Luke Davidson
ME 5659
Pre-Lab

1.) $R_a i_a + C_1 \dot{x}_c = V_a(t)$ ①

$M_0 \ddot{x}_c + B_0 \dot{x}_c = -J_0 \ddot{\theta}_p + C_0 i_a$ ②

$J_1 \ddot{\theta}_p + B_p \dot{\theta}_p + K_p \theta_p = -J_0 \ddot{x}_c$ ③

a.) $\dfrac{\theta_p(s)}{X_c(s)}$ use ③

$$\theta_p(s)\left[J_1 s^2 + B_p s + K_p\right] = -J_0 s^2 X_c(s)$$

$$\frac{\theta_p(s)}{X_c(s)} = \frac{-J_0 s^2}{J_1 s^2 + B_p s + K_p}$$

$$= \frac{-\frac{M_p(L_p)}{2} s^2}{\frac{1}{3} M_p L_p^2 s^2 + B_p s + M_p g \frac{L_p}{2}} \cdot \frac{6}{6} = \frac{-3(M_p)(L_p)s^2}{2 M_p L_p^2 s^2 + 6 B_p s + 3 M_p g L_p}$$

$$\frac{\theta_p(s)}{X_c(s)} = \frac{-3 M_p L_p s^2}{2 M_p L_p^2 s^2 + 6 B_p s + 3 M_p g L_p}$$

b.) assess open loop stability of $\dfrac{X_c(s)}{V_a(s)}$ and $\dfrac{\theta_p(s)}{V_a(s)}$

$\dfrac{X_c(s)}{V_a(s)} = G_{cart}$;

poles of G_{cart} = roots of $V_a(s)$ =

$0,\ -16.5637,\ -0.1347 \pm 4.8145j$

$$\frac{\Theta_p(s)}{V_a(s)} = \frac{\Theta_p(s)}{X_c(s)} \cdot \frac{X_c(s)}{V_a(s)} =$$

$$\boxed{\frac{-3.504s}{s^3 + 16.83s^2 + 27.66s + 384.2} = \frac{\Theta_p(s)}{V_a(s)}}$$

(after minreal)

poles = -16.5637, $-0.1347 \pm 4.8745j$

because all poles in $\frac{\Theta_p(s)}{V_a(s)}$ are in left half plane.

$\frac{\Theta_p(s)}{V_a(s)}$ is open loop absolute stable.

---------------------------------- Part 1b ----------------------------------

To assess the open loop stability of the cart and pendulum systems, we analyze the poles of the transfer functions, or roots of the denominators of the transfer functions.

Poles of Xc(s)/Va(s):

```
  0.0000 + 0.0000i
-16.5637 + 0.0000i
 -0.1347 + 4.8145i
 -0.1347 - 4.8145i
```

Poles of Θp(s)/Xc(s):

```
-16.5637 + 0.0000i
 -0.1347 + 4.8145i
 -0.1347 - 4.8145i
```

Since the poles of Θp(s)/Xc(s) all have negative real parts, they all lie in the left half plane, meaning the system is open-loop stable. Because the pole at the origin in Xc(s)/Va(s) does not lie in the left half plane, the Xc(s)/Va(s) system is open-loop unstable.

2.) a) voltage cart position system using PD controller

① $t_s < 5_s$
② % overshoot < 5%
③ < 5% ss pos. error
④ pend angle stays within $\pm 5°$ of vert for 0.1 m step input

$$G_p = \frac{X_c(s)}{V_a(s)} = \frac{0.05434s^2 + 0.004137s + 1.247}{0.03628s^4 + 0.6107s^3 + 1.004s^2 + 13.94s}$$

$$G_c = K_p + K_d \cdot s$$

$$e^{\left(\frac{-\pi \zeta}{\sqrt{1-\zeta^2}}\right)} < 0.05$$

$$-\pi \zeta < \ln(0.05)\left[\sqrt{1-\zeta^2}\right]$$

$$\pi^2 \zeta^2 < [\ln(0.05)]^2 (1-\zeta^2)$$

$$\zeta^2 \left(\pi^2 + [\ln(0.05)]^2\right) < [\ln(0.05)]^2$$

$$\zeta < \sqrt{\frac{[\ln(0.05)]^2}{\pi^2 + [\ln(0.05)]^2}} \Longrightarrow \boxed{\zeta < 0.69}$$

$$t_s = \frac{4}{\zeta \omega_n} \;;\; \frac{4}{\zeta\omega_n} < 5 \Longrightarrow 4 < 5\zeta\omega_n$$

