Control Design and Simulation of a Ball Balancing Platform System

Control Engineering

David Isaí Zúñiga Rendón
A01378821
dept. Of Science and Engineering
TEC de Monterrey

David O'Neill Suárez
A01378537
dept. Of Science and Engineering
TEC de Monterrey

Santiago Vargas Loyola
A01025503
dept. Of Science and Engineering
TEC de Monterrey

***Abstract*— The theoretical construction and simulation of a ball balancing platform is created by using the *MATLAB* and *Simulink/Simscape* software tools. The mathematical model, CAD, Multibody, model validations and control design strategies are deconstructed, explained and developed in order to intuitively comprehend the inner workings of this project and understand the general concepts of control engineering.**

Keywords— MATLAB, Simulink, ball balancing platform, control engineering, mathematical modelling, joint, frame, block, axis.

I. NOMENCLATURE

Abbreviations

BBP- Ball Balancing Platform

FBD- Free Body Diagram

DOF - Degrees of Freedom

tf - Transfer function

Variables

$a_i[m/s^2]$ *- Absolute acceleration of the ball*

θ *[rad] - Angular position*

$\omega[rad/s]$ *- Angular velocity*

$\alpha_i[rad]$ *- Inclination of platform*

$F_{ri}[N]$ *- Friction on ball exerted by platform*

$I_b[kg.m^2]$ *- Mass moment of inertia of ball*

$m_b[kg]$ *- Mass of ball*

$g\ [m/s^2]$ *- Gravity*

$x_p[m]$ *- Relative x position to platform*

$y_p[m]$ *- Relative y position to platform*

τ *[N.m] - Torque*

$K_t[N.m/Amp]$ *- Motor torque constant*

$J\ [kg.m^2]$ *- Moment of inertia of the rotor*

b [N.m.s] - Motor viscous friction constant

V [volts] - Voltage

$I_a[Amps]$ *- Armature current*

L [H] - Electric inductance

$R\ [\Omega]$ *- Electric resistance*

$E_a[V]$ *- Induced voltage*

$K_b[V/rad/s]$ *- Electromotive force constant*

II. INTRODUCTION

Control Engineering has proven to be one of the most implemented technologies in the past century. Thanks to it, humanity has been able to modify and manipulate a wide number of systems to a point where they have become fully automatic; meaning: without the constant intervention of a user or a physical person. This has been done across many different sectors, from industrial processes, vehicles and energy distribution, all the way to a simple refrigerator or air conditioner.

In a few words, Control Engineering is the branch of engineering which deals with the principles of control theory. It is used to design systems that are capable of working on their own, without the assistance of any third party. For example, in an air conditioning system, there are two entries of information: the temperature of the room which is constantly being measured and the desired value of the temperature set by the user that must be reached. Now, taking into account the behaviour of the mechanical, electrical and electronic elements, a controller is designed and subsequently built. Then, the difference between the set value and the measured value is measured *(error)* and is fed back to this controller which in theory, generates a proper output signal as a consequence.

Behind every control system, there are a number of intricacies and complexities that manifest themselves generally in the dynamic systems that, more often than not, are very difficult to represent since they usually do not follow a linear behavior. Despite these complexities, the operational mathematics behind control theory allows us to represent complex systems of this nature through approximate models that are capable of stabilizing or modulating their behavior and obtaining a desired response.

The study of the control of systems that are naturally unstable is extremely important for solving many practical problems in real life. The actual testing and analysis of unstable systems turns out to be very difficult to execute in real laboratories *(such as position control of aerospace planes or suborbital rockets)* hence it is only possible to analyze such systems through approximate representations and models that can estimate the real behavior. The *Ball Balancing Platform* system assimilates such real unstable systems and gives us an introduction to real control problems and applications. This fundamental purpose of control is the primary reason this project is being created. By utilizing an approximate mathematical model and designing an appropriate controller, the theoretical study of a ball balancing setup can be used to generate an approximate conception of the real modulation and stabilization of a dynamic system. [2]

The main objective of this project is to control the position of a ball to a desired reference point on a plane and to additionally reject disturbances such as the introduction of external forces applied to the system. Although the actual construction and physical implementation of this project did not occur due to the social limitations of current events, the theoretical construction and simulation of the system are activities that allow us to intuitively understand the control strategies and their importance in real applications. This report will cover a *Theoretical Background* that will include basic knowledge about the project and its surroundings. Additionally, the *Activity Description* guides the reader through the entire process, which is divided in the necessary segments to achieve the objectives stated. After that, the *Results* are presented and discussed, finishing up with the *Conclusions* gathered throughout the project.

III. THEORETICAL BACKGROUND

In order to fully understand this project in its entirety, several concepts regarding mechanics, control engineering and control in general are introduced and explained in detail. This section is intended to help the general reader understand these essential notions and unfold the theoretical basics of the system that will be later studied and analyzed.

a) Dynamics

This branch of mechanics studies the relations between the movement and the causes that produce it. A body presents movement as a result of the interactions with other bodies that are described by forces. While the mass of an object is a measure by the resistance that occurs when the object's velocity changes, the weight is the gravitational force of the Earth over a body. [9]

In dynamics, the different types of movement of rigid bodies can be grouped in the following way with its specific equations:

1. Translation: A movement will be of translation if every straight line inside the body maintains the same direction during the movement. If two particles of the same body are named A and B, respectively, the velocity and acceleration of each of particles relate as shown [1]:

$$V_B = V_A$$

$$a_B = a_A$$

2. Rotation around a fixed axis: The particles that form the body move in parallel planes along the centered circles. If this axis of rotation intersects the rigid body, the particles localized on the axis have a velocity and acceleration equal to cero. The linear velocity of a particle results in the vector product of the angular velocity and the position vector [1]:

$$v = \omega \times r$$

Now, the equation for the linear acceleration results in the following, **α** being the angular acceleration:

$$a = \alpha \times r + \omega \times (\omega \times r)$$

3. General plane movement: All the particles of the body move in the same plane. Any plane movement of a plate can be replaced by a defined translation given the movement of an arbitrary reference point *A* in a simultaneous rotation around *A*. The absolute velocity $\mathbf{v}_B$ of a particle *B* of the chain is obtained from the following equation [1]:

$$V_B = V_A + V_{B/A}$$

Where the relative velocity of two points in a rigid body is given by:

$$V_B = V_A + \omega\,\hat{k} \times r_{B/A}$$

The relative acceleration of a_B can be calculated with the following:

$$a_B = a_A + a_{B/A}$$

4. Movement around a fixed point: The tridimensional movement of a rigid body attached to a fixed point *O* [1].

5. General movement: Any movement of a rigid body that doesn't fit to any of the previous categories [1]. The following can be expressed:

$$a_B = a_A + \alpha \times r_{B/A} + \omega \times (\omega \times r_{B/A})$$

b) Linearization

A dynamic model consists in the creation of a group of equations that can relate the relevant variables of the process and that are capable of reproducing the same behaviour. There are different types of dynamic models such as the detailed models, that are generally used for simulations, and the linearized models, used for controllers design. The linear systems are represented by linear equations, while nonlinear systems by nonlinear equations. Both of these can be static or dynamic equations [10].

The linearization of a derivable function $f(x)$in a number $x_1$is the equation of the line tangent to the graph *f* in the point [10]:

$$y = f(x_1) + f'(x_1)(x - x_1)$$

The nonlinearity of a system implies that the application of the same signal in two different points of operation generates two different responses [10].

If the variation $(x - x_1)$ is small, the superior orden terms can be despised. So, for the linearization of $f(x)$it is enough to approximate from the first two terms of the series [10].

c) Frequency domain and Laplace transform

The *Laplace Transform* is a type of integral transformation that has the property of linearity and many other properties that make it very useful to solve linear problems with initial values. As shown in the following equation, *f(t)* is a defined function for $t \geq 0$. Now, the integral of this function times e^{-st} is the *Laplace transform* of *f(t)* [10].

$$\mathcal{L}\{f(t)\} = \int_0^\infty e^{-st} f(t)\, dt$$

d) Control Engineering

This branch of engineering is the one in charge of dealing with the principles of control theory. There are a number of concepts that are relevant to this branch, which definition is going to be presented as follows [11]:

- Controlled variables: Quantity or condition that is measured and controlled.
- Control signal: Magnitude that the controller modifies to affect the value of the controlled variable.
- Plant: Part of the system. Group of elements that operate all together and form the machine wanted to be controlled.
- Process: Progresive continuous marked by a series of gradual changes, one after the other, in a relatively fixed manner, moving towards an established result.
- Closed loop system: System that maintains a determined relation between the output and the reference input, comparing and using the difference as a measure of control [11].

A way of developing a closed loop control system is by using a *PID* controller, which stands for Proportional (P) Integral (I) Derivative (D) controller. Each of these members develops a crucial task for the control system. The proportional control gives an output to the controller that is proportional to the error. The integral control gives an output to the controller that is proportional to the accumulated error, which implies that it is a slow type of control. And the derivative control only acts when there is a change in the absolute value of the error [12].

IV. Activity Description

A seemingly simple system and its control analysis such as the one studied in this document, might appear as a trivial and relatively simple study at first, however the intricacies that make up such devices can add up very quickly and sometimes appear as a convoluted study. Therefore, the activities in this document are divided into five fundamental sections in order to give structure to the document and fully explain the ins and outs of this final project.

First, we dive into the definition of the ball balancing platform apparatus and deconstruct it into its essential elements that make up this particular system. The essential investigation of said system, its state of the art and the general contextualization of the project is described to create an overview of the objectives that are to be achieved. We then proceed to the mathematical characterization of the system, where the theoretical representation of its elements are described and analyzed in order to obtain the mathematical models of the *BBP* system. Then, the digital construction of

the system is created, by modeling the mechanical parts in CAD and bringing them into *"life"* with the Simulink-Simscape environment. After that, we describe the control design theory and the strategies taken to create an appropriate and successful control structure. Finally, we demonstrate the performance of our control strategy by simulating the system and obtaining the desired behaviors.

