Nonlinear Control of Ball and Beam System

***Abstract*—For this lab, we implemented both a proportional integral derivative (PID) and a linear quadratic regulator (LQR) controller to demonstrate nonlinear control techniques. The objective of the control scheme is to track a reference trajectory while operating in a safe region and minimizing tracking error and energy costs. We test our control strategies on a classic example - the ball and beam system - and provide a detailed discussion on the controllers' performance.**

I. Introduction

We investigate nonlinear control techniques for a classic control example - the ball and beam system - such that the position of the ball accurately tracks a defined trajectory. The ball and beam hardware consists of a track, on which a metal ball is free to roll, and a Rotary Servo Base unit which controls the angle of the beam [1]. By controlling this angle, the ball can be balanced at a desired position. We analyze the performance of a proportional integral derivative (PID) and linear quadratic regulator (LQR) controller to drive the ball to a reference trajectory on the beam, specifically focusing on trajectories composed of sine and square waves with adjustable periods and amplitudes. Further, we utilize a Luenberger observer in both controllers to account for uncertainty in sensor readings of the state of the ball. The main objective of the controllers is to control the voltage input to the servo motor to track the reference trajectory of the ball while minimizing the tracking error and energy spent by the motor. Additionally, the controllers must ensure they operate in a safe region (e.g. the ball does not escape the physical limits of the beam).

Organization: The system dynamics are introduced in Section II. In Section III, we present the design of our observer and controllers. The simulation results are discussed in Section IV, followed by the results of our hardware experiments in Section V.

II. System Modeling

The state of the ball and beam system is given by $x = [z, \dot{z}, \theta, \dot{\theta}]$, where z is the ball's position on the beam, $\dot{z}$ is the velocity of the ball, θ is the angle of the servo motor arm, and $\dot{\theta}$ is the angular velocity of the servo arm.

We can derive the equation of motion for the ball by examining the forces acting on the ball [1]. This consists of the gravitational and centrifugal forces, as well as the rolling resistance. The gravitational force in the x-direction, the centrifugal force, and the rolling resistance are expressed, respectively, below:

$$F_g = mg\sin\alpha \tag{1}$$

$$F_c = m(\frac{L}{2} - z)\dot{\alpha}^2 \tag{2}$$

$$F_r = \frac{J_b}{r_b^2}\ddot{z} \tag{3}$$

where m is the ball mass, L is the beam length, r_b is the ball radius, and $J_b = \frac{2}{5}mr_b^2$ is the moment of inertia of the ball [2]. This gives us the following equation of motion of the ball:

$$m\ddot{z} = mg\sin\alpha - m\left(\frac{L}{2} - z\right)\dot{\alpha}^2 - \frac{J_b}{r_b^2}\ddot{z}. \tag{4}$$

We assume that the angle of the beam is small enough such that $\alpha \approx \sin\alpha$. Further, we utilize the kinematics relation $L\sin\alpha \approx r_g\sin\theta$, where r_g is the servo arm length, to perform the following replacement:

$$\sin\alpha = \frac{r_g}{L}\sin\theta, \qquad \dot{\alpha} = \frac{r_g}{L}\cos\theta\dot{\theta}. \tag{5}$$

Finally, we take the motor dynamics as $\tau\ddot{\theta} = -\dot{\theta} + K \cdot V$, where V is the voltage applied to the motor, τ is the delay constant, and K is the motor constant. This leads us to the following ball and beam system equations, with $u = V$ as the control input:

$$\dot{x_1} = x_2 \tag{6}$$

$$\dot{x_2} = \frac{5g}{7}\frac{r_g}{L}\sin x_3 - \frac{5}{7}\left(\frac{L}{2} - x_1\right)\left(\frac{r_g}{L}\right)^2 x_4^2\cos^2 x_3 \tag{7}$$

$$\dot{x_3} = x_4 \tag{8}$$

$$\dot{x_4} = -\frac{x_4}{\tau} + \frac{K}{\tau}u. \tag{9}$$

We present a table with the system parameters below:

r_g	0.0254 m
L	0.4255 m
g	9.81 m/s^2
K	1.5 rad/sV
τ	0.025 s

III. Observer and Controller Design

We will discuss the design of our Luenberger Observer as well as the PID and LQR controllers that we utilize in controlling the ball and beam system [1].

