Lab 6a: Pole Placement for the Inverted Pendulum

Name: Mingjun Wu, Matthew Domine

Date: 4/19/2021

Objectives

The objective of this lab is to achieve simultaneous control of both the angular position of the pendulum and horizontal position of the cart on the track using full-state feedback. We will be considering small angle perturbations and sine wave reference tracking of the cart position. Note that the system is a SIMO – a Single Input Multiple Output – system, since we are trying to control both the position of the cart and the angle of the pendulum by using only the motor voltage.

Theory

The setup consists of a pendulum attached to the movable cart from Labs 3 and 4. The free body diagram of this setup is shown in Figure 1. We ignore friction and assume that the mass of the rod is uniformly distributed, e.g. its center of mass is located at the center of the rod, Lp = L/2. N and P are the horizontal and vertical components, respectively, of the reaction force between the cart and the pendulum. The parameter values of the physical system are given in Table 1 in the Appendix.

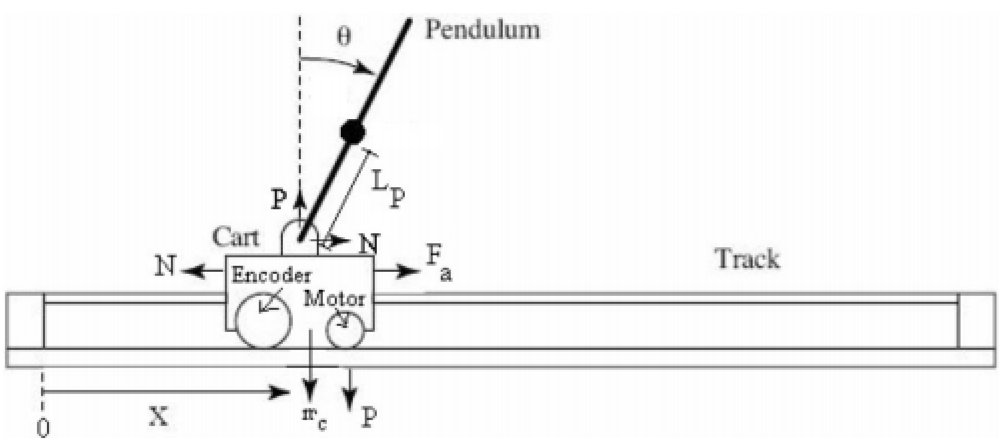

Figure 1: Free body diagram of the inverted pendulum setup (ignoring friction)

Analysis and Results

Pre-Lab

1. **Equations of Motion of the Mechanical System**

Under the small-angle approximation sin θ ≈ θ and cos θ ≈ 1, derive the equations of motion (1) and (2) of the inverted pendulum-cart system. In (1), Fa is the force exerted on the cart by the motor.

$$(M+m)\,\ddot{x} + mL_p\ddot{\theta} = F_a \tag{1}$$

$$mL_p\ddot{x} + \frac{4mL_p^2}{3}\ddot{\theta} - mgL_p\theta = 0 \tag{2}$$

One way of doing this is by considering the free-body diagrams of the cart and the pendulum separately and writing their respective equations of motion.

Hint: Consider the following force/torque balance equations: horizontal acceleration on the cart, acceleration of the center of mass of the pendulum, and torsion of the pendulum. An easy way to analyze the torque equation for the pendulum is to consider its motion as observed from the frame of the cart. However, this will introduce a ‘fictitious’ force, known as D’Alembert’s effect. The associated free body diagram of the cart and the pendulum are given in Figure 2 and Figure 3, respectively. Apply Newton’s laws to obtain the required equations.

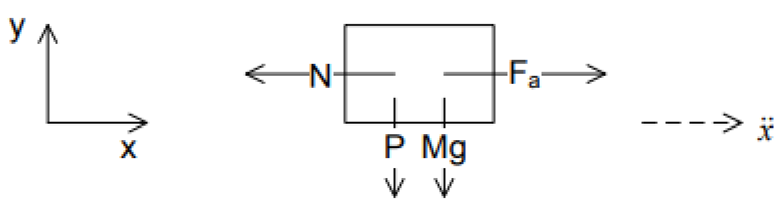

Figure 2: Free body diagram of the cart (ignoring friction; normal force not depicted)

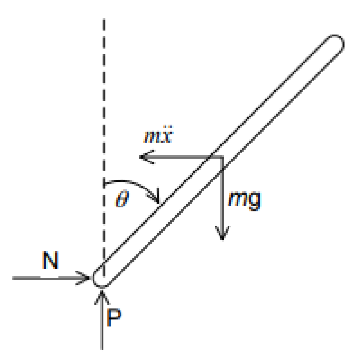

Figure 3: Free body diagram of the pendulum (ignoring friction; the $m\ddot{x}$ vector will appear in the torsion equation due to D’Alembert’s effect)

2. Full System Dynamics of Linearized System

1. Use the motor dynamics derived in Lab 3 (in the form Fa = f(V, x, x˙)) and substitute them into the linearized cart-pendulum dynamics from the previous section to obtain the complete system dynamics. The outputs of interest for us are x, the position of the cart, and θ, the pendulum angle. The control input available is V , the voltage applied to the motor. Thus, our system is a 1-input, 2-output system (SIMO).
2. Now derive the state-space model (A, B, C and D matrices) for the complete system. Use x = x x θ ˙ ˙θ T as your state vector. Make sure you do this derivation symbolically. Your A and B matrix should be of the following form:

$$A = \begin{bmatrix} 0 & a_{12} & 0 & 0 \\ 0 & a_{22} & a_{23} & 0 \\ 0 & 0 & 0 & a_{34} \\ 0 & a_{42} & a_{43} & 0 \end{bmatrix} \qquad B = \begin{bmatrix} 0 \\ b_2 \\ 0 \\ b4 \end{bmatrix} \tag{3}$$

3. Once you have the expressions for your state space representation, use a MATLAB script file to plug in your values. This will make things much easier if parameters need to be adjusted.

3. Analysis and Controller Design

1. Determine the eigenvalues of the state matrix A and the poles of the state-space representation. Is the open-loop system internally stable? Is it BIBO stable? (for this part, you can use the MATLAB command eig).
2. Simulate the output response of the system for a step input in the motor voltage. What would you expect to happen to x and θ in the physical system (assuming infinite track length)? What are the discrepancies between the simulation and the physical system? Why might be the reasons for these discrepancies?
3. We will use a full state-feedback controller to achieve the desired performance specifications. For the purpose of design, we assume that all the state variables are available for measurement and can use them for feedback (i.e. the entire state vector x is known). The full-state feedback controller is u = −Kx. The gain matrix K is chosen such that the closed-loop eigenvalues lie at some desired values. These desired values of the eigenvalues are found based on the performance specifications desired to be achieved. We would like our closed-loop eigenvalues to lie at s1,2 = −2.0 ± 10j and s3,4 = −1.6 ± 1.3j. In lecture you will discuss design techniques for determining these values. For the purpose of this lab we will assume them given.
    a. The feedback gain matrix K is of the form K = [k1 k2 k3 k4]. The closed-loop system matrix is given by AK = A−BK. Using your A and B matrices (with the values plugged in) from (3), compute the matrix AK as a function of the ki .

b. Compute the characteristic polynomial P(K; s) = det(sI − AK) of the closed-loop system as a function of k1 through k4.
c. Compute the desired characteristic polynomial Pdes(s) = Q4 i=1(s−si) determined by the desired locations of the closed-loop poles given above
d. Comparing the coefficients of P(K; s) and P(s), determine the system of linear equations that the gains k1, . . . , k4 have to satisfy. Plug in numerical values and use MATLAB to solve for K.
e. Now verify the result you obtained for K using MATLAB. You may use the acker or place commands for pole placement.

4. With input u = K(r − x), the dynamics of the closed-loop system are x˙ = Ax + Bu = AKx + BKr, where r is your reference input. Observe that the reference input has the same dimension as the system state. Using the calculated gain matrix K and MATLAB, calculate the closed loop transfer function from the first component in the reference r (position reference input) to the output x (the position of the cart). Letting the position reference input, Rpos(s), be a sine wave, plot the position output using frequencies ω = 1, 2, 5 rad/s. Plot these on separate plots with the corresponding reference input superimposed. Be sure that the plots show at least 3 whole periods. Hint. In the frequency domain, the sine wave input is:

$$R_{\text{pos}}(s) = \frac{\omega}{s^2 + \omega^2}$$

Lab

1. Implementing the Controller in Simulink

Implement the state-feedback controller with the lab_6_pseudo_hardware_library.slx provided to you. Here is a review of some of the blocks you will use:

- Cart with Pendulum block: The input is in volts, and the outputs are the encoder values for the cart wheel and cart's pendulum. Remember, these are not in meters nor radians, you need to convert these! If the pendulum exceeds the track limits (−0.5 m to 0.5 m), the simulation will stop.
- dx/dt derivative blocks: The encoders only provide measurements for x and θ. We will need to approximate ˙x and ˙θ in order to perform full state feedback. In this lab, we will use the derivative blocks for this (taking numerical derivatives should generally be AVOIDED). In one of the following labs, you will build an observer and estimate the ˙x and ˙θ from the measurements of ˙x and ˙θ alone.
- Mux and Demux blocks: This will help keep your diagram tidy. You can combine all four signals (x, x, θ, ˙ ˙θ) into a "vector" signal with the mux block, and break them out with the demux block.
- Gain block: You can use a single gain block to implement the matrix K. Change the "multiplication" option from "element-wise" to "matrix".

- Scopes and “To Workspace” blocks: Don’t forget to instrument your system to allow you to save data for your lab report.

Gear Protection In order to prevent the gear from slipping, you must put a saturation block before the Analog Output block. You have two choices:

- a conventional saturation block, set to ±6V as in Lab 3 and 4
- An Embedded MATLAB code block use the following code:

```
function [u_sat,max_lim,min_lim]= fcn(xdot,u)
    % Setting the bounds
    max_lim = min(8,max(4,10*xdot+5));
    min_lim = min(-4,max(-8,10*xdot-5));
    % Dynamic Saturation
    u_sat = max(min_lim,u);
    u_sat = min(max_lim,u_sat);
end
```

You must connect this block to the $\dot{x}$ signal as well as the control signal for saturation. It applies different bounds depending on $\dot{x}$: once the cart is moving, it allows higher “forward” voltages, while prohibiting big “reverse” voltages. The u sat output is the saturated signal, while max_lim and min_lim allow you to plot/analyze the limits.

Units Watch your units. Everything in this system should match the units you used for your state-space derivation. When in doubt, always work with SI units.

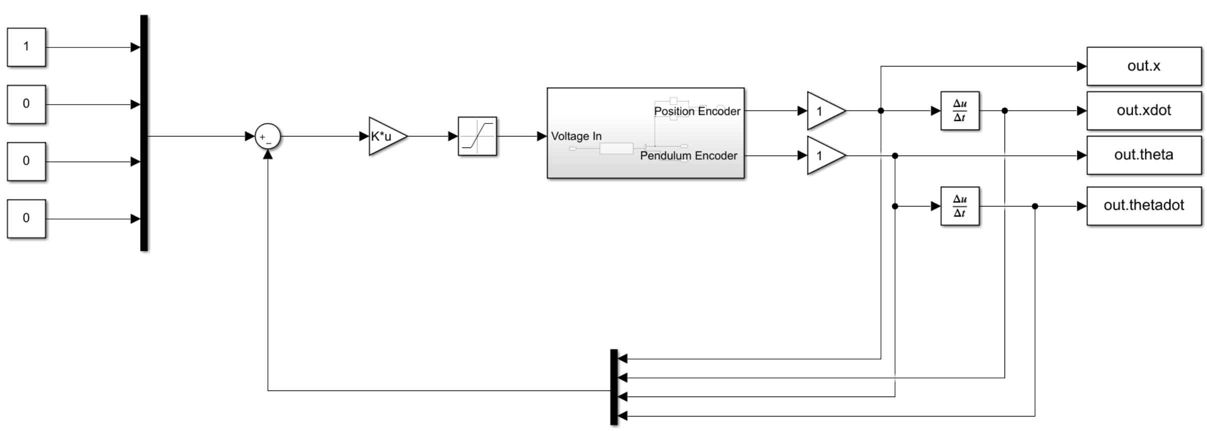

Figure 1: State-feedback Controller With the Lab_6_pseudo_hardware_library.slx

2. Running the Controller on the Hardware

1. Run the controller on the hardware (with reference r set to $r = [0\ 0\ 0\ 0]^T$) and make sure it balances.

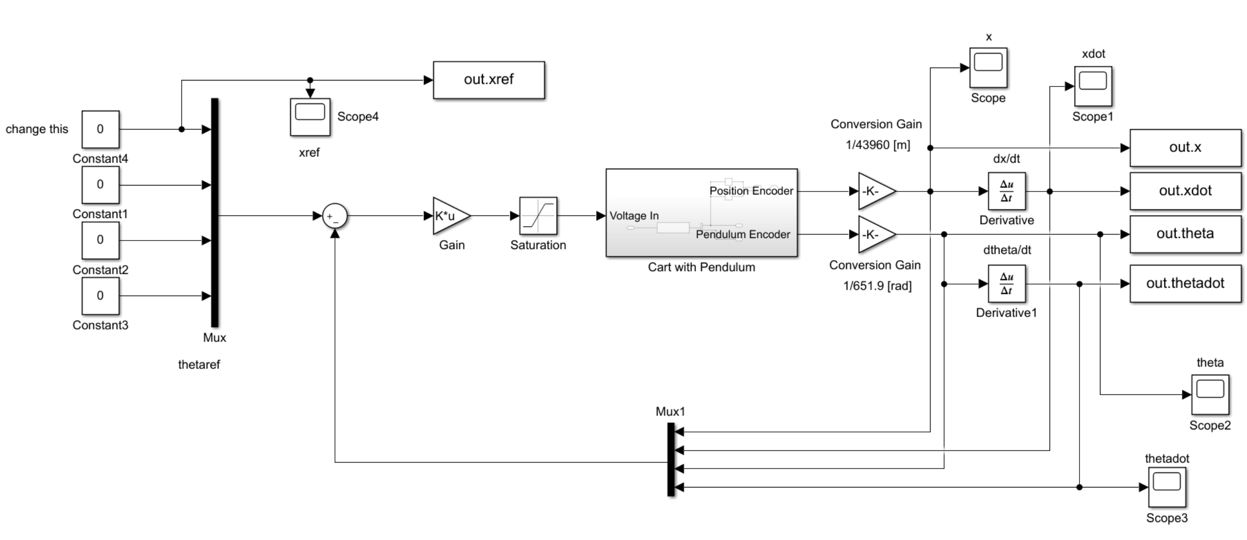

Figure 2: Controller On the Hardware with $r = [0\ 0\ 0\ 0]^T$

```
%% 4.1
clear;clc;close all;
tstop = 20;
out = sim('cartpendulum',tstop);
theta = out.theta.data;
x = out.x.data;
time = out.tout;
plot(time,x)
hold on
yline(0,'r')
xlabel('time(seconds)');ylabel('position(meters)');title('Cart Position')
legend('cart position','reference')
figure
plot(time,theta)
hold on
yline(0,'r')
xlabel('time(seconds)');ylabel('angle(radains)');title('Pendulum Angle')
legend('pendulum angle','reference')
```

