Virtual Construction and Simulation of A Ball&Plate System

AHMET BURAK AKYUZ
Mechatronics Engineering
Bahcesehir University
Istanbul, Turkey

OGUZHAN BERKE YESILDAG
Mechatronics Engineering
Bahcesehir University
Istanbul, Turkey

ONUR KARAER
Mechatronics Engineering
Bahcesehir University
Istanbul, Turkey

YARKIN SEYMEN
Mechatronics Engineering
Bahcesehir University
Istanbul, Turkey

Abstract—*In this paper, an autonomous ball-and-plate system is virtually constructed. The system is dynamically modeled, and the required coefficients for PID control were derived from the calculated system dynamics. The system is unstable and only has two degrees of freedom. It is modeled with two servo motors as the actuators, each controlling one axis. According to design specifications, the system is modelled in CAD software, and the 3D CAD design is exported into simulation software for further inspection and testing. Position sensing is done with the help of the simulating software, and the controllers for the system are tuned according to the simulation results. The overall performance of the system is discussed at the end.*

Keywords— PID Control, Autonomous Control, Ball& Plate System, Simulation and Modelling.

I. Introduction

Mechatronics is described as the synergistic combination of mechanical engineering, electronics, control systems, and computers [1]. Its main concern is maintaining the integrity of the system through successful connections and synergies between different components of the system. This integrity is especially crucial in autonomous systems and usually requires separate and special attention since flaws in interactions between different system components may cause catastrophic failures, and modern system components are already rather complex by themselves.

The Ball-on-Plate system is a great example of an autonomous system. Its inherent nature makes it suitable to be considered a mechatronics system, and the construction of such a system requires a certain level of proficiency in various fields of mechatronics such as mechanical design, control theory, electronics, and software implementation. Combined with its easily comprehendible working principles and ease of manufacturing, it is a great fit to be a practice project for gaining experience for those who are interested in mechatronics, either as a hobby or academically.

Even if the system’s working principles are easy to understand, the inherent nature of the system presents a somewhat challenging control problem in the design process. Such unconventional problems make the importance of mechatronics design methodology more apparent and further underline the importance of integrity in systems in general.

For these advantages, the ball-on-plate system has been a preferred system in academic mechatronics studies, especially by undergraduate students, and consequently, the system has been thoroughly analyzed and numerous studies have been performed on it.

Modeling and simulation are two crucial aspects of the engineering design process. Proper implementation of systems in virtual environments can immensely help in real-life engineering research and development processes by providing a number of advantages such as early fault detection, design improvements via inspection of the system's interaction with the environment, etc.

Ball-on-Plate systems share common basic principles with most real life dynamic platform applications such as 6-DOF flight simulators, animatronics technology, earthquake simulators, etc. Modeling such a system may provide help in these fields as well.

Because of these reasons, we will be focusing on the modeling and virtual construction of such a system in this report.

