Course Name: Reinforcement Learning for Engineers

Course Number and Section: **16:332:515:01**

Instructor: Professor Zoran Gajic

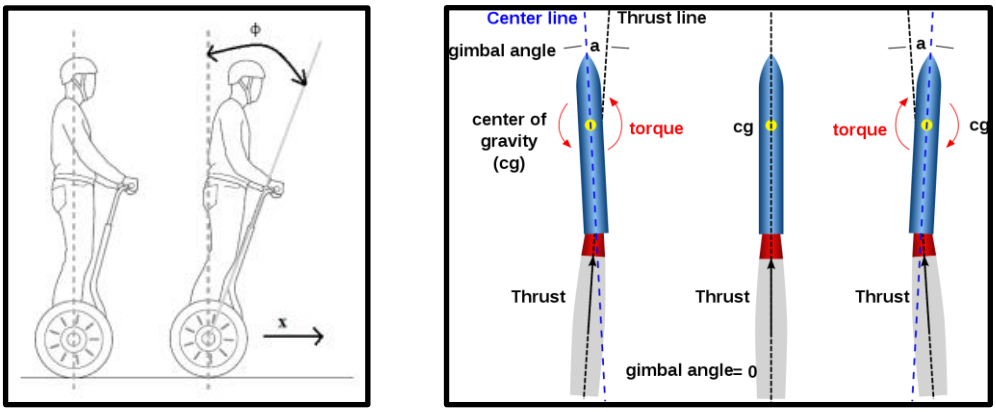
Date Submitted: 11/10/2023

Submitted by: Dhruv Rana (RUID: 209001759)

Control Balance System Using LQR

Introduction

Have you ever tried balancing an object, such as balancing a stick or pencil on your hand? These mechanical systems where the center of mass is balanced over a pivot point are called *balance systems*. A few of the typical balancing systems found in real-world applications are shown in Figure 1. A motorized platform on the Segway Personal Transporter (Figure 1a) stabilizes the person standing on it. The conveyance machine moves along the ground when the rider leans forward, yet it stays upright. Another illustration is a rocket (Figure 1b), where the body of the rocket is stabilized above it using a gimballed nozzle at the bottom of the rocket. In the dynamic realm of personal transporters and rocketry, achieving and maintaining equilibrium is a paramount challenge. The delicate dance between stability and maneuverability is a key determinant of success in these domains. At its core, control theory seeks to regulate the behavior of such systems to achieve desired outcomes. The challenge lies in navigating the inherent complexities and disturbances that can disrupt a system's trajectory. Control balance systems tries to rise to this challenge by counteracting disturbances, and modulating system parameters to maintain stability.



(a) Segway (b) Gimbal Trust

Figure 1: Balance systems. (a) Segway Personal Transporter [1], and (b) Gimbal Trust Rocket [2].

System Modeling

A simplified version of a balance system can be represented as an inverted pendulum on a cart (Figure 2) [3]. To model this system, we choose state variables that represent the position and velocity of the base of the system, x and $\dot{x}$, and the angle and angular rate of the structure above the base, θ and $\dot{\theta}$.

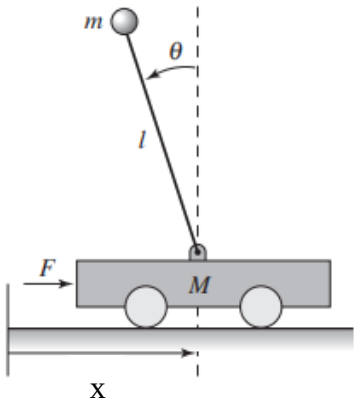


Figure 2 : Cart–pendulum system [3]

Now we can rewrite the dynamics of the system in state space form by defining the state as x = (x $\dot{x}$ θ $\dot{\theta}$), the input as u = F, and the output as y = (x $\dot{x}$ θ $\dot{\theta}$). Typically, we can only measure the cart position and pendulum angle in this type of system. However, for simplicity, we will assume that all states are measurable. Also, I will skip the derivation for the linearized equations of motion represented in state-space form, as it is well known and already derived by multiple other literatures, such as "*Feedback Systems: An Introduction for Scientists and Engineers*" by Åström and Richard.

$$\begin{bmatrix} \dot{x} \\ \ddot{x} \\ \dot{\theta} \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & \frac{-(I+m\,l^2)b}{I(M+m)+M\,m\,l^2} & \frac{m^2\,g\,l^2}{I(M+m)+M\,m\,l^2} & 0 \\ 0 & 0 & 0 & 1 \\ 0 & \frac{-m\,l\,b}{I(M+m)+M\,m\,l^2} & \frac{m\,g\,l(M+m)}{I(M+m)+M\,m\,l^2} & 0 \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{x} \\ \theta \\ \dot{\theta} \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{I+m\,l^2}{I(M+m)+M\,m\,l^2} \\ 0 \\ \frac{m\,l}{I(M+m)+M\,m\,l^2} \end{bmatrix} F$$

$$y = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{x} \\ \theta \\ \dot{\theta} \end{bmatrix}$$

System Analysis

Open-Loop Impulse Response

We will begin by examining the open-loop response of the inverted pendulum system. Specifically, we will examine how the system responds to an impulsive force applied to the cart employing the MATLAB command `impulse`.