Figure 3: Matlab Code

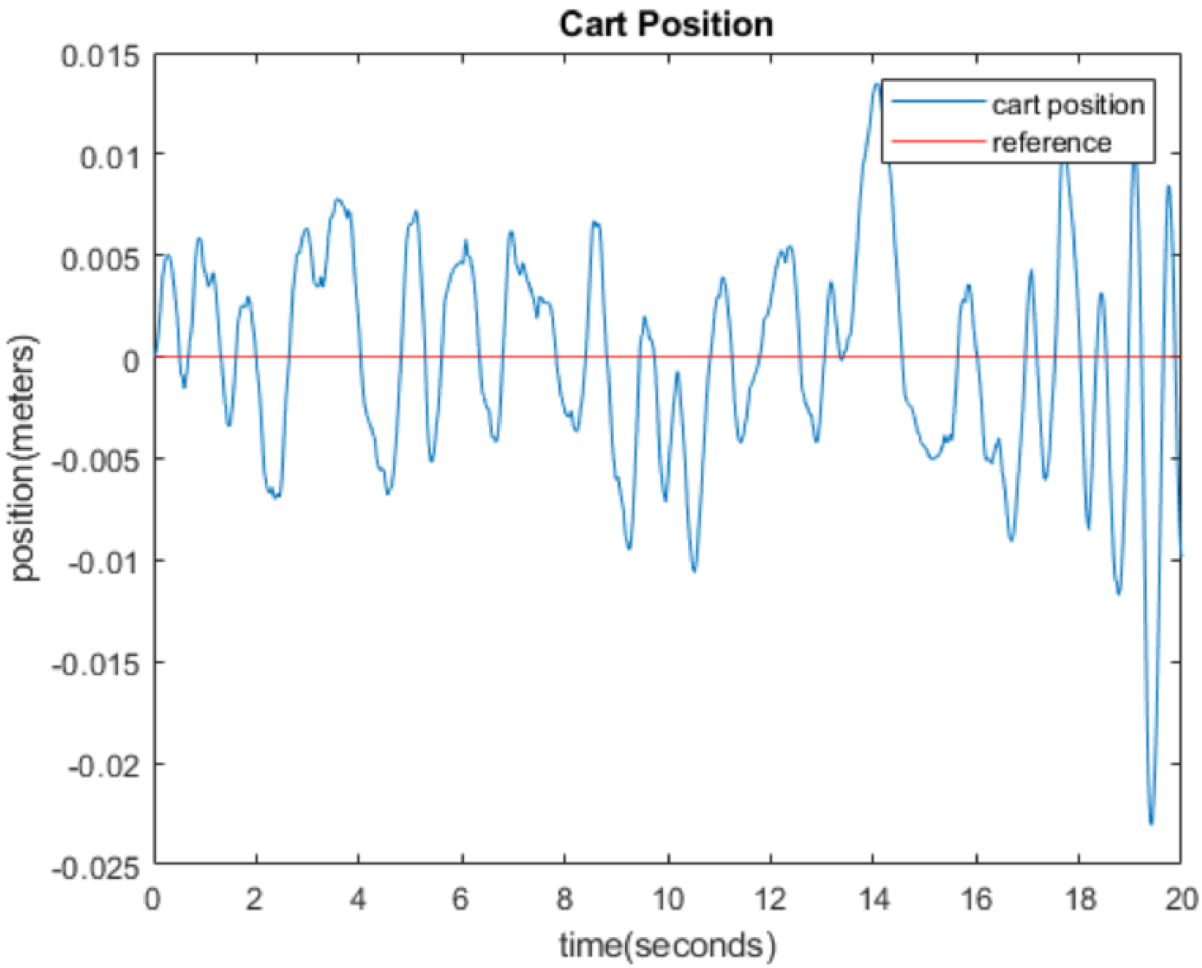

Figure 4: Plot of the Cart Position w/ Reference

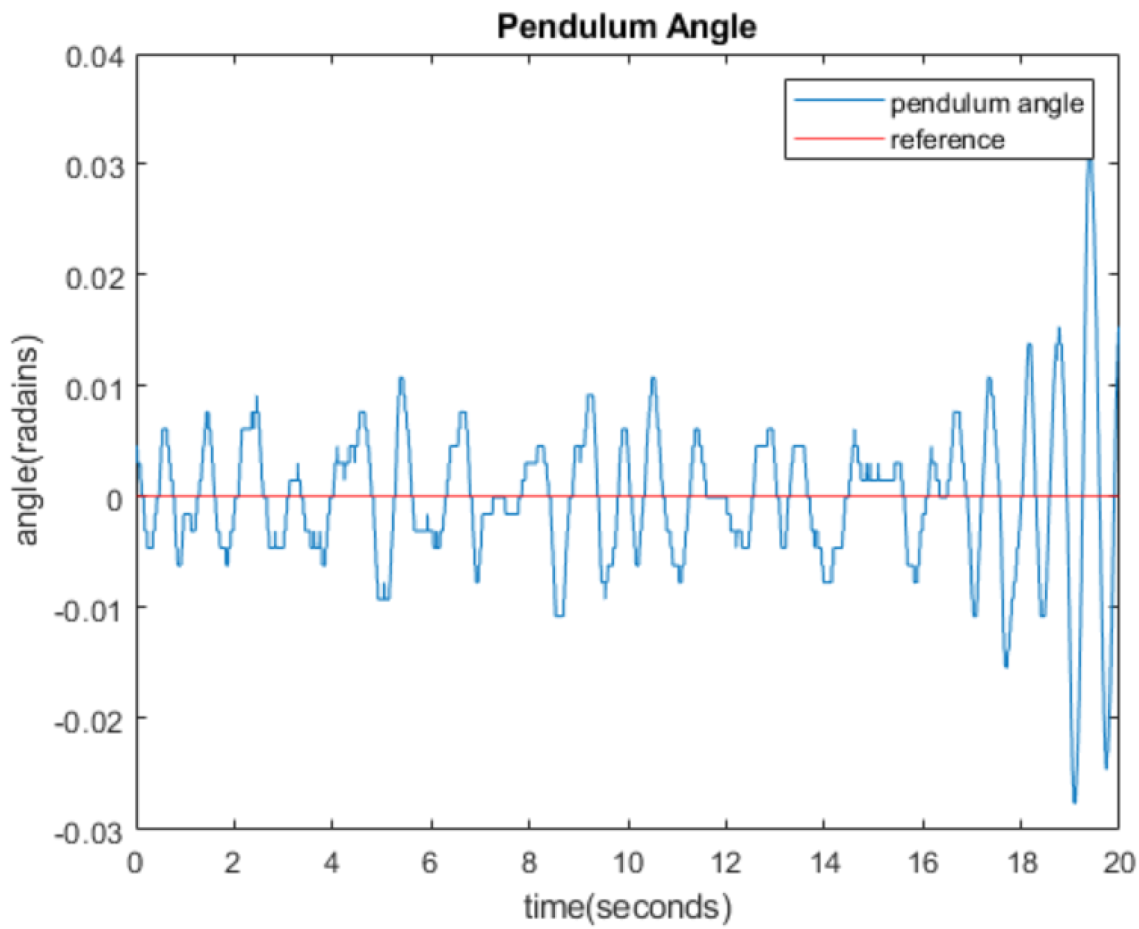

Figure 4: Plot of the Pendulum Angle w/ Reference

2. With the pendulum balancing, manually apply small perturbations to the pendulum and check the response. Make sure to start the system with the cart in the center of the track, so there is enough space for the controller to move the cart and "catch" the pendulum.

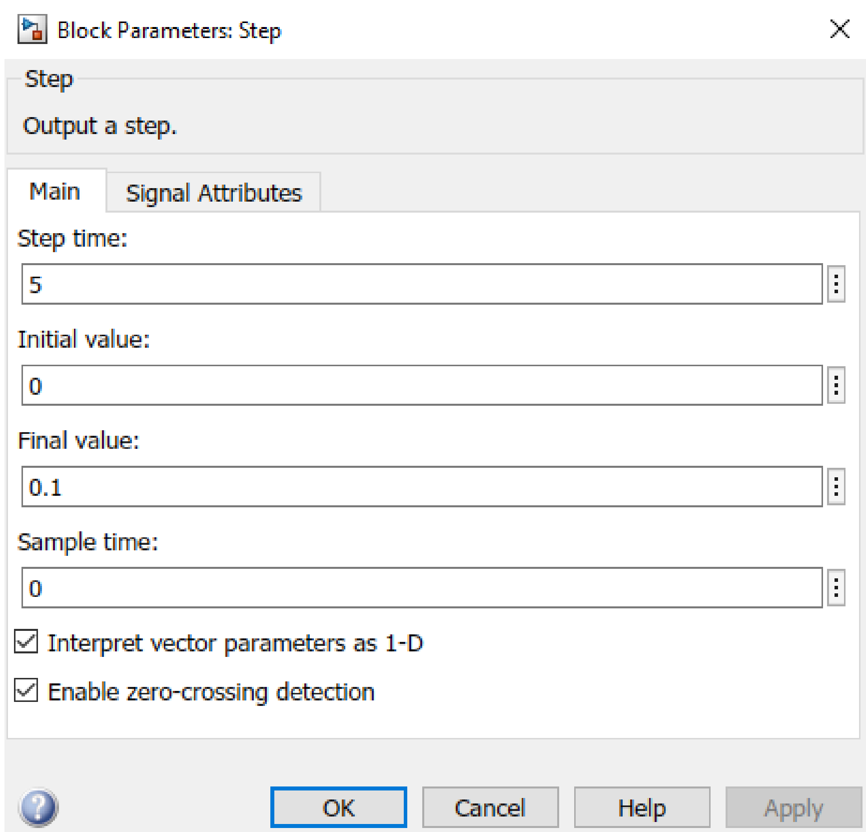

Figure 5: Parameters for the Step Function Block

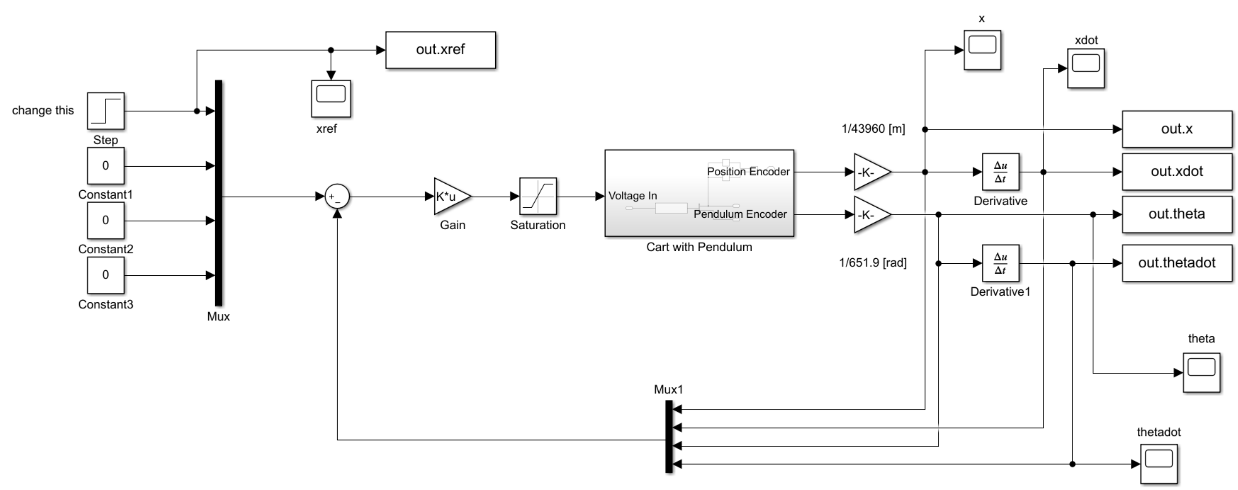

Figure 6: Controller On the Hardware with $r = [step\ 0\ 0\ 0]^T$ *where step is a step input for the position reference of 10 cm 5 seconds after initializing the simulation*

a. Unfortunately we cannot poke the pendulum, instead input a step for the position reference of 10 cm 5 seconds after initializing the simulation (we want to allow the transients to die off). How does the controller actuate the cart (run the simulation a few times to get a general feel for the response)? Hint: note what happens to the pendulum angles first. Why does the system respond like this?
   As it is observed from the cart position plots below, the position response tends to move in an opposite direction for a little right after the initialization and then move toward the desired position which is at 10 cm away. It makes sense that the system response acts like this because if the cart wants to move to the right, then the pendulum would lean backward to the left and could eventually tip over, and hence the controller would try to oppose that by moving in the opposite direction first so that the pendulum could go back to equilibrium or balance itself out. One can note from the pendulum angle plots that the angle of the pendulum changes in the positive direction while the cart position moves in the negative direction at the beginning of the responses.

b. Plot the variation of the cart and pendulum position with time for these 10 cm step inputs. Comment on the controller's general performance. Why does the hardware continue to oscillate about the equilibrium point?
   As the step input is applied to the system the equilibrium is moved (i.e. after 5 seconds the reference position changes to 10 cm) which causes the controller to try to restabilize. This can be observed in the step response where there is a big spike at 5s and then the response moves to and then oscillates about a new equilibrium point. The oscillation of the hardware about the equilibrium point is due to the open loop system being unstable and hence even a very small perturbation could pull the system away from equilibrium which causes the pendulum angle to increase exponentially. The controller attempts to push the system back to its equilibrium point. This continues to happen due to the unstable open loop system would cause it to far way to either the positive or the negative side and the controller keeps pushing it back to zero, therefore it will appear as oscillations in the plot.

```matlab
%% 4.2
clear;clc;close all;
for i = 1:3
tstop = 20;
out = sim('cartpendulumstep',tstop);
theta = out.theta.data;
x = out.x.data;
xref = out.xref.data;
time = out.tout;
figure
plot(time,x)
hold on
plot(time,xref)
xlabel('time(seconds)');ylabel('position(meters)');title('Cart Position')
legend('cart position','reference')
figure
plot(time,theta)
hold on
yline(0,'r')
xlabel('time(seconds)');ylabel('angle(radians)');title('Pendulum Angle')
legend('pendulum angle','reference')
end
```

Figure 7: Matlab Code

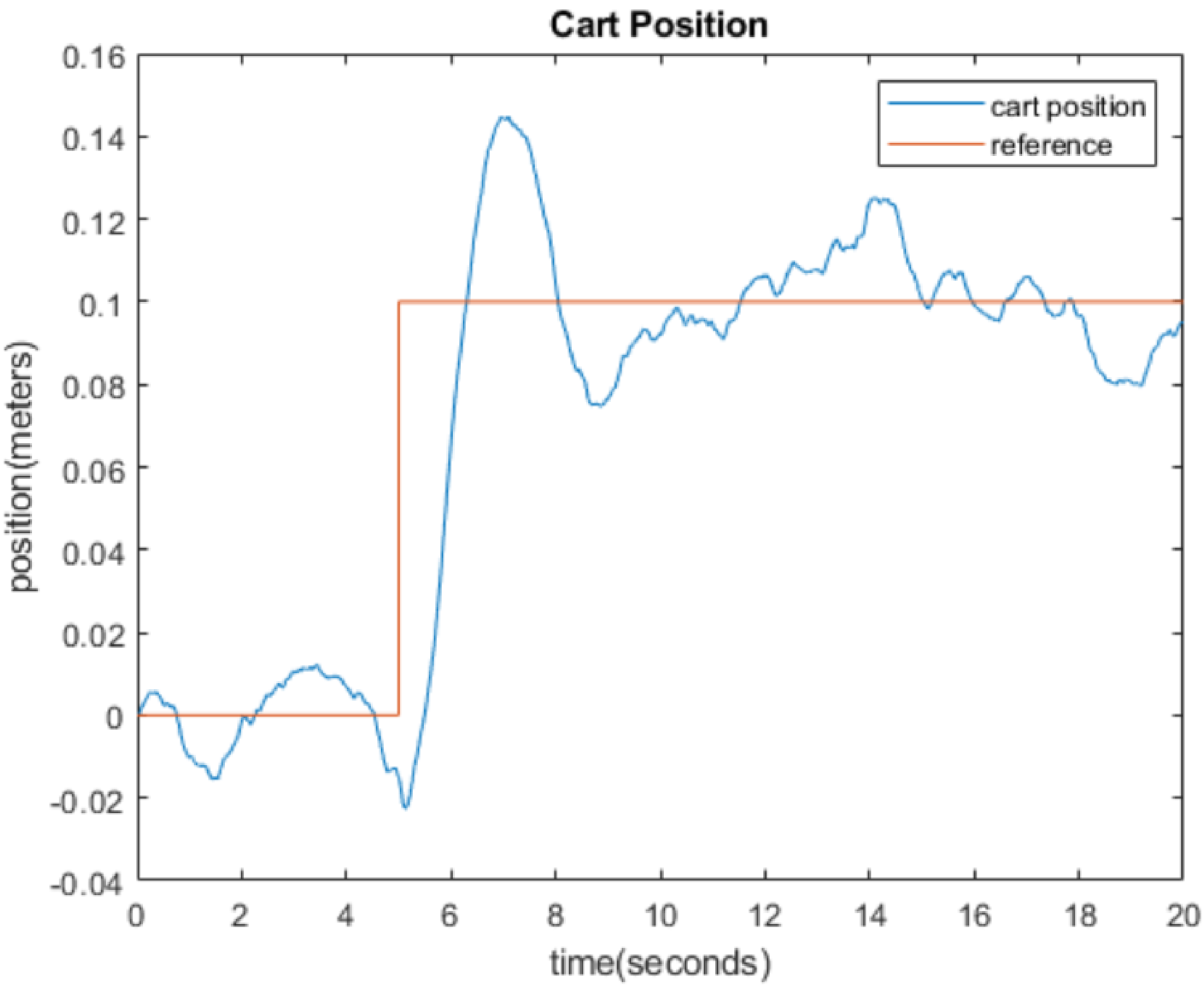

Figure 8: Plot of the Cart Position w/ Reference (Trial 1)

