MXET 375
Applied Dynamic Systems

Multidisciplinary
Engineering Technology
COLLEGE OF ENGINEERING

FINAL PROJECT

Ball and Beam

Submission Date: 12/01/2024

Names: Vedansh Shah, Kyle Rex, Matthew Trevino
Section: 906
UIN: 933000469, 932008894, 332005754

Introduction

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. 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 project was selected because it provides an excellent opportunity to apply key course concepts, such as free body diagrams, equations of motion, and dynamic system modeling, in a practical and engaging way. 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. This system was ultimately chosen for its relevance to the course material and the unique challenges it presents.

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. A diagram of the system can be seen in Figure 1.

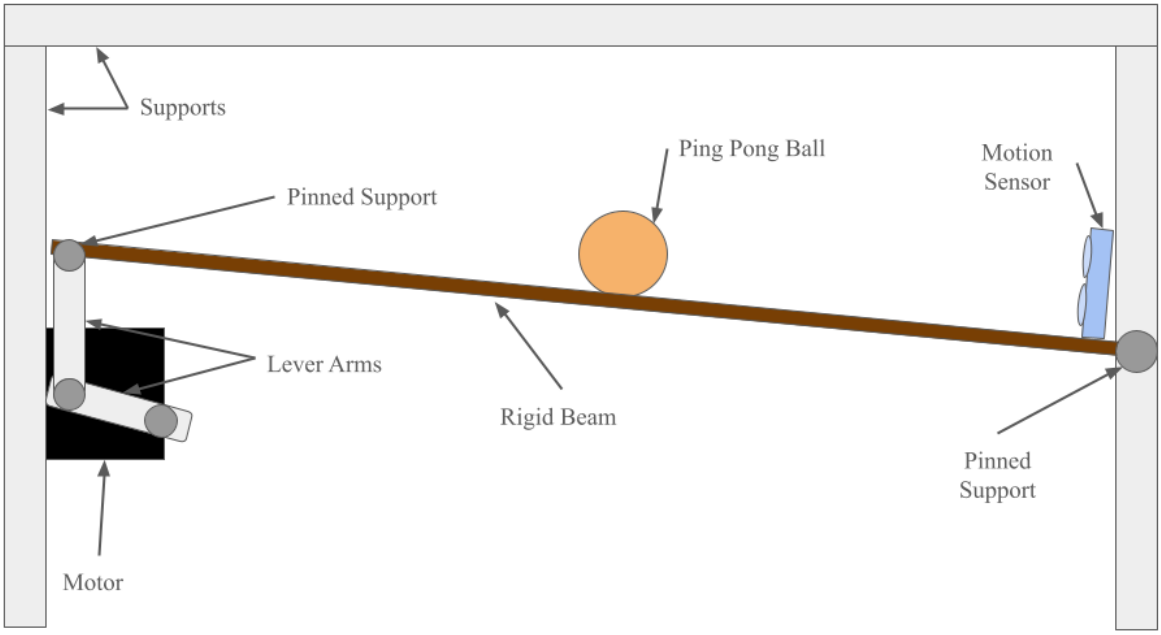

Figure 1: Diagram of the ball and beam system

As shown in the diagram above, the system operates exclusively in the x and y plane, with no motion along the z-axis. One end of the beam is fixed to a support using a 1/4" - 20 x 2" bolt secured with two nuts, allowing free rotation with negligible friction. The other end of the beam is supported by a vertical lever arm, which is connected to a horizontal lever arm. These lever arms

are joined together and to the beam using a #10 - 1" bolt with a nut and several washers acting as spacers, enabling free rotation with minimal friction.

The horizontal lever arm is connected to a DC motor, which rotates to a specific angle based on the ball's position. This rotation transfers through the lever arm assembly, causing the beam to rotate about its fixed end. To determine the motor's required rotation, a program measures the ball's position and commands the motor accordingly. An ultrasonic sensor mounted at the fixed end of the beam tracks the ball's position, rotating with the beam to maintain accurate readings as the ball moves along its length. Based on the sensor readings, a proportional-integral-derivative (PID) controller adjusts the motor's rotation to maintain balance.

