Project -Ball and Beam
ES 245:Control System

Group 10

Nikhil Kumar Lal
Mechanical Engineering
22110167
nikhilkumar.lal@iitgn.ac.in

Manas Kalal
Mechanical Engineering
22110138
manas.kalal@iitgn.ac.in

Nandkishor Kumar Pandit
Mechanical Engineering
22110164
nandkishorkumar.pandit@iitgn.ac.in

Kirtan Patel
Mechanical Engineering
22110185
kirtankumar.patel@iitgn.ac.in

I. Problem Statement

The objective of this project is to design a control system for a Ball and Beam setup. The aim is to control the position of a ball on a beam by adjusting the beam angle, which is in turn controlled by a servo motor. The control system needs to ensure stability, fast response, and minimal steady-state error. The project involves defining system dynamics, analyzing the system's behavior, and implementing controllers like Proportional-Integral-Derivative (PID) to achieve specified performance criteria.

II. Assumptions

- The ball rolls without slipping on the beam.
- The beam is assumed to be rigid and frictionless.
- The servo motor operates linearly within the control range.
- Small-angle approximation for beam angle α is valid: $\sin(\alpha) \approx \alpha$.
- The mass distribution of the ball is uniform.

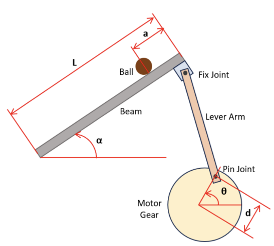

Fig. 1: Ball and Beam System

Task 1: Define the System

1.1 System Dynamics Equations

To start, we need to establish the dynamics of the system. Here's a basic overview:

The ball rolls on the beam with one degree of freedom, i.e., the ball can move along the length of the beam, controlled by the beam's angle. The beam is rotated by a servo motor, which changes the angle θ.

Assumptions::

- The ball experiences pure rolling (no slipping).
- The system is subject to gravity, and we assume small angle approximations (for linearization later).

Dynamics of the Ball:: Using Newton's second law, the dynamics of the ball are derived from the forces and torques acting on it.

a) Translational motion:: The translational motion of the ball's center of mass is:

$$m\ddot{a} = mg\sin(\alpha)$$

Where:

- m is the mass of the ball,
- a is the position of the ball on the beam,
- g is acceleration due to gravity,
- α is the angle of the beam (which is related to the motor angle θ).

For small angles, $\sin(\alpha) \approx \alpha$, so:

$$m\ddot{a} = mg\alpha$$

b) Rotational motion:: The rotational motion (pure rolling) is described by:

$$J\ddot{\theta} = RF$$

Where:

- J is the moment of inertia of the ball,
- R is the radius of the ball,
- F is the force causing the rolling motion.

Since the ball performs pure rolling, $\ddot{a} = R\ddot{\theta}$, and the total force on the ball is:

$$m\ddot{a} + \frac{J}{R^2}\ddot{a} = mg\alpha$$

Thus, the combined equation of motion becomes:

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

1.2 Transfer Function

Now, we relate the motor angle θ to the beam angle α, which controls the ball's position a.

Assuming that $\alpha = d \cdot \theta$ (where d is the lever arm length), the equation of motion becomes:

$$\left(m + \frac{J}{R^2}\right) \ddot{a} = mgd\theta$$

Taking the Laplace transform of both sides (assuming zero initial conditions), we obtain the transfer function:

$$\left(m + \frac{J}{R^2}\right) s^2 A(s) = mgd\Theta(s)$$

Thus, the transfer function $P(s) = \frac{R(s)}{\Theta(s)}$ is:

$$P(s) = \frac{A(s)}{\Theta(s)} = \frac{mgd}{s^2 \left(m + \frac{J}{R^2}\right)}$$

This can be simplified to:

$$P(s) = -\frac{mgd}{L\left(\frac{J}{R^2} + m\right)} \frac{1}{s^2}$$

Where L represents the beam length.

1.3 Linearized State-Space Form

To linearize the system, we consider the small angle approximation ($\sin(\alpha) \approx \alpha$) and express the system in state-space form.

Let:

$$\begin{aligned} x_1 &= a \quad \text{(ball position)}, \\ x_2 &= \dot{a} \quad \text{(ball velocity)}. \end{aligned}$$

The state-space equations become:

$$\dot{x}_1 = x_2$$

$$\dot{x}_2 = \frac{mgd}{m + \frac{J}{R^2}} \theta$$

The state-space form is then:

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\theta$$

Where:

$$\mathbf{A} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} 0 \\ \frac{mgd}{m + \frac{J}{R^2}} \end{bmatrix}$$

The output equation is:

$$y = \mathbf{C}\mathbf{x} + \mathbf{D}\theta$$

Where:

$$\mathbf{C} = \begin{bmatrix} 1 & 0 \end{bmatrix}, \quad \mathbf{D} = 0$$

Thus, the linearized state-space form is:

$$\dot{x}_1 = x_2$$

$$\dot{x}_2 = \frac{mgd}{m + \frac{J}{R^2}} \theta$$

Task 2: Analysis of the System

Assumed Parameter Values

Before performing the analysis, we assume the following typical values for the Ball and Beam system parameters:

- Mass of the ball: $m = 0.01\,\text{kg}$
- Radius of the ball: $R = 0.015\,\text{m}$
- Lever arm offset: $d = 0.03\,\text{m}$
- Length of the beam: $L = 0.34\,\text{m}$
- Gravitational acceleration: $g = 9.81\,\text{m/s}^2$
- Ball's position on the beam: $a = 0\,\text{m}$ (initial)
- Beam angle: $\alpha = 5^\circ = 0.0873\,\text{rad}$
- Servo gear angle: $\theta = 0.05\,\text{rad}$

