EE3510 HW2 105060012張育菘

1. Simulate spring-mass-damper system of unforced response with damping ratio equal to 0.3, 0.6, and 1 (see Fig. 2.46) (p.108)

Ans.

$$\ddot{y} + \frac{b}{M}\dot{y} + \frac{k}{M}y = -\frac{r(t)}{M_s}$$

$$\text{Condition}: \frac{b}{M} = 2\xi\omega_n, \frac{k}{M} = \omega_n^2, = 2\dot{y}(0) = 2, y(0) = -1$$

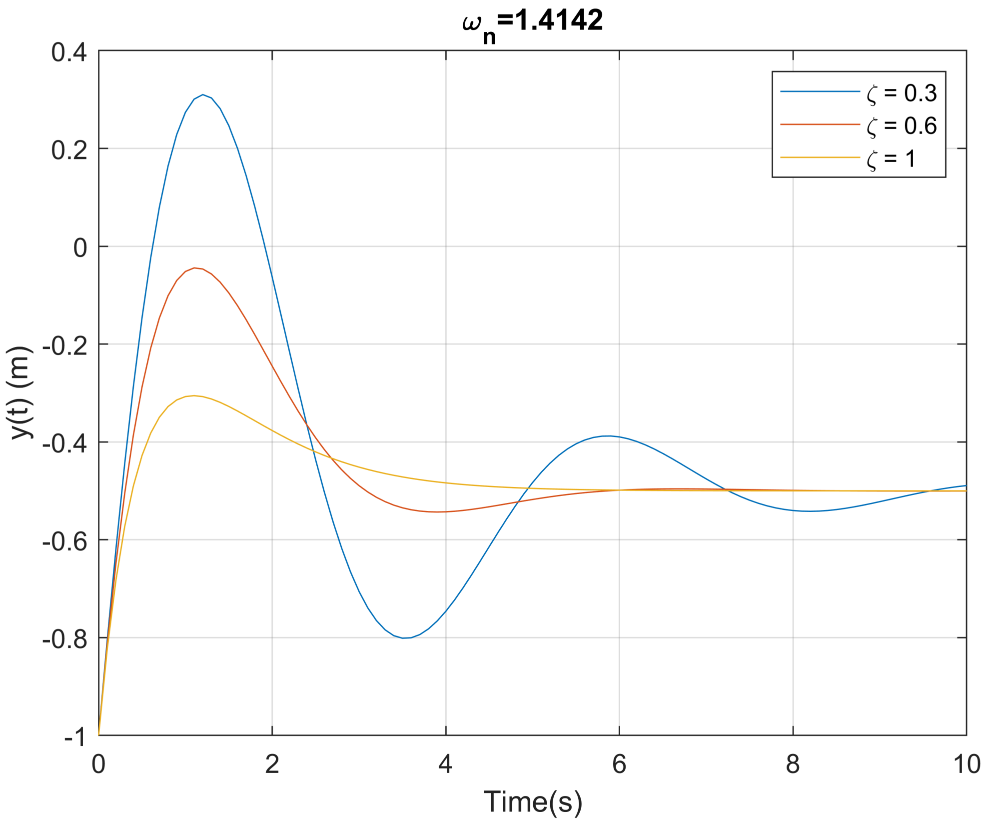
$$\Longrightarrow (s^2Y(s) - \text{s}y(0) - \dot{y}(0)) + \frac{b}{M}(\text{sY}(s) - y(0)) + \frac{k}{M}Y(s) = -\frac{P}{s}$$

