ARTIFICIAL INTELLIGENCE (AI) CONTROLLER DESIGN FOR BEAM AND BALL SYSTEM

IBRAHIM BIN IDERUS

UNIVERSITI MALAYSIA PAHANG

IBRAHIM BIN IDERUS BACHELOR OF ELCTRICAL ENGINEERING (CONTROL & INSTRUMENTATION) 2008-11-17 UMP

UNIVERSITI MALAYSIA PAHANG

BORANG PENGESAHAN STATUS TESIS ♦

JUDUL: ARTIFICIAL INTELLIGENCE (AI) CONTROLLER DESIGN FOR BEAM AND BALL SYSTEM

SESI PENGAJIAN: 2008/2009

Saya **IBRAHIM BIIN IDERUS (850411-07-5279)**
(HURUF BESAR)

mengaku membenarkan tesis (Sarjana Muda/~~Sarjana~~ /~~Doktor Falsafah~~)* ini disimpan di Perpustakaan dengan syarat-syarat kegunaan seperti berikut:

1. Tesis adalah hakmilik Universiti Malaysia Pahang (UMP).
2. Perpustakaan dibenarkan membuat salinan untuk tujuan pengajian sahaja.
3. Perpustakaan dibenarkan membuat salinan tesis ini sebagai bahan pertukaran antara institusi pengajian tinggi.
4. **Sila tandakan (√)

[ ]	**SULIT**	(Mengandungi maklumat yang berdarjah keselamatan atau kepentingan Malaysia seperti yang termaktub di dalam AKTA RAHSIA RASMI 1972)
[ ]	**TERHAD**	(Mengandungi maklumat TERHAD yang telah ditentukan oleh organisasi/badan di mana penyelidikan dijalankan)
[√]	**TIDAK TERHAD**	

Disahkan oleh:

________________________ (TANDATANGAN PENULIS)

________________________ (TANDATANGAN PENYELIA)

Alamat Tetap:

1651 MK 14 BUKIT MINYAK
BUKIT MERTAJAM
14000 PULAU PINANG

AHMAD NOR KASRUDDIN
(Nama Penyelia)

Tarikh: **10 NOVEMBER 2008**

Tarikh: **10 NOVEMBER 2008**

CATATAN:
* Potong yang tidak berkenaan.
** Jika tesis ini SULIT atau TERHAD, sila lampirkan surat daripada pihak berkuasa/organisasi berkenaan dengan menyatakan sekali tempoh tesis ini perlu dikelaskan sebagai atau TERHAD.
♦ Tesis dimaksudkan sebagai tesis bagi Ijazah doktor Falsafah dan Sarjana secara Penyelidikan, atau disertasi bagi pengajian secara kerja kursus dan penyelidikan, atau Laporan Projek Sarjana Muda (PSM).

"I hereby acknowledge that the scope and quality of this thesis is qualified for the award of the Bachelor`s Degree of Electrical Engineering (Electronics)"

Signature : ___________________________

Name : EN AHMAD NOR KASRUDDIN BIN NASIR

Date : 10 NOVEMBER 2008

ARTIFICIAL INTELLIGENCE (AI) CONTROLLER DESIGN FOR BEAM AND BALL SYSTEM

IBRAHIM BIN IDERUS

This thesis is submitted as partial fulfillment of the requirements for the award of the Bachelor of Electrical Engineering (Hons.) (Control and Instrumentasion)

Faculty of Electrical & Electronics Engineering
Universiti Malaysia Pahang

10 NOVEMBER, 2008

"All the trademark and copyright use herein are property of their respective owner. References of information from other sources are quoted accordingly; otherwise the information presented in this report is solely work of the author"

Signature :___________________

Author : IBRAHIM BIN IDERUS

Date : 10 NOVEMBER 2008

‘To my beloved father, mother, brother and sisters.’

ACKNOWLEDGEMENT

In the name of Allah S.W.T. the Most Gracious, the Most Merciful. Praise is to Allah, Lord of the Universe and Peace and Prayers be upon His final Prophet and Messenger, Muhammad S.A.W.

There are several people who contribute in preparing this thesis together with me. There are academicians, parent and students. First of all, I wish to express my sincere gratitude to my supervisor, Mr. Ahmad Nor Kasruddin Bin Nasir for all his advices, guidance, motivation and supports throughout the whole project. As supervisor, he had done great job in ensuring the project reached its objectives.

Also not forgetting my fellow friends who continuously giving the supports and knowledge every time I want it. Without their warmth concern and kindness, it will be difficult for me in the fabrication stage at the beginning of this project. I will never forget the experience and the skills I have gathered from all of you from the very beginning of this project until it is completed. To University Malaysia Pahang (UMP), I am grateful that this institution really wanted to see its students to be skillful and knowledgeable. Even though this project didn't apply any funding I still appreciated because it still manage to produce the desire output

My specially thank to my parents, Mr. Iderus bin Mohamad and Madam Hamidah binti Abd Hamid and my lovely sisters and brothers who had given me moral supports and always pray for my future undertakings. For those I did not mention here, it is not that I forgot, but the space is limited. Thank you for everything.

ABSTRACT

Artificial Intelligence controller design for beam and ball mean we control and balancing the ball always on the top of the beam. Design of a fuzzy controller requires more design decisions than usual, for example regarding rule base, inference engine, defuzzification and data pre- and post processing. The fuzzy logic was applied to this system so that we can make the system in stable condition. This is a difficult control task because the ball does not stay in one place on the beam but moves with an acceleration that is proportional to the tilt of the beam. This is open loop unstable because the ball position increases without limit for a fixed input .That mean we need to regulate the position of the ball on the beam by changing the angle of the beam. The system didn't apply any hardware because it totally software and been designed using Matlab as the medium to gain the desire result. The result gain from this project is very important as the strong basic in adapting to the other systems.

ABSTRAK

Pengawalan kecerdikan buatan digunakan untuk mereka sebuah sistem yang mampu mengawal dan sentiasa mewujudkan keseimbangan bola di atas palang besi pada setiap masa. Pengawalan menggunakan kaedah fuzzy logik dan rangkaian neural di adaptasikan supaya sistem ini berada dalam keadaan yang stabil. Kesukaran yang dihadapi bagi membina sistem ini adalah kerana bola tidak berada pada satu keadaan yang tetap, ia kan bergerak memecut secara kadar langsung dengan kecondongan paksi palang besi. Ini bermaksud sistem ini masih tidak bertindak balas dari input dan masih tidak sempurna kerana pergerakan bola meningkat tanpa limit yang tetap. Untuk itu kita perlu mengawal bola dengan mengawal kecondongan palang besi. Sistem ini tidak menggunakan sebarang alatan atau model tetapi menggunakan sistem perisian sepenuhnya dimana ia direka berdasarkan platform Matlab. Hasil yang diperolehi dari projek ini sangat-sangat penting kerana ia menjadi asas kukuh dalam membina sistem-sistem yang lain

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATION

UMP	- Universiti Malaysia Pahang
FL	- Fuzzy Logic
AI	- Artificial Intelligence
P	- Positive Error
N	- Negative Error
Z	- Zero Error
e(k)	- Error
Δe(k)	- Change of Error
Δu	- Change of Control Signal
$\dot{\theta}$	- Theta Velocity
$\ddot{\theta}$	- Theta Acceleration

LIST OF APPENDICES

CHAPTER 1

INTRODUCTION

1.1 Overview

Nowadays system modeling, analysis and control is very important principles and process in control system. In control system there are a number of generic systems and methods which are encountered in all areas of industry and technology. These report aim to explain these important systems and methods in straightforward terms.

Firstly what is mean by “The Ball and Beam”. The Ball and Beam system is one of the most enduringly popular and important laboratory models for teaching controls system engineering. The ball and beam is widely used because it is very simple to understand as a system and yet the control techniques that can be studied it cover many important classical and modern design methods. The “open loop unstable” is the important property it has.

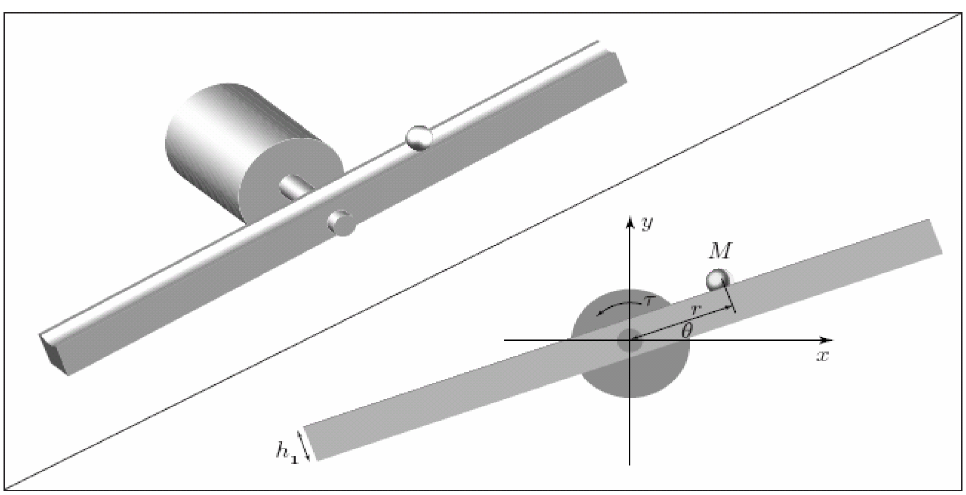


Figure 1.1: Ball and Beam system

The system shown in figure 1 is very simple but need lots of knowledge to control overall system. It consist a steel ball rolling on the top of a long beam. The beam is mounted on the gear and so the beam can titled about its axis by applying an electrical control signal to the gear. The position of the ball can be measured using a special sensor.

The control job is to automatically regulate the position of the ball on the beam by changing the angle of the beam. This is a difficult control task because the ball does not stay in one place on the beam but moves with an acceleration that is proportional to the tilt of the beam. In control technology the system is open loop unstable because the system output (the ball position) increases without limit for a fixed input (beam angle). Feedback control must be used to keep the ball in a desired position on the beam

The ball and beam system is very relevance because most control problems that we meet in practical world are straightforward to control. For fixed input signal the output stays more or less constant. An important set of systems however, either by design or natural, unstable, and feedback control is essential to make them operate safely. Many modern industrial processes and technology systems are intrinsically unstable could be used without stabilizing feedback control. Important practical examples of unstable system are:-

- In the chemical processes industries
- In power generation
- In aerospace

The control of unstable systems is critically important to many of the most difficult control problems and must be studied in the laboratory. The real problem is that real unstable are usually dangerous and cannot be brought into the laboratory. The ball and beam system was developed to resolve this problem. It is simple, safe mechanism and yet it has the important dynamics features of an unstable system

1.2 Problem Statement

Controlling system is very important in nowadays complex system. Which mean the system must be the efficient and user friendly. The real problem here was the movement of the ball on the beam because the ball movement accelerate without any fixed limits. It shown the system was in the open loop unstable condition. My development here was to gain the close loop stable condition using the Artificial Intelligence Controller.

1.3 Project Objectives

The objectives of this project are to;

- ü To design Artificial Intelligence (AI) controller (Fuzzy Logic Controller) for Ball and Beam system
- ü To apply non-linear equation using Lagrian Equation to the beam and ball modeling system
- ü Derive the Mathematical Modeling from the Beam and Ball mechanism

1.4 Scope of Project

The scope of this project is to design the controller for the beam and ball system using fully software application platform (MATLAB). The real system will be converted to mathematical model so that the analysis can be done to control the position of the ball on the right spot. This project will be control by Artificial Intelligence (AI) controller.

i. Modelling the Beam and Ball system (transfer function)
ii. Artificial Intelligence Controller design Fuzzy Logic
(to control the angle of the beam and the ball position)

1.5 Thesis Organization

Chapter 1 discussed briefly about the project in the overview topic. The objectve and scope of the project is also discussed in this chapetr .

Chapter 2 discussed on literature review that related to this project, advantages of fuzzy logic controller, the ball and beam system, and the concept of fuzzy logic.

Chapter 3 discussed about the methodologies of the overall system been designed and also the block diagram of system sequence. This chapter also explains usage of Lagrian Equation to produce the Mathematical Model for the beam and ball mechanism.

Chapter 4 discussed about explanation and step to design the beam and ball modeling using simulink from Matlab®. It also contain the procedure of creating the fuzzy logic controller base from the under damped graph. Beside that it also show the progress of the simulation development.

Chapter 5 will conclude all the discussions on the previous chapter. The recommendation for the future progress also described in this chapter.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction.

This project describes the design of a Artificial Intelligence (AI) controller system for Beam and Ball system, where the student applies a control systems (Fuzzy Logic and Neural Network) approach to real working condition or in easy word was to solve the problem base on the information learned. **The project involves software co-design, so that knowledge from a number of engineering disciplines is necessary for arriving at a workable solution. In the process, the student realizes the advantage of logically analyzing the system requirements according to functional areas, rather than having subjective ideas of the solution at the beginning** [3].

In order to develop this control system, a deep knowledge in the specified field (AI, Lagrangian , Physic etc) must be applied as a thorough analysis of the design alternatives for software modules need to be carried out. **Since the present application involves the use of software components, the designer should be able to translate the real system to suitable equation functional** [2].

