Lab 4: Computational Tools for Transfer Functions and Simulations

Ryan St.Pierre, Doyong Kim, Rohan Rajan

October 11, 2017

1 Experiment 2.1

Reference the Matlab code and diary below for relevant information.

Matlab script: Experiment21.m

```
% Ryan St Pierre (ras70)
% Lab 4
% Experiment 2.1, Nise 7e
% Honor code statement: ras70
clear; format short e; format compact;
delete('diaryExperiment2_1');
echo on; diary diaryExperiment2_1
%% Prelab
%% Prelab-a
P1 = [1 7 2 9 10 12 15];
r1 = roots(P1)

%% Prelab-b
P2 = [1 9 8 9 12 15 20];
r2 = roots(P2)

%% Lab
%% Problem 1

P3 = P1 + P2;
syms s;
P3Display = poly2sym(P3,s)

P4 = P1 - P2;
P4Display = poly2sym(P4,s)
```

```
P5 = conv(P1, P2)
P5Display = poly2sym(P5,s)

%% Problem 2
P6 = conv(conv(conv([1 7], [1 8]),conv([1 3],[1 5])), conv([1 9],[1,10]));
P6Display = poly2sym(P6,s)

%% Problem 3
% We can generate G1 two different ways - both using a maximum of two
% commands. Both ways are shown below

% Define s for future use if we want it
s = tf ('s');

% Way 1 - Using conv generate G1 in one line
G1 = tf(20 * conv(conv([1 2],[1 3]), conv([1 6],[1 8])),...
conv([1,0],conv(conv([1 7],[1 9]), conv([1 10],[1 15]))))

% Way 2 - using 's' to generate G1
G1prime = 20*((s+2)*(s+3)*(s+6)*(s+8))/(s*(s+7)*(s+9)*(s+10)*(s+15));
G1PrimeTF = tf(G1prime)
%% Problem 4
% We can generate G2 two different ways - both using a maximum of two
% commands. Both ways are shown below

% Way 1
G2 = zpk(roots([1 17 99 223 140]), roots([1 32 363 2092 5052 4320]), 1)

% Way 2
G2prime = (s^4+17*s^3+99*s^2+223*s+140)/(s^5+32*s^4+363*s^3+2092*s^2+5052*s+4320);
G2PrimeTF = zpk(G2prime)

%% Problem 5
% G3
G3 = G1 + G2
G3TF = tf(G3)
G3zpk = zpk(G3)

% G4
G4 = G1 - G2
G4TF = tf(G4)
G4zpk = zpk(G4)

% G5
G5 = G1 * G2
G5TF = tf(G5)
G5zpk = zpk(G5)

%% Problem 6
% Problem 6a
```

```
G6_num = 5 * [1 2];
G6_den = conv([1 0],[1 8 15]);
[r6,p6,k6] = residue(G6_num,G6_den)

% Problem 6b
G7_num = G6_num;
G7_den = conv([1 0],[1 6 9]);
[r7,p7,k7] = residue(G7_num,G7_den)

% Problem 6c
G8_num = G6_num;
G8_den = conv([1 0],[1 6 34]);
[r8,p8,k8] = residue(G8_num,G8_den)

diary off; echo off;
```

