Controller comparison and mathematical modelling of ball and beam system

Erick P. Herrera-Granda[1], Israel D. Herrera-Granda[2], Alexis X. Vaca-Espinoza[1], Jonathan G. Loor-Bautista[3], Edgar F. Imbaquingo-Esparza[1]
[1]Escuela de Doctorado en Ingeniería y Producción Industrial,
Universidad Politécnica de Valencia, Edificio 2E. Camino de Vera, s/n, 46022, España
[2]Facultad de Ingeniería en Ciencias Aplicadas, Universidad Técnica del Norte.,
Av. 17 de Julio 5-21 y Gral. José María Córdova, Ibarra, Ecuador
[3]Carreras de Profesionalización EB y EIB, Universidad Nacional de Educación
Barrio Nueva Coca, sector Mushuk Ayllu, vía Loreto Km 1, Orellana, Ecuador

[1]{epherrera, axvacae, efimbaquingoe}@utn.edu.ec, [2]ishergra@doctor.upv.es, [3]jonathan.loor@unae.edu.ec

ABSTRACT.

The present project shows a comparison between three control techniques applied in the ball and beam system. In the ball and beam system problem, a bar is rotated using a motor and this rotation makes a sphere roll over the bar until it reaches the desired position. The mathematical modeling was done writing the dynamic equations on the state space, and the non-linear state equations were represented as a plant system using Simulink. The applied control techniques were state-feedback controller, linear quadratic regulator (LQR) and neural network based NARMA controller. The three techniques were applied for continuous and discrete signals and were tested with and without state observer. Several simulations were carried out using Matlab and Symulink, and results shown that the system can be stabilized by all the controllers with slight differences, state-feedback was faster and NARMA model was smoother and required a smaller input.

Keywords: State-feedback controller, LQR, NARMA, neural network controller, state space.

Introduction

In the present study, a bar and sphere control system is analyzed to explain the most important aspects of control systems. The bar and sphere system is composed of a bar or beam articulated by a motor, and on it is a sphere that can slide freely. The purpose of this control system is to keep the sphere on the bar in the desired position, by means of control techniques that allow the bar to rotate properly and position the sphere in the desired location.

The configuration in which the present project was carried out is presented in Figure 1, where it is observed that the beam is articulated in its central axis, the advantage of that control system is its ease of construction and its mathematical modelling is simple.

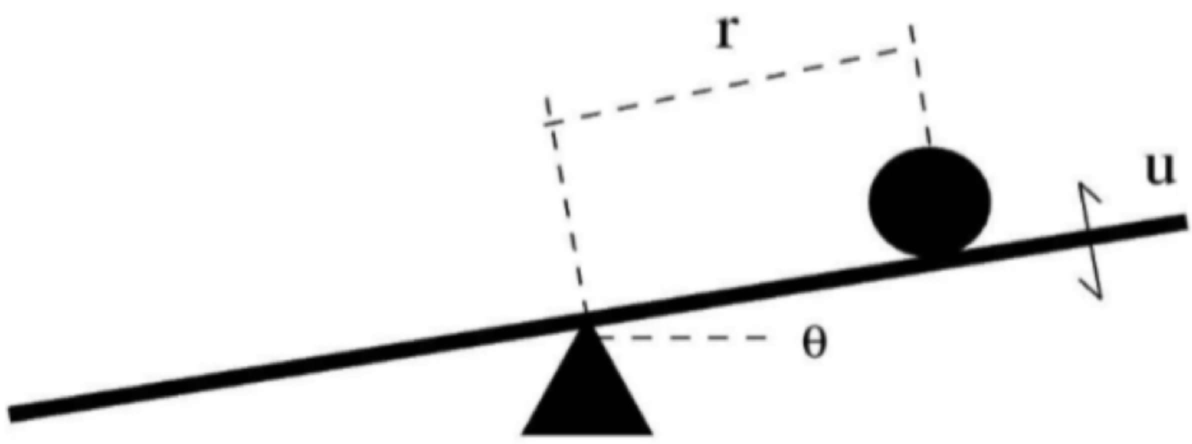

Fig. 1. Ball and beam system and its main variables

This bar and sphere system is considered a subactuated system, because it has one degree of freedom and one single actuator. For the first prototype a study of the linearization and modeling processes in Matlab was carried out, the first objective was to obtain the space state from dynamic equations, then perform a linearization of the state equations and finally model the control system using Matlab.

The controller design techniques employed in this work are state-feedback controller (applying Ackermann's formula), optimal control linear quadratic regulator (LQR) and nonlinear autoregressive mobile average neural network controller (NARMA). These techniques were selected, due to their capacity to control nonlinear systems and the excellent obtained results, achieving stabilization in less than four seconds in all the simulations that were carried out.

The work presented in [6] related to design, construction and modeling of a "Ball and Beam" by Marco Mora Reyes has been taken as a study reference, because it presents a good explanation of mathematical modeling by the Euler-Lagrange method. Also, the work presented in [5] developed by Gregory Cardenas M. was used as a source of information, necessary for the development of the state space and the control system.

Related Works

In the work of [1] the didactic problem of the ball and beam or ball and bar system (B&B) is addressed as a class exaple, which consists in balancing a steel ball on a bar of the same material by means of a single actuator that controls the degrees of freedom of said ball, thereby positioning it on a position determined by the user of the controller. After a modeling process of the physical system, the main functions of the system were obtained, in order to perform a simulation using Simulink. Then the most stable system was determined and a prototype was constructed and implanted in real life, positively closing the learning cycle in the classroom. On the other hand in [2], a prototype for the control of the B&B was developed, by means of a double loop control system and magnetic actuators.

Another prototype for the control of the B&B is detailed in [3], where the control of the position of the ball on the bar is tried by means of actuators that turn the inclination of the bar, and the position and rotation of the ball is monitored in order to feed back the system. Additionally, in [4] the developing of a "Ball and beam balancer" prototype is shown. The prototype used a resistive wire position sensor, a DC servo motor and an Arduino Leonardo microcontroller. Authors used a PID feedback controller algorithm, which depends on the feedback signal provided by a linear potentiometer position sensor, and it can be designed by non-model based methods.

Methodology

This project presents the development and design of the controller to be implemented in the Ball and Beam control system. Initially an analysis of the model was done, in order to get the non-linear functions that represent the plant to be controlled. Then the equations were rewritten as a state space and the resulting system was linearized by Taylor's approximation.

Afterwards, the obtained mathematical model was represented in Matlab using a script for gains and parameters calculation, and Simulink for plant modeling and controller design. Next, simulations on a virtual environment were carried out, using Matlab's V-Realm editor. Finally, a demonstration system's controllability is presented through their respective time response stabilization graphs.

Dynamical modelling

The first part of this project is dedicated to the modeling of the dynamical equations of the ball and beam system. This system consists of a bar which swings a sphere to hold it in the desired position [8]. In this way two different bodies interact, the bar swings and controls to angle and the sphere slides through the bar. In order to control this system, forces in figure 2 were analyzed.