System Dynamics and Transfer Function

The system's transfer function is given by:

$$P(s) = -\frac{mgd}{L\left(\frac{J}{R^2} + m\right)} \frac{1}{s^2}$$

Given the moment of inertia for the ball:

$$J = \frac{2}{5} m R^2$$

Substituting this into the transfer function:

$$P(s) = -\frac{mgd}{L\left(\frac{2}{5}m + m\right)} \frac{1}{s^2}$$

Simplifying:

$$P(s) = -\frac{mgd}{L\left(\frac{7}{5}m\right)} \frac{1}{s^2} = -\frac{5gd}{7L} \frac{1}{s^2}$$

Substituting the numerical values:

$$P(s) = -\frac{5(9.81)(0.03)}{7(0.34)} \frac{1}{s^2} \approx -0.6195 \frac{1}{s^2}$$

2.1 MATLAB Analysis

Poles and Zeros Plot

To perform the poles/zeros analysis in MATLAB, we can represent the transfer function using the assumed parameters:

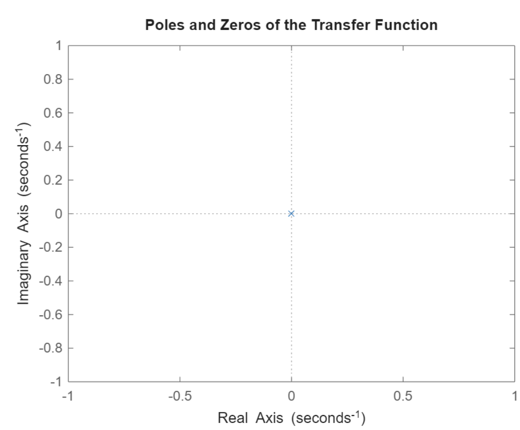

Fig. 2: Pole-Zero Map of the Ball and Beam System

Poles: The system has two poles at the origin (i.e., $s = 0$), indicating an integrator or double integrator system.
Zeros: There are no zeros for this system because the numerator is constant.

Open Loop Step Response

The open-loop step response is obtained for a 1-radian step input in the servo gear angle:

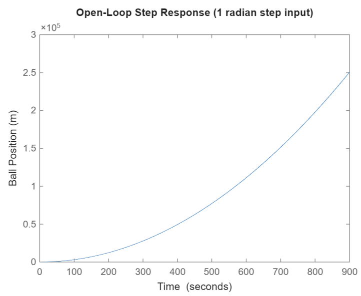

Fig. 3: Open Loop Step Response of the Ball and Beam System

This plot will show the ball's position response to a 1-radian step input in the servo gear angle.

2.2 System Type and Observations

System Type

From the poles/zeros plot, we observe that the system is type 2. This is because it has two poles at the origin, meaning it has two integrators in the open-loop transfer function. Such systems typically have an unbounded response to a step input, as seen in the open-loop response.

Observations from Poles/Zeros Plot

The two poles at the origin indicate that the system is marginally stable. Without feedback control, the system will not return to its equilibrium position and will continue to accelerate, making it inherently unstable in open-loop.

Observations from the Open-Loop Step Response

The step response shows that the ball's position increases without bounds over time, indicating instability. The ball will move away from the initial position, and the system is incapable of self-correction without a controller.

This instability is typical for a type 2 system, as the integrators cause the system to accumulate error over time, making control necessary to stabilize the ball.

Task 3: PID Control

3.1 Design of Controllers

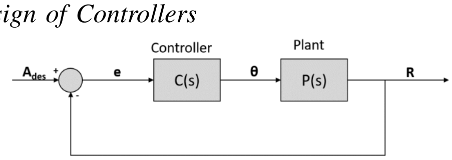

Fig. 4

Proportional Controller (P) A proportional controller applies a control signal proportional to the error signal, defined by $U(s) = K_p E(s)$, where K_p is the proportional gain and $E(s)$ is the error. As the value of K_p increases, the system's responsiveness improves, but excessive K_p can lead to overshoot and instability.

$$u(t) = K_p \cdot e(t) \tag{1}$$

Where:

- K_p is the proportional gain.

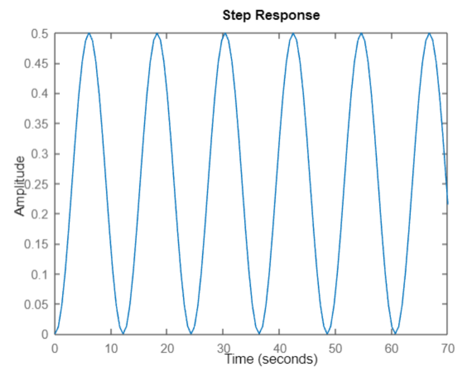

Fig. 5: Performance of Proportional controller(kp=1).

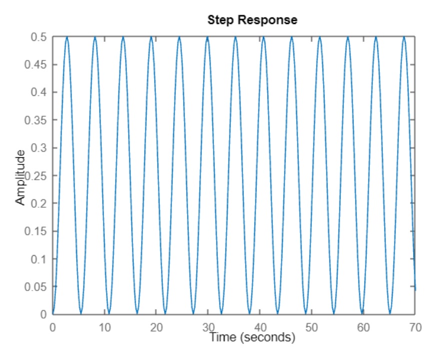

Fig. 6: Performance of Proportional controller(kP= 5).

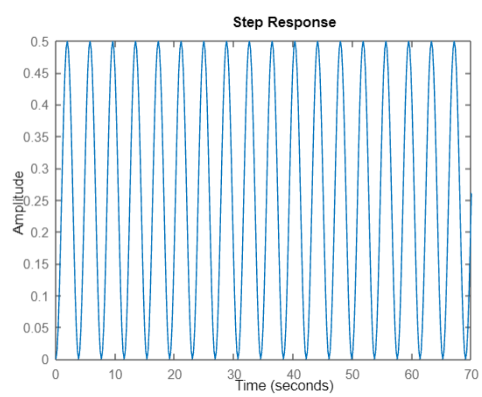

Fig. 7: Performance of Proportional controller(kP= 10).

