Table of Contents

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

```
%Author:Ravikumar M Pise
%PS No:99003747
%Date:7/04/2021
%Version:1.0
```

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

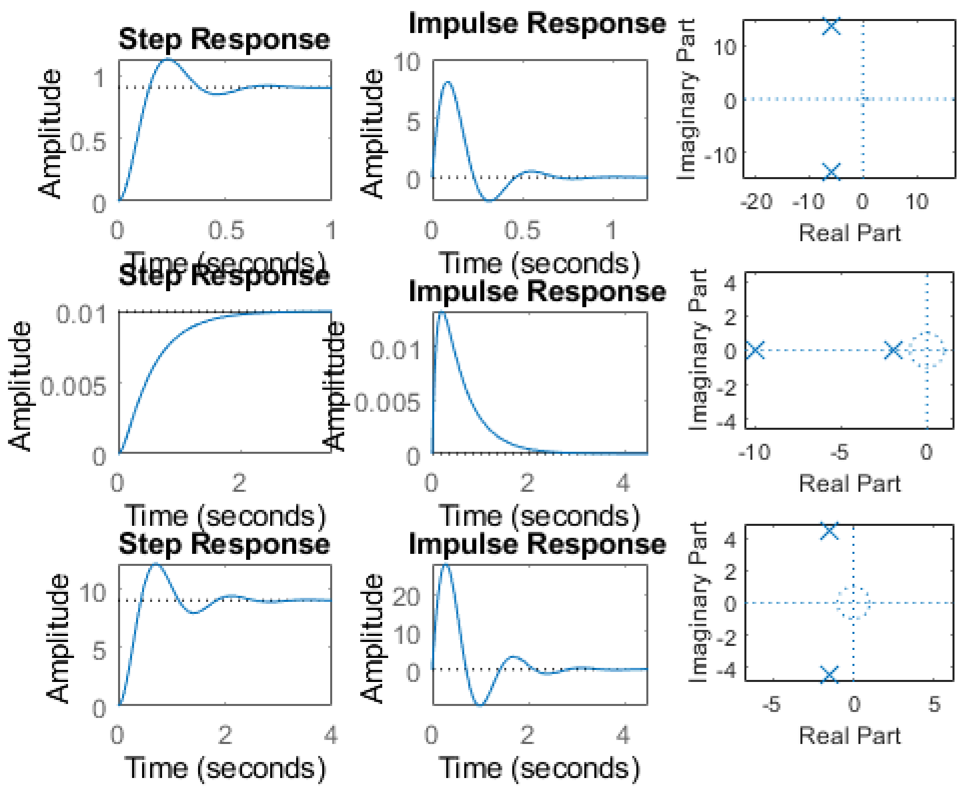
```
%for impulse is 0
%for step is 0
%%FVT
%for impulse is K/((b*L)+(R*J))=0.1667
```

```
%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:

1.1st response of system is accurate because settling time is less 2.1st response speed is high because rise time is less 3.2nd response is more stable because overshoot is zero

Control System-Second Order System:varying zeta value open system

This Document has equation for Second Order System

```
%w=1

jeta=1;
TF=tf([1],[1,(2*jeta),1])
sys = tf([1],[1,(2*jeta),1])
```

```
figure
subplot(2,3,1)
S = stepinfo(sys)
[z,p,k]= tf2zp([1],[1,(2*jeta),1])
zplane(z,p)

jeta=0.7;
TF=tf([1],[1,(2*jeta),1])
sys = tf([1],[1,(2*jeta),1])
%hold on
subplot(2,3,2)
S = stepinfo(sys)
[z,p,k]= tf2zp([1],[1,(2*jeta),1])
zplane(z,p)

jeta=1.5;
TF=tf([1],[1,(2*jeta),1])
sys = tf([1],[1,(2*jeta),1])
subplot(2,3,3)
S = stepinfo(sys)
[z,p,k]= tf2zp([1],[1,(2*jeta),1])
zplane(z,p)

jeta=-1;
TF=tf([1],[1,(2*jeta),1])
sys = tf([1],[1,(2*jeta),1])
subplot(2,3,4)
S = stepinfo(sys)
[z,p,k]= tf2zp([1],[1,(2*jeta),1])
zplane(z,p)


jeta=-0.5;
TF=tf([1],[1,(2*jeta),1])
sys = tf([1],[1,(2*jeta),1])
subplot(2,3,5)
S = stepinfo(sys)
[z,p,k]= tf2zp([1],[1,(2*jeta),1])
zplane(z,p)

jeta=-1.5;
TF=tf([1],[1,(2*jeta),1])
sys = tf([1],[1,(2*jeta),1])
subplot(2,3,6)
S = stepinfo(sys)
[z,p,k]= tf2zp([1],[1,(2*jeta),1])
zplane(z,p)

figure
jeta=0;
TF=tf([1],[1,(2*jeta),1])
sys = tf([1],[1,(2*jeta),1])
S = stepinfo(sys)
[z,p,k]= tf2zp([1],[1,(2*jeta),1])
```

```
zplane(z,p)


TF =

        1
  -------------
  s^2 + 2 s + 1

Continuous-time transfer function.


sys =

        1
  -------------
  s^2 + 2 s + 1

Continuous-time transfer function.


