**KON426E – INTELLIGENT CONTROL SYSTEMS**

**TERM PROJECT - SPRING 2021**

***TEAM 4***



| | |
|---|---|
| **Eymen KOŞAR** | 040170452 |
| **Mehmet Ali ARSLAN** | 040170402 |
| **Eray AYBEK** | 040170413 |
| **Oktay MUTLU** | 040170439 |



| | |
|---|---|
| **DUE DATE** | : 16.06.2022 |
| **COURSE ID & NAME** | : **KON 426E INTELLIGENT CONTROL SYSTEMS** |
| **COURSE COOR.** | : **DOÇ. DR. GÜLAY ÖKE ÖNEL** |

## CONTENTS

## TABLE OF FIGURES

# 1. INTRODUCTION

In this project our aim is control the ball and beam with fuzzy inference system. Firstly, we need to model the ball and beam system with nonlinear equations. Then for the fuzzy inference system we need to determine the inputs and outputs for designing membership functions. After, as a requirement for fuzzy we need to determine the rules. It is good to know that how ball and beam system behave in corresponding circumstances. Determining the best rules for the system depend on your experience on system and system states. So we need to learn the system first. We are considering to design state feedback controller to learn system and how its states behaves while its considered stable. After experiencing that will start to design fuzzy controller.

## 2. BALL AND BEAM SYSTEM

You can see the system illustration below.

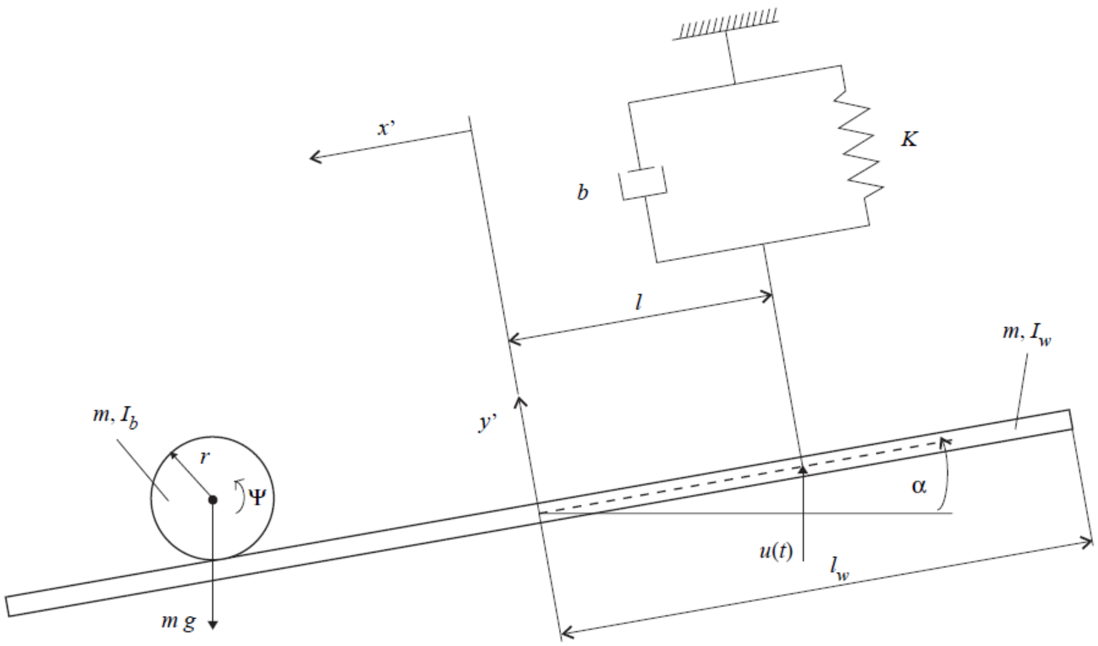


Figure 2-1. Mechanical structure of the ball and beam system

Here is the symbols and its meanings below,

| | | |
|---|---|---|
| $m$ | : | mass of the ball |
| $g$ | : | gravity |
| $r$ | : | roll radius of the ball |
| $I_b$ | : | inertia moment of the ball |
| $I_w$ | : | inertia moment of the beam |
| $M$ | : | mass of the beam |
| $b$ | : | friction coefficient of the drive mechanics |
| $K$ | : | stiffness of the drive mechanics |
| $u(t)$ | : | force of the drive mechanics |
| $l$ | : | radius of force application |
| $l_w$ | : | radius of beam |
| $x'$ | : | ball co-ordinate with respect to the beam |
| $y'$ | : | ball co-ordinate with respect to the beam |
| $\psi$ | : | angle of the ball to the beam |
| $\alpha$ | : | angle of the beam to the horizontal |

Table 2-1. Symbol list of the ball and beam system

We can obtain the dynamics of ball and beam system by using Lagrangian Method. The kinetic energy of the system is sum of ball's and beam's,

$$T = T_b + T_w \tag{2.1}$$

$$T_b = \frac{1}{2} m v_s^2 + \frac{1}{2} I_b \omega_b^2 \tag{2.2}$$

$$T_w = \frac{1}{2} I_w \dot{\alpha}^2 \tag{2.3}$$

Where $v_s$ is the velocity of the centre of mass and $\omega_b$ is angular velocity. They have to be defined as a function of the generalized coordinates.

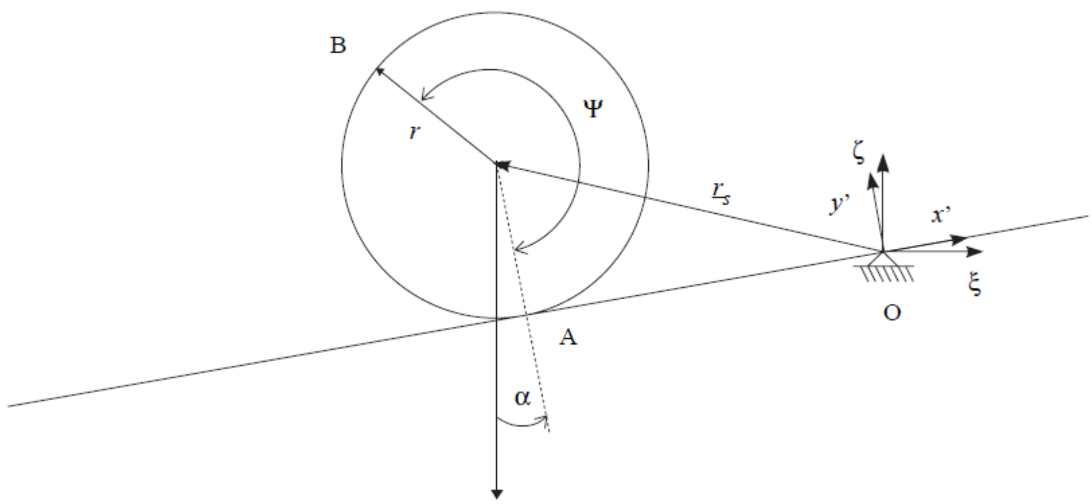


Figure 2-2. The definition of the position vector of the ball

Figure 2.2 shows all the variables required to formulate the velocity $v_s$ with the respect to the inertial system. The following equations will be used,

$$v_s = v_s' + \omega \times r_s \tag{2.4}$$

Where the position vector $r_s = [-x', r]^T$ with respect to the relavite system x' , y' and

$$v_s' = \frac{d' r_s}{dt} = [-\dot{x}' \;\; 0]^T \tag{2.5}$$

$$\frac{dr_s}{dt} = \begin{bmatrix} -\dot{x}' \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \dot{\alpha} \end{bmatrix} \times \begin{bmatrix} -x' \\ r \\ 0 \end{bmatrix} = \begin{bmatrix} -\dot{x}' - \dot{\alpha} r \\ -x'^{\dot{\alpha}} \\ 0 \end{bmatrix} \tag{2.6}$$

According to Equation 2.6, the ball velocity is,

$$v_s^2 = \dot{x'}^2 + 2\dot{x}'\dot{\alpha}r + (\dot{\alpha}r)^2 + \left(x'^{\dot{\alpha}}\right)^2 \tag{2.7}$$

