./

Individual Report – Control Systems



|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Ver. Rel. No.** | **Release Date** | **Prepared. By** | **Reviewed By** | **To be approved By** | **Remarks/Revision Details** |
| 1 | 14-04-2021 | Ravikumar M Pise |  |  | All individual scripts and comparison report |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

**Document History**

Title:Control System-First Order System Open loop with different values 3

This Document has equation for motion differential system 4

Math analysis 4

IVT 4

Analysis (SAS) 10

Title:Control System-First Order System with P,I,D 10

Changing the 3 different gains 11

For P controller 14

Analysis for P 15

For D Controller 15

Analysis for derivative 17

For I Controller 17

Title:Control System-First Order System with positive and negative feedback with P,I,D 19

Changing the gain 19

Analysis: 22

Positive feedback 22

Analysis for Positive feedback 25

Title:Control System-Second Order System:open loop with different value 27

This Document has equation for DC Motor 27

Math analysis 27

IVT 27

Analysis: 31

Control System-Second Order System:varying zeta value open system 32

Analysis 41

This Document has movement of poles for Second Order System 41

Analysis 43

This Document has movement of poles for Second Order System 43

Negtaive Feedback 46

Positive Feedback 51

Analysis 57

Analysis 63

This Document has equation for motion differential system 63

Math analysis 63

Changing the gain of system 64

gain is 0.1 65

Analysis: 66

Change the control function 66

Analysis: 70

This Document has equation for motion differential system 70

Math analysis 70

Negative feedback 70

Analysis: 74

Positive feedback 74

## Title:Control System-First Order System Open loop with different values

%Author:Ravikumar M Pise  
%PS No:99003747  
%Date:10/04/2021  
%Version:1.7

## This Document has equation for motion differential system

%Equation:mdv/dt+bv=u

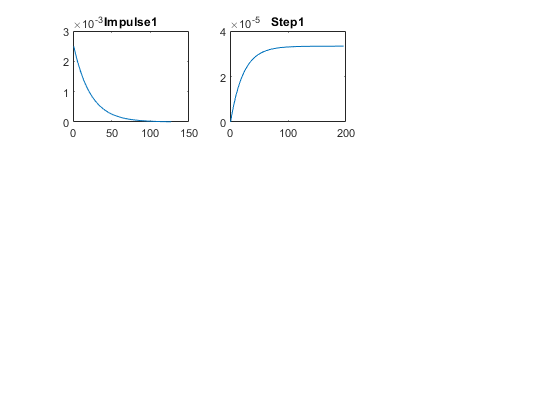
## Math analysis

%dependent variables:v  
%independent variables:t,u  
%constant:m,b  
%Root:-b/m

## IVT

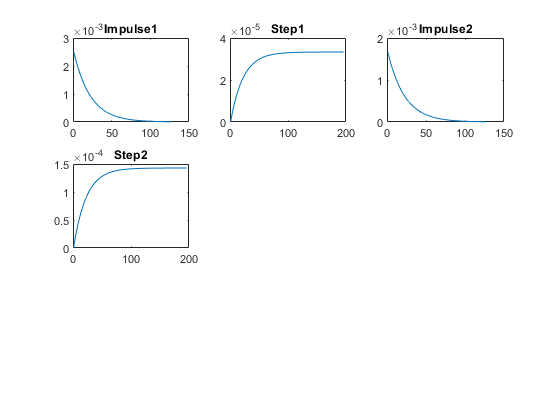
%for impulse is 1/m=0.02  
%for step is 0  
%%FVT  
%for impulse is 0;  
%for step is 1/b=2.4  
  
m=50;  
b=0.4;  
Tau=m/b;  
Transfer\_Function=tf([0,1/b],[Tau,1])  
m1=400;  
b1=30000;  
Tau=m1/b1;  
TF=tf([0,1/b1],[Tau,1])  
T\_R=4\*Tau  
subplot(3,3,1),plot(impulse(TF))  
title("Impulse1")  
subplot(3,3,2),plot(step(TF))  
title("Step1")  
S = stepinfo(TF)

Transfer\_Function =  
   
 2.5  
 ---------  
 125 s + 1  
   
Continuous-time transfer function.  
  
  
TF =  
   
 3.333e-05  
 -------------  
 0.01333 s + 1  
   
Continuous-time transfer function.  
  
  
T\_R =  
  
 0.0533  
  
  
S =   
  
 struct with fields:  
  
 RiseTime: 0.0293  
 SettlingTime: 0.0522  
 SettlingMin: 3.0150e-05  
 SettlingMax: 3.3332e-05  
 Overshoot: 0  
 Undershoot: 0  
 Peak: 3.3332e-05  
 PeakTime: 0.1406



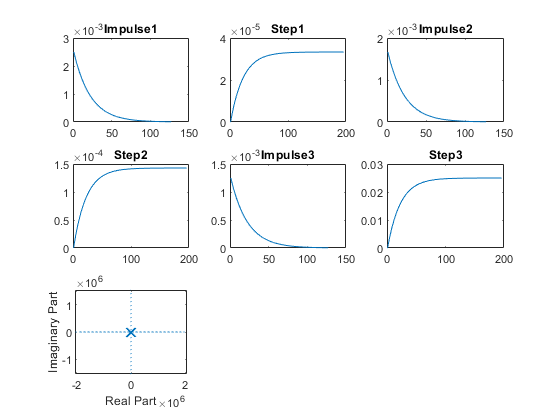
m2=600;  
b2=7000;  
Tau=m2/b2;  
T\_R=4\*Tau  
TF=tf([0,1/b2],[Tau,1])  
subplot(3,3,3),plot(impulse(TF))  
title("Impulse2")  
subplot(3,3,4),plot(step(TF))  
title("Step2")  
S = stepinfo(TF)

T\_R =  
  
 0.3429  
  
  
TF =  
   
 0.0001429  
 -------------  
 0.08571 s + 1  
   
Continuous-time transfer function.  
  
  
S =   
  
 struct with fields:  
  
 RiseTime: 0.1883  
 SettlingTime: 0.3353  
 SettlingMin: 1.2921e-04  
 SettlingMax: 1.4285e-04  
 Overshoot: 0  
 Undershoot: 0  
 Peak: 1.4285e-04  
 PeakTime: 0.9039



m3=800;  
b3=40;  
Tau=m3/b3;  
T\_R=4\*Tau  
TF=tf([0,1/b3],[Tau,1])  
subplot(3,3,5),plot(impulse(TF))  
title("Impulse3")  
subplot(3,3,6),plot(step(TF))  
title("Step3")  
S = stepinfo(TF)  
  
hold on  
  
subplot(3,3,7)  
[z1,p1,k1]= tf2zp([0,1/b1],[m1/b1,1])  
zplane(z1,p1)  
  
hold on  
  
subplot(3,3,7)  
[z2,p2,k2]= tf2zp([0,1/b2],[m2/b2,1])  
zplane(z2,p2)  
  
hold on  
subplot(3,3,7)  
[z3,p3,k3]= tf2zp([0,1/b3],[m3/b3,1])  
zplane(z3,p3)

T\_R =  
  
 80  
  
  
TF =  
   
 0.025  
 --------  
 20 s + 1  
   
Continuous-time transfer function.  
  