$\omega_d =$

$$\omega_n > \frac{4}{5\zeta} \quad \boxed{\omega_n > 1.16}$$

desired closed poles @ $-\zeta\omega_n \pm \omega_d$;

$$\omega_d = \omega_n\sqrt{1-\zeta^2}$$

$$\boxed{\omega_d = 0.8396} = 1.16\sqrt{1-(0.69)^2}$$

desired closed loop poles @ $-\zeta\omega_n \pm \omega_d j$

$$= -(0.69)(1.16) \pm 0.8396j$$

$$= -0.8 \pm 0.8396j$$

$$\phi(-0.8 \pm 0.8396j) = \pm 2.3926$$

$$\alpha = \pi - 2.3926 = 0.749$$

$$z_c = (0.69)(1.16) + \frac{0.8396}{\tan(0.749)} = \quad 1.703 = z_c$$

$$K_d = \frac{1}{|G_c|} = \frac{1}{|(s_{d_1} + z_c)\,G(s=d)|} = \frac{0.1}{0.1005} = \boxed{K_d = 9.95}$$

$$K_p = z_c K_d = 1.703(9.95) = \boxed{K_p = 16.945}$$

---------------------------------- Part 2a ----------------------------------

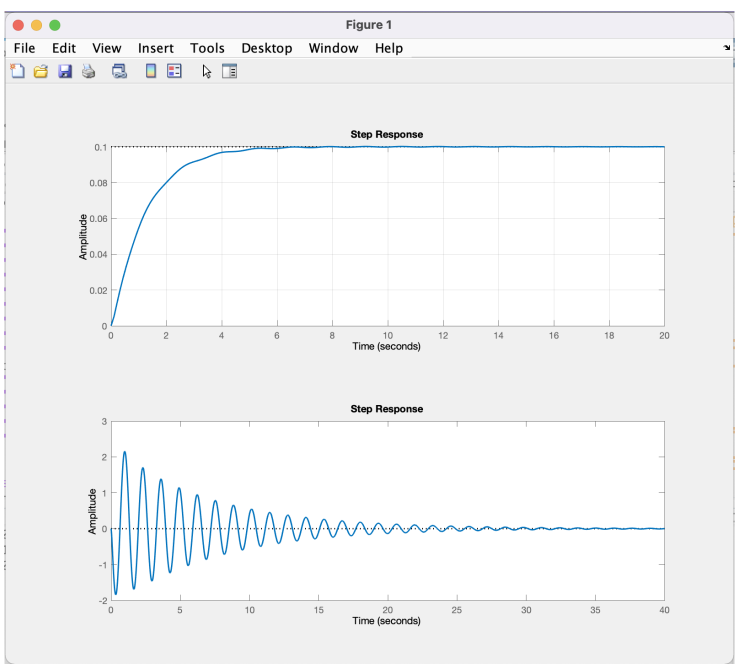
Referencing the calculations done by hand, given the system specifications, desired closed loop poles lie around -0.8 +/- 0.8396i. Calculating Kp and Kd (shown by hand) based off of these desired poles yields:

Kp = < ~16.9
Kd = < ~9.95

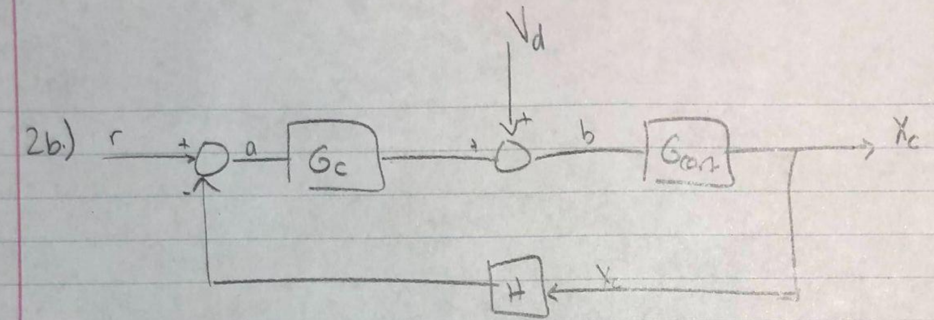
Because these values are approximate values based on the extremes of the design specifications, I further tuned them to best fit the specifications and to be as low as possible. Using Kp = 8.5 and Kd = 0.1, the following results were obtained:

RiseTime: 2.5534
TransientTime: 4.8193
SettlingTime: 4.8193
SettlingMin: 0.0903
SettlingMax: 0.1002
Overshoot: 0.1545
Undershoot: 0
Peak: 0.1002
PeakTime: 9.1940

In further analyzing the response of the system in Figure 1, we can clearly see the cart settling time is less than 5 seconds, there is minimal overshoot and steady-state error, and the pendulum angle stays within +/- 5 degrees of vertical for a 0.1 m step input.



2b.)



