Ball and Beam (Part-1 Report)

Shrishti Mishra (22110246), Debojit Das (22110067), Vardan Mittal (24250104), Aaditya Nimkar (24250002)

Department of Mechanical Engineering, IIT Gandhinagar

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

1. Problem Statement

This project aims to design and implement a control system for manipulating the position of a ball on a beam, utilizing the principles of dynamics and control theory. The system consists of a ball that rolls along a beam, influenced by gravity, while its position is adjusted through a servo motor connected to a lever arm that alters the beam's angle.

The experimental setup will feature a ball in continuous contact with the beam, ensuring pure rolling motion without slipping. The primary objective is to establish a robust controller that takes the servo gear angle as input and outputs the position of the ball along the beam.

The expected outcomes of this project include:

1. A comprehensive understanding of the ball-beam system dynamics and control strategies.
2. Detailed analysis of system behavior through MATLAB simulations, including characterization of transient and steady-state responses.
3. Successful design and implementation of a physical control system that effectively stabilizes the ball on the beam while meeting specified performance criteria.

2. Introduction

The ball and beam system is a widely used experiment in control design labs because of its simplicity and versatility in demonstrating both classical and modern control techniques. The setup is relatively straightforward: a motor is connected to the beam in such a way that it can control the beam's angle. This is typically done using a lever arm or by directly attaching the motor shaft to the center of the beam. In this project, the motor shaft is coupled directly to the beam's center for easier assembly and troubleshooting. The motor adjusts the beam's angle, moving the ball along the beam to a desired position.

An encoder usually provides feedback to measure the motor's position and a sensor (such as a resistive wire or acoustic sensor) to track the ball's position on the beam.

Though the system is simple to assemble, the techniques used to control and evaluate it can be applied to more complex systems. The ball and beam system provides a safe way to explore control strategies for unstable systems. Without feedback, the system is open-loop unstable, meaning that if the beam is tilted at a constant angle, the ball's position will continue to change indefinitely. Therefore, a feedback loop is required to stabilize the ball's position.

While the ball and beam system isn't directly modelled after a real-world system, its dynamics are similar to those found in aerospace applications, such as controlling the vertical thrust in rockets or vertical take-off aircraft. In such systems, the angle of the thrusters must be constantly adjusted to maintain stability. Unstable control problems also arise in industries like chemical processing, where exothermic reactions require careful regulation. This makes the study of the ball and beam system relevant for understanding and solving various real-world control challenges.

3. Apparatus

The design of the apparatus will be as follows:

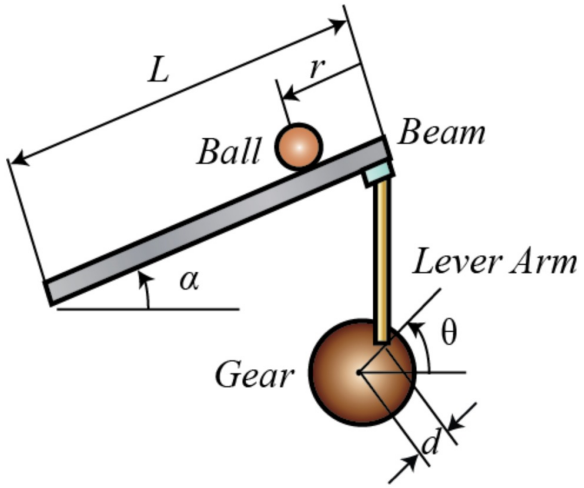

Figure 1. *Proposed Design*

Components

The following components were primarily used:

- **Acrylic Sheet**: Used to create the main frame, including the beam, lever and other supports.
- **Cardboard Base**: The setup is mounted on a flat cardboard base, providing stability.
- **Arduino Uno**: It sends signals to the motor to adjust the beam's angle and receives feedback from the sensors regarding the ball's position on the beam.
- **Ultrasonic Sensor**: An ultrasonic sensor is mounted on the beam to detect the position of the ball along the length of the beam
- **Jumper Wires**: Employed to connect different system components.
- **3D Printed Ring**: Designed for smooth motion as the lever rotates, and ensures precise dimensions and smooth

rotation, helping maintain the balance and accuracy of the beam's motion.