[1] https://github.com/gagandeep25/EE222-Nonlinear-Systems-Ball-and-Beam-Project.git

A. Observer Design

In the ball and beam setup, we only have partial measurement of the full state space. We receive measurement data of the ball position z from the sensor on the Quansar hardware which is susceptible to sensor noise. Therefore, we design a Luenberger observer to provide an estimate $\hat{x}$ of the states using the plant model, knowledge of input u, and measurement of the output y.

Imagine running a simulation copy of the system model in real time with the same input $u(t)$ applied to the actual system:

$$\dot{\hat{x}} = A\hat{x}(t) + Bu(t) \tag{10}$$

$$\hat{y} = C\hat{x}(t) + Du(t) \tag{11}$$

where

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{5gr}{7L} & -\frac{5L}{14}(\frac{r}{L})^2 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & -\frac{1}{\tau} \end{bmatrix}, B = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \frac{K}{\tau} \end{bmatrix}, \tag{12}$$

(13)

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, D = \begin{bmatrix} 0 \end{bmatrix}. \tag{14}$$

We are interested in the error: $e := \hat{x} - x$ between the simulated state and the actual state [3]. We can use the discrepancy between the measured output y and the predicted output $\hat{y}$ to correct the simulation copy:

$$\dot{\hat{x}}(t) = A\hat{x}(t) + Bu(t) + L(\hat{y}(t) - y(t)) \tag{15}$$

Then the error becomes:

$$\dot{e} := \dot{\hat{x}} - \dot{x} \tag{16}$$

$$= A\hat{x}(t) + Bu(t) + L(\hat{y}(t) - y(t)) - Ax(t) \tag{17}$$

$$= A(\hat{x}(t) - x(t)) + Bu(t) + LC(\hat{x}(t) - x(t)) \tag{18}$$

$$= Ae(t) + Bu(t) + LCe(t) \tag{19}$$

$$= (A + LC)e(t) + Bu(t) \tag{20}$$

If (C, A) is observable, we can choose L such that $A+LC$ is Hurwitz. Since $M_o = [C; CA; CA^2; CA^3]$, $rank(M_o) = 4 = n$, and the system is observable. The eigenvalues of $A + LC$ will determine the rate of convergence of $e(t)$. We placed the poles of $A + LC$ at $-10, -12, -15, -18$ which resulted in an observer gain of

$$L = \begin{bmatrix} 25.1320 & 151.9416 & -1.2948 & 32.6879 \\ -0.5331 & -6.4568 & -10.1320 & 619.3871 \end{bmatrix}^\top. \tag{21}$$

B. PID Design

The ball and beam system has relatively low-order dynamics and a known, intuitive behavior. Therefore, a PID controller is well-suited for our objective of achieving stable tracking. The advantages of a PID controller include: 1) a simple, easy, and effective implementation, 2) real-time applicability, 3) direct tunability of gain adjustment, and 4) a robust framework capable of handling basic nonlinearities. In contrast, the disadvantages of a PID controller are: 1) sensitivity to measurement noise, 2) reliance on trial-and-error tuning, and 3) the potential for overshoot or instability due to the integral term. We tune the gains for the PID controller through trial and error, specifically tuning with consideration to the tracking performance, energy cost, and safety violation. We provide the parameters used for simulation and hardware experiments at the end of this section in Table I.

A PID controller consists of three terms: 1) the proportional term, 2) the integral term, and 3) the derivative term. The proportional term reacts immediately to the overall error (e, a sum of the difference between the actual ball position and the desired reference positions, along with half the difference between their velocities), which is

$$P(t) = k_p e(t), \tag{22}$$

where k_p is the gain for the proportional term. The integral term accumulates the error over time and is defined as

$$I(t) = k_i \sum_{\tau=0}^{t} e(\tau)\Delta t \tag{23}$$

where k_i is the gain for the integral term. The derivative term predicts the future error based on the rate of change. Therefore,

$$D(t) = k_d \frac{e(t) - e(t - \Delta t)}{\Delta t}, \tag{24}$$

where k_d is the gain for the derivative term. By summing the PID terms, the desired beam angle θ_d is found as follows:

$$\theta_d(t) = \max\Big(\min\Big(-P(t) - I(t) - D(t), \frac{50\pi}{180}\Big), -\frac{50\pi}{180}\Big) \tag{25}$$

Then, a PID controller determines the voltage that drives the servo toward the desired angle from the current angle $\theta(t)$ by

$$V(t) = k_{servo}(\theta_d(t) - \theta(t)) \tag{26}$$

where k_{servo} is the servo voltage gain.

