Multidisciplinary
Engineering Technology
COLLEGE OF ENGINEERING

MXET 375 Applied Dynamic Systems Laboratory

Final Project

Ball and Beam

Section 906 - Group 27
Kyle Rex, Vedansh Shah, Matthew Trevino

Project Links

Team Folder Link:

https://drive.google.com/drive/folders/1XYHI5zW7jzhonW9jSEi2PNOZHzeOfwnl?usp=drive_link

General access is set to anyone with a link can view.
01 Report and Presentation - Final report in PDF and MS WORD. Final presentation slides in PDF and PPTX.
02 CAD Design - 3 CAD files used to assemble prototype.
03 Codes - Simulink Simulation, Arduino Code, and Matlab Code.
04 Photos - Photos of prototype.
05 Videos - Proposal video and demo video.
06 Other Files - All other files related to project.

Introduction

Course Context

- As part of the MXET 375 course, a final project was assigned to model, simulate, and physically replicate a chosen dynamic system using concepts from the course.
- This project required the use of Simulink to simulate the system, followed by the creation of a physical prototype.
- Data from the prototype was then collected and compared to the simulation to assess the model's accuracy.

Chosen System

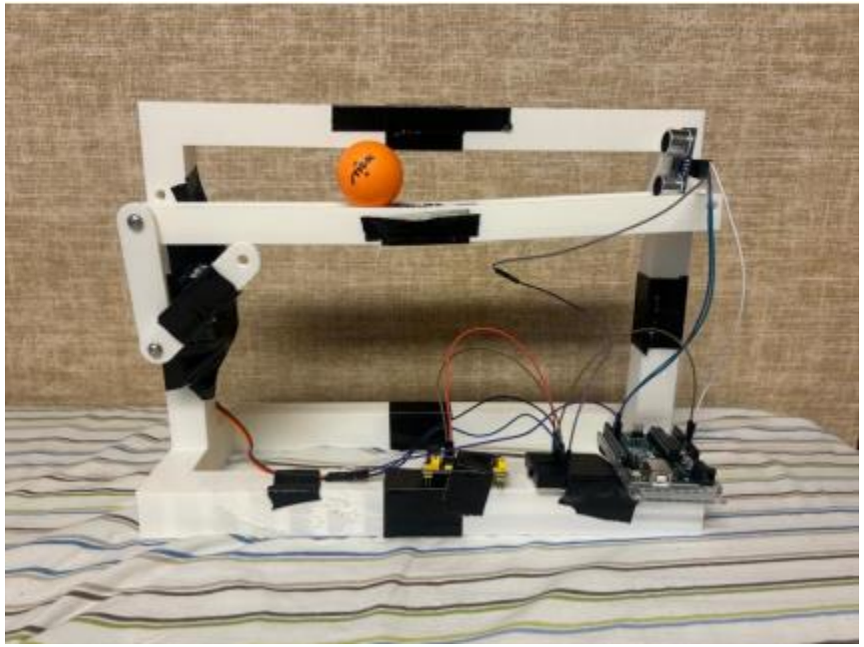
- This report focuses on the modeling, simulation, and dynamics of a "ball and beam" system.
- The ball and beam system is a classic example used in engineering education to illustrate principles of dynamics, control theory, and system modeling.
- This system was ultimately chosen for its relevance to the course material and the unique challenges it presents.

Relevance and Concepts

- This project was selected because it provides an excellent opportunity to apply key course concepts, such as:
  - Free body diagrams
  - Equations of motion
  - Dynamic system modeling
- By designing and implementing a system capable of stabilizing a ball on a beam, essential principles like feedback loops and stability analysis are explored in depth.

