# Nonlinear Control of Ball and Beam System

Carlo Bosio
Mechanical Engineering
University of California, Berkeley
Berkeley, USA
c.bosio@berkeley.edu

Seoyeon Choi
Mechanical Engineering
University of California, Berkeley
Berkeley, USA
seoyeon99@berkeley.edu

Spencer Schutz
Mechanical Engineering
University of California, Berkeley
Berkeley, USA
spencer.schutz@berkeley.edu

***Abstract*—In this report we document our experimentation on a real-world Ball and Beam system. We describe our state estimation algorithm based on an Extended Kalman Filter (EKF), and the two control approaches used, i.e. a Linear Quadratic Regulator (LQR) and a Feedback Linearization (FL) controller. We detail the tuning process of the control and estimation parameters, and report the tracking performances achieved. We discuss observations made during experiments and the most significant differences compared to the simulated system (focus of the first part of the project). The GitHub repository link is https://github.com/spencer-schutz/EE222-Nonlinear-Systems-Ball-and-Beam-Project**

## I. Introduction

The goal of this project is to control a ball and beam system, shown in Figure 1. A metal ball moves along a track that has one pinned end and one end attached to a servo (via a lever arm and gear). As the servo rotates, the track angle changes and gravity accelerates the ball down the track.

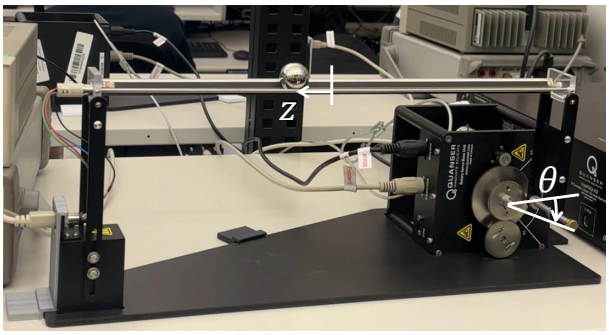



Fig. 1: Ball and beam hardware with labeled states $z$ and $\theta$.

The control objective is to track sine and square wave reference trajectories with minimum energy usage and guaranteed safety. These goals are scored through a cost function that sums tracking error, energy consumption, and safety constraint violation with respective weights $w_t, w_e$, and $w_s$:

$$J_{score} = w_t J_{tracking} + w_e J_{energy} + w_s J_{safety} \tag{1}$$

Minimizing $J_{score}$ requires overcoming challenges including incomplete state information, process and measurement noise, and modeling inaccuracies via state estimation and control.

*Outline:* In Section II we describe the models used for state estimation and control. In Section III we detail our state estimation approach and the two control approaches developed. In Sections IV and V we report our simulated and experimental results. Finally, we conclude in Section VI with some observations and comments.

## II. System Modeling

A diagram of the ball and beam system is shown in Figure 2. The system state is $x = [z, \dot{z}, \theta, \dot{\theta}]^T$, where $z$ is the deviation of the ball from the center of the beam and $\theta$ is the angle of the arm of the servo motor with respect to the horizontal axis (see Figure 1). The control input is $u$, the voltage applied to the motor.

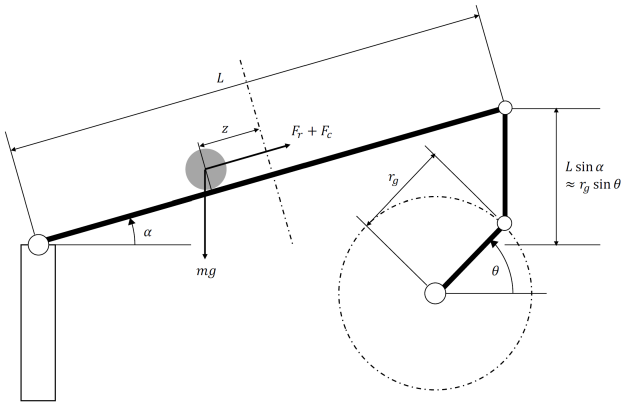



Fig. 2: Ball and beam system diagram.