$a = r - X_c \qquad r = 0,\ H = 1$

$b = V_d + ((r - X_c)G_c) \qquad X_c = b(G_{cart})$

$X_c = (V_d - X_c G_c) G_{cart}$

$\boxed{\text{want } \dfrac{X_c}{V_d}}$

$X_c = V_d G_{cart} - X_c G_c G_{cart}$

$X_c [1 + G_c G_{cart}] = V_d G_{cart}$

$\boxed{\dfrac{c}{k_p c} < 0.05}$

$$\boxed{\frac{X_c}{V_d} = \frac{G_{cart}}{1 + G_c G_{cart}}}$$

- conditions on K_p and K_d to keep 5% ss-error for step disturbance

$$\lim_{s \to 0} \frac{1}{s}\left(\frac{G_{cart}}{1 + G_c G_{cart}}\right) < 0.05$$

$$= \lim_{s \to 0} \frac{0.05434 s^2 + 0.004137 s + 1.247}{0.03628 s^5 + 0.6107 s^4 + 1.004 s^3 + 13.94 s^2} \cdot \left(\frac{1}{.03628 s^4 + .6107 s^3 + 1.004 s^3 + 13.94 s^2 + (K_p + K_d s)(.05434 s^2 + .004137 s + 1.247)}\right)$$

$$\Longrightarrow \frac{1.247}{K_p(1.247)} < 0.05 \Longrightarrow \frac{1}{K_p} < 0.05 \Longrightarrow K_p > 20$$

to keep < 5% ss error for step disturbance

---------------------------------- Part 2b ----------------------------------

The derivation of the TF with disturbance as the input and cart position as the output, Xc(s)/Vd(s), is shown in by hand and is represented numerically as:

```
Xc_Vd_tf =

        0.001972 s^6 + 0.03334 s^5 + 0.1023 s^4 + 1.523 s^3 + 1.309 s^2 + 17.38 s
  ---------------------------------------------------------------------------------------------
  0.001316 s^8 + 0.04451 s^7 + 0.4668 s^6 + 2.547 s^5 + 19.11 s^4 + 41.82 s^3 + 207.8 s^2 +
156.4 s

Continuous-time transfer function.
```

We can calculate the steady-state error in cart position subject to a 1-volt disturbance using the final value theorem. It is the limit as s approaches 0 of s*Xc(s)/Vd(s), where Vd(s) is simply 1. Calculated in MATLAB, the steady-state error is seen to be 0:

```
lim =

0
```

To calculate the Kp and Kd limitations that will keep our steady-state error less than 5 percent for a step disturbance, we can calculate the limit as s approaches 0 of (1/s)*(Xc(s)/Vd(s)), where Gc = Kp + Kd*s. Solving this yields:

Kp > 20

This calculation can be seen by hand in the attached sheet.

3.) a) with $G_c(s) = \frac{0.1(s+10)}{0.02s+1} = \frac{0.1s+1}{0.02s+1}$

$T_s = .001$

method = tustin ; output of matlab =

$$H(z) = \frac{4.902z - 4.854}{z - 0.9512} = \frac{Y}{U}$$

$$\Rightarrow \frac{Y(z - 0.9512)}{z} = \frac{U(4.902z - 4.854)}{z}$$

$$\Rightarrow Y(1 - 0.9512z^{-1}) = U(4.902 - 4.854z^{-1})$$

convert to difference equation

$$y[n] - 0.9512\,y[n-1] = 4.902\,u[n] - 4.854\,u[n-1]$$

$$\Rightarrow y[n] = 4.902\,u[n] - 4.854\,u[n-1] + 0.9512\,y[n-1]$$

---------------------------------- Part 3a ----------------------------------

Using c2d, Ts = 0.001 s, and the Tustin method, H(z) was calculated as:

Hz =

4.902 z - 4.854
\---------------
z - 0.9512

Sample time: 0.001 seconds
Discrete-time transfer function.

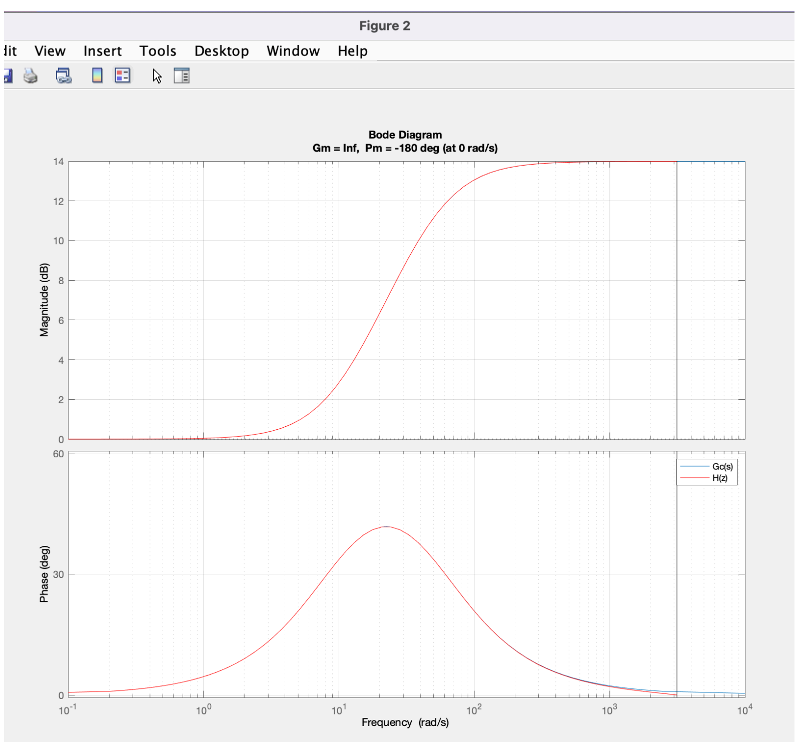
Using this result, the calculation of the difference equation is done by hand on the attached sheet.