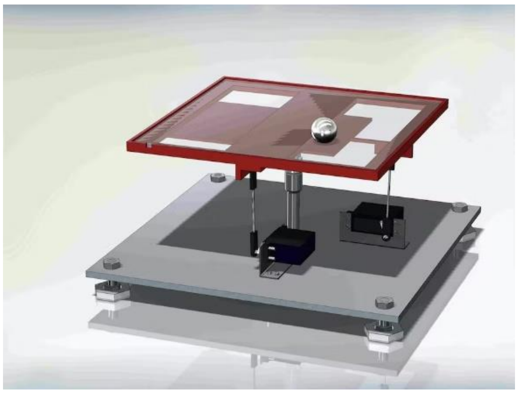

Figure 1. An example Ball-on-Plate System

II. Literature Review

Ball-on-Plate systems have been a popular research topic for mechatronics systems for quite some time, and they have been subjected to various studies. Awtar et al. examine the traditional ball-on-plate system in a very broad perspective, analyzing its potential material cost, performance, and functionality. Later, moving on to review the system dynamically, hardware design, component selection for construction, and controller design, they use a touchscreen instead of a camera to track the position of the ball [1]. In a more recent work, Brezina et al. construct a similar system with a camera, focusing more on the effects of different image processing algorithms on performance [2]. Ho et al.'s study presents a detailed dynamic model of the system along with parameter calculation, and the performance of the simulation and the real-life model are compared afterwards [3]. Another study conducted by Taifour et al. examines the ball-on-plate control in a 1-DoF model. The research presents a different take on perception by choosing ultrasonic distance sensors for distance measuring. The paper focuses on experimental system efficiency observations obtained by changing the coefficients of a PID controller, which is the preferred control method in the system [4]. Fabregas et al. constructed a system with a variable set point. The position of the ball was checked with a fixed point in the system, and set points were modified in order to demonstrate different trajectories (such as a circle, square, and ellipse). Ball position checking and trajectory tracking were performed by different controllers [5]. The efficiency of different control methodologies was also checked on the ball-on-plate system in the past. Fan et al. designed and constructed a ball-on-plate system using the hierarchical fuzzy control method. While it is generally the case to use empty platforms for such constructs, Fan et al. created a platform along with some obstacles and guaranteed that our ball would reach a certain point from another point. The trajectory of the ball in this system is determined by fuzzy logic control [6]. Ball-on-Plate systems of various designs have also been built in the past. Chi-Cheng et al. used image processing along with a 2-DoF manipulator in order to construct a balancing platform, as opposed to the traditional 2-servo motor actuator design. A linear quadratic regulator (LQR) was used as a controller, and several experimental studies were conducted to increase the stability of the system [7]. Park and Lee constructed a similar system with a 6-DOF robotic manipulator. Ball coordinates on the platform were obtained by image processing, and experiments were implemented in real-life. [8]. Kassem et al. designed a ball-on-plate system using a Stewart platform and tested 4 different control strategies (PID, LQR, fuzzy logic, and sliding mode) to control the position of the ball in the system. These four methods are then compared in terms of their efficiency in fixed-point operation and trajectory tracking [9].

III. DESIGN

A. *Design Assumptions*

In order to derive the systems' transfer function and virtually construct the model, some specifications about the materials should be decided. In this system, the ball is assumed to be a solid sphere made from IASA 1030 cold-drawn steel, and the platform is assumed to be made from glass. The design and material selection will be justified in a later section of this report. According to these specifications, our table of parameters becomes:

Parameters	Values
m_b	$0.0139\ (kg)$
r_b	$0.0075\ (m)$
g	$9.81(m/s^2)$
I_b	$3.1292 * 10^{-7}(kg * m^2)$
L_p	$0.2\ (m)$
r_a	$0.021\ (m)$

Table 1. Dimension and Property parameters of the System.

B. *Mathematical Modeling of the System*

In order to start the design process, it is essential to first understand the system's behaviour and its kinematics. For this purpose, mathematical construction of system dynamics is essential.

Before mathematically deriving the system equations, some assumptions are made:

- The ball is assumed to be moving at small speeds and always in contact with the table.
- No slipping is assumed. Friction does no work and is therefore ignored.

The ball-on-plate system has two axes of movement. The ball is controlled by the rotational movement of the platform in the X and Y axes, and this nature of the system results in two dynamic equations. However, since both axes are symmetrical, the equations are the same, and modelling on only one axis is enough [8]. A visual representation of the system can be seen below (see Fig. 2).

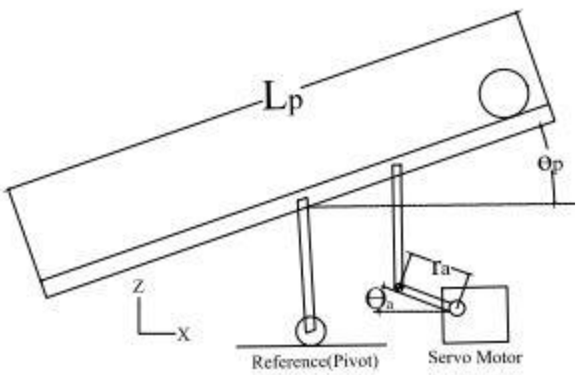

Figure 2. Visual Representation of System Dynamics (X-Axis)

