MCH4921 Mechatronics System Design

Ball Balancing Table Project

Kutay Marangoz, Kaan Özcivan, Yaren Yasmin Engin, Remzi Dağhan Kara

Abstract— **The main objective of this project is to simulate a ball balancing table, focusing on balancing a thrown ball under 10 seconds. Given that the project will be executed in a simulated environment, the results obtained may not directly translate to real-life scenarios. However, they can provide sufficient insights for further development if required. For CAD purposes, SOLIDWORKS was chosen, and the implementation phase involves the use of a Proportional Integral Derivative (PID) controller. Mathematical equations and calculations were performed using Matlab, while Simulink was employed for testing purposes. The conclusion of the entire process involves the transportation of data to Simscape.**

I. Introduction

Application of control theory to the use of dynamic systems for the project has been crucial. Properly using control theory helped us greatly to stabilize the unstable simulative environment. For this Project it was decided to use control theory to stable the unstable-state platform to steady state the balance a thrown ball in the center of the plate. By working and researching extensively for different aspects of the balancing table great understanding and familiarization of a designing a mechatronics system in simulative environment have been gained. After the said mechatronics system is done in the simulative environment it can be further progressed to experiment and monitor it in the physical form as well if needed. Also hardware was needed for the project to progress after the initial design process in SOLIDWORKS. For understanding the position of the thrown ball mono camera is decided to use as perceive sensor. This sensor will send data to microcontroller of the dynamic system. Two servo-motors will be controlled by the microcontroller after receiving the said data. For testing purposes controlling the position of the ball will be done manually as well as randomly to achieve further stabilization data for our success in the system. In the further works the aim to use this dynamic system in other industrial areas by developing it further if possible. And if the project continues in the real life environment results obtained will have a greater understanding of comparison of the differences and similarities between real life and simulative environment.

II. Literature Review

This literature review provides information about the existing projects done regarding ball balancing plate, focusing on the mechanical design, actuator, controller, embedded computing and software choices and the simulation environments that are used on the projects. First Project observed was done by students from KTH Royal Institute of Technology. They used Solid Edge for technical drawing. In this same project, the geometrical analysis was carried out using MATLAB. As hardware components, this project involved two servo motors, Arduino Uno microcontroller and a 4-wire touch panel. The 4-wire analog resistive touch panel was used to determine the position of the ball. The input data given by the panel is processed by the microcontroller and is sent to the servo motor as an output, which generates motion. The simulation is performed using SIMULINK. A low pass filter is implemented in order to regulate the input from the touch panel so that the disturbances and noise is accounted. This project has some undesired outcomes such as the touch panel not reading correct positions of the ball accurately, so the input data contained noise. Moreover, the low pass filter caused delay in the system which caused the motors not reacting to change on time. [1] Another project observed was done by the students of Lebanese American University. The CAD program choice for this project is SOLIDWORKS. For the implementation of the PID controller, MATLAB/Simulink is used. Simscape is used for the PID system block as another approach, which provides a real time perspective. They also tried implementation of a Fuzzy Logic Controller in place of PID in order to work with continuous values instead of only digital values. For real life implementation, image processing is preferred by using a camera positioned above the center point of plate and the MATLAB's Computer Vision Toolbox.[2] Furthermore, the documents of ball balancing table project done by Acrome Inc. was also observed. Components are power distribution box, RC servomotors, 2D Resistive Touch panel, low pass filter and controller. This document provides detailed explanations of the implementation of the equations of motion and the transfer functions, while also giving thorough information about performance measures and control system design.[3] For that reason, this particular document is chosen as a map for the calculations that will be implemented.

III. MATERIALS AND METHODS(DESIGN)

A. *Application That Are Utilized in the Project*

In this project options were given as a choice between CoppeliaSim and Simscape as simulation environments that are to be applied. It has been determined that the application of a PID controller, which will definitely be required, would be more efficient with the usage of Simscape since MATLAB has more refined tools provided with Simulink block diagrams. For CAD program, as progresseion with the project done, the available options were reduced to two which were, utilizing Simscapes' built in tools which contained 3D block diagrams and rigid transformations or sketching the entire plant by using a CAD tool and exporting it to the Simscape environment by transforming the file extensions. As the project continued With the Simscape environment there was a realization that it had many frame calculations therefore it would be very time consuming to build the plant from scratch with Simscape since none of us had any experience with its tools. As a result of this situation, the decision was made to choose a fully functional CAD program for the design. As for the which CAD program, decision was made towards use SOLIDWORKS since it was very compatible with MATLAB compared to the other options there were. Choice of SOLIDWORKS also effected by the experience were given due to available various resources and while also team members being more experienced with SOLIDWORKS environment.

B. *Parameters and Dimensions*

In this project the given basis dimension with a plate having 30cmx30cm cross section or less. According to this limitation it was chosen that some basic parameters which are the ball's radius and mass as 1.25cm and 1kg respectively and the gravitational acceleration as 9.81 m/s2. The remaining dimensions were adjusted according to the ones that were already chosen. The parameters that have been processed in the control functions are as follows:

Table 1. Parameters and Dimensions

Link Length in x-direction	L_x
Link length in y-direction	L_y
Servo Link length	L_s
Radius of the ball	R
Mass of the ball	m
Rotational inertia of the ball	J
Gravitational Acceleration constant	g
Plate angle in x-axis	α
Plate angle in y-axis	β
Servo angle for x-axis	$\theta_{s,y}$
Servo angle for y-axis	$\theta_{s,y}$
Armature resistance of motor	R_a
Armature inductance of motor	L_a
Armature Current	I_a
Armature voltage	V
Back EMF voltage	V_b
Inertia of servo	J_s
Viscous friction coefficient of servo	b_s
Servo-Torque constant	K_t
Back EMF constant	K_e
Angular position of servo shaft	θ_s
Angular velocity of servo shaft	ω_s
Torque produced by servo	T_s

C. *Equations*

1) *Mathematical modeling of servo motor (Actuator)*

By establishing relations between electrical circuits and mechanical motion of the servo motor, mathematical model if the motor is gathered.

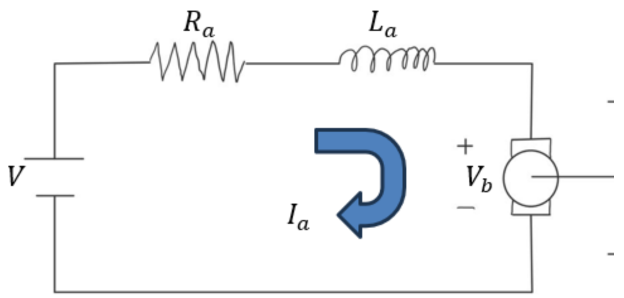

Figure 1. Electrical Circuit of the Servo Motor

For the simplicity of calculations, it is assumed that Motor Torque Constant (Kt) is equal to Back EMF Constant (Ke), and is shown as K in calculations.

