DC Assignment

AV222-Instrumentation and Measurement

Submitted by:

Saurabh Kumar

SC22B146

DC-1

Aim: To plot the Unit Step Response of a Second Order Overdamped Instrument

Code:

```
% Variables
k = 1;
wn = 100;

% Plot for zeta = 1.5
zeta = 1.5; num = k*wn*wn; den = [1 2*zeta*wn wn*wn]; sys = tf(num, den);
stepplot(sys)
hold on

% Plot for zeta = 2
zeta = 2; num = k*wn*wn; den = [1 2*zeta*wn wn*wn]; sys = tf(num, den);
stepplot(sys)

% Plot for zeta = 2.5
zeta = 2.5; num = k*wn*wn; den = [1 2*zeta*wn wn*wn];
sys = tf(num, den);
stepplot(sys)

% Plot for zeta = 3
zeta = 3; num = k*wn*wn; den = [1 2*zeta*wn wn*wn];
sys = tf(num, den);
stepplot(sys)

title('Unit Step Response of Second Order Overdamped Instrument');
legend('zeta = 1.5', 'zeta = 2', 'zeta = 2.5', 'zeta = 3');

hold off
```

Inferences: As zeta increases, the speed of system decreases, thus taking longer time to reach the saturation value.

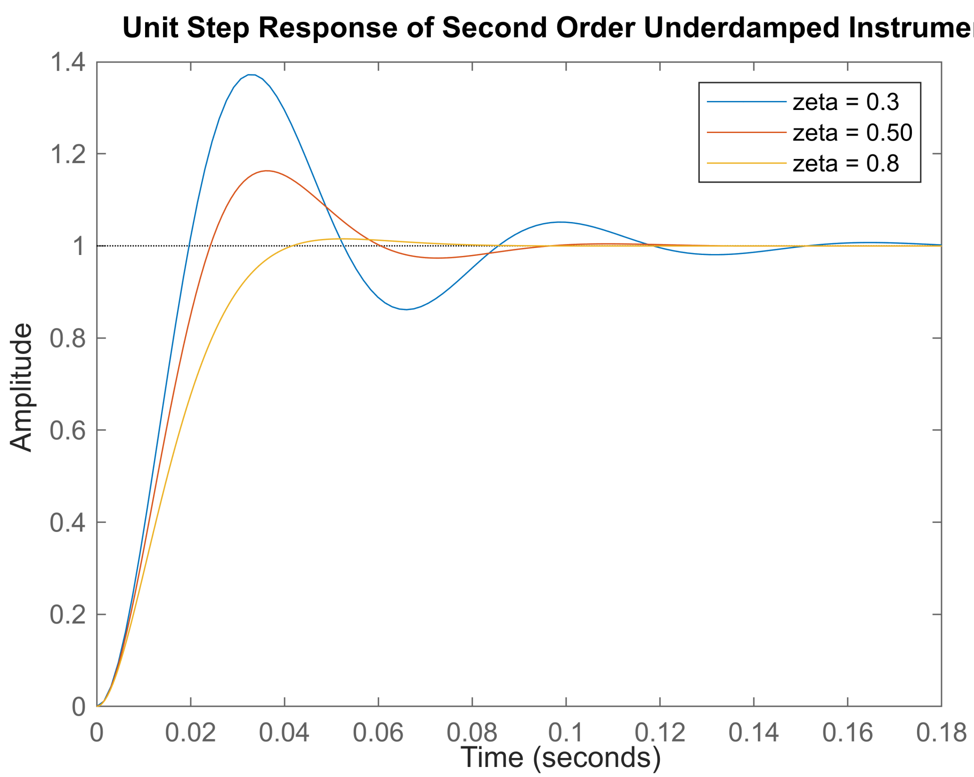
DC-2

Aim: To plot the Unit Step Response of a Second Order Underdamped Instrument

Code:

```
% Variables
A = 1;
k = 1;
wn = 100;

