J Inno Sci Eng,2022, 6(2):160-174
https://doi.org/10.38088/jise.1091154

Research Article

Image-Based Control of 2-DOF Ball Balancing System

Hüseyin Can GURSOY [1], Nurettin Gökhan ADAR[2*]

[1,2*] *Department of Mechatronics Engineering, Bursa Technical University, 16310 Bursa, TURKEY*

Cite this paper as:
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

*Corresponding author: Nurettin Gökhan Adar
E-mail: gokhan.adar@btu.edu.tr

Received Date:27/03/2022
Accepted Date:05/10/2022

Abstract

In this study, the Ball-Plate stabilization system is designed to control with image processing algorithms. The position of the ball is aimed to control by tilting the plate on which the ball is located at a certain position and velocity. The system has two rotational degrees of freedom and is unstable. In the system, two DC motors are used as an actuator, and a camera is used as a feedback sensor. The camera captures the position of the ball and image processing algorithm calculates the that position to blance the plate..PID control is selected for servo motors. Thus, the position of the ball can be controlled so that it converges to the desired point on the plate. Real-time tests are conducted, and Maximum Overshoot and Steady State Error are calculated for both the x and y-axis, and results are given in figures. For the setpoint (15 cm, 15 cm) the Maximum Overshoot and Steady State Error were measured at 40.6% - 8% on the x-axis and 48.6% - 8.6% on the y-axis, while for the setpoint (10 cm, 10 cm) The Maximum Overshoot and Steady State Error were measured at 40.6% - 8% on the x-axis and 48.6% - 8.6% on the y axis.

Keywords: Ball&Plate, PID Control, Image Processing

1. Introduction

Automatic control systems are an indispensable part of today's autonomous systems and are used in all areas where human beings develop technologically, such as input-output control of reactions in chemical and biological studies, power generation facilities, position and velocity control of aircraft in aviation, and space industries. Image processing is one of the important issues of our age, which gives very efficient results when used with automatic control systems. In recent years, these two phenomena have been used together, and visual servoing has been studied frequently in the engineering literature. The information needed by the control system can be obtained by using image processing techniques. By filtering the raw images taken from the camera and by performing mathematical operations, objects can be tracked and detected, and necessary information can be obtained. The Ball and Plate system is an example of visual servoing. The position information of the ball is obtained using the images. After processing the images necessary information is calculated to control the servo motors. The system tries to stabilize the ball by updating the information it receives from the camera.

Awtar et al. discuss the basic design and development of a classical Ball-Plate system according to mechatronic design principles. The realization of the design touches on basic facts such as cost, performance, and functionality. It studies a completely dynamic system review, hardware design, sensor and actuator selection, system modeling, parameter identification, and controller design for the top-plate system. The touchscreen is used to track the ball coordinate instead of the camera [1]. Brezina et al. discuss developing the classic Ball-Plate system. Different methods are tried to improve the system hardware and software. The study especially focuses on the effect of different image processing algorithms on cycle times and delays [2]. Itani designed a ball stabilization system using two Servo Motors, ARDUINO as a microprocessor and a camera for feedback. The transfer function was calculated by mathematical modeling of the system, and a PID controller was designed with a transfer function. The system was implemented in real-time, and the stability and efficiency of the system were shown with graphs [3]. Ho et al. designed both a real-time prototype and a virtual simulation of the Ball-Plate system. A detailed dynamic model of the system was created and the system wascalculated with the coefficients obtained from the dynamic model. The accuracy of the system in the prototype and simulation were examined and compared [4]. Taifour et al. studied on a real-time system design for ball balancing on a single axis. In the system, a Servo Motor wasused as an actuator, while an ultrasonic distance sensor was used for feedback and ARDUINO as a microcontroller. PID was preferred as the controller, and the study aims to experimentally measure the system response and stability according to the changing P, I, and D coefficients [5]. Kocaoğlu designed a Ball Beam system that aims to keep the ball at any selected point on the rod. The current state of the ball was measured by using different sensors, and the ball is displaced by changing the rod's angle through various motors. By applying different control algorithms, the positioning of the ball at the desired point was tried to be ensured appropriately. Settling time, maximum overshoot, and steady-state error were tried to be kept in optimum condition. The PID control method was applied, and the experimental setup was conducted [6]. Fabregas et al. focused on the variable set point for the Ball-Plate system in a simulation. While the position of the ball was usually controlled with a fixed point, the setpoint was changed to a certain trajectory (like a square, circle, or ellipse). Both ball position control and trajectory tracking were performed by different controllers [7]. Fan et al. designed a Ball-Plate system using the Hierarchical Fuzzy Control method. A setup of the system was designed, and the system was run in real-time. While an empty and flat plate was

used in general Ball-Plate systems, obstacles were placed on the plate in this study, and the ball was ensured to reach another point from a certain point. The trajectory of the ball was determined by the Fuzzy Logic Control method [8]. Chen et al. studied the control of Ball-Plate systems on a robot that moved on rough terrain instead of on a fixed ground. The design of a walking robot with 6 parallel legs and the mathematical model of the movement were combined with the model of the classical Ball-Plate system. An experimental setup was conducted, and the system was examined in real-time [9]. Chi-Cheng et al. manufactured the plate of ball balancing system move with a design different from classical methods. Ball coordinates were obtained with image processing algorithms. To stabilize the plate, a manipulator with two degrees of freedom waspreferred rather than a Servo Motor. Linear Quadratic Regulator (LQR) was designed as the controller, and experimental studies were performed to increase the stability of the system [10].