S =

  struct with fields:

        RiseTime: 3.3579
    SettlingTime: 5.8339
     SettlingMin: 0.9000
     SettlingMax: 0.9994
       Overshoot: 0
      Undershoot: 0
            Peak: 0.9994
        PeakTime: 9.7900


z =

  0×1 empty double column vector


p =

    -1
    -1


k =

     1


TF =

         1
```

```
  ---------------
  s^2 + 1.4 s + 1

Continuous-time transfer function.


sys =

         1
  ---------------
  s^2 + 1.4 s + 1

Continuous-time transfer function.


S =

  struct with fields:

        RiseTime: 2.1268
    SettlingTime: 5.9789
     SettlingMin: 0.9001
     SettlingMax: 1.0460
       Overshoot: 4.5986
      Undershoot: 0
            Peak: 1.0460
        PeakTime: 4.4078


z =

  0×1 empty double column vector


p =

  -0.7000 + 0.7141i
  -0.7000 - 0.7141i


k =

     1


TF =

        1
  -------------
  s^2 + 3 s + 1

Continuous-time transfer function.
```

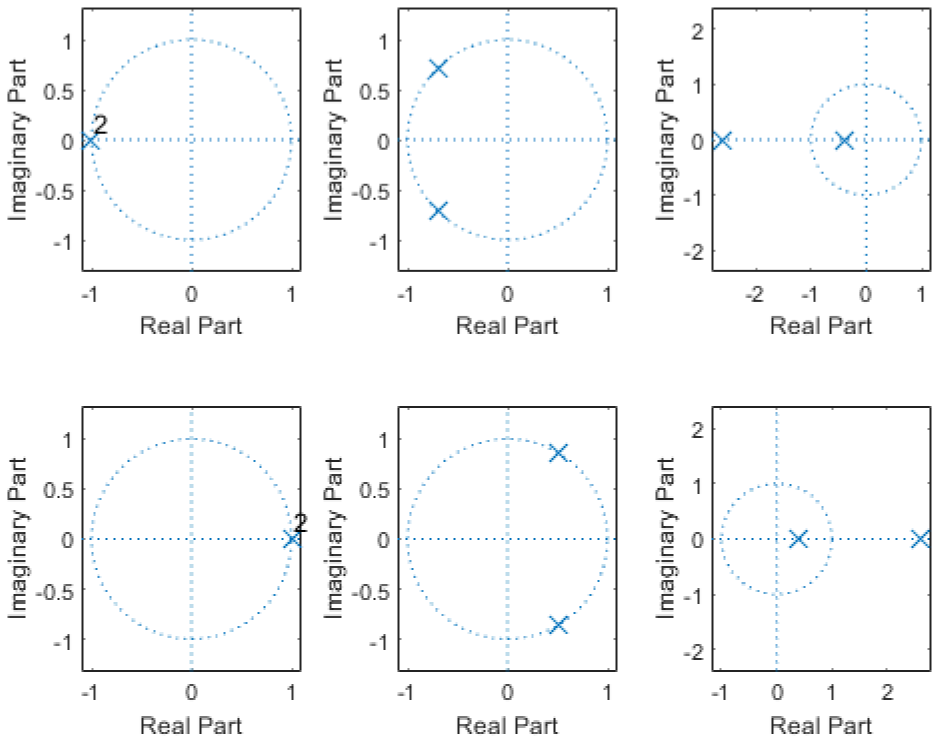
```
sys =

        1
  -------------
  s^2 + 3 s + 1

Continuous-time transfer function.


S =

  struct with fields:

        RiseTime: 5.8584
    SettlingTime: 10.6547
     SettlingMin: 0.9012
     SettlingMax: 0.9999
       Overshoot: 0
      Undershoot: 0
            Peak: 0.9999
        PeakTime: 25.9983


z =

  0×1 empty double column vector


p =

   -2.6180
   -0.3820


k =

     1


TF =

        1
  -------------
  s^2 - 2 s + 1

Continuous-time transfer function.


sys =

        1
  -------------
  s^2 - 2 s + 1
```

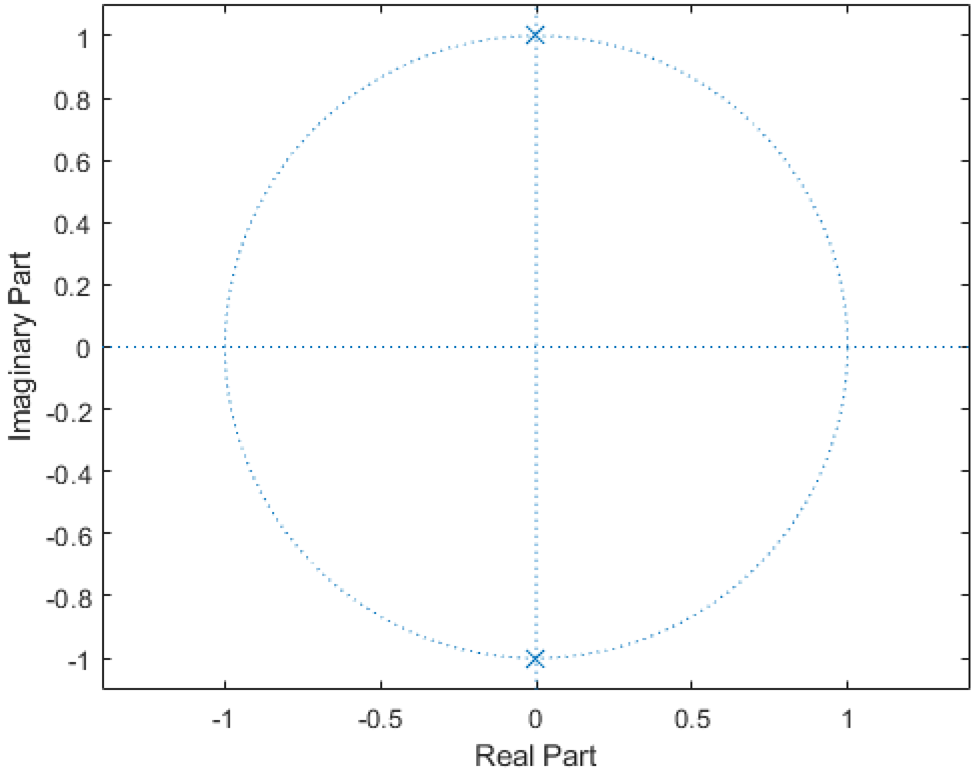
```
Continuous-time transfer function.