For angular velocity $\omega_b$ ,

$$\omega_b = \dot{\psi} + \dot{\alpha} \tag{2.8}$$

$$\omega_b = \frac{\dot{x}'}{r} \tag{2.9}$$

Substituting this on the kinetic energy equation we get,

$$T = \frac{1}{2}m\left(\dot{x'}^2 + 2\dot{x}'\dot{\alpha}r + (\dot{\alpha}r)^2 + \left(x'^{\dot{\alpha}}\right)^2\right) + \frac{1}{2}I_b\left(\frac{\dot{x}'}{r}\right)^2 + \frac{1}{2}I_w\dot{\alpha}^2 \tag{2.10}$$

Lets write potential energy of the system,

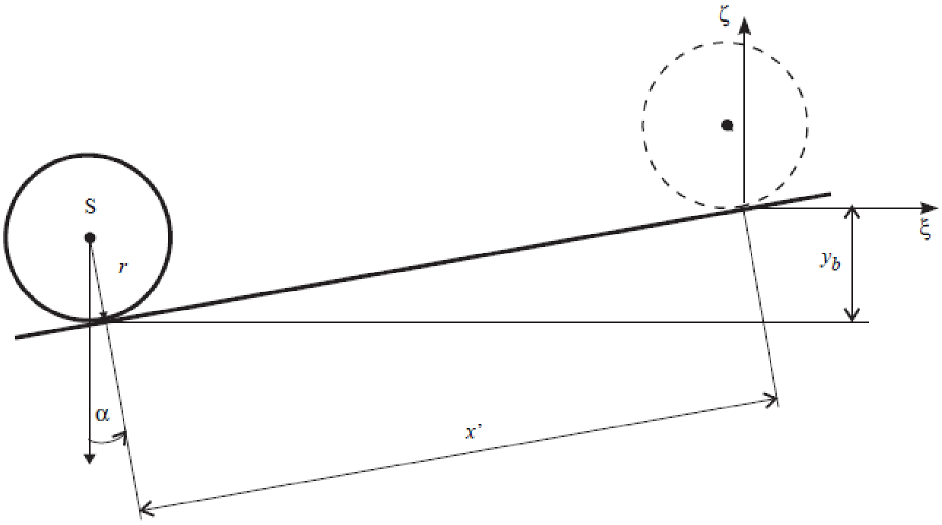


Figure 2-3. Potential energy of the system

Potential energy of the ball:

$$V_b = -mgy_b = -mgx' \sin(a) \tag{2.11}$$

Potential energy of the driving spinning:

$$V_f = \frac{1}{2}Kl^2a^2 \tag{2.12}$$

Summation of Equation 2.11 and 2.12 we can get the potential energy of the system:

$$V = V_b + V_f \tag{2.13}$$

$$V = -mgx' \sin(a) + \frac{1}{2}Kl^2a^2 \tag{2.14}$$

Generalized forces of the system, u(t) is the driving force:

$$Q_a^* = u(t)lcos(a) \qquad and \qquad Q_{x'}^* = 0 \tag{2.15}$$

Lagrange is becomes like this:

$$\mathrm{L} = \mathrm{T} - \mathrm{V} \tag{2.16}$$

$$\mathrm{L} = \frac{1}{2}m\left(\dot{x'}^2 + 2\dot{x'}\dot{\alpha}r + (\dot{\alpha}r)^2 + \left(x'^{\dot{\alpha}}\right)^2\right) + \frac{1}{2}I_b\left(\frac{\dot{x}'}{r}\right)^2 + \cdots$$
$$\ldots + \frac{1}{2}I_w\dot{\alpha}^2 - \left(-mgx'\sin(a) + \frac{1}{2}Kl^2a^2\right) \tag{2.17}$$

Lets write the Lagrange rules,

$$\frac{d}{dt}\left[\frac{\partial \mathrm{L}}{\partial \dot{\alpha}}\right] - \frac{\partial \mathrm{L}}{\partial \alpha} = u(t)lcos(a) \tag{2.18}$$

$$\frac{d}{dt}\left[\frac{\partial \mathrm{L}}{\partial \dot{x'}}\right] - \frac{\partial \mathrm{L}}{\partial x'} = 0 \tag{2.19}$$

From derivations we can obtain this system equations below:

★ $$\left(mx'^2 + I_b + I_w\right)\ddot{a} + \left(2m\dot{x'}x' + bl^2\right)\dot{a} + Kl^2a + (mr^2 + I_b)\frac{1}{r}\ddot{x'} - mgx'\cos(a) = u(t)lcos(a)$$

★★ $$\left(m + \frac{I_b}{r^2}\right)\ddot{x'} + (mr^2 + I_b)\frac{1}{r}\ddot{a} - mx'\dot{a}^2 = mgsin(a)$$

We can transform this equations to state space representation, first of all our states and some abbreviations to simplifying equations are below:

$$x_1 = x' \ \ (Position\ of\ the\ ball)$$

$$x_2 = \dot{x}' \ \ (Velocity\ of\ the\ ball)$$

$$x_3 = a \ \ (Angle\ of\ the\ beam)$$

$$x_4 = \dot{a}\ \ (Angular\ velocity\ of\ the\ beam)$$

$$a_1 = m + \frac{I_b}{r^2} \qquad a_2 = (mr^2 + I_b)\frac{1}{r} \qquad a_3 = mg$$

$$b_1 = I_b + I_w \qquad b_2 = 2m \qquad b_3 = bl^2 \qquad b_4 = Kl^2 \qquad b_5 = (mr^2 + I_b)\frac{1}{r} \qquad b_6 = mg$$

With this abbreviations our starred equations will become like this:

$$\bigstar \qquad (mx'^2 + b_1)\ddot{a} + (b_2\dot{x}'x' + b_3)\dot{a} + b_4 a + b_5\ddot{x}' - b_6 x'\cos(a) = u(t)l\cos(a)$$

$$\bigstar\bigstar \qquad a_1\ddot{x}' + a_2\ddot{a} - mx'^{\dot{a}^2} = a_3\sin(a)$$

Finally if we solve these equations for our states, we get our states' derivatives as this:

$$\dot{x_1} = x_2$$

$$\dot{x}_2 = \frac{a_2\,[\,(b_2 x_1 x_2 + b_3)x_4 + b_4 x_3 - b_6 x_1\cos(x_3)\,] + (m x_1^2 + b_1)\,(a_3\sin(x_3) + m x_1 x_4^2) - a_2\, l\cos(x_3)\,u(t)}{a_1\,(m x_1^2 + b_1) - a_2 b_5}$$

$$\dot{x_3} = x_4$$

$$\begin{aligned}\dot{x}_4 = {} & \frac{-(b_2 x_1 x_2 + b_3)\,x_4 - b_4 x_3 + b_6 x_1\cos(x_3)}{m x_1^2 + b_1} - \frac{b_5\,(a_3\sin(x_3) + m x_1 x_4^2)}{a_1\,(m x_1^2 + b_1) - a_2 b_5} \\ & - \frac{a_2 b_5\,[(b_2 x_1 x_2 + b_3)\,x_4 + b_4 x_3 - b_6 x_1\cos(x_3)]}{(m x_1^2 + b_1)\,(a_1\,(m x_1^2 + b_1) - a_2 b_5)} \\ & + \left(1 + \frac{a_2 b_5}{a_1\,(m x_1^2 + b_1) - a_2 b_5}\right)\frac{l\cos(x_3)\,u(t)}{m x_1^2 + b_1}\end{aligned}$$

We assumed the motor's transfer function as '1'.

# 3. CONTROL METHOD

Since our objective is controlling the system with fuzzy inference system, firstly we decided to control the system with conventional ways to see its behaviour. When we search ball and beam literature widely its seen that there are excessively state feedback controller used. So for the first step we decided to design state feedback controller for our four stated nonlinear system. We will not give the theory of state feedback controller design technique we used since it is not the aim of this document.

After designing state feedback controller its observed that ball and beam system can handle the input first 15-20 seconds than it goes unstable. You can see the beam angles behaviour in Figure 3.1. Also in Figure 3.2 you can see the state feedback designed controller for ball and beam system.