  
S =   
  
 struct with fields:  
  
 RiseTime: 43.9401  
 SettlingTime: 78.2415  
 SettlingMin: 0.0226  
 SettlingMax: 0.0250  
 Overshoot: 0  
 Undershoot: 0  
 Peak: 0.0250  
 PeakTime: 210.9168  
  
  
z1 =  
  
 0×1 empty double column vector  
  
  
p1 =  
  
 -75  
  
  
k1 =  
  
 0.0025  
  
  
z2 =  
  
 0×1 empty double column vector  
  
  
p2 =  
  
 -11.6667  
  
  
k2 =  
  
 0.0017  
  
  
z3 =  
  
 0×1 empty double column vector  
  
  
p3 =  
  
 -0.0500  
  
  
k3 =  
  
 0.0013



## Analysis (SAS)

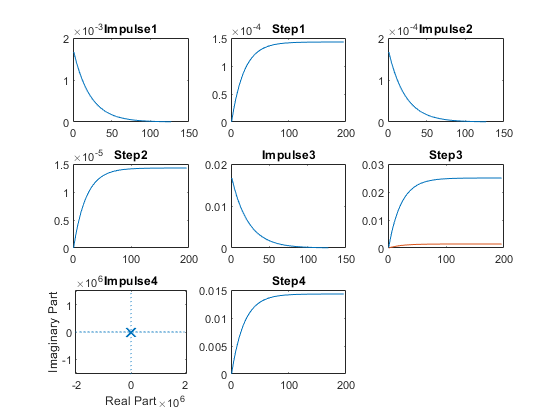
%System is stable  
%No overshoot in first order  
%Hence System 2 is Speed  
  
% Settling time is less in system 2  
%Hence System 2 is Accurate  
  
% Pole position  
%When Pole is near to s plane then less stable compared to the pole at  
%intermediate and pole far from s plane is more stable  
%P1=-75  
%tr and ts is very less  
%P2=-11.66  
%tr and ts is more compared far pole and less compared to near pole of s  
%plane  
%P3=-0.05  
%tr and ts is very high

## Title:Control System-First Order System with P,I,D

## Changing the 3 different gains

m1=600;  
b1=7000;  
Tau=m1/b1;  
TF1=tf([0,1/b1],[Tau,1]);  
T\_R=4\*Tau;  
subplot(3,3,1),plot(impulse(TF1))  
title("Impulse1")  
subplot(3,3,2),plot(step(TF1))  
title("Step1")  
S = stepinfo(TF1)  
p1=pole(TF1)  
  
  
m1=600;  
b1=7000;  
Tau=m1/b1;  
CF=0.1;  
TF2=CF\*tf([0,1/b1],[Tau,1]);  
T\_R=4\*Tau;  
subplot(3,3,3),plot(impulse(TF2))  
title("Impulse2")  
subplot(3,3,4),plot(step(TF2))  
title("Step2")  
S = stepinfo(TF2)  
p2=pole(TF2)  
  
  
m1=600;  
b1=7000;  
Tau=m1/b1;  
CF=10;  
TF3=CF\*tf([0,1/b1],[Tau,1]);  
T\_R=4\*Tau;  
subplot(3,3,5),plot(impulse(TF3))  
title("Impulse3")  
subplot(3,3,6),plot(step(TF3))  
title("Step3")  
S = stepinfo(TF3)  
p3=pole(TF3)  
  
  
m1=600;  
b1=7000;  
Tau=m1/b1;  
CF=100;  
TF4=CF\*tf([0,1/b1],[Tau,1]);  
T\_R=4\*Tau;  
subplot(3,3,7),plot(impulse(TF4))  
title("Impulse4")  
subplot(3,3,8),plot(step(TF4))  
title("Step4")  
S = stepinfo(TF4)  
p4=pole(TF4)

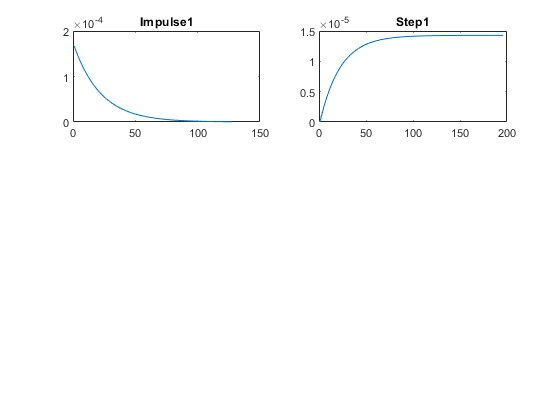
S =   
  
 struct with fields:  
  
 RiseTime: 0.1883  
 SettlingTime: 0.3353  
 SettlingMin: 1.2921e-04  
 SettlingMax: 1.4285e-04  
 Overshoot: 0  
 Undershoot: 0  
 Peak: 1.4285e-04  
 PeakTime: 0.9039  
  
  
p1 =  
  
 -11.6667  
  
  
S =   
  
 struct with fields:  
  
 RiseTime: 0.1883  
 SettlingTime: 0.3353  
 SettlingMin: 1.2921e-05  
 SettlingMax: 1.4285e-05  
 Overshoot: 0  
 Undershoot: 0  
 Peak: 1.4285e-05  
 PeakTime: 0.9039  
  
  
p2 =  
  
 -11.6667  
  
  
S =   
  
 struct with fields:  
  
 RiseTime: 0.1883  
 SettlingTime: 0.3353  
 SettlingMin: 0.0013  
 SettlingMax: 0.0014  
 Overshoot: 0  
 Undershoot: 0  
 Peak: 0.0014  
 PeakTime: 0.9039  
  
  
p3 =  
  
 -11.6667  
  
  
S =   
  
 struct with fields:  
  
 RiseTime: 0.1883  
 SettlingTime: 0.3353  
 SettlingMin: 0.0129  
 SettlingMax: 0.0143  
 Overshoot: 0  
 Undershoot: 0  
 Peak: 0.0143  
 PeakTime: 0.9039  
  
  
p4 =  
  
 -11.6667



## For P controller

figure  
  
m1=600;  
b1=7000;  
Tau=m1/b1;  
CF=0.1;  
TF5=CF\*tf([0,1/b1],[Tau,1]);  
T\_R=4\*Tau;  
subplot(3,2,1),plot(impulse(TF5))  
title("Impulse1")  
subplot(3,2,2),plot(step(TF5))  
title("Step1")  
S = stepinfo(TF5)  
p5=pole(TF5)

S =   
  
 struct with fields:  
  
 RiseTime: 0.1883  
 SettlingTime: 0.3353  
 SettlingMin: 1.2921e-05  
 SettlingMax: 1.4285e-05  
 Overshoot: 0  
 Undershoot: 0  
 Peak: 1.4285e-05  
 PeakTime: 0.9039  
  
  
p5 =  
  
 -11.6667



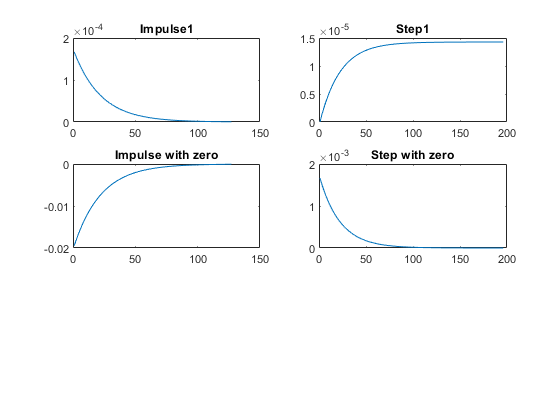
## Analysis for P

%On changing the gain of the transfer function:  
%1. Changing gain: amplitude increasing by factor of gain  
%2. Rise time and settling time remain same  
%3. peak, settling min and settling max is varying by factor of gain  
% There is no change in pole location.