Electrical equations of servo motor:

$$-V(t)+R_aI_a(t)+L_a\frac{dI_a(t)}{dt}+V_b(t)=0$$

$$V_b(t)=K\omega_s(t)\Longrightarrow\ E_b(t)=K\frac{d\theta_s(t)}{dt}$$

$$\boldsymbol{V(t)=R_aI_a(t)+L_a\frac{dI_a(t)}{dt}+K\frac{d\theta_s(t)}{dt}} \qquad \textbf{(1)}$$

Mechanical equations of servo motor:

$$T_s(t)=KI_a(t)$$

$$T_s(t)=J_s\frac{d\theta_s(t)^2}{d^2t}+b_s\frac{d\theta_s(t)}{dt}$$

$$\boldsymbol{KI_a(t)=J_s\frac{d\theta_s(t)^2}{d^2t}+b_s\frac{d\theta_s(t)}{dt}} \qquad \textbf{(2)}$$

Laplace Transform of equations:

$$\boldsymbol{V}(t)=R_aI_a(t)+L_a\frac{dI_a(t)}{dt}+K\frac{d\theta_s(t)}{dt}$$

$$\Longrightarrow\ \boldsymbol{V(s)} \qquad \textbf{(3)}$$
$$\boldsymbol{=R_aI_a(s)+sL_aI_a(s)+sK\theta_s(s)}$$

$$KI_a(t)=J_s\frac{d\theta_s(t)^2}{d^2t}+b_s\frac{d\theta_s(t)}{dt}$$

$$\Longrightarrow\ \boldsymbol{KI_a(s)=s^2J_s\theta_s(s)+sb_s\theta_s(s)} \qquad \textbf{(4)}$$

Obtaining Transfer Function of servo motor:

$$V(s)=R_aI_a(s)+sL_aI_a(s)+sK\theta_s(s)$$

$$\Longrightarrow V(s)-sK\theta_s(s)=I_a(s)(R_a+sL_a)$$

$$\boldsymbol{I_a(s)=\frac{V(s)-sK\theta_s(s)}{R_a+sL_a}} \qquad \textbf{(5)}$$

Substituting equation (5) to (4)

$$K\left[\frac{V(s)-sK\theta_s(s)}{(R_a+sL_a)}\right]=s^2J_s\theta_s(s)+sb_s\theta_s(s)$$

$$\Longrightarrow\ \frac{KV(s)-sK^2\theta_s(s)}{(R_a+sL_a)}=\theta_s(s)(s^2J_s+sb_s)$$

$$\frac{K}{R_a+sL_a}V(s)-\frac{sK^2}{R_a+sL_a}\theta_s(s)=\theta_s(s)(s^2J_s+sb_s)$$

$$\Longrightarrow\ \frac{K}{R_a+sL_a}V(s)=\theta_s(s)\left(s^2J_s+sb_s+\frac{sK^2}{R_a+sL_a}\right)$$

$$\frac{V(s)}{\theta_s(s)}=\left(\frac{R_a+sL_a}{K}\right)\left(\frac{(R_a+sL_a)\left(s^2J_s+sb_s\right)+sK^2}{R_a+sL_a}\right)$$

$$\frac{\theta_s(s)}{V(s)}=\frac{K}{s^2R_aJ_s+sR_ab_s+s^3L_aJ_s+s^2L_ab_s+sK^2}$$

Transfer function of servo motor

$$G_s(s)=\frac{\theta_s(s)}{V(s)}$$

$$\boldsymbol{G_s(s)=\frac{K}{J_sL_as^3+(J_sR_a+L_ab_s)s^2+(K^2+R_ab_s)s}} \qquad \textbf{(6)}$$

2) *Mathematical Modeling of the System*

By mathematical analysis of ball-table interaction, table-motor relation and motor’s conversion from electrical to mechanical, mathematical modeling of system is obtained.

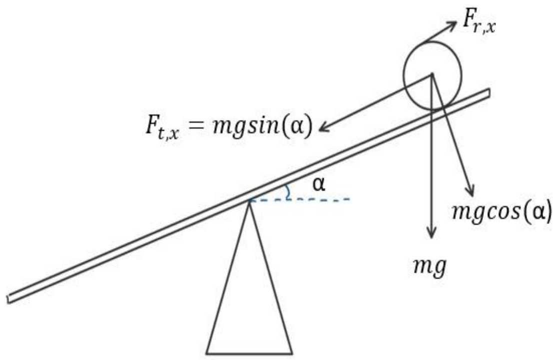

Figure 2. Free-body diagram of 2D System

Application of Newton’s law of motion to the system :

$$\sum\mathbf{F}=\mathbf{F_{t,x}}-\mathbf{F_{r,x}}=\mathbf{m_b}\frac{\mathbf{dx_b}(t)^2}{\mathbf{d^2t}}\rightarrow(\mathbf{7})$$

$$\mathbf{F_{t,x}=m\,g\,sin(\alpha)}\rightarrow(\mathbf{8})$$

$$\boldsymbol{\tau}=\mathbf{F_{r,x}R}=\frac{\mathbf{J}}{\mathbf{R}}\frac{\mathbf{dx_b}(t)^2}{\mathbf{d^2t}}\Longrightarrow\ \mathbf{F_{r,x}}=\frac{\mathbf{J}}{\mathbf{R^2}}\frac{\mathbf{dx_b}(t)^2}{\mathbf{d^2t}}\rightarrow(\mathbf{9})$$

Combining equations 7, 8, and 9

$$\mathrm{m\,g\,sin}(\alpha)-\frac{\mathrm{J}}{R^2}\frac{\mathrm{dx_b}(t)^2}{\mathrm{d^2t}}=\mathrm{m}\frac{\mathrm{dx_b}(t)^2}{\mathrm{d^2t}}$$

Acceleration of ball in x-direction

$$\frac{\mathbf{dx_b}(\boldsymbol{t})^2}{\mathbf{d^2t}} = \frac{\mathbf{mgR^2}}{\mathbf{mR^2+J}}\mathbf{sin}(\boldsymbol{\alpha}) \rightarrow (\mathbf{10})$$

Acceleration of ball in y-direction

$$\frac{\mathbf{dy_b}(\boldsymbol{t})^2}{\mathbf{d^2t}} = \frac{\mathbf{mgR^2}}{\mathbf{mR^2+J}}\mathbf{sin}(\boldsymbol{\beta}) \rightarrow (\mathbf{11})$$

3) Modeling of Ball Balancing Table System

Figure 3 shows the mathematical relation between table and motor. The relation consists of 2 degree of freedom system.

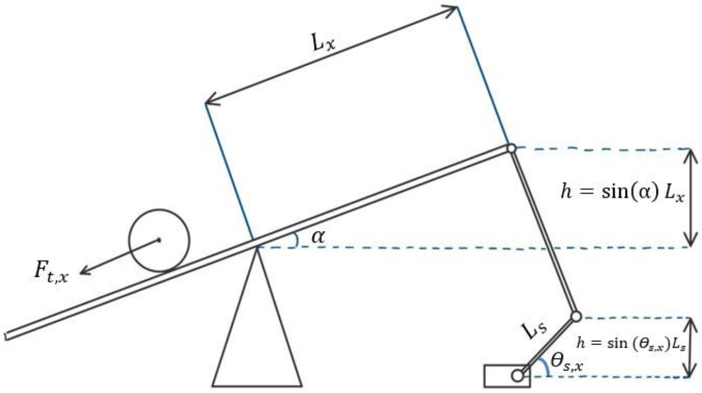

Figure 3. Ball Balancing Table System Relations

In the figure above, d_m stands for the distance between motor's output shaft and center of the joint which connects upper and lower link. L_x is the distance between center of table and center of joint which connects the upper link with the table. h is the distance that is created by the motor output.

