



Universidad de Valladolid



**ESCUELA DE INGENIERÍAS
INDUSTRIALES**

UNIVERSIDAD DE VALLADOLID

ESCUELA DE INGENIERIAS INDUSTRIALES

Grado en Ingeniería Electrónica Industrial y Automática

Ball and Plate System

Autor:

Sanz Gómez, Víctor

Fernando Tadeo Rico

Hautes Etudes d'Ingénieur

UNIVERSITÉ CATHOLIQUE DE LILLE

Valladolid, Marzo 2016.

TFG REALIZADO EN PROGRAMA DE INTERCAMBIO

TÍTULO: BALL AND PLATE SYSTEM

ALUMNO: Víctor Sanz Gómez

FECHA: 4/02/2016

CENTRO: Hautes Etudes d'Ingénieur

TUTOR: Abdelouhab AITOUCHE

RESUMEN

Este proyecto trata sobre el sistema de control a implementar en la planta “ball and plate” (bandeja y bola). Esta planta, de Feedback Instruments, está programada y controlada con LabVIEW y nuestro objetivo final es estudiar este sistema de control y transferirlo al entorno de Matlab/simulink.

Vamos a construir un modelo en espacio de estados y después, le añadiremos un controlador y un observador. Simularemos y compararemos cada sistema, añadiéndole ciertas variaciones y ruidos.

Finalmente también construiremos el modelo de esta misma planta utilizando la lógica difusa. Resulta un proyecto muy interesante para ser utilizado en educación ya que puede proporcionar una visión general y práctica del modelo en espacio de estados, como controlarlo, y una pequeña introducción a la lógica difusa de Takagi-Sugeno.

PALABRAS CLAVE

Control; espacio de estados; lógica difusa; Takagi-Sugeno; Simulink

BALL AND PLATE SYSTEM

Project Report

Víctor SANZ GÓMEZ
Under the supervision of Mr. Abdelouhab AITOUICHE
Hautes Etudes d'Ingenieur, Lille, 2015/2016

Abstract

This Project deals with the control system of the plant “ball and plate”. This plant, from Feedback Instruments, is programmed and controlled with LabVIEW software and our final goal is to study this control system and to transfer it to matlab/simulink environment.

We are going to build a state-space model, and then, we will add a controller and an observer. We will simulate and compare each system, also with noise and variations, and finally we will build a fuzzy logic model of this ball and plate's system. It is a very interesting project for educational purpose that can report us some practical general view of state-space model and how to control it, and a small introduction to the Takagi-Sugeno fuzzy logic.

Contents

1	Introduction	2
I	Space State System Model and Control Design	2
2	Mathematical Modeling	3
2.1	System's description	3
2.2	Plant Model	4
3	Control Design	6
3.1	Controller design: K design	7
3.2	Controller design: H design	11
3.3	Observer design	12
4	Influence of noises and variations	14
4.1	Ball's characteristic's modification	14
4.2	Noise's effect	16

II Takagi-Sugeno Model Design	18
5 Takagi-Sugeno Fuzzy logic	19
5.1 Fuzzy logic modeling	19
6 Simulation's comparison between state-space model and TS fuzzy model	21

1 Introduction

The study of this subject is aimed at students who have being introduced to the automatics and control engineering and want to read some practical examples of the modeling and control of a state-space system or as a little introduction to one of the more useful fuzzy logic's types: Takagi-Sugeno. It could also be interesting for those people who want to build their own ball and plate plant.

This specific plant can be useful for other project's foundation in different real situations, for instance, the dangerous ware transportation by truck or ship where we need to have it very horizontal and without brusque movements. Or more generally, wherever we need something to be equilibrated, as nowadays the trending drones.

Basing our project in the bibliography, we will use for obtaining the mathematical model of the plant principally the own Feedback's document "Ball and Plate Control Experiments 33-240-LabView" and the Final Project Report from Amirkabir University "Modelling and Control of Ball-Plate System".

We will continue with the control design, using the state space model in matlab and simulink. Here the main part of the documentation that will be use is the course's slides "Observation et commande dans l'espace d'etat" by professor Aitouche, apart from some others.

Finally we will use as reference "Takagi-Sugeno Fuzzy Modeling for Process Control" by Kamyar Mehran for the last part of our system where we remodel the plant but with fuzzy logic.

Concluding, this project is a mixture of teorical applied contents, and their representation and simulation in matlab and simulink enviroments.

Part I

Space State System Model and Control Design

2 Mathematical Modeling

In this section we have worked mostly with the documents we have previously mentioned in the introduction. Our goal is to build a mathematical model in the state space in order to base our control system in the state-space environment.

2.1 System's description

First of all, we should describe our system (see figure 1). It is based on the free movement of a plastic ball on a square plate and our mission is to control their position, velocity and acceleration. For doing so, we will have to use different sensors and actuators. In this case we will have two potentiometers and one camera for the input section, and four inductors used as actuators. Let's detail the entire system:

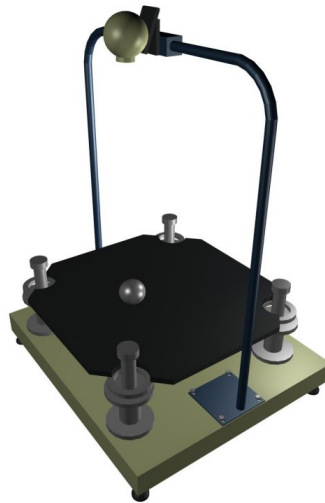


Figure 1: Feedback instruments machine

The plate is held in his middle point by the structure in charge of measuring his angles with two potentiometers, placed in perpendicular directions. This structure also gives more stability to the plate.

In each corner we will find an aluminum ring that due to the electromagnetism, will be in charge of the movement of the plate, and therefore of the ball.

About the inductors that provoke this electromagnetism, each one is controlled in couple with the other inductor in the opposite corner, in order to maintain the equilibrium.

The movement of the ball is obtained by the camera that with an intelligent vision system will be able to determine the position of the ball in the plate, and by simple deductions, the velocity and acceleration.

In brief, our system is composed of: the plate; the ball; two potentiometers that will catch the angles α and β of our plate; one camera in charge of getting the position x and y of the ball; one subsystem in charge of determinate the velocities and accelerations of the ball and plate based on the obtained values; and four inductors in charge of moving the plate and the ball.

- Entries: the values from the camera and potentiometers.
- Outputs: currents for the inductors.

2.2 Plant Model

Now that we have described our entire system, we are able to start the mathematical modeling. We are adding to the report the figure 2 to have a more visual representation of the system in order to making easier the comprehension of this and the following sections.

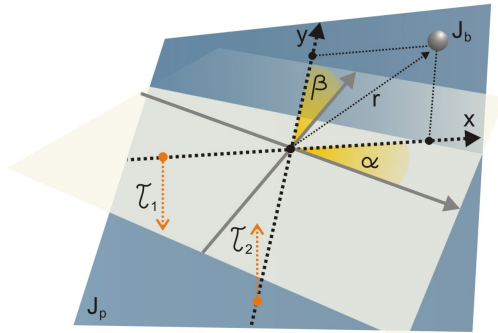


Figure 2: Reference system of the ball and plate

Like in all the documentation we have read, including Feedback's own documentation, we started the modeling with the energy equations of our system, kinetic and potential:

$$T = \frac{1}{2} \left[m(\dot{x}^2 + \dot{y}^2) + I_p(\dot{\alpha}^2 + \dot{\beta}^2) + I_b \left(\frac{\dot{x}^2 + \dot{y}^2}{r^2} \right) + I_b(\dot{\alpha}^2 + \dot{\beta}^2) + m(x\dot{\alpha} + y\dot{\beta})^2 \right] \quad (1)$$

$$W = mgx \sin \alpha + mgy \sin \beta \quad (2)$$

Following the instructions of the Feedback's manual, we have made the Lagrangian, deriving, and linearizing the equations, and after all we have translated the results into the space-state, which is the object of study here.