## For D Controller

m1=600;  
b1=7000;  
Tau=m1/b1;  
CF=tf([1,0],[1]);  
TF6=CF\*tf([0,1/b1],[Tau,1]);  
T\_R=4\*Tau;  
subplot(3,2,3),plot(impulse(TF6))  
title("Impulse with zero")  
subplot(3,2,4),plot(step(TF6))  
title("Step with zero")  
S = stepinfo(TF6)  
p6=pole(TF6)

S =   
  
 struct with fields:  
  
 RiseTime: 0.1883  
 SettlingTime: 0.3353  
 SettlingMin: 4.3838e-08  
 SettlingMax: 1.5917e-04  
 Overshoot: Inf  
 Undershoot: 0  
 Peak: 0.0017  
 PeakTime: 0  
  
  
p6 =  
  
 -11.6667



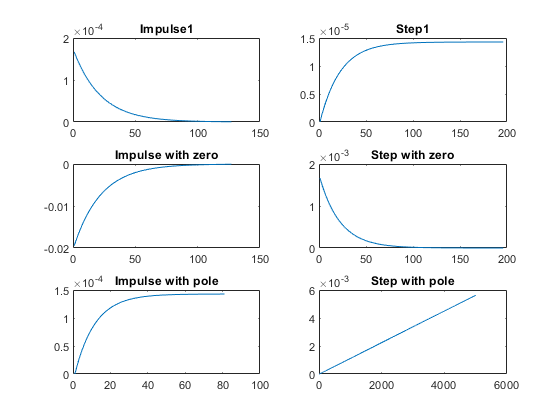
## Analysis for derivative

Pole location doesnot changes one zero added at origin tr and ts is not affected Settling minimum and max varies by adding Kd Peak time is zero Overhsoot is very high

## For I Controller

m1=600;  
b1=7000;  
Tau=m1/b1;  
CF=tf([0,1],[1,0]);  
TF7=CF\*tf([0,1/b1],[Tau,1]);  
T\_R=4\*Tau;  
subplot(3,2,5),plot(impulse(TF7))  
title("Impulse with pole")  
subplot(3,2,6),plot(step(TF7))  
title("Step with pole")  
S = stepinfo(TF7)  
p7=pole(TF7)  
  
% Analysis for I controller  
% Adding integral controler adds one more pole to system at origin  
% It makes system marginally stable  
% Adding pole at 0 makes infinte overshoot  
% It reduces the steady state error

S =   
  
 struct with fields:  
  
 RiseTime: NaN  
 SettlingTime: NaN  
 SettlingMin: NaN  
 SettlingMax: NaN  
 Overshoot: NaN  
 Undershoot: NaN  
 Peak: Inf  
 PeakTime: Inf  
  
  
p7 =  
  
 0  
 -11.6667

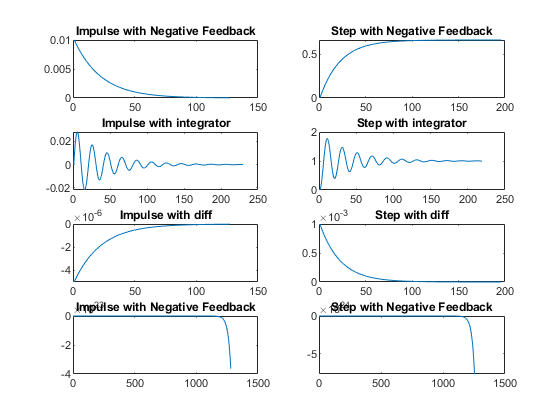


## Title:Control System-First Order System with positive and negative feedback with P,I,D

## Changing the gain

m1=1000;  
b1=5;  
Tau=m1/b1;  
CF=10;  
TF=CF\*tf([0,1/b1],[Tau,1]);  
%S = stepinfo(TF)  
NCTF1=feedback(TF,1);  
subplot(4,2,1),plot(impulse(NCTF1))  
title("Impulse with Negative Feedback")  
subplot(4,2,2),plot(step(NCTF1))  
title("Step with Negative Feedback")  
S = stepinfo(NCTF1)  
p1=pole(NCTF1)  
  
  
  
m1=1000;  
b1=5;  
Tau=m1/b1;  
CF=tf([0,1],[1,0]);  
TF=CF\*tf([0,1/b1],[Tau,1]);  
NCTF2=feedback(TF,1);  
subplot(4,2,3),plot(impulse(NCTF2))  
title("Impulse with integrator")  
subplot(4,2,4),plot(step(NCTF2))  
title("Step with integrator")  
S = stepinfo(NCTF2)  
p2=pole(NCTF2)  
  
  
m1=1000;  
b1=5;  
Tau=m1/b1;  
CF=tf([1,0],[1]);  
TF=CF\*tf([0,1/b1],[Tau,1]);  
T\_R=4\*Tau;  
NCTF3=feedback(TF,1);  
T\_R=4\*Tau;  
subplot(4,2,5),plot(impulse(NCTF3))  
title("Impulse with diff")  
subplot(4,2,6),plot(step(NCTF3))  
title("Step with diff")  
S = stepinfo(NCTF3)  
p3=pole(NCTF3)  
z2=zero(NCTF2)  
  
  
  
m1=-1000;  
b1=5;  
Tau=m1/b1;  
CF=10;  
TF=CF\*tf([0,1/b1],[Tau,1]);  
%S = stepinfo(TF)  
NCTF4=feedback(TF,1);  
subplot(4,2,7),plot(impulse(NCTF4))  
title("Impulse with Negative Feedback")  
subplot(4,2,8),plot(step(NCTF4))  
title("Step with Negative Feedback")  
%S = stepinfo(TF)  
S = stepinfo(NCTF4)  
p4=pole(NCTF4)

S =   
  
 struct with fields:  
  
 RiseTime: 146.4671  
 SettlingTime: 260.8050  
 SettlingMin: 0.6030  
 SettlingMax: 0.6666  
 Overshoot: 0  
 Undershoot: 0  
 Peak: 0.6666  
 PeakTime: 703.0560  
  
  
p1 =  
  
 -0.0150  
  
  
S =   
  
 struct with fields:  
  
 RiseTime: 35.0513  
 SettlingTime: 1.5129e+03  
 SettlingMin: 0.3925  
 SettlingMax: 1.7794  
 Overshoot: 77.9429  
 Undershoot: 0  
 Peak: 1.7794  
 PeakTime: 99.3459  
  
  
p2 =  
  
 -0.0025 + 0.0315i  
 -0.0025 - 0.0315i  
  
  
S =   
  
 struct with fields:  
  
 RiseTime: 439.8407  
 SettlingTime: 783.1973  
 SettlingMin: 2.6276e-08  
 SettlingMax: 9.5404e-05  
 Overshoot: 4.6071e+17  
 Undershoot: 0  
 Peak: 9.9900e-04  
 PeakTime: 0  
  
  
p3 =  
  
 -0.0050  
  
  
z2 =  
  
 0×1 empty double column vector  
  
  
S =   
  
 struct with fields:  
  
 RiseTime: NaN  
 SettlingTime: NaN  
 SettlingMin: NaN  
 SettlingMax: NaN  
 Overshoot: NaN  
 Undershoot: NaN  
 Peak: Inf  
 PeakTime: Inf  
  
  
p4 =  
  
 0.0150



## Analysis:

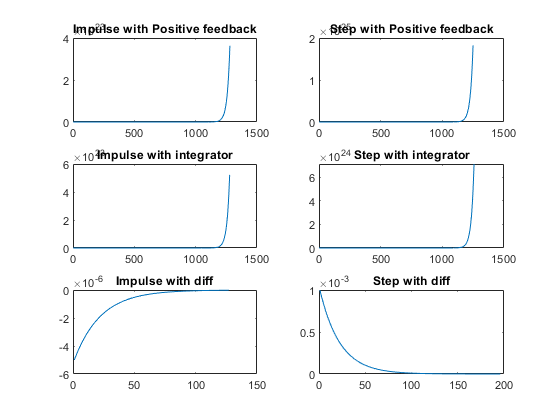
Pole location changes. system is stable By adding integrator with neagtive feedback tr reduces i.e speed of system increases ts reduces i.e accuracy of system increases steady state error reduces

% By adding differentiator with neagtive feedback  
% Rise time increases i.e speed of system decreases  
% Settling Time increase i.e accuracy of system decreases  
% overshoot increases.  
  