With this choice of $x$ and $u$, the continuous-time nonlinear dynamics are given by:

$$\dot{x}_1 = x_2 \tag{2a}$$

$$\dot{x}_2 = \frac{5g}{7}\frac{r_g}{L}\sin x_3 - \frac{5}{7}\left(\frac{L}{2} - x_1\right)\left(\frac{r_g}{L}\right)^2 x_4^2 \cos^2 x_3 \tag{2b}$$

$$\dot{x}_3 = x_4 \tag{2c}$$

$$\dot{x}_4 = -\frac{x_4}{\tau} + \frac{K}{\tau}u \tag{2d}$$

Parameter values for the hardware setup are listed in the Appendix (Table III). The state equations are written succinctly as

$$\dot{x}(t) = f_c(x(t), u(t)). \tag{3}$$

The dynamics (3) will be further manipulated in Section III to meet the specifications of each observer and controller. Measurements of $z$ and $\theta$ are collected with sampling time

$dt$. In Section III, the measurement model (4) will be further manipulated to consider uncertainty.

$$\begin{aligned} y[k] &= Hx[k] \\ H &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \end{aligned} \tag{4}$$

## III. Observer and Controller Design

We implemented an Extended Kalman Filter for our observer and two controllers: 1) LQR; and 2) approximate feedback linearization with LQR. After observing a drop in performance from simulation to hardware, each controller was augmented with integral action. The details are presented in the following subsections.

### A. Extended Kalman Filter

We use an Extended Kalman Filter (EKF) for state estimation. The EKF is composed of a prediction step, propagating an internal state estimate through the dynamics equation, and a measurement update step, correcting the internal estimate based on incoming measurement [1], [2]. We denote with the $p$ subscript the quantities involved in the prediction steps and with the $m$ subscript the ones involved in the measurement update step. We discretize (3) using Forward Euler with timestep $dt$, and add a random process noise $v[k]$ with mean zero and covariance $\Sigma_v$. Furthermore, the measurement equation is augmented with a random measurement noise $w[k]$ with mean zero and covariance $\Sigma_w$.

$$\begin{aligned} x[k] &= x[k-1] + f_c(x[k-1], u[k-1]) * dt + v[k-1] \\ &= f_d(x[k-1], u[k-1], v[k-1]) \quad \text{(5a)} \\ y[k] &= Hx[k] + w[k] \quad \text{(5b)} \end{aligned}$$

The nonlinear discrete dynamics are linearized about the previous estimate $x_m[k-1]$, known $u[k-1]$, and zero variance as follows:

$$A_{EKF}[k-1] = \frac{\partial f_d}{\partial x}(x_m[k-1], u[k-1], 0) \tag{6}$$

The EKF is initialized with a state estimate $x_m[0]$ and a covariance matrix $P_m[0]$. The prediction step is then performed as follows:

$$\begin{aligned} x_p[k] &= f_d(x[k-1], u[k-1], 0) \\ P_p[k] &= A_{EKF}[k-1] P_m[k-1] A_{EKF}[k-1]^T + \Sigma_v. \end{aligned} \tag{7}$$

When a measurement $y[k]$ is received, the measurement update step is performed as follows:

$$\begin{aligned} K[k] &= P_p[k] H^T (H P_p[k] H^T + \Sigma_w)^{-1} \\ x_m[k] &= x_p[k] + K[k](y[k] - H x_p[k]) \\ P_m[k] &= (I - K[k]H) P_p[k], \end{aligned} \tag{8}$$

where $K[k]$ is the Kalman Filter gain at step $k$.

### B. LQR

The first controller we present is a reference-tracking LQR [3]. The reference state $x^*$ is constructed using the given reference ball position $p_{ref}$ and velocity $v_{ref}$. The reference input $u^* = 0$ for all steps.

$$x^*[k] = [p_{ref}[k], v_{ref}[k], 0, 0]^T \tag{9}$$

At each step, the discrete nonlinear dynamics in (5a) is linearized about the current reference state and input with zero process noise.

$$A_{LQR}[k] = \frac{\partial f_d}{\partial x}(x^*[k], u^*[k], 0) \tag{10a}$$

$$B_{LQR}[k] = \frac{\partial f_d}{\partial u}(x^*[k], u^*[k], 0) \tag{10b}$$

The controller chooses a time-varying state feedback gain $K[k]$ to minimize the following quadratic cost:

$$\begin{aligned} J = \sum_{k=0}^{\infty} (x_m[k] - x^*[k])^T Q (x_m[k] - x^*[k])^T + \\ (u[k] - u^*[k])^T R (u[k] - u^*[k]) \end{aligned} \tag{11}$$

