Table of Contents

Title:Control System-Second Order System:open loop with different values

```
%Author:ShivaKumar Naga VAnkadhara
%PS No:99003727
%Date:11/04/2021
%Version:1.7
```

This Document has equation for DC Motor

```
%Equation:Ldi/dt+Ri+Kw=V
%         Jdw/dt+bw=Ki
%T(s)=(K/LJ)/(s^2+((b/J)+(R/L)s+(R*b)/(L*J)+(K*K)/(L*J)
```

Math analysis

```
%dependent variables:w
%independent variables:t
%constant:K,R,L,J,b
%Roots:0.5*(-(b/J)-(R/L))+sqrt((((b*b)/(J*J))+((R*R)/(L*L))-((2*R*b)/
(L*J))-((4*K*K)/(L*J)))
%      0.5*(-(b/J)-(R/L))-sqrt((((b*b)/(J*J))+((R*R)/(L*L))-((2*R*b)/
(L*J))-((4*K*K)/(L*J)))
```

IVT

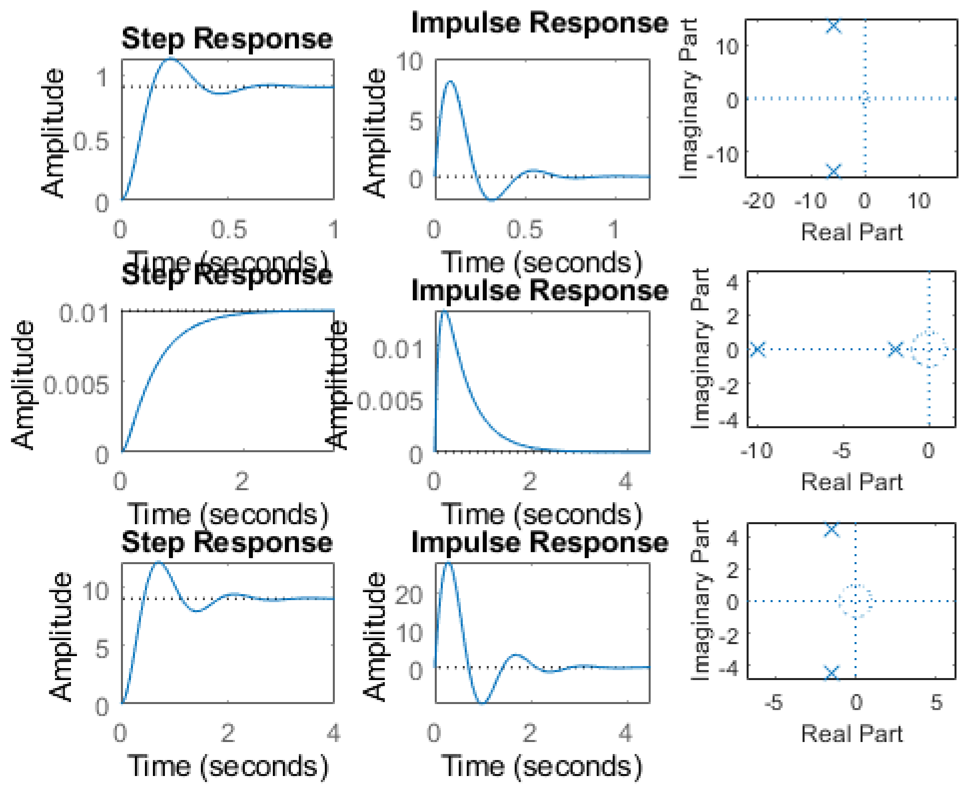
```
%for impulse is 0
%for step is 0
%%FVT
%for impulse is K/((b*L)+(R*J))=0.1667
%for step is K/((R*b)+(K*K))=0.0999001

J = 0.01;
b = 0.1;
K = 1;
R = 1;
L = 0.5;
%TF=tf([K/(J*L)],[1,((b/J)+(R/L)),(((K*K)+(R*b))/(L*J))]);
```

```
sys = tf([K/(J*L)],[1,((b/J)+(R/L)),(((K*K)+(R*b))/(L*J))])
subplot(3,3,1)
step(sys)
subplot(3,3,2)
impulse(sys)
subplot(3,3,3)
%S = stepinfo(sys)
[z,p,k]= tf2zp([K/(J*L)],[1,((b/J)+(R/L)),(((K*K)+(R*b))/(L*J))])
zplane(z,p)
S = stepinfo(sys)

J = 0.1;
b = 1;
K = 0.1;
R = 10;
L = 5;
%TF=tf([K/(J*L)],[1,((b/J)+(R/L)),(((K*K)+(R*b))/(L*J))]);
sys = tf([K/(J*L)],[1,((b/J)+(R/L)),(((K*K)+(R*b))/(L*J))])
subplot(3,3,4)
step(sys)
subplot(3,3,5)
impulse(sys)
subplot(3,3,6)
%S = stepinfo(sys)
[z2,p2,k2]= tf2zp([K/(J*L)],[1,((b/J)+(R/L)),(((K*K)+(R*b))/(L*J))])
zplane(z2,p2)
S = stepinfo(sys)

J = 0.01;
b = 0.01;
K = 0.1;
R = 0.1;
L = 0.05;
%TF=tf([K/(J*L)],[1,((b/J)+(R/L)),(((K*K)+(R*b))/(L*J))]);
sys = tf([K/(J*L)],[1,((b/J)+(R/L)),(((K*K)+(R*b))/(L*J))])
subplot(3,3,7)
step(sys)
subplot(3,3,8)
impulse(sys)
subplot(3,3,9)
%S = stepinfo(sys)
[z1,p1,k1]= tf2zp([K/(J*L)],[1,((b/J)+(R/L)),(((K*K)+(R*b))/(L*J))])
zplane(z1,p1)
S = stepinfo(sys)
```

```
sys =

        200
  ----------------
  s^2 + 12 s + 220

