Indian Institute of Technology Patna

Mechanical Engineering Department

Mechatronics Branch

MH520 : Mini Project

Ball Balancing Platform

Purusottam Pradhan
2211MT12
Sudeep Sapkota
2211MT15
Kundan Saha
2211MT23

Under Guidance of:
Dr.Atul Thakur
Associate Professor
Department of Mechanical Enggineering

ABSTARCT

This project aims to design and implement a ball balancing platform using a PID controller. The platform consists of a flat surface on which a ball can roll and an servo motor that can adjust the angle of the surface to maintain the ball's position at the center. The PID controller is used to compute the required servo motor horn position based on the difference between the ball's current position and the desired position i.e center of the plate. The project includes the design of the hardware and software components, as well as the tuning of the PID controller parameters for optimal performance. Experimental results show that the ball balancing platform can maintain the ball's position at the center with a high degree of accuracy, demonstrating the effectiveness of the PID controller for this application.

ACKNOWLEDGEMENTS

We would like to express our heartfelt gratitude to all those who have contributed to the success of this ball balancing platform project.

We would first like to thank **Dr. Atul Thakur**, supervisor and course examiner for providing us with valuable guidance and insights throughout the entire process.

We would further like to thank **Yuvraj Kamble** , who helped us with everything, and motivating us to do the project.

A special thank you to **Kadali Mani Shankar** for his unfailing presence in the laboratory and helping us in design part of the project.

Furthermore, we would like to thank our senior **Shankar Kumar** for his help with the control theory.

Finally, we express our appreciation to the **Mechatronics, Instrumentation and Control Laboratory, IIT Patna** where this project was completed for providing us with the necessary resources, facilities, and opportunities to pursue our academic and personal goals.

Contents

1 Introduction 2
- 1.1 Objectives . . . 2
- 1.2 Scope . . . 2

2 Theory 3
- 2.1 Mechanical Dynamics . . . 3
  - 2.1.1 Assumptions . . . 3
  - 2.1.2 Equation of motion . . . 4
  - 2.1.3 Laplace Transform . . . 7
- 2.2 Control Theory . . . 7
  - 2.2.1 PID Controller . . . 7
  - 2.2.2 Proportional Control . . . 8
  - 2.2.3 Integrating control . . . 8
  - 2.2.4 Derivative control . . . 9

3 System Design 10
- 3.1 Hardware . . . 11
  - 3.1.1 The Platform . . . 11
  - 3.1.2 Webcam . . . 12
  - 3.1.3 Servo Motor . . . 13
  - 3.1.4 Arduino . . . 13

4 Implementation 14
- 4.1 Electrical circuit . . . 14
- 4.2 Arduino Programming . . . 15
- 4.3 Servo motor control . . . 15

5 Results 16
- 5.1 Balancing the ball without PID controller . . . 16
- 5.2 Balancing the ball with controller . . . 17

6 Conclusion 18

7 Future Work 19

8 Bibliography 21

9 Appendices 22

9.1 Appendix A 22
9.2 Appendix B 23
9.3 Appendix C 24
9.4 Appendix D 25

List of Figures

2.1 Ball and plate system 4

3.1 Essential System Elements 10
3.2 The fabricated setup 11
3.3 Logitech C270 HD WEBCAM 12
3.4 MG90S Servo Motor 13

4.1 Electrical circuit to be implemented 14

9.1 SolidWorks Model 22

Tables

2.1 Ball Plate system parameters 7

3.1 Platform Details 12
3.2 Camera Details 12
3.3 MG90S servo motor specification 13

1.0 Introduction

The concept of ball balancing has been studied extensively in the field of control engineering. The primary objective of ball balancing is to maintain the position of a ball on a platform by applying appropriate control inputs. This concept finds practical applications in various industries such as robotics, automation, and gaming. One of the most common applications of ball balancing is in the development of self-balancing robots, which require the ability to maintain their stability while in motion.

The aim of this project is to design and implement a ball balancing platform using a microcontroller-based control system. The project is intended to serve as a demonstration of the practical applications of control engineering principles in the development of a real-world system.

1.1 Objectives

The main objective of this project is to design and build a ball balancing platform using a microcontroller-based control system. Specifically, the project aims to achieve the following objectives:

- Design and construct a physical platform capable of balancing a ball at the center of the plate.
- Develop a control system that can maintain the position of the ball on the platform.
- Test and evaluate the performance of the ball balancing platform.

1.2 Scope

The scope of this project is limited to the design and implementation of a ball balancing platform using a microcontroller-based control system. The project will involve the development of a physical platform, control system. The platform will be designed to support a ball of a specific size and weight. The control system will be responsible for maintaining the position of the ball on the platform.

2.0 Theory

The ball balancing platform involves the integration of two distinct theoretical domains: mechanical dynamics and control theory. The mechanical dynamics provides a detailed account of the equations and fundamental mathematical principles that are essential to understanding and solving the ball balancing platform problem.The control design theory explains the connections between the three components of a PID (proportional-integral-derivative) controller and their significance in achieving effective control of the ball balancing platform, including the individual effects of each component.

2.1 Mechanical Dynamics

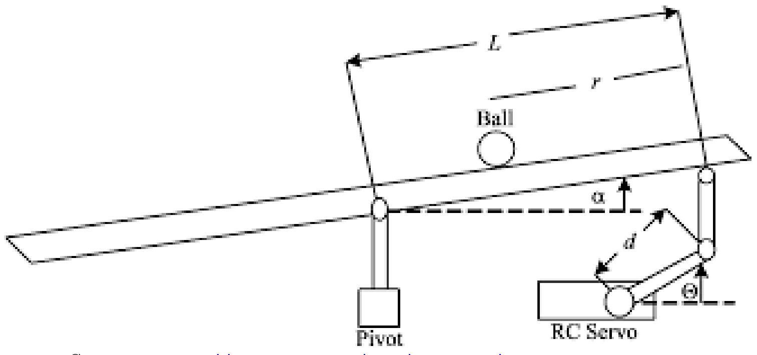
The mathematical model of the ball balancing platform is obtained by applying the laws of physics, which result in one or several differential equations that describe the system's dynamics. The equations of motion, which describe the physical laws governing the system's behavior for the ball balancing platform can be obtained by the Euler-Lagrangian mechanics.