% Plot for zeta = 0.3
zeta = 0.3; num = k*wn*wn; den = [1 2*zeta*wn wn*wn]; sys = tf(num, den);
stepplot(sys)
hold on

% Plot for zeta = 0.5
zeta = 0.5; num = k*wn*wn; den = [1 2*zeta*wn wn*wn]; sys = tf(num, den);
stepplot(sys)

% Plot for zeta = 0.8
zeta = 0.8; num = k*wn*wn; den = [1 2*zeta*wn wn*wn];
```

```
sys = tf(num, den);
stepplot(sys)

title('Unit Step Response of Second Order Underdamped Instrument');
legend('zeta = 0.3', 'zeta = 0.50', 'zeta = 0.8');

hold off
```



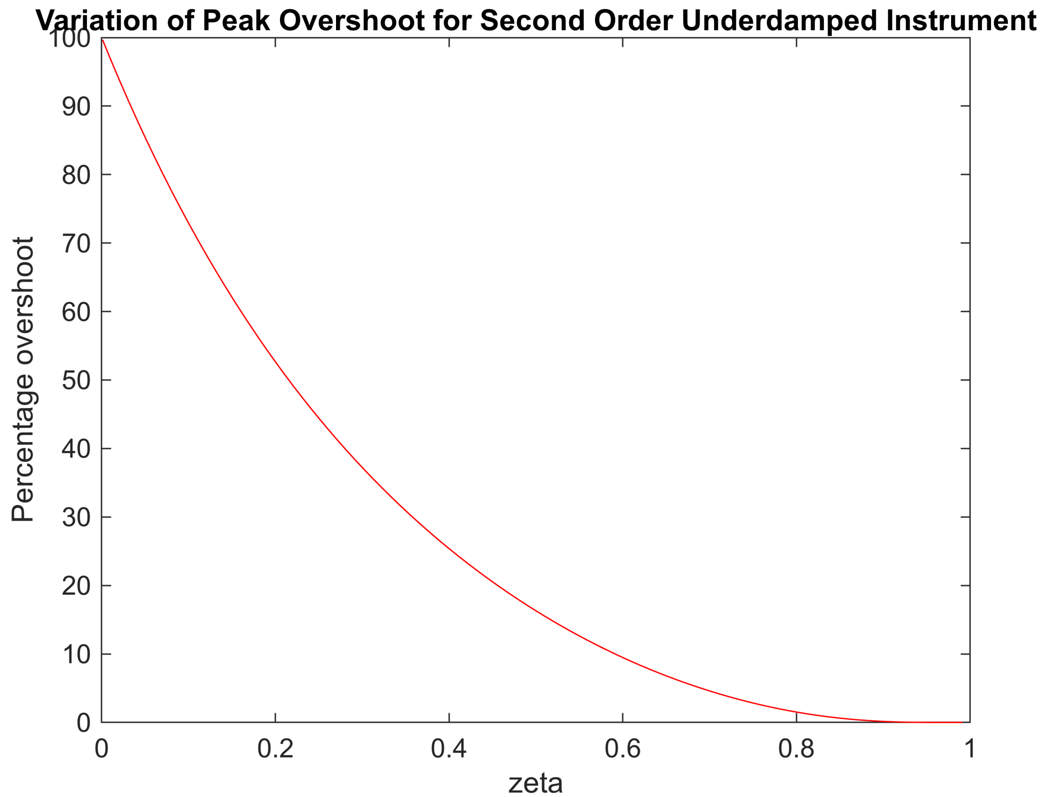
Inferences: As zeta increases, the overshoot increases, thus taking longer to settle.

DC-3

Aim: To plot the Variation of Peak Overshoot for a Second Order Underdamped Instrument

Code:

```
zeta = 0.001:0.01:1;

ytp = A*k + A*k*exp((-zeta.*pi)./(sqrt(1 - zeta.^2)) );
percentOvershoot = (ytp./(A*k) - 1)*100;

plot(zeta, percentOvershoot, 'r')

title('Variation of Peak Overshoot for Second Order Underdamped Instrument');
xlabel('zeta');
ylabel('Percentage overshoot');
```



Inferences: Percentage overshoot decreases with increasing zeta, exponentially.

