MAPUA UNIVERSITY
School of Electrical, Electronics, and Computer Engineering
A.Y 2018-2019

MODULE 4: STEP RESPONSE OF SECOND ORDER SYSTEMS

Score

Submitted by:

Tribiana, Aivje C.
2012101918
ECE131L/E02

Engr. Charmaine Paglinawan
Instructor

PROCEDURES:

1. For each of the following transfer functions, find the damping ratios and natural frequencies.

a. $G(s) = \frac{100}{s^2-22s+100}$ b. $G(s) = \frac{100}{s^2-20s+100}$ c. $G(s) = \frac{100}{s^2-10s+100}$

d. $G(s) = \frac{100}{s^2+100}$ e. $G(s) = \frac{100}{s^2+10s+100}$ f. $G(s) = \frac{100}{s^2+20s+100}$

g. $G(s) = \frac{100}{s^2+22s+100}$

System	Natural Frequency	Damping Ratio
a	6.4174 15.5826	-1 -1
b	10.0000 10.0000	-1 -1
c	10 10	-0.5000 -0.5000
d	10 10	0 0
e	10 10	0.5000 0.5000
f	10.0000 10.0000	1 1
g	6.4174 15.5826	1 1

2. Using different colors for each of the system in (1), use ltiviewer or **pzmap** command to view(plot) their pole-zero maps.

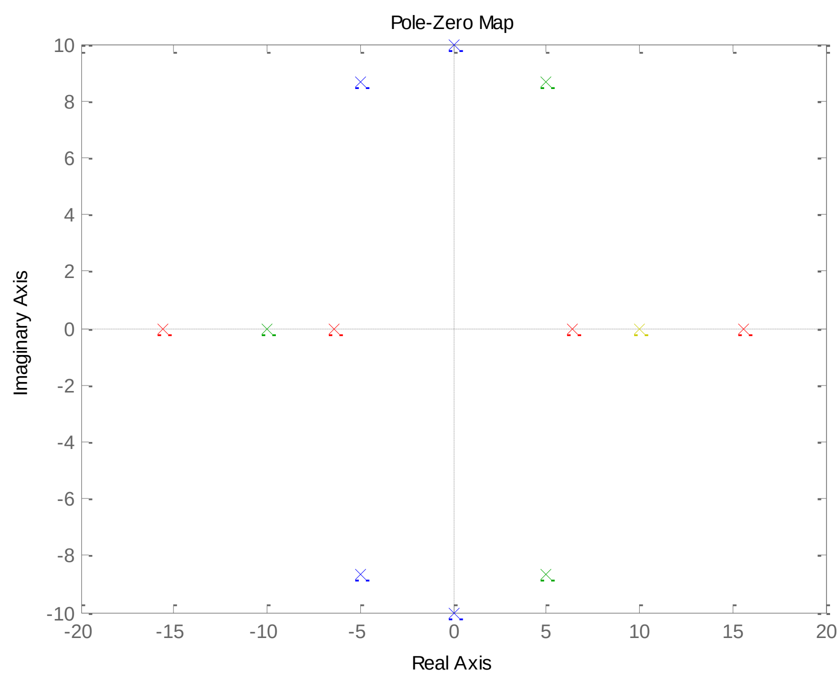


Figure 4.1 Pole-Zero Map of the Systems in (1)