Beam and Ball system is used in providing a way of solution for the unstable condition to be apply in the other system such like chemical plant, aerospace and others that need stability system. Commonly the chemical production plant are the harsh environment which there are lots of danger situation will happen if the system became unstable. **Beam and ball system feature lets this condition to stable by making the system close loop** [1].

For the software platform used here is MATLAB, we need to build a block system (Fuzzy logic and Neural network) which can control the ball on the beam without dropping from the beam **The corresponding program is design base from nonlinear equation.** [4].

One major objective of this design project is to develop skills in order to become proficient in all aspects of the development process, from problem definition and identification of requirements, to planning, design, implementation and testing of the proposed solution

2.2 Fuzzy Logic

A fuzzy control system is a control system based on fuzzy logic - a mathematical system that analyzes analog input values in terms of logical variables that take on continuous values between 0 and 1, in contrast to classical or digital logic, which operates on discrete values of either 0 and 1 (true and false).

Fuzzy logic is widely used in machine control. The term itself inspires a certain skepticism, sounding equivalent to "half-baked logic" or "bogus logic", but the "fuzzy" part does not refer to a lack of rigour in the method, rather to the fact that the logic involved can deal with fuzzy concepts - concepts that cannot be expressed as "true" or "false" but rather as "partially true".

Although genetic algorithms and neural networks can perform just as well as fuzzy logic in many cases (in fact, certain neural networks can be shown to be mathematically equivalent to certain fuzzy logic systems), fuzzy logic has the advantage that the solution to the problem can be cast in terms that human operators can understand, so that their experience can be used in the design of the controller. This makes it easier to mechanize tasks that are already successfully performed by humans.

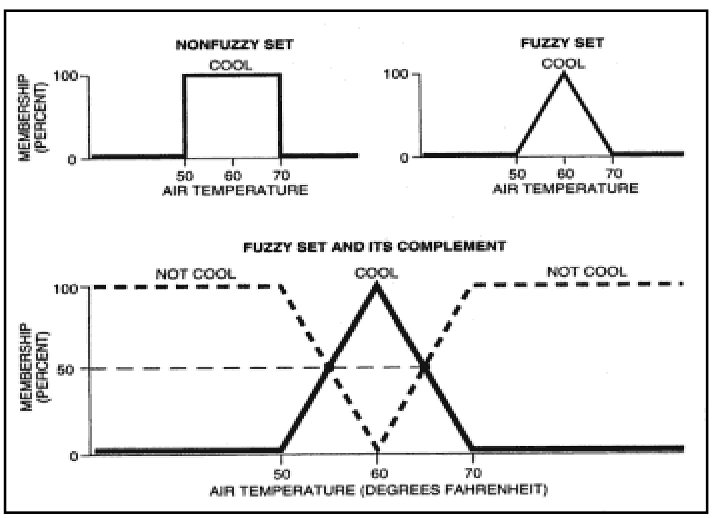


Figure 2.1 : “Centroid” method for Fuzzy Logic

2.2.1 Advantages of Fuzzy Logic Controller

Fuzzy Logic offers several unique features that make it a particularly good choice for many control problems:-

- It is inherently robust since it does not require precise, noise-free inputs and can be programmed to fail safely if a feedback sensor quits or is destroyed. The output control is a smooth control function despite a wide range of input variations.
- Since the Fuzzy Logic controller processes user-defined rules governing the target control system, it can be modified and tweaked easily to improve or drastically alter system performance. New sensors can easily be incorporated into the system simply by generating appropriate governing rules.
- Fuzzy Logic is not limited to a few feedback inputs and one or two control outputs, nor is it necessary to measure or compute rate-of-change parameters in order for it to be implemented. Any sensor data that provides some indication of a system's actions and reactions is sufficient. This allows the sensors to be inexpensive and imprecise thus keeping the overall system cost and complexity low.
- Because of the rule-based operation, any reasonable number of inputs can be processed (1-8 or more) and numerous outputs (1-4 or more) generated, although defining the rule base quickly becomes complex if too many inputs and outputs are chosen for a single implementation since rules defining their interrelations must also be defined. It would be better to break the control system into smaller chunks and use several smaller Fuzzy Logic controllers distributed on the system, each with more limited responsibilities.
- Fuzzy Logic can control nonlinear systems that would be difficult or impossible to model mathematically. This opens doors for control systems that would normally be deemed unfeasible for automation.

2.3 Conclusion

As conclusion, this chapter defines the literature review for this project. The literature reviews about the usage of Artificial Intelligence Controller (fuzzy logic) for the beam and ball system that it can stabilize this system using the more effective controller.

CHAPTER 3

METHODOLODY

3.1 Introduction

Methodologies is one of the most important element to be consider to make sure that the development of the project is smooth and get the expected result. The methodology for this project is divided into five phase:

Phase I : Literature review

i. Surfing internet
ii. Books and writing important point
iii. Discussion with supervisor and lecturers

Phase II : Mathematical Modeling for the Beam and Ball system
- based on internet sources and discussion with supervisor.

Phase III : Design and develop Fuzzy Logic controller.

i. Learn how design the system in the Matlab as selected program to develop overall system
ii. Design the Fuzzy Logic controller using Matlab
iii. Starts develop Fuzzy Logic controller

Phase IV : Integrate both systems and setting the best point to archive the goal or not.

Phase V : Result Analysis

i. Data collection
ii. Data analysis

3.2 The Flow of The System

To design a full Fuzzy Logic controller it needs the right flow to determine the perfect system. The flow showed in the Figure 3.2 describes how we analyze from the mechanical beam and ball system to the mathematical modeling.

Mathematical model is an abstract model that uses mathematical language to describe the behavior of a beam and ball system. It serves in finding an optimal solution to a planning problem and to establish understanding of the relationships among the input data (velocity and acceleration of theta/ position) within a model. For this system we derive the mathematical model into nonlinear equation that fulfills the fuzzy need.

The next step is to design the system using appropriate function block that provided in the Simulink of Matlab. The system must be design in the closed control system because it utilizes feedback. The feedback is used to make the decision about changes to the control signal that drives the plant. In this project it operates at a fix frequency. The frequency of changes to drive signal is usually the same as the position and angle. After reading from error, the fuzzy logic controller recalculating and adjusting the drive signal using the rules provided.

In this system the Artificial Intelligence (AI) controller have been used. A fuzzy control system is a control system based on fuzzy logic - a mathematical system that analyzes analog input values in terms of logical variables that take on continuous values between 0 and 1, in contrast to classical or digital logic, which operates on discrete values of either 0 and 1 (true and false).

.

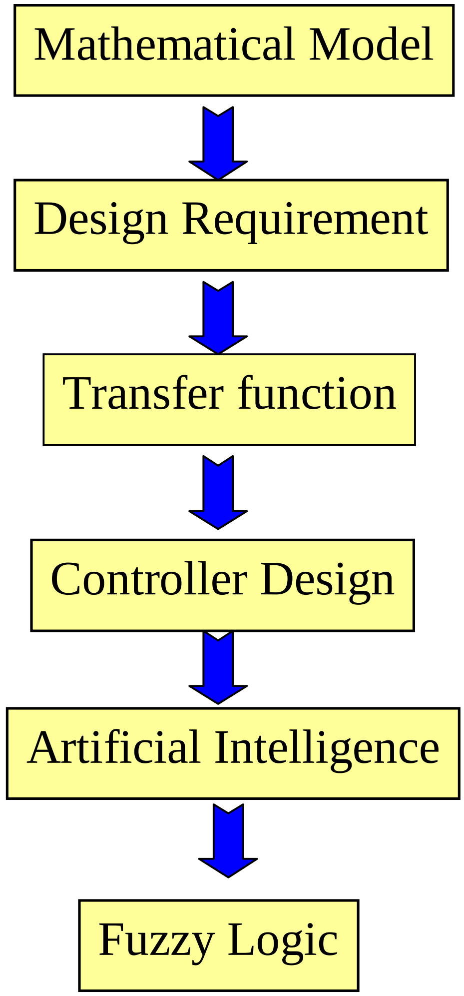


Figure 3.2 : Block diagram of the system

3.3 System Design

3.3.1 Mathematical Modeling Development

The constants are defined as follows:

M = mass of the ball

r = radius of the ball

J = ball's moment of inertia

G = gravitational acceleration

Theta = beam angle coordinate

3.3.2 Lagrange equation

Assume zero rotational inertia

$$J_1 = Mr^2 \tag{1}$$

Kinetic Energy

$$K_2 = \frac{1}{2}J\dot{\theta}^2 + \frac{1}{2}M\dot{x}^2 = \frac{M}{2}\left(r^2\dot{\theta}^2 + \dot{x}^2\right) \tag{2}$$

Potential Energy

$$V = Mgy = Mgr\sin(\theta) \tag{3}$$

Lagrange

$$L = K_e - V = \frac{M}{2}\left(r^2\dot{\theta}^2 + \dot{x}^2\right) - Mgr\sin(\theta) \tag{4}$$

Assume the system has viscous rotational fraction, B , then we have the dissipation power function

$$P = \frac{1}{2} B \dot{q}^2 \tag{5}$$

General coordinates

$$q = \begin{bmatrix} q \\ r \end{bmatrix} \tag{6}$$

Apply torque

$$Q = \begin{bmatrix} t \\ 0 \end{bmatrix} \tag{7}$$

Langrange equation

$$Q = \frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}}\right) - \frac{\partial L}{\partial q} + \frac{\partial P}{\partial \dot{q}}$$

$$= \frac{d}{dt}\begin{bmatrix} Mr^2 \dot{q} \\ M\dot{r} \end{bmatrix} - \begin{bmatrix} -Mgr\cos(q) \\ Mr\dot{q}^2 - Mg\sin(q) \end{bmatrix} + \begin{bmatrix} B\dot{q} \\ 0 \end{bmatrix}$$

$$= \begin{bmatrix} 2Mr\dot{r}\dot{q} + Mr^2\ddot{q} \\ M\ddot{r} \end{bmatrix} - \begin{bmatrix} -Mgr\cos(q) \\ Mr\dot{q}^2 - Mg\sin(q) \end{bmatrix} + \begin{bmatrix} B\dot{q} \\ 0 \end{bmatrix} \tag{8}$$

$$= \begin{bmatrix} 2Mr\dot{r}\dot{q} + Mr^2\ddot{q} \\ M\ddot{r} \end{bmatrix} - \begin{bmatrix} -Mgr\cos(q) \\ Mr\dot{q}^2 - Mg\sin(q) \end{bmatrix} + \begin{bmatrix} B\dot{q} \\ 0 \end{bmatrix}$$

Or

$$\begin{bmatrix} Mr^2 & 0 \\ 0 & M \end{bmatrix}\begin{bmatrix} \ddot{q} \\ \ddot{r} \end{bmatrix} + \begin{bmatrix} B & 2Mr\dot{q} \\ -Mr\dot{q} & 0 \end{bmatrix}\begin{bmatrix} \dot{q} \\ \dot{r} \end{bmatrix} + \begin{bmatrix} Mgr\cos(q) \\ Mg\sin(q) \end{bmatrix} = \begin{bmatrix} t \\ 0 \end{bmatrix} \tag{9}$$

Note that

$$\begin{bmatrix} \ddot{q} \\ \ddot{r} \end{bmatrix} = -\begin{bmatrix} \frac{1}{Mr^{2}}B & \frac{2}{r}\dot{q} \\ -r\dot{q} & 0 \end{bmatrix}\begin{bmatrix} \dot{q} \\ \dot{r} \end{bmatrix} - \begin{bmatrix} \frac{1}{r}g\cos q \\ g\sin q \end{bmatrix} + \begin{bmatrix} \frac{1}{Mr^{2}} \\ 0 \end{bmatrix} t \qquad (10)$$

The nonlinear states are

$$q\ ,\dot{q}\,,\ r\ ,\ \dot{r} \qquad (11)$$

The state equations

$$\frac{d}{dt}q = \dot{q} = f_{1} \qquad (12)$$

$$\frac{d}{dt}\dot{q} = \ddot{q} = -\left(\frac{B}{Mr^{2}} + \frac{2}{r}\dot{r}\right)\dot{q} - \frac{1}{r}g\cos q + \left(\frac{1}{Mr^{2}}\right)t = f_{2} \qquad (13)$$

$$\frac{d}{dt}r = \dot{r} = f_{3} \qquad (14)$$

$$\frac{d}{dt}\dot{r} = r\dot{q}^{2} - g\sin q = f_{4} \qquad (15)$$

The equilibrium is

$$q_{e} = 0,\ \dot{q}_{e} = 0,\ r_{e} = 1,\ \dot{r}_{e} = 0 \qquad (16)$$

Thus we define the states

$$x_{1} = q - q_{e} = q \qquad x_{2} = \dot{q} - \dot{q}_{e} = \dot{q} \qquad (17)$$

$$x_{3} = r - r_{e} = r - 1 \qquad x_{4} = \dot{r} - \dot{r}_{e} = \dot{r} \qquad (18)$$

