Lebanese American University

MCE411 – Mechatronics System Design

Project Report - Ball on a Plate

Group 3 - Spring 2021

Table of Contents:

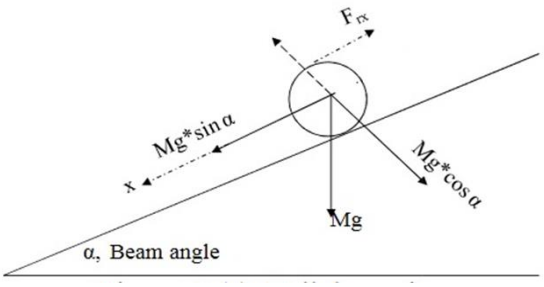
Project Description

The purpose of this project is to identify a complex multi-disciplinary engineering problem, and then try to formulate and solve it by applying principles of engineering, science, and mathematics. Particularly, our goal is to control a ball on a plate. The ball on a plate system is a system that balances a ball on a plate using a control systems technique. The objective of this project is to balance the ball on a plate with finite dimensions. The target at first is to balance the ball in the middle of the plate, then at certain positions in the given space (for example corners of a specified square area), and then on the entire plate. For this project, we will achieve the previous desired behavior using a conventional PID controller, then we will implement it using a Fuzzy Logic controller.

Mathematical Modeling of the System

System Model (Plant)

The first step in solving our problem is to mathematically model our system.



Taking the x-direction:

Assuming no slippage (friction is present), the ball moves along the plate due the force of gravity. The ball's rotational force F_{rx} is related to the torque produced by the ball's rotational motion as follows:

$$T_r = F_{rx}R$$

$$\Sigma T = J\frac{dw_b}{dt} = F_{rx}R$$

$$F_{rx} = \frac{J}{R}\theta''$$

$$F_{rx} = \frac{J}{R^2}x''_b \text{ with } R\theta'' = x'' => \theta'' = \frac{x''}{R}$$

$$\Sigma F = ma$$

$$m_b g \sin\theta - F_{rx} = m_b x''_b$$

$$m_b g \sin\theta - \frac{J}{R^2}x''_b = m_b x''_b$$

$$m_b g \sin\theta = (m_b + \frac{J}{R^2})x''_b$$

$$\text{Linearizing } \theta \approx 0 : \sin\theta = \theta \; and \; \cos\theta = 1$$

$$m_b g\theta_x = (m_b + \frac{J}{R^2})x''_b$$

$$m_b g\theta_x = (m_b + \frac{J}{R^2})x''_b$$

$$m_b g\theta_x(s) = (m_b + \frac{J}{R^2})s^2X_b(s)$$

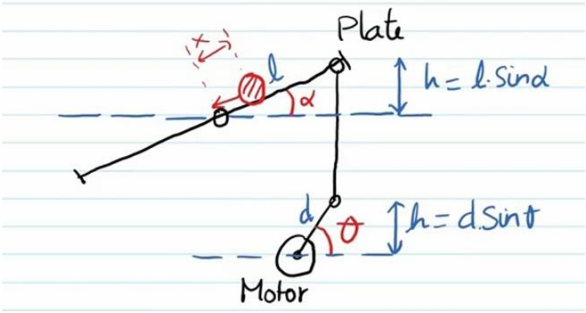
The above equation is the mathematical representation of the system in one direction. Since we assume small angle motions in the two directions, x and y are decoupled. Therefore, the mathematical model above applies to both directions independently. Thus, the transfer function of the system is:

$$\frac{X_b(s)}{\theta_x(s)} = \frac{m_b g}{(m_b + \frac{J}{R^2})s^2}$$

$$\frac{Y_b(s)}{\theta_y(s)} = \frac{m_b g}{(m_b + \frac{J}{R^2})s^2}$$

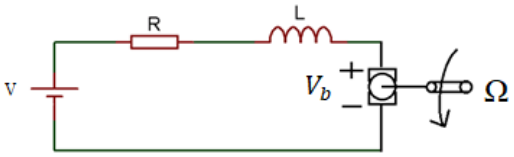
The angle α of the plate is related to that of the servomotor, θ, according to the figure on the right and as follows:

$$\theta_x = \frac{d}{l}\,\alpha_x$$
$$\theta_y = \frac{d}{l}\,\alpha_y$$



Servo Motor Model (Actuator)

Two servo motors are used to control the plate, one in the x-direction and one in the y-direction. Take $k_t = k_e = k$



The motor's electrical equations are as follows:

$$-V + RI_a + L\frac{dI_a}{dt} + V_b = 0$$
$$V = RI_a + L\frac{dI_a}{dt} + k\frac{d\theta}{dt} \qquad (1)$$

The motor's mechanical equations are as follows:

$$\Sigma T = J\frac{d\theta^2}{dt}$$
$$kI_a - c\frac{d\theta}{dt} = J\frac{d\theta^2}{dt}$$
$$J\frac{d\theta^2}{dt} + c\frac{d\theta}{dt} = kI_a \quad (2)$$

Transforming equations (1) and (2) into the Laplace domain, we obtain:

$$(1): V(s) = I(s)\,[R + Ls] + ks\theta(s)$$
$$I(s) = \frac{V(s) - ks\theta(s)}{R + Ls}$$
$$(2): kI(s) = \theta(s)\,[Js^2 + cs]$$

Substituting (1) in (2), we obtain the transfer of the servo motor:

$$k\left[\frac{V(s) - ks\theta(s)}{R + Ls}\right] = \theta(s)\,[Js^2 + cs]$$
$$\frac{k}{R + Ls}\,V(s) = \theta(s)\,[Js^2 + cs + \frac{k^2 s}{R + Ls}]$$

$$G(s) = \frac{\theta(s)}{V(s)} = \frac{k}{J_m L s^3 + (JR + Lc)s^2 + (k^2 + Rc)s}$$

The table below shows the parameters that we chose to model the system:

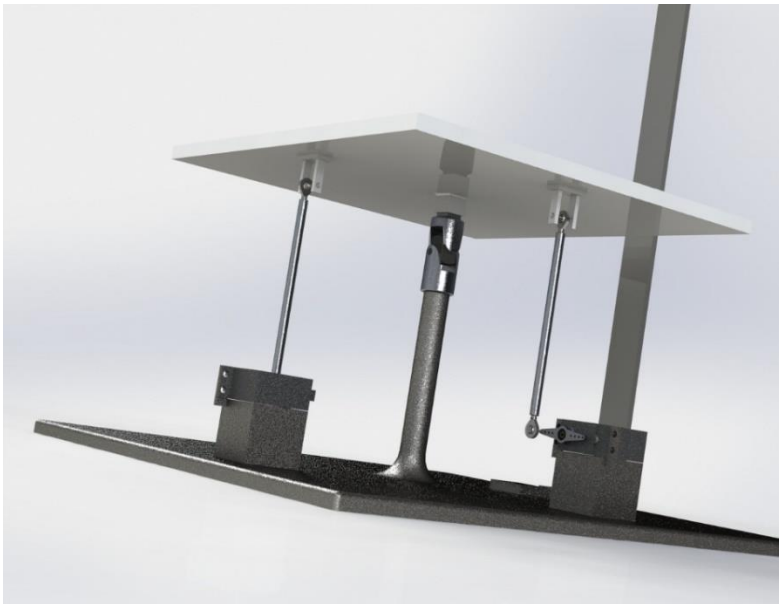
System Parameters		
K	Using the servomotor datasheet, find K = (stall torque*g)/stall current	0.0274
d	Servomotor arm length from CAD model	0.019
l	Length of arm to center of the plate (from CAD model)	0.118
L	Inductance used to model the servomotor	0.00275
R	Resistance used to model the servomotor	4
m	Mass of the solid ball	0.051
r	Radius of the solid ball	0.0124
J	Moment of interia of the solid ball	3.14E-06
Jm	Moment of inertia of the servomotor	0.0032284
c	Frictional torque constant	0.0035077
g	Gravitational constant	9.81