S =

  struct with fields:

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


z =

  0×1 empty double column vector


p =

     1
     1


k =

     1


TF =

       1
  -----------
  s^2 - s + 1

Continuous-time transfer function.


sys =

       1
  -----------
  s^2 - s + 1

Continuous-time transfer function.


S =

  struct with fields:
```

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


z =

  0×1 empty double column vector


p =

   0.5000 + 0.8660i
   0.5000 - 0.8660i


k =

     1


TF =

        1
  -------------
  s^2 - 3 s + 1

Continuous-time transfer function.


sys =

        1
  -------------
  s^2 - 3 s + 1

Continuous-time transfer function.


S =

  struct with fields:

        RiseTime: NaN
    SettlingTime: NaN
     SettlingMin: NaN
     SettlingMax: NaN
       Overshoot: NaN
```

```
      Undershoot: NaN
            Peak: Inf
        PeakTime: Inf


z =

  0×1 empty double column vector


p =

    2.6180
    0.3820


k =

     1


TF =

     1
  -------
  s^2 + 1

Continuous-time transfer function.


sys =

     1
  -------
  s^2 + 1

Continuous-time transfer function.


S =

  struct with fields:

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


z =
```

```
  0×1 empty double column vector


p =

   0.0000 + 1.0000i
   0.0000 - 1.0000i


k =

     1
```





Analysis

To get stable system zeta value should be greater than zero

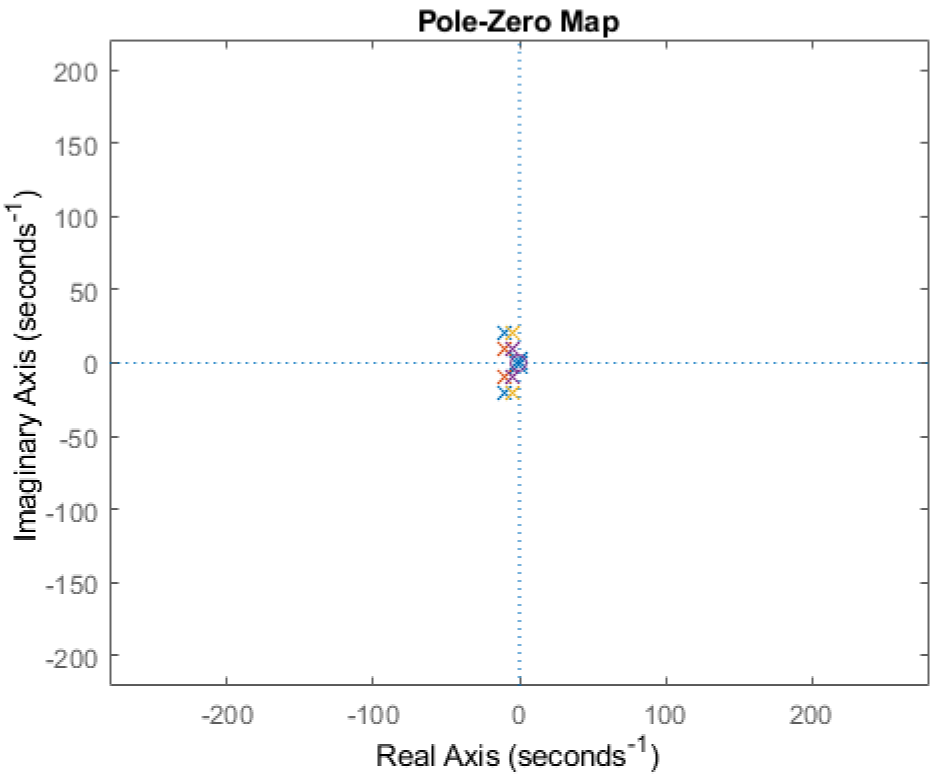
```
% if zeta value is between 0 to 1-->poles are conjugate

% if zeta is greater than 1-->poles are real stable

% if zeta is less then 1 -->Poles are real and unstable
```

