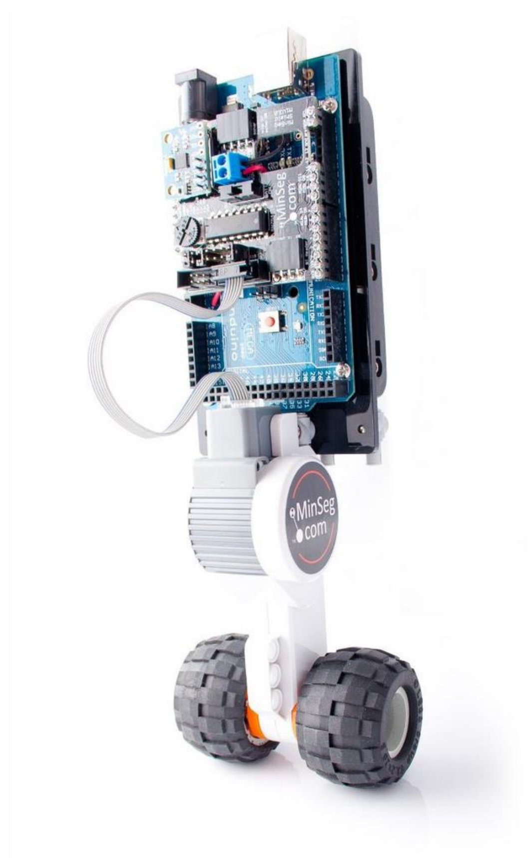
MinSeg Robot Control with Simulink and MATLAB



Nicholas Sandberg

San José State University, Mechanical Engineering Department

December 5, 2023

Table of Contents

Summary

The main objective of this project was to get the MinSeg robot to balance itself using a closed-loop feedback system. This was accomplished by first developing the equations of motion for each subsystem present in the MinSeg robot including mechanical and electrical elements by using the fundamental laws of physics. Once the equations were derived, a controller was designed and simulated in MATLAB based upon the parameters defined by the governing electromechanical equations. Once the control gains were calculated and fine-tuned using multiple different methods, the controller was physically implemented and tested on a live MinSeg robot. Furthermore, I experimented with various settings and parameters and observed the results to determine what effect each change had on the performance of the system. Such changes included trying different weight values to plug into the Linear Quadratic Regulator function and altering the gains of the complimentary filter to adjust the ratio of gyroscope and accelerometer data being used to control the position of the MinSeg.

I learned a lot about mechatronic systems and their design and function throughout the completion of this project. I learned about the underlying physics of MinSeg and how to apply them to solve a real-world problem. I also learned two different methods including Ackerman's Method and the Linear Quadratic Regulator for calculating the proper control gain vectors for a closed loop feedback system to not only properly stabilize the robot, but also how to tweak them to modify the performance characteristics of the controller. I also learned how to integrate knowledge from multiple different disciplines to make my project function as intended including software, hardware, and theory.

It was challenging, but I was able to connect the dots between most of the theory and practice for the scope of this project. I still do not fully understand what effect the changes in pole placement or weights will have on the performance characteristics of the feedback controller, but if I had more time to devote to this project, I would be able to perform more extensive trial and error to draw more accurate conclusions on what effects came of different pole locations or weight parameters.

Part 1: Modeling the MinSeg Robot

In this section, the primary objective was to construct a State-Space model for the MinSeg robot. The equations of motion for the mechanical subsystems of the were derived and analyzed with kinetics and rigid body kinematics. The electrical equation was derived using Kirchoff's voltage law. The electrical and mechanical equations were then combined by using the motor torque equation and then the resulting equations were linearized and put into matrix form. From there, a state-space model with an input and output function was created and the entire system was simulated in MATLAB using a predetermined range of parameter values. The various states were plotted against time on a 2D plot and were analyzed to determine the stability of the open-loop system.

1.1 Derivation of Mechanical Equations of Motion

In this section, the derivations for the equations of motion for the mechanical subsystems were performed. The mechanical subsystems present are the wheel and the pendulum. First, the free body diagram for the pendulum with all acting forces and accelerations was drawn as shown in Figure 1.

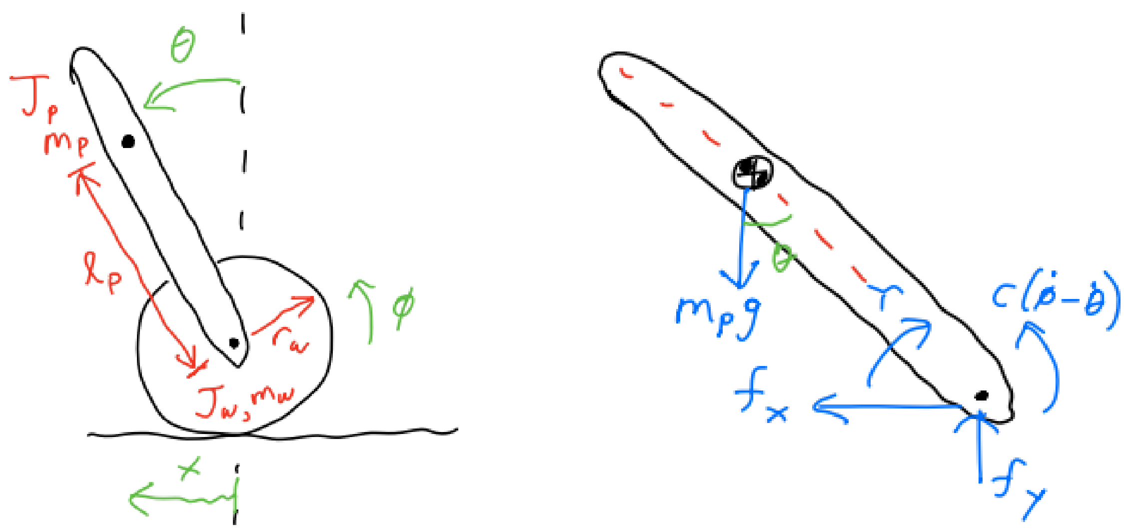


Figure 1. Free body diagram of the pendulum used for kinetic and kinematic analysis.

Newton's 2nd law was applied to derive the equation of motion for the pendulum and the relative acceleration analysis was carried out from the rigid body kinematics of the system as shown below:

Kinematic Analysis

$$\begin{cases} |\overrightarrow{a_B}| = \ddot{x} \\ |\overrightarrow{a_{A/B}}|_n = r\alpha = l_p\dot{\theta}^2 \\ |\overrightarrow{a_{A/B}}|_t = r\omega^2 = l_p\ddot{\theta} \end{cases}$$

Kinetic Analysis

$$\begin{cases} \overline{a_x} = \ddot{x} + l_p\ddot{\theta}\cos(\theta) - l_p\dot{\theta}^2\sin(\theta) \\ m\bar{a}d = m_p(\ddot{x}l_p\cos(\theta) + {l_p}^2\ddot{\theta}) \end{cases}$$

$$\sum F_x = m_p\bar{a}_x \Rightarrow f_x = m_p(\ddot{x} + l_p\ddot{\theta}\cos(\theta) - l_p\dot{\theta}^2\sin(\theta))$$

$$\sum M_B = \bar{I}\alpha + m\bar{a}d \Rightarrow -\tau + c(\dot{\emptyset} - \dot{\theta}) + m_p g l_p\sin(\theta) = J_p\ddot{\theta} + m\bar{a}d$$

$$-\tau + c(\dot{\emptyset} - \dot{\theta}) + m_p g l_p\sin(\theta) = J_p\ddot{\theta} + m\bar{a}d$$

$$-\tau + c(\dot{\emptyset} - \dot{\theta}) + m_p g l_p\sin(\theta) = J_p\ddot{\theta} + m_p(\ddot{x}l_p\cos(\theta) + {l_p}^2\ddot{\theta})$$

The equation of motion for the pendulum subsystem is:

$$\boxed{m_p g l_p\sin(\theta) - \tau + \left[\frac{c}{r_\omega}\right]\dot{x} - c\dot{\theta} = J_p\ddot{\theta} + m_p(\ddot{x}l_p\cos(\theta) + {l_p}^2\ddot{\theta})}$$

Next, the free body diagram for the wheel with all applicable acting forces including gravity, normal force, friction, and the weight of the pendulum was drawn as shown in Figure 2.

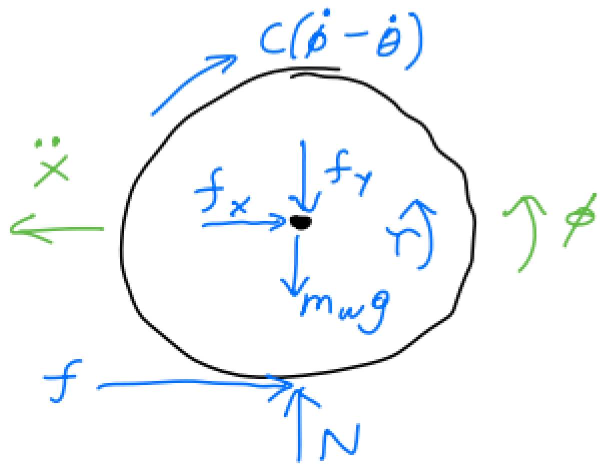


Figure 2. Free Body diagram of the wheel used for kinetic and kinematic analysis.

Newton's 2nd law was applied to derive the equation of motion for the wheel and the relative acceleration analysis was carried out from the rigid body kinematics of the system as follows:

$$\sum M_c = \bar{I}\alpha + m\bar{a}d$$

No Slip Condition

$$\begin{cases} x = r_\omega \emptyset \\ \dot{x} = r_\omega \dot{\emptyset} \\ \ddot{x} = r_\omega \ddot{\emptyset} \end{cases} \begin{cases} \dot{\emptyset} = \frac{\dot{x}}{r_\omega} \\ \ddot{\emptyset} = \frac{\ddot{x}}{r_\omega} \end{cases}$$

$$\tau - c(\dot{\emptyset} - \dot{\theta}) - f_x r_\omega = J_\omega \ddot{\emptyset} + m_\omega r_\omega \ddot{x}$$

$$J_\omega \ddot{\emptyset} + m_\omega r_\omega \ddot{x} + c(\dot{\emptyset} - \dot{\theta}) = \tau - f_x r_\omega$$

The equation of motion for the wheel subsystem is:

$$\left[\frac{J_w}{r_w} + m_\omega r_\omega\right]\ddot{x} + \left[\frac{c}{r_\omega}\right]\dot{x} - c\dot{\theta} = \tau - f_x r_\omega$$

1.2 Derivation of the Electrical Circuit Differential Equation

In this section, the differential equation for the electrical circuit was derived by using Kirchoff's Voltage Law. The circuit diagram was sketched as shown in Figure 3.

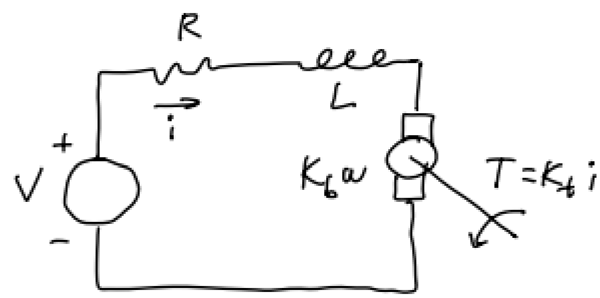


Figure 3. Circuit diagram of the electrical circuit of the MinSeg robot

The derivation using KVL was performed as follows:

Kirchoff's Voltage Law

$$\sum \boldsymbol{V} = \boldsymbol{0}$$

$$V - iR - L\frac{di}{dt} - K_b\omega = 0$$

The electrical circuit differential equation is:

$$V = iR + L\frac{di}{dt} + K_b\omega$$

1.3 Combination of the Mechanical and Electrical Equations

In this section, the mechanical and electrical equations were combined using the mechanical and electrical equations derived in the previous sections. The derivation in this section was performed under the assumption that coil inductance is equal to zero.

$$L = 0 \qquad K_b\omega = K_b(\dot{\emptyset} - \dot{\theta})$$

$$iR = V - K_b(\dot{\emptyset} - \dot{\theta})$$

$$i = \frac{V}{R} - \frac{K_b}{R}\left[\frac{\dot{x}}{r_\omega}\right] + \frac{K_b}{R}\dot{\theta}$$

The electrical equation was related to the mechanical equation using the electric motor torque formula:

$$\tau = K_t i$$

$$\tau = K_t\left[\frac{V}{R} - \frac{K_b}{R}\left[\frac{\dot{x}}{r_\omega}\right] + \frac{K_b}{R}\dot{\theta}\right]$$

The resulting formula for motor torque is:

$$\boxed{\tau = \frac{K_t}{R}V - \frac{K_b K_t}{R r_\omega}\dot{x} + \frac{K_b K_t}{R}\dot{\theta}}$$

1.4. Linearization of Governing Equations of Motion and conversion to Matrix Form

In this section, the motor torque equation derived in the previous section was substituted into both mechanical equations of motion governing the movement of the MinSeg robot. The equations were then linearized by assuming that the deflection angle, θ, will remain small as the robot will be controlled to stay in the upright position.

