Ball and Beam System Analysis

Task 1: Define the System

1.1 Write the system dynamics equations.

To derive the system dynamics equations for the ball and beam system using the Lagrangian equation of motion.

Assumptions

- The ball rolls without slipping on the beam.
- Frictional losses and air resistance are negligible.
- The motor controls the angle θ, which affects α through the lever arm.

System Parameters

- **Ball**: Positioned on the beam, it rolls without slipping.
- **Beam**: Length L, attached to a fixed joint at one end.
- **Lever Arm**: Connected to the beam at the fixed joint and to the motor gear via a pin joint.
- **Motor Gear**: Rotates the lever arm, causing the beam to tilt.
- a: The distance from the fixed joint to the ball.
- L: The total length of the beam.
- α: The angle of the beam relative to the horizontal.
- θ: The angle of the lever arm relative to a reference line.
- d: The distance between the motor gear's center and the pin joint of the lever arm.
- R: Radius of the ball.
- J: Moment of inertia of the ball about its center of mass.

Kinetic Energy Calculation

The kinetic energy of the system consists of two components:

1. **Translational Kinetic Energy**: This corresponds to the motion of the ball's center of mass along the beam.

$$T_{\text{trans}} = \frac{1}{2}m\dot{a}^2$$

2. **Rotational Kinetic Energy**: This corresponds to the ball's rotation about its own center.

$$T_{\text{rot}} = \frac{1}{2}J\omega^2$$

where ω is the angular velocity of the ball. Since the ball rolls without slipping, the linear velocity of the ball's center of mass and its angular velocity are related by:

$$\omega = \frac{\dot{a}}{R}$$

The total kinetic energy T of the system is the sum of these two components:

$$T = \frac{1}{2}m\dot{a}^2 + \frac{1}{2}J\left(\frac{\dot{a}}{R}\right)^2$$

Potential Energy Calculation

The potential energy of the ball is due to its height above the horizontal. If we take the vertical displacement of the ball as $a\sin\alpha$, then the potential energy is given by:

$$V = mga\sin\alpha$$

Lagrangian

The Lagrangian L is defined as the difference between the kinetic energy T and the potential energy V:

$$L = T - V$$

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

Applying the Lagrangian Equation of Motion

The Lagrangian equation of motion is given by:

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

Applying this equation will yield the system dynamics equations for the ball and beam system.

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

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

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

$$\frac{\partial L}{\partial a} = -mg\sin\alpha$$

Including Centrifugal Acceleration

When the beam rotates with an angular velocity $\dot{\alpha}$, the ball experiences a centrifugal acceleration given by:

$$-ma\dot{\alpha}^2$$

where:

- m is the mass of the ball,
- a is the distance of the ball from the pivot point along the beam,
- $\dot{\alpha}$ is the angular velocity of the beam.

Final Equation of Motion

Substituting these terms into the Lagrangian equation gives:

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

where:

- J is the moment of inertia of the ball,
- R is the radius of the ball,
- g is the gravitational acceleration,
- $\ddot{a}$ is the acceleration of the ball along the beam.

Rearranging, we obtain the final equation of motion:

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

Relationship Between Variables

Angular Position: The angle of the beam (α) can be related to the motor gear angle (θ) as follows:

$$\alpha = \frac{d}{L}\theta$$

This relation arises from the geometry of the system, where d represents the effective lever arm of the motor.

Linearization

For small angles α, we can approximate:

$$\sin\alpha \approx \alpha$$

Also, for small perturbations around a stable point, we assume that $\dot{\alpha}^2$ term is negligible compared to the other terms. The equation then simplifies to:

$$\left(m + \frac{J}{R^2}\right)\ddot{r} + mg\alpha = 0$$

Substituting $\alpha = \frac{d}{L}\theta$ into the equation gives:

$$\left(m + \frac{J}{R^2}\right)\ddot{r} + mg\left(\frac{d}{L}\theta\right) = 0$$

Laplace Transform to Obtain the Transfer Function

Taking the Laplace transform of the linearized equation (with zero initial conditions), we get:

$$\left(m + \frac{J}{R^2}\right)s^2 A(s) + mg\left(\frac{d}{L}\right)\Theta(s) = 0$$

where $A(s)$ and $\Theta(s)$ are the Laplace transforms of $a(t)$ and $\theta(t)$, respectively. Rearranging for the transfer function $\frac{A(s)}{\Theta(s)}$:

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

Final Transfer Function

The transfer function between the motor gear angle θ and the ball position a is:

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

Task 2: Analysis of the system

Parameter Values for the Ball-and-Beam System

We assume suitable values for the parameters used in the transfer function of the ball-and-beam system as follows:

- Mass of the ball, m: $0.0023\,\mathrm{kg}$
- Radius of the ball, R: $0.02\,\mathrm{m}$
- Lever arm offset, d: $0.07\,\mathrm{m}$
- Length of the beam, L: $0.25\,\mathrm{m}$
- Ball's moment of inertia, J: $\frac{2}{5}mR^2 = \frac{2}{5}(0.1)(0.02)^2 = 0.000000368\,\mathrm{kg}\cdot\mathrm{m}^2$

Transfer Function Calculation

The transfer function is given by:

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

Acceleration due to gravity:

$$g \approx -9.81\,\mathrm{m/s}^2$$

Thus, the transfer function is approximately:

$$H(s) \approx \frac{1.96}{s^2}$$

Transfer Function Characteristics

Given the transfer function:

$$H(s) = \frac{1.96}{s^2}$$

we can make the following observations:

Type of System

- The transfer function has a denominator of s^2, which means the system is a **second-order system**.
- There are no zeros in the transfer function (numerator is a constant), and there are two poles located at the origin ($s = 0$).
- Since the transfer function contains s^2 in the denominator, this indicates that the system has **two integrators**, making it a **Type 2 system**.

Poles/Zeros Plot Analysis

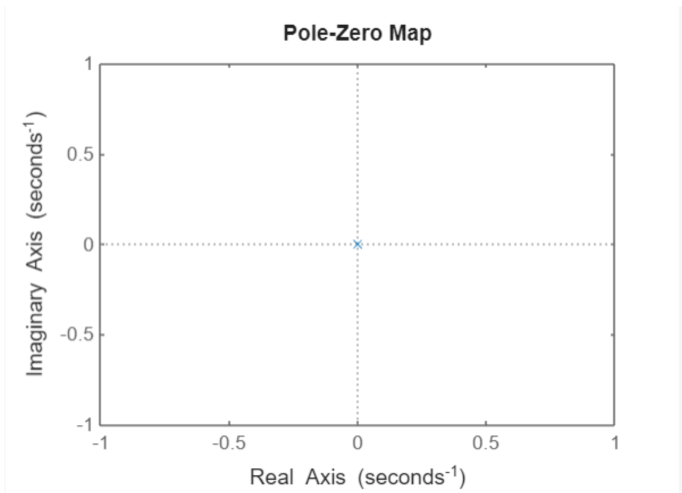

Figure 1: Poles

From the Fig.1 the poles/zeros plot:

- **Poles at the Origin**: There are two poles located at $s = 0$, confirming the system's Type 2 classification. These poles indicate that the system has two integrators in the forward path.
- **No Zeros**: The system does not have any zeros, as the numerator is a constant value.

Open-Loop Step Response Observation

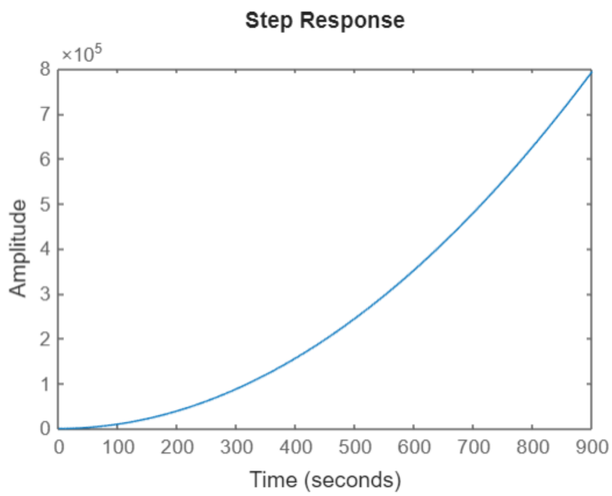

Figure 2: Open-Loop Step Response

From the open-loop step response:

- The response does not settle or approach a steady-state value, instead continuing to rise. This is typical for a system with poles at the origin (pure integrators), which means the system cannot stabilize.
- The increasing amplitude suggests that the system is **unstable** in its open-loop configuration. For a stable system, the response should either converge to a steady-state value or oscillate within a bounded range, which is not happening here.