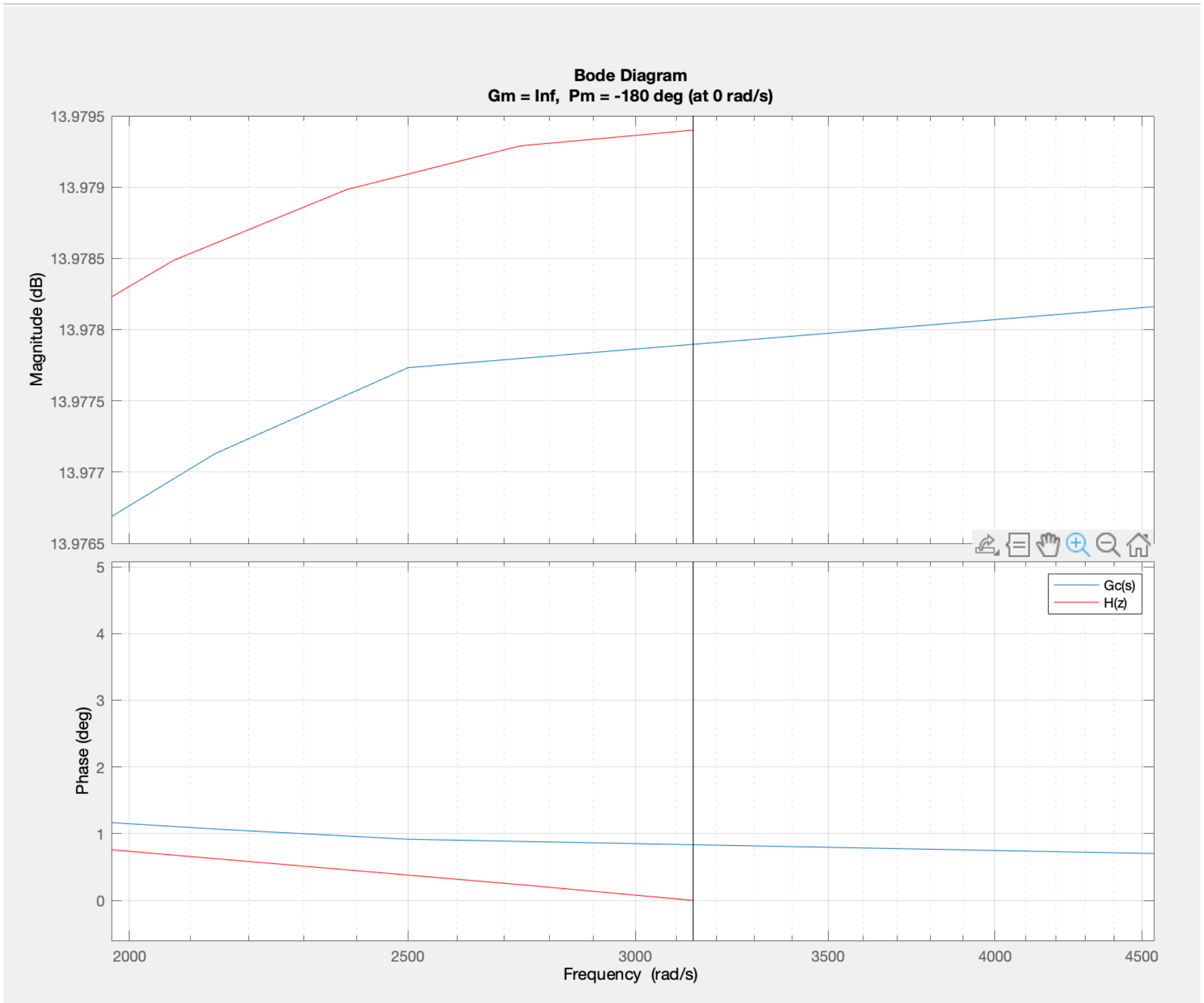
\---

---------------------------------- Part 3b ----------------------------------

Referencing the bode plots of Gc(s) and H(z) in Figure 2, it is seen that the two plots are very similar, only with slight differences at higher frequencies around 200 rad/s. At around 200 rad/s and higher, it can be seen that the discrete approximated controller continues lower in phase and higher in magnitude. A zoomed in image of the bode graphs at a frequency of around 3000 rad/s is shown in Figure 3 to better show the graph differences.

