# 222 Ball and Beam: Simulation

Liz Dietrich, Gagan Deep Goru, Pavan Sumanth, and Yuneil Yeo

## 1 Introduction

The ball and beam system is a very common example for nonlinear control. In this project, we will design two controllers, and an observer, to drive the ball position to a desired trajectory. The main objective of the controllers is to reduce the tracking error and the energy spent by the motor. Further, we must ensure the controllers operate in a safe region (e.g. the ball does not escape the physical limits of the beam). We have chosen to implement (1) a PID controller, (2) an LQR controller, and (3) a Luenberger Observer.

**GITHUB:** `https://github.com/gagandeep25/EE222-Nonlinear-Systems-Ball-and-Beam-Project.git`

## 2 PID

The first controller we implemented is a Proportional Integral Derivative (PID) controller. We chose a PID controller, as the system we are looking at has relatively low-order dynamics and a known, intuitive behavior. A PID controller is well-suited for our objective of achieving stable tracking. Moreover, a PID controller provides sufficient control authority with minimal computational efforts and complexity.

The advantages of a PID controller are 1) 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) potential for overshoot or instability due to the integral term. We tune the gains for the PID controller through trial and error, specifically tunning with consideration to the score based on tracking performance, energy cost, and safety violation.

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 error ($e$, the positional difference between the actual ball position and the desired reference), which is

$$P(t) = k_p e(t), \tag{1}$$

where $k_p$ is the gain for the proportional term. We set it to 7 for our PID controller. The integral term accumulates the error over time, which is

$$I(t) = k_i \sum_{\tau=0}^{t} e(\tau)\Delta t \tag{2}$$

where $k_i$ is the gain for the integral term. We set it to 0.01 for our PID controller. The derivative term predicts the future error based on the rate of change. Therefore,

$$D(t) = k_d \frac{e(t) - e(t - \Delta t)}{\Delta t}, \tag{3}$$

where $k_d$ is the gain for the derivative term. We set it to 8 for our PID controller.

By summing the PID terms, the desired beam angle $\theta_d$ is found as follows:

$$\theta_d(t) = \max\left(\min\left(-P(t) - I(t) - D(t), \frac{55\pi}{180}\right), \frac{-55\pi}{180}\right). \tag{4}$$

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

$$V(t) = k_{servo}(\theta_d(t) - \theta(t)), \tag{5}$$

where $k_{servo}$ is the servo voltage gain. We set it to 2 for our PID controller. We limit the servo voltage to be $\pm 2V$.

# 3 LQR Controller

The second controller we implemented is a Linear Quadratic Regulator (LQR) controller. The LQR controller is guaranteed to stabilize the equilibrium, while also allowing room to tradeoff input energy against the tracking error. Another advantage of the LQR algorithm is that it is computationally simpler than many non-linear control strategies.

To stabilize the trajectory $(x_r(t), u_r(t))$, define the variables

$$\begin{aligned}
&\tilde{x}(t) = x(t) - x_r(t) \qquad \tilde{u}(t) = u(t) - u_r(t) \\
\implies &\dot{\tilde{x}} = f(x(t), u(t)) - f(x_r(t), u_r(t)) \\
\implies &\dot{\tilde{x}} \approx \frac{\partial f}{\partial x}(x_r(t), u_r(t))x + \frac{\partial f}{\partial u}(x_r(t), u_r(t))u \\
&= A\tilde{x} + B\tilde{u}
\end{aligned}$$

For the ball and beam system, the reference trajectory defines the position $(x_{r,1})$, velocity $(x_{r,2})$ and the acceleration $(\dot{x}_{r,2})$ of the ball. We calculate the reference angle and angular velocity based on these as follows

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

Using the information of $x_r(t), u_r(t)$, 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

$$\begin{aligned}
Q &= \begin{bmatrix} 800 & 0 & 0 & 0 \\ 0 & 0.01 & 0 & 0 \\ 0 & 0 & 0.01 & 0 \\ 0 & 0 & 0 & 2.5 \end{bmatrix} \\
R &= 0.5
\end{aligned}$$

The closed loop system is given as $\dot{\tilde{x}} = (A - BK_{LQR})\tilde{x}$, which drives the tracking error $x(t) - x_r(t) \to 0$ in the steady state. (The matrix $A - BK_{LQR}$ is Hurwitz by design)

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 Luerberger Observer (discussed in the next section) which gives an estimate $\hat{x}$ for the state.