The control output

$$u = t - t_e \tag{19}$$

Where

$$-\left(\frac{B}{Mr_e^2} + \frac{2}{r_e}\dot{q}_e \dot{r}_e\right)\dot{q}_e - \frac{1}{r_e} g \cos q_e + \left(\frac{1}{Mr_e^2}\right) t_e = 0 \tag{20}$$

Which implies that

$$t_e = Mgr_e \tag{21}$$

3.3.3 Nonlinear Equations

The new nonlinear equations

$$\dot{x}_1 = x_2 = f_1 \tag{22}$$

$$\dot{x}_2 = -\left(\frac{B}{M(x_3 + r_e)^2} + \frac{2}{(x_3 + r_e)} x_2 x_4\right) x_2 - \frac{1}{(x_3 + r_e)} g \cos(x_1) + \left(\frac{1}{M(x_3 + r_e)^2}\right)(u + t_e) \tag{23}$$

$$\dot{x}_3 = x_4 = f_3 \tag{24}$$

$$\dot{x}_4 = (x_3 + r_e) x_2^2 - g \sin(x_1) = f_4 \tag{25}$$

The equilibrium

$$x_1 = x_2 = x_3 = x_4 = u = 0 \tag{26}$$

Thus

$$a_{11} = \frac{\partial f_1}{\partial x_1}\bigg|_e = 0\big|_e = 0 \qquad a_{12} = \frac{\partial f_1}{\partial x_2}\bigg|_e = 1\big|_e = 1$$

$$a_{13} = \frac{\partial f_1}{\partial x_3}\bigg|_e = 0\big|_e = 0 \qquad a_{14} = \frac{\partial f_1}{\partial x_4}\bigg|_e = 0\big|_e = 0$$

$$a_{21} = \frac{\partial f_2}{\partial x_1}\bigg|_e = \frac{-g}{(x_3 + r_e)}\sin(x_1)\big|_e = 0$$

$$a_{22} = \frac{\partial f_2}{\partial x_2}\bigg|_e = \left(\frac{B}{M\,(x_3 + r_e)^2} + \frac{4}{(x_3 + r_e)} r_2 x_4\right)\bigg|_e = \frac{B}{M}$$

$$a_{23} = \frac{\partial f_2}{\partial x_3}\bigg|_e = \frac{2Bx_2}{M(x_3 + r_e)^3} + \frac{2x_2^2 x_4}{(x_3 + r_e)^2} + \frac{g\cos(x_1)}{(x_3 + r_e)^2} - \frac{2(u + t)}{M(x_3 + r_e)^3}\bigg|_e = -g$$

$$a_{14} = \frac{\partial f_2}{\partial x_4}\bigg|_e = \frac{2x_2^2}{(x_3 + r_e)}\bigg|_e = 0$$

$$a_{31} = \frac{\partial f_3}{\partial x_1}\bigg|_e = 0\big|_e = 0, \qquad a_{32} = \frac{\partial f_3}{\partial x_2}\bigg|_e = 0\big|_e = 0$$

$$a_{33} = \frac{\partial f_1}{\partial x_3}\bigg|_e = 0\big|_e = 0, \qquad a_{34} = \frac{\partial f_3}{\partial x_4}\bigg|_e = 1\big|_e = 1$$

$$a_{41} = \frac{\partial f_4}{\partial x_1}\bigg|_e = -g\cos(x_1)\big|_e = -g \qquad a_{42} = \frac{\partial f_4}{\partial x_2}\bigg|_e = 0\big|_e = 0$$

$$a_{43} = \left.\frac{\partial f_4}{\partial x_3}\right|_e = \left.x_2^2\right|_e = 0 \qquad a_{44} = \left.\frac{\partial f_4}{\partial x_4}\right|_e = \left.0\right|_e = 0$$

$$b_1 = \left.\frac{\partial f_1}{\partial u}\right| = \left.0\right|_e = 0 \qquad b_2 = \left.\frac{\partial f_2}{\partial u}\right|_e = \left.\frac{1}{M(x_3 + r_e)^2}\right|_e = \frac{1}{M}$$

$$b_3 = \left.\frac{\partial f_3}{\partial u}\right| = \left.0\right|_e = 0 \qquad b_4 = \left.\frac{\partial f_4}{\partial u}\right| = \left.0\right|_e = 0$$

The linearized state model

$$\dot{x} = Ax + Bu \tag{27}$$

$$y = Cx \tag{28}$$

Where

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & \frac{B}{M} & -g & 0 \\ 0 & 0 & 0 & 1 \\ -g & 0 & 0 & 0 \end{bmatrix} \qquad B = \begin{bmatrix} 0 \\ \frac{1}{M} \\ 0 \\ 0 \end{bmatrix} \tag{29}$$

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix} \tag{30}$$

CHAPTER 4

FUZZY LOGIC CONTROLLER DESIGN

4.1 Building the Model in Simulink

In this report, rather than express all the forces and geometric constraints (which is difficult to model in Simulink for dynamic systems with constraints) we will model the nonlinear Lagrangian equation of motion directly. This equation gives d/dt(ANG) as a function of the state and input variab, ANG, d/dt(ANG), POS, and d/dt(POS). We will make use of the Nonlinear Function Block to express this function. First, we must express the derivatives of the outputs ANG, ANGDOT, POS and POSDOT.

- Open a new model window in Simulink.
- Insert an Integrator block from the Linear block library.
- Insert a second Integrator to the right of the first, and connect the two with a line.
- Draw a line from the second Integrator .
- Insert an Out block from the Connections block library and connect it to the "ANG" signal line. Also insert another “Out” and label it with “ANGDOT”. Connect if from d/dt(ANG). This will form the output of the system.
- Change the label of the Out block to "ANG" by single-clicking on the existing "Out" label.

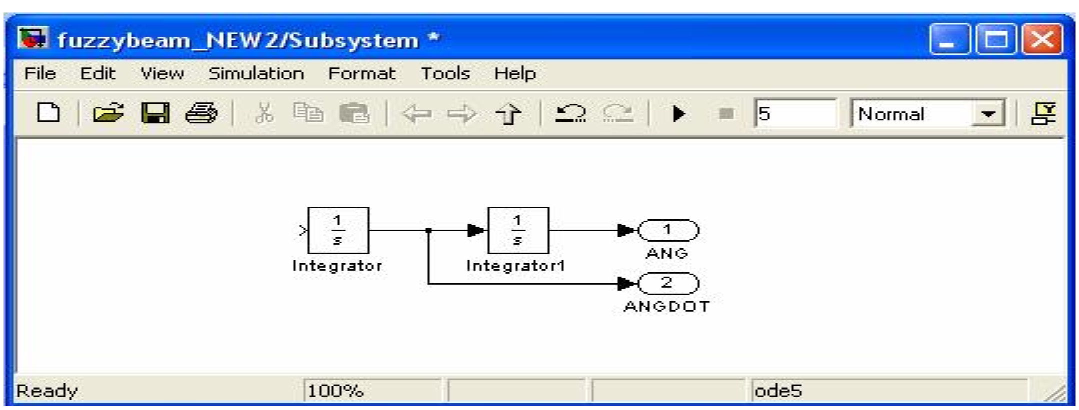


Figure 4.1: Connection of Integrator with Output

Now, we will insert the function which takes the vector [ANG d/dt(ANG) and returns d/dt(ANG).

- Insert a Fcn block from the Nonlinear library and connect its output to the input of the first Integrator.
- Edit the Fcn block by double clicking it, and change it's function to the following:

(B/M)*u(2)-g*u(3)+u(1)/M

This function block takes an input vector, u, where each component is referred to as u[1], u[2], etc. In our case, u[1]=ANG, u[2]=d/dt(ANGDOT),

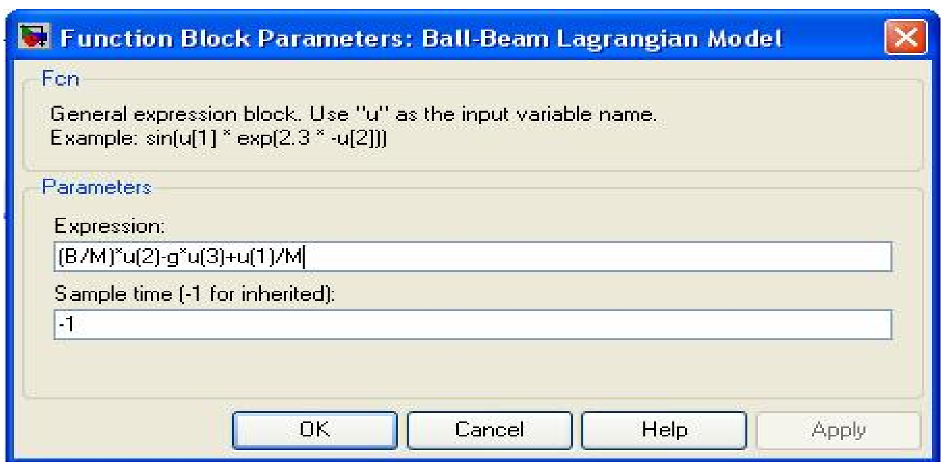


Figure 4.2: Equation 1 Inside Transfer Function

- Close the dialog box and change the label of the Fcn block to "Ball-Beam Lagrangian Model" (you can add newlines in the label by hitting return).

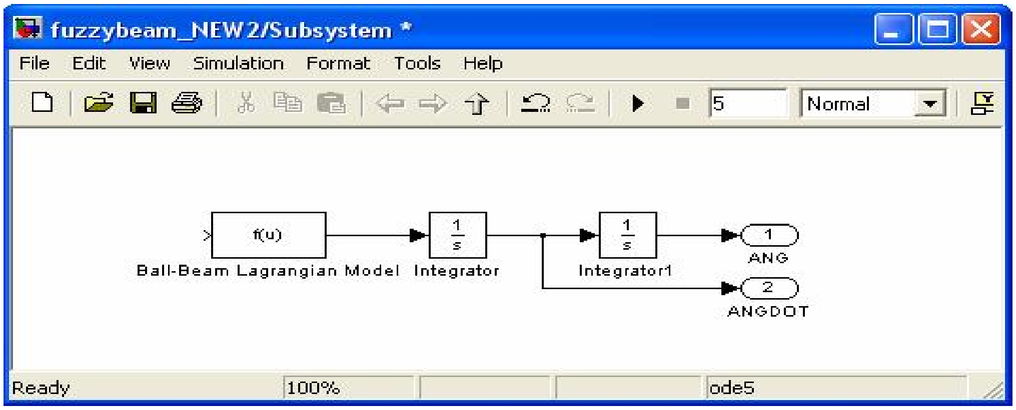


Figure 4.3: Connection of Transfer Function with Integrator

Now, we will begin to construct the function input vector u by feeding back the state signals from the integrators and forming a vector from them with a Mux block.

- Insert a Mux block from the Connections block library and connect its output to the input of the Ball-Beam block.
- Edit the Mux block (by double-clicking on it) and change its number of inputs to 3. The Mux block should now have four inputs.
- Tap a line off the d/dt(ANG) signal (hold Ctrl while drawing) and connect it to the second input of the Mux block.
- Tap a line of the r signal and connect it to the first input of the Mux block.

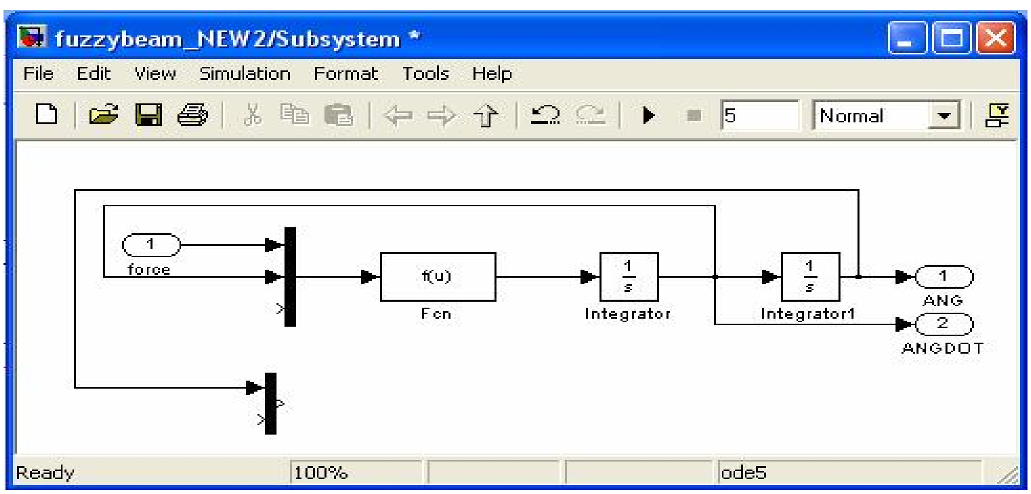


Figure 4.4: Simulation Diagram for Angle

Now we will construct the signals force and d/dt(force) from the input force.