A. *Definition of the ball balancing platform system*

Generally speaking, a *Ball Balancing Platform system* consists of a set of parts that altogether are able to control the position of a ball by moving the platform a certain amount of degrees around two axis.

In a more specific sense, it counts with 7 main body parts. The first one, is the *Base*, this is the plate located at the bottom of the system, to which every other component is coupled to. A *Universal Joint* connects the *Base* and the *Platform*, letting it move only with two *DOF*. Then, comes the *Platform*, which will always remain in contact with the ball, rotating in order to maintain the ball in the desired position. This component surface has 200x200 mm dimensions, while the steel ball has ¾ " of diameter. In addition, the servo motors will be located over the *Base* and under the *Platform*, these two motors will be 12V direct current motors, specifically the *Moog C23-L40 W10*. Now, for the union between the motors and the *Platform* a straight *coupler* was used, one for each motor. The complete assembly of an example of a *Ball Balancing Platform system* is shown on figure 1.

Fig. 1: Ball balancing platform.

Furthermore, for the control elements, there will be used *PID controllers*, due to our extensive experience with this type of control, we considered to be the most appropriate ones. To tackle the two control systems that need to be designed (*coordinate position of the bal x,y, angular position of the servos* $\theta_{1,2}$ *and the voltage applied to each servo* $V_{1,2}$), a *Cascade Control approach* was implemented. Placing two control loops, the inner one for the angle control of the servo motors with respect to the voltage entry, and the outer loop for the position of the ball with respect to the angle of the servo motors.

The objective desired to reach with respect to the establishment time is 3.5 s. Even though this time is arbitrary, we considered it to be an appropriate amount of time for the ball to stabilize in its place.

B. *Mathematical modeling*

The mathematical characterization of any dynamic system is the cornerstone of control design and control engineering. Obtaining the appropriate mathematical model is crucial to analyze and manipulate the system at hand. Due to how the *BBP* is constructed for this project, and how the actuating elements are incorporated into the system, the mathematical characterization of the system as a whole is divided into two sections. First we take a look at the purely mechanical elements of *BBP* which include the platform, the ball and the driving links of the motors. Then we analyze its electromechanical elements which are essentially the *x* and *y* DC Motors.

Before diving into the mathematical modelling of this ball balancing system, it must be stressed that such mathematical characterizations were not entirely derived by ourselves. Scholars and researchers alike have already explained much of the theory behind a similar system, hence the analysis presented in this section is heavily supported by research papers that are cited accordingly. With that being said, we first take a look at the mechanical portion of the *BBP*. This modelling is derived from the already well known Newtonian mechanics where such laws help express the system in a differential equation form. An accurate modelling allows for better system analysis and control design, therefore it is crucial to represent the *BBP* system in its most accurate yet simple form. To begin the analysis, several assumptions are established beforehand to simplify the level of analysis. [2]

- No slipping
- Perfectly spherical and homogeneous ball
- Ball is always touching the platform

Due to the three dimensional nature of our system, this platform can be deconstructed and simplified into a pair of two dimensional *"ball and beam"* systems, both of which consist of the same equations but with different variables due to their respective plane axes. This representation can be observed in the following figure.

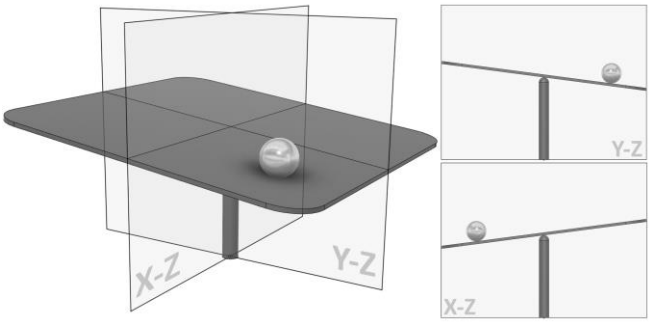


Fig. 2: 3D system represented as a pair of 2D systems.

To derive the expressions that dictate the behavior, the equations of motion through Newtonian Mechanics are applied. To visualize the ball's dynamics, free body diagrams are drawn where the main components are displayed. The constructed *FBDs* can be observed in the next two figures. It must be noted that this representation already includes the DC Motors, along with their driving links.

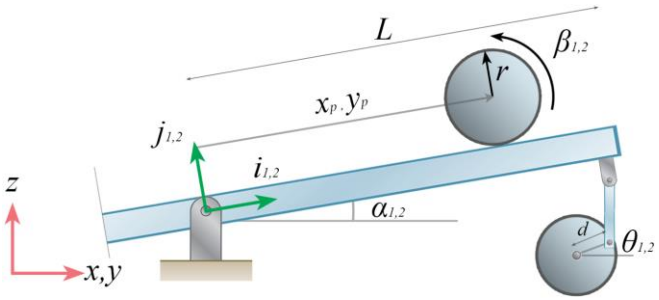


Fig. 3: Two dimensional FBD of the system

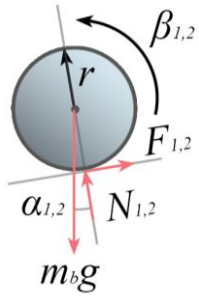


Fig. 4: Two dimensional FBD of the ball

For the purpose of brevity, the explanations between each step will be omitted, however the expressions used to get to the final differential equations are displayed in the mathematical procedure. As previously stated, Newtonian Mechanics were used and the dynamics of the ball along with the platform were established. All of these procedures, substitutions and simplifications can be observed next.

We analyze the relative acceleration of the ball.

$$\mathbf{a}_P = \dot{\mathbf{\Omega}} \times \mathbf{r} + \mathbf{\Omega} \times (\mathbf{\Omega} \times \mathbf{r}) + 2\mathbf{\Omega} \times (\dot{\mathbf{r}})_{Oxyz} + (\ddot{\mathbf{r}})_{Oxyz}$$

$$a = \dot{\omega} \times r + \omega \times (\omega \times r) + 2\omega \times v_{rel} + a_{rel}$$

$$a_1 = \ddot{\alpha_1} e_{k1} \times x_p e_{i1} + \dot{\alpha_1} e_{k1} \times (\dot{\alpha_1} e_{k1} \times x_p e_{i1}) + 2\dot{\alpha_1} e_{k1} \times \dot{x_p} e_{i1} + \ddot{x_p} e_{i1}$$

$$a_1 = \ddot{\alpha_1} x_p e_{j1} - \dot{\alpha_1}^2 x_p e_{i1} + 2\dot{\alpha_1} \dot{x_p} e_{j1} + \ddot{x_p} e_{i1}$$

$$a_1 = (\ddot{x_p} - x_p \dot{\alpha_1}^2) e_{i1} + (x_p \ddot{\alpha_1} + 2\dot{\alpha_1} \dot{x_p}) e_{j1}$$

We establish the equilibrium of torques on the ball from the *FBD* of figure 4.

$$I_b \ddot{\beta}_1 = F_{r1} r$$

$$\beta_1 = -\frac{x_p}{r}$$

$$\ddot{\beta}_1 = -\frac{\ddot{x}_p}{r}$$

$$F_{r1} = -\frac{I_b \ddot{x}_p}{r^2}$$

Now, the equilibrium of forces on the ball can be established through the FBD in figure 3.

$$\sum F_{i1} = m a_{i1}$$

$$F_{r1} - m_b g \sin \alpha_1 = m_b a_{i1}$$

$$m_b (\ddot{x}_p - x_b \dot{\alpha_1}^2) + \frac{I_b \ddot{x}_p}{r^2} + m_b g \sin \alpha_1 = 0$$

$$\left(\frac{I_b}{r^2} + m_b\right) \ddot{x}_p + m_b g \sin \alpha_1 - m_b x_b \dot{\alpha_1}^2 = 0$$

$$\ddot{x}_p = \frac{m_b r^2 (x_b \dot{\alpha_1}^2 - g \sin \alpha_1)}{I_b + m_b r^2}$$

The ball balancing system yields a second order differential equation, hence a linearization is in order to further simplify the analysis. This linearization considers only small deviations in the angle of the platform that are strictly close to cero ($\alpha_i \approx 0$). This way, we can linearize the system to the following expressions.

$$\ddot{x}_p = -\frac{m_b g \alpha_1}{\frac{I_b}{r^2} + m_b} \qquad (1)$$

From the *FBD* we can relate the angle of the platform *(α)* to the angle of the Motor *(θ)*. This relation is given by:

$$\alpha_{1,2} = \frac{d}{L}\theta_{1,2} \tag{2}$$

We substitute the relation in the linearized equation.

$$\ddot{x}_p = -\frac{m_b g \frac{d}{L}}{\frac{I_b}{r^2} + m_b}\theta_1 \tag{3}$$

Substituting the mass moment of inertia of the ball and simplifying. Note that the same procedure was done for the second system, where the respective equivalent expressions were used. Both the *x* and *y* differential equations are obtained.

$$\ddot{x}_p(t) = -\frac{5gd}{7L}\theta_1(t) \tag{4}$$

$$\ddot{y}_p(t) = -\frac{5gd}{7L}\theta_2(t) \tag{5}$$

Finally, we apply the *Laplace Transformation* and obtain the transfer functions of the *BBP* system. The resulting equations now within the frequency domain are shown below.

$$\frac{X(s)}{\theta_1(s)} = -\frac{1}{s^2}\cdot\frac{5gd}{7L} \tag{6}$$

$$\frac{Y(s)}{\theta_2(s)} = -\frac{1}{s^2}\cdot\frac{5gd}{7L} \tag{7}$$

Once obtained the mechanical modelling of *BBP* that relates the *(x, y)* position of the ball to the angular position of the motors (θ_1, θ_2), we turn our focus to the electromechanical portion of the system. We must find the expressions that relate those same angular positions (θ_1, θ_2) to the voltage applied to each motor (V_1, V_2), hence we must analyze those elements and obtain a mathematical model of the DC Motors. It is clear that our system has two control variables, this creates a system that is subdivided into two dependent subsystems, each of them with their own control and manipulated variables. Therefore, the physical setup of a DC motor is analyzed and the mathematical model that relates the angular position θ to the voltage *V* is obtained. To begin the modeling, the equivalent circuit of a fixed field *(permanent magnet DC motor)* is analyzed.