Working in the state-space, we need to define different kinds of variables: input variables, output variables, and state variables. We have taken for the entrance u the variables of angular acceleration $\ddot{\alpha}$ and $\ddot{\beta}$, $u = [\ddot{\alpha}, \ddot{\beta}]'$. The state variables, x_n , will be: the lineal position and velocity, x, \dot{x}, y and \dot{y} and angular position and velocity, $\alpha, \dot{\alpha}, \beta$ and $\dot{\beta}$. So we obtain $x = [x, \dot{x}, y, \dot{y}, \alpha, \dot{\alpha}, \beta, \dot{\beta}] = [x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8]$.

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \\ \dot{x}_5 \\ \dot{x}_6 \\ \dot{x}_7 \\ \dot{x}_8 \end{bmatrix} = \begin{bmatrix} x_2 \\ b \cdot (x_1 x_4 + x_5 x_8) \cdot x_4 - b \cdot g \cdot \sin x_3 \\ x_4 \\ 0 \\ x_6 \\ b \cdot (x_1 x_4 + x_5 x_8) \cdot x_8 - b \cdot g \cdot \sin x_7 \\ x_8 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u_x \\ u_y \end{bmatrix} \quad (3)$$

For the moment, our exit, or output variables, will be $y = x$, so the matrix C in this system will be an identity matrix 8x8 and the matrix D will be null.

To make easier the model, we divided the system (MIMO) in two different subsystems (SIMOs) (system (x,) and system (y,)) and we have omitted all the relations between these two systems due to the fact that the achieved values in this operations are very low and can be despised. Both for the first and the second system we obtain the following matrix for the state-space system:

$$A = \begin{bmatrix} 0 & -1 & 0 & 0 \\ 0 & 0 & -b \cdot g & 0 \\ 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 \end{bmatrix}; \quad (4)$$

$$B = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}; \quad (5)$$

Where b and g are constants values:

$$b = \frac{m}{m + \frac{I_b}{r^2}} = \frac{0.027}{0.027 + \frac{4.32 \cdot 10^{-6}}{0.02^2}} \quad (6)$$

$$g = 9.8$$

m =ball's mass; I_b =ball's moment of inertia; r =ball's radius and g =gravity.

Since now and for the rest of the report, we will refer to these independent subsystems as the complete system of the plant and we will work with them paying attention to only one direction. Let's say direction (x, α) .

In the figure 3 we can see the model description that we have written in matlab, first in one system, and also split in the two independent subsystems, easier to work with:

```

1
2 - b=0.027/(0.027+4.32e-6/0.02^2);
3 - g=9.81;
4
5
6 - %% coupled-system (1 system, 2 inputs, 8 outputs)
7 - A=[0 1 0 0 0 0 0 0; 0 0 -b*g 0 0 0 0 0; 0 0 0 1 0 0 0 0; 0 0 0
8 - B=[0 0; 0 0; 0 0; 1 0; 0 0; 0 0; 0 0; 0 0; 0 1]
9 - C=eye(8);
10 - D=zeros(8,2);
11 - system=ss(A,B,C,D)
12 - eig(A)
13
14 - %% non-coupled-system (2 systems, 1 input, 4 outputs)
15 - Aalpha=[0 1 0 0; 0 0 -b*g 0; 0 0 0 1; 0 0 0 0];
16 - Balpha=[0 0 0 1]';
17 - Calpha=eye(4);
18 - Dalpha=zeros(4,1);
19
20
21 - Abeta=[0 1 0 0; 0 0 -b*g 0; 0 0 0 1; 0 0 0 0];
22 - Bbeta=[0 0 0 1]';
23 - Cbeta=eye(4);
24 - Dbeta=zeros(4,1);

```

Figure 3: model description in matlab

3 Control Design

In this section we will mainly work with Simulink and Matlab, but we will also include some handmade methods in order to show different ways.

First of all, we are going to build in Simulink environment the state space system of the Ball and Plate dynamics that all we know using the data we have

already calculated in the previous point of modeling. The basic state space model that we will use for the next designs is show in the figure 4.

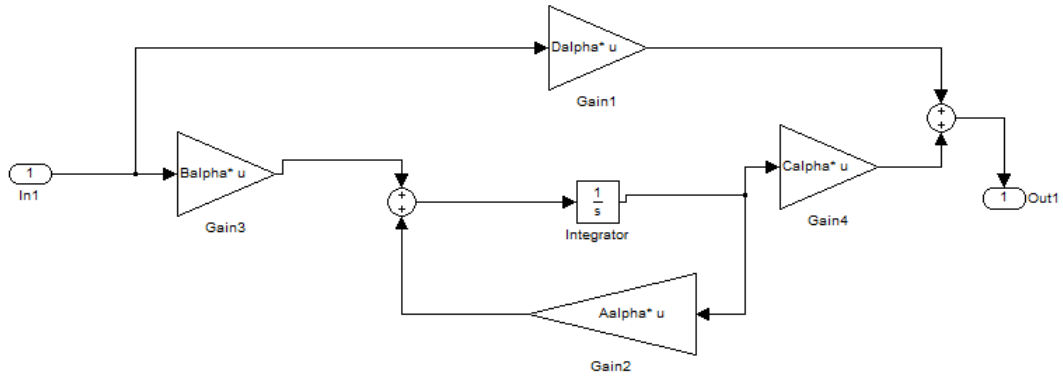


Figure 4: Simulink blocks of the state space model

As you can see, we are still working with the reduced model with only 4 state variables and 1 entry, because it is simpler to work with by this way and to observe the results and it makes no difference in the them using the reduced or the bigger version in the control.

As we are in the state space system, the controller that we will use, it is based on a proportional controller, is showed in the equation 7.

$$\underline{u}(t) = -K\underline{x}(t) + H\underline{y}(t) \quad (7)$$

So first of all we are going to implement the feedback controller and after calculating this K, we will add the H.

3.1 Controler design: K design

In this context we should firstly calculate the eigenvalues of the system without feedback (figure 4). With some common functions in matlab we can derive that they are four zeros. So now we have the necessity of translating these eigenvalues into the negative plane of the s-plane.

In this first attempt we are going to make a simulation of the system locating the four eigenvalues in -1, but this is not possible because matlab we cannot place poles with multiplicity greater than B's rank. So we have chosen the eigenvalues [-1 -0.99 -0.998 -0.999] in order to have a very similar response.

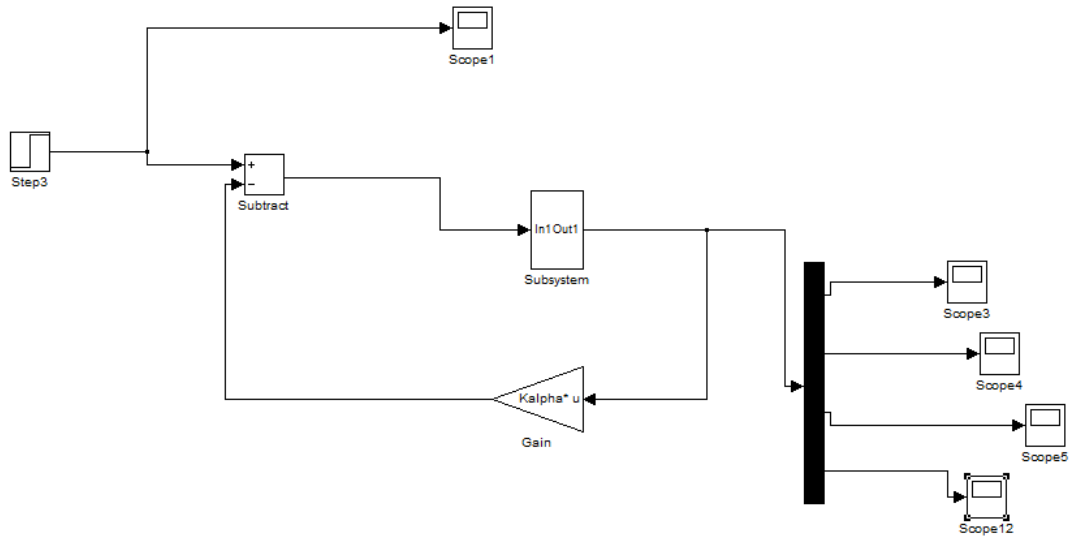


Figure 5: system with feedback

We have calculate the K by matlab with these eigenvalues that we have already said, and at the same time, we derived K by hand for the values -1 (since here we will call A alpha, A ; B alpha, B ; K alpha, K ; etc).

```

Kpplace(A,B,[-1. -0.99 -0.998 -0.999])
K =
    -0.1409    -0.5653     5.9610     3.9870

```

Figure 6: Matlab results for K

The results from matlab are showed in the figure 6, and are almost the same that we obtained manually $(-0.1427 -0.5708 -6 -4)$. We have simulated our system with this K in feedback and the results for position and velocity of both ball and plate can be observed in the figure 7.