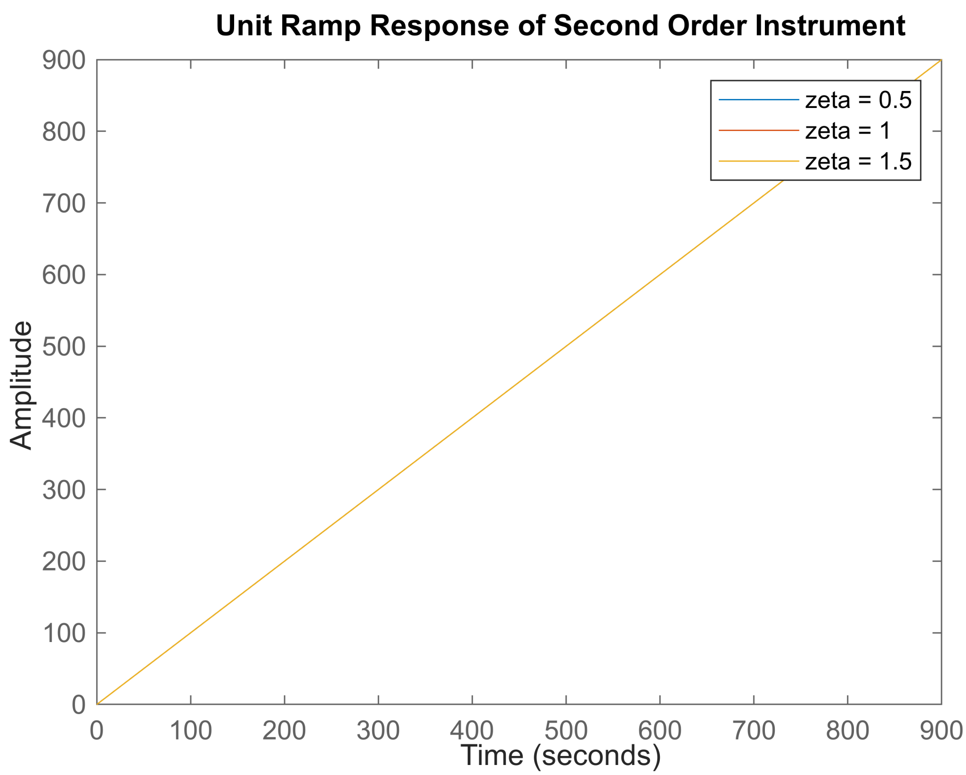
DC-4

Aim: To plot the Unit Ramp Response for a Second Order Instrument

Code:

```
% Variables
A = 1;
k = 1;
wn = 100;

s=tf('s');

% Plot for zeta = 0.3
zeta = 0.5; num = k*wn*wn; den = [1 2*zeta*wn wn*wn]; sys = tf(num, den);
stepplot(sys/s)

hold on

% Plot for zeta = 0.5
zeta = 1; num = k*wn*wn; den = [1 2*zeta*wn wn*wn]; sys = tf(num, den);
stepplot(sys/s)
```

```
% Plot for zeta = 0.8
zeta = 1.5; num = k*wn*wn; den = [1 2*zeta*wn wn*wn];
sys = tf(num, den);
stepplot(sys/s)

title('Unit Ramp Response of Second Order Instrument');
legend('zeta = 0.5', 'zeta = 1', 'zeta = 1.5');

hold off
```



Inferences: The unit ramp response suggest linear increase in output with time, with little variation with zeta.