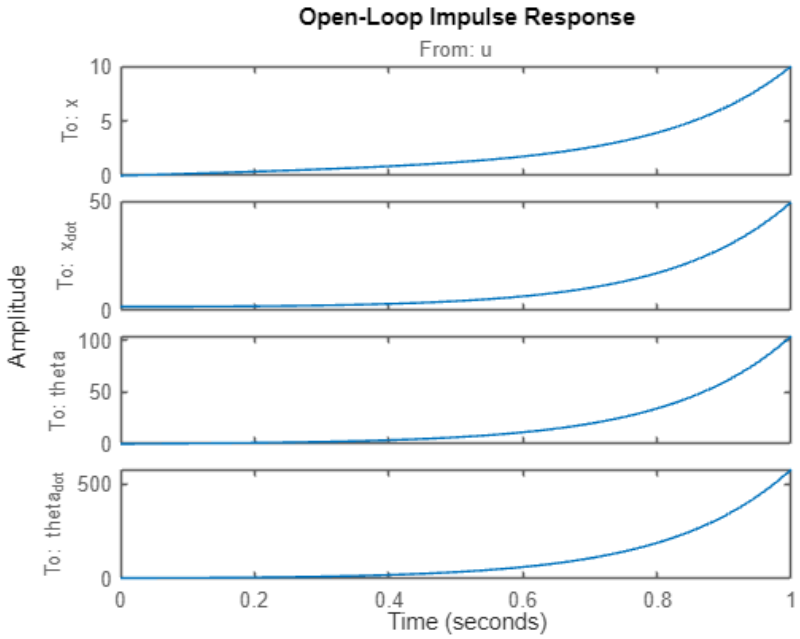


Figure 3 : Open-loop impulse response of the inverted pendulum system

As we can see from the plot, the system response is entirely unsatisfactory. In fact, it is not stable in an open loop. We can also see that the cart's position moves infinitely far to the right, though there is no requirement on cart position for an impulsive force input.

Root Locus

The poles of a system can tell us about its time response, and stability. We will specifically examine the poles and zeros of the system using the MATLAB function `rootlocus`.