Using this state estimate, the servo voltage input is found as $V = u_r - K_{LQR}(\hat{x} - x_r)$. The input is further limited to $\pm 1V$ so that we have low energy consumption while not deteriorating the tracking error by a lot.

# 4 Luenberger Observer

As discussed in the project statement, we do not have measurement of the full state. Therefore, we must design an 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:

$$\dot{\hat{x}} = A\hat{x}(t) + Bu(t) \tag{6}$$

$$\hat{y} = C\hat{x}(t) + Du(t) \tag{7}$$

where

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{5gr}{7L} & -\frac{5L}{14}(\frac{r}{L})^2 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & -\frac{1}{\tau} \end{bmatrix}, B = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \frac{K}{\tau} \end{bmatrix}, C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, D = \begin{bmatrix} 0 \end{bmatrix}. \tag{8}$$

We are interested in the error: $e := \hat{x} - x$ between the simulated state and the actual state [2]. We can use the discrepancy between the measured output $y$ and the predicted output $\hat{y}$ to correct the simulation copy:

$$\dot{\hat{x}}(t) = A\hat{x}(t) + Bu(t) + L(\hat{y}(t) - y(t)) \tag{9}$$

Then the error becomes:

$$\begin{aligned} \dot{e} &:= \dot{\hat{x}} - \dot{x} && (10)\\ &= A\hat{x}(t) + Bu(t) + L(\hat{y}(t) - y(t)) - Ax(t) && (11)\\ &= A(\hat{x}(t) - x(t)) + Bu(t) + LC(\hat{x}(t) - x(t)) && (12)\\ &= Ae(t) + Bu(t) + LCe(t) && (13)\\ &= (A + LC)e(t) + Bu(t) && (14) \end{aligned}$$

### 4.1 Pole Placement

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 $e(t)$. Therefore, as a first attempt in designing this observer, we arbitrarily placed the poles of $A + LC$ at $-10, -12, -15, -18$ which resulted in an observer gain of

$$L = \begin{bmatrix} 25.1320 & 151.9416 & -1.2948 & 32.6879 \\ -0.5331 & -6.4568 & -10.1320 & 619.3871 \end{bmatrix}^\top . \tag{15}$$

### 4.2 Sensor Noise Assumptions

Consider the linearized system from above:

$$\begin{aligned} \dot{\hat{x}} &= A\hat{x}(t) + Bu(t) + w && (16)\\ \hat{y} &= C\hat{x}(t) + Du(t) + v && (17) \end{aligned}$$

with process noise $w$ and sensor noise $v$. We assume that $w$ and $v$ are positive, symmetric, and uncorrelated, with mean zero white noise vectors.

### 4.3 Kalman Filter

Kalman filtering uses observed measurements over time, including noise, to produce estimates of unknown variables. The filter's response can be measured in terms of it's Kalman gain. Kalman gain is the weight given to the measurements and current-state estimate. For example, with a low gain, the filter conforms to the model predictions, while a high gain will place more weight on the most recent measurements [1]. We will utilize this gain in the design of our observer. Consider a quadratic cost function for the expectation of the estimation error [1]:

$$J(x) = E[||x||^2] = \lim_{T\to\infty} \frac{1}{2T} \int_{-T}^{T} ||x||^2 dt. \tag{18}$$