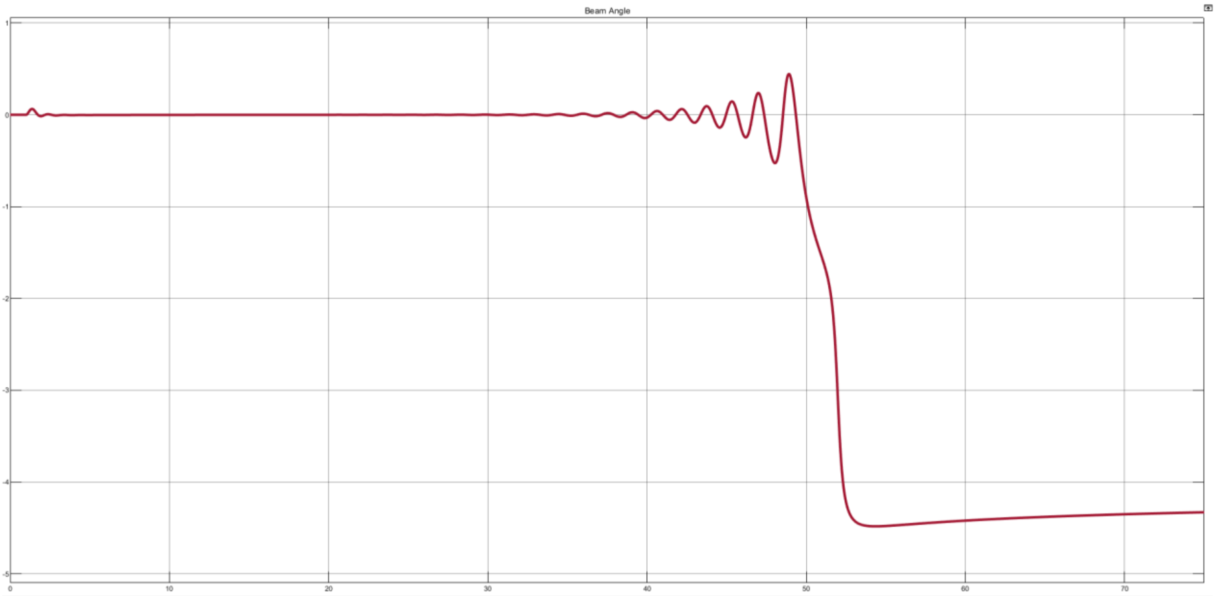


Figure 3-1. Beam angle plot for step input

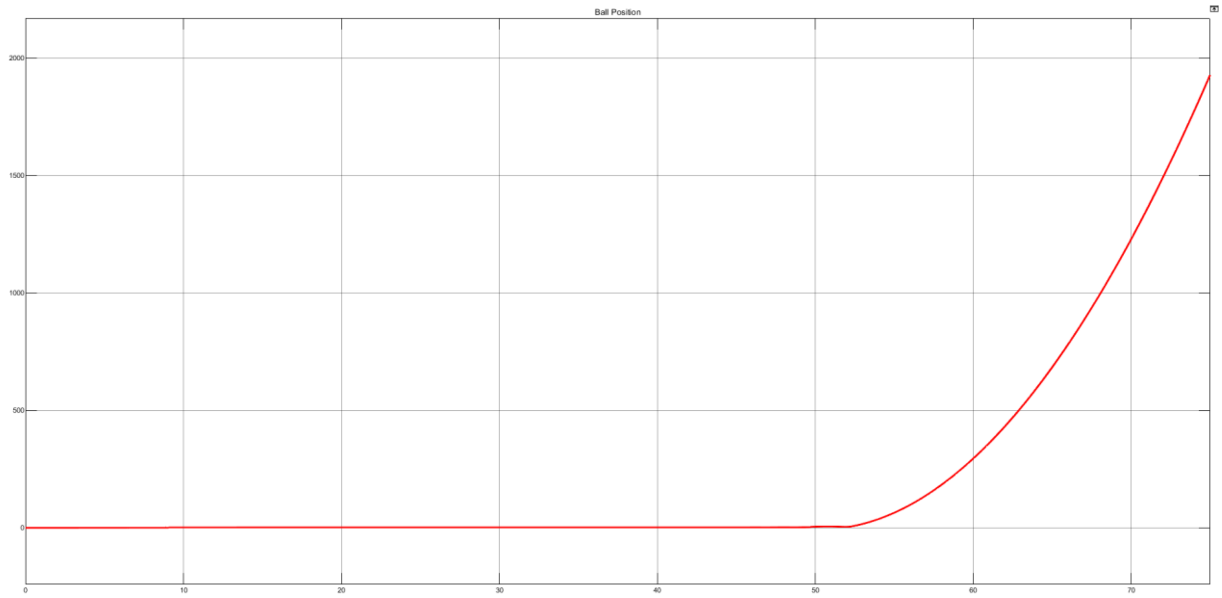


Figure 3-2. Ball position for state feedback controller

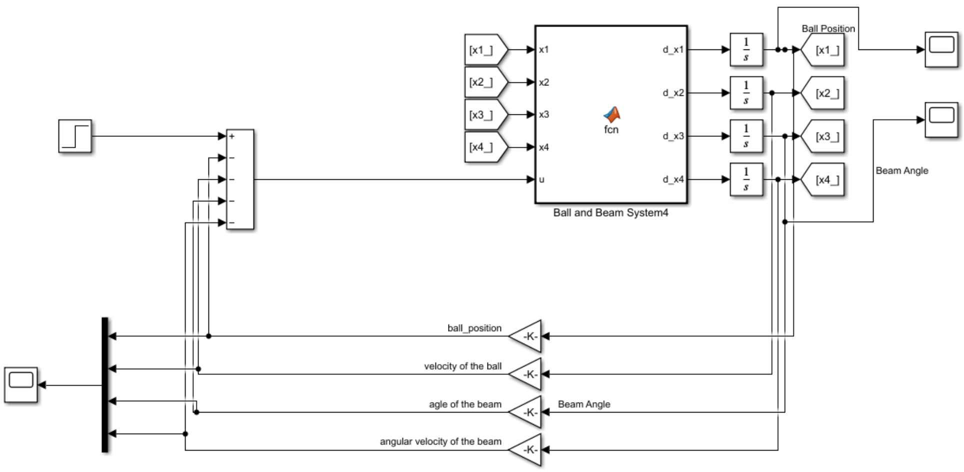


Figure 3-3. Block diagram for state feedback controller

This design gained us a experience that how system behaves while it is stable and how it behaves when it is unstable. It is important to experience that how system is behaving in corresponding circumstaces. Because we will use this gained experience in fuzzy designing.

For fuzzy design we decided design it with two inputs which are ball position and ball velocity. You can see below the membership functions for each input.