This Document has movement of poles for Second Order System

```
zeros = 0;
poles = [-10+20i -10-20i];
gain = 1;
sys = zpk(zeros,poles,gain)
hold on
pzmap(sys)


zeros = 0;
poles = [-10+10i -10-10i];
gain = 1;
```

```
sys = zpk(zeros,poles,gain)
pzmap(sys)


zeros = 0;
poles = [-5+20i -5-20i];
gain = 1;
sys = zpk(zeros,poles,gain)
pzmap(sys)


zeros = 0;
poles = [-5+10i -5-10i];
gain = 1;
sys = zpk(zeros,poles,gain)
pzmap(sys)
```

```
sys =

          s
  -----------------
  (s^2 + 20s + 500)

Continuous-time zero/pole/gain model.


sys =

          s
  -----------------
  (s^2 + 20s + 200)

Continuous-time zero/pole/gain model.


sys =

          s
  -----------------
  (s^2 + 10s + 425)

Continuous-time zero/pole/gain model.


sys =

          s
  -----------------
  (s^2 + 10s + 125)

Continuous-time zero/pole/gain model.
```



Analysis

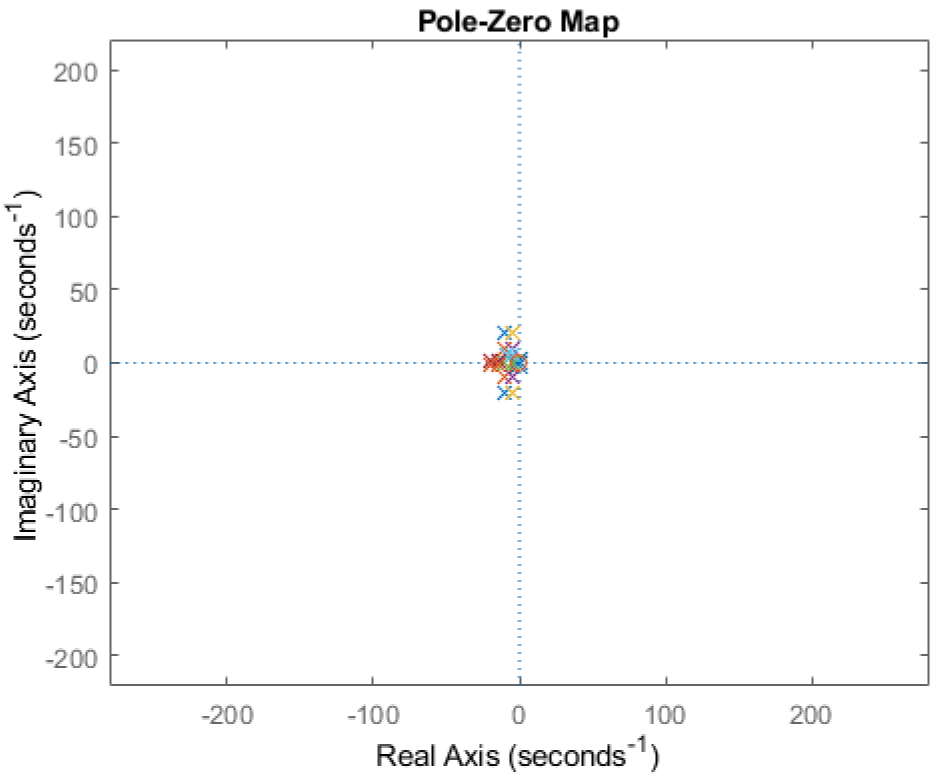
```
% 1.when the pole is moving lef overshoot decreses,Frequency increses
% 2.zeta value is gettting incresed,means damping is less,systemwill
 be stable

% In Vertical Shifting
% 1.Overshoot increases,damping decreases

% diagonal Shifting
% Overshoot,Damping will be same
% Frequency Increases
```

This Document has movement of poles for Second Order System

```
zeros = 0;
poles = [-10 -5];
gain = 1;
sys = zpk(zeros,poles,gain)
hold on
pzmap(sys)
```

```
zeros = 0;
poles = [-9+5i -5+5i];
gain = 1;
sys = zpk(zeros,poles,gain)
hold on
pzmap(sys)


zeros = 0;
poles = [-15+2i -20+2i];
gain = 1;
sys = zpk(zeros,poles,gain)
hold on
pzmap(sys)

zeros = 0;
poles = [-15-2i -20-2i];
gain = 1;
sys = zpk(zeros,poles,gain)
hold on
pzmap(sys)

zeros = 0;
poles = [-15-2i -20-2i];
gain = 1;
sys = zpk(zeros,poles,gain)
hold on
pzmap(sys)


sys =

       s
  ------------
  (s+10) (s+5)

Continuous-time zero/pole/gain model.

Warning: This zpk model has a complex gain or some complex zeros or
 poles that
do not come in conjugate pairs.

sys =

            s
  ---------------------
  (s+(9-5i)) (s+(5-5i))

Continuous-time zero/pole/gain model.

Warning: This zpk model has a complex gain or some complex zeros or
 poles that
do not come in conjugate pairs.
```

```
sys =

             s
  -----------------------
  (s+(15-2i)) (s+(20-2i))

Continuous-time zero/pole/gain model.

Warning: This zpk model has a complex gain or some complex zeros or
 poles that
do not come in conjugate pairs.

sys =

             s
  -----------------------
  (s+(15+2i)) (s+(20+2i))

Continuous-time zero/pole/gain model.

Warning: This zpk model has a complex gain or some complex zeros or
 poles that
do not come in conjugate pairs.

sys =

             s
  -----------------------
  (s+(15+2i)) (s+(20+2i))

Continuous-time zero/pole/gain model.
```



