Ball and Beam Part-I

Group 7: Diya Mehta (22110078), Nishit Mistry (22110172), Vatsal Trivedi (22110276), Utkarsh Srivastava (22110278)

Department of Mechanical Engineering, IIT Gandhinagar

Under Professor Vineet Vashista for the course ES 245: Control Systems

1. Introduction

The ball and beam system consists of a ball that rolls along a beam, with its position controlled by adjusting the angle of the beam through a servo motor. The challenge lies in designing a control system that stabilizes the ball at a desired position on the beam while maintaining smooth and predictable motion.

1.1. Objectives

1. Model the system dynamics of the ball and beam system, including deriving the equations of motion and calculating the transfer function between servo gear angle and ball position.
2. Linearize the system equations and represent them in state-space form for control analysis and design.
3. Perform system analysis using MATLAB, including plotting poles and zeros, and simulating the open-loop step response.
4. Design and tune a PID controller to manipulate the ball's position on the beam, comparing the performance of Proportional (P), Proportional-Derivative (PD), and Proportional-Integral-Derivative (PID) controllers.
5. Optimize the PID controller to meet specified design criteria, including settling time less than 3 seconds and overshoot less than 5
6. Simulate the system in MATLAB/Simulink, generating both open-loop and closed-loop responses, and linearizing the model to design a compensator.
7. Develop a Simscape model of the ball and beam system to simulate its physical behavior in a more realistic environment.
8. Design a CAD model of the ball and beam setup, and build the physical system.
9. Implement the PID controller on the physical system and analyze its performance.

2. System Definition and Dynamics (Task-1)

2.1. System Overview

The ball and beam system consists of a ball that can roll along the length of a beam. The beam is attached to a lever arm at one end, which is connected to a servo motor. The servo motor controls the angle of the lever arm, which in turn adjusts the angle of the beam. As the beam tilts, the ball rolls along the beam under the influence of gravity. The goal is to control the position of the ball by adjusting the angle of the beam, using the servo motor as the input.

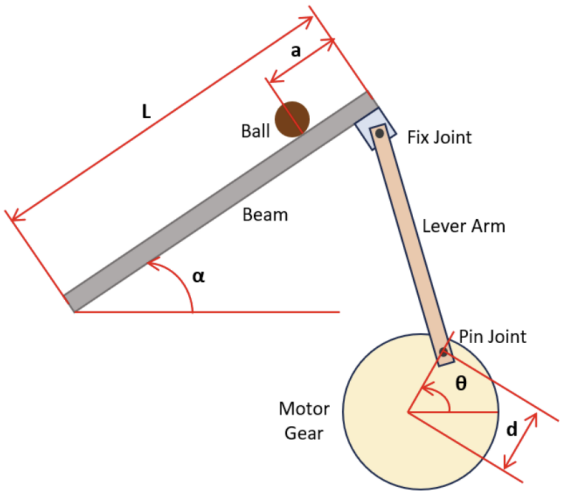

Figure 1. *Ball and Beam Model*

The system has one degree of freedom, which is the position of the ball along the length of the beam. The servo motor angle (θ) is the system input, and the ball position (a) along the beam is the output.

2.2. System Assumptions

1. The ball undergoes pure rolling (no slipping) on the beam.
2. The beam is rigid and lightweight, with negligible bending under the ball's weight.
3. The ball remains in continuous contact with the beam.

2.3. Equations of Motion

To describe the system dynamics, we need to derive the equations of motion for the ball as it rolls on the beam. The forces acting on the ball include gravity and the reaction forces from the beam.

Let:

- m be the mass of the ball.
- M be the mass of the beam.
- R be the radius of the ball.
- g be the acceleration due to gravity.
- θ be the angle of the servo motor.
- α be the angle of the beam relative to the horizontal.
- a be the position of the ball along the beam.
- L be the length of the beam.
- J be the moment of inertia of the ball about its center of mass.

- J_B = be the moment of inertia of the ball and the beam.

The Lagrangian L is given by the difference between the kinetic energy T and the potential energy V:

$$L = T - V$$

Using the expressions for T and V:

$$L = \left(\frac{1}{2}m\dot{a}^2 + \frac{1}{2}m\dot{\alpha}^2a^2 + \frac{1}{2}J\frac{\dot{a}^2}{R^2} + \frac{1}{2}J_B\dot{\alpha}^2\right) - \left(mga\sin(\alpha) + Mg\frac{L}{2}\sin(\alpha)\right)$$

The equation of motion is derived from the Euler-Lagrange equation:

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{a}}\right) - \frac{\partial L}{\partial a} = 0$$

Let's compute each term:

1. $\frac{\partial L}{\partial \dot{a}}$:

$$\frac{\partial L}{\partial \dot{a}} = m\dot{a} + \frac{J\dot{a}}{R^2}$$

2. $\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{a}}\right)$:

$$\frac{d}{dt}\left(m\dot{a} + \frac{J\dot{a}}{R^2}\right) = m\ddot{a} + \frac{J\ddot{a}}{R^2}$$