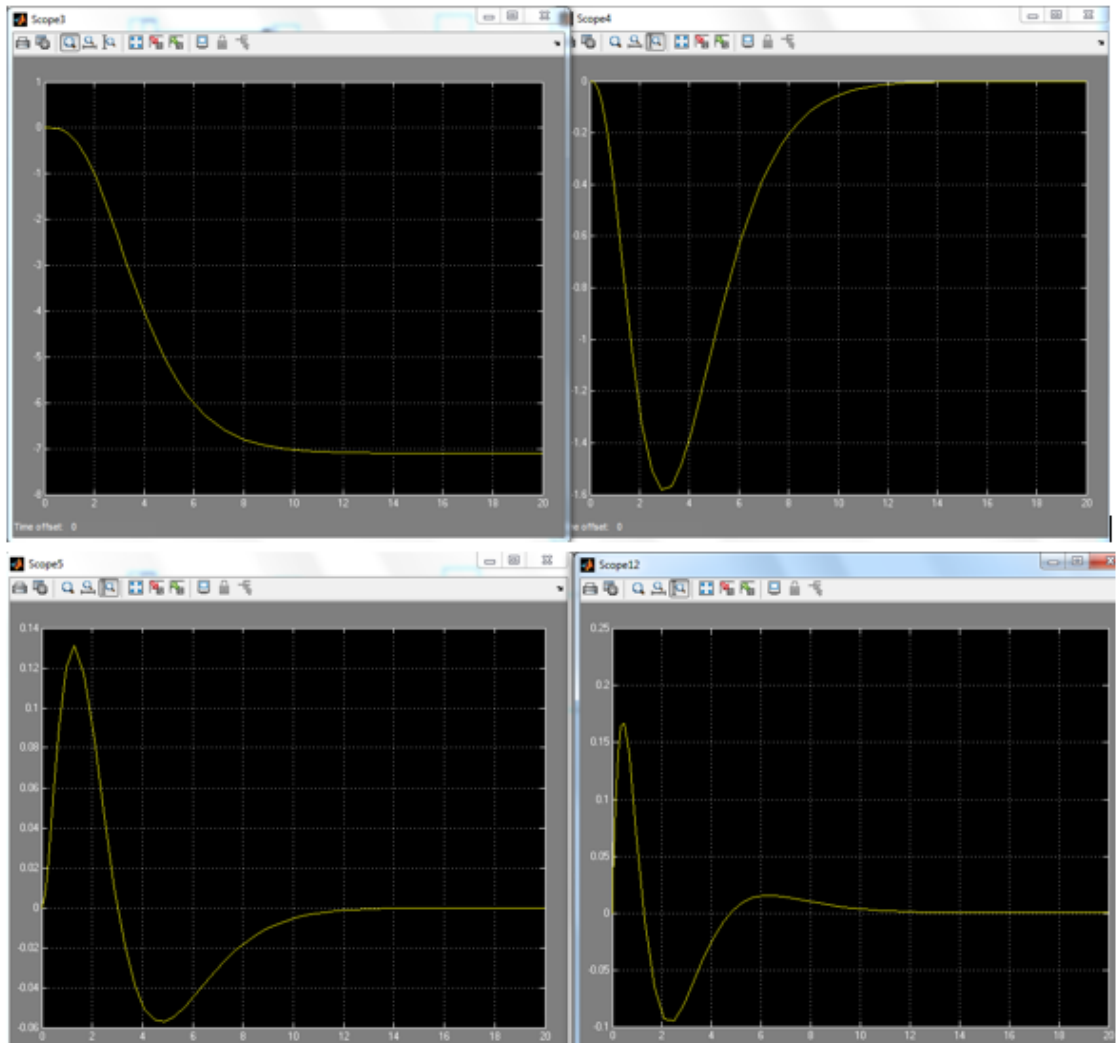


Figure 7: Results from the simulation. scope3: x ; scope4: \dot{x} ; scope5: α ; scope12: $\dot{\alpha}$.

As we expected and as we can see, each graphic tends to different values in a stable way, which means that for the poles that we have chosen before, we get a stable system.

In order to improve and get the best poles for our system, now we are going to study aswell the differences of speed if we change one of its eigenvalues. We will compare the answer of the ball velocity for an impulse entry for this diferent values of poles. We have chosen the velocity randomly.

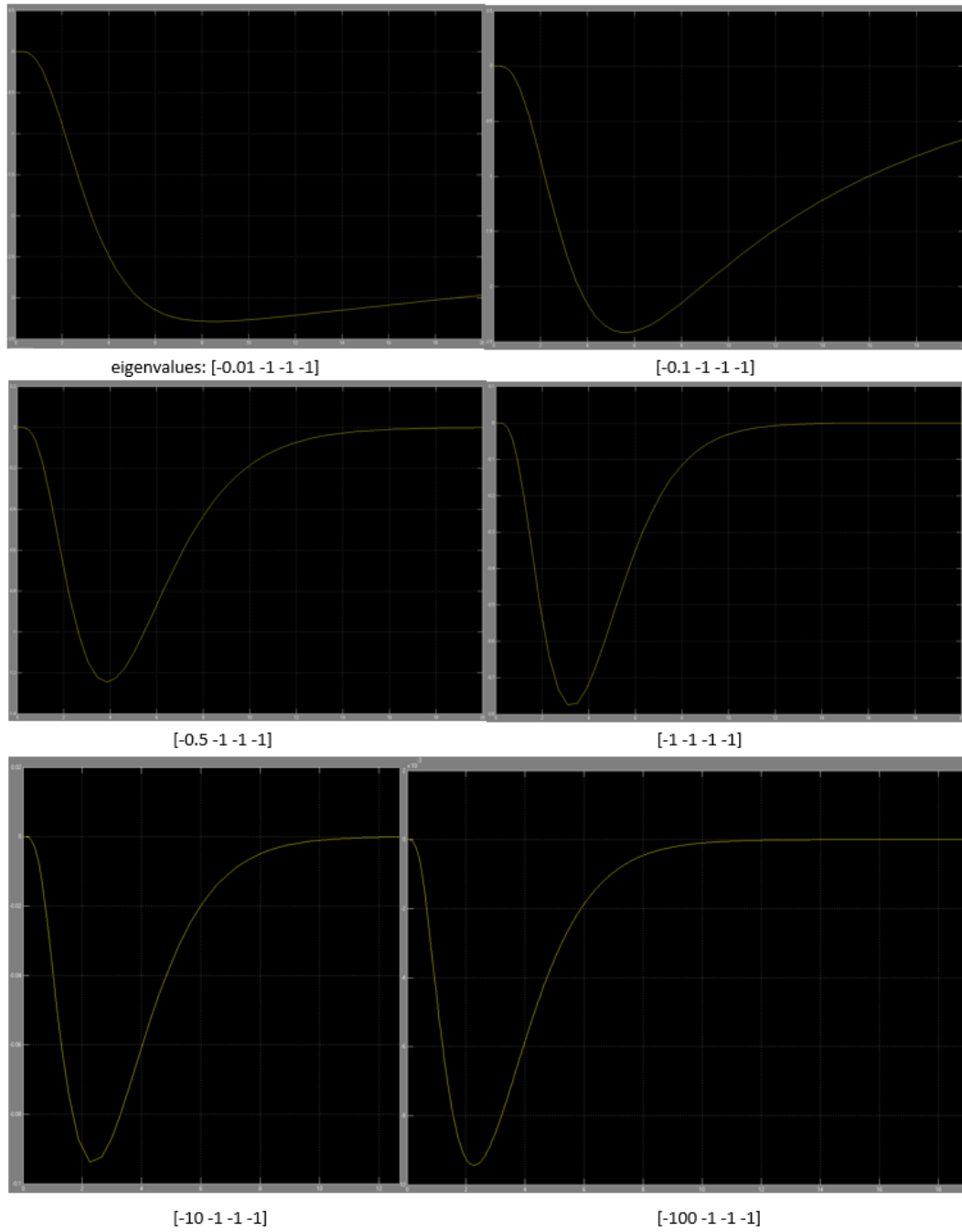


Figure 8: Ball's velocity answer for different eigenvalues

As we can see in figure 8, the highest value for the first eigenvalue, the fastest is the response of the ball's velocity. So after watching and comparing these results we have decided to select a pole in -5 in order to have a fast response but without locating the pole in the very left.