Applying linear ratio operation in order to obtain h:

$$\sin(\theta_{s,x})L_s = \sin(\alpha)L_x = h$$

Relation in x-direction:

$$\mathbf{sin}(\boldsymbol{\alpha}) = \frac{\boldsymbol{L_s}}{\boldsymbol{L_x}}\mathbf{sin}(\boldsymbol{\theta_{s,x}}) \rightarrow (\mathbf{12})$$

Relation in y-direction:

$$\mathbf{sin}(\boldsymbol{\beta}) = \frac{\boldsymbol{L_s}}{\boldsymbol{L_y}}\mathbf{sin}(\boldsymbol{\theta_{s,y}}) \rightarrow (\mathbf{13})$$

Substituting equation (12) to (10) to obtain the acceleration in x-direction

$$\frac{\mathrm{dx_b}(t)^2}{\mathrm{d^2t}} = \frac{\mathrm{mgR^2}}{\mathrm{mR^2+J}}\frac{L_s}{L_x}\sin(\theta_{s,x})$$

$$\Longrightarrow \frac{\mathbf{dx_b}(\boldsymbol{t})^2}{\mathbf{d^2t}} = \frac{\mathbf{mgR^2}\boldsymbol{L_s}}{(\mathbf{mR^2+J})\boldsymbol{L_x}}\mathbf{sin}(\boldsymbol{\theta_{s,x}}) \rightarrow (\mathbf{14})$$

Substituting equation (11) to (10) to obtain the acceleration in y-direction

$$\frac{\mathrm{dy_b}(t)^2}{\mathrm{d^2t}} = \frac{\mathrm{mgR^2}}{\mathrm{mR^2+J}}\frac{L_s}{L_y}\sin(\theta_{s,y})$$

$$\Longrightarrow \frac{\mathbf{dy_b}(\boldsymbol{t})^2}{\mathbf{d^2t}} = \frac{\mathbf{mgR^2}\boldsymbol{L_s}}{(\mathbf{mR^2+J})\boldsymbol{L_y}}\mathbf{sin}(\boldsymbol{\theta_{s,y}}) \rightarrow (\mathbf{15})$$

a) Linearization around operating point

Linearizing the nonlinear equations about the operating point $x = 0\,, y = 0$ for small angles

$$\sin(\theta_{s,x}) \approx \theta_x\,, \cos(\theta_{s,x}) \approx 1$$

$$\sin(\theta_{s,y}) \approx \theta_y\,, \cos(\theta_{s,y}) \approx 1$$

Equations (12) and (13) are re-written as

$$\frac{\mathrm{dx_b}(t)^2}{\mathrm{d^2t}} = \frac{m\mathrm{g}\mathrm{R}^2L_s}{(\mathrm{m}R^2+J)L_x}\theta_x$$

$$\frac{\mathrm{dy_b}(t)^2}{\mathrm{d^2t}} = \frac{\mathrm{mgR^2}L_s}{(\mathrm{mR^2+J})L_y}\theta_y$$

b) Laplace transform of Equations

$$\frac{\mathrm{dx_b}(t)^2}{\mathrm{d^2t}} = \frac{m\mathrm{g}\mathrm{R}^2L_s}{(\mathrm{mR^2+J})L_x}\theta_x$$

$$\Longrightarrow\ s^2X_b(s) = \frac{mg\mathrm{R}^2L_s}{(\mathrm{mR^2+J})L_x}\theta_x(s)$$

$$\frac{\mathrm{dy_b}(t)^2}{\mathrm{d^2t}} = \frac{\mathrm{mgR^2}L_s}{(\mathrm{mR^2+J})L_y}\theta_y$$

$$\Longrightarrow\ s^2Y_b(s) = \frac{\mathrm{mgR^2}L_s}{(\mathrm{mR^2+J})L_y}\theta_y(s)$$

c) Obtaining Transfer Function of the system

$$s^2X_b(s) = \frac{mgR^2L_s}{(mR^2+J)L_x}\theta_x(s)$$

$$\Rightarrow X_b(s) = \frac{\frac{mgR^2L_s}{(mR^2+J)L_x}}{s^2}\theta_x(s)$$

$$s^2Y_b(s) = \frac{mgR^2L_s}{(mR^2+J)L_y}\theta_y(s)$$

$$\boldsymbol{\Rightarrow Y_b(s) = \frac{\frac{mgR^2L_s}{(mR^2+J)L_y}}{s^2}\theta_y(s)}$$

Transfer function of system

$$\boldsymbol{G_x(s) = \frac{X_b(s)}{\theta_x(s)} = \frac{\frac{mgR^2L_s}{(mR^2+J)L_y}}{s^2}} \qquad \textbf{(16)}$$

$$\boldsymbol{G_y(s) = \frac{Y_b(s)}{\theta_y(s)} = \frac{\frac{mgR^2L_s}{(mR^2+J)L_y}}{s^2}} \qquad \textbf{(17)}$$

D. CAD Design

For technical drawing in the project, Solidworks application is used. Design process started with calculations, planning, and drawing sketches. As it was declared in the instructions the plate (*figure 4*) is desgined according to the upper limit which is 300mm x 300mm for relatively easier control operations with ratioed dimension of other parts since the transfer function changes according to them. After the plate, the desired connection locations for plate arms and universal joint were extruded by 3mm.

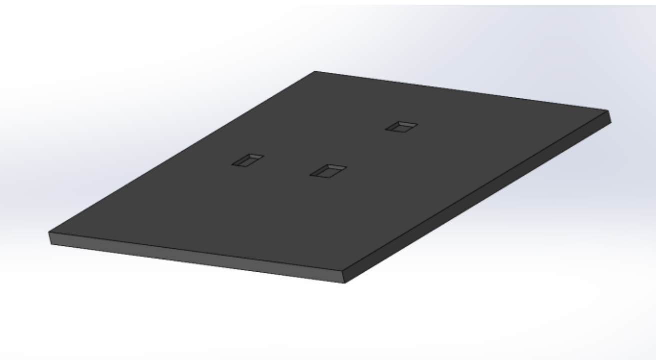

Figure 4: The Plate

After the inital top plate design completed, a base (*figure 5*) for the servo motor and bottom half of the universal joint connections designed. The base had designed to be bigger than the top plate in order to make the placement of the servo motors more flexible for possible dimension changes. The first approach for the servo motor selection was importing them externally since many other project have already utilized servo motors but during the importation the team had some combability errors between the imported part and the CAD program. In order to solve this problem, the servo motors (figure 5) are designed manually.

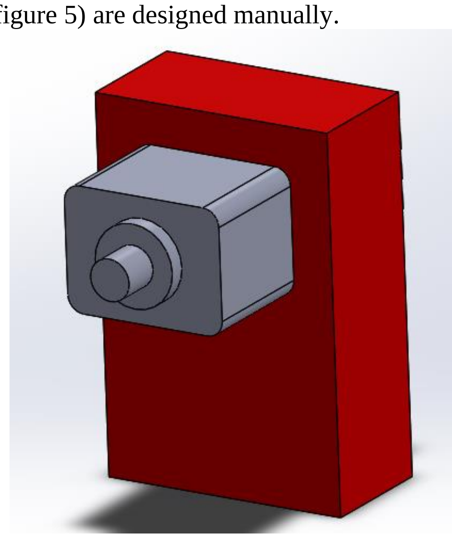

Figure 5 : Servo Motor

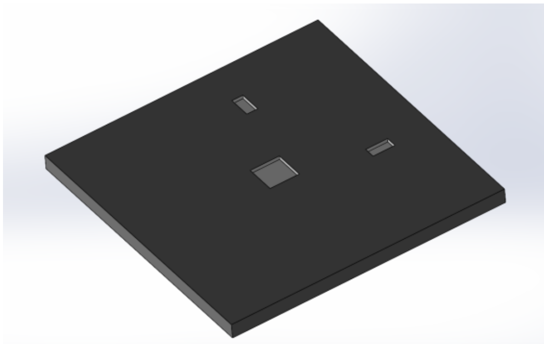

Figure 6: The Base

After finishing the plate and base, the design process of the universal joint(figüre 8) and arms to connect it (figüre 7, figüre 9) between plate and base has been initiated. This process was one of the most challenging parts of the technical drawing phase for many aspects encountered. In order to overcome this challenge the project had been worked on several different sketches to find an optimal option that fits best for the main design of the plant. Ultimately the decision was choosing a design that prioritized simplicity without compromising robustness.

