```
% Name: Lamin Jammeh
% Class: EE480 Online
% Semster: Fall 2023
% HW_7

% Basic Problems
%% ********** 3.24a **********
clear;
clc;
% note H(s) = Y(s)/X(s) and X(s) = Y(s)/H(s)

syms t s; %Define the syms function interms of t and s
Y_s = 4 / (s*((s+1)^2 + 1)); %Define H(s) in laplace domain
h_t = exp(-t) * cos(t)*heaviside(t); %Define y(t) in time domain
H_s = laplace(h_t); %Define Y(s) in laplace transform
X_s = Y_s / H_s
x_t = ilaplace(X_s)

%% ********** 3.28a **********
clear;
clc;
% note the steady response y_ss(t)=limy(t) as t approaches infinity

syms t s; %define a syms function interms of t and s
H_s = (s^2 + 4) / (s*((s+1)^2 + 1)); %defines H(s)
num = [1,4]; %defines the numerator of the H(s)
denum = [1,2,2,2];
H=tf(num,denum);
pzmap(H);
title("Q3.28a pzmap of H(s)");
grid on;
% define x and y axis limits
xlim([-6,0]) % assign limits to y axis
ylim([-2,2]) % assign limits to y axis

% condition for BIBO stable
poles = pole(H); %use pole function to determine the poles values
disp(poles) %display poles values
if pole(H) < 0
    disp('System is BIBO stable')
else
    disp('System is not BIBO stable')
end

%% ********** 3.28a **********
clear;
clc;
% note the steady response y_ss(t)=limy(t) as t approaches infinity
%note y(t) = h(t)*x(t) where *=convolution
```

```
syms t s o f; %define a syms function interms of t, s and o

%Step1 Define the transfer function
H_s = (s^2 + 4) / (s*((s+1)^2 + 1)); %defines H(s)

%Step 2 define x(t)
x_t = 2*cos(o*t)*heaviside(t)
X_s = laplace(x_t) % define laplace of x(t)
Y_s = H_s * X_s % define out Y(s) in laplce domain
% note @ steady state Y(s)=0
Output_ss = Y_s ==0;
%Step3 solve for s and use f to determin omega
S=solve(Output_ss,s) %solve for s
%Note Omega = 2*pi*f
f = S/(i*2*pi)

%% ********** 3.42a **********
clear;
clc;

%use syms function interms of s
syms s;
P_s = s^2 + s + 1;
P = [1,1,1];
Q_s = 2*s^3 + 3*s + s + 1;
Q = [2,3,1,1];
Z_s = conv(P,Q)

%% ********** 3.42a **********
clear;
clc;

syms t s;
Y_s = (s+2) / s^2 * (s+1) * ((s+4)^2 + 9);
N = [1,1];
D = s^2 * (s+1) * ((s+4)^2 + 9);
%Split the D into 3 diff polynomials
D_1 = [1 0 0];              %ploynomial for S^2
D_2 = [1 1];                %ploynomial for S+1
D_3 = [1 8 25];             %ploynomial for (s+4)^2+9
D_4= conv(D_1,D_2,'full');   % multiple D1*D2 to D4
D_s = conv(D_3,D_4,'full')    % multiple D3*D to get final polynomial
d_t = ilaplace(D)
```

3.42

Saturday, October 14, 2023 12:58 PM

a) $P_{(s)} = s^2 + s + 1 \qquad Q_{(s)} = 2s^3 + 3s^2 + s + 1$

$Z_{(s)} = P_{(s)}\ Q_{(s)}$

$Z_{(s)} = [s^2 + s + 1] \times [2s^3 + 3s^2 + s + 1]$

$Z_{(s)} = [2s^5 + 3s^4 + s^3 + s^2] + [2s^4 + 3s^3 + s^2 + s] + [2s^3 + 3s^2 + s + 1]$

