# Controlling a Furuta Pendulum System

Cuesta Bret, Cristina and Langner, Alexander

Group - FP12

April 23rd, 2025

***Abstract*—The main goal for this report is to design a full-state feedback controller for a Furuta pendulum system in a downward configuration using the LQR method. The controller will be tested and compared to a manually tuned controller, as well as to other LQR controllers designed with different Q/R matrices. The Furuta pendulum system is commonly used when teaching control theory due to its complex dynamics and nonlinearity. A popular solution for finding the optimal gains for this type of system is LQR control, in which the values for two matrices (Q & R) are chosen ahead of time based on the states and/or inputs the user wants to prioritize. After testing the performance of an LQR controller which placed equal importance on platen and pendulum angles against that of a controller that was manually tuned with the same philosophy, the LQR controller performed slightly better in all aspects: the platen RMS was 2.8% smaller, the rise time was 0.22% smaller, and the command RMS was 15.8% smaller than the manual method. In additional tests, the Q matrix was modified to prioritize pendulum control more strongly, and so the maximum pendulum angle was able to be lowered by 69%, despite a higher platen RMS value and a longer rise time. The R matrix was also modified, resulting in a 1600% increase in command RMS when lowered by a factor of 0.1 and a 13.7% increase when raised by a factor of 10 due to overall less efficient motor effort management.**

***Index Terms*—Gains, Impulse Response, LQR, Shannon**

## I. Introduction

In regards to control theory, there is perhaps no system more well known than the Furuta pendulum, which is characterized by a motor-driven arm that rotates in the horizontal plane and is rigidly connected to another arm that can freely rotate in the vertical plane (Fig. 1). Whether the free arm is being balanced in an upward position or carried along in a downward position, trying to control this system with any level of precision can be quite challenging due to the complex dynamics that stem from the blend of gravitational and centripetal forces. With an accurate model of the system, however, it is possible to obtain an optimal controller through a variety of potential methods. In this report, a full-state feedback controller will be designed for this system using a model of a "pendulum-down" Furuta pendulum system (Fig. 2) obtained from known system dynamics [1] [2], and then its performance will be tested against that of other controllers.

In full-state feedback control, a complete model of a system's dynamics is used to find the optimal values for the gains of that system given a set of desired performance metrics. In this case, the gains $k_1$, $k_2$, $k_3$, and $k_4$ correspond to the angle of the platen (driven arm), $\theta_1$, the angle of the pendulum (free arm), $\theta_2$, the rotational velocity of the platen, $\dot{\theta}_1$, and the rotational velocity of the pendulum, $\dot{\theta}_2$, respectively. These values are used to find the amount of control effort needed, $u$, in real time (1). Usually, finding these gains involves using pole-placement methods, but in the case of a system with multiple angles such as this, it might be simpler to use a linear-quadratic regulator (LQR) approach. In the simplified LQR objective function (2), the value for $J$ is minimized with respect to the input vector, $\underline{u}$, which in this case is one value. $Q$, the matrix of weights for the state vector $\underline{x}$ (the four aforementioned $\theta$ values), and $R$, the matrix of weights for the input vector, are chosen ahead of time based on which states or inputs the engineer wants to prioritize controlling [3]. If more emphasis should be placed on controlling the first two states, for example, then the first two values of the diagonal $Q$ matrix should be larger than the others. Once the $Q$ and $R$ values have been chosen, the optimal gains can be found using engineering software that implements the Riccati differential equation [4].

$$u(t) = -\begin{bmatrix} k_1 & k_2 & k_3 & k_4 \end{bmatrix} \begin{bmatrix} \theta_1 \\ \theta_2 \\ \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix} \tag{1}$$

$$J = \int_{t_0}^{t_f} \left( \underline{x}^T Q \underline{x} + \underline{u}^T R \underline{u} \right) dt \tag{2}$$

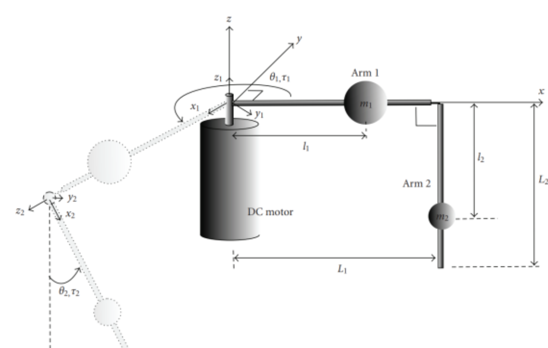


Fig. 1. Standard schematic for a Furuta pendulum system [1].