The program, implemented on an Arduino, uses a PID algorithm to process the ball's position and compute the necessary motor angle. The sensor reads the ball's position, converts it into a distance value in centimeters, and feeds this value into the PID controller. The controller then calculates the adjustment needed for the servo motor, which is mounted to the beam at a baseline horizontal position of 60°. Adjustments within a range of ±80° are applied to the motor, and the system continues operating until the ball is balanced at the beam's center.

Several systems were considered for this project, but the ball and beam design was ultimately chosen for its relevance to course material and the unique challenges it presented. Concepts from MXET 375, particularly modeling translational and rotational systems, are integral to this system’s operation. The system integrates translational motion (the linear motion of the ball) and rotational motion (the angular rotation of the beam), requiring simultaneous analysis of both. While these topics were covered in the course, the project offered an opportunity for deeper exploration.

Another deciding factor was the inclusion of a negative feedback loop, which added complexity to the system. Feedback control is a topic covered in other courses but not explicitly within the context of dynamic systems modeling. This project required external research to effectively implement and simulate feedback dynamics, making it a valuable learning opportunity. Overall, the ball and beam system combines course concepts with additional challenges, which is why it was selected.

This project’s reliance on translational and rotational modeling further supports its relevance. Since the objective is to balance a ball at the beam's center, translational motion forms the foundation for deriving the equations of motion. Additionally, the motion sensor measures the ball’s linear position, emphasizing the importance of translational modeling. The interplay between translational and rotational motion will be described in greater detail in the next section.

The project aims to balance the ball at the beam’s center, regardless of its initial position, within ten seconds. Once this goal is achieved, experimental data from the physical system will be collected and compared to results from a Simulink simulation. Simulation parameters will be

adjusted to align the results of the simulation with the physical system, ultimately resulting in an accurate and reliable model.

Background and Dynamics Modeling

To begin modeling the ball and beam system, a free body diagram of the system was constructed. This diagram can be seen in Figure 2.

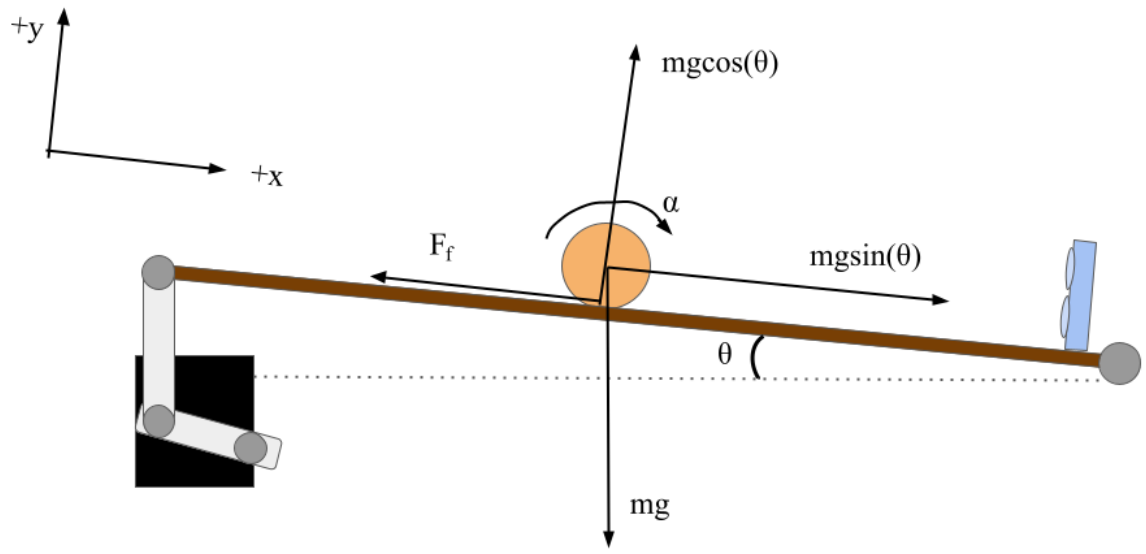

Figure 2: Free body diagram of the ball and beam system

The free body diagram in Figure 2 illustrates the forces acting on the ball at any given angle, denoted as θ. When the ball moves along the beam, two primary forces are always at play. The first is the gravitational force, represented by the vector pointing vertically downward and labeled as 'mg.' This force is defined by the product of the ball's mass (measured in kilograms) and the gravitational constant (measured in meters per second squared).