In this report, the dynamical representation of the system is derived with Euler-Lagrange Equations:

$$\frac{d}{dt}\left[\frac{\vartheta\mathcal{L}}{\vartheta\dot{q}_i}\right]-\frac{\vartheta\mathcal{L}}{\vartheta q_i}=Q_i$$

and the Lagranian Equation is:

$$\mathcal{L}=T-V$$

T and V are the kinetic and potential energy of the system respectively. The kinetic energy of the ball and the platform are derived as:

$$T_b = \frac{1}{2}\left[m_b(\dot{x}_b{}^2 + \dot{y}_b{}^2) + \frac{I_b}{r_b{}^2}(\dot{x}_b{}^2 + \dot{y}_b{}^2)\right] = \frac{1}{2}(m_b + \frac{I_b}{r_b{}^2})(\dot{x}_b{}^2 + \dot{y}_b{}^2)$$

$$T_p = \frac{1}{2}\left[(I_p + I_b)(\dot{\alpha}^2 + \dot{\beta}^2) + m_b(x_b{}^2\dot{\alpha}^2 + 2x_b\dot{\alpha}y_b\dot{\beta} + y_b{}^2\dot{\beta}^2\right]$$

Where T_b and T_p are the kinetic energy of the ball and plate. m_b, r_b, x_b *and* $\dot{x_b}$ are the mass, radius, position and velocity of the ball on the x-axis. y_b *and* $\dot{y_b}$ are the position and velocity of the ball on y-axis. I_b *and* I_p are moment of inertia of the ball and plate and finally, α *and* β are the angle of the plate with X and Y axes respectively.

Total kinetic energy of the system is the sum of the kinetic energies of the ball and plate and is derived as:

$$T = T_b + T_p =$$

$$\frac{1}{2}\left(m_b + \frac{I_b}{r_b{}^2}\right)(\dot{x}_b{}^2 + \dot{y}_b{}^2) + \frac{1}{2}(I_p + I_b)(\dot{\alpha}^2 + \dot{\beta}^2) + \frac{1}{2}m_b(x_b{}^2\dot{\alpha}^2) + 2x_b\dot{\alpha}y_b\dot{\beta} + y_b{}^2\dot{\beta}^2)$$

Since reference point is selected as the pivot of the plane, potential energy calculation becomes:

$$V_b = m_b gh = m_b g(x_b \sin\alpha + y_b \sin\beta)$$

Where g is the gravitational acceleration.

Using the above Lagranian equation:

$$\mathcal{L} = T - V =$$

$$\frac{1}{2}\left(m_b + \frac{I_b}{r_b{}^2}\right)(\dot{x}_b{}^2 + \dot{y}_b{}^2) + \frac{1}{2}(I_p + I_b)(\dot{\alpha}^2 + \dot{\beta}^2) + \frac{1}{2}m_b(x_b{}^2\dot{\alpha}^2) + 2x_b\dot{\alpha}y_b\dot{\beta} + y_b{}^2\dot{\beta}^2 - m_b g(x_b \sin\alpha + y_b \sin\beta)$$

Next equations are calculated by taking the derivative of the previous equation with respect to time, position and velocity of the ball:

$$\frac{\vartheta\mathcal{L}}{\vartheta\dot{x_b}} = \left(m_b + \frac{I_b}{r_b{}^2}\right)\dot{x}_b$$

$$\frac{d}{dt}\frac{\vartheta\mathcal{L}}{\vartheta\dot{x_b}} = \left(m_b + \frac{I_b}{r_b{}^2}\right)\ddot{x}_b$$

$$\frac{\vartheta\mathcal{L}}{\vartheta x_b} = m_b(x_b\dot{\alpha} + y_b\dot{\beta})\dot{\alpha} - m_b g \sin\alpha$$

Substituting these equations into the first equation and rearranging it, dynamic model becomes:

$$\left(m_b + \frac{I_b}{r_b{}^2}\right)\ddot{x}_b - m_b(x_b\dot{\alpha} + y_b\dot{\beta})\dot{\alpha} + m_b g \sin\alpha = 0$$

Above equation is non-linear but a linearized equation is needed to derive the transfer function. By linearizing it according to small deviations in α and β the equation becomes:

$$\left(m_b + \frac{I_b}{r_b{}^2}\right)\ddot{x}_b + \frac{m_b g r_a}{L_p}\theta_x = 0$$

Same equation can be also applied to y-axis since the model is symmetrical:

$$\left(m_b + \frac{I_b}{r_b{}^2}\right)\ddot{y}_b + \frac{m_b g r_a}{L_p}\theta_y = 0$$