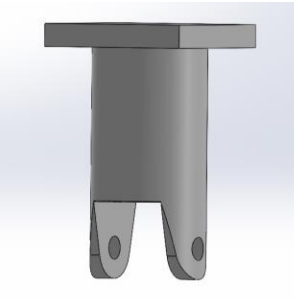

Figure 7: universal joint top

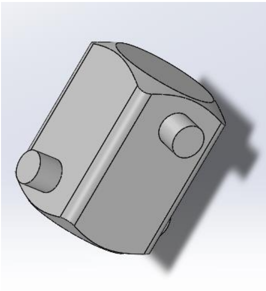

Figure 8: Universal Joint middle

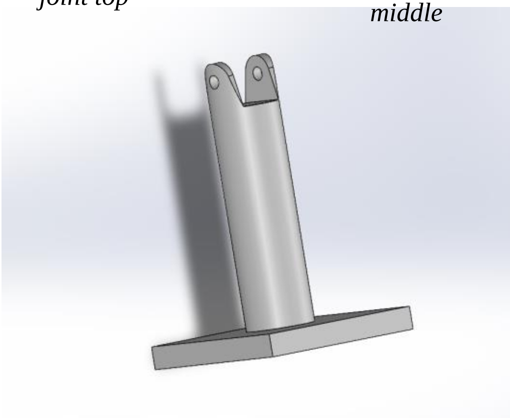

Figure 9: Univesial Joint Bottom

After completing the Universal joint, the first approach for the servo links was to use a revolute joint since the references that this project has built upon utilized revolute joints (*figure 10*, *figure 11*) for connections because they were simpler and directly effected the specific points in the plate that needed to be controlled. However as project continued, the revolute joints had caused simulation errors in MATLAB and error messages suggested that DOF in the joints should be increased so according to this suggestion, the revolute joints are replaced with ball joints (*figure 12, figure 13, figure 14*) and problem had solved.

Figure 10: Revolute Plate Arm

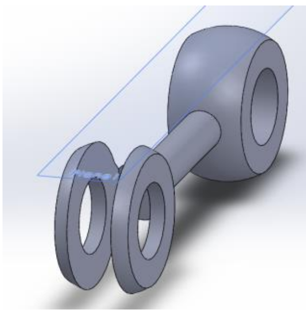

Figure 11: Revolute Servo Link

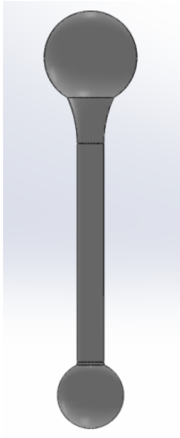

Figure 12: spherical Servo Link

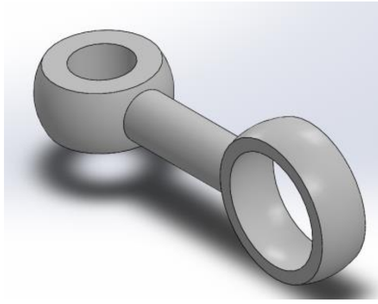

Figure 13: Spherical Servo Link

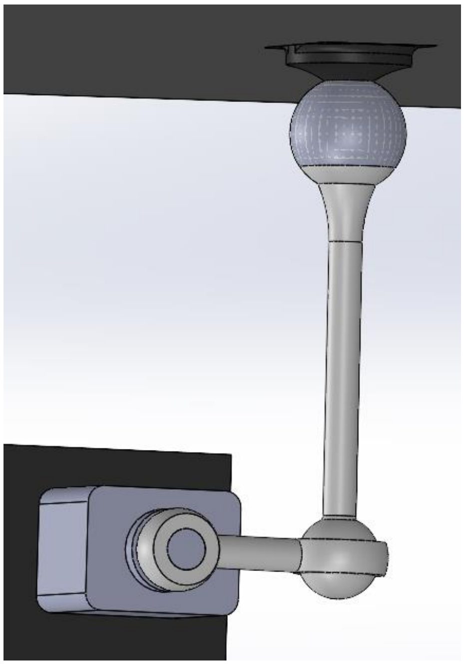

Figure 14: Servo Link and Plate Arm final Connection

During the process of designing the servo links and arms to establish the connection between the servo motor and the top plate (*figure 15)* , numerous calculations were made about height, length and distences relative to plate's center. This proved to be a complex task, given the spherical joints' diameters involved additional computations since they affected the joint movement. The plate arms' distance is the most crutial aspect of the assembly process because it directly affects the transfer funtion. 52.5 mm length for the distance between plate arm's center and plate's center is chosen as final dimension for further calculations. The plate arms are connected to the plate with an additional part which is named connection port (*figure 16*) .

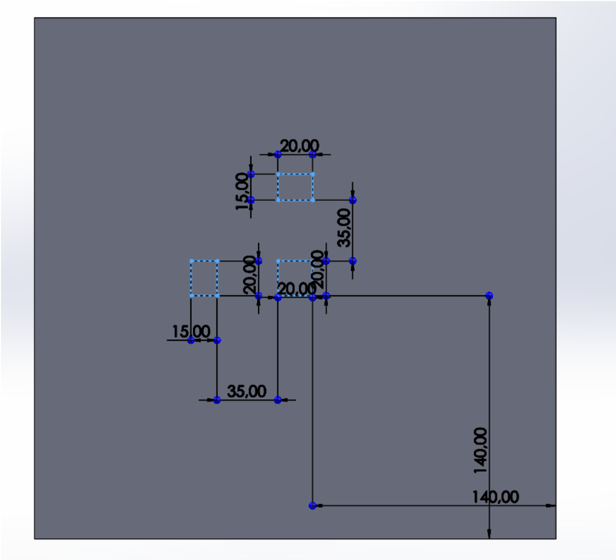

Figure 15: Dimensions of Displacement

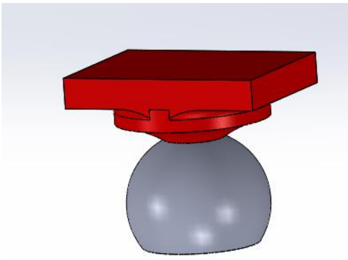

Figure 16: Connection Port

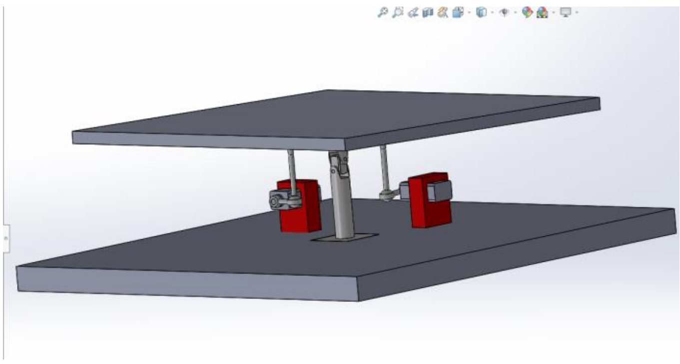

Figure 17 :Final Assembly (Spherical)

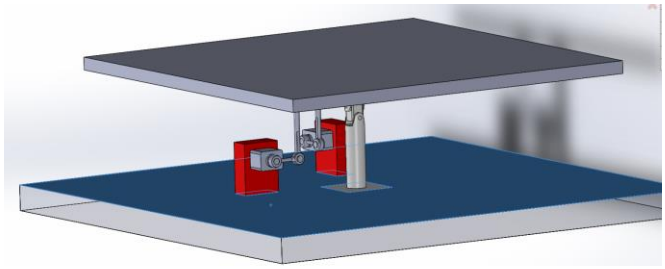

Figure 18: Final Assembly (Revolute)

E. PID Controller Design

In order to stabilize the system, a PID controller is modelled and built on Simulink with the block diagrams.

Parameter Values

Parameter Symbol	Value	Units
L_x	52.5	[mm]
L_y	52.5	[mm]
L_s		[mm]
r	12.5	[mm]
m	0.065	[kg]
J	0.00001015625	[kg*m^2]
g	9.81	[m/s^2]
R_a	4	[ohm]
L_a	0.00275	[H]
J_s	0.0032284	[kg*m^2]
b_s	0.0035077	[N*s/m^2]
K	0.0274	-

SolidWorks Material List was utilized in order to calculate the values of the ball. AISI 321 Annealed Stainless Steel is the material of choice for the ball.