Negtaive Feedback

```
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))]);
CF=10
sys = CF*TF
NCTF1=feedback(sys,1)
subplot(3,2,1)
step(NCTF1)
title("Step with negative")
subplot(3,2,2)
impulse(NCTF1)
title("impulse with negative")
S = stepinfo(NCTF1)
[wn,zeta]=damp(NCTF1)
```

```
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))]);
CF=tf([1,0],[1])
sys = CF*TF
NCTF2=feedback(sys,1)
subplot(3,2,3)
step(NCTF2)
title("Step with diff")
subplot(3,2,4)
impulse(NCTF2)
title("impulse with diff")
S = stepinfo(NCTF2)
[wn,zeta]=damp(NCTF2)
```

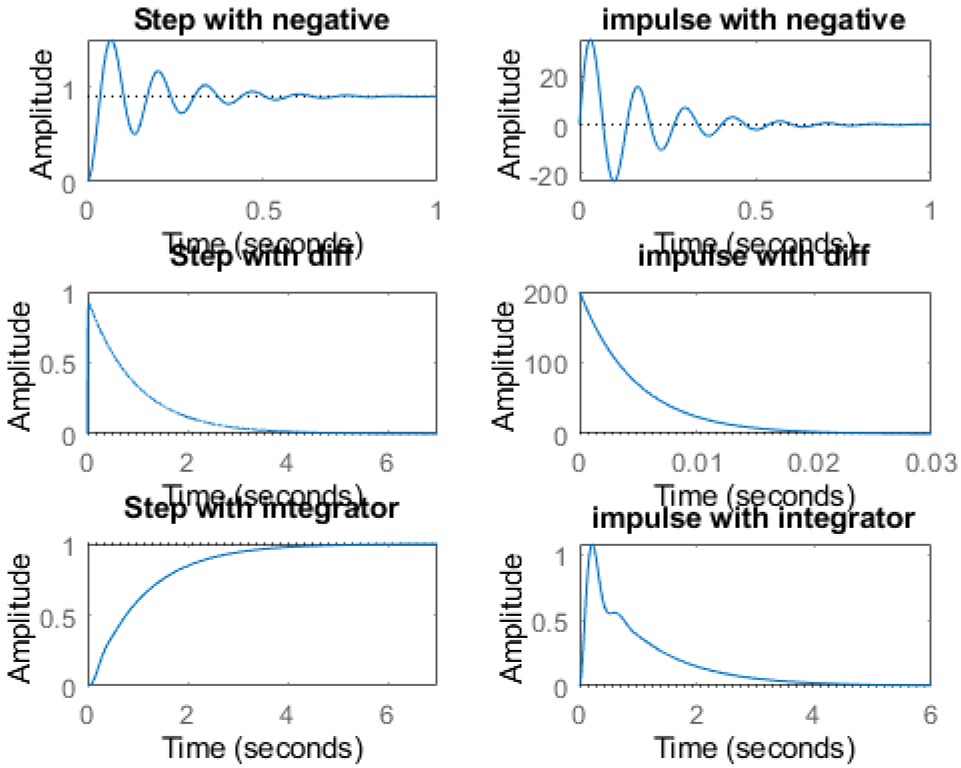
```
J = 0.01;
b = 0.1;
K = 1;
R = 1;
L = 0.5;
TF=tf([K/(J*L)],[1,((b/J)+(R/L)),(((K*K)+(R*b))/(L*J))]);
CF=tf([1],[1,0])
sys = CF*TF
NCTF3=feedback(sys,1)
subplot(3,2,5)
step(NCTF3)
title("Step with integrator")
subplot(3,2,6)
impulse(NCTF3)
title("impulse with integrator")
S = stepinfo(NCTF3)
[wn,zeta]=damp(NCTF3)
```

```
CF =

    10


sys =

        2000
  ----------------
  s^2 + 12 s + 220

Continuous-time transfer function.


NCTF1 =

        2000
```

```
  -----------------
  s^2 + 12 s + 2220

Continuous-time transfer function.


S =

  struct with fields:

        RiseTime: 0.0245
    SettlingTime: 0.6206
     SettlingMin: 0.4993
     SettlingMax: 1.5026
       Overshoot: 66.7860
      Undershoot: 0
            Peak: 1.5026
        PeakTime: 0.0667


wn =

   47.1169
   47.1169


zeta =

    0.1273
    0.1273


CF =

  s

Continuous-time transfer function.


sys =

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

Continuous-time transfer function.


NCTF2 =

        200 s
  -----------------
  s^2 + 212 s + 220
```

```
Continuous-time transfer function.


S =

  struct with fields:

        RiseTime: 0
    SettlingTime: 3.7813
     SettlingMin: 6.5963e-04
     SettlingMax: 0.9234
       Overshoot: Inf
      Undershoot: 0
            Peak: 0.9234
        PeakTime: 0.0253


wn =

    1.0429
  210.9571


zeta =

     1
     1


CF =

  1
  -
  s

Continuous-time transfer function.


sys =

          200
  --------------------
  s^3 + 12 s^2 + 220 s

Continuous-time transfer function.


NCTF3 =

              200
  --------------------------
  s^3 + 12 s^2 + 220 s + 200

Continuous-time transfer function.
```

```
S =

  struct with fields:

        RiseTime: 2.2719
    SettlingTime: 4.1463
     SettlingMin: 0.9044
     SettlingMax: 0.9993
       Overshoot: 0
      Undershoot: 0
            Peak: 0.9993
        PeakTime: 7.6683


wn =

    0.9549
   14.4725
   14.4725


zeta =

    1.0000
    0.3816
    0.3816
```