3. Using **step** command, plot the step responses of the systems given in (1)

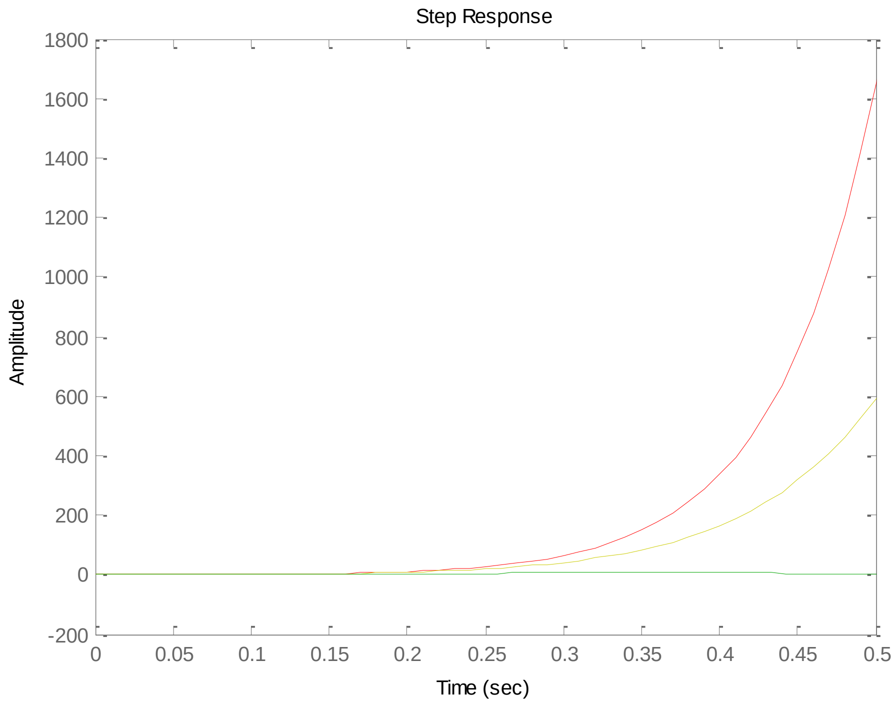


Figure 4.2a Step Responses of the Systems $G_{a(s)}$, $G_{b(s)}$, $G_{c(s)}$, and $G_{d(s)}$

For systems (d), (e), (f), (g).