The following table shows the parameters used for the PID controller for simulation and hardware testing.

TABLE I: Parameters of PID Controller

	Simulation	Hardware
k_p	7	2
k_i	0.01	0.4
k_d	8	0
k_{servo}	2	3

In addition, during hardware testing, we noticed that the angle of the beam does not move when the voltage is of small magnitude. Consequently, the tracking performance deteriorates while the energy cost continues to increase, due to non-zero voltage. Therefore, we made the following adjustment to our voltage calculations:

$$V_{servo} = \begin{cases} V(t) + 0.6, \text{if } V(t) < 0, \\ V(t) - 0.6, \text{if } V(t) > 0. \end{cases} \tag{27}$$

This adjustment helped when the ball settled to the incorrect position due to the lack of beam movement.

Further, we limited the servo voltage to be $\pm 2V$ in simulation and $\pm 1V$ for hardware experiments. We made this adjustment to the bound on the voltage for optimal performance in terms of energy cost. The angle of the beam then is adjusted based on V_{servo} such that

$$V_{servo} = \begin{cases} 1 & \text{if } V_{servo} > 1 \\ -1 & \text{if } V_{servo} < -1 \\ V_{servo} & \text{otherwise .} \end{cases} \tag{28}$$

The bounds on θ_d in 25 and V_{servo} in 28 ensure that the system doesn't violate safety restrictions.

C. LQR Design

An LQR controller is an optimal control method for linear systems, consisting of two main parts: the cost function and the controller gain. The LQR controller is guaranteed to stabilize the equilibrium, while also allowing for tradeoff between the input energy and tracking error. Further, the LQR algorithm is computationally simpler than many non-linear control strategies.

To use the LQR controller, we must first linearize the ball and beam system. Jacobian linearization is a technique to linearize our nonlinear system about an equilibrium or operating point. The resulting linearized equations can be used to control the system in a neighborhood of this point. To linearize the state equations about the point consider the model $\dot{x} = f(x, u)$. Let $\xi = (x^{eq}, u^{eq})$ be an equilibrium/operating point of this model. Then, we can approximate f in a neighborhood ξ as

$$f(x,u) \approx f(x^{eq}, u^{eq}) + \frac{\partial f}{\partial x}(\xi)(x - x^{eq}) + \frac{\partial f}{\partial u}(\xi)(u - u^{eq}).$$

If we define $\tilde{x} := x - x^{eq}$ and $\tilde{u} := u - u^{eq}$, we write our linearized model as $\dot{\tilde{x}} = \dot{x} = f(x,u) \approx A\tilde{x} + B\tilde{u}$. For our system, this results in matrices A and B (when $\xi = (0,0)$):

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{5gr_g}{7L} & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & -\frac{1}{\tau} \end{bmatrix}, B = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \frac{K}{\tau} \end{bmatrix}. \tag{29}$$

The LQR control is designed for this linearized model which finds the optimal control $\tilde{u} = -K_{LQR} \cdot \tilde{x}$ to minimize the following cost

$$J = \int_0^\infty [x(t)^\intercal Q x(t) + u(t)^\intercal R u(t)] dt$$

where $Q = Q^\intercal \succeq 0$ and $R = R^\intercal \succeq 0$. Through trial and error, we chose $R = 0.5$ and

$$Q = \begin{bmatrix} 800 & 0 & 0 & 0 \\ 0 & 0.01 & 0 & 0 \\ 0 & 0 & 0.01 & 0 \\ 0 & 0 & 0 & 2.5 \end{bmatrix}.$$