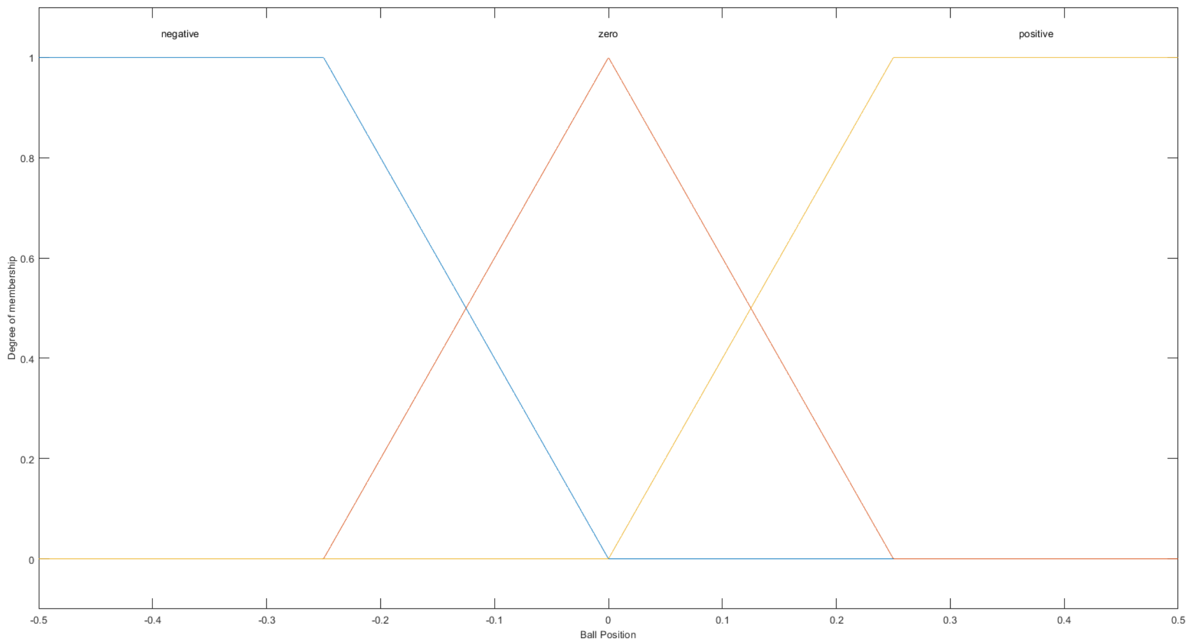


Figure 3-4. Membership function for ball position

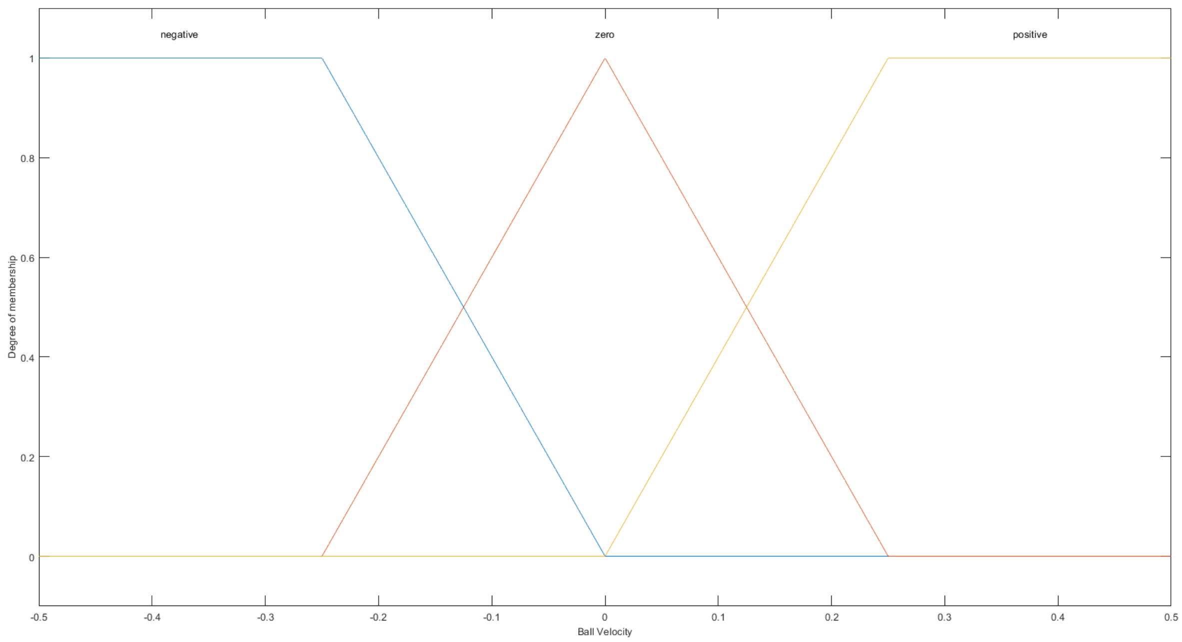


Figure 3-5. Membership function for ball velocity

When it comes to rules, we have 9 rules as expected and they are:

1. If ball position is zero and ball velocity is zero then torque is zero,
2. If ball position is zero and ball velocity is negative then torque is positive,
3. If ball position is zero and ball velocity is positive then torque is negative,
4. If ball position is negative and ball velocity is zero then torque is positive,

5. If ball position is negative and ball velocity is negative then torque is big positive,
6. If ball position is negative and ball velocity is positive then torque is zero,
7. If ball position is positive and ball velocity is zero then torque is negative,
8. If ball position is positive and ball velocity is negative then torque is zero,
9. If ball position is positive and ball velocity is positive then torque is big negative.

Also, you can see in Figure 3-5 that our membership functions for the output which is torque.

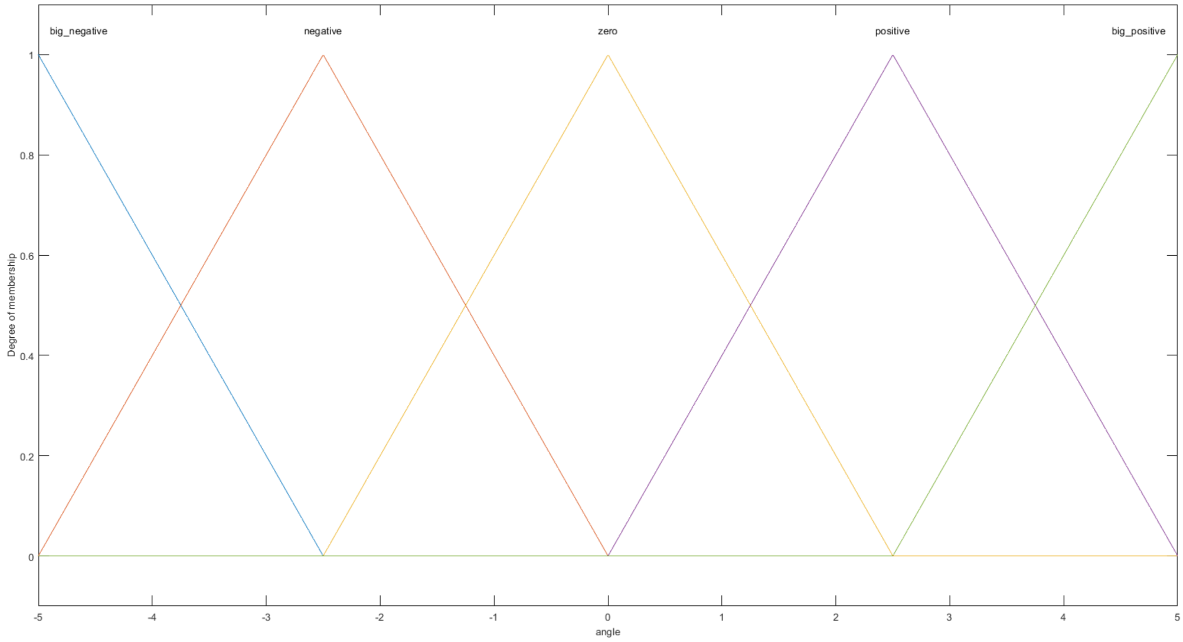


Figure 3-6. Membership functions for output

Also You can see the 3D plot of our rule table in Figure 3-6.

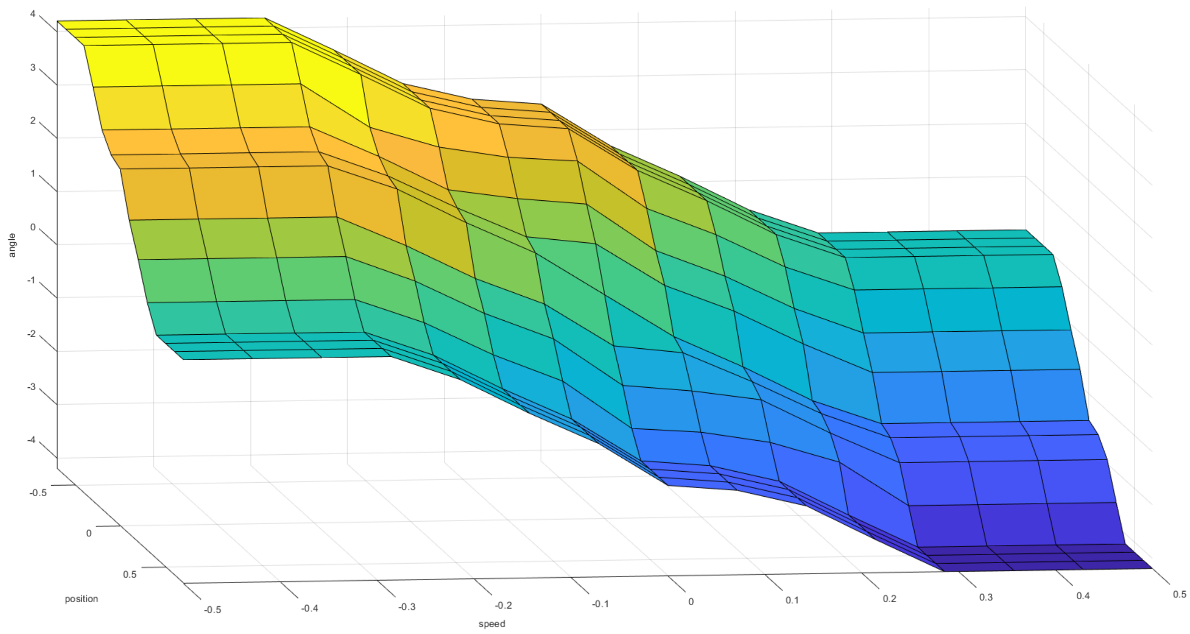


Figure 3-7. 3D Graph for the rule table

After designing, we obtained that we can not control the ball and beam system with only fuzzy. You can see beam angle plot in Figure 3-7 and ball position in Figure 3.8.

