2(c) 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; B4= 9 M4= 5 K4=1;
```

Code:

```
%For negative feedback
clc;
B1= 0.5
M1= 5;
K1 =1;
P=5;

sys = tf([P*K1],[M1,B1,2*K1])
subplot(4,4,1);
impulse(sys);
title('Impulse Input for k');
subplot(4,4,2);
step(sys);
title('Step Input for k');
subplot(4,4,3);
[z,p,k]= tf2zp([P*K1],[M1,B1,2*K1])
pzmap(sys)
subplot(4,4,4)
bode(sys)
margin(sys)
[Gm,Pm,Wcg,Wcp] = margin(sys)
hold on;
S = stepinfo(sys)

B2= -9
```

```
M2= 5;
K2=1;
P2=5;

sys = tf([P2*K2],[M2,B2,2*K2])
subplot(4,4,5);
impulse(sys);
title('Impulse Input for k- Unstable');
subplot(4,4,6);
step(sys);
title('Step Input for k- Unstable');
subplot(4,4,7);
[z,p,k]= tf2zp([P2*K2],[M2,B2,2*K2])
pzmap(sys)
subplot(4,4,8)
bode(sys)
margin(sys)
[Gm,Pm,Wcg,Wcp] = margin(sys)
hold on;
S = stepinfo(sys)

% For Positive feedback using I & D

B3= 9
M3= 5;
K3=1;
sys = tf([K3],[M3,B3,0,0])
subplot(4,4,9);
impulse(sys);
title('Impulse Input for Positive feedback 1/s ');
subplot(4,4,10);
step(sys);
title('Step Input for Positive feedback 1/s');
subplot(4,4,11);
[z,p,k]= tf2zp([K3],[M3,B3,0,0])
pzmap(sys)
subplot(4,4,12)
bode(sys)
margin(sys)
[Gm,Pm,Wcg,Wcp] = margin(sys)
hold on;
S = stepinfo(sys)


B4= 9
M4= 5;
K4=1;
sys = tf([K4,0],[M4,B4,0])
subplot(4,4,13);
impulse(sys);
title('Impulse Input for Positive feedback s ');
subplot(4,4,14);
step(sys);
title('Step Input for Positive feedback s');
```

```
subplot(4,4,15);
[z,p,k]= tf2zp([K4,0],[M4,B4,0])
pzmap(sys)
subplot(4,4,16)
bode(sys)
margin(sys)
[Gm,Pm,Wcg,Wcp] = margin(sys)
hold on;
S = stepinfo(sys)
```

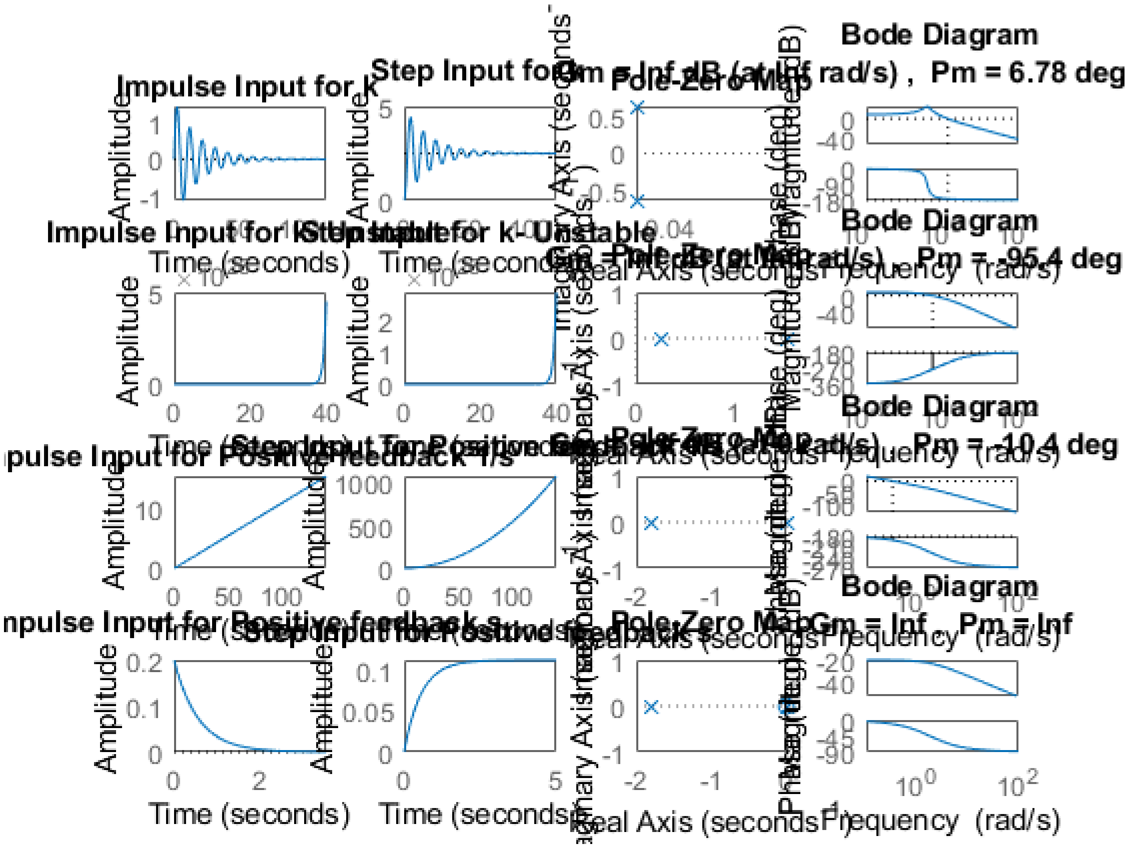
```
B1 =