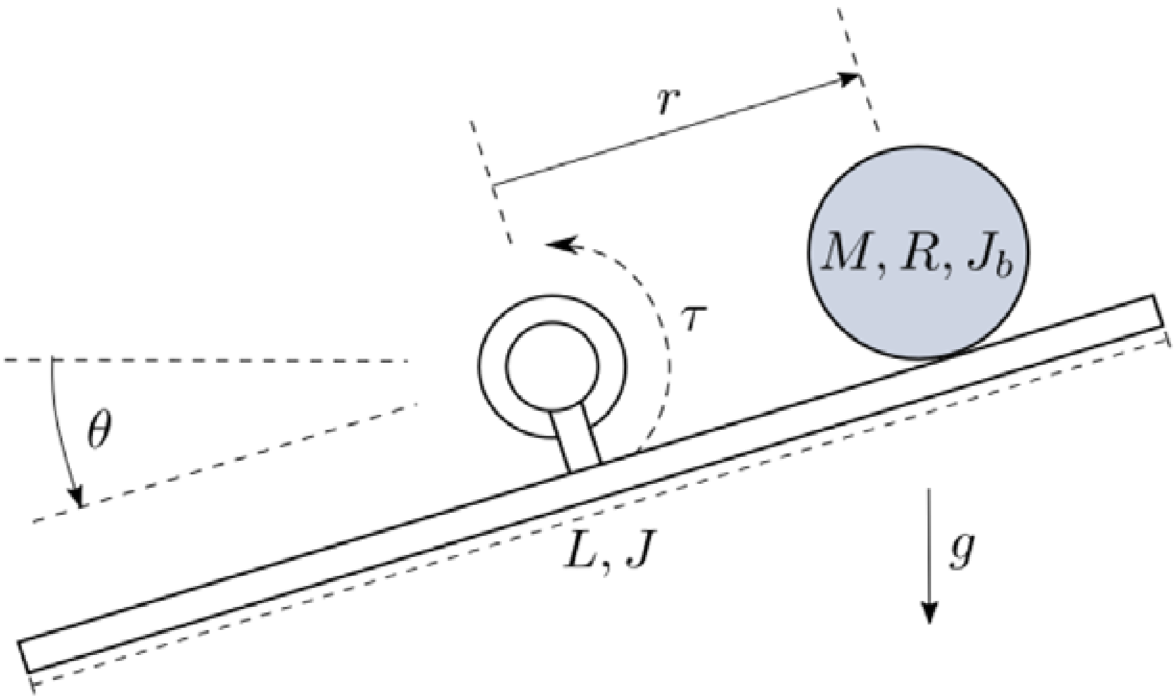

Fig. 2. Ball and beam system, variables and parameters

Where:

ϒ(*t*): Sphere Position

θ: Rail Angle

τ: Applied Torque

g: Gravity acceleration

J: Moment of inertia of the rail

Jb: Moment of inertia of the ball

m: Mass of the Sphere

R: Sphere radius

To start modeling the bar-sphere system, the free-body diagram of Figure 2 is established in which a polar coordinate system is established with respect to the central axis of the bar, considering that the sphere can move freely over the bar. The sphere must remain in contact with the bar without slipping. Defining θ as the rotation angle of the bar respect to the horizontal, and r the position of the sphere, the following L-equations of movement are obtained [4][9]:

$$0 = \left(\frac{J_b}{R^2} + M\right)\ddot{r} + Mg\,sin\,\theta - Mg\,sin\,\theta - Mr\dot{\theta}^2 \quad (1)$$

$$\tau = (Mr^2 + J + J_b)\ddot{\theta} + 2Mr\dot{r}\dot{\theta} + \; Mgr\,cos\theta \quad (2)$$

Where τ is the momentum or torque applied to the bar; *J* is the inertia momentum of the bar; M and J_b are the mass and the moment of inertia of the sphere respectively ; R is the sphere radius and *g* is the acceleration of gravity. Model (1) can be simplified if the moment τ is replaced, making θ =u.

Space state modeling was used, where the state variables are ϒ, θ and their respective derivatives. Next, the state variables are:

$$x_1 \;=\; r \quad (3)$$

$$x_2 \;=\; \dot{r} \quad (4)$$

$$x_3 \;=\; \theta \quad (5)$$

$$x_4 \;=\; \dot{\theta} \quad (6)$$

It is known, that the applied torque τ is the manipulated system variable and r is the position of the sphere and was taken as the output of the system, so it turns out that:

$$u \;=\; \tau \quad (7)$$

$$y \;=\; r \quad (8)$$

To linearize the system without neglecting values, linearization by Taylor's approximation was used. Where the non-linear system equations in state space results, in [10][11]:

$$\dot{x}_1 \;=\; x_2 \quad (9)$$

$$\dot{x}_2 = b(x_1{x_4}^2 - g\;sin(x_3)) \quad (10)$$

$$\dot{x}_3 = x_1 \quad (11)$$

$$\dot{x}_4 = \frac{-2mx_1x_2x_4 - mgx_1\cos\,(x_3) + u}{m{x_1}^2 + J + J_b} \quad (12)$$

Where:

$$b = \frac{m}{\frac{J_b}{r^2} + m} \quad (13)$$

Linearization

The system is written in an state space array of the form:

$$f(x,u) = \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \end{bmatrix} = \begin{bmatrix} x_2 \\ b(x_1 {x_4}^2 - g\, sin(x_3)) \\ x_4 \\ \dfrac{-2mx_1x_2x_4 - mgx_1cos(x_3) + u}{m{x_1}^2 + J + Jb} \end{bmatrix} \tag{14}$$

Once the state equations f_i were obtained, the equilibrium point x_q was determined for taking each state derivative as zero x=0, and an input u_q= 0. Replacing all the values the state space matrix results in four equilibrium equations:

$$x_{2q} = 0 \tag{15}$$

$$b(x_{1q}{x_{4q}}^2 - g\, sin(x_{3q})) = 0 \tag{16}$$

$$x_{4q} = 0 \tag{17}$$

$$\frac{-2mx_{1q}x_{2q}x_{4q} - mgx_{1q}cos(x_{3q}) + u_q}{m{x_{1q}}^2 + J + Jb} = 0 \tag{18}$$

Replacing equations (15) and (17), in equations (16) and (18) results in:

$$\dot{x} = \begin{bmatrix} x_{1q} \\ x_{2q} \\ x_{3q} \\ x_{4q} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Then, the state matrices of the bar-sphere system are obtained. For the state space of (19), A matrix is defined as follows:

$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx + Du \end{aligned} \tag{19}$$

$$A = \left.\frac{\partial f_i}{\partial x_j}\right|_{\substack{x=x_q \\ u=u_q}} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \frac{\partial f_1}{\partial x_3} & \frac{\partial f_1}{\partial x_4} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \frac{\partial f_2}{\partial x_3} & \frac{\partial f_2}{\partial x_4} \\ \frac{\partial f_3}{\partial x_1} & \frac{\partial f_3}{\partial x_2} & \frac{\partial f_3}{\partial x_3} & \frac{\partial f_3}{\partial x_4} \\ \frac{\partial f_4}{\partial x_1} & \frac{\partial f_4}{\partial x_2} & \frac{\partial f_4}{\partial x_3} & \frac{\partial f_4}{\partial x_4} \end{bmatrix} \tag{20}$$

$$\begin{bmatrix} 0 & 1 & 0 & 0 \\ {x_{4q}}^2 & 0 & -gcosx3 & 2x_{1q}x_{4q} \\ 0 & 0 & 0 & 1 \\ \frac{-2mx_{1q}x_{2q}x_{4q} - mgx_{1q}cosx_{3q} + U_q}{m{x_1}^2 + J + J_b} & \frac{2mx_{1q}x_{2q}x_{4q}}{J + J_b} & \frac{mgx_{1q}\sin(x_{3q})}{J + J_b} & -\frac{2mx_{1q}x_{2q}x_{2q}}{J + J_b} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & -bg & 0 \\ 0 & 0 & 0 & 1 \\ \frac{mg}{J + J_b} & 0 & 0 & 0 \end{bmatrix} \quad (21)$$

Matrices B and C are as follows:

$$B = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \frac{1}{J + J_b} \end{bmatrix} \quad (22)$$

$$C = [1 \quad 0 \quad 0 \quad 0] \quad (23)$$

The values for the system constants are:

- Mass of the Sphere: 0,1 [*Kg*]
- Radius of the Sphere: 0,1 [*m]*
- Moment of Inertia of the Sphere: 0,004 [$kg\ m^2$]
- Moment of Inertia of the Bar: 0,02083 [$kg\ m^2$]
- Acceleration of Gravity: 9,8 [m/s^2]

By entering these values in Matlab, the matrices of the linearized bar-sphere system are obtained:

$$\begin{bmatrix} \dot{x}1 \\ \dot{x}2 \\ \dot{x}3 \\ \dot{x}4 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & -1.96 & 0 \\ -39,47 & 0 & 0 & 1 \\ & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x1 \\ x2 \\ x3 \\ x4 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 40.73 \end{bmatrix} u \quad (24)$$

$$y = [1 \quad 0 \quad 0 \quad 0]x \quad (25)$$

Controller design and simulation

In Matlab's software constants, k_1, k_2, k_3, k_4 gains, and the respective matrices A, B, Cand D, are replaced to obtain the simulation graphs.