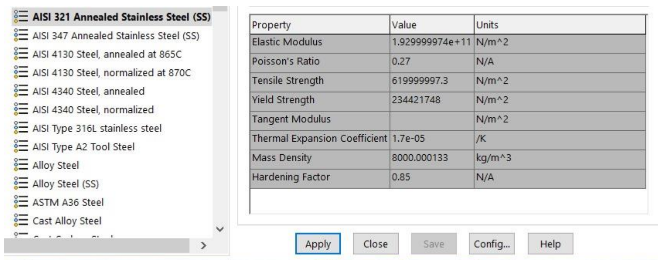

Fig. 19 SolidWorks Material List

Servo motor's parameter values are taken from Mechatronics System Design Project Report- Ball on Plate submitted to Lebanese American University.[2] Servo motor of selection is

MG995 High Speed Metal Gear Dual Ball Bearing Servo. Datasheet of MG995 can be accessed from the appendix.

Mathematical Substitution of Transfer Function

When the numerical values for parameters are substituted to the transfer function, the system's transfer functions become:

$$\boldsymbol{G_x(s)} = \frac{\frac{\mathbf{mgR^2}L_S}{(\mathbf{mR^2+J})L_y}}{s^2} = \frac{\mathbf{1.962}}{s^2} \qquad \mathbf{(18)}$$

$$\boldsymbol{G_y(s)} = \frac{\frac{\mathbf{mgR^2}L_S}{(\mathbf{mR^2+J})L_y}}{s^2} = \frac{\mathbf{1.962}}{s^2} \qquad \mathbf{(19)}$$

With the numerical values taken from Lebanese American University [2] to the servo motor transfer function(6), the servo motor transfer function becomes:

$$\boldsymbol{G_s(s)} = \frac{\boldsymbol{K}}{\boldsymbol{J_s L_a s^3 + (J_s R_a + L_a b_s)s^2 + (K^2 + R_a b_s)s}}$$

$$= \frac{\mathbf{0.0274}}{\mathbf{0.0000088781}\boldsymbol{s^3} + \mathbf{0.01292324618}\boldsymbol{s^2} + \mathbf{0.01478156}\boldsymbol{s}} \qquad \mathbf{(20)}$$

Goal of the PID Controller

The PID controller will be used to accomplish the aim of the system to control the table and balance the ball on the table's surface. This should be accomplished under 10 seconds. Performance measurements required for this goal are Settling Time, Percentage Overshoot, and Steady State Error.

Settling Time

Time it takes for the system's response to reach and stay within the specific error band around the desired value is settling time. Importance of the settling time is that the shorter it is, faster the system response will become. Settling time of this system is chosen to be 10 seconds.

Percentage Overshoot

The measure of how much the system's response exceeds the desired value is referred as percentage overshoot. Lower percentage overshoot indicates less oscillation of the system response.

Steady State Error

Steady state error is the difference between the actual output and the desired output when the system reaches a constant state.

Cascade Control System Block Diagram

Cascade Control System Block Diagram of the active ball balancing table consists of inner negative loop and an outer negative feedback loop.

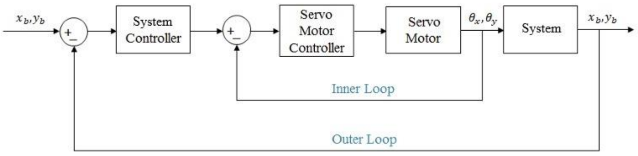

Fig. 20 Cascade Control System Block Diagram

Inner Loop of the System

Inner loop of the system consists of servo motor transfer function and servo motor controller. Input of the inner loop is position information of the ball and the output is the servo motor arm angular rotation which moves the system to balance the ball. Construction of PID of the inner loop will be done by utilizing Matlab and its Control System Designer tool.

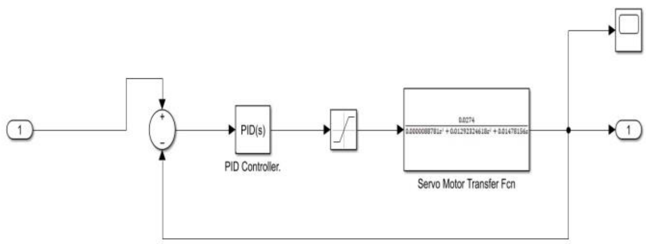

Fig 21. Initial Transfer Function of The System

The initial equations (*figure 21*) for the servo motor created in unwanted results with root locuses (*fig 22*) and stepsize (*fig 23*) of the servo motor. This situation caused the simulation to react uncontrollably to the servomotors' motion with transfer function's initial equations. As the project continued, in order to reach the s^2 on the denominator part of the servo motor transfer function, like that of the servo motor transfer function of reference [2], zeros added to the system using the Control System Designer of Matlab. According to this adjustment, Matlab provided Kp, Ki, and Kd values and with trial and error methods within the simulation environment, tuning Kp, Ki, and Kd values of PID controller manually, final equation (*figure 27*) was obtained.

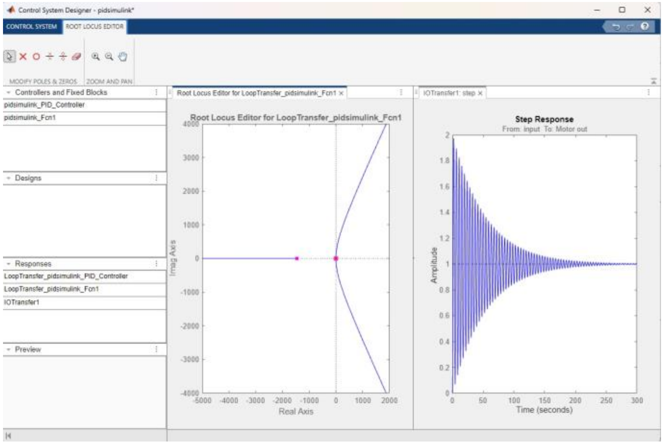

Fig 22. Root Locus

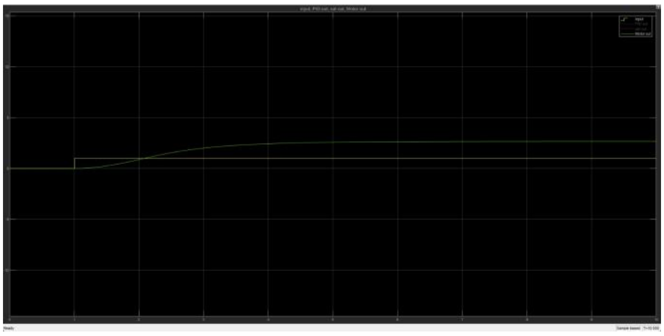

Fig 23. Step Size

IV. Results and Discussion

A. SIMULATION

When the SOLIDWORKS design is exported to the Simscape environment, a base block diagram *(figure 24)* is created automatically by the MATLAB. The ball which should be balanced right at the center of the plate is added to the system as solid component with a rigid transformation relative to the plate and predetermined parameters. Then it is placed near to the end points of the plate. In order to prevent the ball to pass through the plate, a 'spherical contact force' is added between plate and ball which can be found within the simscape's stock multibody library. For the ball to move realistically on the plate's surface, a 6DOF joint is added between plate and ball as well. In order to apply further feedback operations for the general controller and transfer functions to be utilized according to position data, a transform sensor is added between plate and ball. X and Y translations are pointed as ball_pos_x and ball_pos_y outputs with the usage of PS-simulink converters.The final steps inside the main plant of the simulink model is to give motion inputs to the revolute joints of the servomotors. In order to do this, two inport blocks are needed for the communication between actuation signal of the servo motors that will be obtained with negative feedback transformation functions and servomotor link. The inputs are named as servo_X_angle and servo_Y_angle. The motion inputs are needed to be activated within the revolute joint and torques should be set to automatically calculated. For the connection between de signal and the motion input segment of the joint, a simulink-PS converter is needed thus the final block diagram for the plant have been created (*figure 25)*