\---




Bode Diagram
Gm = Inf, Pm = -180 deg (at 0 rad/s)
Magnitude (dB)
13.9795
13.979
13.9785
13.978
13.9775
13.977
13.9765
Phase (deg)
5
4
3
2
1
0
Gc(s)
H(z)
2000
2500
3000
3500
4000
4500
Frequency (rad/s)

---------------------------------- Part 4a ----------------------------------

The eigenvalues of A and the poles of Gcart are calculated below and are clearly equivalent:

Eigenvalues of A:

```
  0.0000 + 0.0000i
-16.5637 + 0.0000i
 -0.1347 + 4.8145i
 -0.1347 - 4.8145i
```

Poles of Gcart:

```
  0.0000 + 0.0000i
-16.5637 + 0.0000i
 -0.1347 + 4.8145i
 -0.1347 - 4.8145i
```

We can also confirm that the system is fully controllable by calculating the controllability Gramian and its rank:

```
controllability =

   1.0e+04 *

        0    0.0001   -0.0025    0.0415
   0.0001   -0.0025    0.0415   -0.6875
        0   -0.0004    0.0059   -0.0896
  -0.0004    0.0059   -0.0896    1.4795


rank =

     4
```

We can see the controllability matrix is full rank, or rank 4, confirming that the system is fully controllable.

---------------------------------- Part 4b ----------------------------------

Using Ackermann's formula, I was able to calculate a gain vector K such that the :closed loop poles would all be placed at -1:

```
K =

    0.0291  -11.0629    5.8346   -1.0670
```

To confirm this controller gain vector is correct, we can calculate Acl = A-BK and its eigenvalues, which should be equal to -1:

Acl =

0	1.0000	0	0
-0.0436	-0.1742	-7.2691	1.6031
0	0	0	1.0000
0.1019	0.4074	-5.9432	-3.8258

Eigenvalues of Acl:

-1.0003 + 0.0003i
-1.0003 - 0.0003i
-0.9997 + 0.0003i
-0.9997 - 0.0003i

As we can see, the eigenvalues of Acl are all extremely close to -1, confirming the controller is correct.

---------------------------------- Part 4c ----------------------------------

Using the Linear Quadratic Regulator (LQR) method, I was able to reduce the pendulum angle to less than +/- 0.5 deg (0.0087 rad) within 10 seconds while keeping the input voltage to less than +/- 5 Volts, as seen in the figures below. Voltage was calculated using the equation U = -Klqr * X. The Q and R cost values I used were:

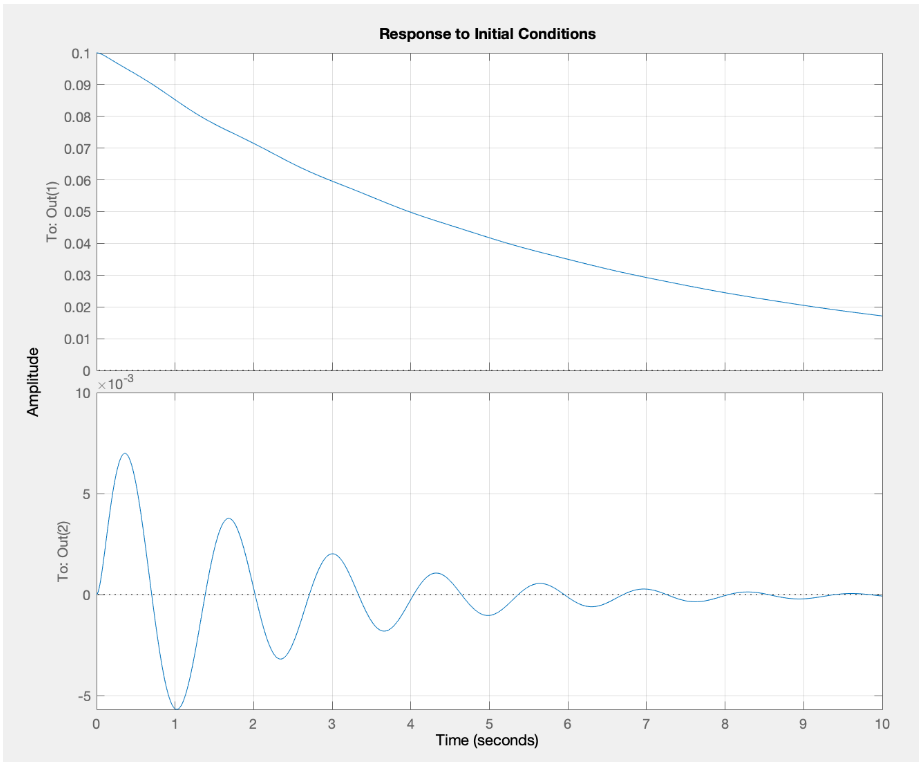
Q =

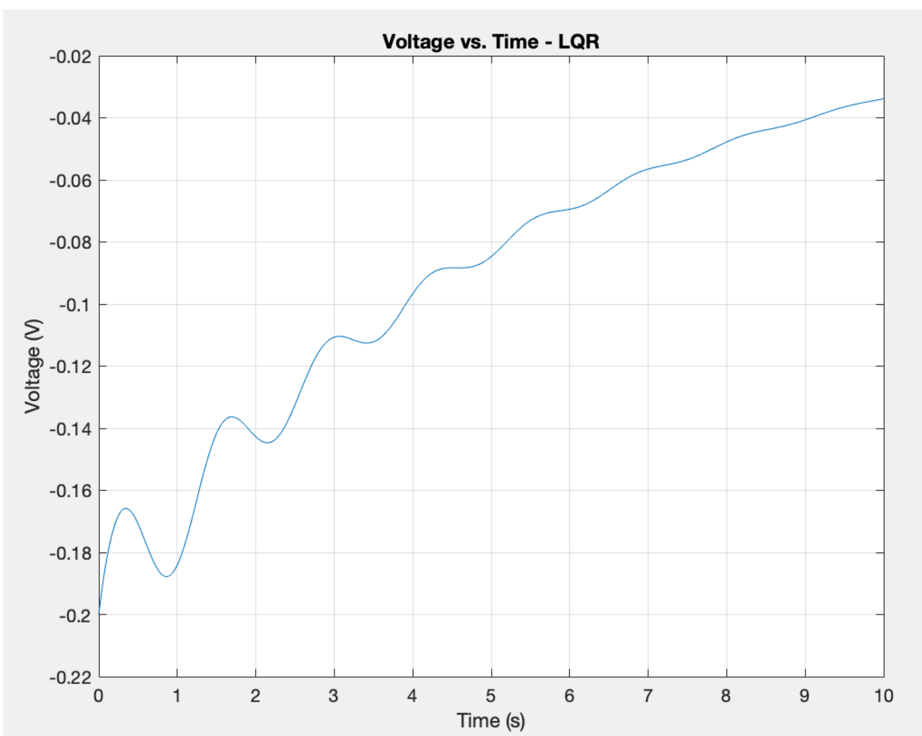
10	0	0	0
0	3	0	0
0	0	1	0
0	0	0	2

R =

2.5000

The rows in the Q matrix are associated with each state: Xc, X-dot_c, Θp, Θp-dot. When a higher number is added to the diagonal, the cost of that specific state is increased, meaning it is more computationally expensive to alter that state. R is similar to Q, except it works on the input matrix, thus mainly affecting the input voltage for our system. I altered the costs of cart position, velocity, angle and angular velocity until I was satisfied with the response of my system.