3. $\frac{\partial L}{\partial a}$:

$$\frac{\partial L}{\partial a} = -ma\dot{\alpha}^2 + mg\sin(\alpha)$$

Now, plugging these into the Euler-Lagrange equation:

$$m\ddot{a} + \frac{J\ddot{a}}{R^2} - ma\dot{\alpha}^2 + mg\sin(\alpha) = 0$$

2.4. System's transfer function between motor gear angle (θ) and ball position (a)

We substitute $\alpha = \frac{d\theta}{L}$ into the equation of motion:

$$m\ddot{a} + \frac{J\ddot{a}}{R^2} - \frac{mad^2\dot{\theta}^2}{L^2} + mg\sin(\frac{d\theta}{L}) = 0$$

Since we are linearizing around $\alpha = 0$, we ignore the higher-order terms and simplify the equation to:

$$\left(m + \frac{J}{R^2}\right)\ddot{a} = -mg\alpha$$

Substituting $\alpha = \frac{d\theta}{L}$, the equation becomes:

$$\left(m + \frac{J}{R^2}\right)\ddot{a} = -mg\frac{d\theta}{L}$$

Now, taking the Laplace transform of both sides, assuming the initial conditions are zero:

$$\left(m + \frac{J}{R^2}\right)s^2A(s) = -mg\frac{d}{L}\Theta(s)$$

Solving for $\frac{A(s)}{\Theta(s)}$:

$$\frac{A(s)}{\Theta(s)} = \frac{-mgd/L}{(m + \frac{J}{R^2})s^2}$$

2.5. State Space Representation

Let the state variables be $x_1 = a$ (ball's position) and $x_2 = \dot{a}$ (ball's velocity).

The equation of motion is:

$$\left(m + \frac{J}{R^2}\right)\ddot{a} = -mg\frac{d}{L}\theta$$

Substituting $x_2 = \dot{a}$ and $\dot{x_2} = \ddot{a}$:

$$\left(m + \frac{J}{R^2}\right)\dot{x_2} = -mg\frac{d}{L}\theta$$

Now, the state-space equations become:

$$\dot{x_1} = x_2$$

$$\dot{x_2} = -\frac{mgd}{L\left(m + \frac{J}{R^2}\right)}\theta$$

The output equation is:

$$y = x_1$$

Thus, the state-space representation can be written as:

$$\begin{bmatrix}\dot{x_1}\\ \dot{x_2}\end{bmatrix} = \begin{bmatrix}0 & 1\\ 0 & 0\end{bmatrix}\begin{bmatrix}x_1\\ x_2\end{bmatrix} + \begin{bmatrix}0\\ -\frac{mgd}{L\left(m+\frac{J}{R^2}\right)}\end{bmatrix}\theta$$

And the output equation is:

$$y = \begin{bmatrix}1 & 0\end{bmatrix}\begin{bmatrix}x_1\\ x_2\end{bmatrix}$$

3. System Analysis (Task-2)

3.1. Parameters

- Mass of the ball (m): 0.0016 kg
- Radius of the ball (R): 0.02 m
- Length of the beam (L): 0.235 m
- Lever arm offset (d): 0.045 m
- Moment of inertia of the ball (J): $\frac{2}{3}mR^2 = 4.267e-7$ kg-m^2
- Gravitational acceleration (g): -9.81 m/s^2

3.2. Poles and Zeros Analysis

The system's transfer function is a second-order system with a double pole at the origin, as evident from the denominator s^2. There are no zeros in this system, and the poles of the system are located at: $s = 0, s = 0$.

This indicates that the system is unstable in its open-loop configuration, as the presence of poles at the origin suggests that the system is neither stable nor asymptotically stable without feedback control. This is typical for an integrator-type system, where the output will continue to grow over time in response to an input.

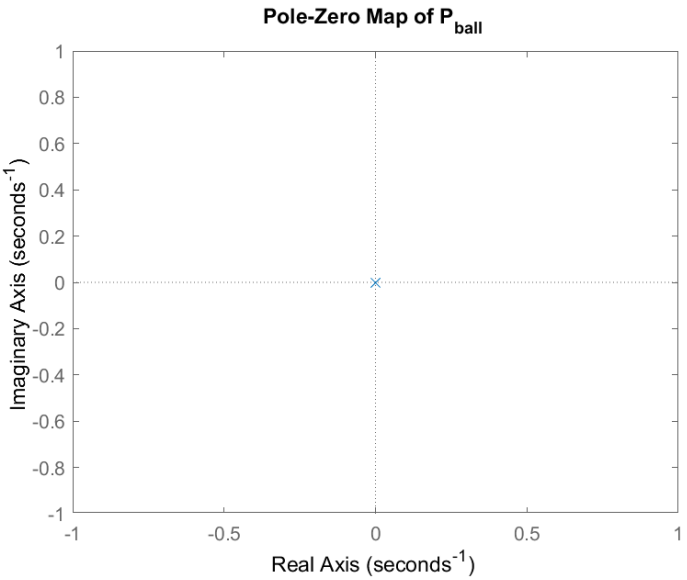

Figure 2. *Matlab Pole-Zero Plot*

3.3. Open-loop Step Response

The open-loop step response represents the system's reaction to a step input. Since the system's transfer function has poles at the origin, the open-loop response will exhibit unbounded growth, indicating instability.