Summary

- **System Type**: Type 2 (second-order system with two integrators)
- **Poles/Zeros Plot**: Two poles at the origin, indicating marginal stability.
- **Open-Loop Response Observation**: The system is unstable in open-loop form.

To achieve stability and desired performance, a compensator or controller (e.g., PID controller or lead compensator) would need to be designed to modify the system dynamics appropriately.

Task 3: PID control

A PID controller constantly calculates an error, e(t), which is the difference between a target value (setpoint) and the actual measured value (process variable). It then adjusts the system by applying corrections based on three components: proportional, integral, and derivative. These three terms work together to modify the error and create a control signal. PID stands for proportional-integral-derivative, referring to the three elements that influence how the system responds to the error.

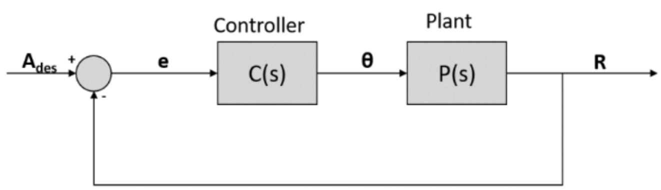

Figure 3: PID Controller

$$\theta(t) = K_p e(t) + K_i \int e(t)\, dt + K_d \frac{de(t)}{dt} \tag{1}$$

- e - the tracking error
- a - desired output
- R - actual output
- θ - control signal
- K_p - proportional gain
- K_i - integral gain
- K_d - derivative gain

The transfer function for a PID controller is:

$$H(s) = K_p + \frac{K_i}{s} + K_d s = \frac{K_d s^2 + K_p s + K_i}{s}$$

Proportional controller

The system's response to a step input of 0.125 m and By setting

$$k_p = 1$$

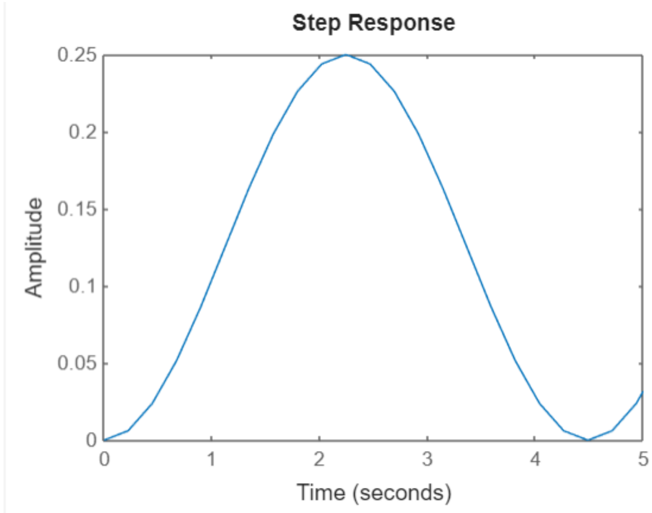

Figure 4: Step Response with Proportional controller

with

$$k_p = 10$$

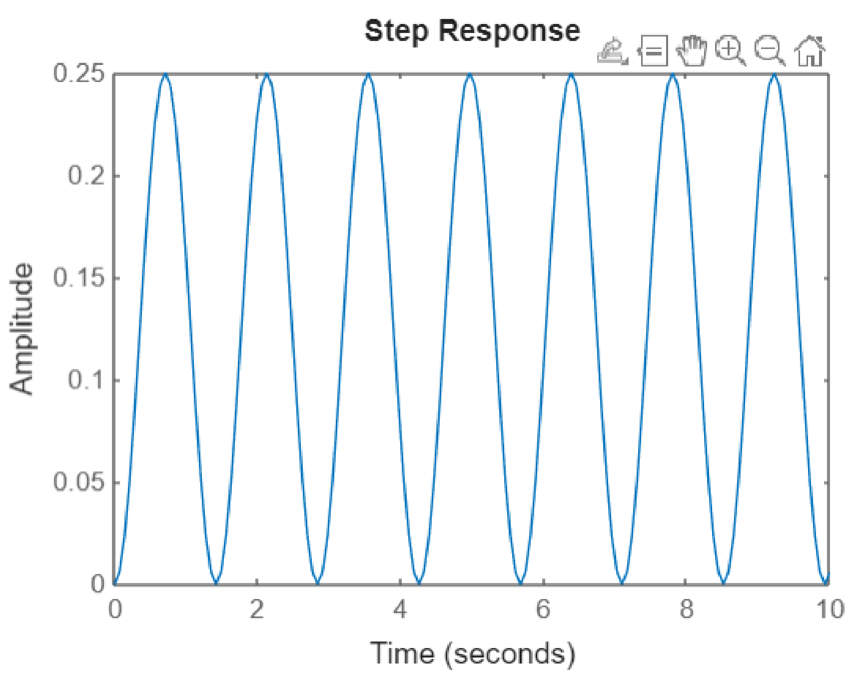

Figure 5: Step Response with Proportional controller

Proportional-derivative controller

Now, by adding derivative controller with

$$k_d = 1$$

$$k_p = 10$$

Plot will be

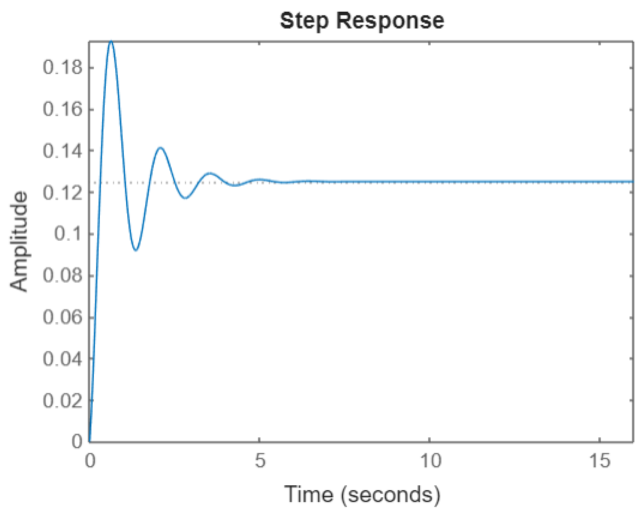

Figure 6: Step Response with Proportional-derivative controller

Now with

$$k_d = 10$$

$$k_p = 10$$

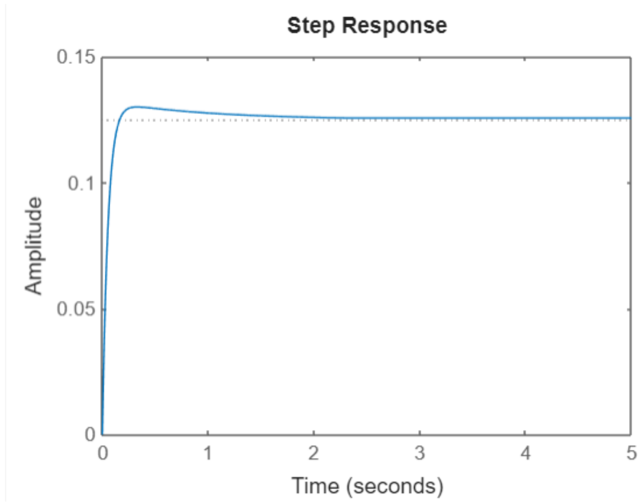

Figure 7: Step Response with Proportional-derivative controller

Proportional-integral-derivative controller

Now, by adding derivative controller with

$$k_p = 10$$

$$k_i = 1$$

$$k_d = 10$$

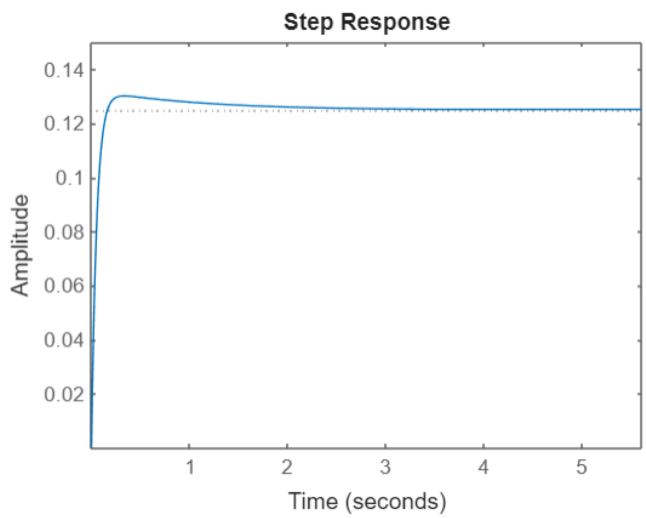

Figure 8: Step Response with Proportional-integral-derivative controller

By varing

$$k_p = 10$$
$$k_i = 0$$
$$k_d = 10$$

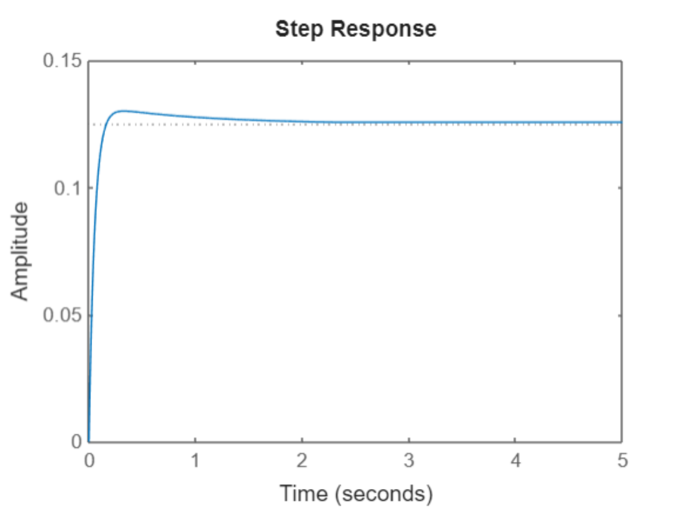

Figure 9: Step Response with Proportional-integral-derivative controller

Observations for Proportional controller

Transient Response:

- **Rise Time:** The system reaches its peak value very quickly.
- **Overshoot:** The system goes past the desired value by a large amount.
- **Oscillations:** The system continues to oscillate, with the size of the oscillations staying quite large.

Steady-State Response:

- **Steady-State Error:** The system does not reach the desired value and stays off by about 0.125 meters , similar to the case when $K_p = 1$.

Set Point Error:

The proportional controller cannot remove the steady-state error, even with a higher K_p value. This is a typical limitation of proportional control, which only uses the current error to adjust the control signal.