Small Angle, θ, the following was assumed:

$$\sin(\theta) = \theta \qquad cos = 1 \qquad \dot{\theta}^2\sin(\theta) = 0$$

These values were then substituted into the differential equation for the pendulum:

$$m_p g l_p \sin(\theta) - \left[\frac{K_t}{R}V - \frac{K_b K_t}{R r_\omega}\dot{x}\right] + \left[\frac{c}{r_\omega}\right]\dot{x} - c\dot{\theta} = J_p\ddot{\theta} + m_p(\ddot{x} l_p \cos(\theta) + {l_p}^2\ddot{\theta})$$

$$[m_p g l_p]\theta - \left[\frac{K_t}{R}\right]V + \left[\frac{K_b K_t}{R r_\omega}\right]\dot{x} + \left[\frac{-K_b K_t}{R}\right]\dot{\theta} + \left[\frac{c}{r_\omega}\right]\dot{x} - [c]\dot{\theta} = [J_p]\ddot{\theta} + [m_p l_p]\ddot{x} + [m_p {l_p}^2]\ddot{\theta}$$

The linearized equation for the motion of the pendulum is as follows:

$$\boxed{[m_p l_p]\ddot{x} + [J_p + m_p {l_p}^2]\ddot{\theta} + \left[\frac{-K_b K_t}{R r_\omega} - \frac{c}{r_\omega}\right]\dot{x} + \left[\frac{K_b K_t}{R} + c\right]\dot{\theta} - [m_p g l_p]\theta = \left[\frac{-K_t}{R}\right]V}$$

The small angle assumption values were then substituted into the differential equation for the wheel:

$$\left[\frac{J_w}{r_w} + m_\omega r_\omega\right]\ddot{x} + \left[\frac{c}{r_\omega}\right]\dot{x} - c\dot{\theta}$$
$$= \left[\frac{K_t}{R}V - \frac{K_b K_t}{R r_\omega}\dot{x} + \frac{K_b K_t}{R}\dot{\theta}\right] - r_\omega\left[m_p(\ddot{x} + l_p\ddot{\theta}\cos(\theta) - l_p\dot{\theta}^2\sin(\theta))\right]$$

$$\left[\frac{J_\omega}{r_\omega} + m_\omega r_\omega + m_p r_\omega\right]\ddot{x} + \left[\frac{c}{r_\omega} + \frac{K_b K_t}{R r_\omega}\right]\dot{x} + [r_\omega m_p l_p]\ddot{\theta} + \left[-c - \frac{K_b K_t}{R}\right]\dot{\theta} = \frac{K_t}{R}V$$

The linearized equation for the motion of the wheel is as follows:

$$\boxed{\left[\frac{J_\omega}{{r_\omega}^2} + m_\omega + m_p\right]\ddot{x} + [m_p l_p]\ddot{\theta} + \left[\frac{c}{{r_w}^2} + \frac{K_b K_t}{R {r_\omega}^2}\right]\dot{x} + \left[\frac{-c}{r_\omega} - \frac{K_b K_t}{R r_\omega}\right]\dot{\theta} = \left[\frac{K_t}{R r_\omega}\right]V}$$

Once both equations were linearized, they were put into matrix form resulting in the following M, C, K, and F matrices.

$$\begin{bmatrix} \left(\frac{J_\omega}{r\omega^2} + m_\omega + m_p\right) & (m_p l_p) \\ (m_p l_p) & (J_p + m_p {l_p}^2) \end{bmatrix}\begin{bmatrix}\ddot{x} \\ \ddot{\theta}\end{bmatrix} + \begin{bmatrix} \left(\frac{c}{r\omega^2} + \frac{K_b K_t}{R {r_\omega}^2}\right) & \left(\frac{-K_b K_t}{R r_\omega} - \frac{c}{r_\omega}\right) \\ \left(\frac{-c}{r_\omega} - \frac{K_b K_t}{R r_\omega}\right) & \left(\frac{K_b K_t}{R} + c\right) \end{bmatrix}\begin{bmatrix}\dot{x} \\ \dot{\theta}\end{bmatrix}$$
$$+ \begin{bmatrix} 0 & 0 \\ 0 & -m_p g l_p \end{bmatrix}\begin{bmatrix} x \\ \theta \end{bmatrix} = \begin{bmatrix} \frac{K_t}{R r_\omega} \\ \frac{-K_t}{R} \end{bmatrix}\begin{bmatrix} V \\ V \end{bmatrix}$$

$$M = \begin{bmatrix} \left(\frac{J_\omega}{r\omega^2} + m_\omega + m_p\right) & (m_p l_p) \\ (m_p l_p) & (J_p + m_p {l_p}^2) \end{bmatrix}$$

$$C = \begin{bmatrix} \left(\frac{c}{r\omega^2} + \frac{K_b K_t}{R{r_\omega}^2}\right) & \left(\frac{-K_b K_t}{R r_\omega} - \frac{c}{r_\omega}\right) \\ \left(\frac{-c}{r_\omega} - \frac{K_b K_t}{R r_\omega}\right) & \left(\frac{K_b K_t}{R} + c\right) \end{bmatrix}$$

$$K = \begin{bmatrix} 0 & 0 \\ 0 & -m_p g l_p \end{bmatrix}$$

$$F = \begin{bmatrix} \frac{K_t}{R r_\omega} \\ \frac{-K_t}{R} \end{bmatrix}$$

1.5 Conversion of Matrix Equation to State-Space Form

In this section, the matrix equation was converted to the state-space form by using the following formula:

$$\dot{q}(t) = \boldsymbol{A} q(t) + \boldsymbol{B} u(t)$$

Where:

$$A = \begin{bmatrix} 0_{2x2} & I_{2x2} \\ -M^{-1}K & -M^{-1}C \end{bmatrix}$$

$$B = \begin{bmatrix} 0_{2x1} \\ \hline M^{-1}F \end{bmatrix}$$

and

$$0_{2x2} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \qquad I_{2x2} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \qquad 0_{2x1} = \begin{bmatrix} 0 \\ \hline 0 \end{bmatrix}$$

Thus, we have the A and B matrix that make up the input equation of the state-space model:

$$\boldsymbol{A} = \begin{bmatrix} \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} & \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \\ -\begin{bmatrix} \left(\frac{J_\omega}{r\omega^2} + m_\omega + m_p\right) & (m_p l_p) \\ (m_p l_p) & (J_p + m_p {l_p}^2) \end{bmatrix}^{-1} \begin{bmatrix} 0 & 0 \\ 0 & -m_p g l_p \end{bmatrix} & -\begin{bmatrix} \left(\frac{J_\omega}{r\omega^2} + m_\omega + m_p\right) & (m_p l_p) \\ (m_p l_p) & (J_p + m_p {l_p}^2) \end{bmatrix}^{-1} \begin{bmatrix} \left(\frac{c}{r\omega^2} + \frac{K_b K_t}{R{r_\omega}^2}\right) & \left(\frac{-K_b K_t}{R r_\omega} - \frac{c}{r_\omega}\right) \\ \left(\frac{-c}{r_\omega} - \frac{K_b K_t}{R r_\omega}\right) & \left(\frac{K_b K_t}{R} + c\right) \end{bmatrix} \end{bmatrix}$$

$$\boldsymbol{B} = \left[\frac{\begin{bmatrix} 0 \\ 0 \end{bmatrix}}{\begin{bmatrix} \left(\frac{J_\omega}{r\omega^2} + m_\omega + m_p\right) & (m_p l_p) \\ (m_p l_p) & (J_p + m_p {l_p}^2) \end{bmatrix}^{-1} \begin{bmatrix} \frac{K_t}{R r_\omega} \\ \frac{-K_t}{R} \end{bmatrix}} \right]$$

Once the input function of the state-space model was created, a MATLAB script was written using the parameter values in Table 1.

Parameter	Description	Value	Unit
m_p	Pendulum mass	0.285	Kg
m_w	Wheel mass	0.025	Kg
J_p	Pendulum inertia	0.0010	Kg-m^2
J_w	Wheel inertia	0.0013	Kg-m^2
r_w	Wheel radius	0.022	m
l	Pendulum length	0.1	m
c	Rotational damping	0.0001	N.m.s/rad
k_t	Torque constant	0.3	N.m/A
k_b	Back-emf constant	0.5	V.s/rad
R	Coil resistance	5	Ω
g	Gravitational acceleration	9.81	m/s^2

Table 1. Parameter values used to simulate the state-space system of the MinSeg robot in MATLAB.

1.6: Verification of State Space Model with Eigenvalues

In this section, the state-space system was verified by comparing the eigenvalues of the A matrix and the values of the B matrix calculated in the MATLAB script to those specified by the project instructions. The desired eigenvalues of matrix A and the values of matrix B are as follows:

eig(A)) = **[0; 6.9; -6.0; -38]**

and

B = **[0; 0; 1.14; -24.01]**

The values in the state space model matched those specified by the project instructions, therefore, the state-space system is valid and will successfully balance the MinSeg robot using it to complete the remainder of the process.

1.7: Simulation of Open Loop State State-Space Model in MATLAB

In this section, the state space-system was simulated for a small initial angle θ for zero input voltage, thus creating an open-loop system. The output equation for the state-space model was created as shown below:

$$y(t) = C_1 q(t) + Du(t)$$

Where:

$$C_1 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad D = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

The simulation was run, and the various states of the system were plotted against time including wheel position, pendulum angle, wheel velocity, and pendulum angle vector as shown in Figures 4, 5, 6, and 7 below respectively:

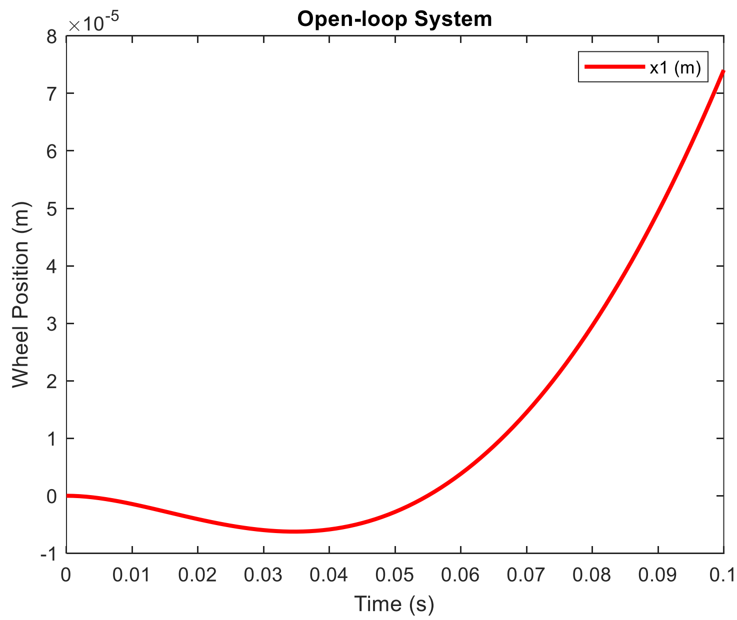


Figure 4. Wheel position vs. time plot for the open-loop simulation

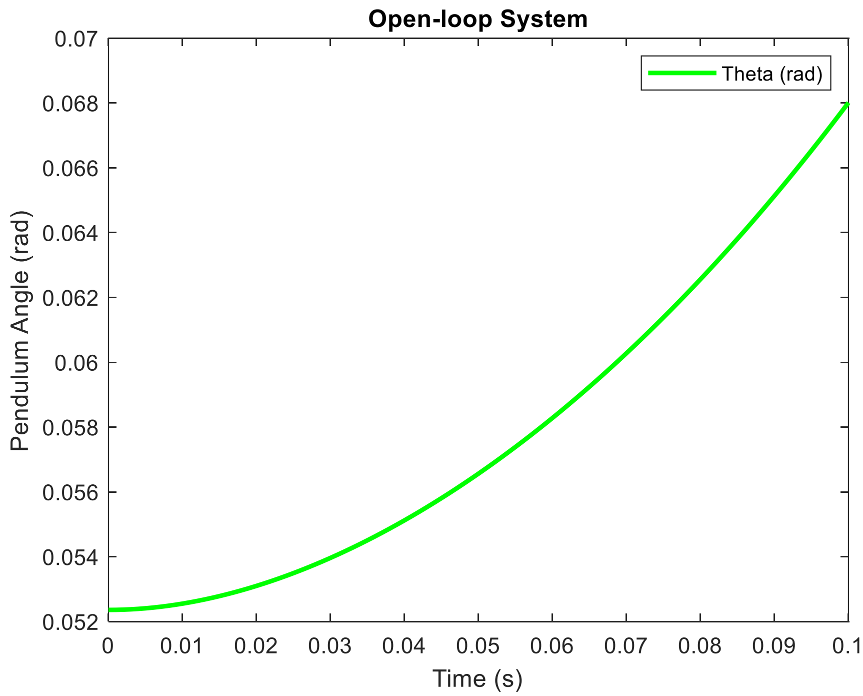


Figure 5. Pendulum Angle vs. time plot for the open-loop simulation

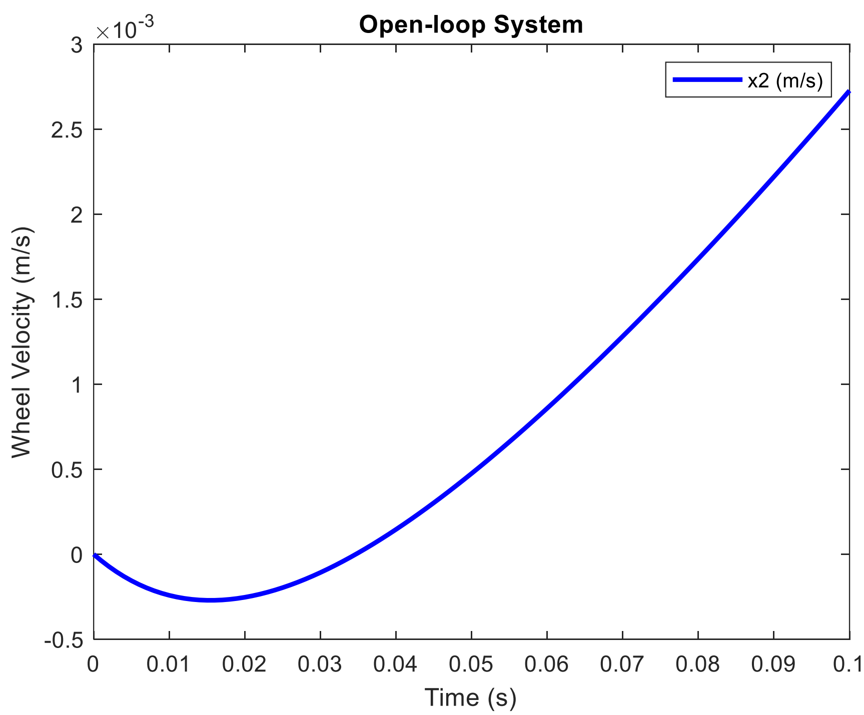


Figure 6. Wheel velocity vs. time plot for the open-loop simulation

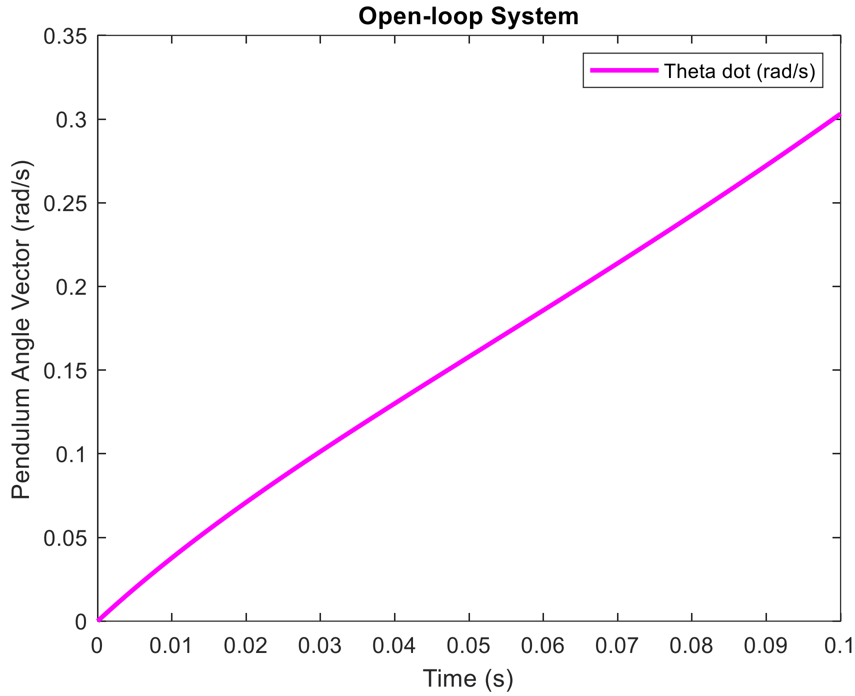


Figure 7. Pendulum angle vector vs. time plot for the open-loop simulation

The open loop system is unstable due to the lack of control from the feedback system, thus causing the pendulum angle and wheel position to deviate further from zero as time progresses.

Part 2: Designing a Full-State Feedback Controller

In this section, the state-space system developed in Part 1 was stabilized and regulated using a full-state feedback control law with a calculated vector of control gains. By using Ackerman's method to calculate gain values based on input pole locations, the system was stabilized and able to be fine-tuned or adjusted at will.

2.1 Using Ackerman's Method to place eigenvalues and generate control gain vector

In this section, MATLAB's built in "acker" function was used to place the eigenvalues of the system in optimal locations for system stability in the complex, left hand plane when plotted in 2D. Based upon results from trial and error, the ideal pole locations were as follows:

[-6, -6, -6, -6]

2.2 Implementation of full-state feedback loop in state-space model

In this section, the gain control vector, K, that was calculated in the previous section was used to simulate the system response in MATLAB to an initial condition of 10 degrees. The input voltage was monitored to ensure that it would not exceed the maximum value of 5V.

2.3 Fine tuning of pole locations and simulation results

The simulation was run using the aforementioned parameters, and the various states of the system were plotted against time including wheel position, pendulum angle, wheel velocity, and pendulum angle vector as shown in Figures 8, 9, 10, and 11 respectively. Input voltage was also plotted vs. time in Figure 12.

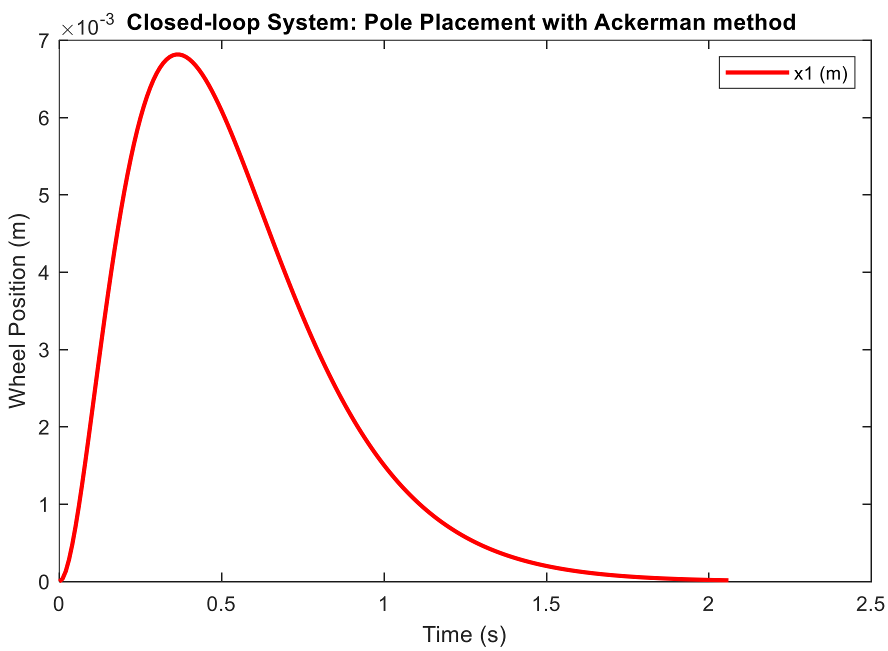


Figure 8. Wheel position vs. time plot for the closed-loop simulation using control gains calculated with Ackerman's Method.

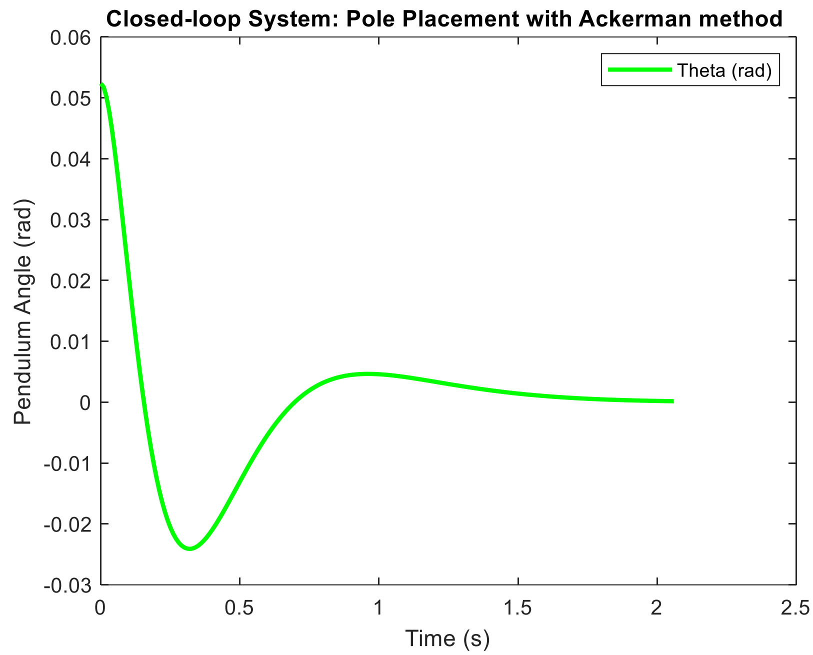


Figure 9. Pendulum Angle vs. time plot for the closed-loop simulation using control gains calculated with Ackerman's Method.