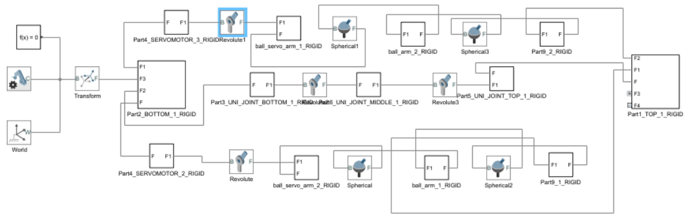

Figure 24. Base Block Diagram

Figure 25 . Final Plant

After completing the main plant and its general connections the next step that has been taken is constructing the feedback system (*figure 26)* according to the position data, transfer function of the servo and PID values. A negative feedback control system approach is utilized in this project. The system initialized with a 0 constant in order to prevent the unnecessary variables that can disturb the negative feedback summation operator. A "-1" gain added to both position output since if it was not added, the servo links would try to move backwards than intended. The position data of the ball relative to plate can be observed with a scope. These position data are sent to the summation operator for the servo_X and servo_Y segments to convert them into the desired motion inputs for the servo motors' revolute joint to be actuated in the desired conditions. The converted data which is motion outputs can also be observed with a scope.

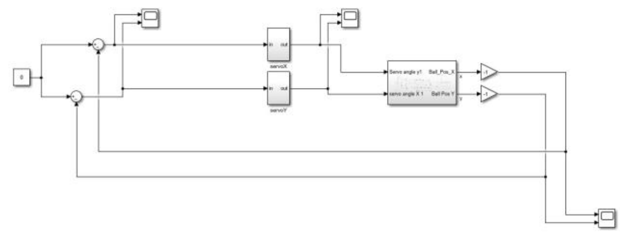

Figure 26. Main Feedback System

The transfer function, that is acquired from the theoretical calculations and with many trials and errors in the MATLAB calculations, is implemented in the servomotor section of the feedback system (*figure 27)*. 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. The servo motor block needs a PID controller within it, along with a positive feedback loop, to control it. A saturation block is also added to the system in order to limit the signal from overshooting and underdamping. After the construction the Simulink (*figure 26)* and Simscape (*figure 25)* diagrams the simulation (*figure 29*, *figure 30)* process has been started.

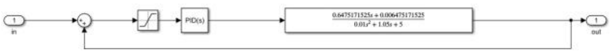

Figure 27 :Servo Motor Transfer Function and PID

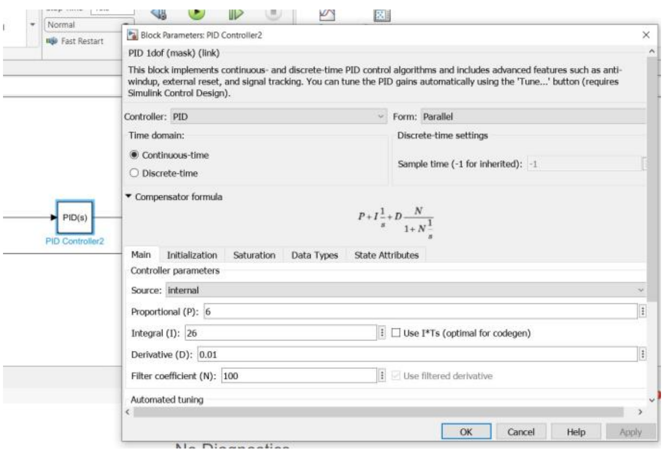

Figure 28 : Final PID values

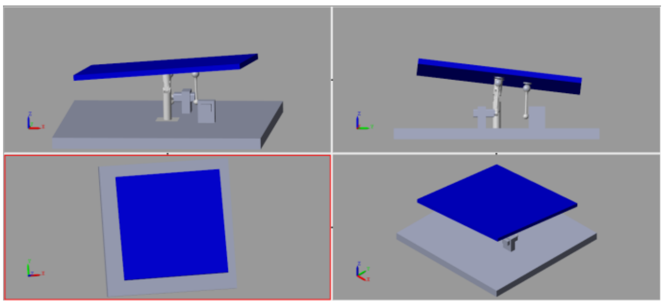

Figure 29 :Final Assembly (Spherical)

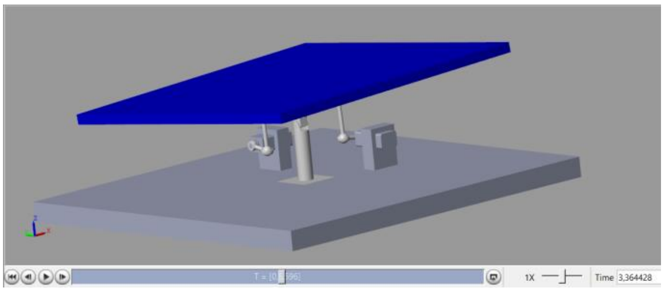

Figure 30 :Final Assembly (Spherical)

B. DISCUSSION

The first approach in this project was to utilize revolute joint (*figure 31)* between servo links and plate arms but if it were to apply motion inputs, an assembly error (*figure 32)* during the simulation process would be encountered which the team has solved this issue by replacing those revolute joints with ball joints.

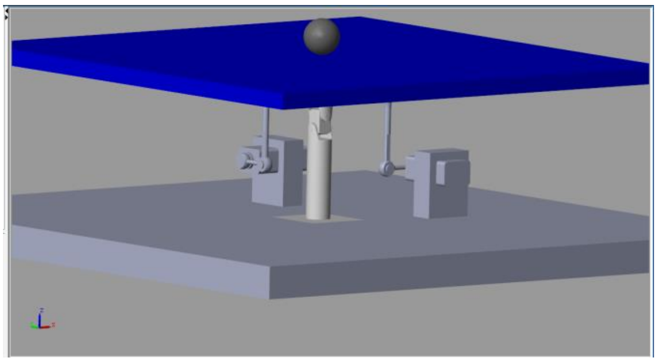

Figure 31 :Revolute Joint Simulation

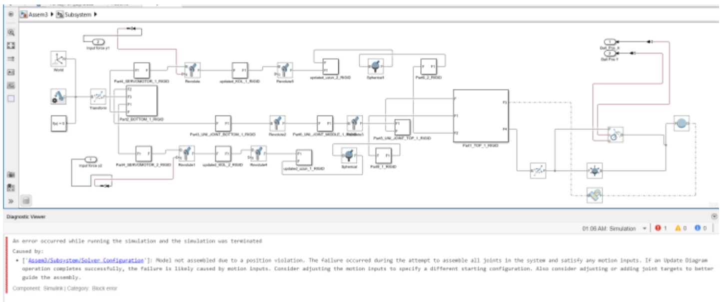

Figure 32 :Assembly Error

The PS in Simulink-PS converter stands for physical signal. Without a physical signal the joints can not be actuated since the rigid design is trying to mimic a real life representation in the simulation so it is necessary to add a Simulink-PS converter between the the joints and external inputs but the Simulink-PS converter can not be activated when it is directly connected, meaning an error would likely be encountered if a simulation is trying to be run without any configuration within the Simulink–PS converter which has happened during this project. The error message (*figure 33)* the MATLAB gives to the user suggest that second order filtering (*figure 34*) of the input and setting the derivative calculation as automatic (*figure 35)* would resolve this issue. During this experimentation, following the instructions which the MATLAB has given resolved the issue.

