The Control of a Gantry

Author: Brent Justice

Course: AAE 36401

Date: 9/25/2014

TA: Abi Akomandu

Introduction

Objectives

Part (i)

- Determine the equations of motion for a gantry system on a track
- Determine state representation of system
- Determine theoretical natural frequency of the pendulum
- Determine natural frequency of the pendulum experimentally

Part (ii)

- Generate Bode plot of transfer function for voltage input to angle output
- Determine theoretical natural frequency for coupled cart-pendulum system
- Experimentally determine coupled natural frequency

Part (iii)

- Design controller for steadying pendulum as quickly as possible given a disturbance
- Use pole placement method to determine feedback constants
- Use SIMULINK to test and iterate poles
- Test hardware with poles and compare to SIMULINK outputs

Part (iv)

- Design integral control for moving cart 0.5m down the track and steadying the pendulum
- Use pole placement method to determine feedback constants
- Use SIMULINK to test and iterate poles
- Test hardware with poles and compare to SIMULINK outputs

Intended methods

An experimental apparatus containing a cart on a one dimensional track was utilized for reaching the above objectives. More specifically, the cart moved along the track with the use of an electric motor. The motor was controlled through the specification of input voltage. As the cart moved along the track, position was recorded. A pendulum attached to the cart is free to swing. Angle of the pendulum is recorded.

SIMULINK was utilized for controlling the cart and producing simulated cart-pendulum response.

Procedure

Definition of Variables

variable	units	Description
R_m	Ω	Motor armature resistance
L_m	mH	Motor armature inductance
K_t	$N-m/A$	Motor torque efficiency

η_m	%	Motor efficiency
K_m	V-s/rad	Back electromotive force
J_m	$kg-m^2$	Rotor moment of inertia
k_g		Planetary gearbox ratio
η_g	%	Planetary gearbox efficiency
M_c	kg	Cart mass
M_w	kg	Cart weight mass
L_t	m	Track length
T_c	m	Cart travel
P_r	m/tooth	Rack pitch
r_{mp}	m	Motor pinion radius
N_{mp}		Motor pinion number of teeth
r_{pp}	m	Position pinion radius
N_{pp}		Position pinion number of teeth
K_{EP}	m/count	Cart encoder resolution
B_{eq}	Kg/s	Viscous damping
B_{emf}	Kg/s	Back emf damping
ω_p	Rad/s	Natural frequency of pendulum only
ω_n	Rad/s	Natural frequency of cart-pendulum system
ζ		Damping ratio
$\lvert G_\alpha(i\omega)\rvert$	Rad/V	Transfer function magnitude
τ_s	s	Settling time
α	rads	Angle of the pendulum
K values		State feedback constants
x		State variables
A		State space matrix – EOM
B		State space vector - voltage
v		Voltage input
C		State space vector – choose output of interest
u		State space input – integral controller
D		State space vector

Schematic and Description of Apparatus

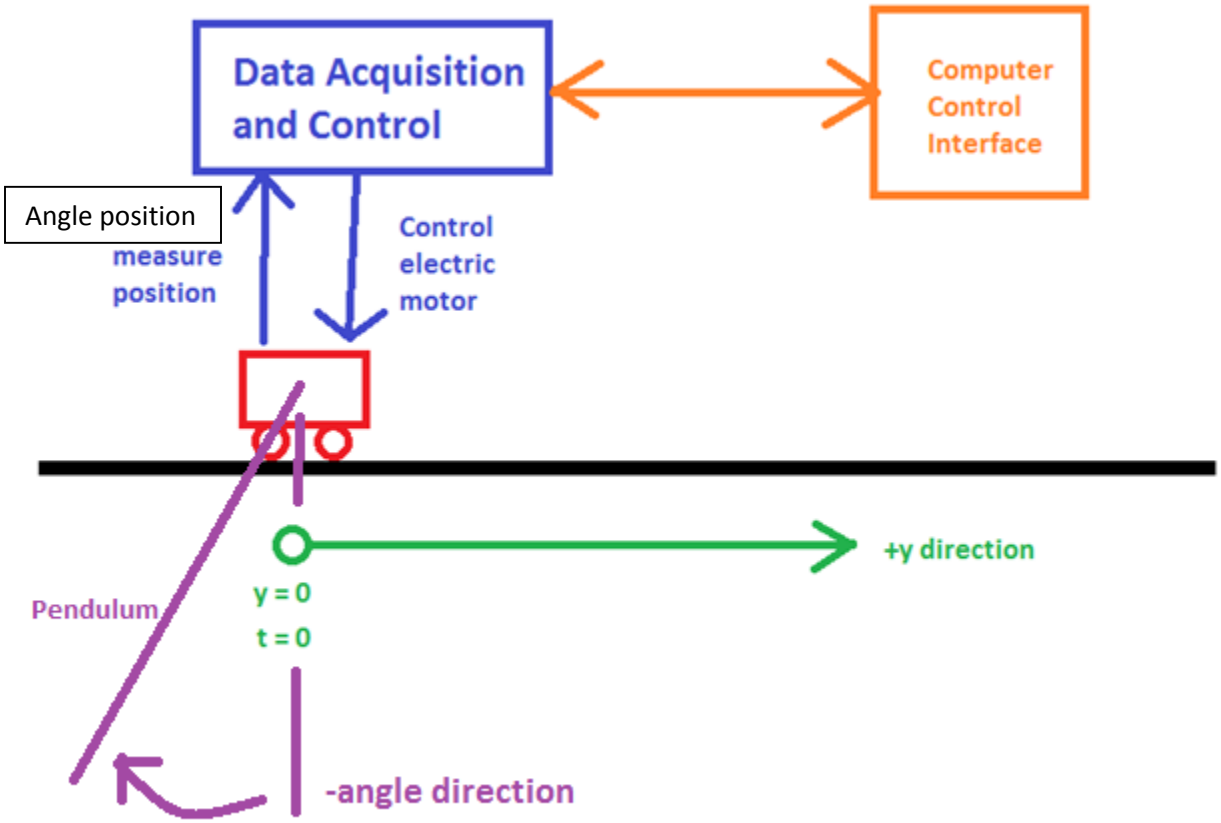

Figure (0): Experimental Apparatus

From Figure 0, we see that the experimental apparatus consists of a car on a one-dimensional track. At time = 0, the initial position is defined as y=0. We also see that as the pendulum hangs downwards, the angle (α) is 0. Angle increases as the pendulum moves counter-clockwise in direction.

The cart motion is controlled by an electric motor. The cart position and pendulum position are both observed and recorded. SIMULINK is used for controlling the cart and for acquiring data.

One limitation of the experiment is the length of the track – care must be taken to ensure that the cart does not crash into the end of the track. Another limitation is that angle is only recorded with a precision of approximately 0.1 degrees.