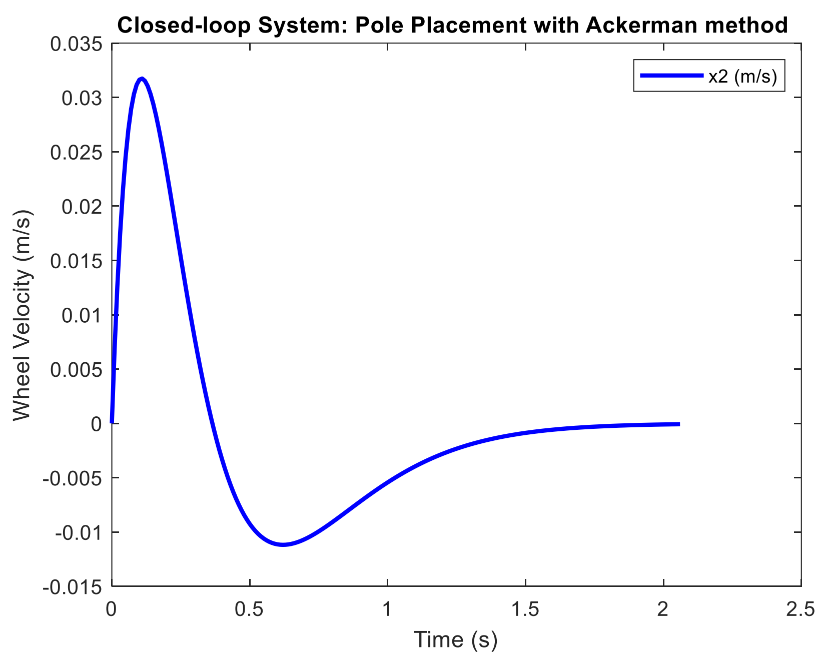


Figure 10. Wheel velocity vs. time plot for the closed-loop simulation using control gains calculated with Ackerman's Method.

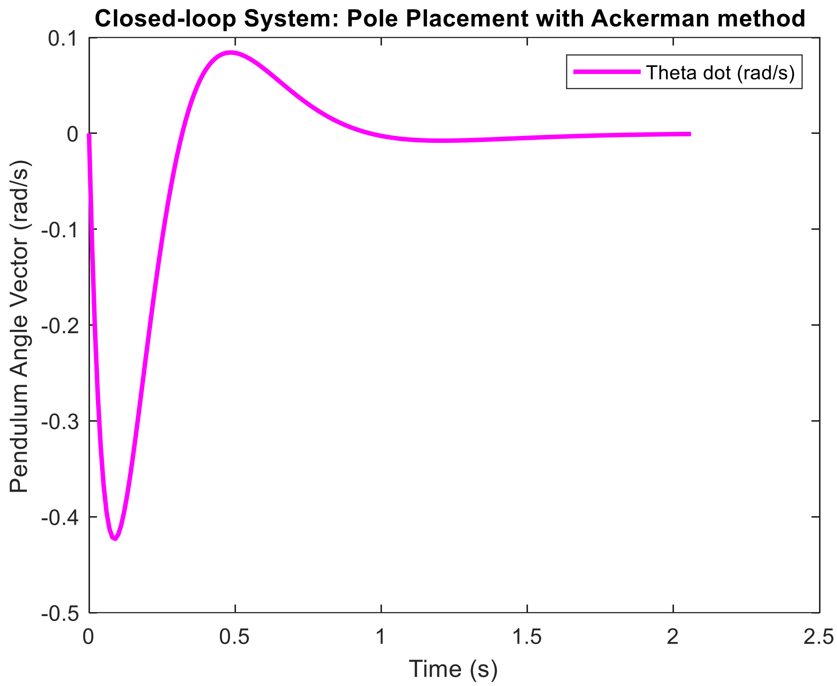


Figure 11. Pendulum Angle Vector vs. time plot for the closed-loop simulation using control gains calculated with Ackerman's Method.

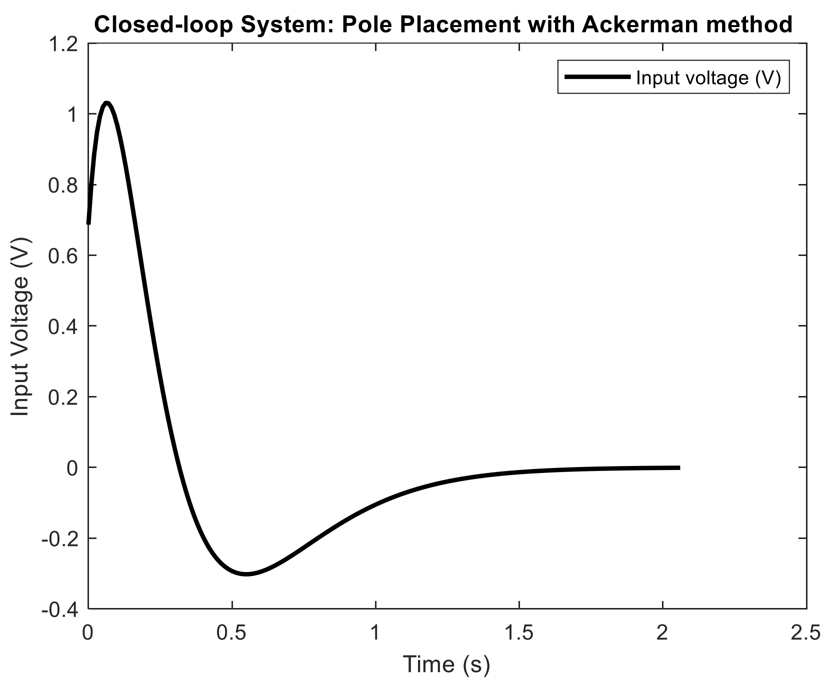


Figure 12. Input voltage vs. time plot for the closed-loop simulation using control gains calculated with Ackerman's Method.

Part 3: Designing a Linear Quadratic Regulator (LQR)

In this section, an alternative method for pole placement was explored. The linear quadratic regulator (LQR) creates a feedback gain vector using 2 user defined weight matrices to fine-tune the performance of the feedback controller. Once the Q and R matrices were determined, then the feedback gain vector was calculated using the LQR method. From there, the resulting state-space model was simulated for a small angle initial condition and the response characteristics of the system were stabilized.

3.1 Using MATLAB's LQR function to obtain feedback gain vector

In this section, MATLAB's "lqr" function was utilized to take a user defined Q and R matrix and calculate a control gain vector that would be used to add a full-state feedback loop to control the step response.

3.2 Fine tuning of feedback gain matrix and simulation results

The simulation was run using the aforementioned parameters, and the various states of the system were plotted against time including wheel position, pendulum angle, wheel velocity, and pendulum angle vector as shown in Figures 13, 14, 15, and 16 respectively. Input voltage was also plotted vs. time in Figure 17.

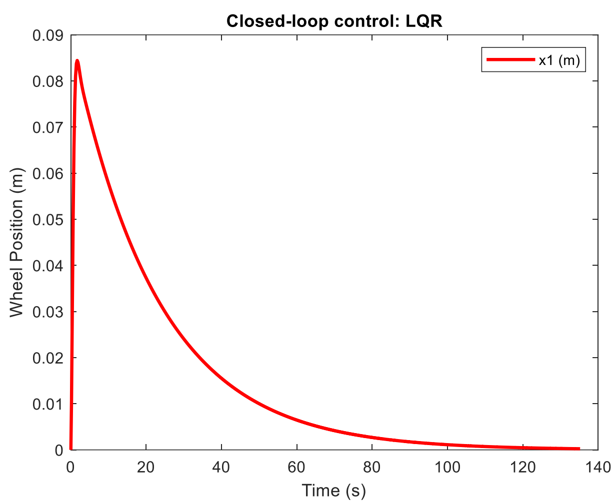


Figure 13. Wheel Position vs. time plot for the closed-loop simulation using control gains calculated with MATLAB's LQR function.

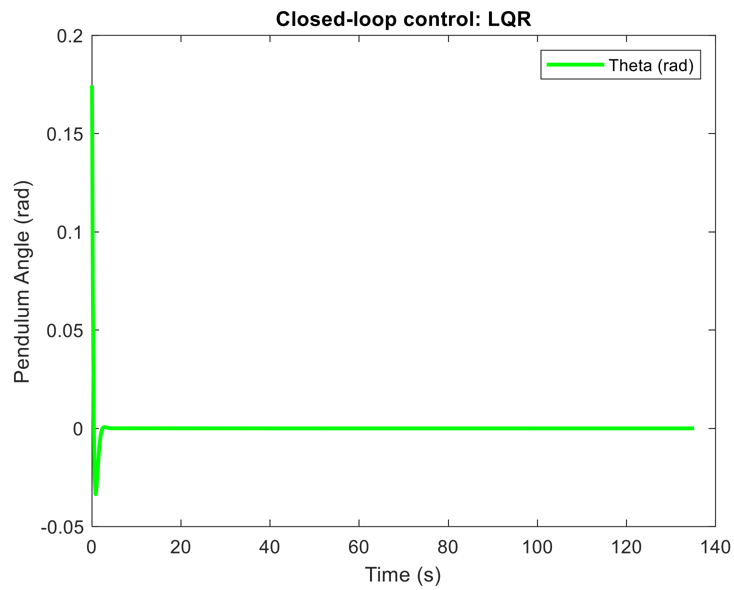


Figure 14. Pendulum Angle vs. time plot for the closed-loop simulation using control gains calculated with MATLAB's LQR function.

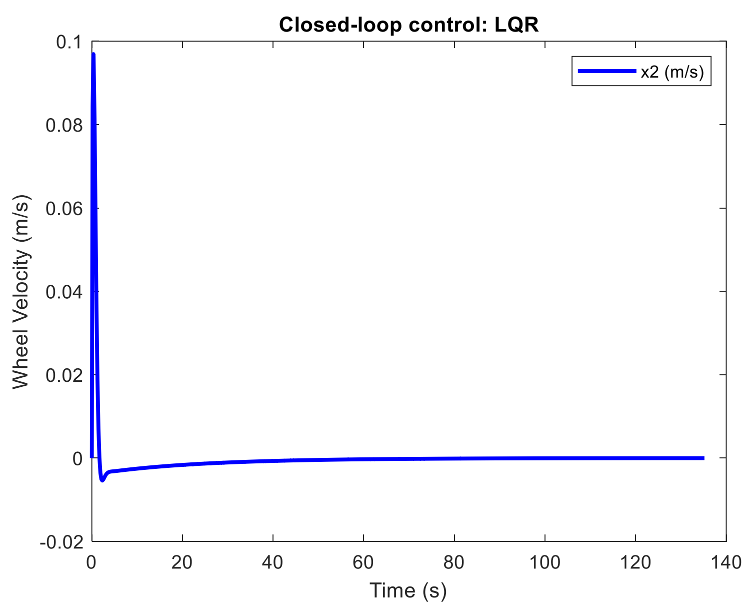


Figure 15. Wheel Velocity vs. time plot for the closed-loop simulation using control gains calculated with MATLAB's LQR function.

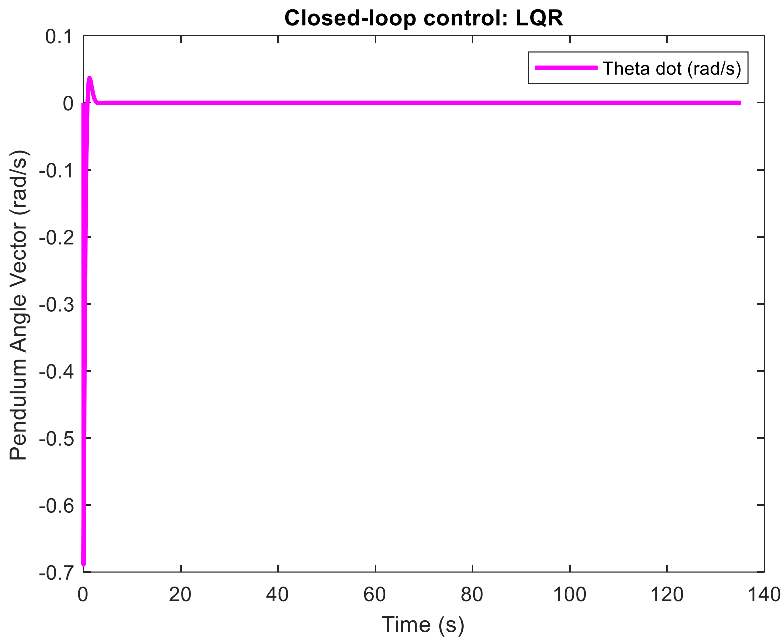


Figure 16. Pendulum Angle Vector vs. time plot for the closed-loop simulation using control gains calculated with MATLAB's LQR function.

