3.2	Free body diagrams of the system.....	26
Figure 3.3	Model of Inverted Pendulum in Simulink	31
Figure 4.1	Fuzzy Logic controller overview.....	32
Figure 4.2	Block diagram of fuzzy controller.....	33
Figure 4.3	Block diagram of the system with fuzzy logic controller.....	35
Figure 4.4	Rearranged block diagram of the system with Fuzzy Logic Controller.	36
Figure 4.5	Input 1 of Fuzzy Logic Controller 1.....	38
Figure 4.6	Input 2 of Fuzzy Logic Controller 1.....	38
Figure 4.7	Output of Fuzzy Logic Controller 1.....	39
Figure 4.8	Input 1 of Fuzzy Logic Controller 2.....	39
Figure 4.9	Input 2 of Fuzzy Logic Controller 2.....	40
Figure 4.10	Output of Fuzzy Logic Controller 2.....	40
Figure 4.11	Inverted Pendulum in Simulink.....	42

Figure 4.12	Inverted Pendulum's pole dynamics with cart.....	43
Figure 4.13	Control of Inverted Pendulum with fuzzy controller in Simulink.....	43
Figure 5.1	Block diagram of discrete control system.....	45
Figure 5.2	Block diagram of Inverted Pendulum.....	47
Figure 5.3	Implementation of Block Diagram.....	48
Figure 5.4	Simulink diagram of balance control by ANFIS.....	49
Figure 5.6	Input 1 of TSK for ANFIS implementation of position control.....	50
Figure 5.7	Input 2 of TSK for ANFIS implementation of position control.....	50
Figure 5.8	Output of TSK for ANFIS implementation of position control.....	51
Figure 5.9	Input 1 of TSK for ANFIS implementation of angle control.....	51
Figure 5.10	Input 2 of TSK for ANFIS implementation of angle control.....	52
Figure 5.11	Output of TSK for ANFIS implementation of position control.....	52
Figure 6.1	Open loop pole zero plot of Inverted Pendulum.....	54
Figure 6.2	Open loop impulse response of Inverted Pendulum.....	55
Figure 6.3	Open loop step response of Inverted Pendulum.....	55
Figure 6.4	Root locus plot for open loop inverted pendulum.....	56
Figure 6.5	Step response for pendulum's angle.....	57
Figure 6.6	Step response for cart's position.....	57
Figure 6.7	Training error when ROI is 0.50.....	59
Figure 6.8	Training error when ROI is 0.48.....	59
Figure 6.9	Training error when ROI is 0.46.....	60
Figure 6.10	Training error when ROI is 0.44.....	60
Figure 6.11	Training error when ROI is 0.42.....	61
Figure 6.12	Training error when ROI is 0.41.....	61
Figure 6.13	Training error when ROI is 0.39.....	62
Figure 6.14	Training error when ROI is 0.37.....	62
Figure 6.15	Training error when ROI is 0.32.....	63
Figure 6.16	Training error when ROI is 0.30.....	64
Figure 6.17	Training error when ROI is 0.27.....	64
Figure 6.18	Training error when ROI is 0.23.....	65
Figure 6.19	Training error when ROI is 0.21.....	65

Figure 6.20	Training error when ROI is 0.18.....	66
Figure 6.21	Comparison of output response of cart's position.....	67
Figure 6.22	Comparison of output response of pendulum's angle.....	68

List of Tables

Table 1.1	Different Activation function with their symbols and MATLAB commands.....	3
Table 2.1	Two passes in the hybrid learning algorithm for ANFIS.....	21
Table 4.1	Inputs and outputs of FLC.....	36
Table 4.2	Standard labels of quantization.....	37
Table 4.3	Fuzzy rule matrix for control position FLC.....	41
Table 4.4	Fuzzy rule matrix for control position FLC.....	41
Table 6.1	ANFI'S parameters for position control.....	58
Table 6.2	ANFI'S parameters for angle control.....	63
Table 6.3	Statistical results for position and angle control of IP.....	66
Table 6.4	Comparison of output response of carts's position.....	67
Table 6.5	Comparison of output response of pendulum's angle.....	68

List of Abbreviations and Constants Used

Symbols	Descriptions
b	Friction on cart
l	Length to pendulum centre of mass
..	Cart position Coordinate
..	Cart Acceleration
..	Pendulum angle from the vertical
..	Pendulum Angular Acceleration
r(s)	Reference Signal
e(s)	Error signal
u(s)	Plant input
g(s)	Plant
y(s)	Output pendulum's angle
F(s)	Force
x	State vector
u	Input vector
y	Output vector
	Settling time
	Rising time
	Steady state error
M	Mass of the Cart (900 gm= 0.9 in Kg)
m	Mass of the Pendulum (100 gm =0.1 in Kg)
L _p	Length of Pendulum (47 cm=0.47 in m)
l	Length of pendulum to Center of ravity(23.5 cm=0.235 in m)
i	Moment of Inertia of Pendulum(5.3 gm-m ² =0.0053 in Kg.m ²)
r	Radius of Pulley(2.3 cm = 0.023 in m)

$\tau(\tau)$	Time Constant of Motor (0.5 seconds)
K_f	Gain of Feedback ($9/\pi$ V/rad/sec = 2.8648 in V/rad/sec)
K_m	Gain of Motor (17 rad/sec/V = 17 in rad/sec/V)
b	Friction of the Cart (0.000 N/m/sec = 0 in N/m/sec)
g	Acceleration due to Gravity = 9.8 m/sec^2

Literature Survey

Fuzzy set theory was first presented by Zadeh [1] in his seminal paper in 1965. Fuzzy logic was developed later from fuzzy set theory to reason with uncertain and vague information and to represent knowledge in an operationally powerful form. The main reason to develop fuzzy logic from fuzzy set theory was to build a conceptual framework for linguistically represented knowledge. Fuzzy logic is an extension of multi-valued logics. It offers the base for approximate reasoning with uncertain and vague propositions.

The fuzzy logic control application theory spread out in middle of 70's when Mamdani [2-4] in his research papers proposed a fuzzy logic in process control, in which fuzzy set theory are used for the processing by a computer, and the obtained information can be used in fuzzy control algorithm to control a process without having an explicit model of the process.

The mathematical approximation to Fuzzy Logic was given by Kosko and Wang[11, 12] in their research paper in which they proved that the Fuzzy system are universal approximators and such system can be used for control analysis, in which approximation modeling is done.

Schwartz and Piero [13, 14] provided a mathematical foundation in fuzzy logical system including approximation reasoning. This concept opened a new way in control application problems which revolutionized the market mainly in the field of consumer electronic and electrical product.

By the advent of 1990 hybrid system architectures was considered in attention of the researchers which included fuzzy system and neural network system all together as a single system in which both system best features are employed. Most of the approaches was based on Takagi Sugeno [5] model which is a generalization of fuzzy logic.

The idea of combining fuzzy logic and neural network was given by Gupta and Ching [9]. They had described a simple way to the learning of fuzzy logic control system either

offline or online. The learning and the adaptability of the newly formed system was much rewarded.

Ayoubi [15] described a method in an algorithm form to train the fuzzy network using Back Propagation (BP) which requires a desired set of training input output data. These data set modelling parameters are obtained online or offline through simulations.

Arabshahai [20] introduced the concept of acceleration of back propagation using previous described fuzzy method. The Fuzzy Logic Controller is trained in order to intimate the input and output data set for the model. His research work included the optimized use of hybrid system using back-propagation.

Ollenu [21] used the input output linearization method in which outer loop comprised of adaptive fuzzy network. The fuzzy output is used as a supporting input to the adaptive generated output and is used for cancelling the effect of disturbances. The fuzzy structure is determined adaptively in his work. He also described that the liner control feedback law which is used in the inner loop. The approach is very much restricted and is very difficult to modify for other applications. This approach is not used in the application for the non linear system which are feedback linearizable.

Tanuka [16] in his research work used fuzzy identified model , which is a replaced with the plant dynamics. Now the back propagated error is used for training the controller of the system for the attainment of best performance.

Spooner [17] in his paper presented adaptive control technique for non linear system using Lyapunov theory and fuzzy system. The adaptability of the system using this approach has not given the desired performance and results while considering non linear plant application. His work was restricted to feedback linearizable plants only.

Watanabe [18] used fuzzy gaussian neural network for control of mobile robot in which delta rule i.e. back propagation is used for training the robot. But no systematic work of back propagated error is presented for the plant in his work study.

Adaptive Neuro Fuzzy Inference system (ANFIS) was proposed by Shing and Jang [10] in which the training of network was given by temporal back-propagation and least square simultaneously. Temporal backpropagation was originally proposed by Derrick [8]. For the error back-propagation simplified difference equation of the model of plant is used. Extending this approach Shing proposed an error back-propagation by calculation of the Jacobian of system's model.

Lawrence Bush [24] described the application of a Fuzzy Logic Controller (FLC) for the inverted pendulum problem. FLCs have been used to control many dynamic systems. The inverted pendulum represents a challenging control problem, which continually moves toward an uncontrolled state. Three versions of an FLC for the inverted pendulum problem were discussed in his paper including their strengths, weaknesses and performance.

Dario Maravall [23] has proved an excellent effect of vertical force as regards the stabilization of the inverted pendulum (IP) and has demonstrated how the fuzzy control design methodology can be used to construct a hybrid fuzzy control system that incorporating Takagi–Sugeno fuzzy control structure for stabilization of the Inverted Pendulum via a vertical force.

Teng, F.C. [27] in his study showed that the Matlab Simulink based software packages are suitable for the design of real-time computer control experiments. The operating environment, operation procedure and limitations of the software namely, WinCon 2.0, Real Time Toolbox 3.0 and xPC Target 1.0 are discussed. Application of the software is demonstrated through the real-time implementation of PID, LQR and fuzzy logic controllers. Sample implementations and some experimental results obtained from the

real-time control of a moving cart, single link inverted pendulum, self-erecting inverted pendulum, double-link inverted pendulum and pole balancing mobile robot are presented.

Mohanlal and P.P. Kaimal, [25] revealed a method of exact modeling and optimal control of the inverted pendulum on a cart, which is a benchmark nonlinear dynamic system. Conventionally, the TSK (Takagi-Sugeno-Kang) fuzzy modeling blends local linear models to represent a nonlinear system, which in general does not exactly represent the nonlinear system under consideration.

Jung, Park, Suck Gyu Lee [19] discussed a synthesized method of control inputs to control both position and angle of the inverted pendulum simultaneously. As well known, the tradeoffs in control between position and angle make it difficult to control them simultaneously. To resolve the control conflict between position and angle, his paper introduced the new weighting factors describing contribution degree of the control inputs. To verify the validity of the proposed algorithm, computer simulations as well as experiments with prototypes of the inverted pendulum system are conducted.

Yamakita, M. Hoshino, and T. Furuta [22] discussed inverted pendulums which is a typical examples of unstable systems and have been used for the verification of designed control systems. In particular, the control of the systems has been known as a good example to show the power of modern control theory. They showed two kinds of experimental system using inverted pendulums. One is state transfer control of a double pendulum (TITech double Pendulum) and the other is stabilization and hand-over control of a single inverted pendulum in 3D space by visual feedback using two manipulators. In the former experiment, one can demonstrate a standard stabilizing regulation method and nonlinear control techniques. In the latter one, a feedback stabilization and reliable control by non-contact measurement can be made.

Changsheng Yu and Li Xu [26] described hardware of an inverted pendulum system with low-cost and high reliability and stability, meanwhile, a novel expert fuzzy control scheme and the MATALB-based real-time control approach was proposed by him. The

number of fuzzy control rules was significantly reduced by combining related variables, and the control performance was greatly improved by adopting suitable rule-base according to the angle of the pole with the vertical. The proposed control scheme was implemented in MATLAB and showed good performance in the real-time fuzzy control of the developed inverted pendulum.

Keith [29] had done a complete analysis of this system using the Routh Criterion, by multiplying out the characteristic equation as a polynomial of s and studying the coefficients. Although correct, his approach is unnecessarily abstruse. The system is the ideal example root-locus analysis. Siebert explains that a physical interpretation for the need for this integrator arises from the fact that we are using a voltage-controlled motor using force as output variable. Without the integrator a constant angular error only achieves a constant cart velocity, which is not enough to make the pendulum upright. In order to get underneath" the pendulum, the cart must be accelerated therefore, we need the integrator.

Jang [23] in his research paper described an innovative approach to design the fuzzy controller without restoring to the domain knowledge about the plants which are to be controlled. The fuzzy controller which he considered was implemented by the help of ANFIS while the plant can be implemented either as a neural network or as an ANFIS. He further showed that the plant can be best implemented as an adaptive network if the difference equation of the plant dynamics is available. He also showed that, fuzzy controller can evolve automatically in which desired membership function is acquired using if then rule base to reach at desired goal , which is based on the learning rule of the adaptive network.

L Rajaji , C. Kumar and M Vasudevan [30] in the combined innovative approach developed the usage of Fuzzy Logic and ANFIS for the application of higher power performance of soft starter fed induction motor drive for blowers, fans, pumps etc. in which speed is adjusted by adjusting firing angle of the appropriate thyristors for any given operating torque and load.

INTRODUCTION

In the Neuro-Fuzzy systems the neural networks (NN) are incorporated in fuzzy systems, which can use fuzzy knowledge automatically by learning algorithms of NNs. They can be visualized as a mixture of local experts. Adaptive Neuro-Fuzzy inference system (ANFIS) is one of the of Neuro Fuzzy systems in which a fuzzy system is incorporated in the framework of adaptive networks. ANFIS constructs an input-output mapping based both on human knowledge i.e. rules and on generated input-output data pairs.

A fuzzy system provides a framework for taking into account the flexible nature of variables, and representation of incomplete information. This is one of the alternatives to classical approach and is based on the concept that, humans think using linguistic terms such as “much” or “more” and others rather than numbers. The concept is described in natural language i.e. linguistic variables, by L. A. Zadeh using fuzzy sets introduced by him in 1965. The fuzzy systems are based on conditional if-then rules, which use fuzzy sets as linguistic terms or variables in antecedent and conclusion parts. A collection of these fuzzy if then rules can be found out from human experts or alternatively they can be generated from observed data. The main advantage of such fuzzy systems is the easiness to interpret the knowledge in the rule base.

In most of the fuzzy systems, fuzzy if-then rules are obtained from an expert. However, this methodology of knowledge acquisition has great disadvantages; because not every expert wants to share his/her knowledge. For this reason, NNs were incorporated into fuzzy systems, which can acquire knowledge automatically by learning algorithms of NNs. These systems are called Neuro- Fuzzy systems and have advantages over fuzzy systems, i.e., acquired knowledge is very easy to understand. Like in NNs, knowledge is saved in connection weights, but the knowledge can also be easily interpreted as fuzzy if then rules. The Neuro- Fuzzy systems can be visualized as a mixture of local experts and their parameters are updated using gradient descent and least squares optimization methods. The most frequently used NNs in Neuro-Fuzzy systems are radial basis function networks (RBFN). Their popularity is due to the simplicity of structure, well-established theoretical basis and faster learning than in other types of NNs. Adaptive network based fuzzy inference system or adaptive Neuro-Fuzzy inference system

(ANFIS), first proposed by Jang (1993), is one of the examples of RBFN Neuro Fuzzy systems in which a fuzzy inference system is implemented in the framework of adaptive networks. ANFIS forms an input output mapping based both on human knowledge (based on fuzzy if then rules) and on generated input output data pairs by using a hybrid algorithm that is the combination of the gradient descent and least square estimates. ANFIS is not a black box model and optimization techniques can be well applied to it, which is computationally efficient and is also well suited to mathematical analysis. Therefore, it can be used in modeling and controlling studies, and also for estimation purposes also.

The inverted pendulum presents a very good example for control engineers to prove a modern control theory. This can be determined by the facts that the inverted pendulum is marginally stable, in control sense having distinctive time variant mathematical model. Inverted Pendulum is a good model for, an automatic aircraft landing system, the attitude control of a space booster rocket and a satellite , stabilization of a cabin in a ship, aircraft stabilization in the turbulent air-flow etc. To solve such problem with non-linearity and time variant system, there are alternatives such as a real time computer simulation of these equations or linearization. The inverted pendulum is a highly nonlinear and open-loop unstable system. This means that standard linear techniques cannot be modeled as the nonlinear dynamics of the system. On simulating the system the pendulum falls over quickly. The characteristics of the inverted pendulum make identification and control more challenging. The inverted pendulum is an intriguing subject from the control point of view due to their intrinsic 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. 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. Moreover, application of these control techniques to a two or three stage inverted pendulum may

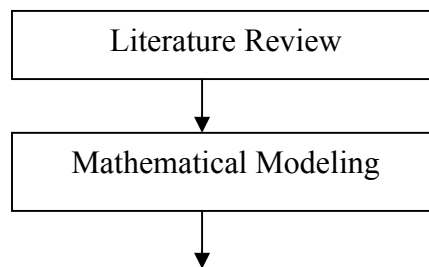
result in a very critical design of control parameters and difficult stabilization. However, using artificial intelligence controllers such as artificial neural network and fuzzy logic controllers, the controller can be design without require the model to be linearized. The non-linearized model can be simulated directly using the Matlab application to see result. Here we are considering the the Artificial Intelligence aspect for the inverted pendulum i.e. using the help of fuzzy controllers by application of ANFIS.

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

1. A progressive model can be built. 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 insightful measure of the ability of the method to provide control.
3. The cart/pole system is a common test case for fuzzy logic so any results can be compared to previous work in the field. In order to perform sound criticism of any controllers developed, a reference model must be designed at the outset of this work.

Research Methodology

The following flow chart best describes the outline of thesis and the research methodology used.



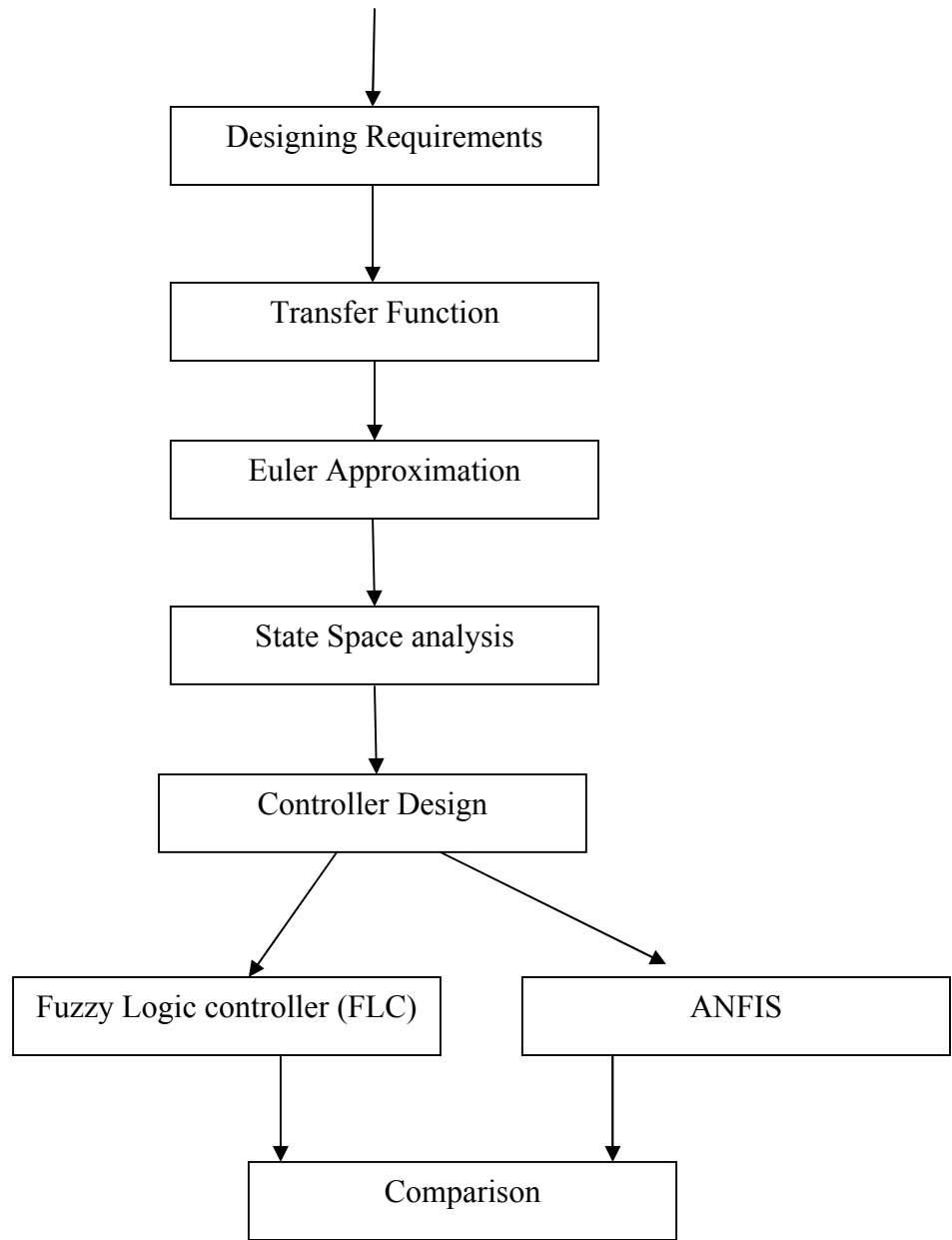


Figure Flow chart of Research Methodology

CHAPTER 1

ELEMNTS OF ARTIFICIAL INTELLIGENCE SYSTEMS

1.1 Introduction

Basic characteristics of NNs will be summarized in this section. First, a neuron model and architecture of NN will be described. After that, learning in NNs will be explained. Also, adaptive network will be given as an example of NN with back propagation and hybrid learning algorithms.