As observed, the system remains marginally stable when a proportional gain is applied. However, as the value of K_p

increases from 1 to 5 and then to 10, the system becomes unstable, indicating that higher gains negatively impact system stability.

Proportional-Derivative Controller (PD) For the proportional-derivative controller, the control input also depends on the derivative of the error:

$$u(t) = K_p \cdot e(t) + K_d \cdot \dot{e}(t) \tag{2}$$

Where:

- K_d is the derivative gain.

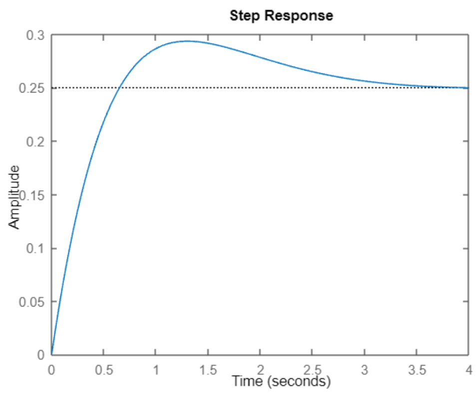

Fig. 8: Performance of Proportional-derivative controller(kp=10,kd=10).

When $K_p = 10$ and $K_d = 10$, the system is stable, but the overshoot is too high, and the settling time needs to be reduced. We see that by increasing K_d, we can reduce the overshoot and slightly decrease the settling time. Therefore, increase K_d to 20 . The expected output should reflect these improvements.

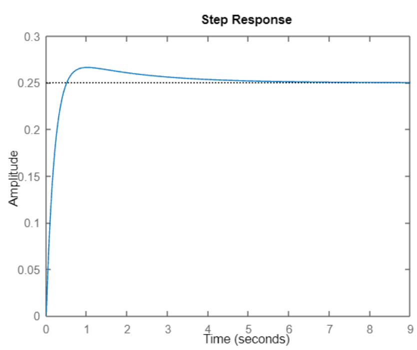

Fig. 9: Performance of Proportional-derivative controller(kp=10,kd=20).

The overshoot criterion is met but the settling time needs to come down a bit. To decrease the settling time we may try increasing the K_p slightly to increase the rise time. The derivative gain (K_d) can also be increased to take off some of the overshoot that increasing K_p will cause. After playing with the gains a bit, the following step response plot can be achieved with K_p = 15 and K_d = 40:

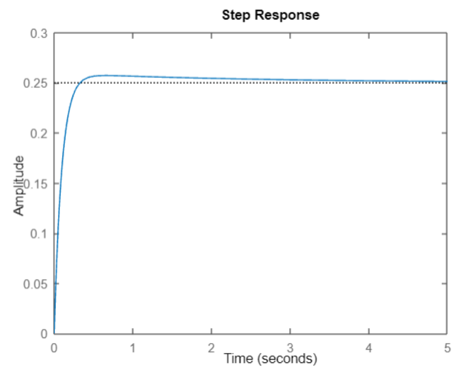

Fig. 10: Performance of Proportional-derivative controller(kp=15,kd=40).

Proportional-Integral-Derivative Controller (PID) For the full PID controller, the control input depends on the proportional, derivative, and integral of the error:

$$u(t) = K_p \cdot e(t) + K_i \cdot \int e(t)\,dt + K_d \cdot \dot{e}(t) \tag{3}$$

Where:

- K_i is the integral gain.

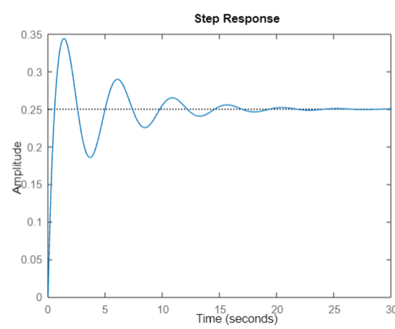

Fig. 11: Performance of Proportional-integral-derivative controller(Pkp=10,kd=10,ki=15).

The PID controller with parameters $P = 10$, $I = 15$, and $D = 10$ on the ball and beam system shows multiple overshoots. This happens because the integral action accumulates errors, which affects the proportional response. The derivative action helps dampen the system's response, reducing overshoot and eventually minimizing the steady-state error. However, achieving this stability takes much longer than with just a PD controller, so it may be better to use only the proportional and derivative components for this system.

To achieve a smoother response, we can increase the derivative action to control the aggressive behavior of the proportional action. We can also increase the integral action to further reduce the steady-state error. The smoother PID controller still doesn't perform better than the PD controller because it takes longer to reach the setpoint.

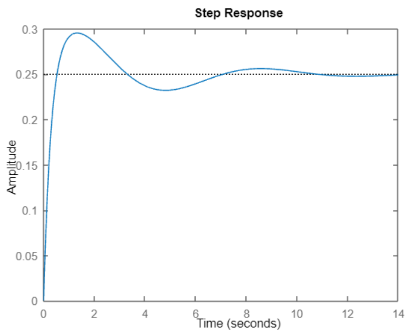

Fig. 12: Performance of Proportional-integral-derivative controller(kp=10,kd=15,ki=10).