Observations for Proportional-derivative controller

Transient Response:

- **Rise Time:** The system reaches its peak value quickly, in about 1 second.
- **Overshoot:** The system goes a bit above the desired value before coming back down.
- **Oscillations:** After reaching the peak, the system has some smaller oscillations, but they settle fairly quickly.

Steady-State Response:

- **Steady-State Error:** The system doesn't fully reach the target value and remains off by about 0.125 meters.

Set Point Error:

The PD controller, like the proportional controller, cannot remove the steady-state error. This happens because the derivative part only affects how quickly the error changes, not the size of the error itself.

Observations for Proportional-integral-derivative controller

Transient Response:

- **Rise Time:** The system quickly reaches its peak value in about 1 second.
- **Overshoot:** The system goes slightly above the desired value before settling.
- **Oscillations:** After reaching the peak, the system has small oscillations, but they stop quickly.

Steady-State Response:

- **Steady-State Error:** The system has almost no error at steady state, meaning it stays at the desired value.

Set Point Error:

The PID controller removes the steady-state error completely, showing that it works well for precise control.

Designing a controller using Root-Locus method

The main idea of root locus design is to predict how a system will behave when it's closed-loop by looking at the root locus plot of the open-loop system. By adding zeros or poles (through a compensator), you can change the root locus and, as a result, improve the system's behavior when it's closed-loop.