% By adding proportional with neagtive feedback  
% boosts the amplitude of the system

## Positive feedback

figure  
m1=1000;  
b1=5;  
Tau=m1/b1;  
CF=10;  
TF=CF\*tf([0,1/b1],[Tau,1]);  
%S = stepinfo(TF)  
PCTF1=feedback(TF,-1);  
subplot(3,2,1),plot(impulse(PCTF1))  
title("Impulse with Positive feedback")  
subplot(3,2,2),plot(step(PCTF1))  
title("Step with Positive feedback")  
S = stepinfo(PCTF1)  
p4=pole(PCTF1)  
  
m1=1000;  
b1=5;  
Tau=m1/b1;  
CF=tf([0,1],[1,0]);  
TF=CF\*tf([0,1/b1],[Tau,1]);  
PCTF2=feedback(TF,-1);  
subplot(3,2,3),plot(impulse(PCTF2))  
title("Impulse with integrator")  
subplot(3,2,4),plot(step(PCTF2))  
title("Step with integrator")  
S = stepinfo(PCTF2)  
p5=pole(PCTF2)  
  
  
m1=1000;  
b1=5;  
Tau=m1/b1;  
CF=tf([1,0],[1]);  
TF=CF\*tf([0,1/b1],[Tau,1]);  
T\_R=4\*Tau;  
PCTF3=feedback(TF,-1);  
T\_R=4\*Tau;  
subplot(3,2,5),plot(impulse(PCTF3))  
title("Impulse with diff")  
subplot(3,2,6),plot(step(PCTF3))  
title("Step with diff")  
S = stepinfo(PCTF3)  
p6=pole(PCTF3)  
z2=zero(PCTF3)

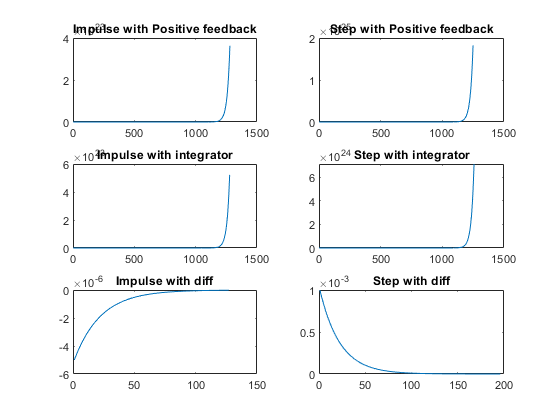
S =   
  
 struct with fields:  
  
 RiseTime: NaN  
 SettlingTime: NaN  
 SettlingMin: NaN  
 SettlingMax: NaN  
 Overshoot: NaN  
 Undershoot: NaN  
 Peak: Inf  
 PeakTime: Inf  
  
  
p4 =  
  
 0.0050  
  
  
S =   
  
 struct with fields:  
  
 RiseTime: NaN  
 SettlingTime: NaN  
 SettlingMin: NaN  
 SettlingMax: NaN  
 Overshoot: NaN  
 Undershoot: NaN  
 Peak: Inf  
 PeakTime: Inf  
  
  
p5 =  
  
 -0.0342  
 0.0292  
  
  
S =   
  
 struct with fields:  
  
 RiseTime: 438.9619  
 SettlingTime: 781.6325  
 SettlingMin: 2.6329e-08  
 SettlingMax: 9.5595e-05  
 Overshoot: Inf  
 Undershoot: 0  
 Peak: 0.0010  
 PeakTime: 0  
  
  
p6 =  
  
 -0.0050  
  
  
z2 =  
  
 0

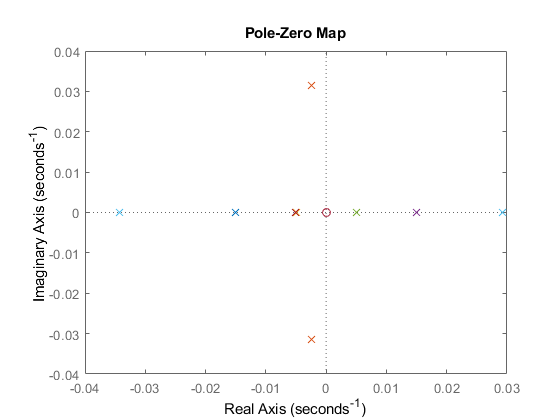


## Analysis for Positive feedback

Pole location changes. system is unstable in open loop positive feedback.

% By adding integrator with neagtive feedback  
% system is not stable, one root lies right side  
  
% on adding differentiator to positive feedback system, system is  
% becoming stable and poles got shifted to left side  
% The system is unstable in case of positive feedback with gain  
% and integrator  
  
  
figure  
hold on  
pzmap(NCTF1)  
pzmap(NCTF2)  
pzmap(NCTF3)  
pzmap(NCTF4)  
pzmap(PCTF1)  
pzmap(PCTF2)  
pzmap(PCTF3)





## Title:Control System-Second Order System:open loop with different value

## 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)

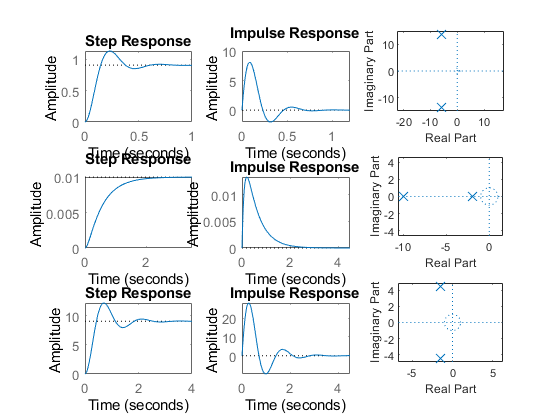
## Math analysis

%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:

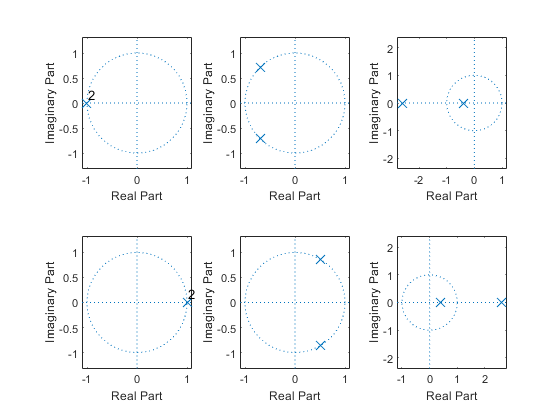
1.1st response of system is accurate because settling time is less 2.1st response speed is high because rise time is less 3.2nd response is more stable because overshoot is zero

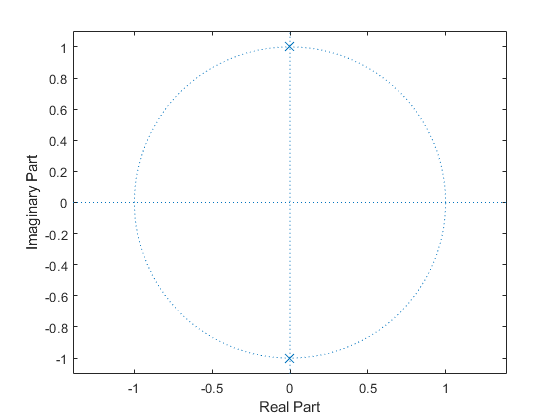
## Control System-Second Order System:varying zeta value open system

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

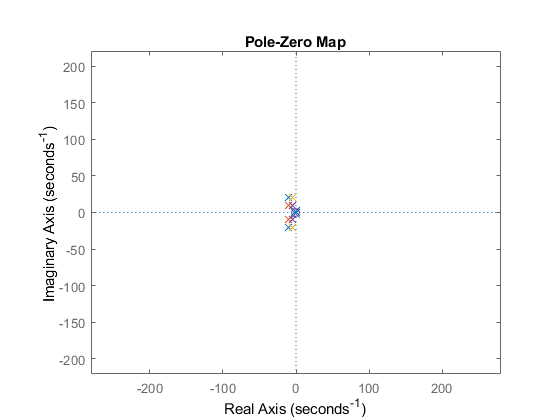
To get stable system zeta value should be greater than zero

% 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.