Procedure of Experiments

Set SIMULINK data buffer to 30 seconds

Part (i)

- Start data recording
- Move pendulum to about 15 degrees and let it swing
- Record angle over time

Part (ii)

- Set X_MAX to +/- 0.6 m to keep cart from running off track
- Apply sinusoidal voltage input to cart motor with amplitude of 3 volts. Conduct 3 tests where frequency = 3 rad/s, frequency = natural frequency of gantry, and frequency = 7 rad/s.
- Record angle over time

Part (iii)

- Select and enter values for K (Feedback constants)
- Tap the pendulum
- Record pendulum angle

Part (iv)

- Select and enter values for K (feedback constants)
- Set cart final position to 0.5 m
- Record cart position and pendulum angle
- Iterate testing for various K values

Results

Table 1: natural frequency of pendulum through linear approximation and experimental data

Part (i): experimental and linear approximation		
variable	value	units
$\omega_p\ (linear)$	4.7543	rad/s
$\omega_p\ (experimental)$	4.7866	rad/s

Table 2: natural frequency of cart-pendulum system determined analytically

Part (ii): computed from eigenvalues		
variable	value	units
$\omega_n\ (calculated)$	4.8065	rad/s
$\zeta\ (calculated)$	0.0352	

Table 3: natural frequency of cart-pendulum system determined analytically through BODE plot

Part (ii): computed frequency from Bode Plot		
variable	value	units
$\omega_n\ (Bode)$	4.8	rad/s

Table 4: cart-pendulum BODE plot inspection at various frequencies vs experimental results

Part (ii): Transfer function magnitude determined experimentally and by BODE analysis			
variable	From Bode plot	From experimental results	units
$\lvert G_\alpha(i\omega)\rvert\ where\ \omega = 3\ rad/s$	0.064	0.074	Rad/V
$\lvert G_\alpha(i\omega)\rvert\ where\ \omega = \omega_n$	0.842	0.363	Rad/V
$\lvert G_\alpha(i\omega)\rvert\ where\ \omega = 7\ rad/s$	0.071	0.078	Rad/V
$\lvert G_\alpha(i\omega)\rvert\ where\ \omega = 3\ rad/s$	-23.95	-22.16	DB Rad/V
$\lvert G_\alpha(i\omega)\rvert\ where\ \omega = \omega_n$	-1.49	-8.80	DB Rad/V
$\lvert G_\alpha(i\omega)\rvert\ where\ \omega = 7\ rad/s$	-22.98	-22.16	DB Rad/V

Table 5: SIMULINK inspection of cart-pendulum system with proportional control

Part (iii): SIMULINK inputs and output for $\dot{\alpha}(0) = \frac{\pi}{2}$		
variable	value	units
$\dot{\alpha}(0)$ $[input]$	1.571	rad/s
$Poles$	-1.82 + 1.91i -1.82 - 1.91i -10 -20	
K Values	40.13 -63.95 19.30 2.16	
τ_s (α)	2.4819	s

Table 6: experimental and SIMULINK inspection of cart-pendulum system with proportional control

Part (iii): experimental data vs SIMULINK outputs		
variable	value	units
$\dot{\alpha}(0)$ $[measured]$	1.534	rad/s
$Poles$	-1.82 + 1.91i -1.82 - 1.91i -10 -20	
K Values	40.13 -63.95 19.30 2.16	
τ_s $(\alpha - SIMULINK)$	2.462	s
τ_s $(\alpha - experimental)$	2.346	s

Note that the poles and K values were not changed from Table 5 to Table 6. It was determined that the SIMULINK created poles generated an acceptable response for the cart-pendulum system. It was not necessary to use different poles.

Table 7: SIMULINK inspection of cart-pendulum system with integral control

Part (iv): SIMULINK inputs and output		
variable	value	units
$Poles$	-1.82 + 1.91i -1.82 - 1.91i -10 -20 -30	
K Values	847.6 7.2 237.5 88.4 -1204.0	
$\tau_s\ (\alpha)$	2.274	s

Table 8: experimental and SIMULINK inspection of cart-pendulum system with integral control

Part (iv): experimental data vs SIMULINK outputs		
variable	value	units
$Poles$	-2+2i -2-2i -7+3i -7-3i -10	
K Values	112.4 -28.2 31.3 9.0 -134.1	
$\tau_s\ (\alpha: SIMULINK)$	2.19	s
$\tau_s\ (\alpha: experimental)$	2.10	s

Analysis and Discussion

The nonlinear equations of motion for the pendulum-cart system are given below:

$$\begin{aligned}
&\left((M_c+M_p)I_p + M_cM_pl_p^2 + M_p^2l_p^2\sin(\alpha)^2\right)\ddot{x}_c + \left(I_p + M_pl_p^2\right)B_{eq}\dot{x}_c \\
&= \quad M_pl_pB_p\cos(\alpha)\dot{\alpha} + \left(M_p^2l_p^3 + I_pM_pl_p\right)\sin(\alpha)\dot{\alpha}^2 + \left(I_p + M_pl_p^2\right)F_c + M_p^2l_p^2g\cos(\alpha)\sin(\alpha);
\end{aligned}$$

$$\begin{aligned}
&\left((M_c+M_p)I_p + M_cM_pl_p^2 + M_p^2l_p^2\sin(\alpha)^2\right)\ddot{\alpha} + \left(M_c+M_p\right)B_p\dot{\alpha} \\
&= \quad M_pl_p\cos(\alpha)B_{eq}\dot{x}_c - \left(M_c+M_p\right)M_pgl_p\sin(\alpha) - M_p^2l_p^2\sin(\alpha)\cos(\alpha)\dot{\alpha}^2 - M_pl_p\cos(\alpha)F_c;
\end{aligned}$$

$$F_c = \frac{\eta_g K_g \eta_m K_t \left(vr_{mp} - K_gK_m\dot{x}_c\right)}{R_m r_{mp}^2}. \qquad (1)$$

Through use of the small angle approximation, these equations can be reduced to the following linearized equations of motion:

$$\begin{aligned}
\ddot{x}_c &= \frac{-(I_p + M_pl_p^2)B_{eq}\dot{x}_c + M_pl_pB_p\dot{\alpha} + M_p^2l_p^2g\alpha + (I_p + M_pl_p^2)F_c}{(M_c+M_p)I_p + M_cM_pl_p^2} \\
\ddot{\alpha} &= \frac{M_pl_pB_{eq}\dot{x}_c - (M_c+M_p)B_p\dot{\alpha} - (M_c+M_p)M_pgl_p\alpha - M_pl_pF_c}{(M_c+M_p)I_p + M_cM_pl_p^2} \\
F_c &= \frac{\eta_g K_g \eta_m K_t \left(vr_{mp} - K_gK_m\dot{x}_c\right)}{R_m r_{mp}^2}.
\end{aligned} \qquad (2)$$

By setting all of the derivatives to zero, we are able to obtain the following equilibrium states:

$$\alpha^e = 0 \qquad (3)$$
$$x_c^e = any\ value \qquad (4)$$

We are interested in the equilibrium point of α. This tells us that in order to arrest the motion of the pendulum, we must drive α to zero.