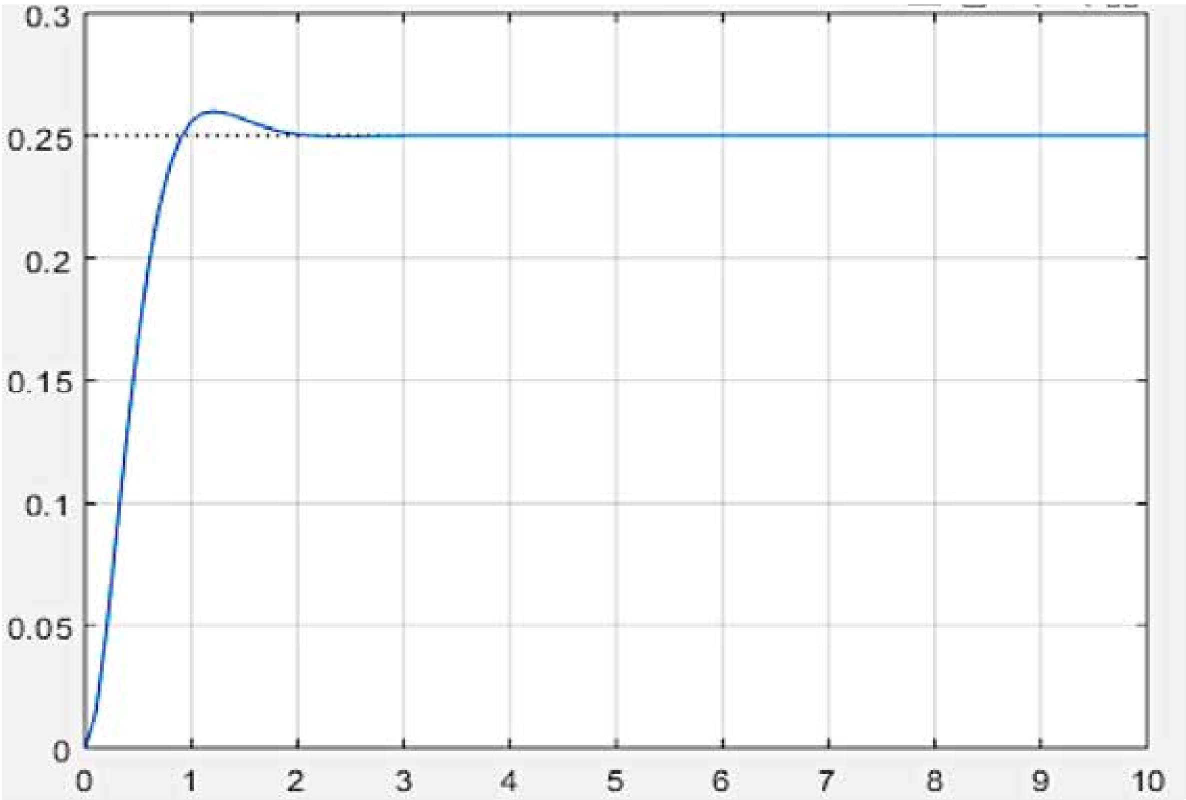

Fig. 3. System response bar-sphere with τ = 0 and initial conditions at zero and unaltered variables.

Figure 5 shows the response of the system with an initial condition θ = 0.1 [rad] and with an input torque of zero τ = 0. For this graph a time window of three seconds was taken and, by modifying the initial condition of the initial angle of the 0.1 rad, it is observed that the system becomes unstable, so the sphere tends to fall from the bar.

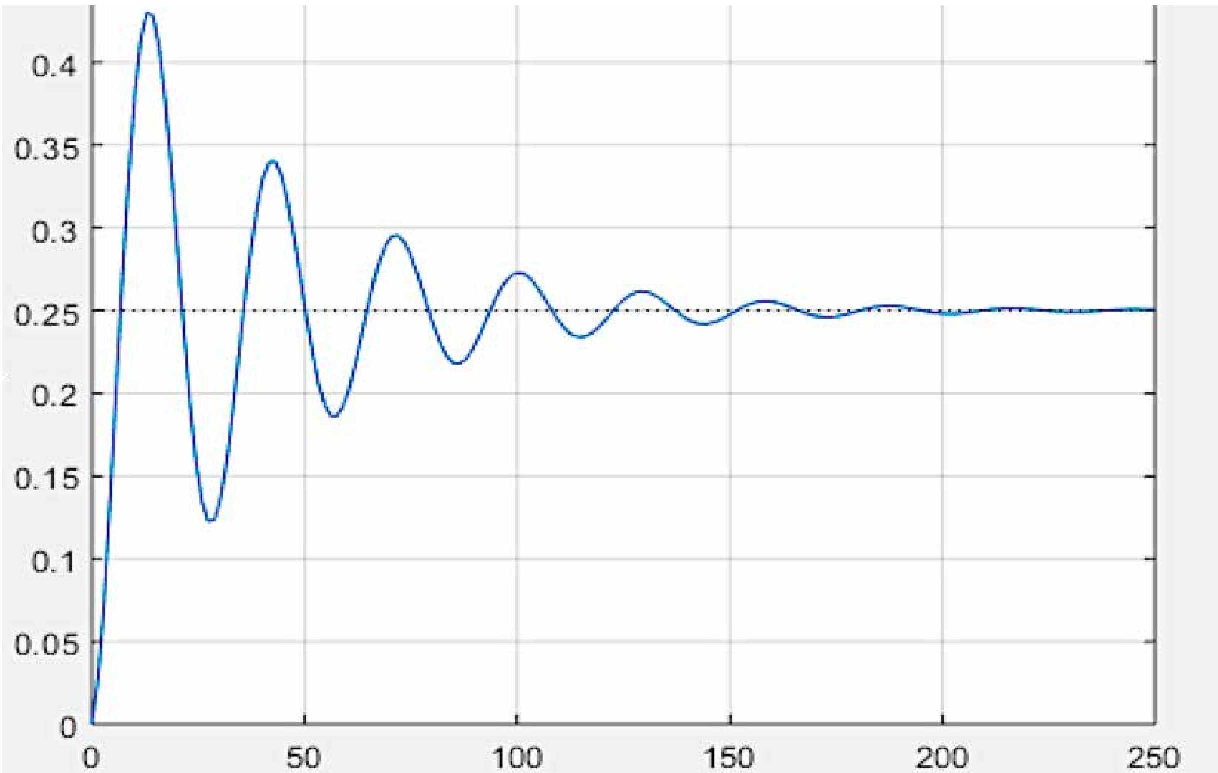

Fig. 4. System response bar-sphere with an initial conditionθ=0.1 [rad] and an input torque equal to zero τ=0

Simulation is carried out in Simulink, for which the parameters were defined as follows:

$$m = 0.111,\ R = 0.015,\ g = -9.8,\ L = 1.0,\ d = 0.03,\ J = 9.99e^{-6},\ H = -\frac{m\,g}{\frac{J}{R^2}+m},\ kp = 0.1,\ kd = 0.1.$$

Afterwards ball and beam system was created using Matlab ss function [7]. The controllability and observability matrixes were obtained, and both of theme were full rank, which suggests that a controller and an observer can be implemented. Applied code is detailed below.

```
sys=ss(A,B,C,D)
pole (sys)
Sc=ctrb(sys)
So=obsv(sys)
rank(So)
rlocus(sys)
```

To see the ball and beam system running in Matlab, a Simulink 3D animation is created, which is a VR SINK block that designs the bar and sphere control system

In Matlab v-realm building, for the rotation and translation of the system two blots of VR Signal Expander were created as shown in Figure 9, 10 and 11 [7].