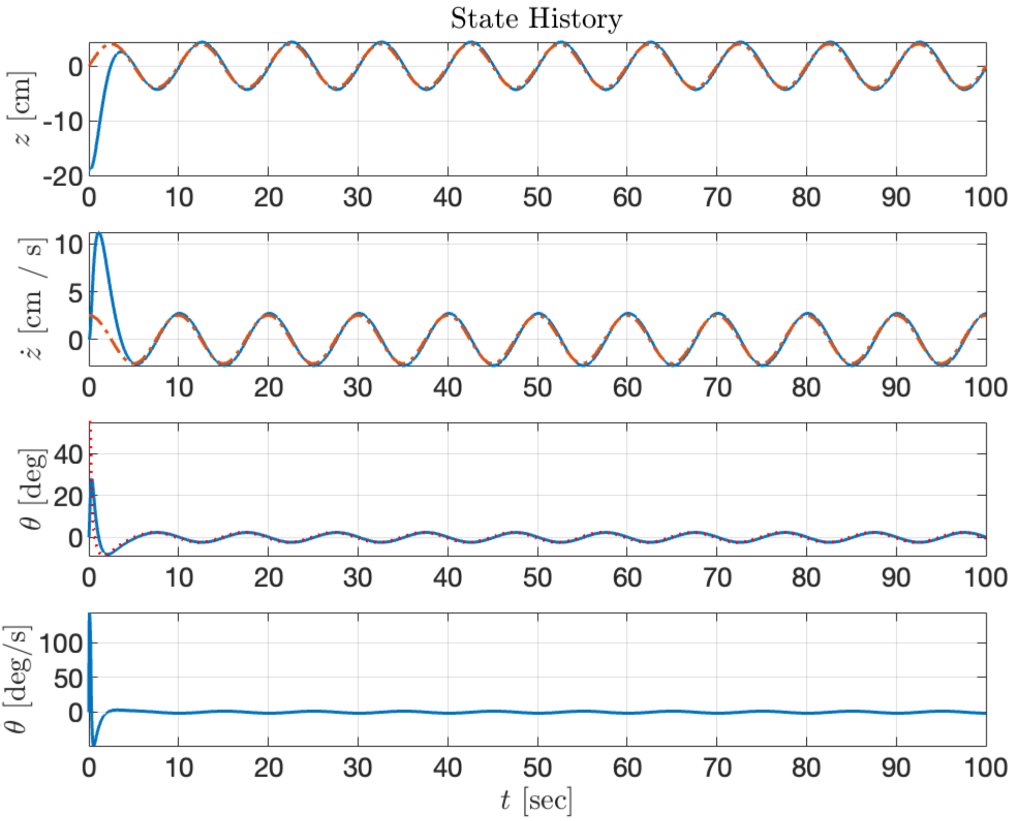
The observer that minimizes this cost function for (16) is a Luenberger observer with gain $L = \Pi_+ C^\top V^{-1}$ [1, 4] which can be obtained from `kalman`, such that `[kalmf,L,P] = kalman(sys,Q,R)`. Here, `sys` is the state space representation of our system, and $Q$ and $R$ represent the mean and covariance of $w$ and $v$, respectively. Using [3] as a starting point, we designed $Q = diag(10, 100, 10, 100)$ and $R = diag(0.1, 0.1)$. The resulting gain matrix $L$ is fed back into the Luenberger observer, replacing the values calculated in (15).