Diary: diaryExperiment2_1

```
%% Prelab
%% Prelab-a
P1 = [1 7 2 9 10 12 15];
r1 = roots(P1)
r1 =
  -6.8731e+00 + 0.0000e+00i
   7.6317e-01 + 1.0822e+00i
   7.6317e-01 - 1.0822e+00i
  -1.0000e+00 + 0.0000e+00i
  -3.2661e-01 + 1.0667e+00i
  -3.2661e-01 - 1.0667e+00i

%% Prelab-b
P2 = [1 9 8 9 12 15 20];
r2 = roots(P2)
r2 =
  -8.1330e+00 + 0.0000e+00i
   6.9962e-01 + 9.8801e-01i
   6.9962e-01 - 9.8801e-01i
  -1.2183e+00 + 0.0000e+00i
  -5.2398e-01 + 1.0501e+00i
  -5.2398e-01 - 1.0501e+00i

%% Lab
%% Problem 1

P3 = P1 + P2;
syms s;
P3Display = poly2sym(P3,s)
P3Display =
2*s^6 + 16*s^5 + 10*s^4 + 18*s^3 + 22*s^2 + 27*s + 35

P4 = P1 - P2;
P4Display = poly2sym(P4,s)
P4Display =
- 2*s^5 - 6*s^4 - 2*s^2 - 3*s - 5

P5 = conv(P1, P2)
P5 =
     1    16    73    92   182   291   433   599   523   609   560   465   300
P5Display = poly2sym(P5,s)
P5Display =
s^12 + 16*s^11 + 73*s^10 + 92*s^9 + 182*s^8 + 291*s^7 + 433*s^6 + 599*s^5 +
    523*s^4 + 609*s^3 + 560*s^2 + 465*s + 300

%% Problem 2
P6 = conv(conv(conv([1 7], [1 8]),conv([1 3],[1 5])), conv([1 9],[1,10]));
P6Display = poly2sym(P6,s)
P6Display =
```

```
s^6 + 42*s^5 + 718*s^4 + 6372*s^3 + 30817*s^2 + 76530*s + 75600

%% Problem 3
% We can generate G1 two different ways - both using a maximum of two
% commands. Both ways are shown below

% Define s for future use if we want it
s = tf ('s');

% Way 1 - Using conv generate G1 in one line
G1 = tf(20 * conv(conv([1 2],[1 3]), conv([1 6],[1 8])),...
conv([1,0],conv(conv([1 7],[1 9]), conv([1 10],[1 15]))))

G1 =

  20 s^4 + 380 s^3 + 2480 s^2 + 6480 s + 5760
  -------------------------------------------
  s^5 + 41 s^4 + 613 s^3 + 3975 s^2 + 9450 s

Continuous-time transfer function.


% Way 2 - using 's' to generate G1
G1prime = 20*((s+2)*(s+3)*(s+6)*(s+8))/(s*(s+7)*(s+9)*(s+10)*(s+15));
G1PrimeTF = tf(G1prime)

G1PrimeTF =

  20 s^4 + 380 s^3 + 2480 s^2 + 6480 s + 5760
  -------------------------------------------
  s^5 + 41 s^4 + 613 s^3 + 3975 s^2 + 9450 s

Continuous-time transfer function.

%% Problem 4
% We can generate G2 two different ways - both using a maximum of two
% commands. Both ways are shown below

% Way 1
G2 = zpk(roots([1 17 99 223 140]), roots([1 32 363 2092 5052 4320]), 1)

G2 =

                 (s+7) (s+5) (s+4) (s+1)
  -----------------------------------------------------
  (s+16.79) (s^2 + 4.097s + 4.468) (s^2 + 11.12s + 57.6)

Continuous-time zero/pole/gain model.


% Way 2
```

```
G2prime = (s^4+17*s^3+99*s^2+223*s+140)/(s^5+32*s^4+363*s^3+2092*s^2+5052*s+4320);
G2PrimeTF = zpk(G2prime)

G2PrimeTF =

                 (s+7) (s+5) (s+4) (s+1)
  ---------------------------------------------------
  (s+16.79) (s^2 + 4.097s + 4.468) (s^2 + 11.12s + 57.6)

Continuous-time zero/pole/gain model.


%% Problem 5
% G3
G3 = G1 + G2

G3 =


  21 (s+16.76) (s+7.999) (s+6.003) (s^2 + 6.079s + 9.499) (s^2 + 3.033s + 2.607)
       (s^2 + 11.46s + 59.47)


  ------------------------------------------------------------------------------

           s (s+15) (s+16.79) (s+10) (s+9) (s+7) (s^2 + 4.097s + 4.468)
                (s^2 + 11.12s + 57.6)


Continuous-time zero/pole/gain model.

G3TF = tf(G3)

G3TF =


  21 s^9 + 1078 s^8 + 2.331e04 s^7 + 2.843e05 s^6 + 2.156e06 s^5 + 1.043e07 s^4
               + 3.173e07 s^3 + 5.816e07 s^2 + 5.842e07 s + 2.488e07

  ------------------------------------------------------------------------------

  s^10 + 73 s^9 + 2288 s^8 + 4.057e04 s^7 + 4.5e05 s^6 + 3.239e06 s^5