WHOLE MATLAB CODE

```
% Luke Davidson
% ME 5659 Pre-Lab
clc;
clear all;
close all;
%% Constants
% problem constants
Mp = 0.23; % pendulum mass (kg)
Mc = 0.96; % cart mass (kg)
Jm = 3.90e-7; % motor rotor mass moment of inertia (kg-m^2)
N = 3.71; % gearbox gear ratio (3.71:1)
rg = 0.00635; % motor pinion gear radius (m)
Lp = 0.6413; % pendulum ("long" pendulum) length (m)
Bc = 5.4; % cart linear viscous friction/damping coeff. (N-s/m)
Bm = 0.000018; % motor shaft viscous friction/damping coeff. (Nm-s/rad)
Kt = 0.00767; % motor torque constant (Nm/Amp)
g = 9.81; % local acceleration due to gravity (m/s^2)
KB = 0.00767; % motor back -EMF constant (V-s/rad)
Bp = 0.0024; % pendulum axis viscous friction/damping coeff. (Nm-s/rad)
Ra = 2.6; % motor armature (winding) resistance (Ohms)
M0 = Mp + Mc + Jm*N^2/rg^2;
J1 = (1/3)*Mp*Lp^2;
J0 = (1/2)*Mp*Lp;
B0 = Bc + Bm*N^2/rg^2;
C0 = N*Kt/rg;
Kp = (1/2)*Mp*g*Lp;
C1 = N*KB/rg;
a4 = M0*J1-J0^2;
a3 = M0*Bp + B0*J1 + C0*C1*J1/Ra;
a2 = M0*Kp + B0*Bp + C0*C1*Bp/Ra;
a1 = B0*Kp + C0*C1*Kp/Ra;
a0 = 0;
b2 = C0*J1/Ra;
b1 = C0*Bp/Ra;
b0 = C0*Kp/Ra;
%% Question 1
% This is the TF , Theta(s)/X(s), which can be used to determine the
% response of the pendulum to cart motion
T_over_x = tf([-J0 0 0],[J1 Bp Kp]);
% cart TF , Xc(s)/Va(s)
num = [b2 b1 b0];
den = [a4 a3 a2 a1 a0];
Gcart = tf([b2 b1 b0],[a4 a3 a2 a1 a0]);
                % Part 1b
theta_div_v = minreal(T_over_x*Gcart);
isstable(theta_div_v);
```

```
fprintf('------------------------------------ Part 1b
------------------------------------\n')
fprintf('\tTo assess the open loop stability of the cart and pendulum
systems,\n')
fprintf('we analyze the poles of the transfer functions, or roots of the
denominators\n')
fprintf('of the transfer functions.\n\n')
fprintf('\tPoles of Xc(s)/Va(s):\n')
disp(pole(Gcart))
fprintf('\tPoles of Θp(s)/Xc(s):\n')
disp(pole(theta_div_v))
fprintf('\tSince the poles of Θp(s)/Xc(s) all have negative real parts, they
all\n')
fprintf('lie in the left half plane, meaning the system is open-loop stable.
Because\n')
fprintf('the pole at the origin in Xc(s)/Va(s) does not lie in the left half
plane,\n')
fprintf('the Xc(s)/Va(s) system is open-loop unstable.')
fprintf('\n--------------------------------------------------------------------------------
-----------\n\n\n')
%% Question 2
s = tf('s');
% PD controller , pick some gains
Kp = 9; Kd = 0.1; %16.945, 5
Gc = Kp+Kd*s;
% 10cm reference input applied to closed -loop cart system
subplot (2,1,1);
step (0.1* feedback(Gcart*Gc ,1) ,20); grid on;
stepinfo (0.1* feedback(Gcart*Gc ,1));
% pendulum response to same 10cm cart position reference input
% (convert radian output to degrees)
subplot (2,1,2);
step (0.1* feedback(Gcart*Gc ,1)*T_over_x *180/ pi)
set(findall(gcf ,'type','line'),'linewidth',1.5) % make plot lines bolder
fprintf('------------------------------------ Part 2a
------------------------------------\n')
fprintf('\tReferencing the calculations done by hand, given the system
specifications,\n')
fprintf('desired closed loop poles lie around -0.8 +/- 0.8396i. Calculating Kp
and Kd (shown\n')
fprintf('by hand) based off of these desired poles yields:\n\n')
fprintf('\tKp = < ~16.9\n')
fprintf('\tKd = < ~9.95\n\n')
fprintf('\tBecause these values are approximate values based on the extremes of
the design\n')
fprintf('specifications, I further tuned them to best fit the specifications
and to be as low\n')
fprintf('as possible. Using Kp = 8.5 and Kd = 0.1, the following results were
obtained:\n\n')
```

```
disp(stepinfo(0.1* feedback(Gcart*Gc ,1)));
fprintf('\tIn further analyzing the response of the system in Figure 1, we can
clearly\n')
fprintf('see the cart settling time is less than 5 seconds, there is minimal
overshoot\n')
fprintf('and steady-state error, and the pendulum angle stays within +/- 5
degrees of\n')
fprintf('vertical for a 0.1 m step input.')
fprintf('\n--------------------------------------------------------------------
-----------\n\n\n')
Xc_Vd_tf = Gcart/(1 + Gcart*Gc);
s = tf('s');
Xc_tf_s = Xc_Vd_tf * s;
[Num,Den] = tfdata(Xc_tf_s,'v');
syms s
sys_syms=poly2sym(Num,s)/poly2sym(Den,s);
lim = limit(sys_syms,s,0);
Gcart;
fprintf('----------------------------------- Part 2b
-----------------------------------\n')
fprintf('\tThe derivation of the TF with disturbance as the input and cart
position as\n')
fprintf('the output, Xc(s)/Vd(s), is shown in by hand and is represented
numerically as:\n')
Xc_Vd_tf
fprintf('\tWe can calculate the steady-state error in cart position subject to
a 1-volt\n')
fprintf('disturbance using the final value theorem. It is the limit as s
approaches 0 of\n')
fprintf('s*Xc(s)/Vd(s), where Vd(s) is simply 1. Calculated in MATLAB, the
steady-state\n')
fprintf('error is seen to be 0:\n')
lim
fprintf('\tTo calculate the Kp and Kd limitations that will keep our
steady-state error\n')
fprintf('less than 5 percent for a step disturbance, we can calculate the limit
as s\n')
fprintf('approaches 0 of (1/s)*(Xc(s)/Vd(s)), where Gc = Kp + Kd*s. Solving
this yields:\n\n')
fprintf('\tKp > 20\n\n')