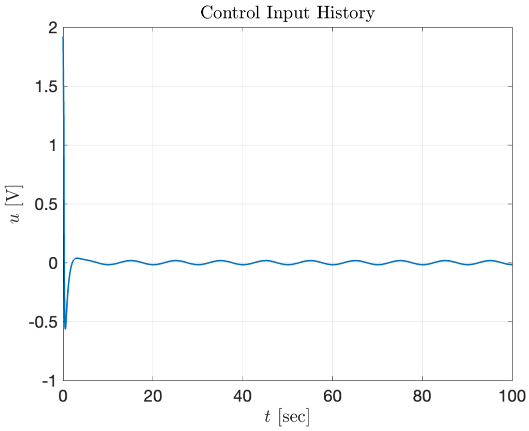
## 5 Results

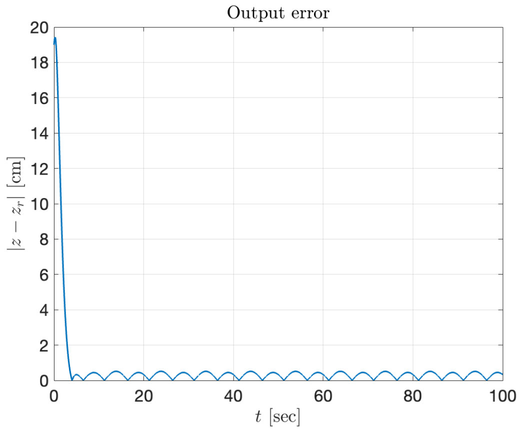
In this section, we test our controllers and observer by evaluating how well the controller can track the reference trajectories: sine wave and square waves. Plots for State History, Control Input History, and Output Error are for the case with the amplitude of 0.04m and the period of 10 seconds, with the initial position of -0.19 m from the sensor. We simulated our controllers and observers for T = 100 seconds.