$Z_{(s)} = 2s^5 + 5s^4 + 6s^3 + 5s^2 + 2s + 1$

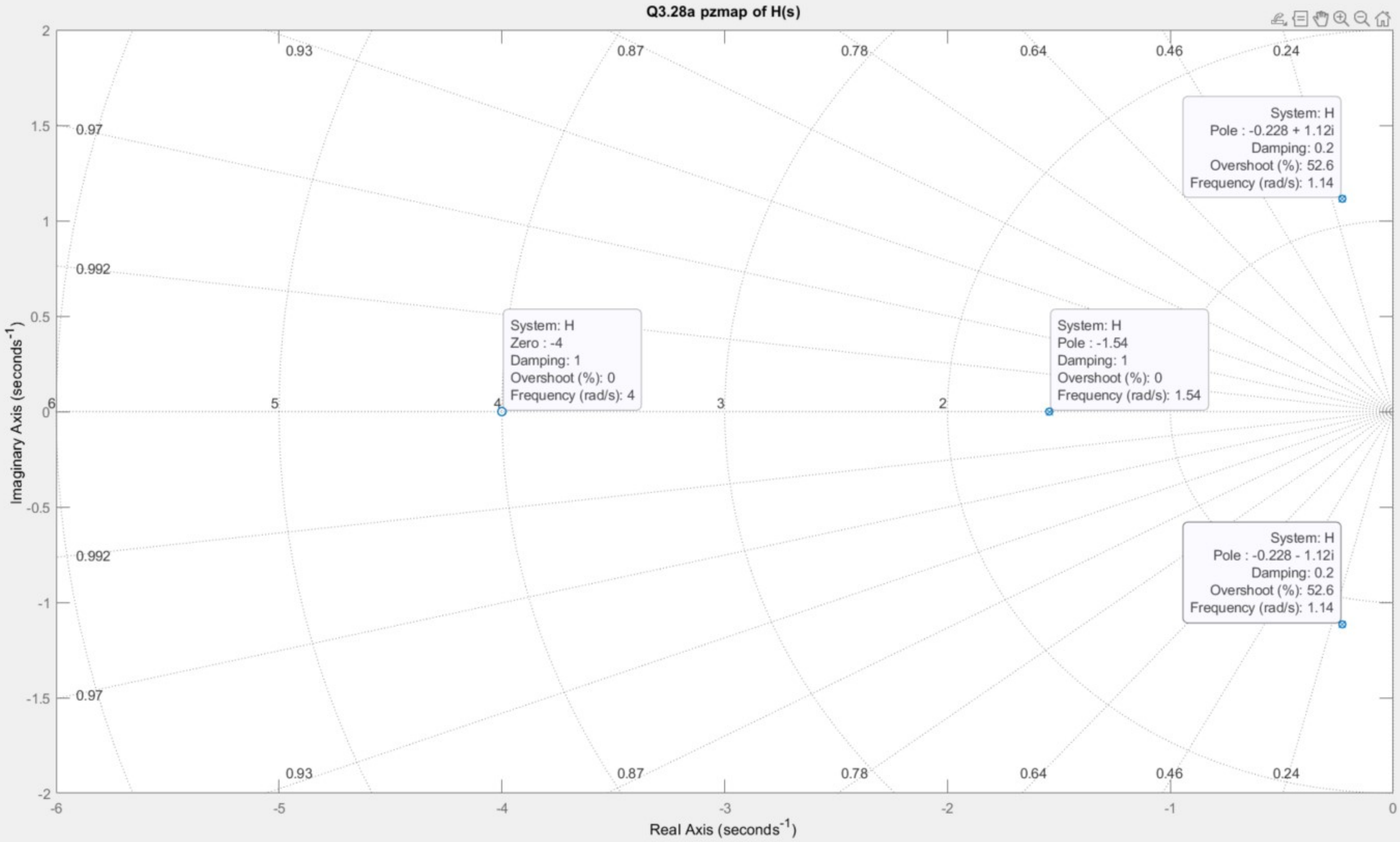
b) $Y_{(s)} = \dfrac{(s+1)}{s^2(s+1)((s+4)^2 + 9))}$

$D_{(s)} = s^2(s+1)\left((s+4)^2 + 9\right)$

$= (s^3 + s^2)(s^2 + 8s + 16 + 9)$

$= s^5 + 8s^4 + 25s^3 + s^4 + 8s^3 + 25s^2$

$= s^4 + 9s^4 + 33s^3 + 25s^2$



Q3.28a pzmap of H(s)
System: H
Pole : -0.228 + 1.12i
Damping: 0.2
Overshoot (%): 52.6
Frequency (rad/s): 1.14
System: H
Zero : -4
Damping: 1
Overshoot (%): 0
Frequency (rad/s): 4
System: H
Pole : -1.54
Damping: 1
Overshoot (%): 0
Frequency (rad/s): 1.54
System: H
Pole : -0.228 - 1.12i
Damping: 0.2
Overshoot (%): 52.6
Frequency (rad/s): 1.14
0.93
0.87
0.78
0.64
0.46
0.24
0.97
0.992
Imaginary Axis (seconds^{-1})
Real Axis (seconds^{-1})