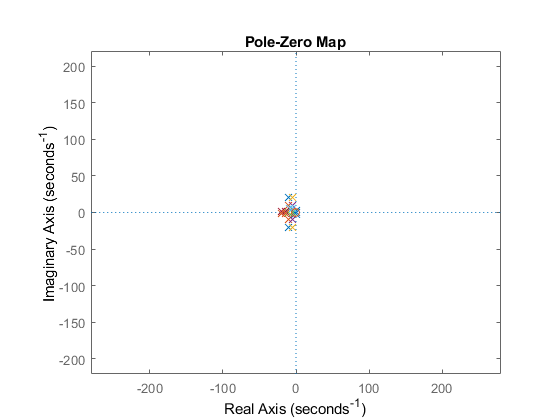
## Analysis

% 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)

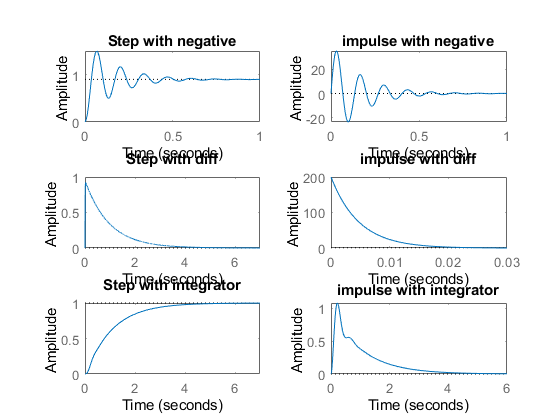
sys =  
   
 s  
 ------------  
 (s+10) (s+5)  
   
Continuous-time zero/pole/gain model.  
  
Warning: This zpk model has a complex gain or some complex zeros or poles that  
do not come in conjugate pairs.   
  
sys =  
   
 s  
 ---------------------  
 (s+(9-5i)) (s+(5-5i))  
   
Continuous-time 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

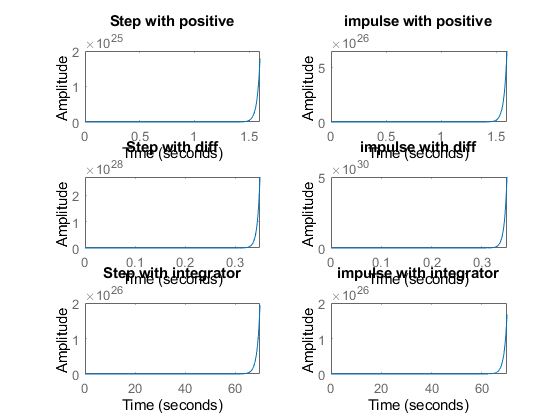
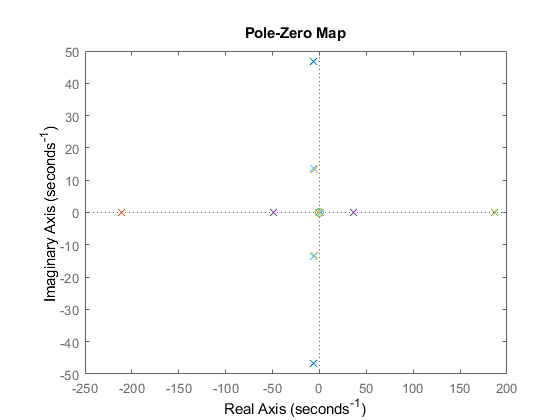


figure  
hold on  
pzmap(NCTF1)  
pzmap(NCTF2)  
pzmap(NCTF3)  
pzmap(PCTF1)  
pzmap(PCTF2)  
pzmap(PCTF3)



## 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

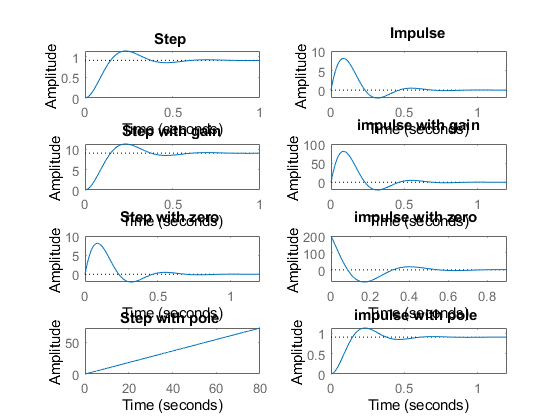
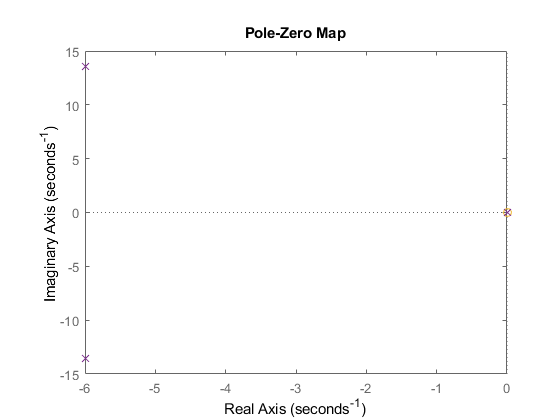


figure  
hold on  
pzmap(sys1)  
pzmap(sys2)  
pzmap(sys3)  
pzmap(sys4)



## 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

## This Document has equation for motion differential system

%Equation:v=u+(dv/dt)T

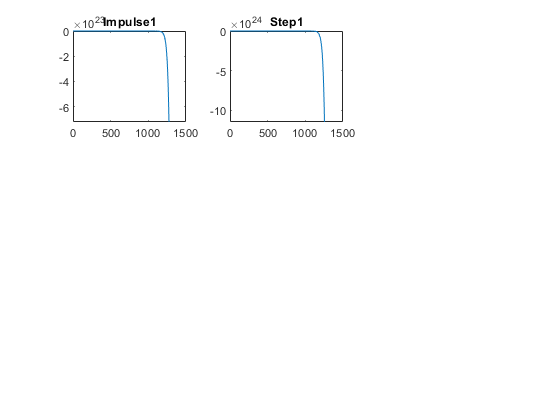
## Math analysis

%dependent variables:v  
%independent variables:t  
%constant:T  
%Root:1/T

## Changing the gain of system

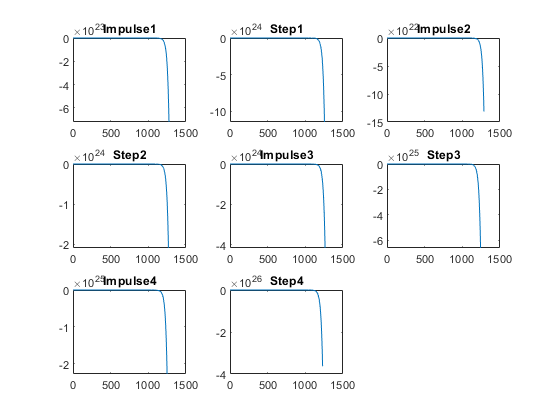
%gain is 1  
T1=40;  
Tau=T1;  
TF1=tf([0,-1],[T1,-1]);  
T\_R=4\*Tau;  
subplot(3,3,1),plot(impulse(TF1))  
title("Impulse1")  
subplot(3,3,2),plot(step(TF1))  
title("Step1")  
S = stepinfo(TF1);  
p1=pole(TF1)  
z1=zero(TF1)

p1 =  
  
 0.0250  
  
  
z1 =  
  
 0×1 empty double column vector



## gain is 0.1

T1=40;  
Tau=1/T1;  
CF=0.1;  
TF2=CF\*tf([0,-1],[T1,-1]);  
T\_R=4\*Tau;  
subplot(3,3,3),plot(impulse(TF2))  
title("Impulse2")  
subplot(3,3,4),plot(step(TF2))  
title("Step2")  
S = stepinfo(TF2);  
  
%gain is 10  
T1=40;  
Tau=1/T1;  
CF=10;  
TF3=CF\*tf([0,-1],[T1,-1]);  
T\_R=4\*Tau;  
subplot(3,3,5),plot(impulse(TF3))  
title("Impulse3")  
subplot(3,3,6),plot(step(TF3))  
title("Step3")  
S = stepinfo(TF3);  
  