Objective

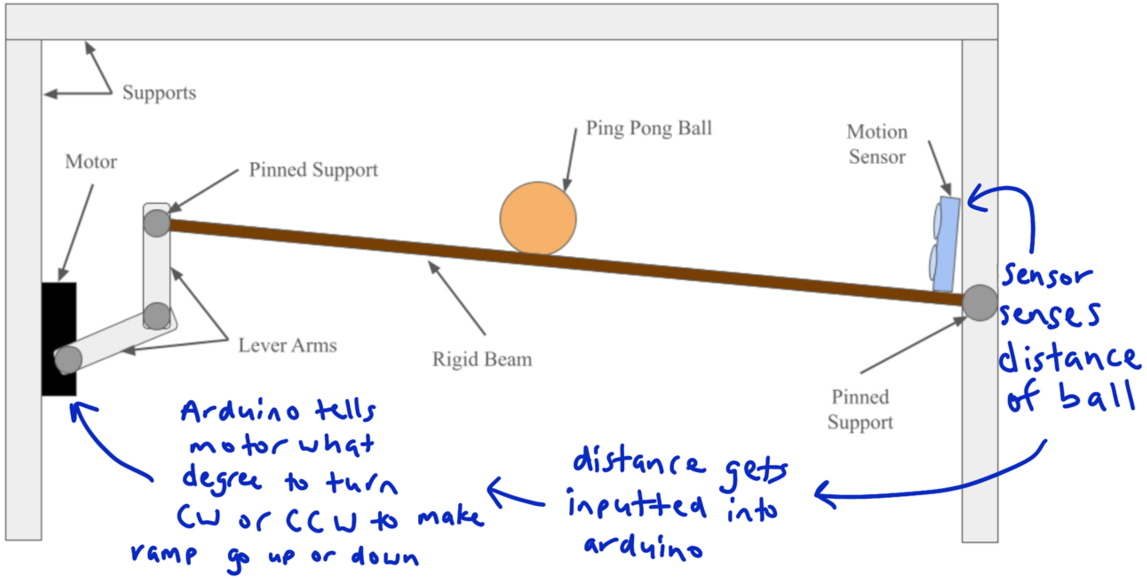
- The primary goal of the system is to balance a ping pong ball at the center of a beam, regardless of its initial position, using feedback control.



Project Description

The project involves a ball placed on a rotating beam, granting the ball one degree of freedom to roll along its length.

- **Mechanism:** One end of the beam is fixed, while the other is connected to a system of lever arms. These arms are driven by a servo motor that adjusts the beam's tilt angle, θ, through the rotation of a servo motor gear.
- **Physics:** Tilting the beam creates a slope, causing the ball to roll along the beam under the influence of gravity, with its motion depending on the tilt angle.
- **Objective:** The goal is to design and implement a control system that adjusts θ to precisely control the ball's position along the beam.



Project Concept

Implementation Plan

1. Mechanical Setup

- **Fabrication:**
  - Construct the beam, backboard, and motor arms with precise dimensions for stability and functionality.
  - Ensure proper alignment of the beam to allow smooth rolling of the ball.
- **Hardware Integration:**
  - Mount the servo motor securely to drive the lever arms.
  - Attach sensors (e.g., position or distance sensors) at appropriate locations to measure the ball's position effectively.

2. Electrical & Hardware Configuration

- **Connections:**
  - Connect the servo motor and sensors to the Arduino Uno, ensuring proper wiring and power supply.
  - Validate all electrical connections to avoid issues during debugging.
- **Sensor Calibration:**
  - Configure sensors to accurately detect the ball's position on the beam.
  - Test sensor responsiveness and output consistency.

3. System Control Development

- **Controller Implementation:**
  - Write Arduino code to:
    - Process sensor input.
    - Calculate the required servo angle using a control algorithm (e.g., PID).
    - Adjust the motor to tilt the beam and balance the ball at the desired position.
- **Testing and Debugging:**
  - Test the motor control code to ensure smooth and responsive operation.
  - Debug integration issues between sensors, motor, and Arduino.

4. Simulation and Modeling

- **Initial Simulation:**
  - Develop a Simulink model to simulate the system's dynamics, validating the feasibility of the control strategy.
- **Final Simulation:**
  - Refine the Simulink model to include accurate system parameters (e.g., beam dimensions, ball mass, motor characteristics).
  - Simulate the ball's motion in response to different control inputs, visualizing its position over time.
  - Compare simulation results with experimental data to validate the model.