               + 1.502e07 s^4 + 4.25e07 s^3 + 6.491e07 s^2 + 4.082e07 s


Continuous-time transfer function.

G3zpk = zpk(G3)

G3zpk =
```

```
  21 (s+16.76) (s+7.999) (s+6.003) (s^2 + 6.079s + 9.499) (s^2 + 3.033s + 2.607)
             (s^2 + 11.46s + 59.47)


  ---------------------------------------------------------------------------

           s (s+15) (s+16.79) (s+10) (s+9) (s+7) (s^2 + 4.097s + 4.468)
                    (s^2 + 11.12s + 57.6)


Continuous-time zero/pole/gain model.


% G4
G4 = G1 - G2

G4 =

  19 (s+16.81) (s+8.001) (s+5.997) (s+3.252) (s+1.496) (s^2 + 4.335s + 6.018)
                (s^2 + 10.74s + 55.47)
  ---------------------------------------------------------------------------
          s (s+15) (s+16.79) (s+10) (s+9) (s+7) (s^2 + 4.097s + 4.468)
                (s^2 + 11.12s + 57.6)

Continuous-time zero/pole/gain model.

G4TF = tf(G4)

G4TF =


  19 s^9 + 962 s^8 + 2.049e04 s^7 + 2.469e05 s^6 + 1.862e06 s^5 + 9.034e06 s^4
                   + 2.791e07 s^3 + 5.284e07 s^2 + 5.577e07 s + 2.488e07

  ---------------------------------------------------------------------------

  s^10 + 73 s^9 + 2288 s^8 + 4.057e04 s^7 + 4.5e05 s^6 + 3.239e06 s^5 +
                   + 1.502e07 s^4 + 4.25e07 s^3 + 6.491e07 s^2 + 4.082e07 s


Continuous-time transfer function.

G4zpk = zpk(G4)

G4zpk =

  19 (s+16.81) (s+8.001) (s+5.997) (s+3.252) (s+1.496) (s^2 + 4.335s + 6.018)
                   (s^2 + 10.74s + 55.47)
  ---------------------------------------------------------------------------
          s (s+15) (s+16.79) (s+10) (s+9) (s+7) (s^2 + 4.097s + 4.468)
```

```
                    (s^2 + 11.12s + 57.6)

Continuous-time zero/pole/gain model.


% G5
G5 = G1 * G2

G5 =

                  20 (s+8) (s+7) (s+6) (s+5) (s+4) (s+3) (s+2) (s+1)
  ---------------------------------------------------------------------------------
  s (s+15) (s+16.79) (s+10) (s+9) (s+7) (s^2 + 4.097s + 4.468) (s^2 + 11.12s + 57.6)

Continuous-time zero/pole/gain model.

G5TF = tf(G5)

G5TF =


  20 s^8 + 720 s^7 + 1.092e04 s^6 + 9.072e04 s^5 + 4.49e05 s^4 + 1.346e06 s^3
                   + 2.362e06 s^2 + 2.192e06 s + 8.064e05

  ------------------------------------------------------------------------------

  s^10 + 73 s^9 + 2288 s^8 + 4.057e04 s^7 + 4.5e05 s^6 + 3.239e06 s^5
                   + 1.502e07 s^4 + 4.25e07 s^3 + 6.491e07 s^2 + 4.082e07 s


Continuous-time transfer function.

G5zpk = zpk(G5)

G5zpk =

                  20 (s+8) (s+7) (s+6) (s+5) (s+4) (s+3) (s+2) (s+1)
  ---------------------------------------------------------------------------------
  s (s+15) (s+16.79) (s+10) (s+9) (s+7) (s^2 + 4.097s + 4.468) (s^2 + 11.12s + 57.6)

Continuous-time zero/pole/gain model.


%% Problem 6
% Problem 6a
G6_num = 5 * [1 2];
G6_den = conv([1 0],[1 8 15]);
[r6,p6,k6] = residue(G6_num,G6_den)
r6 =
  -1.5000e+00
   8.3333e-01
```

```
   6.6667e-01
p6 =
    -5
    -3
     0
k6 =
     []

% Problem 6b
G7_num = G6_num;
G7_den = conv([1 0],[1 6 9]);
[r7,p7,k7] = residue(G7_num,G7_den)
r7 =
  -1.1111e+00
   1.6667e+00
   1.1111e+00
p7 =
    -3
    -3
     0
k7 =
     []

% Problem 6c
G8_num = G6_num;
G8_den = conv([1 0],[1 6 34]);
[r8,p8,k8] = residue(G8_num,G8_den)
r8 =
  -1.4706e-01 - 4.1176e-01i
  -1.4706e-01 + 4.1176e-01i
   2.9412e-01 + 0.0000e+00i
p8 =
  -3.0000e+00 + 5.0000e+00i
  -3.0000e+00 - 5.0000e+00i
   0.0000e+00 + 0.0000e+00i
k8 =
     []

diary off; echo off;
```