$$\begin{Bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \ddot{\theta}_1 \\ \ddot{\theta}_2 \end{Bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1.2940 & -9.2376 & 0.06205 \\ 0 & -53.963 & 6.5932 & -2.5876 \end{bmatrix} \begin{Bmatrix} \theta_1 \\ \theta_2 \\ \dot{\theta}_1 \\ \dot{\theta}_2 \end{Bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 9.2376 \\ -6.3133 \end{bmatrix} V$$

$$\begin{Bmatrix} \theta_1 \\ \theta_2 \end{Bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \begin{Bmatrix} \theta_1 \\ \theta_2 \\ \dot{\theta}_1 \\ \dot{\theta}_2 \end{Bmatrix} + \begin{bmatrix} 0 \\ 0 \end{bmatrix} V$$

Fig. 2. Final enumerated state-space equations for the pendulum system in the "pendulum-down" configuration [2].

Simply put, the goal of this report is to use the LQR method to design a full-state feedback controller for a Furuta pendulum system in the "pendulum-down" configuration. Along the way, a manual-tuning method will also be tested and compared to the LQR method. Finally, the effects that changing the values for $Q$ and $R$ have on the system will be observed and discussed.

## II. Procedure

*Hardware and Software*

The hardware setup for this lab includes assembling the motor/flywheel system using the "Out Of The Box" Furuta Pendulum kit as performed in Lab 1 [5] with the addition of the pendulum attachment as seen in Fig. 1. The resulting system should be a DC motor/flywheel with a spinning platen, a base, a clamp, and a pendulum encoder to record the platen angle. Next, a LabVIEW Virtual Instrument (VI) was programmed. A base VI was given and used as a starting point. First, the platen rotation error was calculated by subtracting the desired and actual rotation angles. The "MathToolHigherOrder" subVI was used to compute the numerical derivative (over 7 points) of the platen rotation error, the actual platen rotation, and pendulum rotation. The pendulum rotation was filtered using a lowpass Butterworth filter using a sampling frequency of 1,000 and a cutoff frequency of 10,000. Finally, to assemble the LQR (1) control, four gains, $k_1$, $k_2$, $k_3$, and $k_4$, were created as user inputs. The gains were multiplied by the platen rotation error, the pendulum rotation error, the platen velocity error, and the pendulum velocity error respectively and added together to give the output command signal.

*State-Space Equations*

To perform this lab, two state-space equations needed to be found to describe the dynamics of the system. These equations were then required to calculate the $Q$ and $R$ matrices needed for LQR method and to obtain the gains that the user would input into the VI.

*A. System Dynamics*

In order to determine the state-space equations, the motor specifications and system dynamics had to be obtained by following the procedure from Lab 3 [6]. Using calipers, the distance from the platen center of rotation to the center of mass of the pendulum ($L_1$) was measured, as well as the distance between the point of rotation to the longest edge of the pendulum ($d_2$). The distance between the point of rotation to the center of mass of the pendulum was also found ($d_1$), as well as the total length of the pendulum ($d_3$). The pendulum was then weighed on a lab scale ($m_{pend}$), and so was the additional weight from the screw and two nuts ($m_{nuts}$). The moment of inertia for the pendulum arm, $\hat{J}_2$, was calculated, along with the moment of inertia for the point of rotation arm, $\hat{J}_o$, and the base friction coefficient, $b_1$ [6]. The square wave used to find the data had a command signal of 70 with a frequency of 0.1 Hz and amplitude of 300 degrees. The DC gain, $K_{DC}$, and the time constant, τ, were obtained for the new system following the procedure from Lab 1 [5] and using the square wave data saved.

The pendulum was calibrated to find $b_1$ using the VIs provided. The exponential decay and damping of the system were evaluated by performing a free-swing experiment with the pendulum. The state-space equations (Fig. 2) were set up using the parameters obtained and an impulse response was graphed using MATLAB.

*Manually Tuning the Pendulum*

To evaluate the system's performance and how the gains worked, several manual tuning simulations were conducted. First, the pendulum was immobilized in the downward position with tape. This was done so as to isolate and focus on the platen rotation. Then the VI was set so that the platen tracked a square wave at a frequency of 0.25 Hz with a ±60º amplitude. $k_2$ and $k_4$ were set to 0 since those gains control the pendulum and $k_1$ and $k_3$ were manually tuned so the wave was as close to the desired amplitude as possible, minimizing the error. Then the tape was removed, and the same gains were tested with the pendulum free to move. The system performance was recorded for both test cases and the data was saved to Excel. Following this, all the gains were manually tuned with the objective of reducing the pendulum rotation as much as possible while still tracking the wave at the original parameters. The data for the third test was also recorded.