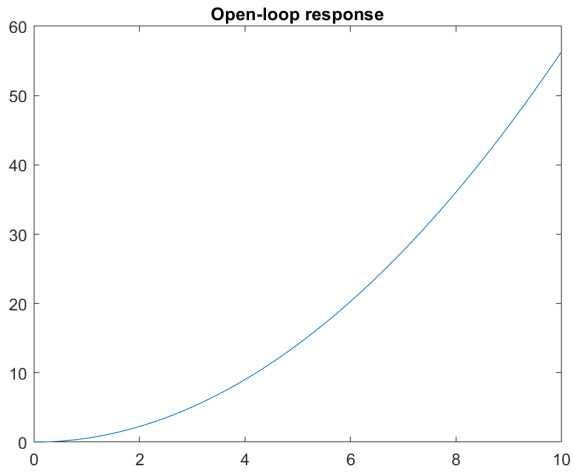

Figure 3. *Matlab Open-loop Step Response Plot*

The step response shows that when a 1-radian step is applied to the motor gear, the ball's position continues to increase without settling, confirming that the system is unstable. The ball's position grows without bounds as time progresses due to the lack of a stabilizing force or control mechanism.

3.4. System Type and Observations

Based on the transfer function and poles, the ball and beam system is classified as a type 2 system because it has two integrators (poles at the origin). Type 2 systems have infinite steady-state error for step-inputs in open-loop form. Additionally, they are prone to instability, as seen from the step response.

Observations from Poles/Zeros Plot:

- The poles are located at the origin, confirming the presence of two integrators in the system.
- The system is inherently unstable without feedback, as the ball's position increases indefinitely in response to any disturbance or input.

Observations from Open-Loop Step Response:

- The system exhibits unbounded growth in the ball's position due to the lack of stability in the open-loop form.
- The need for a feedback control mechanism, such as a PID controller, is critical to stabilizing the system and achieving the desired control performance.

The open-loop analysis demonstrates that the ball and beam system is unstable and cannot control the ball's position effectively without feedback. The transfer function indicates the need for a controller to stabilize the ball at a desired position on the beam.

4. PID Control Design (Task-3)

4.1. Controller Structure

A PID controller computes the control input $u(t)$ to the system based on the error $e(t)$, which is the difference between the desired ball position $a_{\text{desired}}(t)$ and the actual ball position $a(t)$. The control input is defined as:

$$u(t) = K_p e(t) + K_i \int e(t)\,dt + K_d \frac{de(t)}{dt}$$

Where:

- K_p is the proportional gain,
- K_i is the integral gain,
- K_d is the derivative gain,
- $e(t) = a_{\text{desired}}(t) - a(t)$ is the error signal.

4.2. Proportional Controller (P-Controller)

A proportional controller adjusts the control signal in proportion to the error. The control law is simplified to:

$$u(t) = K_p e(t)$$

The step response for different values of K_p was simulated in MATLAB to observe the system's behavior.

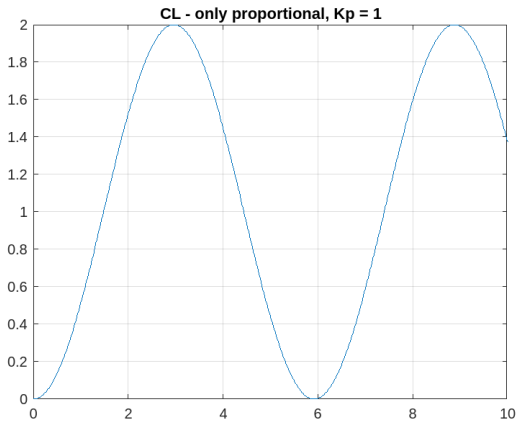

Figure 4. *P-Controller with K_p= 1*

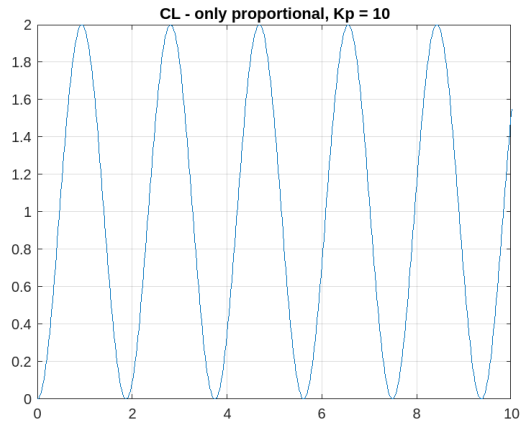

Figure 5. *P-Controller with K_p= 10*

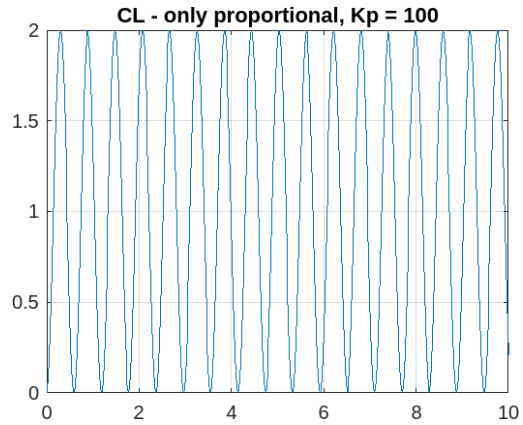

Figure 6. *P-Controller with K_p= 100*

4.2.1. Observations.

- Transient Response: As K_p increases, the rise time significantly decreases, improving the initial speed of response.
- Steady-State Response: While the system keeps on oscillating for K_p values, but a higher K_p causes more oscillation, making the steady-state response less stable at extreme values.
- Set Point Error: A moderate increase in K_p reduces the set point error by making the system respond more quickly. However, a high K_p introduces large oscillations, leading to significant errors in reaching the desired set point.