To calculate the value of K_{LQR}, we used the built-in `lqr` MATLAB function during simulation. We initially planned for a time-varying K_{LQR}, with $x^{eq} = x_r$ and $u^{eq} = u_r$, where (x_r, u_r) is the reference trajectory. However, this was not feasible to implement on Simulink, and hence we used an approximate time-invariant K_{LQR} evaluated at $(x^{eq}, u^{eq}) = (0,0)$, which we further refined based on experimental results for hardware testing. This resulted in the following K_{LQR} for simulation and hardware testing, respectively:

$$K_{sim} = \begin{bmatrix} 40 & 41.774 & 9.123 & 1.731 \end{bmatrix},$$
$$K_{hardware} = \begin{bmatrix} 50 & 35 & 9.123 & 1.731 \end{bmatrix}.$$

It is important to note that the ball and beam system only measures the position and angle, which is not the full-state measurement. LQR controller best works with a full state feedback control, which is why we implemented a Luenberger Observer (discussed in Section III) which gives an estimate $\hat{x}$ for the state. Using this state estimate, the servo voltage is found to be $V_{servo} = -K_{LQR}(\hat{x} - x_r)$. Note that the reference trajectory x_r only defines the position ($x_{r,1}$), velocity ($x_{r,2}$) and acceleration ($\dot{x}_{r,2}$) of the ball. We calculate the reference angle and angular velocity based on these as follows

$$x_{r,3} \approx \arcsin(\frac{7L}{5gr_g}\dot{x}_{r,2})$$
$$x_{r,4} = 0 \implies \dot{x}_{r,4} = 0 \implies u_r = 0.$$

We implemented multiple changes during hardware testing to improve the performance of the LQR controller. For instance, we made the same adjustment presented in 27, and adjusted the sensor output for ball position (x_1) slightly as shown below, to account for measurement errors

$$x_1 \leftarrow \frac{20}{19} x_{1,meas} - 0.0185. \tag{30}$$

We have also limited the beam angle θ_d to $[-\frac{50\pi}{180}, \frac{50\pi}{180}]$, and experimentally adjusted the overall gain as shown below, to ensure that safety restrictions are not violated while minimizing the score

$$V_{servo} = 0.8 \cdot (-K_{LQR}(\hat{x}(t) - x_r(t))). \tag{31}$$

We further clipped the servo voltage at $\pm 1V$ to reduce the energy cost without much reduction in the tracking cost:

$$V_{servo} = \begin{cases} 1V & \text{if } V_{servo} > 1V \\ -1V & \text{if } V_{servo} < -1V \\ V_{servo} & \text{otherwise.} \end{cases} \tag{32}$$

IV. Simulation Results

In this section, we test our controllers and observer in simulation by evaluating how well the controllers track sine and square wave reference trajectories. Specifically, we present results for waves with an amplitude of 0.04 m and a period of 10 sec with an initial position of -0.19 m from the sensor. We simulated all trajectories for $T = 100$ sec.

The resulting state history can be seen for the PID controller in Fig. 1 and 2 and Fig. 3 and 4 for the LQR controller. Further, Table II presents the performance statistics for the PID and LQR controllers.

TABLE II: Simulation Performance

Reference	Controller	Tracking Cost	Energy Cost	Total Score
Sine Wave	LQR	0.74	0.06	0.81
	PID	0.87	0.03	0.90
Square Wave	LQR	2.91	0.60	3.51
	PID	3.58	0.15	3.73

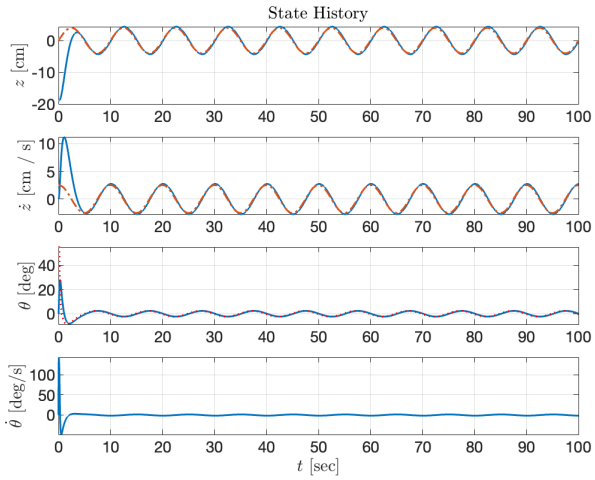

Fig. 1: Performance of PID controller for a sine wave reference trajectory. The blue line represents the actual system, and the red line represents the desired trajectory.

