2(b) Second Order MSD Equation

Table of Contents

Author: Pushkar Antony PS Number: 99003729 Date: 7th April 2021. Version: 1.0.

Plant Description

The Mass-damper Spring Second order system is taken as Plant. It is used in as suspension.

```
% Equation: Mx''(t)+ Bx'(t) + Kx(t)= Kf(t).
% f = force; B= coefficient of friction; M = mass ; v= velocity;
 k=spring
%constant.
% Values: K1= 0.9 B1= 0.4 M1=1000 Wn=0.03 ; K2= 1 B2= 0.5 M2= 500
 Wn=0.44;
%K3= 3 B3= 1.7 M3= 340 Wn=0.09;
```

Code:

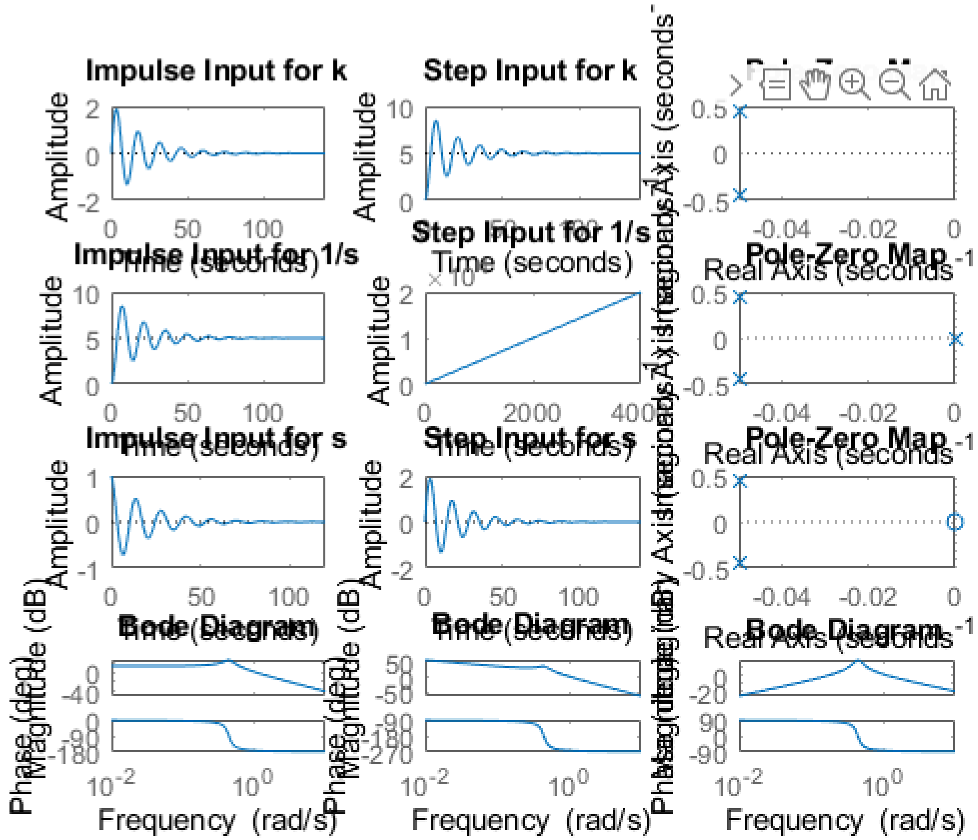
```
clc;
B1= 0.5
M1= 5;
K1 =1;
P=5;
sys = tf([P*K1/M1],[1,B1/M1,K1/M1])
subplot(4,3,1);
impulse(sys);
title('Impulse Input for k');
subplot(4,3,2);
step(sys);
title('Step Input for k');
subplot(4,3,3);
[z,p,k]= tf2zp([P*K1/M1],[1,B1/M1,K1/M1])
pzmap(sys)
subplot(4,3,10);
bode(sys)
hold on;
S = stepinfo(sys)

sys = tf([P*K1/M1],[1,B1/M1,K1/M1,0])
subplot(4,3,4);
impulse(sys);
title('Impulse Input for 1/s');
subplot(4,3,5);
step(sys);
```

```
title('Step Input for 1/s');
subplot(4,3,6);
[z,p,k]= tf2zp([P*K1/M1],[1,B1/M1,K1/M1,0])
pzmap(sys)
subplot(4,3,11);
bode(sys)
hold on;
S = stepinfo(sys)

sys = tf([P*K1/M1,0],[1,B1/M1,K1/M1])
subplot(4,3,7);
impulse(sys);
title('Impulse Input for s');
subplot(4,3,8);
step(sys);
title('Step Input for s');
subplot(4,3,9);
[z,p,k]= tf2zp([P*K1/M1,0],[1,B1/M1,K1/M1])
pzmap(sys)
subplot(4,3,12);
bode(sys)
hold on;
S = stepinfo(sys)
```

```
B1 =

    0.5000


sys =

          1
  -----------------
  s^2 + 0.1 s + 0.2

Continuous-time transfer function.


z =

  0×1 empty double column vector


p =

  -0.0500 + 0.4444i
  -0.0500 - 0.4444i


k =

     1
```

```
S =

  struct with fields:

        RiseTime: 2.5448
    SettlingTime: 78.1524
     SettlingMin: 2.5361
     SettlingMax: 8.5106
       Overshoot: 70.2118
      Undershoot: 0
            Peak: 8.5106
        PeakTime: 7.0248


sys =

            1
  --------------------
  s^3 + 0.1 s^2 + 0.2 s

Continuous-time transfer function.


z =

  0×1 empty double column vector


p =

   0.0000 + 0.0000i
  -0.0500 + 0.4444i
  -0.0500 - 0.4444i


k =

     1


S =

  struct with fields:

        RiseTime: NaN
    SettlingTime: NaN
     SettlingMin: NaN
     SettlingMax: NaN
       Overshoot: NaN
      Undershoot: NaN
            Peak: Inf
        PeakTime: Inf
```

```
sys =

          s
  -----------------
  s^2 + 0.1 s + 0.2

Continuous-time transfer function.


z =

     0


p =

  -0.0500 + 0.4444i
  -0.0500 - 0.4444i


k =

     1


S =

  struct with fields:

        RiseTime: 0
    SettlingTime: 81.5509
     SettlingMin: -1.3280
     SettlingMax: 1.8877
       Overshoot: Inf
      Undershoot: Inf
            Peak: 1.8877
        PeakTime: 3.5124
```