## 5.1 PID

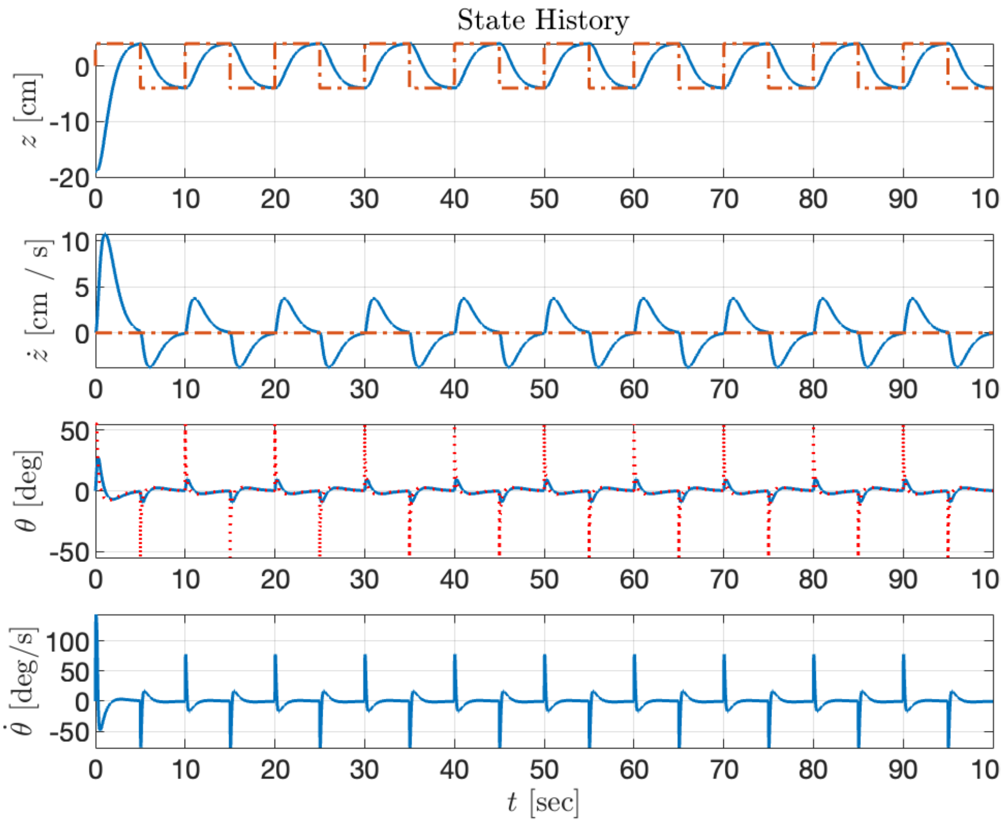
### 5.1.1 Sine Waves

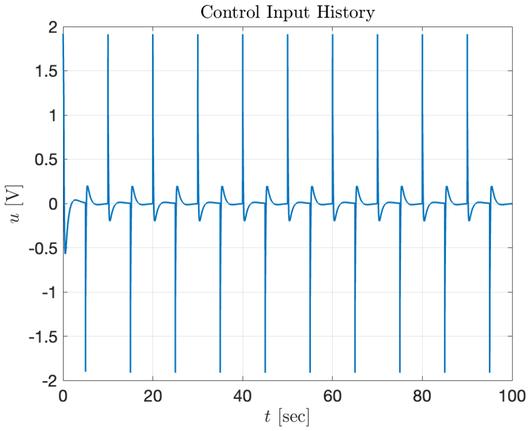


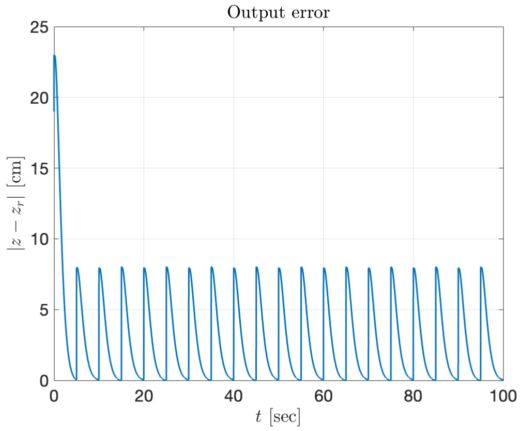




### 5.1.2 Square Waves





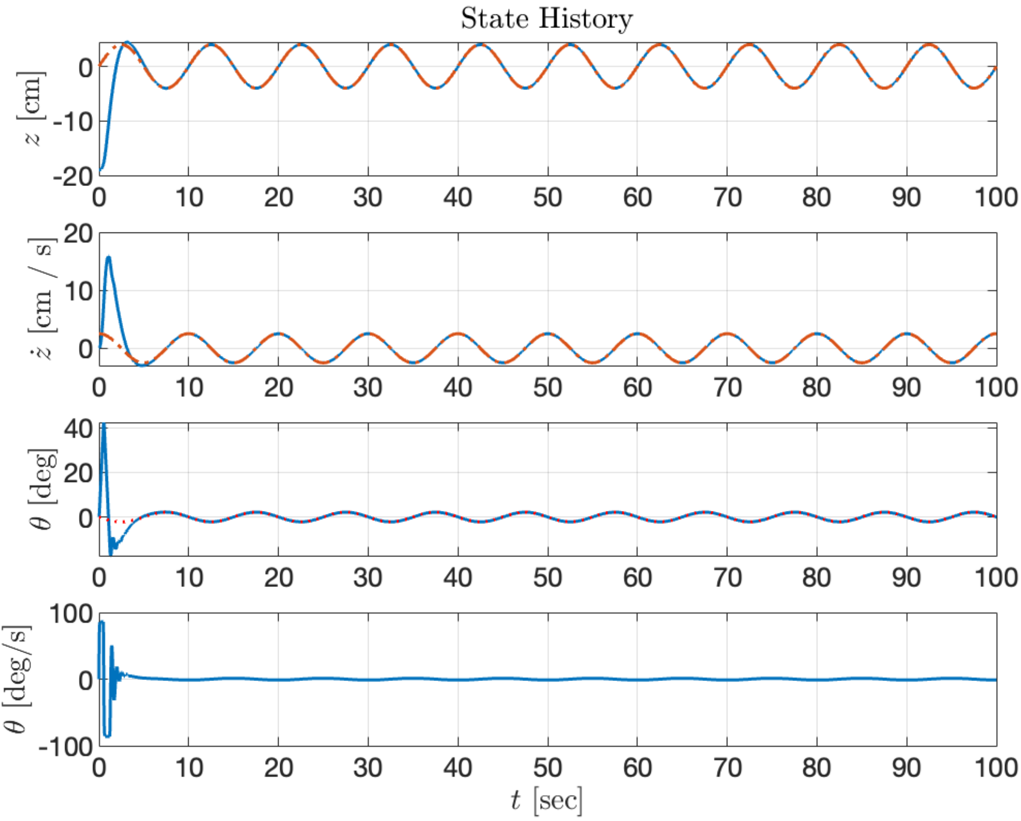


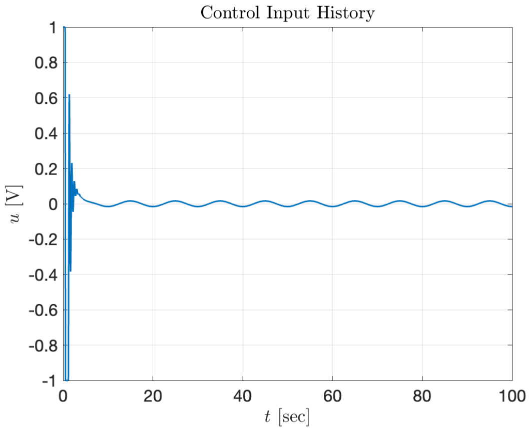
### 5.1.3 Single Run Score

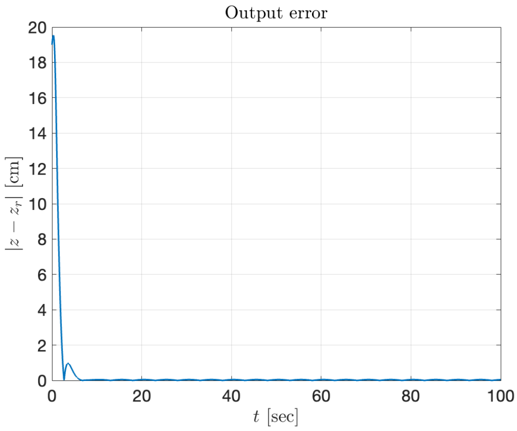
- Average Tracking Error: 0.0005 (Sine Waves) & 0.0020 (Square Waves)
- Average Energy Consumption: 0.0067 (Sine Waves) & 0.0304 (Square Waves)
- Safety Constraint Violation: 0 (Sine Waves) & 0 (Square Waves)
- Tracking Cost: 0.87 (Sine Waves) & 3.58 (Square Waves)
- Energy Cost: 0.03 (Sine Waves) & 0.15 (Square Waves)
- Safety Cost: 0.00 (Sine Waves) & 0.00 (Square Waves)
- Total Score: 0.90 (Sine Waves) & 3.73 (Square Waves)