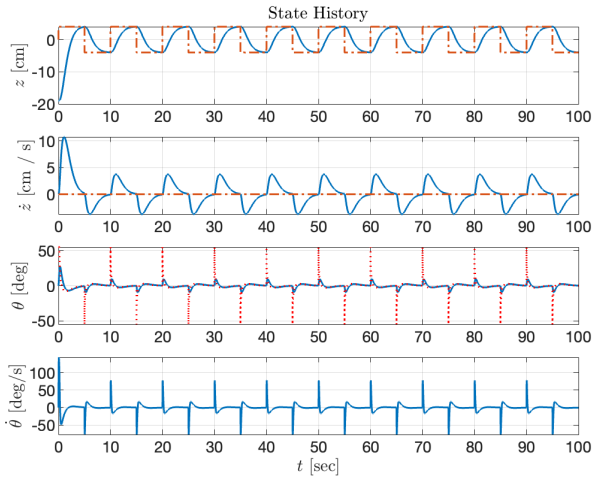

Fig. 2: Performance of PID controller for a square wave reference trajectory.

Generally, the LQR controller performs better than the PID for both sine and square waves. This could be due to many reasons, such as tuning or choice of linearization point. It can also be seen that both the PID and LQR controllers tend to perform better on sine waves than square waves in terms of both tracking performance and energy cost. We provide a further comparison of controller performance in simulation in the Appendix.

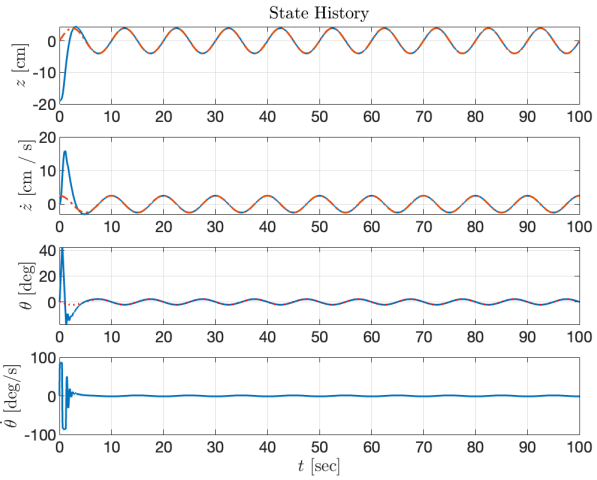

Fig. 3: Performance of LQR controller for a sine wave reference trajectory.

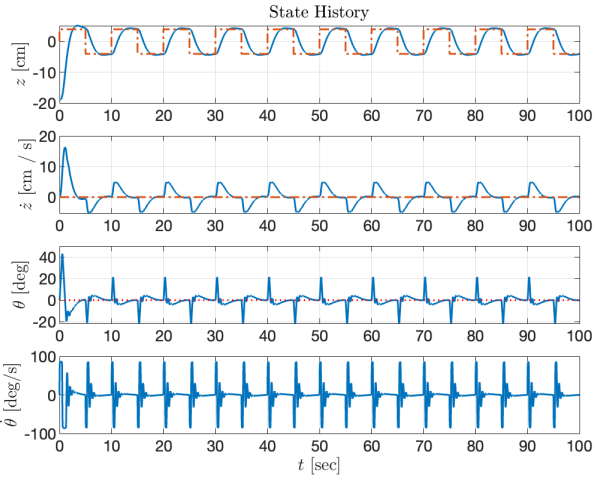

Fig. 4: Performance of LQR controller for a square wave reference trajectory.

V. Hardware Experiment Results

The PID and LQR controllers were tested on the Quansar ball and beam hardware. An unknown evaluation trajectory was used to assess the ability of the controllers. The LQR controller exhibited far better performance, as can be seen in Table III.

