HW 3

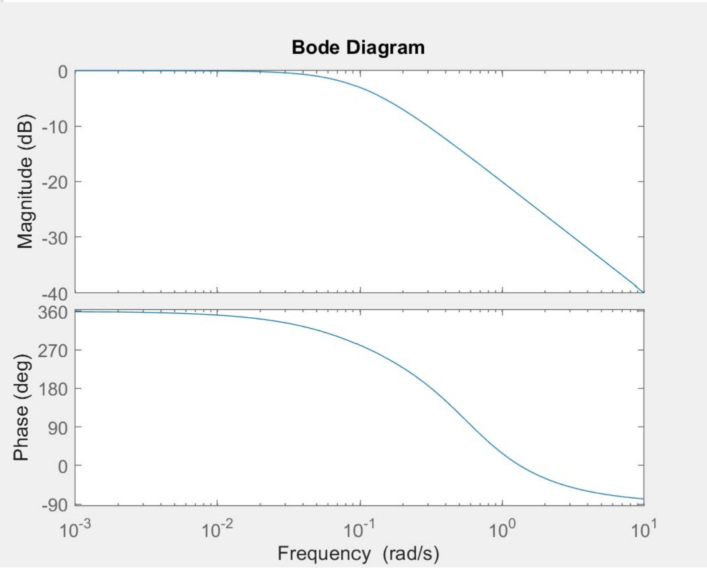
Wednesday, November 15, 2023 11:07 PM

https://github.com/axgib/Classes/tree/main/MAE144

Q1

```
%% MT1 MAE 144 redo - HW 3
clear; clc;

s = tf('s');

% dy(t)/dt + a0*y(t) = b0 * u(t - d)
% s*Y(s) + a0*Y(s) = b0 * U(s - d)
a0 = 0.1;
b0 = 0.1;

G = tf(b0, [1 a0], 'InputDelay', 6)%Y(s)/U(s) = e^(-6s)*(b0)/(s-a)
G = pade(G, 2);
```

```
G =

                 0.1
  exp(-6*s) * -------
               s + 0.1

Continuous-time transfer function.
Model Properties
>>
```

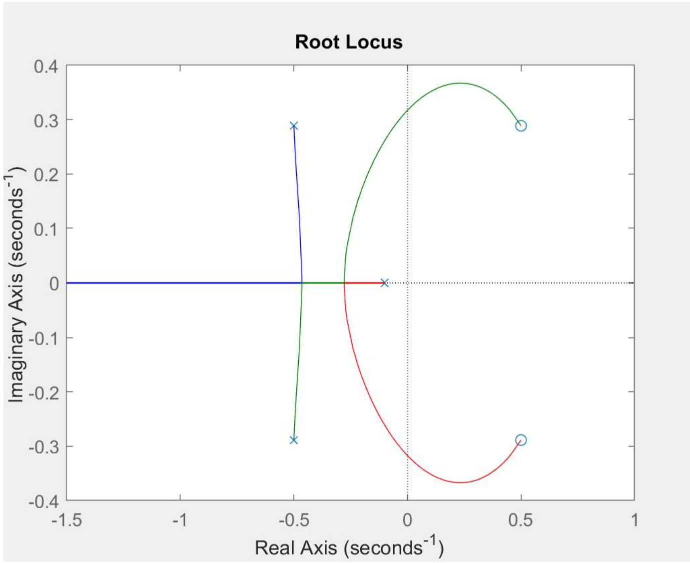


Q2

```
%% Find the Ku and omega_u (critical K and omega)
figure;
Ku = 3.3
rlocus(Ku*G)

w_u = 0.317; %rad/s
```

```
%% P2 GM
% GM = 0.6*Ku
para = [0.6, inf, 0]; %PID parameters

GM = 0.6*Ku %This is the amount of gain margin from stable we have to work
%with given the ZP conditions

%Dpid = Kp*(1 + 1/(Ti*s) + Td*s) too tough to calc
```