- **Sensor Holder**: Designed to securely hold the sensors in place while ensuring alignment with the beam for accurate readings.

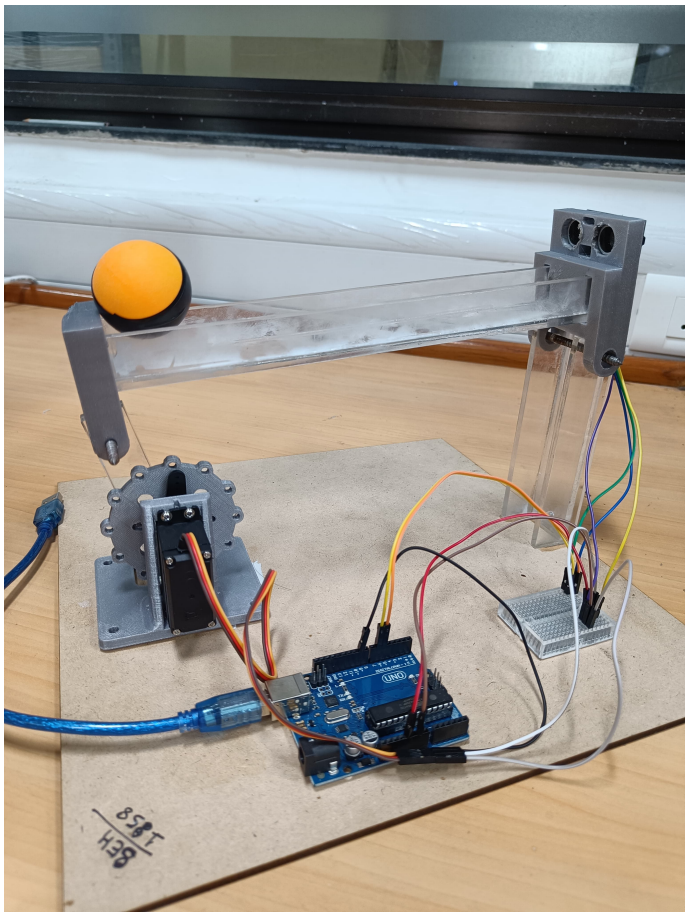

Figure 2. *Ball and beam assembly*

4. Theoretical Calculation

1. Total Kinetic Energy for the Ball:

$$KE = \frac{1}{2}m\dot{x}^2 + \frac{1}{2}I\omega^2$$

(by no slip condition, $\omega = \frac{\dot{x}}{r}$)

2. Total Potential Energy for the Ball (assuming thickness $\ll L$ of the beam):

$$PE = mg(L - x)\sin(\alpha)$$

3. Relationship for Angular Acceleration:

$$\alpha = k\theta$$

4. Using Lagrangian:

$$L = KE - PE = \frac{1}{2}\left(m + \frac{I}{r^2}\right)\dot{x}^2 - mg(L - x)\sin(k\theta)$$

Now using Euler-Lagrangian formulation:

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

$$\frac{\partial L}{\partial x} = \left(m + \frac{I}{r^2}\right)\dot{x}$$

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{x}}\right) = \left(m + \frac{I}{r^2}\right)\ddot{x} = -mg\sin(k\theta) \qquad \text{(A)}$$

$$\frac{\partial L}{\partial x} = mg\sin(k\theta)$$

Now for the transfer function, taking Laplace transform on both sides in (A):

$$\left(m + \frac{I}{r^2}\right)s^2X(s) = -mgk\Theta(s)$$

Now,

$$\text{TF} = \frac{C(s)}{R(s)} = \frac{-mgk}{m + \frac{I}{r^2}s^2}$$

For state space representation:

$$\left(m + \frac{I}{r^2}\right)s^2X(s) = -mgk\Theta(s)$$

Taking inverse Laplace:

$$\left(m + \frac{I}{r^2}\right)\ddot{x} = -mg\theta$$

Let:

$$x_1 = x, \quad x_2 = \dot{x} \quad \Rightarrow \quad \dot{x}_1 = x_2$$

$$\dot{x}_2 = -\frac{mgk}{m + \frac{I}{r^2}}\theta$$

The system can be represented as:

$$\begin{bmatrix}\dot{x}_1 \\ \dot{x}_2\end{bmatrix} = \begin{bmatrix}0 & 1 \\ 0 & 0\end{bmatrix}\begin{bmatrix}x_1 \\ x_2\end{bmatrix} + \begin{bmatrix}0 \\ -\frac{mgk}{m+\frac{I}{r^2}}\end{bmatrix}\theta$$

The output equation is:

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

5. MATLAB formulation

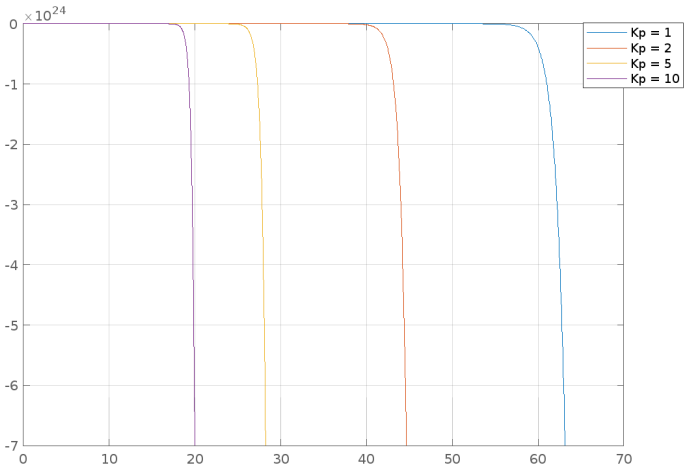

Figure 3. *Kp plots*

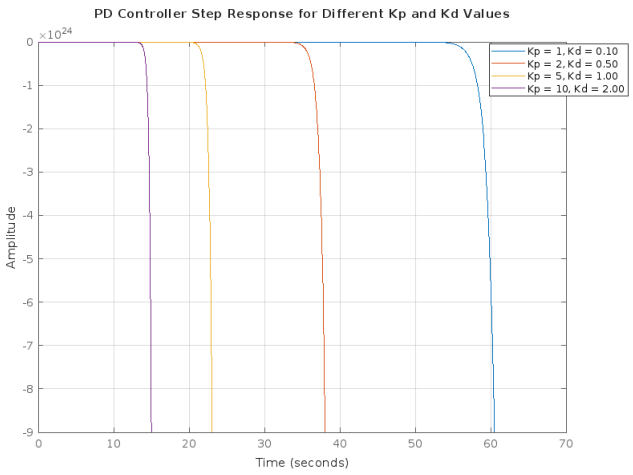

Figure 4. *PD plots*

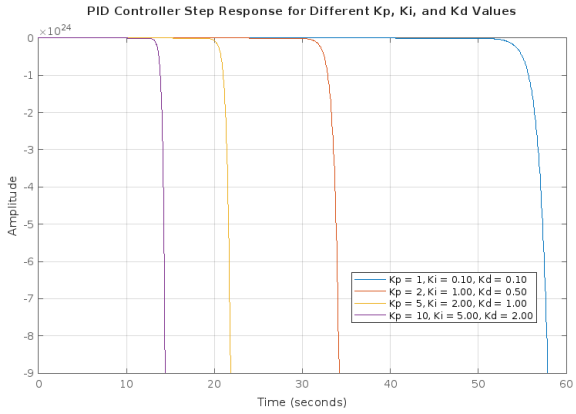

Figure 5. *PID plots*

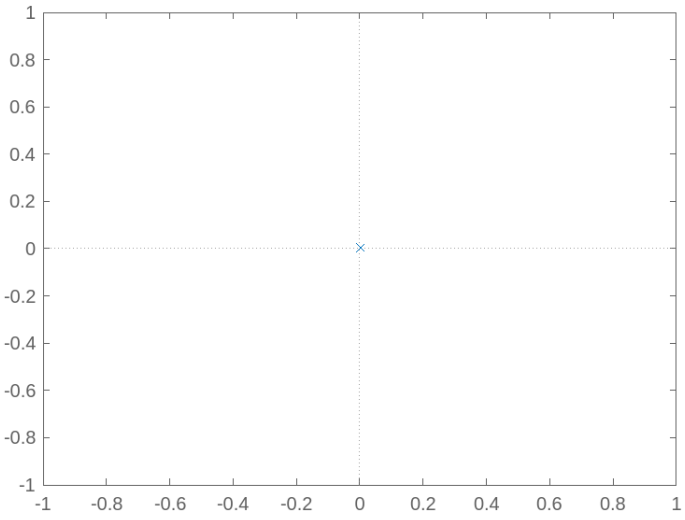

Figure 6. *Zeroes/Poles plots*

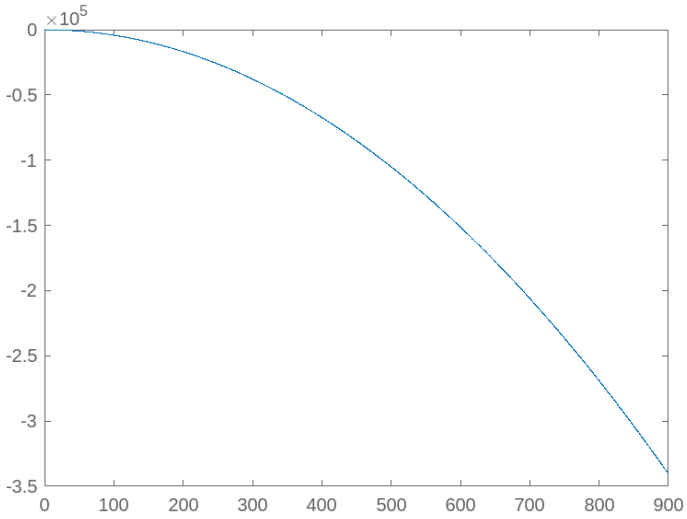

Figure 7. *Plot of Servo Response*

Figure 2 shows the step response of a system with different proportional gain (Kp) values (1, 2, 5, and 10). The response curves show how increasing Kp affects the system's response time - higher Kp values result in faster responses, as shown by the earlier steep drops in the curves.

Figure 3 represents a PD (Proportional-Derivative) controller step response with varying Kp and Kd values. Each curve represents a different combination of gains: Kp = 1, Kd = 0.10 Kp = 2, Kd = 0.50 Kp = 5, Kd = 1.00 Kp = 10, Kd = 2.00

Figure 4 shows the complete PID controller step response with different combinations of all three gains (Kp, Ki, and Kd). Kp = 1, Ki = 0.10, Kd = 0.10 Kp = 2, Ki = 1.00, Kd = 0.50 Kp = 5, Ki = 2.00, Kd = 1.00 Kp = 10, Ki = 5.00, Kd = 2.00

Figure 6 is the system response curve. The plots show how the servo motor angle responds to a unit step input The x-axis represents time in seconds. The steep drops show the motor moving to its commanded position. Different PID parameters affect how quickly and smoothly the motor moves. Higher Kp shows the fastest response but might be too aggressive; lower Kp shows slower, more gentle movement, and adding Kd helps dampen the motion. Ki helps overcome static friction and ensures position accuracy.