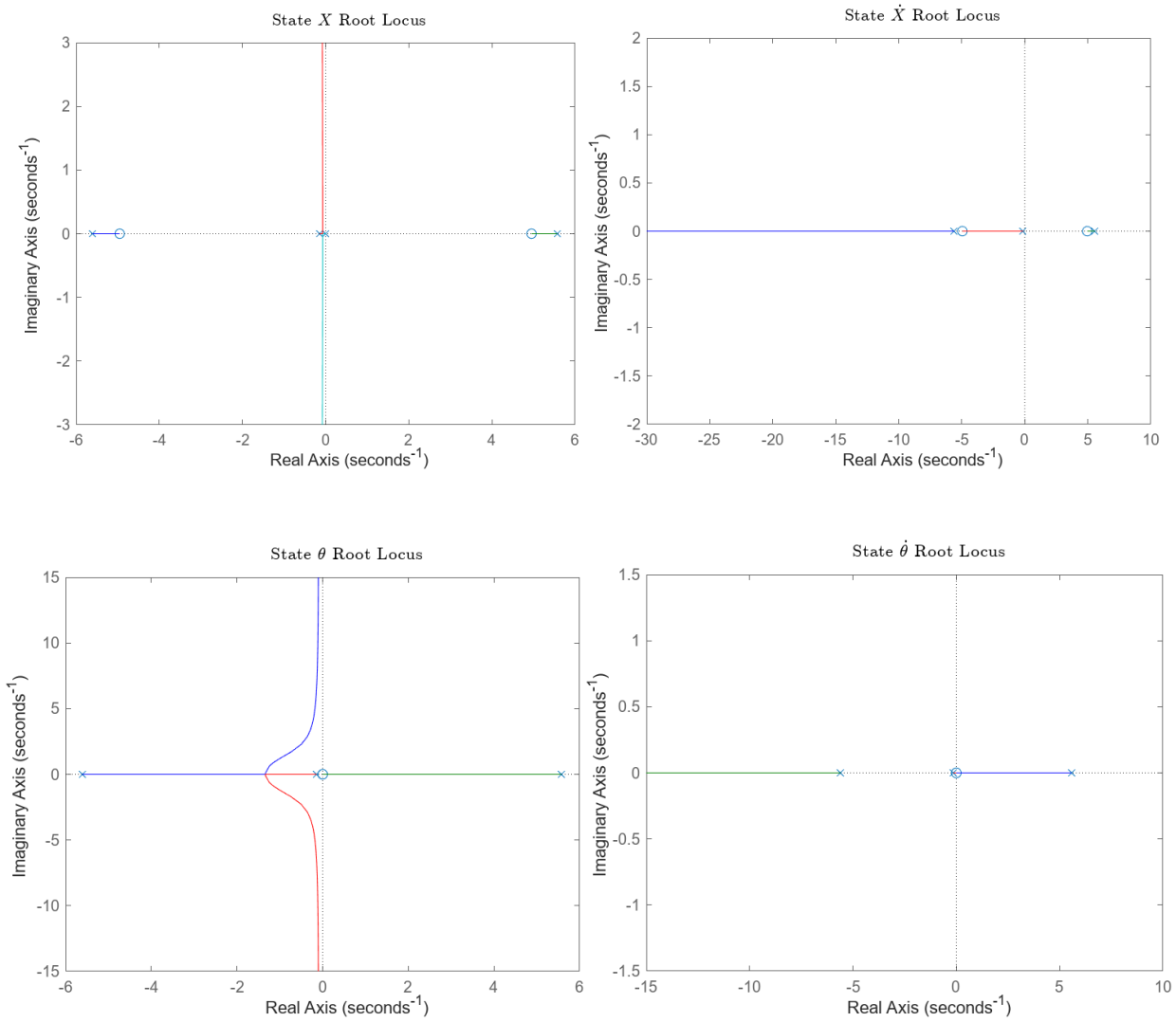


Figure 4 : Root Locus of the inverted pendulum system

The pole in the right half of the complex s-plane indicates that the system is unstable since the pole has a positive real part.

Open-Loop Step Response

Since the system has a pole with positive real part its response to a step input will also grow unbounded. We will verify this using the command `lsim` which can be employed to simulate the response of LTI models to arbitrary inputs. In this case, a 1-Newton step input will be used.

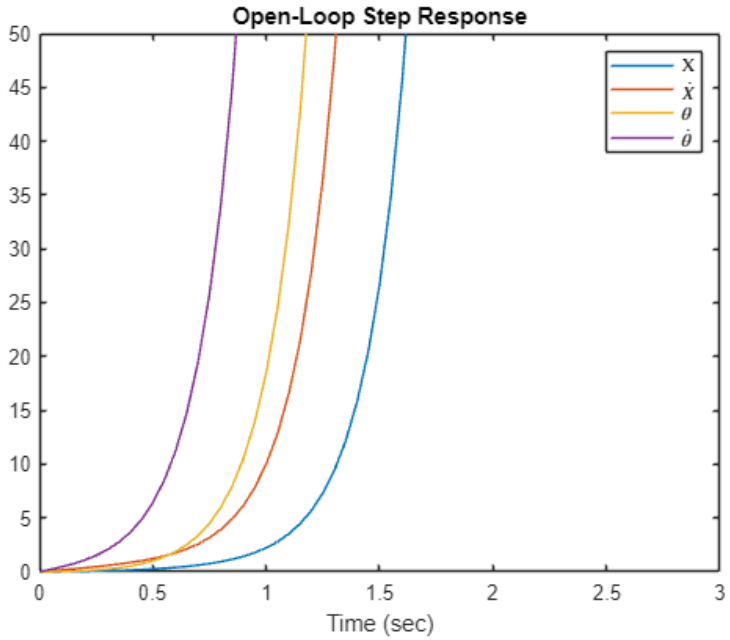


Figure 5 : Open-loop step response of the inverted pendulum system

The above results confirm our expectation that the system's response to a step input is unstable. It is apparent from the analysis above that some sort of control will need to be designed to improve the response of the system.

System Control

We can design several controllers that can control this system and meet our design requirements. For example, PID controllers, Pole placement, etc. However, PID can control only one state, and we would have to give up control over the other states. On the other hand, pole placement strategies try to find poles such that all the design requirements are met, but this strategy is hard to get an intuition on how the poles influence the system. Thus, we will use a linear quadratic regulator (LQR), which is similar to pole placement as it tries to find a matrix K for the full state feedback, but it gives greater control over the input and output of the system.