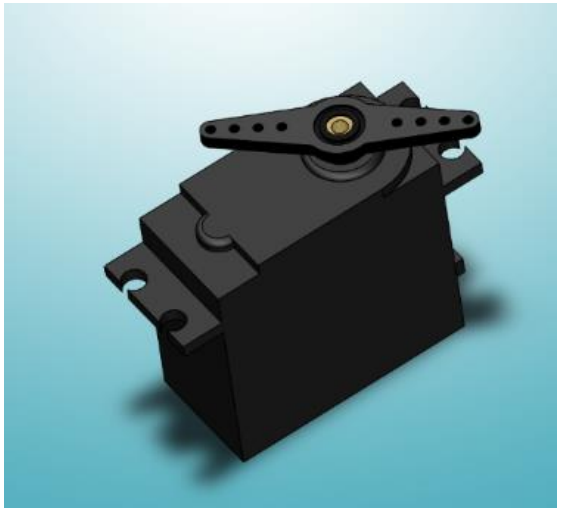
Note: These parameters are approximated. When using the true values, we were getting the error: “the transfer function is not finite”. The transfer function was being divided by 0. It seems that this error is common. The solution was to change the step size to use another solver, however, this was not doable when using Simscape. That is why we used these values.

CAD Design

Next, we modeled our system on SOLIDWORKS. The rendered images below show the CAD design of the ball on a plate system, along with the components needed to control the position of the ball, including a universal joint, two servo motors, and a camera module.

Note: When converting from SOLIDWORKS to Simscape, the assembly was complex. It was easier to redesign the assembly using Simscape. In addition, the types of joints were causing so many errors when doing the Simscape model. At the end, we used revolute joints for the motors, spherical joints between the links and the plate, and a universal joint in the center of the plate (illustrated in the *Appendix*).





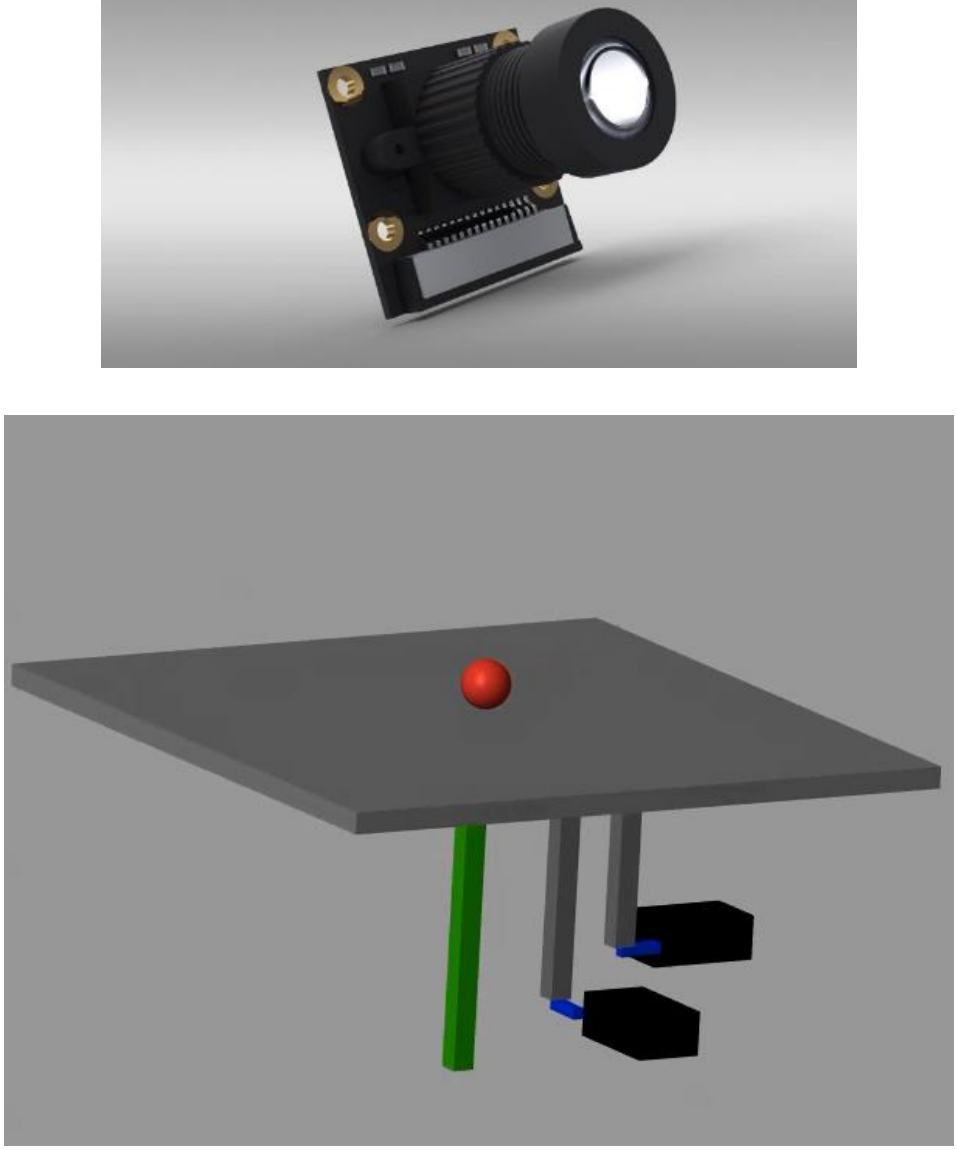


Figure 1: Simscape Model.

MATLAB/Simulink Implementation

Implementation of the PID Controller using Transfer Functions

The third step is to implement a PID controller that can control the position of the ball on the plate. The first approach is to use the transfer functions of the plant and servo motor (previously derived in *Mathematical Modeling of the System*) as shown in the block diagram below. The Ball-Plate system has 2-DOF, where each actuator is responsible for one degree of freedom since the ball's motion in the X and Y directions are decoupled. Therefore, decentralized control can be used by treating this MIMO system as two separate SISO systems, where each direction is actuated, measured, and controlled separately [8].

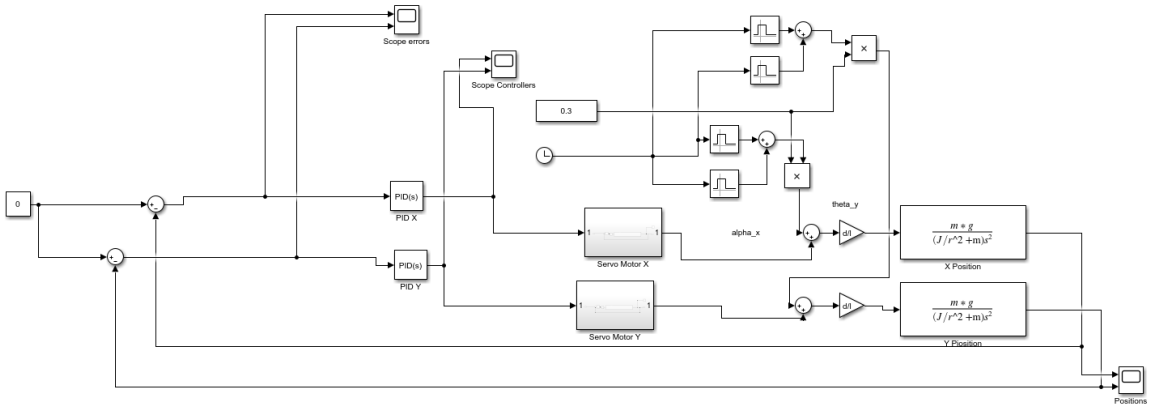


Figure 2: PID-System Block Diagram using Transfer Functions.