2.1.1 Assumptions

The assumptions that we have taken for getting the equation of motions of the ball on the platform is given below:

- The ball is rolling and not slipping on the platform.
- Platform is friction less.
- The ball is homogeneous and an ideal sphere.
- There is no upward motion of the ball relative to the plate i.e. there is always contact between the plate and the ball.

2.1.2 Equation of motion



Source:https://sciendo.com/pdf/10.4467/2353737XCT.18.077.8559

Figure 2.1: Ball and plate system

The equations of motion for the ball-plate system[Fig2.1] can be obtained using the Lagrangian formalism. The Euler-Lagrange equation of ball-plate system is

$$\frac{d}{dt}\frac{\partial T}{\partial \dot{q}_i} - \frac{\partial T}{\partial q_i} + \frac{\partial V}{\partial q_i} = Q_i \tag{2.1}$$

Where:
q_i stands for i-direction coordinate
T kinetic energy of the ball-plate system.
V potential energy of the ball-plate system.
Q_i composite force.

Since our ball-plate system is a two degrees of freedom system, we can describe the position of the ball on the plate using the coordinates (x,y), and the orientation of the plate using the angles α and β. We can assume that the center of the x-y coordinates corresponds to the center of the plate.

The kinetic energy of the ball consists of translational energy and rotational energy with respect to its center

$$T_b = \frac{1}{2}m_b\left(\dot{x_b}^2 + \dot{y_b}^2\right) + \frac{1}{2}I_b\left(\omega_x^2 + \omega_y^2\right) \tag{2.2}$$

Where:
m_b is mass of the ball.
I_b is moment of inertia of the ball.
$\dot{x_b}$ is translational velocity of ball along x-axis.
$\dot{y_b}$ is translational velocity of ball along y-axis.
ω_x is rotational velocity of the ball along x-axis.
ω_y is rotational velocity of the ball along y-axis.
r_b is the radius of the ball.

The relation between rotational and translational velocities of the ball is given by

$$\dot{x_b} = r_b \omega_x \tag{2.3}$$

$$\dot{y_b} = r_b \omega_y \tag{2.4}$$

By substituting equation (2.3) and (2.4) into equation (2.2), we get

$$T_b = \frac{1}{2}\left[m_b\left(\dot{x_b}^2 + \dot{y_b}^2\right) + \frac{I_b}{r_b^2}b\left(\dot{x_b}^2 + \dot{y_b}^2\right)\right] = \frac{1}{2}\left(m_b + \frac{I_b}{r_b^2}\right)\left(\dot{x_b}^2 + \dot{y_b}^2\right) \tag{2.5}$$

The kinetic energy of the plate consists of its rotational energy with respect to its center of mass. The equation is as follow

$$T_p = \frac{1}{2}\left(I_p + I_b\right)\left(\dot{\alpha}^2 + \dot{\beta}^2\right) + \frac{1}{2}m_b\left(x_b\dot{\alpha} + y_b\dot{\beta}\right) \tag{2.6}$$

$$T_p = \frac{1}{2}\left(I_p + I_b\right)\left(\dot{\alpha}^2 + \dot{\beta}^2\right) + \frac{1}{2}m_b\left(x_b^2\dot{\alpha}^2 + 2x_b\dot{\alpha}y_b\dot{\beta} + y_b^2\dot{\beta}^2\right) \tag{2.7}$$

Where x_b position of the ball on the plate with respect to x-axis.
y_b position of the ball on the plate with respect to y-axis.
α angle of inclination of the plate along x-axis.
β angle of inclination of the plate along y-axis.

Now the total kinetic energy of the whole system can be written as

$$T = T_b + T_p = \frac{1}{2}\left(m_b + \frac{I_b}{r_b^2}\right)\left(\dot{x_b}^2 + \dot{y_b}^2\right) + \frac{1}{2}\left(I_p + I_b\right)\left(\dot{\alpha}^2 + \dot{\beta}^2\right) + \frac{1}{2}m_b\left(x_b^2\dot{\alpha}^2 + 2x_b\dot{\alpha}y_b\dot{\beta} + y_b^2\dot{\beta}^2\right) \tag{2.8}$$

The potential energy of the ball can be calculated as

$$V_b = m_b g h = m_b g\left(x_b sin\alpha + y_b sin\beta\right) \tag{2.9}$$

And the Lagrangian L of the system can be written as

$$L = T_p + T_b - V_b \tag{2.10}$$

Assume generalized torques as τ_x and τ_y which are exerted torques on the horizontal plate.