$$\Longrightarrow Y(s) = \frac{-s^2 + \left(2 - \frac{b}{M}\right)s - P}{s\left(s^2 + \frac{b}{M}s + \frac{k}{M}\right)}$$

```
close all; clear;

% Damping ratio is (b/M)/(2wn), wn = sprt(k/M) = sqrt(2)
% the magnitude of the step response is P

fprintf('Using Inverse Laplace Transform to get the output signal in time domain\n');
```

```
Using Inverse Laplace Transform to get the output signal in time domain
```

```
wn = sqrt(2);
zeta = [0.3 0.6 1];

% function [y_t] = ilap(P,damp_ratio,wn)
y_t1 = ilap(1,0.3,sqrt(2));
y_t2 = ilap(1,0.6,sqrt(2));
y_t3 = ilap(1,1,sqrt(2));

t = [0:0.1:10];
figure;
plot(t, y_t1(t), t, y_t2(t), t, y_t3(t)); grid;
xlabel('Time(s)'), ylabel('y(t) (m)');
title(['\omega_n=',num2str(wn)]);
legend(['\zeta = ',num2str(zeta(1))], ['\zeta = ',num2str(zeta(2))],['\zeta = ',num2str(zeta(3)
```



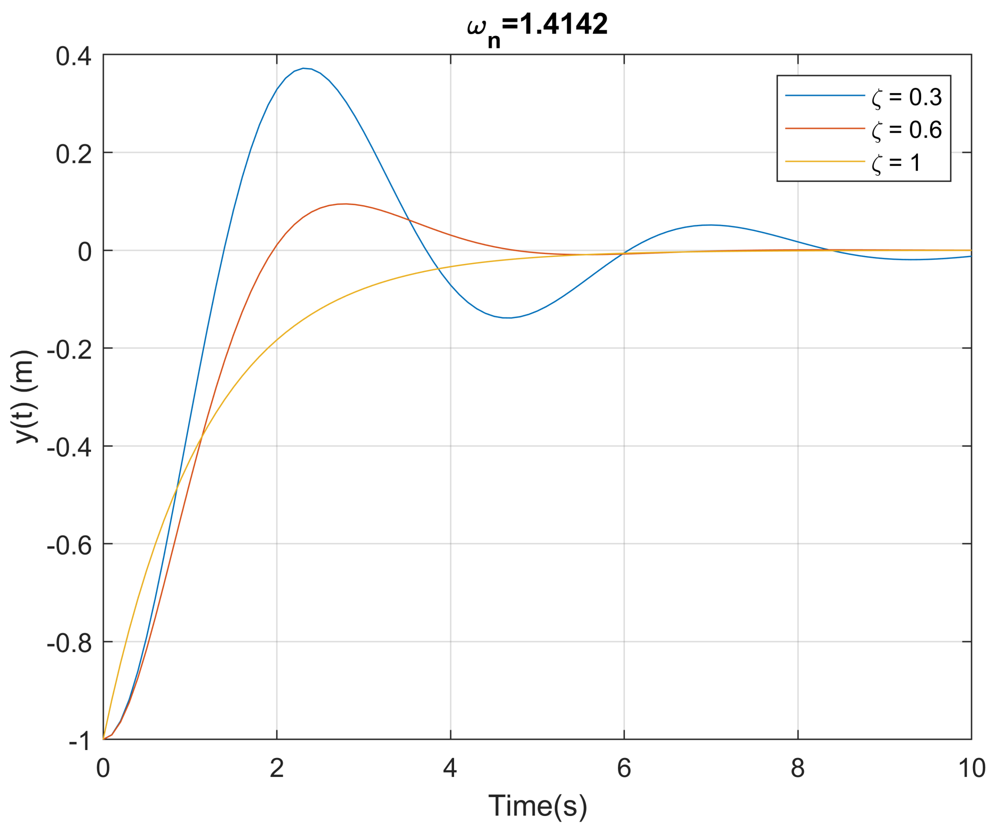
$$\implies y(t) = \frac{y(0)}{\sqrt{1-\zeta^2}} e^{-\zeta\omega_n t} \sin\left(\omega_n \sqrt{1-\zeta^2}\, t + \theta\right), \theta = \cos^{-1}\zeta$$

```
fprintf('by equation of output signal in time domain\n');
```

```
by equation of output signal in time domain
```

```
y0 = -1;
wn = sqrt(2);
zeta = [0.3 0.6 1];
t = [0:0.1:10];
y = zeros(3,length(t));

%Compute Unforced Response to an Initial Condition
for i=1:1:length(zeta)-1
    c = (y0/sqrt(1-zeta(i)^2));
    for j=1:1:length(t)
        y(i, j) = c*exp(-zeta(i)*wn*t(j)).*sin(wn*sqrt(1-zeta(i)^2)*t(j)+acos(zeta(i)));
    end
end

for j=1:1:length(t)
    y(length(zeta), j) = c*exp(-zeta(i)*wn*t(j)).*sin(wn*0*t(j)+acos(zeta(i)));
end

figure; plot(t,y); grid;
```

```
xlabel('Time(s)'), ylabel('y(t) (m)');
title(['\omega_n=',num2str(wn)]);
legend(['\zeta = ',num2str(zeta(1))], ['\zeta = ',num2str(zeta(2))],['\zeta = ',num2str(zeta(3
```