Controllability and Observability

Before we design our controller, we will first verify that the system is controllable. Satisfaction with this property means that we can drive the state of the system anywhere we like in a finite time. For the system to be completely state controllable, the controllability matrix must have a rank where the rank of a matrix is the number of linearly independent rows (or columns).

$$C = [\, B \mid AB \mid A^2B \mid \cdots \mid A^{n-1}B \,]$$

We can easily test for controllability using the MATLAB command `ctrb` to generate the controllability matrix and the MATLAB command `rank` to test the rank of the matrix. Indeed, we find that the controllability matrix is full rank. Therefore, we have verified that our system is controllable and thus we should be able to design a controller that achieves the given requirements.

$$O = \begin{bmatrix} C \\ CA \\ CA^2 \\ \vdots \\ CA^{n-1} \end{bmatrix}$$

Similarly, we can find the observability matrix using the MATLAB command `obsv` and check the rank to see if all the state variables in the system are observable for the full state feedback. After running the command, we find that observability matrix is full rank, and all the system states are observable.

Linear Quadratic Regulator (LQR)

The Linear Quadratic Regulator (LQR) is a well-known method that provides optimally controlled feedback gain that ensures the stability of the closed-loop and high performance of system. For the derivation of the linear quadratic regulator, we consider the linear system state-space representation in the form:

$$\frac{dx(t)}{d\tau} = Ax(t) + Bu(t), \qquad x(t) = x_0 \tag{1}$$

$$y(t) = Cx(t)$$

Where our goal is to minimize the input u given a quadratic performance criterion associated to the dynamic system.

$$J(x(t_0)) = \frac{1}{2}\int_0^\infty (x^T(t)Qx(t) + u^T(t)Ru(t))\,dt, \qquad Q \geq 0, \quad R > 0 \tag{2}$$

The solution to this dynamic optimization problem via successive iterations created by Vaisbord is given as follows [4]:

Step 0: Check that the optimal LQ controller exists, that is, check that (A, B) is controllable (or at least detectable) and $(A,\mathrm{Chol}(Q))$ is observable (or at least detectable).

Step 1: Initialization is required with an admissible (stabilizing) feedback control input, $u_1\big(x(t)\big) = -K_1x(t)$, such that the feedback (closed-loop) system is asymptotically stable, that is

$$\frac{dx_1(t)}{dt} = (A - BK_1)x(t), \qquad x_1(0) = x(0) = x_0, \qquad Re\{\lambda_i(A - BK)\} < 0, \quad i = 1,2,\dots,n \tag{3}$$

Step 2: Solve the algebraic Lyapunov equation

$$(A - BK_1)^T P_1 + P_1(A - BK_1) + Q + K_1^T R K_1 = 0 \quad \Rightarrow \quad P_1 \tag{4}$$

Note that the algebraic Lyapunov equation has a unique solution under assumption that the coefficient matrix is asymptotically stable, [5]. Evaluate the approximate performance criterion as

$$J_1(x(0)) = \frac{1}{2}x^T(0)P_1x(0) \tag{5}$$

and implement the following control to the system

$$u_2(x(t)) = -R^{-1}B^T P_1 x(t) = -K_2 x(t) \tag{6}$$

Evaluate the approximate state trajectories and the approximate system performance values

$$\frac{dx_2(t)}{dt} = (A - BK_2)x(t), \qquad x_2(0) = x(0) = x_0 \quad \Rightarrow \quad x_2(t) = e^{(A-BK_2)t}x(0) \tag{7}$$

$$J_2(x(0)) = \frac{1}{2}x^T(0)P_2 x(0), \qquad (A - BK_2)^T P_2 + P_2(A - BK_2) + Q + K_2^T R K_2 = 0 \tag{8}$$

Observe that

$$J_1(x(0)) = \frac{1}{2}x^T(0)P_2 x(0) \quad \geq \quad J_2(x(0)) = \frac{1}{2}x^T(0)P_2 x(0) \tag{9}$$

Step k: Repeat the process *k*-times until the convergence to the optimal control, optimal system trajectory and the optimal performance is obtained, solve iteratively the algebraic Lyapunov equations

$$(A - BK_k)^T P_k + P_k(A - BK_k) + Q + K_k^T R K_k = 0, \quad K_k = R^{-1}B^T P_{k-1}, \quad k = 3, \ldots \tag{10}$$

Observe that

$$J_1(x(0)) \geq J_2(x(0)) \geq J_3(x(0)) \geq \cdots \geq J_k(x(0)) \geq \cdots \geq J_{opt}(x(0)) = \frac{1}{2}x^T(0)P_{opt}x(0) \tag{11}$$

Results

By implementing the Vaisbord Algorithm in MATLAB, with the parmaeters listed in Appendix A with X0 = [1 -2 30^o 5^o] , $Q = I$, R $= 1$. We find that the optimal performance we can achieve after 7 iterations is J_{opt} = 13.7395.