2 Experiment 2.2

(See attached Maple worksheet and circuit diagram.)

3 Mechanical System Analysis

Figures one, two, and three all show the system response from a specific input from the transfer function

$$H(s) = \frac{f_v s + k}{M_1 s^2 + f_v s + k} \tag{1}$$

where f_v is the total damping between mass 1 and mass 2, k is the spring constant, and M is the mass in kg.

As seen in Fig. 1, the system response for the under damped, critically damped, and over damped scenarios follow a standard 2nd order system behavior. The undamped system response in particular is oscillating from 0 to 2 in amplitude for an infinite amount of time. All system responses overshoot their final value. The amount of this overshoot for each system response in increasing order are: overdamped, critically damped, underdamped, undamped. This percent overshoot corresponds directly to the peak time. The responses with larger overshoots peak at a higher value, but take longer to get to these peaks. Said differently the responses with lower overshoots rise quicker, but peak at lower values. Finally, all the responses take a different amount of time to settle - overdamped takes the quickest, then critically damped, then underdamped. As prior mentioned the undamped response never settles, but rather oscillates.

In Fig. 2 however, the undamped system response is increasing in amplitude as the time increases. With an input of $sin(t)$, the system is being excited at its natural frequency thus resulting in an undamped response that is exhibits increased oscillations over time. Vibrating undamped systems to their natural frequency can be dangerous since the system response will tend to increase until the system fails due to an inevitable mechanical failure. In other words, the an undamped system is not stable given an input at its natural frequency. This means the system is not BIBO stable, since its output is not bounded for the input at its natural frequency (which is bounded). However, the system is stable with regards to the definition that says a system is stable if its natural response decays to zero or oscillates as time goes to infinity.

Fig. 3 shows that when the system is excited close to, but not quite, the natural frequency, the undamped response will again increase and oscillate over time, however, Fig. 3 clearly shows that the system response will diminish over time as well. At around 30 seconds is when the undamped system response in Fig. 3 started to diminish. Vibrating systems close to its natural frequency will not make the system response increase forever, that being said, Fig. 3 shows that the mechanical build of

the system itself should be checked to make sure that it can withhold the maximum system response when vibrating close to its natural frequency.