Using Euler- Lagrange equation:

$$\frac{d}{dt}\frac{\partial T}{\partial \dot{\alpha}} - \frac{\partial L}{\partial \alpha} = \left(I_p + I_b\right)\ddot{\alpha} + m_b x_b^2\ddot{\alpha} + 2m_b x_b\dot{x}_b\dot{\alpha} + m_b x_b y_b\ddot{\alpha} + m_b\dot{x}_b y_b\dot{\beta} + m_b x_b\dot{y}_b\dot{\beta} - mgcos\alpha = \tau_x \tag{2.11}$$

$$\frac{d}{dt}\frac{\partial T}{\partial \dot{\beta}} - \frac{\partial L}{\partial \beta} = \left(I_p + I_b\right)\ddot{\beta} + m_b y_b^2\ddot{\beta} + 2m_b y_b\dot{y}_b\dot{\alpha} + m_b x_b y_b\ddot{\beta} + m_b\dot{y}_b m_b\dot{\alpha} + m_b y_b\dot{x}_b\dot{\alpha} - mgcos\beta = \tau_y \tag{2.12}$$

$$\frac{d}{dt}\frac{\partial T}{\partial \dot{x}_b} - \frac{\partial L}{\partial x_b} = \left(m_b + \frac{I_b}{r_b^2}\right)\ddot{x}_b - m_b\left(x_b\dot{\alpha} + y_b\dot{\beta}\right)\dot{\alpha} + m_b g sin\alpha = 0 \tag{2.13}$$

$$\frac{d}{dt}\frac{\partial T}{\partial \dot{y}_b} - \frac{\partial L}{\partial y_b} = \left(m_b + \frac{I_b}{r_b^2}\right)\ddot{y}_b - m_b\left(y_b\dot{\beta} + x_b\dot{\alpha}\right)\dot{\beta} + m_b g sin\beta = 0 \tag{2.14}$$

So the non-linear differential equations of motion for the ball-plate-system are as follows:

$$\left(m_b + \frac{I_b}{r_b^2}\right)\ddot{x}_b - m_b\left(x_b\dot{\alpha} + y_b\dot{\beta}\right)\dot{\alpha} + m_b g sin\alpha = 0 \tag{2.15}$$

$$\left(m_b + \frac{I_b}{r_b^2}\right)\ddot{y}_b - m_b\left(y_b\dot{\beta} + x_b\dot{\alpha}\right)\dot{\beta} + m_b g sin\beta = 0 \tag{2.16}$$

$$\tau_x = \left(I_p + I_b + m_b x_b^2\right)\alpha + 2m_b x_b \dot{x}_b \dot{\alpha} + m_b x_b y_b \alpha + m_b \dot{x}_b y_b \dot{\beta} + m_b x_b \dot{y}_b \dot{\beta} - mgcos\alpha \tag{2.17}$$

$$\tau_y = \left(I_p + I_b + m_b y_b^2\right)\beta + 2m_b y_b \dot{y}_b \dot{\alpha} + m_b x_b y_b \beta + m_b \dot{y}_b x_b \dot{\alpha} + m_b y_b \dot{x}_b \dot{\alpha} - mgcos\beta \tag{2.18}$$

The equation (2.15) and (2.16)shows the relation between plate's state and ball's state which represent the plate's inclination.
The equation (2.17) and (2.18) shows the effect of external torque on the plate-ball system.

The relationship between the plate inclination angle and the servo horn inclination angle is given by

$$\alpha = \frac{r_a}{L_p}\theta_x \tag{2.19}$$

$$\beta = \frac{r_a}{L_p}\theta_y \tag{2.20}$$

When the angular velocity $\dot{\alpha}$ and $\dot{\beta}$ is low, it can be approximated as follows:
$\dot{\alpha}\dot{\beta}$=0, $\dot{\alpha}^2$=0, $\dot{\beta}^2$=0

By linearizing the equations of motion for $\theta_x = 0$ and $\theta_y = 0$, and after substituting equation (2.19) and (2.20) into motion equations (2.15) and (2.16), the result is as follows:

$$\left(m_b + \frac{I_b}{r_b^2}\right)\ddot{x}_b + \frac{m_b g r_a}{L_p}\theta_x = 0 \tag{2.21}$$

$$\left(m_b + \frac{I_b}{r_b^2}\right)\ddot{y}_b + \frac{m_b g r_a}{L_p}\theta_y = 0 \tag{2.22}$$

Since the ball used is a solid sphere ball, then the moment of inertia of the ball is equal to

$$I_b = \frac{2}{5}m_b r_b^2$$

Further, if the complete expression of the moment of inertia for the ball, Ib is inserted, the linearized equation can be written as follows below.Noteworthy is that the theoretical system is independent of the mass as well as of the radius of the ball.

$$\ddot{x}_b = \frac{3}{5} g\alpha \tag{2.23}$$

$$\ddot{y}_b = \frac{3}{5} g\beta \tag{2.24}$$

2.1.3 Laplace Transform

Equations (2.23) and (2.24) is defining the dynamics of the system within the time domain. To allow controller design of the system, these equations need to be translated into the frequency domain using Laplace transformation. Now the above two equations in the frequency domain can be written as

$$s^2 X = \frac{3}{5} gA \tag{2.25}$$

$$s^2 Y = \frac{3}{5} gB \tag{2.26}$$

mass of the ball(m_b)	0.003kg(Hallow table tennis ball) 0.02177kg(water filled TT ball)
radius of the ball(r_b)	0.015m
Inertial of the ball(I_b)	0.00003kg.m^2(hollow)
Center to edge distance(Plate)	125mm
Servo horn length	1.5mm

Table 2.1: Ball Plate system parameters

2.2 Control Theory

To regulate a system's output by adjusting its input, engineers often use a PID (Proportional-Integral-Derivative) controller. In this section its different parts will be explained in details.

2.2.1 PID Controller

A PID (Proportional-Integral-Derivative) controller is a type of feedback controller that is widely used in engineering applications to regulate a system's output by adjusting its input. The controller works by continuously measuring the error between the desired setpoint and the actual output of the system, and generating a control signal that adjusts the system's input in response to this error.

The PID controller has three distinct components: proportional, integral, and derivative. The proportional component generates a control signal that is proportional to the error signal, while the integral component integrates the error over time to reduce steady-state errors. The derivative component calculates the rate of change of the error signal to improve the controller's response to changes in the system's output.