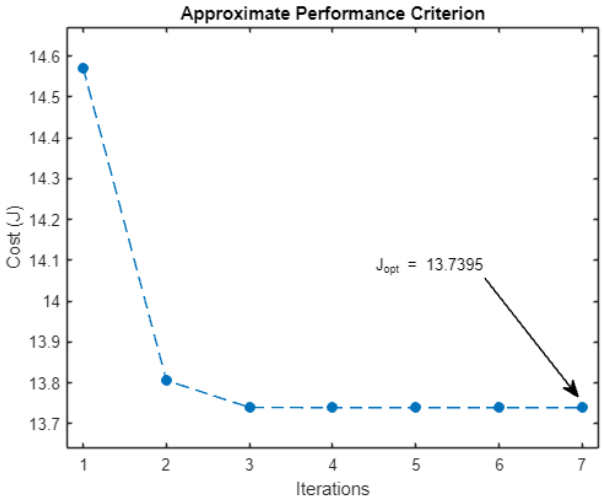


Figure 6 : Approximate performance Criterion

Iteration #	Performance Cost (J)
1	14.57146026972481
2	13.80599889418425
3	13.74002652583613
4	13.73945497636709
5	13.73945491762834
6	13.73945491762836
7	13.73945491762831

Table 1 : Approximate performance Criterion

From this we can also find the approximate trajectory of all the states. We can observe that all the states settle at zero (0) just as intended.

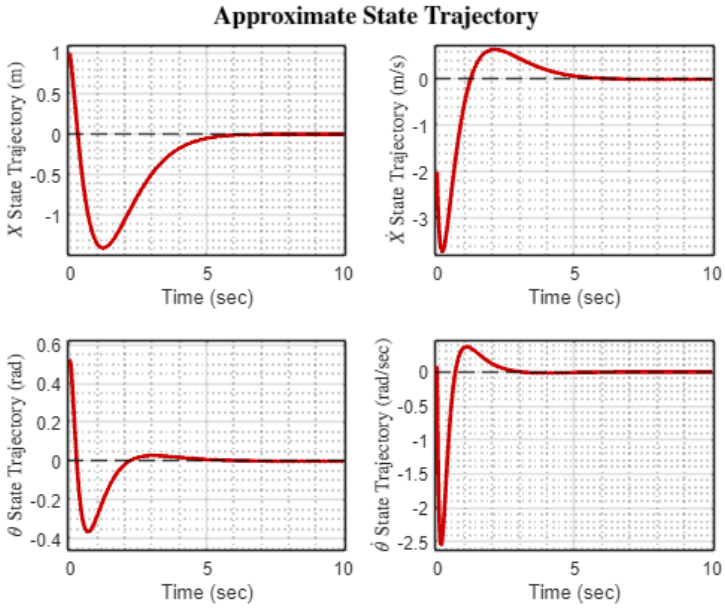


Figure 7 : Approximate State Trajectory

The input that led to this intended state trajectories is:

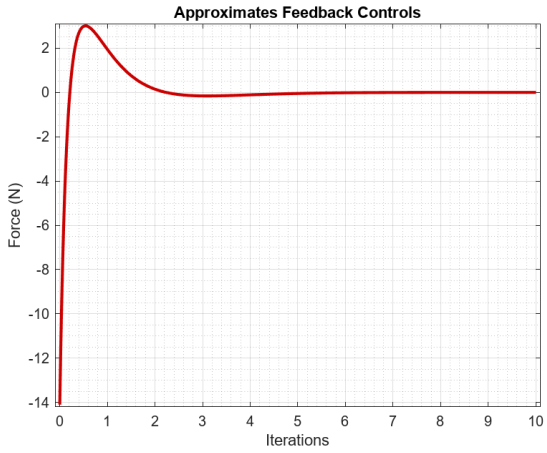


Figure 8 : Approximate Feedback Controls

After 7 iterations we find that the optimal gain K is :

$$K_{opt} = [-1.0000 \quad -2.0405 \quad 20.3838 \quad 3.9313]$$

Simulink System Simulation

We must remember that till now we have been operating on the linearized model of the inverted pendulum system. However, we can create the non-linear model using Simulink. Simulink has an add-on Simscape which allows the creation of rigid body systems and test them.