Math Analysis:

Independent: Time(t) Dependent: Velocity(v) and Force(f) Constant: Mass(M), Frictional Coefficient(B), Spring constant(K)

```
% Roots:((-B/M)+-sqrt(sq(B/M)-4K/M))/2

% IVT:
% 1. For step input: 0
% 2. For impulse input: 0

% FVT:
% 1. For step input: 1
% 2. For impulse input: K/M

% Time Response Results:
% K1= 0.9 B1= 0.4 M1=1000
%        RiseTime: 2.5448
%    SettlingTime: 78.1524
%     SettlingMin: 2.5361
%     SettlingMax: 8.5106
%       Overshoot: 70.2118
%      Undershoot: 0
%            Peak: 8.5106
%        PeakTime: 7.0248
```

```
%K2= 1 B2= 0.5 M2= 500
%        RiseTime: NaN
%    SettlingTime: NaN
%     SettlingMin: NaN
%     SettlingMax: NaN
%       Overshoot: NaN
%      Undershoot: NaN
%            Peak: Inf
%        PeakTime: Inf

%K3= 3 B3= 1.7 M3= 340
%        RiseTime: 0
%    SettlingTime: 81.5509
%     SettlingMin: -1.3280
%     SettlingMax: 1.8877
%       Overshoot: Inf
%      Undershoot: Inf
%            Peak: 1.8877
%        PeakTime: 3.5124
```

Comparison Analysis:(Speed, Accuracy and stability):

1) with proportionality controller, only the amplitude changes and all

```
%other stats are same as in 2nd order system without controller.
% 2) On adding an integrator controller, a pole is getting added at
 the
%origin and makes the system marginally stable.
% 3) On adding a differentiator controller, a zero is added to the
 origin
%making an unstable system stable.
% 4) On adding a differentiator controller, the overshoot increases
 and
%also the response time increase.
```

Published with MATLAB® R2020b