- Insert an In block on the left side of your model window. Change its label to "force".
- Connect the input of the force block to the third input of the Mux block

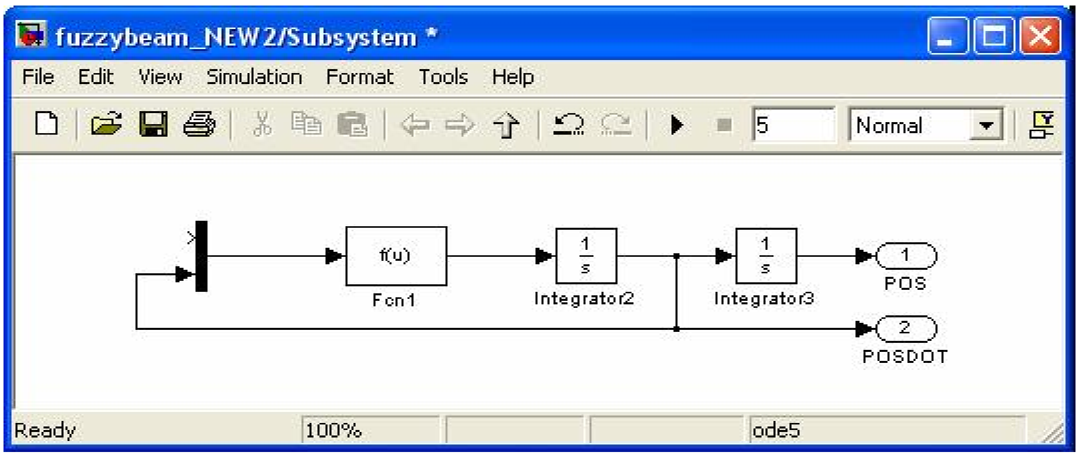


Figure 4.5: Simulation Diagram for Position

Now, we will insert the function which takes the vector [POS d/dt(POS) and returns d/dt(POS).

- Insert a Fcn block from the Nonlinear library and connect its output to the input of the first Integrator.
- Edit the Fcn block by double clicking it, and change it's function to the following:

-g*u(1)+u(2)

This function block takes an input vector, u, where each component is referred to as u[1], u[2], etc. In our case, u[1]=POS, u[2]=d/dt(POSDOT),

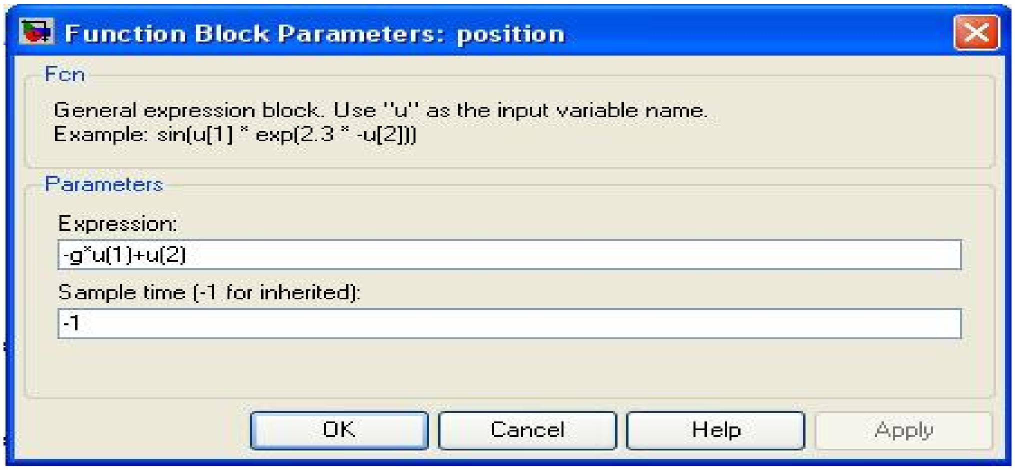


Figure 4.6: Equation II Inside Transfer Function

- Close the dialog box and change the label of the Fcn block to "Position"

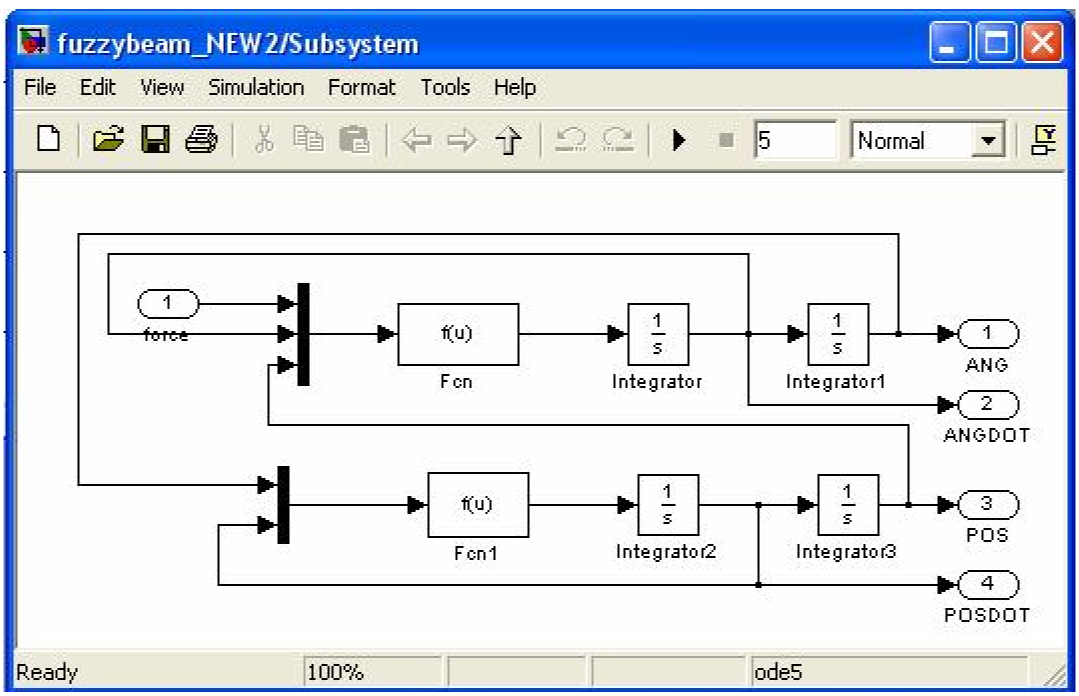


Figure 4.7: Beam and Ball Transfer Function

Save your model as "ball.mdl". Open Loop Response To generate the open-loop response, it is first necessary to contain this model in a subsystem block.

- Create a new model window (select New from the File menu in Simulink or hit Ctrl-N).
- Insert a Subsystem block from the Connections block library.
- Open the Subsystem block by double clicking on it. You will see a new model window labeled "Subsystem".
- Open your previous model window named ball.mdl. Select all of the model components by selecting Select All from the Edit menu (or hit Ctrl-A).
- Copy the model into the paste buffer by selecting Copy from the Edit menu (or hit Ctrl-C).
- Paste the model into the Subsystem window by selecting Paste from the Edit menu (or hit Ctrl-V) in the Subsystem window
- Close the Subsystem window. You will see the Subsystem block in the untitled window with one input terminal labeled force and four outputs terminal labeled ANG, ANGDOT, POS and POSDOT.
- Resize the Subsystem block to make the labels visible by selecting it and dragging one of the corners.
- Label the Subsystem block "Ball and Beam Model".

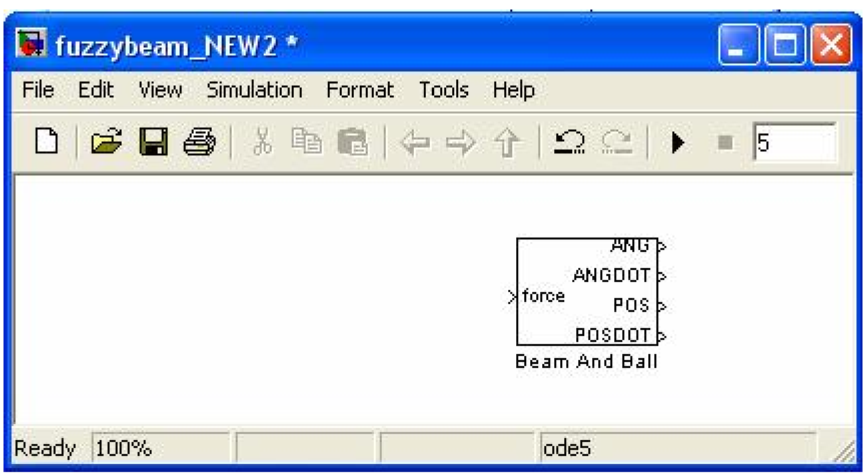


Figure 4.8: Beam and Ball Subsystem

- Insert a Step block (from the Sources block library) and connect it to the input of the Ball and Beam Model.
- Edit the Step block (by double clicking on it to bring up the dialog box) and change the Step Time value to 0. Close the Step block dialog box.
- Insert a Scope block (from the Sinks block library) and connect it to the output of the Ball and Beam Model.

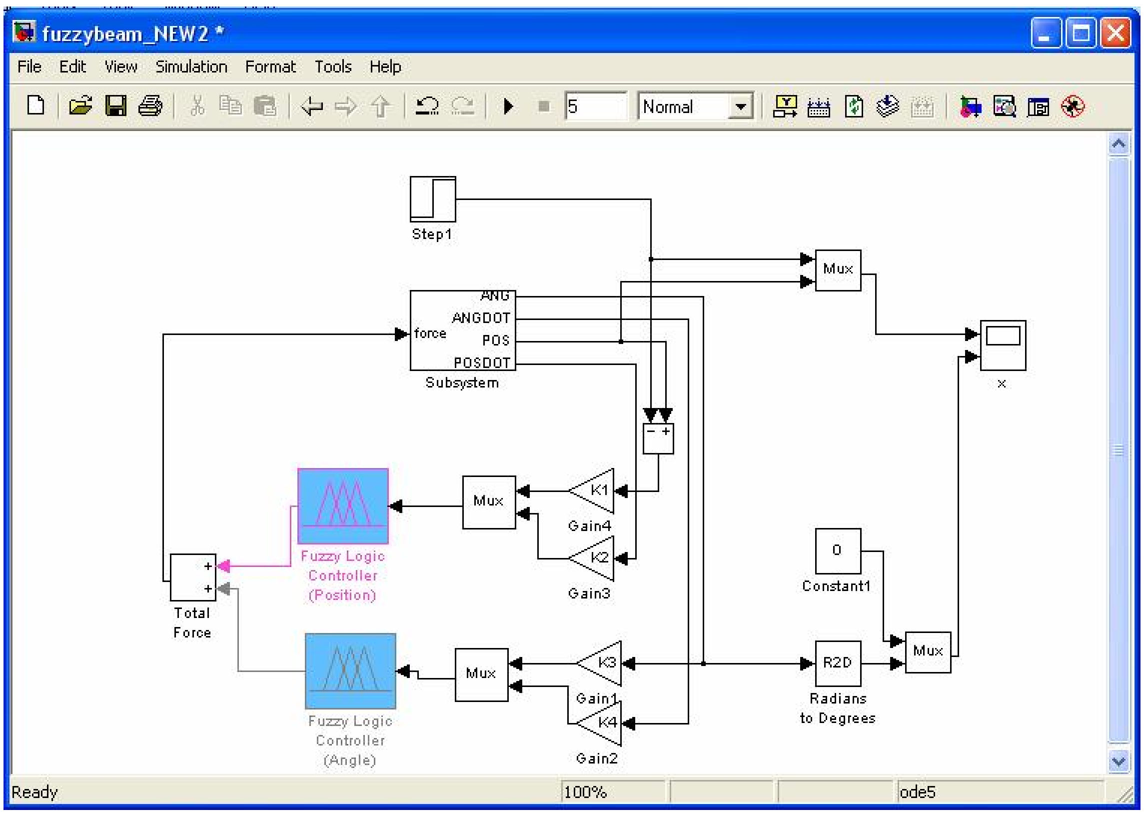
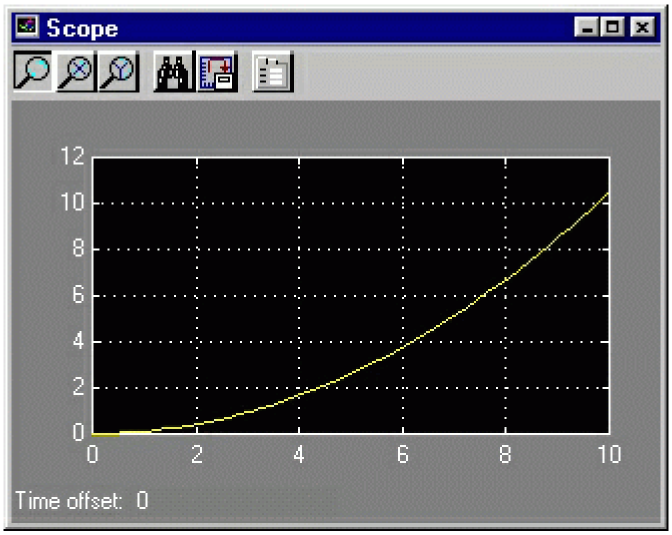


Figure 4.9: Overall Simulation System

Before obtaining a step response, we must set the physical parameters Enter the following commands at the MATLAB prompt.

```
M = 2;
g = 9.81;
B = 0.1;
```

We are now ready to run the simulationStart the simulation by selecting Start from the Simulation menu (or hit Ctrl-t). When the simulation is finished, open the Scope by double clicking on it and hit the Scope's autoscale button. You will see the following response.