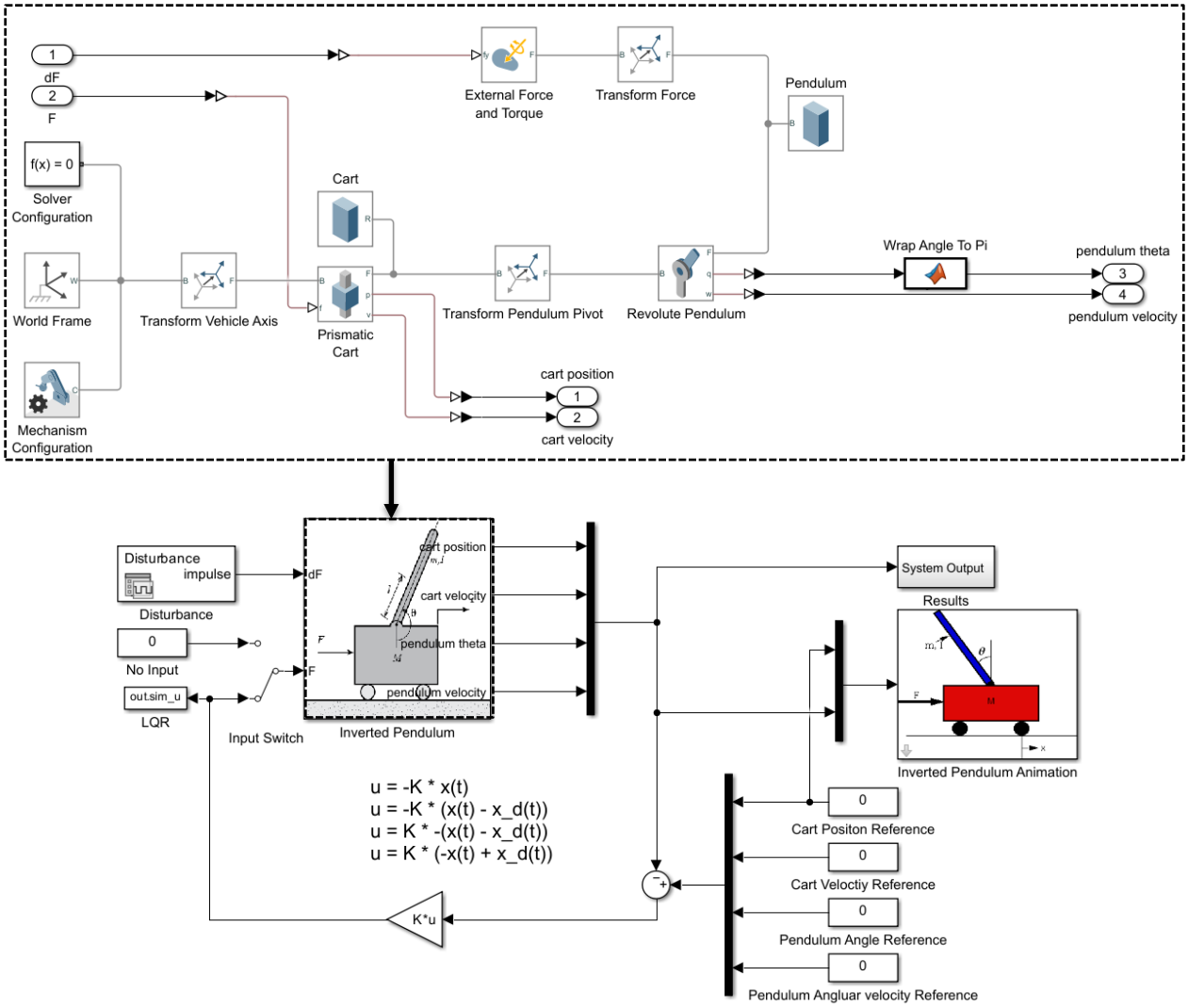


Figure 9 : Non-Linear Inverted Pendulum System w/ LQR

Running the simulation for 20 seconds and having an impulse disturbance at 11 seconds of 1 Newton. We got the following results.