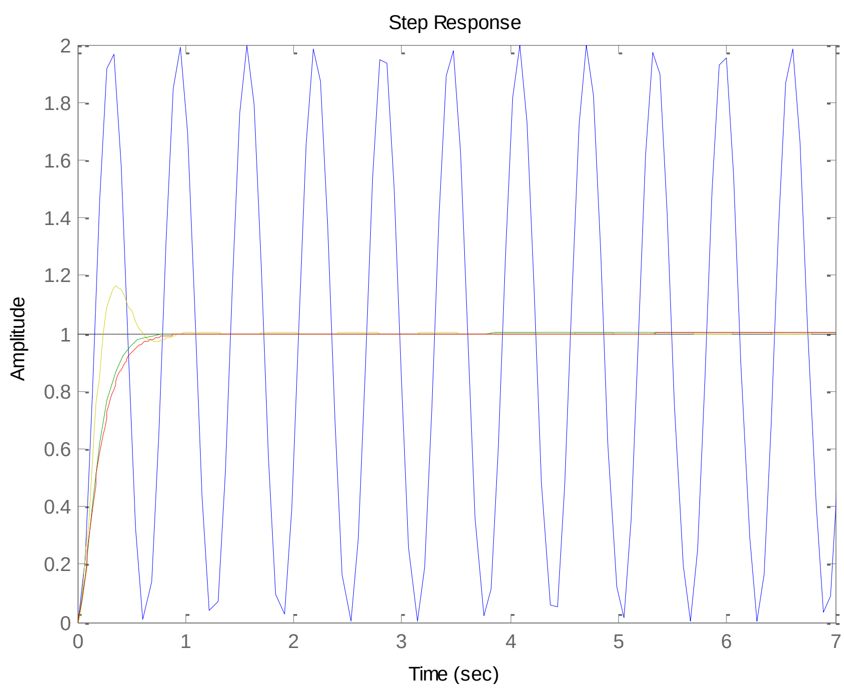
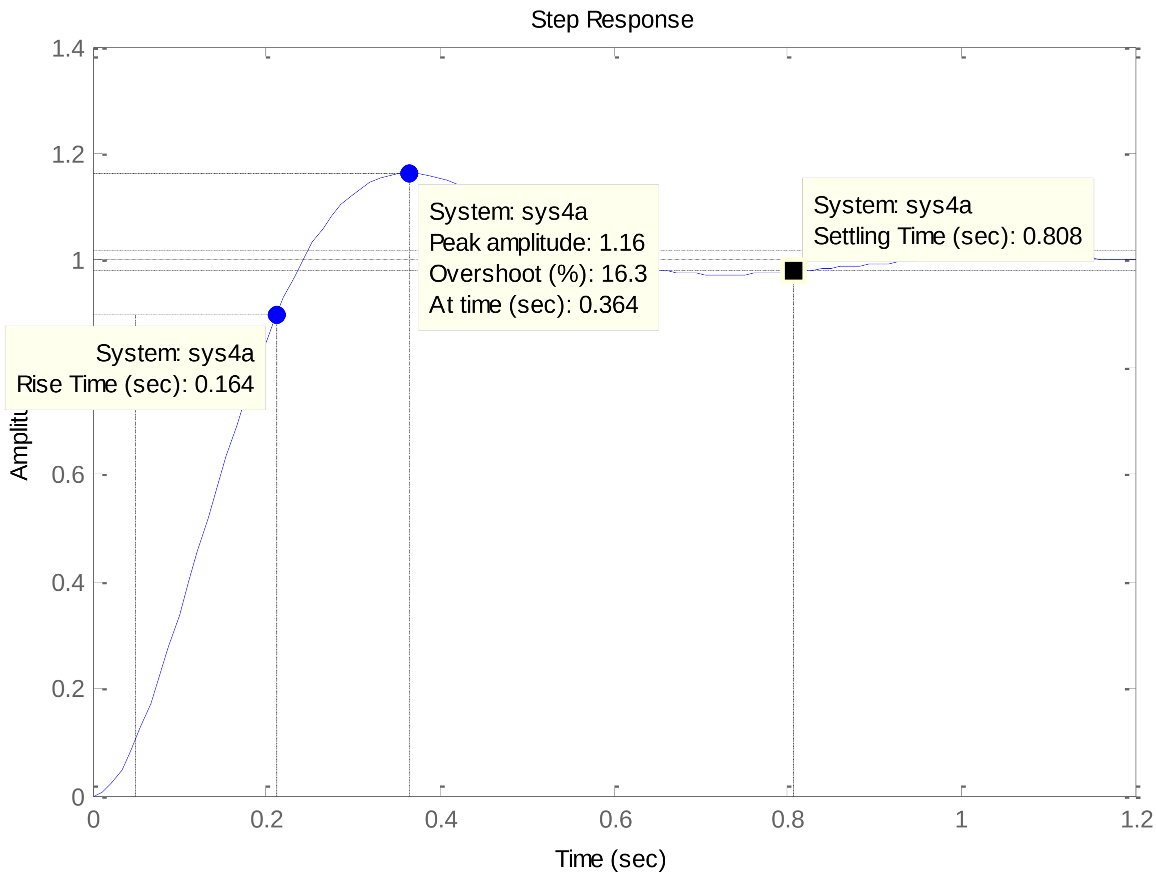


Figure 4.2b Step Responses of the Systems $G_{a(s)}$, $G_{b(s)}$, $G_{c(s)}$, and $G_{d(s)}$

4. Using the **ltieview** command, determine the peak time, percent overshoot, settling time and rise time of $G(s) = \frac{100}{s^2+10s+100}$ by right-clicking the mouse anywhere in the plot and selecting the characteristics.

% Overshoot	Peak Time (sec)	Rise Time (sec)	Settling Time (sec)
16.3	0.364	0.164	0.808



5. Create a MATLAB m-file secondordersys.m that will accept ç and ω_n plot the pole-zero map and step response showing %OS, rise time, settling time, and peak time.