From this plot it is clear that the system is unstable in open-loop causing the ball to roll right off the end of the beam. Therefore, some method of controlling the ball's position in this system is required. Later in this report, we will implement fuzzy logic controller.

4.2 Fuzzy Control System Design

Basically, the difficult task of modeling and simulating complex real-world systems for control systems development, especially when implementation issues are considered, is well documented. Even if a relatively accurate model of a dynamic system can be developed, it is often too complex to use in controller development, especially for many conventional control design procedures that require restrictive assumptions for the plant (e.g., linearity). It is for this reason that in practice conventional controllers are often developed via simple models of the plant behavior that satisfy the necessary assumptions, and via the ad hoc tuning of relatively simple linear or nonlinear controllers.

Regardless, it is well understood (although sometimes forgotten) that heuristics enter the conventional control design process as long as you are concerned with the actual implementation of the control system. It must be acknowledged, moreover, that conventional control engineering approaches that use appropriate heuristics to tune the design have been relatively successful. You may ask the following questions: How much of the success can be attributed to the use of the mathematical model and conventional control design approach, and how much should be attributed to the clever heuristic tuning that the control engineer uses upon implementation? And if we exploit the use of heuristic information throughout the entire design process, can we obtain higher performance control systems?

Fuzzy control provides a formal methodology for representing, manipulating, and implementing a human's heuristic knowledge about how to control a system. In this section we seek to provide a philosophy of how to approach the design of fuzzy controllers. This will lead us to provide a motivation for, and overview of, the entire book.

The fuzzy controller block diagram is given in Figure 1.2, where we show a fuzzy controller embedded in a closed-loop control system. The plant outputs are 1.3 Fuzzy Control System Design 11 denoted by $y(t)$, its inputs are denoted by $u(t)$, and the reference input to the fuzzy controller is denoted by $r(t)$.

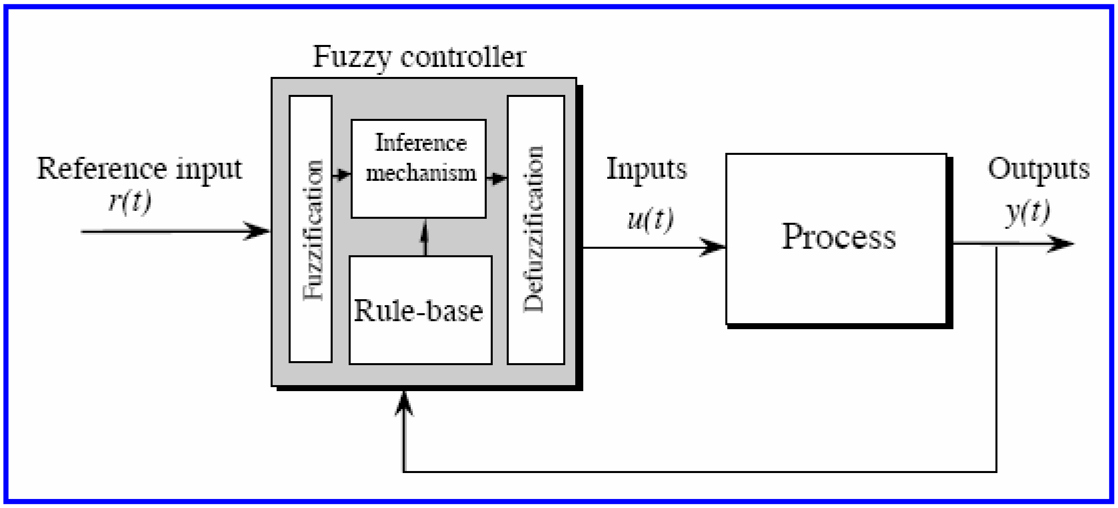


Figure 4.2.1: Fuzzy controller architecture.

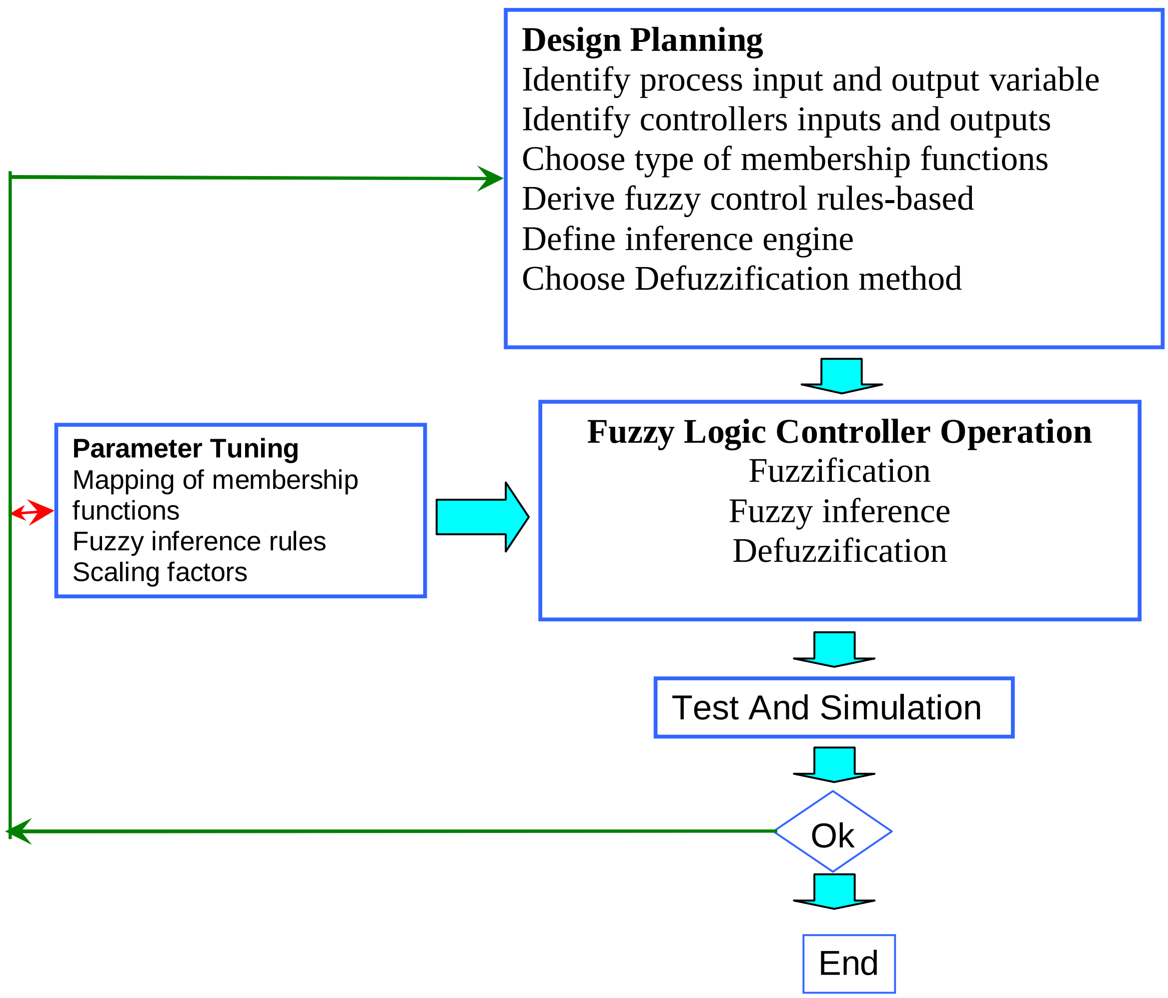
The fuzzy controller has four main components: (1) The **"rule-base"** holds the knowledge, in the form of a set of rules, of how best to control the system. (2) The **inference mechanism** evaluates which control rules are relevant at the current time and then decides what the input to the plant should be. (3) The **fuzzification interface** simply modifies the inputs so that they can be interpreted and compared to the rules in the rule-base. And (4) the **defuzzification** interface converts the conclusions reached by the inference mechanism into the inputs to the plant.

Basically, you should view the fuzzy controller as an artificial decision maker that operates in a closed-loop system in real time. It gathers plant output data $y(t)$, compares it to the reference input $r(t)$, and then decides what the plant input $u(t)$ should be to ensure that the performance objectives will be met. To design the fuzzy controller, the control engineer must gather information on how the artificial decision maker should act in the closed-loop system.

Sometimes this information can come from a human decision maker who performs the control task, while at other times the control engineer can come to understand the plant dynamics and write down a set of rules about how to control the system without outside help. These "rules" basically say, "If the plant output and reference input are behaving in a certain manner, then the plant input should be some value."

A whole set of such **"If-Then"** rules is loaded into the rule-base, and an inference strategy is chosen, then the system is ready to be tested to see if the closed-loop specifications are met. This brief description provides a very high-level overview of how to design a fuzzy control system. Below we will expand on these basic ideas and provide more details on this procedure and its relationship to the conventional control design procedure.

4.3 Flow Chart of Fuzzy Logic Controller



4.4 Derivation of fuzzy rules

- The fuzzy rules can be formulated based on the conditions of error, E, and the rate of change of error, ΔE, and the output is the change in the control signal, Δu
- This condition is transform into fuzzy rules using the following formula

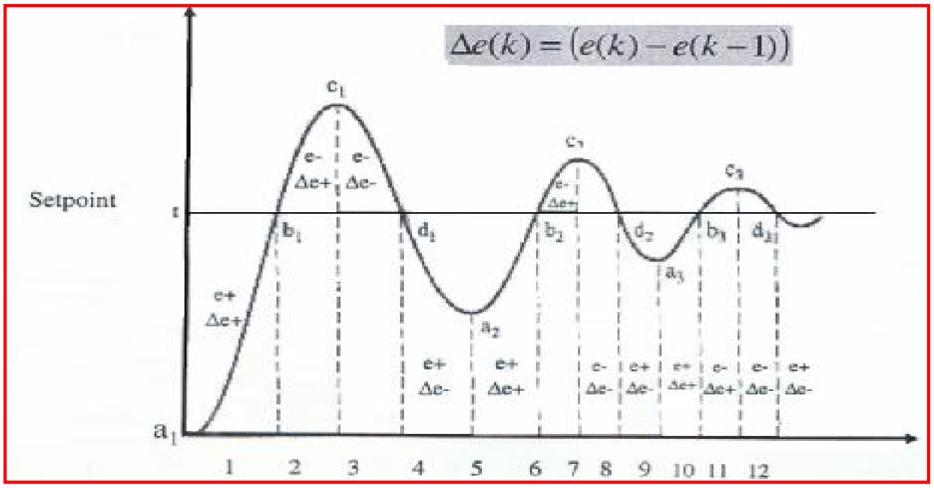


Figure 4.4.1: Observation of system response for deriving fuzzy control rules

$$e(k) = r(k) - y(k)$$

$$\Delta e(k) = e(k) - e(k-1)$$

- The prototype of fuzzy control rules is tabulated in Table 1 and a justification of fuzzy control rules is added in table 2. (These are for only 3 quantization P, Z and N)
- As an example, based on the under damped 2^{nd} order system response, the corresponding rule of Region 2 can be formulated as rule R1 below which decreases the system overshoot and for Region 3, rule R2 has the effect of shortening the rise time

- More specially,

R1: if (E is negative and ΔE is positive)
Then ΔU is positive

R2: if (E is negative and ΔE is negative)
Then ΔU is positive

- Prototype of fuzzy control rules with term set derive such below

Rule No.	E	AE	Au	Reference Point
1	P	Z	P	a1,a2,a3
2	Z	N	N	d1,d2,d3
3	N	Z	N	c1,c2,c3
4	Z	P	P	b1,b2,b3
5	Z	Z	Z	set point
6	P	N	P	1 (rise time), 5
7	N	N	N	2 (overshoot), 6
8	N	P	N	3,7
9	P	P	P	4,8
10	P	N	Z	9
11	N	P	Z	10

Table 4.1: Fuzzy Rules

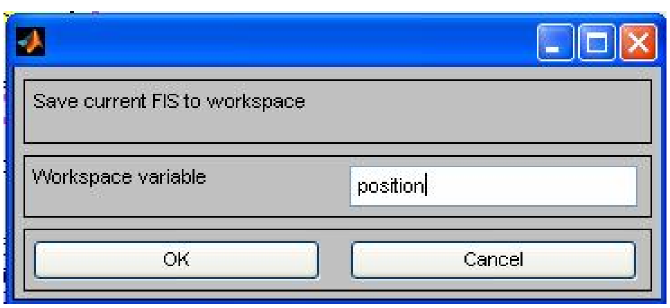
4.5 Building the Fuzzy Controller

To start this system from scratch, type

fuzzy

at the MATLAB prompt. The generic untitled FIS Editor opens, with one input, labeled **input1**, and one output, labeled **output1**. For this example, we will construct a two-input, one output system, so go to the **Edit** menu and select **Add input**. A second yellow box labeled **input2** will appear. The two inputs we will have in position are **position** and **delposition**. Our one output is **force**. We'd like to change the variable names to reflect that:

i. Click one on the box (yellow) on the left marked **input1** (the box will be highlighted in red).
ii. In the white edit field on the right, charge input1 to **position** and press **Return.**
iii. Click once on the box (yellow) marked **input2** (the box will be highlighted in red).
iv. In the white field on the right, change input2 to **delposition** and press **Return**
v. Click once on the box (blue) on the ringht marked **output1**
vi. In the white edit field on the right, change output to **force.**
vii. From the File menu, select **Export and then To Workspace.**



viii. Enter the variable name **position** and click OK
ix. Repeat the same step for **angle** part

You will see the diagram updated to reflect the new names of the input and output variables. There is now a new variable in the workspace called **position** that contains all the information about this system. By saving to the workspace with a new name, you also rename the entire system. Your window will look something like this.