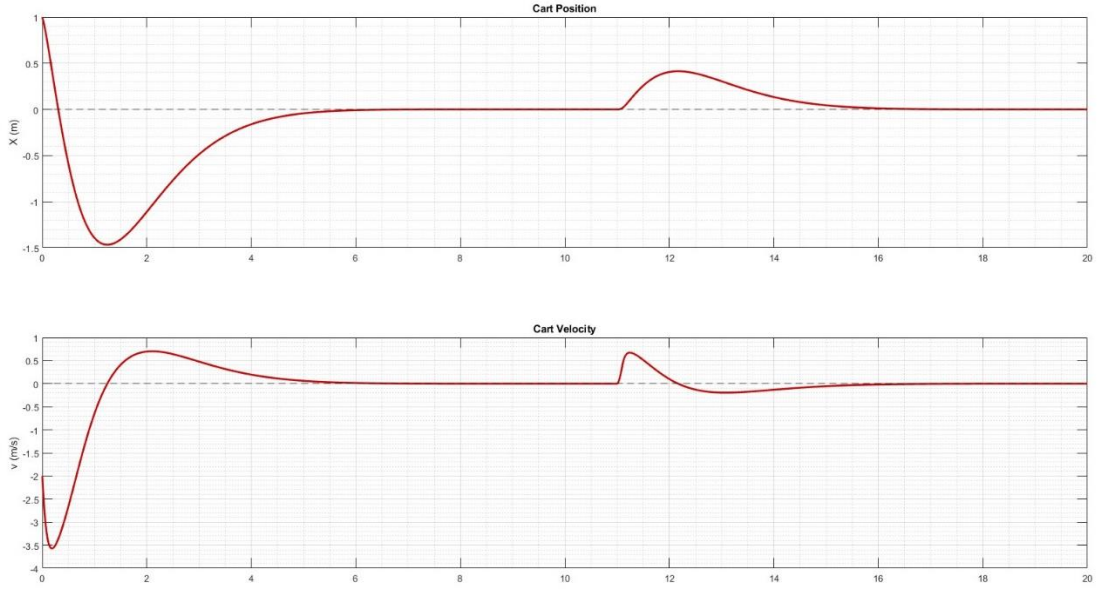


Figure 9 : Cart Information

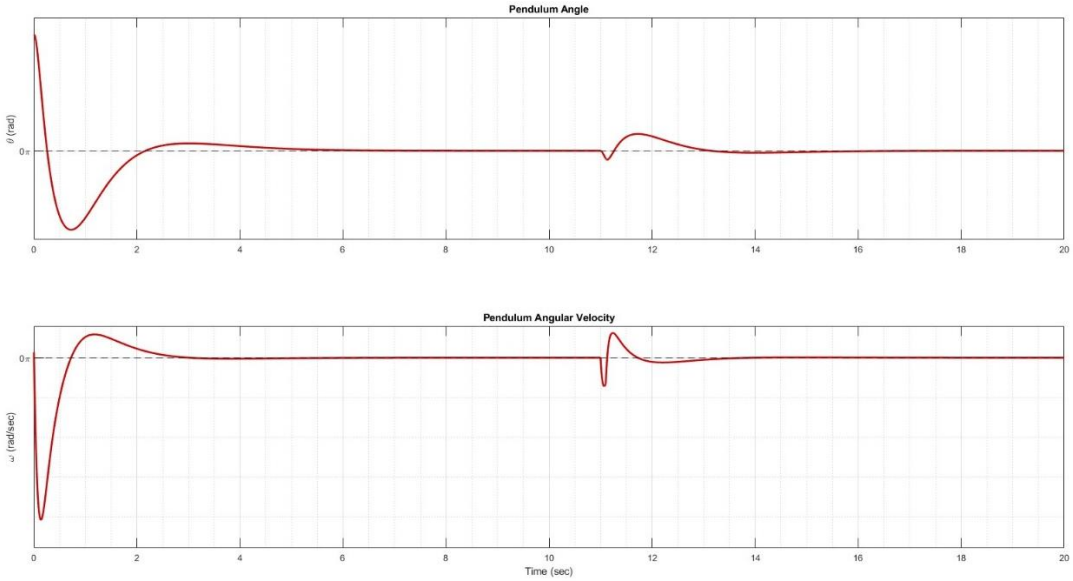


Figure 10 : Pendulum Information

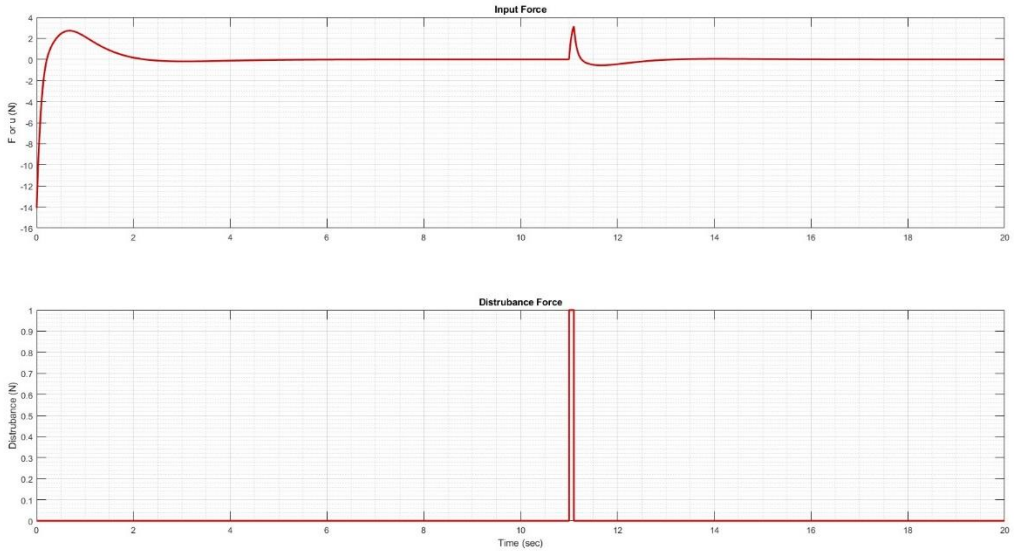


Figure 11 : System Input's

We can observe that the input force and state trajectories are very similar to their approximates up until 10 seconds. Not only that, but we can also see that the system has disturbance rejection and will go back to zero (0) after a disturbance in the system states.

Extension

I even added a reference tracking mechanism for the cart, so it goes to specified location and balances the pendulum there. Additionally, I even made a simple visualization of the system to understand what is really happening. All of this and the code can be found on my google drive (https://drive.google.com/drive/folders/1gSiAVmu7fHkg4wjYcDXRVpnMLrhNoQfe?usp=sharing). All you must do is download the MATLAB folder and run the file "Initialization_and_Setup.mlx", it will run all the tests and simulation mentioned in the report. Simulink will open and close several times to record the results. After all the simulations are done a message box will pop up and Simulink will open for your own experimentation.

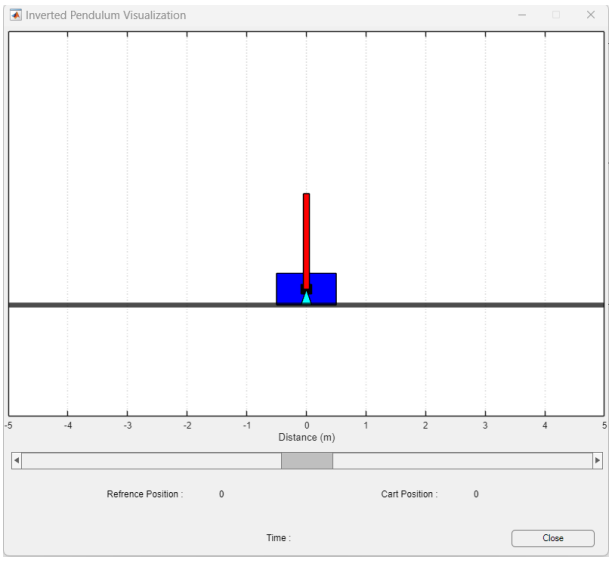


Figure 12 : Simulink System Animation

Conclusion

This paper presented a lab report on the design and implementation of a linear quadratic regulator (LQR) for controlling an inverted pendulum system. The paper first modeled the system using state-space representation and analyzed its open-loop response. The paper then applied the Vaisbord algorithm to find the optimal feedback gain matrix