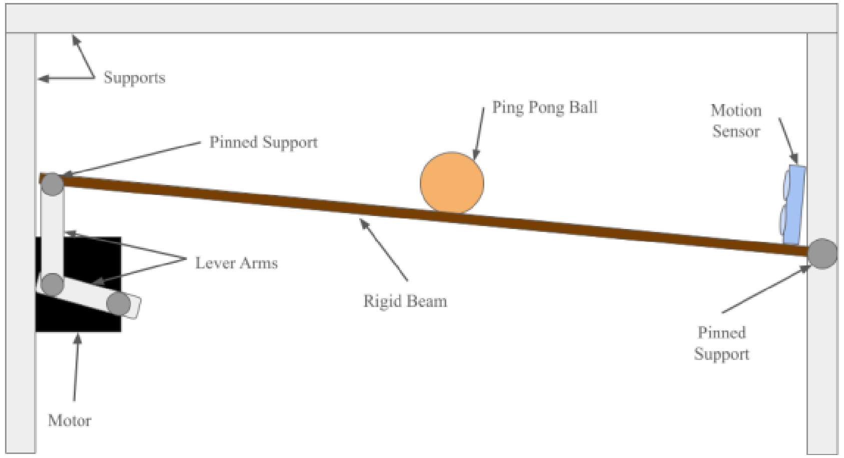
5. Final Integration and Optimization

- **System Integration:**
  - Combine mechanical, electrical, and control systems into a cohesive setup.
  - Test the integrated system to ensure it performs as expected.
- **Optimization:**
  - Fine-tune the control parameters (e.g., PID gains) to achieve smooth, precise ball balancing.
  - Address any inconsistencies between the simulation and physical system.

Mechanical Design

System Overview

- **Operation**: The system functions in the x and y plane, with no movement along the z-axis.
- **Key Components**:
  - **Rigid Beam**: 34 cm long, mounted at one end to a pinned support.
  - **Lever Arms**: 10 cm long, transferring motion from the motor to the beam.
  - **Servo Motor**: 20kg.cm capacity, adjusts beam angle based on ball position.
- **Assembly**: Components were designed in CAD and fabricated via 3D printing. Secured with nuts and bolts for key joints, and super glue/duct tape for additional support.



Mechanical Operation

- The servo motor rotates the horizontal lever arm, which tilts the beam to control the ball's position.
- The ball rolls under gravity as the beam tilts, with the motor providing precise control.
- The beam is supported by two pinned supports, allowing for smooth rotation with minimal friction.

Integration and Design

- **Precision**: 3D-printed components ensure consistent dimensions and robust assembly.
- **Stability**: The system design prioritizes stable connections to minimize errors during operation.

Electrical/Electronic System Design

Components

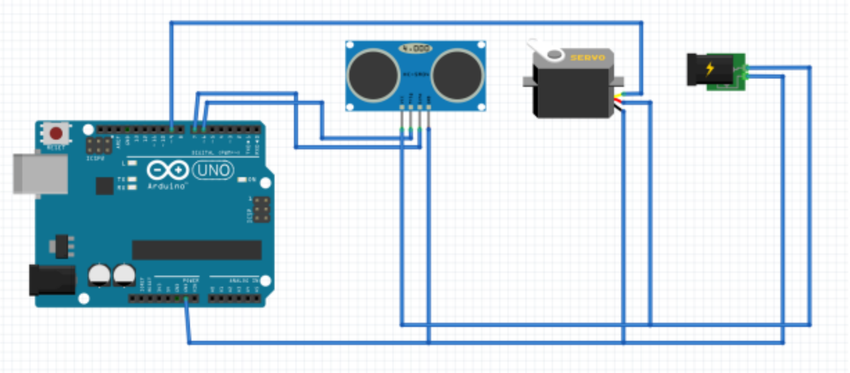
- **5V Power Module**: Provides power to the servo motor and ultrasonic sensor, powered by an AC adapter.
- **Ultrasonic Sensor**: Measures the ball's position along the beam by detecting distance.
- **Servo Motor**: Adjusts the tilt of the beam based on input from the Arduino.
- **Arduino Uno**: Microcontroller that processes sensor data and controls the servo motor via PWM signals.