*LQR Control Method to Find Gains*

Using MATLAB, the state-space equations, and the lqr() function, the gains necessary were found based on user input for matrices Q and R. Four different tests were conducted, the R matrix was left at 1 for the first two tests and the Q matrix was tested for different values while assessing system performance and stabilization of the pendulum. The wave tracked had a frequency of 0.25 Hz and amplitude of ±60º. The first test focused on platen and pendulum displacement/angle costs, this was achieved by increasing the first two values of the Q matrix, thus increasing $k_1$ and $k_2$. The Q values obtained were 25 which means that more emphasis was put in getting the desired platen angle and decreasing the pendulum rotation. The second test focused on getting smaller pendulum angles by modifying all the gain costs and increasing the speed costs. The Q matrix obtained (Table I) placed more emphasis on the platen angle and decreased the platen and pendulum velocities. Finally, the last two tests used the first Q matrix but changed the R matrix to 0.1 and to 10, thus obtaining a new set of gains.

*Data Manipulation*

For all the tests conducted, a plot of platen rotation vs. time was used to find the rise time. To visualize the oscillation of the pendulum, a pendulum rotation vs. time graph was also plotted for all scenarios. The Root Mean Square (RMS) of the platen rotation and command signal were also calculated. Performance was evaluated using these RMS values and the platen rise time. The pendulum oscillation angle will also be used to determine the performance. A smaller pendulum angle and smaller RMS values would mean better system performance.

## III. Results

After recording all the data from each test, the results were put together to characterize the performance of the system and evaluate pendulum stability. The impulse response for the state space equations (Fig. 2) can be seen in Fig. 3.

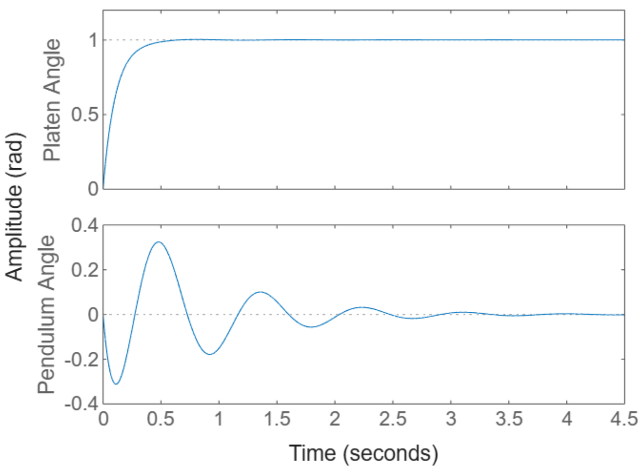


Fig. 3. Impulse Response for the state-space equations (Fig. 2). Top graph $\Theta_1$, bottom graph $\Theta_2$.

Table I shows the gains used for all tests as well as the values for the Q and R matrices. The Q matrix is a diagonal matrix, so the values in the table correspond to the diagonal.

TABLE I
Q and R Matrices and Corresponding Gains

| Test | Q | R | $k_1$ | $k_2$ | $k_3$ | $k_4$ |
|---|---|---|---|---|---|---|
| 1 | N/A | N/A | 12 | 0 | 0.11 | 0 |
| 2 | N/A | N/A | 12 | 0 | 0.11 | 0 |
| 3 | N/A | N/A | 5 | -6 | 0.11 | -0.1 |
| 4 | [25,25,1,1] | 1 | 5 | -3.6095 | 0.8514 | -0.1145 |
| 5 | [25,35,10,2.5] | 1 | 5 | -4.8570 | 2.0816 | -0.3390 |
| 6 | [25,25,1,1] | 0.1 | 15.811 | -14.277 | 3.5939 | -0.0103 |
| 7 | [25,25,1,1] | 10 | 1.5811 | -0.5770 | 0.2162 | -0.0439 |

N/A = LQR method not used for the test

The graphs for the pendulum rotation versus time for tests 4-7 can be seen in Fig. 4-7.