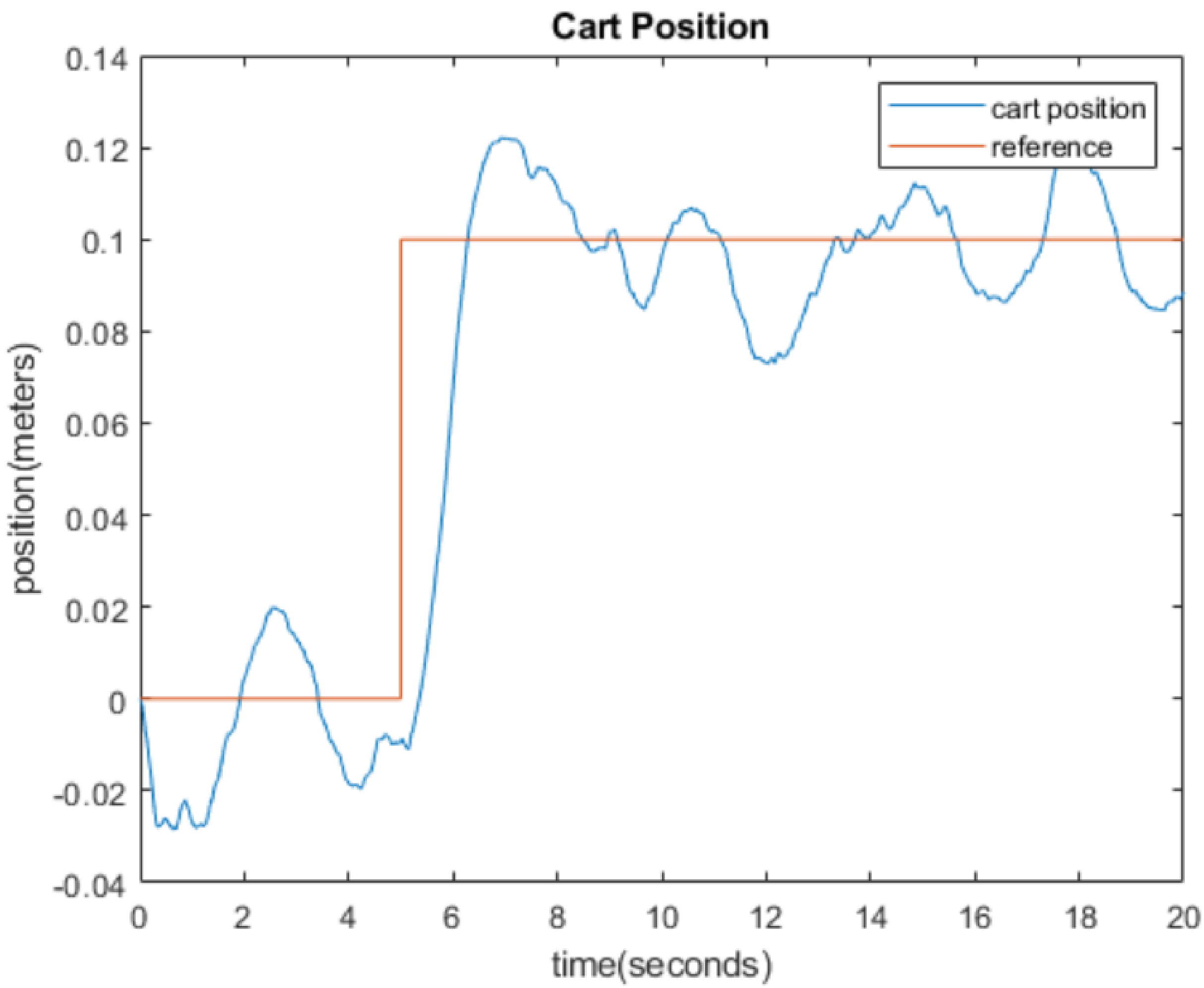

Figure 9: Plot of the Cart Position w/ Reference (Trial 2)

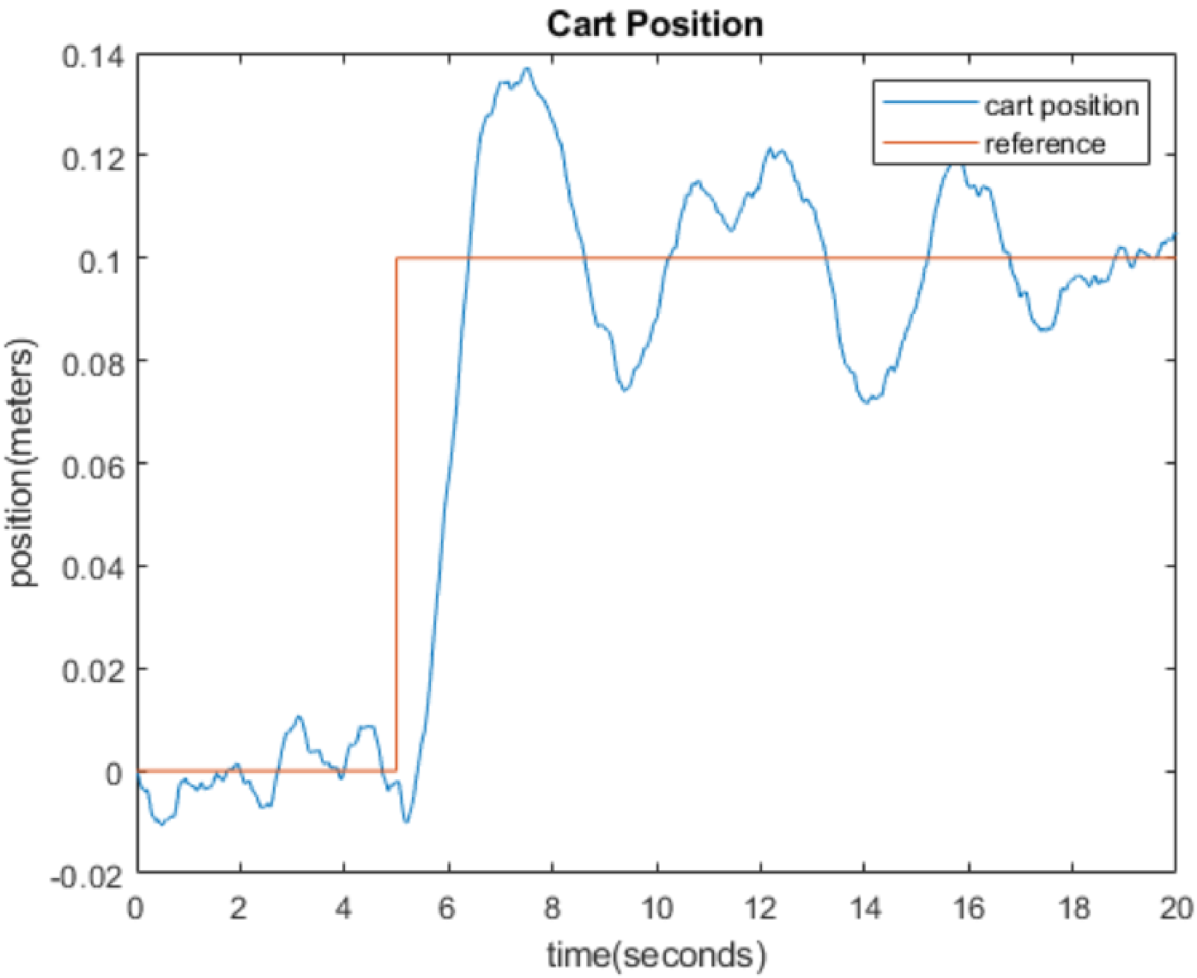

Figure 10: Plot of the Cart Position w/ Reference (Trial 3)

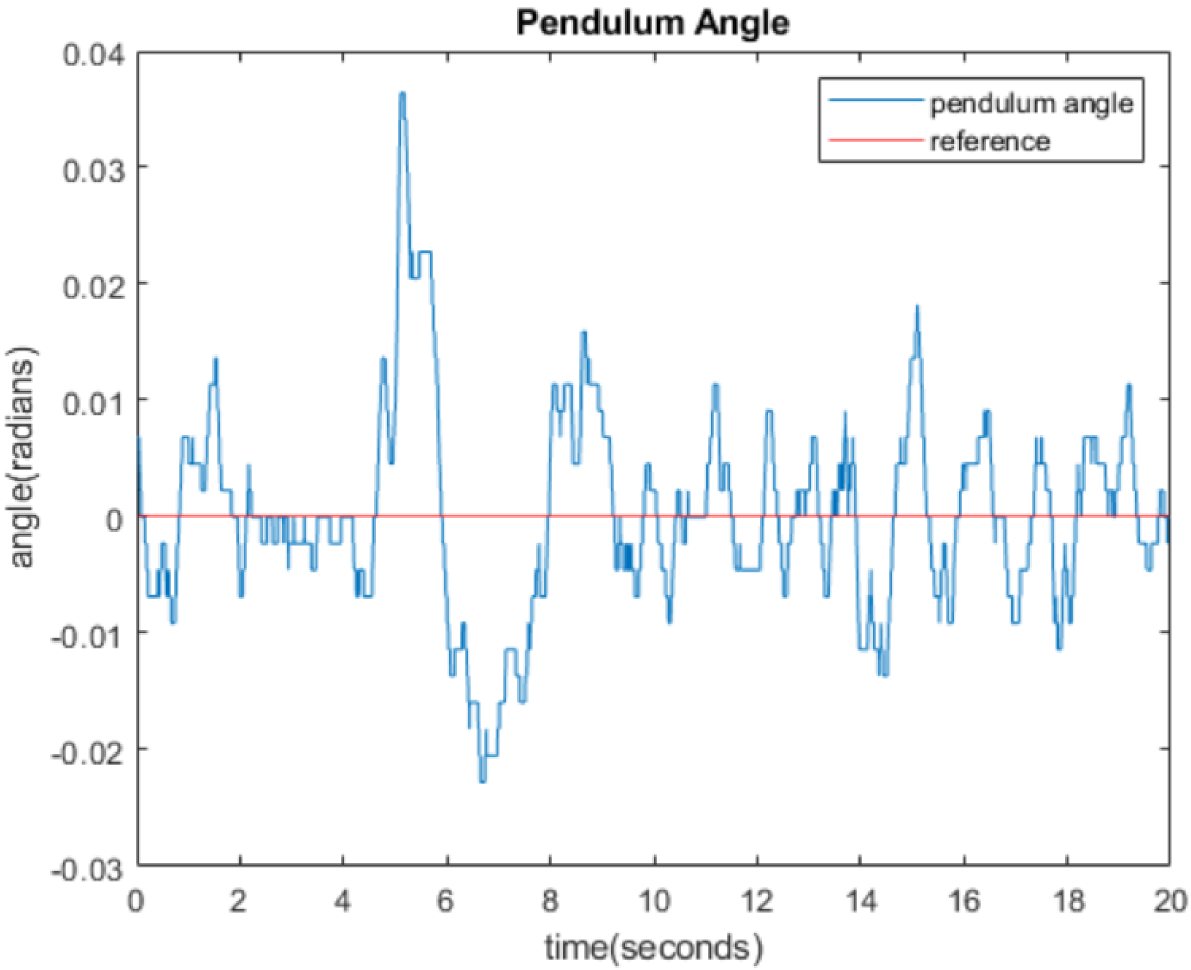

Figure 11: Plot of the Pendulum Angle w/ Reference (Trial 1)

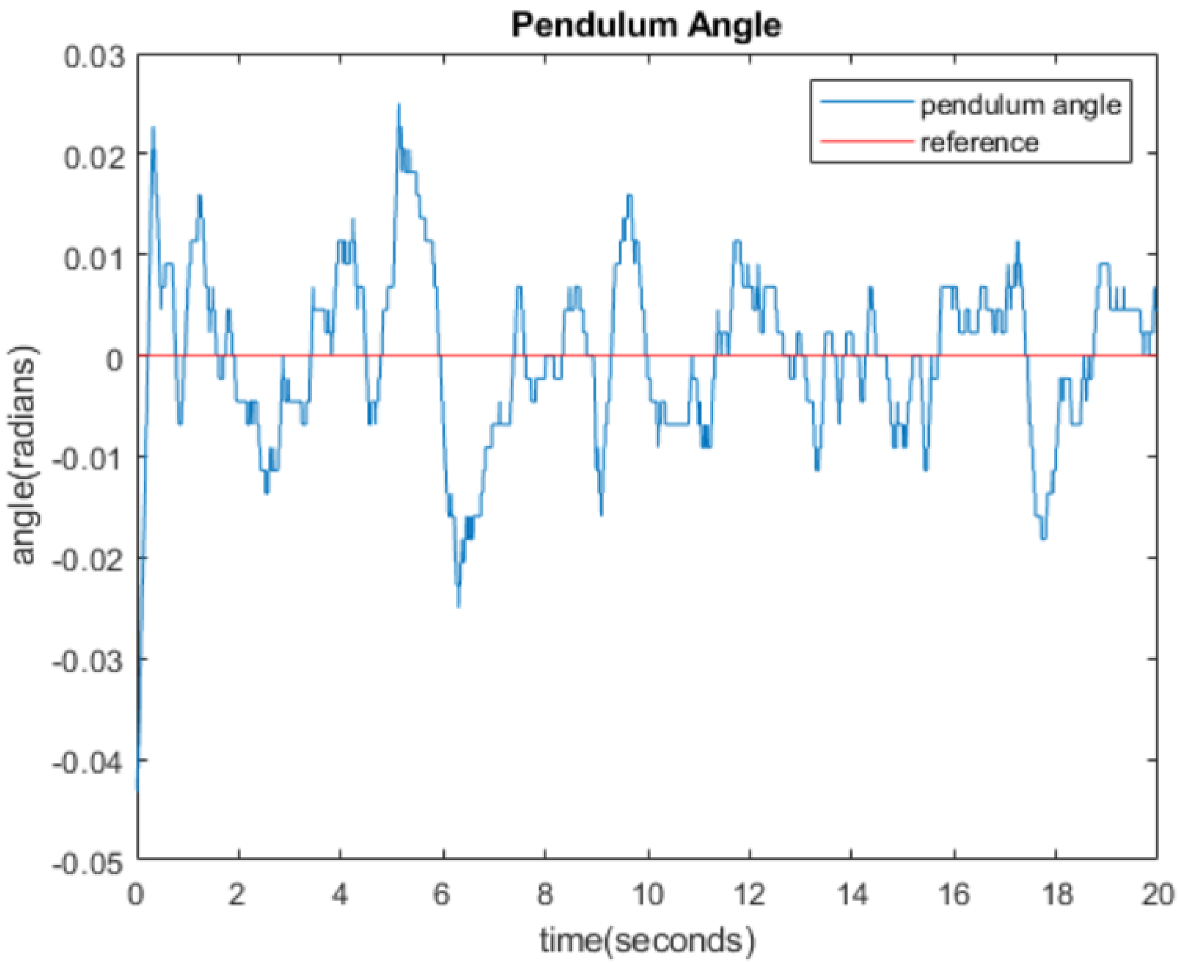

Figure 12: Plot of the Pendulum Angle w/ Reference (Trial 2)

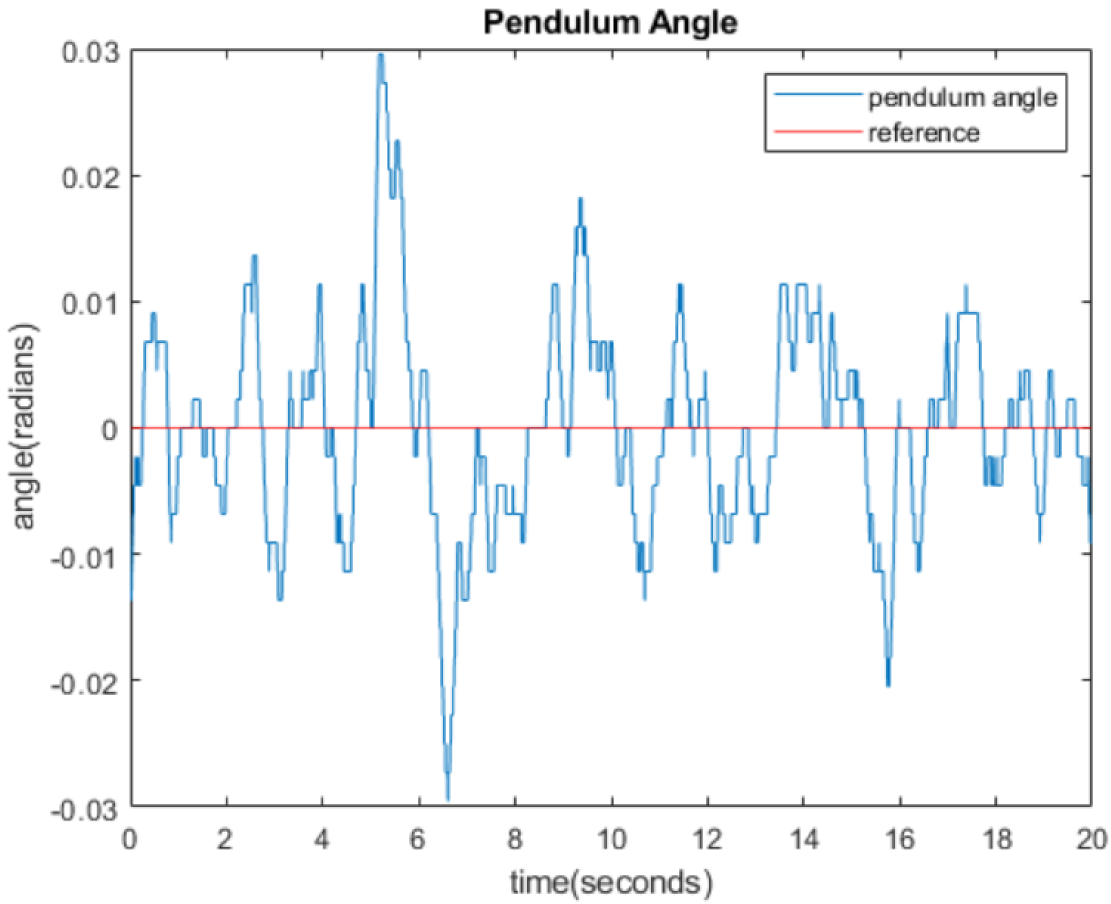

Figure 13: Plot of the Pendulum Angle w/ Reference (Trial 3)

3. Introduce a sine wave reference signal and analyze the results. Your reference input will be $r = [M\sin \omega t\ 0\ 0\ 0]^T$. Start with a reasonable amplitude (M ≈ 0.1 m), and use frequencies ω = 1, 2, 5 rad/s. Check your response and include plots of cart position, cart velocity and pendulum angle in your report (both reference and output signals). Be sure that the plots show at least 3 whole periods.
   a. Compare the outputs of the prelab simulation to the "hardware" simulation. For each frequency in both simulations, specifically note the gain and phase lag. You can estimate these from the plots. Are they close? If not, explain possible causes for the difference.

Note: comment about gain and phase lag are below the plots, please scroll down and find them.