System Operation

- The ultrasonic sensor sends real-time positional data to the Arduino.
- The Arduino processes the data using a PID controller to calculate the motor's required tilt angle.
- The servo motor adjusts the beam's angle accordingly, maintaining the ball's balance at the center.

Electrical Integration

- All components share a common ground to ensure stable operation.
- The system operates with continuous feedback, adjusting motor angles dynamically to keep the ball balanced.



Mathematical Model

General Translational Motion Equations

- $m\ddot{x} = \sum F$ - General translational equation of motion.
- $m\ddot{x} = F_G - F_f$ - Substitute the gravitational force and frictional force into right hand side of equation

Gravitational Force Derivation

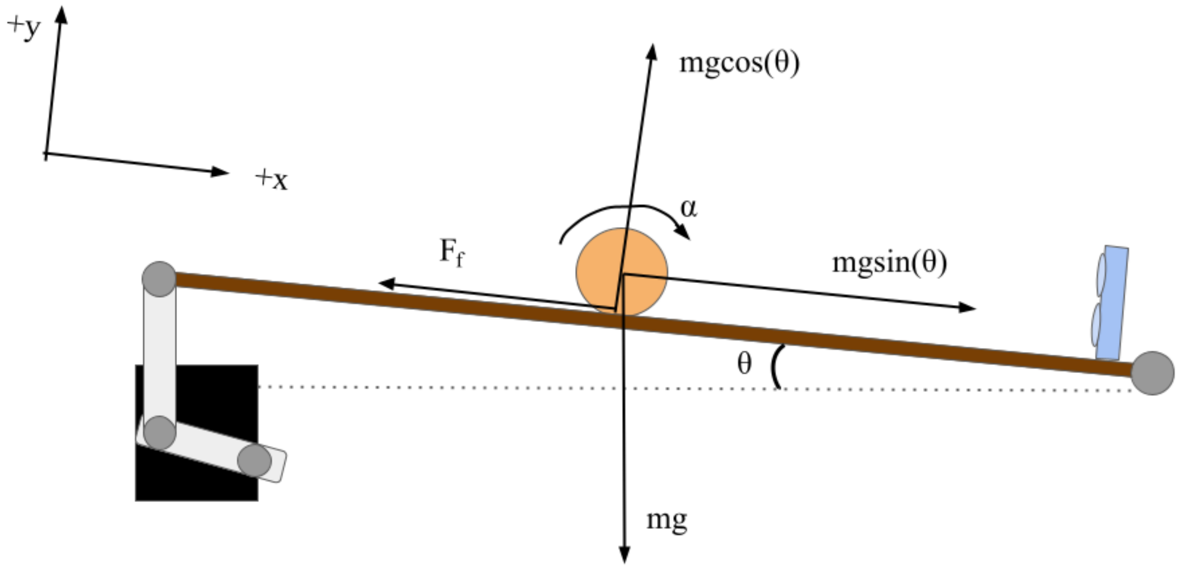
- $mgsin(\Theta)$ - Force due to gravity.
- $\sin(\Theta) = \frac{h}{L}$ - Replace sin(Θ) with known quantities.
- $\frac{h}{L} = \frac{kx}{L}$ - Relate h to x.

Frictional Force Derivation

- $T = F_f R$ - Relate frictional force to torque.
- $F_f = \frac{T}{R}$ - Rearrange torque expression.
- $I\ddot{\alpha} = \sum T$ - General rotational equation of motion.
- $I\ddot{\alpha} = \mathrm{T}$ - Simplify torque in terms of rotational acceleration.
- $\alpha = \frac{x}{R}$ - Relate rotational displacement to linear displacement.

Substitutions

- $F_f = \frac{I\ddot{x}}{R^2}$ - Substitute expressions into the frictional force formula.
- $m\ddot{x} = \frac{mgk}{L} - \frac{I\ddot{x}}{R^2}$ - Substitute gravitational force and frictional force.
- $\ddot{x} = (\frac{mgk/L}{m+I/R^2})x$ - Rearrange for final equation of motion.