The diagram also shows two additional force components derived from the gravitational force. These components arise because of the need to redefine the standard x-y coordinate system to simplify the analysis. Under the traditional x-y coordinate system, the gravitational force vector maintains its magnitude and direction regardless of the beam's angle. However, this leads to two-dimensional motion for the ball, unnecessarily complicating the system's analysis. To address this, the coordinate system was redefined, as shown in Figure 2, to align one axis with the inclined plane. Under this new system, the motion of the ball becomes one-dimensional, and the gravitational force must be split into two components: an x-component, mgsin(θ), parallel to the beam, and a y-component, mgcos(θ), perpendicular to the beam. Since the ball does not move in the y-direction, only the x-component, mgsin(θ), contributes to the ball's motion.

The ball experiences two types of motion: translational and rotational. Translational motion refers to the ball's movement along a straight path, while rotational motion involves the ball rolling

about its center of mass. The gravitational force described earlier acts at the ball's center of mass, contributing to its translational motion. Additionally, frictional force, labeled as Ff, influences both translational and rotational motion.

The translational contribution of the frictional force arises from the interaction between the ball's surface and the beam. This interaction generates resistance opposing the ball's motion. In the free body diagram, gravity causes the ball to move in the +x direction, while the frictional force acts in the -x direction, resisting motion. Together, these forces govern the ball's translational movement.

For rotational motion, frictional force plays a crucial role. As the ball rolls down the incline, a torque is generated due to the application of the frictional force at the surface of the ball, some distance from its center (equal to the ball's radius). Although the gravitational force acts at the center of mass and does not produce torque, the frictional force applied at the surface generates a torque, calculated as the cross product of the frictional force and the ball's radius. This torque is responsible for the ball's rotational motion.

With the forces identified, the equation of motion for the system can now be derived. First define the summation of forces on the ball which can be seen in Equation 1 where m is the mass of the ball and x is the position of the ball.

$$m\ddot{x} = \Sigma F_x \qquad \textbf{Equation 1}$$

Next, the two forces acting on the ball were substituted in the right hand side of Equation 1. This addition can be seen in Equation 2 where F_G is the force due to gravity and F_f is the force due to friction.

$$m\ddot{x} = F_G - F_f \qquad \textbf{Equation 2}$$

Once the forces were identified, they could be replaced with known variables. As was explained previously, the force due to gravity is equivalent to mgsin(θ). However, the angle of the beam is not being measured throughout the experiment. Thus, sin(θ) can be replaced by its trigonometric equivalent. This equation can be seen in Equation 3 where h is the height of the beam and L is the length of the beam.

$$sin(\theta) = \frac{h}{L} \qquad \textbf{Equation 3}$$

To simplify the model, h can be replaced by the position of the ball, x, multiplied by a constant, k. This change can be seen in Equation 4 where k is a constant, x is the position of the ball, and h is the height of the beam.

$$\frac{h}{L} = \frac{kx}{L} \qquad \textbf{Equation 4}$$

The force due to friction, however, must be manipulated in terms of known variables before it can be substituted into Equation 2. Based on the description of the free body diagram, it was determined that a force applied some distance from the axis of rotation can generate a torque. This equation can be seen in Equation 5 where τ is the torque, F_f is the force due to friction, and R is the radius of the ball.

$$\tau = F_f R \qquad \textbf{Equation 5}$$

Equation 5 can be rearranged to isolate the friction force as seen in Equation 6.

$$F_f = \frac{\tau}{R} \qquad \textbf{Equation 6}$$

Equation 6 isolates the frictional force, but the torque is still an unknown quantity. This problem can be solved by analyzing the rotational motion of the ball. The general equation for rotational motion can be seen in Equation 7 where I is the moment of inertia of the ball, α is the angular displacement of the ball, and τ is the torque.