C. *Controller Design*

Applying the Lagrange equation along with integration property, transfer function of the system becomes:

$$G(s) = \frac{X_b(s)}{\theta_x(s)} = -\frac{m_b g r_a}{L_p\left(m_b + \frac{I_b}{r_b{}^2}\right)s^2}\frac{rad}{m}$$

After substituting the values (Table 1) to the equation and converting it into the desired units, the function becomes:

$$G(s) = \frac{0.4216}{s^2}\frac{deg}{cm}$$

The servo motors in the system are also modelled using some assumptions for realism purposes and a transfer function between voltage and resulting velocity is acquired as follows:

$$G_M(s) = \frac{\vartheta(s)}{V_M(s)} = \frac{K_M}{t_s s + 1} = \frac{100}{0.01s + 1}$$

Where t_s is the time constant and K_M is the motor gain coefficient.

For the system, A PID Controller is decided for control implementation. PID, short for Proportional Integral Derivative controller is an automatically optimized and accurate control system used to regulate different parameters like temperature, pressure and speed at desired values[1]. The mathematical representation of a PID is as shown below:

$$u(t) = K_p e(t) + K_i \int_0^t e(s)ds + K_d \frac{d}{dt}e(t)$$

Expressing it in Laplace Domain:

[1] Definition from: https://www.ssla.co.uk/pid-controller/

$$K(s) = K_p + K_i * \frac{1}{s} + K_d * s$$

$u(t)$ is the output, $e(t)$ is the error signal and K_p, K_i and K_d are proportional, integral and derivative gain respectively.
A block diagram of the system is shown below (Fig. 3).

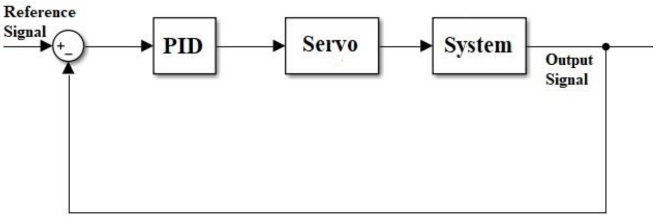

Figure 3. Closed-Loop Block Diagram of the System

The closed-loop function and characteristic equation of the system are as follows:

$$\frac{Output\ Signal}{Input\ Signal} = \frac{(K_p + K_d s)(G(s))(G_M(s))}{1 + (K_p + K_d s)(G(s))(G_M(s))}$$

Setting the numerator of the denominating term to 0:

$$0.01s^3 + s^2 + 40.6K_d s + 40.6K_p = 0$$

The characteristic equation is one order higher than the general equation. So, by ascending the general equation by one order, it becomes:

$$P_D(s) = (As + B)(\ s^2 + 2\xi\omega_n s + \omega_n{}^2)$$
$$= A\ s^3 + (2A\xi\omega_n + B)s^2 + (A\omega_n{}^2 + 2B\xi\omega_n)s + B\omega_n{}^2$$

ξ is the damping ratio and ω_n is the natural frequency of the system. These values are calculated regarding overshoot and settling time and are decided in accordance with some design requirements. For these reasons, %2 overshoot with a 1 second settling time is chosen and ξ and ω_n are calculated as 0.7797 and 5.12 respectively. After the desired values are substituted into the equations our coefficients are:

$$A = 0.01, \qquad B = 0.92$$
$$K_p = 0.57, \qquad K_D = 0.18$$

IV. Virtual Construction

Several steps need to be completed in order to implement the model virtually. Generally speaking, these steps include: Mechanical Modelling (CAD), Implementation of Controllers and Sensing. Each of these tasks are also divided into sub-tasks and are further explained in their corresponding sections.

A. *Mechanical Modelling*

In order to successfully construct a system, it should be first designed mechanically. For this task, a variety of CAD software are considered and, in the end, SOLIDWORKS is decided because of its 3D Drawing capabilities, ease of use and relatively rich extension availability which expedite the exporting process considerably. We choose to construct a 20x20 cm square platform. For DoF availability a universal joint is used for the platform and 2 cylindrical joints for the arms. An example image of the drawing can be seen below (Fig. 4).