TABLE III: Hardware Performance

Reference	Controller	Tracking Cost	Energy Cost	Total Score
Evaluation Trajectory	LQR	0.28	1.31	**1.59**
	PID	5.35	3.43	8.78

We largely attribute the huge discrepancy in performance to tuning. The LQR controller from simulation immediately

performed better than the PID controller; therefore, we chose to focus on tuning the LQR parameters. We note, that PID controllers can often perform much better than as seen here. Given more time, we would have worked to tune the PID terms further. This performance discrepancy can also be seen between Fig. 5 and 6 which display the control inputs and outputs of the LQR controller and Fig. 7 and 8 which correspond to the PID controller.

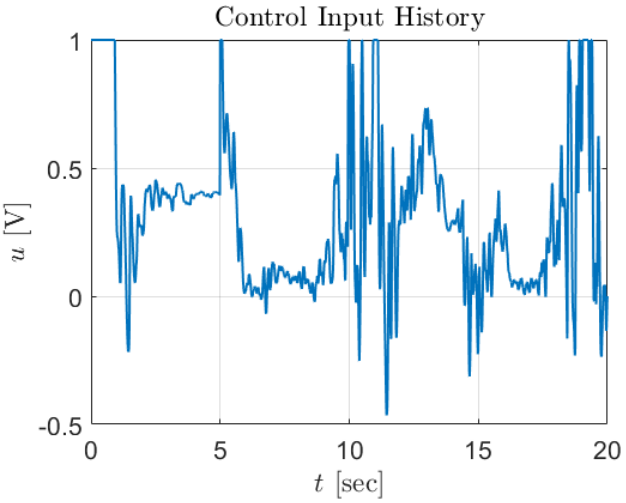

Fig. 5: LQR Control Inputs

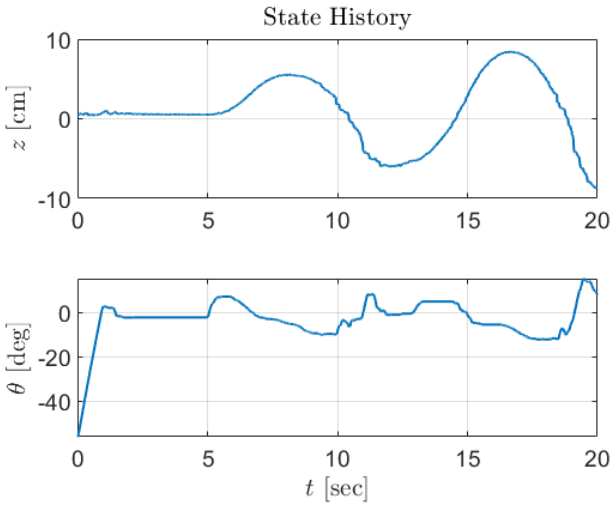

Fig. 6: LQR Output

VI. Conclusion

This project investigates the use of PID and LQR controllers for the control of a ball and beam system in reference tracking. Additionally, a Luenberger observer was implemented to estimate the state of the ball. We analyzed the efficacy of this filter empirically through comparisons with the true state and saw very little estimation error.

The simulation portion of this lab indicated that while both controllers were effective, the LQR outperformed the