$$I\ddot{\alpha} = \Sigma\tau \qquad \textbf{Equation 7}$$

Since there is only one torque applied to the ball, the torque in Equation 6 can be defined as seen in Equation 8.

$$I\ddot{\alpha} = \tau \qquad \textbf{Equation 8}$$

To eliminate all unknown variables, the angular displacement of the ball can be defined in terms of its linear displacement. This can be seen in Equation 9 where α is the angular displacement of the ball, x is the linear displacement of the ball, and R is the radius of the ball.

$$\alpha = \frac{x}{R} \qquad \textbf{Equation 9}$$

At this point, all unknown variables have been defined in terms of known quantities. Equation 9 and Equation 8 can be combined with Equation 6 to yield the frictional force equation that can be seen in Equation 10.

$$F_f = \frac{I\ddot{x}}{R^2} \qquad \textbf{Equation 10}$$

Equation 10 can then be substituted into Equation 2 to yield the equation of motion that can be seen in Equation 11.

$$m\ddot{x} = \frac{mgk}{L} - \frac{I\ddot{x}}{R^2} \qquad \textbf{Equation 11}$$

Equation 11 can then be rearranged to yield the final equation of motion that can be seen in Equation 12.

$$\ddot{x} = \left(\frac{\frac{mgk}{L}}{m+\frac{I}{R^2}}\right)x \qquad \textbf{Equation 12}$$

Equation 12 represents the final equation of motion that is used to build a Simulink simulation model of the ball and beam system.

Simulations and Experimental Setup

To simulate the system, the differential equation shown in Equation 12 needed to be constructed in Simulink. The final Simulink block diagram modeled using this equation can be seen in Figure 3.