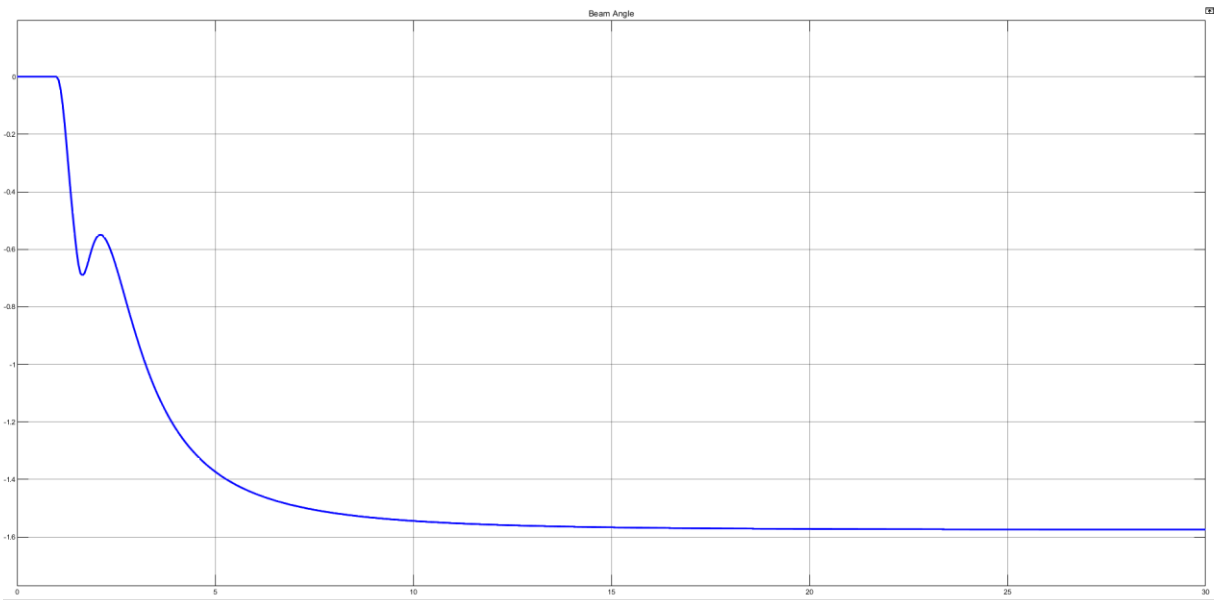


Figure 3-8. Beam angle plot with fuzzy

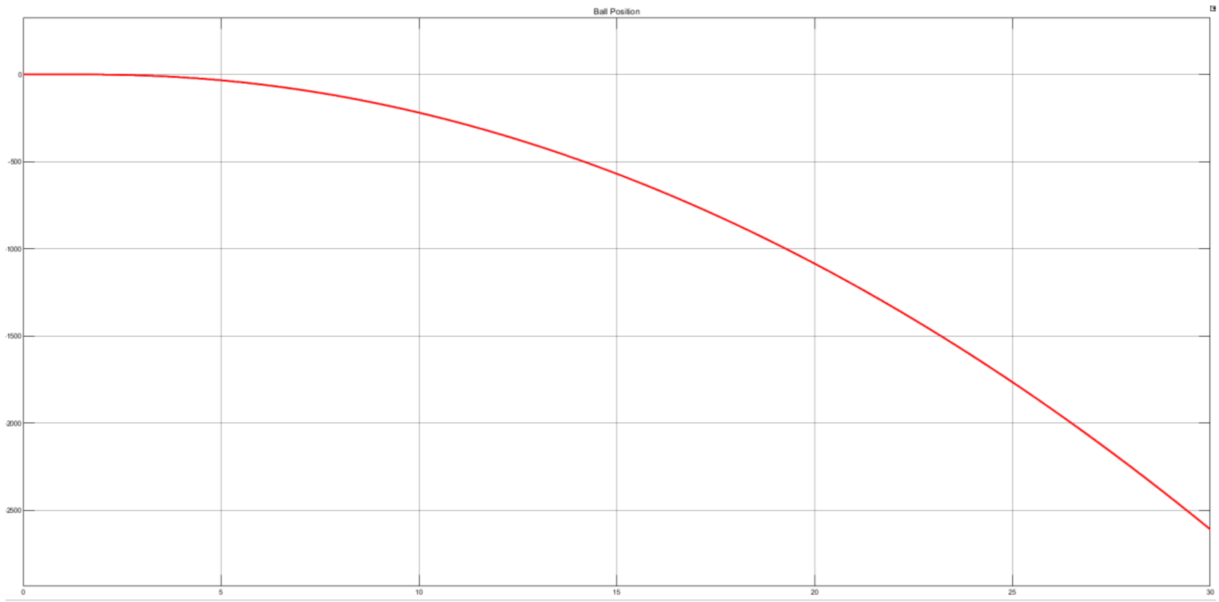


Figure 3-9. Ball position graph with fuzzy controller

After seeing this results we decided the combine the state feedback controller and fuzzy inference controller to control ball and beam system. After implementing and changing membership boundaries we can control the system succesfully. We did not use the state feedback gains we obtained at the first for this structure. We used genetic algorithm to find state feedback gains for the sytem in Figure 3-9.

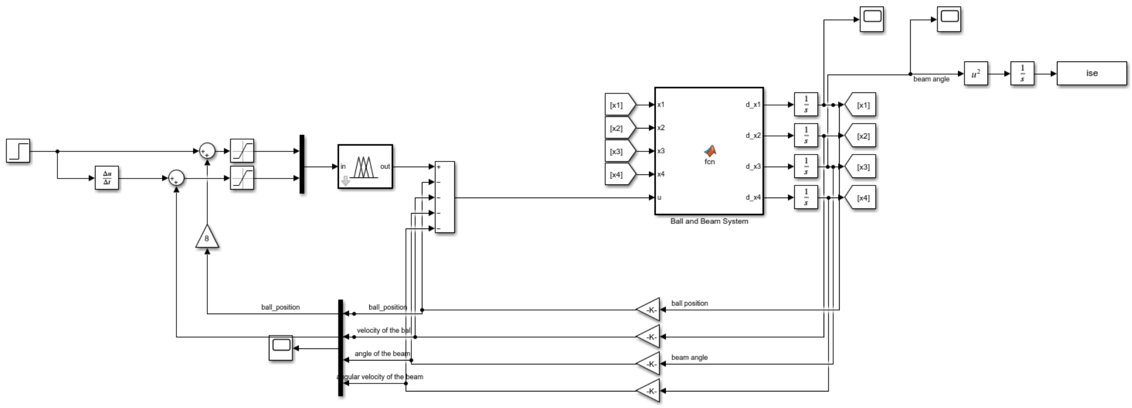


Figure 3-10. The system with fuzzy and state feedback controller combined

We used ISE criteria as objective function in genetic algorithm. Which was given in Equation 3.1.

$$ISE = \int_{0}^{\infty} e^2(t)dt \tag{3.1}$$

Genetic algorithm gives us the gains as;

$$State\ Feedback\ Gains = \begin{bmatrix} 6.27 \\ 5.82 \\ 15.78 \\ 1.39 \end{bmatrix} \tag{3.2}$$

Also we multiplied the ball position which is fuzzy's first input by 8 to firing rules more properly. You can see the simulation result for the given control method in Chapter 4. After combining fuzzy and state feedback controller we updated the membership boundaries due to our experience and it provide better controller performance for the system.