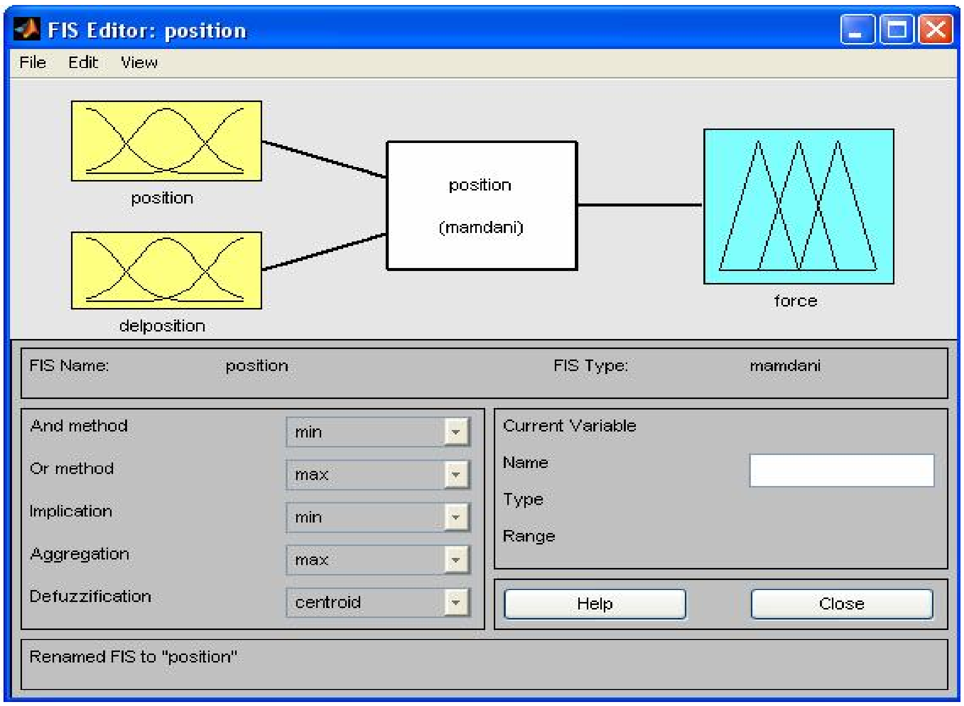


Figure 4.5.1 : FIS Editor for Position

Leave the inference options in the lower left in their default positions for now. You've entered all the information you need for this particular GUI. Next define the **membership functions** associated with each of the variables. To do this, open the Membership Function Editor. You can open the Membership Function Editor within the FIS Editor window, select **Edit > Membership Functions**.

4.5.1 Membership Function

The membership function is a graphical representation of the magnitude of participation of each input. It associates a weighting with each of the inputs that are processed, define functional overlap between inputs, and ultimately determines an output response. The rules use the input membership values as weighting factors to determine their influence on the fuzzy output sets of the final output conclusion. Once the functions are inferred, scaled, and combined, they are defuzzified into a crisp output which drives the system. There are different membership functions associated with each input and output response. Some features to note are:

SHAPE - triangular is common, but bell, trapezoidal, haversine and, exponential have been used. More complex functions are possible but require greater computing overhead to implement.. HEIGHT or magnitude (usually normalized to 1) WIDTH (of the base of function), SHOULDERING (locks height at maximum if an outer function. Shouldered functions evaluate as 1.0 past their center) CENTER points (center of the member function shape) **OVERLAP** (N&Z, Z&P, typically about **50% of width** but can be less).

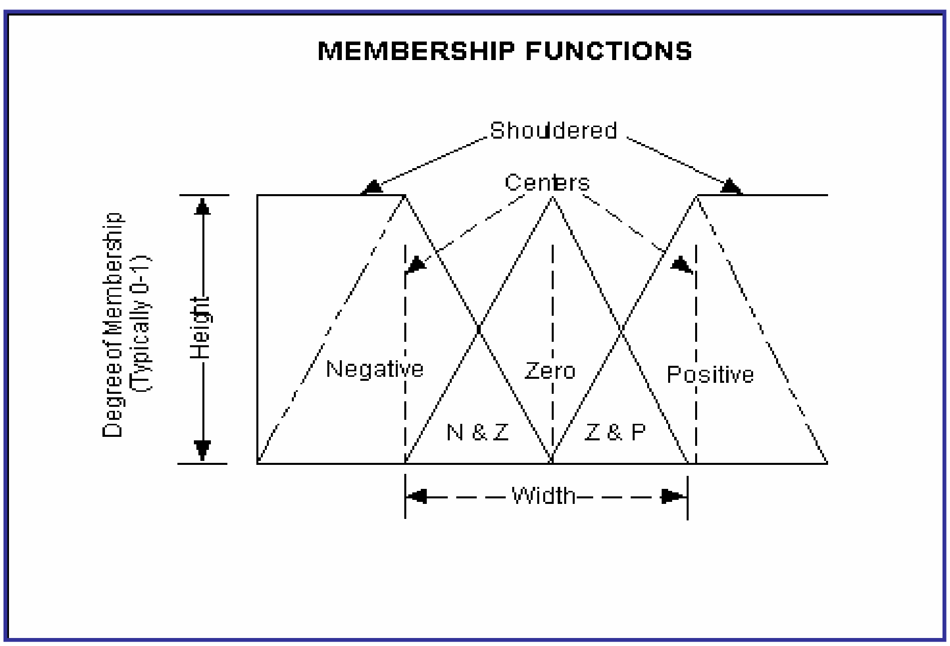


Figure 4.5.1(a): The features of a membership function

The degree of membership (DOM) is determined by plugging the selected input parameter (error or error-dot) into the horizontal axis and projecting vertically to the upper boundary of the membership function(s).

The process of specifying the input membership functions for this two input **position** problem is as follows:

i. Select the input variable, **position,** by double-clicking on it. Set both the **Range** and **Display Range** to the vector [-1,1].

ii. Select MFs.. from the Edit menu. The window below opens.

iii. Use the tab to choose gaussmf for **MF Type** and 3 for **Number of MFs.** This adds three Gaussian curves to the variable service.

iv. Click once on the curve with leftmost hump. Change the name of the curve to **N**. Adjust the shape of the membership function in desired parameter change, and then click in the membership function. The parameter for this curve is [-1 -0.5 0].

v. Name the curve with the middle hump, **Z**, and the curve with the rightmost hump, **P**. The associated parameter Z is [-0.5 0 0.5] and **P** is [0 0.5 1].

vi. Select the input variable, **delposition**, by clicking on it. Set both range and the display range to the vector [-1 1].

vii. Finally to create the membership functions for the output variable, **force**. The Variable Palette is use to create output variable membership function. On the left, selecting the output variable, **force**. The input ranged from -1 to 1 but the output scale is going to be a force **50 percent overlapping**.

The system should look something like this.

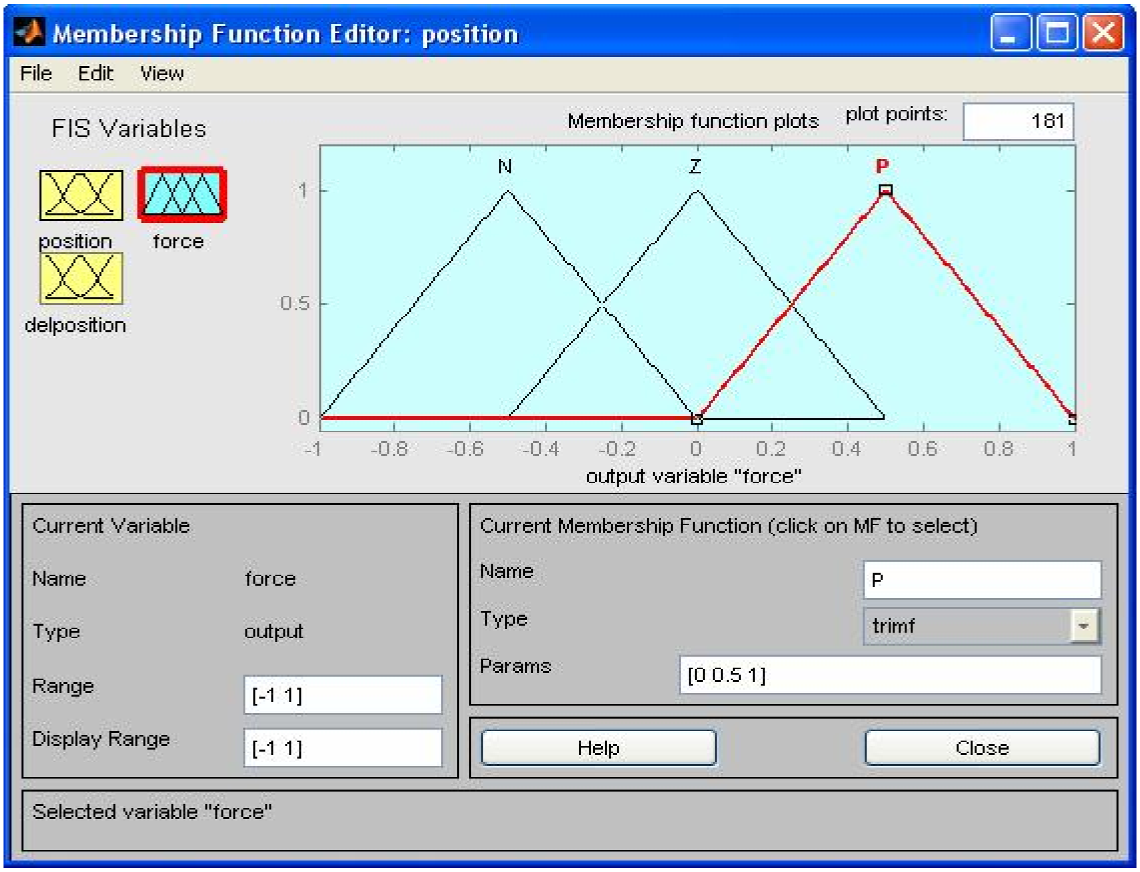


Figure 4.5.1(b): Membership Function for Position

All previous step have been same aplly to the **angle** part. The variable have been named, and the membership functions have appropriate shapes and name, next step is to write down the rules.

4.5.2 Rules Design

THE RULE MATRIX

In the previous report, the concept of linguistic variables was presented. The fuzzy parameters of **error, e(k)** and **error-dot, ∆e(k)** were modified by the adjectives "negative" is **N**, "zero" is **Z**, and "positive" is **P**. To picture this, imagine the simplest practical implementation, a 3-by-3 matrix.

The columns represent "negative error", "zero error", and "positive error" inputs from left to right. The rows represent "negative", "zero", and "positive" "error-dot" input from top to bottom. This planar construct is called a rule matrix. It has two input conditions, "error" and "error-dot", and one output response conclusion (at the intersection of each row and column). In this case there are **eleven** possible logical products **(AND)** output response conclusions.

Although not absolutely necessary, rule matrices usually have an **odd number** of rows and columns to accommodate a "zero" center row and column region. This may not be needed as long as the functions on either side of the center overlap somewhat and continuous dithering of the output is acceptable since the "zero" regions correspond to **"no change"** output responses the lack of this region will cause the system to continually hunt for "zero". It is also possible to have a different number of rows than columns. This occurs when numerous degrees of inputs are needed.

The maximum number of possible rules is simply the product of the number of rows and columns, but definition of all of these rules may not be necessary since some input conditions may never occur in practical operation. The primary objective of this construct is to map out the universe of possible inputs while keeping the system sufficiently under control.