4.3. Proportional-Derivative Controller (PD-Controller)

Adding a derivative term improves the system's stability by providing a damping effect that counteracts rapid changes in the error. The control law is:

$$u(t) = K_p e(t) + K_d \frac{de(t)}{dt}$$

The PD controller reduces overshoot and improves transient response, but steady-state error may still persist. The step response for different values of K_d and K_p was simulated to assess the impact of derivative action.

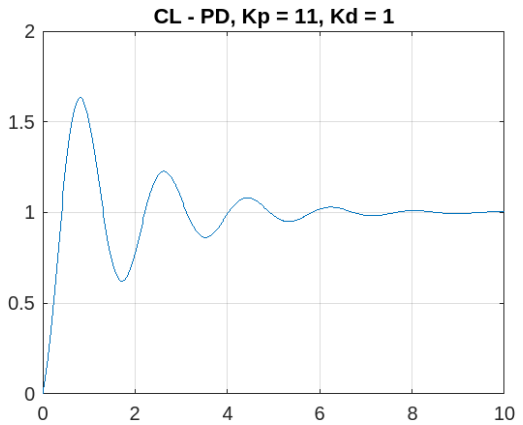

Figure 7. *PD-Controller with K_d= 11*

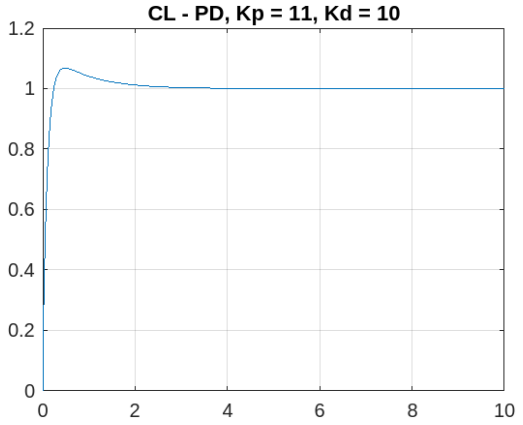

Figure 8. *PD-Controller with K_d= 10*

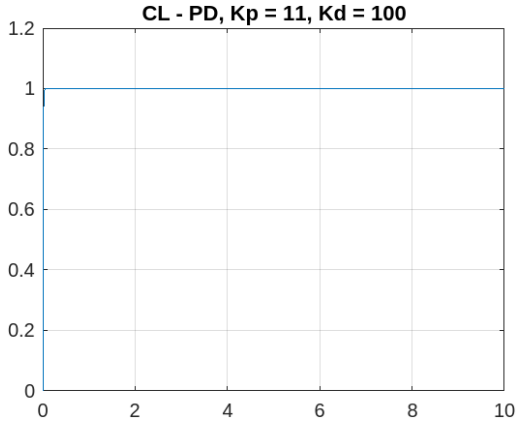

Figure 9. *PD-Controller with K_d= 100*

4.3.1. Observations.

- Transient Response: As K_d increases, the derivative action helps in reducing the oscillations allowing the system to react faster to the changes.
- Steady-State Response: The system stabilizes much quicker with higher K_d, providing a more stable and quicker response to reach steady state.
- Set Point Error: Increasing K_d has no impact on the set point error directly, but the minimized overshoot results in a more precise approach to the set point.

4.4. Proportional-Integral-Derivative Controller (PID-Controller)

A PID controller combines the benefits of proportional, integral, and derivative actions. The integral term eliminates steady-state error by accumulating the error over time, while the derivative term improves transient response. The control law is:

$$u(t) = K_p e(t) + K_i \int e(t)\,dt + K_d \frac{de(t)}{dt}$$

The step response for different values of K_d,K_p and K_i was simulated to assess the impact of integral action.