1.2 Neural Networks (NN)

1.2.1 Models of Neuron

A neuron is a special nervous cell in organisms, which have electric activity. These cells are mainly intended for the operation of the organism. The biological neuron is shown schematically in Figure 1.1. A neuron consists of a cell body, which is surrounded by a membrane. The neuron has dendrites and axons, which are its inputs and outputs of neuron. Axons of neurons join to dendrites of other neurons by forming synaptic contacts (synapses).

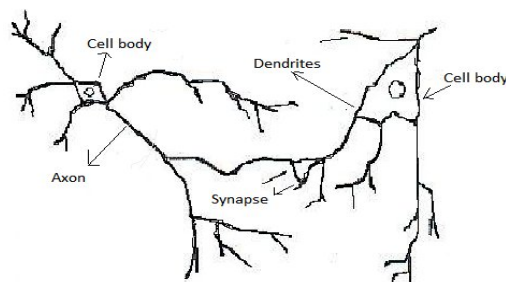


Figure 1.1 Biological Neuron

Input signals of the dendrite tree are weighted and added in the cell body and formed in the axon, where the output signal is generated. The signal's intensity, consequently, is a function of a weighted sum of the input signal. The output is passed through the branches of the axon and reaches the synapses. Through the synapses the signal is transformed into a new input signal for neighbor neurons. The input signal can be either positive or negative (exciting or inhibiting), depending on the synapses (Aliev 2001). In accordance with the biological model, different mathematical models were suggested. The mathematical model of the neuron, which is usually utilized under the simulation of NN, can be shown in Figure 1.2. The neuron receives a set of input signals x_1, x_2, \dots, x_n (vector X) which are usually the output signals of other neurons. Each input signal is multiplied to a corresponding connection weight, w , and analogue of the synapse's efficiency. Weighted input signals come to the summation module corresponding to cell body, where their algebraic summation is executed and the excitement level of neuron is determined:

$$n = \sum_{i=1}^R W_i X_i \quad 1.1$$

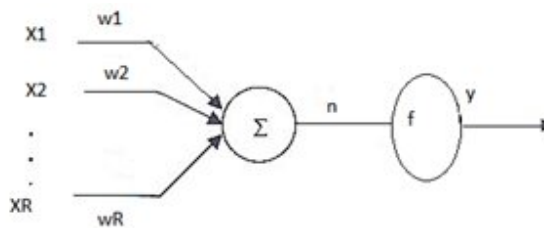


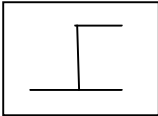
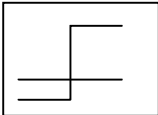
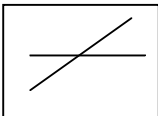
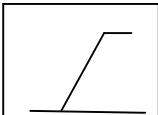
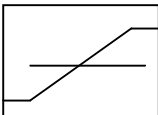
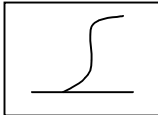
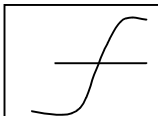
Figure 1.2 Mathematical neuron

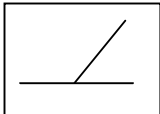

The output signal of a neuron is determined by conducting the excitement level through the function f , called activation function as in Equation 1.2.

$$y = f(n) \quad 1.2$$

The following activation functions can be utilized as function f

Table 1.1 Different Activation function with their symbols and MATLAB commands

Name	Input/Output Relation	Icon	MATLAB Function
Hard Limit	$y = 0 \quad n < 0$ $y = 1 \quad n \geq 0$		Hardlim
Symmetrical Hard Limit	$y = -1 \quad n < 0$ $y = +1 \quad n \geq 0$		Hardlims
Linear	$y = n$		Purelin
Saturating Linear	$y = 0 \quad n < 0$ $y = n \quad 0 \leq n \leq 1$ $y = 1 \quad n > 1$		Satlin
Symmetric Saturating Linear	$y = -1 \quad n < -1$ $y = n - 1 \quad -1 \leq n \leq 1$ $y = 1 \quad n > 1$		Satlins
Log-Sigmoid	$y = \frac{1}{1 + e^{-n}}$		Logsig
Hyperbolic Tangent Sigmoid	$y = \frac{e^n - e^{-n}}{e^n + e^{-n}}$		Tansig

Positive Linear	$y = 0 \quad n < 0$ $y = n \leq n$		Poslin
Competitive	$y = 1$ neuron max n <i>with</i> $y = 0$ all other neurons		Compet

1.2.2 Architectures of Neural Networks

The totality of the neurons, connected with each other and with the network. The input vector comes to the network by activating the input neurons. A set of input signals of a network's neurons is called the vector of input activeness. Connection weights of neurons are represented in form of matrix W , element w_{ij} of which is the connection weight between i -th and j -th neurons. During the network functioning process, the input vector is transformed into output one, i.e. some information processing is performed. The computational power of the network, thus, solves problems with its connections. Connections link inputs of one neuron with output of others. The connection strengths are given by weight coefficients. NN can also consist a bias term, which acts on a neuron like an offset. The function of the bias is to provide a threshold for the activation of neurons. The bias can be connected all neurons in network.

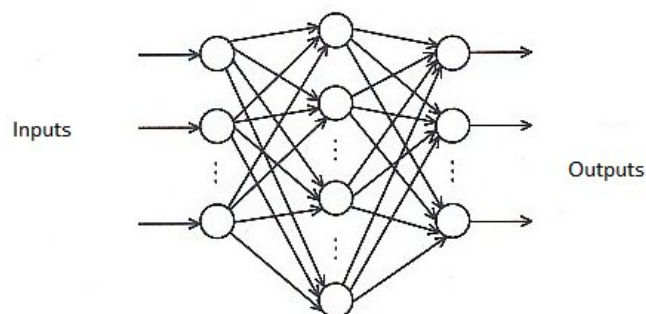


Figure1.3 Basic structure of neural network

NNs can be divided into two types of architectures: feed forward networks and recurrent NNs. Feed forward networks have no feedback connections. In this type of network, neurons of the j -th layer receive signals from environment (when $j=1$) or the neurons of previous the $(j-1)$ -th layer when $(j>1)$ and pass their outputs to neurons of the next $(j+1)$ -th layer to the environment (when j is the last layer). Feed forward networks can be single-layer or multi-layer. Multilayer NNs consist of input, output and hidden layer. The use of hidden layers allows an increase in the computational power of the network. Choosing the optimal structure of a network provides an increase in reliability and computational power, and a decreased processing. The multilayer perceptron (MLP), adaptive network, radial basis function neural network (RBFNN), the learning vector quantization (LVQ) network, and the group-method of data handling network can be given the examples of feedforward networks. Recurrent neural networks (RNN) have structures similar to standard feedforward NN with layers of nodes connected via weighted feed-forward connections, but also include time delayed feedback or recurrent connections in the architecture. The important advantage of the RNN is the ability to approximate a continuous or discrete nonlinear dynamic system by neural dynamics defined by a system of nonlinear differential equations. This offers the opportunities for applications to adaptive control problems. Examples of the RNN include the Hopfield network, the Elman network, and the Jordan network.

1.2.3 Learning in Neural Networks

Generally, learning is the process by which the NN adapts itself to a stimulus, and eventually it produces a desired response. It is also a continuous classification process of input stimuli: when a stimulus appears at the network, the network either recognizes it or it develops a new classification. Actually, during the process of learning, the network adjusts its parameters, the synaptic weights, in response to an input stimulus so that its actual output response converges to the desired output response. When the actual output response is the same as the desired one, the network has completed the learning phase.

Learning rules for networks are described by mathematical expressions called learning equations. The neurons in NNs may be interconnected in different ways; however, the

learning process is not same for the all. It is known that, different learning methodologies suit different people. Like this, different learning techniques suit different NNs. There are two general categories of learning in NNs, supervised and unsupervised learning. In supervised learning, both the input and the actual response and the desired response are available and are used to formulate a cost (error) measure. If the actual response differs from the target response, the NN generates an error signal, which is then used to calculate the adjustment that should be made to the network's weights so that actual output matches the target output (Jain 1997).

Unlike supervised learning, there is no target output in unsupervised learning. During the training period, the network receives at its input many different input patterns and it arbitrarily organizes the pattern into categories. When a stimulus is later applied, the network provides an output response indicating the class to which the stimulus belongs. If a class cannot be found for the stimulus, a new class is generated. This type of learning sometimes referred to as self-organizing learning.

1.2.4 Learning Algorithms

A learning algorithm is a mathematical tool that outlines the methodology and the speed for NN to reach the steady state of its parameters, weights and thresholds successfully. It starts with an error function (energy function), which is expressed in terms of weights. The objective is to minimize the error in the set of weights. When the error function is zero, or small enough, the steady state of the network and of the weights is reached. During learning, the error function decreases and the weights are updated. The decrease may be accomplished with different optimization techniques such as the Delta rule, Boltzman's algorithm, the backpropogation learning algorithm and simulation annealing. The selection of the error function and the optimization method is important, because it may increase stability, instability or a solution trapped in a local minimum. Backpropogation learning algorithm (Rumelhart et al 1986) is the basic learning mechanism and it is very popular in the literature. In this algorithm, the network output, on presentation of input data, is compared with the desired output and a measure of the error is obtained. This error measure is then used to incrementally modify appropriate

weights in the connection matrices in order to reduce the error. Following numerous presentations of training data, the overall error of the network is expected to be reduced to an acceptable level and the network has then learned how to solve the problem posed by training data. Example of the backpropagation learning algorithm will be given in the following section.

1.2.5 Adaptive networks

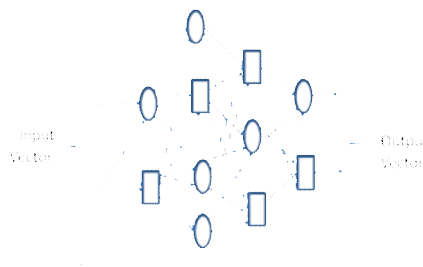


Figure 1.4 Adaptive network

An adaptive network (Figure 1.4) is an example of multilayer feedforward NN in which each node performs a particular function (node function) on incoming signals as well as a set of parameters pertaining to this node. The formulas for the node functions may vary from node to node, and the choice of each node function depends on the overall input-output function, which the adaptive network is required to carry out. The links in an adaptive network only show the flow direction signals between nodes. No weights are associated with the links .

The basic learning rule of adaptive networks is the back propagation learning rule. However, since it is slow and tends to become trapped in local minima, a hybrid learning rule algorithm was proposed to speed up the learning algorithm by Jang in 1993.

1.2.5.1 Back propagation of Adaptive Networks

Suppose that adaptive network in Figure 1.4 has N layers. (j,i) is the node in the i th position of the j th layer and $O_{j,i}$ is the node output. Since a node output depends on its incoming signal and its parameter set, it can be defined as follows:

$$O_{j,i} = f_{j,i}(O_i^{j-1}, \dots, O_{\#(j-1)}^{j-1}, a, b, c, \dots) \quad 1.3$$

a, b, c are the parameters pertaining to this node and f is the node function. Assuming the training data has P entries, an error measure can be defined for the p th ($1 \leq p \leq P$) entry of training data as the sum of the square errors and it is equal to:

$$E_p = \sum_{k=1}^N (y_{dk} - y_k)^2 \quad 1.4$$

where y_d is the k th component of the p th target vector, and y_k is the k th component of the actual output vector produced by presentation of p th input vector. Therefore, the overall error measure can be expressed as:

$$E = \sum_{p=1}^P E_p \quad 1.5$$

When E_p is equal to zero, the network is able to reproduce exactly the desired output vector in the p th training data pair. Therefore, aim is to minimize an overall measure. For this, first error rate $\frac{\partial E}{\partial y}$ for p th training data and for each node output should be calculated. The error rate for output node at layer N can be calculated from Equation 1.4

$$\frac{\partial E_p}{\partial y_{ip}} = -2(y_{dik} - y_{ip}) \quad 1.6$$

For internal node at the i th position of layer j , the error rate can be derived by the chain rule of differential calculus as follows:

$$\frac{\partial E_p}{\partial y_{ip}^j} = \sum_{k=1}^{\#j+1} \frac{\partial E_p}{\partial y_{kp}^{j+1}} \frac{\partial y_{kp}^{j+1}}{\partial y_{ip}^j} \quad 1.7$$

where ($1 \leq j \leq N-1$). That is, the error rate of an internal node at layer j can be expressed as a linear combination of the error rates of the layer $j+1$. Therefore, for all $1 \leq j \leq N$ and $1 \leq i \leq \#(j)$ error rates are found by applying Equation 1.6 and 1.7. The underlying procedure is called backpropagation since the error rates are obtained sequentially from the output layer back to the input layer. The gradient vector is defined as the derivative of the error measure with respect to α and equals to:

$$\frac{\partial E_p}{\partial \alpha} = \sum_{y^* \in S} \frac{\partial E_p}{\partial y^*} \frac{\partial y^*}{\partial \alpha} \quad 1.8$$

where S is the set of nodes whose outputs depend on α . Then, the derivative of the overall measure E with respect to α is written as:

$$\frac{\partial E}{\partial \alpha} = \sum_{p=1}^P \frac{\partial E_p}{\partial \alpha} \quad 1.9$$

According to the simple steepest descent, the update formula for the generic parameter α is as follows:

$$\Delta \alpha = -\frac{\partial E}{\partial \alpha} \quad 1.10$$

in which η is a learning rate which can be further expressed as:

$$\eta = \frac{k}{\sqrt{\sum \left(\frac{\partial E}{\partial \alpha}\right)^2}} \quad 1.11$$

where k is the “step size”, the length of each transition in the parameter space. It is observed that if k is small, the gradient method will closely approximate the gradient path, but convergence will be slow since gradient must be calculated many times. On the other hand, if k is large, convergence will initially be vary fast, but the algorithm will oscillate about the optimum. For this reason, it is proposed that if the error measure undergoes four consecutive reductions, k should be increased by 10%, if the error measure undergoes two consecutive combinations one increase and one reduction, it should be decreased by 10%.

1.2.5.2 Hybrid Learning Rule

Hybrid learning rule combines the gradient method and the least square estimate (LSE) to identify the parameters. Assuming that the adaptive network has only one output, the output can be expressed as:

$$\text{Output} = F(I, S) \quad 1.12$$

where I is the set of input variables and S is the set of parameters. "If there exists a function H such that the composite function $H \circ F$ is linear some of the elements of S , these elements can be identified by the least square estimates.". Hence parameter set S can be decomposed into two sets: S_1 and S_2 such that $H \circ F$ is linear in the elements of S_2 . Then applying H to Equation 1.12, it becomes as follows:

$$\text{Output} = H \circ F(I, S) \quad 1.13$$

which is linear in the elements of S_2 . Now given values of elements of S_2 , P training data is plugged into Equation 1.12 and a matrix equation obtained:

$$AX = B \quad 1.14$$

where X is an unknown vector whose elements are parameters in S_2 . Since number of training data pairs is usually greater than number of linear parameters, this is an over-determined problem and there is no exact solution for Equation 1.14. To deal with this problem, a sequential method of LSE was proposed by Jang. According to this method X is calculated iteratively using the sequential formulas adopted in literature as:

$$X_{i+1} = X_i + S_{i+1} a_{i+1} (b_{i+1}^T - a_{i+1}^T X_i) \quad 1.15$$

$$S_{i+1} = S_i - \frac{S_i a_i a_{i+1}^T S_i}{1 + a_{i+1}^T S_i a_{i+1}} \quad i = 0, 1, \dots, P - 1 \quad 1.16$$

Where a_i^T is the i th row vector of matrix A defined in Equation 1.14 and b_i^T is the i th element of B . S_i is called the covariance matrix. Each epoch of hybrid learning procedure consists of a forward pass and backward pass. In forward pass, input data and functional signals go forward are supplied to calculate each node output until the matrices A and B in Equation 1.14 are obtained. Then, the parameters in 2 S are identified by the sequential least squares formulas. After identifying parameters in S_2 the functional signals keep going forward till the error measure is calculated. In the backward pass, the error rates propagate from the output end toward the input end, and the parameters in S_1 are updated by the gradient method (Equation 1.10).

1.3 Fuzzy Logic

This section presents general information about the theory of fuzzy logic. Definition of a fuzzy set and linguistic variable conception are presented.

1.3.1 The Fuzzy set

A “fuzzy set” is a simple extension of the definition of a classical set in which the characteristic function is permitted to have any values between 0 and 1. A “fuzzy set” A in X can be defined as a set of ordered pairs:

$$A = \{(x, \mu_A(x)) | x \in X\} \quad 1.17$$

where $\mu_A(x)$ is called membership function for the fuzzy set A . It maps each x to a membership grade between 0 and 1. Examples of membership functions (Triangular, Trapezoidal and Gaussian) can be seen in Figure 1.5, Figure 1.6 and figure 1.7 and described with the following formulas: **Triangular Membership function**

$$\text{triangle}(x; a, b, c) = \begin{cases} 0 & x \leq a \\ \frac{x-a}{b-a} & a \leq x \leq b; \frac{c-x}{c-b} & b \leq x \leq c \\ 0 & c \leq x \end{cases}$$

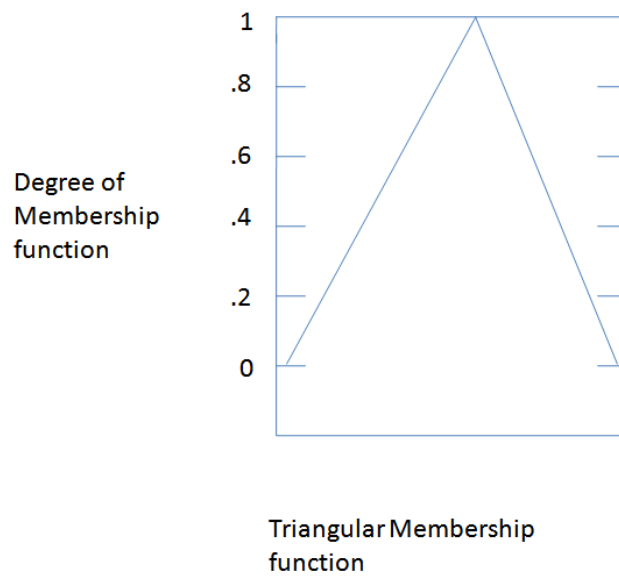
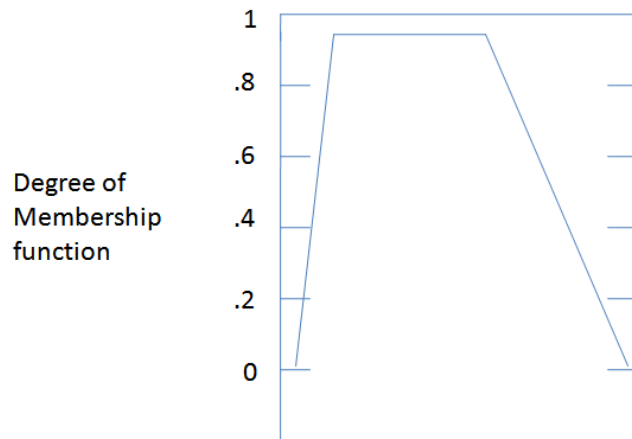


Figure1.5 Triangular Membership function

1) Trapezoidal Membership Function

$trapezoidal(x; a, b, c, d) =$

$$\left\{ \begin{array}{ll} 0 & x \leq a \\ \frac{x-a}{b-a} & a \leq x \leq b \\ 1 & b \leq x \leq c \\ \frac{d-x}{d-c} & c \leq x \leq d \\ 0 & d \leq x \end{array} \right.$$

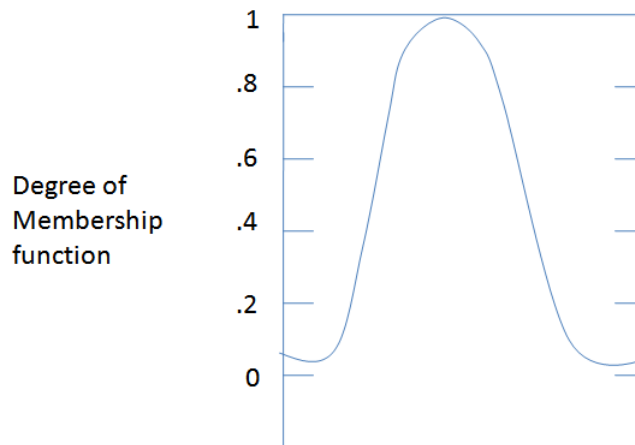


Trapezoidal Membership function

Figure 1.6 Trapezoidal Membership function

2) **Gaussian Membership Function:**

$$gaussian(x; c, \sigma) = e^{\frac{-(x-c)^2}{2\sigma}}$$



Gaussian Membership function

Figure 1.7 Gaussian Membership function

1.3.2 Linguistic Variables

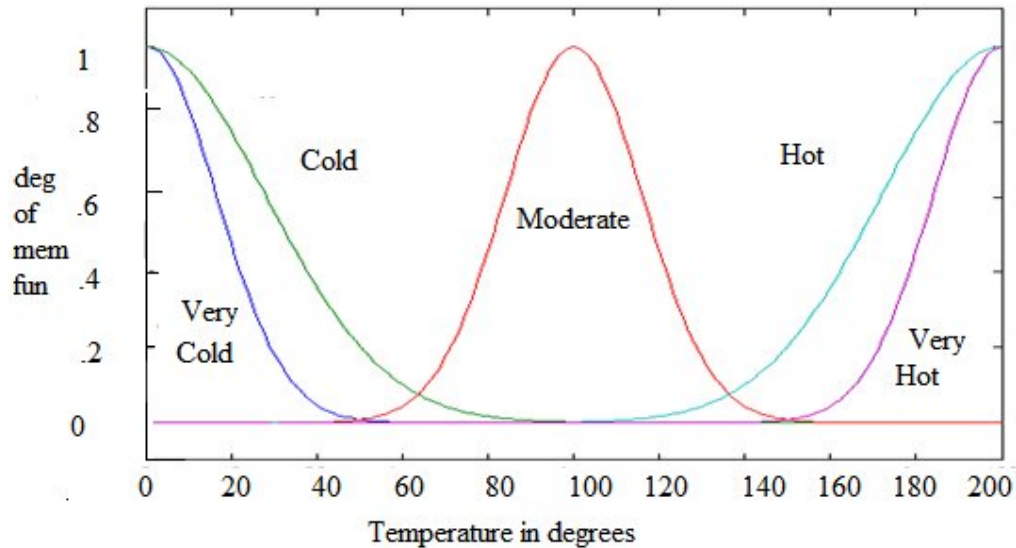


Figure 1.8 Membership functions of the term set temperature

The concept of linguistic variables was introduced by Zadeh in 1973 to provide a basis for approximate reasoning. A linguistic variable was defined as a variable whose values are words or sentences. For instance, temperature can be linguistic variable if its values are linguistic rather than numerical, i.e., hot, very hot, cold, very cold, etc., rather than 20, 21, 23, 45.... Figure 1.8 illustrates the term set temperature expressed by the Gaussian MFs

1.3.3 Fuzzy if then Rules

A fuzzy if-then rule (fuzzy rule, fuzzy implication, or fuzzy conditional statement) is expressed as follow:

If x is A then y is B

where A and B linguistic values defined by fuzzy sets. “x is A” is called “antecedent” or “premise”, while “y is B” is called the “consequence” or “conclusion”. Some of the if-then rule examples can be given below:

- If pressure high, then volume is small.
- If the speed is low AND the distance is small, then the force on brake should be small.