GM = 1.98

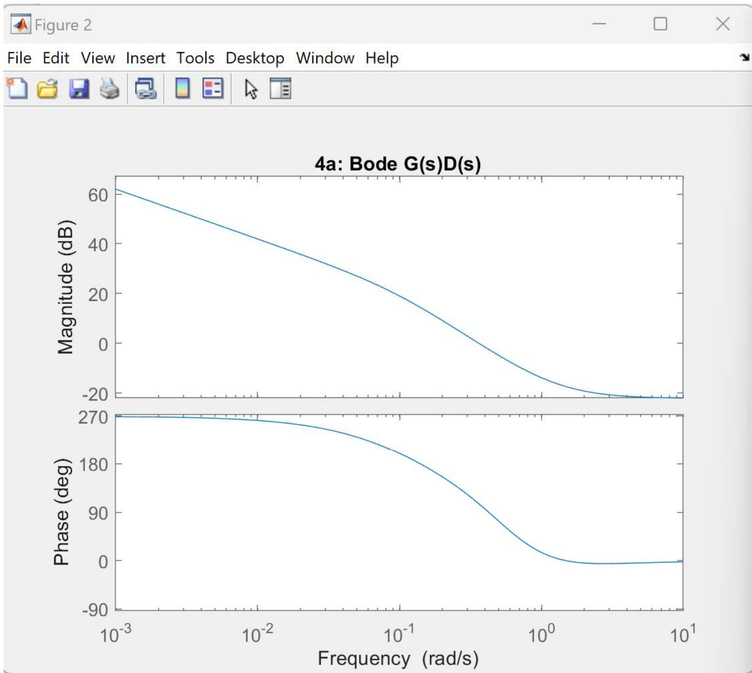


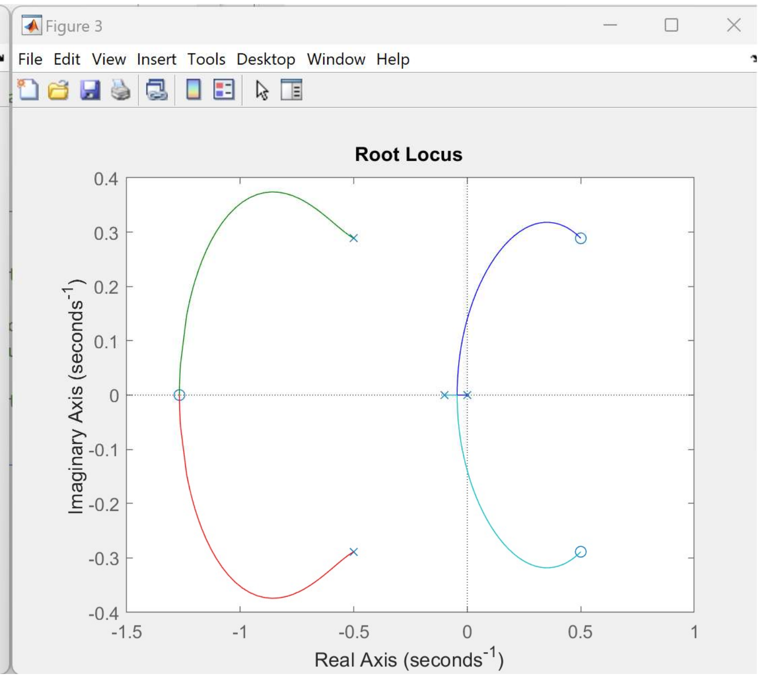
Q3

```
%% P3 Dial down beta para(1), Dial up gamma para(3)

%We build a PID controller with real parameters once the parameter
% give us our PID control constants of K, Ti, Tp.
%With PID control we can automatically control the temperature of the cell
% bath with our 2 temperature system by bringing water in and out to get
% our desired temperature.
% If the frequency is too high or too low our controller can act with
%a higher response to get our temperature back to the target
```