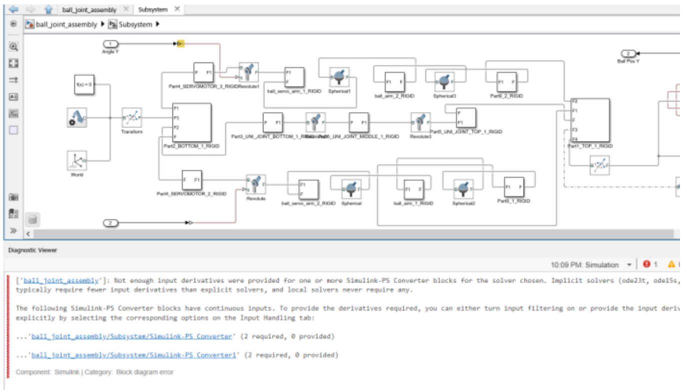

Figure 33 : Simulink-PS Converter Error Message

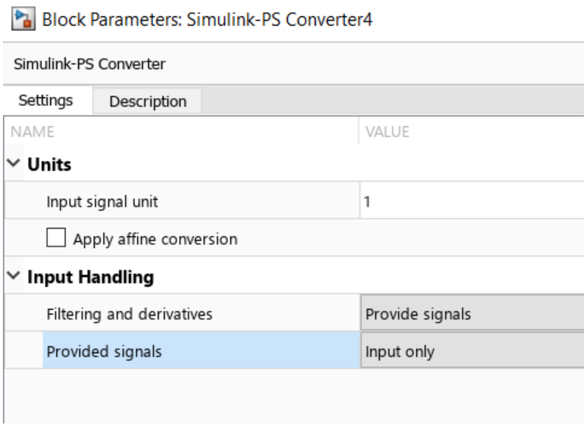

Figure 34 : Second Order Filtering

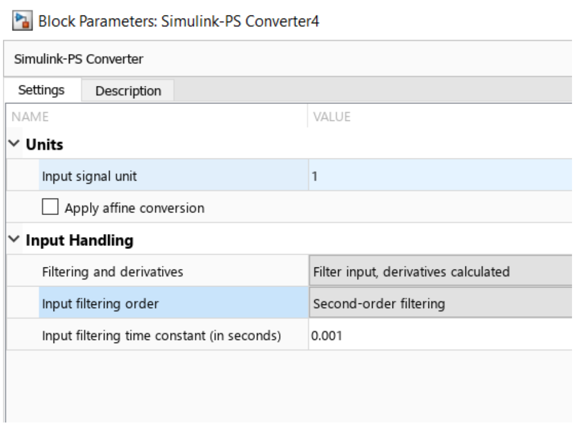

Figure 35 : Derivative Automation

After making the model with ball joint, the first idea that the team has come up with was creating the contact force between the ball and the plate with an external library named Simscape Multibody Contact Forces. This Simulink diagram's (*figure 36)* main difference from the main design is Simulink block diagram having an outport block which is connected to the 'out' port of the contact force. the out segment is connected to a GoTo block with a bus creator block. With the usage of bus creator, the GoTo block recieves many signals from the output port of the contact force such as normal force, friction force, penetration velocity but since only the position of the ball is concerend, the signal from bus creator were translated to 2 outputs (X and Y signals) with bus selector block mention earlier which is connected to a From Block. The purpose of the From Block in the diagram were to set a invisible connection between the signal connected to GoTo block and the bus slector by selecting the same name tag which is 'ballpos' in this project. The design (*figure 37)* is discarded due to many visual errors (*figure 38, figure39)* during the simulation process.

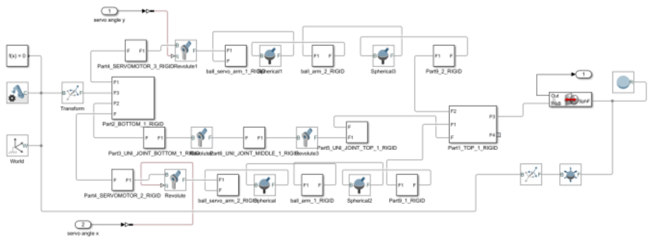

Figure 36: Simscape Multibody Contact Forces

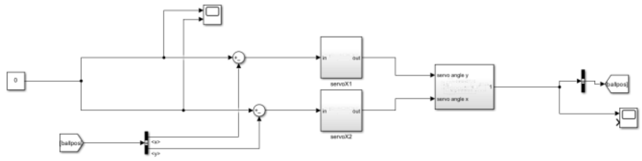

Figure 37: The Discarded Simulink Model

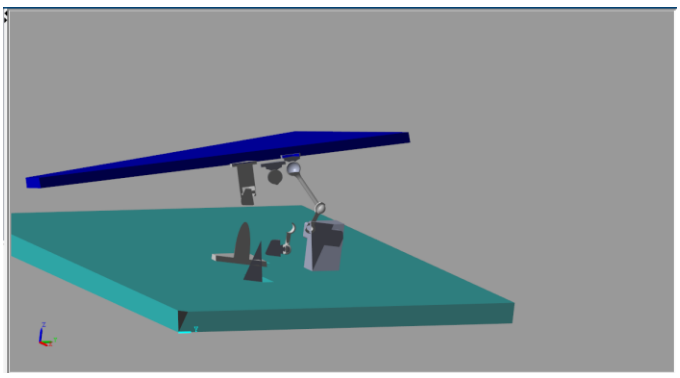

Figure 38: Visual Bug

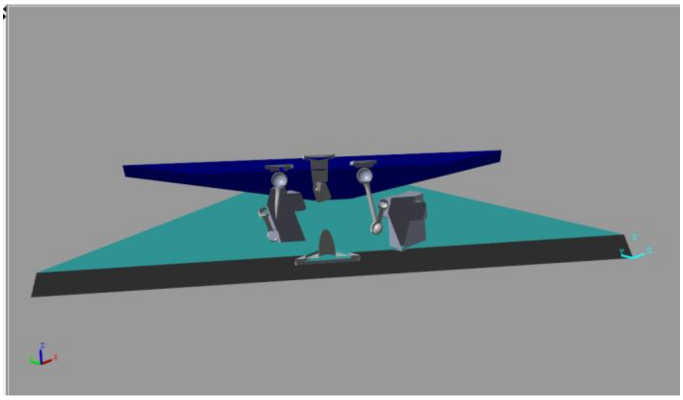

Figure 39: Visual Bug

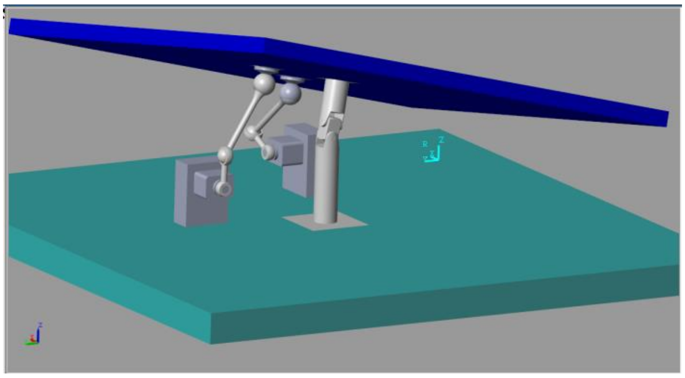

Figure 41: Constraintless Model

During the first stages of the second simulation model, the rigid transformation of the ball was relative to the world frame which was causing the simulation to end earlier than the settled time with the degenerate mass distribution error (*figure 40)*. In later stages the rigid transformation has been made relative to the plate's center of mass which resolved the issue.