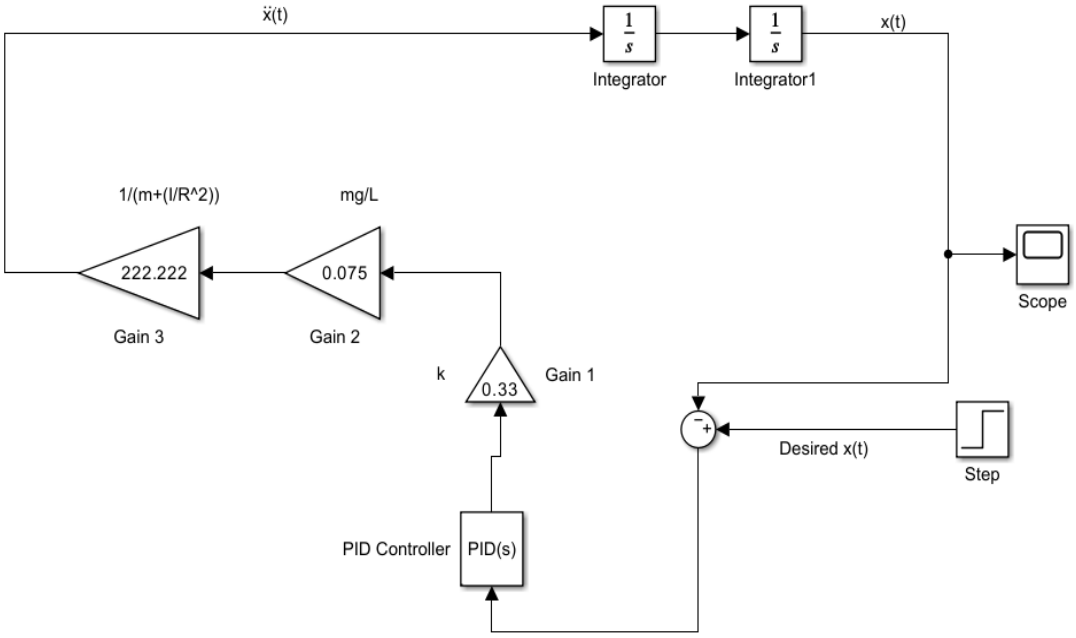

Figure 3: Simulink block diagram

To begin constructing the block diagram, a starting point was defined as $\ddot{x}$. This point is the line in the top left corner of Figure 3. In order to make the necessary comparison, the x variable needed to be isolated, so two integrator blocks were inserted following the line defined as $\ddot{x}$. At this point in the block diagram, the measured x position needed to be compared with the desired x position. This was done by subtracting a step block from the measured x position. The step block is set to 17 cm or 0.17 m, since this is the center of the beam, and thus the desired position of the ball. Based on the difference between x and the step block, a PID controller block would calculate the error and output the necessary correction to the system. To generate the desired behavior of the system, P was set to 5, I was set to 1, and D was set to 0.5. Next, the corrected x position needed to be multiplied by the constants in Equation 12. This was done by grouping the constant values into three different groups, each of which was represented by a gain block. The first gain block following the PID block represents the constant k. The value assigned as the constant k was 0.33. As explained in the previous section, the model is simplified under the assumption that h=kx. It was determined that for any given x position, h is roughly one-third of x. This is why k is set to 0.33. The second gain block represents $\frac{mg}{L}$. The value shown in Figure 3**,** 0.075, was calculated by inserting the mass of the ball, gravitational acceleration, and length of the beam into the

expression $\frac{mg}{L}$. A similar method was used in the third gain block. The mass, moment of inertia, and radius of the ball was inserted into the expression, $m + \frac{I}{R^2}$. Since this expression is in the denominator of Equation 12, the value in the previous expression was multiplied to the power of negative 1, yielding the value 222.222 shown in the third gain block. At this point, all variables and constants were accounted for, resulting in the completion of the cycle. The constants used in the Simulink model can be seen in Table 1.

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

Once the Simulink model was built, a few adjustments were made before the simulation was run. To begin, an initial position of 0.3 m was set to the second integral block. The solver was then changed to "ode23t" in the model settings. At this point, the simulation was run, and the resulting motion of the ball was plotted in the scope block. The results of this experiment are shown and analyzed in the next section.

The simulation and experimental setups were developed independently, with the goal of later comparing their results to validate the system's performance. This approach ensured flexibility in design while allowing for direct evaluation of the similarities and differences between the theoretical model and the physical prototype.

The second part of the project involved creating a physical prototype to get experimental data from and compare it to the simulation. The prototype is designed to balance a ping pong ball on a tilting beam by controlling the beam's angle via a servo motor, as shown in Figure 4. The

system consists of three primary components: the mechanical structure, the electrical setup, and the control system.