Park and Lee, used a 6-degree-of-freedom Robot Manipulator as an actuator to move the plate instead of servo motors. Ball coordinates are obtained with Image Processing algorithms. The experimental setup is implemented in real-time [11]. Kassem et al designed a Ball-Plate system using the 6 DOF Stewart Platform. PID, LQR, Sliding Mode, and Fuzzy Logic Controller are used separately to control the ball position in the system. Four control methods are compared in terms of efficiency and accuracy when the system is being operated both for a fixed setpoint and for following a trajectory. The system is tested both in the simulation and on the experimental setup [12].

In this study, the position of the ball on the plate with two degrees of freedom was automatically controlled with image processing techniques. The system consisted of two Servo motors, a camera, a controller, a plate, and mechanical parts that enable the system to move. Servo Motors rotated the plate in two different axes, and the ball could move as the force (gravity) acting on the ball changes. To calculate the change in the position of the ball against the angle change of the motors, the mathematical modeling of the system was derived, and the Transfer Function was obtained. A PID-based controller was designed using the transfer function. The position information of the ball was obtained by using image processing algorithms through a camera placed on the system, and this coordinate ws used as feedback. Maximum Overshoot and Steady State Error were calculated for both the x and y axes, and the results are given in figures.

2. Materials and Methods

2.1. Experimental Setup and Mathematical Model of Ball Balancing Table

The experimental setup of the ball-plate system consisted of the servo motors, Arduino Uno, camera, and mechanical parts, and was given in Figure 1. TowerPro MG996 servo motors were used as actuators. HXSJ A870 webcam having a resolution of 640x480 pixelswas peferred, and a connection wa established via the USB port. The top plate was made of 30x30x3 mm wood. Mechanical parts were manufactured by using the additive manufacturing technique with PLA filament at 80% fill rate and 0.2 mm layer thickness. The system was an example of an electromechanical system, and its mathematical modeling was examined, and the transfer function of the system was calculated to design a controller.

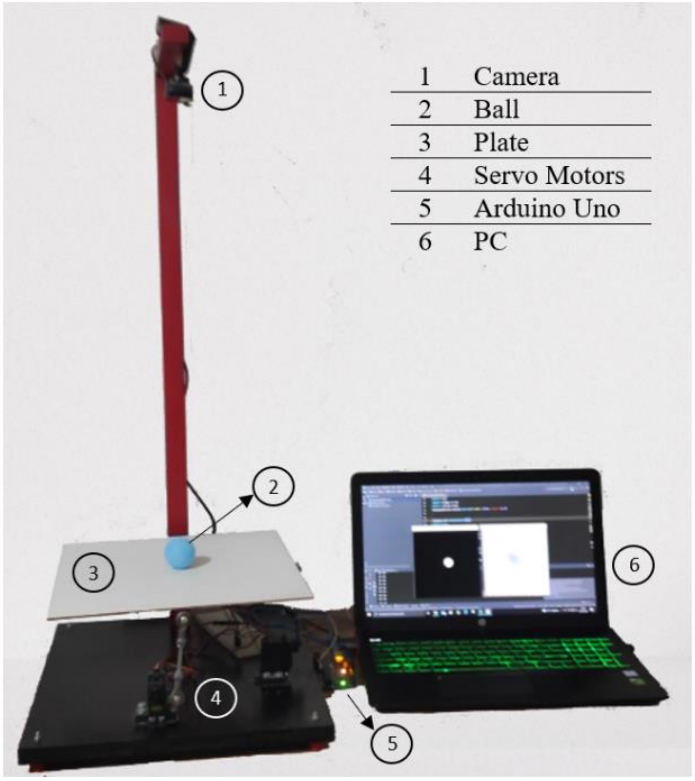

Figure 1. Experimental Setup

The plate on which the ball was located can move in two axes. However, since both axes were symmetrical with respect to each other, their dynamic equations were considered the same, and the system could only be modeled on one axis [11]. Therefore, the mathematical modeling and transfer function were calculated only for the x-axis. The ball and plate system model was presented in Figure 2.

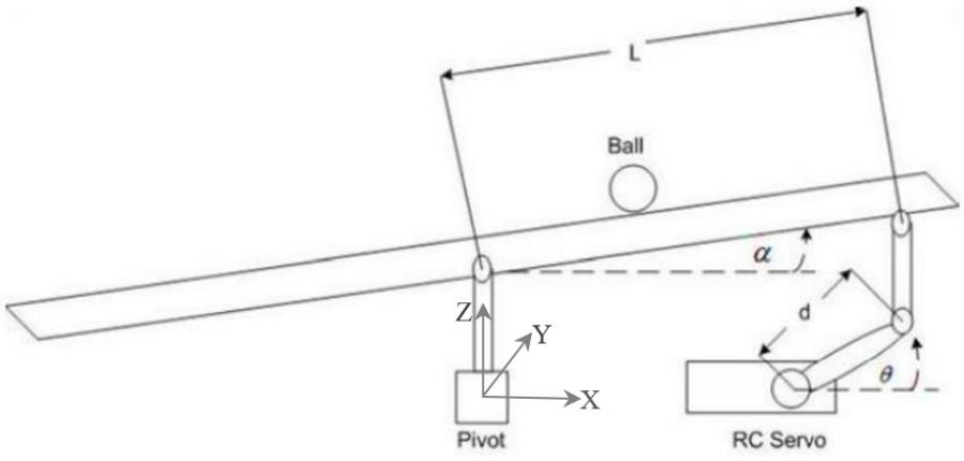

Figure 2. Dynamic Model of the System (x-axis) [3]

The parameters of the system are given in Table 1.