that minimizes a quadratic performance criterion. The paper also simulated the closed-loop system using Simulink and verified its stability and disturbance rejection properties. The paper demonstrated the effectiveness of the LQR method for achieving the desired performance of the balance system. The paper also extended the system to include reference tracking and visualization. The paper concluded that the LQR is a powerful and versatile technique for optimal control of linear systems.

Appendix A : Inverted Pendulum Model Parameters

(M)	Mass of the Cart	0.5 kg
(m)	Mass of the pendulum	0.2 kg
(b)	Coefficient of friction for cart	0.1
(l)	Length to pendulum center of mass	0.3 m
(L)	Length to pendulum	0.6 m
(I)	Mass moment of inertia of the pendulum	0.006 kg m^2
(g)	Gravity	9.8 m/s^2
(x)	Cart's Position	m
($\boldsymbol{\theta}$)	Pendulum's Angle	rad
(F)	Force applied on the cart	N

References

[1] Sevekari, Pranav et al. "Robust Control for Stable and Safe Performance of a Two Wheeled Human Transporter." IFAC-PapersOnLine 53, 616-621, 2020.

[2] "Gimbaled Thrust." *NASA Glenn Research Center*, 20 November 2018, https://www.grc.nasa.gov/www/k-12/rocket/gimbaled.html.

[3] Åström, Karl Johan, and Richard M. Murray. *Feedback Systems: An Introduction for Scientists and Engineers*. Princeton University Press, 2008.

[4] E. Vaisbord, "An approximate method for the synthesis of optimal control," Automation and Remote Control,Vol. 24, 1626-1632, December 1963.

[5] Z. Gajic, Lyapunov Matrix Equation in Systems Stability and Control, Academic Press, 1995 (Chapter 8, Section 8.1 Kleinman's Algorithm for Riccati Equation, pages 90-95).