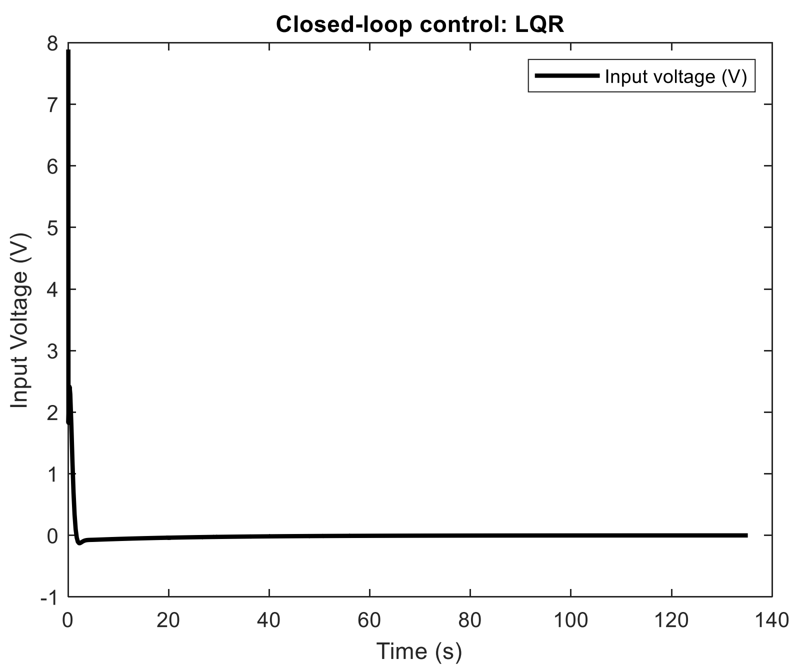


Figure 17. Input voltage vs. time plot for the closed-loop simulation using control gains calculated with MATLAB's LQR function.

Part 4: Implementation of the Feedback Controller

In this section, the feedback controller that was designed and developed in the previous section was fully implemented to balance the MinSeg robot. The sensors on the MinSeg robot, specifically the gyroscope, encoder and the accelerometer collected data that was fed into the feedback system to apply the necessary amount of control to stabilize the robot. This was accomplished with the use of a Simulink block diagram.

4.1 Creation of closed-loop feedback controller system in Simulink: Sensors

In this section, the sensors subsystem of the feedback control system was created as shown in figure 18. The sensors that were used in this system were the gyroscope and accelerometer.

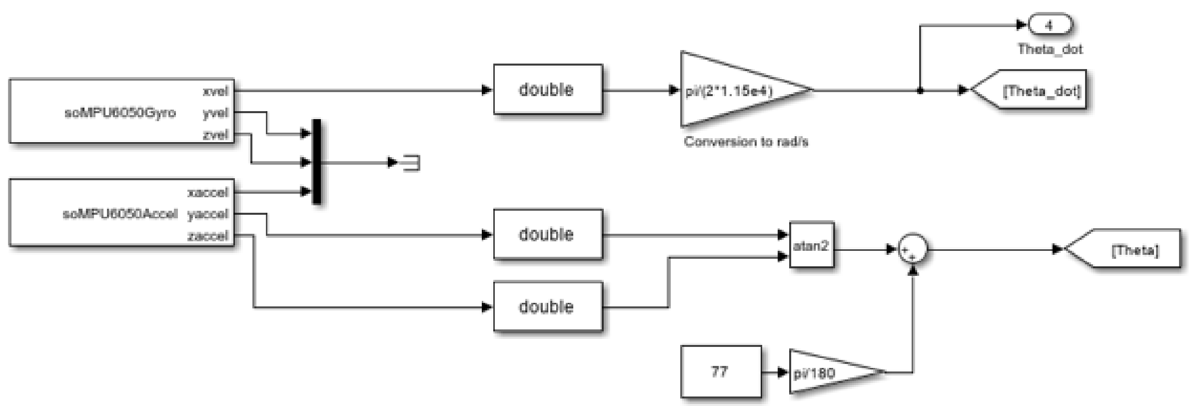


Figure 18. Sensor subsystem of the feedback control system.

4.2 Creation of complimentary filter for gyroscope and accelerometer data

In this section, a complimentary filter was created to receive data from the gyroscope and the accelerometer as shown in figure 19.

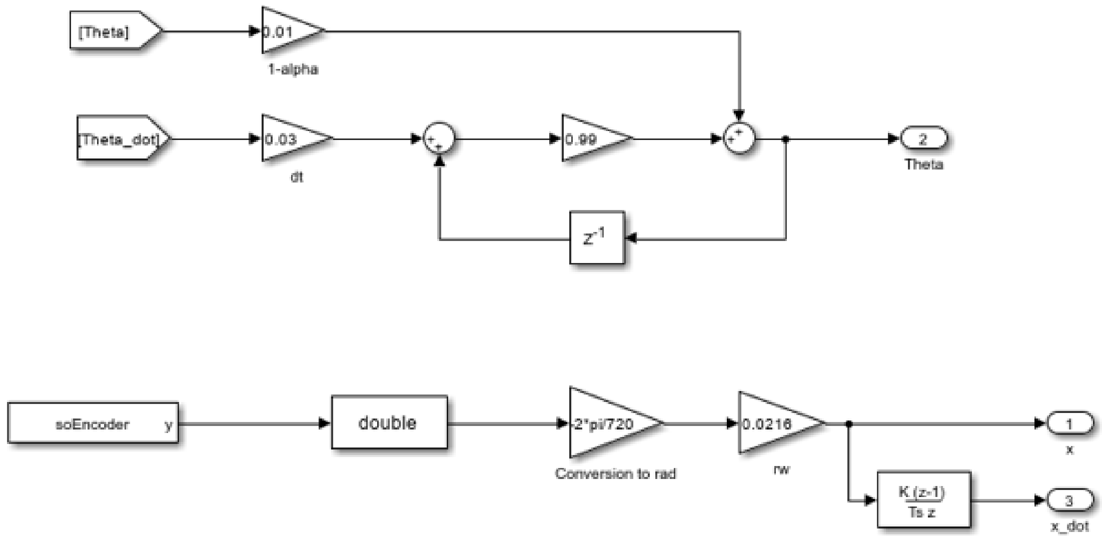


Figure 19. Complimentary filter for the feedback control loop: it determines the ratio of gyroscope to accelerometer data that is fed to system

4.3 Creation of motor driver block for Simulink model

After the creation of the sensor subsystem and the complimentary filter, the motor driver was created and implemented into the closed loop feedback system. This motor driver has a built-in friction correction bias to control the amount of PWM needed for the motor to physically move. The block diagram for this subsystem is depicted in Figure 20.

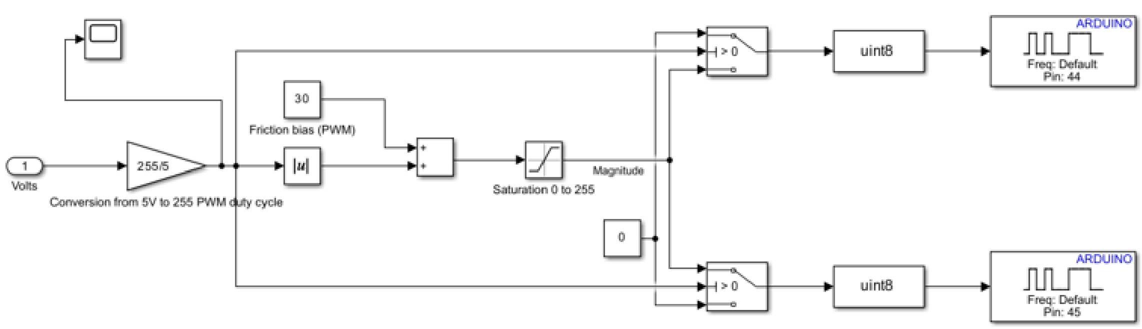


Figure 20. Motor driver used in the closed-loop feedback system

4.4 Finalized Simulink model for closed-loop feedback system

In this section, the subsystems created in the previous sections were tied together to create one complete closed-loop feedback system. The resulting block diagram is depicted in Figure 21.

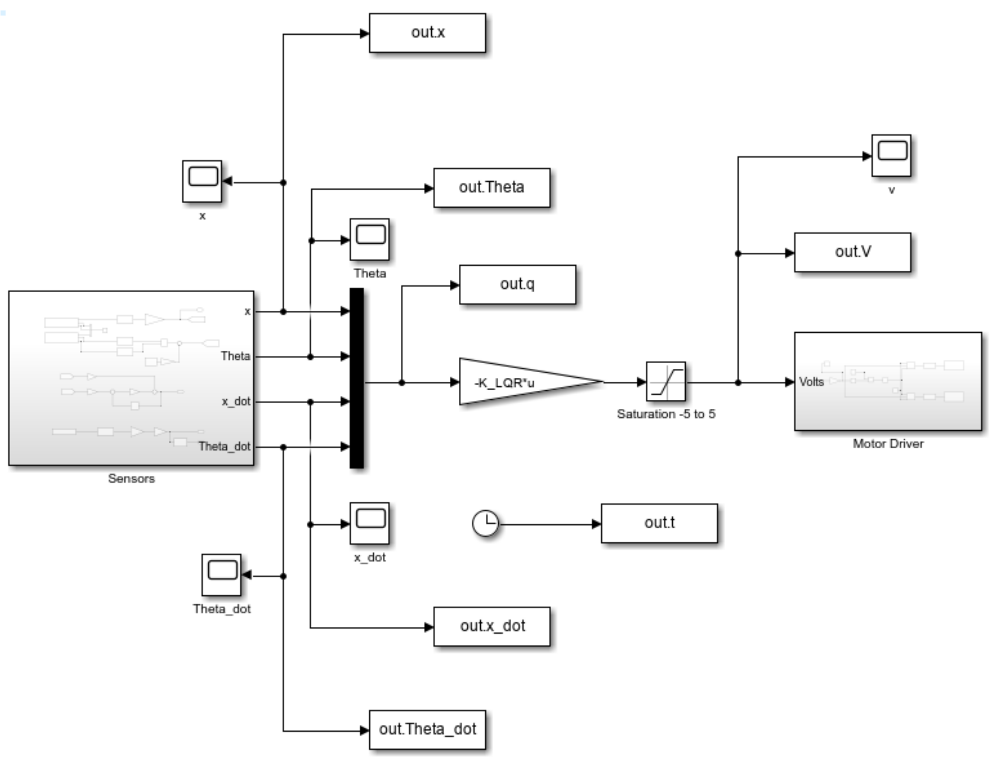


Figure 21. Complete closed-loop feedback system utilizing all of the subsystems previously constructed. System is designed to export data to the MATLAB workplace for further analysis.

4.5. Test LQR values for closed-loop feedback system

Once the closed-loop feedback system was complete, it came time to calculate the feedback gain control vector. According to the project instructions, the following Q and R matrices were to be used to plug into MATLAB's "lqr" function:

$$Q = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 100 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 100 \end{bmatrix} \qquad R = 1$$

4.6 Creation of feedback gain vector based on test LQR values

Once the weight matrices defined in the previous section were plugged into the LQR function, a feedback gain control vector was generated. At this point, the system was ready to be put to the test and balance the MinSeg robot.

4.7 Fine tuning of LQR values

At first, the robot was able to balance but the performance characteristics of the feedback controller were not perfect. They had room for improvement. In this section of the project, the weight matrices were tweaked to provide the optimal feedback control gain vector that would result in the desired performance characteristics.

4.8 Closed-loop feedback simulation results: Identical weights for x and θ

In this section, the Q matrix was adjusted so that the weights corresponding to the wheel position and the pendulum angle were identical. The simulation was then run, and the various states of the system were plotted against time including wheel position, pendulum angle, wheel velocity, and pendulum angle vector as shown in Figures 22, 23, 24, and 25 below respectively. Input voltage was also plotted vs. time in Figure 26.