Mathematical Model

Simulink Simulation

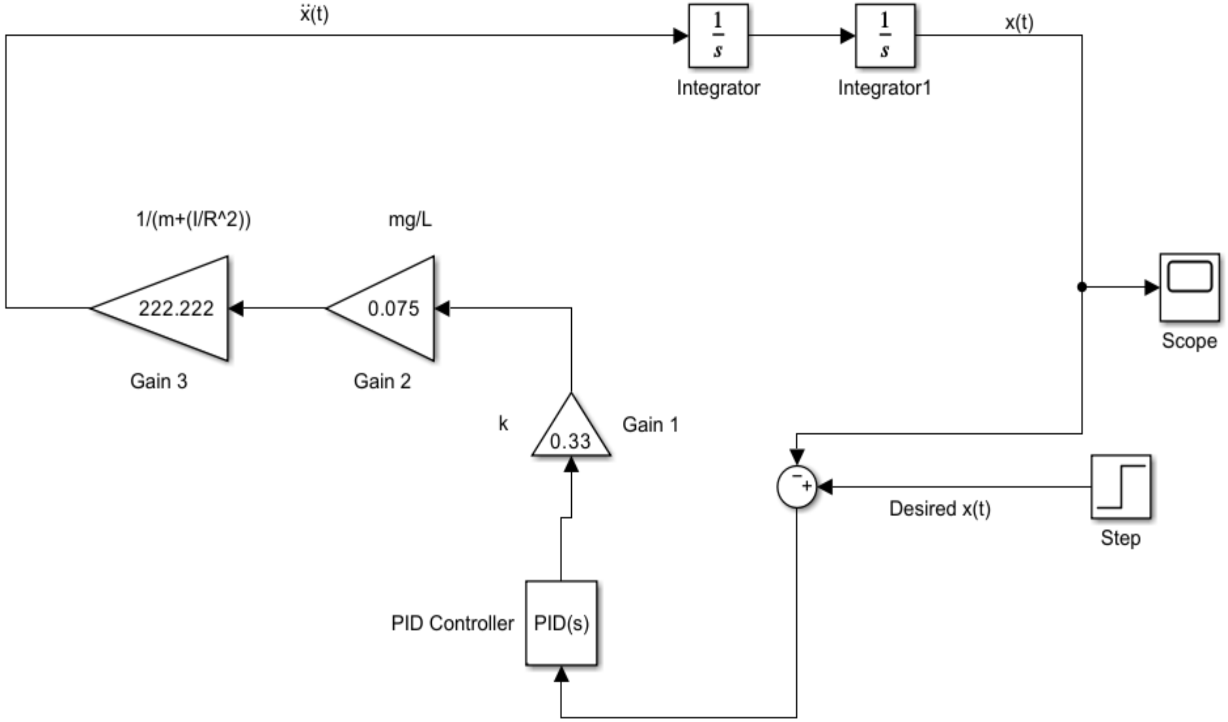


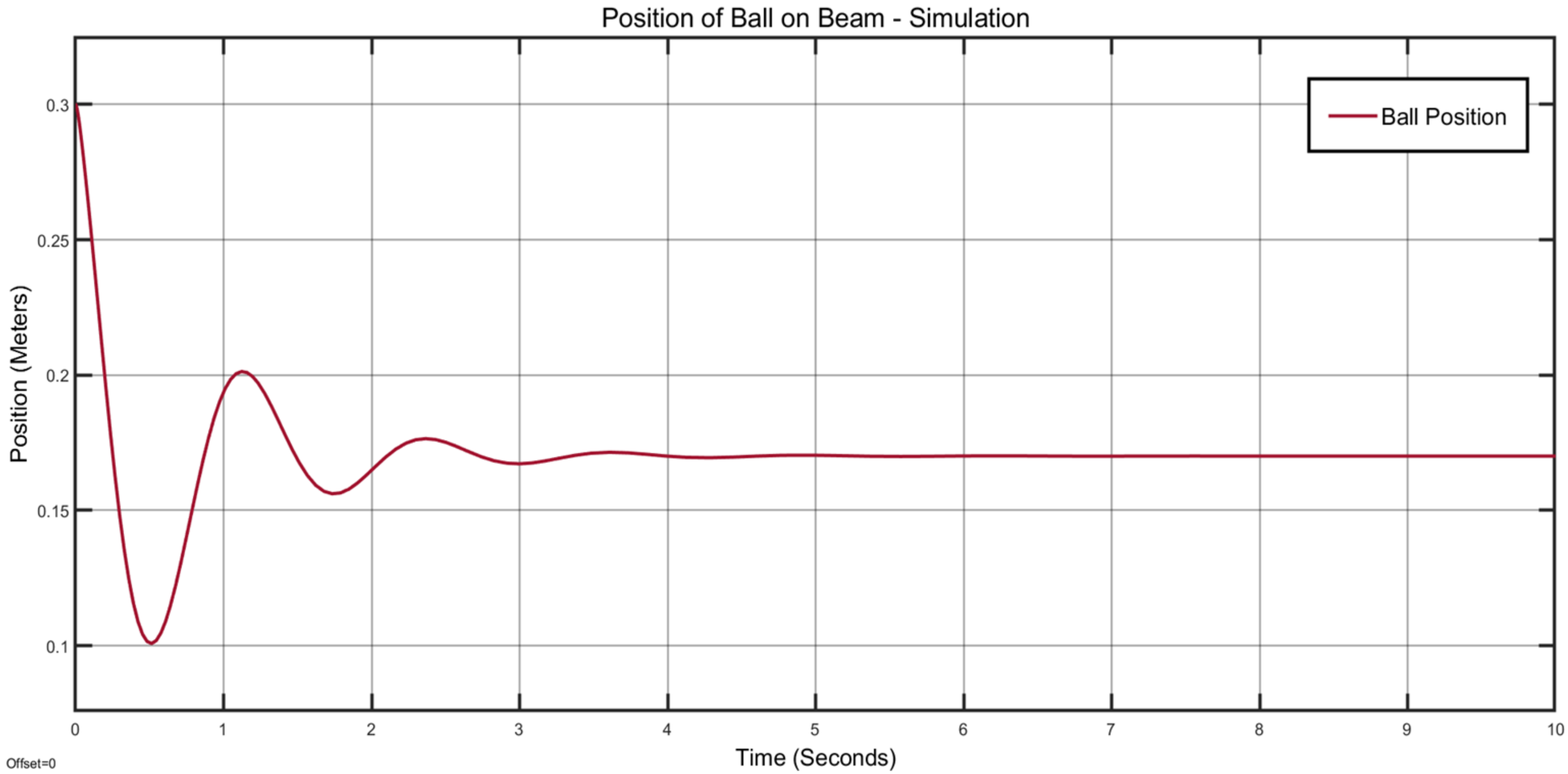
Table 1: Values for Simulink Model

Mass (m)	0.0027 kg
Gravitational Acceleration (g)	$9.8\ \frac{m}{s^2}$
k	0.33
Length of Beam (L)	0.3528 m
Radius of Ball (R)	0.020 m
Moment of Inertia (I)	$7.2 * 10^{-7}\ kgm^2$
Initial Position	0.3 m
Step Value	0.17 m
P	5
I	1
D	0.5