Table 1. Parameters of The System

Parameters	Value
m_b(kg)	0.0025
r_b(m)	0.02
g(m/s^2)	9.81
I_b(kg.m^2)	3.2 x 10^{-6}
L_p(m)	0.3
r_a(m)	0.024

In this study, the dynamical model of the system is derived with Euler-Lagrange Equations.

$$\frac{\mathrm{d}}{\mathrm{dt}}\left(\frac{\partial \mathcal{L}}{\partial \dot{q}_i}\right) - \frac{\partial \mathcal{L}}{\partial q_i} = \mathrm{Q}_i \tag{1}$$

and the Lagrangian is as:

$$\mathcal{L} = \mathrm{T} - \mathrm{V} \tag{2}$$

where T is Kinetic Energy, and V is the Potential Energy of the system, respectively.

The kinetic energy of a ball and the table are calculated as follows:

$$T_b = \frac{1}{2}\left[m_b(\dot{x}_b{}^2+\dot{y}_b{}^2)+\frac{I_b}{r_b^2}(\dot{x}_b{}^2+\dot{y}_b{}^2)\right] = \frac{1}{2}\left(m_b + \frac{I_b}{r_b^2}\right)(\dot{x}_b{}^2+\dot{y}_b{}^2) \tag{3}$$

$$T_p = \frac{1}{2}(I_P + I_b)(\dot{\alpha}^2 + \dot{\beta}^2) + \frac{1}{2}m_b(x_b^2\dot{\alpha}^2 + 2x_b\dot{\alpha}y_b\dot{\beta} + y_b^2\dot{\beta}^2) \tag{4}$$

where T_b is the kinetic energy of the ball, T_p is the kinetic energy of plate, m_b is the mass of the ball, r_b is the radius of the ball, x_b is the position of the ball on the x-axis, y_b is the position of the ball on the y-axis, x ̇_b is the velocity of the ball on the x-axis, y ̇_bis the velocity of the ball on the y-axis, I_(b) is the ball's moment of inertia, I_p is the moment of inertia of the plate, α is the angle of the plate on the x-axis, and β is the angle of the plate on the y-axis.

The total kinetic energy of a system is a sum of the kinetic energy of a ball and the kinetic energy of a table and is as follows:

$$T = T_b + T_p = \frac{1}{2}\left(m_{b+}\frac{I_b}{r_b^2}\right)(\dot{x}_b{}^2+\dot{y}_b{}^2) + \frac{1}{2}(I_P + I_b)(\dot{\alpha}^2 + \dot{\beta}^2) + \frac{1}{2}m_b(x_b^2\dot{\alpha}^2 + 2x_b\dot{\alpha}y_b\dot{\beta} + y_b^2\dot{\beta}^2) \tag{5}$$

The total Potential Energy of a system occurs with the ball because the reference is selected as the pivot point of the plate. The potential energy of the system is calculated:

$$V_b = m_b g h = m_b g(x_b sin\,\alpha + y_b sin\beta) \tag{6}$$

where g (m/s^2) is gravity.

Using Equations (5) and (6), The Lagrangian is expressed as:

$$\mathcal{L} = \frac{1}{2}(m_{b+}\ \frac{I_b}{r_b^2})(\ \dot{x}_b{}^2 + \dot{y}_b{}^2) + \frac{1}{2}(I_P + I_b)(\dot{\alpha}^2 + \dot{\beta}^2)$$
$$+ \frac{1}{2} m_b(x_b^2\dot{\alpha}^2 + 2x_b\dot{\alpha}y_b\dot{\beta} + y_b^2\dot{\beta}^2)\ - m_b g(x_b sin\,\alpha + y_b sin\beta) \tag{7}$$

Equations (8), (9), and (10) are calculated by taking the derivatives of Equation (7) according to the time, the position of the ball, and the velocity of the ball.

$$\frac{\partial \mathcal{L}}{\partial \dot{x}_b} = (m_{b+}\ \frac{I_b}{r_b^2})\,\dot{x}_b \tag{8}$$

$$\frac{\partial}{dt}\frac{\partial \mathcal{L}}{\partial \dot{x}_b} = (m_{b+}\ \frac{I_b}{r_b^2})\,\ddot{x}_b \tag{9}$$

$$\frac{\partial \mathcal{L}}{\partial x_b} = m_b(x_b\dot{\alpha} + y_b\dot{\beta})\dot{\alpha} - m_b g sin\,\alpha \tag{10}$$

If the equations (9) and (10) are substituted into equation (1) and rearranged, the dynamical model of the system is obtained as:

$$(m_{b+}\ \frac{I_b}{r_b^2})\,\ddot{x}_b - m_b(x_b\dot{\alpha} + y_b\dot{\beta})\dot{\alpha} + m_b g sin\,\alpha\ = 0 \tag{11}$$

Equation (11) is a non-linear mathematical model. To derive the Transfer Function of the system, linear model is needed. The small deviation of α and β in Equation (12) is linearized as:

$$(m_{b+}\ \frac{I_b}{r_b^2})\ddot{x}_b + \frac{m_b g r_a}{L_p}\theta_x = 0 \tag{12}$$

Since the system is symmetrical, the mathematical model of the y-axis is the same as the x-axis and is as follows:

$$(m_{b+}\ \frac{I_b}{r_b^2})\ddot{y}_b + \frac{m_b g r_a}{L_p}\theta_y = 0 \tag{13}$$