To analyze these equations of motion, we set the following state variables:

$$x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} x_c \\ \alpha \\ \dot{x_c} \\ \dot{\alpha} \end{bmatrix} \qquad (5)$$

It follows that the derivatives of these state variables can be computed with the following model:

$\dot{x} = Ax + Bv$	(6)
$where\ v = voltage\ input$ $where\ \dot{x} = state\ space\ output$	

From the linearize equations of motion and from known variables, we can compute the following:

$$A = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1.5216 & -11.6513 & 0.0049 \\ 0 & -26.1093 & 26.8458 & -0.0841 \end{bmatrix} \text{ and } B = \begin{bmatrix} 0 \\ 0 \\ 1.5304 \\ -3.5261 \end{bmatrix} \quad (7)$$

From equation 6, we see that voltage is the only input. There are 4 outputs corresponding to the 4 state space variables. If we wanted to know the transfer function corresponding to voltage as an input and α as an output, we would create the following value for C:

$C = [0\ 1\ 0\ 0]$	(8)

Note that this vector corresponds to the placement of α in the state space vector. It follows that the transfer function from voltage input to α output can be computed as such:

$G_\alpha(s) = \frac{\alpha(s)}{V(s)} = C(sI - A)^{-1}B = \frac{-3.526s}{s^3 + 11.74s^2 + 26.96s + 263.4}$	(9)

Part (i)

In the first part of the lab, we assume that the cart is fixed to the origin and does not move. However, the pendulum is free to swing. This assumption simplifies the equations of motion in Equation 2 to the following linearized equation of motion:

$(I_p + M_p l_p^2)\ddot{\alpha} + M_p l_p g \alpha = 0$	(10)

This reduces further to a simple polynomial where:

$$\ddot{\alpha} + \omega_p^2 \alpha = 0 \quad \text{where} \quad \omega_p = \sqrt{\frac{M_p l_p g}{I_p + M_p l_p^2}}. \quad (11)$$

The constant denoted as ω_p is the natural frequency of the pendulum. Equation 11 and known parameters were used to calculate a theoretical natural pendulum frequency in Table 1.

Natural frequency was also determined experimentally. The cart was held down and the pendulum was disturbed and allowed to swing freely. Figure 1, below, illustrates the collected steady-state data:

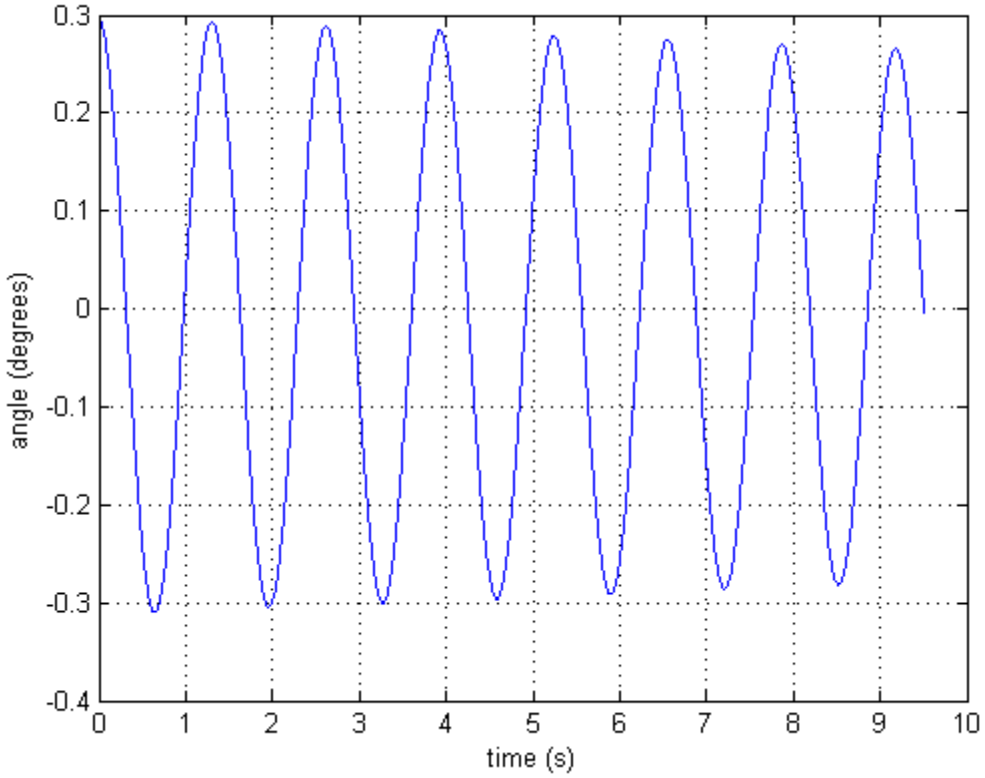

Figure 1: Cart was held down, pendulum was disturbed and allowed to swing freely

We observe that the pendulum swings with at a constant frequency. This frequency was determined and corresponds to the experimentally determined pendulum natural frequency – shown in Table 1.

Part (ii)

Every square matrix can be represented as a characteristic polynomial. The characteristic polynomial of the matrix "A" in Equation 7 is seen below:

$charpoly(A) = s^4 + 11.74s^3 + 26.96s^2 + 263.37s$	(12)

When we take the roots of this polynomial, we find the following:

$roots\big(charpoly(A)\big) = \begin{cases} 0 \\ -11.40 \\ -0.17 + 4.80i \\ -0.17 - 4.80i \end{cases} = eigenvalues(A)$	(13)

We note that the eigenvalues coincide with the roots of the characteristic polynomial. The complex eigenvalues contain the information pertaining to the natural frequency of the cart-pendulum system. More specifically, we observe that:

$(s + 0.17 + 4.80i)(s + 0.17 - 4.80i) = s^2 + 2\zeta\omega_n s + \omega_n^2$	(14)

From equation 14, we are able to calculate the natural frequency and the damping ration found in Table 2.

Recall from equation 9 where we discussed how to compute the transfer function relating voltage input to angle output. Taking the Bode plot of this transfer function, we get:

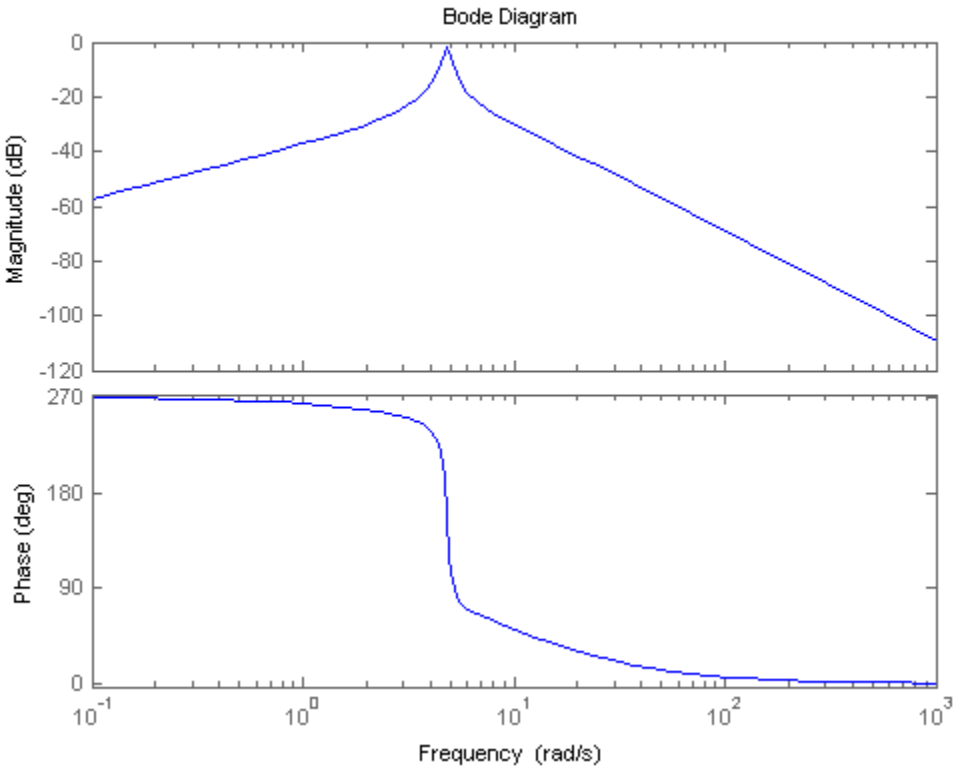

Figure 2: Bode plot of voltage input to angle output transfer function

From the Bode plot, we observe that the natural frequency occurs at 4.8 radians/sec. (Shown in Table 3.) This corresponds to the turning point in the magnitude curve. Recall that the natural frequency exists at a negative complex pole. Negative complex poles will cause a change in the magnitude slope by -40 dB/sec. As seen from the Bode plot, the incoming magnitude slope is 20 dB/sec and the outgoing magnitude slope is -20 dB/sec at the natural frequency point.

Next, we applied various input voltage frequencies to the cart and observed the resulting angle output over time. The following results were obtained:

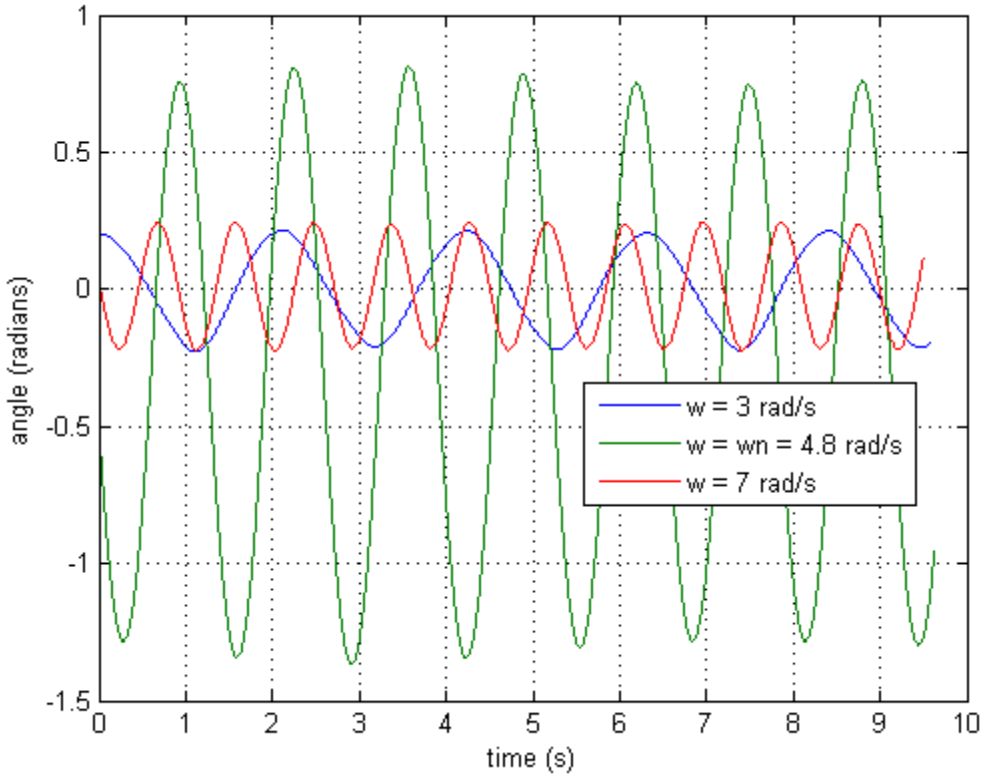

Figure 3: steady state angle output of pendulum as a function of input voltage frequency

During the transient period, not seen here, the pendulum motion was erratic and difficult to predict. After approximately 10 seconds, steady state was achieved, as seen above. In this mode, frequency output and magnitude remained constant. At $\omega = \omega_n$, the pendulum motion was very erratic. The amplitude of the pendulum swing was very large. Eventually, the pendulum settled into a strange motion where the positive peak was less than the absolute value of the negative peak.

The magnitude of the transfer function at various frequencies can be directly observed from the Bode magnitude plot. The magnitude of the transfer function can also be extracted from the real-world data using the steady-state of the linear system:

$\alpha_{ss}(t) = A\lvert G_\alpha(i\omega)\rvert sin(\omega t + \angle G_\alpha(i\omega))$	(15)

By assuming that the trig term is equal to zero at the peaks, we are able to simplify this to:

$\lvert G_\alpha(i\omega)\rvert = \frac{peaks(\alpha_{ss}(t))}{A}$	(16)

These magnitudes were found and tabulated in Table 4. Note that the magnitudes are within 10% error for the case of w=3 and 2=7 rad/s. However, when w=wn, the results are off. This is most likely attributed to the strange, nonsinusoidal oscillation of the real-world data. It is possible that in this case, the system had not yet reached steady state.

Part (iii)

The objective of the controller for part 3 is to steady the oscillations of the pendulum as quickly as possible given any perturbation of the pendulum. The designed controller uses proportional feedback constants in conjunction with the linearized system of equations. Recall that when the nonlinear system of equations were simplified, the small angle approximation was assumed. As such, this controller assumes that the pendulum will remain at small angles. Otherwise, the controller may begin to fail.

The feedback constants are chosen such that the system is stable – this corresponds to a system where all of the poles lie in the left-hand plane. More specifically, we desire that:

$\zeta = 0.6901 \; and \; \omega_n = 2.6346 \, rad/s$	(17)

From equation 14, we note that these criteria yield 2 poles. As such, we must find 2 more poles since the system contains 4 state variables. These 2 remaining poles are arbitrarily chosen to lie on the negative axis. MATLAB is utilized to find the feedback constants that generate these 4 specified poles.