Simulation Results

Simulation Results

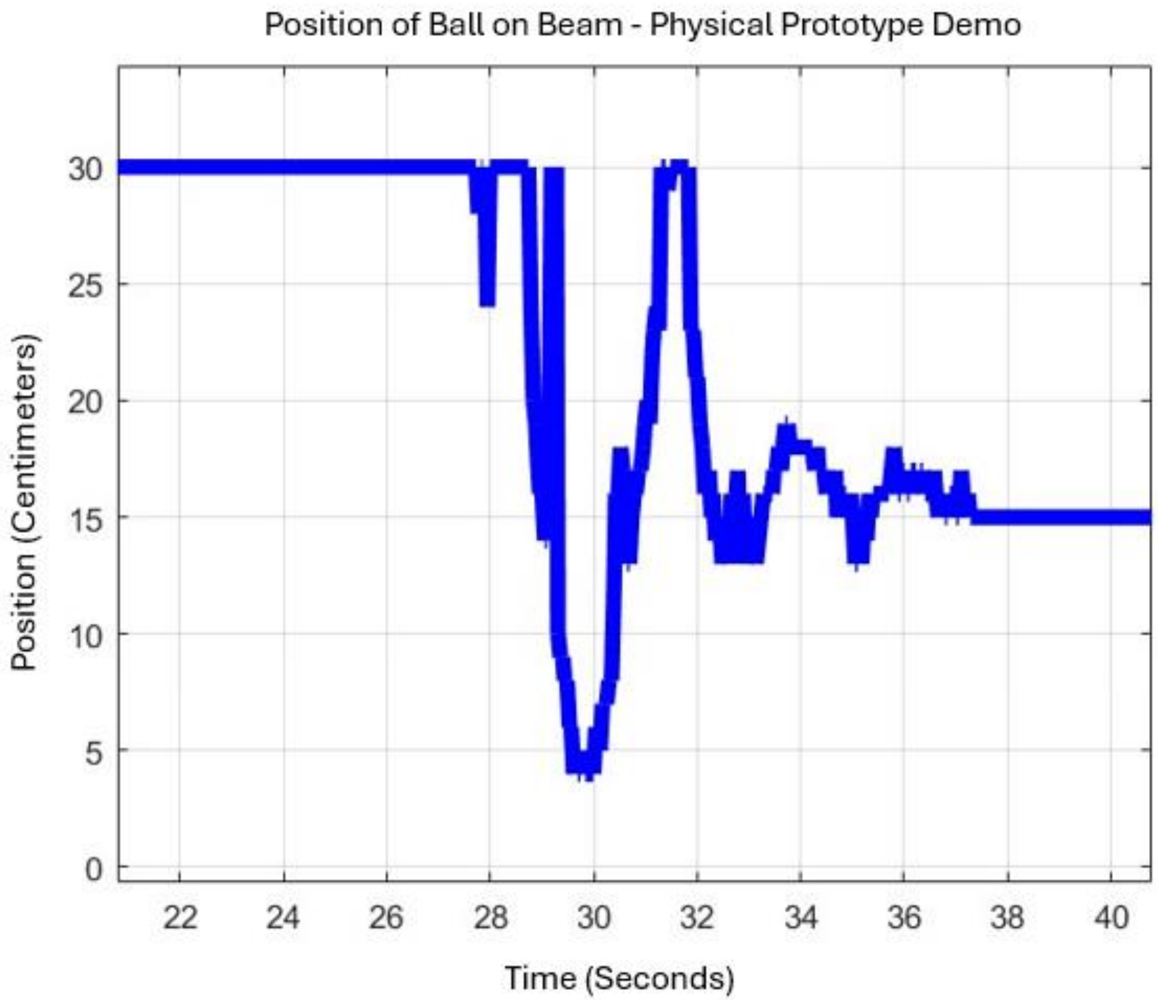
- Initial position is set to 0.3 meters or 30 centimeters.
- Oscillations begin to dampen due to the PID controller.
- System reaches steady state of 0.17 meters or 17 centimeters after around 4-5 seconds.
- PID values contribute to the behavior, stability, and low steady state error of the system.



Experimental Results

Experimental Results

- Initial position is set to 0.3 meters or 30 centimeters.
- Oscillations begin to dampen due to the PID controller.
- System reaches steady state of 0.15 meters or 15 centimeters after around 4-5 seconds.
- While there remain distinct differences between the two plots the similarities most definitely outweigh the differences.
- The general behavior of both systems is relatively similar having the initial position beginning at 0.3 meters (30 centimeters) then having a steep initial drop, a steep incline, one more steep drop, gradual incline, gradual drop, small incline, small drop, before both leveling out near 0.15-0.17 meters (15-17 centimeters).
- The accuracy between the simulation to the physical system cannot be denied but there is absolutely still room for improvement.