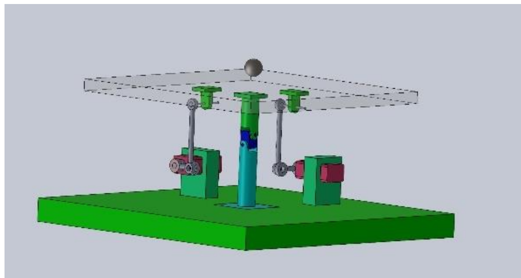

Figure 4. 3D CAD Drawing for Ball-on-Plate System

Regarding materials, a solid sphere made from AISI 1020, Cold Rolled steel along with a glass platform is decided. Arm heads and joints are from standard alloy metal and their connecting rods are from polymer. Finally, servos along with the base is constructed as plastic.

Final design is also mechanically analyzed for bending, buckling and resonance and the model is verified to be operatable. An example result of one of the tests conducted can be seen below (Fig. 5).

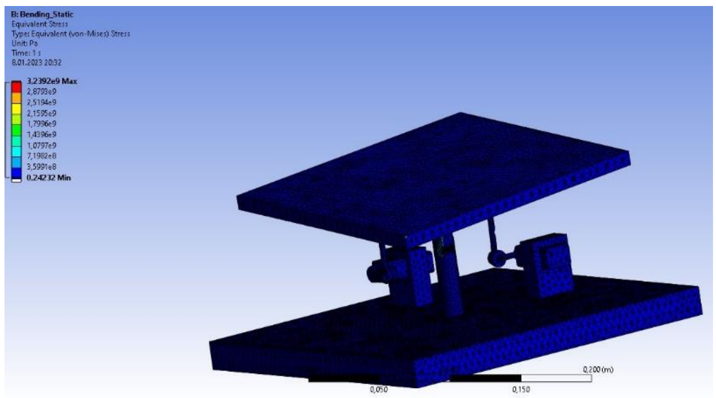

Figure 5. Bending Analysis Result in Terms of Total(Von-Mises) Stress.

B. *Controller Implementation*

After the construction of the model, it is needed to export it into a suitable simulation software to control it. Similar to the previous step, a variety of simulation softwares are considered and Simscape Multibody is decided because of some software advantages that it provides. After exporting the model into Simscape, two seperate PID controllers, one for each axis, is implemented with previously derived coefficients.Each connected to one servo. The Simscape Model for PIDs can be seen below (Fig.6).

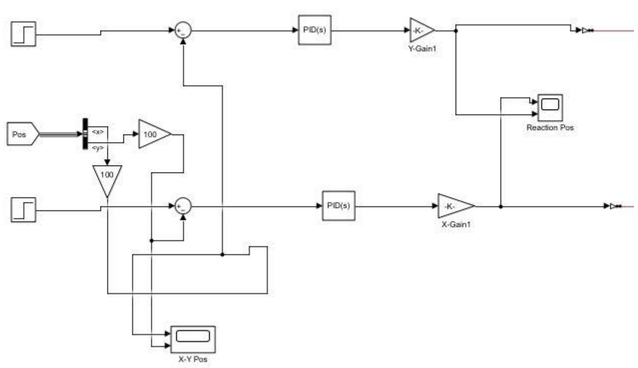

Figure 6. Control Implementation for Ball-on-Plate System

Outputted position is fed to the system as meters and are converted into centimeters with the help of gain blocks. In the next step the scaled-up positions are negatively fed to the controllers which operate with the previously calculated coefficients. Finally, the output is converted from degrees to radians and are sent to the servo motors.

C. Implementation of Sensing

A resistive touchscreen is decided to be used for sensing. A resistive touchscreen operates by sensing the direct pressure applied to it. A resistive touch screen consists of a touch layer placed on top of a standard display. The touch layer normally includes two transparent electrical layers separated by a small gap. Pressing the display's surface causes the two separate layers to come into contact, which creates an electrical connection that can be sensed and located[2]. An example touchscreen can be seen below (Fig. 6).

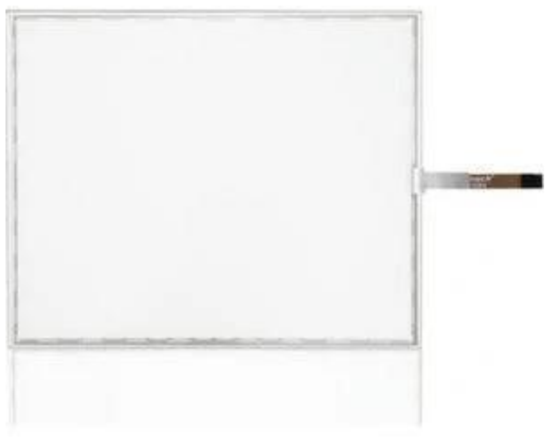

Figure 6. A Standard Touchscreen

"Simulink Contact Forces" library is used to simulate the touchscreen. Its visualized working principles can be observed below (Fig. 7).