The output of each servo motor is the angle θ, which is related to the angle of the plate α by the relation d/l. The angle α is the input of the plant (ball-plate). The servo motor block needs a PID controller within it, along with a feedback loop, to control it. We also added a saturation block to (+/-7.8V) to make sure the voltage – output of the PID – meet the specifications of the servo motor. This is normally achieved using a PWM signal in hardware.

The disturbances are added using a clock block, sum block to add several disturbances and multiplication block (Screenshot in *Appendix*).

The servo motor block diagram is shown below:

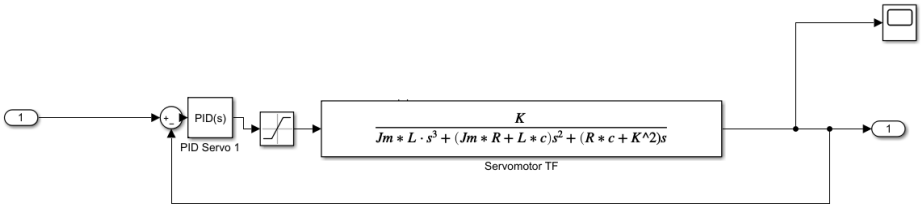


Figure 3: Servo Motor Block Diagram.

The outer loop of the block diagram is used to control the entire ball-plate system through the servo motor block.

The simulation results below show the process of controlling the ball at coordinates (0,0). We can see that the ball's X and Y coordinates reach a value of around -2×10^{-3} at the end of the simulation, which is approximately equal to 0.

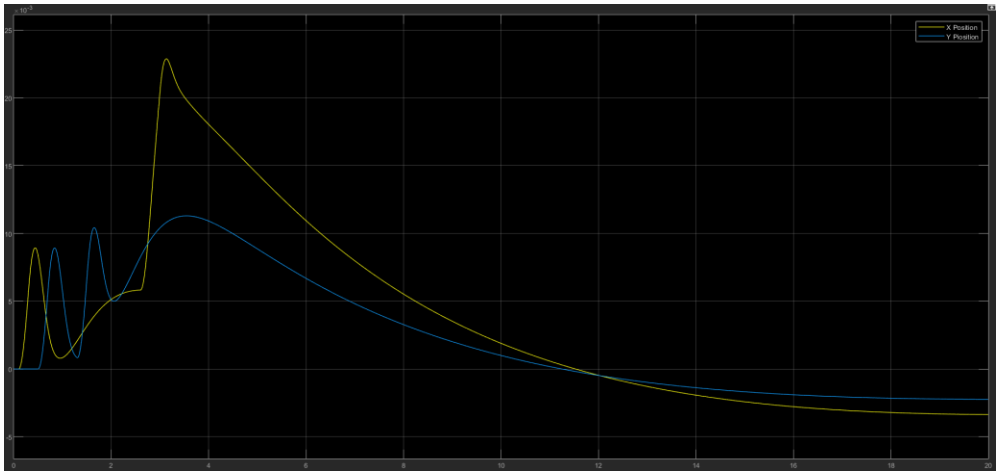


Figure 4: Simulation of X and Y Position of the Ball.

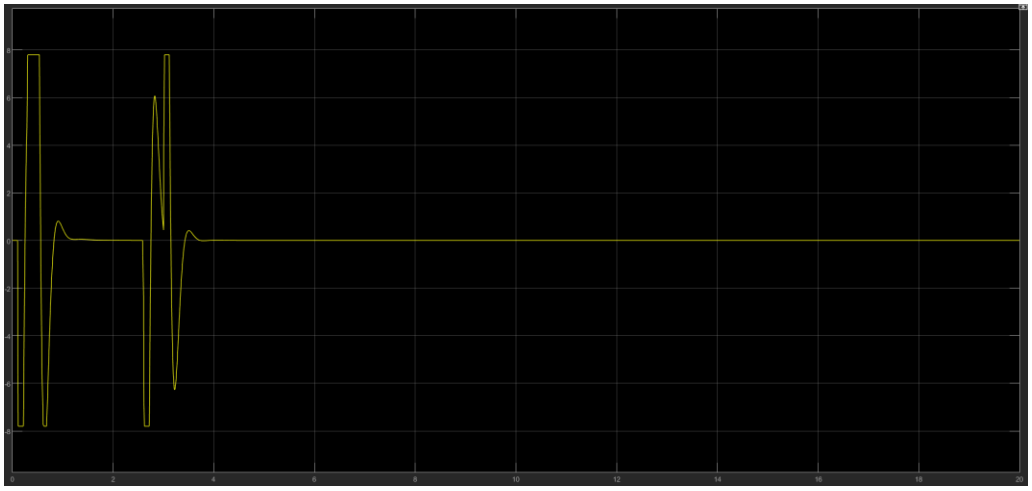


Figure 5: Actuator Input Profile.

Implementation of the PID Controller using Simscape

The second approach is to use the Simscape model of the plant, and the transfer function of the servo motor (model of the servo is the same as used above) as shown in the block diagram below.

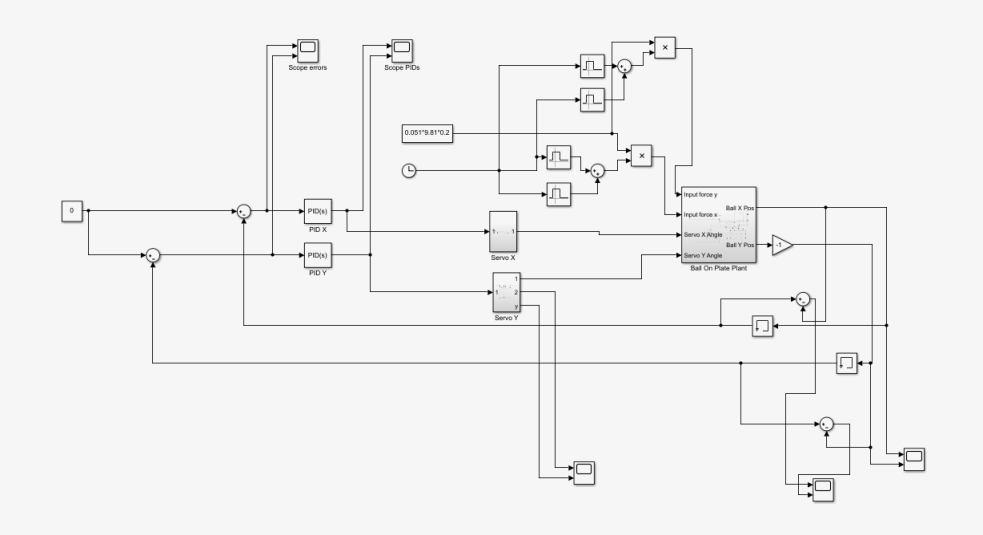


Figure 6: PID-System Block Diagram using Simscape.