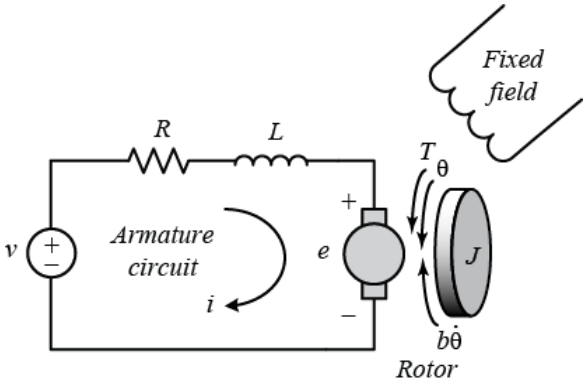


Fig. 5: Equivalent circuit of a fixed field DC motor [5]

The governing equations of the physical setup are the following.

$$V = L\frac{di_a}{dt} + Ri_a + E_a \tag{8}$$

$$\tau = J\ddot{\theta} + b\dot{\theta} \tag{9}$$

$$\tau = K_t i_a \tag{10}$$

$$E_a = K_b\dot{\theta} \tag{11}$$

We apply the Laplace transform to each equation:

$$V(s) = (Ls + R)I_a(s) + E_a(s) \tag{12}$$

$$\tau(s) = (Js^2 + bs)\theta(s) \tag{13}$$

$$\tau(s) = K_t I_a(s) \tag{14}$$

$$E_a(s) = K_b\dot{\theta}(s) \tag{15}$$

Calculating the complex derivative of equation *(13)*, substituting and simplifying.

$$\tau(s) = (Js + b)\dot{\theta}(s)$$

$$V(s) = \left(\frac{Ls + R}{K_t}\right)(Js + b)\dot{\theta}(s) + K_b\dot{\theta}(s)$$

Obtaining the transfer function of the angular velocity of the DC motor.

$$\frac{\dot{\theta}(s)}{V(s)} = \frac{K_t}{(Js + b)(Ls + R) + K_t K_b} \tag{16}$$

Neglecting the armature inductance to yield a first order *tf.*

$$\frac{\dot{\theta}(s)}{V(s)} = \frac{K}{\tau s + 1} \tag{19}$$

Where K and τ are the equivalent simplification expressions.

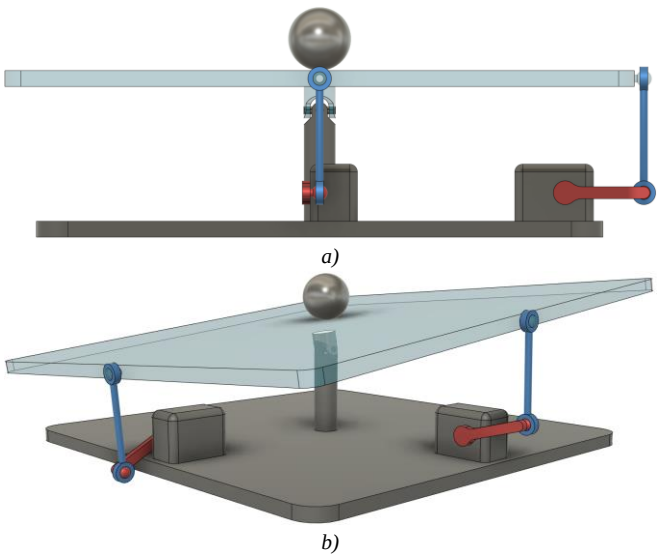
Finally, integrating the previous expressions and obtaining the *tf* of the angular position.

$$\frac{\theta(s)}{V(s)} = \frac{1}{s} \cdot \frac{\dot{\theta}(s)}{V(s)} \qquad (20)$$

C. CAD and Multibody

To create the 3D model and eventually implement the system through Simulink, the CAD software Fusion 360 was used to prototype ideas and test the most favourable *BBP* mechanisms. The fundamental elements of the system were modelled based on already existing *BBP* toy models, one of which was previously presented in figure 1. In that figure, the fundamental elements such as the platform, ball, driving links and coupler links can be observed. These main elements were modelled based on that figure and the *FBD* from figure 3 as well. Different joints to assemble the platform and the base were tested and after a couple iterations, a *Universal Joint* was used to create that particular linkage. This joint was the appropriate way of assembling both parts since it virtually restricts the system to two *DOF*.

By evaluating this idea and testing the motion, it was decided to work with this setup and create the final model. The elements of the entire system with the exception of the ball, were exported as STEP files and later used as the solids for the Simscape Multibody modelling. The final CAD model can be observed next.



a)

b)

Fig. 6: Final CAD Model

Materials were assigned to each element since the inertial properties are needed for the Multibody simulation. The materials assigned included common materials such as aluminum, steel and a touch glass for the platform. Fusion 360's library was used to sample the materials and assign the one that fits the most to the part. In the next table the mechanical properties of each element are shown where the dimensions, materials and mass are determined.

TABLE I
PROPERTIES OF THE *BBP* ELEMENTS

Part	Dimensions	Material	Mass
Platform	L = 100 mm	Glass	501.93 g
Driving link	d = 25 mm	Aluminum 6061	1.136 g
Ball	r = 10 mm	Steel	32.882 g
Coupler	l = 31 mm	Aluminum 6061	0.733 g

For the creation of the Simscape model a number of components were implemented for the mechanism to work properly, including a variety of joints, rigid transforms and forces. As per usual in a Multibody environment, the elemental blocks were added though a new Simscape Multibody model. This model was opened by typing the command *smnew* in the command window, displaying the fundamental blocks and a variable-step automatic solver which was later modified so that MATLAB could converge to a solution. These common blocks include the mandatory *Solver Configuration* block, the *World Frame* block which creates a ground frame for the physical model to be in, and the *Mechanism Configuration* block in which the gravitational force is specified.

To bring in the CAD models into Simulink, the block *File Solid* was used to import the STEP files previously exported. These Simscape blocks require the user to specify either a mass or density for the simulation to run, hence the calculation of masses for each element. The "*Base"* to which the rest of the parts are coupled to, is first added. Three *Frames* must be specified, making sure the *Z* axis are aligned according to the required joint. These *Frames* can be observed in the following figure.

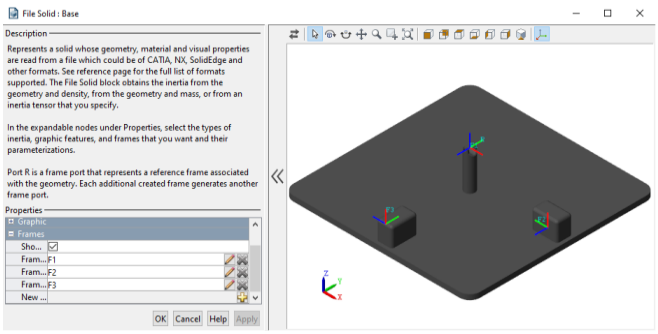


Fig. 7: Base and its established frames

Once the *Frames* are set and aligned, three connecting branches derived from the frames to which the next components are connected to. These components include

the *Universal Joint Peg*, which consists of two *Revolute Joints* due to the two *DOF* and both *x,y Driving Links.* The *Universal Joint Peg* is imported and the reference frames are established.

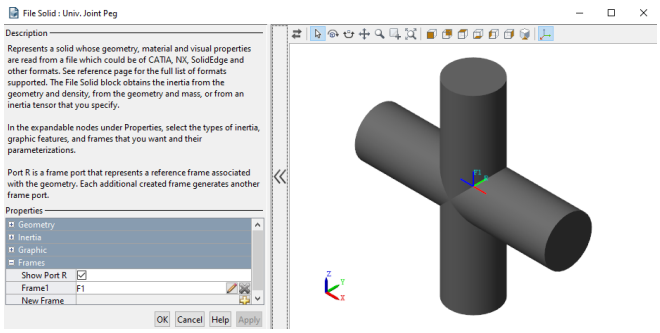


Fig. 8: Universal Joint Peg and its established frame

After that, the platform is added, coupled to the previous *Universal Joint Peg* with a respective *Frame*, however, two additional *Frames* are added in order to include que *x,y Couplers* of the motors. The imported component and its established frames are observed below.

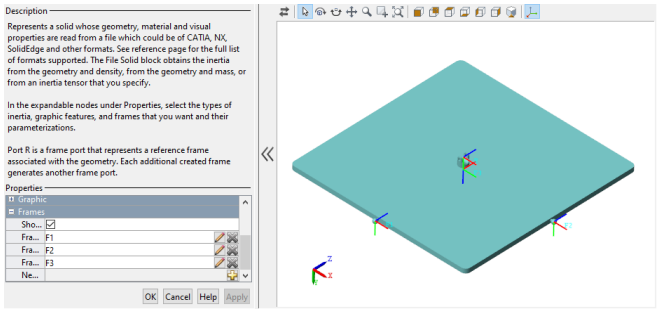


Fig. 9: Platform and its established frames

The branches generated by the *Frames* from the *Base* are used to add the *Driving Links* through *Revolute Joints* which will be in charge of moving the platform, in which the sensing of angular velocity and actuation by torque are activated. Once these *Revolutes* are set in place and configured, the *Driving Links* are imported and placed appropriately. The additional *Frames* are shown next.