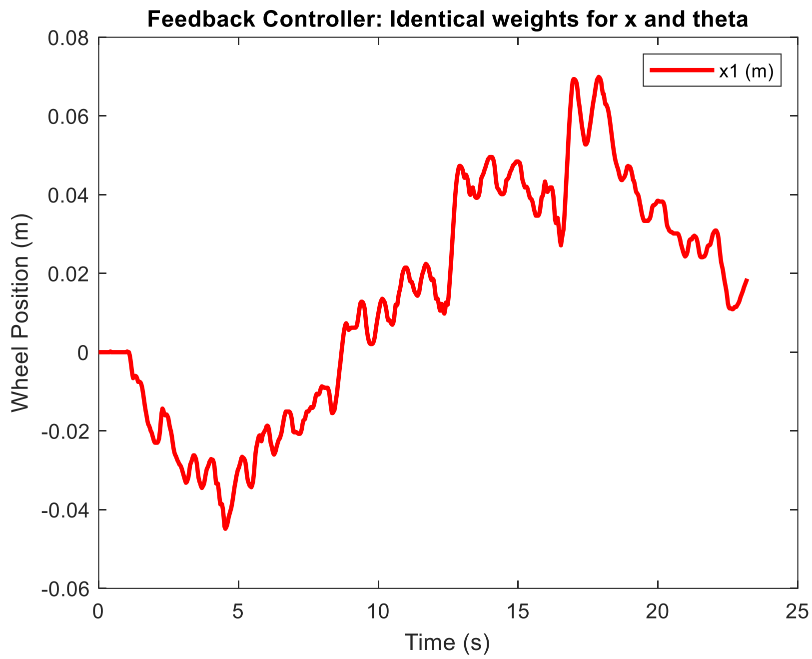


Figure 22. Wheel Position vs. time plot for data collected from closed-loop feedback system controlling the MinSeg robot using identical weights for x and θ

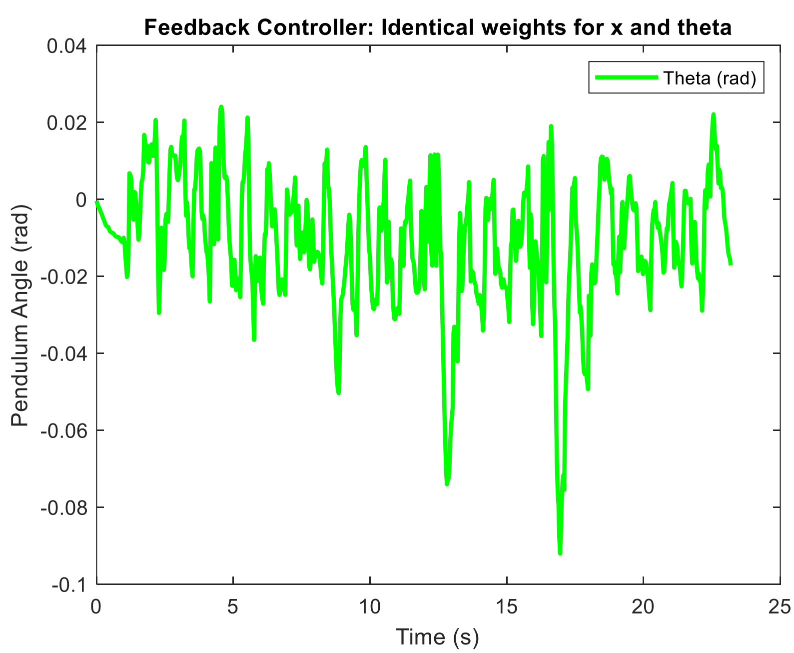


Figure 23. Pendulum angle vs. time plot for data collected from closed-loop feedback system controlling the MinSeg robot using identical weights for x and θ

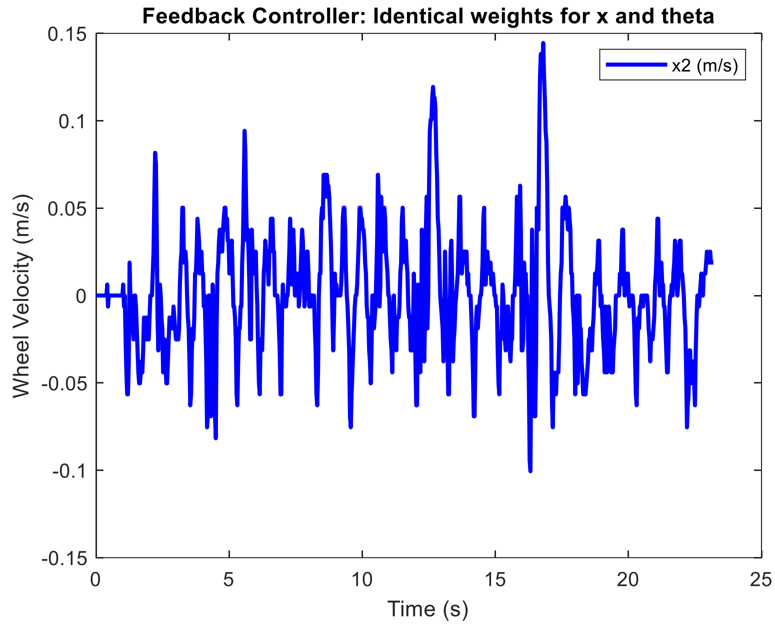


Figure 24. Wheel Velocity vs. time plot for data collected from closed-loop feedback system controlling the MinSeg robot using identical weights for x and θ

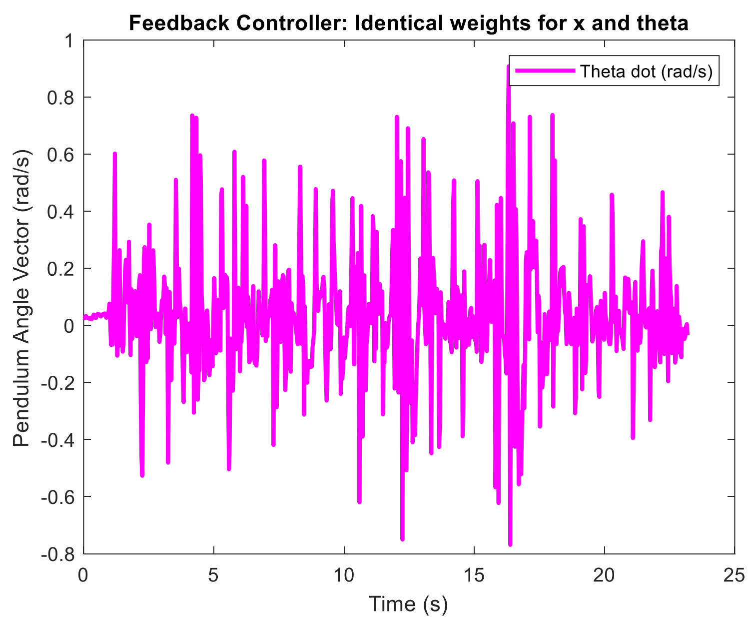


Figure 25. Pendulum Angle Vector vs. time plot for data collected from closed-loop feedback system controlling the MinSeg robot using identical weights for x and θ

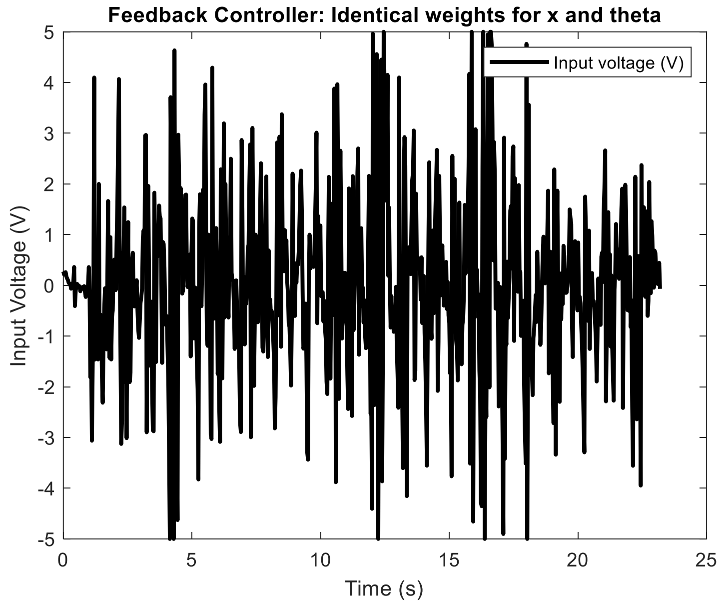


Figure 26. Input Voltage vs. time plot for data collected from closed-loop feedback system controlling the MinSeg robot using identical weights for x and θ

4.9 Closed-loop feedback simulation results: Fine tuning of control gains

In this section, the weight matrices were fine tuned to further improve the performance characteristics of the closed-loop feedback controller. Then, the simulation was run, and the various states of the system were plotted against time including wheel position, pendulum angle, wheel velocity, and pendulum angle vector as shown in Figures 27, 28, 29, and 30 below respectively. Input voltage was also plotted vs. time in Figure 31.

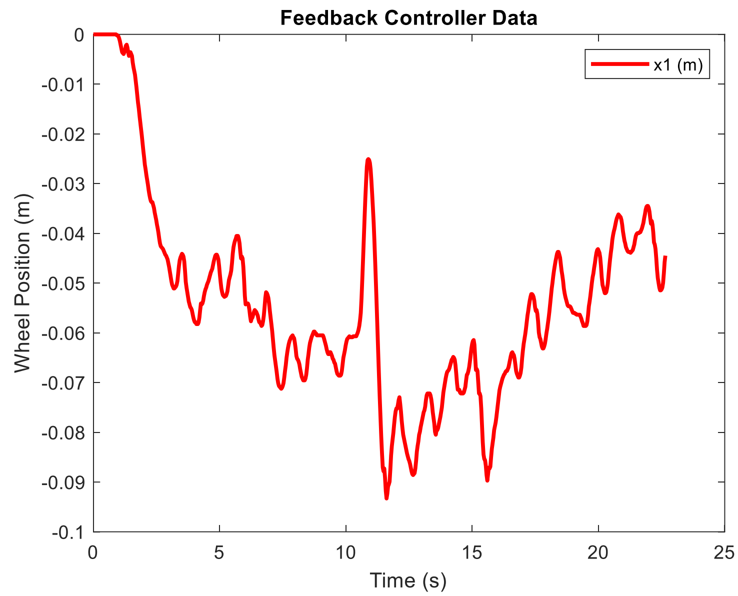


Figure 27.S Wheel Position vs. time plot for data collected from closed-loop feedback system controlling the MinSeg robot using fine-tuned control gains

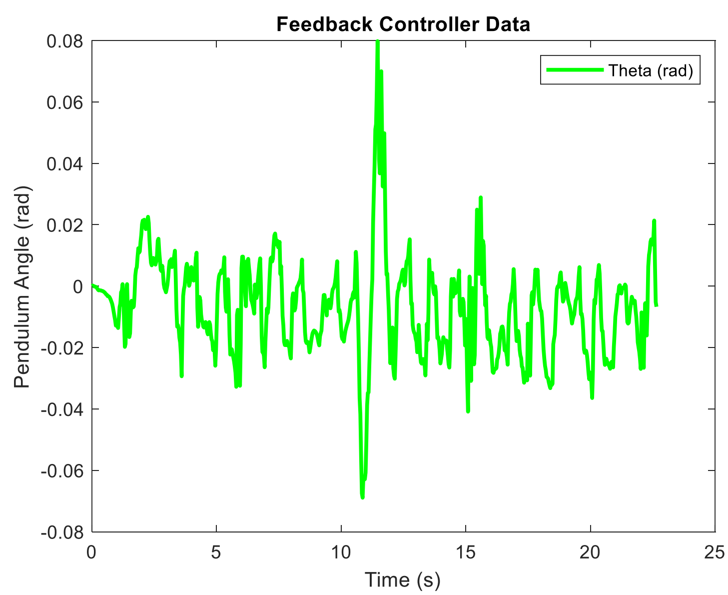


Figure 28. Pendulum Angle vs. time plot for data collected from closed-loop feedback system controlling the MinSeg robot using fine-tuned control gains