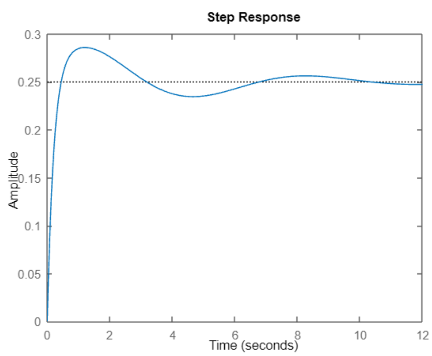

Fig. 13: Performance of Proportional-integral-derivative controller(Pkp=12,kd=20,ki=15).

Observations on Controller Types

Proportional Controller Observations

1) $K_p = 1$:
    - **Transient Response**: Slow rise time and settling time, with noticeable overshoot.
    - **Steady-State Response**: Significant steady-state error; the system does not stabilize at the set point.
    - **Set Point Error**: Large steady-state error due to insufficient gain.
2) $K_p = 5$:
    - **Transient Response**: Improved rise time and reduced settling time, but still exhibits overshoot.
    - **Steady-State Response**: Steady-state error decreases but is still present.
    - **Set Point Error**: Moderate steady-state error; performance improves over $K_p = 1$.
3) $K_p = 10$:
    - **Transient Response**: Fast rise time and settling time with increased overshoot.
    - **Steady-State Response**: Much lower steady-state error, but overshoot can lead to instability.
    - **Set Point Error**: Minimal steady-state error; closer to desired set point.

Proportional-Derivative (PD) Controller Observations

1) $K_p = 10$, $K_d = 10$:
    - **Transient Response**: Improved rise time and reduced overshoot compared to the proportional controller.
    - **Steady-State Response**: Some steady-state error remains.
    - **Set Point Error**: Steady-state error is reduced compared to the P controller.
2) $K_p = 10$, $K_d = 20$:
    - **Transient Response**: Further reduction in overshoot; faster response.
    - **Steady-State Response**: Steady-state error is still present but minimized.
    - **Set Point Error**: The system responds well with lower steady-state error than the previous configuration.
3) $K_p = 15$, $K_d = 40$:
    - **Transient Response**: Very fast rise time with minimal overshoot, indicating good damping.
    - **Steady-State Response**: Very low steady-state error.
    - **Set Point Error**: Excellent performance with negligible steady-state error.

Proportional-Integral-Derivative (PID) Controller Observations

1) $K_p = 10$, $K_d = 10$, $K_i = 15$:
    - **Transient Response**: Similar multiple overshoots, but slightly better performance.
    - **Steady-State Response**: Further reduction in steady-state error.
    - **Set Point Error**: Improved performance, but still oscillatory.
2) $K_p = 10$, $K_d = 15$, $K_i = 10$:
    - **Transient Response**: Improved damping with less overshoot.
    - **Steady-State Response**: Steady-state error decreases effectively.
    - **Set Point Error**: Excellent overall performance with reduced oscillations.
3) $K_p = 12$, $K_d = 20$, $K_i = 15$:
    - **Transient Response**: Fast rise time with smooth transitions and minimal overshoot.
    - **Steady-State Response**: Low steady-state error achieved quickly.
    - **Set Point Error**: Very small steady-state error; system reaches the set point efficiently.

Conclusion

In summary, as the gains increase in each controller type, the transient response improves with faster rise times and reduced settling times. However, higher proportional gains may lead to overshoot and instability. The addition of derivative action helps dampen the response, while integral action eliminates steady-state error but can introduce overshoot and longer settling times. The PD controller generally performs better in terms of speed and stability compared to the PID controller for the ball and beam system, which benefits from tuning to achieve optimal performance.

Task 3.3: PID Controller Design

In this task, we aim to design a PID controller that meets the performance criteria of a settling time of less than 3 seconds and an overshoot of less than 5%. The design process involves calculating the necessary parameters, simulating the system's response, and tuning the controller gains to achieve the desired transient and steady-state performance.

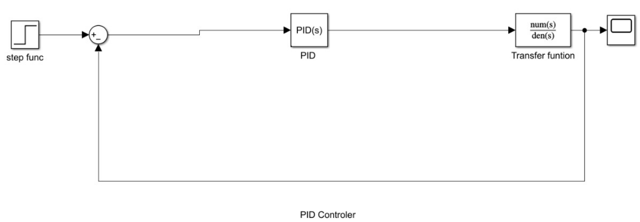

Fig. 14: Block Diagram.

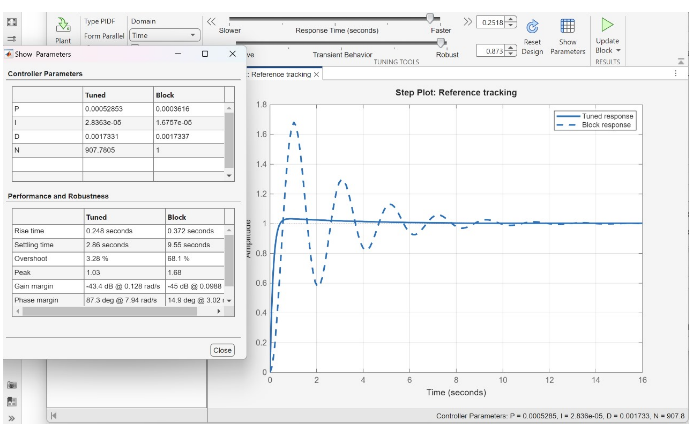

Fig. 15: Response when tune with PID controller.

Task 4: Simulation of Ball and Beam System

Task 4.1: Build the ball beam model in Simulink and generate the system's open loop response.

To simulate a ball and beam system in MATLAB using Simulink and generate the system's open-loop response

When the simulation is finished, open the Scope by double clicking on it.

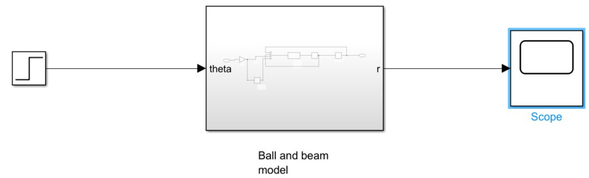

Fig. 16: open loop simulation.