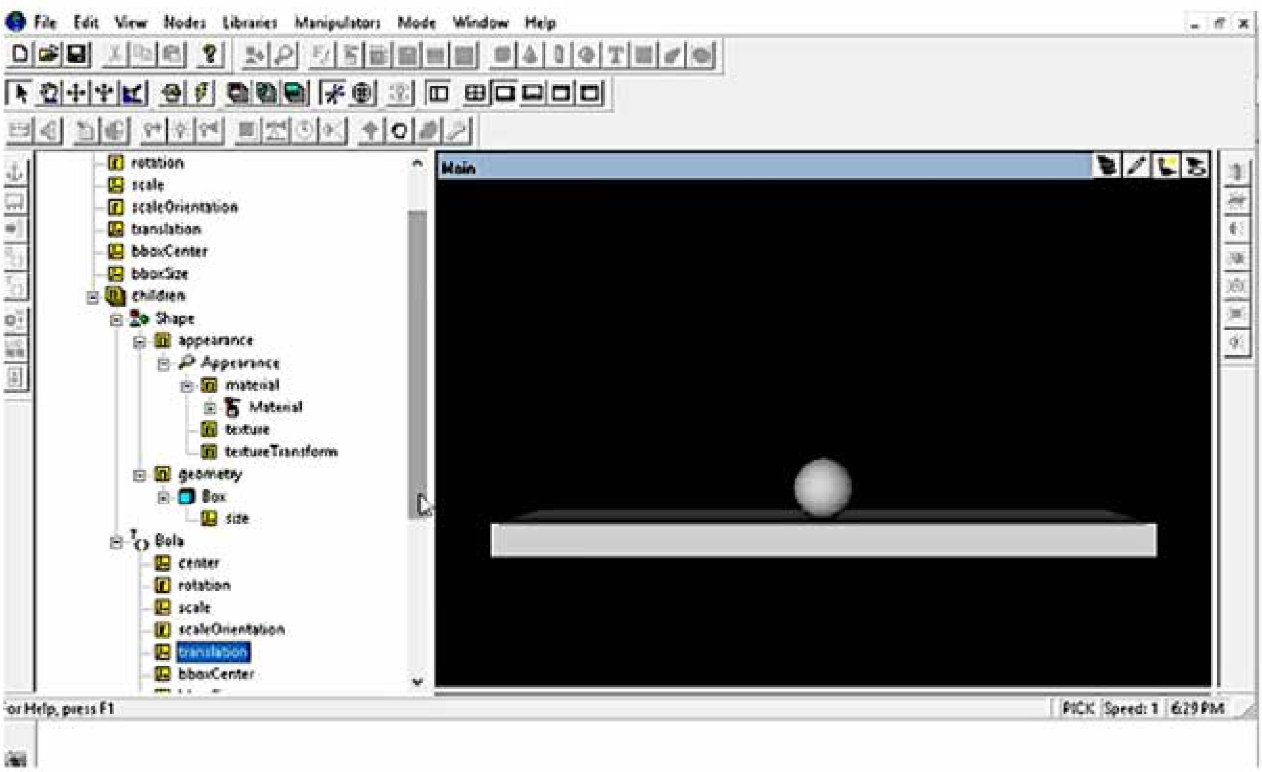

Fig. 5. V-Realm building for the simulation using Matlab

State-feedback Controller. As shown in figures 6 to 8 the system is stabilized in less than 2 seconds, this stabilization is achieved by means of the gains obtained in the previous section.

$$u(t) = -Kx(t) + r(t) \tag{26}$$

The plant is modeled by means equations of state equations:

$$\dot{x}(t)=Ax(t)+Bu(t) \quad (27)$$

$$y(t)=Cx(t) \quad (28)$$

Substituting in equation (27) and the given value of u(t).

$$\dot{x}(t)=Ax(t)+B[-Kx(t)+r(t)] \quad (29)$$

The Simulink model developed for discrete time is presented in figure 6, the required input through the time is shown in figure 7 and the time response of the four states is presented in figure 8.