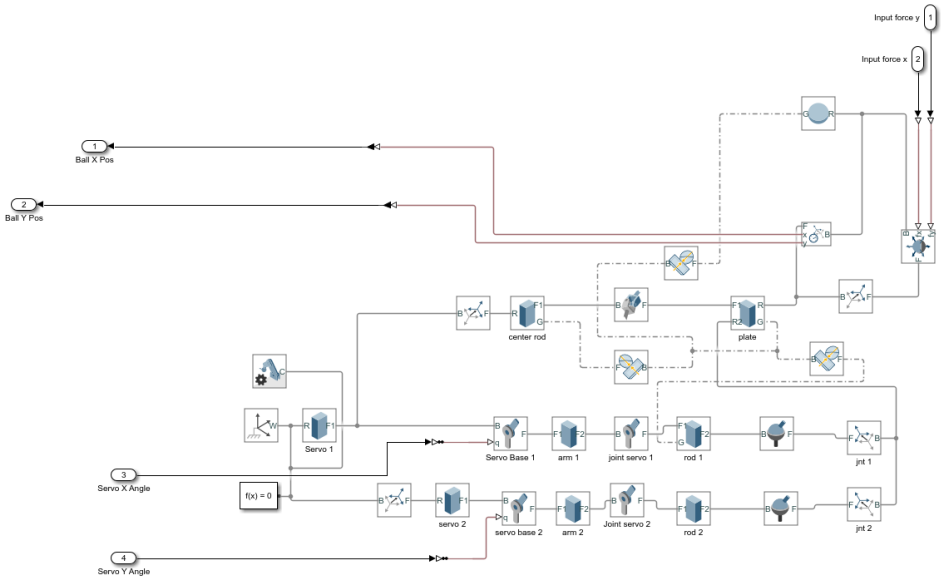


Figure 7: Ball-Plate Simscape Model.

The Simscape model of the plant, along with the rest of the block diagram needed to control it, provide a real-time perspective of how the ball is controlled at a certain position on the plate.

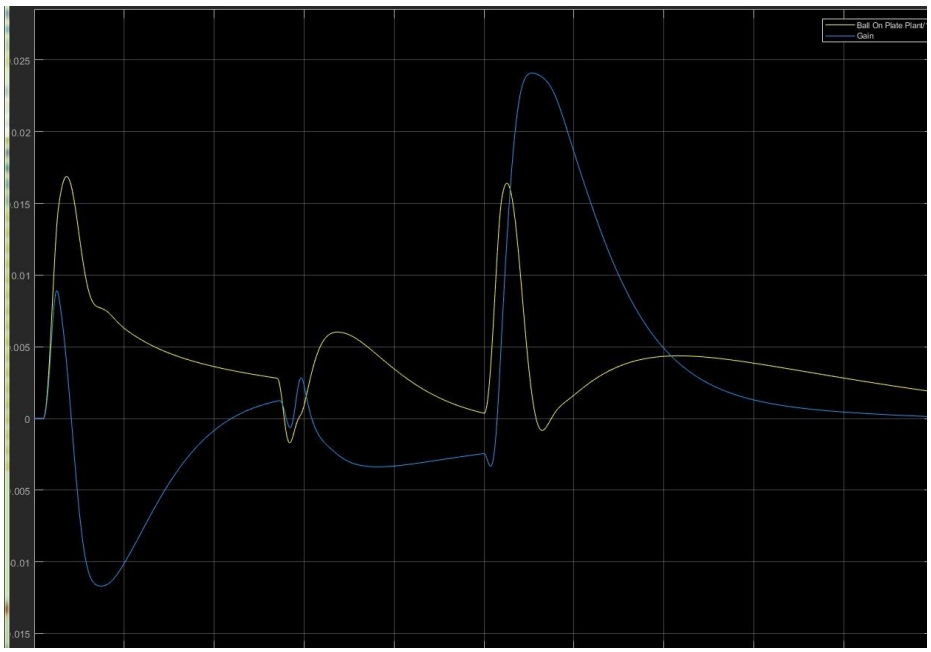


Figure 8: Simulation of X (Yellow) and Y (Blue) Position of the Ball.

Implementation of the Fuzzy Logic Controller using Transfer Functions

To control the ball-plate system, we also chose to implement a fuzzy controller in the place of the PID controller that was implemented earlier. Fuzzy control is important since it takes on continuous values instead of working with digital values made up of 0s and 1s. The block diagram below includes the transfer function representation of the ball-plate system, as well as that of each servo motor with their respective PID controllers. The input of each servo motor is controlled by a Mamdani fuzzy logic controller with two inputs (the error in position of the ball e and the rate of change of that error $\frac{de}{dt}$). The output of the fuzzy controller is the angle θ of each servo motor. The universe of discourse of each of the controller's inputs and output is:

- e = [-0.125 0.125]
- $\frac{de}{dt}$ = [-0.03 0.03]
- θ = [-0.08 0.08]

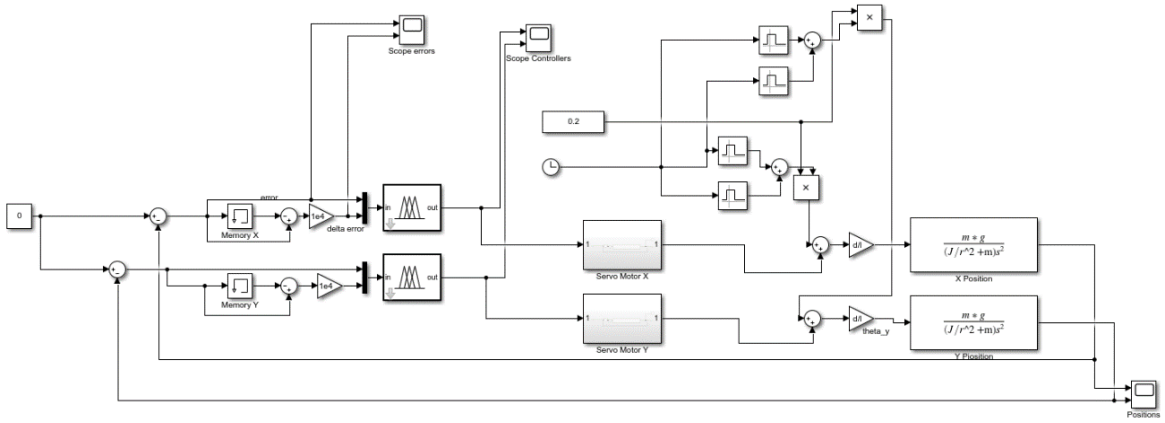


Figure 9: Fuzzy Logic-System Block Diagram using Transfer Functions.

Fuzzy Associative Matrix					
e \ e-dot	NL	NS	Z	PS	PL
NL	NL	NL	NM	NS	PS
NS	NL	NM	NS	PS	PM
Z	NL	NS	Z	PS	PL
PS	NM	NS	PS	PM	PL
PL	NS	NS	PM	PL	PL

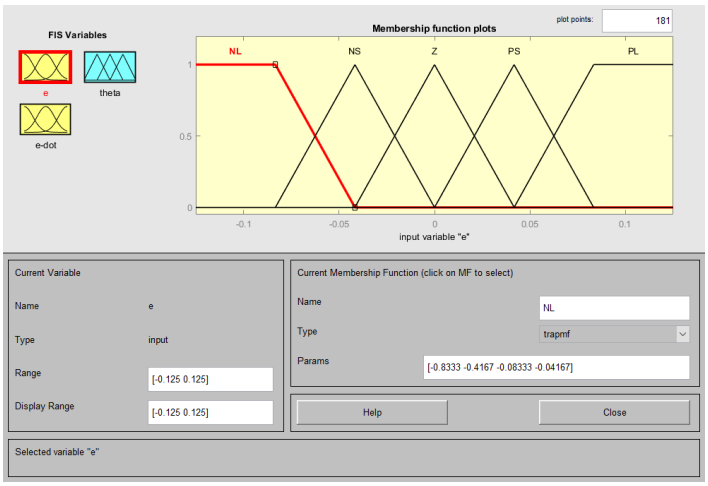


Figure 10: Error (Input) Membership Functions.