SIMULNIK was used with these input feedback constants to yield the following simulated response:

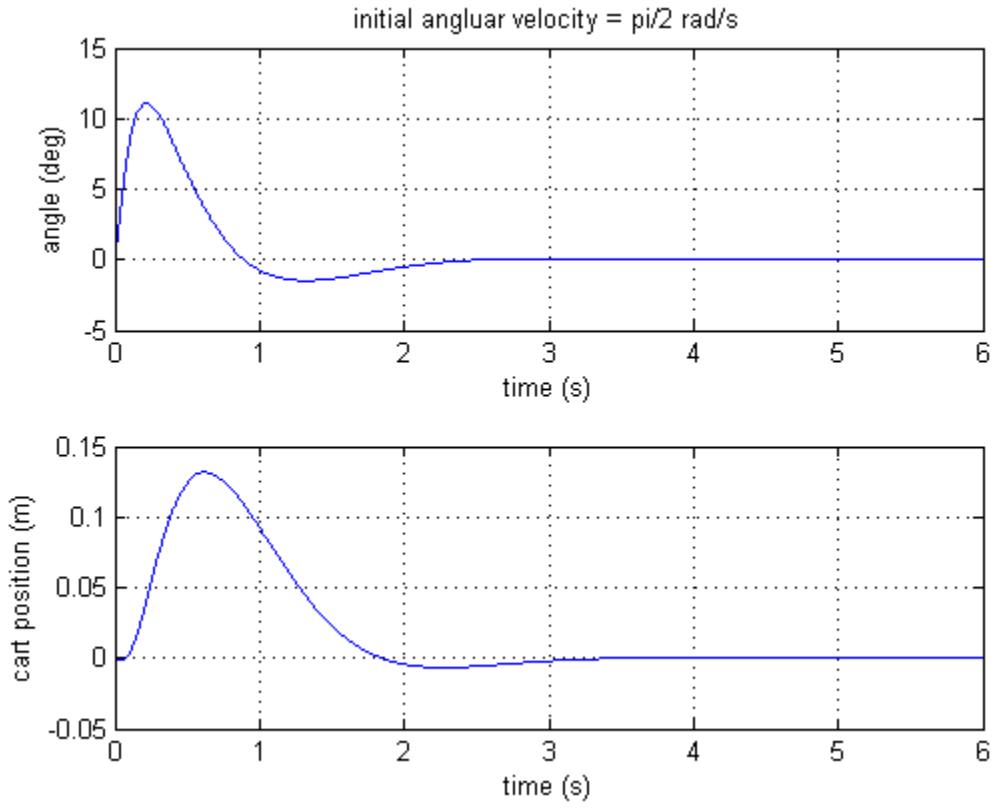

Figure 4: SIMULINK response with proportional feedback constants

Next, these feedback constants were used to control the gantry. The results are seen below:

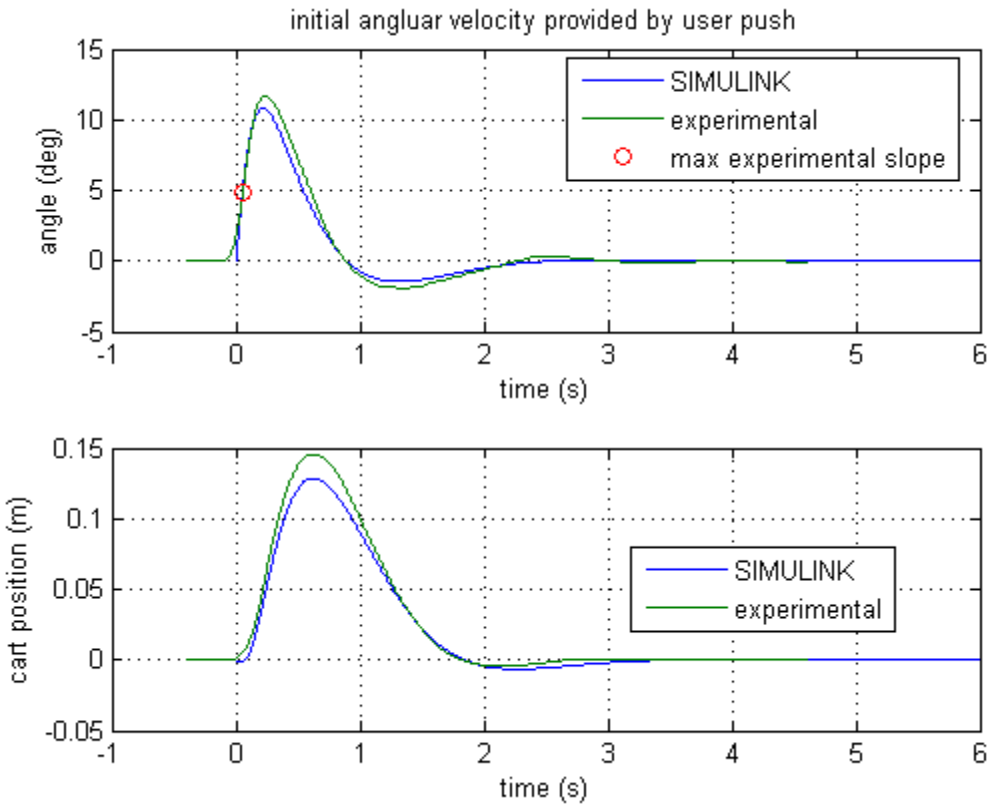

Figure 5: SIMULINK and gantry response with proportional feeback constants

For Figure 4, the cart was initially disturbed with an initial angular velocity of the pendulum. For Figure 5, this initial angular velocity was simulated with a tap on the pendulum. The resulting data was analyzed to estimate an initial angular velocity. This angular velocity was then input into the SIMULINK simulation, paired with the feedback constants to generate the SIMULINK response.

From Figure 5, we note that the SIMULINK response and the real-world response are closely matched. Table 6 tabulates the inputs and outputs. We see that the settling time values are within 5% error of each other. This is evidence that the controller closely matches design specifications.

Part (iv)

The objectives for part 4 are to move the cart 0.5 meters down the track and to steady the oscillations of the pendulum as quickly as possible. We desire a controller capable of meeting these objectives with a settling time of less than 2.2 seconds for the angle. To accomplish this, an integral controller was utilized.

The integral controller introduces a new state variable:

$x_5 = \int_0^t \big(u(\sigma) - x_1(\sigma)\big)d\sigma \; or \; equivelantly \; \dot{x_5} = u - x_1$	(18)

This new state variable results in a new state space system:

$$\dot{x} = A_i x + B_i v + Du \quad (19)$$

where:

$$A_i = \begin{bmatrix} & A & & \begin{bmatrix}0\\0\\0\\0\end{bmatrix} \\ \begin{bmatrix}-1 & 0 & 0 & 0\end{bmatrix} & & 0 \end{bmatrix} \quad \text{and} \quad B_i = \begin{bmatrix} B \\ 0 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \\ 0 \end{bmatrix}$$

$$x = \begin{bmatrix} x_c \\ \alpha \\ \dot{x}_c \\ \dot{\alpha} \\ \int (u - x_c) \end{bmatrix} \quad \text{and} \quad D = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}.$$

We see from Equation 19 that the term "u" is a new input. It follow from Equation 18 that the steady solution of x1 is:

$$x_1^e = steady\ state = u_0 \quad (20)$$

From equation 20, we see the usefulness of the new state variable – the steady state x1 can be chosen by selecting an appropriate input value for "u". "u" is called the reference signal.