## 5.2 LQR

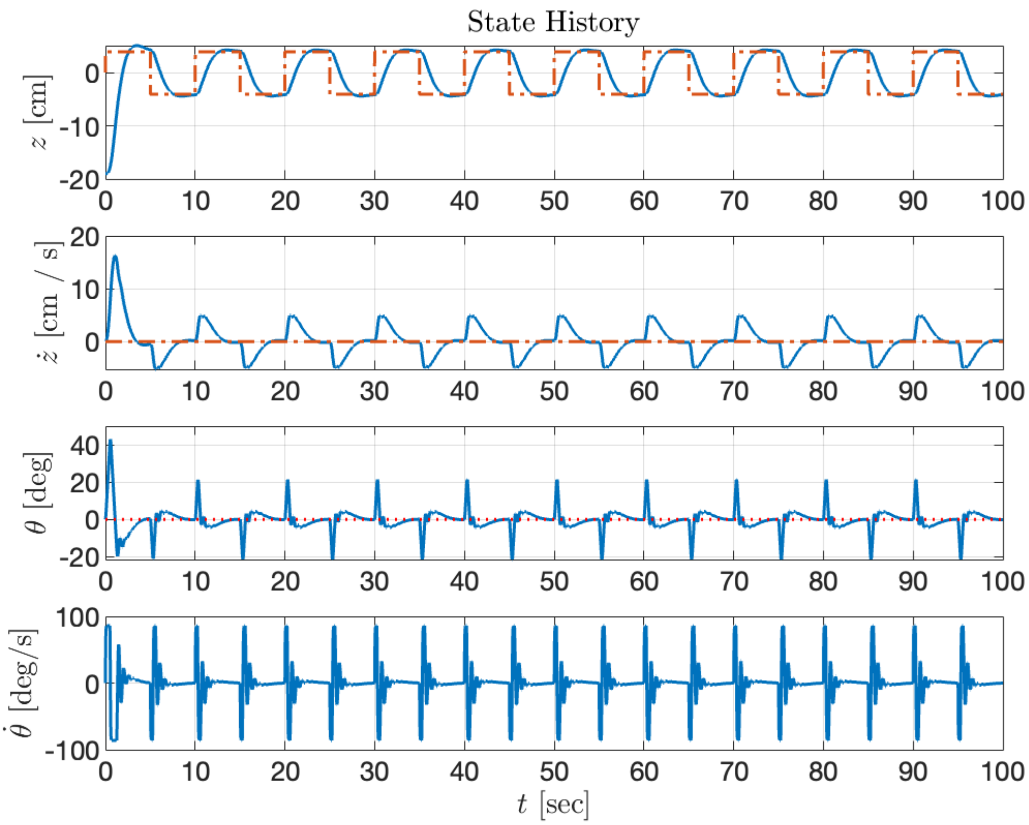
### 5.2.1 Sine Waves

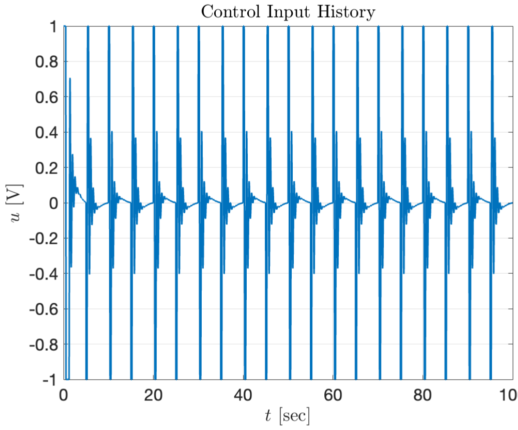


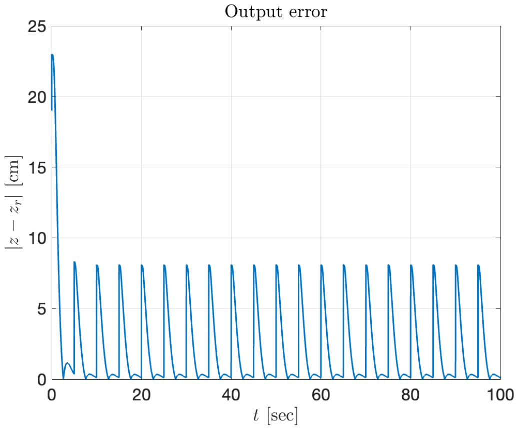




### 5.2.2 Square Waves







### 5.2.3 Single Run Score

- Average Tracking Error: 0.0004 (Sine Waves) & 0.0016 (Square Waves)
- Average Energy Consumption: 0.0128 (Sine Waves) & 0.1191 (Square Waves)
- Safety Constraint Violation: 0 (Sine Waves) & 0 (Square Waves)
- Tracking Cost: 0.74 (Sine Waves) & 2.91 (Square Waves)
- Energy Cost: 0.06(Sine Waves) & 0.6 (Square Waves)
- Safety Cost: 0 (Sine Waves) & 0 (Square Waves)
- Total Score: 0.81 (Sine Waves) & 3.5061 (Square Waves)