## 4. SIMULATION RESULTS

For step input, you can see the simulation results below.

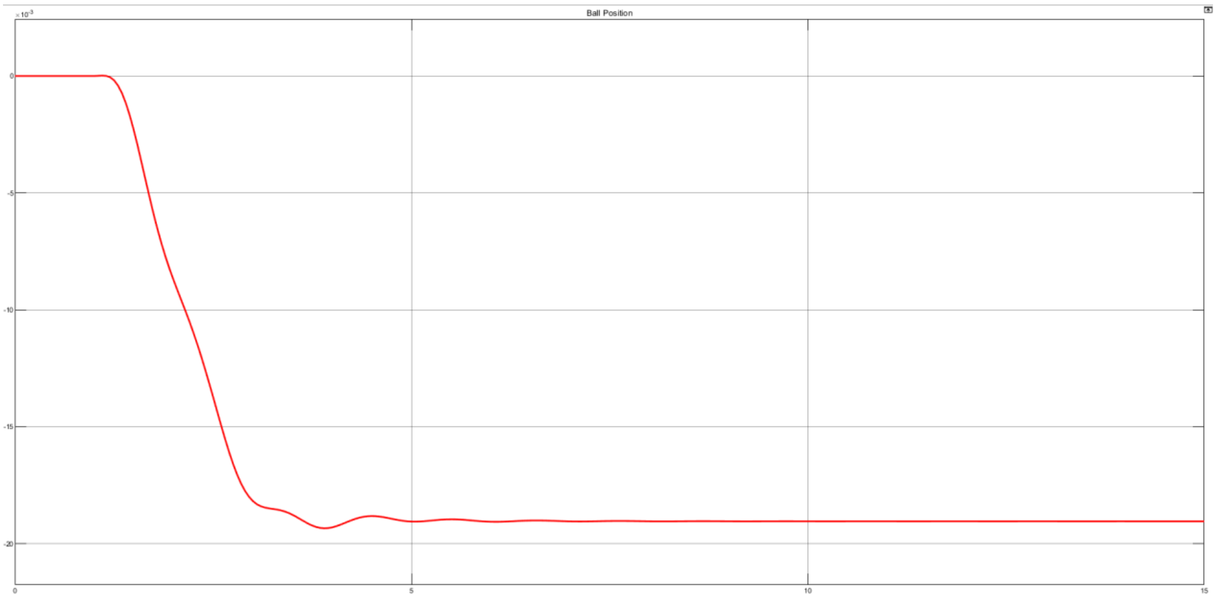


Figure 4-1. Ball position for step input

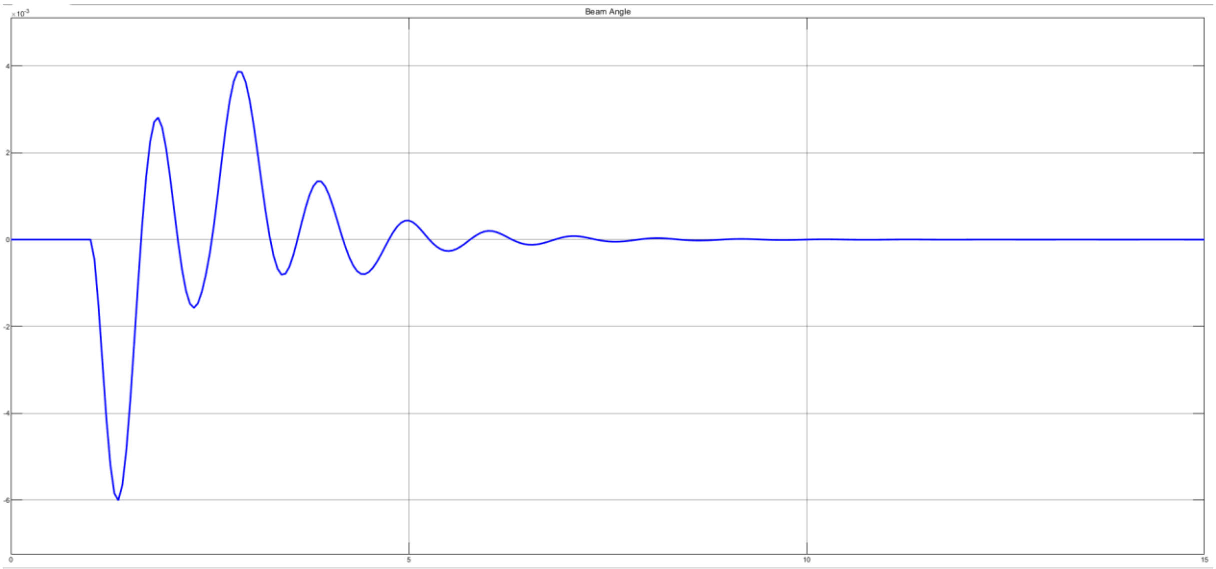


Figure 4-2. Beam angle for step input

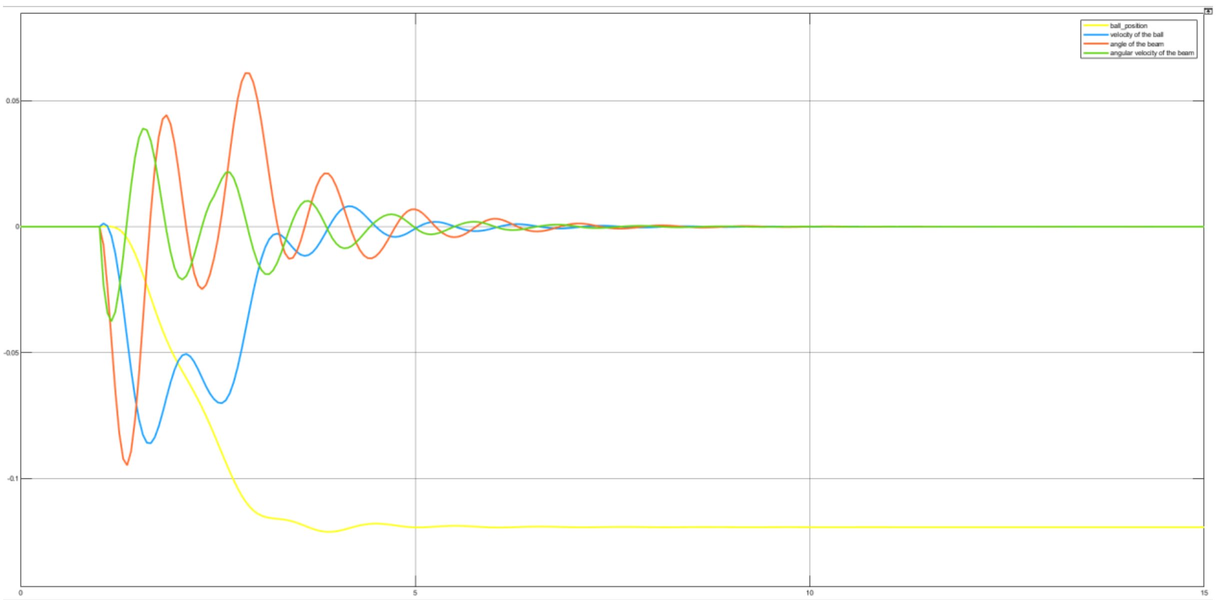


Figure 4-3. All states in one plot