3.2 Controller design: H design

To guarantee the null steady errors, now we are adding the H matrix in our system that we previously mentioned in equation 7. This H matrix will be also very useful for watching and comparing the different responses, because it scales the answer in order to adjust it to the entry.

As we should know and we will discover in the equation 8, H can be only use in systems with the same amount of inputs and outputs. As we have only one input, we have to select only one output variable. This means that our H will not be a matrix neither a vector, but a scalar.

We have chosen the position of the ball because it is the simplest and in some way the most important, because the final goal of the system is just to move and hold the ball in a certain place on the plate. We will also see in the next subsection that this variable is the only one observable, ergo we will use anyway the ball's position.

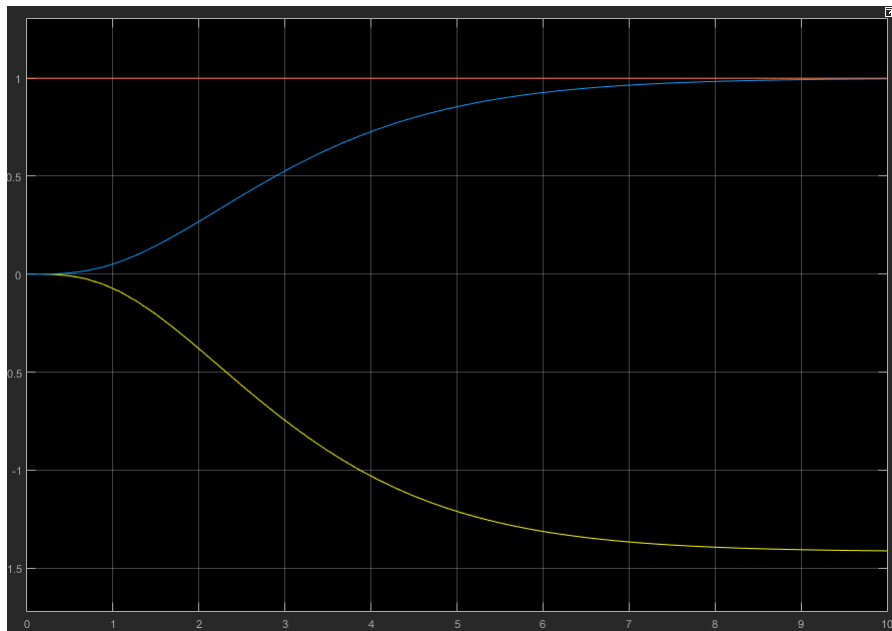


Figure 9: Ball's position outputs

Using the new matrix C: $C1 = [1, 0, 0, 0]$; we use the equation 8 in matlab

to derive H1 valor (H for the ball's position output).

$$H1 = -(C1 \cdot (A - B \cdot K)^{-1} \cdot B)^{-1} \quad (8)$$

The result of this equation is the scalar that will redimensionate our system. In figure 9 we can appreciate the comparison between the ball's position output before (in yellow), and after (in blue) adding this H scalar to our system with its step entry (in brown).

We can deduce that we have achieved our goal because now the answer is supposed to directly follow the input, which means that simplifies the analysis of the controller and the system.

Once we have finished our controller, continuing with the objective of improving our system, we are going to add to the plant and controller of our system, an observer.

3.3 Observer design

The observer in a system helps in the control of the plant with a feedback of the observer states. This means that if after a period of time, the virtual states of our system start to differ gently with the real states, the observer will reduce this error.

First of all we have measured the observability of both, our system SIMO and the subsystems of each SISO system which outputs compose the big system (the same system (x, α) or (y, β) but only with one output: ball's position, ball's velocity, plate's position or plate's velocity).

We have obtained as result that the SIMO system is observable but the only SISO subsystem observable is the first, the one which output is the ball's position, where we have already used for the H design (see subsection 3.2). This means that we can only derive the state space variables if we have as output at least the ball's position.

```

% first case: all states are measured
L=place(A',C',[-2. -0.99 -0.998 -0.999]);
L=L'
% second case the position 1 is measured
L1=place(A',C1',[-2 -1 -0.998 -0.999]);
L1=L1'

```

Figure 10: The observers

So as we can see in figure 10, we will try the two different observers for the two different observable systems: the first case where we don't have the H control and we have four outputs; and the second case where we add H to the controller but only the ball position is measured (SISO system). We will compare both of them with a controller without observer.

We designed for making the comparison the figure 11.

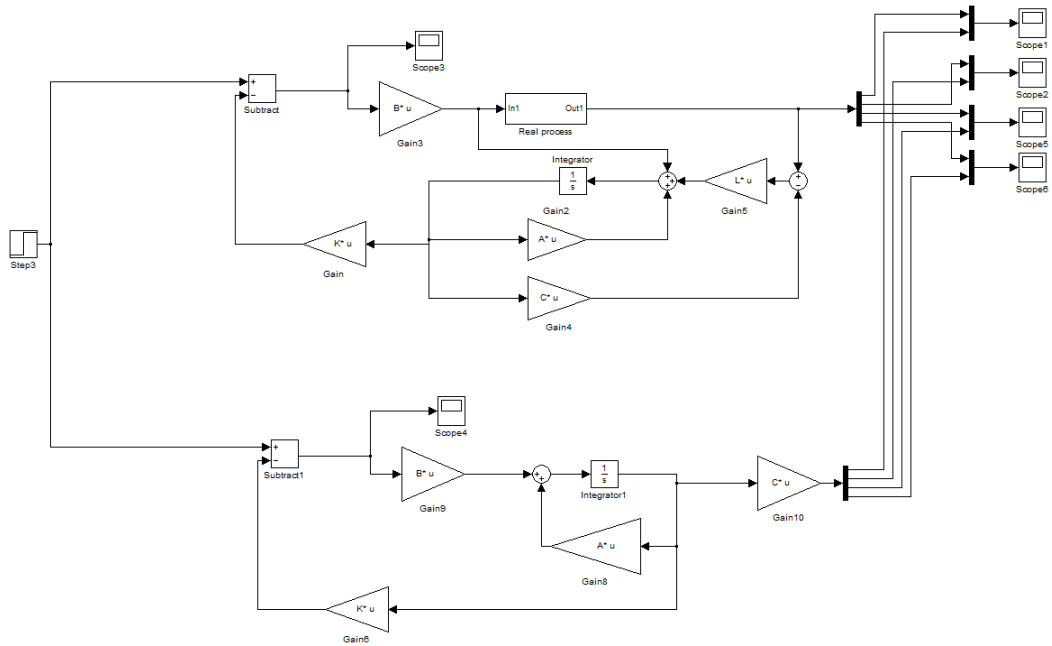


Figure 11: SIMO system diagram, with and without observer

$$L = \begin{bmatrix} 2.0000 & 1.0000 & 0 & 0 \\ 0 & 0.9900 & -7.0071 & 0 \\ 0 & 0 & 0.9980 & 1.0000 \\ 0 & 0 & 0 & 0.9990 \end{bmatrix}$$

Figure 12: SIMO L matrix

We are not attaching the graphic results for this comparison because there is no difference between the system with and without observer; the answers of both are identical.

If we watch the answer for the second case, the SISO system, we will obtain the same result, figures 13 and 14.

In this case we have also changed the poles for placing the L matrix, and also changing the value of the step entry. But we have obtained the same result than with the previous case. The answer for both systems, with and without observer, is the same than the answer showed in figure 9.