## 5.3 Comparative Analysis: PID vs LQR Score Trends

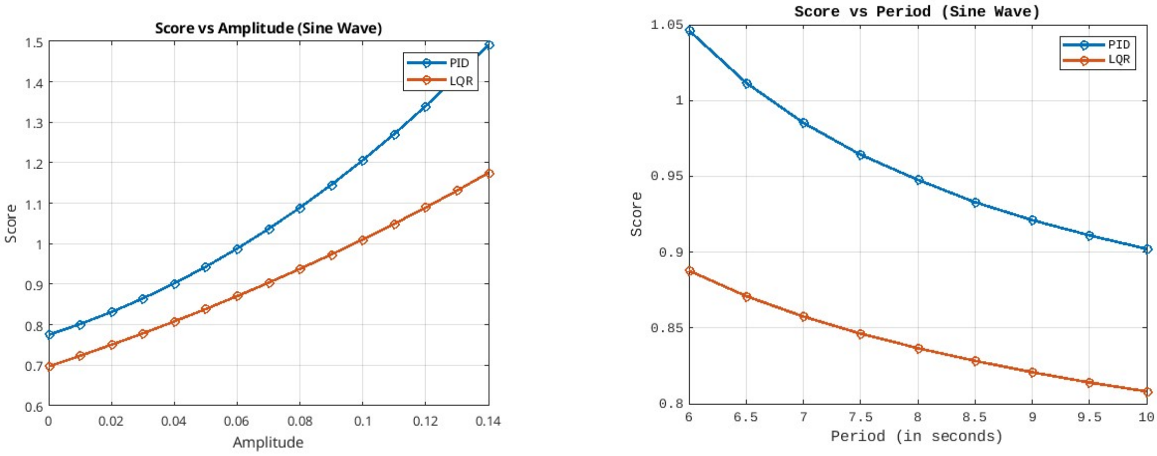


Figure 5: Variation in score across amplitude and period for the sine wave

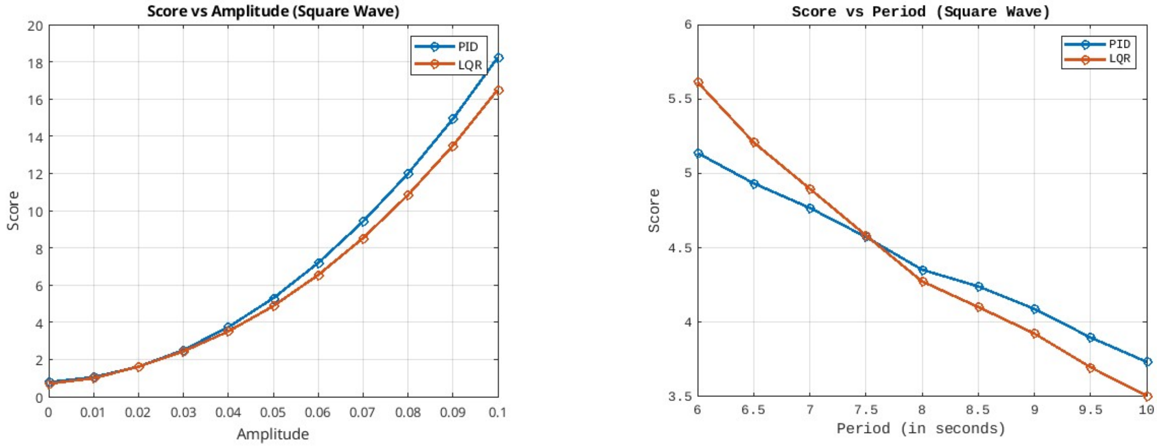


Figure 6: Variation in score across amplitude and period for the square wave

When considering a single run, as seen in Section 5.1 and 5.2, the highest magnitude of the control input is greater for PID compared to LQR. The PID performs slightly better than LQR in terms of the score for both sine and square waves. Both PID and LQR tend to do better for sine waves compared to square waves in terms of both tracking performance and energy cost.

Further, to better understand how each controller performs, we compare scores across various amplitudes and periods 5, 6. 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.

**Please note that we have tested our controller with Simulink and checked that both controllers work.**

## References

[1] Matthias Althoff. Continuous Control (MIMO). Lecture Notes, Cyber-Physical Systems, 2022. Technical University of Munich.

[2] Murat Arcak. Feedback Aspects. Lecture Notes, Linear Systems 221A, 2023. University California, Berkeley.

[3] Rahul Soni and Sathans. Optimal control of a ball and beam system through LQR and LQG. In *2018 2nd International Conference on Inventive Systems and Control (ICISC)*, pages 179–184, 2018.

[4] Robert F. Stengel. *Optimal Control and Estimation.* Dover Books on Mathematics, 1994.