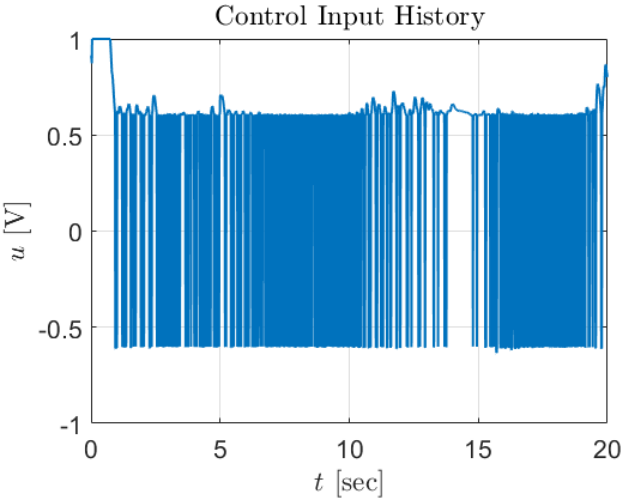

Fig. 7: PID Control Inputs

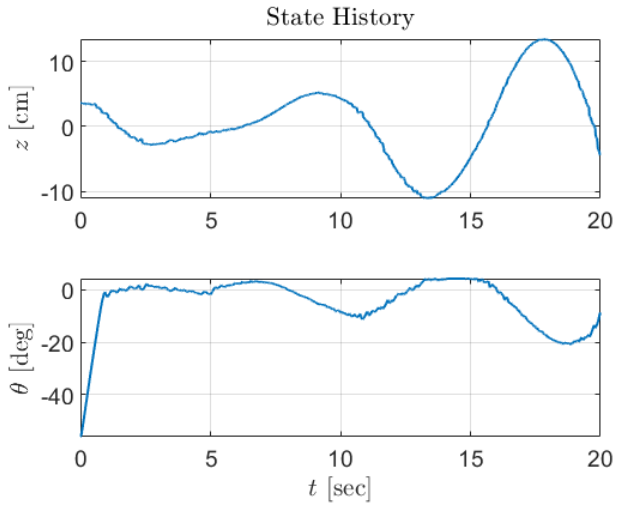

Fig. 8: PID Output

PID when tuned for a specific reference trajectory. Due to the fine-tuning required for the PID controller, it performed far worse during hardware experiments. The LQR controller had less variance in parameters from simulation to hardware experimentation, providing us with an easy starting point.

The performance of both of the controllers was worse on hardware than in simulation, specifically when tested on sine waves. For the PID controller, the applied servo voltage oscillated heavily when testing on the hardware. Consequently, the energy cost of the PID controller was significantly higher. While the LQR controller had a similar issue, the difference in energy cost and servo voltage was significantly less than the PID controller. Because of this phenomenon, we had to greatly adjust the gains of both controllers to ensure optimal energy performance. Additionally, we set the bounds on the servo voltage lower, to ensure optimal energy costs. Finally, we manually adjusted the computed servo voltage because we noticed that the angle of the beam did not move when the

magnitude of the servo voltage was small, but non-zero. This can be seen above in Equation (27).

In conclusion, we found there to be a discrepancy in the performance of our controllers between simulation and hardware testing due to modeling errors in our system equations. For instance, the model does not account for friction between the ball and the beam, gravitational forces when adjusting the angle of the beam, and more. However, we still see that the LQR controller was able to track the reference trajectory quite successfully.