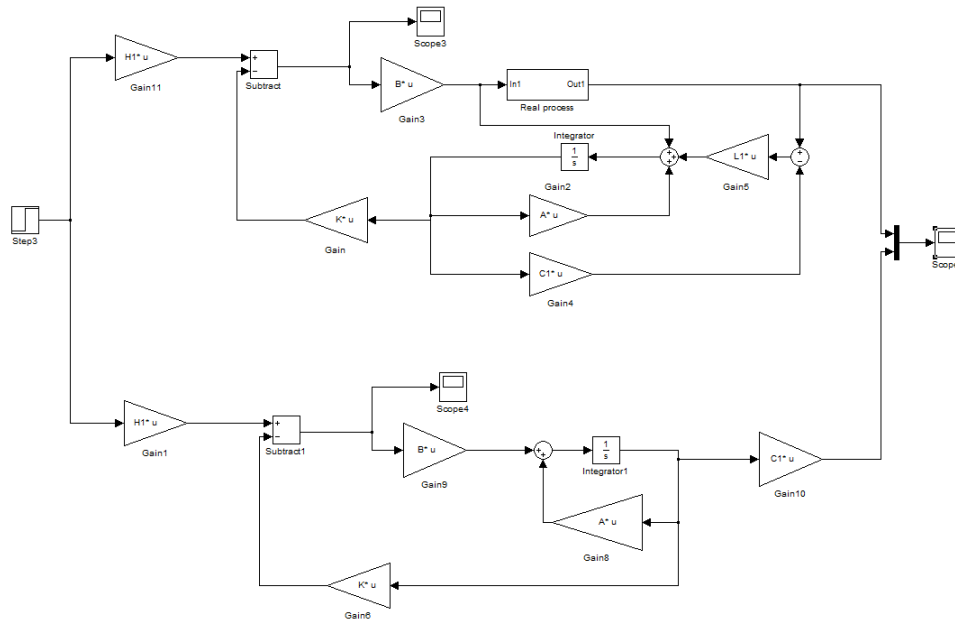


Figure 13: SISO system diagram, with and without observer

$$L1 = \begin{bmatrix} 4.9970 \\ 8.9880 \\ -0.9968 \\ -0.2846 \end{bmatrix}$$

Figure 14: SISO L matrix

4 Influence of noises and variations

In order to test the robustness of our system we will make a little battery of tests adding to it some different variations of some constants and also we will add some noise to watch how the system answers and if it is adequate for our demands.

4.1 Ball's characteristic's modification

The first of the robustness test that we will do is to modificate our b constant (equation 6), that depends on the ball's mass, radius, and inertia's moment. We have chosen this constant because is directly relationated with variables where there could not be a good precision in the meassurement instruments, as it could happen with the mass or the radius.

We have modified this b value in $\pm 5\%$ in the H coupled system (defined in subsection 3.2) and the answer to this variation is shown in figure 15.

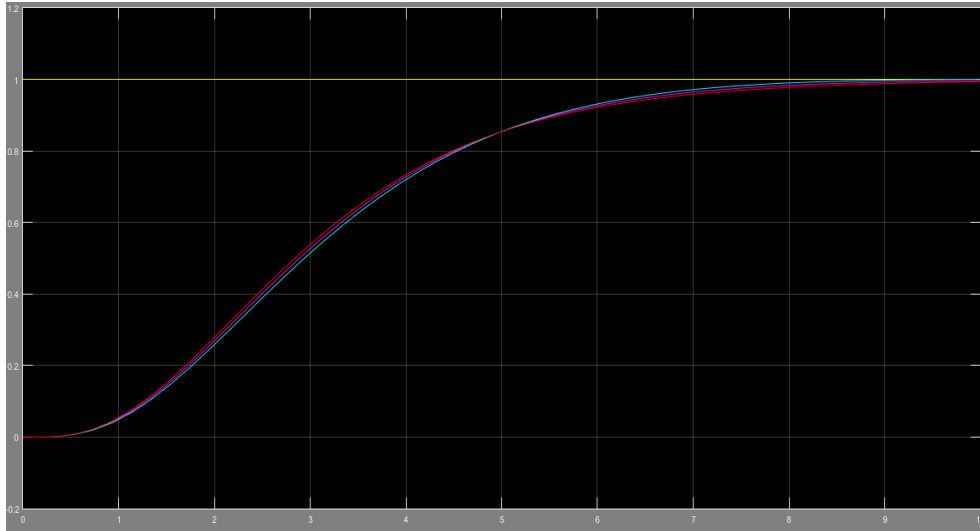


Figure 15: b variation responses

We can perceive some slightly differences between the three graphics, one for each b value (105%, 100% and 95%). In fact, this is a clearly example of an under, over and critically damped responses, as we can see in the zoom that we made from figure 15 in the figure 16, where we can see one of the responses above the reference (the entry).

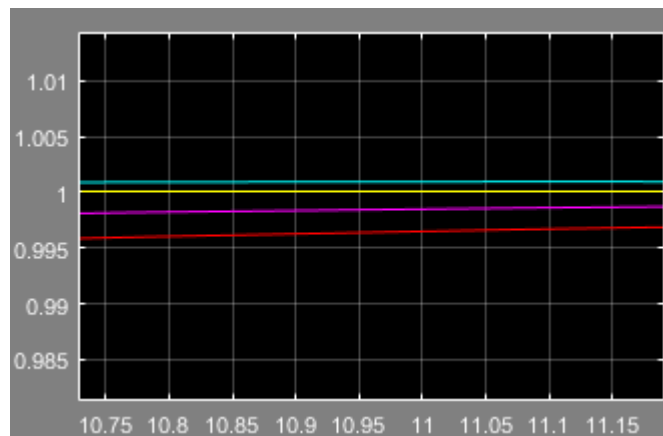


Figure 16: b variation responses zoom

The reason of this behavior is that our feedback element (the controller K):

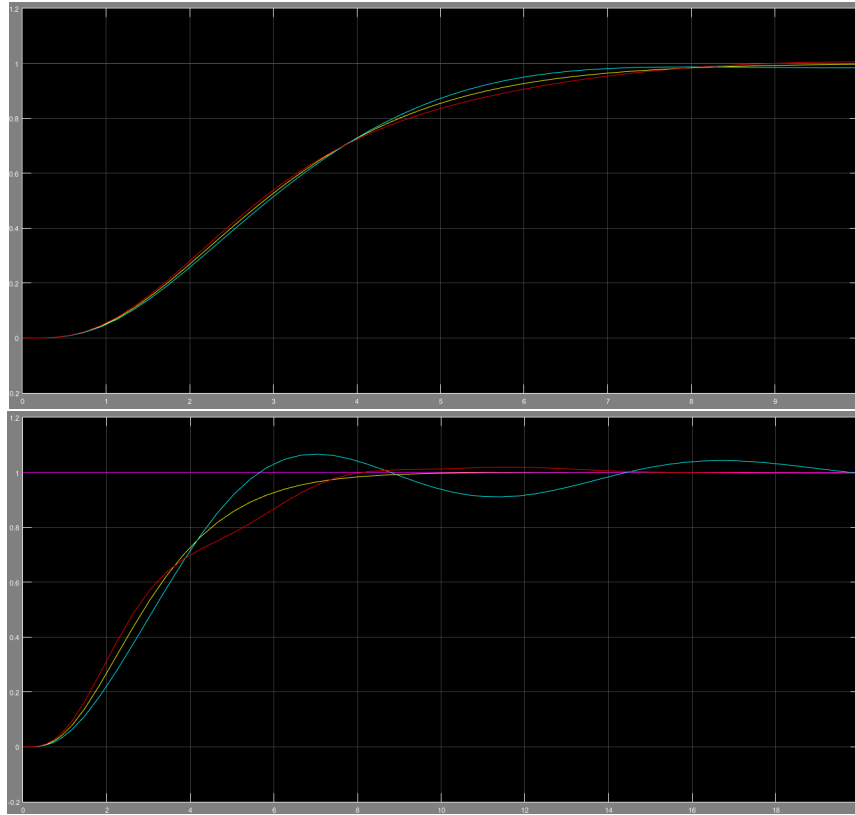


Figure 17: b variation responses ($\pm 5\%$ and $\pm 20\%$ respectively) for system with observer

subsection 3.1) has been calculated with the original A matrix (equation 4), composed with the values of the original b .

