1(b) First Order Equation

Table of Contents

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

Plant Description

The Mass-damper first order system is taken as Plant. Equation: f= Bv + M v' f = force; B= coefficient of friction; M = mass ; v= velocity. Values: B1= 0.4, M1=1000; B2= 0.5, M2= 500; B3= 1.7, M3= 340;

Code:

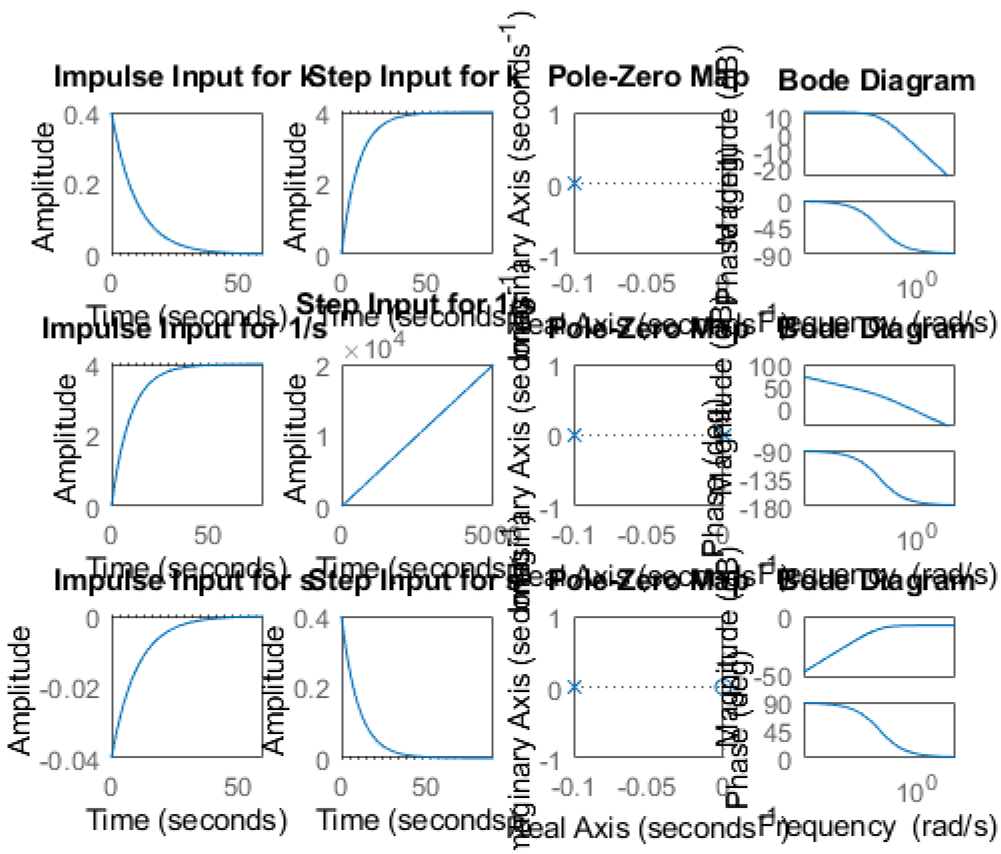
```
clc;
B1= 0.5;
M1= 5;
P = 2;

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

sys = tf([P/M1],[1,B1/M1,0])
subplot(3,4,5);
impulse(sys);
title('Impulse Input for 1/s');
subplot(3,4,6);
step(sys);
title('Step Input for 1/s');
subplot(3,4,7);
[z,p,k]= tf2zp([P/M1],[1,B1/M1,0])
pzmap(sys)
subplot(3,4,8);
```

```
bode(sys)
hold on;
S = stepinfo(sys)

sys = tf([P/M1,0],[1,B1/M1])
subplot(3,4,9);
impulse(sys);
title('Impulse Input for s');
subplot(3,4,10);
step(sys);
title('Step Input for s');
subplot(3,4,11);
[z,p,k]= tf2zp([P/M1,0],[1,B1/M1])
pzmap(sys)
subplot(3,4,12);
bode(sys)
hold on;
S = stepinfo(sys)
```

```
sys =

    0.4
  -------
  s + 0.1

Continuous-time transfer function.


z =

  0×1 empty double column vector


p =

   -0.1000


k =

    0.4000


S =

  struct with fields:

        RiseTime: 21.9701
    SettlingTime: 39.1207
     SettlingMin: 3.6180
     SettlingMax: 3.9999
       Overshoot: 0
      Undershoot: 0
```

```
            Peak: 3.9999
        PeakTime: 105.4584


sys =

      0.4
  -----------
  s^2 + 0.1 s

Continuous-time transfer function.


z =

  0×1 empty double column vector


p =

         0
   -0.1000


k =

    0.4000


S =

  struct with fields:

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


sys =

   0.4 s
  -------
  s + 0.1

Continuous-time transfer function.


z =
```

```
     0


p =

   -0.1000


k =

    0.4000


S =

  struct with fields:

        RiseTime: 21.9701
    SettlingTime: 39.1207
     SettlingMin: 1.0521e-05
     SettlingMax: 0.0382
       Overshoot: 0
      Undershoot: 7.2058e+17
            Peak: 0.4000
        PeakTime: 0
```



Math Analysis

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

```
% Roots:(-B)/M

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

% FVT:
% 1. For step input: 1/B
% 2. For impulse input: 0

% Time Response Results:
% Rise Time :4tau = (4M)/B; where tau = M/B
```

Comparison Analysis:(Speed, Accuracy and stability):

1) when a Proportionality controller is introduced, only the amplitude

```
%is getting incremented and all other parameters like rise time,
 settling
%time remain same as first order without controller.
% 2) when an integrator controller is introduced, a pole gets added at
 the
%origin and makes the system marginally stable.
% 3) When a differentiator controller is introduced, a zero gets added
 to
%the origin making any unstable system also stable.
% 4) PID controllers control the whole system making them unstable to
%stable, more stable, add poles, add zeros.
```

Published with MATLAB® R2020b