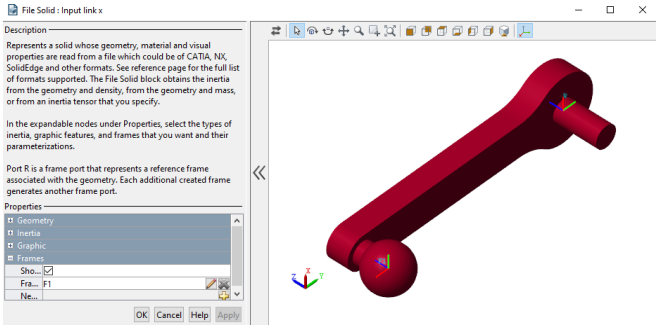


Fig. 10: Input Links and their established frames

These previous *Links* are followed by two *Couplers*, both of which are paired together by two *Spherical Joints*. The respective *Frames* for these *Couplers* are established and the parts are assembled together through the respective joints. The component and its frames can be observed in the next figure.

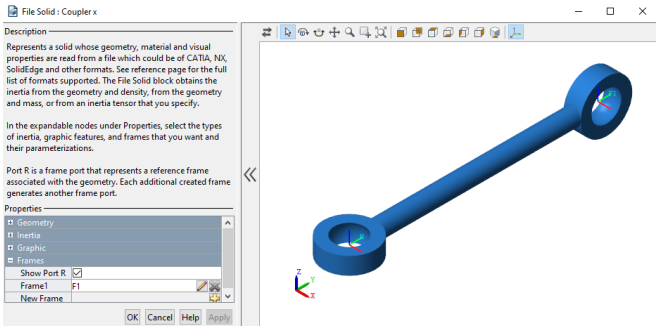


Fig. 11: Input Links and their established frames

Next, the *Ball* is created through a *Solid Block*, where the radius is determined. A rigid transform is used to translate the *Ball* to the correct position and a *6-DOF Joint* block is used to connect the *Platform* and the *Ball*. Additionally, a *Spatial Contact Force* block is added between these components in order to activate contact between them. This block lets us specify the physical properties between the two objects: frictional force, damping, stiffness, etc.

Finally, a *Transform Sensor* block between the *Platform* and the *Ball*. This block allows the tracking of the coordinate positions of the *Ball.* This block is critical since it will eventually be used to measure the output signals and use them as the feedback for the controller. The final Simscape Multibody model can be observed in the next figures.

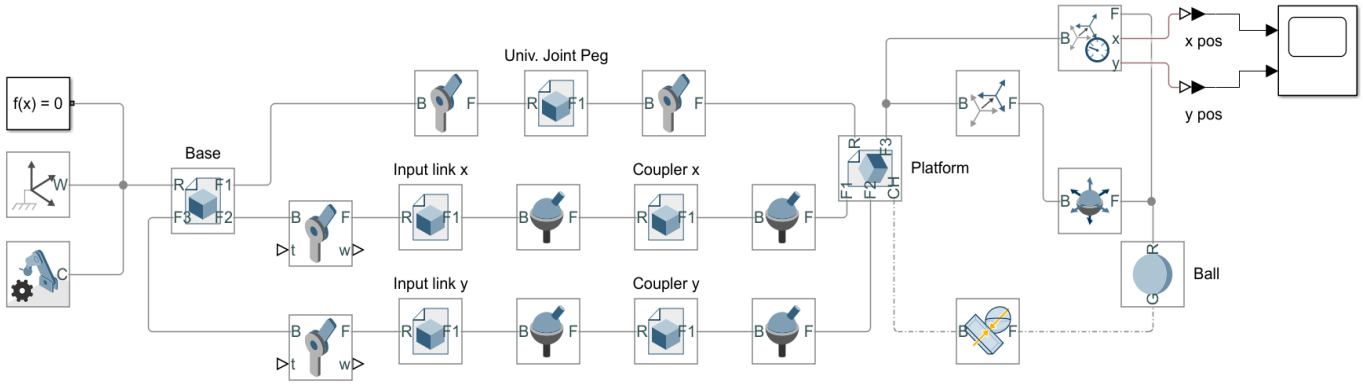


Fig. 12: Simscape Multibody blocking

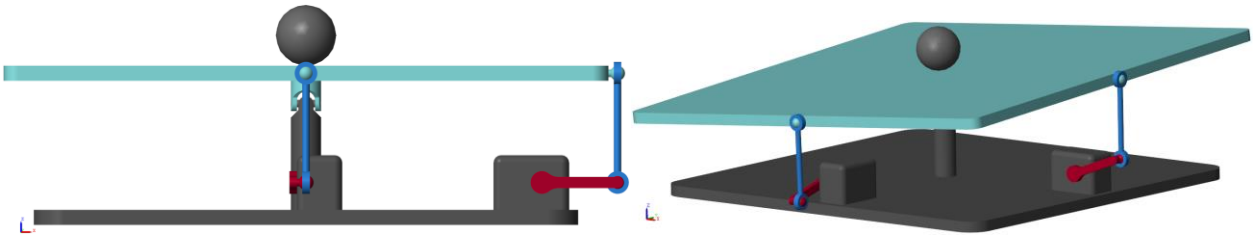


Fig. 13: Simscape Multibody assembly

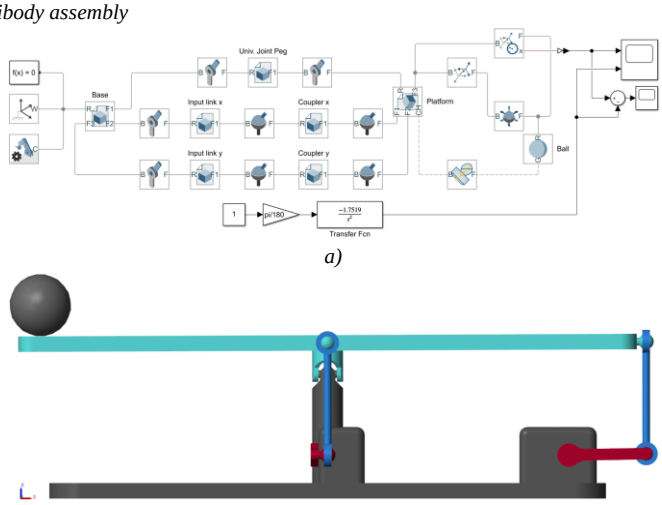
D. *Validations of the mathematical models*

The first mathematical model that was validated was the mechanical portion of the *BBP* system. With the Multibody assembly completed, this physical plant is validated with the transfer function obtained during the mathematical analysis. To do this, the values of the variables *L* and *d* from Table I are substituted in the expression that describes the plant model *(see equation 6 and 7)* and the final transfer functions are obtained. These final functions are shown below.

$$\frac{X(s)}{\theta_1(s)} = -\frac{1.7519}{s^2} \qquad (21)$$

$$\frac{Y(s)}{\theta_2(s)} = -\frac{1.7519}{s^2} \qquad (22)$$

With the final *tf* and the *"real"* system assembled in Simulink-Multibody, the mathematical model is validated by making a comparison between the physical response of the Multibody assembly and the response of the *tf*. For this, an initial angle condition of 1° degree is established for the *x-Revolute Joint* coupled to the *x-driving link* and subsequently, an input of 1° is also introduced to the *tf*. The simulation is then executed for 2.5 seconds and the linear position curves of both systems and the error between the models are observed. This validation is shown in the results section. The Simulink block diagram with which this validation was carried out is shown in the following figure.



a)

b)

Fig. 14: BBP system validation

Now that the mechanical portion of the system has been validated, we jump to the validation of the DC Motor used. The scientific literature has validated the first order behaviour of a *DC Motor* a number of times already, so in reality there would be no need to validate the model for this project, however we believe it's important to show how the *"real"* system or at least its Simscape counterpart, behaves similar to the first order *tf* approximation. To validate the *DC Motor*, first a real device needs to be chosen and its properties must then be introduced into Simscape. The appropriate *DC Motor* that was chosen can be observed in the following figure, which corresponds to a *Permanent Magnet DC Motor C23-L40 10*. The parametric values can be observed in the table below. *[13]*

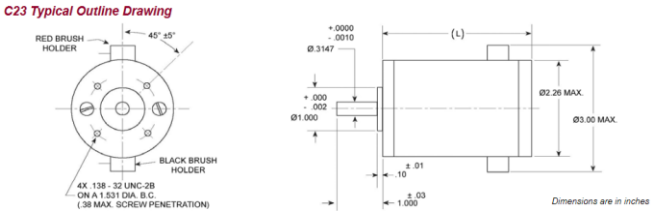


Fig. 15: Permanent Magnet DC Motor C23-L40 10

TABLE II
DC MOTOR C23-40 10 PARAMETRIC PROPERTIES

Armature resistance	0.7 Ohm
Armature inductance	0.5 mH
Back-emf constant	0.0342 V/(rad/s)
Rotor Inertia	282.5 g*cm^2
Rated Voltage	12v - 24v
Rated Current	4.9 A

All of these parameters are introduced into Simscape's electrical DC Motor equivalence. A DC source is fed to the DC motor with a terminal voltage or 12v, a *Translational Sensor* is used to record the velocity while a *Voltage Sensor* is used to record the voltage throughout the simulation. Two *ToWorkspace* blocks are used to export the values to MATLAB so that the *System Identification* tool could be used. The Solver *ode14x* is defined for the model settings and the simulation is executed for 0.4 seconds. The validation circuit in Simulink can be observed in the following figure.

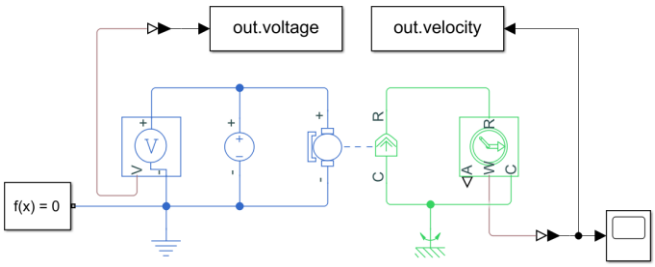


Fig. 16: DC Motor Validation circuit

Once the simulation is executed and the values are recorded in the MATLAB workspace, the command "s*ystemIdentification"* is typed in the command window to open the *SystemID* tool. The time domain data is imported and the *input*, *output* and *sample time* boxes are filled in. Please note that the *sample time* is calculated by dividing the simulation time over the number of data values. With these values introduced, the data is imported into the *SystemID* tool. This tool is shown in the next figure.