However, this changes in the ball's characteristics are not going to disturb the systems response in any important way. In any case, it will reach the final value a bit slower.

This b modification when applied to the system with the observer, figure 13, results in a more dramatical divergation from the original, and if the ball's characteristics differ more than $\pm 20\%$ with its original values, we can have serious problems to accomplish a decent time of stability, or even the system's stability, as we can see in figure 17.

4.2 Noise's effect

The other modification we have made is about the influence of noise in the system. As in the previous robustness test, we have made this test to the

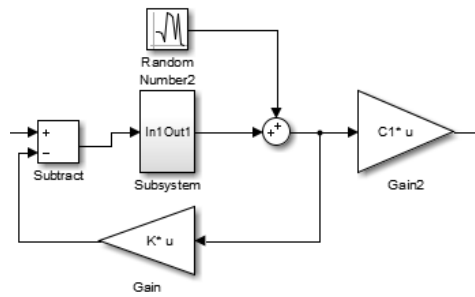


Figure 18: Location of noise's block

controlled system with and without observer.

We have added one noise block in the plant's output, before the feedback, in order to simulate the possible noises in the input system of the sensors, or even noises in their signal treatment.

We have made two different tests, the first one for a noise of 0.1% and the second one of 0.01% of the entry state value.

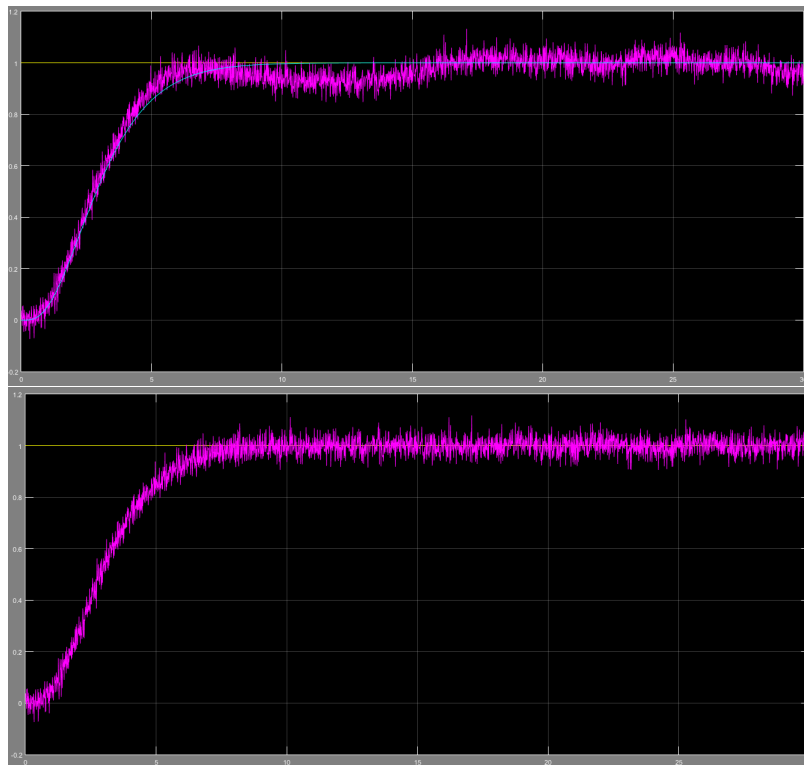


Figure 19: noise's response (0.1%), without and with observer, respectively

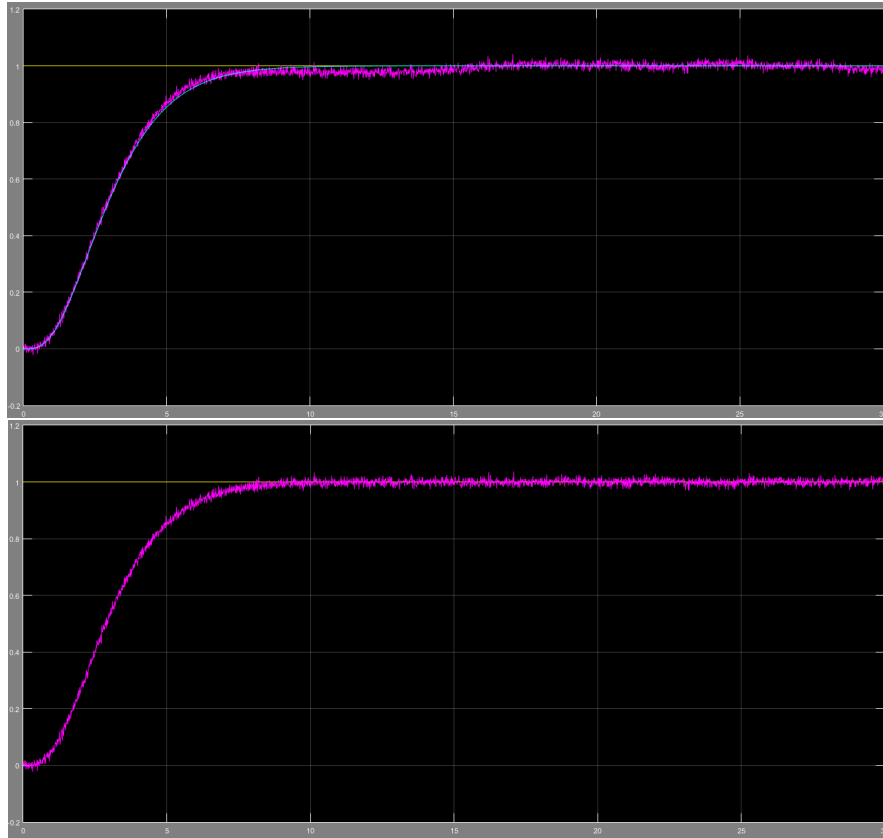


Figure 20: noise's response (0.01%), without and with observer, respectively

The results are, unlike before, better in the system with observer than without it. We can see in figures 19 and 20.

Briefly, we can say that depending on the type of robustness desired we can make a system more or less prepared for each situation. For example, if we know in advance that the noises are going to be depreciable for the system, but we are going to use balls of different mass and radius, we prefer the system without observer.

In the other hand, if we will use always the same ball, we will use the observer in order to minimize the noise's (and others) effects.

Part II

Takagi-Sugeno Model Design

5 Takagi-Sugeno Fuzzy logic

Now we are going to study the fuzzy logic's TS type and its direct application to our system. We will need for doing so, first of all, to remodel our system into a fuzzy model, and then to make a controller. After these we will simulate it and compare the results.

5.1 Fuzzy logic modeling

First of all, we are going to build a fuzzy model for our system. Nevertheless, we are not starting from zero. We will use what we have already done in subsection 2.2 until equation (3).

As we have done before, we will consider that the motion of the ball is independent of the axis, so we get the next equation (we will continue working only in x and α axis):

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} x_2 \\ b \cdot (x_1 x_4^2 - g \cdot \sin x_3) \\ x_4 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} u_x$$

Or expressed in the way $\dot{x} = A \cdot x + B \cdot u$:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ b \cdot x_4^2 & 0 & -b \cdot g \cdot \frac{\sin x_3}{x_3} & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} x + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} u_x \quad (9)$$

Once we get here, we start the fuzzy modeling. Our goal is to derive the TS fuzzy model from this system. This fuzzy model corresponds to a model that follows some rules. Our rules will follow the same structure than the next example:

$$\text{IF } z_1 \text{ IS } P_i \text{ AND } z_2 \text{ IS } Q_j \text{ THEN } A(z_1, z_2) \text{ IS } A_k \quad (10)$$

Where z_1 and z_2 are fuzzy variables, P and Q fuzzy sets, and A the matrix that we already know.

We will design the fuzzy model by operating with the nonlinear terms in the equation (9). In this case, these terms are x_3 and x_4 , or in other words, angle position and velocity. First we will need the variables limits. Measured in radians and radians/second, we obtained the following limits: $x_3 \in [-0.175, 0.175]$ and $x_4 \in [-2.25, 2.25]$.