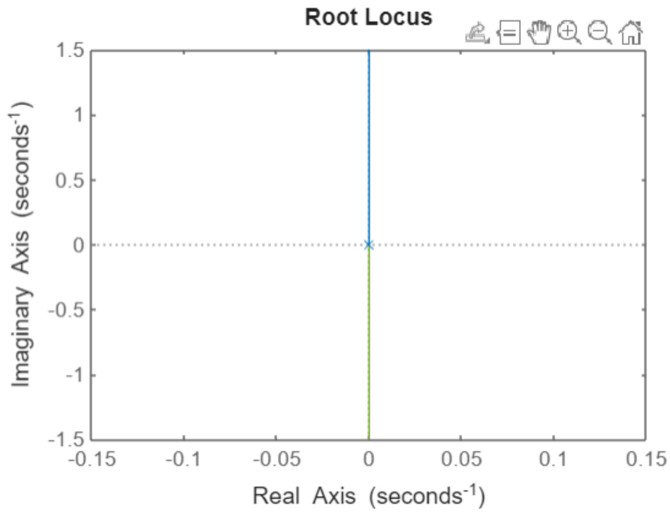

Figure 10: Root Locus

The system has two poles at the origin which go off to infinity along the imaginary axes.

Design Criteria

The design requirements on the root locus observed using the `sgrid` command. This command creates a grid that shows constant damping ratio ζ and natural frequency ω_n. These values are calculated using the following equations, based on the maximum overshoot (Mp) and settling time (T_s):

$$Mp = e^{\frac{-\pi\zeta}{\sqrt{1-\zeta^2}}}$$

$$T_s = \frac{4}{\zeta\omega_n}$$

Since our our design criteria is settling time (T_s) less than 3 seconds and overshoot (Mp) less than 5

By soving above equations we will get

$$\zeta \approx 0.7, \quad \omega_n \approx 1.905\,\mathrm{rad/s}$$

From the above graph The region between the two dotted diagonal lines shows where the percent overshoot is less than 5%. The area outside the curved line represents locations where the settling time is under 3 seconds. However, no part of the plot meets the design criteria indicated by these lines. To fix this and move the root locus into the left-hand plane for stability, we will attempt to add a lead compensator to the system.

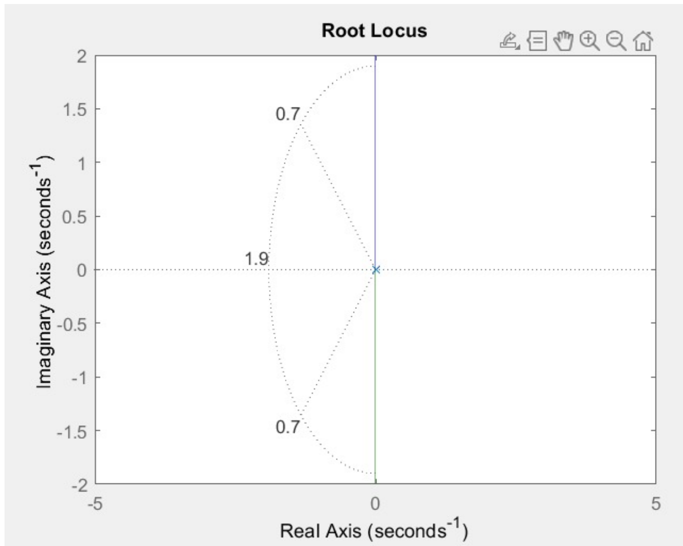

Figure 11: Root Locus

Lead compensator

This type of compensator moves the root locus into the left-hand plane, which generally improves the system's stability and transient response.

The transfer function of a lead compensator is given by:

$$C(s) = K_c \cdot \frac{s + z_o}{s + p_o}$$

Where:

- K_c is the compensator's gain
- z_o is the location of the zero
- p_o is the location of the pole

To design a lead compensator, the zero is typically placed near the origin to cancel out one of the plant's poles. The pole of the compensator is then placed to the left of the origin, which shifts the root locus further to the left, resulting in a faster and more stable system response. For this taking