The optimal cost is

$$J^* = (x_m[k]^T - x^*[k]) P[k] (x_m[k]^T - x^*[k]) \tag{12}$$

where $P[k]$ is the solution to the discrete time algebraic Riccati equation (DARE) at step $k$ (dropped from right side for brevity):

$$P[k] = A^T P A + Q - A^T P B \left(R + B^T P B\right)^{-1} B^T P A \tag{13}$$

At each step, $P[k]$ is used to calculate the feedback gain $K[k]$ (dependency on $k$ dropped from right side for brevity),

$$K[k] = (R + B^T P B)^{-1} B^T P A \tag{14}$$

and the applied input is

$$u[k] = u^*[k] - K[k](x_m[k] - x^*[k]). \tag{15}$$

The linearized dynamics are controllable and observable, so a unique $P[k] \succ 0$ exists for all k and the controller is locally asymptotically stabilizing for the nonlinear system.

### C. Approximate Feedback Linearization + LQR

The second controller we present combines approximate feedback linearization with LQR. Feedback linearization works by applying a control input that cancels nonlinear terms and adds linear state feedback, making the closed-loop system linear [4]. In our approach, we utilize an approximate full state feedback linearization by treating certain terms as disturbances. Then, we design the feedback gains using LQR to ensure asymptotic stability in the closed-loop system.

A system is fully feedback linearizable when its relative degree equals the order of the system. In our case, we view the term $-\frac{5}{7}(\frac{L}{2} - x_1)(\frac{r_g}{L})^2 x_4^2 \cos^2 x_3$ in (2d) as a disturbance to the system and ignore it to make the system fully feedback linearizable. We show the reason behind this choice in detail

later. We define the output as $y = x_1$ and the tracking error as

$$e(t) := y(t) - y_d(t) \tag{16}$$

where $y_d(t)$ is the reference position trajectory. Then, we can write the system as

$$\begin{aligned} e^{(1)} &= y^{(1)} - y_d^{(1)} = x_2 - y_d^{(1)} \\ e^{(2)} &= y^{(2)} - y_d^{(2)} = a \sin x_3 - y_d^{(2)} \\ e^{(3)} &= y^{(3)} - y_d^{(3)} = a x_4 \cos x_3 - y_d^{(3)} \\ e^{(4)} &= y^{(4)} - y_d^{(4)} \\ &= -\frac{a}{\tau} x_4 \cos x_3 - a x_4^2 \sin x_3 + \frac{ak}{\tau} \cos x_3 u - y_d^{(4)} \end{aligned} \tag{17}$$

where the superscript denotes the order of time derivatives and constant $a = \frac{5g}{7} \frac{r_g}{L}$. Notice that $\frac{ak}{\tau} \cos x_3 \neq 0$ when $x_3 \neq \pm \frac{\pi}{2}$. Since $x_3$'s safety bound is $[-\frac{\pi}{3}, \frac{\pi}{3}]$, we can conclude that $\frac{ak}{\tau} \cos x_3 \neq 0$. Therefore, the relative degree for (17) is 4, so the system is now fully feedback linearizable. With this, we define $e = \left[e, e^{(1)}, e^{(2)}, e^{(3)}\right]^\intercal$, $f(x) = -\frac{a}{\tau} x_4 \cos x_3 - a x_4^2 \sin x_3$, $g(x) = \frac{ak}{\tau} \cos x_3$, and the system becomes

$$\begin{aligned} \dot{e}_1 &= e_2 \\ \dot{e}_2 &= e_3 \\ \dot{e}_3 &= e_4 \\ \dot{e}_4 &= f(x) - y_d^{(4)} + g(x) u \end{aligned} \tag{18}$$

If we include the previously ignored term $-\frac{5}{7}(\frac{L}{2} - x_1)(\frac{r_g}{L})^2 x_4^2 \cos^2 x_3$, we get

$$g(x) = -\frac{10}{7} \frac{k}{\tau} (\frac{L}{2} - x_1)(\frac{r_g}{L})^2 x_4 \cos^2 x_3 \tag{19}$$

It is hard to achieve $g(x) \neq 0$, since we would need to make $x_4$, the angular velocity, to be nonzero. Even if we treat $-\frac{5}{7}(\frac{L}{2} - x_1)(\frac{r_g}{L})^2 x_4^2 \cos^2 x_3$ as disturbance, our controller still results in satisfactory control performance.