When the Laplace transform is applied to the expression in Equation (12) and $X_b(s)$ and $\theta_x(s)$ are selected as output and input of a system, the transfer function of a system is calculated as:

$$P(s) = \frac{output}{input} = \frac{X_b(s)}{\theta_x(s)} = -\frac{m_b g r_a}{L_p(m_{b+}\frac{I_b}{r_b^2})s^2}\ \frac{rad}{m} \tag{14}$$

According to the parameters in Table 1, the transfer function is defined as:

$$\mathrm{TF} = \mathrm{G(s)}\ = \frac{0.107}{s^2}\frac{deg}{cm} \tag{15}$$

The servo motor was worked in real-time and the data set was obtained as input voltage and output velocity. Using this data, the transfer function of the servo motor was calculated as follows:

$$G_M(s) = \frac{\vartheta(s)}{V_M(s)}\ = \frac{K_M}{\tau s+1}\ =\ \frac{100}{0.01s\ +1} \tag{16}$$

where K_M is the motor gain coefficient, and τ is the time constant of the motor.

2.1.1. PID Controller Design

The mathematical model of the PID controller is as follows:

$$\mathrm{u(t)} = K_p \mathrm{e(t)}\ +\ K_i \int_0^t e(s)ds + K_d \frac{de(t)}{dt} \tag{17}$$

where u(t) is the output of the controller, e(t) is the control error, Kp is the proportional coefficient, Ki is the integral coefficient, and Kd is the derivative coefficient.

The closed-loop block diagram of the system is given in Figure 3 where G(s), and GM(s) is the system's and servo motor's transfer functions. PD controller is selected to control the system.

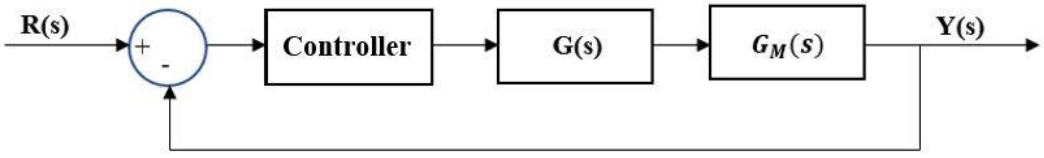

Figure 3. Closed-Loop Block Diagram of the System

The close-loop transfer function and the characteristic equation of the system are given below in Equation (18) and Equation (19), respectively. If the denominator of the closed-loop transfer function is set to zero, the characteristic equation is calculated as follows:

$$\frac{\mathrm{Y(s)}}{\mathrm{R(s)}} = \frac{(K_p + K_d s)(\mathrm{G(s)})(G_M(s))}{1 + (K_p + K_d s)(\mathrm{G(s)})(G_M(s))} \tag{18}$$

$$0.01s^3 + s^2 + 10.7K_d s + 10.7K_p = 0 \quad (19)$$

Since the characteristic equation in (19) is the third order and the general characteristic equation formula is the second order, the factor (a*s +b) is added to the general characteristic equation.

$$P_D(s) = (\mathrm{as} + \mathrm{b})\,(s^2 + 2\xi\omega\mathrm{ns} + {\omega_n}^2) \quad (20)$$

where ξ is the damping ratio, and ω_n is the undamped natural frequency.

"ξ" and "ω_n" values are calculated using "overshoot" and "settling time" parameters. The overshoot value and the settling time are chosen as %2 and 1 second, respectively. The ξ is calculated as 0.7797 and the ω_n value as 5.12. Since the ξ is less than 1, it is expected that the system will oscillate and reach the set point. With the calculated values, the formula is rearranged, and Equality (21) is obtained as follows:

$$P_D(s) = s^2 + 8\mathrm{s} + 26.32 \quad (21)$$

$$P_C(s) = P_D(s) \quad (22)$$

If the equality in (22) is achieved and both equations are equalized, a, b, K_p, and K_d coefficients are obtained as follows:

$$\mathrm{a}=0.01 \qquad \mathrm{b}=0.92 \qquad K_p = 2.45 \qquad K_d = 0.69 \quad (23)$$

As obtained in Equation (23), the "K_p" value was 2.45, and the "K_d" value was 0.69. While the system was running, steady-state error was always detected because of the friction forces that were neglected while creating the dynamic model of the system and the gaps in the mechanical connections. The "I" coefficient of the PID works to eliminate this continuous steady-state error. Therefore, the controller was designed as PD, but the "I" coefficient was experimentally optimized and used as K_i=0.03.

2.2. Image Processing

Image processing is a set of operations for making meaningful inferences from an image. These operations are obtained by using mathematical operations to be performed on the pixels that make up the image. With different algorithms, a lot of information can be obtained from the image and used for different purposes such as control of the mechatronics systems. In this study, image processing is used for object recognition and object tracking. Algorithms are written on the PYTHON.

2.2.1. Color Spaces and Masking

To analyze the concept of color in the digital world and to correspond to a mathematical equivalent, there is more than one characterization method. Each of these methods is called Color Space. Each Color Space is used for different

purposes. The raw image obtained from the camera is in “RGB” Color Space format. In this format, each color is obtained as a result of combining the red, green, and blue colors in different proportions.

However, since different light sources was used in the study, a Color Space with the light intensity (brightness) parameter was chosen. “HSV” Color Space format was selected to solve this problem. In this format, each color was expressed in terms of hue, saturation, and brightness. In this way, different brightness values could be detected, and a protection algorithm could be written in response to light level changes. The ball on the table image from the camera was in “RGB” format. Then the image was converted form “”RGB” to “HSV” format, and was given in Figure 4.