Matlab script:4.10 Assignment Part 2

```
%% Doyong Kim
%% ME344 lab 4
%% Experiment 3 part 4.10
%% calling anonymous function and other variables
clear
clc
s = tf('s');
G = @(M, fv, K) (fv.*s+K)/(M.*(s^2)+fv.*s+K);
t = linspace(0,10,1000);
u = transpose(linspace(1,1,1000));
%% undamped simulation fv = 0
G1 = G(1,0,1)
a = lsim(G1,u,t);

%% underdamped simulation fv = 1
G2 = G(1,1,1)
b = lsim(G2,u,t);

%% Critically damped simulation fv = 2
G3 = G(1,2,1)
c = lsim(G3,u,t);

%% Overdamped simulation fv = 4
G4 = G(1,4,1)
d = lsim(G4,u,t);

%% Plots
figure(1)
plot(t,linspace(1,1,1000),t,a,t,b,t,c,t,d)
legend('Location','southeast','X2 position','Undamped','Underdamped',...
    'Critically damped','Overdamped')
xlabel('Time in Seconds')
ylabel('System Response')
title('System Response vs Time with System Input 1')
savefig('plot1.fig')
%% with input = sin(t) and time of 50 seconds
tt = linspace(1,50,1000);
uu = sin(tt);
% for undamped fv = 0
GG1 = G(1,0,1)
aa = lsim(GG1,uu,tt);
% for underdamped fv = 1
GG2 = G(1,1,1)
bb = lsim(GG2,uu,tt);
% for Critically damped fv = 2
GG3 = G(1,2,1)
```

```
cc = lsim(GG3,uu,tt);
% for Overdamped fv = 4
GG4 = G(1,4,1)
dd = lsim(GG4,uu,tt);
%% Plots
figure(2)
plot(tt,sin(tt),tt,aa,tt,bb,tt,cc,tt,dd)
legend('Location','northwest','X2 position','Undamped','Underdamped',...
    'Critically damped','Overdamped')
ylabel('System Response')
xlabel('Time in Seconds')
title('System Response vs Time with System Input Sin(t)')
savefig('plot2.fig')
%% with input = sin(0.9*t) and time of 50 seconds
ttt = linspace(1,50,1000);
uuu = sin(0.9.*ttt);
% for undamped fv = 0
GGG1 = G(1,0,1)
aaa = lsim(GGG1,uuu,ttt);
% for underdamped fv = 1
GGG2 = G(1,1,1)
bbb = lsim(GGG2,uuu,ttt);
% for Critically damped fv = 2
GGG3 = G(1,2,1)
ccc = lsim(GGG3,uuu,ttt);
% for Overdamped fv = 4
GGG4 = G(1,4,1)
ddd = lsim(GGG4,uuu,ttt);
%% Plots
figure(3)
plot(ttt,uuu,ttt,aaa,ttt,bbb,ttt,ccc,ttt,ddd)
legend('Location','northwest','X2 position','Undamped','Underdamped',...
    'Critically damped','Overdamped')
ylabel('System Response')
xlabel('Time in Seconds')
title('System Response vs Time with System Input Sin(0.9t)')
savefig('plot3.fig')
```

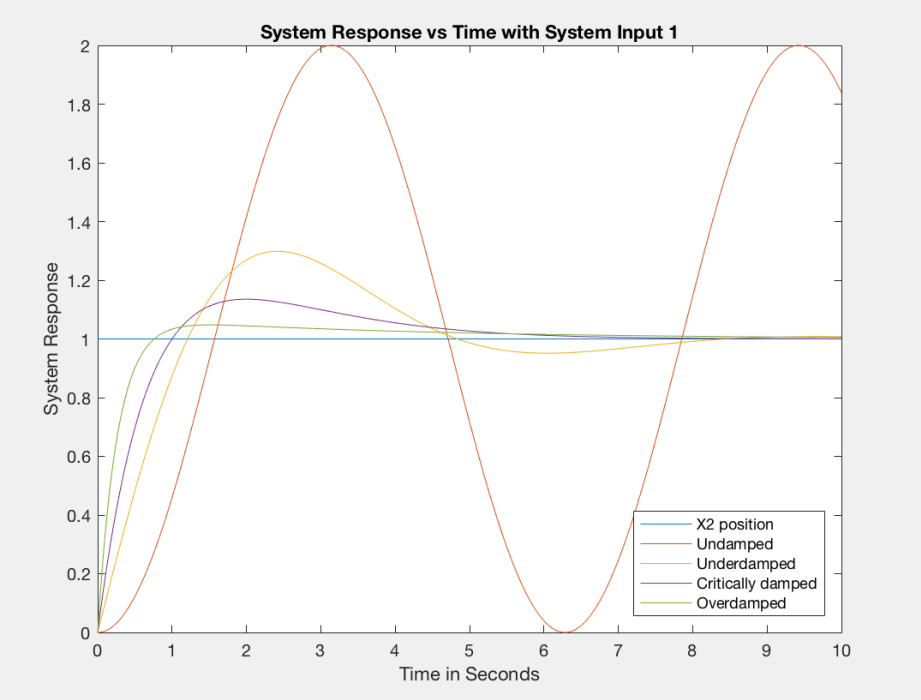

Figure 1: Step response for undamped $\zeta = 0$, underdamped $\zeta < 1$, critically damped$\zeta = 1$, and overdamped systems $\zeta > 0$