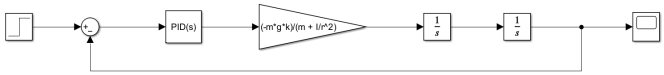

Figure 8. *Simulink*

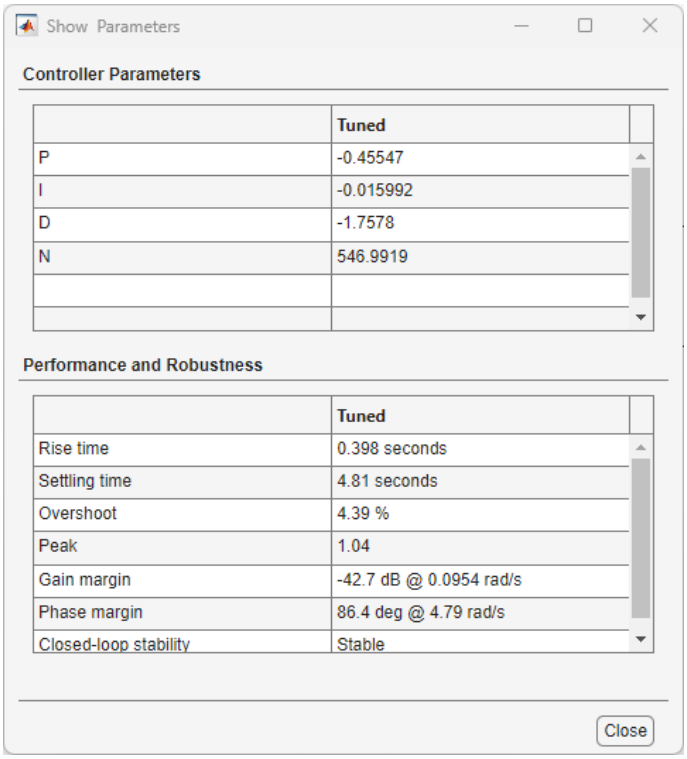

Figure 9. *Controller parameters*

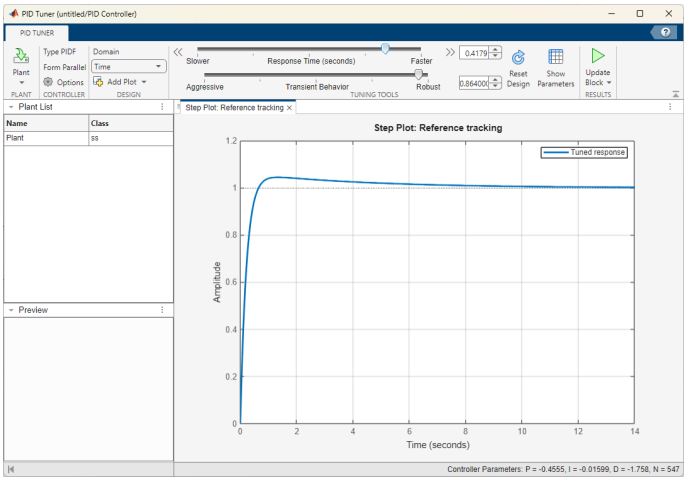

Figure 10. *PID tuner*

The Simulink model includes the following components:

- **Reference Input**: The setpoint or reference input represents the desired state for the system, such as a particular angle or position for the balanced object. In this case, it is represented by a step input, which generates a sudden change in the reference value.

- **Summation Block**: This block calculates the error signal by subtracting the feedback (measured system state) from the reference input. The error signal is then fed into the PID controller to generate the appropriate control signal.

- **PID Controller**: The proportional-integral-derivative (PID) controller is used to minimize the error between the system output and the reference input. The controller output is used to adjust the system dynamics.

$$\text{PID}(s) = K_p + \frac{K_i}{s} + K_d s$$

where K_p, K_i, and K_d are the proportional, integral, and derivative gains, respectively.

- **System Dynamics (Transfer Function)**: This block represents the dynamic system you are controlling, which has a transfer function given by:

$$\frac{-mgk}{m + \frac{I}{r^2}}$$

Here, m is the mass, g is the gravitational constant, k is the control input gain, I is the inertia, and r is the radius. This block models how the control input influences the system's behavior over time.

- **Integrator Blocks**: Two integrator blocks represent the system's state equations. The first integrator models the velocity or rate of change of position, while the second integrator models the position itself. Both integrators have the transfer function:

$$\frac{1}{s}$$

where s is the Laplace variable, and the output of the integrators represents the state of the system.

- **Feedback Path**: The system includes a feedback loop where the system's output is fed back and subtracted from the reference input to minimize the error. This feedback is critical for maintaining balance or control of the system.