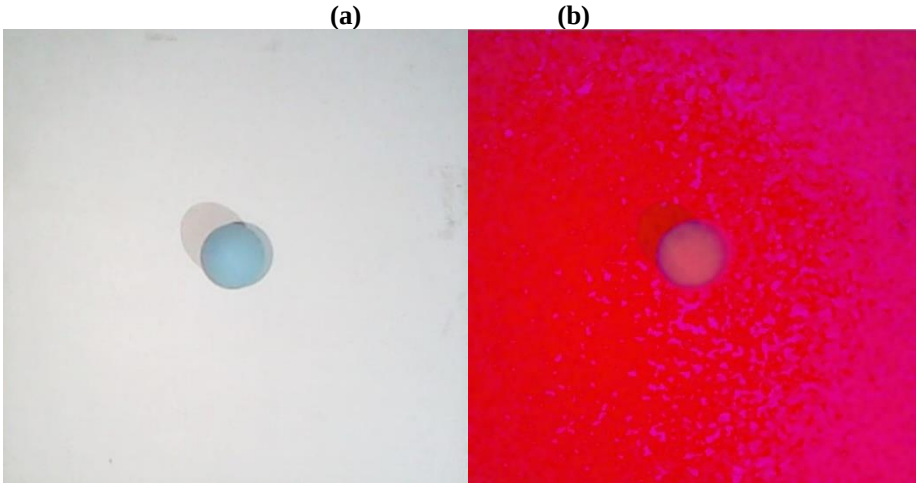

Figure 4. a) “RGB” Color Space Image **b)** “HSV” Color Space Image

In ideal light conditions, the average “HSV” values in pixels containing the ball were measured as (102, 67, 216), respectively. “HSV” threshold values were determined to operate system in different light conditions. . This process was calculated by experimentally measuring the “HSV” values of the pixels of the ball under different light conditions like unstable brightness levels, changeable light color and intensity. Measurements were made under the extreme conditions that could occur, and a more stable algorithm was established in this way. These threshold values were limited to a minimum of (90,50,50) and a maximum of (110, 255, 255) to be “H”, “S”, and “V”, respectively.

To make edge detection operations easier, masking was done with the determined threshold values. The purpose of masking was to make the pixel values of different surface pixels more observable. Pixels within the range (ball) were obtained in white, and pixels out of the range (plane) were obtained in black. Figure 5 shows the image with the masking process.

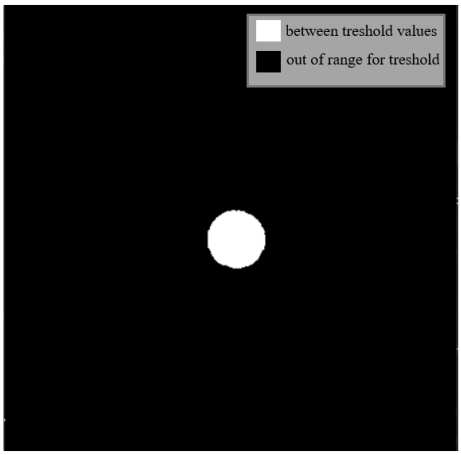

Figure 5. Masked Image

2.2.2. Coordinate Detection

The ball coordinates, which was input to the control algorithm and then used as feedback, were obtained at this stage. To obtain the position information of the ball, the boundaries between the ball and the ground were determined. An edge detection algorithm was used to determine the boundaries. This process was done by frequency analysis and the separation of color regions of different intensities. The function detected the sudden color transition from 0 black pixels to 255 white pixels as border pixels. In this way, the white ball in the image was separated from the black background. The border-image of the spherical ball in the 2D image was a regular circle, and the center point of this circle was the desired coordinate information. To find the center point of the circle, the smallest rectangle that could contain the borders drawn. This rectangle was a square. The center point of the drawn square could be accepted as the center point of the circle. In this way, the coordinate information of the center point of the ball was calculated. In Figure 6, there is the image of the ball whose coordinate was found.

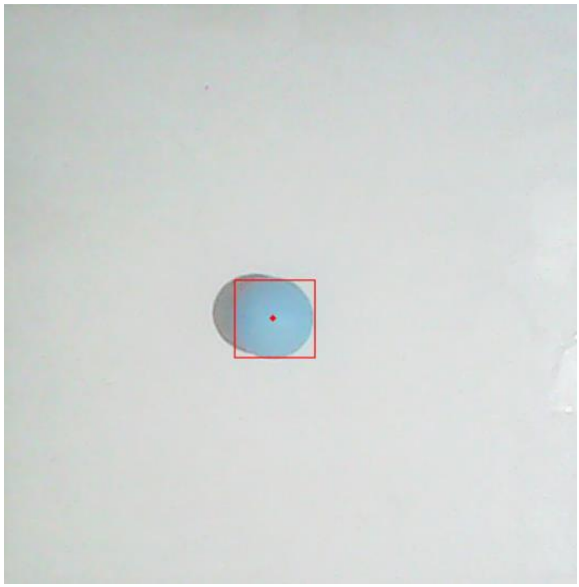

Figure 6. The Center Point of The Ball

3. Results and Discussion

The real-time image taken from the camera is processed with image processing algorithms. The first of these algorithms is pre-processing. The image is resized to the desired size and converted from “RBG” color space to “HSV” format. Then, maximum and minimum threshold values are obtained using the experimentally measured “HSV” pixel values of the ball. Masking processing is done by using threshold values. Finally, the edges of the ball are found by an edge finder function that performs frequency analysis of the pixel values in the masked, black and white image. The center point of the ball is obtained from the found edges.