%gain is 100  
T1=40;  
CF=100;  
TF4=CF\*tf([0,-1],[T1,-1]);  
T\_R=4\*Tau;  
subplot(3,3,7),plot(impulse(TF4))  
title("Impulse4")  
subplot(3,3,8),plot(step(TF4))  
title("Step4")  
S = stepinfo(TF4);



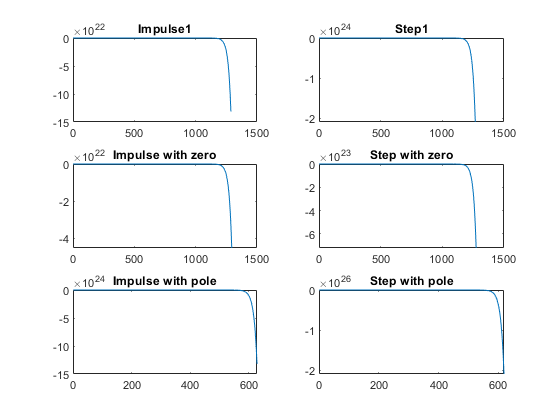
## Analysis:

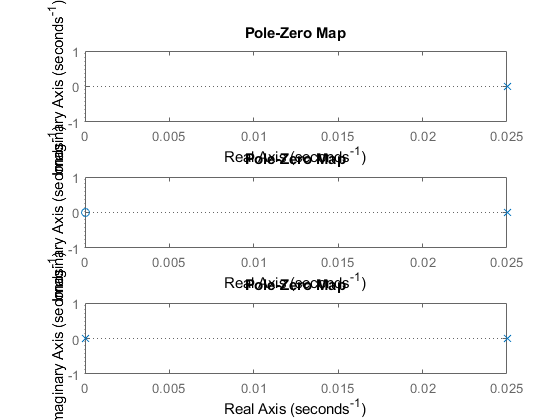
On changing the gain of the transfer function:  
Justification:  
The system is unstable whether the gain is increasing or decreasing because the pole location is at right half of S Plane  
1. all time response is NaN,Inf by this we can conclude that the system  
is not settling,speed is not known,Syatem is not accurate

## Change the control function

figure  
% system with proportion  
T1=40;  
CF=0.1;  
TF5=CF\*tf([0,-1],[T1,-1]);  
T\_R=4\*Tau;  
subplot(3,2,1),plot(impulse(TF5))  
title("Impulse1")  
subplot(3,2,2),plot(step(TF5))  
title("Step1")  
S = stepinfo(TF5)  
  
% system with differentiator  
T1=40;  
CF=tf([1,0],[1]);  
TF6=CF\*tf([0,-1],[T1,-1]);  
T\_R=4\*Tau;  
subplot(3,2,3),plot(impulse(TF6))  
title("Impulse with zero")  
subplot(3,2,4),plot(step(TF6))  
title("Step with zero")  
S = stepinfo(TF6)  
  
% system with integrator  
T1=40;  
CF=tf([0,1],[1,0]);  
TF7=CF\*tf([0,-1],[T1,-1]);  
T\_R=4\*Tau;  
subplot(3,2,5),plot(impulse(TF7))  
title("Impulse with pole")  
subplot(3,2,6),plot(step(TF7))  
title("Step with pole")  
S = stepinfo(TF7)  
  
%poles printing  
figure  
subplot(3,1,1)  
pzmap(TF5)  
subplot(3,1,2)  
pzmap(TF6)  
subplot(3,1,3)  
pzmap(TF7)

S =   
  
 struct with fields:  
  
 RiseTime: NaN  
 SettlingTime: NaN  
 SettlingMin: NaN  
 SettlingMax: NaN  
 Overshoot: NaN  
 Undershoot: NaN  
 Peak: Inf  
 PeakTime: Inf  
  
  
S =   
  
 struct with fields:  
  
 RiseTime: NaN  
 SettlingTime: NaN  
 SettlingMin: NaN  
 SettlingMax: NaN  
 Overshoot: NaN  
 Undershoot: NaN  
 Peak: Inf  
 PeakTime: Inf  
  
  
S =   
  
 struct with fields:  
  
 RiseTime: NaN  
 SettlingTime: NaN  
 SettlingMin: NaN  
 SettlingMax: NaN  
 Overshoot: NaN  
 Undershoot: NaN  
 Peak: Inf  
 PeakTime: Inf





## Analysis:

%1. Proportional: 1 pole  
%2. Differentiator:1 pole 1 zero  
%3. Integrator: 2 poles

## This Document has equation for motion differential system

%Equation:v=u+(dv/dt)T

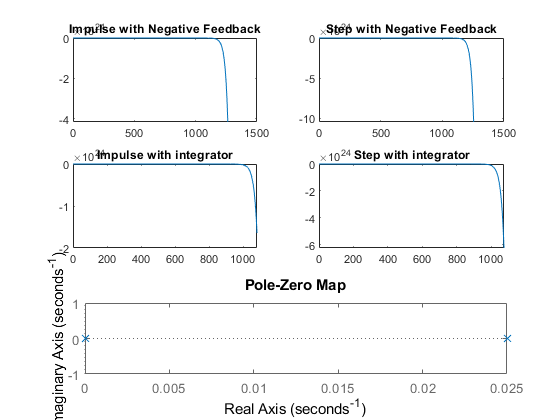
## Math analysis

%dependent variables:v  
%independent variables:t  
%constant:T  
%Root:1/T

## Negative feedback

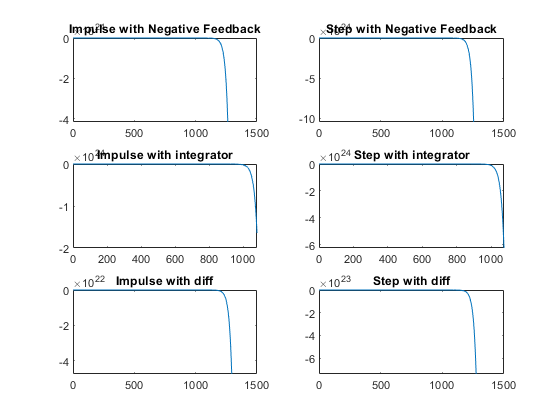
T1=40;  
Tau=1/T1;  
CF=10;  
TF=CF\*tf([0,-1/T1],[1,-Tau]);  
%S = stepinfo(TF)  
NCTF1=feedback(TF,1);  
subplot(3,2,1),plot(impulse(NCTF1))  
title("Impulse with Negative Feedback")  
subplot(3,2,2),plot(step(NCTF1))  
title("Step with Negative Feedback")  
S1 = stepinfo(NCTF1)  
p1=pole(NCTF1)  
  
T1=40;  
Tau=1/T1;  
CF=tf([0,1],[1,0]);  
TF=CF\*tf([0,-1/T1],[1,-Tau]);  
NCTF2=feedback(TF,1);  
subplot(3,2,3),plot(impulse(NCTF2))  
title("Impulse with integrator")  
subplot(3,2,4),plot(step(NCTF2))  
title("Step with integrator")  
S2 = stepinfo(NCTF2)  
p2=pole(NCTF2)  
z2=zero(NCTF2)

S1 =   
  
 struct with fields:  
  
 RiseTime: NaN  
 SettlingTime: NaN  
 SettlingMin: NaN  
 SettlingMax: NaN  
 Overshoot: NaN  
 Undershoot: NaN  
 Peak: Inf  
 PeakTime: Inf  
  
  
p1 =  
  
 0.2750  
  
  
S2 =   
  
 struct with fields:  
  