Q4

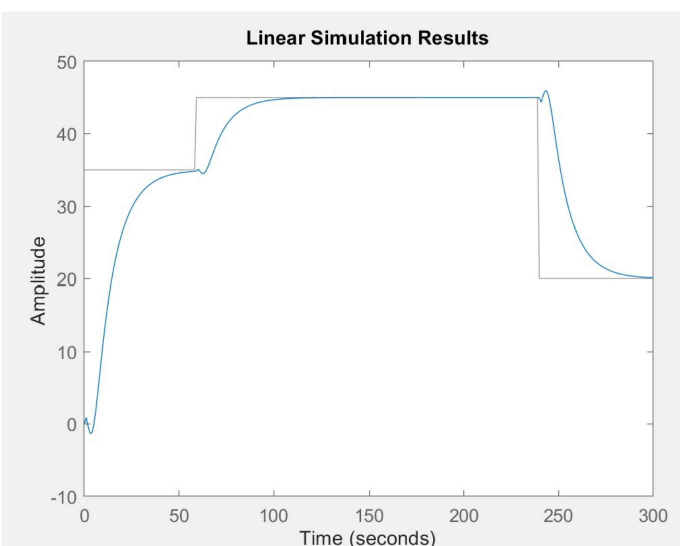
```
%% P4a Z-N PID Bode G(s)D(s),

para = [0.6, 0.5, 0.125];
Tu = 1/w_u;

Kp = para(1) * Ku;
Ti = para(2) * Tu;
Tp = para(3) * Tu;

w_i = 1/Ti;
w_d = 1/Tp;

D = Kp * (1 + 1/(Ti*s) + (Tp * s))
figure;
bode(G*D)
title("4a: Bode G(s)D(s)")
figure;
rlocus(G*D)
```



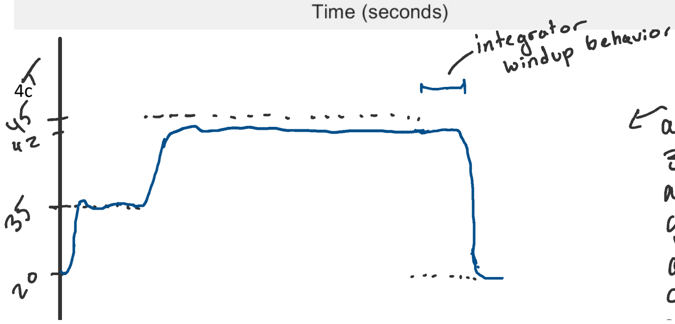


Q 4b

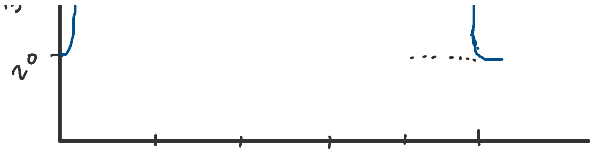
```
%% P4b Profile:
% 35C for 1hr
% 45C for 3hr
% 20C for 1hr
% assume PM is 70deg and crossover is 0.2rad/s

t = linspace(0, 300, 300); %300min in 5hours
u(1:1*60 - 1) = 35;
u(1*60: 4*60) = 45;
u(4*60: 5*60) = 20;

figure;
lsim(G, u , t)
```



4c



there will be a strong delay when we attempt to change temps because for 3hrs the controller was trying to reach a temp that it could not. Integrator got very large over a long time and thus will take time to get down again so that the system



got very large over a long time
and thus will take time to get
down again so that the system
can change temperature.

Q 5

```
%% 5a What controller is better than a PID
% Lead lag cuz you 1)dont get integrator windup as we saw in exercise 4c if
% the controller is not meeting it's target temperature it will try to
% keep building so when it is time to come down again there is a large
% delay before the system is able to change temperature
% 2) dont amplify high freq - PIDs also amplify high frequencies which is
% bad for things like noise amplification. usually noise operates at the
% higher frequencies so we making this worse with a PID.

%more detail
```

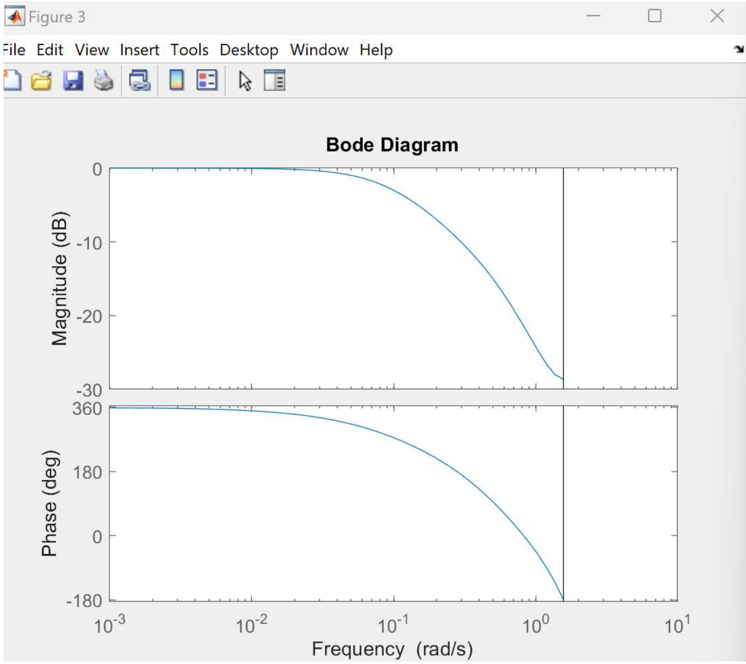
```
%% 5b Change to the physical system
% Getting rid of the delay would allow us to control the system faster and
% with greater accuracy. Maybe we more directly change the water
% temperatuer with a shorter pipe from the valve source to the bath so
% there is less convective transport delay. Maybe we also suppliment the
% bath with additional heating elements so we can reach our desired
% temperature too.
```