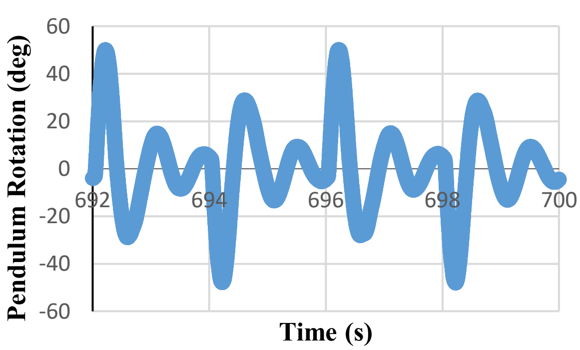


Fig. 4. Pendulum Rotation (deg) vs Time (s) for test 4.

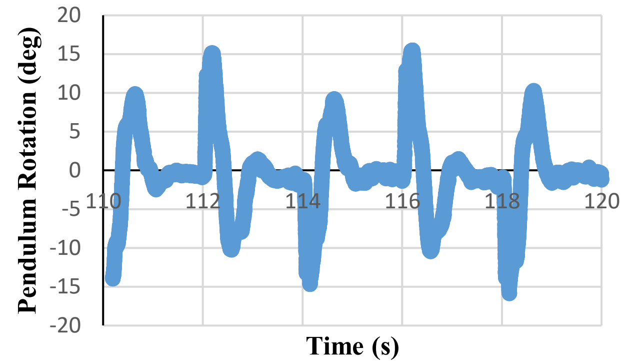


Fig. 5. Pendulum Rotation (deg) vs Time (s) for test 5.

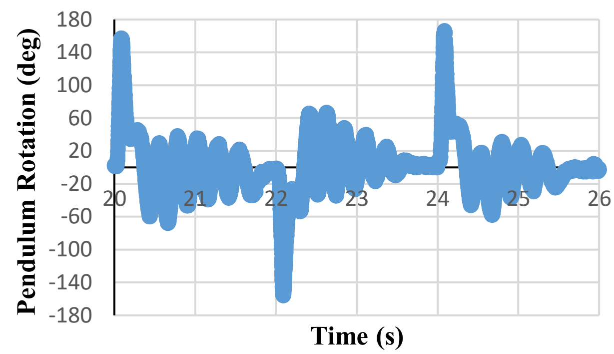


Fig. 6. Pendulum Rotation (deg) vs Time (s) for test 6.

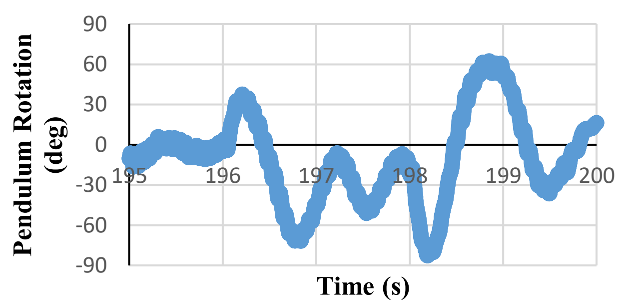


Fig. 7. Pendulum Rotation (deg) vs Time (s) for test 7.

The RMS values for the command signal and the platen rotation error as well as the rise time for the platen were calculated and shown in Table II.

TABLE II
RMS Values and Rise Time

| Test | Platen RMS | Command RMS | Rise Time |
|---|---|---|---|
| 1 | 28.1145 | 365.396 | 0.1490 |
| 2 | 24.6732 | 322.044 | 0.1502 |
| 3 | 38.5794 | 205.569 | 0.6032 |
| 4 | 37.5086 | 173.141 | 0.6019 |
| 5 | 47.0076 | 1566.29 | 1.7724 |
| 6 | 61.1974 | 2964.26 | 0.6111 |
| 7 | 51.8487 | 196.829 | N/A |

N/A = platen didn't reach desired values

## IV. Discussion

To review, the objective of this report was to design and test a full-state feedback controller for a Furuta pendulum using the LQR method, as well as to compare its performance to that of a manually tuned controller and other LQR controllers that had different $Q$ and $R$ values. The gains and the square-wave platen performance for each controller, as well as the $Q$ and $R$ matrices and pendulum rotation graphs for the LQR controllers, were all recorded and detailed in the Results section.

*Impulse Response Test*

Before performing the controller tests, an impulse response simulation of the system was generated in MATLAB using the state-space equations of the system (Fig. 3). Physically, the data makes sense. It is expected that the pendulum, an underdamped system, would be pushed along and then continue rocking back and forth even after the platen has stopped, which is exactly what is shown in the graph. This is a strong indication that the values for the state-space equations of the system (Fig. 2) are accurate.