The output of the controller is the sum of the contributions of the proportional, integral, and derivative components, each multiplied by a corresponding gain factor. The proportional, integral, and derivative gains can be tuned to achieve the desired level of control performance.The regulated output signal, discussed in detail below, is given by equation

$$u(t) = K_p e(t) + K_i \int_0^t e(t)dt + K_d \frac{de(t)}{dt} \tag{2.27}$$

2.2.2 Proportional Control

The proportional controller is designed to reduce the steady-state error of a control system by adjusting the proportional gain of the system. The proportional gain, denoted as KP, is a constant factor that determines the response of the system to the error signal. As the proportional gain is increased, the proportional response of the system is also increased, resulting in a decrease in the steady-state error.The output signal of the proportional controller is given by the product of the error signal and the proportional gain, as shown in the equation below:

$$u_p(t) = K_p e(t) \tag{2.28}$$

In other words, the proportional term or output signal of the controller is directly proportional to the error signal, and the proportionality constant is the proportional gain KP. Therefore, increasing KP amplifies the controller's response to the error signal, reducing the steady-state error of the system.

2.2.3 Integrating control

The integral controller in a control system is designed to correct the accumulated offset that may be caused by the system's response to changes in the input signal. It does this by integrating the error signal over time, taking into account both the magnitude and the duration of the error.

By summing the error signal over time, the integral controller can eliminate the steady-state error that may be present in the system. However, it is important to note that setting the integral gain, KI, too high can cause the system's response to become worse. This is because the integral term may cause the system to exhibit transient or oscillatory behavior, leading to instability in the control system. Therefore, it is important to strike a balance between the proportional and integral gains of the controller to achieve the desired level of control performance.

$$u_i(t) = K_i \int_0^t e(t)dt \tag{2.29}$$

2.2.4 Derivative control

The derivative controller is an essential component of the PID controller, which is used to improve the performance of a control system. The derivative controller calculates the rate of change of the error signal with respect to time, also known as the error response slope. The derivative gain, KD, is then multiplied by the error response slope to produce the derivative term of the controller.

The derivative term can be mathematically represented by the following equation:

$$u_d(t) = K_d \frac{de(t)}{dt} \tag{2.30}$$

In other words, the derivative term is proportional to the rate of change of the error signal, and the proportionality constant is the derivative gain KD. By including the derivative term in the PID controller, the controller can effectively anticipate changes in the error signal and react accordingly. This leads to a more responsive control system and helps to reduce overshoot and settling time.

However, it is important to note that setting the derivative gain too high can cause the control system to become unstable, leading to oscillations or even system failure. Therefore, it is necessary to choose an appropriate value for the derivative gain to achieve the desired level of control performance.

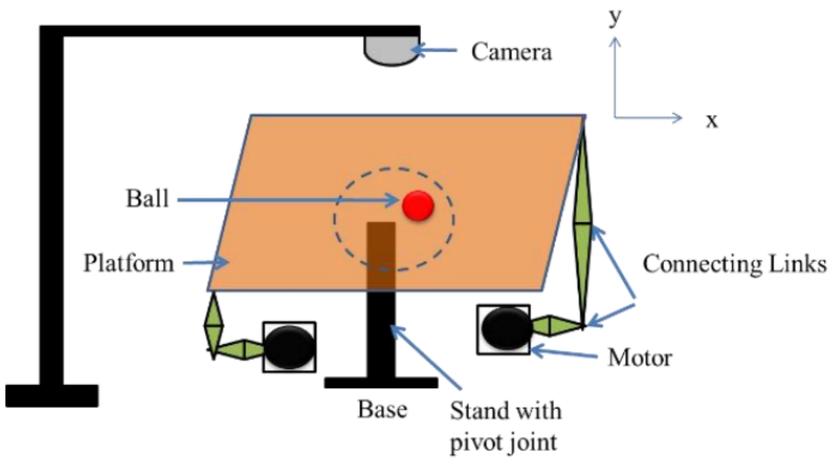
3.0 System Design

In order to achieve the objective of stabilizing a ball on a flat surface, the system needs to be able to adjust the inclination of the platform around both the x-axis and y-axis. This is accomplished using two servo motors that are connected to the platform via two pairs of ball joint arms.

The servo motors are used to rotate the platform around the x-axis and y-axis, respectively, in order to adjust the angle of the surface and control the position of the ball. The pairs of arms are used to transmit the rotational force from the servo motors to the platform, allowing precise control of the inclination of the surface in two dimensions.

By adjusting the inclination of the platform in real-time using the servo motors, the control system can maintain the ball in a stable position on the surface. This is achieved by constantly monitoring the position of the ball and adjusting the inclination of the surface as necessary to counteract any external disturbances or variations in the ball's position.

A web cam will provide ball position as input to the microcontroller and as of the platforms inclination, servo motor will be used to calculate the necessary output.



Source:

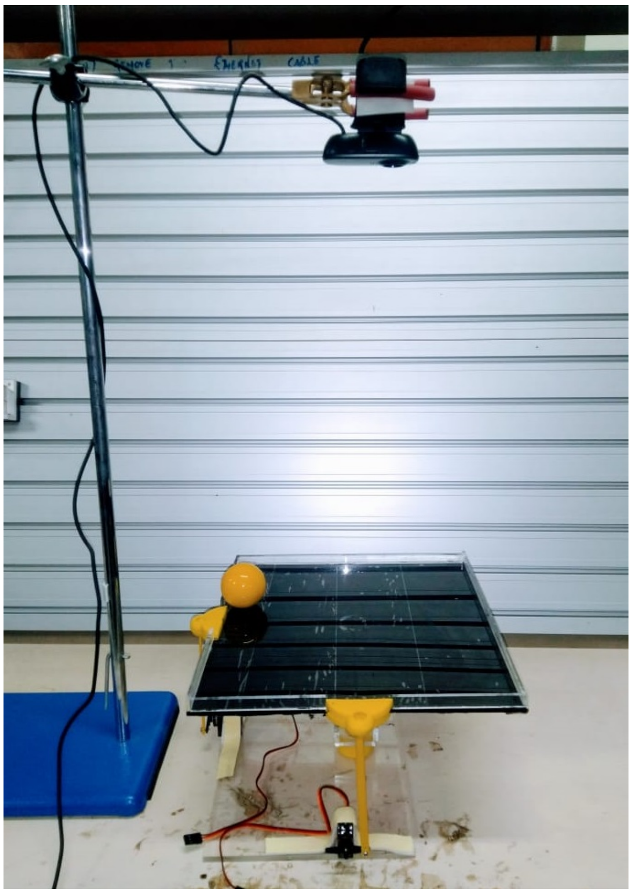
Figure 3.1: Essential System Elements

3.1 Hardware

The central working parts of the physical model, are the fabricated Platform, a webcam, two servo motors and a Arduino Uno microcontroller. The webcam provides the position of the ball, which act as the input data, the microcontroller processes the input and generates an output sent to the servo motors to generate physical motion.

3.1.1 The Platform

This is the platform which we have fabricated in the MICL Lab,IIT Patna, which consists of a webcam, two servomotor, a platform which is made up off acrylic sheet, a universal joint, and the ball joints/spherical joints which we have fabricated using 3D printing.



Source:

Figure 3.2: The fabricated setup

The platform details are given below:

Thickness	3mm
Length and Breadth	125mm
Platform Material	Acrylic
Area of the plate	$0.0625m^2$
Weight	0.0225kg

Table 3.1: Platform Details

3.1.2 Webcam

To determine the position of the ball a Logitech C270 HD WEBCAM is used, which is mounted over the platform as shown in the Fig[3.2]. And the detailed view of the webcam is given below [Fig3.3]. Using this webcam we get the position of the ball, which is achieved by image processing. And the Image processing MATLAB code is given in the Appendix.

Source:

Figure 3.3: Logitech C270 HD WEBCAM

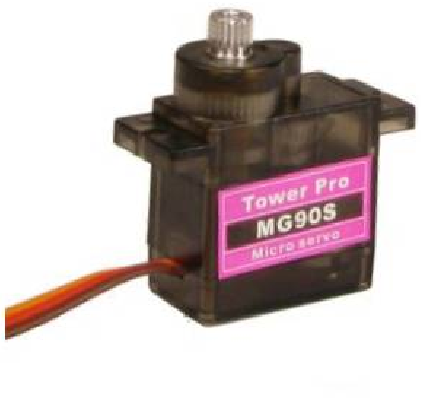
The camera details are given below

Model	Logitech C270 HD WEBCAM
Max Resolution	720p/30fps
Camera megapixel	0.9
The diagonal field of view (dFoV)	55°

Table 3.2: Camera Details

3.1.3 Servo Motor

In our project we have used the MG90S servo motors[Fig3.4] as the actuator. By the use of two MG90S servo motor we adjust the inclination of the plate. And the code for controlling the servo motor is given in the Appendix.



Source:

Figure 3.4: MG90S Servo Motor

The MG90S Servo Motor specifications are given below:

Model	MG90S
Operating voltage	4.8V 6.6V
Stall torque @4.8V	1.8kg-cm
Stall torque @6.6V	2.2kg-cm
Rotation	0°-180°
Weight of motor	13.4gm

Table 3.3: MG90S servo motor specification

3.1.4 Arduino

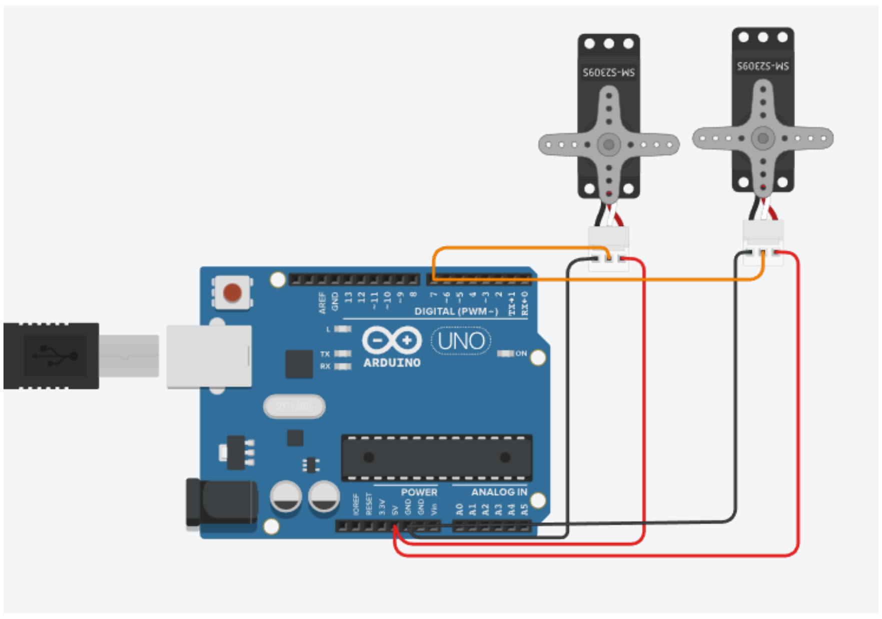
The core of system revolves around the microcontroller Arduino Uno based ATmega328P, programmable through Arduino's own IDE. The Arduino Uno possess fourteen digital outputs and inputs, whereof six are pulse width modulating. An additional 6 inputs are analog. Some other features are 16 MHz quartz crystal, USB connection to power jack, reset button and in circuit serial programming header.

4.0 Implementation

The process of programming the physical system model involves accounting for the differences between theoretical predictions and actual outcomes. Various parameters and simplifications may be disregarded during the implementation process, but these are later compensated for in order to achieve the desired response.It is important to note that this process is repeated for each rotational axis. In other words, the methodology is applied separately to each axis in order to ensure that the final physical system model accurately reflects the desired response.