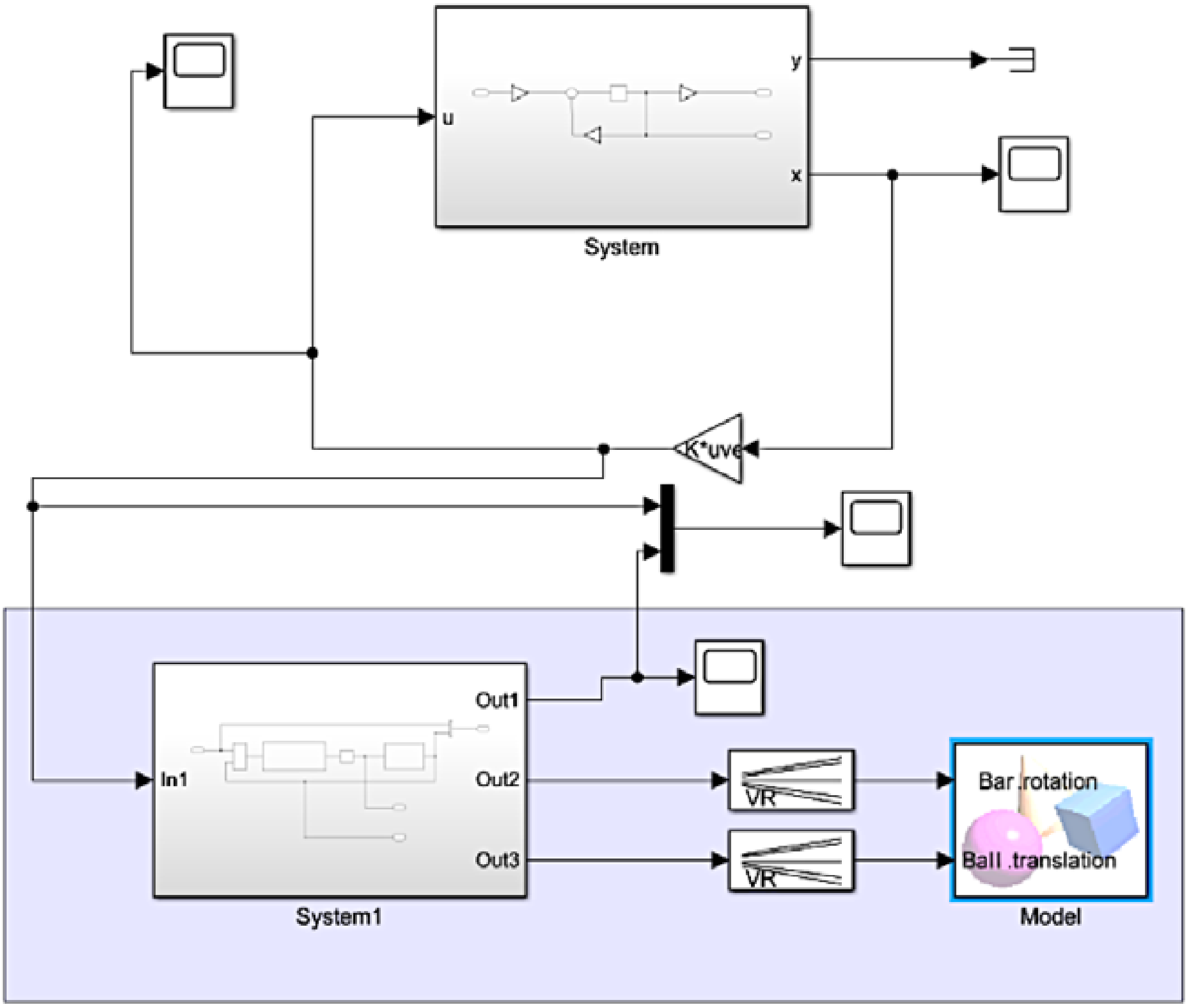

Fig. 6. Simulink model for state-feedback controller

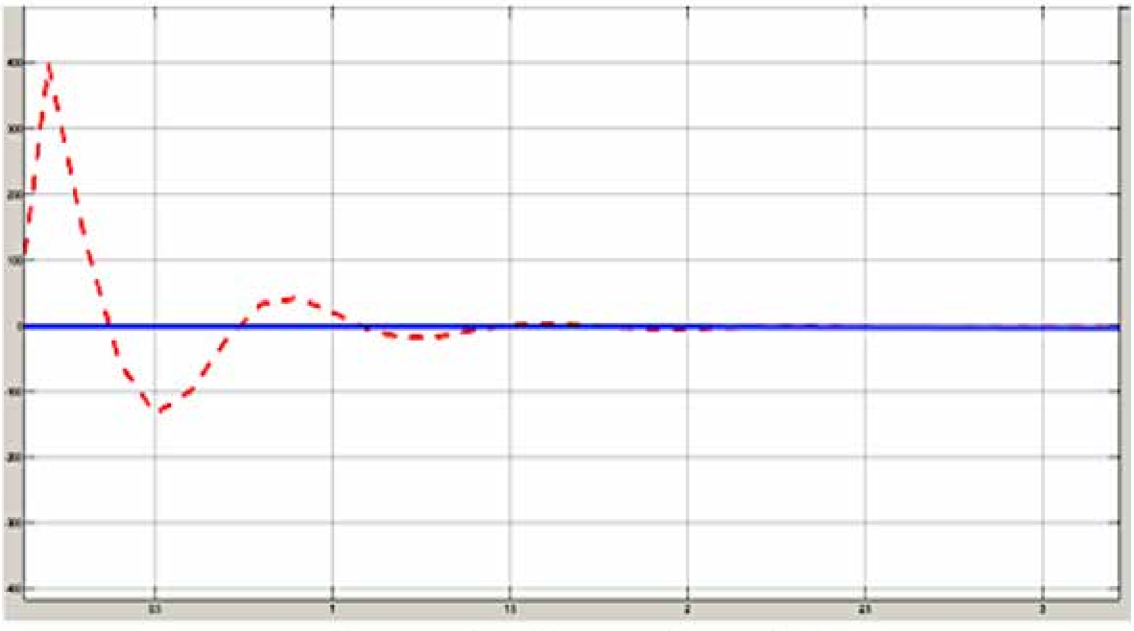

Fig. 7. Required input through time

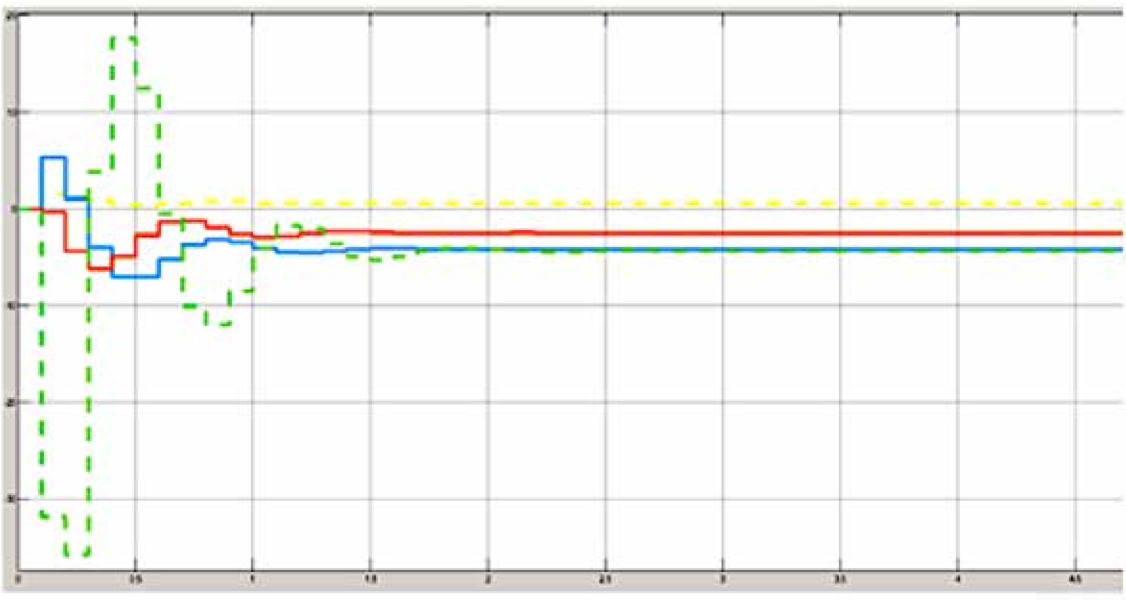

Fig. 8. System time response of the four states

Linear-Quadratic Regulator (LQR). The next applied controller was the Linear-Quadratic Regulator, which uses optimal control theory in order to manage the system at minimum cost. This controller is defined fin state space as follows:

$$\dot{x} = Ax + Bu; x \in \mathbb{R}^n; u \in \mathbb{R}^p \quad (30)$$

$$y = Cx \quad (31)$$

Performance criteria $J = \int_0^\infty [x^T(t)Qx(t) + u^T(t)Rx(t)]dt$; where Q is negative definite and R is positive definite. Then the optimal control that minimizes J is given by the linear state feedback law:

$$u(t) = -Kx(t) \quad (32)$$

Where

$$K = R^{-1} B^{T} P \quad (33)$$

Where P is the only positive definite solution of the Riccati's algebraic equation (EAR).

$$A^{T} P + PA - PBR^{-1} B^{T} P + Q = 0 \quad (34)$$

Figure 9 shows the graph of the controlled states for LQR controller. The time response graph shows that the controller stabilized the states in a few seconds.

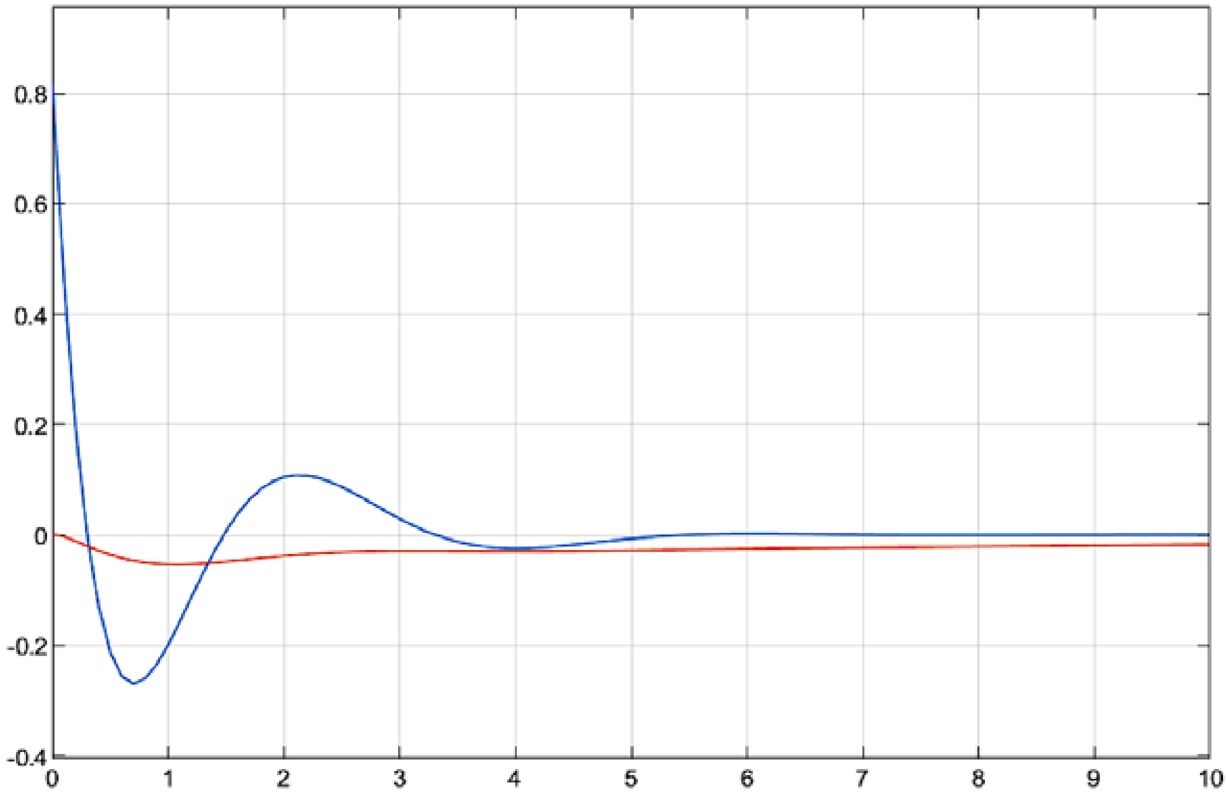

Fig. 9. System time response for the LQR controller

NARMA-L2 Neural Controller. A standard model that is used to represent general nonlinear discrete time systems is the nonlinear autoregressive mobile average (NARMA) model:

$$y(k+d) = N[y(k), y(k-1), \dots, y(k-n+1), u(k), u(k-1), \dots, u(k-n+1)] \quad (36)$$

Where u(k) is the system input e and y(k) is the system output. Using NARMA-L2 model the controller law is:

$$u(k+1) = \frac{y_r(k+d) - f[y(k), \dots, y(k-n+1), u(k), \dots, u(k-n+1)]}{g[y(k), \dots, y(k-n+1), u(k), \dots, u(k-n+1)]} \quad (37)$$

NARMA controller structure, from Matlab's network controller Deep Learning Toolbox block library in Simulink, is shown in figure 10.