Similarly to part 3, the equations of motion utilized for the part 4 model are linearized. As such, it is assumed that the pendulum will remain at small angles, or the model might become unreliable.

The feedback constants are chosen such that the settling time of the angle of the pendulum is less than 2.2 seconds and that the max overshoot is less than 5%. Recall that the objectives of this test are to have the gantry move 0.5 meters down the track and to arrest the oscillations of the pendulum as quickly as possible. Similarly to the method described in part 3, 5 poles are arbitrarily chosen such that they lie on the left-hand plane. This corresponds to a stable system. MATLAB is utilized to find the feedback constants that correspond to these poles.

SIMULINK was used to iterate on these pole guesses until a feasible solution was found. The results are seen below:

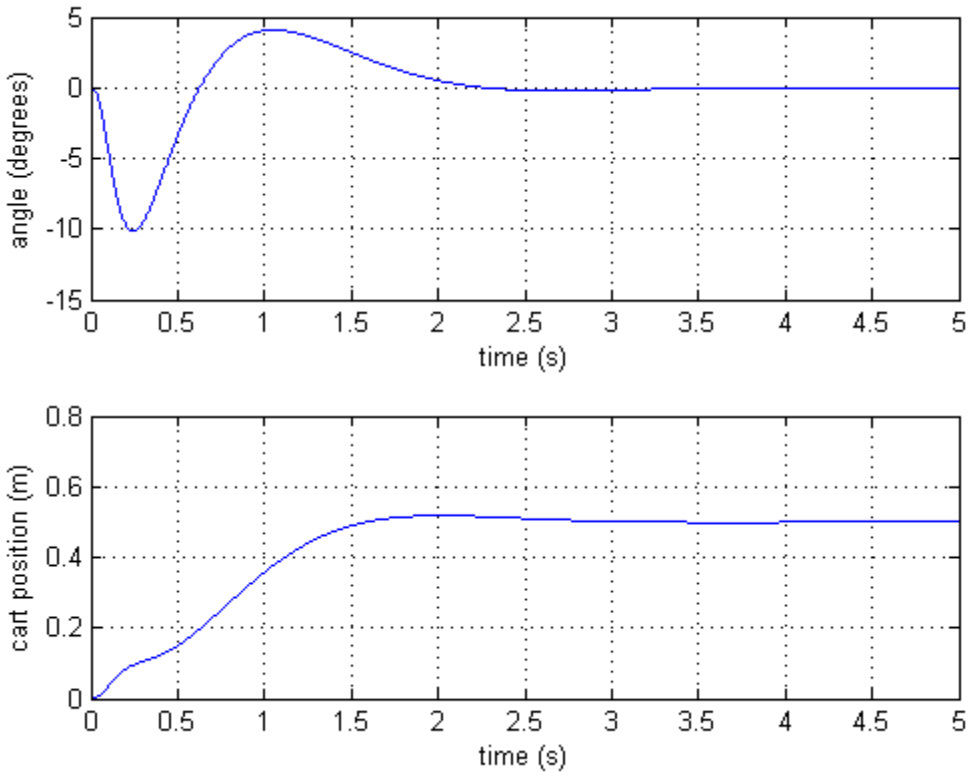

Figure 5: SIMULINK model response of gantry with integral controller

The inputs for the SIMULINK response in Figure 5 can be found in Table 7. However, when these input feedback constants were used on the gantry controller, the resulting gantry motion was very erratic. This can be attributed to the very large K-values. The hardware was not able to respond quickly enough to accommodate the large gains.

As such, different feedback constant values were used for the gantry during the experiment. The results are seen in Figure 6.

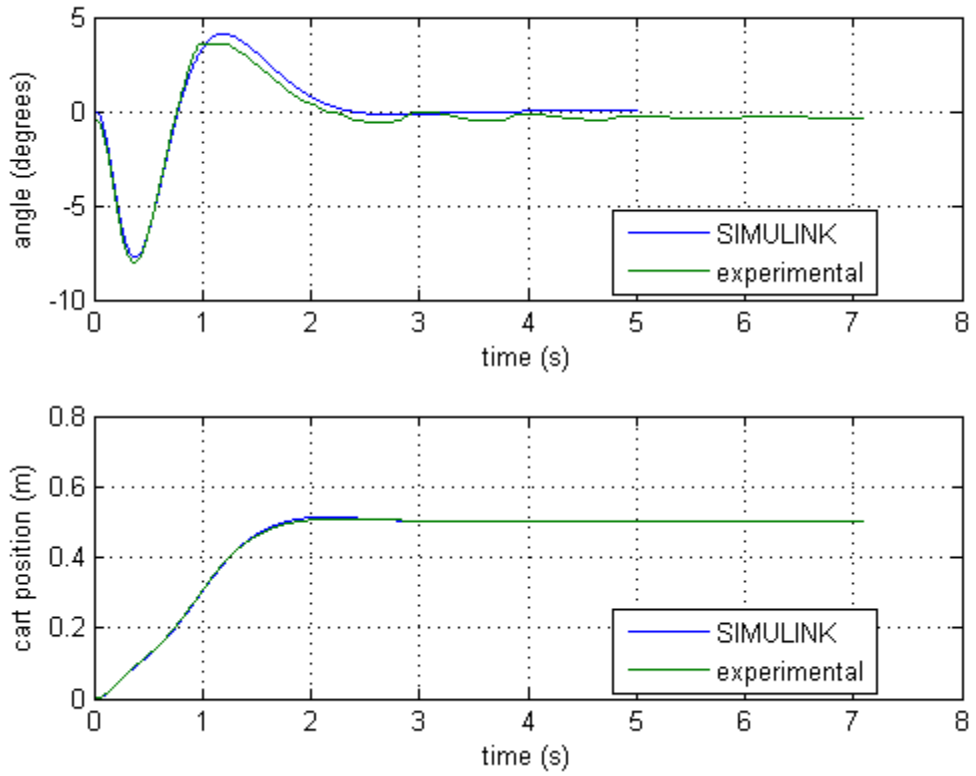

Figure 6: SIMULINK and gantry response with integral controller

The inputs and outputs for the gantry response seen in Figure 6 are tabulated in Table 7. We see that the SIMULINK response had a slightly longer settling time. Nonetheless, the real-world response and the SIMULINK response are very closely matched and the settling times are still within 10% error agreement. In both the simulation and the experimental data, the settling time for angle was less than 2.2 seconds, and the steady state position of the cart was 0.5 m, so the controller did indeed meet design specifications.