APPENDIX

A. Comparative Analysis: PID vs LQR Score Trends

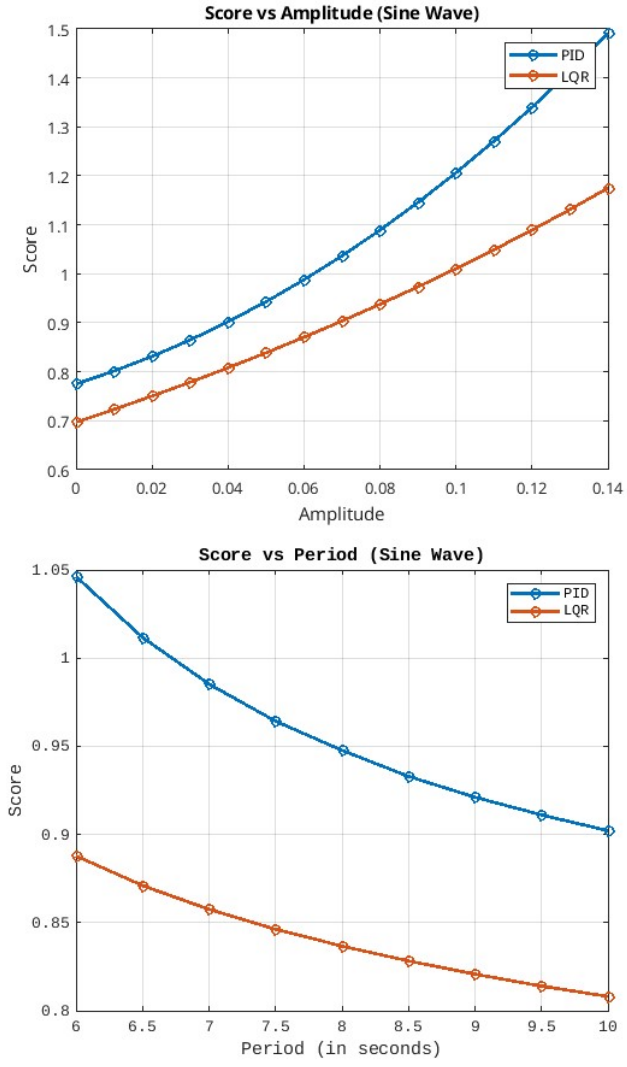

Fig. 9: Variation in score across amplitude and period for the sine wave

To better understand how the PID and LQR controllers perform in simulation, we compare scores across various amplitudes and periods 9, 10.

First, we hold the period constant. For the sine wave, we see a nearly linear increase in score for LQR and a slightly worse than linear increase for PID. For the square wave, we see a more exponential increase, where the score starts to blow up for both LQR and PID. Next, we hold the amplitude constant. For the sine wave, we see a fairly slow decline in score for both PID and LQR; however, the PID has overall higher scores. For the square wave, we see a nearly linear decrease in score for PID and a slightly faster decrease for LQR. Overall, we see slightly lower score trends for the LQR controller.

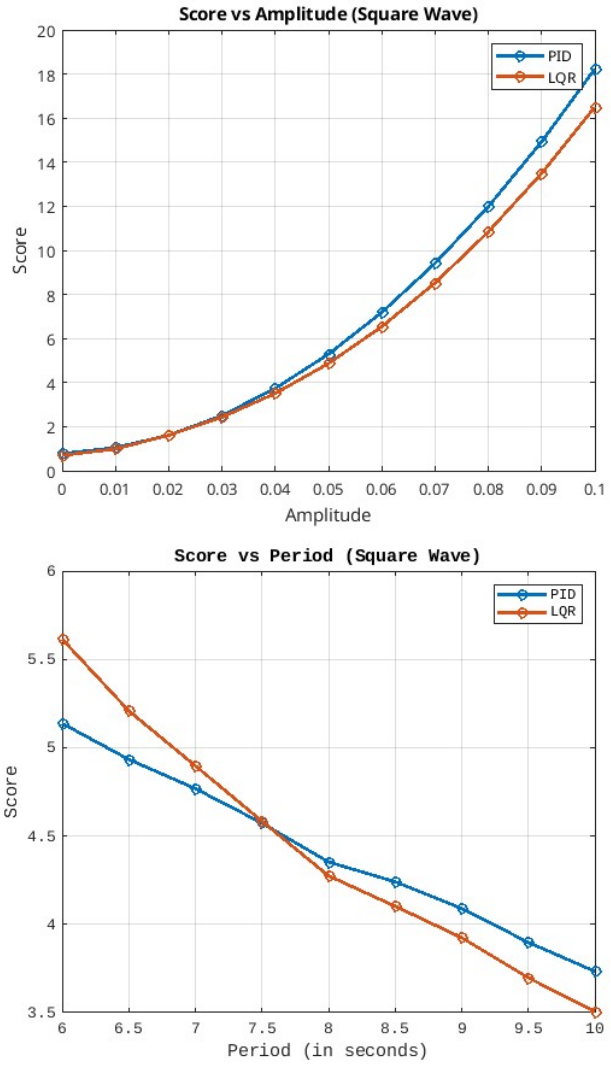

Fig. 10: Variation in score across amplitude and period for the square wave

REFERENCES

[1] R. I. of Technology, "Srv02 experiment 3: Ball & beam," https://www.se.rit.edu/~se463/ResearchProject/Quanser/SRV02_Exp3_Ball%20&%20Beam.pdf, 2025, accessed: 2025-04-30. [Online]. Available: https://www.se.rit.edu/~se463/ResearchProject/Quanser/SRV02_Exp3_Ball%20&%20Beam.pdf

[2] U. B. EE222, "Nonlinear systems: Analysis, stability, and control course project," 2025.

[3] M. Arcak, "Feedback Aspects," Lecture Notes, Linear Systems 221A, 2023, university California, Berkeley.