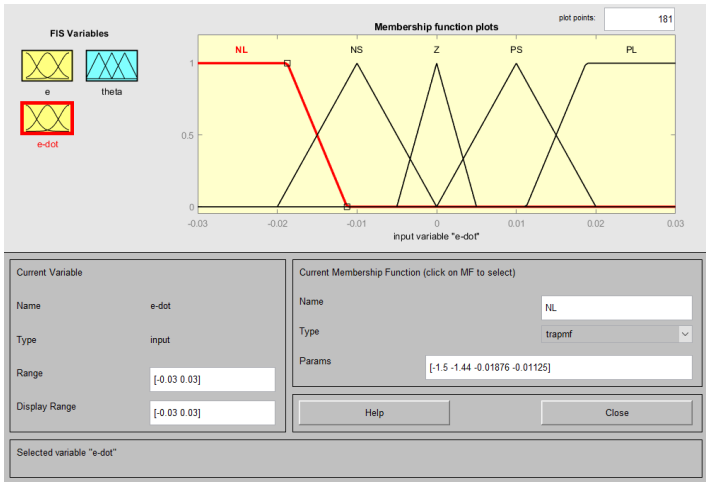


Figure 11: Rate of Change of Error (Input) Membership Functions.

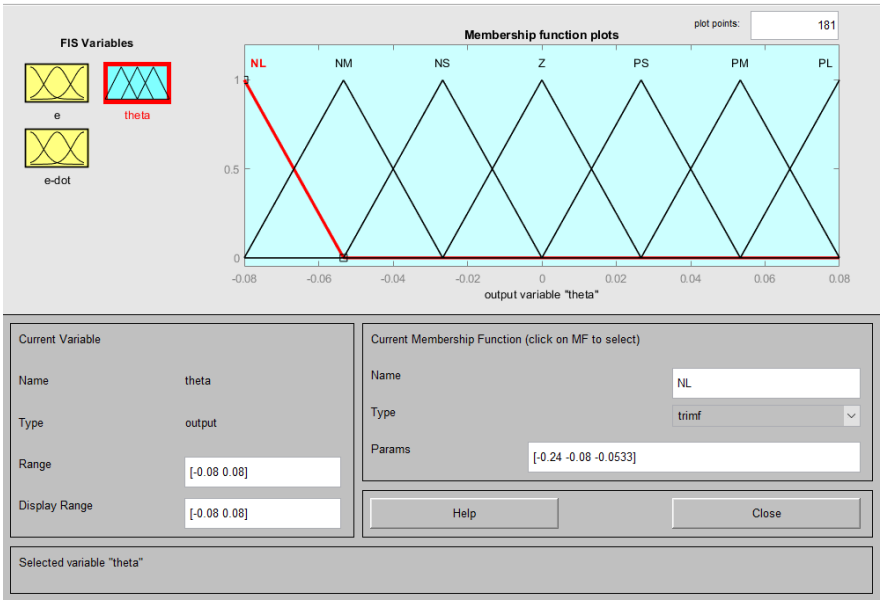


Figure 12: Output Theta Membership Functions.

To get e-dot, we used a memory block in Simulink, where the output of the block is the old value, and the input is the current value, so we get the difference between the two values, then we divide it by 1e-4 (1e-4 is the step size that we used first).

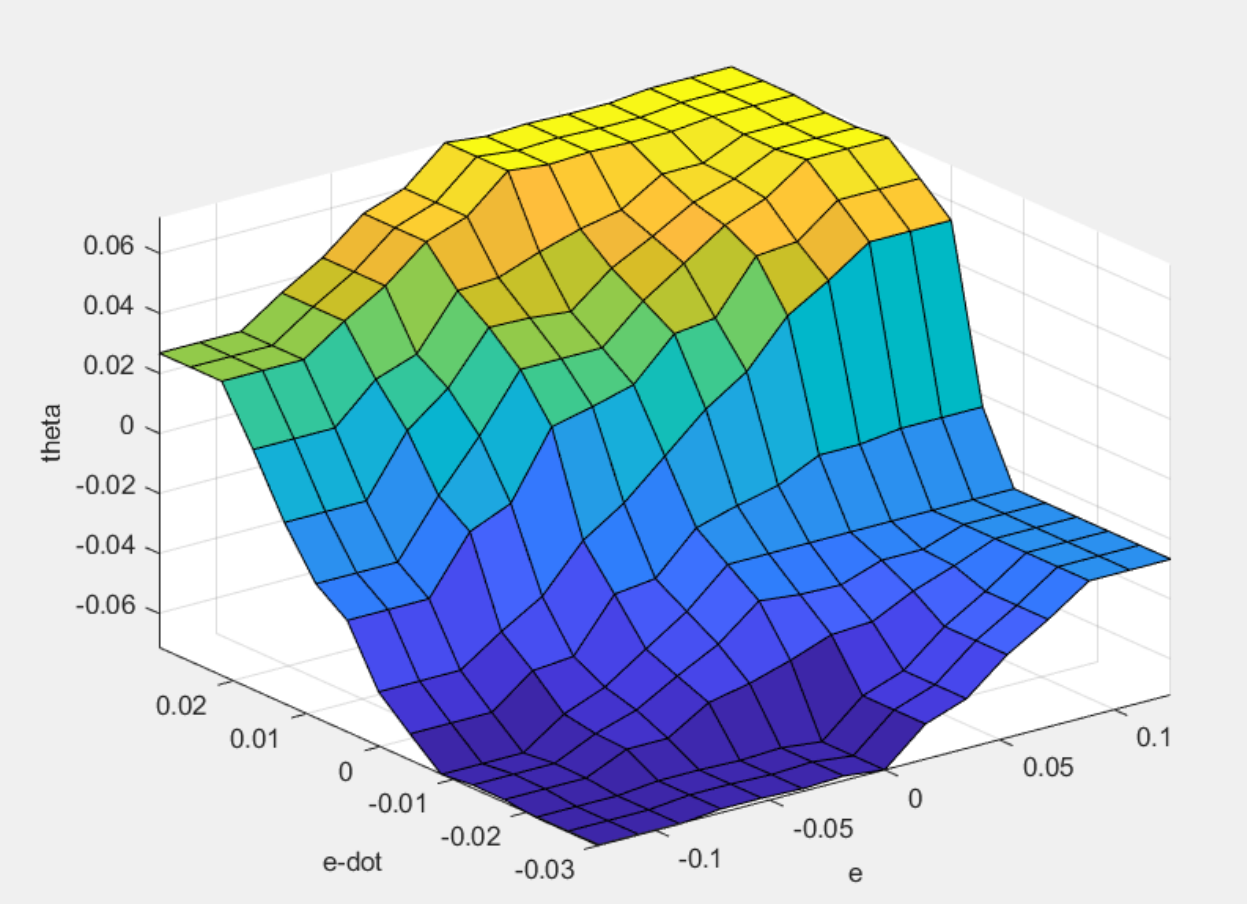


Figure 10: Fuzzy Logic 3D Control Surface.