1.3.4 Fuzzy Reasoning

Fuzzy reasoning, approximate reasoning, is an inference procedure whose outcome is conclusion for a set of fuzzy if-then rules. The steps of fuzzy reasoning can be given as follows:

1. “Input variables are compared with the MFs on the premise part to obtain the membership values of each linguistic label (fuzzification).
2. The membership values on the premise part are combined through specific fuzzy set operations such as: min, max, or multiplication to get firing strength (weight) of each rule.
3. The qualified consequent (either fuzzy or crisp) is generated depends on the firing strength.
4. The qualified consequents are aggregated to produce crisp output according to the defined methods such as: centroid of area, bisector of area, mean of maximum, smallest of maximum and largest of maximum (defuzzification)

1.3.5 Fuzzy Systems

Fuzzy systems are made of a knowledge base and reasoning mechanism called fuzzy inference engine. The structure of fuzzy inference engine is shown in Figure 1.9. A fuzzy inference engine combines fuzzy if-then rules into a mapping from the inputs of the system into its outputs, using fuzzy reasoning methods. That is, fuzzy systems represents on linear mapping accompanied by fuzzy if-then rules from the rule base. Each of these rules describes the local mappings. The rule base can be constructed either from human expert or automatic generation that is extraction of rules using numerical input-output data.

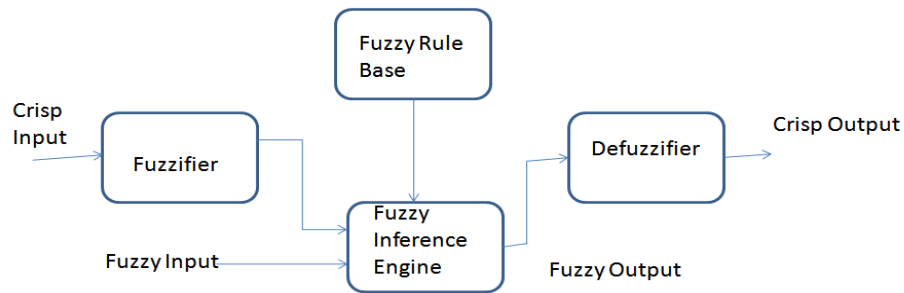


Figure 1.9 Fuzzy Inference Engine

Mamdani and Takagi-Sugeno fuzzy systems are the examples of fuzzy inference systems. Mamdani fuzzy inference system was first used to control a steam engine and boiler combination by a set of linguistic rules obtained from human operators . Figure 1.10 illustrates how a two rule Mamdani fuzzy inference system derives the overall output z when subjected to two numeric inputs x and y . Takagi-Sugeno fuzzy inference system was first introduced by Takagi and Sugeno . The difference of Takagi- Sugeno model is that each rule has a crisp output, and the overall output is determined as weighted average of single rules output. This type of fuzzy inference system is shown in Figure 1.11

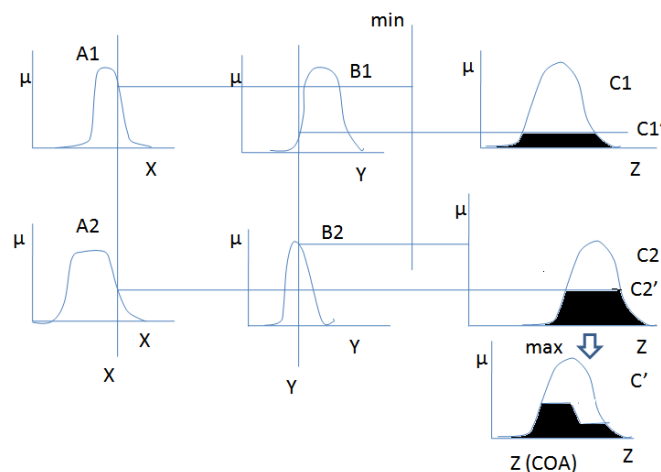
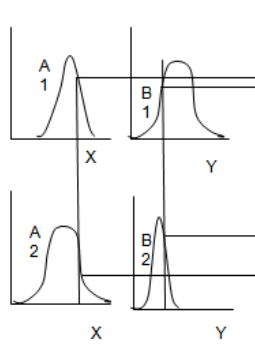


Figure1.10 Mamdani Fuzzy Inference System



W1 $f_1 = p_1x + q_1y + r_1$

$f_2 = p_2x + q_2y + r_2$

W2

$$\Rightarrow f = \frac{w_1 f_1 + w_2 f_2}{w_1 + w_2}$$

Figure1.11 Takagi-Sugeno Fuzzy Inference System

CHAPTER 2

ANFIS (ADAPTIVE NEURO FUZZY INFERENCE SYSTEM)

2.1 Introduction

ANFIS are a class of adaptive networks that are functionally equivalent to fuzzy inference systems. ANFIS represent Sugeno Tsukamoto fuzzy models. ANFIS uses a hybrid learning algorithm. Using a given input/output data set, the toolbox function `anfis` constructs a fuzzy inference system (FIS) whose membership function parameters are tuned (adjusted) using either a back propagation algorithm alone, or in combination with a least squares type of method. This allows your fuzzy systems to learn from the data they are modeling. The basic idea behind these neuro-adaptive learning techniques is very simple. These techniques provide a method for the fuzzy modeling procedure to learn information about a data set, in order to compute the membership function parameters that best allow the associated fuzzy inference system to track the given input/output data. This learning method works similarly to that of neural networks.

2.2 ANFIS Architecture

In ANFIS, Takagi-Sugeno type fuzzy inference system is used. The output of each rule can be a linear combination of input variables plus a constant term or can be only a constant term. The final output is the weighted average of each rule's output. Basic ANFIS architecture that has two inputs x and y and one output z is shown in Figure 3.10. The rule base contains two Takagi-Sugeno if-then rules as follows:

Rule1: If x is A_1 and y is B_1 , then $f = p x + q y + r$

Rule2: If x is A_2 and y is B_2 , then $f = p x + q y + r$

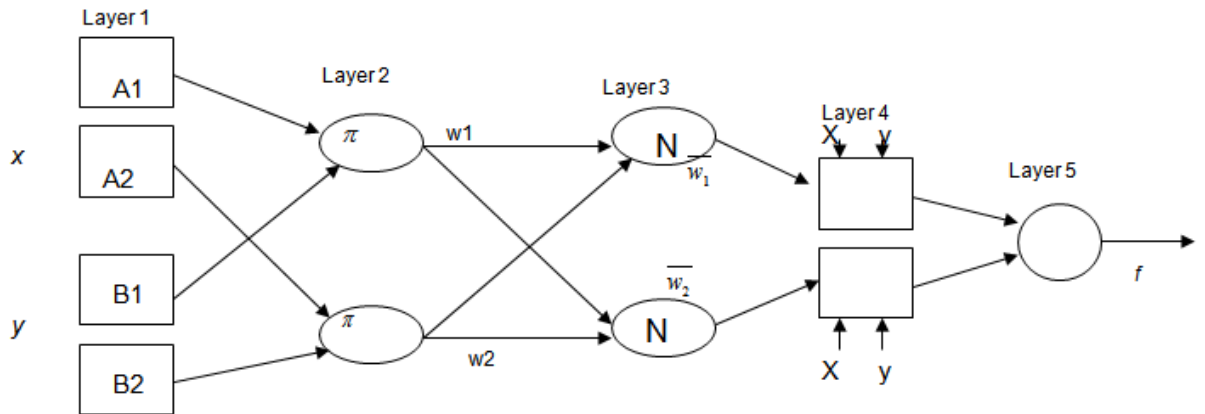


Figure 2.1 Basic structure of ANFIS

The node functions in the same layer are the same as described below:

Layer 1: Every node i in this layer is a square node with a node function as:

$$O_{1,i} = \mu_{A_i}(x) \quad \text{for } i = 1,2 \quad 3.1$$

$$O_{1,i} = \mu_{B_{i-2}}(y) \quad \text{for } i = 1,2 \quad 3.2$$

where x is the input to node i , and A_i (or B_{i-2}) is a linguistic label (such as “small” or “large”) associated with this node. In other words, $O_{1,i}$ is the membership grade of a fuzzy set A and it specifies the degree to which the given input x satisfies the quantifier A . The membership function for A can be any appropriate membership function, such as the Triangular or Gaussian. When the parameters of membership function changes, chosen membership function varies accordingly, thus exhibiting various forms of membership functions for a fuzzy set A . Parameters in this layer are referred to as “premise parameters”.

Layer 2: Every node in this layer is a fixed node labeled as Π , whose output is the product of all incoming signals

$$O_{2,i} = w_i = \mu_{A_i}(x)\mu_{B_i} \quad \text{for } i = 1,2 \quad 3.3$$

Each node output represents the firing strength of a fuzzy rule

Layer 3: Every node in this layer is a fixed node labeled N. The i th node calculates the ratio of the rule's firing strength to the sum of all rules' firing strengths:

$$O_{3,i} = w_i = \frac{w_i}{w_1 + w_2} \quad \text{for } i = 1, 2 \quad 3.4$$

Outputs of this layer are called “normalized firing strengths”

Layer 4: Every node i in this layer is an adaptive node with a node function as:

$$O_{4,i} = \bar{w}_i f_i = w_i(p_i x + q_i y + r_i) \quad 3.5$$

where \bar{w}_i is a normalized firing strength from layer 3 and $\{p_i, q_i, r_i\}$ is the parameter set of this node. Parameters in this layer are referred to as “consequent parameters”.

Layer 5: The single node in this layer is a fixed node labeled Σ that computes the overall output as the summation of all incoming signals:

$$\text{overall output } O_{5,1} = \sum_i \bar{w}_i f_i = \frac{\sum_i w_i f_i}{\sum_i w_i} \quad 3.6$$

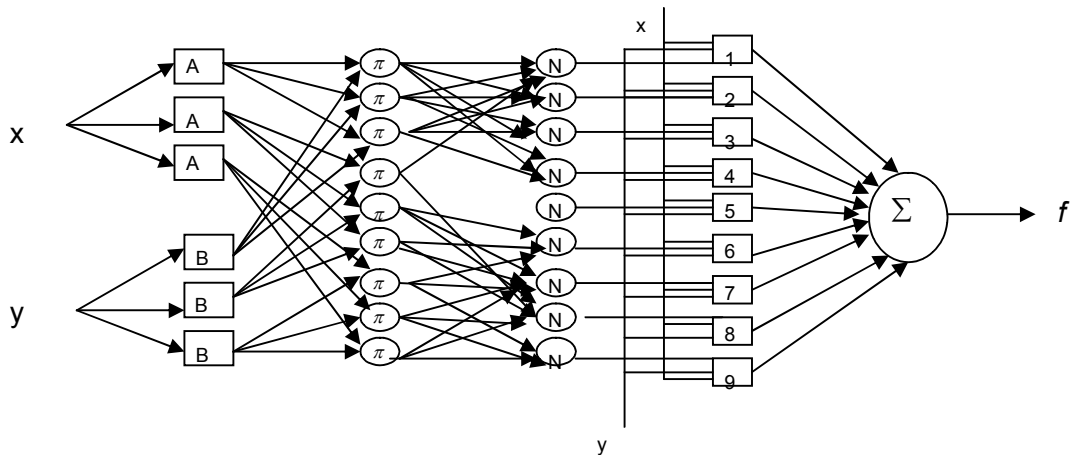


Figure2.2 ANFIS Architecture with nine rules

Thus an adaptive network, which is functionally equivalent to the Takagi- Sugeno type fuzzy inference system, has been constructed. Other example of ANFIS with nine rules can be shown in Figure2.2. Three membership functions are associated with each input, so the input space partitioned into nine fuzzy subspaces. The premise part of a rule describes a fuzzy subspace, while the consequent part specifies the output within this fuzzy subspace.

2.3 ANFIS Learning algorithm

2.3.1 Hybrid Learning Algorithm - I

- In the forward pass the algorithm uses least-squares method to identify the consequent parameters on the layer 4.
- In the backward pass the errors are propagated backward and the premise parameters are updated by gradient descent.

2.3.2 Hybrid Learning Algorithm – II

Table 2.1 Two passes in the hybrid learning algorithm for ANFIS.

	Forward Pass	Backward Pass
Premise Parameters	Fixed	Gradient Descent
Consequent Parameters	Least-square estimator	Fixed
Signals	Node outputs	Error signals

2.3 ANFIS as a Universal Approximator

When the number of rules is not restricted, a zero-order Sugeno model has unlimited approximation power for matching any nonlinear function arbitrarily well on a compact set. This can be proved using the Stone-Weierstrass theorem. Let D be a compact

space of N dimensions, and let F be a set of continuous real-valued functions on D satisfying the following criteria:

1) Stone-Weierstrauss theorem - I

- Let D be a compact space of N dimensions, and let F be a set of continuous real-valued functions on D satisfying the following criteria:
- **Identity function:** The constant $f(x) = I$ is in F.
- **Separability:** For any two points $x_1 \neq x_2$ in D, there is an f in F such that $f(x_1) \neq f(x_2)$.
- **Algebraic closure:** If f and g are any two functions in F, then fg and $af + bg$ are in F for any two real numbers a and b.

2) Stone-Weierstrauss theorem – II

- Then F is dense on $C(D)$, the set of continuous real-valued functions on D.
- For any $\epsilon > 0$ and any function g in $C(D)$, there is a function f in F such that $|g(x) - f(x)| < \epsilon$ for all x in D.
- The ANFIS satisfies all these requirements

From the proposed ANFIS architecture (Figure 3.11), the output f can be defined as

$$f = \frac{w_1}{w_1+w_2} f_1 + \frac{w_2}{w_1+w_2} f_2 \quad 3.7$$

$$f = \bar{w}_1 (p_1 x + q_1 y + r_1) + \bar{w}_2 (p_2 x + q_2 y + r_2) \quad 3.8 \quad \text{where}$$

p_1, q_1, r_1, q_2 and r_2 are the linear consequent parameters. The methods for updating the parameters are listed as below:

1. *Gradient decent only:* All parameters are updated by gradient decent back propagation.
2. *Gradient decent and One pass of Least Square Estimates (LSE):* The LSE is applied only once at the very beginning to get the initial values of the consequent parameters and then the gradient descent takes over to update all parameters.
3. *Gradient and LSE:* This is the hybrid learning rule. Since the hybrid learning approach converges much faster by reducing search space dimensions than the original back propagation method, it is more desirable. In the forward pass of the

hybrid learning, node outputs go forward until layer 4 and the consequent parameters are identified with the least square method. In the backward pass, the error rates propagate backward and the premise parameters are updated by gradientdescent.

CHAPTER 3

THE INVERTED PENDULUM

3.1 Introduction

Remember when you were a child and you tried to balance a broom-stick on your index finger or the palm of your hand? You had to constantly adjust the position of your hand to keep the object upright. An INVERTED PENDULUM does basically the same thing. However, it is limited in that it only moves in one dimension, while your hand could move up, down, sideways, etc. Just like the broom-stick, an Inverted Pendulum is an inherently unstable system. Force must be properly applied to keep the system intact. To achieve this, proper control theory is required. The Inverted Pendulum is essential in the evaluating and comparing of various control theories.

The *inverted pendulum* (IP) is among the most difficult systems to control in the field of control engineering. Due to its importance in the field of control engineering, it has been a task of choice to be assigned to Control Engineering students to analyze its model and propose a linear compensator according to the PID control law. The inverted pendulum offers a very good example for control engineers to verify a modern control theory. This can be explained by the facts that inverted pendulum is marginally stable, in control sense, has distinctive time variant mathematical model. Inverted Pendulum is a very good model for the attitude 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. To solve such problem with non-linear time variant system, there are alternatives such as real time computer simulation of these equations or linearization.

3.2 Modeling of inverted pendulum

The inverted pendulum is a highly nonlinear and open-loop unstable system. 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 characteristics of the inverted pendulum make identification and control more challenging. The Inverted Pendulum problem resembles the control systems that exist in robotic arms. The dynamics of Inverted Pendulum simulates the dynamics of robotic arm in the condition when the center of pressure lies below the centre of gravity for the arm so that the system is also unstable. Robotic arm behaves very much like Inverted Pendulum under this condition. The inverted pendulum system is a classic control problem that is used in universities around the world. It is a suitable process to test prototype controllers due to its high non-linearities and lack of stability. The system consists of an inverted pole with mass, m , hinged by an angle θ from vertical axis on a cart with mass, M , which is free to move in the x direction as shown in Figure 3.1. A force, F is required to push the cart horizontally. In this chapter, the dynamical equations of the system will be derived.

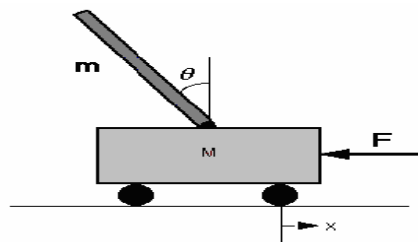


Figure 3.1 Free body diagram of the inverted pendulum system

This section outlines the assumptions, design requirements and physical properties of the model and its components

Assumptions for inverted pendulum are as follows.

1. The system starts in a state of equilibrium meaning that the initial conditions are therefore assumed to be zero.
2. The pendulum does not move more than a few degrees away from the vertical to satisfy a linear model.
3. A step input is applied.

3.3 Force Analysis and System Equations

To derive the suitable mathematical model for an inverted pendulum system, consider Figure 3.2

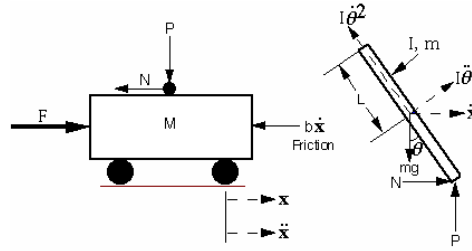


Figure 3.2 Free body diagrams of the system.