Now we are going to introduce our fuzzy variables z_1 and z_2 , that correspond to $z_1 = b \cdot x_4^2$ and $z_2 = -b \cdot g \cdot \frac{\sin x_3}{x_3}$. So now the matrix A from equation 9 can we written as:

$$\dot{A}(z) = \begin{bmatrix} 0 & 1 & 0 & 0 \\ z_1 & 0 & z_2 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (11)$$

The next steps should be to calculate the maximums and the minimums of this fuzzy variables derivating its equations with respect to x_3 and x_4 limits. In our system we obtained the following:

$$\begin{aligned} \max z_1 &= 3.616; & \min z_1 &= 0; \\ \max z_2 &= -6.964; & \min z_2 &= -7. \end{aligned}$$

Now we can represent z_1 and z_2 using the membership functions $M_1(z_1)$, $M_2(z_1)$, $N_1(z_2)$ and $N_2(z_2)$ as follows:

$$z_1 = b \cdot x_4^2 = M_1 \cdot 3.616 + M_2 \cdot 0,$$

$$z_2 = -b \cdot g \cdot \frac{\sin x_3}{x_3} = N_1 \cdot (-6.964) + N_2 \cdot (-7).$$

where, for z_1

$$M_1(z_1) = \frac{z_1(x_4) - \min z_1}{\max z_1 - \min z_1},$$

$$M_2(z_1) = \frac{\max z_1 - z_1(x_4)}{\max z_1 - \min z_1}$$

and

$$M_1(z_1) + M_2(z_1) = 1,$$

with their equivalents for z_2 : $N_1(z_2)$ and $N_2(z_2)$. We will call the fuzzy sets “Big”, “Small”, “Fast” and “Slow” referring to the angle’s amplitude and speed.

Now we can determine the whole nonlinear system using the following rules:

Rule 1: IF z_1 is “Big” and z_2 is “Fast” THEN $\dot{x} = A_1 \cdot x + B \cdot u$;

Rule 2: IF z_1 is “Big” and z_2 is “Slow” THEN $\dot{x} = A_2 \cdot x + B \cdot u$;

Rule 3: IF z_1 is “Small” and z_2 is “Fast” THEN $\dot{x} = A_3 \cdot x + B \cdot u$;

Rule 4: IF z_1 is “Small” and z_2 is “Slow” THEN $\dot{x} = A_4 \cdot x + B \cdot u$;

where these new subsystems have the following A matrix in the form of matrix (11):

$$\begin{aligned}
 A_1 &= \begin{bmatrix} 0 & 1 & 0 & 0 \\ 3.616 & 0 & -6.964 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} & A_2 &= \begin{bmatrix} 0 & 1 & 0 & 0 \\ 3.616 & 0 & -7 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \\
 A_3 &= \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & -6.964 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} & A_4 &= \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & -7 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}
 \end{aligned} \tag{12}$$

Finally, we have to deal with a defuzzification process in order to have a linear \dot{x} answer and the possibility of using this system in non fuzzy enviroments. This \dot{x} will be derived from the following equation:

$$\dot{x} = h_1(z) \cdot A_1 \cdot x + h_2(z) \cdot A_2 \cdot x + h_3(z) \cdot A_3 \cdot x + h_4(z) \cdot A_4 \cdot x + B \cdot u \tag{13}$$

where

$$\begin{aligned}
 h_1(z) &= M_1(z_1) \cdot N_1(z_2), \\
 h_2(z) &= M_1(z_1) \cdot N_2(z_2), \\
 h_3(z) &= M_2(z_1) \cdot N_1(z_2), \\
 h_4(z) &= M_2(z_1) \cdot N_2(z_2).
 \end{aligned} \tag{14}$$

Once we have calculated the A matrix from equation (11), which results we can see for our obtained limits in (12), and once we have derived h_1 , h_2 , h_3 and h_4 from (14), we get our final equation system, shown in equation (13).

6 Simulation’s comparison between state-space model and TS fuzzy model

In this section we will deal with the simulation of this previous fuzzy controller by simulink in order to see how it works and if its outputs are similar to the outputs obtained by the state-space model design.

Starting with the code already used, in figure 3, we have written the following in figure 21, that has all the necessary to build our fuzzy system in symulink.

```

b=0.027/(0.027+4.32e-6/0.02^2);
g=9.81;

Mz=b*2.25^2;           %maximum z1
mz=0;                  %minimum z1
Nz=-b*g*sin(0.175)/0.175; %maximum z2
nz=-7;                 %minimum z2

Aalpha=[0 1 0 0; 0 0 -b*g 0; 0 0 0 1; 0 0 0 0];
Balpha=[0 0 0 1]';
Calpha=eye(4);
Dalpna=zeros(4,1);
|
A1=[0 1 0 0; Mz 0 Nz 0; 0 0 0 1; 0 0 0 0];
A2=[0 1 0 0; Mz 0 nz 0; 0 0 0 1; 0 0 0 0];
A3=[0 1 0 0; mz 0 Nz 0; 0 0 0 1; 0 0 0 0];
A4=[0 1 0 0; mz 0 nz 0; 0 0 0 1; 0 0 0 0];

```

Figure 21: Model's code in matlab

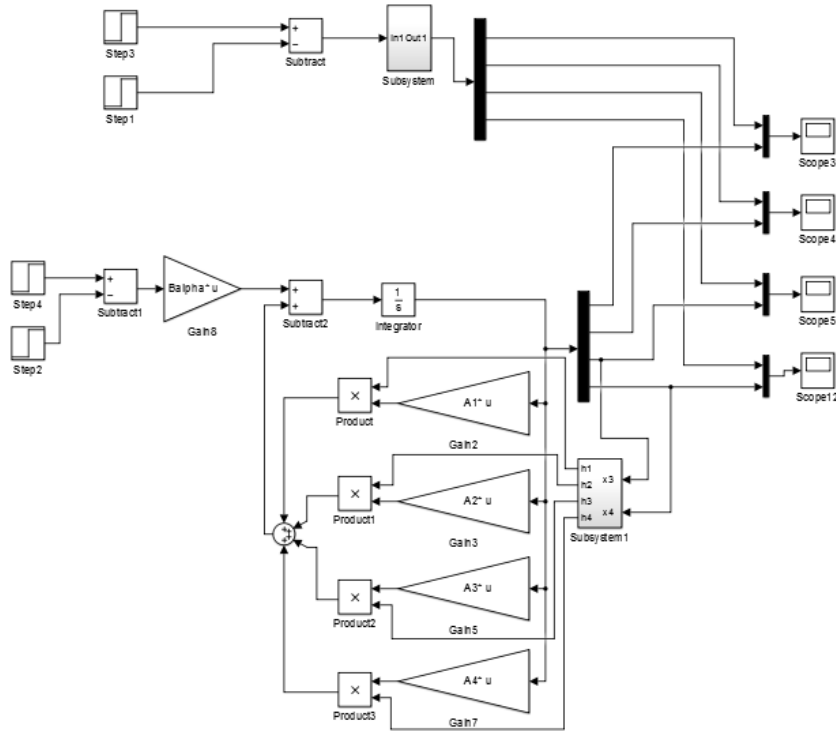


Figure 22: Comparison between state-space and fuzzy models

In the following figures 22 and 23, we have designed the model in symulink enviroment, we made a subsystem for the calculus of our variables h due to their

more complex configuration.