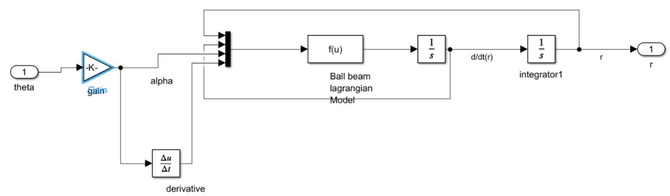

Fig. 17: Inside ball and beam model.

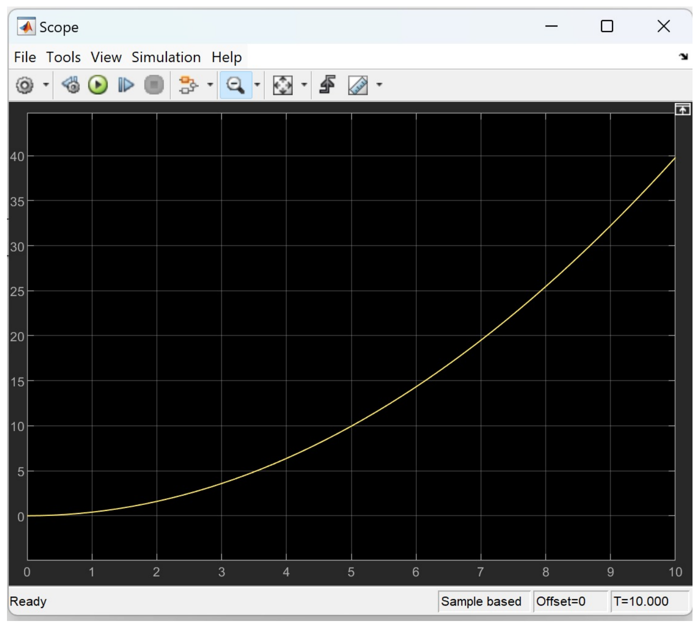

Fig. 18: open loop response.

Task 4.2: Linearizing the Model and Designing a Compensator

In this section, we linearize the ball and beam system model and design a lead/lag compensator to meet the specified design criteria. The design goals are:

- Overshoot of less than 5%.
- Settling time of less than 5 seconds.

To achieve these performance criteria, we first linearize the system around its equilibrium point using a small-angle approximation for the beam angle. The linearized model is then used to design a compensator, such as a Proportional-Integral-Derivative (PID) controller, to stabilize the system.

By tuning the compensator parameters, we ensure that the closed-loop response satisfies the desired performance metrics. Finally, the closed-loop response of the system is generated and analyzed to verify that the overshoot is below 5% and the settling time is less than 5 seconds.

Fig. 19: Linearization of the open loop model in simulation.

We can verify this model by obtaining an open-loop step response.

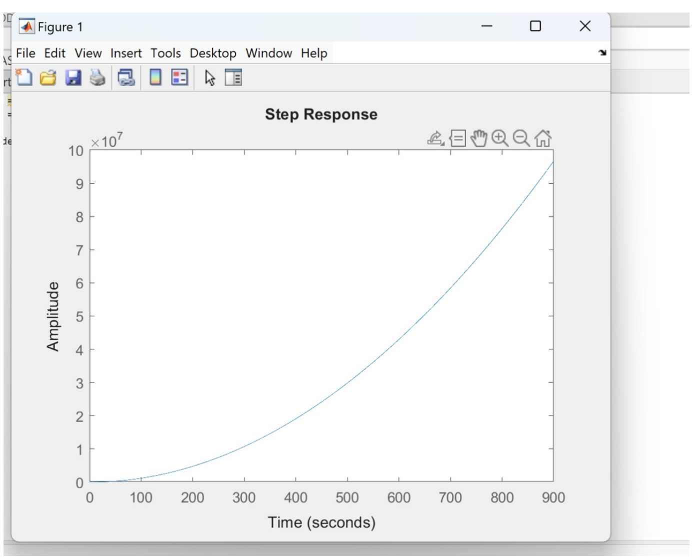

Fig. 20: Open-loop step response.

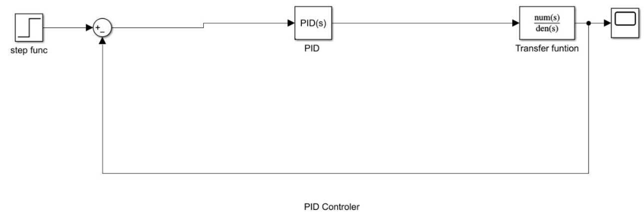

Fig. 21: Closed-loop model in simulation.

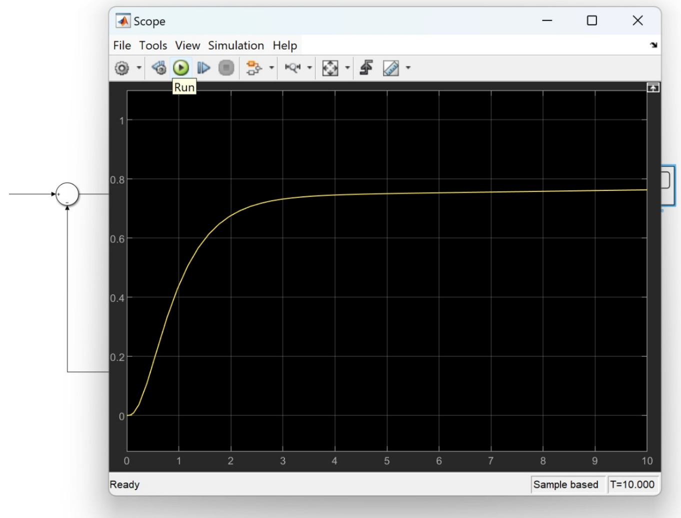

Fig. 22: Closed-loop response.