Summing the forces of the cart in the horizontal direction

$$F = M\ddot{x} + b\dot{x} + N \quad 3.1$$

Summing the forces of the pendulum in the horizontal direction

$$N = M\ddot{x} + ml\ddot{\theta}\cos\theta - ml\dot{\theta}^2\sin\theta \quad 3.2$$

Substituting equation 3.2 into equation 3.1

$$(M + m)\ddot{x} + b\dot{x} + ml\ddot{\theta}\cos\theta - ml\dot{\theta}^2\sin\theta = F \quad 3.3$$

Sum the forces perpendicular to the pendulum

$$P\sin\theta + N\cos\theta - mg\sin\theta = ml\ddot{\theta} + m\ddot{x}\cos\theta \quad 3.4$$

To get rid of the P and N terms in the equation above, sum the moments around the centroid of the pendulum.

$$-Pl\sin\theta - Nl\cos\theta = I\ddot{\theta} \quad 3.5$$

Combining equation 3.4 and equation 3.5 we have

$$(I + ml^2)\ddot{\theta} + mgl\sin\theta = -ml\ddot{x}\cos\theta \quad 3.6$$

Since Matlab can only work with linear functions, this set of equations should be linearized about $\theta = \pi$. Assume that $\theta = \pi + \phi$ (where ϕ represents a small angle from the vertical upward direction). Therefore $\cos\theta = -1, \sin\theta = -\phi$ and $\frac{d\theta}{dt} = 0$

After linearization the two equations of motion become (where u represents the input)

$$(I + ml^2)\ddot{\phi} - mgl\phi = ml\ddot{x} \quad 3.7$$

$$(M + m)\ddot{x} + b\dot{x} - ml\ddot{\phi} = u \quad 3.8$$

3.4 Transfer Function of the Inverted Pendulum

Laplace transform of the system equations

$$(I + ml^2)\phi(s)s^2 - mgl\phi(s) = -mlX(s)s^2 \quad 3.9$$

$$(M + m)X(s)s^2 + bX(s)s - ml\phi(s)s^2 = U(s) \quad 3.10$$

Since the angle ϕ is the output of interest, solving equation 3.9 for X(s),

$$X(s) = \left[\frac{(I+ml^2)}{ml} + \frac{g}{s^2} \right] \phi(s) \quad 3.11$$

Substituting equation 3.11 into equation 3.10

$$(M + m) \left[\frac{(I+ml^2)}{ml} + \frac{g}{s} \right] \phi(s)s^2 + b \left[\frac{(I+ml^2)}{ml} + \frac{g}{s} \right] \phi(s) - ml\phi(s)s^2 = U(s) \quad 3.12$$

Re-arranging, the transfer function is

$$\frac{\phi(s)}{U(s)} = \frac{\frac{ml}{s^2}}{s^4 + \frac{(I+ml^2)}{q}s^2 - \frac{(M+m)mgl}{q}s^2 - \frac{bmg}{q}} \quad 3.13$$

where, $q = [(M + m)(I + ml^2) - ml^2]$

From the transfer function above it can be seen that there is both a pole and a zero at the origin. These can be canceled and the transfer function of the angle is shown as in equation 3.14 and equation 3.15

$$\frac{\phi(s)}{U(s)} = \frac{\frac{ml}{q}s}{s^3 + \frac{(I+ml^2)}{q}s^2 - \frac{(M+m)mgI}{q}s - \frac{bmgI}{q}} \quad 3.14$$

$$\frac{\phi(s)}{U(s)} = \frac{4.545s}{s^3 + 1.818s^2 - 31.1818s - 4.4545} \quad 3.15$$

Solving for the transfer function of the position, the following equation 3.16 will be derived

$$\frac{X(s)}{U(s)} = \frac{1.8182s - 4.4545}{s^3 + 1.818s^2 - 31.1818s - 4.4545} \quad 3.16$$

3.5 State-Space analysis of Inverted Pendulum

The linearized system equations can also be represented in state-space form as shown in equation 3.17 till equation

$$3.19 \begin{bmatrix} \ddot{x}(t) \\ \dot{x}(t) \\ \ddot{\phi}(t) \\ \dot{\phi}(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & \frac{1}{I(M+m)+(Mml^2)} & \frac{m^2gl^2}{I(M+m)+(Mml^2)} & 0 \\ 0 & 0 & 0 & 1 \\ 0 & \frac{mlb}{I(M+m)+(Mml^2)} & \frac{mgI(M+m)}{I(M+m)+(Mml^2)} & 0 \end{bmatrix} \begin{bmatrix} x(t) \\ \dot{x}(t) \\ \phi(t) \\ \dot{\phi}(t) \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{0}{I(M+m)+(Mml^2)} \\ 0 \\ \frac{ml}{I(M+m)+(Mml^2)} \end{bmatrix} u(t)$$

Substituting all the parameter values into the equation 3.17 yielding equation 3.18 and equation 3.19

$$\begin{bmatrix} \ddot{x}(t) \\ \dot{x}(t) \\ \ddot{\phi}(t) \\ \dot{\phi}(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -0.1818 & 2.6727 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0.4545 & 31.1818 & 0 \end{bmatrix} \begin{bmatrix} x(t) \\ \dot{x}(t) \\ \phi(t) \\ \dot{\phi}(t) \end{bmatrix} + \begin{bmatrix} 0 \\ 1.8182 \\ 0 \\ 4.5455 \end{bmatrix} u(t) \quad 3.18$$

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

3.19

The C matrix is 2 by 4, because both the cart's position and the pendulum's position are part of the output. For the state-space design, a multi-output system will be controlled so that the cart's position and pendulum's angle can be observed from the first row and second row of the output respectively.

3.6 Programming tool which are used for control of Inverted Pendulum

There are several programming tools which may be employed for controlling the nonlinear system considered here. We may also incorporate basic C /C++ programming or even Java may be used .One may also used LabView to control the IP. But we have used the MatLab and Simulink to control the inverted pendulum.

3.6.1 Matlab

Matlab (short for MATrix LABoratory) is a language for technical computing, developed by the The Mathworks, Inc. It provides a single platform for computation, visualization, programming and software development. All problems and solutions in Matlab are expressed in notation used in linear algebra and essentially involve operations using matrices and vectors. Matlab can be used to solve problems in control systems.

In addition, Matlab can be used to build Graphical User Interfaces (GUIs) so that user-friendly custom software can be developed. The Matlab software environment has a core module (called Matlab) and associated with that are a set of "Toolboxes" that perform specialized computations.

3.6.2 What is m-File

An m-file, or script file, is a simple text file where a Matlab command can be placed. When the file is run, Matlab reads the commands and executes them exactly if each command is typed sequentially at the Matlab prompt. All m-file names must end with the extension '.m' (e.g. plot.m). By creating a new m-file with the same name as an existing m-file, Matlab will choose the one which appears first in the path order (help path for more information). To make life easier, choose a name for mfile which doesn't already exist.

3.6.3 What is Simulink

Simulink is a tool for modeling, simulating and analyzing multi domain dynamic systems. Its primary interface is a graphical block diagramming tool and a customizable set of block libraries.. Simulink offers tight integration with the rest of the MATLAB environment and both drive MATLAB or can be scripted from it. Simulink is widely used in control theory and digital signal processing for multi domain simulation and design..the above diagram presents an inverted pendulum model in simulink.

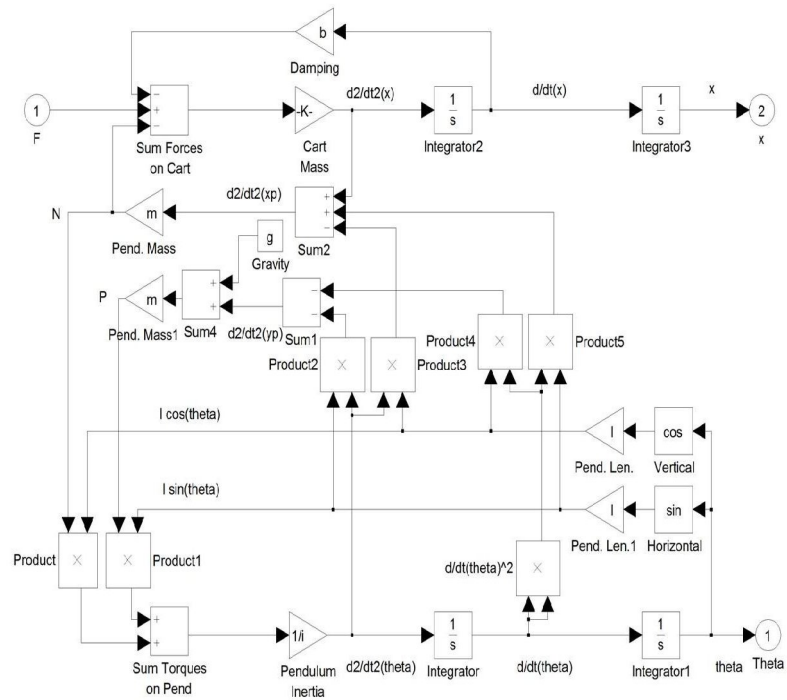


Figure 3.3 Model of Inverted Pendulum in Simulink

CHAPTER 4

FUZZY LOGIC CONTROLLER FOR INVERTED PENDULUM

4.1 Introduction

The concept of Fuzzy Logic Controller (FLC) was conceived by Lotfi Zadeh, a professor at the University of California at Berkley, and presented not as a control methodology, but as a way of processing data by allowing partial set membership rather than crisp set membership or non-membership. This approach to set theory was not applied to control systems until the 70's due to insufficient small-computer capability prior to that time. Professor Zadeh reasoned that people do not require precise, numerical information input, and yet they are capable of highly adaptive control. If feedback controllers could be programmed to accept noisy, imprecise input, they would be much more effective and perhaps easier to implement. FLC provides a simple way to arrive at a definite conclusion based upon vague, ambiguous, imprecise, noisy, or missing input information. FLC's approach to control problems mimics how a person would make decisions, only much faster.

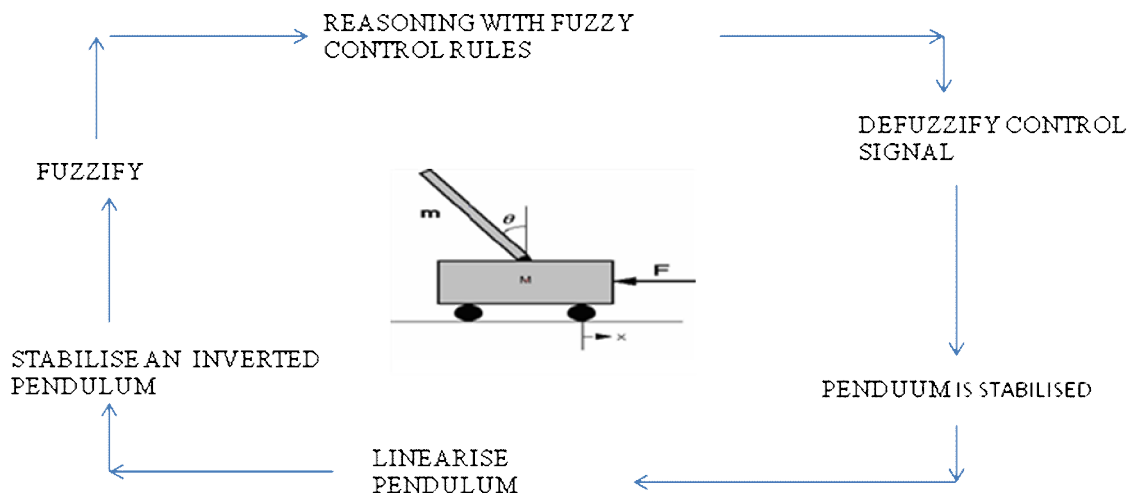


Figure 4.1 Fuzzy Logic controller overview

When idea of fuzzy logic is applied to control, it is generally called as ' fuzzy control. Fuzzy control is the first ever known to which fuzzy logic is applied. Main areas of application are numerous, such as control of cement kilns, electric trains, water purification plants etc.

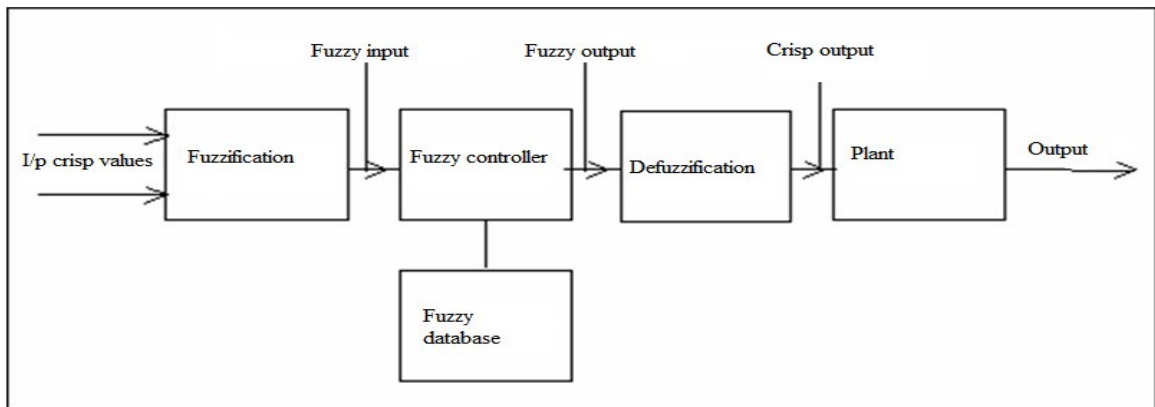


Figure 4.2 Block diagram of fuzzy controller

4.2 Planning Of Fuzzy Controllers

In designing fuzzy control, the problem of determining the form of the control rules is very crucial. This problem can be divided into two parts, i.e. determination of the antecedents and the determination of the consequent. For the antecedent part, three things have to be determined.

- 1) The input information for x_1 , x_2 , and etc which must be used in the antecedents is selected 49
- 2) Determination of the conditions, that is, the fuzzy partitions of the input
- 3) Determination of the parameters for the fuzzy variables.

For the consequent part, output is generally the control input for the process controlled. The only remaining problem is the fuzzy parameters. Determination of the consequent is not difficult, and the only problem is the wholly determination of the antecedents.

4.2.1 How is FLC different from conventional control methods

FLC incorporates a simple, rule-based IF X AND Y THEN Z approach to a solving control problem rather than attempting to model a system mathematically. The FLC model is empirically-based, relying on an operator's experience rather than their technical understanding of the system. For example, rather than dealing with temperature control in terms such as "SP =50°C", "T <100°C", or "21°C <TEMP <56°C", terms like "IF (process is too cool) AND (process is getting colder) THEN (add heat to the process)" or "IF (process is too hot) AND (process is heating rapidly) THEN (cool the process quickly)" are used. These terms are imprecise and yet very descriptive of what must actually happen. Consider what you do in the shower if the temperature is too cold: you will make the water comfortable very quickly with little trouble. FLC is capable of mimicking this type of behavior but at very high rate.

4.2.2 How Does FLC Work

FLC requires some numerical parameters in order to operate such as what is considered significant error and significant rate-of-change-of-error, but exact values of these numbers are usually not critical unless very responsive performance is required in which case empirical tuning would determine them. For example, a simple temperature control system could use a single temperature feedback sensor whose data is subtracted from the command signal to compute "error" and then time differentiated to yield the error slope or rate-of-change-of-error, hereafter called "error-dot". Error might have units of degs or F and a small error considered to be 2F while a large error is 5F. The "error-dot" might then have units of degs/min with a small error-dot being 5F/min and a large one being 15F/min. These values don't have to be symmetrical and can be "tweaked" once the system is operating in order to optimize performance. Generally, FLC is so forgiving that the system will probably work the first time without any tweaking

4.3 Fuzzy of Inverted Pendulum system

FLC was conceived as a better method for sorting and handling data but has proven to be an excellent choice for many control system applications since it mimics human control logic. It can be built into anything from small, hand-held products to large computerized process control systems. It uses an imprecise but very descriptive language to deal with input data more like a human operator. It is very robust and forgiving of operator and data input and often works when first implemented with little or no tuning. In this study, a very new technique, fuzzy controller, has been employed for stabilization of the inverted pendulum. Fuzzy controller, when inverted pendulum is considered, is a very good choice for control strategy. The reasons are non-linearity, complex mathematical computation and realtime computation need. Fuzzy controller fits the requirements very well in such cases. Moreover, applying a new technique offers new horizons to the researcher, especially for higher complex systems which need direct expert control

Here it is possible to give two inputs to the FLC as shown in Figure 5.11. The proposed defuzzification methods for the FLC are Sugeno or Mamdani. This is because both of these techniques are commonly used in designing the FLC.

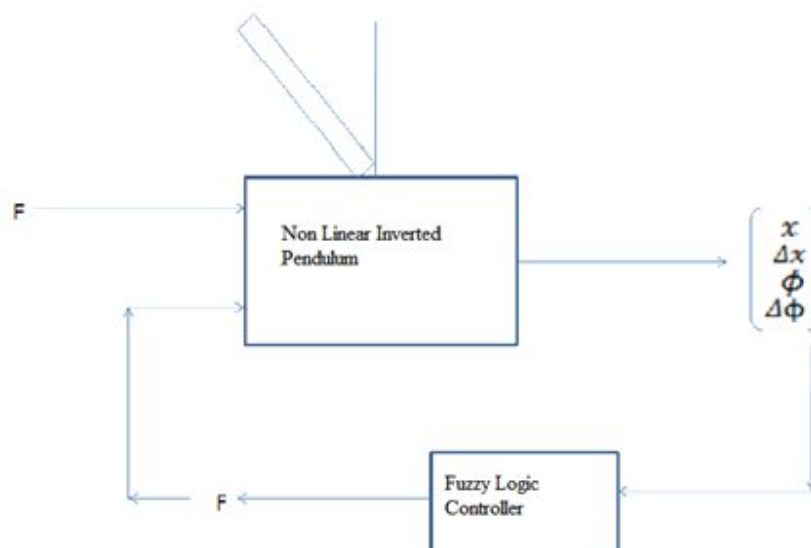


Figure 4.3 Block diagram of the system with fuzzy logic controller

In order to implement four inputs to the controllers, the FLC were divided into two. As can be seen in Figure 5.11, the ‘FLC 1’ controls the cart’s position, whereas the ‘FLC 2’ controls the pendulum’s angle. The ‘FLC 1’ received x and \dot{x} as the inputs while the ‘FLC 2’ received ϕ and $\dot{\phi}$ as the inputs. The control variables of both FLC were summed together to become a force and the input to the system

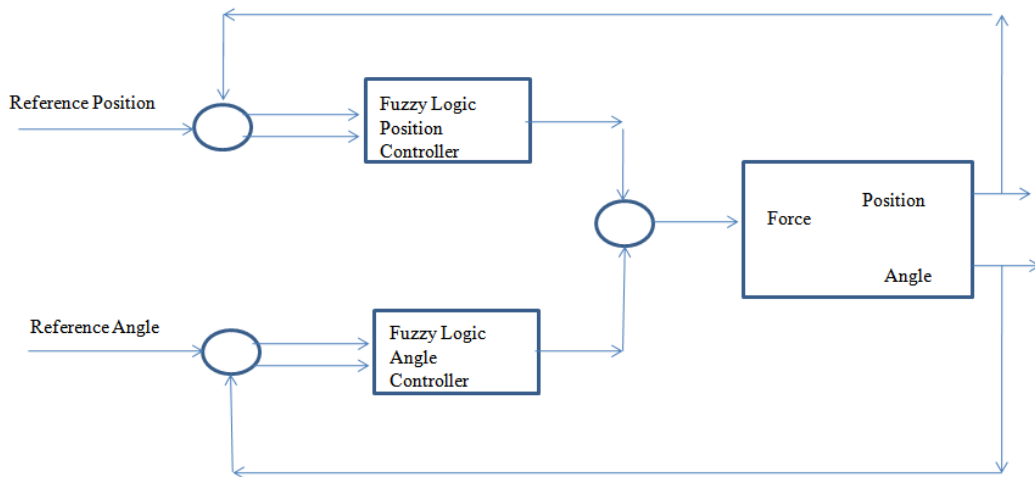


Figure 4.4 Rearranged block diagram of the system with Fuzzy Logic Controller

4.4 Inputs and Outputs of Fuzzy Logic Controllers

Through observations the input and output fuzzy variables has been identified. This is shown in Table 4.1

Table 4.1 Inputs and outputs of FLC

No	Input	Output
1.	Angle between pendulum shaft/pole and vertical line α	Force to push the cart, F
2.	Angular velocity of pendulum shaft/pole $\dot{\alpha}$	Force to push the cart, F

3.	Difference (error signal) between position of cart and set point position, x	Force to push the cart, F
4	Rate at which the error of position changes \dot{x}	Force to push the cart, F

4.5 Membership functions of the Inverted Pendulum's FLC

The input and output variables have to be quantized into several modules or fuzzy subsets and the appropriate labels are assigned in this controller. Two fuzzy logic controllers are needed to excite the stabilize force to the cart. There are FLC to control position and FLC to control angle. According to the complexity of this inverted pendulum system, seven fuzzy subsets are needed to quantize each fuzzy variable for both FLC as shown in Table 4.2

Table 4.2: Standard labels of quantization

Linguistic Term	Label
Negative Big	NB
Negative Medium	NM
Negative Small	NS
Zero	ZE
Positive Small	PS
Positive Medium	PM
Positive Big	PB

Figure 4.5 till Figure 4.10 show the membership functions of FLC's. The triangular shape was used to design the FLC. In addition, the membership functions were evenly distributed. It was done so that the tuning process of the controller can be easily done. As we can see from the figure, the ranges of the membership functions are different to each other. It was the crucial step to determine the appropriate values. It is needed to know the

nature of the plant from the expert knowledge. This can be acquired from the previous research papers that were done by the expert.