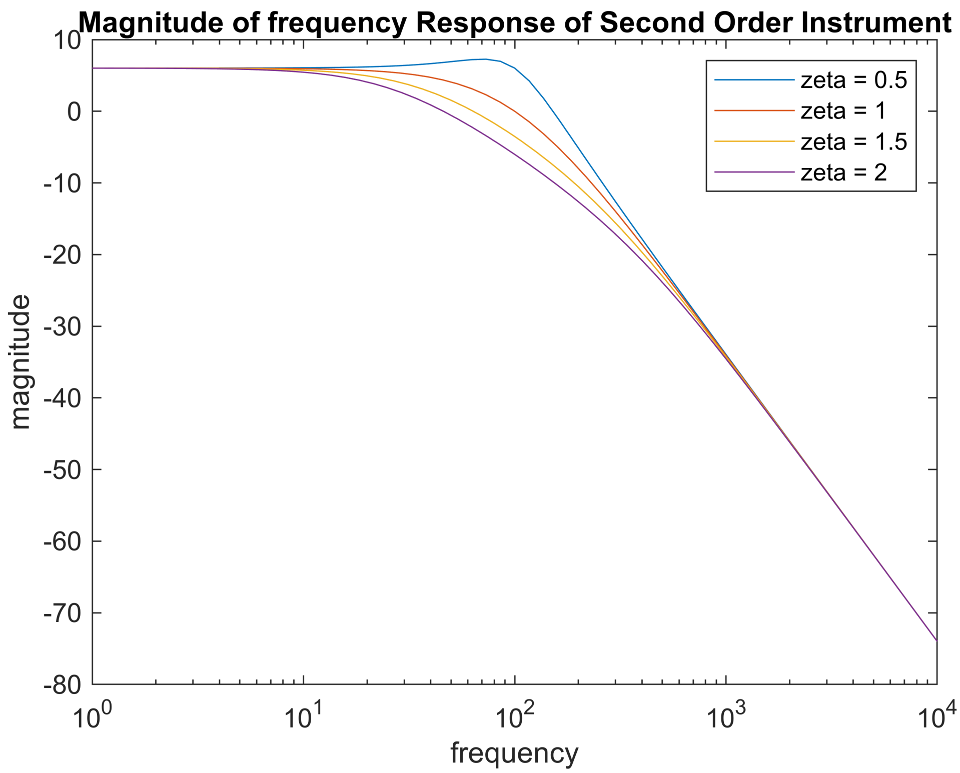
DC-5

Aim: To plot the Magnitude of frequency/sine Response of Second Order Instrument

Code:

```
% Variables
k = 2;
wn = 100;

% Plot for zeta = 1.5
zeta = 0.5; num = k*wn*wn; den = [1 2*zeta*wn wn*wn]; sys = tf(num, den);
[mag,~,wout] = bode(sys);
mag_dB = 20*log10(mag(:));
```

```
semilogx(wout, mag_dB)
hold on

% Plot for zeta = 2
zeta = 1; num = k*wn*wn; den = [1 2*zeta*wn wn*wn]; sys = tf(num, den);
[mag,~,wout] = bode(sys);
mag_dB = 20*log10(mag(:));
semilogx(wout, mag_dB)

% Plot for zeta = 2.5
zeta = 1.5; num = k*wn*wn; den = [1 2*zeta*wn wn*wn]; sys = tf(num, den);
[mag,~,wout] = bode(sys);
mag_dB = 20*log10(mag(:));
semilogx(wout, mag_dB)

% Plot for zeta = 3
zeta = 2; num = k*wn*wn; den = [1 2*zeta*wn wn*wn]; sys = tf(num, den);
[mag,~,wout] = bode(sys);
mag_dB = 20*log10(mag(:));
semilogx(wout, mag_dB)

title('Magnitude of frequency Response of Second Order Instrument');
legend('zeta = 0.5', 'zeta = 1', 'zeta = 1.5', 'zeta = 2');
xlabel('frequency');
ylabel('magnitude')

hold off
```



Inferences: The speed of response increases with zeta. For zeta > 1, the sytem shows a overshoot near w = wn.