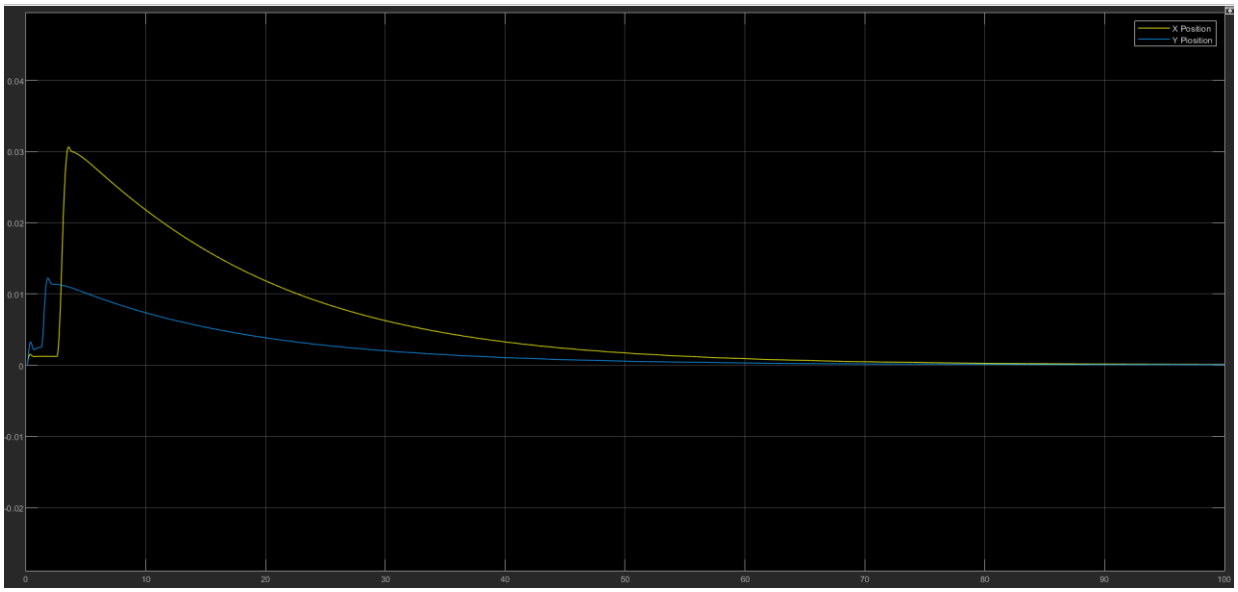


Figure 11: Simulation of X and Y Positions of the Ball.

Implementation of the Fuzzy Logic Controller using Simscape

The second approach is to use the Simscape model of the plant, and the transfer function of the servo motor as shown in the block diagram below, with the fuzzy logic controller.

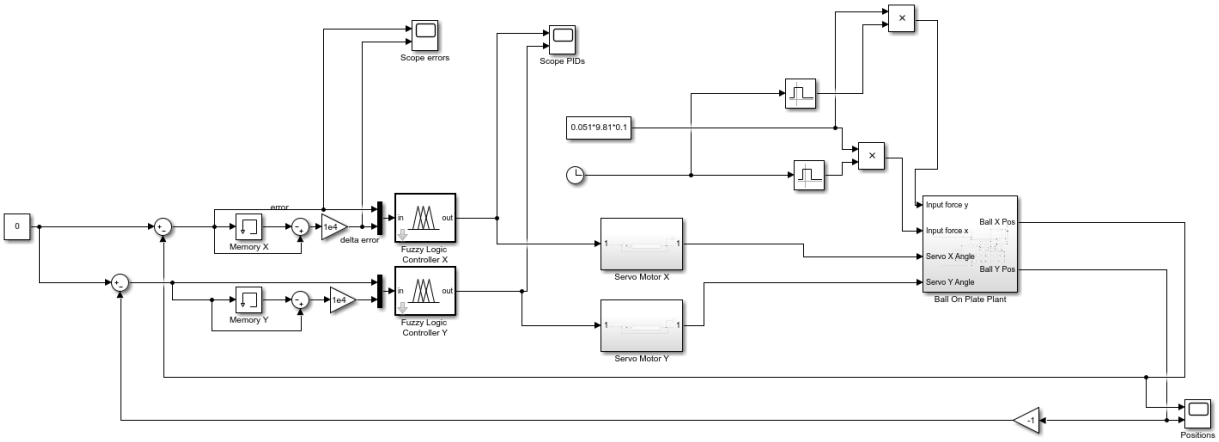


Figure 12: Fuzzy Logic-System Block Diagram using Simscape.

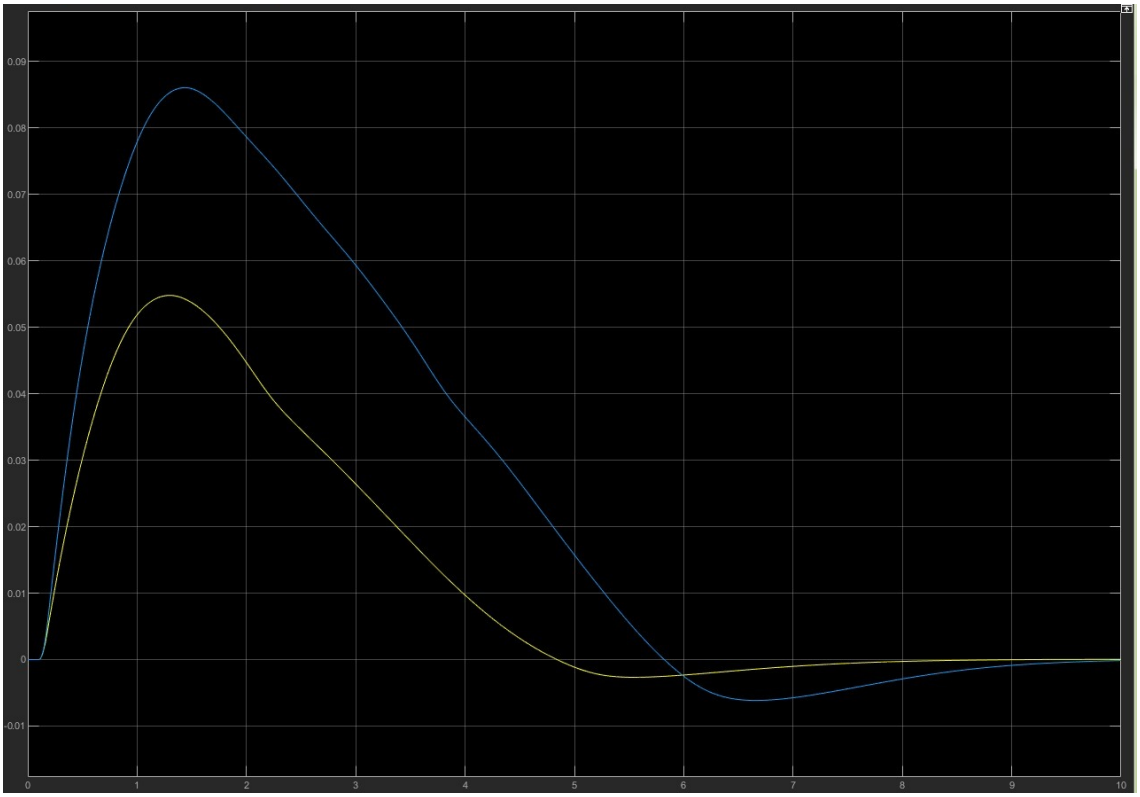
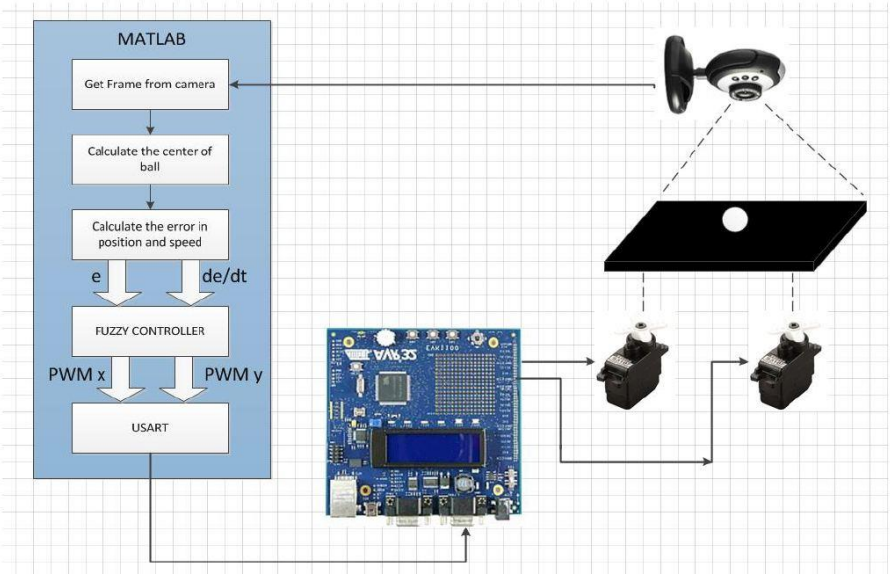


Figure 13: Simulation of X and Y Positions of the Ball.