*Manually Tuned Test Cases*

Looking at the platen performance of the first two test cases (taped vs. untaped pendulum) in Table II, it is apparent that the platen RMS and command RMS were 12.2% and 11.9% lower respectively for the untaped test. It would seem that the natural swinging of the pendulum and the inertia it carried helped the platen reach its destination with more energy efficiency, even if the change in rise time was negligible. Comparing the second and third tests (only $k_1$ and $k_3$ vs. all gains tuned): attempting to limit the motion of the pendulum seems to have forced the platen to move more slowly, as indicated by the 56% increase in platen RMS and 302% boost in rise time. This was predicted to happen when the value for $k_1$ was intentionally lowered from 12 to 5, as shown in Table I. To put it another way, limiting the angle of the pendulum was prioritized over getting good platen performance. Still, it took less motor effort to get there, with the command RMS dropping by 36%.

*Manual Tuning vs. LQR*

Between the third and fourth test cases (manually tuned vs. LQR control), the “optimal” values for $k_2$ and $k_3$ were found to be significantly different. As a result, the platen performance improved, though not by much, with a 2.8% decrease in platen RMS, a 0.22% drop in rise time, and most significantly, a 15.8% decrease in command RMS. It’s worth noting here though that the values for $Q$, [25, 25, 1, 1], were chosen to prioritize the platen and pendulum angles equally, and to an extent, the same philosophy was used during the manual tuning method. A shift in priorities for either of these methods would undoubtedly lead to different results, as shown in test cases 5-7.

*LQR Experimentation*

In order to observe the effects that changing $Q$ and $R$ would have on the system, some additional tests were also conducted. First, in an attempt to limit the pendulum angle deviations, the $Q$ matrix was changed from [25, 25, 1, 1] to [25, 35, 10, 2.5] so that more importance would be placed on controlling the angle of the pendulum (tests cases 4 and 5), specifically by increasing the second value from 25 to 35, but also by increasing the third and fourth values, which are related to velocity. The result was a 20% increase in platen RMS, a 194% increase in rise time and a 69% decrease in the maximum pendulum angle (Fig. 4 and 5), which is pretty much exactly what the intention was. It is also worth mentioning that the command effort shot up by more than 800%, which is probably related to the fact that in such a large effort to prioritize the pendulum angle, the motor would rapidly change its rotation speed to match the requirements at the time, which is reflected in the “shakiness” of the pendulum (Fig. 5).

After experimenting with the $Q$ matrix, the $R$ matrix was then modified while keeping the $Q$ values from test case 4 consistent, and the response was recorded (test cases 6 and 7). Decreasing the $R$ value by a factor of 0.1 resulted in much less importance being placed on limiting the command effort. Thus, it increased by a whopping 1600%, and the pendulum angle became highly unstable due to the controller constantly trying to correct itself (Fig. 6). Increasing the $R$ value by a factor of 10 actually ended up increasing the command effort as well by 13.7%. The reason for this is that since the system moved so slowly, it never even ended up reaching the desired platen rotation angle and thus it never got a break (Fig. 7), while the initial system did reach its desired angle and so the motor could rest for half the time. In neither case did any aspect of the platen performance improve.

Ultimately, changing the $Q$ and $R$ values does not inherently lead to “better” or “worse” results; it simply leads to different optimal results. The matrix values used in test case 4 were designed to equally prioritize responses of both arms, which was the intention, and so given the option, we would not change those values.

*Limitations and Potential Improvements*

As far as analytical limitations go, both the moment of inertia for the base of the pendulum and any Coulomb forces acting on the system (e.g. dry friction) were ignored. With further testing and some simple modifications to the procedure, the addition of these factors have the potential to improve the accuracy of the values in the state-space equations (Fig. 2). For hardware, all of the 3D printed components could be made out of stiffer material such as metal, and optimizing the circumference of the track for the ball bearings could reduce frictional forces or energy losses that impact the system. Finally, measuring the angular velocity values instead of calculating them using derivatives could lead to better controller performance overall.

## References

[1] B. S. Cazzolato and Z. Prime, “On the Dynamics of the Furuta Pendulum,” *Journal of Control Science and Engineering*, 2011.

[2] C. Cuesta Bret, “Characterization of a Furuta Pendulum System,” University of Florida, unpublished.

[3] J. Brooks, “L18_03_24_25.pdf”, University of Florida, unpublished.

[4] J. Brooks, “L19_03_26_25.pdf”, University of Florida, unpublished.

[5] C. Cuesta Bret and A. Langner, “Models of a Motor/Flywheel System,” University of Florida, unpublished.

[6] J. Brooks and S. Ridgeway, “EML 4314C - SP 2025 - Lab 3 Assignment.pdf”, University of Florida, unpublished.