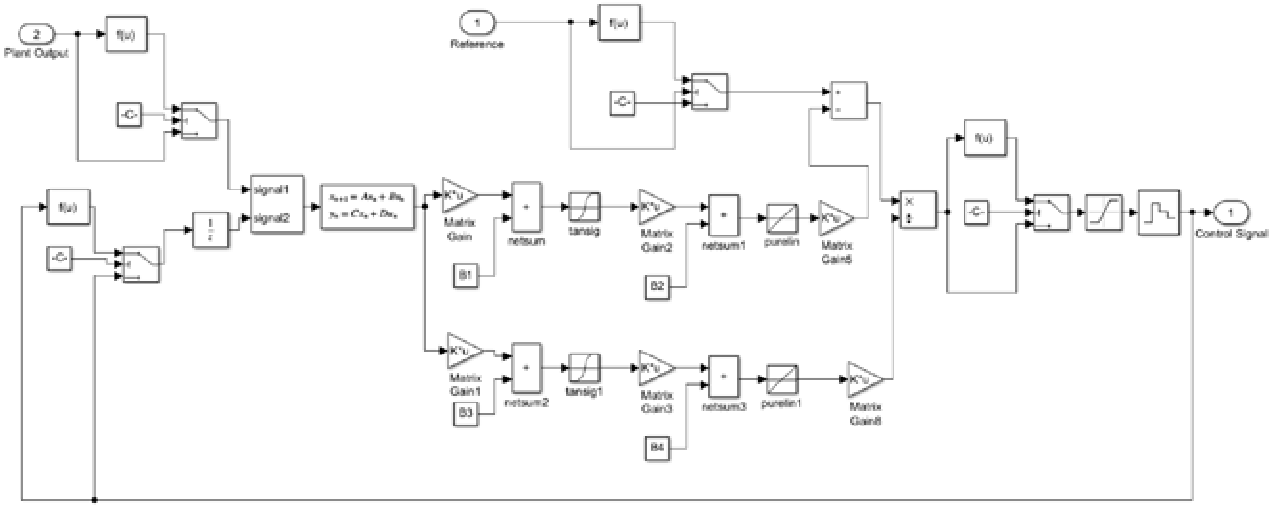

Fig. 10. Structure of Matlab's NARMA controller

For the training stage of the controller 10000 samples of random input signal values were used under Levenberg-Marquardt method, resulting in the neural network shown in figure 11.