Positive Feedback

```
figure
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))]);
CF=10
sys = CF*TF
PCTF1=feedback(sys,-1)
subplot(3,2,1)
step(PCTF1)
title("Step with positive")
subplot(3,2,2)
impulse(PCTF1)
title("impulse with positive")
S = stepinfo(PCTF1)
[wn,zeta]=damp(PCTF1)



J = 0.01;
b = 0.1;
```

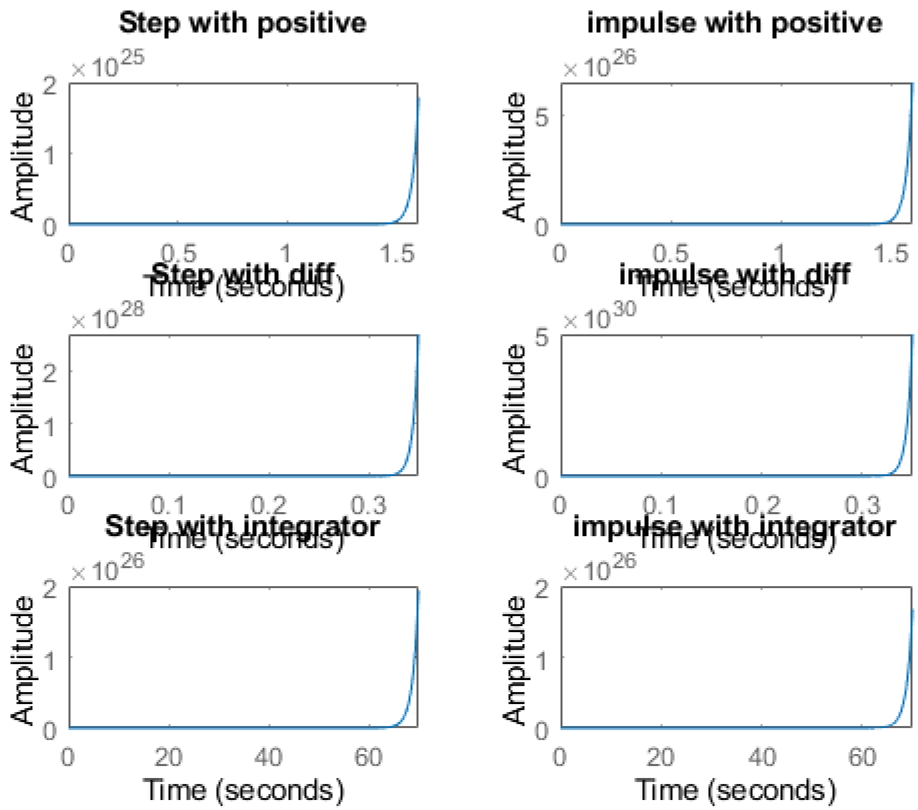
```
K = 1;
R = 1;
L = 0.5;
TF=tf([K/(J*L)],[1,((b/J)+(R/L)),(((K*K)+(R*b))/(L*J))]);
CF=tf([1,0],[1])
sys = CF*TF
PCTF2=feedback(sys,-1)
subplot(3,2,3)
step(PCTF2)
title("Step with diff")
subplot(3,2,4)
impulse(PCTF2)
title("impulse with diff")
S = stepinfo(PCTF2)
[wn,zeta]=damp(PCTF2)




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))]);
CF=tf([1],[1,0])
sys = CF*TF
PCTF3=feedback(sys,-1)
subplot(3,2,5)
step(PCTF3)
title("Step with integrator")
subplot(3,2,6)
impulse(PCTF3)
title("impulse with integrator")
S = stepinfo(PCTF3)
[wn,zeta]=damp(PCTF3)
```

```
CF =

    10


sys =

        2000
  ----------------
  s^2 + 12 s + 220

Continuous-time transfer function.


PCTF1 =
```