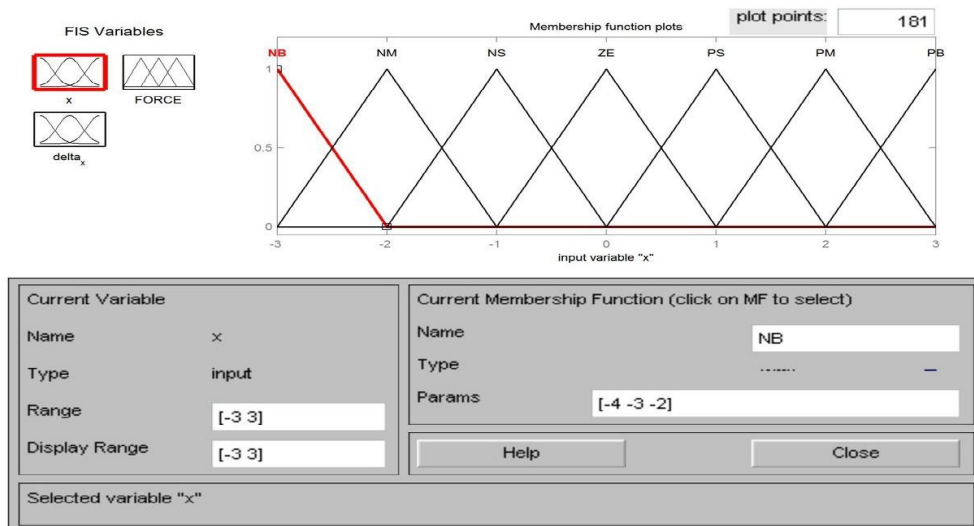


Figure 4.5 Input 1 of Fuzzy Logic Controller 1

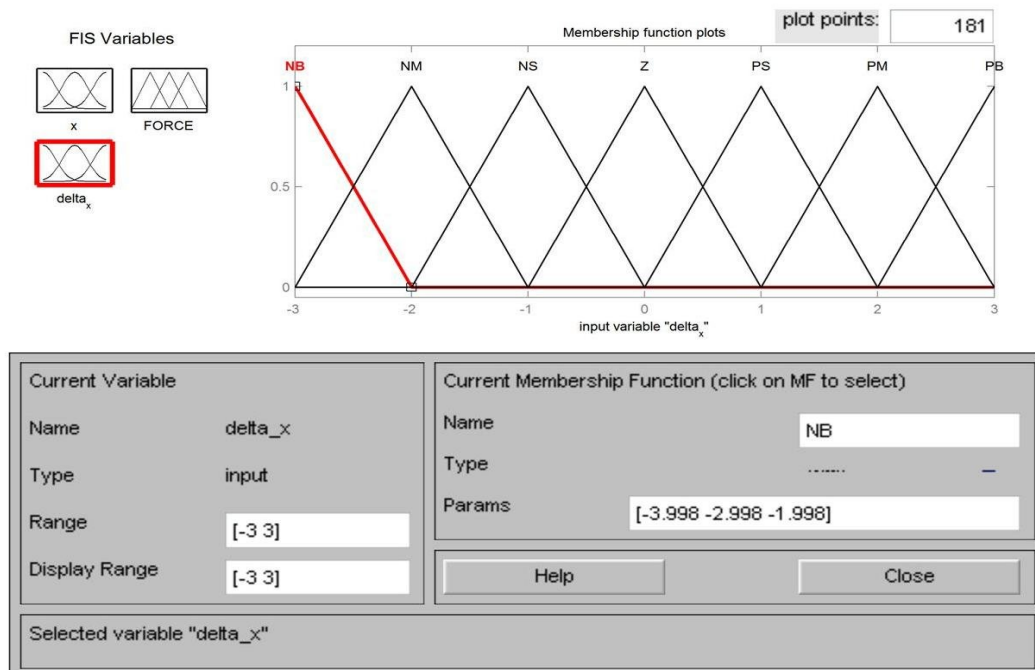


Figure 4.6 Input 2 of Fuzzy Logic Controller1

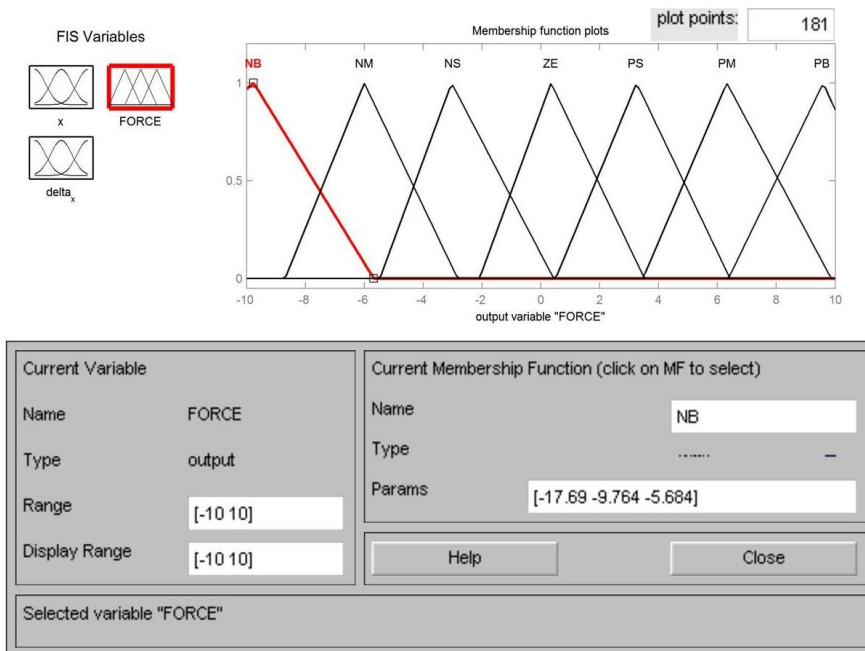


Figure 4.7: Output of Fuzzy Logic Controller 1

Membership functions for each of the fuzzy subsets of the control angle FLC

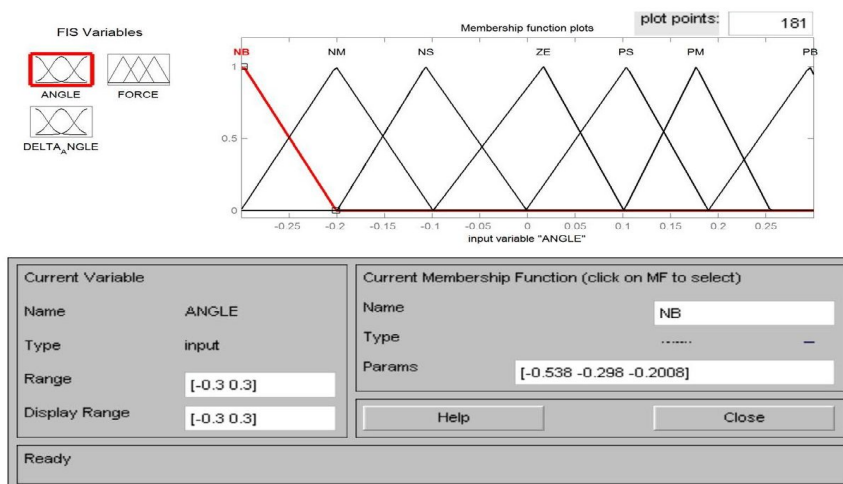


Figure 4.8 Input 1 of Fuzzy Logic Controller 2

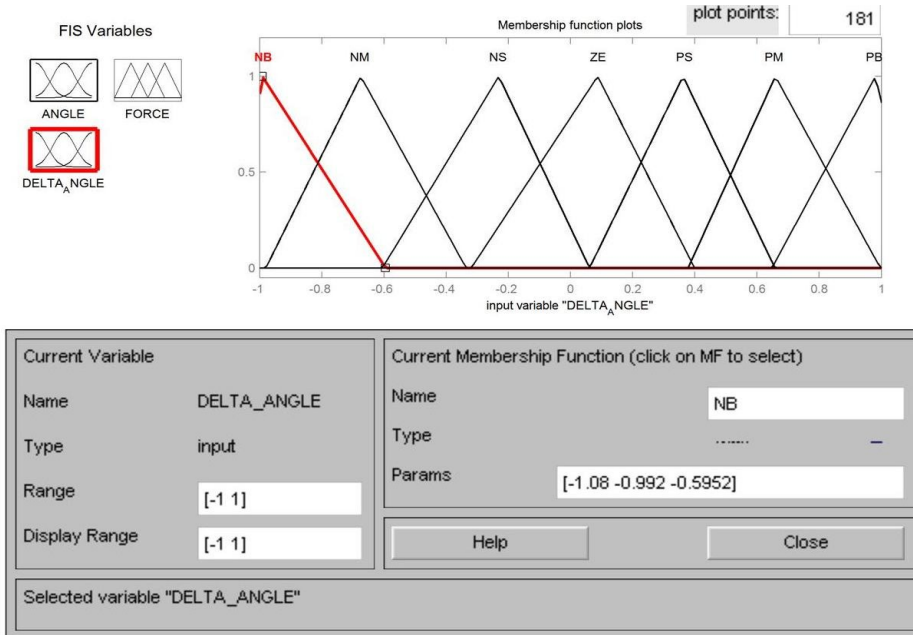


Figure 4.9 Input 2 of Fuzzy Logic Controller 2

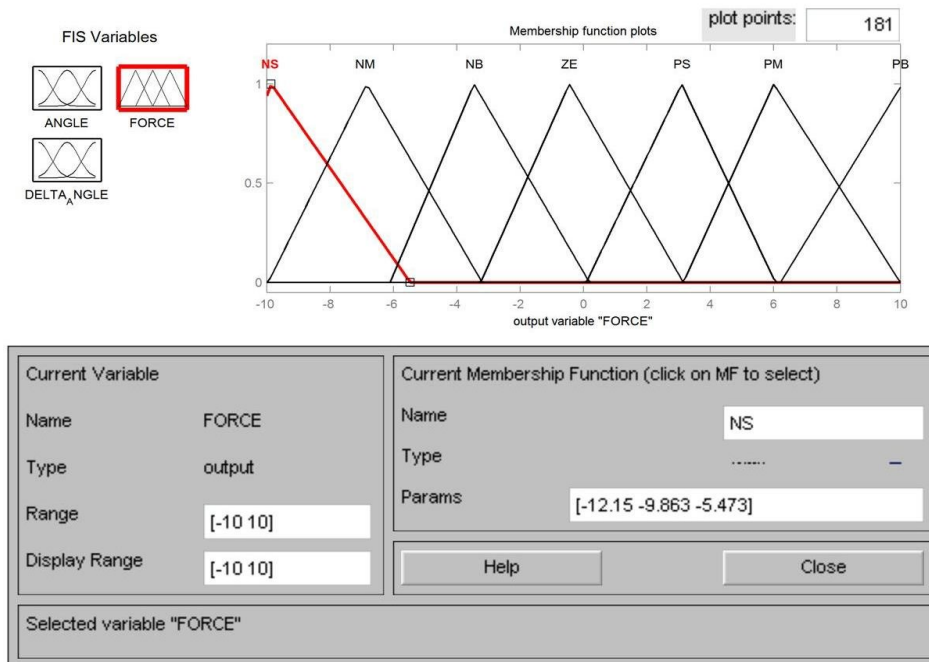


Figure 4.10 Output of Fuzzy Logic Controller 2

4.6 Rule base

The following rule base is formed for two fuzzy controllers which is applied for simulation study to control the inverted pendulum.

Table 4.3 Fuzzy rule matrix for control position FLC

		Position(X)						
		NB	NM	NS	ZE	PS	PM	PB
Del Position (ΔX)	NB	NB	NM	NS	NS	PS	PM	PB
	NM	NULL	NM	NS	NS	PS	PM	NULL
	NS	NULL	NULL	NS	NS	PS	NULL	NULL
	ZE	NB	NM	NS	ZE	PS	PM	PB
	PS	NULL	NULL	NS	PS	PS	NULL	NULL
	PM	NULL	NM	NS	PS	PS	NULL	NULL
	PB	NB	NM	NS	PS	PS	NULL	PB

Table 4.4 Fuzzy rule matrix for control position FLC

		Angle(θ)						
		NB	NM	NS	ZE	PS	PM	PB
Del Angle ($\Delta \theta$)	NB	NB	NM	NS	NS	PS	PM	PB
	NM	NULL	NM	NS	NS	PS	PM	NULL
	NS	NULL	NULL	NS	NS	PS	NULL	NULL
	ZE	NB	NM	NS	ZE	PS	PM	PB
	PS	NULL	NULL	NS	PS	PS	NULL	NULL
	PM	NULL	NM	NS	PS	PS	NULL	NULL
	PB	NB	NM	NS	PS	PS	PM	NULL

The output of the controllers is the forces which will be summed up together to produce equivalent force to the plant system. The tuning process of the FLC is more difficult as compared conventional controllers like PID controllers. This is because there are a lot of

parameters need to be implemented of Fuzzy logic Controller of inverted pendulum in Simulink.

The above mentioned model of inverted pendulum may be used as subsystem to give out the values of angle and position which is shown in the following figure. The simple non linear inverted pendulum with position, delta position .angle and delta angle is also shown below by the help of Simulink. Figure 4.11 shows the schematic diagram of the Nonlinear Inverted Pendulum System with two FLC controllers in Simulink. As can be seen from the figure, there are 2 FLC's in the diagram. Each of them has a function to control the position of a cart and to control the angle of the pendulum respectively considered such as membership functions, gains, range values and etc.

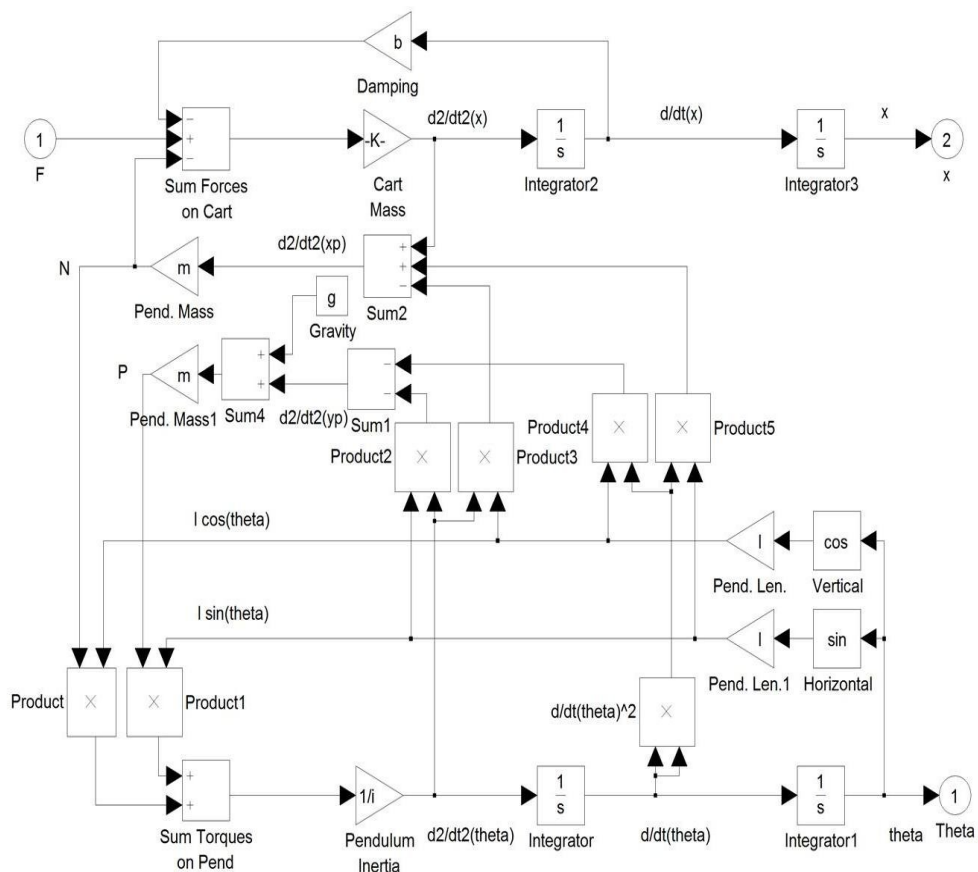


Figure 4.11 Inverted Pendulum in Simulink

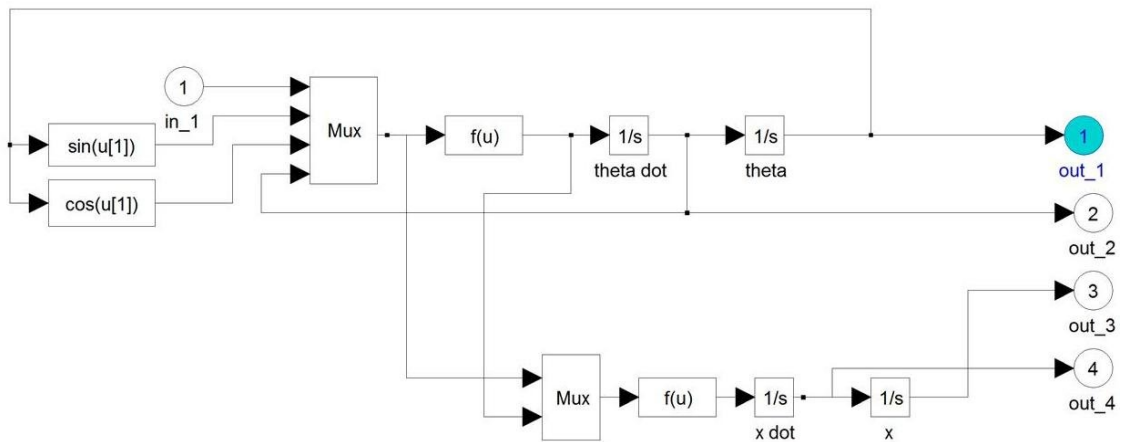


Figure 4.12 Inverted Pendulum's pole dynamics with cart

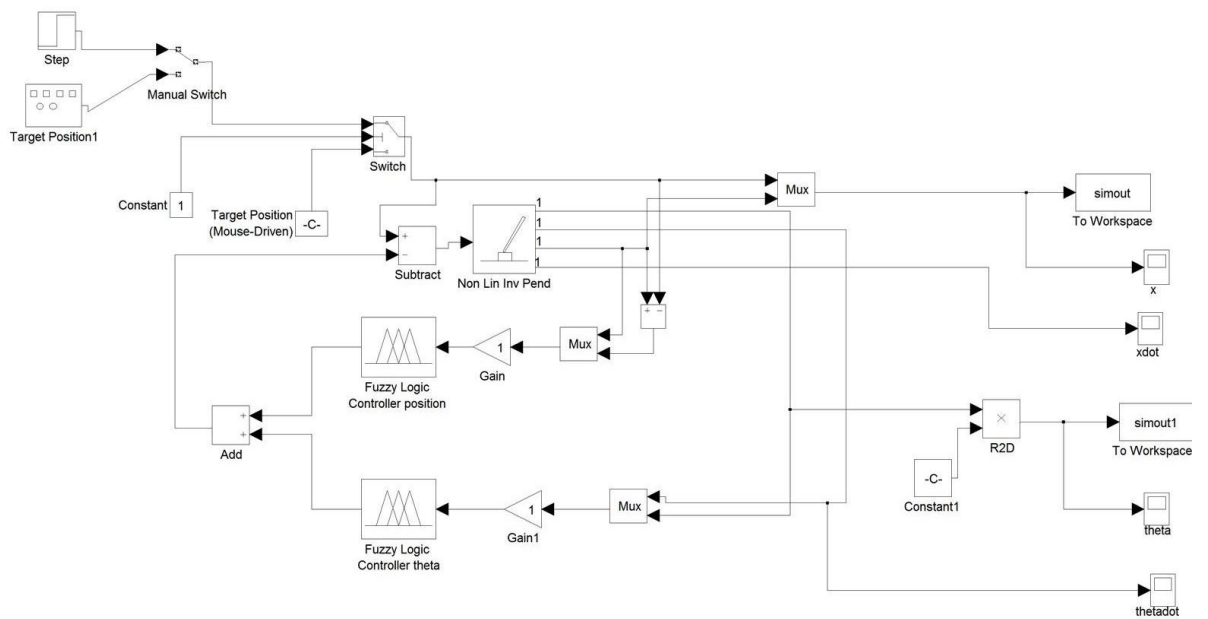


Figure 4.13 Control of Inverted Pendulum with fuzzy controller in Simulink

The above mentioned case is described by using Mamdani inference system mechanism but if we have to use the ANFIS using fuzzy controllers then we have to use Sugeno type of inference mechanism as because of the fact that ANFIS supports only Sugeno type. So we have to design a neuro fuzzy controller in which output function is linear combination

of input variables i.e in this the inference mechanism is proceeded by taking help of Sugeno type.

4.7 Conclusion

As a conclusion, the control objective in stabilizing the inverted pendulum has been achieved by using two block of Fuzzy logic controller. This can be verified through the output responses of the system which satisfy the design criteria. The controller design could stabilize the pendulum rod and keeping the cart in a desired position. Fuzzy logic has provided a simple way without going through the mathematical approach as conventional controller in order to arrive at a definite conclusion based upon nonlinear and an unstable of inverted pendulum system.

INVERTED PENDULUM WITH ANFIS

5.1 Introduction

The ANFIS architecture is completely equivalent to fuzzy inference system (fuzzy controller when used in control system). However by implementing as an ANFIS we can easily employ the backpropagation type learning procedure to find its parameter for achieving a minimal error measure. The plant can be implemented as a concise adaptive network if the difference or the differential equations of the plant dynamics are available. Based on the adaptive rule of the adaptive neural network the fuzzy controller can evolve automatically to acquire desired membership function of fuzzy if then rule for achieving control goal.

The block diagram of feedback control system in discrete time domain as shown in figure 5.1 where the delay through the controller is assumed to be small and plant state variable are assumed to be accessible with accuracy, for an example in a regulator control problem, the control goal is to keep the plant state as close possible to desired zero state, or the variable.

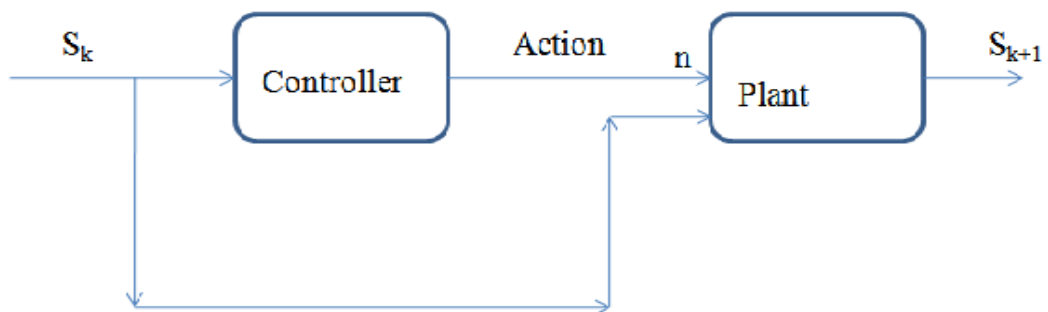


Figure 5.1 Block diagram of discrete control system

Before finding a way to derive the plant state to zero state in above figure at first the mathematical expression for both the controller block and for the plant block are calculated. First step is referred to as implementation of the block diagram. As in our case the controller is fuzzy controller so the obvious candidate for implementing the controller is to use the ANFIS architecture. The reason is for the three cause

- (1) The function behavior of the ANFIS is exact the same as a fuzzy controller with Takagi-Sugeno's fuzzy if-then rules.
- (2) The learning rule of ANFIS provides an effective method for achieving the desired input -output mapping.
- (3) An ANFIS is actually an universal approximator provided the rule no is not restricted.

As for the implementation of the plant block any mathematical expression that can well initiate the input-output behavior of the plant are qualified candidate. From this point of view, the selection of plant implementation is quite flexible and it all depends on how well we know about the plant, for an instance if we know nothing about the plant, except its input output data pairs then ANFIS is best suited candidate to solve out the problem. But on the other hand, if plant dynamic equation are available with deterministic coefficient then we go for simple adaptive network based on numerical method (such forward Euler integration, Runge Kutta method etc) to implement in the plant block.