The input of the transfer function is calculated in “cm” as length units. However, the ball coordinate information is obtained as pixel size. Therefore, an appropriate conversion is made from the pixel size to the length dimension. The 30x30 cm sized plate is measured as 440 pixels in the image from the camera, and the coordinate information is multiplied by a calculated coefficient and converted to “cm”. The coefficient is found as “0.0682” by doing 30 cm divided by 440 pixels.

The position information of the ball is sent to two separate PID controllers for the x and y axes, and the angle values of the servo motors are obtained at the controller output. The whole system works in real-time. The general block diagram of the system is given in Figure 7.

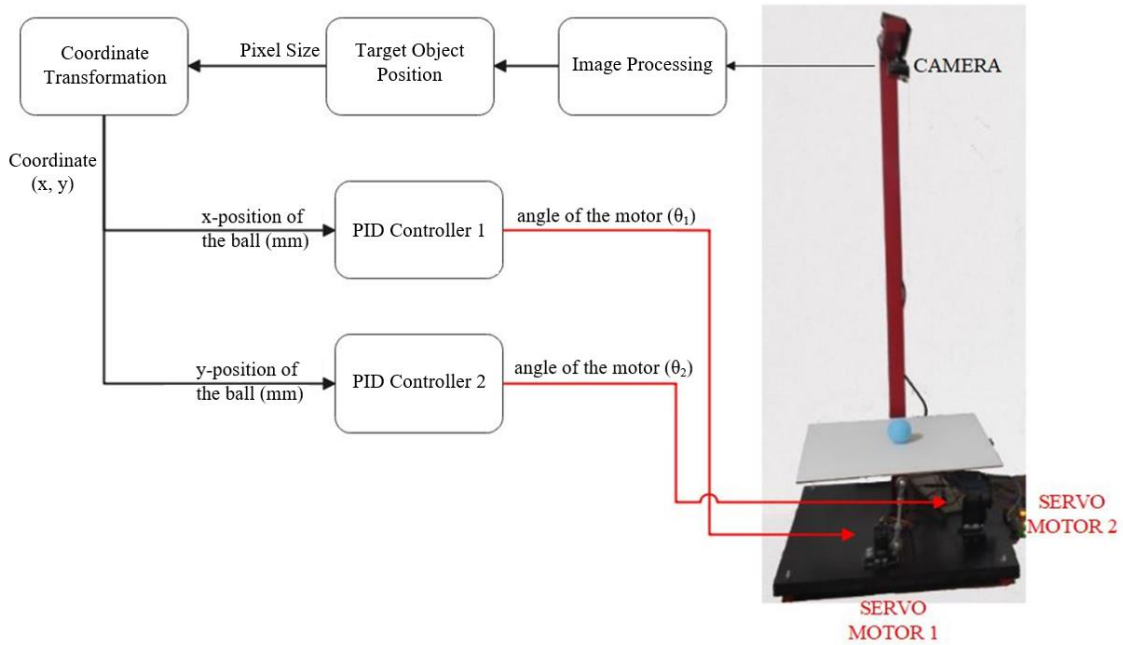

Figure 7. General Block Diagram of System

The Kp and Kd values calculated in Equation (22) are theoretical values that are found for optimum operating conditions in the system. However, the system does not work satisfactorily with these calculated values in real-time. While creating the dynamic model of the system, some forces such as friction were ignored, and it was assumed that the ball would move without slipping. The gap in the structure of the U-joint used at the pivot point caused some losses and errors.

Therefore, the theoretically calculated PID coefficients are different from the most ideal coefficients in practice. The parameters which the system can operate in the most optimized way in practice are tested on the PID controller and areobtained as Kp=2.1, Kd=1.1, and Ki=0.03. In theory, a “PD” controller is designed, but since a “steady-state error” is encountered in the system, a PID controller is used in practice by adding the “integral value (Ki)”.

In the real-time system, the position graphs of the ball are obtained for different setpoints. In Figure 8 and Figure 9, there are position-time graphs on the x-axis and y-axis of the ball that the setpoint is set to be in the middle of the plate (15 cm, 15 cm). In this graph, the Maximum Overshoot for the X-axis is 40.6%, and the Steady State Error is 8%, while the Maximum Overshoot for Y-Axis is 48.6%, and the Steady-State Error is 8.6%.