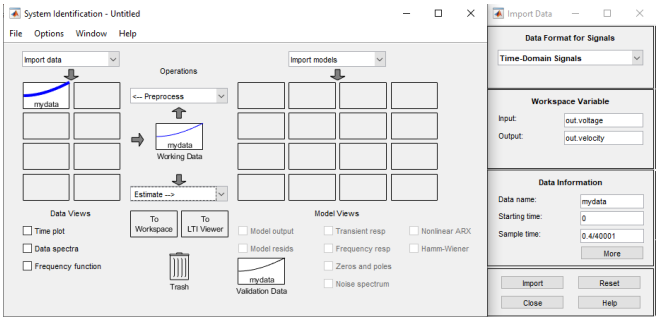


Fig.17: System Identification tool

Now that the data of the *"real" DC Motor* has been imported and visualized, a transfer function estimation is done. Since we essentially desire a first order *tf*, we set the *number of poles* to 1 and the *number of zeros* to 0. This estimation tool is shown in the next figure. The tool allows us to estimate a transfer function that assimilates the behaviour of the "real" *DC Motor*. The *Best Fits* and the comparison between both systems are shown in the results section.

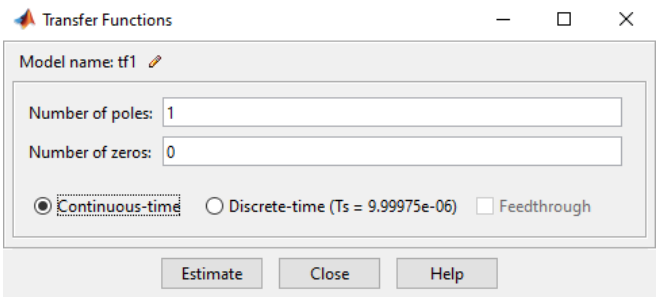


Fig. 18: Transfer function estimation tool

Now that the mathematical model has been validated, the estimated *tf* that can now be used for the control design aspect of the project. This *tf* assimilates the behaviour of the real Motor quite precisely. The *tf* is shown below.

$$\frac{\theta_{1,2}(s)}{V_{1,2}(s)} = \frac{1}{s} \cdot \frac{1796}{s + 61.41} \qquad (23)$$

E. Control Design

This section describes how the design of the control system was carried out. Based on the literature consulted, it was decided to use a cascade control scheme like the one shown in the following figure.

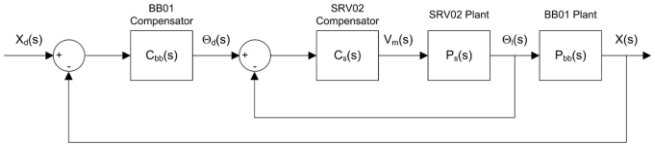


Fig. 19: Cascade Control scheme. [7][8].

As it can be seen, the internal loop corresponds to the motor control system while the external loop includes both the motor and the plant. Now, to obtain the gains of the controllers in a cascade arrangement there are various methodologies which vary in complexity and degree of accuracy. However, due to the time and knowledge constraints of the project, it was decided to proceed with the application of a method that requires a combination of mathematical calculations and heuristic adjustments.

This method consists of the following. First, the inner loop is analyzed using the *pole placement methodology*. The gains obtained by this method will be used to control the DC Motor. Because this loop was analyzed independently, the gains of the controller will be accurate and will not require further adjustments. Then, the outer loop will be analyzed, which also includes the inner loop. This can be observed in the following figure in which the inner loop *(simplified in a single block)* is outlined in royal blue and the outer loop in cyan.

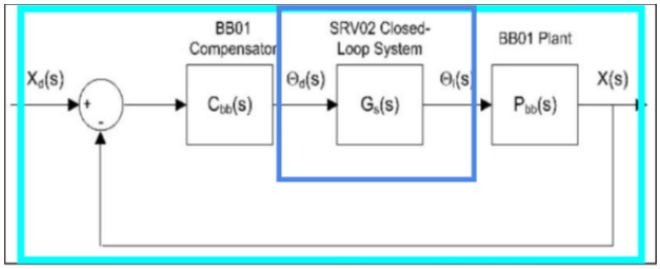


Fig. 20: Scheme showing the inner loop as a single block which is assumed to have an ideal behavior [7][8].

The last figure makes it clear that to obtain the gains using the *pole placement methodology* it is necessary to make some assumptions. In this case, it is considered that the closed loop control system corresponding to the motor has perfect behavior, that is, with a settling time very close to zero, without overshoot and without stationary error. Once this assumption is made, the inner subsystem can be ignored in order to analyze the plant alone. This methodology will give inaccurate results for the outer loop, but will provide a good starting point for further heuristic plant tuning.

The calculations performed to obtain the gains of the controllers using the aforementioned methodology are developed below. The type of controller that was used was the PID type since, if the model had to be built, this type of controller is the one that has the best ability to deal with disturbances and / or elements that are not considered in an idealized mathematical model.

The calculation of the inner loop controller gains begins with the transfer function that describes the motor, which was obtained using the System ID. The transfer function that describes the motor is shown below.

$$\frac{\theta(s)}{V(s)} = \frac{1796}{s^2 + 61.41s} \tag{24}$$

The desired settling time and overshoot values are now set.

$$M_P = 5\%$$

$$t_s = 0.2\ s$$

Subsequently, the characteristic equation that models the desired behavior of the system is obtained.

$$\xi = \frac{-\ln\left(\frac{M_P}{100}\right)}{\sqrt{\pi^2 + \ln\left(\frac{M_P}{100}\right)^2}} \quad ; \quad t_s = \frac{4}{\xi\ \omega_n} \tag{25}$$

Substituting the desired parameters and solving for ξ and ω_n we obtain

$$\xi = 0.6901\ ;\ \omega_n = 28.98$$

Substituting the values in the quadratic equation for harmonic motion, we have

$$s^2 + 2s\xi\omega_n + {\omega_n}^2 \tag{26}$$

$$s^2 + 40s + 839.84 \tag{27}$$

Since the closed-loop transfer function will be of third order, equation 27 is multiplied by a non-dominant pole, in this case by (s + 200), which results from multiplying its real root times 10. Therefore the desired polynomial results in the following expression.

$$s^3 + 240s^2 + 8839.84s + 167968 \tag{28}$$

Now we proceed to calculate the closed loop transfer function.

$$G_{LC} = \frac{\left(K_P + \frac{K_I}{s} + K_D s\right)\left(\frac{1796}{s^2 + 61.41s}\right)}{1 + \left(K_P + \frac{K_I}{s} + K_D s\right)\left(\frac{1796}{s^2 + 61.41s}\right)} \tag{29}$$

Simplifying

$$G_{LC} = \frac{1796\ (K_D s^2 + K_P s + K_I)}{s^3 + (1796K_D + 61.41)s^2 + (1796K_P)s + 1796K_I} \tag{30}$$

Finally, equating the denominator of Equation 30 and Equation 28 and solving for the gains, the following results are obtained for the motor's PID controller.

$$K_P = 4.922$$
$$K_I = 93.523$$
$$K_D = 0.0994$$

Now, considering the assumption discussed earlier in this section, we proceed to calculate the gains of the outer loop following the same procedure used for the inner loop. The desired settling time and overshoot values are first set.

$$M_P = 5\%$$
$$t_s = 3.5\ s$$

Subsequently, the characteristic equation that models the desired behavior of the system is obtained.

$$\xi = \frac{-\ln\left(\frac{M_P}{100}\right)}{\sqrt{\pi^2 + \ln\left(\frac{M_P}{100}\right)^2}} \quad ; \quad t_s = \frac{4}{\xi\ \omega_n} \tag{31}$$

Substituting the desired parameters and solving for ξ and ω_n we obtain

$$\xi = 0.6901\ ;\ \omega_n = 1.6561$$

Substituting the values in the quadratic equation for harmonic motion, we have

$$s^2 + 2.2858s + 2.743 \tag{32}$$

Since the closed-loop transfer function will be of third order, equation 32 is multiplied by a non-dominant pole, in this case by (s + 11.43) which results from multiplying the real root of equation 31 by 10.

$$s^3 + 13.7158s^2 + 28.8697s + 31.3525 \tag{33}$$

Now we proceed to calculate the closed loop transfer function.

$$G_{LC} = \frac{\left(K_P + \frac{K_I}{s} + K_D s\right)\left(\frac{1.75119}{s^2}\right)}{1 + \left(K_P + \frac{K_I}{s} + K_D s\right)\left(\frac{1.75119}{s^2}\right)} \tag{34}$$

Simplifying

$$G_{LC} = \frac{1.75119\ (K_D s^2 + K_P s + K_I)}{s^3 + (1.75119K_D)s^2 + (1.75119K_P)s + 1.75119K_I} \tag{35}$$

Finally, equating the denominator of Equation 35 and Equation 33 and solving for the gains, the following results are obtained for the motor's PID controller.

$$K_P = 16.486$$
$$K_I = 17.903$$
$$K_D = 7.83$$

Once the gain values for both controllers have been obtained, they are entered into MATLAB-Simulink to test the response of the system and make the necessary adjustments to the gains of the external loop controller.

F. Simulink Interface and HMI

Up to this point, we have analyzed all the components that make up the system in its entirety, hence we can begin the *"construction"* of the control system in Simulink. The entire control structure with the respective controllers and plants are introduced. Due to the way the PID controllers were designed, the predetermined PID blocks of Simulink cannot be used since they include the derivative filter *N*. This small filter affects the dynamics of the controlled system quite substantially, hence our own PID controllers must be constructed. This is by no means a complicated task, in fact is extremely easy, the only problem that arises when constructing our own controller structure is the manifestation of *algebraic loops*, which can cause severe problems when running the simulation. Essentially, *algebraic loops* can make it difficult for the simulation to converge to a single solution. To avoid this problem, the relative and absolute tolerances are increased to a suitable value. The PID controllers are shown below.