$$z_o = 0.03$$

$$p_o = 4$$

we will get plot

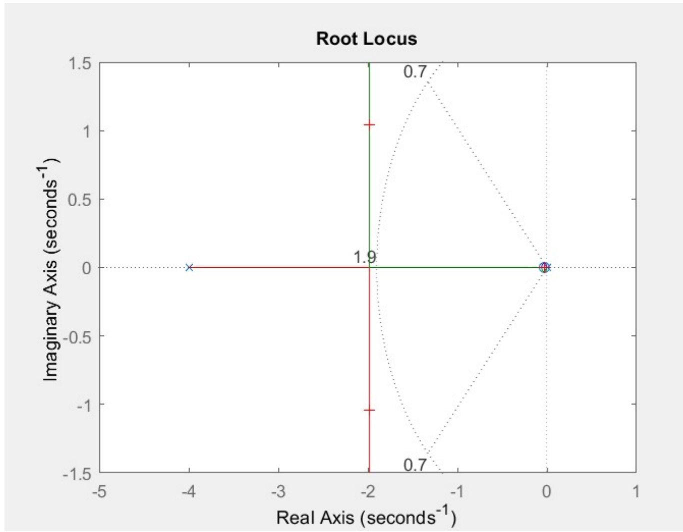

Figure 12: Gain selection

```
Select a point in the graphics window

selected_point =

  -1.9830 + 1.0422i



k =

    2.6260



poles =

  -1.9846 + 1.0422i
  -1.9846 - 1.0422i
  -0.0307 + 0.0000i
```

Figure 13: Output in the command window

Closed-Loop Response

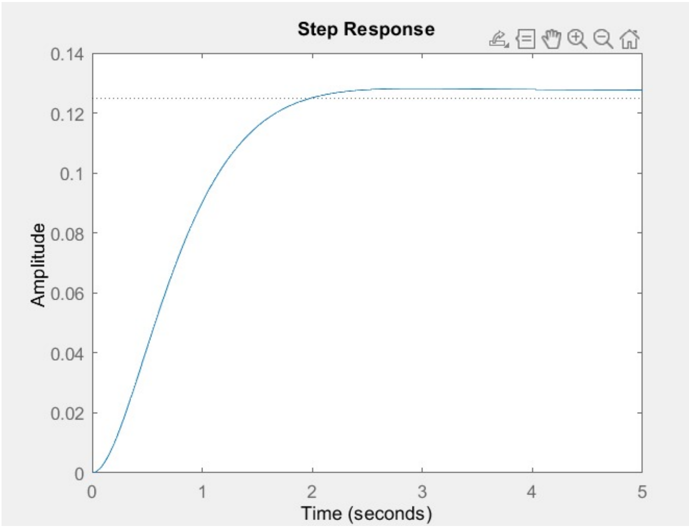

Figure 14: Closed loop response

From the above response we can say that settling time and percentage overshoot designcriteria are met

Build the ball beam model in Simulink

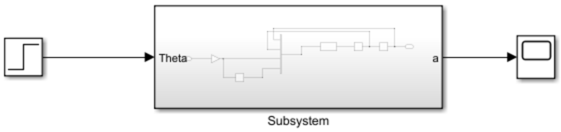

Figure 15: Simulink simulation

For this we will get the system's open loop response

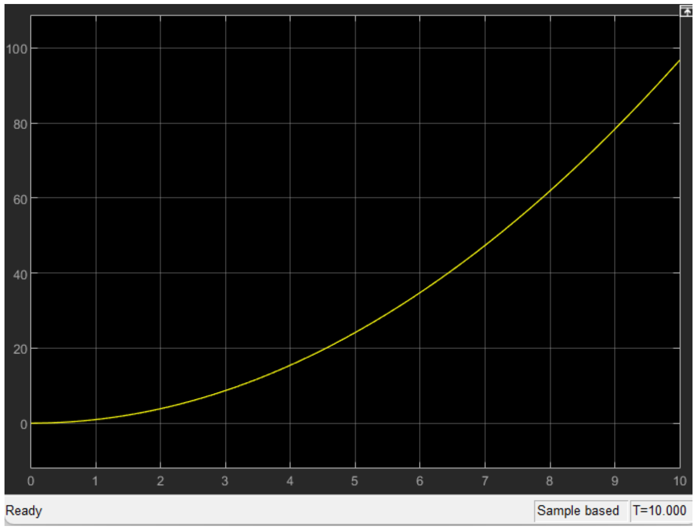

Figure 16: open loop response

Lead compensator

Since the system is unstable we are going to designing the Lead compensator using Root Locus Controller.

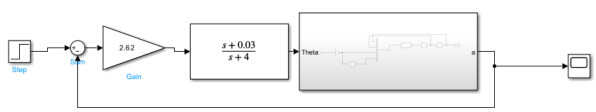

Figure 17: Root Locus Controller

For this we will get the system's closed-loop response.