The unity feedback is taken to linearize the model exact sensor behaviour is not captured as we are unable to use the hardware through Matlab.

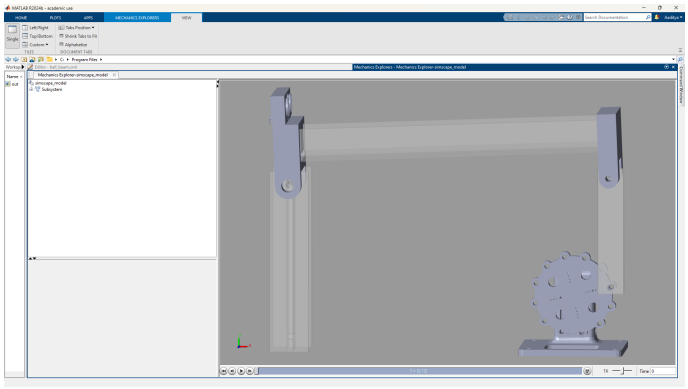

Figure 11. *Simscape model*

6. Implementation Challenges

During the implementation of this project, we encountered several challenges that impacted the development process. These challenges are outlined below:

- **Issues with Simscape model Import from SolidWorks:** While attempting to import the SolidWorks assembly files into Simscape using the official plugins, we encountered multiple issues. These problems included incompatibility between the model structure and the Simscape environment, as well as errors during the file import process. This significantly delayed the integration of the mechanical system into the simulation environment.
- **Problems with MATLAB-Arduino Communication:** Initially, we intended to use MATLAB to directly control the Arduino, allowing us to leverage MATLAB's PID tuning tools for real-time system adjustment. However, continuous errors occurred during the communication between MATLAB and Arduino. Despite troubleshooting the connection and software settings, the errors persisted, which eventually forced us to abandon this approach.
- **Faulty Sensor:** One of the ultrasonic sensors used in the system was faulty, which led to no data output. After performing diagnostic tests, we identified the need for a sensor replacement. The faulty sensor caused interruptions in the development process until a new sensor was procured and integrated into the system. Further the sensor is noisy and need to do some signal processing and filtering.
- **Ball Weight Issue:** The ball used in the balancing mechanism was found to be too light for the system. As a result, the ball is not moving even when a large change of angle is made. To address this, we decided to increase the weight of the ball.

Despite these challenges, we were able to find solutions or workarounds for most issues and continue with the project development. The adjustments made in response to these problems helped improve both the mechanical and control aspects of the system.

7. Conclusions and Recommendations

The ball and beam balance system is a very fascinating one to study. The indirect control of ball position through beam angle control is interesting to implement. Constructing and troubleshooting the prototype was quite interesting. The real complexity came when implementing the PID control and fine-tuning it—this part of the project was a roller coaster. Nonetheless, the experience of working with hardware to implement digital control systems was gratifying.

There is a distinct difference between achieving success in simulation and replicating that success in the physical system. Going through the entire design process, from modelling and simulation to the final implementation, deepened our understanding of the system's real-world behaviour. This experience proved invaluable when it came to troubleshooting and debugging the hardware.

One of the most challenging aspects was implementing an observer using the Arduino and fine-tuning the PID controller. Additionally, assembling the physical structure required precision, especially in ensuring smooth lever motion and accurate sensor placement. Although there were some difficulties along the way, we overcame them, and the project was a valuable learning experience overall.

8. Acknowledgements

We would like to sincerely thank Professor Vineet Vashista for his invaluable guidance and support throughout the project and for equipping us with the necessary concepts. His expertise and enthusiasm significantly enriched our learning experience.

Additionally, we extend our heartfelt gratitude to Teaching Assistants Rajdeep and Jenish for their dedication and assistance during our frequent review meetings, which were instrumental in helping us navigate our challenges.

We are also thankful to IIT Gandhinagar for providing us with the platform and resources necessary for the successful execution of this project.

Finally, we would like to express our appreciation to everyone who contributed to making this course and project a rewarding and enlightening experience.