State feedback refers to the placement of poles for controlling a closed-loop feedback system. Firstly, we recognize that for a system to be stable, all poles must be negative. Second, recall from Equation 13 that the poles directly impact the eigenvalues of the system of equations of motion. From equation 14, we then saw how the eigenvalues directly impact the natural frequency and damping ratio response of the system. More specifically, a pair of complex poles has an associated natural frequency and damping ratio. One design method is to choose values for damping ratio and natural frequency and find these resulting complex poles. Then, assume that these poles are the dominant poles. Dominant poles are the poles closest to the imaginary axis – their response takes the longest time to die out. As such, their response dominates the response of the system. The remaining poles are chosen to be far left of the complex dominant poles such that the system remains stable, yet these subdominant poles have effects that die out much faster than the complex pole.

One restriction of this method is that as feedback values become too large, the real-world equipment is not able to compensate fast enough. As such, care must be taken to keep the feedback constants at low values that the hardware is able to support.

Conclusion and Recommendation

Main Points

The motion of a cart-gantry system can be described by a system of nonlinear differential equations. These equations can be linearized with the small angle approximation. The linearized system of differential equations can then be converted to a state space representation.

Transfer functions can be obtained from the state space representation to describe how a certain output is effected by certain inputs. One transfer function of interest is how voltage input effects the angle output of the gantry pendulum. We recognize that if the cart is held still, the pendulum has an associated natural frequency. When we draw a Bode plot of this transfer function, we see that the magnitude response is greatest at the natural frequency. We also observe that the steady-state amplitude of the angle response is proportional to the magnitude of the transfer function. To prove this, we apply a sinusoidal voltage input at various frequencies. As the frequency approaches the natural frequency of the pendulum, the motion becomes very large and erratic.

For part 3, we investigate a controller for the purposes of steadying the motion of the pendulum as quickly as possible for any given disturbance on the pendulum. We recognize that the feedback constants can be chosen through intelligent pole placement. More specifically, we choose 2 dominant complex poles with an associated damping ratio and natural frequency. The remaining poles are chosen far left of these dominant poles. A SIMULINK model was built to test these poles and compare the output with the experimental data. The resulting datasets had good agreement with less than 5% error in settling time. This means that SIMULINK can be used test find and test poles with hardware testing in the loop.

Part 4 was very similar to part 3. The new goal of the controller was to move the cart 0.5m down the track and then settle the pendulum motion as quickly as possible. A new state feedback variable was added to allow for the incorporation of an integral control. The integral control addition allowed for a new input that specifies the distance for the cart to be moved down the track. Similarly to as in part 3, specifications for settling time and max overshoot are given – this yields a specified natural frequency and damping ratio – which yields specified dominant complex poles. The remaining poles are chosen far left of these dominant poles. Another SIMULINK model was built to simulate and test chosen poles. However, it was discovered that as feedback constants become too large, the hardware is unable to compensate quickly enough. When poles were changed slightly to accommodate this, the SIMULINK and experimental data were in good agreement. This means that SIMULINK can be utilized for choosing and testing poles without hardware in the loop.

Theoretical/ Experimental Limitations

As discussed, the nonlinear equations of motion were linearized with the small angle approximation in order to simplify the state space representation. As such, the state space becomes unreliable as the gantry pendulum approaches large angles.

We recognize that the Bode plot is a useful tool for determining the steady state response for a transfer function over a range of input frequencies. We are able to generate these Bode plot results experimentally, but this would take a great deal of time since each input frequency data point would require its own test.

We also recognize that SIMULINK is a useful tool for testing the response of various poles. The pole placement method described in this report allows for a rough solution. These poles must be tweaked and iterated with SIMULINK in order to generate an acceptable solution. Without SIMULINK, these poles would have to be iterated upon with hardware in the loop. This is not always cheap or easy. Furthermore, this takes more time.

We also recognize that large feedback values may cause a feasible response within SIMULINK, but fail when used with the hardware. This can be attributed to the fact that the hardware cannot compensate quickly enough with large feedback values. Care must be taken to identify the response limitations of the hardware.

Personal Lessons Learned and Suggestions for Improvement.

I have personally benefited from learning how SIMULINK can be used to model and simulate real-world control problems. Pole placement methods produce close-to-accurate poles, but SIMULINK can be effectively utilized for iterating pole values and obtaining optimized feedback constants with hardware out of the loop. This is a very powerful tool.

The lab experience was very beneficial and allowed for control theory to be applied to real-world hardware. My only suggestion for improvement is that the due date for lab reports be 1.5 weeks after the lab. These reports have a great deal of content – often difficult to accomplish within 1 week for a 1 credit hour class. I greatly enjoy the content of this lab and I do not wish to rush completion of the report.

Appendix

Code part 1,2:

```
clc; clear; close all;
addpath data

%% Given constants
Rm =    2.6; %motor armature resistance (ohms)
Lm =    0.18; %motor armature inductance (mH)
Kt =    0.00767; %motor torque constant (N-m/A)
nu_m =  1; %motor efficiency ()
Km =    0.00767; %back-electromotive-force(EMF) constant (V-s/rad)
Jm =    3.90E-07; %rotor moment of inertia (kg-m^2)
Kg =    3.71; %planetary gearbox ratio ()
nu_g =  1; %planetary gearbox efficiency ()
Mc2 =   0.57; %cart mass (kg)
Mw =    0.37; %cart weight mass (kg)
Mc =    1.0731; %total cart weight mass including motor inertia (kg)
Beq =   5.4; %viscous damping at motor pinion (N-s/m)
Lt =    0.99; %track length (m)
Tc =    0.814; %cart travel (m)
Pr =    1.664E-3; %rack pitch (m/tooth)
rmp =   6.35E-3; %motor pinion radius (m)
Nmp =   24; %motor pinion number of teeth ()
rpp =   0.01482975; %position pinion radius (m)
Npp =   56; %position pinion number of teeth ()
KEP =   2.275E-5; %cart encoder resolution (m/count)
Mp =    0.23; %long pendulum mass with T-fitting (kg)
Mpm =   0.127; %medium pendulum mass with T-fitting (kg)
Lp =    0.6413; %long pendulum length from pivot to tip (m)
Lpm =   0.3365; %medium pendulum length from pivot to tip (m)
lp =    0.3302; %long pendulum length: pivot to center of mass (m)
lpm =   0.1778; %medium pendulum length: pivot to center of mass (m)
Jp =    7.88E-3; %long pendulum moment of inertia _ center of mass (kg-m^2)
Jpm =   1.2E-3; %medium pendulum moment of inertia _ center of mass (kg-m^2)
Bp =    0.0024; %viscous damping at pendulum axis (N-m-s/rad)
g =     9.81; %gravitational constant (m/s^2)

%% Part 1
load part_1_theta
t = theta.time;
t = t-t(1);
theta = theta.signals.values;
figure(1)
plot(t,theta)
xlabel('time (s)')
ylabel('angle (degrees)')
grid on

%find period of osciallation
[~,ipeak]=findpeaks(theta);
for n = 1:1:(length(ipeak)-1)
    period(n) = t(ipeak(n+1)) - t(ipeak(n));
end
period = mean(period) %s
```

```
%find natural frequency
wp = 2*pi/period %rad/s

%% Part 2
%Part a: Hand in your values for the damping ratio _ and natural frequency !n
that you calculated.
denom = [1 11.74 26.96 263.4];
r = roots(denom);
poly1 = poly(r(2:3))
wn = sqrt(poly1(3))
damping = poly1(2)/(2*wn)

%part c - bode plot
H = tf([-3.526 0],[1 11.74 26.96 263.4]);
figure(2)
bode(H)

%part d - plot experimental data
load part_2_w3_theta
t3 = theta.time - theta.time(1);
theta3 = theta.signals.values;
amp3 = (max(theta3)-min(theta3))/2;
load part_2_wn_theta
tn = theta.time - theta.time(1);
thetan = theta.signals.values;
ampn = (-min(thetan))/1;
load part_2_w7_theta
t7 = theta.time - theta.time(1);
theta7 = theta.signals.values;
amp7 = (max(theta7)-min(theta7))/2;

figure(3)
plot(t3,theta3,tn,thetan,t7,theta7)
xlabel('time (s)')
ylabel('angle (radians)')
grid on
legend('w = 3 rad/s','w = wn = 4.8 rad/s','w = 7 rad/s',0)

%part e - find magnitude of xfer function at various w
A = 3;  %voltage sinusoidal amplitude input
MAG3_calc = amp3/A
MAGn_calc = ampn/A
MAG7_calc = amp7/A
MAG3db_calc = mag2db(MAG3_calc);
MAGndb_calc = mag2db(MAGn_calc);
MAG7db_calc = mag2db(MAG7_calc);

[MAG3, ~] = bode(H,3)
[MAGn, ~] = bode(H,wn)
[MAG7, ~] = bode(H,7)
MAG3db = mag2db(MAG3);
MAGndb = mag2db(MAGn);
MAG7db = mag2db(MAG7);
```