Generally Speaking, the plots in both simulations are not exactly the same but close. One could claim that they are in agreement with each other. The response of the hardware simulation tends to have a larger gain as well as smaller phase lag at both 1 and 2 rad/s, but the gain and the phase lag in both simulations are very close at 5 rad/s. The discrepancy is most likely caused by the linearity of the systems, which a linearized system is used in the prelab simulation and a nonlinear system is used in the hardware simulation.

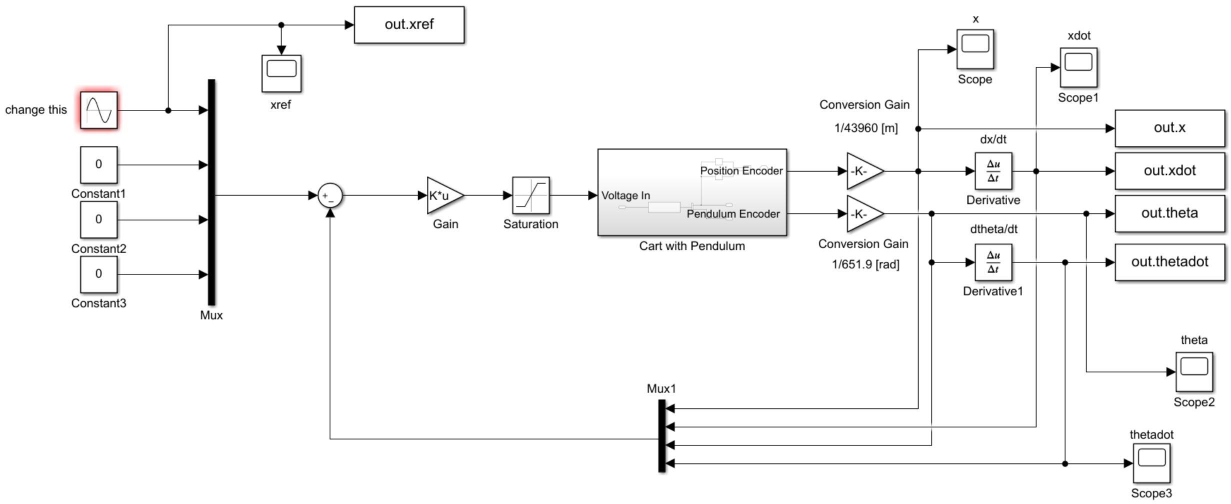

Figure 14: Controller On the Hardware with $r = [M\sin\omega t\ 0\ 0\ 0]^T$ *where* $M = 0.1m$ *and* $\omega = 1, 2, 5$ rad/s

```
clear;clc;close all;
tstop = 20;
out = sim('cartpendulumsineh',tstop);
theta = out.theta.data;
thetadot = out.thetadot.data;
x = out.x.data;
xdot = out.xdot.data;
xref = out.xref.data;
time = out.tout;
plot(time,x)
hold on
plot(time,xref)
xlabel('time(seconds)');ylabel('position(meters)');title('Cart Position')
legend('cart position','reference')
figure
plot(time,xdot)
hold on
yline(0,'r')
xlabel('time(seconds)');ylabel('velocity(m/s)');title('Cart Velocity')
legend('cart velocity','reference')
figure
plot(time,theta)
hold on
yline(0,'r')
xlabel('time(seconds)');ylabel('angle(radians)');title('Pendulum Angle')
legend('pendulum angle','reference')
figure
plot(time,thetadot)
hold on
yline(0,'r')
xlabel('time(seconds)');ylabel('anglular velocity(rad/s)');title('Pendulum Angular Velocity')
legend('pendulum angular velocity','reference')
```

Figure 15: MATLAB Code

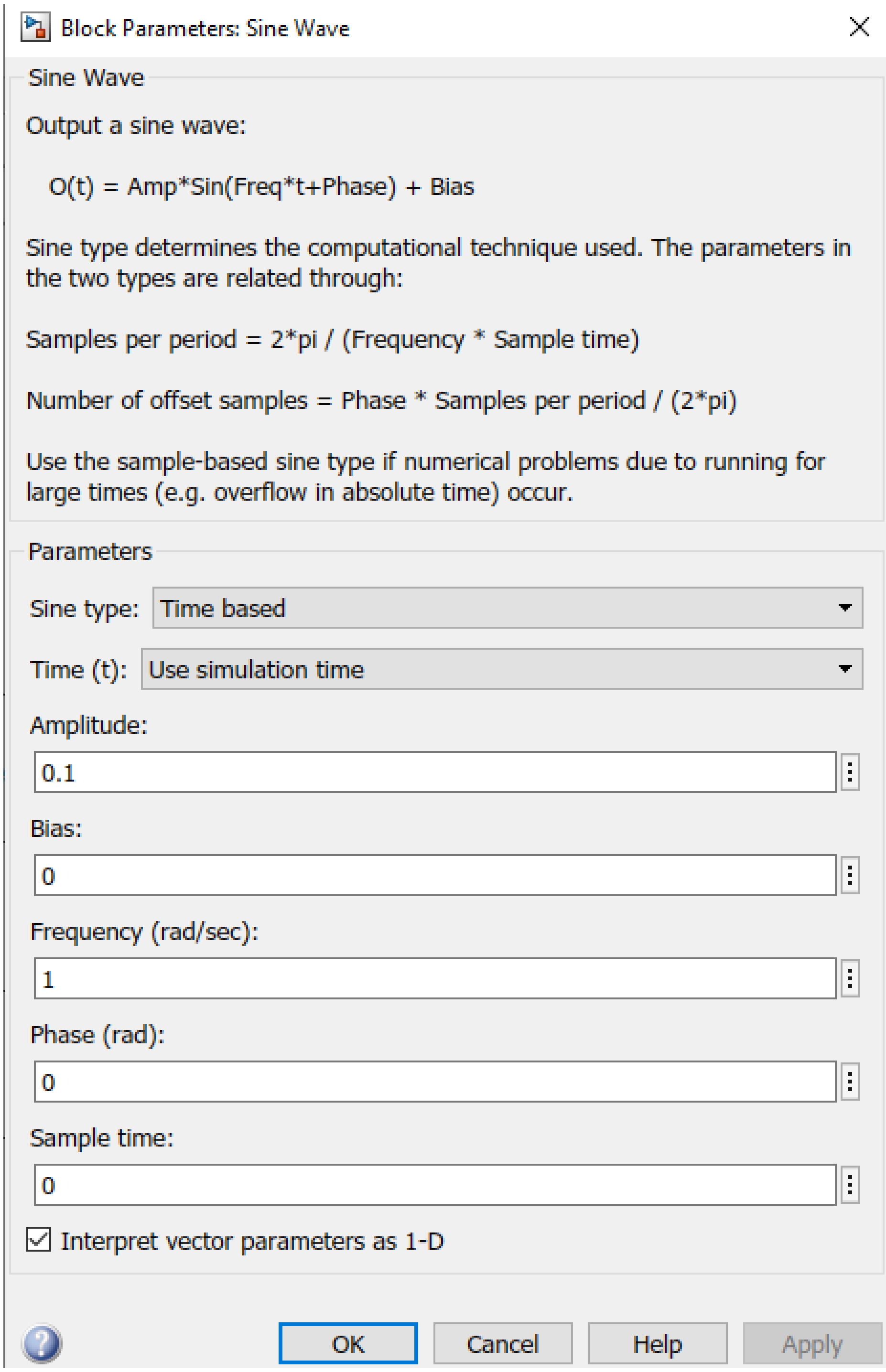

Figure 16: Sine Wave Block Parameters (ω = 1 rad/s)

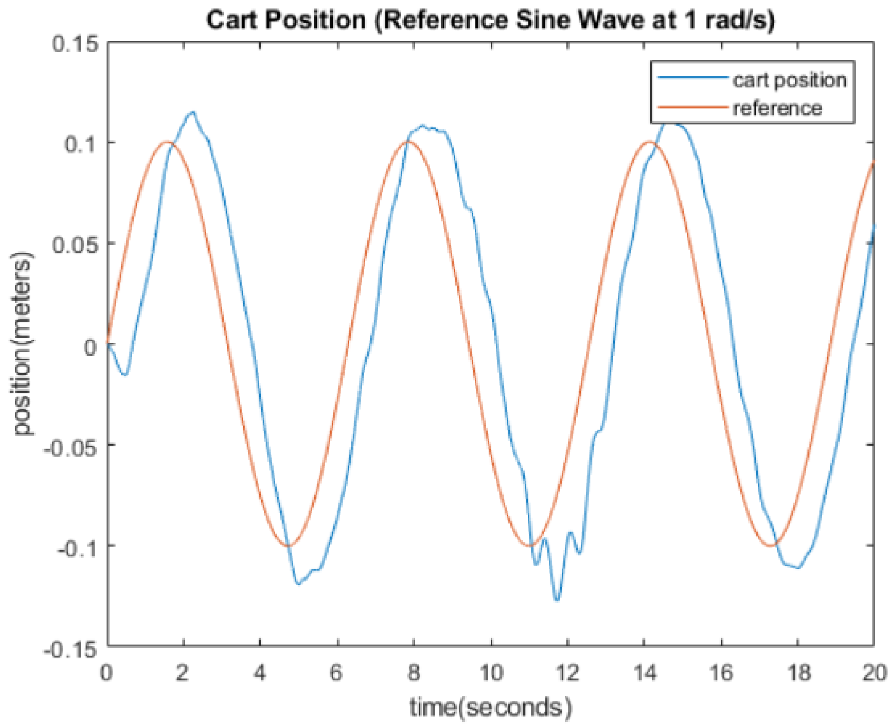

Figure 17: Plot of the Cart Position w/ Reference Sine Wave at 1 rad/s Using Hardware Simulation

The reference response lag the cart position reponses about 30 degrees and the gain is about 1.2.

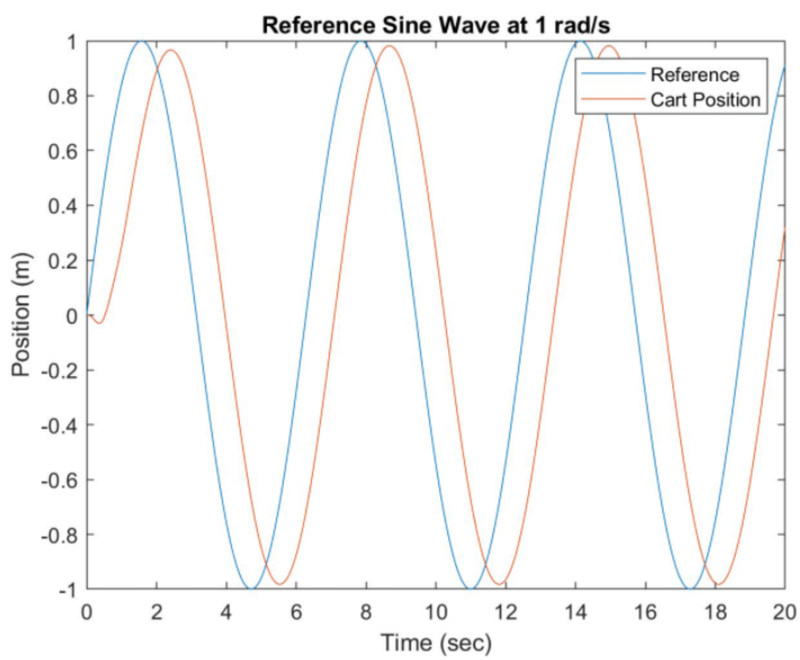

Figure 18: Plot of the Cart Position w/ Reference Sine Wave at 1 rad/s Using Prelab Simulation

The reference response lag the cart position reponses about 45 degrees and the gain is about 0.95.

Conclusion: The hardware simulation has a larger gain and a smaller phase shift than the prelab simulation.

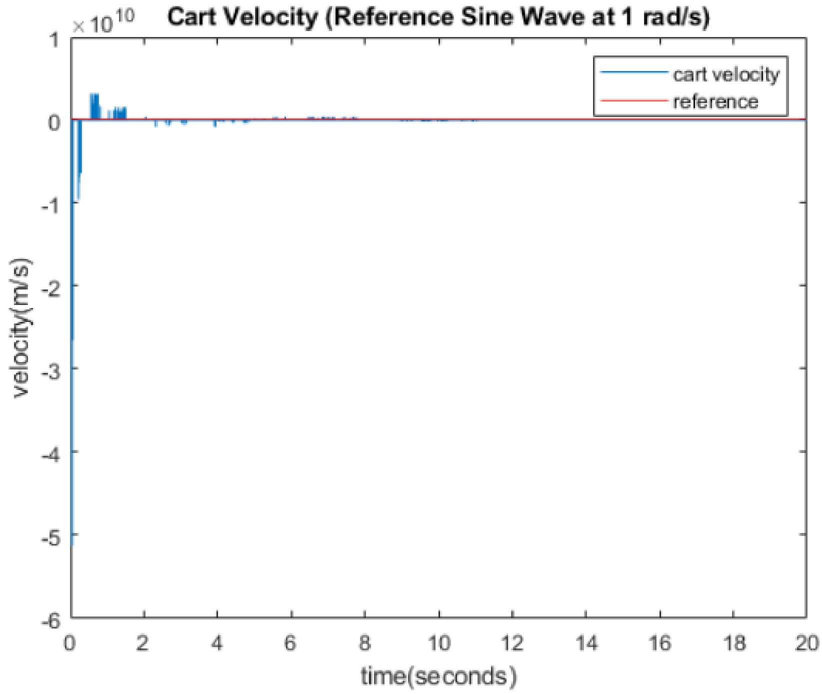

Figure 19: Plot of the Cart Velocity w/ Reference Sine Wave at 1 rad/s

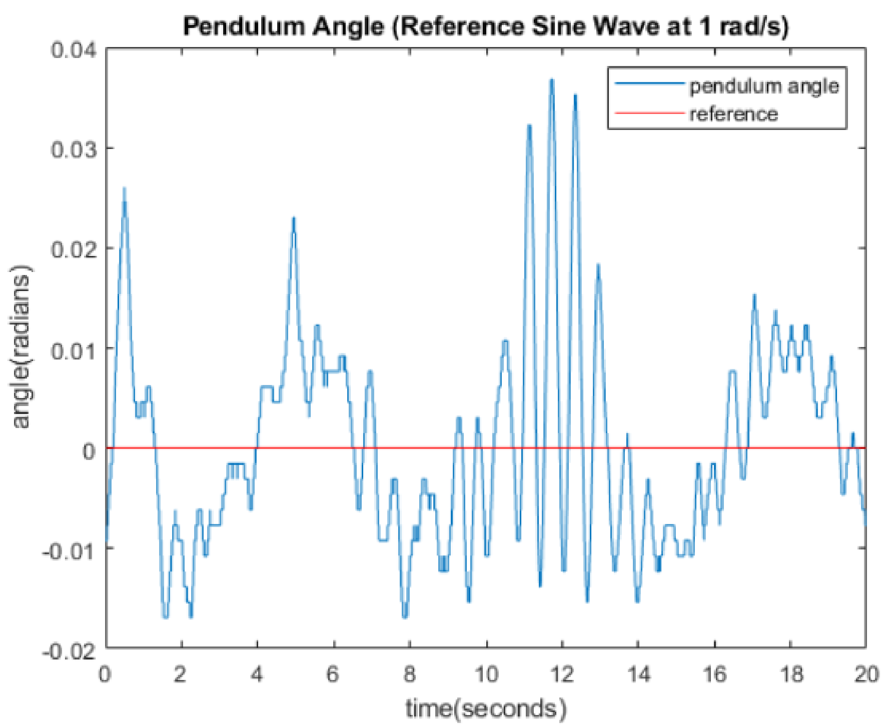

Figure 20: Plot of the Pendulum Angle w/ Reference Sine Wave at 1 rad/s

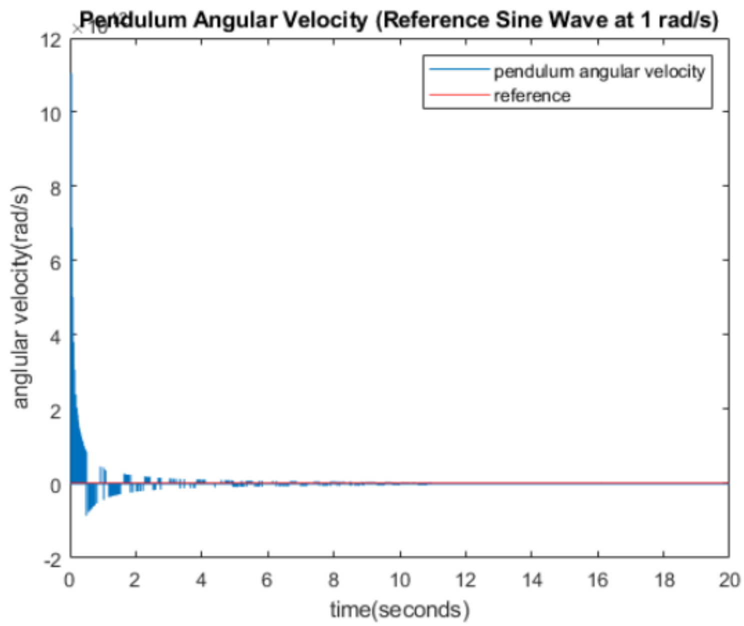

Figure 21: Plot of the Pendulum Angular Velocity w/ Reference Sine Wave at 1 rad/s