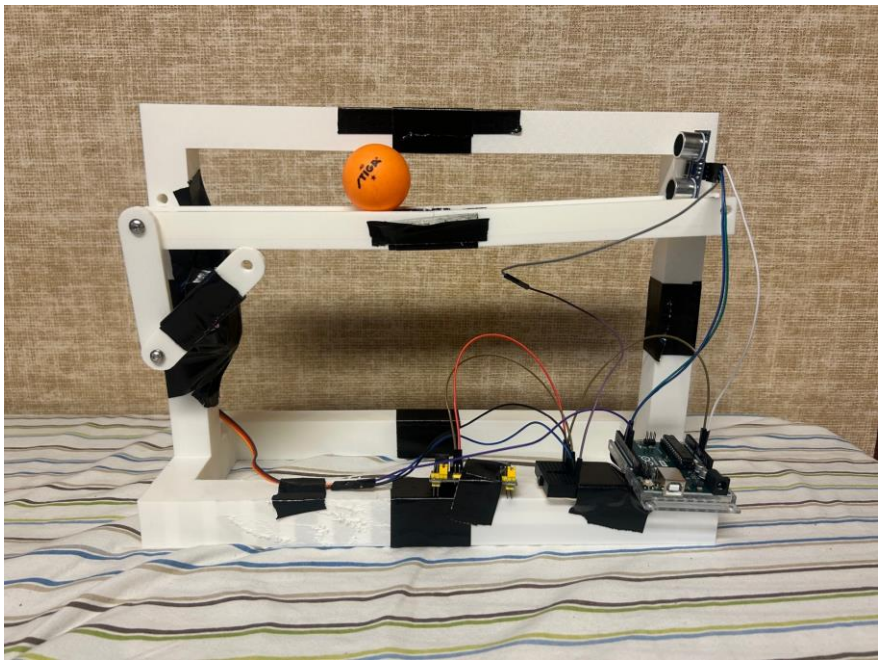

Figure 4. Fully Assembled Ball-and-Beam Prototype Setup

The mechanical structure includes a rigid beam, lever arms, and a support stand, all of which were designed using CAD software and fabricated through 3D printing. The beam is connected to two lever arms, which are in turn driven by a 20kg.cm servo motor capable of precise angle adjustments. The rigid beam is 30cm long, lever arms are 10cm long, and the center of the motor is placed 30cm to the left and 5cm down relative to the fixed end of the beam. The servo motor tilts the beam around a fixed pivot point, allowing the ball to roll along its length under the influence of gravity. The entire structure is secured using a combination of nuts and bolts for the fixed pivot points and lever arms, while superglue and duct tape were used to hold other components in place during assembly.

The electrical system integrates a 5V power module, an ultrasonic sensor, the servo motor, and an Arduino Uno microcontroller. The 5V power module, powered by an AC adapter, provides the necessary power to the servo motor and the ultrasonic sensor. The ultrasonic sensor measures the distance of the ping pong ball from one end of the beam, providing real-time positional feedback. The sensor outputs are fed into the Arduino, which processes the data and generates appropriate PWM signals to control the servo motor’s angle. All components share a common ground, with the grounds for the power module, motor, sensor, and Arduino connected to ensure stable operation.

The control system is implemented on the Arduino, which continuously adjusts the motor’s angle to maintain the desired position of the ball on the beam. The ultrasonic sensor, positioned at one end of the beam, measures the ball's distance and sends the data to the Arduino. Based on this

feedback, the Arduino uses a control algorithm to determine the necessary tilt angle of the beam, which is applied by the servo motor. This closed-loop feedback system ensures that the ball remains balanced and responds dynamically to changes in position. This feedback system is controlled using PID logic and therefore uses the values of 2.5, 0, and 1.1 for the constants of P, I and D respectively. The code sequence also allows for the control of the steady state position of the ball relative to the sensor allowing the user to control where the ball's final position will be along the beam.

The assembly process prioritized precision and stability to minimize experimental errors. The 3D-printed components ensure consistent dimensions, while the use of nuts and bolts provides robust connections at critical points. Despite the use of superglue and duct tape for secondary attachments, the structure remains durable and functional for both simulations and experimental testing. This prototype represents a cohesive integration of mechanical design, electrical circuitry, and control logic, providing a reliable platform for analyzing and testing the performance of the ball-and-beam system.

Evaluate Experimental Results

The experimental results included plots for the position of the ball on the beam for both the Simulink simulation and the physical system itself. These results can be seen in Figure 5 and Figure 6 respectively. The x-axis for both plots is time measured in seconds and the y-axis is position measured in meters for Figure 5 and centimeters for Figure 6.