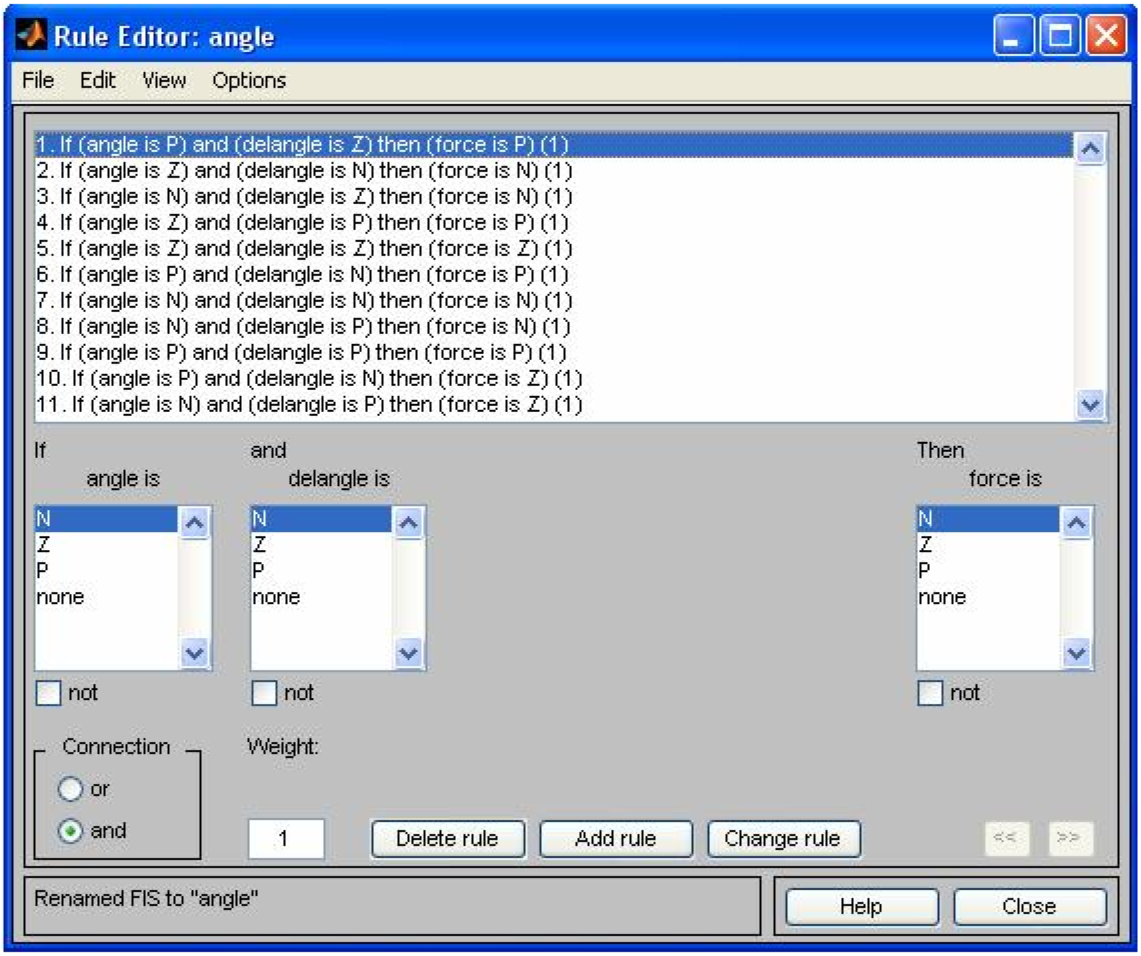


Figure 4.5.2(a): Rule Editor for Fuzzy Logic Controller (angle)

Constructing rules using the graphical Rule Editor interface is fairly self evident. Based on the descriptions of the input and output variables defined with the FIS Editor, the Rule Editor allows you to construct the rule statements automatically, by clicking on and selecting one item in each input variable box, one item in each output box, and one connection item. Choosing none as one of the variable qualities will exclude that variable from a given rule. Choosing not under any variable name will negate the associated quality. Rules may be changed, deleted, or added, by clicking on the appropriate button.

To insert the first rule in the Rule Editor, select the following:

- P under the variable **Angle**
- Z under the variable **Delangle**
- The **and** radio button, in the **Connection** block
- P under the output variable, **Force**

The resulting rule is

1. If (**angle is P**) and (**delangle is Z**) then (**force is P**) (1)

The numbers in the parentheses represent weights that can be applied to each rule if desired. You can specify the weights by typing in a desired number between zero and one under the Weight setting. If you do not specify them, the weights are assumed to be **unity (1).**

Follow a similar procedure to insert the second until eleven rules in the Rule Editor to get

2. If (**angle is Z**) and (**delangle is N**) then (**force is N**) (1)
3. If (**angle is N**) and (**delangle is Z**) then (**force is N**) (1)
4. If (**angle is Z**) and (**delangle is P**) then (**force is P**) (1)
5. If (**angle is Z**) and (**delangle is Z**) then (**force is Z**) (1)
6. If (**angle is P**) and (**delangle is N**) then (**force is P**) (1)
7. If (**angle is N**) and (**delangle is N**) then (**force is N**) (1)
8. If (**angle is N**) and (**delangle is P**) then (**force is N**) (1)
9. If (**angle is P**) and (**delangle is P**) then (**force is P**) (1)
10. If (**angle is P**) and (**delangle is N**) then (**force is Z**) (1)
11. If (**angle is N**) and (**delangle is P**) then (**force is Z**) (1)

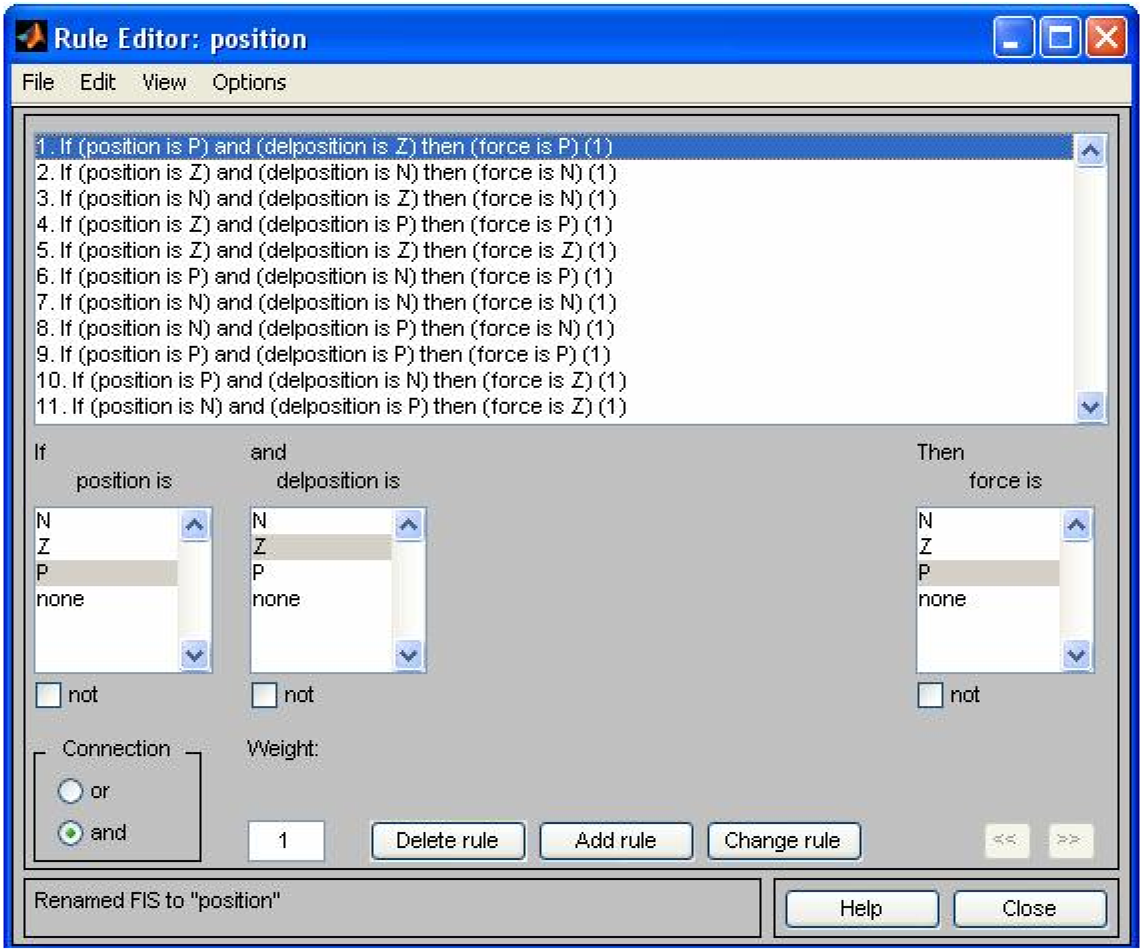


Figure 4.5.2.2. : Rule Editor For Fuzzy Logic Controller (position)

The resulting rule is

1. If (**position is P**) and (**delposition is Z**) then (**force is P**) (1)
2. If (**position is Z**) and (**delposition is N**) then (**force is N**) (1)
3. If (**position is N**) and (**delposition is Z**) then (**force is N**) (1)
4. If (**position is Z**) and (**delposition is P**) then (**force is P**) (1)
5. If (**position is Z**) and (**delposition is Z**) then (**force is Z**) (1)
6. If (**position is P**) and (**delposition is N**) then (**force is P**) (1)
7. If (**position is N**) and (**delposition is N**) then (**force is N**) (1)
8. If (**position is N**) and (**delposition is P**) then (**force is N**) (1)
9. If (**position is P**) and (**delposition is P**) then (**force is P**) (1)
10. If (**position is P**) and (**delposition is N**) then (**force is Z**) (1)
11. If (**position is N**) and (**delposition is P**) then (**force is Z**) (1)

At this point, the fuzzy inference system has been completely defined, in that the variables, membership functions, and the rules necessary to calculate force are in place.

4.6 Gains Analyze For Stable State

In providing the stable condition for the ball to avoid dropping from beam we use gain as the value from suitable gains value that position before the Position and Angle Fuzzy Logic Controller. The suitable condition for position of the ball is shown below. The reference value for the ball position is at value 1. So to tune it in this simulation we use Unit Step. The final value that filled in is 1.

Tuning the exactly value that suit for the system need a lot of time to try and error for this project. The suppose graph for position is where we can derive the system produce the undreamed graph. It showed that the graph have rise time, overshoot, settling time and constant time.

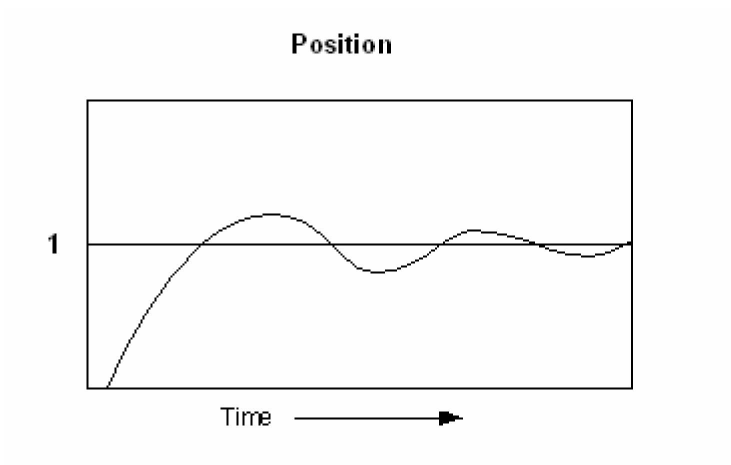


Figure 4.6(a): Under damped Graph for Position of the Ball

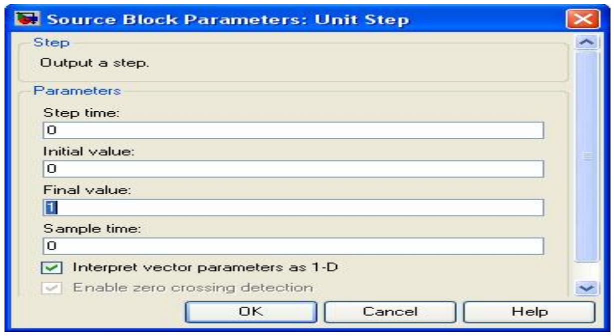


Figure 4.6(b): Unit step

To derive the suitable gains value we use two ways of method. The first method is by using signal constraint block parameter. It provides easier ways on tuning the right gains value using simpler method. The things that we need to adjust is using the constraint line that shown below. Just before we setting the pattern of constraint we must make sure that K1, K2, K3 and K4 parameter values change to 1.

It can search the suitable gains values range beyond each boundry. The range for amplitude versus time less effective because of the bigger range of parameter. the value that successfully meet the requirement will be display at the progress block. The main problem happen for this method and still unsolved was it cannot be reset back to the original form. The method need to be redevelod each time we change the position of the bondry line.