We separate $u = u_{cancel} + u_{LQR}$ where

$$\begin{aligned} u_{cancel} &= g(x)^{-1} \left( -f(x) + y_d^{(4)} \right) \\ u_{LQR} &= -k_1 e_1 - k_2 e_2 - k_3 e_3 - k_4 e_4 \end{aligned} \tag{20}$$

Here, $u_{cancel}$ cancels the nonlinear terms and $u_{LQR}$ provides feedback control using LQR. After applying $u_{cancel}$ to the system, the system can be shown in linear form:

$$\begin{aligned} \dot{e} &= \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} e + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} u_{LQR} \\ y &= \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix} e \end{aligned} \tag{21}$$

This system is controllable and observable, so LQR can be applied. We discretize A and B in (21) with zero-order hold (ZOH) and apply discrete LQR described in Section III-B.

Notice that when calculating $e$, we need up to fourth derivative of $y_d$. Given a second-order reference, we fit a polynomial to obtain the third and fourth derivatives. In practice, however, polynomial fitting may be unreliable or computationally expensive for certain trajectories. In such cases, we simply set the third and fourth derivative of $y_d$ to be zero and demonstrate that this approximation has minimal impact on performance.

### D. Integral Action

Due to significant model mismatch between simulation and hardware experiments, we added an integral control term to the LQR and FL controllers for hardware experiments. Assuming the nominal controller (i.e. LQR or FL) outputs a control $u[k]$, the control to be applied to the hardware $u_h[k]$ is obtained as

$$u_h[k] = u[k] - K_I * z[k] \tag{22}$$

where $z$ is the integral term computed as

$$z[k] = z[k-1] + e[k-1]\, dt \tag{23}$$

and $e[k-1]$ is the position tracking error at step $k-1$.

This integral term was particularly useful to compensate for slight offsets in the rest position of the beam and friction. These unmodeled contributions could be better compensated, for instance, through a learned feedforward term.

## IV. Simulation Results

We test our controllers in simulation on sine waves and square waves as references. Table I summarizes the results, Fig. 3 and Fig. 4 shows the state, error, and control input history plots, and the parameters used in simulation are listed in the Appendix (Table IV). LQR and FL + LQR achieve similar total scores for both references. LQR tends to yield a lower tracking cost, whereas FL + LQR provides a lower energy cost.

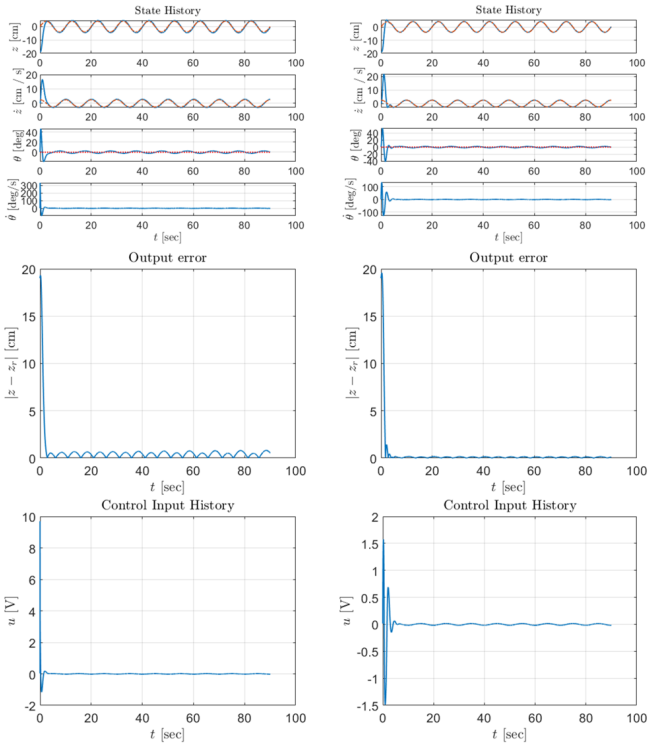
TABLE I: Simulation performance

| Reference | Controller | Tracking Cost | Energy Cost | Total Score |
|---|---|---|---|---|
| Sine Wave | LQR | 0.71 | 0.16 | 0.87 |
| | FL + LQR | 0.73 | 0.14 | **0.86** |
| Square Wave | LQR | 2.69 | 0.51 | **3.20** |
| | FL + LQR | 2.87 | 0.42 | 3.29 |