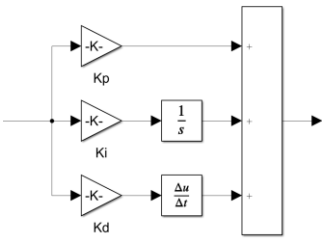


Fig. 21: Simulink self built PID controllers

Once the controllers are created, the entire system is constructed and organized along the previously consulted cascade control structure. To maintain order between each component in the Simulink environment, the *DC Motors*,

BBP plant and *PID controllers* were enclosed into different subsystems, this way a more clean and clear structure can be observed. where each element can easily be identified. This control structure and the completed system can be observed in the following figure.

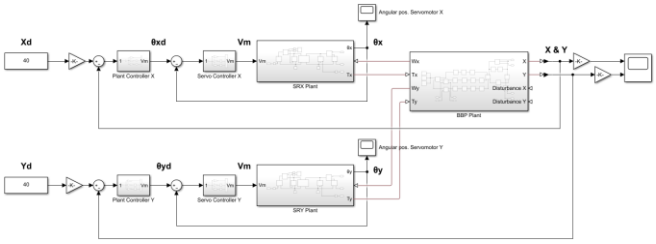


Fig. 22: Simulink Interface and control structure of the entire system

The control system is tested and the set-points are modified in order to troubleshoot any problems. Once this is done and the controlled behavior is cleared of any issues, the subsystem for disturbance induction is constructed along with the *HMI* for easy parameter manipulation. First, the disturbance subsystem is composed essentially of different *Pulse Generators* that activate at different times. These signals are connected to an *External Force Block* which is subsequently connected to a *Frame* attached to the steel *Ball*. This prompts a response in the *Ball* that assimilates a behavior along the lines of a finger briefly pushing into it. Finally, the *HMI* is constructed by using *Knobs*, *Switches* and *Dashboard Scopes* to facilitate the manipulation of the reference coordinates, the external loop *PID gains* and the introduction of disturbances. The final Simulink and *Human-Machine Interface* is shown in the next figure.

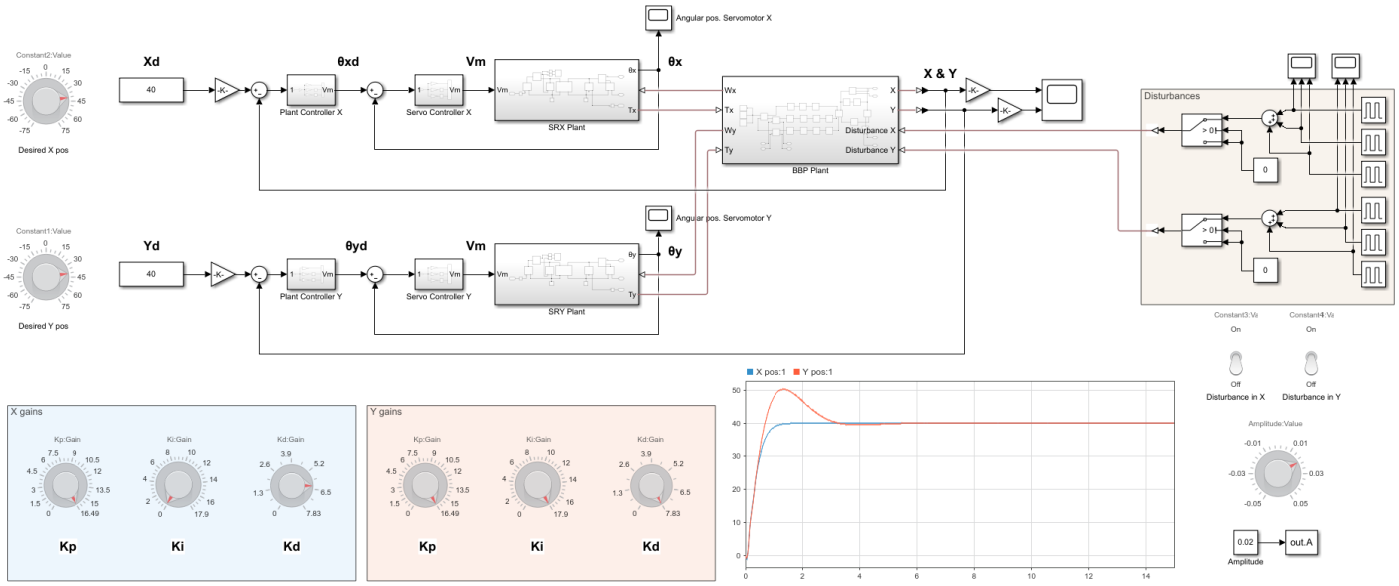
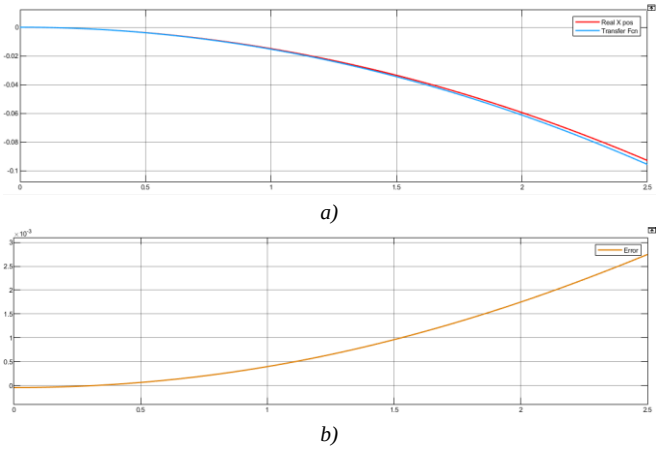


Fig. 23: Simulink control structure and HMI

V. RESULTS

In this section we are going to address the results obtained during the development of the project. Before jumping into the controlled behavior, we must first take a look at the validation of the mechanical and electrical portion of the system. First of all, the *BBP* system was validated according to the procedure described in the previous section. The next figure shows the results of this validation and the error between the "real" system and the system characterized by the transfer function.



a)

b)

Fig. 24: BBP system validation. a) "Real" and transfer function's linear position. b) Error between systems,

Once the *BBP* system is validated, the same validation results must be analyzed for the DC Motors. In order to do this, the *tf* of the motor is obtained using the *System ID* function of MATLAB, just as it was described in the previous section. In the next figure the estimation of the *tf* can be observed where a *Best Fit* of 99.13% is obtained and the overlapping of both the "real" Motor and *tf* can be appreciated. This is shown in the next figure.

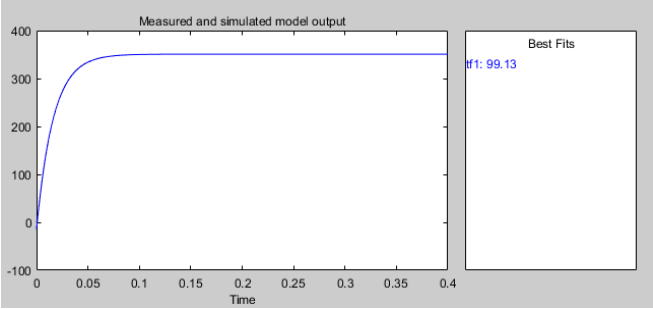


Fig. 25: System Identification tool. Best Fit of 99.13% for the FO tf.

Once the *tf* is obtained, the system is once again validated by comparing the outputs of the *tf* with the outputs of the "real" motor. In order to do this, the motor is supplied with 12 V and both the velocity and angular position of the motor and the *tf* are plotted side by side. The next figures show the circuit used to validate the motor and the outputs of velocity and angular position respectively.

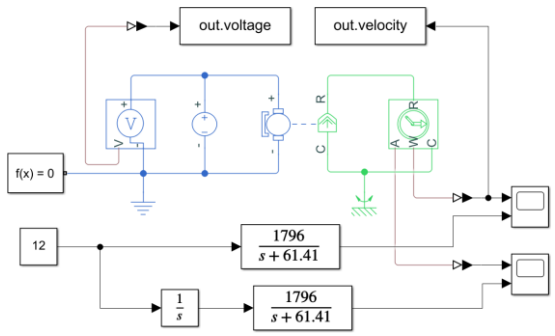
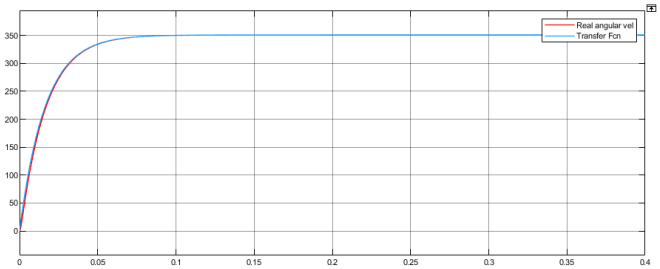
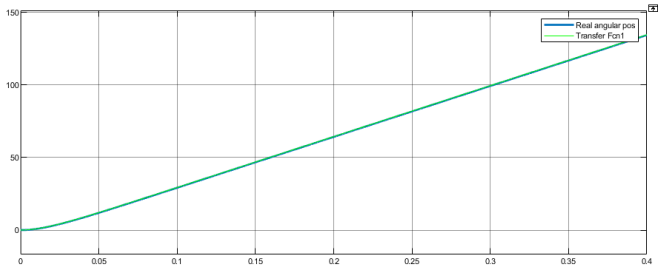


Fig. 26: DC Motor transfer function validation and comparison circuit.



a)



b)

Fig. 27: DC Motor tf validation. a) "Real" and tf angular velocity. b) "Real" and tf angular position

Now that both mathematical models have been compared and validated, the theoretical gains obtained for both PIDs in the previous section are inputted into the corresponding PID controllers and then tested. The next picture shows the response of the DC Motor alone after the implementation of the PID controller, where the settling time and the overshoots can be abbreviated.