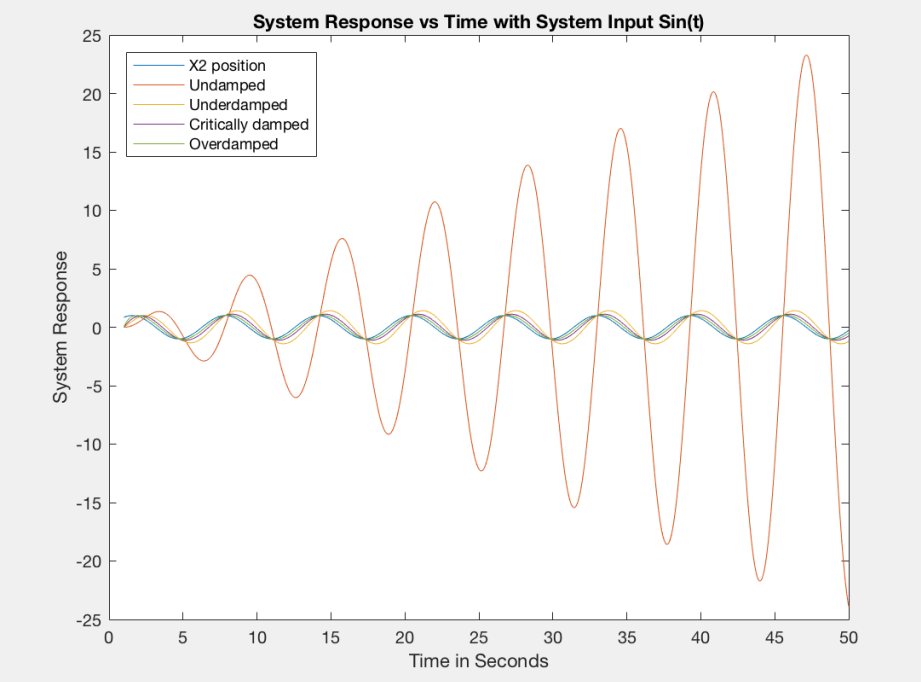

Figure 2: Response for $r(t) = \sin(t)$ for undamped $\zeta = 0$, underdamped $\zeta < 1$, critically damped$\zeta = 1$, and overdamped systems $\zeta > 0$

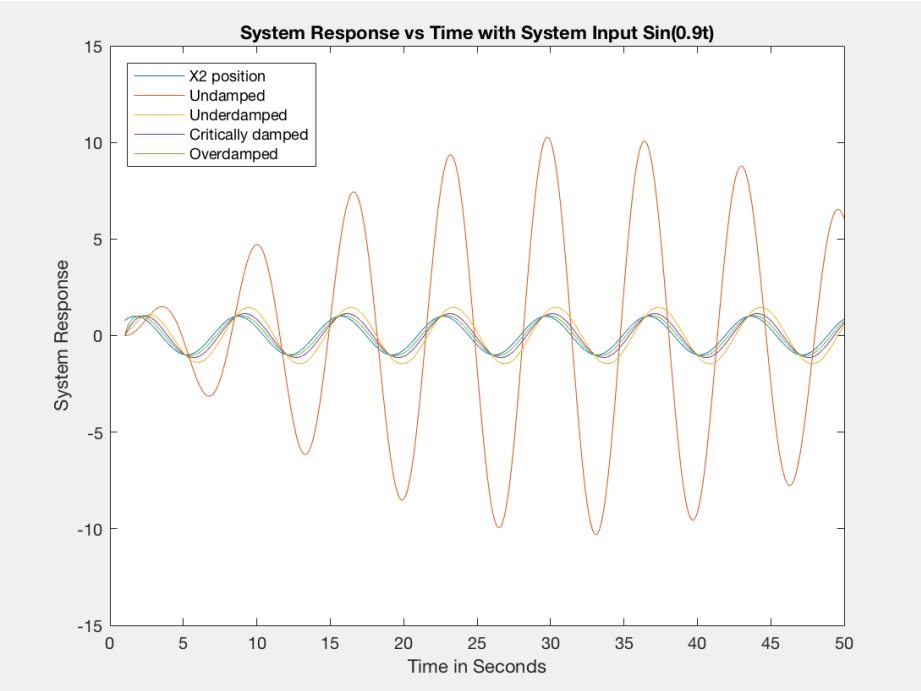

Figure 3: Response for $r(t) = \sin(0.9t)$ for undamped $\zeta = 0$, underdamped $\zeta < 1$, critically damped$\zeta = 1$, and overdamped systems $\zeta > 0$