Controls System Lab 4 Colin Roskos

Introduction:

Study of L.T.I. systems and their representation. We will continue the use of Matlab as a mathematical modelling software, and its use of representing pole-zero-gain representations. Along with analysis by plotting poles and zeros, time response, and arbitrary inputs.

Conclusion:

In this lab we learned and implemented the ability to: map poles and zeros using the transfer function – function – tf, we learned the use of impulse and step functions to simulate the step and impulse response of a given system equation, the use of variable substitution in the tf function and their implementation, and the comparison between analytical methods for solving ode systems and a s model solution.

Code:

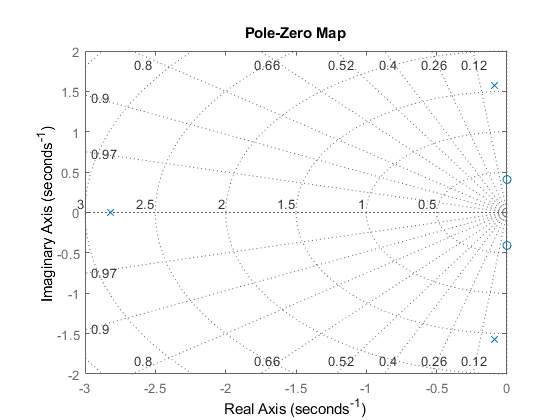
%% Exercise 1:

% G = ( 6s^2 + 1 ) / ( s^3 + 3s^2 + 3s + 7 )

G = tf([6 0 1], [1 3 3 7]);

pzmap(G); sgrid;

% 3 poles and 2 zeros, all are lhp.



%% Exercise 2:

% B/A = T

T1 = tf([1], [1 .2 1]);

T2 = tf([1 0], [1, .2 1]);

t = 0:0.01:10;

subplot(1,2,1);

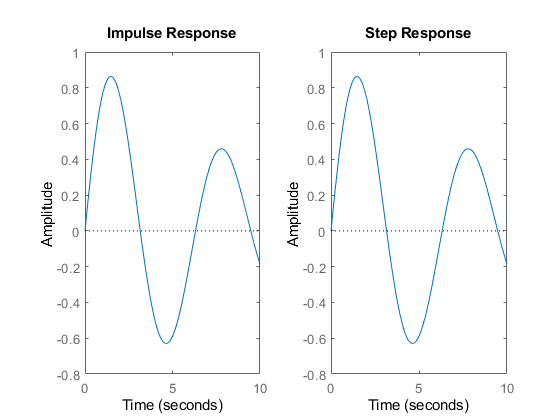
impulse(T1,t);

subplot(1,2,2);

step(T2,t);

% these are the same because step response is 1/s, impulse is 1

% T1\*H = 1/D \* 1 = 1/D , T2\*H = s/D \* 1/s = 1/D



%% Exercise 3:

% U = X/R

t = 0:0.01:10;

z = [3 6 12];

for i = 1:3

s = tf('s');

U = ( 15/ z(i) )\*( s + z(i) ) / ( s^2 + 3\*s + 15 );

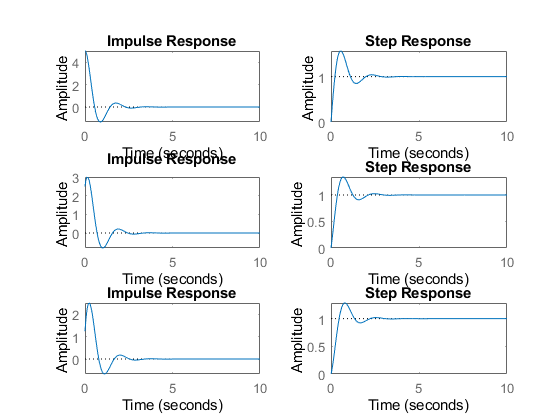
subplot(3, 2, 2\*i-1);

impulse(U, t);

subplot(3, 2, 2\*i);

step(U, t);

end%for



%% Exercise 4

% y'' + 4y' + 4y = u, y(0) = y'(0) = 0

% determine analytically and with the step function

t = 0:0.01:10;

% analytical solution

[t, y] = ode45(@fun, t, [0 0]);

subplot(2,1,1);

plot(t, y);

Y2 = tf([1], [1 4 4]);

subplot(2,1,2);

step(t, Y2);