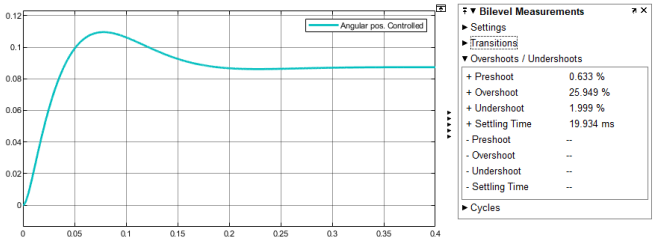


Fig. 28: DC Motor controlled response after PID controller implementation

Once it has been verified that the motor performs according to what was planned, the entire system is tested at different reference points in order to observe the general behavior and to discover any possible limiting conditions of the *BBP* system itself. Three tests were carried out, all of them with three cycles of disturbances. For the first one, a reference of x = 50mm and y = 50 mm was used. The second test included a reference of x = -60 mm and y = -30 mm. Finally for the third test, the limiting conditions were found with the theoretical *PID gains*, namely with the coordinates x = 75 mm and y = -75 mm. These tests are shown in the next three figures and a video evidence was recorded and shown next.

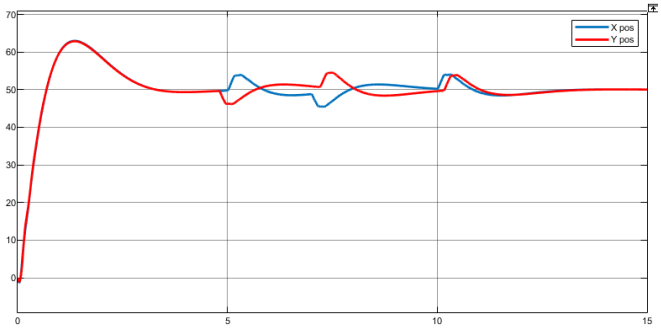


Fig. 29: Test 1. Controlled response with disturbances for a reference of x and y at 50mm and 50 mm respectively.

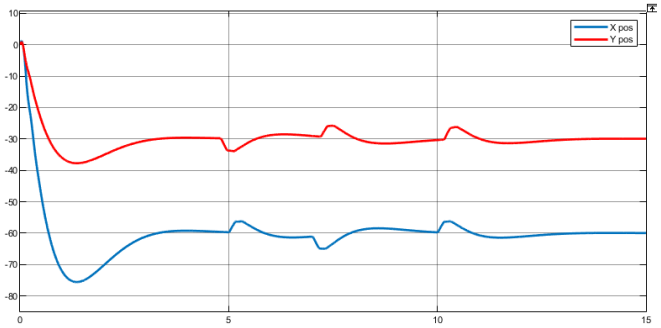


Fig. 30: Test 2. Controlled response with disturbances for a maximum reference value of x and y of -60mm and -30 mm respectively.

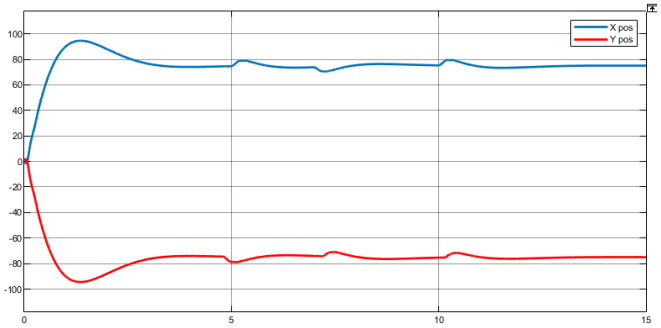


Fig. 31: Test 3. Controlled response with disturbances for a maximum reference value of x and y of 75mm and -75 mm respectively.

The behavior of the ball on the platform for these first three tests can be consulted in the following link.

https://drive.google.com/file/d/1mnzA9bwJQ3ZCz5tya1sZBnK5eEvsWAte/view?usp=sharing

Once the system was thoroughly tested using the calculated gains for both PIDs, the system was tuned in order to achieve a better behaviour. For this tuning, only the gains of the outer loop PID were changed. In order to obtain a quick response that resembles the original desired behaviour, *Kp, Ki and Kd* were set at 16.49, 0 and 6.5 respectively. This achieves an overdamped response that is shown below.

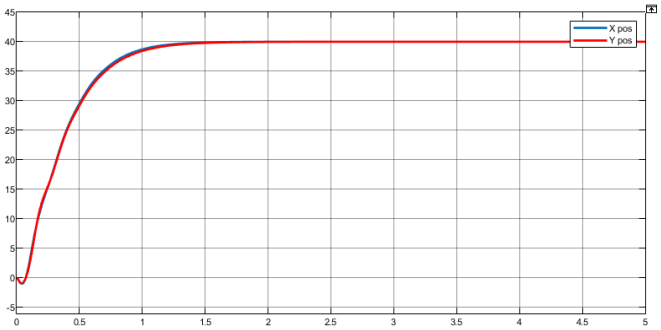


Fig. 32: Test 4. Tuned response for quick overdamped response. Kp, Ki, Kd at 16.49, 0 and 6.5 respectively.

Secondly, in order to show an unstable or critically unstable system, the gains *Kp, Ki and Kd* were set to 16.49, 17.6 and 0.6 respectively. This also represents the point at which the system is at the verge of destabilizing.

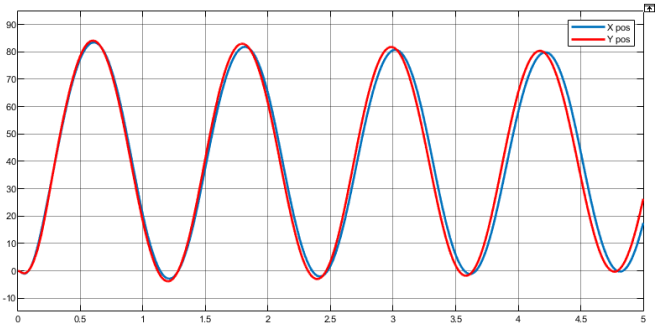


Fig. 33: Test 5. Tuned response for a critically stable system. Kp, Ki, Kd at 16.49, 17.6 and 0.6 respectively.

Last but not least, the system was tuned in order to show a fast underdamped response, one that allows the ball to reach the desired point very quickly and but with the presence of an overshoot.

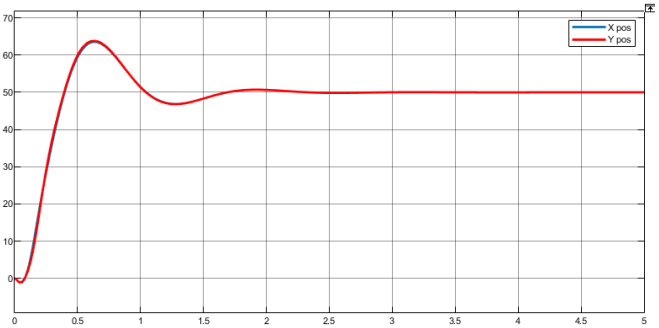


Fig. 34: Test 6. Tuned response for a fast underdamped response. Kp, Ki, Kd at 16.49, 0 and 2.4 respectively.

The three tuned behaviors of the ball can be consulted in the next link.

https://drive.google.com/file/d/1bytsOOw-WGybqh63NEVepdxV_kpyFEBn/view?usp=sharing

VI. Discussion

First of all, from figure 24a it can clearly be seen that the mathematical model of the *BBP* system is indeed correct. The trajectory of the ball over the inclined surface is very similar between the *Multibody* model and the *tf*. This will continue to be valid as long as the angle of the platform is close to zero since the operating point of this linearization was made with this assumption taken into account. Figure 24b shows the error between both the real and the *tf* where a maximum error of around 2.5×10^{-3}m can be observed. This curve essentially tells us that the discrepancy between the

transfer function and the real system is approximately at most 2.5 millimeters which, for the scope and precision of our project, is more than suitable. This assimilation allows us to ensure that the models used in equations 21 and 22 will reflect the appropriate behavior of the plant. Furthermore, this is also one of the reasons why the settling time chosen for the DC Motors was set to 0.2 seconds, to ensure fast reactions to disturbances, thus keeping the platform tilted inside the range of small angles. On the other hand, the mathematical model of the motor can be assumed to be almost perfect, since a fit of 99.13% was obtained using the *System ID* tool from MATLAB. This is made even more evident by the results shown in figure 27, in which the difference between the transfer function and the "real" system is almost non-existent.

Regarding the results obtained using the calculated gains for both PIDs of each plane *(inner and outer loops)*, it can be said that they were really satisfactory. Even without fine tuning the system reacts fast, moving the ball to the desired position in less than 5 seconds, which is already close to the designed settling time of 3.5 seconds. As it can be seen in figures 29, 30 and 31, the system also reacts promptly to disturbances. Nevertheless, the system must be tuned in order to achieve more satisfactory results. Despite these little variations due to different reference points, the initial results are impressive to say the least, the control structure works flawlessly even with the introduction of disturbances. Figure 29 is a good example that shows how both the *x* and *y* reference achieve the stationary error and no matter how many disturbances are applied, the controllers react accordingly to take the ball to the desired position.

The tuning of the *PID* controller, on the other hand, was conducted using heuristic methods in order to enhance the system response to a more suitable behavior according to the needs. The first interesting result from this process was the one obtained in figure 32. This response shows an overdamped response when *Ki* decreases to a value of 0. The figure shows a system that stabilizes quite fast and with no overshoot present at all *(see figure 32)*. This result is interesting and very favorable since it fundamentally eliminates the limiting conditions due to the disappearance of the overshoot and also makes the response much faster, in fact, several tests were conducted off the record where the ball could literally be placed at the very edge of the platform, which gave quite eye-opening results. This behavior is also very interesting since it eliminates the need for an integrator gain, making it possible to actually design only for a PD controller.