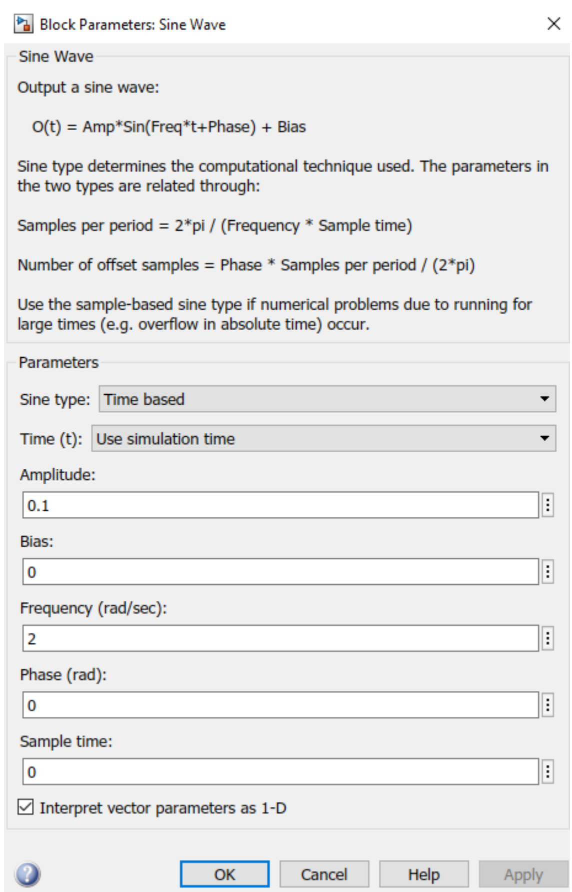

Figure 22: Sine Wave Block Parameters (ω = 2 rad/s)

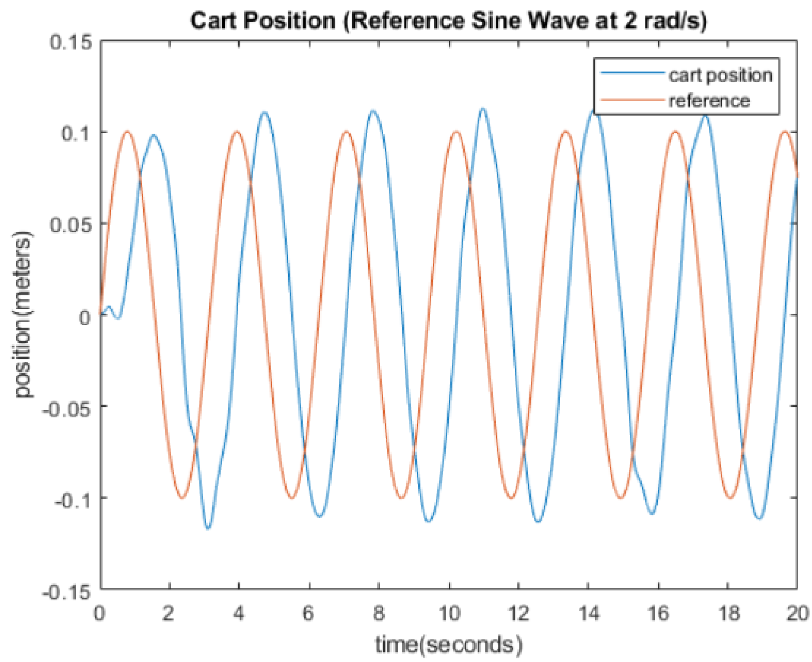

Figure 23: Plot of the Cart Position w/ Reference Sine Wave at 2 rad/s Using Hardware Simulation

The reference response lag the cart position reponses about 45 degrees and the gain is about 1.2.

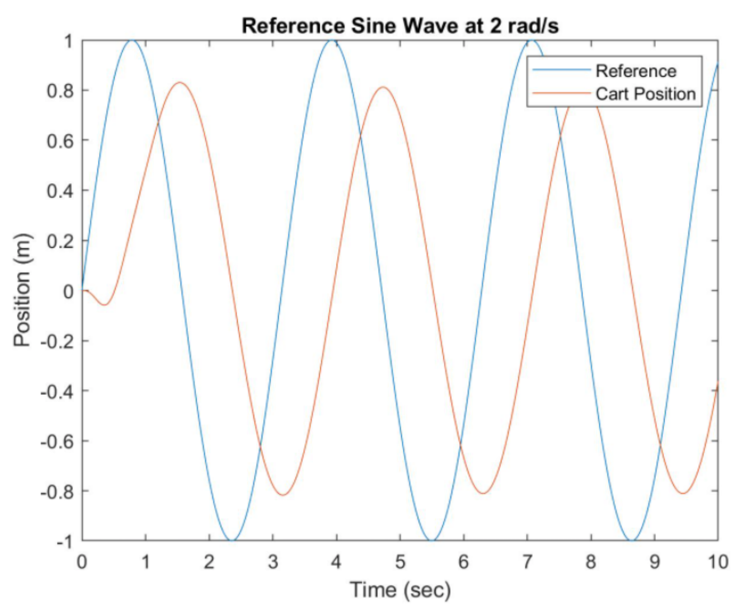

Figure 24: Plot of the Cart Position w/ Reference Sine Wave at 2 rad/s Using Prelab Simulation

The reference response lag the cart position reponses about 60 degrees and the gain is about 0.8.

Conclusion: The hardware simulation has a larger gain and a smaller phase shift than the prelab simulation.

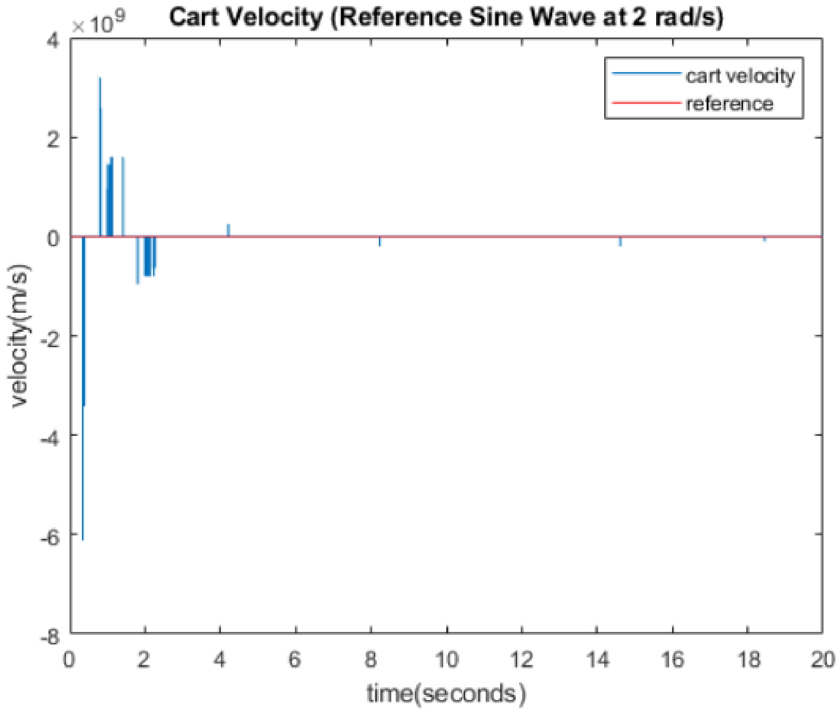

Figure 25: Plot of the Cart Velocity w/ Reference Sine Wave at 2 rad/s Using Hardware Simulation

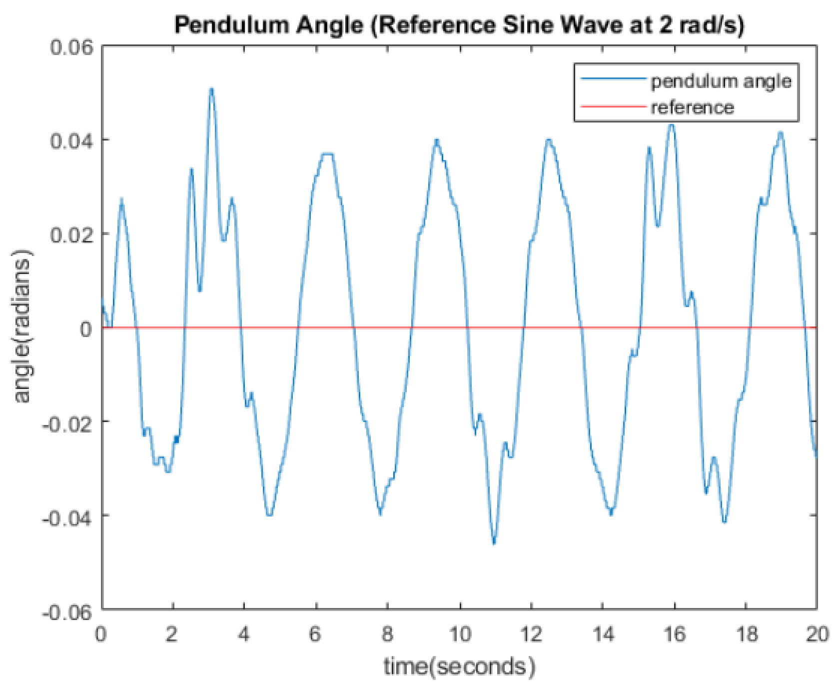

Figure 26: Plot of the Pendulum Angle w/ Reference Sine Wave at 2 rad/s Using Prelab Simulation

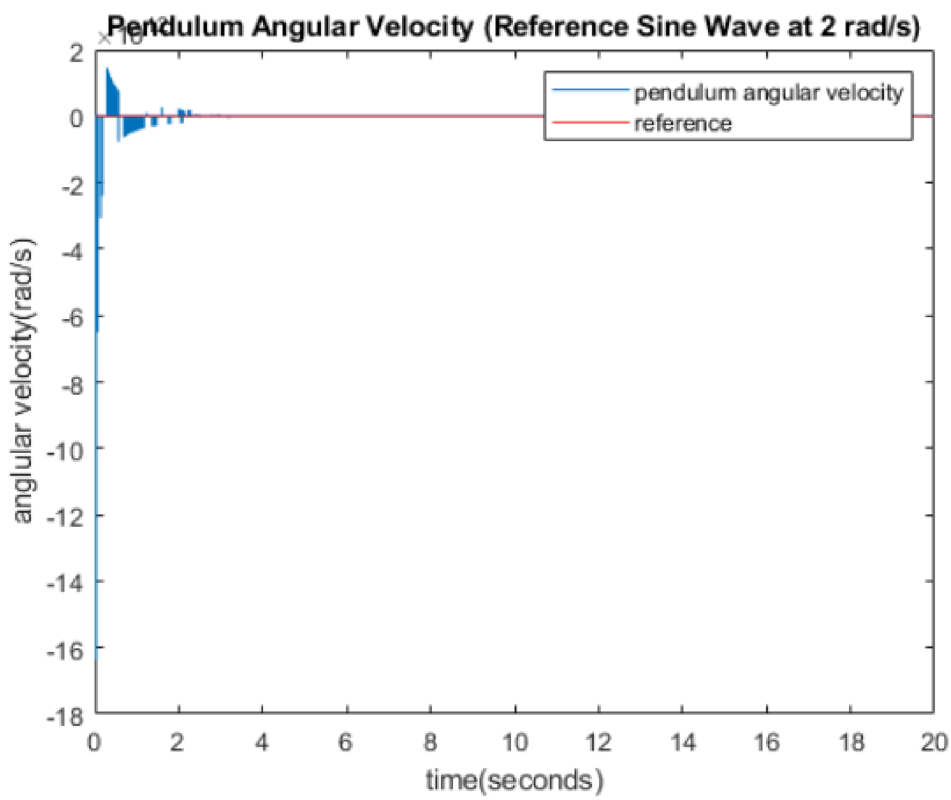

Figure 27: Plot of the Pendulum Angular Velocity w/ Reference Sine Wave at 2 rad/s

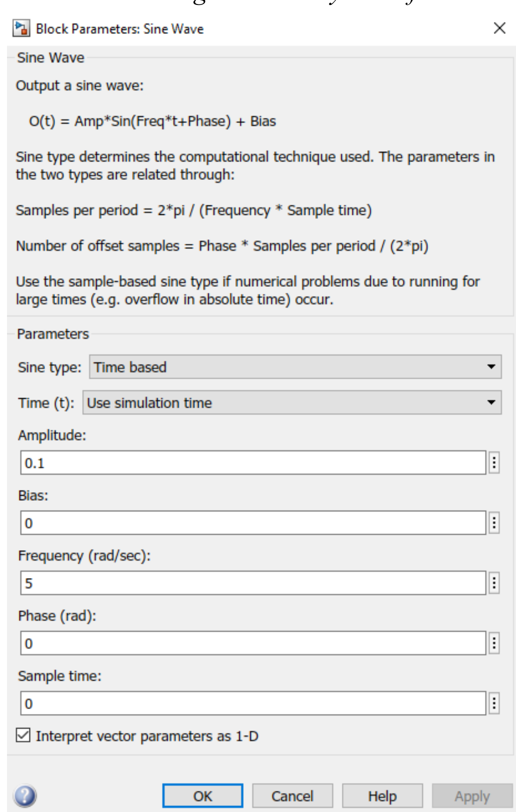

Figure 28: Sine Wave Block Parameters (ω = 5 rad/s)

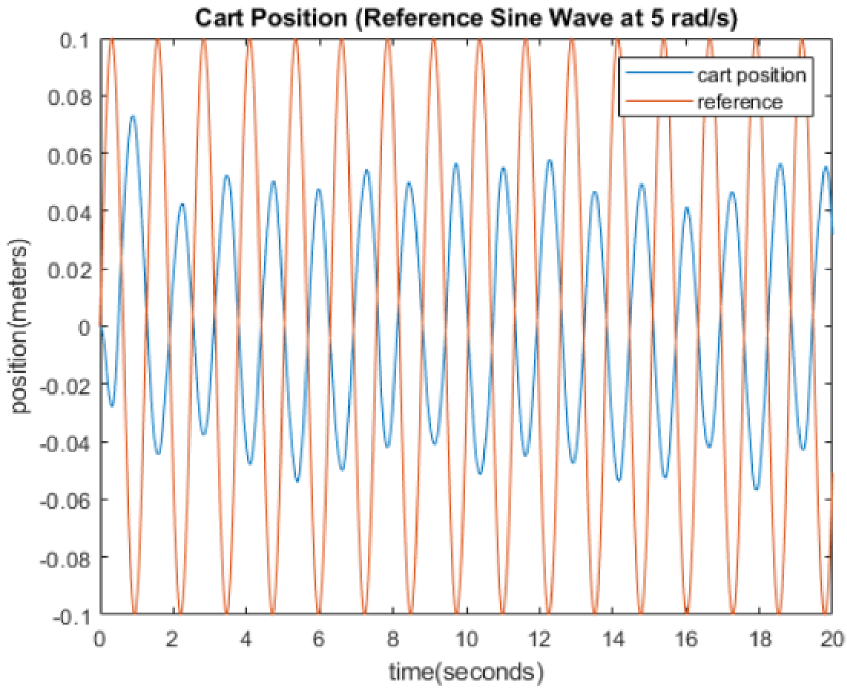

Figure 29: Plot of the Cart Position w/ Reference Sine Wave at 5 rad/s Using Hardware Simulation

The reference response lag the cart position reponses about -180 degrees and the gain is about 0.4.

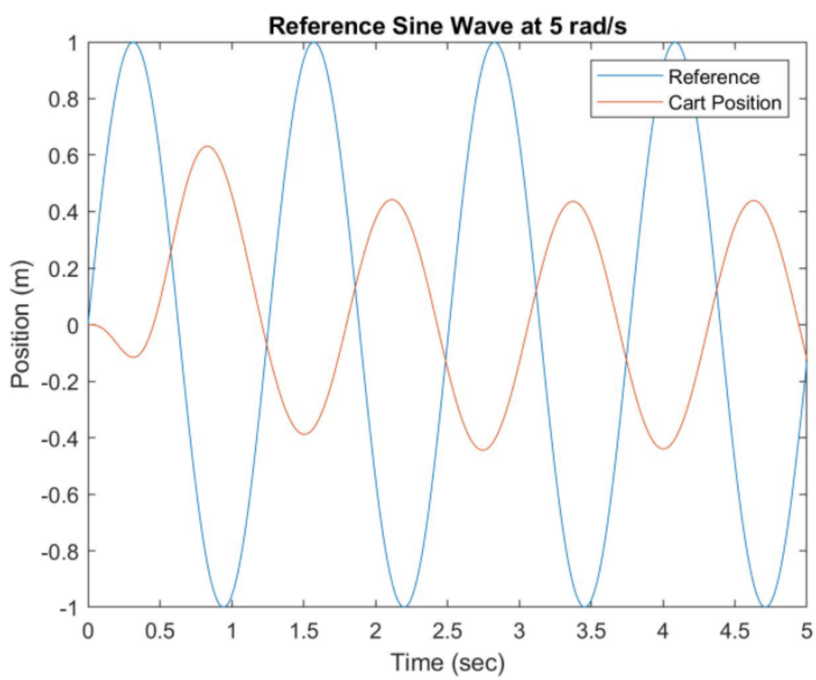

Figure 29: Plot of the Cart Position w/ Reference Sine Wave at 5 rad/s Using Prelab Simulation

The reference response lag the cart position reponses about -180 degrees and the gain is about 0.4.

Conclusion: The gains and phase lag are very close in both simulations now as the angular frequency increases.