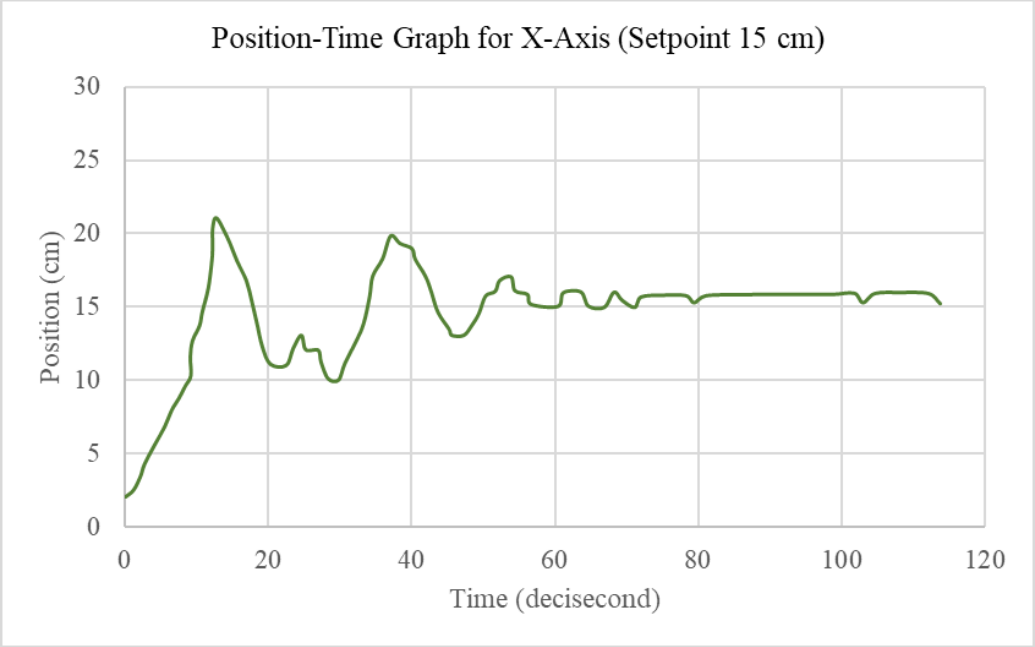

Figure 8. Position-Time Graph for x-Axis (Setpoint 15 cm)

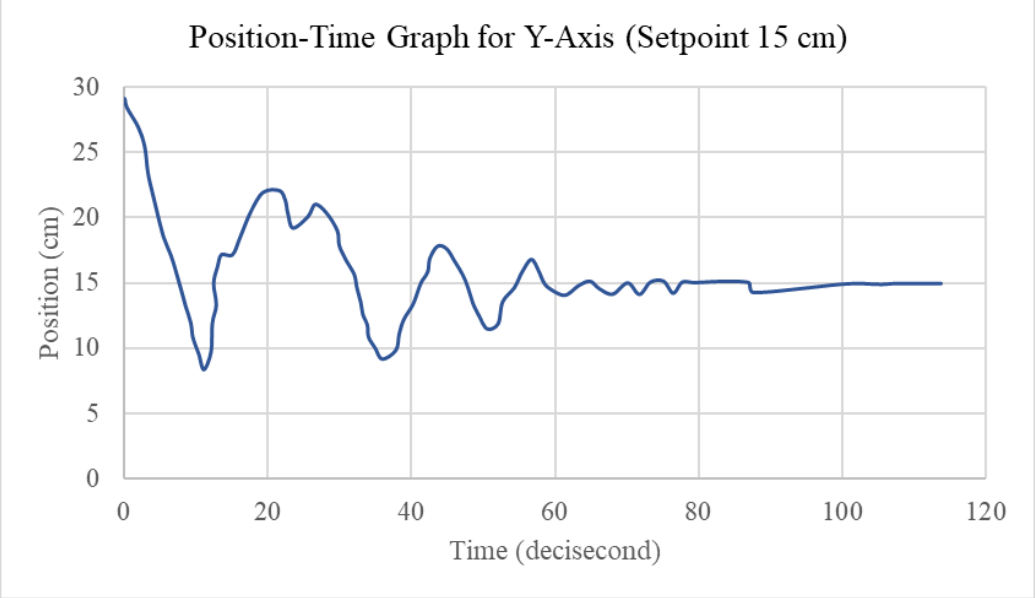

Figure 9. Position-Time Graph for y-Axis (Setpoint 15 cm)

In Figure 10 and Figure 11, there are position-time graphs of the ball on the x-axis and y-axis where the setpoint is set at 10 cm for the x-axis and 10 cm for the y-axis. The Maximum Overshoot for the x-axis is 25%, and the Steady State Error is 5%, while the Maximum Overshoot for y-Axis is 49.5%, and the Steady State Error is 6%.

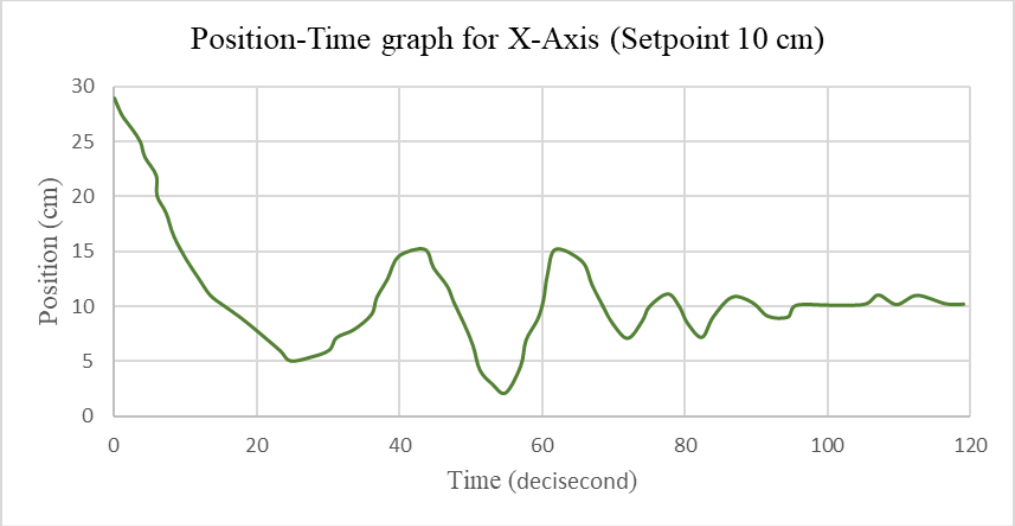

Figure 10. Position-Time Graph for x-Axis (Setpoint 10 cm)