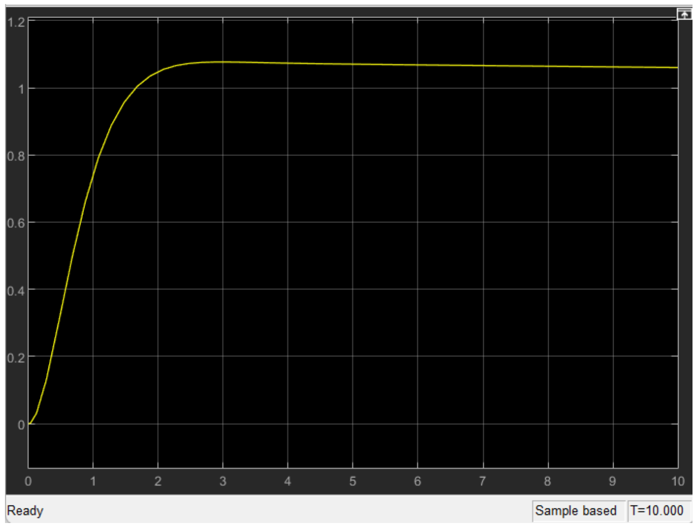

Figure 18: System's closed-loop response.

So, we got design criteria - overshoot of less than 5percentage and settling time of less than 5 seconds.

Simscape

We have done Half the simscape model

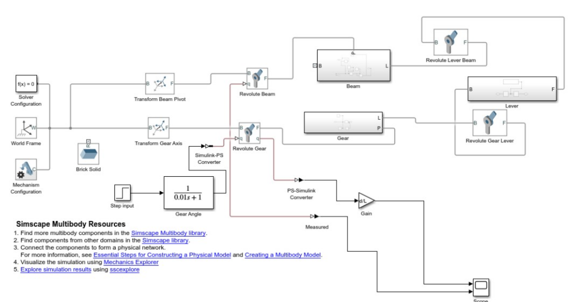

Figure 19: Simscape

Task 5

Physical Setup

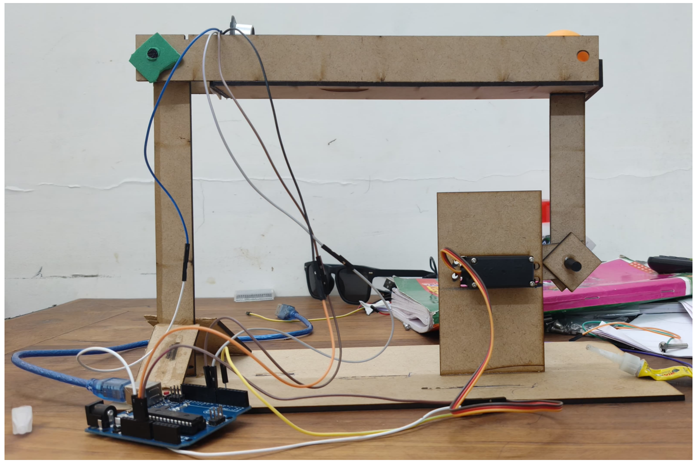

Figure 20: Physical Setup

CAD Model

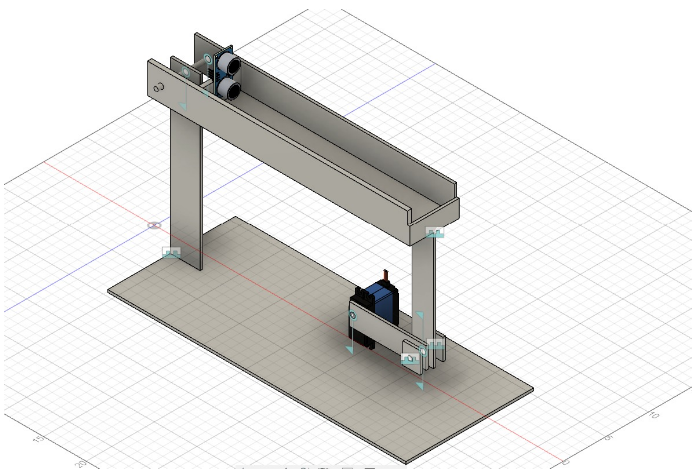

Figure 21: CAD Model