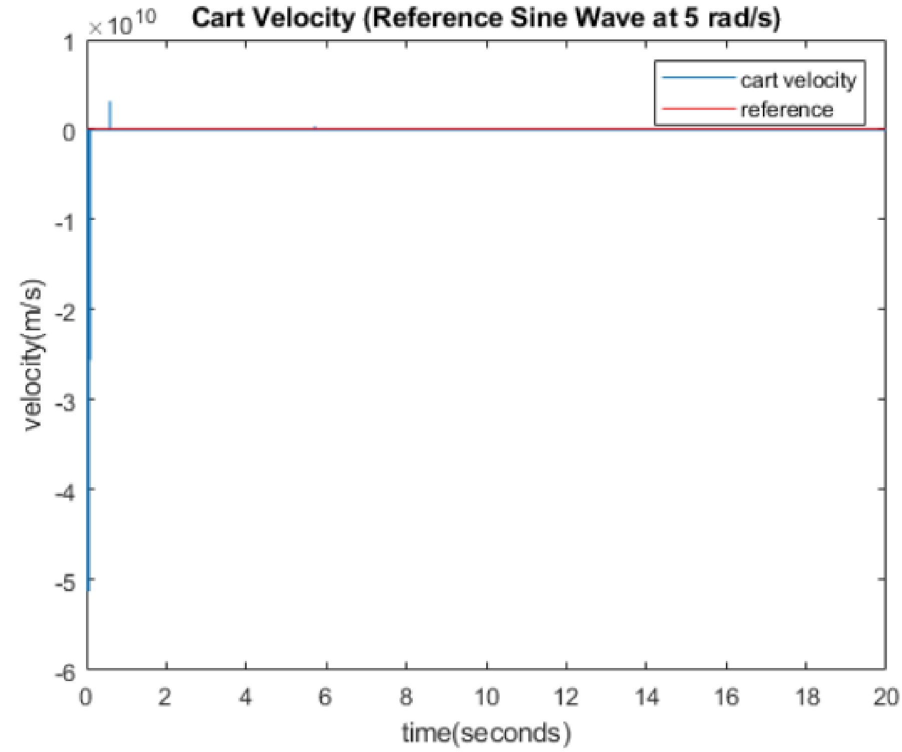

Figure 30: Plot of the Cart Velocity w/ Reference Sine Wave at 5 rad/s

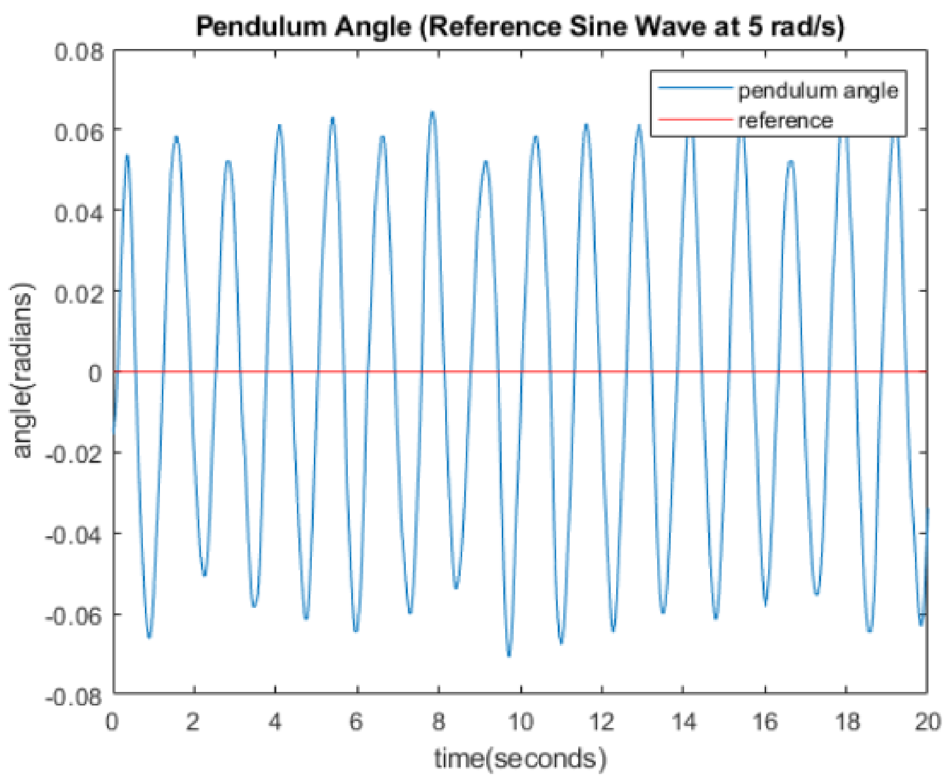

Figure 31: Plot of the Pendulum Angle w/ Reference Sine Wave at 2 rad/s

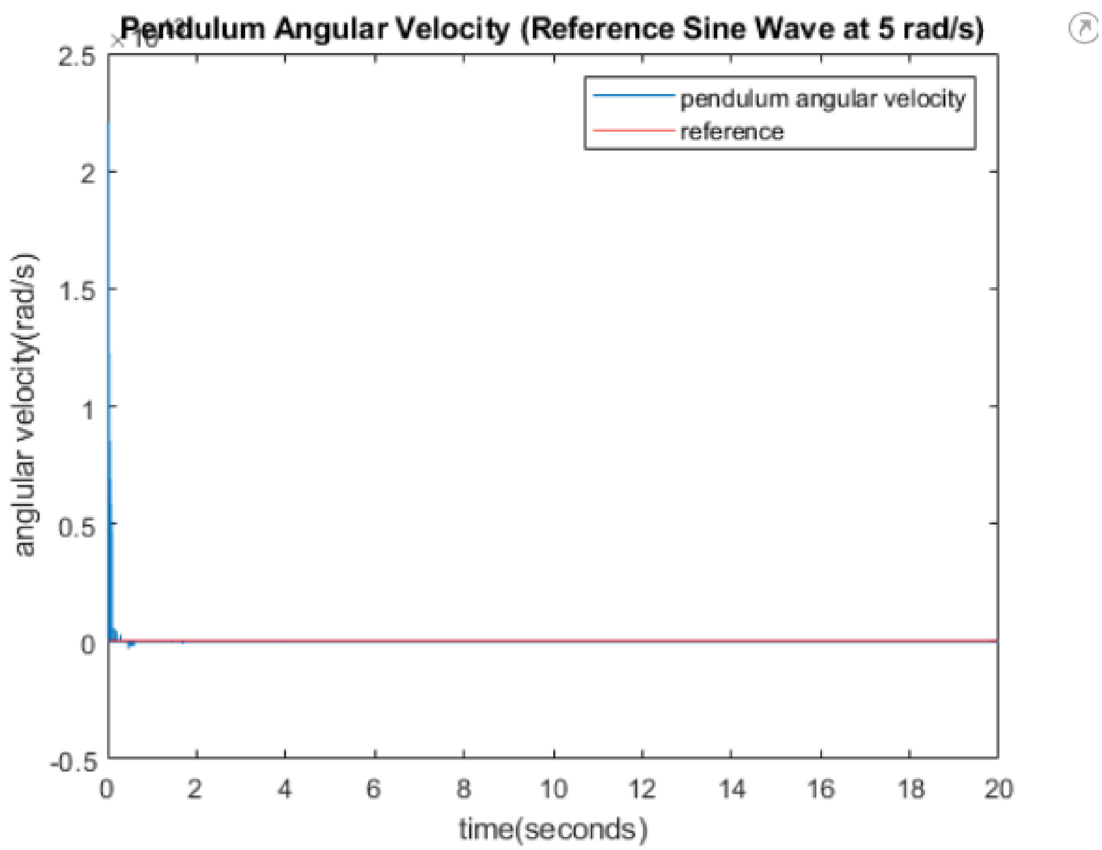

Figure 32: Plot of the Pendulum Angular Velocity w/ Reference Sine Wave at 5 rad/s

b. Slightly change the position of the desired closed-loop poles. Try a couple of different values and run the resulting controllers on the hardware. Again include plots of cart position, cart velocity and pendulum angle in your report. You do not need to repeat part 3a, just pick any of the 3 frequencies.
   Angular Frequency ω: 1 rad/s
   Changing the closed poles slightly:
   1. **-1.8 $\pm$ 10j and -1.4 $\pm$1.3j**
   2. **-2 $\pm$ 9j and -1.6 $\pm$1.1j**
   3. **-1.8 $\pm$ 9j and -1.4 $\pm$1.1j**
   4. **-2.2 $\pm$ 11j and -1.8 $\pm$1.5j**
   5. **-3 $\pm$ 15j and -2 $\pm$1.8j**

```
%1st
%closed loop poles: -1.8 +/- 10j and -1.4 +/- 1.3j
des1 = [-1.8+10i -1.8-10i -1.4+1.3i -1.4-1.3i];
k1 = acker(A,B,des)

%2nd
%closed loop poles: -2 +/- 9j and -1.6 +/- 1.1j
des2 = [-2+9i -2-9i -1.6+1.1i -1.6-1.1i];
k2 = acker(A,B,des2)

%3rd
%closed loop poles: -1.8 +/- 9j and -1.4 +/- 1.1j
des3 = [-1.8+9i -1.8-9i -1.4+1.1i -1.4-1.1i];
k3 = acker(A,B,des3)

%4th
%closed loop poles: -2.2 +/- 11j and -1.8 +/- 1.5j
des4 = [-2.2+11i -2.2-11i -1.8+1.5i -1.8-1.5i];
k4 = acker(A,B,des4)
```

Figure 33: Matlab Code for Computing the K Gains

```
k1 =

  -11.1073  -13.3821  -46.1394   -5.7725


k2 =

   -9.4455  -12.9361  -40.9547   -5.8074


k3 =

   -7.8713  -11.7631  -39.0819   -5.0597


k4 =

  -20.3637  -18.5395  -58.9473   -8.5057
```

Figure 34: Output Result of the K Gain Value

4.3b (Pole Trial 1)

```
clear;clc;close all;
tstop = 20;
out = sim('cartpendulumsinehp1',tstop);
theta = out.theta.data;
x = out.x.data;
xdot = out.xdot.data;
xref = out.xref.data;
time = out.tout;
plot(time,x)
hold on
plot(time,xref)
xlabel('time(seconds)');ylabel('position(meters)');title('Cart Position Pole Trial 1 (Reference Sine Wave at 1 rad/s)')
legend('cart position','reference')
figure
plot(time,xdot)
hold on
yline(0,'r')
xlabel('time(seconds)');ylabel('velocity(m/s)');title('Cart Velocity Pole Trial 1 (Reference Sine Wave at 1 rad/s)')
legend('cart velocity','reference')
figure
plot(time,theta)
hold on
yline(0,'r')
xlabel('time(seconds)');ylabel('angle(radians)');title('Pendulum Angle Pole Trial 1 (Reference Sine Wave at 1 rad/s)')
legend('pendulum angle','reference')
```

Figure 35: Matlab Code for the plots

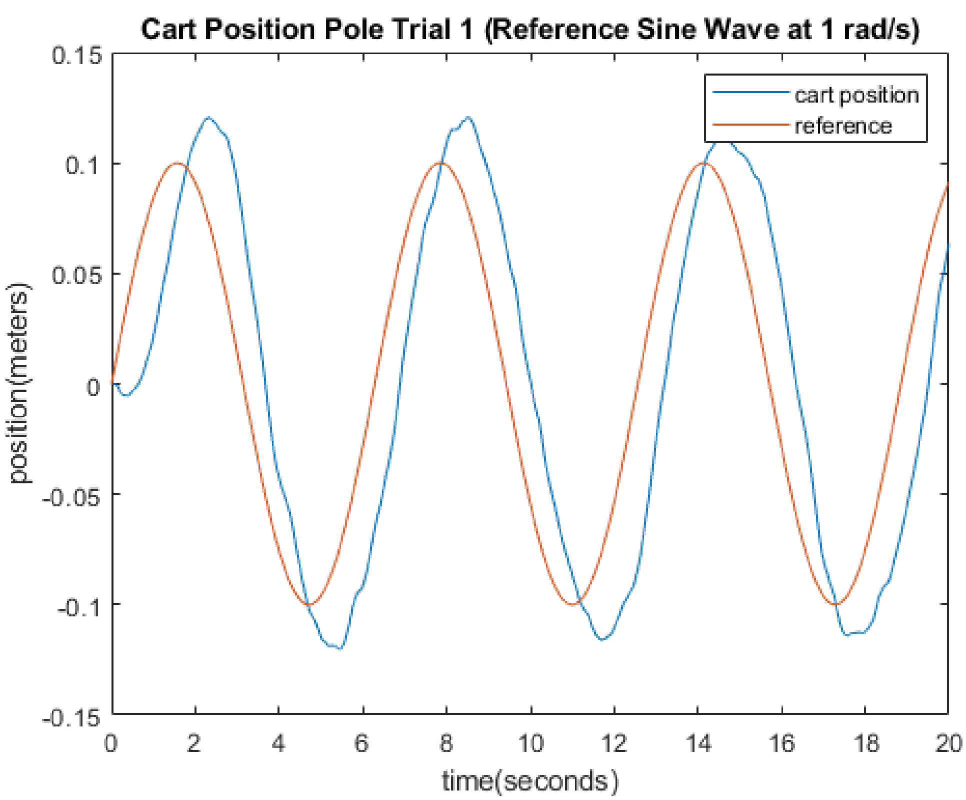

Figure 36:Plot of the Cart Position w/ Reference Sine Wave for Trial 1

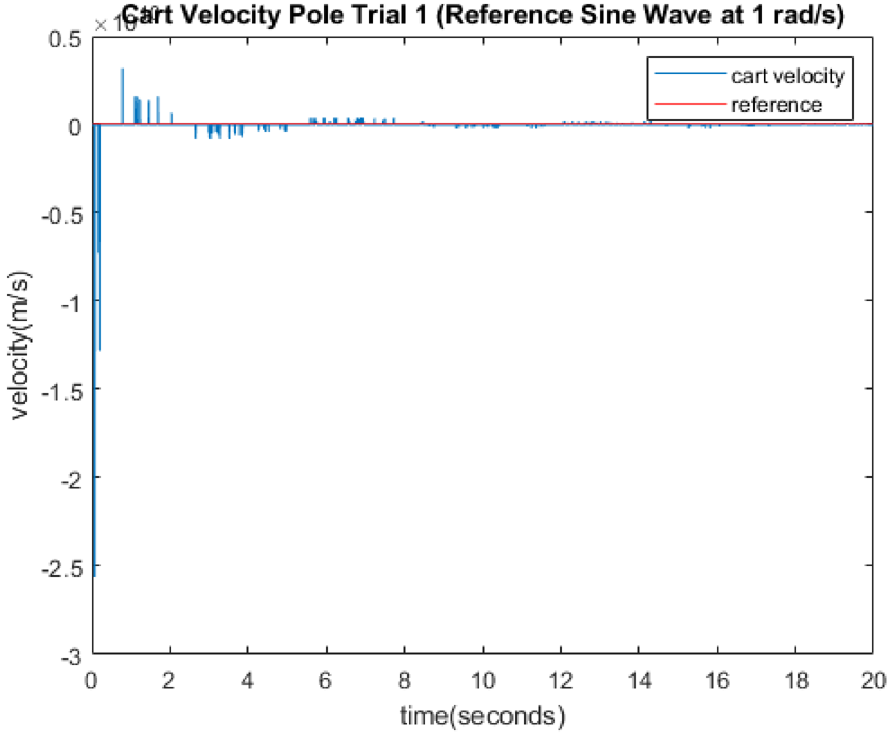

Figure 37: Plot of the Pendulum Angular Velocity w/ Reference Sine Wave for Trial 1