Experiment Video:

Experiment Video:

https://youtu.be/5NaJbI3RjyU

Conclusion

- **Project Goals:**
  - Modeled and stabilized a ball-and-beam system using theoretical and practical methods.
  - Explored the dynamics of translational and rotational motion, including friction and gravity effects.
- **Key Challenges:**
  - Balancing rotational and translational dynamics in the equations of motion.
  - Ensuring the model's accuracy with real-world variables like material imperfections and friction.
  - Implementing feedback in Simulink and fine-tuning Arduino code for hardware stability.
- **Solutions:**
  - Refined the model iteratively through testing and validation.
  - Applied PID control and adjusted simulation parameters for realistic feedback.
  - Addressed sensor inaccuracies and ensured reliable controller-motor communication.
- **Future Enhancements:**
  - Add non-linearities (e.g., variable friction, beam deflection).
  - Use alternative materials for better stability.
  - Explore advanced feedback systems like adaptive or non-linear controllers.
- **Lessons Learned:**
  - Importance of integrating analysis, modeling, and experimentation.
  - Gained skills in adaptability, problem-solving, and control system design.
  - Insights applicable to robotics, vehicle dynamics, and precise control systems.

References

J. Apkarian, M. Levis, H. Gurocak. *Ball and Beam Experiment for MATLAB/Simulink Users.* (2015). Accessed: Nov. 30, 2024. [Online]. Available: https://www.quanser.com/products/ball-and-beam/

RealPars, *PID Controller Explained.* (Dec. 20, 2021). Accessed: Nov. 24, 2024. [Online Video]. Available: https://www.youtube.com/watch?v=fv6dLTEvl74

Nikolai, K. *Simulink Control Systems and PID, Matlab R2020b.* (Jan. 21, 2021). Accessed: Nov. 24, 2024. [Online Video]. Available: https://www.youtube.com/watch?v=PRFCBVTFy90&t=1210s

Distribution of Work

Name	Assignments
Kyle Rex	1. Purchase all materials required for prototype: Motor, Arduino, Sensor, 5V Converter, Joints, and Cardboard.
	2. Create the physical initial prototype to be ready for testing: Build ramp, backboard, and motor arms. Setup motor and sensors. Setup Arduino Uno connections.
	3. Create MATLAB code to model the distance of the ball throughout the physical demonstration to compare with the output of the Simulink simulation.
Vedansh Shah	1. Create Arduino code for prototype: Create Arduino code to control motor to balance ball.
	2. Configure all parts appropriately to test and make the initial prototype work: Debug, connect, and complete system.
	3. Improve initial prototype into a final prototype by improving design where required.
Matthew Trevino	1. Create an initial Simulink simulation to make sure modeling it is feasible.
	2. Create final Simulink simulation modeling the complete system.
	3. Create an output that shows the change in distance of the ball throughout the simulation trying to get it to match as closely as possible to the physical demonstration result.

Timeline

Task #	TASK TITLE	TASK OWNER	START DATE	DUE DATE	Week 1					WEEK 2							WEEK 3						
					T	W	R	F	S	S	M	T	W	R	F	S	S	M	T	W	R	F	S
1	Simulation				9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27
1.1	Create an initial Simulink simulation	Matthew Trevino	11/9	11/13	X	X	X	X	X														
1.2	Improve the Simulink simulation	Matthew Trevino	11/13	11/18						X	X	X	X	X									
1.3	Complete the Simulink simulation completely	Matthew Trevino	11/18	11/22											X	X	X	X					
2	Physical Model Design and Assembly																						
2.1	Purchase materials required for the physical model	Kyle Rex	11/5	11/10	X	X																	
2.2	Assemble the physical model, and make CAD for arms, beam, and stand.	Kyle, Vedansh	11/10	11/20			X	X	X	X	X	X	X	X	X	X							
2.3	Create MATLAB code to record the position of the ball over time	Kyle Rex	11/20	11/22													X	X					
3	Data collection and demonstration																						
3.1	Create the code that controls the motor based on a measured position and fine tune code.	Vedansh, Kyle	11/9	11/20	X	X	X	X	X	X	X	X	X	X	X	X							
3.2	Collect data from both the Simulink simulation and the physical model	Vedansh Shah	11/20	11/22													X	X					
3.3	Work on project report and make sure project is ready for presentation	Kyle, Matthew, Vedansh	11/22	12/3															X	X	X	X	X

Thank You