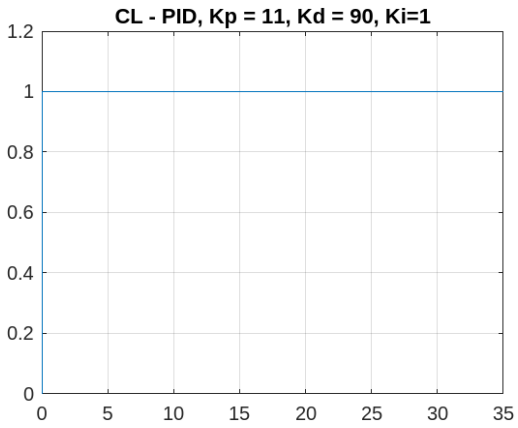

Figure 10. *PID-Controller with K_i= 1*

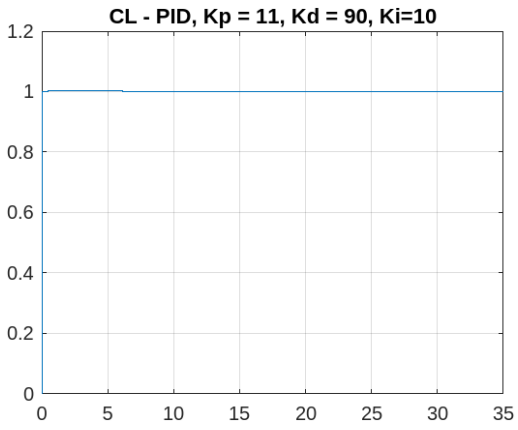

Figure 11. *PID-Controller with K_i= 10*

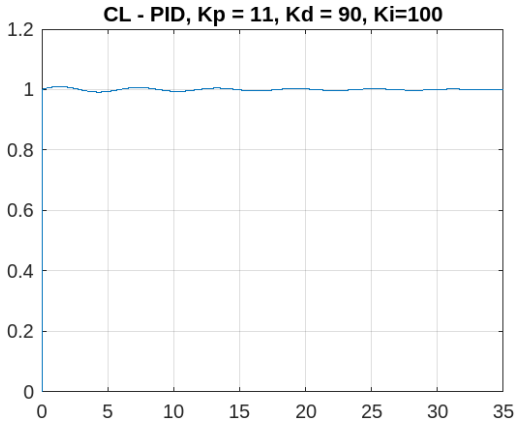

Figure 12. *PID-Controller with K_i= 100*

4.4.1. Observations.

- The transient response remains almost same, with only slight increases in overshoot as the integral gain increases.
- The primary role of the integral term is to eliminate steady-state error. As K_i increases, the system becomes more accurate in tracking the set point. However, an overly high K_i introduces an overshoot to the system.

4.5. Controller Design for Specific Criteria

The goal is to design a PID controller that satisfies the following criteria:

- Settling time $<$ 3 seconds,
- Overshoot $<$ 5%.

Through iterative tuning of the PID controller gains in MATLAB, the following gains were found to meet the desired specifications:

Using: $k_d = 11; k_p = 90; k_i = 1;$

Proportional-Integral-Derivative (PID) Control

Settling Time:	2.9656 s
Overshoot:	0.1335 %

4.5.1. Performance Results.

- Settling time: 2.96 seconds,
- Overshoot: 0.13%.

The PID controller successfully meets the design requirements, providing a fast and stable response with minimal overshoot and no steady-state error.

5. Simulink and Simscape Simulation (Task-4)

5.1. Open-loop Simulink Model

The first step was to develop the open-loop system model in Simulink, where no control feedback was applied. The purpose of this simulation was to observe the system's natural response when the servo gear angle is actuated without any corrective input.

1. System Dynamics Blocks: The system's dynamics were modeled using Simulink blocks. Key components such as the ball's motion on the beam, beam angle control via the servo motor, and gravity were included.
2. Input Signal: A step input of 1 radian was applied to the servo motor, representing a sudden change in the motor's angle.
3. Simulation Configuration: The simulation was configured for a reasonable time duration to capture the transient and steady-state response.

We add the function that takes the vector $[r, \dot{r}, \alpha, \dot{\alpha}]$ and returns $\ddot{r}$ for the ball and beam model.

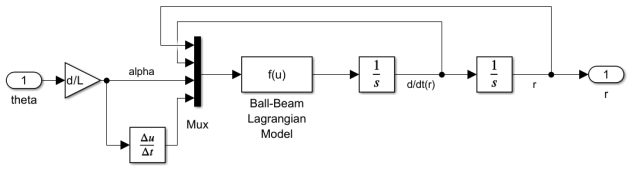

Figure 13. *Ball and Beam Model*

For non-linearzed, we use the Ball and Beam Model, for linearized system, we use the transfer function.

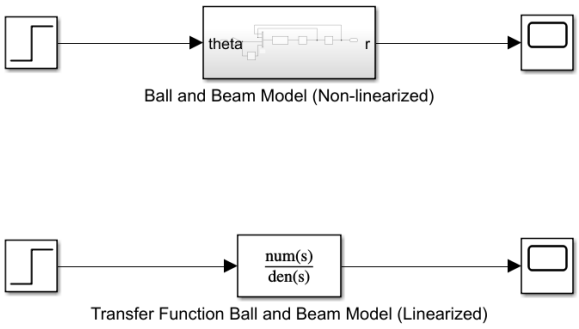

Figure 14. *Open-Loop System (Simulink)*

In both, the step input open-loop resopnse for T=10 sec is:

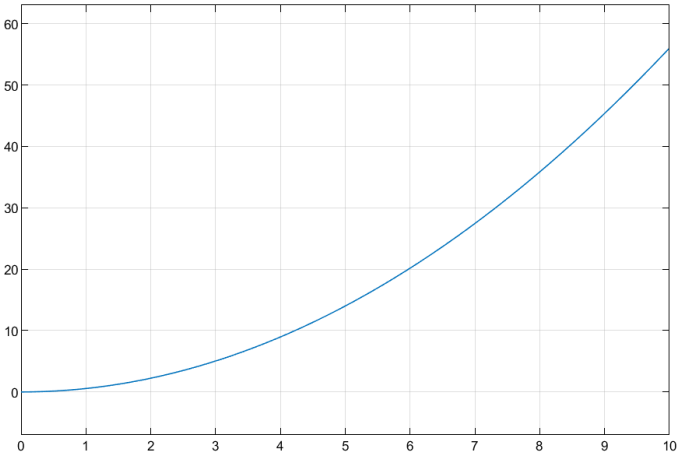

Figure 15. *Open-loop response for step input (Simulink)*

Open-Loop Response Observations:

1. The ball's position deviates significantly from the setpoint, confirming the system's instability.

5.2. Closed-Loop Simulink Model with Compensator

5.2.1. Compensator Design. For the given ball and beam system, our goal is to design a compensator of the form:

$$C(s) = \frac{s+a}{s+b}$$

where a and b are parameters to be tuned. The compensator modifies the behavior of the system by adding poles and zeros, affecting both the transient and steady-state performance. The primary objective is to achieve a settling time of less than 5 seconds and an overshoot of less than 5%.

5.2.2. Root Locus Design Approach. The Root Locus method involves plotting the location of the system poles as the gain is varied. The compensator adds a zero at $s = -a$ and a pole at $s = -b$, which influences the shape of the root locus plot. The key steps in this approach are:

1. **Initial Plant Transfer Function:** We first define the plant's transfer function, which is a representation of the dynamics of the ball on the beam.

$$P(s) = \frac{-(mgd/L)}{(J/R^2 + m)\,s^2}$$

2. **Open-Loop Transfer Function:** The compensator is then cascaded with the plant, and the open-loop transfer function becomes:

$$G_{OL}(s) = \frac{s+a}{s+b} \cdot P(s)$$

3. **Root Locus Plot:** By generating the root locus, we can visually assess how varying the system gain moves the poles of the system. Adjusting a and b iteratively shifts the location of poles:

   - Increasing a tends to shift poles leftward, reducing the settling time.
   - Adjusting b affects the damping ratio, helping to reduce overshoot.

   Through multiple iterations of adjusting a and b and analyzing the root locus, we tuned the system to place the poles in locations that correspond to the desired performance.

4. **Closed-Loop Response:** After determining suitable values for a and b, we close the loop with unity feedback and observe the step response.

We have added a lead compensator since a=0.01 and b=5. A lead compensator is a control system component that adds a zero and a pole to the open-loop transfer function, typically positioned in such a way that the zero is closer to the origin than the pole. This configuration increases the phase margin of the system, thereby improving stability and transient response.

5.2.3. Final Compensator Design. After completing the iterative process using the root locus methods, we arrived at final values for $a = 0.01$ and $b = 5$ that optimized the system's performance. The compensator was able to modify the system's dynamics in a way that resulted in:

- A settling time of less than 5 seconds
- An overshoot of less than 5%

5.2.4. Simulink Modeling. Simulink Diagram: The closed-loop model was built in Simulink using the following components:

1. Gain block,
2. Compensator block,
3. Transfer function block representing the Ball and Beam system dynamics,
4. Sum block for calculating the error between the desired and actual positions,
5. Step input signal for the servo motor.

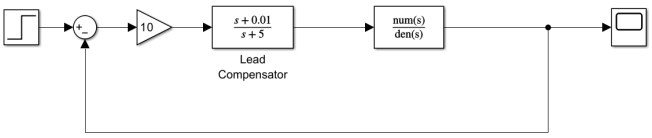

Figure 16. *Closed-loop System (Simulink)*

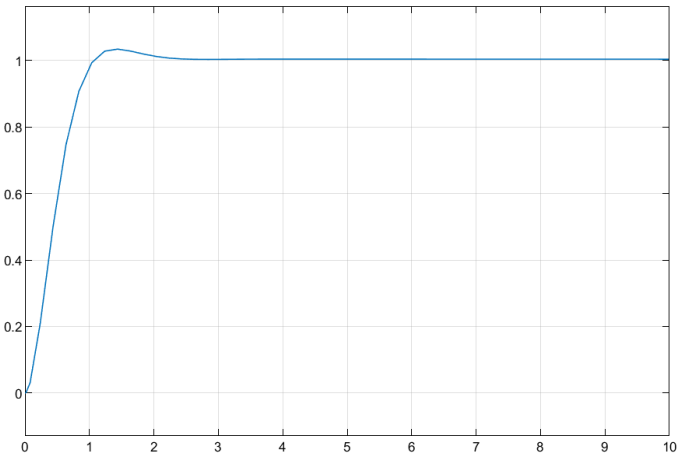

Figure 17. *Closed-loop response for step input (Simulink)*

Closed-Loop Response Observations:

1. Simulating the compensator system in MATLAB, we find the settling time to be 1.83 seconds and the overshoot to be 3.45 %.
2. The compensator successfully stabilizes the ball's position with minimal overshoot and a fast settling time.
3. The system reaches the desired position smoothly without sustained oscillations or instability.