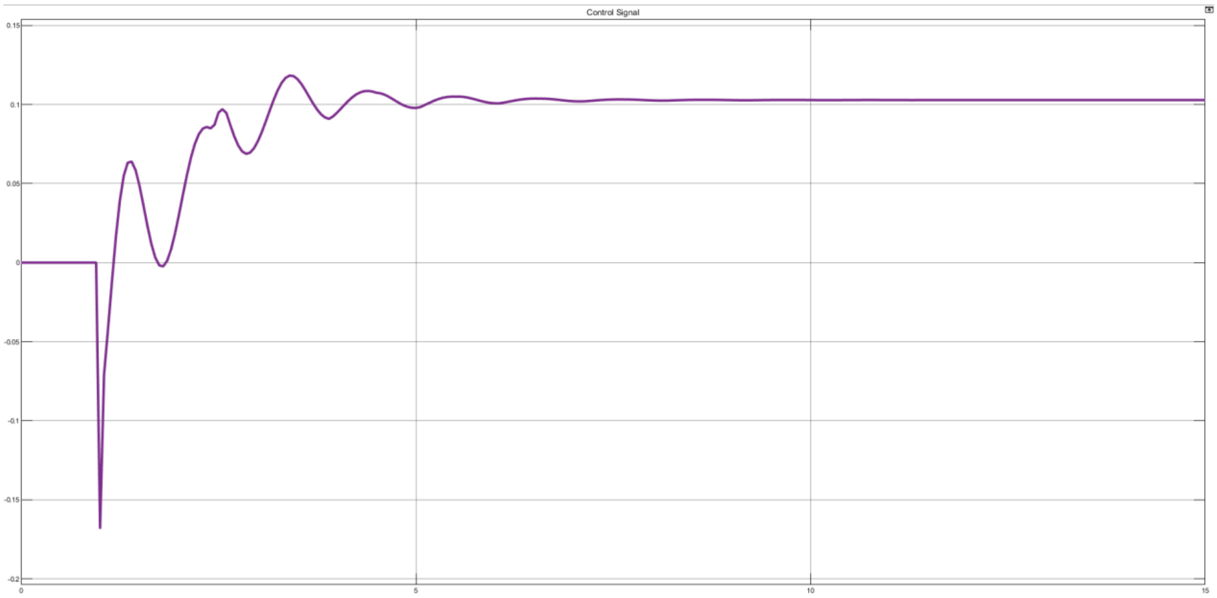


Figure 4-4. Control signal for step input

## [redacted] 4.1. Bonus, System Performance In Disturbance

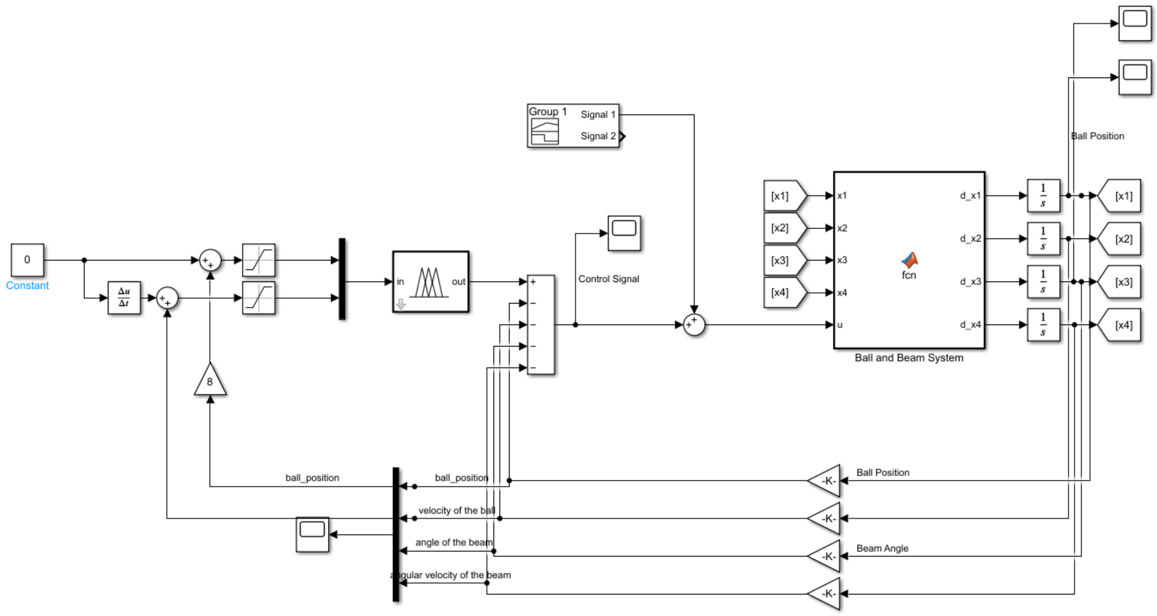


Figure 4-5. Block diagram to see systems disturbance performance

You can see the given disturbance from in Figure 4-6.