Task 4.3: Simscape Model for the Ball Beam System

The system's physical parameters are defined in the following sections.

A. System's Physical Parameters

The physical parameters of the ball and beam system are defined in the `ball_beam_properties.m` file. These parameters play a crucial role in determining the system's dynamics and the behavior of the ball along the beam under the influence of external forces:

- **Mass of the ball** (m): 0.01
- **Radius of the ball** (R): 0.015
- **Distance from pivot to beam** (d): 0.03
- **Length of the beam** (L): 0.34
- **Moment of inertia of the ball** (J): 9.0e-7

B. System Configuration

The system's mechanical configuration is modeled using multiple revolute joints and transformation elements, ensuring accurate simulation of the physical behavior. The key components are:

- **Revolute Joints**: These joints provide rotational motion between connected parts, such as the beam, lever, and gear, allowing for dynamic movement within the system.
- **Transformation Elements**: These elements are used to manage the spatial transformations of the pivot and gear axes. They ensure proper alignment and motion of the beam and gear as the system operates.

This configuration, combined with the system's physical parameters, ensures accurate modeling and simulation of the ball and beam system.

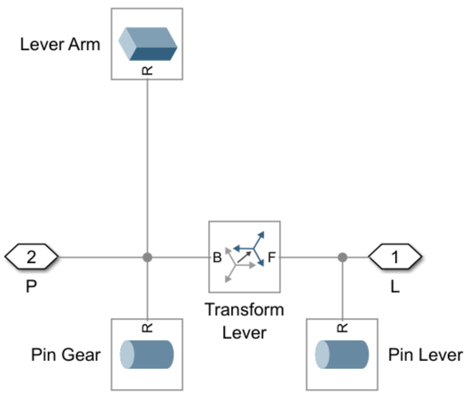

Fig. 23: Gear.

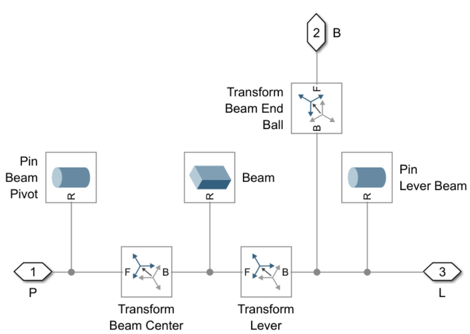

Fig. 24: **Beam.**

Task 5: Physical System and PID Implementation

C. Fabrication Details

The physical system for the Ball and Beam control was constructed using acrylic materials due to their sturdiness and accuracy. Bearings and metal rods were also used to create a revolute joint, which ensures smooth movement of the beam. The dimensions of the bearings are 5 mm (inner diameter) and 16 mm (outer diameter). The details of the system components are provided in Table I.

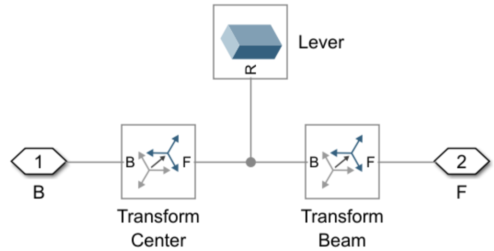

Fig. 25: **Lever.**

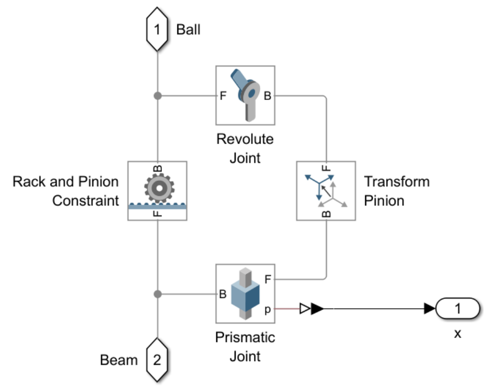

Fig. 26: **Ball Beam Constraints.**

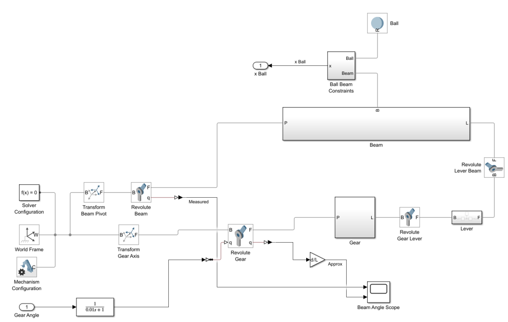

Fig. 27: **Ball on Beam.**

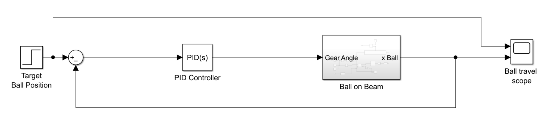

Fig. 28: **The resulting closed-loop system.**

D. Challenges and Observations

The PID control system was implemented to stabilize the ball at a setpoint of 10 cm from one end of the beam. The main challenges encountered during the development process