5.3. Simscape Model

1. Physical Components: The ball, beam, servo motor, and gravity were modeled using Simscape components.
2. Simulation: The physical model was simulated to observe the system's real-world dynamics and controller performance.

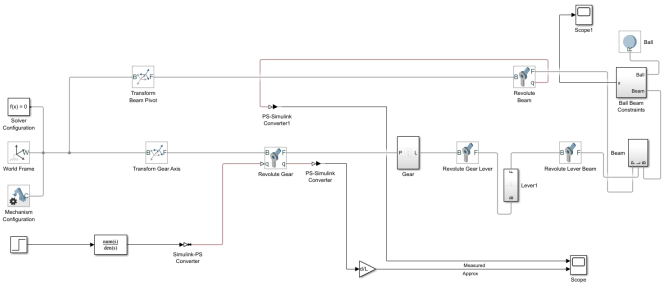

Figure 18. *Ball and Beam Simscape System*

Figure 19. *Ball and Beam Simscape Model*

Observations from Simscape Simulation:

1. The Simscape model provides a more realistic simulation, including the effects of friction, damping, and real-world constraints.

6. Physical System Design (Task-5)

6.1. Components of the Physical System

The primary components used to construct the physical system are as follows:

Ball: A small spherical object that rolls along the beam.

Beam: A rigid, flat beam upon which the ball rolls. The length of the beam was selected to provide enough range for the ball to move freely while remaining compact for control.

Lever Arm: A connecting rod that transmits the angular movement from the servo motor to the beam.

Servo Motor: The actuator responsible for adjusting the beam's angle. The motor is controlled by an external controller to set the desired angle and adjust the beam's tilt.

Ultrasonic Sensor: To measure the ball's position along the beam. Power Supply: The system requires a stable power source to operate the servo motor and sensors.

6.2. CAD Model Design

The physical design of the system was modeled in Autodesk Inventor to ensure proper dimensions, alignment of components, and overall system integration.

The beam was modeled as a rectangular structure, with proper attachment points for the lever arm and sensors.The servo

motor was positioned at one end of the beam, with a lever arm attached to transfer the rotational motion of the servo motor into linear tilting of the beam. A platform was designed to hold the beam in place, ensuring it was free to tilt along its axis while allowing the ball to roll.