 RiseTime: NaN  
 SettlingTime: NaN  
 SettlingMin: NaN  
 SettlingMax: NaN  
 Overshoot: NaN  
 Undershoot: NaN  
 Peak: Inf  
 PeakTime: Inf  
  
  
p2 =  
  
 0.1711  
 -0.1461  
  
  
z2 =  
  
 0×1 empty double column vector



T1=40;  
Tau=1/T1;  
CF=tf([1,0],[1]);  
TF=CF\*tf([0,-1/T1],[1,-Tau]);  
T\_R=4\*Tau;  
NCTF3=feedback(TF,1);  
T\_R=4\*Tau;  
subplot(3,2,5),plot(impulse(NCTF3))  
title("Impulse with diff")  
subplot(3,2,6),plot(step(NCTF3))  
title("Step with diff")  
p3=pole(NCTF3)  
z3=zero(NCTF3)  
S3 = stepinfo(NCTF3)

p3 =  
  
 0.0256  
  
  
z3 =  
  
 0  
  
  
S3 =   
  
 struct with fields:  
  
 RiseTime: NaN  
 SettlingTime: NaN  
 SettlingMin: NaN  
 SettlingMax: NaN  
 Overshoot: NaN  
 Undershoot: NaN  
 Peak: Inf  
 PeakTime: Inf



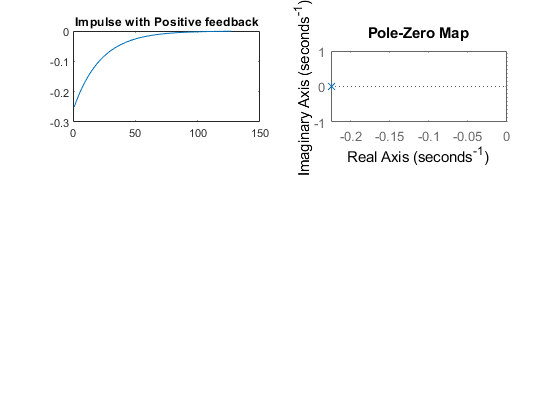
## Analysis:

1.System is becomming stable when Positive feedback is added with  
proportional controller  
Because the pole is shifting from Right half of S plane to Left Part of  
S Plane.  
Pole location is:-0.225

## Positive feedback

figure  
T1=40;  
Tau=1/T1;  
CF=10;  
TF=CF\*tf([0,-1/T1],[1,-Tau]);  
%S = stepinfo(TF)  
PCTF1=feedback(TF,-1);  
subplot(3,2,1),plot(impulse(PCTF1))  
title("Impulse with Positive feedback")  
subplot(3,2,2),plot(step(PCTF1))  
title("Step with Positive feedback")  
S = stepinfo(PCTF1)  
p4=pole(PCTF1)  
pzmap(PCTF1)

S =   
  
 struct with fields:  
  
 RiseTime: 9.7645  
 SettlingTime: 17.3870  
 SettlingMin: -1.1111  
 SettlingMax: -1.0050  
 Overshoot: 0  
 Undershoot: 0  
 Peak: 1.1111  
 PeakTime: 46.8704  
  
  
p4 =  
  
 -0.2250



T1=40;  
Tau=1/T1;  
CF=tf([0,1],[1,0]);  
TF=CF\*tf([0,-1/T1],[1,-Tau]);  
PCTF2=feedback(TF,-1);  
subplot(3,2,3),plot(impulse(PCTF2))  
title("Impulse with integrator")  
subplot(3,2,4),plot(step(PCTF2))  
title("Step with integrator")  
p5=pole(PCTF2)  
S = stepinfo(PCTF2)  
  
T1=40;  
Tau=1/T1;  
CF=tf([1,0],[1]);  
TF=CF\*tf([0,-1/T1],[1,-Tau]);  
T\_R=4\*Tau;  
PCTF3=feedback(TF,-1);  
T\_R=4\*Tau;  
subplot(3,2,5),plot(impulse(PCTF3))  
title("Impulse with diff")  
subplot(3,2,6),plot(step(PCTF3))  
title("Step with diff")  
p6=pole(PCTF3)  
z2=zero(PCTF3)  
S = stepinfo(PCTF3)  
  
%%Analysis:  
%1. on adding differentiator to positive feedback system, system is  
% becoming stable and poles got shifted to left side  
%2. The system is unstable in case of positive feedback with gain  
% and integrator  
%3. As the system is unstable in case of gain and integrator we are not  
% getting parameters, also the peak is infinite

p5 =  
  
 0.0125 + 0.1576i  
 0.0125 - 0.1576i  
  
  
S =   
  
 struct with fields:  
  
 RiseTime: NaN  
 SettlingTime: NaN  
 SettlingMin: NaN  
 SettlingMax: NaN  
 Overshoot: NaN  
 Undershoot: NaN  
 Peak: Inf  
 PeakTime: Inf  
  
  
p6 =  
  
 0.0244  
  
  
z2 =  
  
 0  
  
  
S =   
  
 struct with fields:  
  
 RiseTime: NaN  
 SettlingTime: NaN  
 SettlingMin: NaN  
 SettlingMax: NaN  
 Overshoot: NaN  
 Undershoot: NaN  
 Peak: Inf  
 PeakTime: Inf

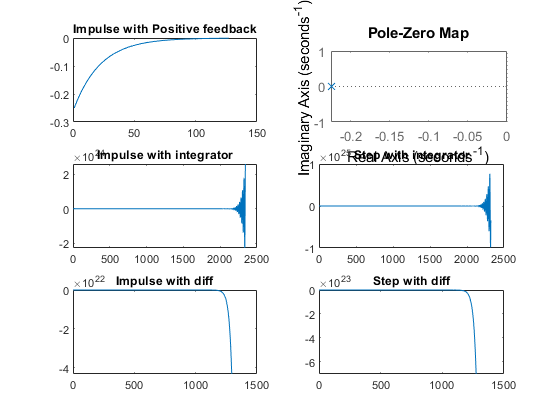
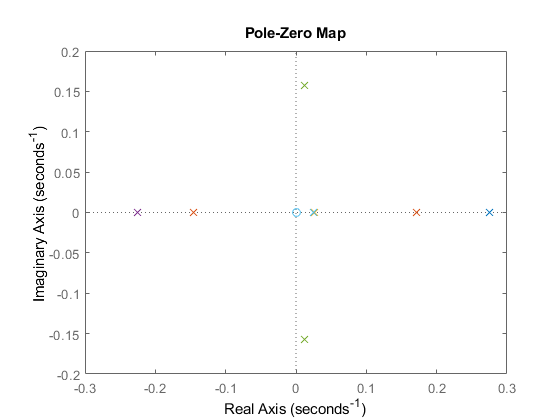


figure  
hold on  
pzmap(NCTF1)  
pzmap(NCTF2)  
pzmap(NCTF3)  
pzmap(PCTF1)  
pzmap(PCTF2)  
pzmap(PCTF3)



[Published with MATLAB® R2020b](https://www.mathworks.com/products/matlab)

# Comparison of MATLAB and SciLab

This document contains the migration of system transfer function from MATLAB to SciLab.

# MATLAB

Plant Description:

%% Title:Control System-First Order System

%Author:Sushma L Nagangoudra

%PS No:99003752

%Date:7/04/2021

%Version:1.0

%% This Document has equation for motion differential system

%Equation:mdv/dt+bv=u

%TF=v(s)/u(s)=(01/b)

%% Math analysis

%dependent variables:v

%independent variables:t,u

%constant:m,b

%Root:-b/m

%%IVT

%for impulse is 1/m=0.02

%for step is 0

%%FVT

%for impulse is 0;

%for step is 1/b=2.4

%for impulse is 0;

%for step is 1/b=0.033

m=50;

b=0.4;

Tau=m/b;

Transfer\_Function=tf([0,1/b],[Tau,1])

%%FVT

%for impulse is 1/m=0.0025

%for step is 0

%%FVT

m1=400;

b1=30000;

Tau=m1/b1;

TF=tf([0,1/b1],[Tau,1])

T\_R=4\*Tau

subplot(3,3,1),plot(impulse(TF))

title("Impulse1")

subplot(3,3,2),plot(step(TF))

title("Step1")

S = stepinfo(TF)

%%IVT

%for impulse is 1/m=0.00166

%for step is 0

%%FVT

%for impulse is 0;