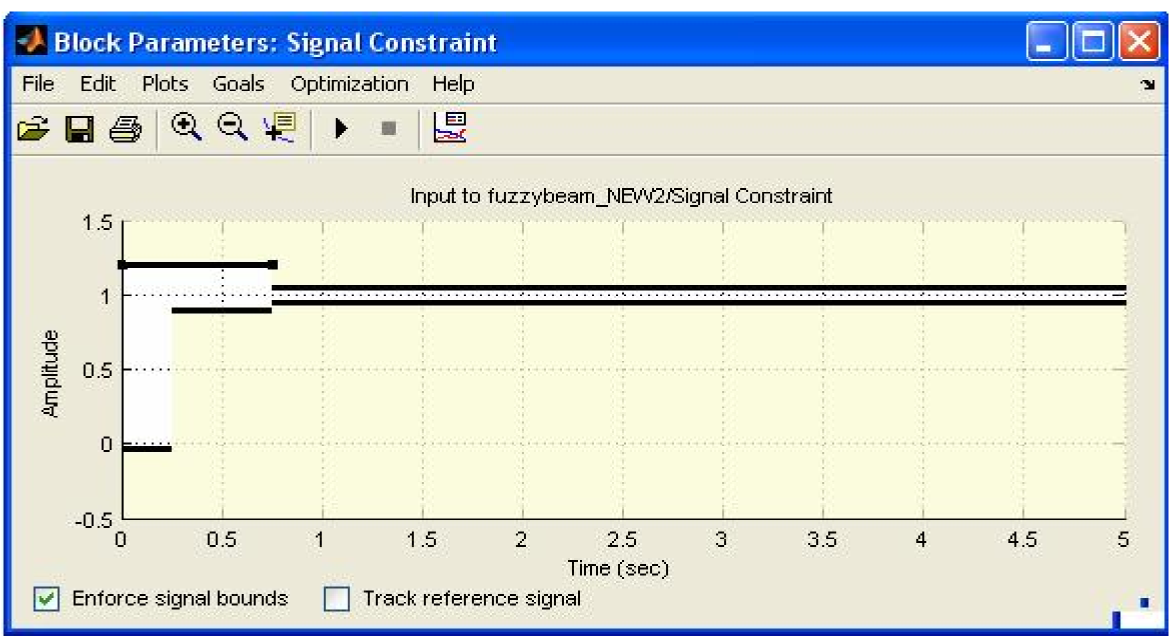


Figure 4.7: Signal Constraint Block Parameter

4.7 Problem Encounter and Terms

There are several errors that encountered during the development of this project. The final desired result for this project can't be obtained. There are several problem that occur that disturbance stable output condition. The first problem occur in this project is how to determined the perfect gains value to derive the under damped graph in the scope that show that the system was in the stable position condition. Basically the first method was by using try and error for the gains value for the gains block for K1, K2, K3 and K4. The possibility to obtain the perfect value that match for all gain is very low and it can be call less effective.

The second method that been used was signal constraint function block. Problem here is because this method needs to be rebuilt each time we run the searching progress. Each time we rebuilt we need more time and also actual setting must be done back. The future solution must be finding the appropriate technique that can provide the perfect value and it must be user friendly.

Besides that, there has some lack of knowledge in determine the overlapping for the membership function. For this system the overlapping that can be allowed is around 1 to 50 percent overlapping. Need to be remained each percentages can influence the result. So we suggest that fuzzy logic can be added as the compulsory learning subject in the electrical course.

CHAPTER 5

CONCLUSION AND RECOMMENDATION

5.1 Conclusion

At the end of this project, the overall progress doesn't achieve the objectives cause of the several problem mentions on the previous chapter. It means the beam and ball system still in the unstable condition even using Fuzzy Logic Controller. But the part of converting the mechanical beam and ball system to the Mathematical Modeling was successfully achieved using the Lagrian Equation (L= Ke -V). The non-linear equation finally can be use as the transfer function for the ball and beam system. This system can be successful stable if we can get the suitable gains value to imbalance the fuzzy logic controller for position and angle. The modeling system need the other method on tuning the membership parameter and also for determines the exact gains value.

5.2 Recommendation

For the future development and enhancement, there are much more can do with this system like integrate the mechanical beam and ball system to prove our simulation are exact as the mechanical system. Concept of mechanical system showed in the APPENDIX A. The beam and ball system also can be controlled using another method of controller such like PID controller (proportional–integral–derivative controller) that using linear equation. The LQR and Neural Network controller also can be including for this system.

Basically if we used varied of controller we can make the comparison about the system performance and also can search for the advantages/disadvantages for the stable condition. The fuzzy logic controller also can be implementing in the Inverted Pendulum System because the fuzzy system is very efficient to control the open loop system in the controlling the industrial mechanism.

5.2.1 Costing and Commercialization

The total cost for this project is only RM 500.00. This project is designed to be very affordable and still can handle any system that interface with the unstable condition. This project has very high potential because industrial need to control their machine so they can produce the perfect product by using the efficient machine handling. It very important because it can reduce their labor cost and also provides the perfect final good rather than human can do.

REFERENCES

[1] 20 January 2008, Nonlinear System Example, citing internet source url http://www.ece.nmsu.edu/ctrlsys/help/bnb.pdf

[2] 20 January 2008, Artificial Neural Network Identification & Control of an Inverted Pendulum, citing source url http://www.BSweeney_Meng04_report.pdf

[3] 20 January 2008, Artificial Neural Network Identification & Control of an Inverted Pendulum, citing source url http://www.TCallinan_MEng_03.pdf /

[4] 11 April 2008,Ball and Beam (Peter Wellstead) , citing internet source url http://control-systems-principles.co.uk

[5] 11 April 2008, Design Of Fuzzy Controller, citing internet source url Jan Jantzen jj@iau.dtu.dk

[6] 2 June 2008, Fuzzy Logic Toolbox, citing internet source url http://www.mathworks.com/

[7] 2 June 2008, Ball-on-Beam Balancer, citing internet source url http://www.personal.psu.edu/users/a/x/axa943/ball_on_beam_balancer.htm

[8] 2 June 2008, Modeling the Ball and Beam Experiment by Carnegie Mellon, citing internet source url
http://www.engin.umich.edu/group/ctm/examples/ball/ball.html

[9] 23 July 2008, Artificial Intelligence by Michael Negnevitsky,
http://www.pearsoned.co.uk/

[10] 21 August 2008, Encoder (The Newletter of the Seatle Robotic Society), citing internet source url
http://www.seattlerobotics.org/encoder/mar98/fuz/flindex.html

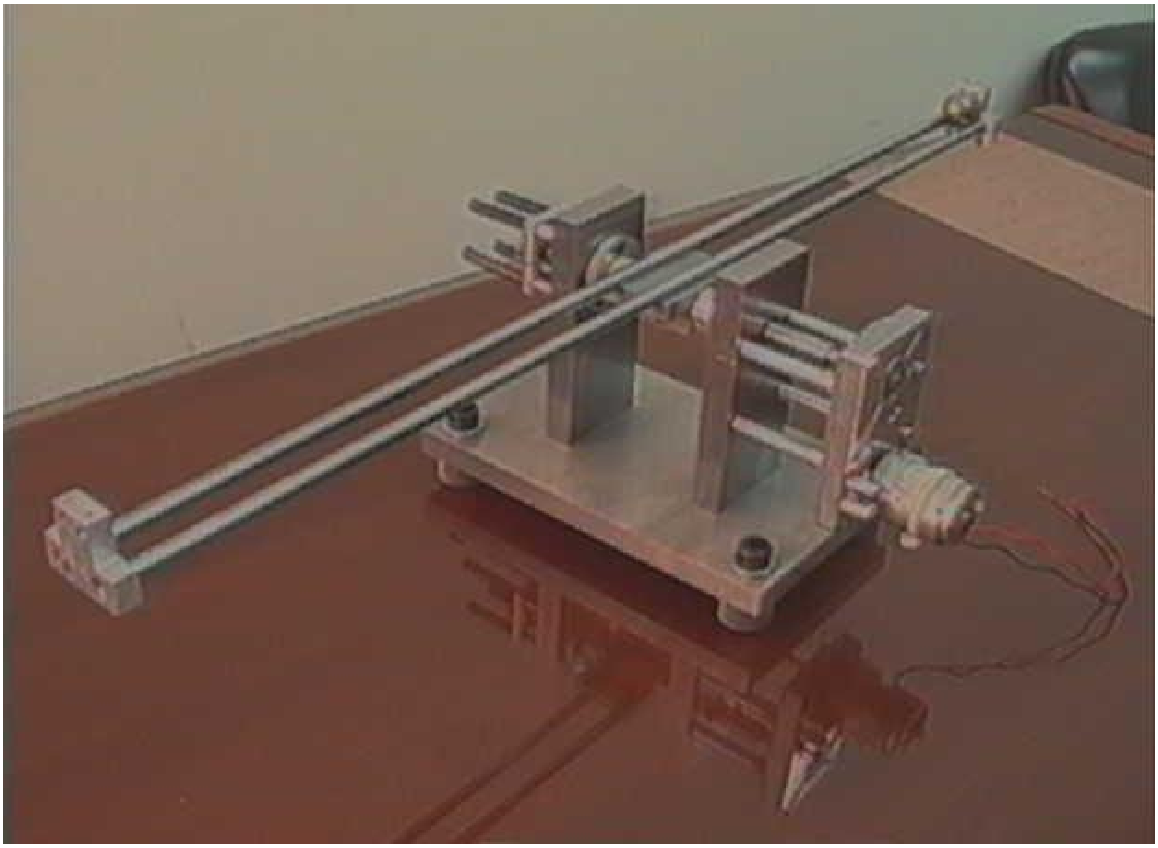
[11] 23 September 2008, Fuzzy Control System,
Citing internet source url
http://en.wikipedia.org/

[12] Artificial Intelligence, A Guide to Intelligence System, Second Edition 2006 by ADDISON WESLEY

[13] Fuzzy Logic , Father of Fuzzy Logic ,citing internet source url
http://www.public.iastate.edu/~baldusb/homepage.htm

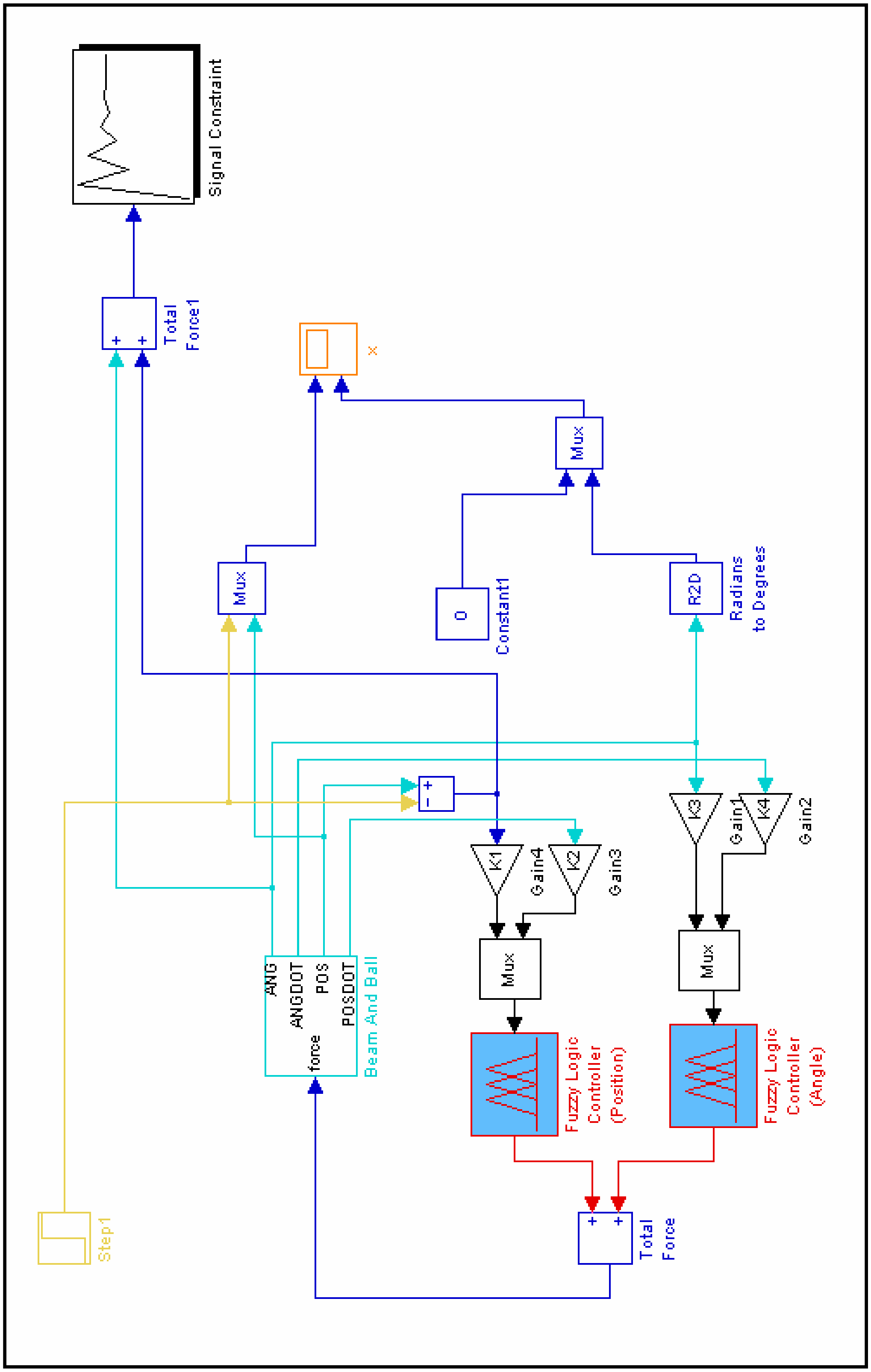
APPENDIX A

THE MECHANICAL CONCEPT FOR BEAM AND BALL SYSTEM



APPENDIX B

BEAM AND BALL SYSTEM SIMULINK DIAGRAM


Signal Constraint
Total Force1
x
Mux
Mux
0
Constant1
R2D
Radians to Degrees
K1
Gain4
K2
Gain3
K3
Gain1
K4
Gain2
Mux
Mux
ANG
ANGDOT
POS
POSDOT
force
Beam And Ball
Fuzzy Logic Controller (Position)
Fuzzy Logic Controller (Angle)
Step1
Total Force