2. Simulate Example 2.19 (Series connection) (p.120)

$$G(s) = \frac{1}{500s^2}, G_c(s) = \frac{s+1}{s+2}$$

$$\text{sys1} = G_c(s)G(s)$$

$$\text{sys2} = \frac{G_c(s)G(s)}{1 + G_c(s)G(s)}$$

```
close all; clear;
numg = [1]; deng = [500 0 0]; sysg = tf(numg,deng);
numgc = [1 1]; dengc = [1 2]; sysgc = tf(numgc,dengc);
sys1 = series(sysg,sysgc)
```

```
sys1 =
 
         s + 1
  ------------------
  500 s^3 + 1000 s^2
 
Continuous-time transfer function.
```

```
sys2_num = series(sysg,sysgc);
sys2 = feedback(sys2_num,[1])
```

```
sys2 =

            s + 1
  --------------------------
  500 s^3 + 1000 s^2 + s + 1

Continuous-time transfer function.
```

3. Simulate Example 2.20 (The feedback function with unity feedback) (p.123)

$$G(s) = \frac{1}{500s^2}, G_c(s) = \frac{s+1}{s+2}$$

$$T_1(s) = \frac{G_c(s)G(s)}{1+G_c(s)G(s)}, T_2(s) = \frac{G(s)}{1 \mp G_c(s)G(s)}$$

$$T_{2-1}(s) = \frac{G(s)}{1+G_c(s)G(s)}, T_{2-2}(s) = \frac{G(s)}{1-G_c(s)G(s)}$$

```
close all; clear;
numg = [1]; deng = [500 0 0]; sysg = tf(numg,deng);
numgc = [1 1]; dengc = [1 2]; sysgc = tf(numgc,dengc);
T1_num = series(sysg,sysgc);
T1 = feedback(T1_num,[1],-1)
```

```
T1 =

            s + 1
  --------------------------
  500 s^3 + 1000 s^2 + s + 1

Continuous-time transfer function.
```

```
T2_1 = feedback(sysg,sysgc,-1)
```

```
T2_1 =

            s + 2
  --------------------------
  500 s^3 + 1000 s^2 + s + 1

Continuous-time transfer function.
```

```
T2_2 = feedback(sysg,sysgc,+1)
```

```
T2_2 =

            s + 2
  --------------------------
```

```
  500 s^3 + 1000 s^2 - s - 1

Continuous-time transfer function.
```

```
function [y_t] = ilap(P,damp_ratio,wn)
    a_2 = 2*damp_ratio*wn; a_3 = wn^2;

    a = [1 a_2 a_3 0]; b = [-1 (2-a_2) -P];

    syms s;
    y_b = 0; y_a = 0;
    for i=1:1:length(b)
        y_b = y_b + b(i)*(s^(length(b)-i));
    end
    for i=1:1:length(a)
        y_a = y_a + a(i)*(s^(length(a)-i));
    end

    y = y_b./y_a;
    y_t = ilaplace(y); y_t = vpa(y_t);
    y_t = matlabFunction(y_t);

end
```