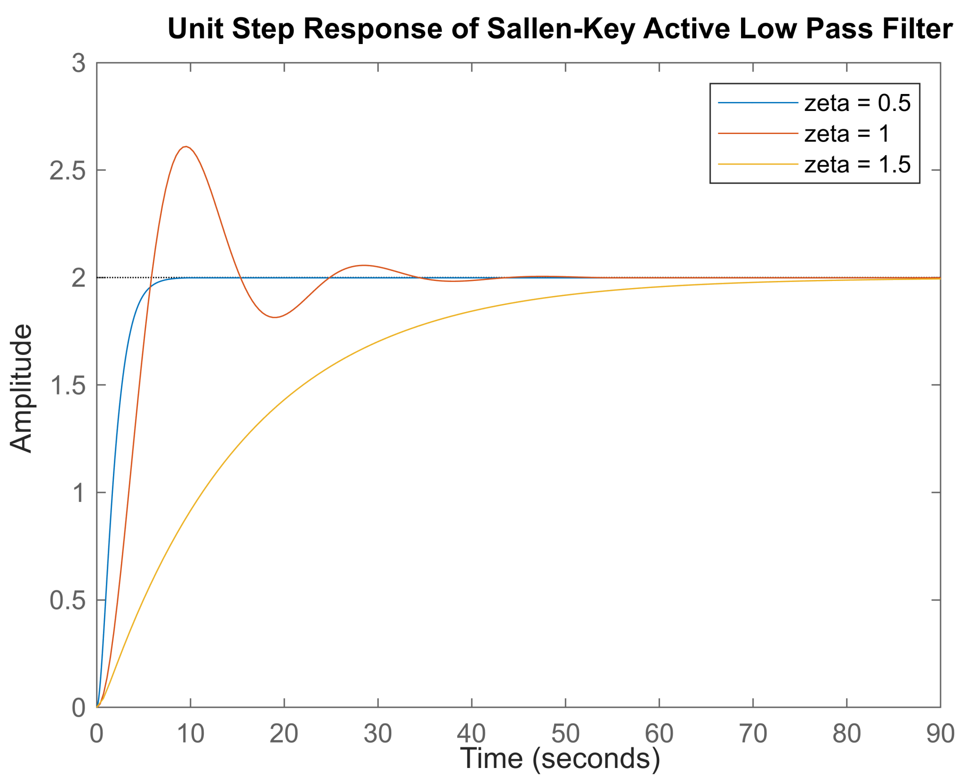
DC-6

Aim: To plot the unit step response of a Sallen-Key Active Low Pass Filter (LPF).

Code:

```
% system 1
% variables
R1 = 1; R2 = 1; C1 = 1; C2 = 1;
wn = 1 / (sqrt(R1*R2*C1*C2));
zeta = ( (R1 + R2)*C2 ) / ( 2*(sqrt(R1*R2*C1*C2)) );

num = k*wn*wn;
den = [1 2*zeta*wn wn*wn];
sys = tf(num, den);
step(sys)
hold on

% system 2
R1 = 1; R2 = 1; C1 = 8; C2 = 1;
wn = 1 / (sqrt(R1*R2*C1*C2));
zeta = ( (R1 + R2)*C2 ) / ( 2*(sqrt(R1*R2*C1*C2)) );
```

```
num = k*wn*wn;
den = [1 2*zeta*wn wn*wn];
sys = tf(num, den);
step(sys)

% system 3
R1 = 1; R2 = 1; C1 = 1; C2 = 8;
wn = 1 / (sqrt(R1*R2*C1*C2));
zeta = ( (R1 + R2)*C2 ) / ( 2*(sqrt(R1*R2*C1* ...
    C2)) );

num = k*wn*wn;
den = [1 2*zeta*wn wn*wn];
sys = tf(num, den);
step(sys)

title('Unit Step Response of Sallen-Key Active Low Pass Filter');
legend('zeta = 0.5', 'zeta = 1', 'zeta = 1.5');
hold off
```



Inferences: The unit response of sallen-key low pass filter is similar to a typical second-order system.

The speed of response increases with zeta. The system shows large overshoot for very low zeta.