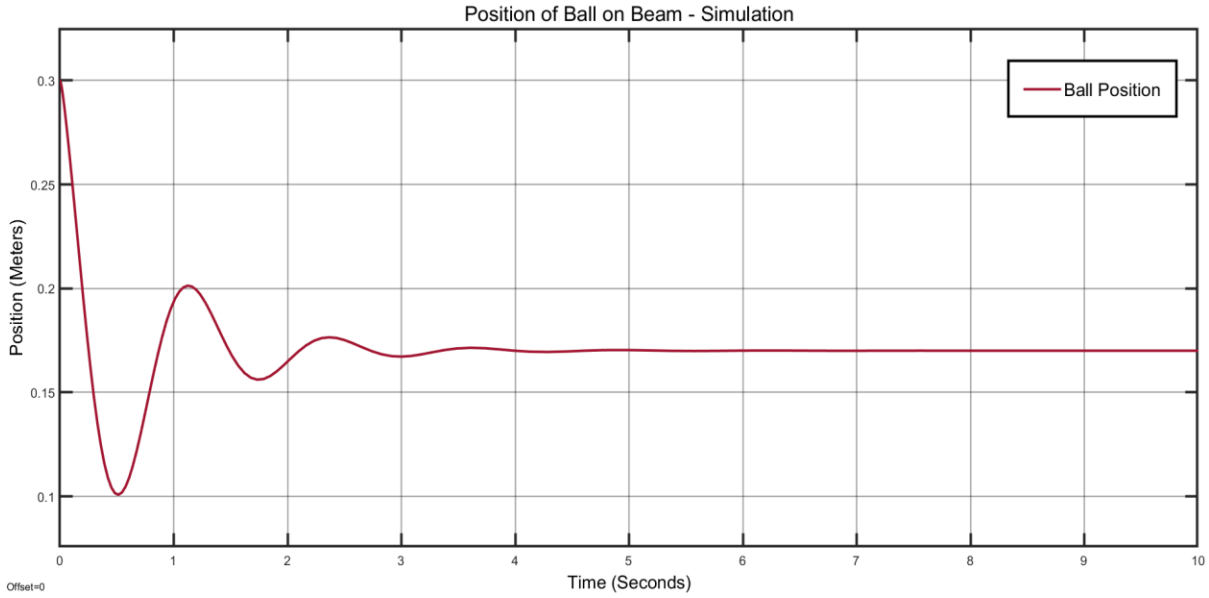

Figure 5: Results of Simulink Simulation

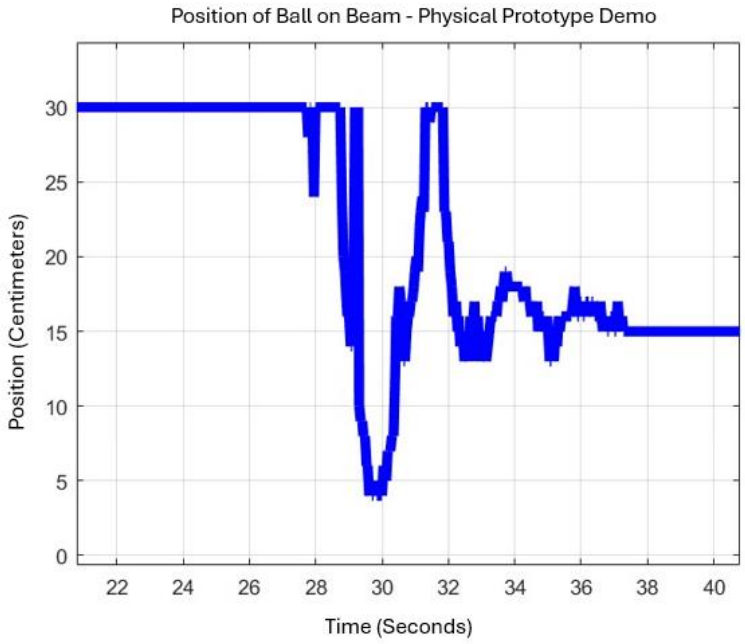

Figure 6: Results of Physical System Demonstration

Figure 5 was created using the internal plotting function in Simulink using the Scope block while Figure 6 was created using a MATLAB code sequence that created a plot measuring the distance output of the physical Arduino controlled ultrasonic sensor on the prototype.

Figure 5 represents a simulation of the ball and beam system modeled in Simulink, displaying the ball's position as a function of time. The behavior indicates a system with well-tuned control dynamics. Initially, the system is set to begin with the ball position at approximately 0.3 meters. Following this, the system demonstrates damped oscillations as the control algorithm works to stabilize the ball's position. The oscillations diminish over time, and the system settles into a steady-state position near 0.17 meters after approximately 4-5 seconds. The smooth nature of the response, characterized by minimal oscillatory behavior after stabilization, is caused by the presence of a well-designed PID continuous feedback controller which ensures stability, precision, and low steady-state error.