4.1 Electrical circuit

The electrical circuitin for our project is shown in the Fig4.1 and the respective components. Besides wiring the components are the Arduino Uno microprocessor, two servo motors, a small breadboard can be used. The Arduino powered through USB and the servo motors from power source stepped up to 5V.



Source:

Figure 4.1: Electrical circuit to be implemented

4.2 Arduino Programming

We have taken the position of the ball as the input using image processing which is done in MATLAB. And using this co-ordinates of the ball, according to the position of the ball we have actuate the servo motor arm. We have program the Arduino using MATLAB and the complete program is given in the Appendix.

4.3 Servo motor control

The error control signal is sent to each servo motor independently.The control signal enables angular displacement of the motors by utilization of the servo library of the MATLAB. Each motor is calibrated to maintain a horizontal platform initially.

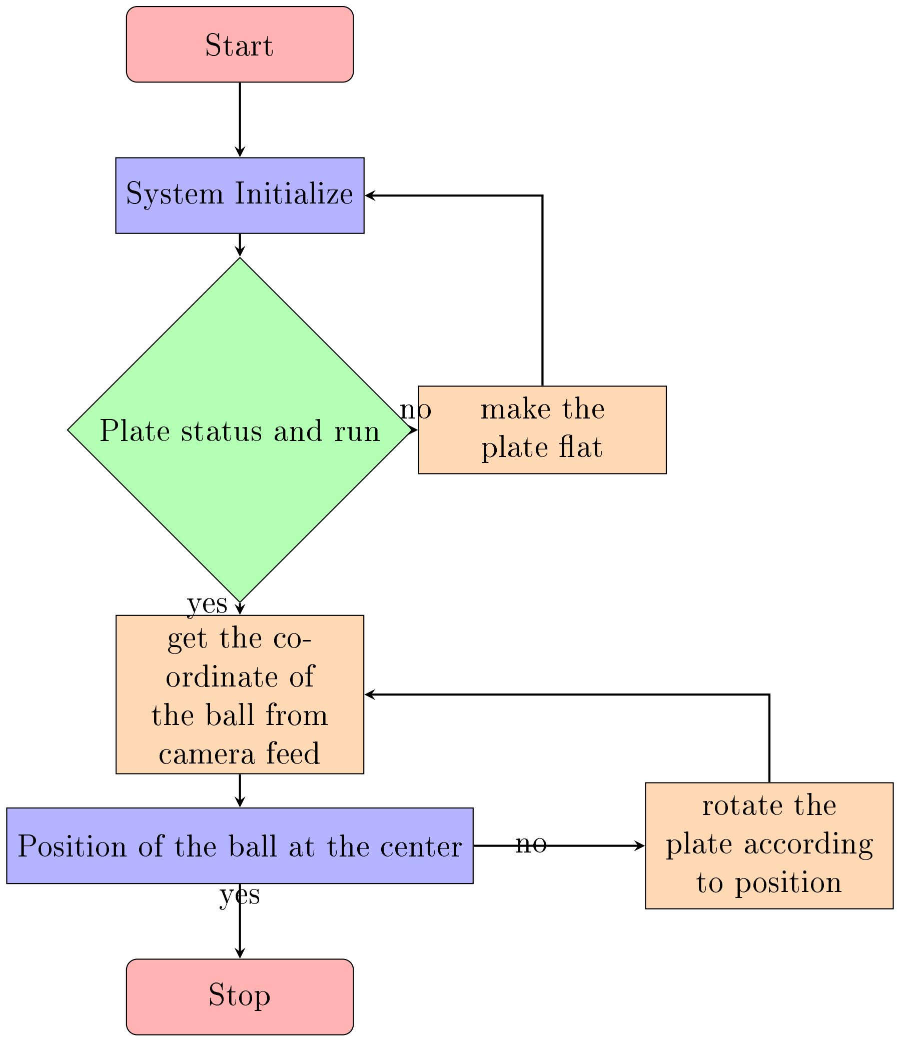
5.0 Results

5.1 Balancing the ball without PID controller

In this Ball balancing without PID controller we have written some rules for the system such that it can balance the ball at the center of the plate. In this we achieve the ball balancing objective by tuning the maximum and minimum angle the plate can tilt. By this method we get the ball balanced but it takes more time in our case it takes around 30sec. The video link for the experiment is given below. The steps for the above problems are given below. And the MATLAB code corresponding to this problem is given in the Appendix.

Link for the video:
https://drive.google.com/file/d/11QNlLAo9diwM3cgUX_ffhqHuL1rlzaAj/view?usp=sharing

The flow chart for the above problem is given below


Start
System Initialize
Plate status and run
no
make the plate flat
yes
get the co-ordinate of the ball from camera feed
Position of the ball at the center
no
rotate the plate according to position
yes
Stop

5.2 Balancing the ball with controller

Our team encountered a significant issue while attempting to implement a PID (Proportional-Integral-Derivative) control system for the ball balancing platform project. Specifically, we faced problems with the image processing component, which prevented us from continuously tracking the position of the ball in real-time. As a result, we were unable to provide the necessary input to the PID controller, rendering it ineffective. This issue is critical to the success of the project, as it is impossible to keep the ball balanced without accurate and timely position tracking. Despite our best efforts, we have not been able to resolve the issue, and we are currently exploring different approaches to improve our image processing capabilities. We understand the importance of this project and are committed to finding a solution that will allow us to move forward and achieve our goals.