Image Processing

If we were implementing the project in real life, a camera will be positioned above the center of the plate. After image processing, we can get the position (X,Y) of the ball, then do the control part, either using PID or fuzzy logic controller. Below are previous hardware implementations of the project:



We implemented using MATLAB toolboxes (computer vision toolbox – image processing toolbox) a function that:

1. Takes as input a video
2. Asks the user to draw:
    a. The offset from the video border to the plate border horizontally.
    b. The offset from the video border to the plate border vertically.
    c. The plate dimension along the x axis.
    d. The plate dimension along the y axis.
3. The frames of the video will be processed, and the results will be shown.

For our project, we saved the Simscape animation simulated above from a top view, we also saved the positions that we got during the simulation using the "To File" block in Simulink. The image processing function analyzed the video, then the function compared the results of the Simscape simulation and those of the image processing function.

More explanation is provided in the code.

The results are shown below:

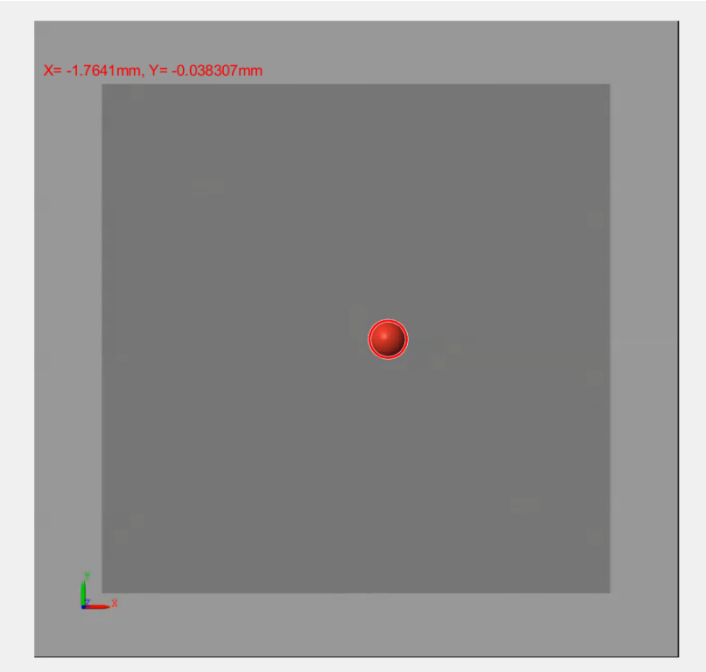


Figure 17: Image Processing Screenshot while Processing

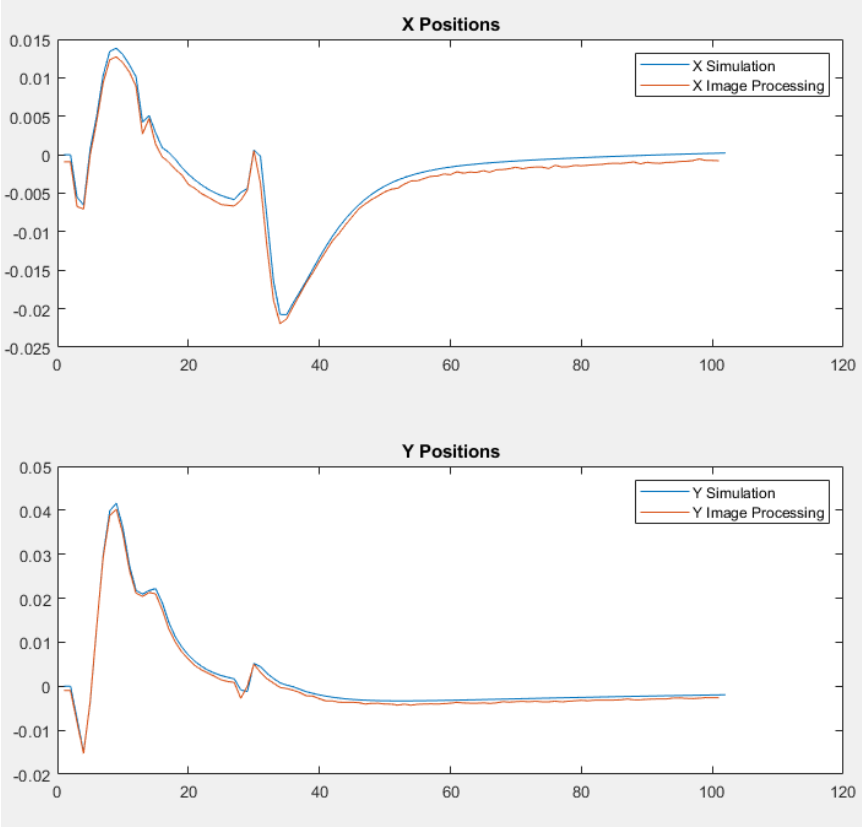


Figure 18: Results of the Image Processing Function.

Project Constraints, Notes and Conclusion

1. We did not take into consideration the friction force when modeling the plant, however, when using the Simscape model, we took the values used by [7] and we implemented it in the contact block in Simscape.
2. During simulation, we encountered many errors related to the joints. One of the errors was that we were getting a singularity when using Simscape. This was solved by using a step of 1e-5 and a tolerance of 1e-2. The tolerance is big; hence the calculations are not very accurate. This error took us so many hours to find a working configuration for. Therefore, the simulation was very slow.
3. Computing power: we were using a 16GB RAM CPU and some simulations needed 20-25 minutes.
4. In the image processing, the function provided by MATLAB is slow, when implementing it on hardware we need to use another function or to use OpenCV in c-code or python with Mex.
5. We did not take into consideration the inertia and the mass of most of the components in the Simscape design (we did it for the plate and the ball only).
6. The contact block was introduced in Simscape 2020, we did not know that at first, so we needed to update MATLAB.

In conclusion, this project covers the multidisciplinary aspect of controlling a ball at a certain position on a plate. To achieve this, we implemented two control methods, a conventional PID controller and a fuzzy logic controller. Moreover, each controller was implemented using the mathematical models of the system, as well as Simscape, which provides a simulation that is close to the real-time application of the system. Moreover, this project was able to strengthen our knowledge in building a mechatronics system.

Group Communication

The communication between members of the team was done through a *WhatsApp* group, *Zoom* for virtual meetings, in addition to file sharing and collaboration on *Google Drive*.

References

[1] M. M. Alwan, “Design and Implementation of Classical Sliding Mode Controller for Ball and Plate System,” *Journal of Engineering*, vol. 23, no. 6, pp. 1–19, Jun. 2017.

[2] F. I. Robayo Betancourt, S. M. Brand Alarcon, and L. F. Aristizabal Velasquez, “Fuzzy and PID controllers applied to ball and plate system,” *2019 IEEE 4th Colombian Conference on Automatic Control (CCAC)*, 2019.