## V. Hardware Experiment Results

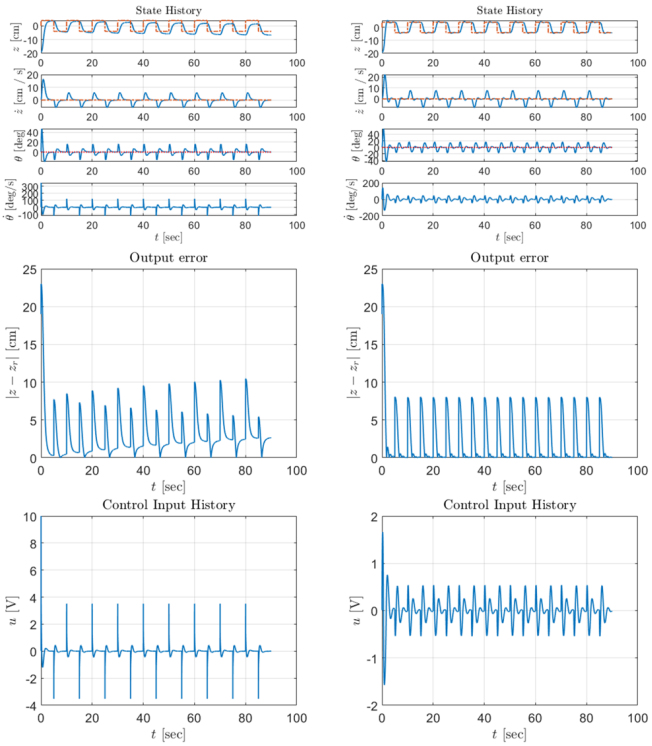
We test our controllers in the real hardware testbed shown in Fig. 1. The experiments for benchmarking were conducted on the given evaluation trajectory. Table II summarizes the results, Fig. 5 shows the state and control input history plots, and the Appendix (Table V) contains the parameters used in hardware simulations. In hardware experiments, FL + LQR shows much better score than LQR with lower cost in both tracking cost and energy cost.

The careful tuning of control and state estimation parameters was crucial to achieving a good performance. This process was conducted via trial and error and guided by our intuition. More systematic approaches to parameter tuning could be adopted with additional time availability.




(a) LQR result (b) FL + LQR result

Fig. 3: Simulation plots with sine waves as reference.




(a) LQR result (b) FL + LQR result

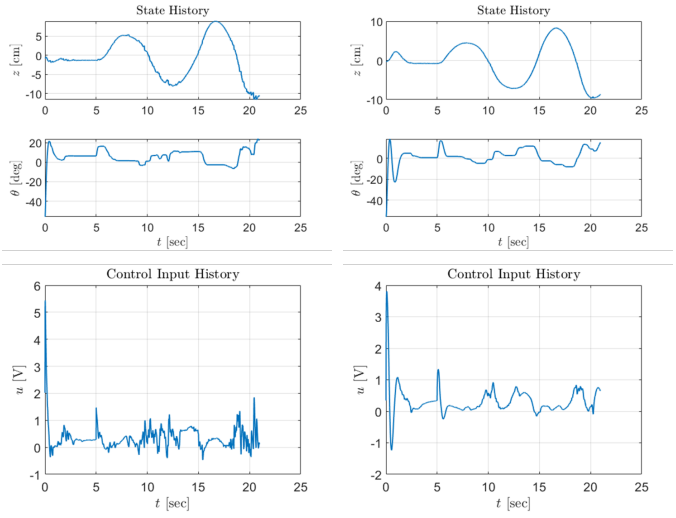
Fig. 4: Simulation plots with square waves as reference.

We observed a significant degree of variability in the performances across different experimental set ups. In fact, switching testbed requires retuning most control parameters. Carefully cleaning the beam guides and sphere decreases the measurement noise.

It is also interesting to notice that both our approaches rely on an LQR for parameter tuning, but the parameters used differ significantly. This might be due to the fact that FL cancels the nonlinearity, while the plain LQR is a first order approximation.