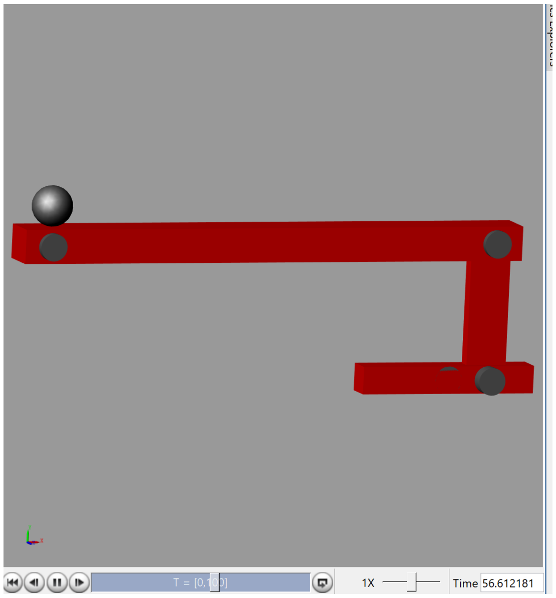

Fig. 29: **Screenshot of the system.**

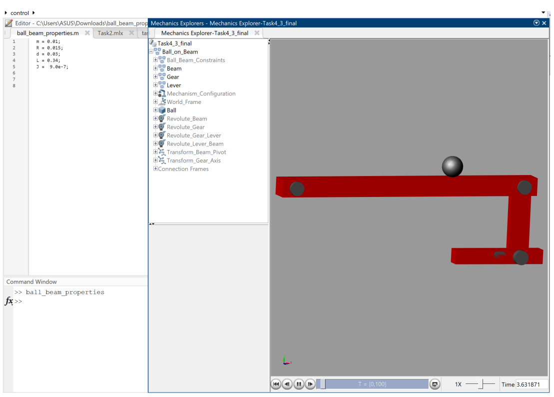

Fig. 30: **Simscape model for the ball beam system.**

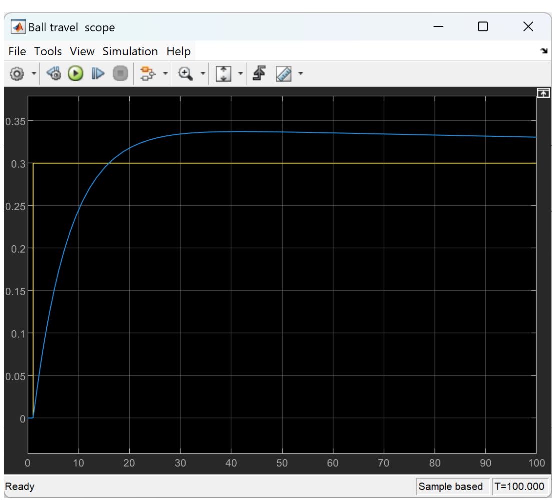

Fig. 31: **Output of Simscape Model.**

are listed in Table II.

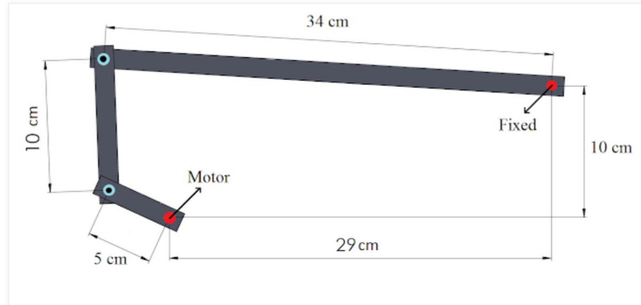

Fig. 32: Side View of the Setup

TABLE I: Fabrication Details of Ball and Beam System Components

Component	Dimensions (mm)	Material	Weight (g)
2*Arm 1	Length: 50 Width: 20, Thickness: 5	2*Acrylic	2*35
2*Arm 2	Length: 100 Width: 20, Thickness: 5	2*Acrylic	2*55
2*Beam	Length: 340 Width: 20, Thickness: 5	2*Acrylic	2*210
Base	150 x 350	Acrylic	300
Bearings	Inner: 5 mm, Outer: 16 mm	Metal	15
Metal Rods	40,20,10	Metal	10
Ultrasonic	N/A	N/A	17
Ball	Diameter: 4 (cm)	T.T Ball	11

The system demonstrated successful control, and the ball reached steady-state near the 10 cm setpoint, as shown in the distance vs. time plot in Fig. 37.

E. Images of the Setup

Three images of the physical setup are shown below:

Fig. 33: Complete Ball and Beam Setup

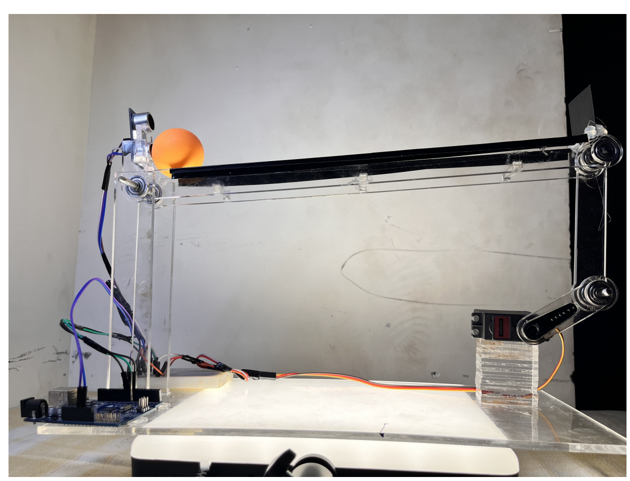

Fig. 34: Close-up of Servo Motor and Arm 1

TABLE II: Challenges and Solutions

Challenge	Solution
Delay in sensor feedback	Tuning of the PID controller to optimize response time.
Vibration due to lightweight acrylic material	Introduction of damping mechanisms in the base structure.
Initial oscillations of the ball	Adjustments in proportional and derivative gains.