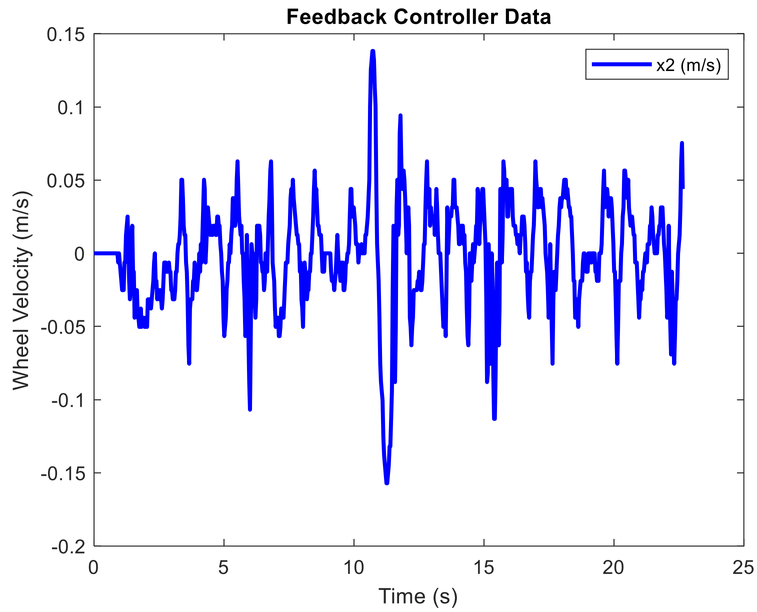


Figure 29. Wheel velocity vs. time plot for data collected from closed-loop feedback system controlling the MinSeg robot using fine-tuned control gains

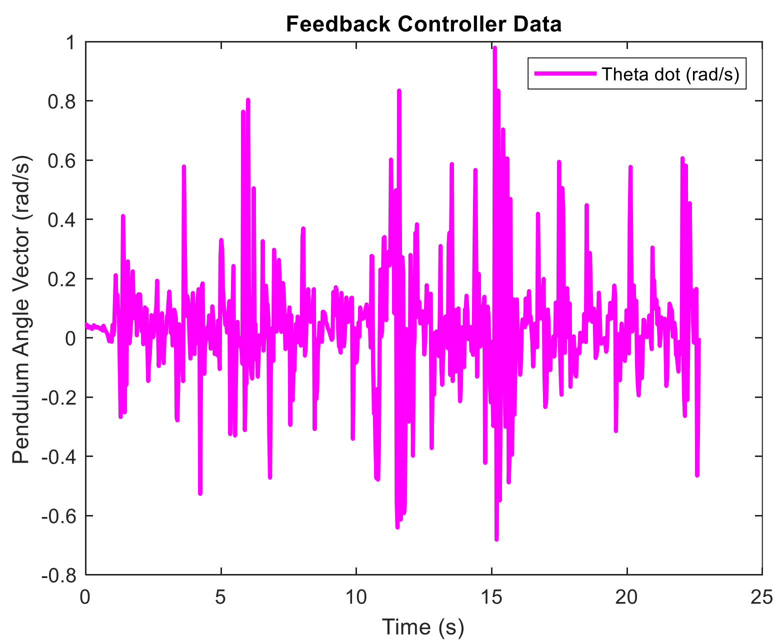


Figure 30. Pendulum Angle Vector vs. time plot for data collected from closed-loop feedback system controlling the MinSeg robot using fine-tuned control gains

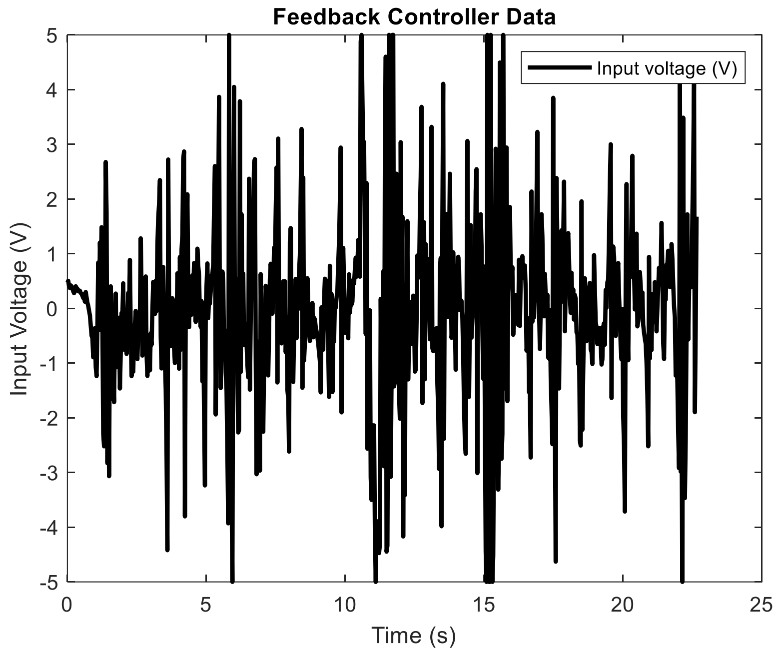


Figure 31. Pendulum Angle Vector vs. time plot for data collected from closed-loop feedback system controlling the MinSeg robot using fine-tuned control gains

4.10 closed-loop feedback system performance analysis with complimentary filter disabled

In this section, the simulation was run using the same parameters as the previous section, however the gain on the complimentary filter was adjusted so that the state of the pendulum angle was determined solely by the accelerometer data. The simulation was then run, and the various states of the system were plotted against time including wheel position, pendulum angle, wheel velocity, and pendulum angle vector as shown in Figures 32, 33, 34, and 35 below respectively. Input voltage was also plotted vs. time in Figure 36.

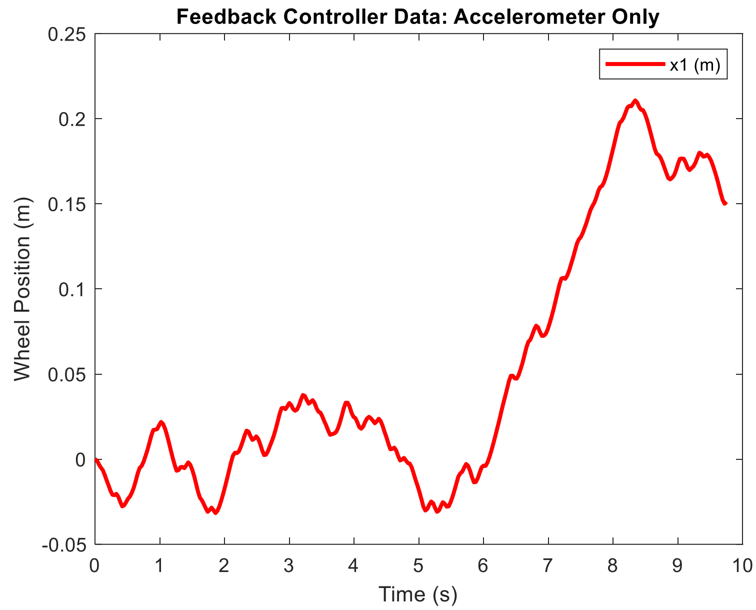


Figure 32. Wheel position vs. time plot for data collected from closed-loop feedback system controlling the MinSeg robot using only data collected from the accelerometer.

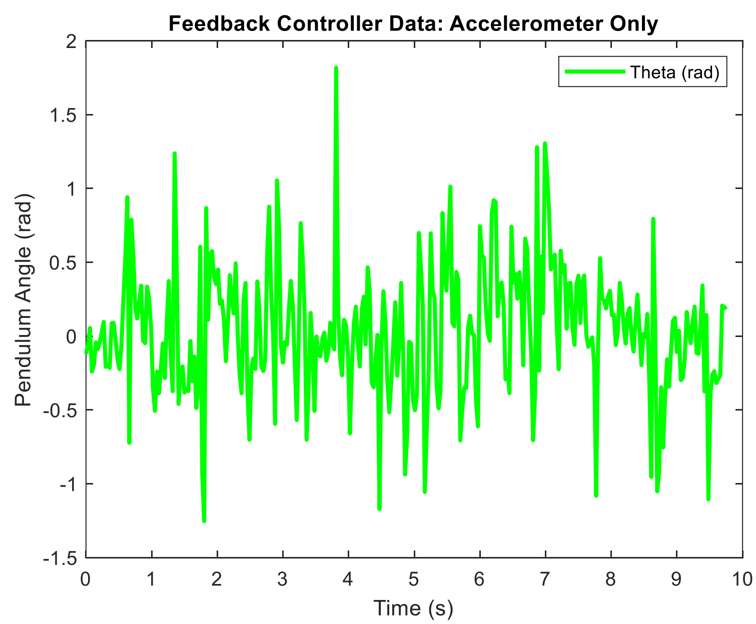


Figure 33. Pendulum angle vs. time plot for data collected from closed-loop feedback system controlling the MinSeg robot using only data collected from the accelerometer.

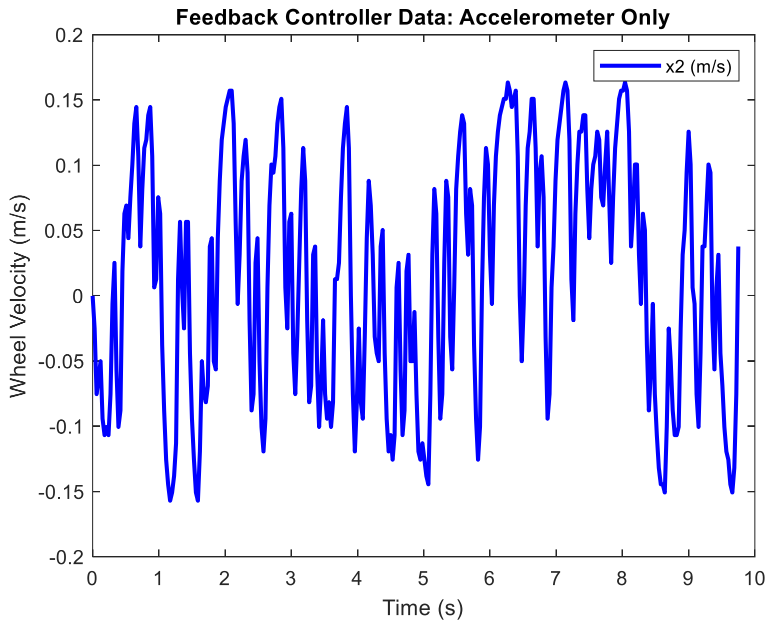


Figure 34. Wheel velocity vs. time plot for data collected from closed-loop feedback system controlling the MinSeg robot using only data collected from the accelerometer.

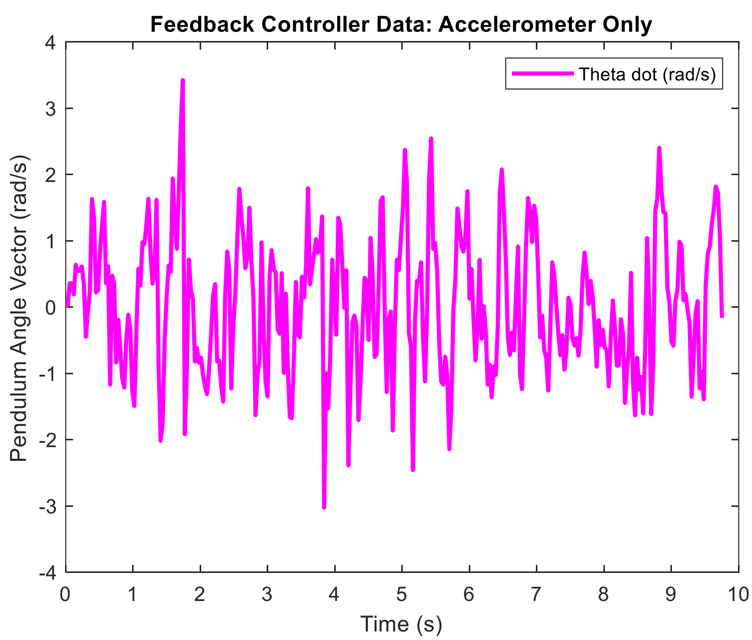


Figure 35. Pendulum angle vector vs. time plot for data collected from closed-loop feedback system controlling the MinSeg robot using only data collected from the accelerometer.