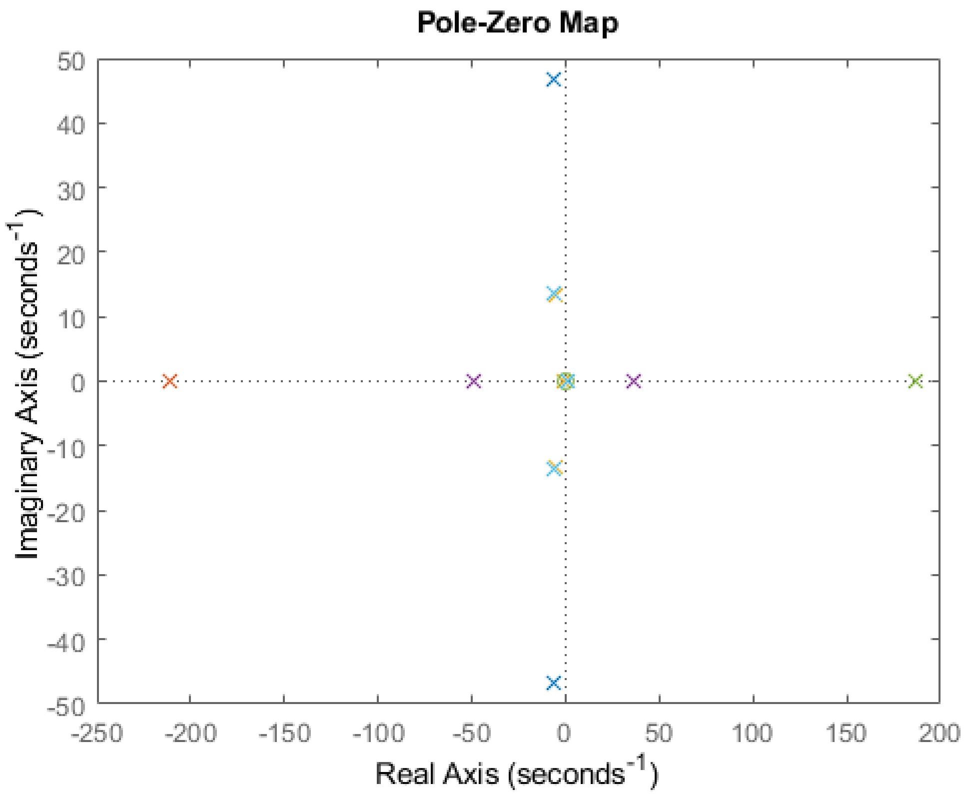
```
        2000
  -----------------
  s^2 + 12 s - 1780

Continuous-time transfer function.


S =

  struct with fields:

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


wn =

   36.6146
   48.6146


zeta =

    -1
     1


CF =

  s

Continuous-time transfer function.


sys =

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

Continuous-time transfer function.


PCTF2 =

        200 s
  -----------------
  s^2 - 188 s + 220
```

```
Continuous-time transfer function.


S =

  struct with fields:

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


wn =

    1.1776
  186.8224


zeta =

    -1
    -1


CF =

  1
  -
  s

Continuous-time transfer function.


sys =

          200
  --------------------
  s^3 + 12 s^2 + 220 s

Continuous-time transfer function.


PCTF3 =

             200
  --------------------------
  s^3 + 12 s^2 + 220 s - 200
```

```
Continuous-time transfer function.


S =

  struct with fields:

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


wn =

    0.8653
   15.2030
   15.2030


zeta =

   -1.0000
    0.4231
    0.4231
```




```
figure
hold on
pzmap(NCTF1)
pzmap(NCTF2)
pzmap(NCTF3)
pzmap(PCTF1)
pzmap(PCTF2)
pzmap(PCTF3)
```



Analysis

by adding positive feedback with integrator damping will be less,so the system will be stable

```
% Overshooot is infinity,in differentiator closed loop, system is
 unstable
```

```
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))]);
CF=1;
sys1 = CF*TF;
subplot(4,2,1)
step(sys1)
title("Step ")
subplot(4,2,2)
impulse(sys1)
title("Impulse")
S = stepinfo(sys1);
[wn,zeta]=damp(sys1)
p1=pole(sys1)
```

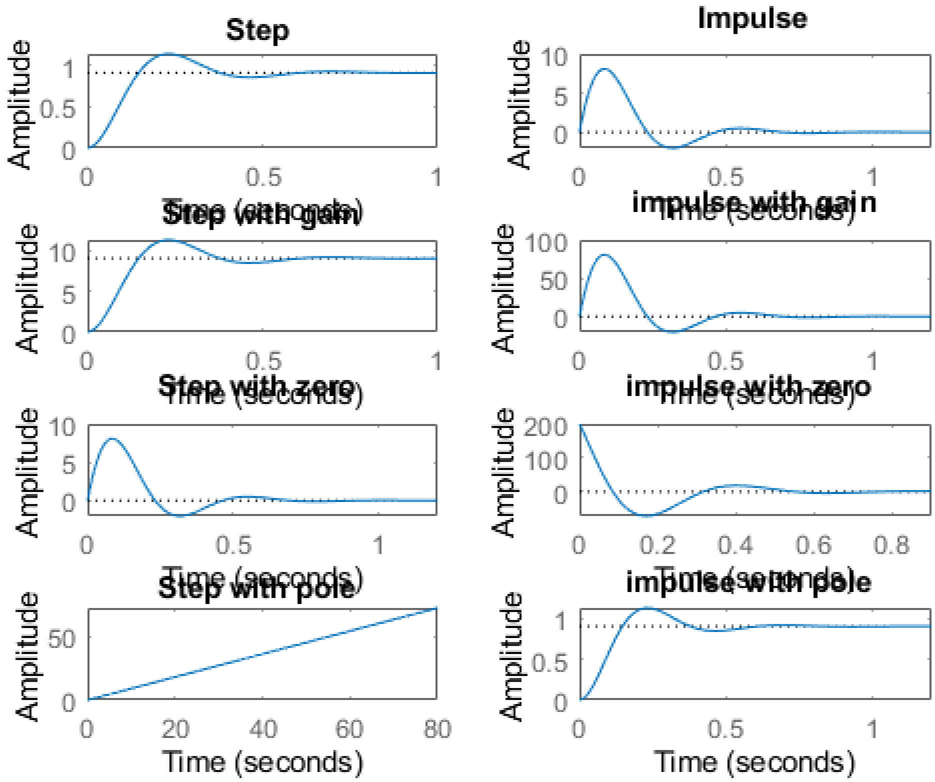
```
z1=zero(sys1)


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))]);
CF=10;
sys2 = CF*TF;
subplot(4,2,3)
step(sys2)
title("Step with gain")
subplot(4,2,4)
impulse(sys2)
title("impulse with gain")
S = stepinfo(sys2)
[wn,zeta]=damp(sys2)
p2=pole(sys2)
z2=zero(sys2)



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))]);
CF=tf([1,0],[1]);
sys3 = CF*TF;
subplot(4,2,5)
step(sys3)
title("Step with zero ")
subplot(4,2,6)
impulse(sys3)
title("impulse with zero ")
S = stepinfo(sys3)
[wn,zeta]=damp(sys3)
p3=pole(sys3)
z3=zero(sys3)




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))]);
CF=tf([1],[1,0]);
sys4 = CF*TF;
```

```
subplot(4,2,7)
step(sys4)
title("Step with pole ")
subplot(4,2,8)
impulse(sys4)
title("impulse with pole ")
S = stepinfo(sys4)
[wn,zeta]=damp(sys4)
p4=pole(sys4)
z4=zero(sys4)
```