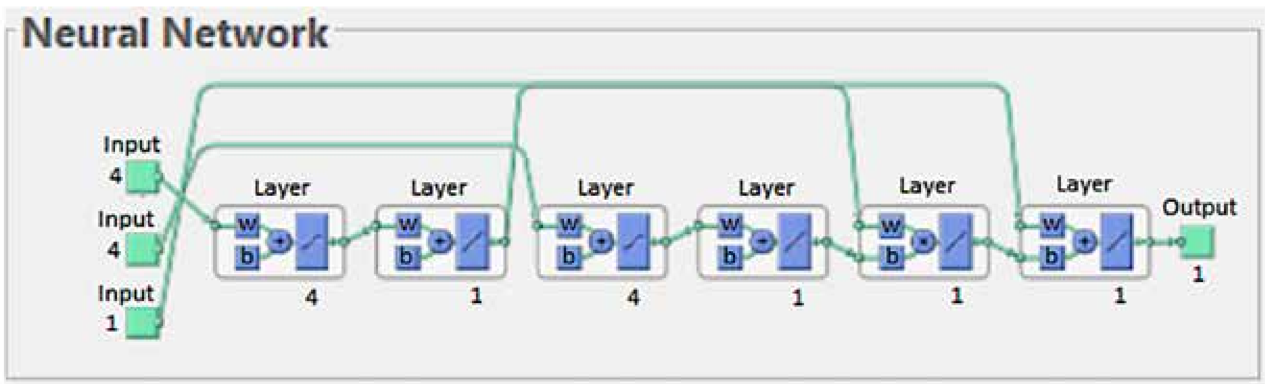

Fig. 11. Neural network obtained from training stage

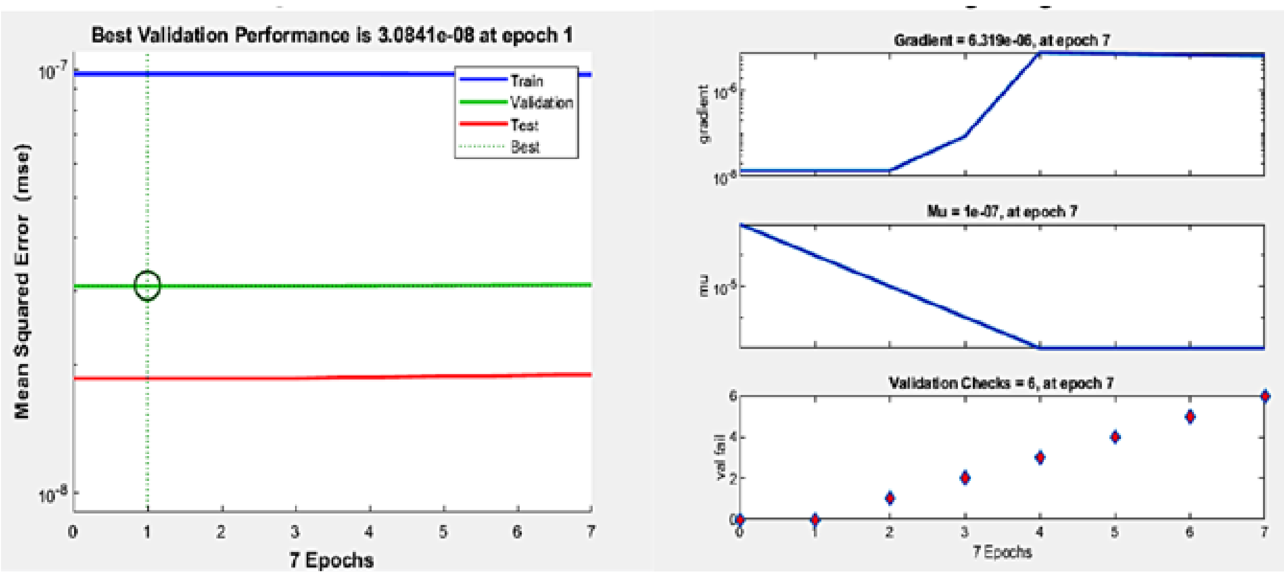

Fig. 12. Performance and training state

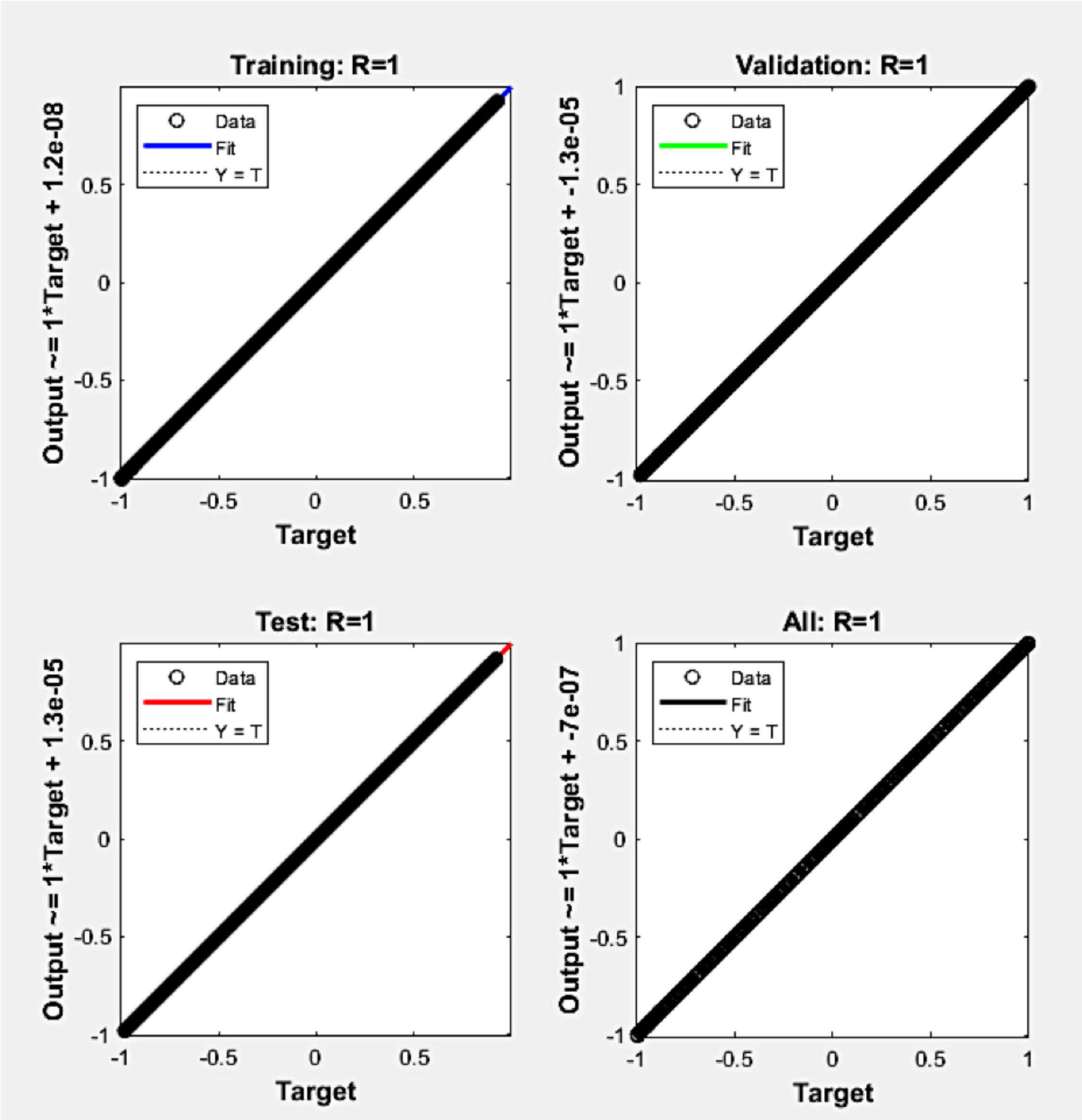

Fig. 13. Regression

Several simulations were carried out, employing constant and variable input references. Figure 14. shows the time response of the ball position, using the NARMA-L2 controller, for a desired reference pulse signal of $\frac{\pi}{36}$ rad amplitude and a period of 15 seconds, where desired ball position was achieved in less than 3 seconds.

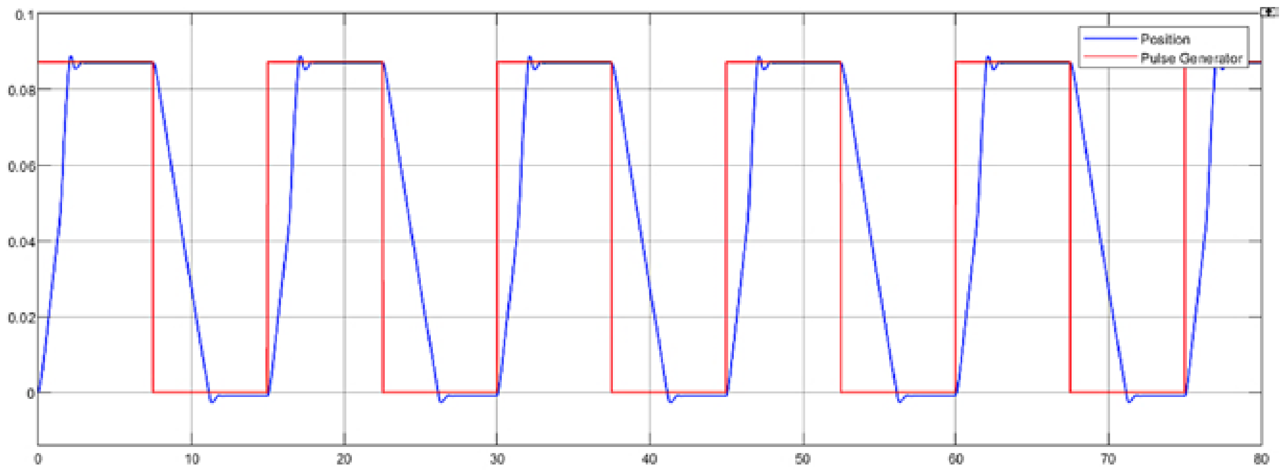

Fig. 14. System time response of NARMA controller for a pulse desired reference signal of π/36 rad amplitude and a period of 15 seconds

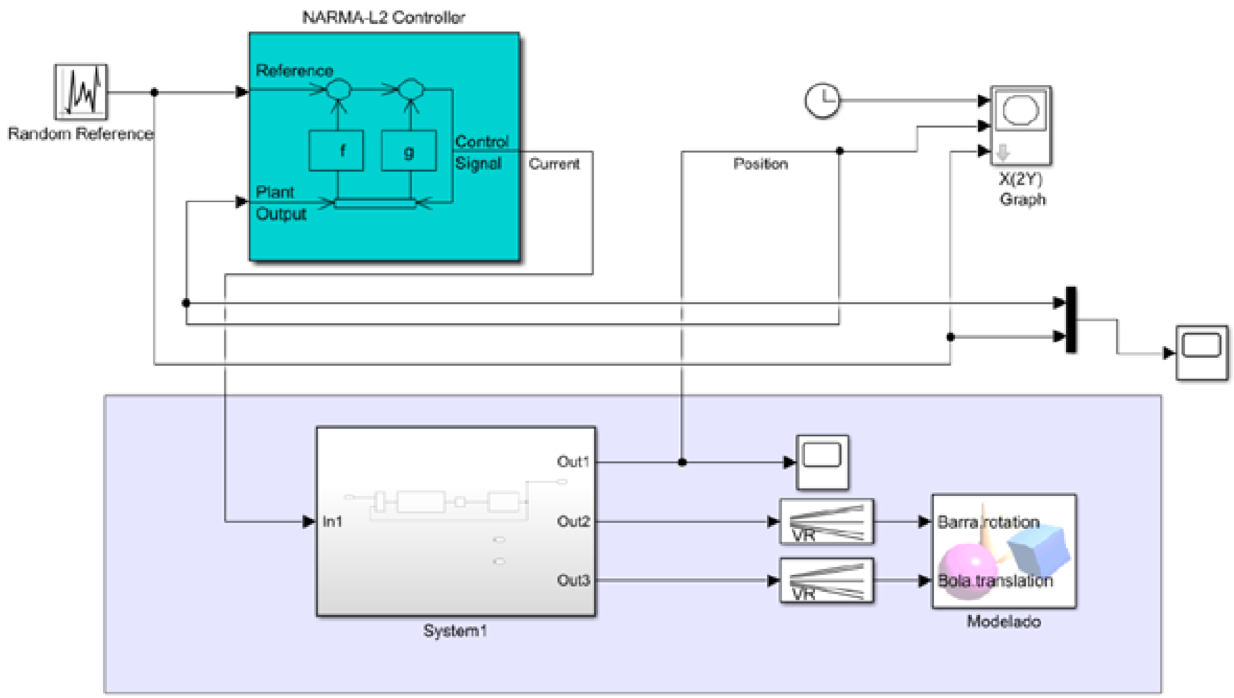

Fig. 15. Symulink model of the NARMA controller implemented in the system

State Observer. The previous state feedback controller was developed assuming that we have full access to the four state values, but considering the results of the observability matrix, an observer was developed and implemented in the Simulink interface, in order to estimate the value of $\dot{r}$, θ and $\dot{\theta}$ given the position of the sphere Υ. This, also helped to solve synchronization problems employing transformations of the state space to the discrete time using a sampling time of 0,1 seconds by means of zero order hold blocks of Simulynk, which allowed to match the desired output by giving the corrected input in the precise time period. The observer gains were also calculated by Ackerman's criteria and the block diagram and states time response are shown in figures 7 and 8.