6.0 Conclusion

In conclusion, our team has made significant progress in the ball balancing platform project by fabricating the model. However, we faced an unexpected obstacle in implementing a PID control system due to issues with the image feed. Despite our best efforts, we were unable to overcome this challenge, which had a significant impact on the project's progress. Nevertheless, we are committed to continuing our work on the project and finding a solution to this problem. We believe that with further investigation and exploration of different approaches, we can overcome this obstacle and achieve our goals for the ball balancing platform project. While this setback was unexpected and challenging, we are confident that we can find a way to move forward and achieve success. Our team is dedicated to this project and will continue to work tirelessly to overcome this challenge and achieve our objectives.

7.0 Future Work

Several future works and improvements are necessary to enhance the ball balancing platform's controller design and construction. One of the most crucial steps is to incorporate powerful and precise servo motors that can provide better control over the plate's movements. The use of consistent servo motor brands will ensure that the platform responds accurately and consistently to input commands. This improvement will significantly enhance the stability and reliability of the platform's movement control, enabling the ball to balance more effectively. In summary, enhancing the servo motors' power and precision will be a significant step towards achieving better control and performance of the ball balancing platform.

In terms of construction, it is essential to address the issue of play in the joints and links and improve the tolerances of the system. This problem is a significant factor in the shattering of the touch panel or platform and is primarily caused by the servo motors. However, the play is further compounded by the insufficient tolerances and clearances in the joints. Therefore, to ensure a more stable and robust platform, we need to focus on improving the precision and accuracy of the manufacturing process, including the use of higher-quality materials and tighter tolerances in the joints and links. By addressing these construction issues, we can reduce the play in the system, which will lead to a more reliable and durable ball balancing platform.

The ball balancing platform project faced limitations in achieving completely accurate results and data readings due to slow serial communication between the Arduino and the computer through a USB connection. This slow communication resulted in a slower response of the servo motors, which made it challenging to maintain the ball's balance on the platform. To address this issue, we need to investigate alternative methods of data extraction and analysis that are more efficient and faster. This would allow for more accurate and timely readings, leading to better analysis and discussion of the results. However, due to time constraints, we were unable to explore these alternative methods in the scope of this project. Nevertheless, it is an area that warrants further investigation and could significantly enhance the overall performance of the ball balancing platform.

In the future, it is essential to address the camera feed problem in the ball balancing platform project. One potential solution is to explore the use of alternative camera systems or sensors that can provide a more continuous and stable image feed. Additionally, we could investigate methods of image processing and analysis that are better suited to handle the complexity of the platform's movement control system. For instance, we could explore the use of machine learning algorithms to improve the real-time tracking and position detection of the ball. Another approach is to optimize the design and placement of the camera to ensure optimal viewing angles and lighting conditions. This could involve exploring different camera models and positioning techniques to achieve the best possible image quality and feed stability. Overall, addressing the camera feed problem will be critical to improving the performance and accuracy of the ball balancing platform, and it is an area that warrants further investigation and experimentation in future work.

8.0 Bibliography

1. ALEXANDER HASP FRANK, MORGAN TJERNSTRÖM, Construction and theoretical study of a ball balancing platform, KTH ROYAL INSTITUTE OF TECHNOLOGY SCHOOL OF INDUSTRIAL ENGINEERING AND MANAGEMENT.
   https://www.diva-portal.org/smash/get/diva2:1373784/FULLTEXT01.pdf

2. Meriem Hamdoun1, Mohamed Ben Abdallah, Mounir Ayadi, Frédéric Rotella,Irène Zambettakis, Functional observer-based feedback controller for ball balancing table.

3. Ahmed Itani, BALL PLATE BALANCING SYSTEM USING IMAGE PROCESSING, GRADUATE SCHOOL OF APPLIED SCIENCES OF NEAR EAST UNIVERSITY.

4. Volodymyr Samotyy, Sergii Telenyk, Petro Kravets ,Volodymyr Shymkovych, Taras Posvistak, A real time control system for balancing a ball on a platform with FPGA parallel implementation.

5. Tom Trapp and Wil Selby, Ball and Beam Balance, Massachusetts Institute of Technology.
   https://s3-us-west-2.amazonaws.com/selbystorage/wp-content/uploads/2016/05/SelbyTrappFInalReport.pdf

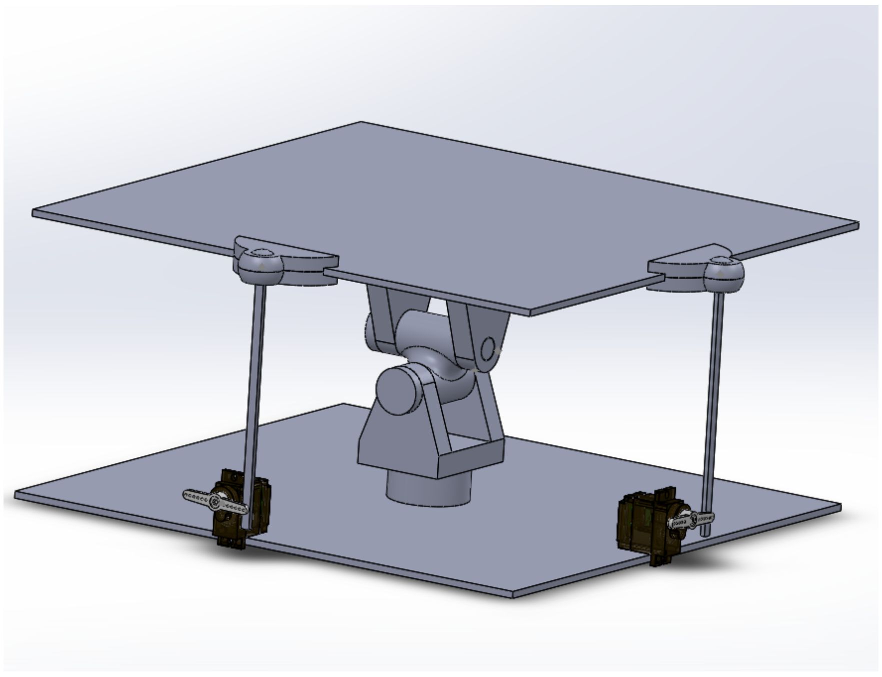
6. Ehsan Lakzaei, Balancing ball on a plate, Eindhoven University of Technology
   https://www.researchgate.net/publication/344687938_Balancing_ball_on_a_plate-Final
   Report