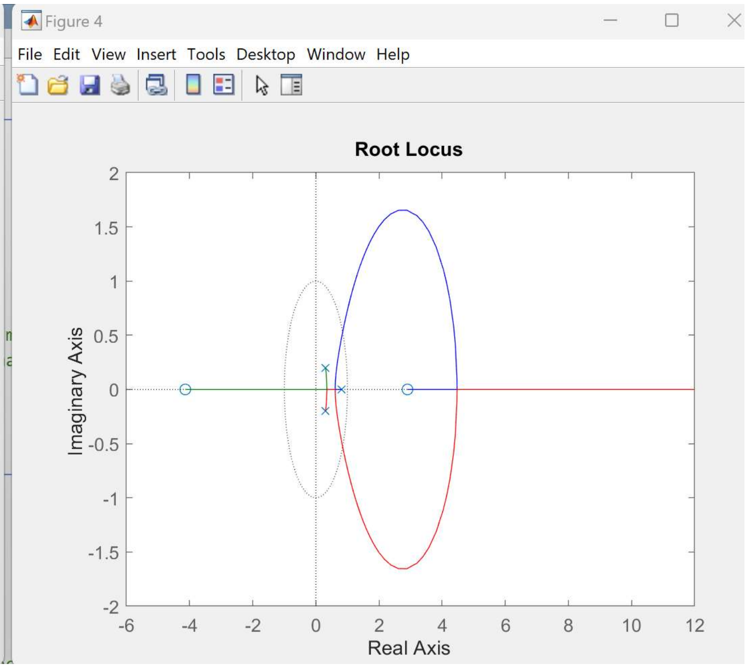
Q6

```
%% 6a DAC - G(s) - ADC
h = 2; %sec
Gz = c2d(G, 2, 'zoh')

figure;
bode(Gz)
figure;
rlocus(Gz)
% Degree of denominator is larger than the degree on the numerator therfore
% the system is causal. System is unstable because of the many zero outside
% the unit circle.
```

```
%% 6b Deadbeat control
z = tf('z');
Tz = 1/z^(3 - 2);

Dz = (Tz/Gz)/(1 - Tz)

%This is a deadbeat controller.
%Yes we expect a significant amount of intersample ripple because of the
%unsteady pole/zero cancellation. Deadbeat control is based on the idea of
%using the inverse of plant to cancel it's own dynamics. It does this by
%canceling the pole/zeros. But we have a zero outside of the unit circle
%thus our system isn't stable. We do have minimal phase.
%I propose that we use a ripple free deadbeat controller
```

```
%% 6c
%Given our untable system and the temperature control, this controller
%would not be a suitable fit because of the large amount of oscilitory
%behavior in the continous time domain. While it may look like we are under
%control in the discrete time, we might not be sampling to show the true
%behavior of the continous time conditions of our cell plant. And with are
%high and low water temperature baths it would be even harder to fix
%becuase finer adjustments are hard with that large of a temperature range.
%Further, the delay compounds the stability.
```






```
Dz =

      -z^4 + 1.435 z^3 - 0.6401 z^2 + 0.1108 z
  ---------------------------------------------------
  0.009627 z^4 + 0.002259 z^3 - 0.1274 z^2 + 0.1156 z
```

Extra Credit - ran out of time >:(

```
%% Extra Credit
clc;

syms y(t) u(t)
ode = diff(y,t) == -0.1*y(t) + 0.1*u(t - 6)

cond1 = y(0) == 25;
cond2 = u(t) == 50;

conds = [cond1, cond2]
ySol(t) = dsolve(ode == 35, conds)
```