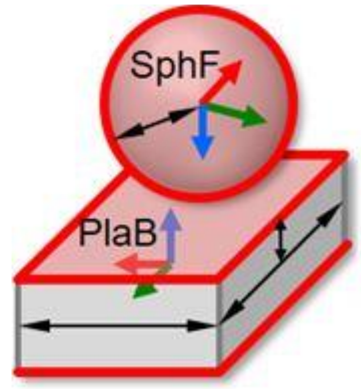

Figure 7. Sensing Block Working Principle

The block used for sensing requires the dimensions of the plate and the body that will be contacting to it. After setting the dimensions in accordance with the CAD design, a static and kinetic friction force between the platform and the sphere were implemented. The friction coefficients between glass and metal are taken from the paper of Buckley [10]. The output is strictly limited to only X and Y axes displacements to further simulate a touchscreen.

SIMULATION AND RESULTS

For the system, CAD drawing is constructed according to some specifications along with previously calculated system dynamics and control theory. A 20x20 cm platform is constructed with 2 servo motors. Platforms material is taken as glass to simulate a resistive touchscreen and the ball is assumed to be a solid sphere made from steel. After the sketching the model is then exported into Simscape Multibody simulation environment. The servo actuators are limited to operate between -15° and 15° to simulate a fail-safe design to prevent structural failure from excessive control inputs. A touchscreen is simulated with the help of "Simulink Contact Forces" library and the surface is implemented with static and dynamic friction coefficients to further simulate real life conditions. The surface outputs are limited to x and y axes positions and are fed to the controllers. For control, two PID controllers (one for each axis) are implemented. Controller coefficients are set as initial calculated parameters and further tuned according to simulation results. The tuned coefficients are decided as $K_p = 0.55$ and $K_D = 0.18$. Furthermore, an Integral coefficient of $K_I = 0.04$ is experimentally implemented to reduce steady-state error. In the simulation, the ball is initially gently placed at the middle of the platform and a disturbance is applied to the system at 1 second to observe the transient behavior of the system. The X and Y axes positions of the ball are as follows (Reference settling point is 1 cm) (See Fig. 8 and Fig. 9):

[2] Definiton from: https://www.gsmarena.com/glossary.php3?term=resistive-touchscreen