    0.5000


sys =

          5
  -----------------
  5 s^2 + 0.5 s + 2

Continuous-time transfer function.


z =

  0×1 empty double column vector


p =

  -0.0500 + 0.6305i
  -0.0500 - 0.6305i


k =

     1


Gm =

   Inf


Pm =

    6.7782


Wcg =

   Inf
```

```
Wcp =

    1.1803


S =

  struct with fields:

        RiseTime: 1.7526
    SettlingTime: 75.6433
     SettlingMin: 0.9814
     SettlingMax: 4.4486
       Overshoot: 77.9429
      Undershoot: 0
            Peak: 4.4486
        PeakTime: 4.9673


B2 =

    -9


sys =

         5
  ---------------
  5 s^2 - 9 s + 2

Continuous-time transfer function.


z =

  0×1 empty double column vector


p =

    1.5403
    0.2597


k =

     1

Warning: The closed-loop system is unstable.

Gm =
```

```
    Inf


Pm =

  -95.4008


Wcg =

    Inf


Wcp =

     0.5531


S =

  struct with fields:

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


B3 =

     9


sys =

        1
  -------------
  5 s^3 + 9 s^2

Continuous-time transfer function.


z =

  0×1 empty double column vector


p =

         0
```

```
         0
   -1.8000


k =

    0.2000

Warning: The closed-loop system is unstable.

Gm =

     0


Pm =

  -10.4065


Wcg =

     0


Wcp =

    0.3306


S =

  struct with fields:

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


B4 =

     9


sys =

       s
  -----------
  5 s^2 + 9 s
```

```
Continuous-time transfer function.


z =

     0


p =

         0
   -1.8000


k =

    0.2000

Warning: The closed-loop system is unstable.

Gm =

   Inf


Pm =

   Inf


Wcg =

   NaN


Wcp =

   NaN


S =

  struct with fields:

        RiseTime: 1.2206
    SettlingTime: 2.1734
     SettlingMin: 0.1005
     SettlingMax: 0.1111
       Overshoot: 0
      Undershoot: 0
            Peak: 0.1111
        PeakTime: 5.8588
```



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: 1.7526
%    SettlingTime: 75.6433
%     SettlingMin: 0.9814
%     SettlingMax: 4.4486
%       Overshoot: 77.9429
%      Undershoot: 0
%            Peak: 4.4486
%        PeakTime: 4.9673
```

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

%K4= 1 B4= 9 M4= 5;
%        RiseTime: 1.2206
%    SettlingTime: 2.1734
%     SettlingMin: 0.1005
%     SettlingMax: 0.1111
%       Overshoot: 0
%      Undershoot: 0
%            Peak: 0.1111
%        PeakTime: 5.8588
```

Comparison Analysis:(Speed, Accuracy and stability):

```
%-ve feedback
% 1) On adding a -ve feedback, the stability of the system reaches at
 a
%faster speed.
% 2) The gain margin is infinity and the phase margin is +ve value
 making
%the system stable.
% 3) When the gain margin is infinity and the phase margin is -ve
 value the
%system becomes unstable.
%+ve feedback
% 1)  When the gain margin is 0 and the phase margin is -ve value the
%system becomes unstable.
% 2) When both gain margin and phase margin are infinity, the errors
 get
%accumulated and makes the system unstable.
```

Published with MATLAB® R2020b