5.2 Applying ANFIS to Inverted Pendulum

It doesn't matter that what method we choose for the plant block the figure 5.1 is still an adaptive network in which learning rule can be applied to find the desired input-output pair mapping from \mathbf{S}_k to \mathbf{S}_{k+1} for all k . Since the desired state in our case is the zero vector, the training data is of the form

$$(\mathbf{S}; \mathbf{0}) \quad 5.1$$

Where \mathbf{S} is the input part vector and $\mathbf{0}$ is the output vector after small time step h . In other words given any state \mathbf{S} at time t , and expect that the controller will drive the plant to a

state at time $t+h$ as possible as close to zero vector. The desired goal is not achieved in a single time step h but we have to do no of iteration i.e. a collective effect of training data is required to make out reach the plant state at origin.

Now considering the ANFIS with our problem of inverted pendulum as shown in the following figure 5.2 Here our control goal is to keep balancing the ploe by supplying the appropriate force F to the cart.

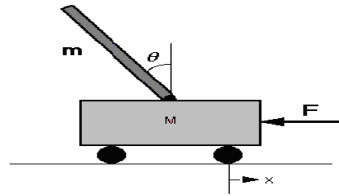


Figure 5.2 Block diagram of Inverted Pendulum

Let $x_1(t) = \theta(t)$ i.e. angle of the pole with respect to vertical and let $x_2(t) = \dot{\theta}(t)$ i.e. the angular velocity of the pole, then the state equation can be expressed as $\dot{x}_1 = x_2$

$$\dot{x}_2 = \frac{g \sin(x_1) + \cos(x_1) \left(\frac{-F - m l x_2^2 \sin(x_1)}{M+m} \right)}{l \left(\frac{4}{3} \frac{m \cos^2(x_1)}{M+m} \right)} \quad 5.2$$

$$= H_2(x_1, x_2, F) \quad 5.3$$

Since the inverted pendulum is the well behaved non linear dynamical system with precisely defined differential equation, so we can use a two step forward Euler integration to approximate its state at time $(t+h)$

$$x_1(t + 0.5h) = 0.5h x_2(t) + x_1(t) \quad 5.4$$

$$x_2(t + 0.5h) = 0.5h H_2(x_1(t + 0.5h), (x_2(t + 0.5h), F) + x_2(t + 0.5h)$$

Where $h=0.5$ sec , these above four equations are node function of node 1,2,3 & 4 respectively as shown in the plant block of following figure 5.3. Here F is a function of x_1 and x_2 and it is kept constant during the period $[t, t+h]$. The controller block is implemented as an ANFIS with 4 rules and each rule is assigned with two membership function. But in general there is no way to tell in advance that how many rules are required to achieve the control goal. In simulation the rule no is arbitrarily set to four and control goal is achieved.

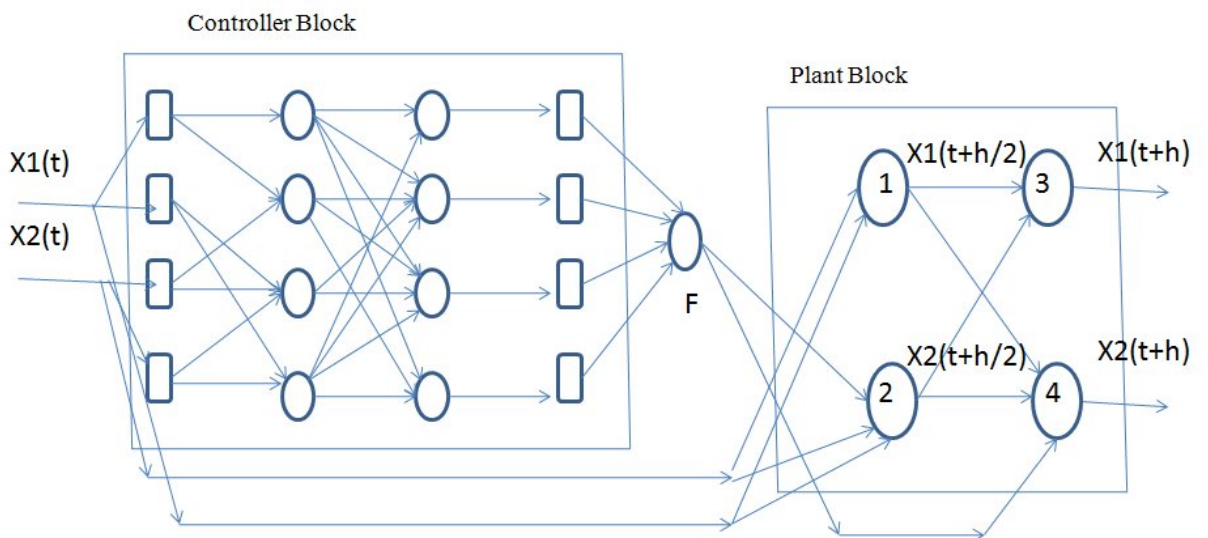


Figure 5.3 Implementation of Block Diagram

The inverted pendulum is expected to start from initial point somewhere from the origin in state space, therefore the input part of training data are chosen to scatter evenly over a square region in a state space.

We assume that we have no prior knowledge about the inverted pendulum system which is available. Therefore the consequent parameters are all set to zero showing that the control action is zero initially. The premise parameter is set in such a way that the membership function can cover the operating range totally with sufficient overlapping with each other. Here the θ is usually of higher order than that of ϕ . The learning process

converges within less than 45 epoches. Infact only after 10 epoches of learnig the fuzzy controller thus obtained can even balance the pole already.

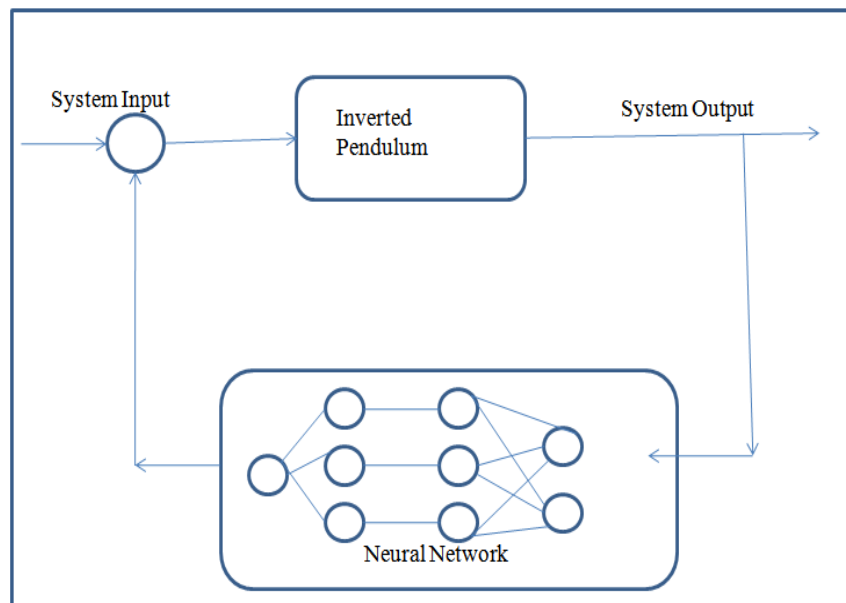


Figure 5.4 Simulink diagram of balance control by ANFIS

5.3 Control Strategy

Now for the control of inverted pendulum ANFIS is used. ANFIS employs a type of inference mechanism which is slightly differently as used in FIS, here Sugeno type of inference mechanism is used to design the controller. In Sugeno inference mechanism the output is linear combination of input membership function. The position controller and angle controller uses 2 input and one output and for that rule base is also formed. Following are the figures which show that how we have used the position and angle as to control the inverted pendulum.

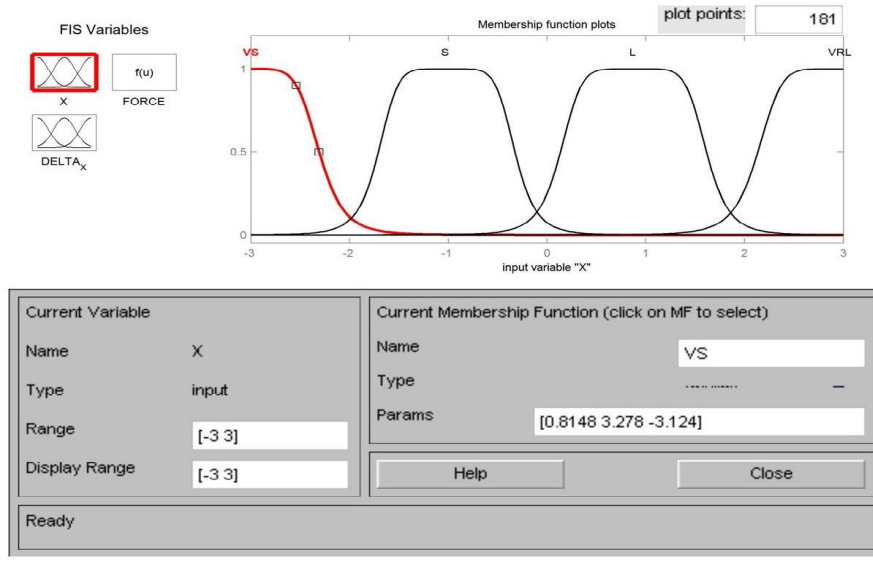


Figure 5.6 Input 1 of TSK for ANFIS implementation of position control

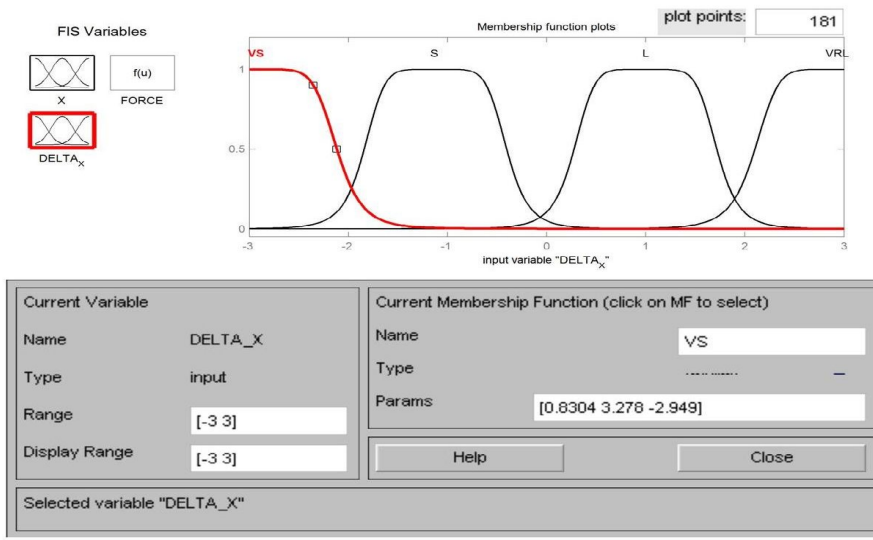


Figure 5.7 Input 2 of TSK for ANFIS implementation of position control

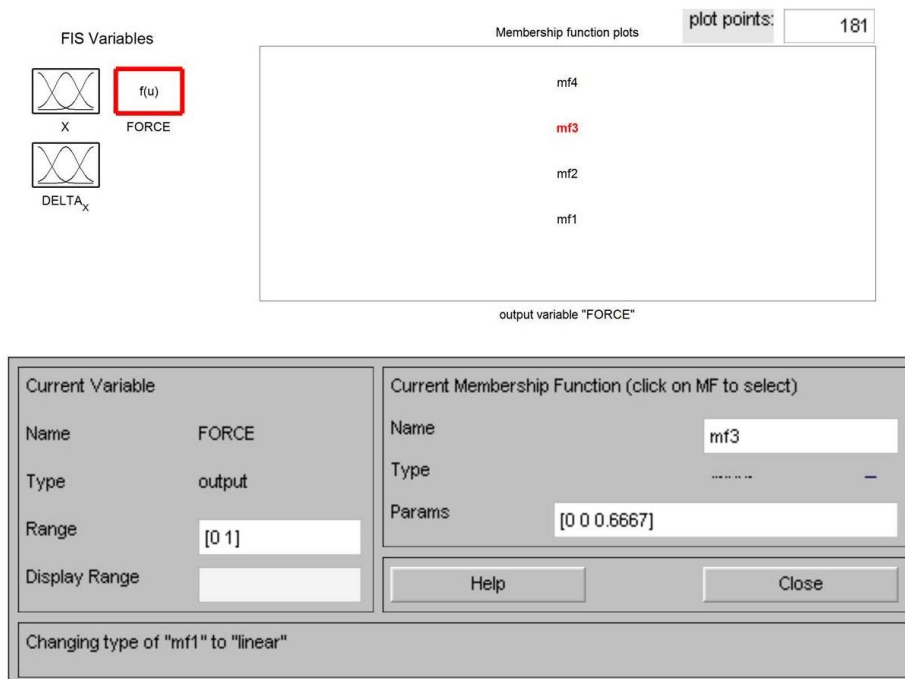


Figure 5.8 Output of TSK for ANFIS implementation of position control

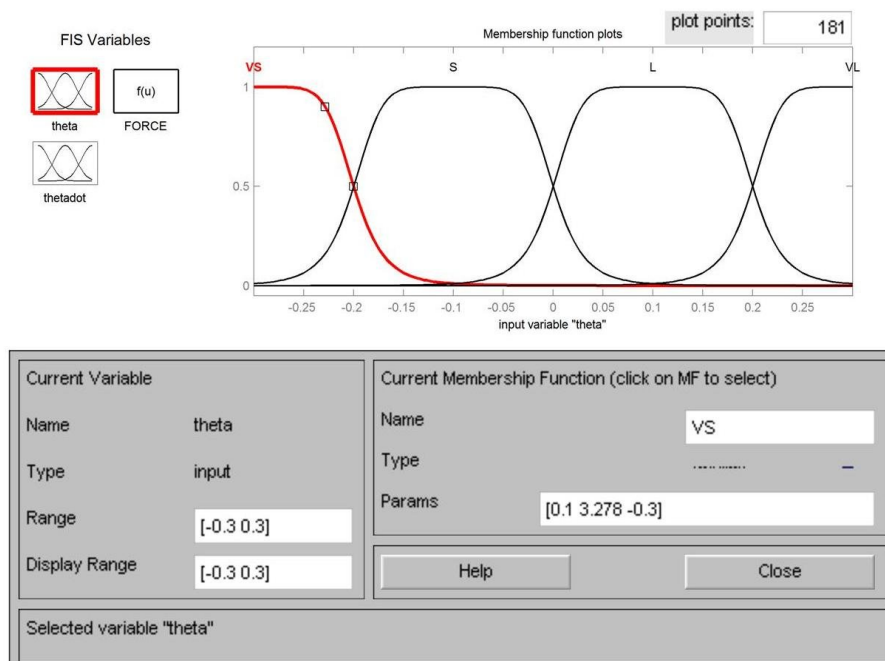


Figure 5.9 Input 1 of TSK for ANFIS implementation of angle control

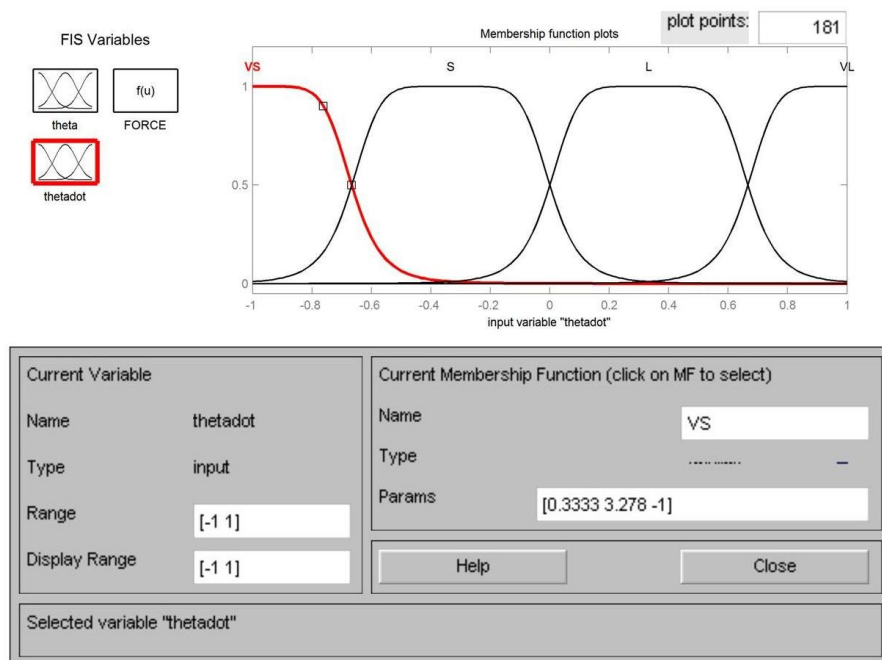


Figure 5.10 Input 2 of TSK for ANFIS implementation of angle control

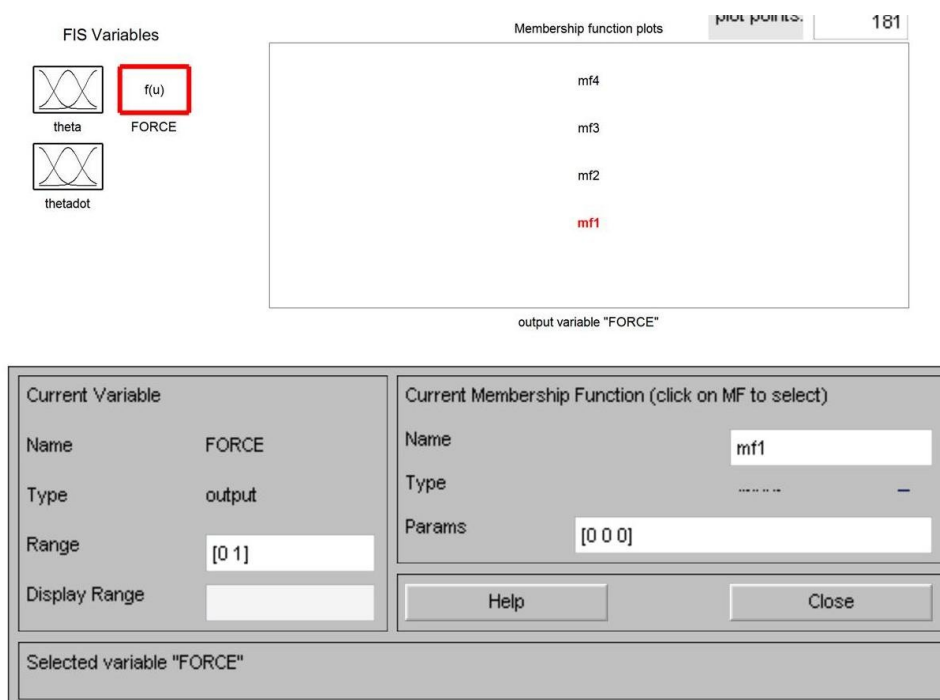


Figure 5.11 Output of TSK for ANFIS implementation of position control

5.4 Conclusion

The ANFIS is applied here for controlling of inverted pendulum in which two different controllers are used to train the system to get the desired optimum output. We have used two input pair of input to get a desired single output which is based on Sugeno inference system for both of the controllers. The simulation scheme is presented in the result and discussion chapter.