TABLE II: Hardware Performance

| Reference | Controller | Tracking Cost | Energy Cost | Total Score |
|---|---|---|---|---|
| Evaluation Trajectory | LQR | 0.35 | 1.73 | 2.08 |
| | FL + LQR | 0.19 | 1.44 | **1.63** |




(a) LQR result (b) FL + LQR result

Fig. 5: Hardware experiment plots

## VI. Conclusion

We evaluated LQR and FL + LQR on both simulation and hardware. In simulation, their total performance was comparable: LQR was a little bit better on square-wave references, which has large jumps in trajectory, while FL + LQR performed better on sine-wave references, which has smooth changes throughout the trajectory. Approximate feedback linearization cancels dominant nonlinearities, yielding smoother control inputs for FL + LQR, whereas LQR reacts aggressively to large errors—reducing them quickly but with larger steady-state error when deviations are small (Fig. 3). In the control input plots, LQR control spikes to nearly 10, compared to about 1.5 for FL + LQR.

In hardware, however, those sharp LQR controls proved detrimental: the high and sharp changing inputs increased both tracking and energy costs, degrading overall performance. FL + LQR's smoother inputs required less energy and produced fewer oscillations and noise, resulting in superior tracking

(Fig. 5).

We also encountered significant model–plant mismatch during the hardware experiments. —most notably the EKF's measurement-noise gain. Real sensors proved far less precise than their simulated counterparts, therefore we had to increase the noise gain ten times in hardware. Similarly, the effect of integral control was powerful in the hardware experiments. In simulations, the integral term had little impact on performance. In hardware, a steady-state error persisted despite extensive tuning; adding and adjusting an integral gain effectively eliminated that bias.

The hardest part of the hardware experiments was that the parameters were extremely sensitive to the experiment bench. Whenever we switched benches during lab sessions, we had to retune everything—most critically, the measurement-noise. It would be nice to stick to the same bench during lab sessions and commit to that bench. Even on a single bench, results varied because we manually placed the ball at the beam's center. One simple trick that improved repeatability was wiping the ball and beam with a microfiber cloth before each trial. This reduced oscillations in measurement noise and noticeably enhanced performance.

## REFERENCES


[1] H. K. Khalil, "Extended high-gain observers as disturbance estimators," *SICE Journal of Control, Measurement, and System Integration*, vol. 10, no. 3, pp. 125–134, 2017.

[2] M. Mueller, "C231b lecture notes."

[3] F. Borrelli, "C231a lecture notes."

[4] M. Arcak, "C222 lecture notes."


## APPENDIX

TABLE III: Ball and beam parameters

| Parameters | Values |
|---|---|
| $r_g$ | 0.0254 m |
| $L$ | 0.4255 m |
| $g$ | 9.81 m/s$^2$ |
| $K$ | 1.5 rad/sV |
| $\tau$ | 0.025 s |

TABLE IV: Parameters used in simulation

| Parameters | Values |
|---|---|
| $\hat{x}_m(0)$ | $[-0.19, 0, 0, 0]$ |
| $P_m(0)$ | $\mathrm{diag}([0.05, 0, 0.01, 0])$ |
| $\Sigma_{vv}$ (Process noise) | $\mathrm{diag}([0.05, 0.001, 0.0175, 0.00175])$ |
| $\Sigma_{ww}$ (Measurement noise) | $\mathrm{diag}([0.05, 0.0175])$ |
| $Q$ (LQR) | $diag([1000, 500, 1, 1])$ |
| $R$ (LQR) | 0.1 |
| $K_I$ (LQR) | 0.1 |
| $Q$ (FL + LQR) | $diag([800, 50, 1, 1])$ |
| $R$ (FL + LQR) | 0.3 |
| $K_I$ (FL + LQR) | 0.0 |

TABLE V: Parameters used in hardware experiment

| Parameters | Values |
|---|---|
| $\hat{x}_m(0)$ | $[0, 0, 0, 0]$ |
| $P_m(0)$ | $\mathrm{diag}([0.01, 0.01, 0.01, 0.01])$ |
| $\Sigma_{vv}$ (Process noise) | $\mathrm{diag}([0.01, 0.05, 0.02, 0.2])$ |
| $\Sigma_{ww}$ (Measurement noise) | $\mathrm{diag}([10, 1])$ |
| $Q$ (LQR) | $diag([1000, 100, 0.1, 1e-10])$ |
| $R$ (LQR) | 0.4 |
| $K_I$ (LQR) | 3.0 |
| $Q$ (FL + LQR) | $diag([1000, 100, 0.1, 1e-10])$ |
| $R$ (FL + LQR) | 0.06 |
| $K_I$ (FL + LQR) | 2.0 |