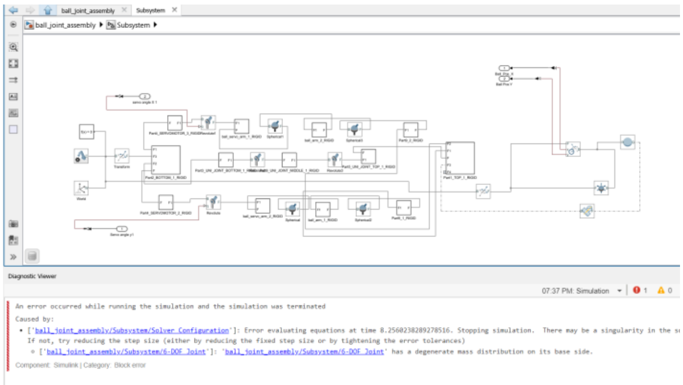

Figure 40:Degenerate Mass Distribution

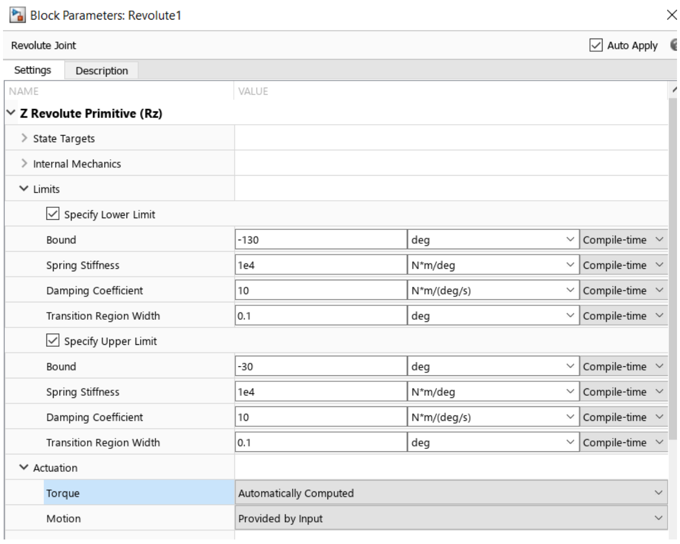

Figure 42: Joint Restrictions

Since the Simscape can not evaluate the joint constraint within the Solidworks it also can not simulate them, for example ball joints between servo links and plate arms are initially constrainless in the Simscape simulation which results in visually impossible movements in real life (*figure 41)*. In order to resolve this issue, some joint constraints are given to the revolute joint (*figure 42)* since constraining one joint stabilizes the other ones too.

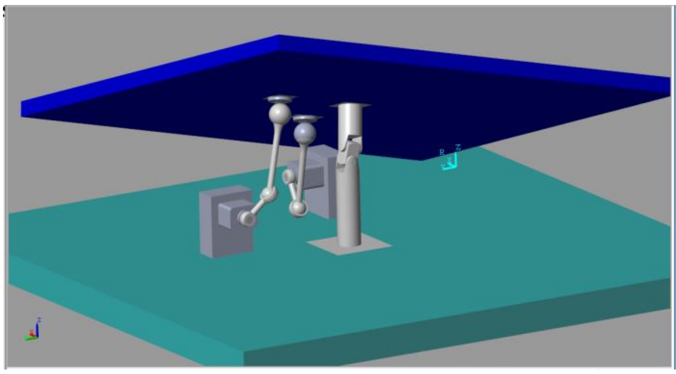

Figure 43: Constrained Model

The last problem that was encountered and couldn't been resolve was the initial position of one of the edges of the plate being too low than the desired position (*figure 44, figure 45)* and the model's inability to fully centralize the ball on the center of the plate. In the final stage of the project, the servo motors' actuations/ movements are relatively around the desired state but as stated before they are not fully capable of recovering the ball before it falls off the plate. Retunning the PID values or changing some gain coefficients would be a good approach in order to resolve this issue as these methods have been tried and resulted in some relative succes in the servo motors' reactiveness, more specifically retunning the P to 6 from 1.93830768402209, I to 26 from 19.0464862289854 and D to 0.01 from 0 in the PID controller have resulted in some relative increase in the speed of reactievenes of the motors but further arrangements have made the simulation to collapse/ break.

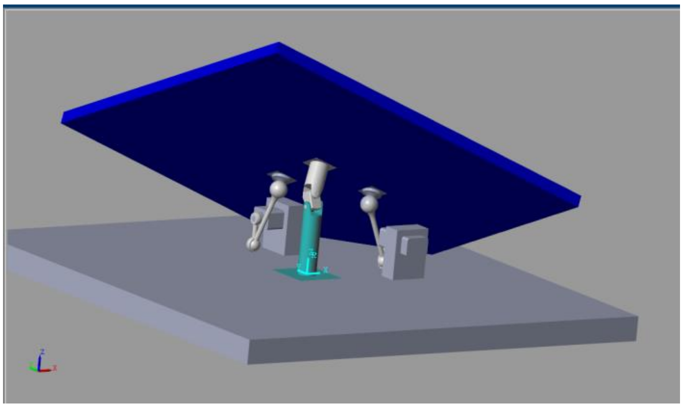

Figure 44: Initial Position

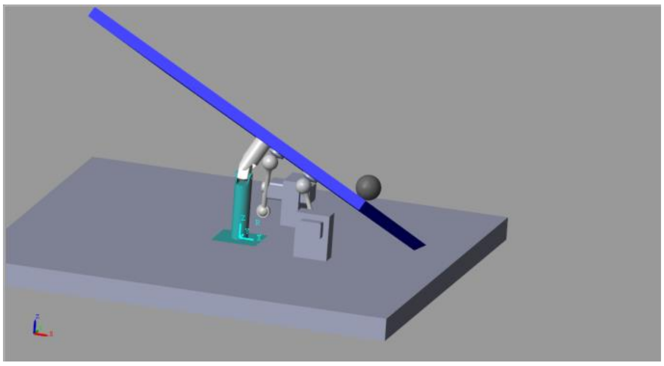

Figure 45: Initial Position

V. CONCLUSION

The project goal is stabilizing the thrown ball in the center of plate under 10 seconds with implementation of various sections that had to be completed in order to obtain the desired result. These sections included parts such as; theoretical calculations, technical drawings, controller design and simulation.

After doing extensive research and various calculations through project, there were many problems encountered with different parts with the sections. To overcome these problems, application of the various problem solving methods were implemented which were experimenting with the different parts to find optimal solution such as different joint parts, using trial and error and comparison of different models to find errors to remove them.

In order to eliminate the problems in the technical drawing, many sketches have been disposed and redone due to connection problems with joints' bottom and top parts. Since SOLIDWORKS had joint movements problems the design was reimagined and redrawn with spherical joint to overcome this problem.

By applying these different problem solving methods, technical drawing and theoretical calculations were completed successfully with no errors while the controller design was partially successful. Since the simulation operations advance concurrently with controller design this results in simulation also being partially successful.

When it comes to eliminating the problems in PID and simulation, by using Control System Designer of Matlab, with trial and error methods within the simulation environment and tuning Kp, Ki, and Kd values of PID controller manually, a partially successful system was established. Although the system did not completely work as desired, it partially succeeded. With further tuning of the PID controller, the system would completely work as desired.

After all these steps the system managed to operate partially for the desired goal, more specifically the ball managed to balance itself for 4 seconds however due to partially successful PID system it was not able to operate fully for the aimed purpose. Further tuning the PID system shows promise since increasing the P, I and D values resulted in stabilization during this project

APPENDIX

Appendixes should appear before the acknowledgment.

ACKNOWLEDGMENT

The preferred spelling of the word "acknowledgment" in America is without an "e" after the "g". Avoid the stilted expression, "One of us (R. B. G.) thanks . . ." Instead, try "R. B. G. thanks". Put sponsor acknowledgments in the unnumbered footnote on the first page.

REFERENCES

[1] A. H. Frank, M. Tjernström, Construction and theoretical study of a ball balancing platform. Limitations when stabilizing dynamic systems through implementation of automatic control theory (2019) [accessed Oct. 2023]
[2] J. El Khoury et al. Lebanese American University MCE411- Mechatronics System Design Project Report – Ball on a Plate (2021) [accessed Oct. 2023]
[3] Acrome Inc. Ball Balancing Table Info Sheet (2014) [accessed Oct. 2023]