However, on the other hand, the gain *Kd* seems to affect the settling time of the response and the damping effect; as *Kd* decreases, the system becomes more unstable / underdamped. This can be seen in figures 33 and 34. Taking all this into account, an optimal configuration was designed which has a settling time of around 3 seconds and an overshot slightly smaller to that of the original system. This is shown in figure 34. All in all, the response and control dynamic of the system was extremely successful. The cascade control structure however *(see figures 19 and 22)*, was probably the most complicated part to analyze. Just like it was mentioned in the *Activity Description*, an assumption *(that was based on the supporting research [7][8])* was imposed where the inner closed loop was considered as a perfect element. Although the supporting research does make this same assumption, they additionally implement a filter later on to compensate for this simplification assumption. In the case of our project, this assumption although not entirely correct, it did bring us to very close and favorable results that only had to be slightly modified later.

VII. Conclusion

The main objective of this project stated in the introduction is to control the position of a ball to a desired reference point on a plane and to additionally reject disturbances such as the launch of external forces applied to the system. According to the results presented, it is safe to say that the objective was successfully fulfilled. The development of the project was only carried out in a remote manner, meaning that there was no construction and/or physical implementation of the model due to the social situation contemporary to the happening of this project. The usage of academic software like *Matlab* and *Simulink* allowed this project to take place. It was theoretically constructed and simulated, letting us intuitively understand the control strategies and their importance in real applications.

The division to two independent 2D systems that are analog to the *Ball and Beam* setup was the correct way to confront the 3D nature of the system. It is also important to mention that supporting papers were used to understand the appropriate control structure and design of a system of this essence. The employment of the transfer function approach and the cascade control system turned out to be a proper way to tackle the control issue. By having two subsystems, each with their own controlled and manipulated variables, the cascade control system lets us have an inner loop and an outer loop.

Unlike the obtention of the inner loop *tf*, which was pretty straightforward, the outer loop technique was a little more complicated. For the pole placement approach, the closed loop *tf* is needed, so we had to come up with a solution for the outer loop controller. If we had used the *tf* of the inner

loop to obtain the outer loop *tf*, this would have resulted in a higher order equation, highly complicating the process. Because of this, we decided to consider the inner loop as an ideal controller, from which there is no error nor delay. The usage of this method proved to be successful for our application, obtaining the expected results.

This project specifically doesn't have any industrial application, it was developed mainly for academic purposes. This is why our values of establishment time and overshoot were arbitrary. The objective of 3.5 s as the establishment time desired was successfully achieved.

There were a few complications presented along the development of the project. Each and one of them was dealt with and overcome, although it is important to mention that the primary obstacles were faced with the *Simulink* program, due to the lack of experience with it. It is fair to say that the development of this project provided us with much experience and knowledge. It was mentally challenging and gave the opportunity to implement numerous concepts seen along our mayor.

VIII. Appendix

- MATLAB and CAD files

https://drive.google.com/file/d/152FfBAv6VnEVB--piwpIEwIbO9x6_a73/view?usp=sharing

- Video of first three tests:

https://drive.google.com/file/d/1mnzA9bwJQ3ZCz5tya1sZBnK5eEvsWAte/view?usp=sharing

- Video of tuning tests:

https://drive.google.com/file/d/1bytsOOw-WGybqh63NEVepdxV_kpyFEBn/view?usp=sharing

- Permanent Magnet DC Motor C23-L40 Datasheet

Part Number*		C23-L33					C23-L40				
Winding Code**		10	20	30	40	50	10	20	30	40	50
L = Length	inches	3.33					4				
	millimeters	84.6					101.6				
Peak Torque	oz-in	125.0	125.0	125.0	125.0	125.0	250.0	250.0	250.0	250.0	250.0
	Nm	0.883	0.883	0.883	0.883	0.883	1.765	1.765	1.765	1.765	1.765
Continuous Stall Torque	oz-in	16.5	16.5	16.5	16.5	16.5	27.0	27.0	27.0	27.0	27.0
	Nm	0.117	0.117	0.117	0.117	0.117	0.191	0.191	0.191	0.191	0.191
Rated Terminal Voltage	volts DC	12 -24	12 - 24	12 -36	12 - 60	12 - 60	12 - 24	12 - 48	12 - 60	12 - 60	12 - 60
Terminal Voltage	volts DC	12	12	24	36	48	12	24	36	48	60
Rated Speed	RPM	4700	2150	4200	3750	3000	2300	3600	3500	2850	2250
	rad/sec	492	225	440	393	314	241	377	367	298	236
Rated Torque	oz-in	7.5	12.6	12.7	14.4	15.8	17.3	25.5	25.3	25.6	24.2
	Nm	0.05	0.09	0.09	0.10	0.11	0.12	0.18	0.18	0.18	0.17
Rated Current	Amps	4.75	4.3	3	2	1.4	4.9	4.3	2.75	1.8	1.1
Rated Power	Watts	26.1	20.0	39.5	40.0	35.1	29.4	67.9	65.5	54.0	40.3
	Horsepower	0.03	0.03	0.05	0.05	0.05	0.04	0.09	0.09	0.07	0.05
Torque Senstivity	oz-in/amp	2.65	4.25	6.2	10.25	15.75	4.84	7.74	12	18.5	28.75
	Nm/amp	0.0187	0.0300	0.0438	0.0724	0.1112	0.0342	0.0547	0.0847	0.1306	0.2030
Back EMF	volts/KRPM	2	3.15	4.6	7.6	11.5	3.58	5.72	8.82	13.82	21.22
	volts/rad/sec	0.0191	0.0301	0.0439	0.0726	0.1098	0.0342	0.0546	0.0842	0.1320	0.2026
Terminal Resistance	ohms	0.60	1.00	1.70	4.00	9.00	0.70	0.96	2.30	5.50	12.00
Terminal Inductance	mH	0.35	0.94	2.00	5.50	13.00	0.50	1.30	3.10	7.36	18.00
Motor Constant	oz-in/watt^1/2	3.4	4.3	4.8	5.1	5.3	5.8	7.9	7.9	7.9	8.3
	Nm/watt	0.024	0.030	0.034	0.036	0.037	0.041	0.056	0.056	0.056	0.059
Rotor Inertia	oz-in-sec²	0.0022	0.0022	0.0022	0.0022	0.0022	0.004	0.004	0.004	0.004	0.004
	g-cm²	155.4	155.4	155.4	155.4	155.4	282.5	282.5	282.5	282.5	282.5
Friction Torque	oz-in	3.0	3.0	3.0	3.0	3.0	3.0	3.0	3.0	3.0	3.0
	Nm	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02
Thermal Resistance	°C/watt	6.2	6.2	6.2	6.2	6.2	5.4	5.4	5.4	5.4	5.4
Damping Factor	oz-in/KRPM	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1
	Nm/KRPM	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001
Weight	oz	27	27	27	27	27	38	38	38	38	38
	g	765	765	765	765	765	1077	1077	1077	1077	1077
Electrical Time Constant	millisecond	0.5833	0.9400	1.1765	1.3750	1.4444	0.7143	1.3584	1.3478	1.3382	1.5000
Mech. Time Constant	millisecond	26.07623	17.2056	13.72994	11.82747	11.44547	16.91906	9.052773	9.100907	9.00927	8.237676
Speed/Torque Gradient	rpm/oz-in	-113.2075	-74.69655	-59.60729	-51.34788	-49.68944	-40.39891	-21.61598	-21.73091	-21.51211	-19.66971

IX. References

[1]. Beer, P. F., Johnston, E.R., Cornwell, P. J. & Self, B.P. (2017). *Mecánica Vectorial para Ingenieros. Dinámica.*. México D.F. : McGraw Hill Education.

[2]. A. Hasp. M. Tjernström. (2019). *Construction and theoretical study of ball balancing platform.* KTH Royal Institute of Technology. Stockholm, Sweden.

[3]. Acrome. *Ball Balancing Table.* Retrieved from: https://acrome.net/product/ball-balancing-table

[4]. Control for MATLAB/Simulink. *DC Motor speed: System Modeling.* Retrieved from: https://ctms.engin.umich.edu/CTMS/index.php?example=MotorSpeed&section=SystemModeling

[5]. M. Keshmiri. A. Fellah. A. Mohebbi. (2012). *Modeling and control of ball and beam system using model based and non-model based control approaches.* Concordia University. Montreal. Canada. Retrieved from: https://www.researchgate.net/publication/235800576_Modeling_and_control_of_ball_and_beam_system_using_model_based_and_non-model_based_control_approaches

[6]. A. Ali. (2017). *Design and implementation of a ball and beam system using a PID controller*. Sudan University of Science and Technology. Recovered from: https://www.researchgate.net/publication/317079231_Design_and_Implementation_of_Ball_and_Beam_System_Using_PID_Controller

[7]. Nainesh Bum. (2013). *Experimental Control Design for Ball and Beam system.* University of Southampton. Retrieved from: https://issuu.com/naineshbumb/docs/ballandbeam

[8]. Quanser Inc. (2011). *Ball and Beam workbook*. Ontario, Canada. Retrieve from: http://docshare02.docshare.tips/files/27419/274198965.pdf

[9]. J.C. Moreno Marín and S. Heredia Avalos (n.d.) *Resumen de Física. Escuela Politécnica Superior*. Retrieved from https://web.ua.es/es/cursos-cero/documentos/-gestadm/dinamica-teoria.pdf

[10]. Zill, G. D., Wright, S. W. (2015). *Ecuaciones Diferenciales con problemas con valores en la frontera.* México: Cengage Learning.

[11]. Ogata, Katsuhiko (1997). *Modern Control Engineering*. Upper Saddle River, N.J.: Prentice Hall.
[12]. Mazzone, V. (2002). *Controladores PID.* Retrieved from https://www.eng.newcastle.edu.au/~jhb519/teaching/caut1/Apuntes/PID.pdf
[13]. Permanent Magnet Brush DC Motors Technical Datasheet - moc23series. Retrieved from: https://www.moog.com/content/dam/moog/literature/MCG/moc23series.pdf