RESULTS AND DISCUSSION

6.1 Open Loop Results

As mentioned in Chapter 3, the mathematical model and equations using the transfer function and state-space of the inverted pendulum have been determined. By implementing all these equations into Matlab M-file command, the following results could be seen. From the Figure 6.1 to Figure 6.5, it can be noticed that the inverted pendulum system is not stable without controller. The curve of the pendulum's angle and cart's position were approached infinity as the time increases. Therefore, some controllers need to be designed in order to stabilize the system

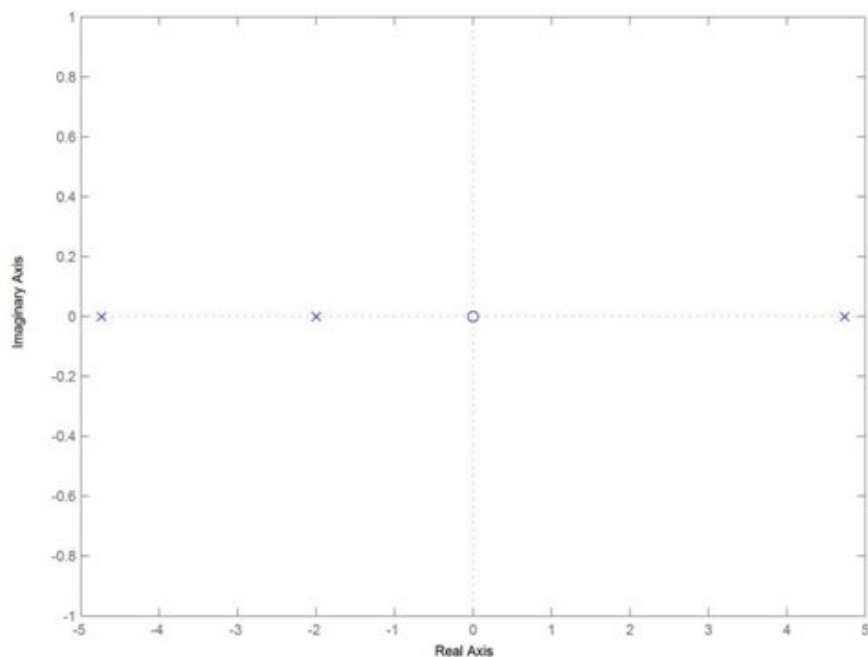


Figure 6.1 Open loop pole zero plot of Inverted Pendulum

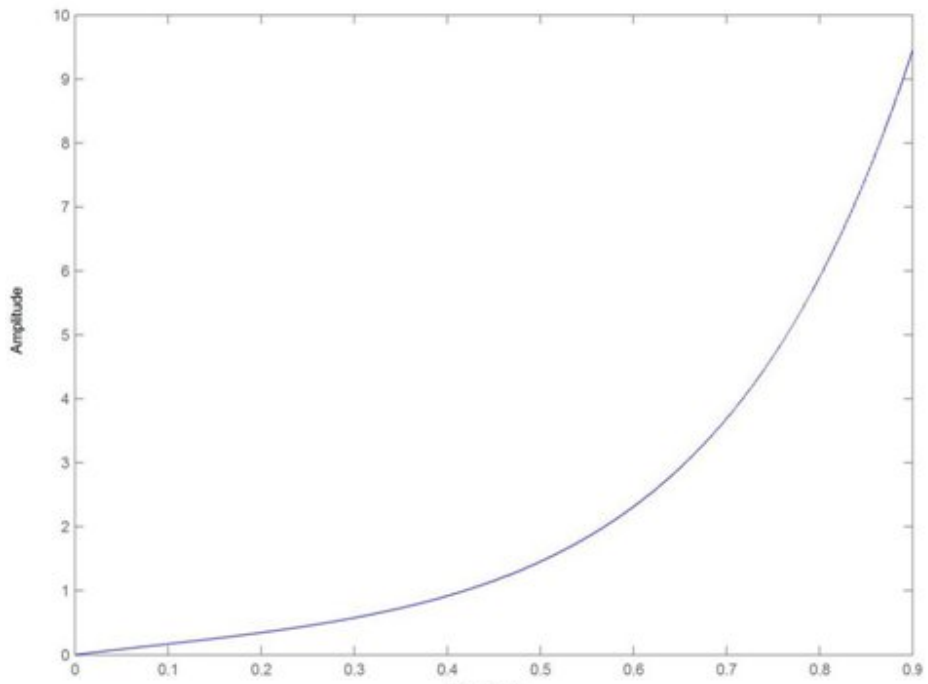


Figure 6.2 open loop impulse response of Inverted Pendulum

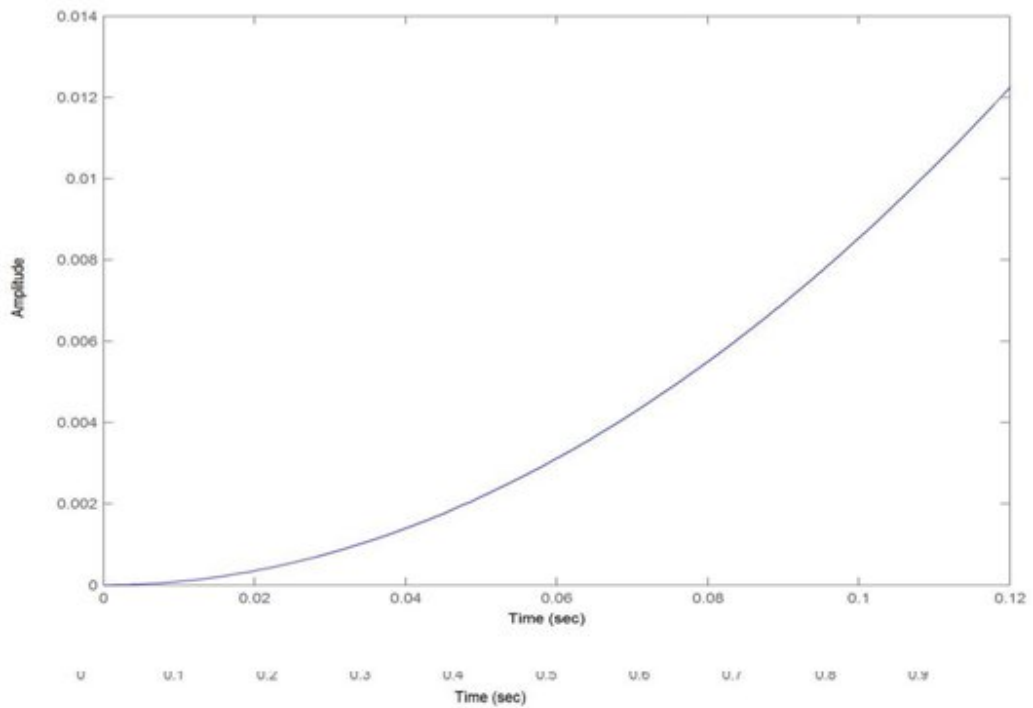


Figure 6.3 Open loop step response of Inverted Pendulum

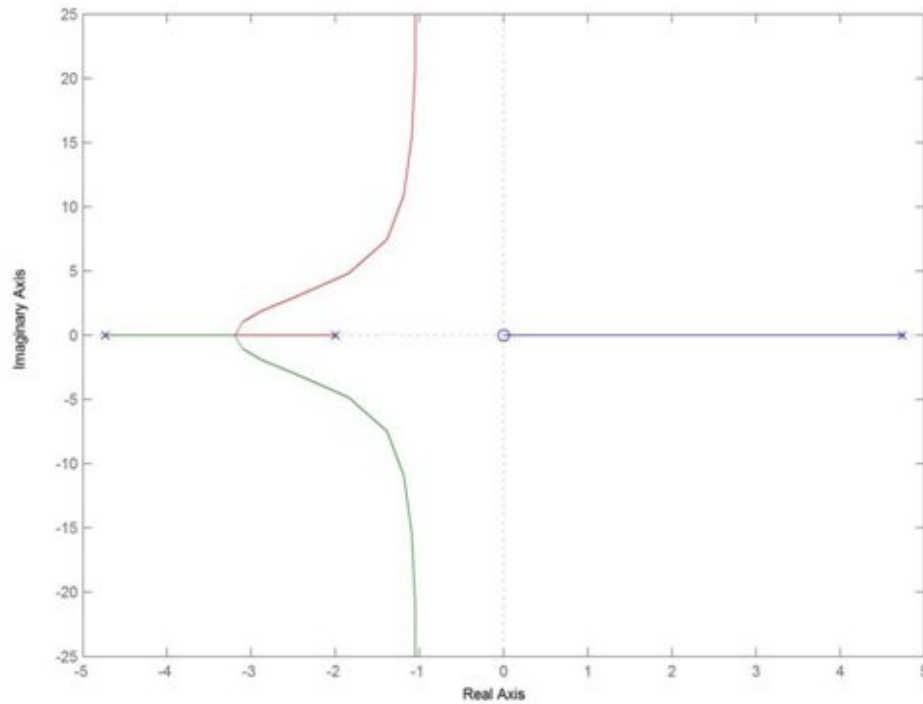


Figure 6.4 Root locus plot for open loop inverted pendulum

6.2 Fuzzy Logic Controller Control Method

The above discussed uncontrolled inverted pendulum is open loop type of control application. Here the graphs obtained from the transfer function of the system reveals that the system is highly non linear. So to linearise the system we have used the fuzzy logic as it is easy to deal with with better characteristics as when dealt with conventional controller.

6.2.1 Pendulum's Angle and Cart's Position

As discussed in Chapter 3 & 4, four inputs have been applied to the FLC. The FLC was divided into two FLC's so that each FLC can receive two inputs from the system. The output of each of the FLC were summed up together to produce the equivalent force. This force was fired as the input to the system. Figure 6.1 and Figure 6.2 show the output responses (angle and position) of the pendulum and cart respectively. Figure 6.3 and

Figure 6.4 represents the system output response of cart position and angle of pendulum from the vertical.

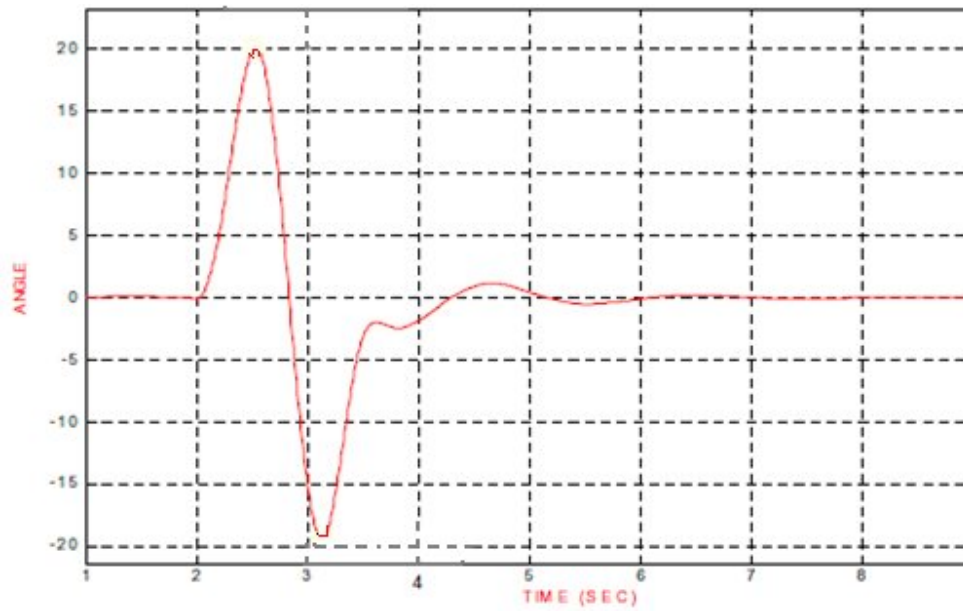


Figure 6.5 Step response for pendulum's angle

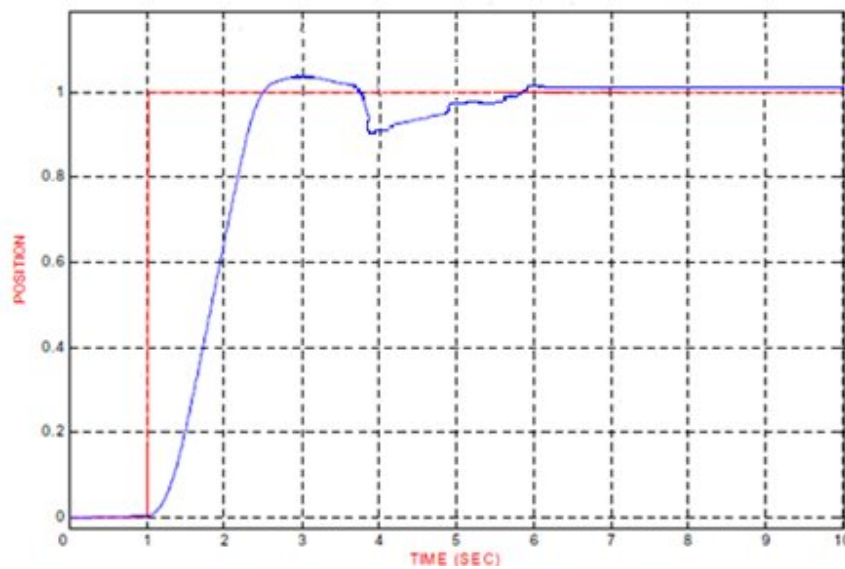


Figure 6.6 Step response for cart's position

Applying the m-file command into matlab, controller block would be generated. The block was exported into simulink diagram so that the simulation could be operated and the system response could be analyzed. Figure 6.3 and Figure 6.4 show the system output response of cart's position and the system output response of pendulum's angle respectively.

6.3 ANFIS Training Parameters

Following are the tables and graphs of the training errors with different values for the range of influence for both the position and angle control. We have seen from the obtained graphs that as the value of range of influence is decreased the training error also get decreased. So there comes a point at which the error becomes saturated with increased no of epochs.

Table 6.1 ANFI'S parameters for position control

Position controller			
S . No.	No of Epoches	Range of influence	Training error
1.	30	0.50	0.77391
2	30	0.48	0.62786
3	30	0.46	0.56885
4	30	0.44	0.4672
5	30	0.42	0.29009
6	30	0.41	0.24947
7	30	0.39	0.32321
8	30	0.37	0.38514
9	30	0.35	0.29741
10	10	0.32	0.22532

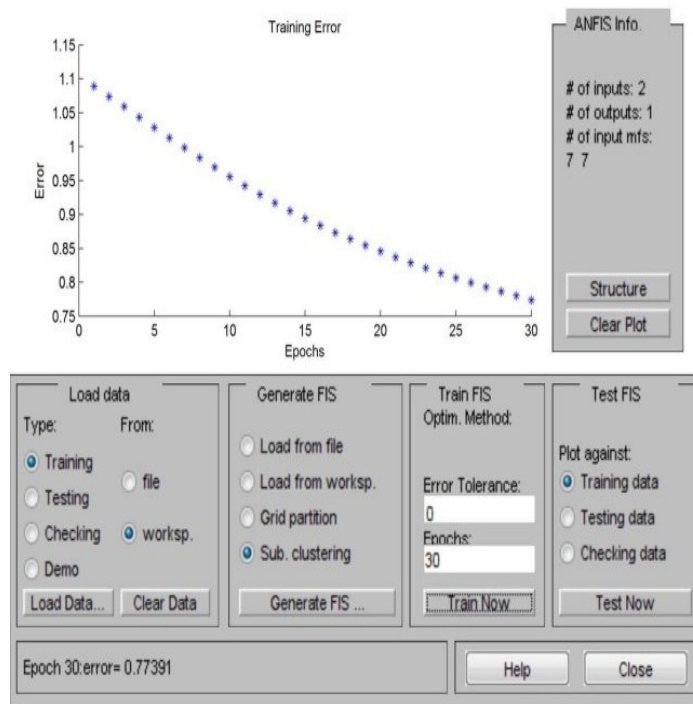


Figure 6.7 Training error when ROI is 0.50

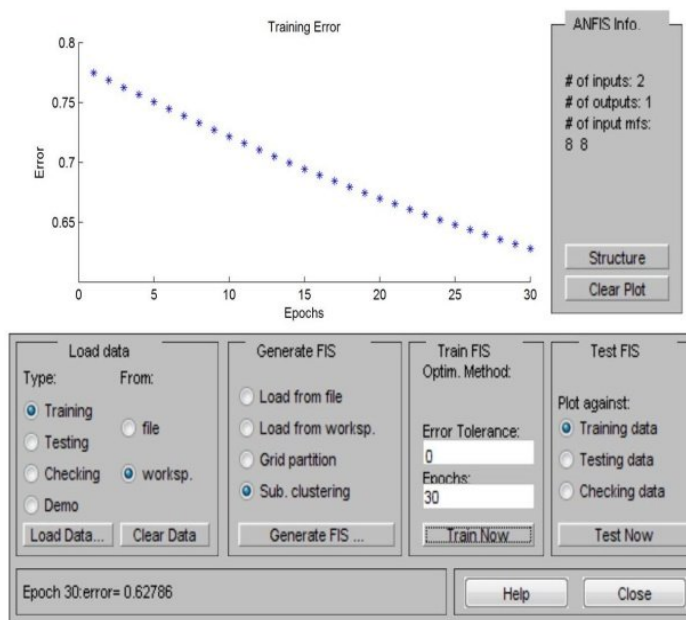


Figure 6.8 Training error when ROI is 0.48

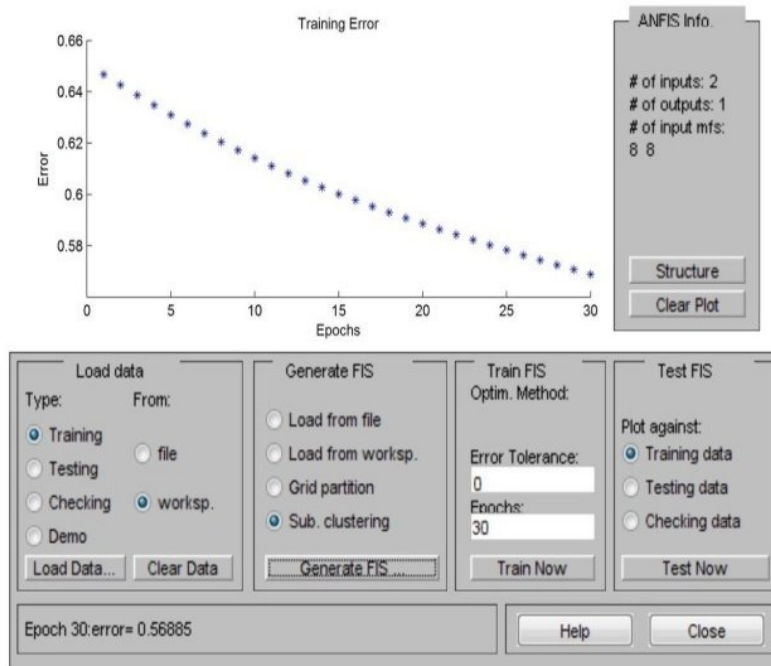


Figure 6.9 Training error when ROI is 0.46

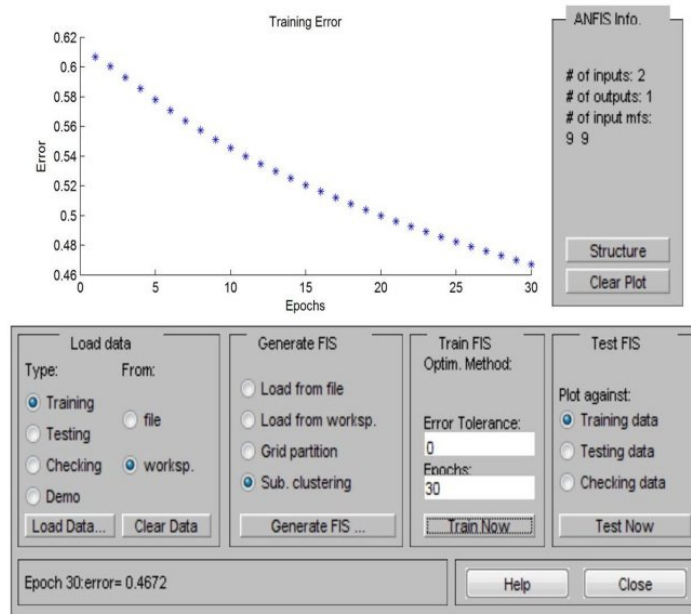


Figure 6.10 Training error when ROI is 0.44

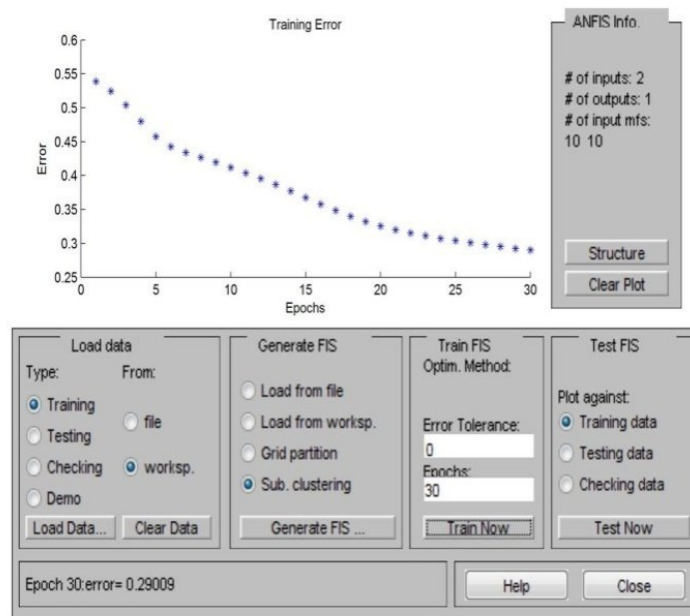


Figure 6.11 Training error when ROI is 0.42

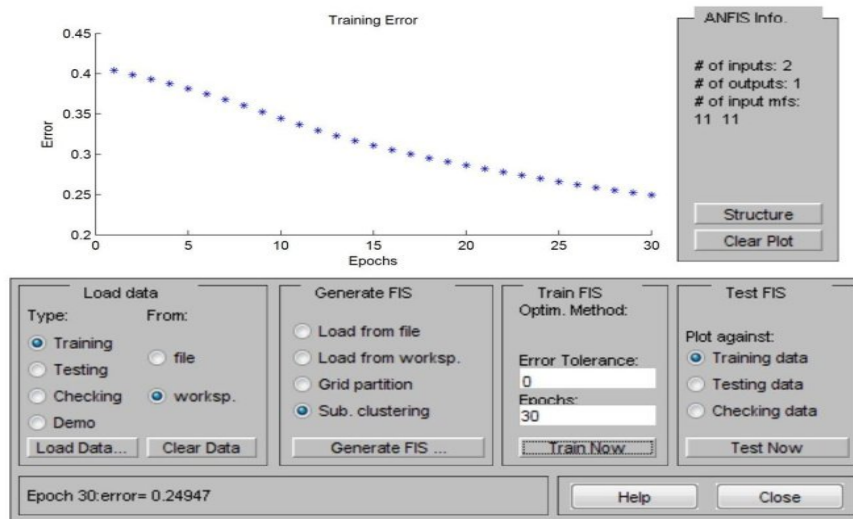


Figure 6.12 Training error when ROI is 0.41

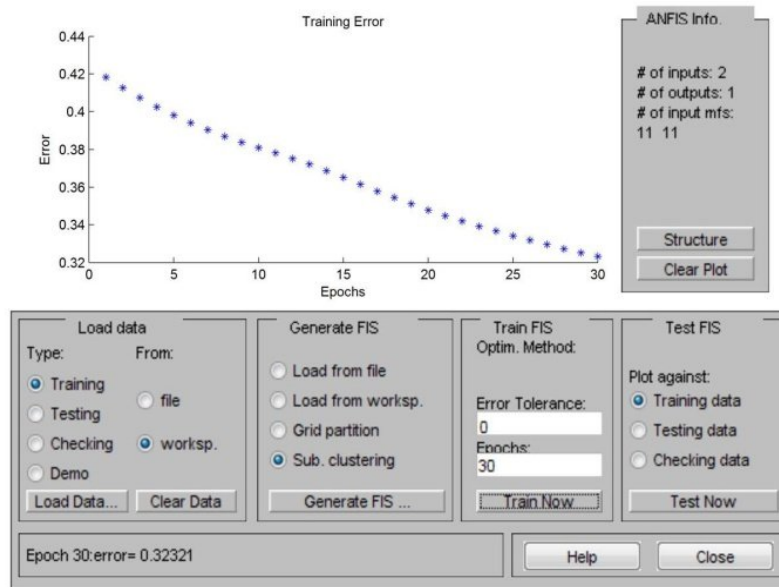


Figure 6.13 Training error when ROI is 0.39

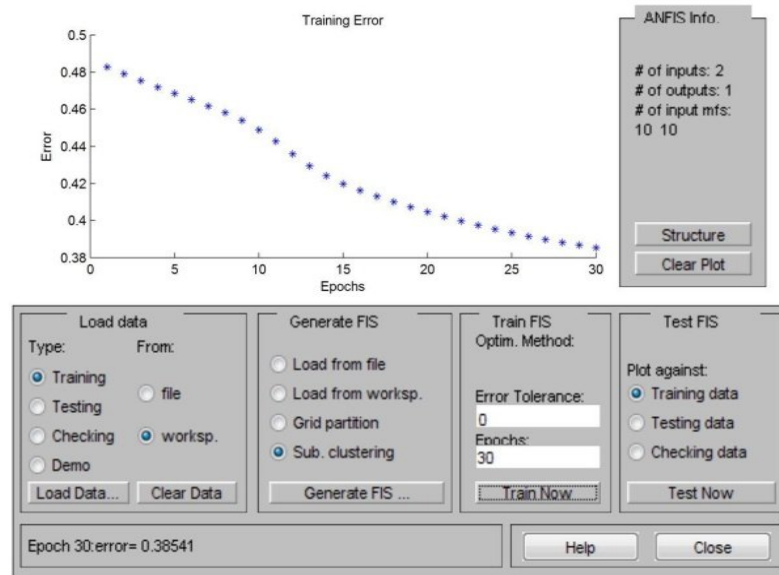


Figure 6.14 Training error when ROI is 0.37



Figure 6.15 Training error when ROI is 0.32

Table 6.2 ANFI'S parameters for angle control

Angle controller			
S . No.	No of Epoch's	Range of influence	Training error
1.	30	0.30	0.97639
2	30	0.27	0.47207
3	30	0.23	0.20726
4	30	0.21	0.1705
5	30	0.18	0.034199
6	30	0.17	0.032298
7	30	0.10	0.018847
8	30	0.08	0.00054337
9	30	0.06	0.00057157
10	10	0.04	0.00051356