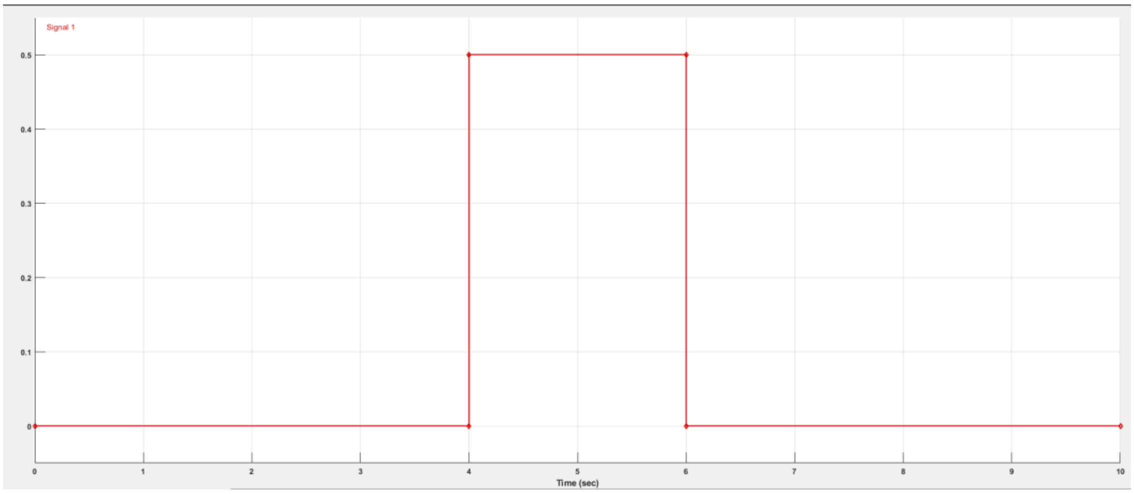


Figure 4-6. Given disturbance

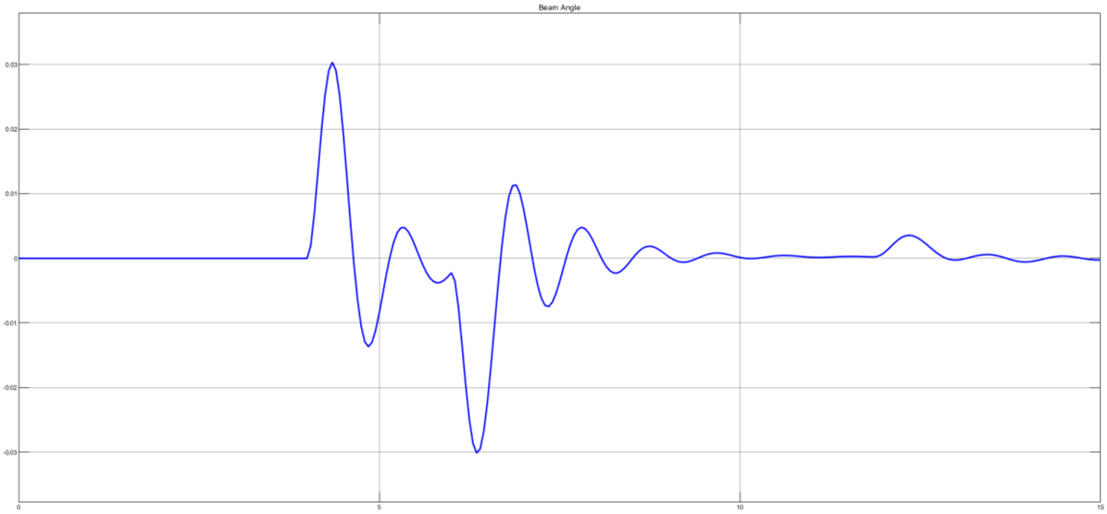


Figure 4-7. Beam angle for disturbed system

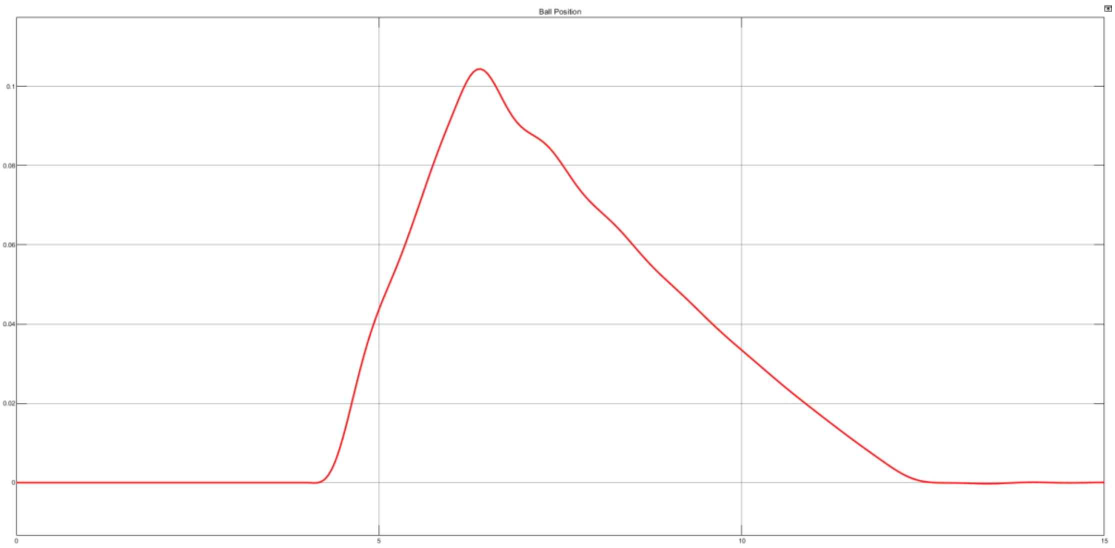


Figure 4-8. Ball position for disturbed system

# 5. CONCLUSION

Firstly, we searched literature to find a widely used method to control nonlinear ball and beam system. After researching literature, its obtained that state feedback controller is the commonly used to control ball and beam system. So, we designed our four stated system to state feedback controller to control it. We can not say that our system was succesful for state feedback controller but we learned and gained experience of systems dynamics. And that gained experience lead up to design fuzzy for the ball and beam. We implemented a fuzzy for two input and one output but this was not enough to control the system since it has four states. If we design a fuzzy with four inputs with 3 membership, this situation will obligate us to determine 81 rules. So we decided to combine state feedback and fuzzy to control ball and beam. After combining two controllers we determined feedback gains with genetic algorithm. And changed some membership boundaries to increase performance of the system. After some error-trial we found some optimum values to take stable response from step input.

# 6. CODES

```
%% Position Controller

position_fuzzy = mamfis(...
                'NumInputs', 2, 'NumInputMFs', 3,...
                'NumOutputs', 1, 'NumOutputMFs', 5,...
                'AddRule', 'none');

position_alt = -0.7;
position_ust = 0.7;
position_gap = linspace(position_alt, position_ust, 5);

speed_alt = -0.5;
speed_ust = 0.5;
speed_gap = linspace(speed_alt, speed_ust, 5);

angle_alt = -0.2;
angle_ust = 0.2;
angle_gap = linspace(angle_alt, angle_ust, 5);


position_fuzzy.Inputs(1).Name = 'position';
position_fuzzy.Inputs(1).Range = [position_alt position_ust];
position_fuzzy.Inputs(1).MembershipFunctions(1).Name = 'negative';
position_fuzzy.Inputs(1).MembershipFunctions(1).Type = 'trapmf';
position_fuzzy.Inputs(1).MembershipFunctions(1).Parameters = [position_gap(1) position_gap(1) position_gap(2) position_gap(3)];

position_fuzzy.Inputs(1).MembershipFunctions(2).Name = 'zero';
position_fuzzy.Inputs(1).MembershipFunctions(2).Type = 'trimf';
position_fuzzy.Inputs(1).MembershipFunctions(2).Parameters = [position_gap(2) position_gap(3) position_gap(4)];

position_fuzzy.Inputs(1).MembershipFunctions(3).Name = 'positive';
position_fuzzy.Inputs(1).MembershipFunctions(3).Type = 'trapmf';
position_fuzzy.Inputs(1).MembershipFunctions(3).Parameters = [position_gap(3) position_gap(4) position_gap(5) position_gap(5)];

position_fuzzy.Inputs(2).Name = 'speed';
position_fuzzy.Inputs(2).Range = [speed_alt speed_ust];
position_fuzzy.Inputs(2).MembershipFunctions(1).Name = 'negative';
position_fuzzy.Inputs(2).MembershipFunctions(1).Type = 'trapmf';
position_fuzzy.Inputs(2).MembershipFunctions(1).Parameters = [speed_gap(1) speed_gap(1) speed_gap(2) speed_gap(3)];

position_fuzzy.Inputs(2).MembershipFunctions(2).Name = 'zero';
position_fuzzy.Inputs(2).MembershipFunctions(2).Type = 'trimf';
position_fuzzy.Inputs(2).MembershipFunctions(2).Parameters = [speed_gap(2) speed_gap(3) speed_gap(4)];

position_fuzzy.Inputs(2).MembershipFunctions(3).Name = 'positive';
position_fuzzy.Inputs(2).MembershipFunctions(3).Type = 'trapmf';
position_fuzzy.Inputs(2).MembershipFunctions(3).Parameters = [speed_gap(3) speed_gap(4) speed_gap(5) speed_gap(5)];

position_fuzzy.Outputs(1).Name = 'angle';
position_fuzzy.Outputs(1).Range = [angle_alt angle_ust];

position_fuzzy.Outputs(1).MembershipFunctions(1).Name = 'big_negative';
position_fuzzy.Outputs(1).MembershipFunctions(1).Type = 'trimf';
position_fuzzy.Outputs(1).MembershipFunctions(1).Parameters = [angle_gap(1) angle_gap(1) angle_gap(2)];

position_fuzzy.Outputs(1).MembershipFunctions(2).Name = 'negative';
position_fuzzy.Outputs(1).MembershipFunctions(2).Type = 'trimf';
position_fuzzy.Outputs(1).MembershipFunctions(2).Parameters = [angle_gap(1) angle_gap(2) angle_gap(3)];

position_fuzzy.Outputs(1).MembershipFunctions(3).Name = 'zero';
position_fuzzy.Outputs(1).MembershipFunctions(3).Type = 'trimf';
position_fuzzy.Outputs(1).MembershipFunctions(3).Parameters = [angle_gap(2) angle_gap(3) angle_gap(4)];

position_fuzzy.Outputs(1).MembershipFunctions(4).Name = 'positive';
position_fuzzy.Outputs(1).MembershipFunctions(4).Type = 'trimf';
position_fuzzy.Outputs(1).MembershipFunctions(4).Parameters = [angle_gap(3) angle_gap(4) angle_gap(5)];

position_fuzzy.Outputs(1).MembershipFunctions(5).Name = 'big_positive';
position_fuzzy.Outputs(1).MembershipFunctions(5).Type = 'trimf';
position_fuzzy.Outputs(1).MembershipFunctions(5).Parameters = [angle_gap(4) angle_gap(5) angle_gap(5)];
```

```
position_fuzzy.Outputs(1).MembershipFunctions(5).Parameters = [angle_gap(4) angle_gap(5) angle_gap(5)];

rules = [...
    "position == zero & speed == zero       => angle = zero"; ...
    "position == zero & speed == negative   => angle = positive"; ...
    "position == zero & speed == positive   => angle = negative"; ...

    "position == negative & speed == zero       => angle = positive"; ...
    "position == negative & speed == negative   => angle = big_positive"; ...
    "position == negative & speed == positive   => angle = zero"; ...

    "position == positive & speed == zero       => angle = negative"; ...
    "position == positive & speed == negative   => angle = zero"; ...
    "position == positive & speed == positive   => angle = big_negative"; ...
    ];

position_fuzzy = addRule(position_fuzzy, rules);
```

## 7. REFERENCES


- Amira GMBH, Ball and Beam BW500 Setup for Laboratary Experiments, 1999.