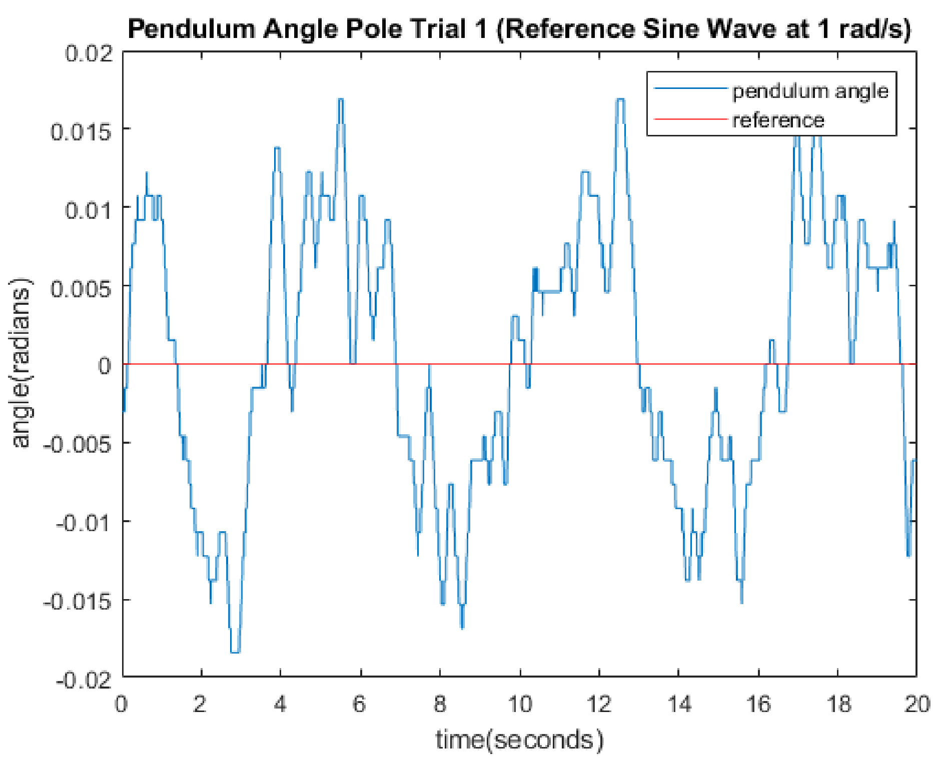

Figure 38: Plot of the Pendulum Angle w/ Reference Sine Wave for Trial 1

4.3b (Pole Trial 5)

```
clear;clc;close all;
tstop = 20;
K = [-49.9368  -33.3440 -106.1082  -15.6020];
out = sim('cartpendulumsinehpt',tstop);
theta = out.theta.data;
x = out.x.data;
xdot = out.xdot.data;
xref = out.xref.data;
time = out.tout;
plot(time,x)
hold on
plot(time,xref)
xlabel('time(seconds)');ylabel('position(meters)');title('Cart Position Pole Trial 5 (Reference Sine Wave at 1 rad/s)')
legend('cart position','reference')
figure
plot(time,xdot)
hold on
yline(0,'r')
xlabel('time(seconds)');ylabel('velocity(m/s)');title('Cart Velocity Pole Trial 5 (Reference Sine Wave at 1 rad/s)')
legend('cart velocity','reference')
figure
plot(time,theta)
hold on
yline(0,'r')
xlabel('time(seconds)');ylabel('angle(radians)');title('Pendulum Angle Pole Trial 5 (Reference Sine Wave at 1 rad/s)')
legend('pendulum angle','reference')
```

Figure 39: Matlab Code for the plots

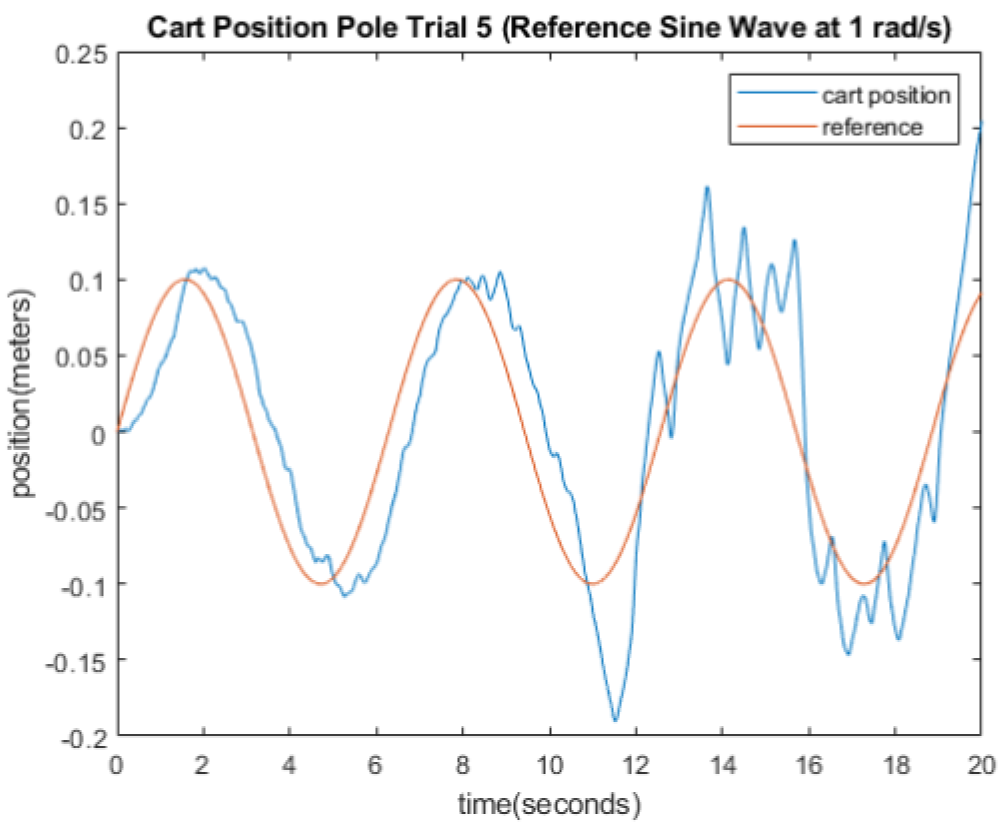

Figure 40:Plot of the Cart Position w/ Reference Sine Wave for Trial 5

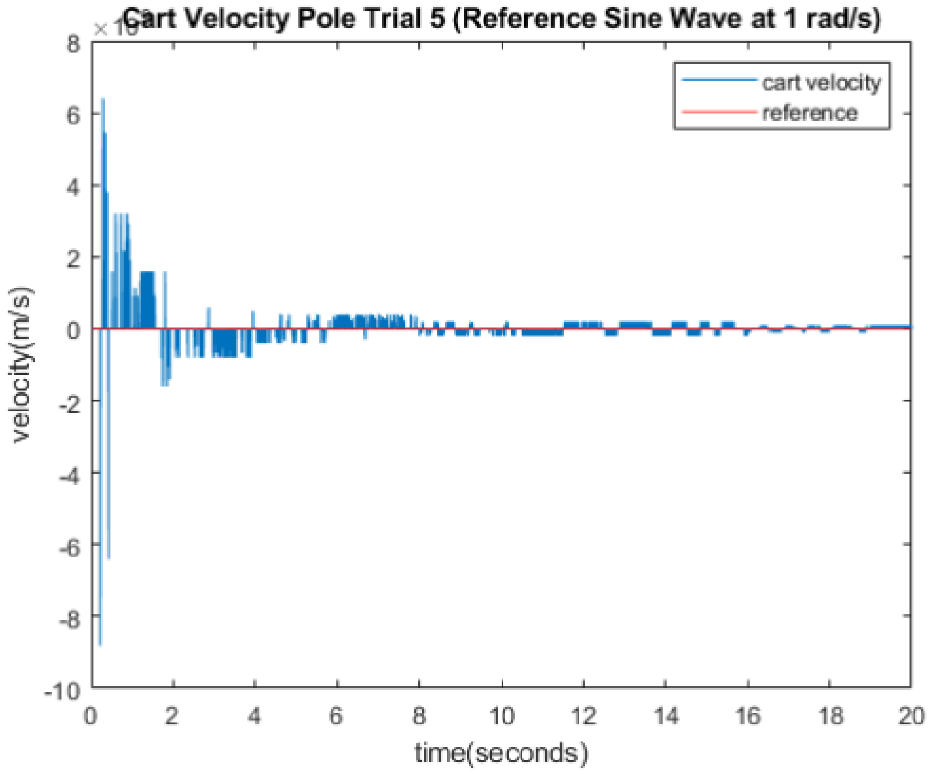

Figure 41: Plot of the Cart Velocity w/ Reference Sine Wave for Trial 5

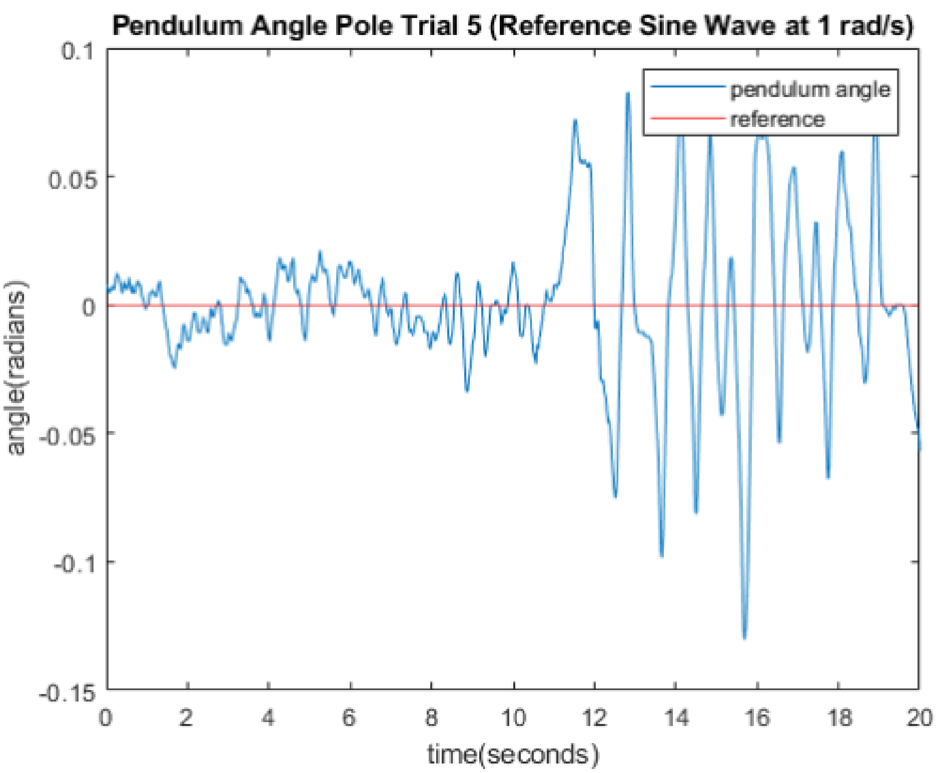

Figure 42: Plot of the Pendulum Angle w/ Reference Sine Wave for Trial 5

4.3b (Pole Trial 3)

```
clear;clc;close all;
tstop = 20;
out = sim('cartpendulumsinehp3',tstop);
theta = out.theta.data;
x = out.x.data;
xdot = out.xdot.data;
xref = out.xref.data;
time = out.tout;
plot(time,x)
hold on
plot(time,xref)
xlabel('time(seconds)');ylabel('position(meters)');title('Cart Position Pole Trial 3 (Reference Sine Wave at 1 rad/s)')
legend('cart position','reference')
figure
plot(time,xdot)
hold on
yline(0,'r')
xlabel('time(seconds)');ylabel('velocity(m/s)');title('Cart Velocity Pole Trial 3 (Reference Sine Wave at 1 rad/s)')
legend('cart velocity','reference')
figure
plot(time,theta)
hold on
yline(0,'r')
xlabel('time(seconds)');ylabel('angle(radians)');title('Pendulum Angle Pole Trial 3 (Reference Sine Wave at 1 rad/s)')
legend('pendulum angle','reference')
```

Figure 43: Matlab Code for the plots

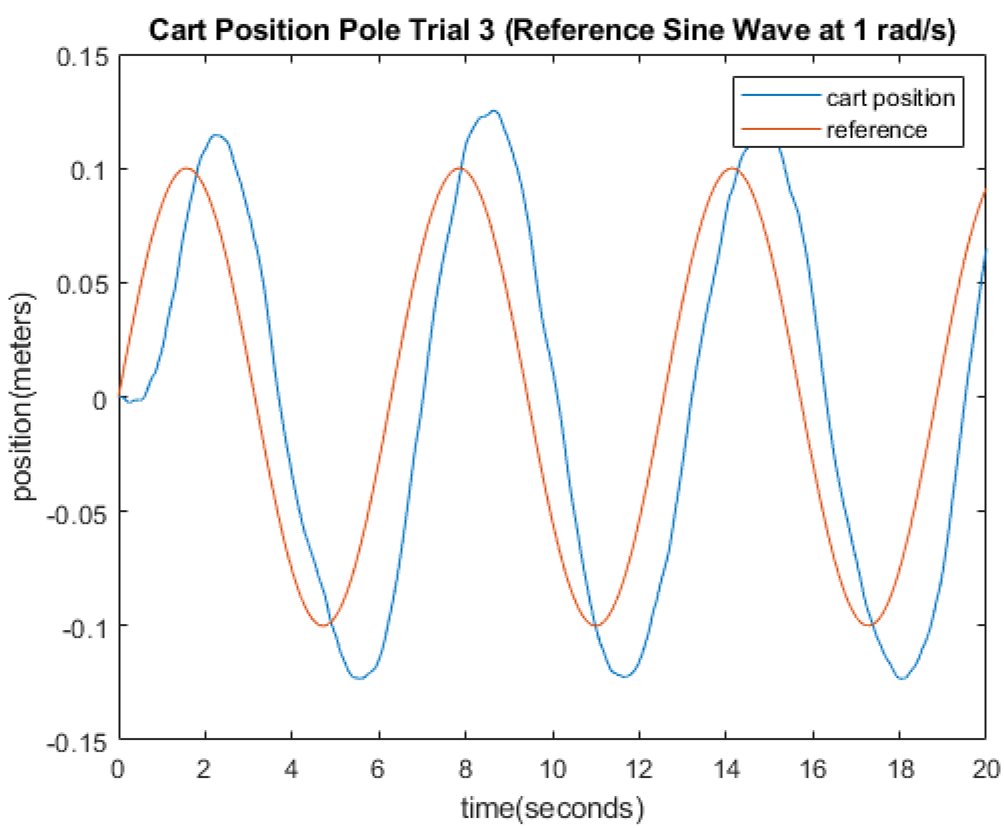

Figure 44:Plot of the Cart Position w/ Reference Sine Wave for Trial 3

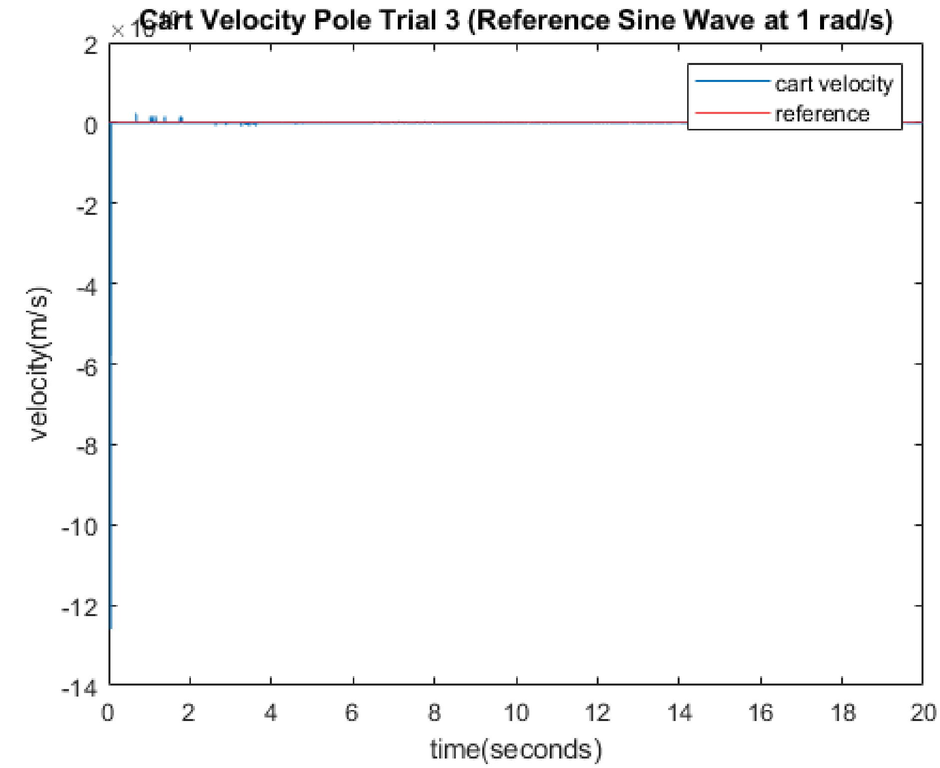

Figure 45: Plot of the Cart Velocity w/ Reference Sine Wave for Trial 3

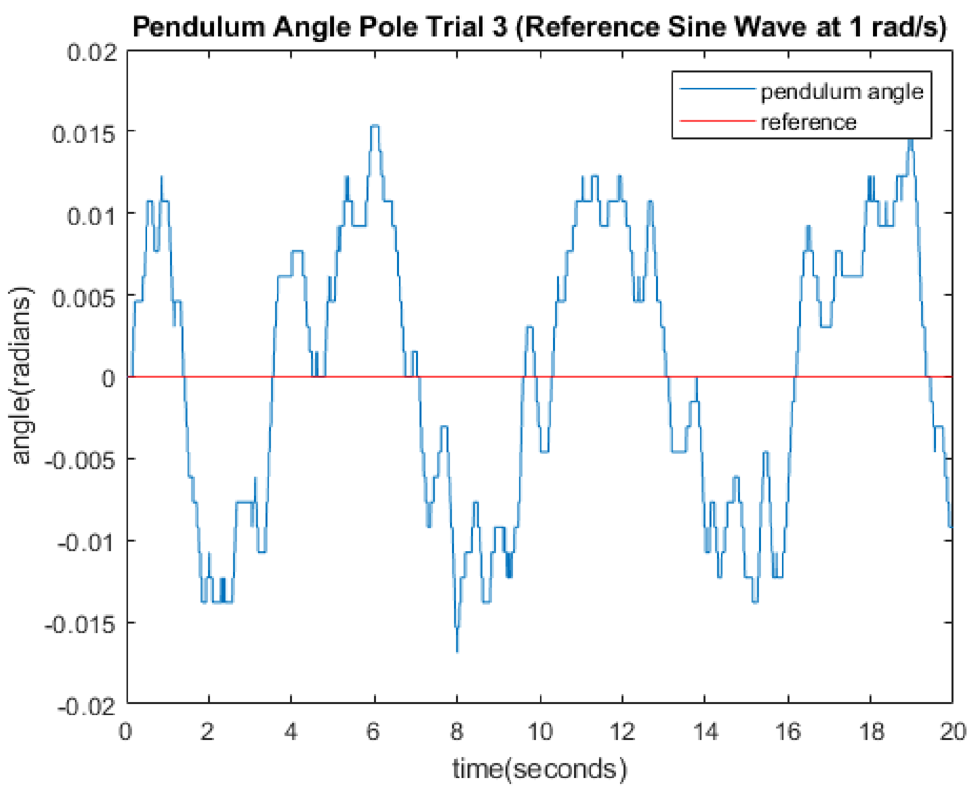

Figure 46: Plot of the Pendulum Angle w/ Reference Sine Wave for Trial 3

4.3b (Pole Trial 4)

```
clear;clc;close all;
tstop = 20;
out = sim('cartpendulumsinehp4',tstop);
theta = out.theta.data;
x = out.x.data;
xdot = out.xdot.data;
xref = out.xref.data;
time = out.tout;
plot(time,x)
hold on
plot(time,xref)
xlabel('time(seconds)');ylabel('position(meters)');title('Cart Position Pole Trial 4 (Reference Sine Wave at 1 rad/s)')
legend('cart position','reference')
figure
plot(time,xdot)
hold on
yline(0,'r')
xlabel('time(seconds)');ylabel('velocity(m/s)');title('Cart Velocity Pole Trial 4 (Reference Sine Wave at 1 rad/s)')
legend('cart velocity','reference')
figure
plot(time,theta)
hold on
yline(0,'r')
xlabel('time(seconds)');ylabel('angle(radians)');title('Pendulum Angle Pole Trial 4 (Reference Sine Wave at 1 rad/s)')
legend('pendulum angle','reference')
```