MATLAB m-file:

```
function[sys]=secondordersystem
W=input('W= ')
Z=input('Z= ')
sys=tf(W^2,[1 2*W*Z W^2])
figure(1);pzmap(sys);
figure(2);step(sys);
ltiview(sys)
end
```

6. Test your m-file using the following values for ç and ω_n.

ω_n	ç	% Overshoot	T_{peak}	T_{rise} (sec)	$T_{settling}$ (sec)
10	**0.1**	72.9	0.32	0.111	3.84
10	**0.2**	52.7	0.32	0.121	1.96
10	**0.3**	37.2	0.331	0.133	1.12
10	**0.4**	25.4	0.345	0.147	0.841
10	**0.5**	16.3	0.364	0.164	0.808
10	**0.6**	9.47	0.396	0.186	0.594
10	**0.7**	4.6	0.442	0.213	0.598
10	**0.8**	1.52	0.525	0.247	0.376
10	**0.9**	0.152	>0.7	0.288	0.47
1	**0.1**	72.9	3.2	1.11	38.4
2	**0.1**	72.9	1.6	0.556	19.2
3	**0.1**	72.9	1.07	0.371	12.8
4	**0.1**	72.9	0.8	0.278	9.59
5	**0.1**	72.9	0.64	0.222	7.68
6	**0.1**	72.9	0.533	0.185	6.4
7	**0.1**	72.9	0.457	0.159	5.48
8	**0.1**	72.9	0.4	0.139	4.8
9	**0.1**	72.9	0.356	0.124	4.26
10	**0.1**	72.9	0.32	0.111	3.84

SEATWORK:

Create a MATLAB m-file "underdampedsys.m" that will accept the parameters σ and ω_d, for a 2nd order underdamped system.The program will output the pzmap and the step response.

Test the m-file using:

Damping Constant, σ	**Damped Frequency, ω_d**
-1	1
-1	25
-1	5
-1	75
-1	10
-2.5	1
-5	1
-7.5	1
-10	1
-2.5	2.5
-5	5
-7.5	7.5
10	10

SYNTAX:

1. a.

```
>> sysa = tf(100, [1 -22 100])

Transfer function:
      100
----------------
s^2 - 22 s + 100

>> [Wn, Z] = damp(sysa)

Wn =

    6.4174
   15.5826


Z =

    -1
    -1
```

b.

```
>> sysb = tf(100, [1 -20 100])

Transfer function:
      100
----------------
s^2 - 20 s + 100

>> [Wn, Z] = damp(sysb)

Wn =

   10.0000
   10.0000


Z =

    -1
    -1
```

c.

```
>> sysc = tf(100, [1 -10 100])

Transfer function:
      100
----------------
s^2 - 10 s + 100

>> [Wn, Z] = damp(sysc)

Wn =

    10
    10


Z =

   -0.5000
   -0.5000
```

d.

```
>> sysd = tf(100, [1 0 100])

Transfer function:
   100
---------
s^2 + 100

>> [Wn, Z] = damp(sysd)

Wn =

    10
    10


Z =

     0
     0
```

e.

```
>> syse = tf(100, [1 10 100])

Transfer function:
      100
----------------
s^2 + 10 s + 100

>> [Wn, Z] = damp(syse)
```

```
Wn =

    10
    10


Z =

    0.5000
    0.5000
```

f.

```
>> sysf = tf(100, [1 20 100])

Transfer function:
      100
----------------
s^2 + 20 s + 100

>> [Wn, Z] = damp(sysf)

Wn =

   10.0000
   10.0000


Z =

     1
     1
```

g.

```
>> sysg = tf(100, [1 22 100])

Transfer function:
      100
----------------
s^2 + 22 s + 100

>> [Wn, Z] = damp(sysg)

Wn =

    6.4174
   15.5826

Z =

     1
     1
```

2.

```
>>
pzmap(sysa,'r',sysb,'y',sysc,'g',sysd,'b',syse,'o',
sysf,'p',sysg,'v')
```

3.

```
>> step(sysa,'r')
>> hold on
>> step(sysb,'y')
>> step(sysc,'g')
```

4.

```
>> sys4a = tf(100, [1 10 100])

Transfer function:
      100
----------------
s^2 + 10 s + 100

>> syms s
>> ltiview(sys4a)
```

5.

```
function[sys]=secondordersystem
W=input('W= ')
Z=input('Z= ')
sys=tf(W^2,[1 2*W*Z W^2])
figure(1);pzmap(sys);
figure(2);step(sys);
ltiview(sys)
end
```