fprintf('This calculation can be seen by hand in the attached sheet.')
fprintf('\n--------------------------------------------------------------------
-----------\n\n\n')
%% Question 3
Gcs = tf([0.1 1],[.02 1]);
Hz = c2d(Gcs,0.001,'tustin');
figure
margin(Gcs)
hold on
```

```
margin(Hz,'r')
grid on
legend('Gc(s)','H(z)')
hold off
fprintf('----------------------------------- Part 3a
-----------------------------------\n')
fprintf('\tUsing c2d, Ts = 0.001 s, and the Tustin method, H(z) was calculated
as:\n')
Hz
fprintf('\tUsing this result, the calculation of the difference equation is
done by hand\n')
fprintf('on the attached sheet.')
fprintf('\n-----------------------------------------------------------------------------
-----------\n\n\n')
fprintf('----------------------------------- Part 3b
-----------------------------------\n')
fprintf('\tReferencing the bode plots of Gc(s) and H(z) in Figure 2, it is seen
that\n')
fprintf('the two plots are very similar, only with slight differences at higher
frequencies\n')
fprintf('around 200 rad/s. At around 200 rad/s and higher, it can be seen that
the discrete\n')
fprintf('approximated controller continues lower in phase and higher in
magnitude. A zoomed\n')
fprintf('in image of the bode graphs at a frequency of around 3000 rad/s is
shown in Figure\n')
fprintf('3 to better show the graph differences.')
fprintf('\n-----------------------------------------------------------------------------
-----------\n\n\n')
%% Question 4
% state -space representation of the system , with states:
% x1 = xc
% x2 = xc ’
% x3 = theta_p
% x4 = theta_p ’
A= [0 1 0 0;
0 -0.1674554527e2 0.1470698608e1 0.4878730578e-2;
0 0 0 1;
0 0.3916781211e2 -0.2638554173e2 -0.8752843619e-1];
B= [0;
0.1497917088e1;
0;
-0.3503626434e1];
C=[1 0 0 0;
0 0 1 0];
D=[0;
0];
% the C/D matrices define outputs:
% y1 = xc
```

```
% y2 = theta_p
% form the state -space
sys = ss(A,B,C,D);
eig(A);
pole(Gcart);
controllability = ctrb(A,B);
rank = rank(ctrb(A,B));
K = acker(A,B,[-1 -1 -1 -1]);
Acl = A - B*K;
eig(Acl);
fprintf('----------------------------------- Part 4a
-----------------------------------\n')
fprintf('\tThe eigenvalues of A and the poles of Gcart are calculated below and
are\n')
fprintf('clearly equivalent:\n\n')
fprintf('\tEigenvalues of A:\n')
disp(eig(A))
fprintf('\tPoles of Gcart:\n')
disp(pole(Gcart))
fprintf('\tWe can also confirm that the system is fully controllable by
calculating the\n')
fprintf('controllability Gramian and its rank:\n')
controllability
rank
fprintf('\tWe can see the controllability matrix is full rank, or rank 4,
confirming that\n')
fprintf('the system is fully controllable.')
fprintf('\n------------------------------------------------------------------------
-----------\n\n\n')
fprintf('----------------------------------- Part 4b
-----------------------------------\n')
fprintf("\tUsing Ackermann's formula, I was able to calculate a gain vector K
such that the\n:")
fprintf('closed loop poles would all be placed at -1:\n')
K
fprintf('')
fprintf('\tTo confirm this controller gain vector is correct, we can calculate
Acl = A-BK and\n')
fprintf('its eigenvalues, which should be equal to -1:\n')
Acl
fprintf('\tEigenvalues of Acl:\n')
disp(eig(Acl))
fprintf('\tAs we can see, the eigenvalues of Acl are all extremely close to -1,
confirming\n')
fprintf('the controller is correct.')
fprintf('\n------------------------------------------------------------------------
-----------\n\n\n')
Q = [10 0 0 0;    % cart pos 30; 50;
    0 3 0 0;    % cart vel 5; 20;
```

```
    0 0 1 0;     % cart angle 1; 1;
    0 0 0 2];     % cart angular rate 15; 15;
R = 2.5; %12.5; 30;
Klqr = lqr(A,B,Q,R);
syslqr = ss((A-B*Klqr),B,C,D);
x0 = [0.1;0;0;0];
t = 0:0.01:10;
[y,t,x] = initial(syslqr,x0,t);
size = size(x,1);
for i = 1:size
   V(i,:) = -Klqr*x(i,:)';
end
plot(t,V); grid on
ylabel('Voltage (V)'); xlabel('Time (s)'); title('Voltage vs. Time - LQR');
figure
initial(syslqr,x0,t); grid on;
fprintf('----------------------------------- Part 4c
-----------------------------------\n')
fprintf('\tUsing the Linear Quadratic Regulator (LQR) method, I was able to
reduce the\n')
fprintf('pendulum angle to less than +/- 0.5 deg (0.0087 rad) within 10 seconds
while keeping the\n')
fprintf('input voltage to less than +/- 5 Volts, as seen in the figures below.
Voltage was\n')
fprintf('calculated using the equation U = -Klqr * X. The Q and R cost values I
used were:\n')
Q
R
fprintf('\tThe rows in the Q matrix are associated with each state: Xc,
X-dot_c, Θp, Θp-dot.\n')
fprintf('When a higher number is added to the diagonal, the cost of that
specific state\n')
fprintf('is increased, meaning it is more computationally expensive to alter
that state.\n')
fprintf('R is similar to Q, except it works on the input matrix, thus mainly
affecting\n')
fprintf('the input voltage for our system. I altered the costs of cart
position, velocity,\n')
fprintf('angle and angular velocity until I was satisfied with the response of
my system.')
fprintf('\n-----------------------------------------------------------------------------
-----------\n\n\n')
```