7. Dr. Pranav Bhounsule, Geoffrey Chiou, Andy Plascencia, Tyler Rowe, BALANCING A BALL AND BEAM WITH PID, The University of Texas at San Antonio
   https://pab47.github.io/reports/Beam_Ball_Balancing.pdf

9.0 Appendices

9.1 Appendix A

Solidworks Model



Source:

Figure 9.1: SolidWorks Model

9.2 Appendix B

Image Processing code

```
clear;clc;

    cam=webcam("Logi C310 HD WebCam");
path=[];

    while true
e=snapshot(cam);
e=rgb2gray(e);
e=imcrop(e,[0 0 420 420]);
imshow(e)

    [center,radius]=imfindcircles(e,[30,80],"ObjectPolarity","bright","Sensitivity",0.9,"EdgeThreshold",
viscircles(center,radius);
viscircles([210 210],20);
center;
path=[path;center];

    end
```

9.3 Appendix C

Servo Calibration code

```
clear all;
clc;

    a = arduino('COM6', 'Uno', 'Libraries', 'Servo');
servo1 = servo(a, 'D4');
servo2=servo(a,'D7');
servo1_cal_angle=58;
servo2_cal_angle=69;
alpha1=0;
alpha2=0;
writePosition(servo1,(1/180)*(servo1_cal_angle+alpha1));
writePosition(servo2,(1/180)*(servo2_cal_angle+alpha2));
```

9.4 Appendix D

Controller Code

```
clear all;
close all;
clc;

    cam=webcam("Logi C310 HD WebCam");%calling the webcam
%for craeting the path list later on
path=[];

    % capturing an image before balancing
e=snapshot(cam);
e=rgb2gray(e);
e=imcrop(e,[0 0 420 420]);
imshow(e);
[center,radius]=imfindcircles(e,[30,80],"ObjectPolarity","bright","Sensitivity",0.9,"EdgeThreshold",0.3
viscircles(center,radius);

    a = arduino('COM6', 'Uno', 'Libraries', 'Servo');% calling the arduino board
servo1 = servo(a, 'D4');% servo motor 1 connected to the 4th port
servo2 = servo(a,'D7'); % servo motor 2 connected to the 7th port

    %servo motor calibrated angle
servo1_cal_angle=58;
servo2_cal_angle=69;

    %keeping the plate flat before balancing the plate
theta1=0;
theta2=0;
writePosition(servo1,(1/180)*(servo1_cal_angle+theta1));
writePosition(servo2,(1/180)*(servo2_cal_angle+theta2));

    %defining the inital pre error value for the derivative control
pre_error_x=0;
pre_error_y=0;

    z=0;

    while true
% capturing contineous image of the ball for image processing
e=snapshot(cam);
e=rgb2gray(e);
```

```
e=imcrop(e,[0 0 420 420]);
figure(1);
hold on;
imshow(e);
%capturing the location of the centers of the ball while it is travelling
[center,radius]=imfindcircles(e,[30,80],"ObjectPolarity","bright","Sensitivity",0.9,"EdgeThreshold",0.3
viscircles(center,radius);
%defining the center of the plate and putting a circle in the center
%location
viscircles([210 210],40);
center;
path=[path;center];

    %defining the origin of the plate
x_coordinates=[210;210;210;];
y_coordinates=[210;210;210;];

    %capturing the x-coordinate and y=coordinate of the ball
x_coordinates=[x_coordinates;path(:,1)];
y_coordinates=[y_coordinates;path(:,2)];

    %plotting the center of the ball

    plot(x_coordinates(end),y_coordinates(end),'m*','MarkerSize',10);

    hold off;

    %distance from the center of the plate in the x-direction
error_x=x_coordinates(end)-210;

    %distance from the center of the plate in the y-direction
error_y=y_coordinates(end)-210;

    %defining the distance from the center of the plate
distance=((error_x)2̂+(error_y)2̂)0̂.5;

    %————————————————————————————————————-%
%Implementing proportional-integral-derivative control

    k_p_x=0.036;
k_p_y=0.029;

    k_d_x=0.01;
k_d_y=0.01;
```

```
    %for derivative control
delta_x=(error_x-pre_error_x);

    delta_y=(error_y-pre_error_y);

    theta1=k_p_x*error_x+k_d_x*delta_x;
theta2=k_p_y*error_y+k_d_y*delta_y;

    writePosition(servo1,(1/180)*(servo1_cal_angle+theta1));
writePosition(servo2,(1/180)*(servo2_cal_angle+theta2));

    pre_error_x=x_coordinates(end)-210;
pre_error_y=y_coordinates(end)-210;

    %if the ball get stuck in a position, 35 away from the center then we try to get the ball
%to the center.

if abs(x_coordinates(end-1)-x_coordinates(end))<=0.5 & distance >= 35 theta1=(17/210)*error_x;
theta2=(17/210)*error_y;
writePosition(servo1,(1/180)*(servo1_cal_angle+theta1));
writePosition(servo2,(1/180)*(servo2_cal_angle+theta2));

    end

    z=z+3;
t=size(x_coordinates(:,1));
figure(2);
hold on;
plot(z,error_x,'*');
ylim([-300 300]);
xlim([0 z+100]);
xlabel('time');
ylabel('error_x');
grid on;
hold off;

    figure(3);
hold on;
plot(z,error_y,'*');

    xlim([0 z+100]);
ylim([-300 300]);
xlabel('time');
ylabel('error_y');
grid on;
```

```
hold off;

    end
```