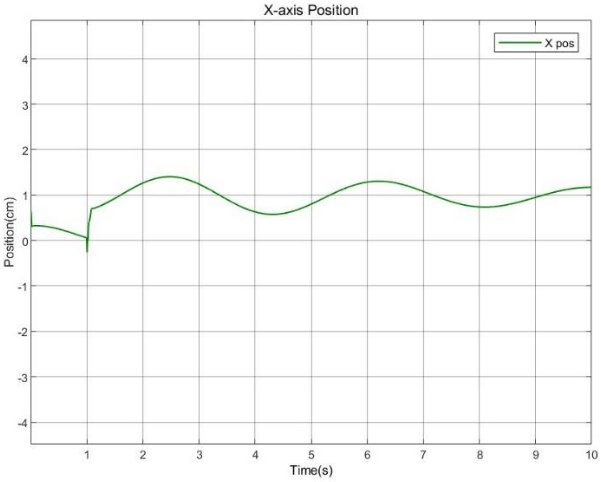

Figure 8. X-axis position of Ball

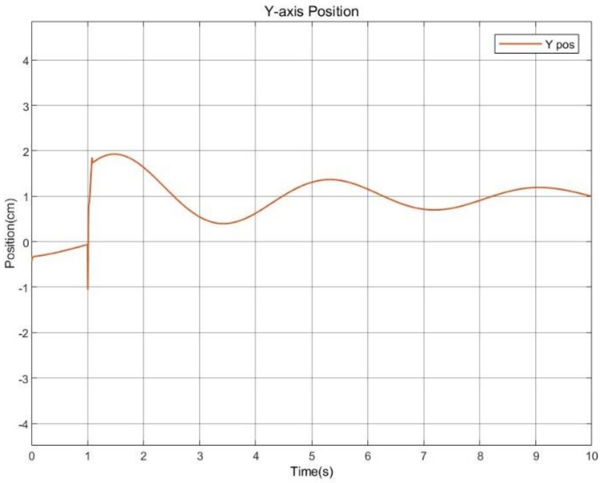

Figure 9. Y-axis position of Ball

It is inspected from the figures that the maximum position and displacement for X-axis occurs after the disturbance at approximately 2.5 seconds with 1.406 cm position and 0.4 cm displacement. For Y-axis the maximum values occur around 1.5 seconds with a position value of 1.93 cm and a displacement of approximately 0.93 cm. The maximum overshoot values for X and Y axes are therefore calculated as %40.06 and %93 respectively. Both systems are operating in reducing oscillations and seem to enter a quasi-equilibrium state (8.09 and 7.2 seconds for X and Y axes). Taking those states as goals for stability and given that the disturbance is applied at 1 second the average settling time is calculated to be approximately 6.64 seconds (7.09 and 6.2 seconds respectively).

CONCLUSION

In this study, a Ball-on-Plate system was virtually constructed. The system was designed with 2 servo motors and a touchscreen for sensing. The dynamics of the system were calculated using Euler-Lagrange operations and an initial design for a PD controller with coefficients $K_p = 0.57$, $K_D = 0.18$ were obtained. After simulation testing the coefficients were tuned to become $K_p = 0.55$,$K_D = 0.18$. An experimentally calculated Integral term of $K_I = 0.04$ is added to improve performance and reduce steady-state error. The settling times for axes were observed as 7.09-6.2 seconds and maximum overshoot values were %40.06 - %93 for X and Y axes respectively.

The system is implemented without the mechanical modelling of servo motors and some assumptions regarding their properties are made through the calculations. Also, system dynamics were calculated with neglecting the work done by friction. Further mechanical modelling of servo motors along with a more thorough dynamic modelling which also takes work done by friction into account will result in a more detailed construction of the system. Causing the system to operate more efficiently.

REFERENCES

[1] Awtar, S., Bernard, C., Boklund, N., Master, A., Ueda, D., Craig K. (2002). Mechatronic Design Of A BallOn-Plate Balancing System. Mechatronics, 12(2), 217-228 pJ. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.

[2] Brezina, A., Tkacik, M., Tkacik, T., Jadlovska S. (2019). Upgrade of the Ball and Plate Laboratory Model. IFAC-PapersOnLine, 52(27), 277-282 p.

[3] Ho, M., Rizal, Y., Chu, L. (2013). Visual Servoing Tracking Control of a Ball and Plate System: Design, Implementation and Experimental Validation. International Journal of Advanced Robotic Systems, 10(7).

[4] Taifour, A., Ahmed, A., Almahdi, H., Osama, A., Naseraldeen, A. (2017). Design and Implementation of Ball and Beam System Using

[5] Fabregas, E., Chacón, J., Dormido-Canto, S., Farias, S., Dormido G. (2015). Virtual Laboratory of the Ball and Plate System. IFAC-PapersOnLine, 48(29), 152-157 p.

[6] Fan, X., Zhang, N., Teng, S. (2004). Trajectory Planning And Tracking Of Ball And Plate System Using Hierarchical Fuzzy Control Scheme. Fuzzy Sets and Systems, 144(2), 297-312p

[7] Chi-Cheng, C., Tsai, C. (2016). Visual Servo Control for Balancing a Ball-Plate System. International Journal of Mechanical Engineering and Robotics Research, 5(1).

[8] Park, J., Lee, Y. (2003). Robust Visual Servoing For Motion Control Of The Ball On A Plate. Mechatronics, 13(7), 723-738 p.

[9] Kassem, A., Haddad, H., Albıtar, C. (2015). Comparison Between Different Methods of Control of Ball and Plate System with 6DOF Stewart Platform. IFAC-PapersOnLine, 48(11), 47-52 p.

[10] Buckley, H. (1973). Friction Behaviour of Glass and Metals in Contact with Glass in Various Environments. NASA Technical Note, NASA TN D-7529.

[11] Gursoy C. H. and Adar, G. N. (2022). Image-Based Control of 2-DOF Ball Balancing System. Journal of Innovative Science and Engineering.6(2):160-174