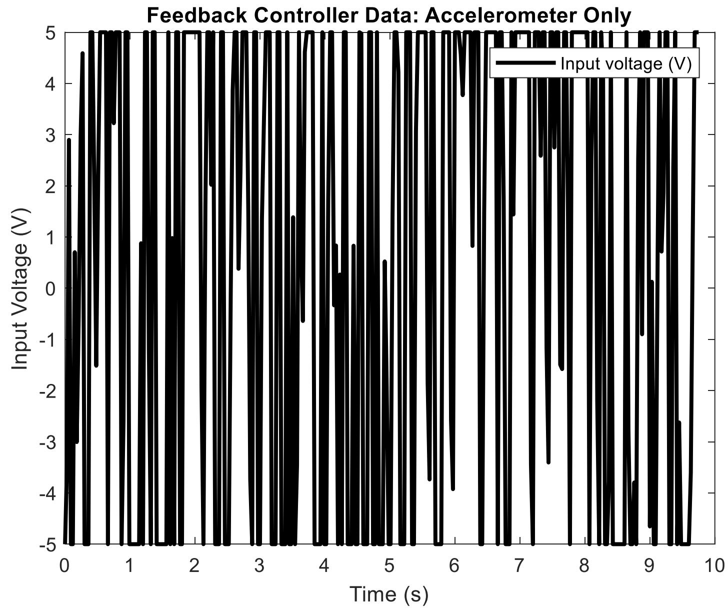


Figure 36. Input voltage vs. time plot for data collected from closed-loop feedback system controlling the MinSeg robot using only data collected from the accelerometer.

In this section, the simulation was run using the same parameters as the previous section, however the gain on the complimentary filter was adjusted so that the state of the pendulum angle was determined solely by the gyroscope data. The simulation was run, and the various states of the system were plotted against time including wheel position, pendulum angle, wheel velocity, and pendulum angle vector as shown in Figures 37, 38, 39, and 40 below respectively. Input voltage was also plotted vs. time in Figure 41.

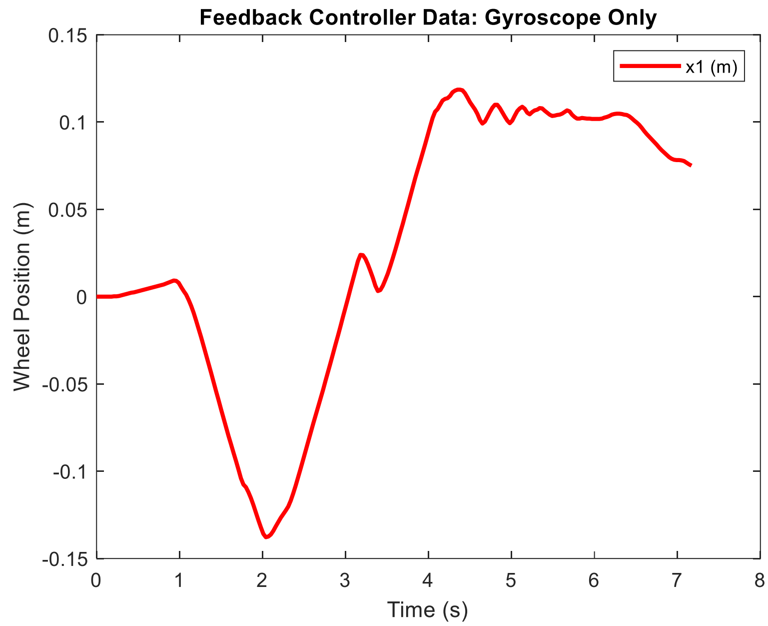


Figure 37. Wheel position vs. time plot for data collected from closed-loop feedback system controlling the MinSeg robot using only data collected from the gyroscope.

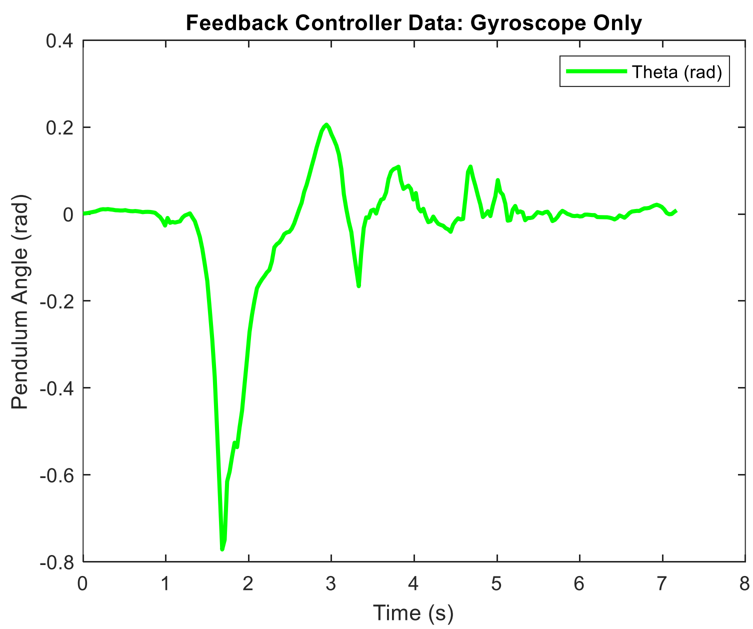


Figure 38. Pendulum angle vs. time plot for data collected from closed-loop feedback system controlling the MinSeg robot using only data collected from the gyroscope.

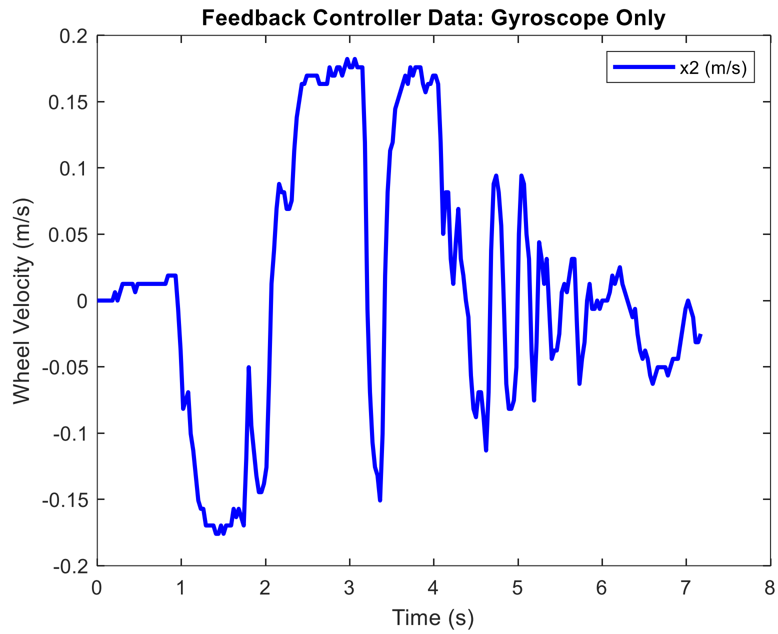


Figure 39. Wheel velocity vs. time plot for data collected from closed-loop feedback system controlling the MinSeg robot using only data collected from the gyroscope.

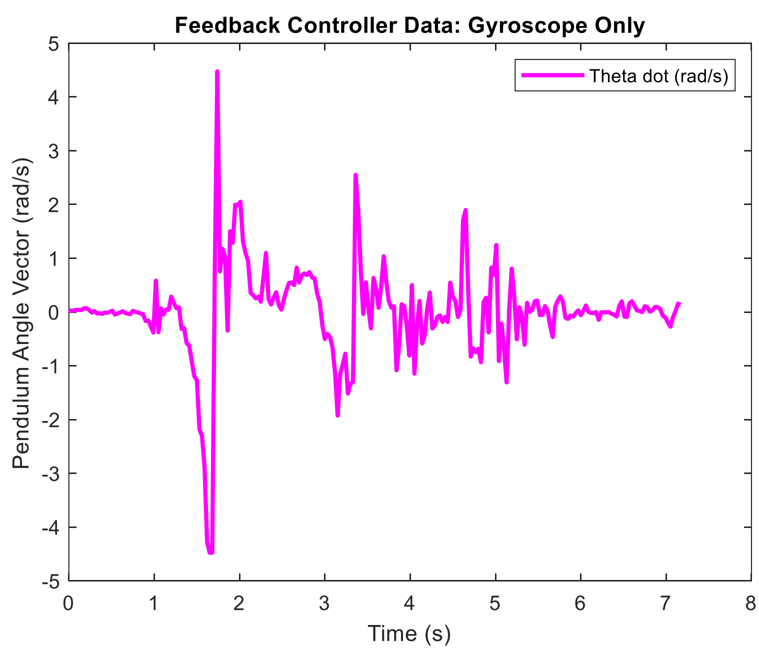


Figure 40. Pendulum angle vector vs. time plot for data collected from closed-loop feedback system controlling the MinSeg robot using only data collected from the gyroscope.

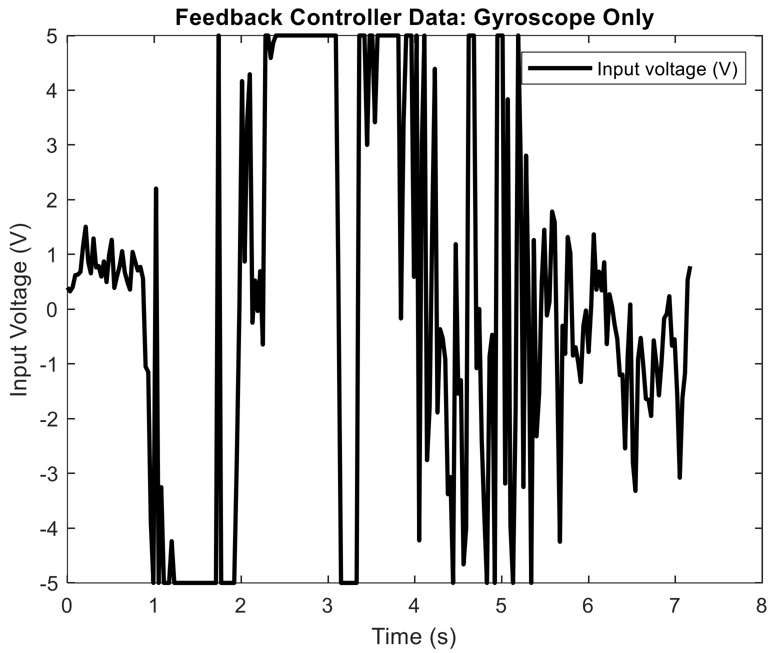


Figure 41. Input voltage vs. time plot for data collected from closed-loop feedback system controlling the MinSeg robot using only data collected from the gyroscope.

Conclusions

I learned a lot in the process of completing this project. While I was already familiar with using the laws of physics to model a system and create differential equations of motion, I had no prior experience with state space models or applying said equations to a physical project to solve a real-world problem. I gained a deeper understanding of the difference between open loop systems and the various types of closed loop feedback controllers. I also learned how to tweak the performance characteristics of the feedback controller and the different metrics that would be used to analyze said performance characteristics.

I was able to get the robot to balance successfully and fine tune the LQR settings for reasonably optimal performance characteristics. If I had more time to invest in this project, I would have liked to perform further trial and error to gain a deeper understanding of how the choice in weight matrices would influence the performance system of a whole. I successfully balanced my robot through trial and error, but I would like to understand more as to why certain weights make the system balance better.

References

MinSegShield M1V4 Single Axis MinSeg Kit: Minseg.Com, MinSeg.Com, https://minseg.com/products/minsegshield-m1v3-2-single-axis-minseg-uno-mega-due-compatible. Accessed 20 Nov. 2023.