Figure 47: Matlab Code for the plots

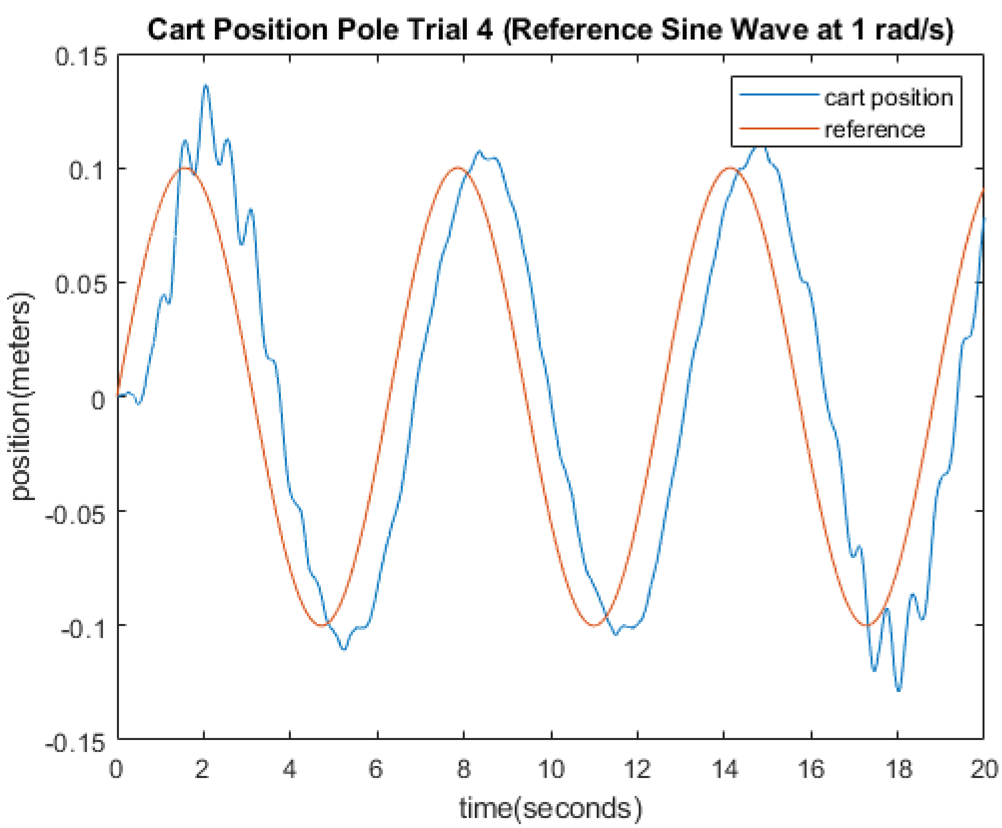

Figure 48:Plot of the Cart Position w/ Reference Sine Wave for Trial 4

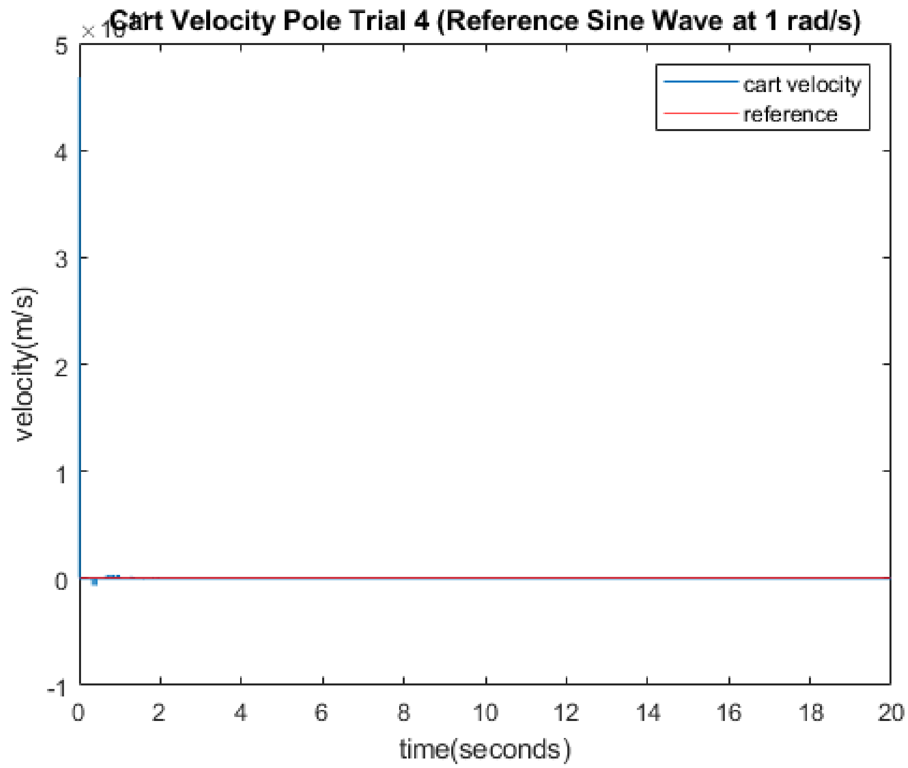

Figure 49: Plot of the Cart Velocity w/ Reference Sine Wave for Trial 4

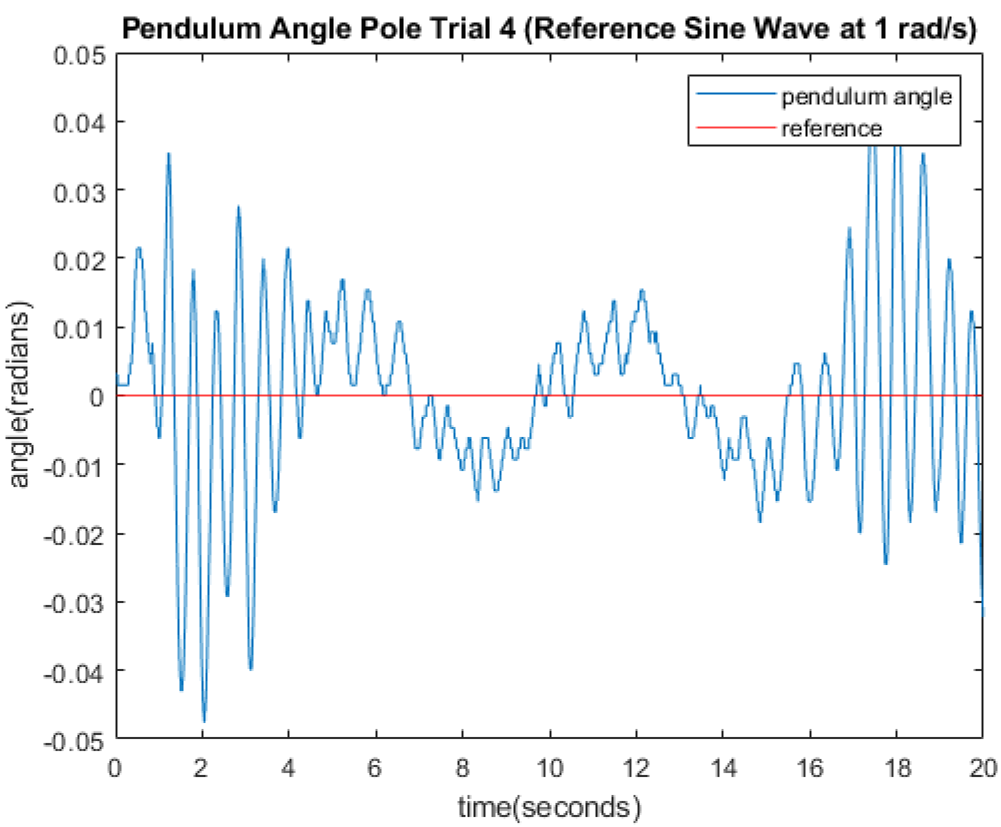

Figure 50: Plot of the Pendulum Angle w/ Reference Sine Wave for Trial 4

Discuss how the changes in the position of the poles affect the behavior of the system.

I changed the position of the closed loop poles and used four different trials of closed loop poles.

**I decreased only the real part of the desired poles slightly in the 1st trial as well as decreasing only the imaginary part of the desired poles slightly in the 2nd trial.
I decreased both the real part and imaginary part of the desired poles slightly in the 3rd trial as well as increasing both of them slightly in the 4th trial.
Finally I increased both the real part and imaginary part of the poles in the 5th trial.**

There is no significant difference of the responses in trail 1 and trail 2 so that only the plots for trail 1 are included in this lab. By comparing the plots in trail 3, 4 and 5, one could observe that the responses become noisier and more oscillating as the positions of the closed loop poles move further away from the origin. The responses in trail 3, which the closed loop poles are the closest to the origin in this case, are the cleanest and the smoothest among the five trials.

Theoretically speaking, the closed loop poles have larger radial distance as well as larger angle as both the real and imaginary parts of the poles are increased which results in a faster response with smaller overshoot and larger natural frequency.

4. Plot the cart velocity $\dot{x}$ and the pendulum's angular velocity $\dot{\theta}$, which are obtained by numerically differentiating the signals x and θ, respectively.

```
figure(1)
plot(out.x_dot.Time, out.x_dot.Data)
hold on
yline(0,'-r','LineWidth',2)
title('Velocity xdot vs Time');
legend('Xdot','reference');
xlabel('Time (s)');ylabel('Velocity (m/s)');


figure(2)
plot(out.theta_dot.Time, out.theta_dot.Data)
hold on
yline(0,'-r','LineWidth',2)
title('Anular Velocity theta dot vs Time');
legend('thetadot','Reference');xlabel('Time (s)');ylabel('Angular Velocity (m/s)');
```

Figure 51: Matlab Code for the plots

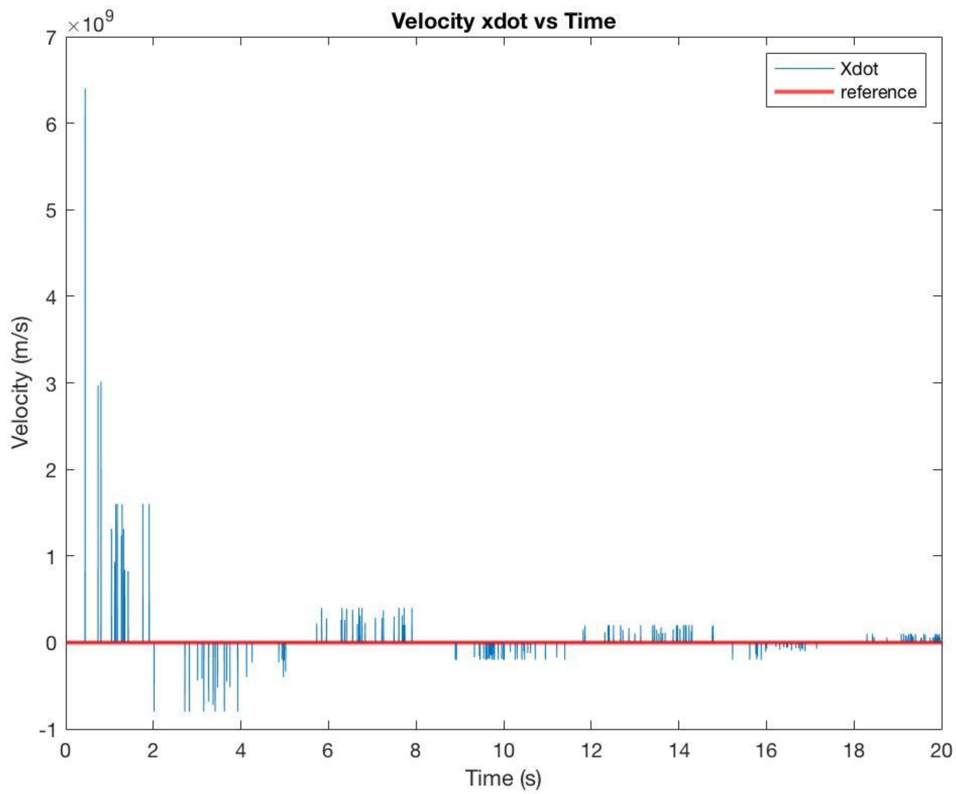

Figure 52: Plot of the Cart Velocity w/ Reference Sine Wave at 1 rad/s

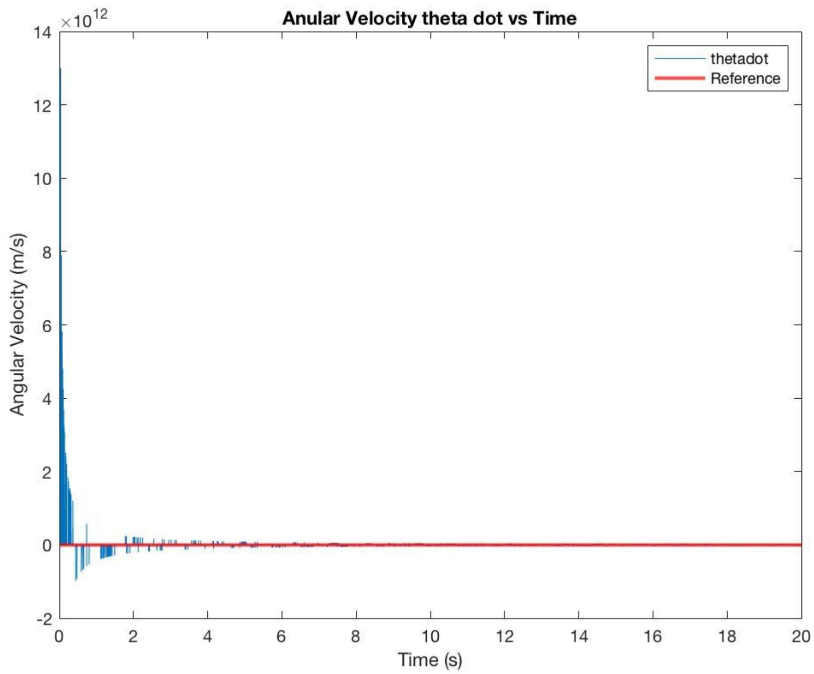

Figure 53: Plot of the Pendulum Angular Velocity w/ Reference Sine Wave at 1 rad/s

Comment on the quality of the obtained signals.

As it is seen from the plots above, the obtained signals are noisy, especially at the beginning of the responses. There are huge step sizes seen in the plots, which come as a result of using the derivative block that is highly sensitive to the dynamics of the entire model. This is seen in the fluctuations of the plot. The derivative block isn't continuous and it approximates the derivative of the input signal with respect to the simulation time. In any physical digital system, there will always be noise due to inherent quantization of numerical representation format and the finite-difference approximations will greatly amplify any noise in the signals.

A System Parameters

Parameter	Value	Description
	439.6 counts/cm	Resolution of the cart position encoder
	651.9 counts/rad	Resolution of the angle encoder
M	$0.94\,\mathrm{kg}$	Mass of cart and motor
m	$0.230\,\mathrm{kg}$	Mass of pendulum
L_p	$0.3302\,\mathrm{m}$	Pendulum distance from pivot to center of mass
I_c	$m\,L_p^2/3$	Moment of inertia of pendulum about its center
I_e	$4m\,L_p^2/3$	Moment of inertia of pendulum about its end
K_t	$7.67 \cdot 10^{-3}\,\mathrm{Nm/A}$	Motor torque constant
K_m	$7.67 \cdot 10^{-3}\,\mathrm{Vs/rad}$	Motor back EMF constant
K_g	3.71	Motor gearbox ratio
R_m	$2.6\,\Omega$	Motor winding resistance
r	$6.36 \cdot 10^{-3}\,\mathrm{m}$	Radius of motor gear
J_m	$3.9 \cdot 10^{-7}\,\mathrm{kg\,m^2}$	Motor moment of inertia

Table 1: Parameters of the inverted pendulum setup