Continuous-time transfer function.
```

```
z =

  0×1 empty double column vector


p =

  -6.0000 +13.5647i
  -6.0000 -13.5647i


k =

   200


S =

  struct with fields:

        RiseTime: 0.0993
    SettlingTime: 0.5669
     SettlingMin: 0.8527
     SettlingMax: 1.1356
       Overshoot: 24.9123
      Undershoot: 0
            Peak: 1.1356
        PeakTime: 0.2303


sys =

         0.2
  ------------------
  s^2 + 12 s + 20.02

Continuous-time transfer function.


z2 =

  0×1 empty double column vector


p2 =

   -9.9975
   -2.0025


k2 =

    0.2000
```

```
S =

  struct with fields:

        RiseTime: 1.1351
    SettlingTime: 2.0652
     SettlingMin: 0.0090
     SettlingMax: 0.0100
       Overshoot: 0
      Undershoot: 0
            Peak: 0.0100
        PeakTime: 3.6758


sys =

       200
  --------------
  s^2 + 3 s + 22

Continuous-time transfer function.


z1 =

  0×1 empty double column vector


p1 =

  -1.5000 + 4.4441i
  -1.5000 - 4.4441i


k1 =

   200


S =

  struct with fields:

        RiseTime: 0.2882
    SettlingTime: 2.3810
     SettlingMin: 8.0006
     SettlingMax: 12.2393
       Overshoot: 34.6325
      Undershoot: 0
            Peak: 12.2393
        PeakTime: 0.7061
```



Analysis

1.If rise time is less the system is not much stable and its speed 2.If the rise time is high the system may behave more stable its not speed in nature. 3.If the Over shoot is less the system is kind of stable. 4.If the Over shoot is more the system may behave less stable. 5.If settling time is less accuracy is high. 6.If the settling time is high accuracy is less. 7.In the above systems system 2 is more stable because overshoot is 0. 8.Peak time is inversly proportional to overshoot. so if peak time is more system is stable. 9.when we add proportional to the open loop no parameters get changed only peak time and overshoot changes.

Published with MATLAB® R2021a