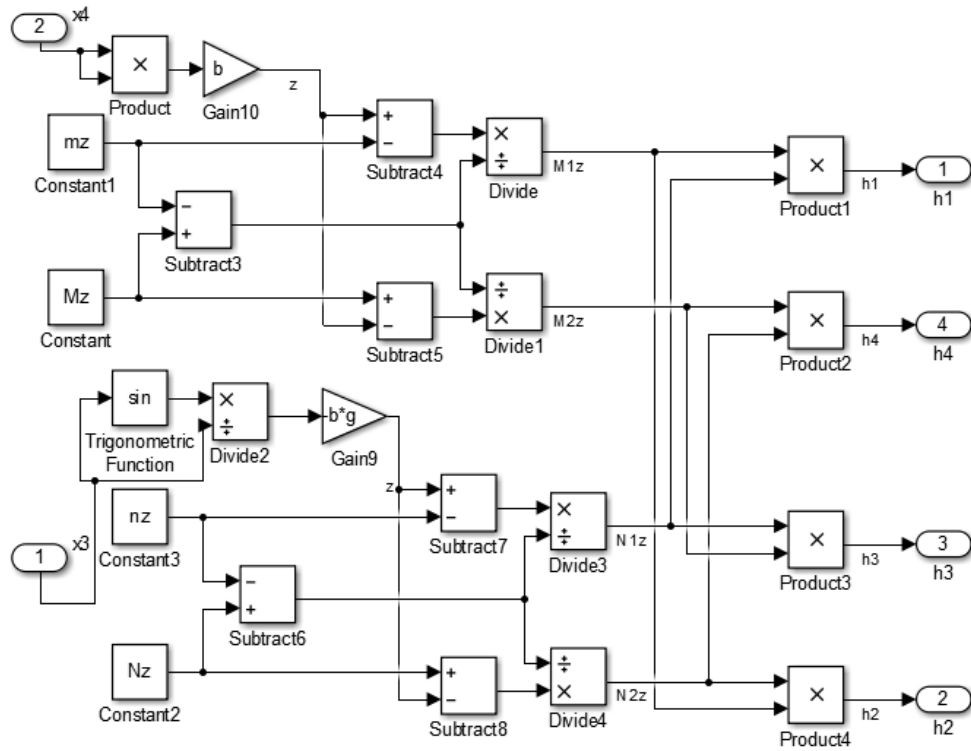


Figure 23: h calculus subsystem

The comparison makes no proper differences between these two systems, and the results are the same for angle's position and velocity, and almost the same for the ball's position and velocity. As we can see in the figure 24.

Actually, we can also add to our new system the K controller derived in subsection 3.1. This is not any fuzzy controller that we would build for the fuzzy model, but it is possible to use it in order to make the previous comparison with a controller.

We are adding only the answer of the ball's position, showed in figure 25. As we can see, now there is no difference between this two outputs

In conclusion, we can say that our fuzzy model has practically the same behaviour than the state-space model. Which gives us certainty that our fuzzy model is well built.

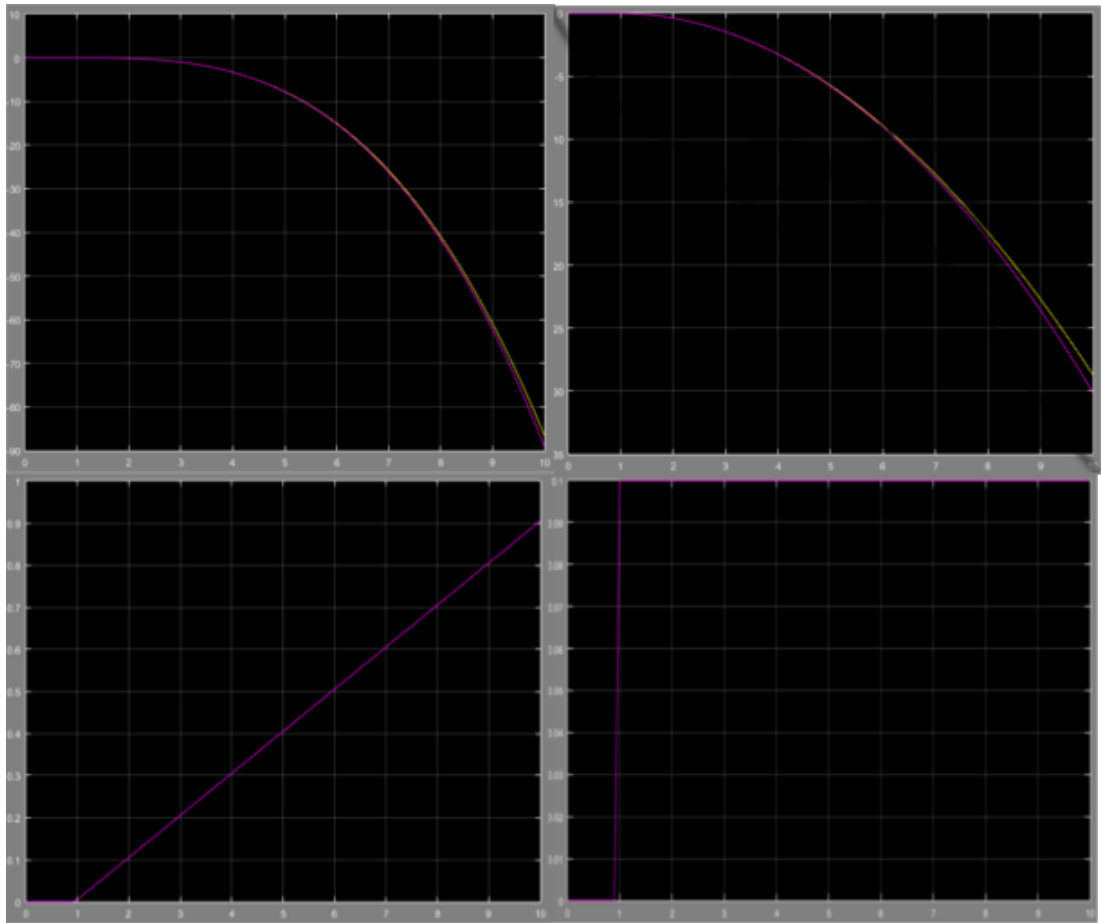


Figure 24: x, \dot{x}, α and $\dot{\alpha}$ respectively

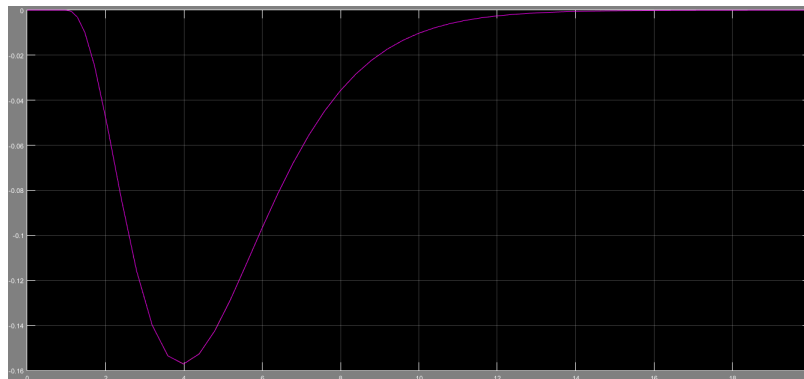


Figure 25: comparison with K

References

- [1] Feedback Instruments Ltd. “Ball and Plate Control Experiments 33-240-LabView”, 2013
- [2] Abdelouhab Aitouche “Observation and Control of Space State Systems”
- [3] Mohammad Nokhbeh, Daniel Khashabi and Dr. H.A. Talebi “Modelling And Control Of Ball-Plate System” Final Project Report from Amirkabir University of Technology, 2011.
- [4] Kamyar Mehran “Takagi-Sugeno Fuzzy Modeling for Process Control” Industrial Automotion, Robotics and Artificial Intelligence (EEE8005), Newcastle University, 2008.
- [5] Katsuhiko Ogata “Modern Control Engineering”, 1970
- [6] P. de Otaola Arca and Dr. Zamora Macho “Construcción y Diseño del Sistema de Control de Posición de una Esfera Sobre una Plataforma Movil”, Universidad Pontificia ICAI ICADE Comillas, Madrid, 2015
- [7] A. Jadlovská, S. Jajcisin and R. Lonseak “Modelling and PID Control Of Nonlinear Educational Model Ball & Plate”, 17th International Conference on Process Control, Slovakia, 2009
- [8] Sören Georg, Matthias Müller and Horst Schulte “Wind Turbine Model and Observer in Takagi-Sugeno Model Structure” The Science of Making Torque, Wind 2012, Oldenburg, 2012
- [9] M. Ali Rastin, S. Ali A. Moosavian, Erfan Talebzadeh and Mojtaba Alaeddin “Trajectory Tracking And Obstaacle Avoidance Of A Ball And Plate System Using Fuzzy Logic”, 13th Iranian Conference on fuzzy systems, 2013.