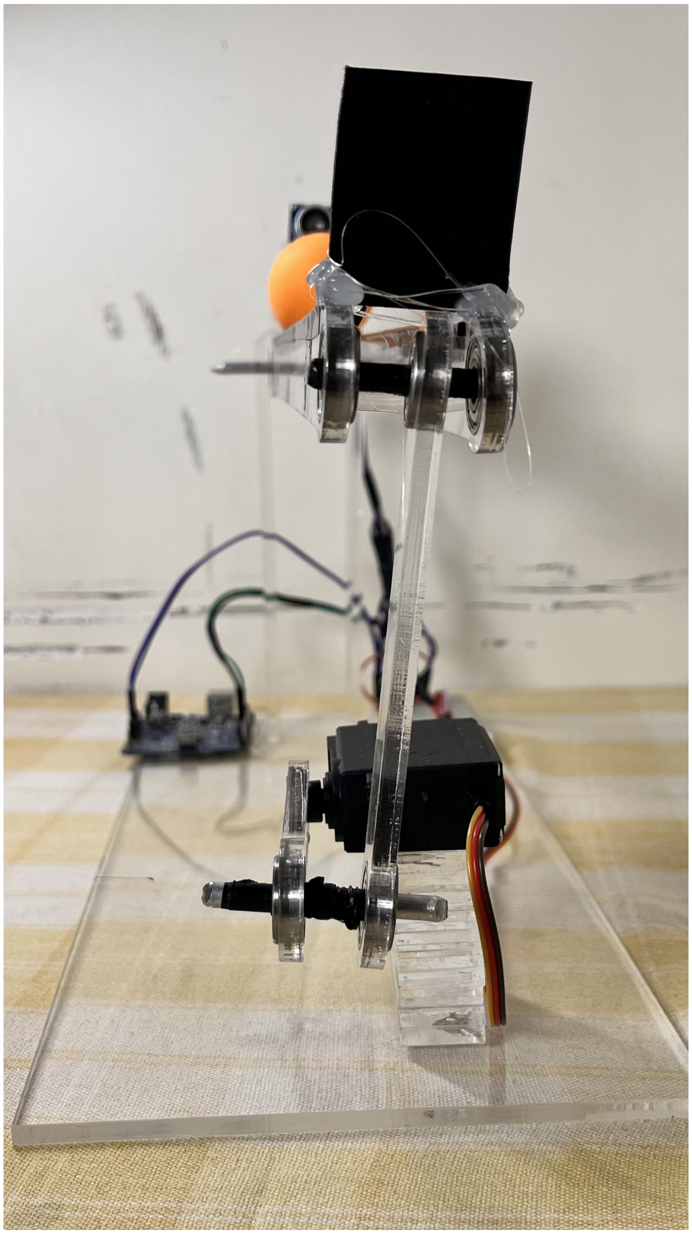

Fig. 35: Another Side View of the Setup

F. Real-Time Plot of Ball's Position

During testing, the real-time distance versus time plot, shown in Fig. 37, was generated using MATLAB. The ball stabilized at approximately 10 cm as per the design requirements.

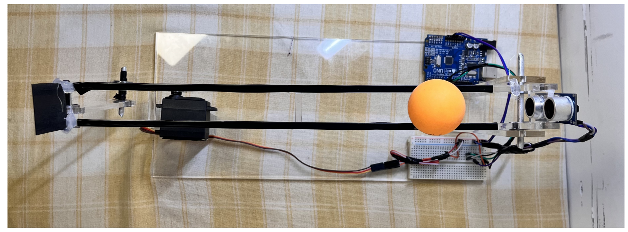

Fig. 36: Top View of the Setup

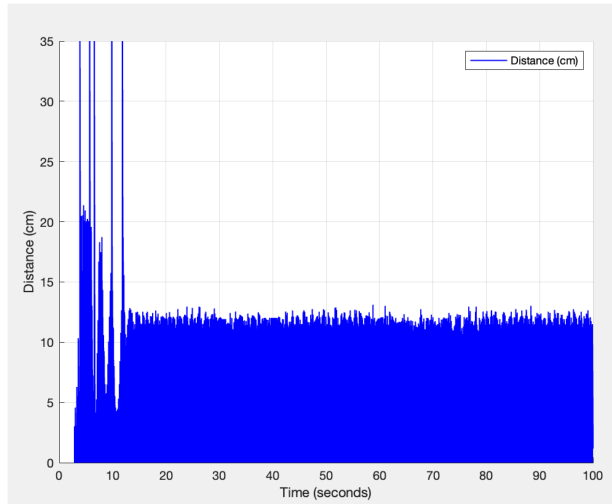

Fig. 37: Real-time Distance vs Time Plot

G. Conclusion

The physical system was fabricated using acrylic materials and revolute joints with bearings and metal rods, which allowed smooth movement of the beam. A PID controller was implemented to successfully stabilize the ball at the desired position. The challenges encountered during development were addressed through systematic PID tuning and mechanical adjustments, and the final system achieved the required performance metrics.

REFERENCES

[1] Control Tutorials for MATLAB and Simulink. "Ball & Beam: Simulink Simscape." [Online]. Available: https://ctms.engin.umich.edu/CTMS/index.php?example=BallBeamsection=SimulinkSim [Accessed: 20-Oct-2024].

[2] M. Tutorials, "Balancing of Ball on Beam using Arduino," [Online]. Available: https://mechatronicstutorials.blogspot.com/2014/07/balancing-of-ball-on-beam-using-arduino.html. [Accessed: 20-Oct-2024.

[3] YouTube. "YouTube Video 1," [Online]. Available: https://www.youtube.com/watch?v=y5Hmr-rR-yk. [Accessed: 20-Oct-2024].

[4] YouTube. "YouTube Video 2," [Online]. Available: https://www.youtube.com/watch?v=gfOChWXEgq4. [Accessed: 20-Oct-2024.

[5] YouTube. "YouTube Shorts," [Online]. Available: https://www.youtube.com/shorts/0JT0tff6nsA. [Accessed: 20-Oct-2024.