Figure 6 corresponds to the actual performance of the physical system under similar conditions. The behavior deviates from the simulated response, showing sharp and irregular transitions instead of smooth oscillations. Initially, the system is set to begin with the ball position at approximately 30 centimeters. When the ball is then placed on the beam, the beam begins to move in an attempt to counteract the ball’s motion and center it. This is seen by the abrupt change in the ball’s position from rolling towards the sensor, getting within about 5 centimeters, to rolling the ball back the other direction returning to almost 30 centimeters again. After this initial abrupt countermeasure, the beam begins to make smaller movements to counter the ball’s motion until returning the ball to its steady state position in the center of the beam. This can be seen by the damping oscillations in the plot that gradually stabilize the ball around a steady-state value of approximately 15 centimeters after 8-9 seconds. The sudden and sharp transitions in the plot suggest that the physical system is either subject to external noise or disturbances not accounted for in the simulation or exhibits limitations in the implementation of the control algorithm, such as latency, nonlinearity, or actuator constraints.

A comparison of the two plots highlights notable differences in the stability and control dynamics between the simulated and physical systems. The simulation demonstrates smooth transitions and effective stabilization, indicative of an idealized and well-tuned control model. In contrast, the physical system reaches stability more slowly with larger oscillations, likely due to unmodeled real-world factors such as mechanical imperfections, sensor noise, or unaccounted nonlinearities. While the simulation achieves steady-state convergence efficiently, the physical system exhibits a slower and less precise response, with much larger oscillations. These differences underscore the importance of accounting for real-world constraints and uncertainties when transitioning from simulation to physical implementation, as well as the potential need for further tuning of the controller or enhancements in system design to improve robustness.

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.

Conclusion

This project provided an in-depth exploration of the dynamics and control principles necessary to model and stabilize a ball and beam system. Through the derivation of equations of motion, the redefinition of coordinate systems, and the inclusion of key forces such as friction and gravity, a comprehensive understanding of the system's behavior was achieved. The combination of translational and rotational dynamics presented unique challenges that required a careful balance of theoretical analysis and practical considerations. Additionally, the interactive process of validating the model against real-world behavior helped ensure the system’s accuracy and effectiveness.

One of the most significant challenges faced during the project was accurately modeling the interplay between rotational and translational motion. Capturing the effects of friction, torque, and beam incline while maintaining simplicity in the equations of motion proved to be complex. Another obstacle was ensuring that the model reflected real-world behavior, as factors such as material imperfections and friction coefficients added variability to the system. Addressing these challenges required repeated refinements and a deep understanding of the fundamental concepts taught in the course.

Incorporating feedback into the Simulink model was another significant hurdle. Feedback is a critical component of control systems but, implementing it effectively within Simulink required additional research and a trial-and-error approach. Ensuring that the simulated feedback loop mimicked real-world behavior demanded a detailed understanding of PID control and fine-tuning of simulation parameters. On the hardware side, preparing the physical prototype for successful operation presented further challenges. The Arduino code controlling the prototype had to be repeatedly fine-tuned and adjusted to achieve stable performance. This process involved testing various parameters, addressing sensor inaccuracies, and ensuring smooth communication between the controller and the motor. These steps were time-intensive but essential to achieving a functional and reliable system.

In future iterations, several enhancements could be explored to improve the project. Incorporating non-linearities such as variable friction coefficients or the effects of beam deflection could create a more realistic model. Furthermore, exploring alternative materials for the ball and beam could improve stability and efficiency. Integrating advanced feedback control systems, such as adaptive or non-linear controllers, would allow for more dynamic and precise ball stabilization. These adjustments could significantly enhance both the accuracy and robustness of the system.

The lessons learned from this project are highly valuable for future engineering endeavors. By applying theoretical knowledge to solve practical problems, this project demonstrated the importance of integrating analysis, modeling, and experimentation. Addressing challenges like the interplay between translational and rotational motion, implementing feedback in Simulink, and fine-tuning the prototype underscored the importance of adaptability, creativity, and perseverance in engineering design. These skills and insights can be applied to a wide range of real-world systems, such as robotic manipulators, dynamic stability controls in vehicles, or other systems

requiring precise control and modeling. Ultimately, this project provided a solid foundation for addressing more advanced engineering challenges in the future.

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