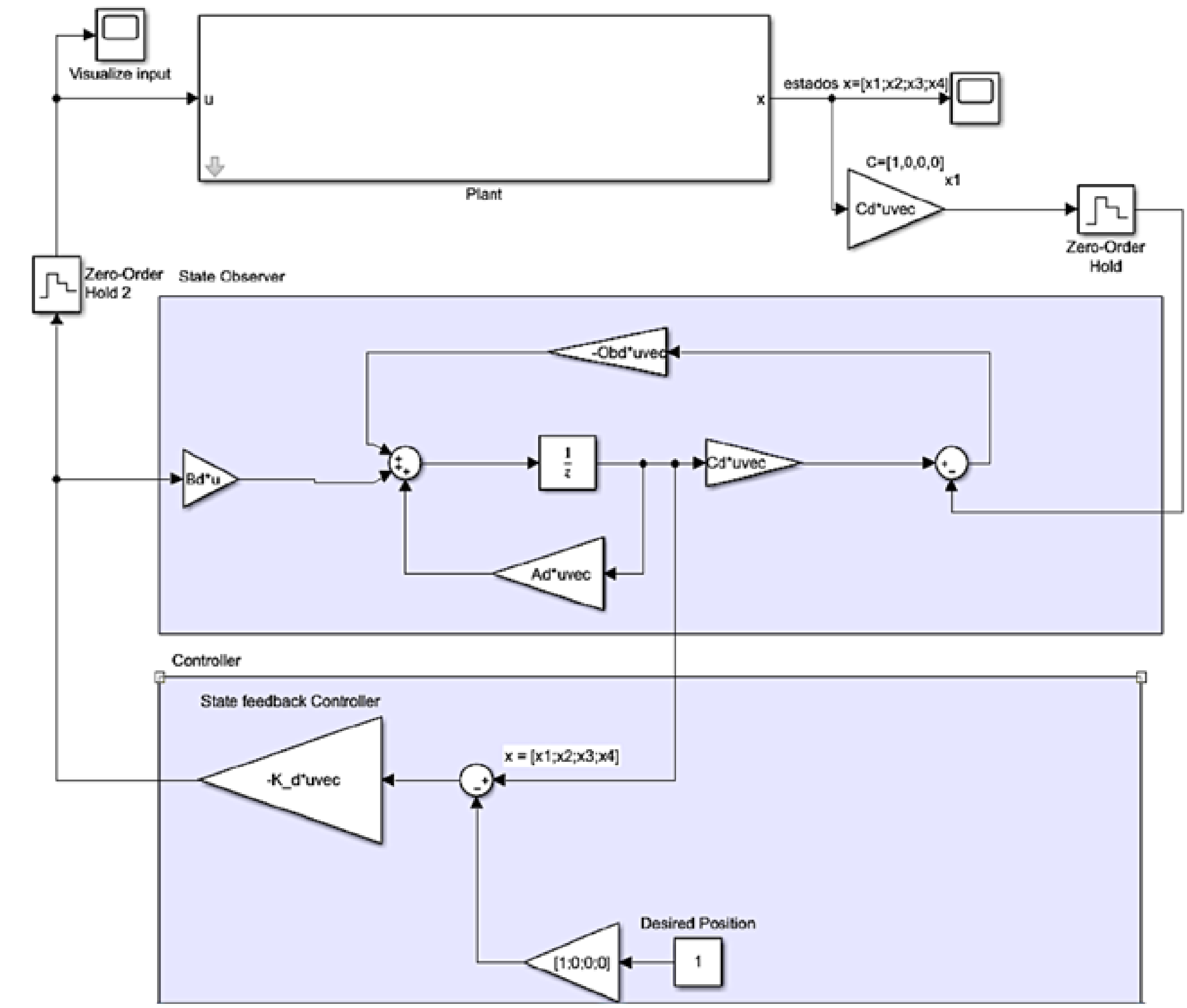

Fig. 15. Simulink model of observer and controller integrated for the system

Finally, as shown in figures 8, 9 and 14, where time response of the states using each controller is presented, overshoot, undershoot, rise time and slew rate, were selected as comparison parameters, in order to perform a comparison of controller's behavior. The results of this parameters for each controller are summarized in table 1.

Tabla 1. Performance characteristics for controller's time response performance

Performance specification	***State Feedback***	***LQR***	***NARMA***
Overshoot	25.37%	8.79%	23.78%
Undershoot	27.56%	2.43%	19.67%
Rise time	153.32 (ms)	587 (ms)	175.54 (ms)
Slew rate	1.78 (/s)	89.45 (/s)	5.46 (/s)

Conclusions and future work

From this work it can be concluded that, the development of a mathematical model by means of a state space, can be used for simulation purposes and brings a lot of possibilities for developing and testing a lot of control techniques and it can bring a lot of information for choosing the best technique for each system.

The three control techniques brought satisfying results, since all of them achieved steady state un a few seconds, but some differences were found. The state-feedback controller was one of the fastest because it was able to stabilize the system in around three seconds, but it required high torque input values and it had some overshoot problems. The linear quadratic regulator had a smoother time response under the same input reference and it required an input torque of smaller amplitude, but it took a larger time period to stabilize the four states. In general, the best results were achieved using the neuronal network-based controller NARMA-L2, because it had a really fast stabilization of around three seconds without significant overshooting problems, and it required a smaller and smoother input.

The implementation of a state observer could represent a challenging task, since two feedback systems have to be linearized and synchronized and by means of a linear multivariate equation given by the means, the input is estimated by the controller and the three states are also estimated by means of the output. This can lead to synchronization problems because the linearization by Taylor's approximation works fine close to the equilibrium point but having two linearized systems running at the same time can easily become unstable. In order to avoid this issue, in this work we converted the system to the discrete time, but other solutions like exact feedback linearization probably could overcome this issue and could be developed as future work.

References

Cuevas López, F., Mora Reyes, M.A., Olvera Mera, C.: Diseño, construcción y control de un "ball and beam." (2008).

Lopez, F., Monroy, P., Antolines, J.D.R.: Control de posición de un sistema bola y viga con actuadores magnéticos. Tecnura. 15, 12–23 (2011).

Blas, A., Martínez, O., Montesinos, J., Solıs, C.: El Sistema Barra-Esfera (Ball & Beam) en un Laboratorio de Robótica. In: II Congreso Internacional de Robótica y Computación. pp. 206–213. , Baja California (2015)

Saad, M., Khalfallah, M.: Design and Implementation of an Embedded Ball-beam Controller Using PID Algorithm. Univers. J. Control Autom. 5, 63–70 (2017)

G. C. M., «Modelamiento y Simulación del Sistema Ball and Beam,» Universidad de la Frontera, Temuco-Chile, pp. 1-12, 2010.

Marco Antonio Reyes Cisneros and Peter Wellstead." Sistema de Balancín y Pelota: Principios Básicos". División de Electrónica y Computación, CUCEI, Universidad de Guadalajara, México.

MatLab/Simulink, 7.7.0 R2008b For Unix MathWorks Ins.

Yi Guo,David J. Hill,Zhong-Ping Jiang."Global Nonlinear Control Of The Ball and Beam

System”.The University of Sydney. Australia 2017.

Shamsher Ali Naz,Reza Katebi and Luisella Balbis.Implementation of Kalman Filter On Ball And Beam Experiment Using LabView.Industrial Control Unit, Department of Electronics and Electrical Engineering University of Strathclyde, Glasgow, U.K..

A. S. Rivera Pabón y F. López Caamal, «MODELADO, CONTROL Y SIMULACIÓN DE UN SISTEMA ESFERA VIGA POR RETROALIMENTACIÓN DE ESTADOS,» Jóvenes en la ciencia, vol. 3, Nº 2, pp. 536-541, 2017.

L. A. B. Sánchez, O. M. Fuentes y J. J. M. García, «El Sistema Barra-Esfera (Ball & Beam) en un Laboratorio de Robótica,» Cinvestav, 2015.