Figure 6.16 Training error when ROI is 0.30

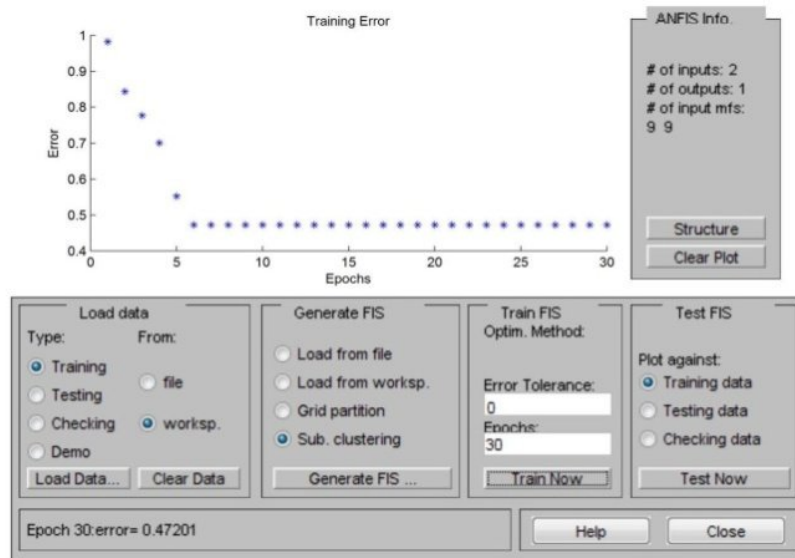


Figure 6.17 Training error when ROI is 0.27

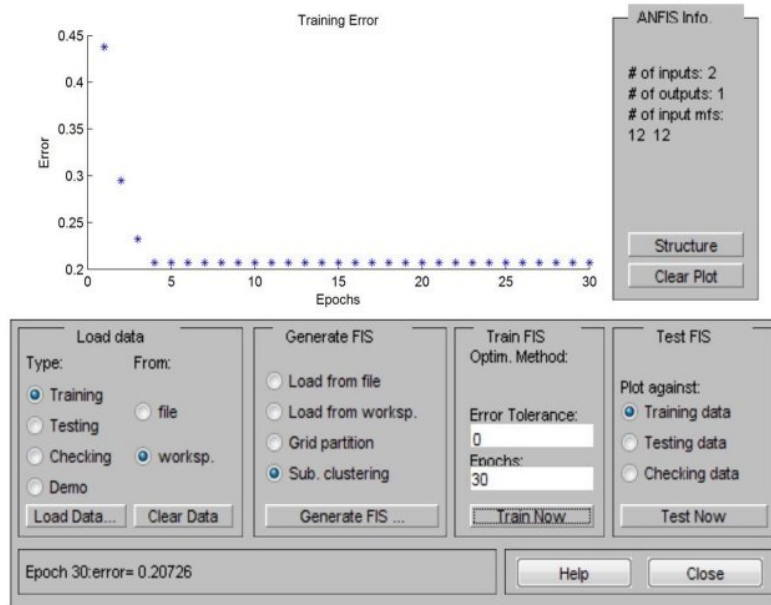


Figure 6.18 Training error when ROI is 0.23

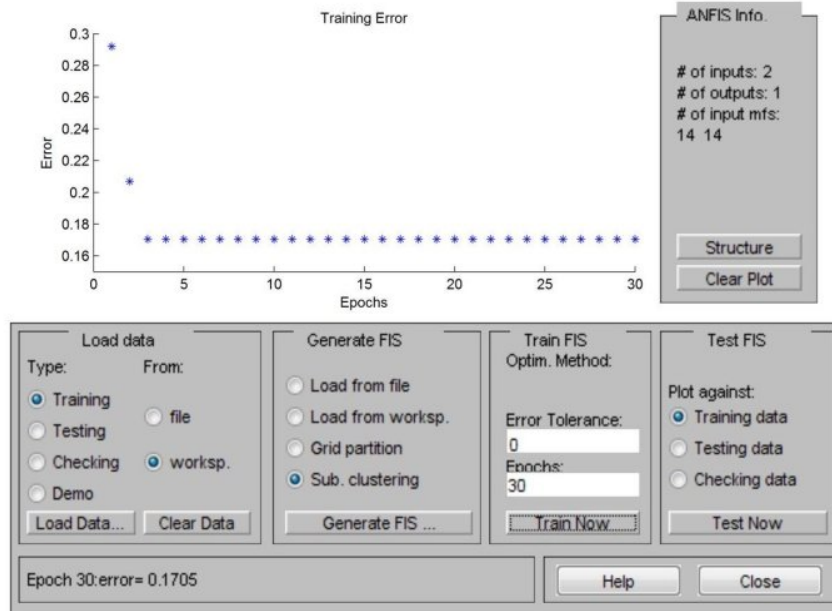


Figure 6.19 Training error when ROI is 0.21

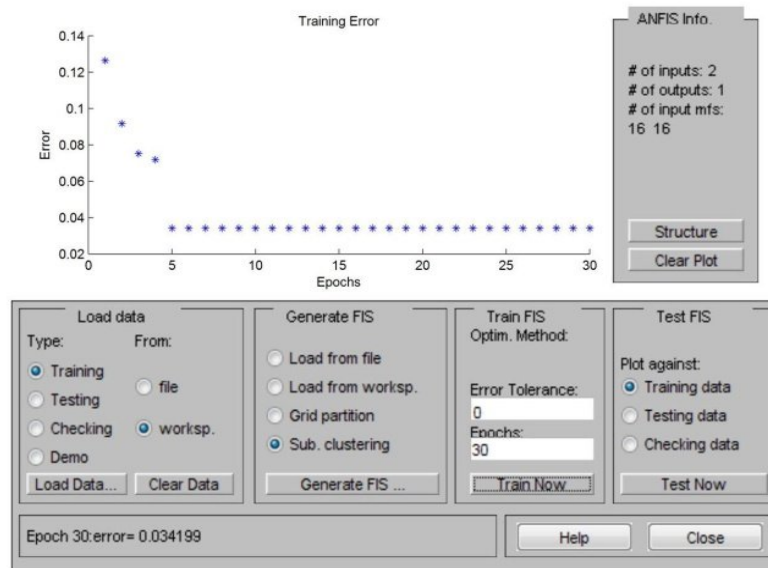


Figure 6.20 Training error when ROI is 0.18

6.4 Statistical Results

For the controller of position the average training error for whole data set considered here is found to be 0.421386. So on calculation the standard deviation of position is 0.030227. For the angle controller the average error from the obtained data is found to be 0.191319. So the standard deviation for the angle control is calculated to be as 0.0884708.

Table 6.3 Statistical results for position and angle control of IP

Position controller			Angle controller		
Average Training Error	Variance	Std. Deviation	Average Training Error	Variance	Std. Deviation
0.421386	0.17386	0.030227	0.191319	0.2974404	0.0884708

6.5 Graphical Results and Comparison

The graphical results showing the comparison of the control of inverted pendulum is presented here with two different possible control i.e. position control of cart and angle control of inverted pendulum both in ANFIS and FLC case. For the case of simplicity step response is used for the analysis

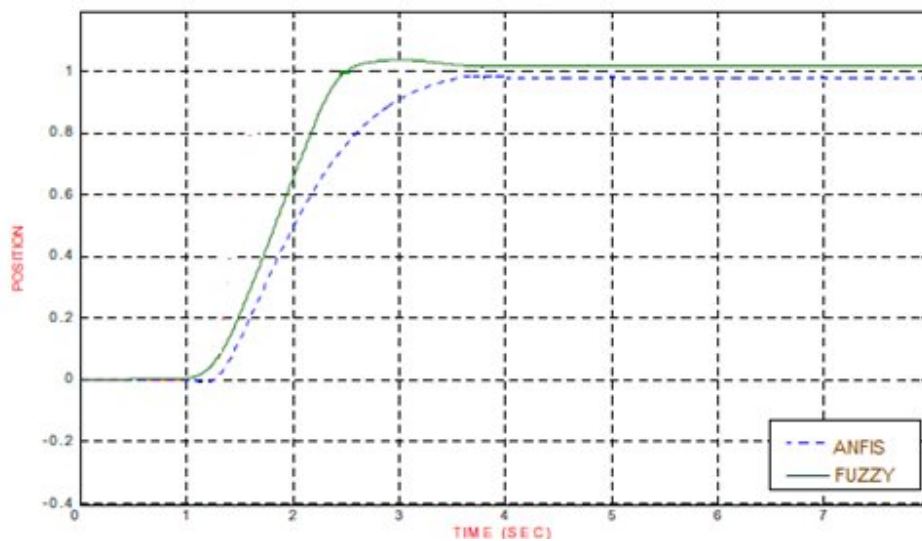


Figure 6.21 Comparison of output response of cart's position

Table 6.4 Comparison of output response of cart's position

Characteristics	Fuzzy	ANFIS
Rise time (T_r)	2.2 sec	2.6 sec
Settling time(T_s)	3.2 sec	3.5 sec
Steady State error	0.14	0.2
Maximum Overshoot	0.13	0

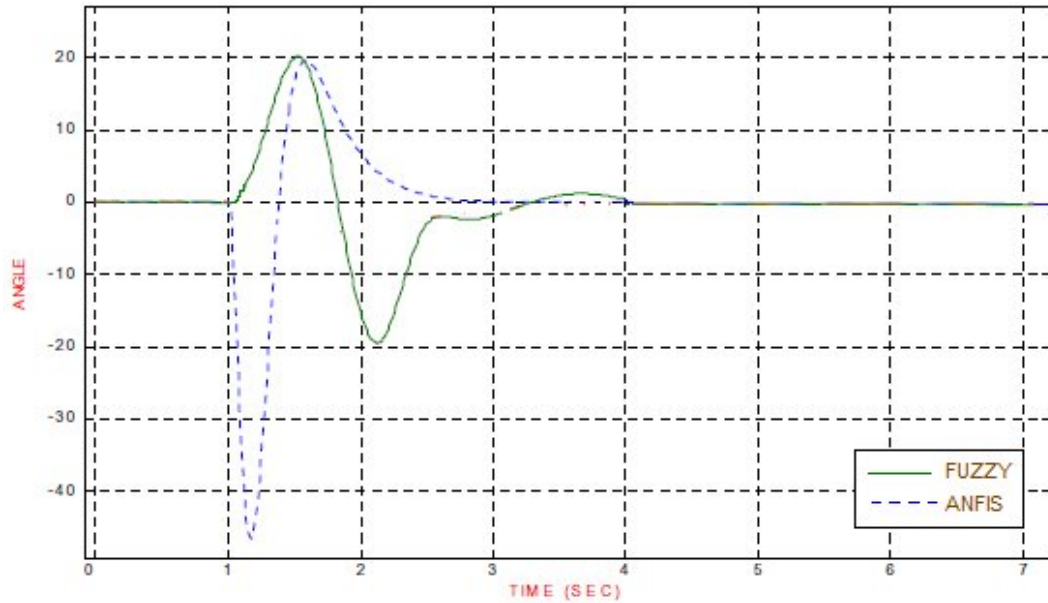


Figure 6.22 Comparison of output response of pendulum's angle

Table 6.5 Comparison of output response of pendulum's angle

Characteristics	Fuzzy	ANFIS
Settling time(T_s)	3.8 sec	2.3 sec
Steady State error	0	0
Maximum Overshoot	19° to -19°	19° to -38°

From the above table we can clearly say that while controlling pendulum of inverted pendulum the application ANFIS is much better than the simple FIS usage. The rise time and the steady state error is much suppressed as when it was controlled with FLC in position control. But when it was case of angle control the ANFIS again gives better performance than that of the basic FLC for the angle control also.

CONCLUSION AD FUTURE WORK

The stability and the response characteristics obtained here from both of the methods that is fuzzy approach and the ANFIS approach shows that ANFIS gives better performance when it comes to a non linear relationship between input and output. Thus if we have input output pairs of training data set and the system shows some nonlinear behavior the we can go for ANFIS approach. The control of inverted pendulum can also be done by applying some other neuro fuzzy architectural techniques like FALCON or GARIC or NEFCON or NEFPROX or FINEST or SONFIN or EFuNN or dmEFuNN. An integrated neuro-fuzzy model is interpretable and capable of learning in a supervised mode (or even reinforcement learning like NEFCON). In FALCON,GARIC, ANFIS, NEFCON, SONFIN, and FINEST the learning process is only concerned with parameter level adaptation within fixed structures. For large-scale problems, it will be too complicated to determine the optimal premise-consequent structures, rule numbers etc. User has to provide the architecture details (type and quantity of MF's for input and output variables),type of fuzzy operators etc. The programming approach like C, C++,Java may also be used to control the pendulum of inverted pendulum.

Although the controllers has been successfully stabilized the system, implementation of the algorithm into real hardware is very important. Since FLC could stabilize the nonlinear system, it is expected that the FLC will work well if it is implemented into real system. Therefore, further work for the proposed approaches will have great advantage. AI controller technique should be improved so that a robust controller and a better response can be achieved. The limitation in a FLC design should be eliminated. A lot of techniques can be figure out from the previous papers and research that were successfully done.

REFERENCES

- [1] L A Zadeh. “Fuzzy Sets” *Information and Control* , Vol. 8, pp. 353-383, 1965.
- [2] P J King and E H Mamdani “The Application of Fuzzy Control System to Industrial Process”, *Automatica* Vol. 13 pp. 235-242, 1977.
- [3] E H Mamdani “Application of Fuzzy algorithms for control of simple dynamic plant”, in *Proceeding of IEEE* Vol. 121, pp. 1585-1588, 1977.
- [4] T. J. Procyk and E. H. Mamdani “A linguistic self-organizing process controller”, *Automatica*, Vol. 15, pp. 15–30, 1979.
- [5] T. Takagi and M. Sugeno, “Fuzzy identification of systems and its applications to modeling and control”, *IEEE Transaction on System, Man, Cybernetics*, Vol. SMC-15, no. 1, pp. 116–132, Jan. 1985.
- [6] *William McC Siebert, “Circuits, Signals, and Systems”, The MIT Electrical Engineering and Computer Science Series, MIT Press, 1986.*
- [7] Cheun Chien Lee, “Fuzzy Logic in Control System: Fuzzy Logic Controller - Part I & 11”, *IEEE Transactions on System, Man, and Cybernetics*, Vol. 20, No. 2, pp. 403-435,1990.
- [8] Derrick H Nguyen and Bernard Widrow “Neural Network for self learning Control System”, *IEEE Transactions on Control System Magazine*, Vol. 15 pp. 18-23, April 1990.
- [9] M.M. Gupta and Ching J. Qi (1991) “Fusion of Fuzzy Logic and Neural Networks with Applications to Decision and Control Problems”, in *Proceedings of the American Control Conference*, pp. 30-31. 1991.

- [10] Jyh-Shing and R. Jang. “Self-Learning Fuzzy Controllers Based on Temporal Back propagation”, *IEEE Transactions on Neural Networks*, Vol.3, No.5, pp. 714-723, 1992.
- [11] Bart Kosko, “Fuzzy systems as universal approximators”, in *Proceedings of IEEE International Conference Fuzzy Systems*, pp. 1143–1162, Mar. 1992.
- [12] L. X.Wang, “Fuzzy systems are universal approximators”, in *Proc. IEEE International Conference Fuzzy Systems*, Mar. 1992, pp. 1163–1170.
- [13] Daniel G. Schwatz and George J. Klir “Application of Fuzzy Sets and Approximate Reasoning”, in *Proceeding of IEEE*, Vol. 82, No. 4, pp. 482-498, April 1994.
- [14] Piero P. Bonssone, Vivek Badami, Kenneth H. Chiang, Pratap S. Khedkar, Kenneth W. Marcelle, and Michel J. Schutten, “Industrial Applications of Fuzzy Logic at General Electric,” in *Proceeding of IEEE*, Vol. 83, No. 3, March 1995
- [15] M Ayoubi “Neuro Fuzzy structure for rule generation and application in the fault diagnostic of technical process”, in *Proceeding of the American Control Conference*. Vol.2. pp. 2757-2761, June 1995.
- [16] Kazuo Tanuka, Manabu Sano and Hiroyuki Watanabe “Modeling and Control of Carbon mono-oxide concentration using Neuro Fuzzy Techniques” *IEEE transaction on Fuzzy System*, Vol. 3, No. 3, pp. 271-279, August 1995.
- [17] Jeffery T. Spooner and Kevin M. Passino “Stable direct adaptive control using Fuzzy System and Neural Network” *IEEE Transaction on Fuzzy Systems*, Vol. 4, No.3pp 339-358, August 1996.
- [18] Wanatabe Keigo, Jun Tang Masatosi and Tosio Fakud “A Fuzzy Gaussian Neural Network and its application to mobile robot control” *IEEE Transaction on Control System Technology*, Vol. 4, No. 2, March 1996.

- [19] Jung Park and Suck Gyu Lee “Advanced Motion Control”, in *Proceeding of IEEE 4th International Workshop*, Vol. 2, pp. 619-624, 18-21 March 1996.
- [20] Payman Arabshahai, Jai Jo Choi, Robert J Marks II, Thomas P. Candel “Fuzzy Parameter adaptation in Optimization”, *IEEE Transaction on Computational Science and Engineering*, Vol. 6, No. 5, pp. 56-57, May 1996.
- [21] A. Trebi Ollenu and B. A. White “Robust output tracking for Multiple input and Multiple output system: An Adaptive Fuzzy approach” in *Proceeding of IEEE Control Theory Application*, Vol. 144, No. 6, pp. 537-544, November 1997
- [22] Yamakita, M. Hoshino, and T. Furuta, “Non Linear System Control Techniques”, in *Proceedings of American Control Conference*, Vol. 8, pp. 490 - 494 , June 1999.
- [23] J.S. Roger Jang, “Fuzzy Controller Design without Domain Expert”, in *Proceeding of AAAI of Control System Design*, Vol. 27, No. 3, pp. 205-213, June 2001.
- [24] Lawrence Bush, “Fuzzy Logic Controller for the Inverted Pendulum Problem”, *IEEE Control System Magazine*, pp. 238-245, November 27, 2001.
- [25] Mohanlal, P.P. Kaimal, and M.R. Vikram “Decision and Control”, in *Proceedings of the 41st IEEE Conference Sarabhai Space Centre*, Vol. 3, pp. 3255-3260, Dec. 2002.
- [26] Changsheng Yu and Li Xu “Development of Inverted Pendulum System and Fuzzy Control Based on MATLAB”, in *Proceedings of the 5th World Congress on Intelligent Control and Automation*, pp. 298-307, June 2004.
- [27] Dario Maravall, Changjiu Zhou, and Javier Alonso “Hybrid Fuzzy Control of the Inverted Pendulum via Vertical Forces”, *Wiley Periodicals Inc.*, pp. 635-641, March 2005.

- [28] Songmoung Nundrakwang, Taworn Benjanarasuth, Jongkol Ngamwiwit and Noriyuki Komine, “Hybrid controller for Swinging up Inverted Pendulum system”, in *Proceeding IEEE of System Man and Cybernetics*, Vol. 4, pp 750-758, June2005.
- [29] Keith B Oldham, “The Fractional Calculus: Theory and Applications of Differentiation and Integration to arbitrary order”, *IEEE transaction on fractional control*, Vol.12, pp. 435-437, June 2005.
- [30] L Rajaji , C. Kumar and M Vasudevan “Fuzzy and ANFIS based soft starter fed induction motor drive for high performance ” in *Proceeding of Asian Research Publishing Network Journal of Engineering and Applied Sciences* . Vol. 3 No. 4, August 2008.