%for step is 1/b=0.02

m2=600;

b2=7000;

Tau=m2/b2;

T\_R=4\*Tau

TF=tf([0,1/b2],[Tau,1])

subplot(3,3,3),plot(impulse(TF))

title("Impulse2")

subplot(3,3,4),plot(step(TF))

title("Step2")

S = stepinfo(TF)

%%IVT

%for impulse is 1/m=0.00125

%for step is 0

%%FVT

%for impulse is 0;

%for step is 1/b=0.025

m3=800;

b3=40;

Tau=m3/b3;

T\_R=4\*Tau

TF=tf([0,1/b3],[Tau,1])

subplot(3,3,5),plot(impulse(TF))

title("Impulse3")

subplot(3,3,6),plot(step(TF))

title("Step3")

S = stepinfo(TF)

hold on

subplot(3,3,7)

[z1,p1,k1]= tf2zp([0,1/b1],[m1/b1,1])

zplane(z1,p1)

hold on

subplot(3,3,7)

[z2,p2,k2]= tf2zp([0,1/b2],[m2/b2,1])

zplane(z2,p2)

hold on

subplot(3,3,7)

[z3,p3,k3]= tf2zp([0,1/b3],[m3/b3,1])

zplane(z3,p3)

%% Analysis (SAS)

% Rise time

%T1=0.02929

%T2=0.18831

%T3=44

%Hence System 2 is Speed

% Settling time

%S1=0.05216

%S2=0.335232

%S3=80

%Hence System 2 is Accurate

% Pole position

%P1=-75

%P2=-11.66

%P3=-0.05

%When pole is far from s-plane then the system is more stable (rising and

%setting time are very less)

%When pole is near from s-plane then the system is less stable (rising and

%setting time are very high)

# SciLab

Code:

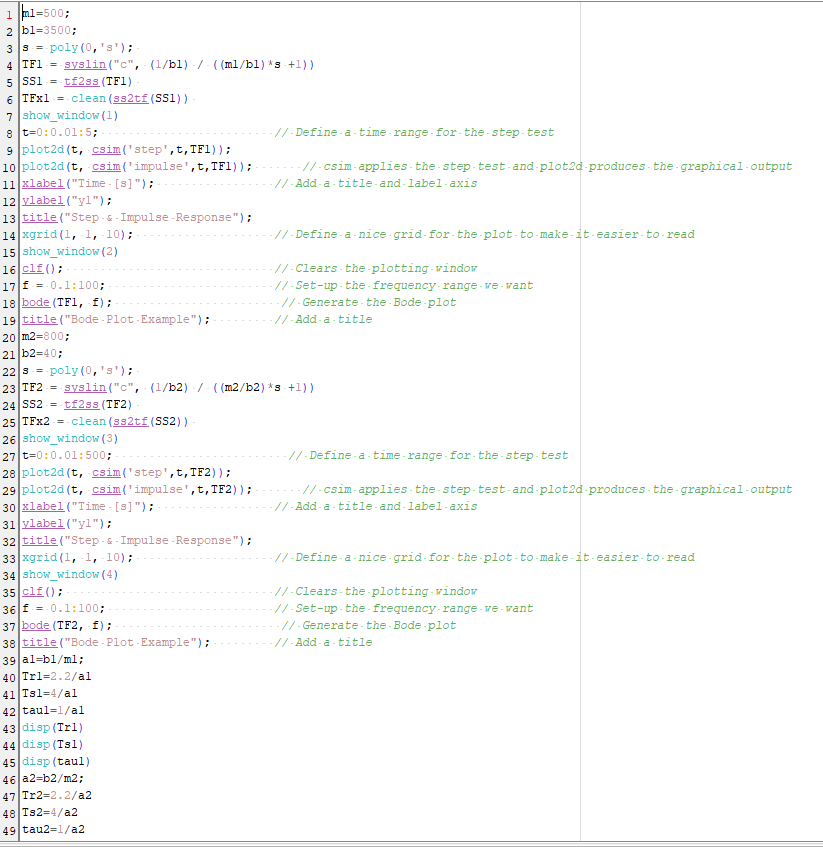


Figure 1: SciLab code for 1st order Equation of Motion

Plots:

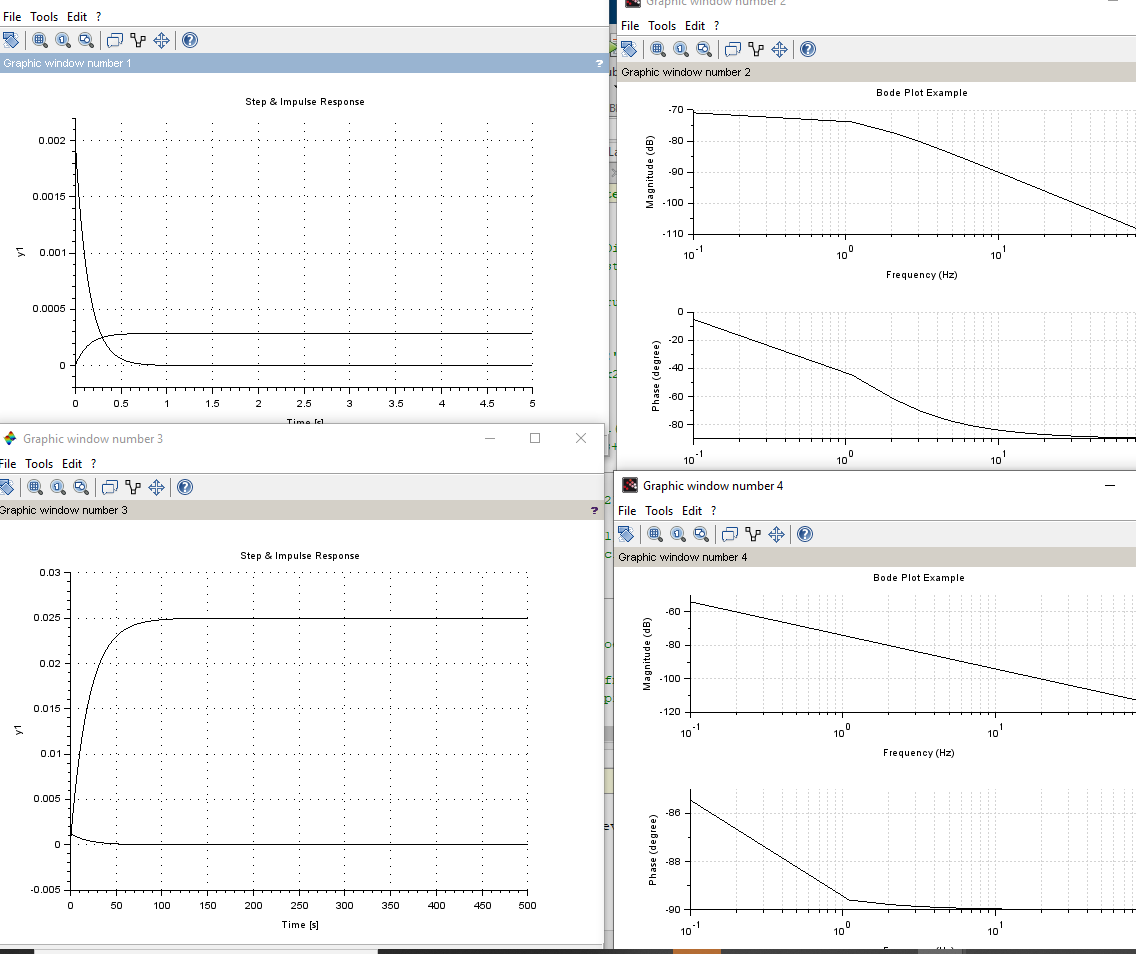


Figure 2: Plot of the 1st order Equation of Motion using SciLab.

# Analysis from the scripts:

* The position of the poles for the given transfer function remains same in both the tools.
* The frequency response and the time domain responses in the respective tool gives the same stability of the system.

The rise time, settling time and the peak value remains the same.