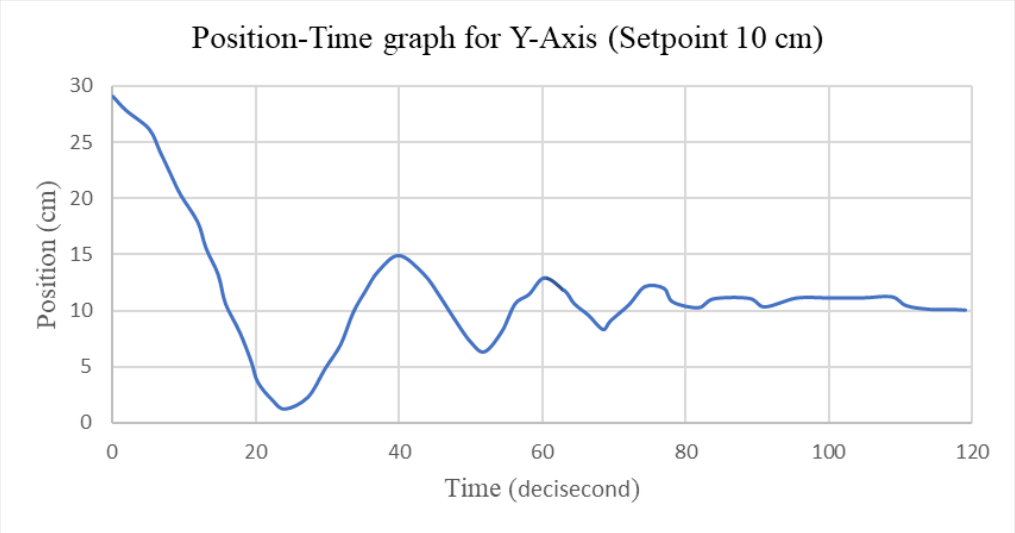

Figure 11. Position-Time Graph for y-Axis (Setpoint 10 cm)

4. Conclusions

In this study, a ball balancing system was designed by using image processing. The system included two Servo Motors as an actuator and a camera as a feedback sensor. The instantaneous coordinates of the ball were found by processing the image taken from the camera.

The dynamic model of the system was calculated using Lagrange-Euler Equationsand the transfer function was obtained to compute the PD coefficients. A PD controller was designed with Kp =2.45 and Kd =0.69. In the real-time

implementation, the PD controller was not satisfied. Thus PID controller was selected as Kp =2.45, Kd =0.69 and Ki= 0.003.

The time-dependent position of the ball was given with figures for each setpoint. When the setpoint was given as the midpoint of the plate, Maximum Overshoot was 25%-49.5%, and the Steady State Error was 5%-6% in the x and y axes, respectively.

The gaps in the U-joint used in the system cause the plate to rotate around its axis. This important mechanical problem causes the system not to work stably in the regions far from the center. If the mechanical problem is eliminated, the system will operate more stable.

NOMENCLATURE

α : Slope of the plate at x-axis
β : Slope of the plate at y-axis
m_b : Mass of the ball
r_b : Radius of the ball
x_b : Position of the ball at x-axis
y_b : Position of the ball at y-axis
$\mathcal{L}$: Lagrange
q_i : Joint variable
T : Kinetic energy of the system
V : Potential energy of the system
Q : Generalized forces
T_b : Kinetic energy of the ball
I_b : Inertia moment of the ball
x_b' : Linear velocity of the ball at the x-axis
y_b' : Linear velocity of the ball at the y-axis
w_x : Angular velocity of the ball at the x-axis
w_y : Angular velocity of the ball at the y-axis
r_a : Arm length of the servo motor
L_p : Length of the plate
θ_x : Angle of the servo motor
ξ : Damping ratio
ω_n : Undamped natural frequency
t_s : Settling time

References

[1] Awtar, S., Bernard, C., Boklund, N., Master, A., Ueda, D., Craig K. (2002). Mechatronic Design Of A Ball-On-Plate Balancing System. Mechatronics, 12(2), 217-228 p.

[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] Itani, A. (2017). Ball Plate Balancing System Using Image Processing.

[4] 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).

[5] Taifour, A., Ahmed, A., Almahdi, H., Osama, A., Naseraldeen, A. (2017). Design and Implementation of Ball and Beam System Using PID Controller. Automatic Control and Information Sciences, 3(1), 1-4 p.

[6] Kocaoğlu, S. (2013). Pıd Kontrollü Top Çubuk Sisteminin Tasarımı ve Kontrolü Üzerine Bir Araştırma.

[7] 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.

[8] 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.

[9] Chen, Z., Gao F.,Sun, Q., Tian, Y., Liu, J., Zhao, Y. (2019). Ball-on-plate motion planning for six-parallel-legged robots walking on irregular terrains using pure haptic information. Mechanism and Machine Theory, Volume 141, 136-150 p.

[10] 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).

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

[12] 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.

[13] Gözde, H. (2019). Evolutionary Computation Based Control For Ball And Plate Stabilization System. Balkan Journal Of Electrıcal & Computer Engıneerıng, 7(1), 45-46 p.

[14] Kuo, B. (1999). Otomatik Kontrol Sistemleri, Litaratür, 7th edition, 88-90 p. ISBN:975-8431-64-1.