[3] A. Itani, “BALL PLATE BALANCING SYSTEM USING IMAGE PROCESSING,” thesis, NICOSIA, 2017.

[4] A. Rastogi, R. Shooja, and C. Chiru, rep., “BALANCING BALL ON PLATE USING FUZZY LOGIC, ”2013.

[5] M. B. A. Hussien , M. G. E. M. Yousif, A. O. H. A. Altoum, and M. F. A. Abdullraheem, “DESIGN AND IMPLEMENTATION OF BALL AND PLATE CONTROL SYSTEM USING PID CONTROLLER”, Sudan University of Science and Technology, rep., 2017.

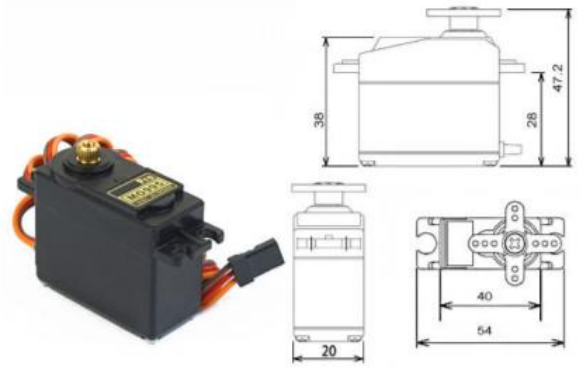
[6] K. Kaplan and I. M. Kundakcı, “Fuzzy Logic Based Ball on Plate Balancing System Real Time Control by Image Processing,” *International Journal of Natural and Engineering Sciences*, vol. 10, no. 3, pp. 28–32, Dec. 2016.

[7] N. Khaled, B. Pattel, and A. Siddiqui, *Digital Twin Development and Deployment on the Cloud: Developing Cloud-Friendly Dynamic Models Using Simulink/Simscape and Amazon AWS*. London: Elsevier, AP, Academic Press, 2020.

[8] Jihadsamarji, “jihadsamarji/Ball-Plate-Control,” *GitHub*. [Online]. Available: https://github.com/jihadsamarji/Ball-Plate-Control. [Accessed: 22-May-2021].

Appendix

MG995 High Speed
Metal Gear Dual Ball Bearing Servo

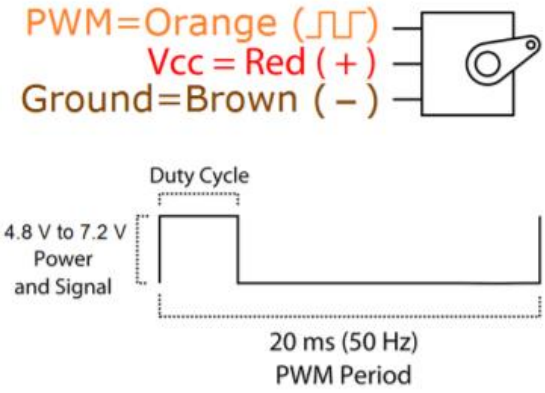


The unit comes complete with 30cm wire and 3 pin 'S' type female header connector that fits most receivers, including Futaba, JR, GWS, Cirrus, Blue Bird, Blue Arrow, Corona, Berg, Spektrum and Hitec.

This high-speed standard servo can rotate approximately 120 degrees (60 in each direction). You can use any servo code, hardware or library to control these servos, so it's great for beginners who want to make stuff move without building a motor controller with feedback & gear box, especially since it will fit in small places. The MG995 Metal Gear Servo also comes with a selection of arms and hardware to get you set up nice and fast!

Specifications

- Weight: 55 g
- Dimension: 40.7 x 19.7 x 42.9 mm approx.
- Stall torque: 8.5 kgf·cm (4.8 V), 10 kgf·cm (6 V)
- Operating speed: 0.2 s/60º (4.8 V), 0.16 s/60º (6 V)
- Operating voltage: 4.8 V a 7.2 V
- Dead band width: 5 µs
- Stable and shock proof double ball bearing design
- Temperature range: 0 ºC – 55 ºC



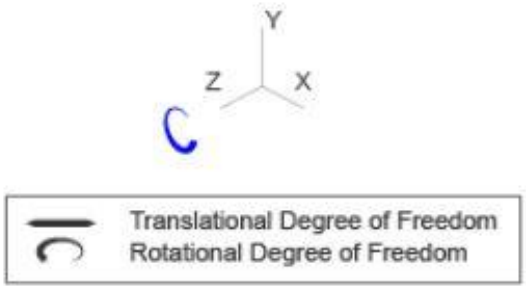


Figure 19: Revolute Joint

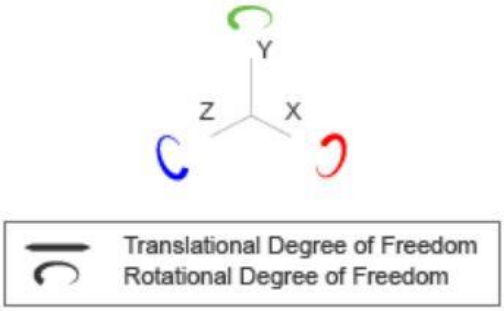


Figure 20: Spherical Joint.

Joint Degrees of Freedom

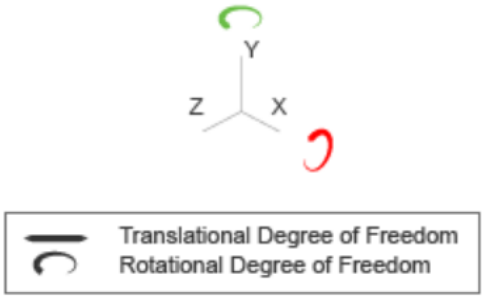


Figure 21: Universal Joint

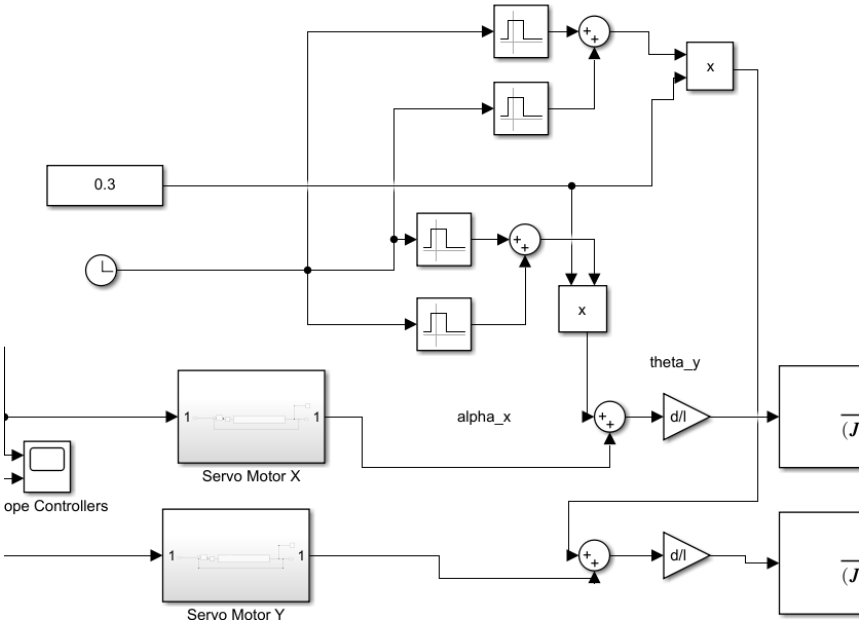


Figure 22: Disturbances (Different for Simscape)