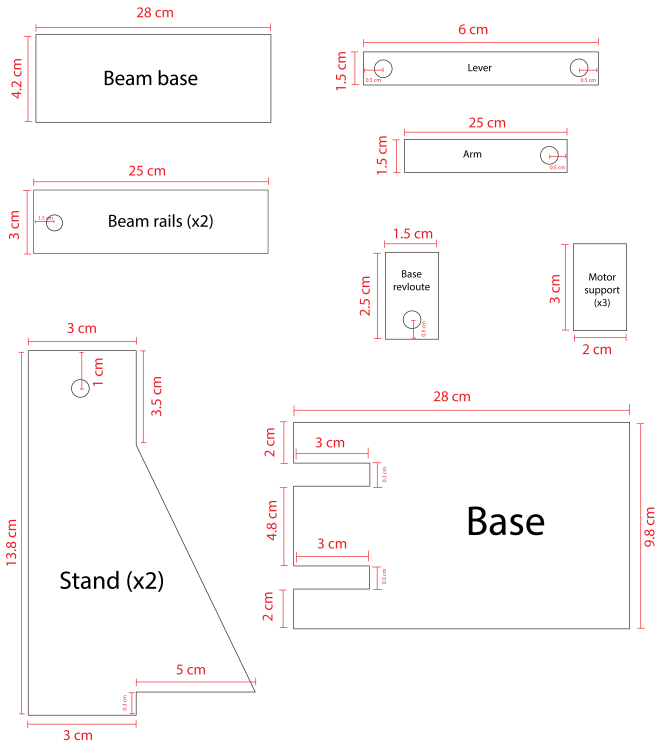

Figure 20. *Components of CAD Design Model*

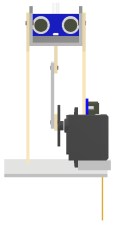

Figure 21. *Front View of CAD Design Model*

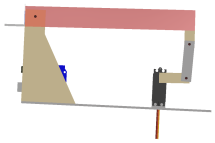

Figure 22. *Side View of CAD Design Model*

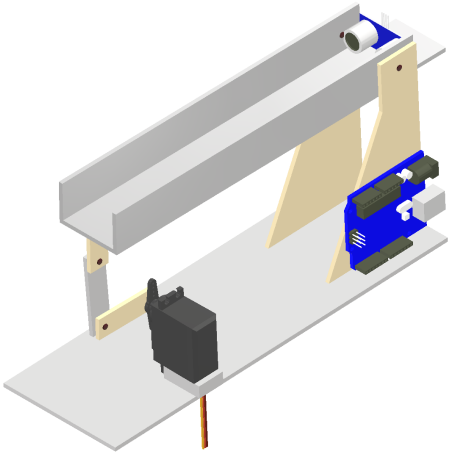

Figure 23. *Isometric View of CAD Design Model*

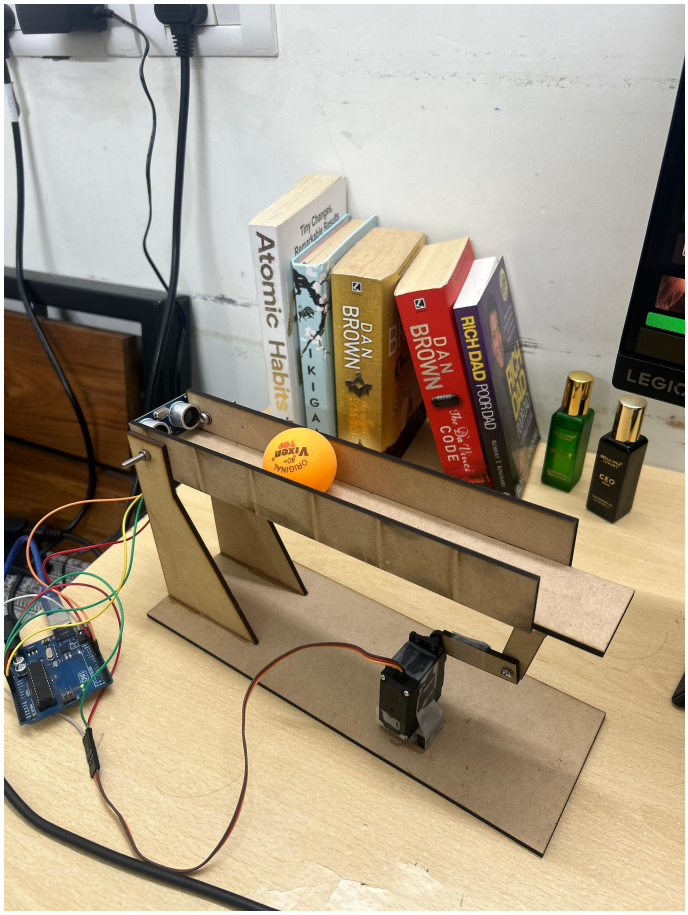

Figure 24. *Physical Setup*

6.3. Physical System Assembly

Once the design was finalized in CAD, the system was physically constructed. The following steps were followed to assemble the ball and beam system:

1. Beam Setup: The beam was mounted on a stable base with a pivot point allowing it to tilt. The ball was placed on the beam, free to roll along its length.
2. Servo Motor Installation: The servo motor was mounted at the end of the beam and connected to the lever arm. The lever arm was secured to the beam at a fixed offset distance to allow smooth motion.
3. Sensor Placement: The sensor was plased as to be able to measure the ball position. It was connected to Arduino UNO.
4. Arduino UNO and Power Supply: The Arduino UNO was programmed with the PID control algorithm and connected to the servo motor and sensor. A power supply was provided to operate the motor and sensor.
5. Wiring and Connections: Proper wiring was done to connect the components, and all connections were securely

fastened.

6.4. PID Controller Implementation

A PID controller was implemented to control the servo motor's angle based on the ball's position. The controller's parameters were tuned based on the simulation results as the starting point to achieve the desired control performance in the physical system. Based on the system reponse, the gains were adjusted. By manually tuning the physical system, we set:

$$K_p = 3, K_i = 0, K_d = 0.5$$

. Challenges in Implementation:

1. Noise and Sensor Accuracy: Some challenges were faced with sensor accuracy and noise, which impacted the control signal. This was mitigated by careful calibration.

2. Friction and Slipping: While pure rolling was assumed, minor slipping due to friction was observed during testing. This required slight adjustments to the controller gains to accommodate real-world behavior.

3. Time Delay: Small time delays were noticed in the physical system response, requiring the adjustment of the derivative term in the PID controller to prevent overshoot and instability.

6.5. Observations and Results

The physical system closely mirrored the simulated model's behavior. The PID controller was effective in stabilizing the ball at a set position on the beam.

- Steady-State Performance: The ball reached the desired position and maintained it without drifting.

- Transient Performance: The system's response was fast, with minimal oscillations, and it met the transient response criteria established in the design.

7. Results and Discussion

Overall, the simulated model provided a good foundation for understanding the system dynamics and establishing initial PID gains. However, the transition to the physical system required careful consideration of the non-linearities and disturbances that were not fully captured in the simulation.

The effectiveness of the PID control strategy was demonstrated through successful stabilization of the ball at desired positions on the beam. The results underscored the importance of tuning and adapting the controller to real-world conditions.

8. Conclusion

The ball and beam pmodelling successfully demonstrated the design and implementation of a PID control system to manipulate the position of a ball on a beam using a servo motor. Through comprehensive analysis and simulation, we established the system dynamics and developed effective controller strategies. The PID controller, particularly the proportional-integral-derivative configuration, proved effective in achieving minimal steady-state error and satisfactory transient response.

The transition from simulation to physical implementation highlighted the complexities of real-world dynamics, necessitating iterative tuning to adapt the controller to actual conditions. Overall, this project reinforced the importance of applying theoretical control principles in practical scenarios and provided valuable insights for future work in control systems.

9. Acknowledgments

We would like to express our sincere gratitude to our professor and the teaching assistants for their guidance and support throughout this project. Their expertise and feedback were invaluable in helping us complete this work successfully.