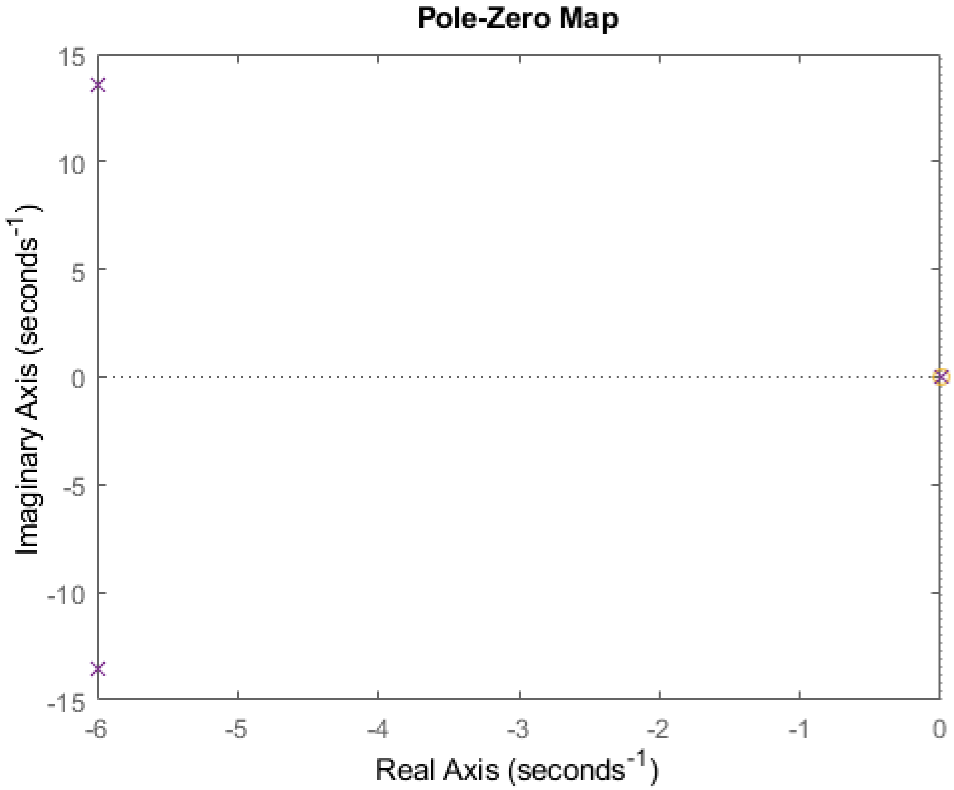
```
wn =

   14.8324
   14.8324


zeta =

    0.4045
    0.4045


p1 =

  -6.0000 +13.5647i
  -6.0000 -13.5647i


z1 =

  0×1 empty double column vector


S =

  struct with fields:

        RiseTime: 0.0993
    SettlingTime: 0.5669
     SettlingMin: 8.5269
     SettlingMax: 11.3557
       Overshoot: 24.9123
      Undershoot: 0
            Peak: 11.3557
        PeakTime: 0.2303


wn =

   14.8324
   14.8324
```

```
zeta =

    0.4045
    0.4045


p2 =

  -6.0000 +13.5647i
  -6.0000 -13.5647i


z2 =

  0×1 empty double column vector


S =

  struct with fields:

        RiseTime: 0
    SettlingTime: 0.6520
     SettlingMin: -2.0155
     SettlingMax: 8.0919
       Overshoot: Inf
      Undershoot: Inf
            Peak: 8.0919
        PeakTime: 0.0844


wn =

   14.8324
   14.8324


zeta =

    0.4045
    0.4045


p3 =

  -6.0000 +13.5647i
  -6.0000 -13.5647i


z3 =

     0
```

```
S =

  struct with fields:

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


wn =

         0
   14.8324
   14.8324


zeta =

   -1.0000
    0.4045
    0.4045


p4 =

   0.0000 + 0.0000i
  -6.0000 +13.5647i
  -6.0000 -13.5647i


z4 =

  0×1 empty double column vector
```




```
figure
hold on
pzmap(sys1)
pzmap(sys2)
pzmap(sys3)
pzmap(sys4)
```



Analysis

without gain-->Complex conjugate poles

```
% when we add differentiator,with different constant values,the
 Transfer function is settling at zero in step response

% Poles are getting changed when we change constant values

% Pole Location is not getting changed irrespective of controller
 taken into consederation

% By adding defferential controller,IVT changes from 0 to some value.
% FVT remains same for impulse response.
% by adding integrator in step response the system is not settling.
```

Published with MATLAB® R2020b