CODE PART 3:

```
clc; clear; close all
addpath code
addpath data

%% Part 3
%initialize variables
run setup_lab_ip02_spg

%part a: simulink with adot = pi/2
p3 = -10
p4 = -20
K = place (A,B,[(-1.8182+1.9067i), (-1.8182-1.9067i), p3, p4])
X0 = [0, 0, 0, pi/2]';
sim('s_spg_pp')
t_sim = alpha_out.Time;
theta_sim = alpha_out.Data;      %deg
xc_sim = xc_out_mm.Data * 0.001;  %m
theta_properties_sim =
stepinfo([0;theta_sim],[0;t_sim],0.5,'RiseTimeLimits',[0
1],'SettlingTimeThreshold',0.05)

figure(1)
subplot(2,1,1)
plot(t_sim,theta_sim)
xlabel('time (s)')
ylabel('angle (deg)')
grid on
title('initial angluar velocity = pi/2 rad/s')
subplot(2,1,2)
plot(t_sim,xc_sim)
xlabel('time (s)')
ylabel('cart position (m)')
grid on

%part b: experimental results
load('part_3_theta_brent')
load('part_3_x_brent')
t = theta.time(1:end/2);
t = t - t(1) - 0.4;
theta = theta.signals.values(1:end/2)*57.2957795;
xc = x.signals.values(1:end/2);

%part c: find max slope
[Y,I] = max(diff(theta))
max_slope = Y/(t(I+1) - t(I))   %deg/s

%part c: sim results with same initial velocity
X0 = [0, 0, 0, max_slope*0.0174532925]';
sim('s_spg_pp')
t_sim = alpha_out.Time;
theta_sim = alpha_out.Data;      %deg
xc_sim = xc_out_mm.Data * 0.001;  %m
theta_properties = stepinfo([0;theta],[0;t],0.5,'RiseTimeLimits',[0
1],'SettlingTimeThreshold',0.05)
```

```
theta_properties_sim =
stepinfo([0;theta_sim],[0;t_sim],0.5,'RiseTimeLimits',[0
1],'SettlingTimeThreshold',0.05)

figure(2)
subplot(2,1,1)
plot(t_sim,theta_sim,t,theta,t(I),theta(I),'o')
legend('SIMULINK','experimental','max experimental slope',0)
xlabel('time (s)')
ylabel('angle (deg)')
grid on
title('initial angluar velocity provided by user push')
subplot(2,1,2)
plot(t_sim,xc_sim,t,xc)
legend('SIMULINK','experimental',0)
xlabel('time (s)')
ylabel('cart position (m)')
grid on
```

CODE PART 4:

```
clc; clear; close all
addpath code
addpath data

%% Part 3
%initialize variables
run setup_lab_ip02_spg

%part a: simulink
Ai = [A, zeros(4, 1);-1, 0, 0, 0, 0];
Bi = [B; 0];
p3 = -10;
p4 = -20;
p5 = -30;
K = place (Ai,Bi,[(-1.8182+1.9067i), (-1.8182-1.9067i), p3, p4, p5])
sim('aae364gantry2')
t_sim = alpha_out.Time;
theta_sim = alpha_out.Data;      %deg
xc_sim = xc_out.Data;  %m
figure(1)
subplot(2,1,1)
plot(t_sim,theta_sim)
xlabel('time (s)')
ylabel('angle (degrees)')
grid on
subplot(2,1,2)
plot(t_sim,xc_sim)
xlabel('time (s)')
ylabel('cart position (m)')
grid on
theta_properties_sim =
stepinfo([0;theta_sim],[0;t_sim],0.5,'RiseTimeLimits',[0
1],'SettlingTimeThreshold',0.05)
xc_properties_sim = stepinfo([0;xc_sim],[0;t_sim],0.5,'RiseTimeLimits',[0
1],'SettlingTimeThreshold',0.05)
```

```
%part b: experimental
load('part_4_theta_drake')
load('part_4_x_drake')
t = theta.time(1:end/2);
t = t - t(1);
theta = theta.signals.values(1:end/2)*57.2957795;
xc = x.signals.values(1:end/2);

K = [112.4296, -28.2173, 31.3039, 8.9735, -134.139]
sim('aae364gantry2')
t_sim = alpha_out.Time;
theta_sim = alpha_out.Data;      %deg
xc_sim = xc_out.Data;  %m
figure(2)
subplot(2,1,1)
plot(t_sim,theta_sim,t,theta)
legend('SIMULINK','experimental',0)
xlabel('time (s)')
ylabel('angle (degrees)')
grid on
subplot(2,1,2)
plot(t_sim,xc_sim,t,xc)
legend('SIMULINK','experimental',0)
xlabel('time (s)')
ylabel('cart position (m)')
grid on
theta_properties_sim =
stepinfo([0;theta_sim],[0;t_sim],0.5,'RiseTimeLimits',[0
1],'SettlingTimeThreshold',0.05)
xc_properties_sim = stepinfo([0;xc_sim],[0;t_sim],0.5,'RiseTimeLimits',[0
1],'SettlingTimeThreshold',0.05)
theta_properties = stepinfo([0;theta],[0;t],0.5,'RiseTimeLimits',[0
1],'SettlingTimeThreshold',0.05)
xc_properties = stepinfo([0;xc],[0;t],0.5,'RiseTimeLimits',[0
1],'SettlingTimeThreshold',0.05)
```