MANIPAL INSTITUTE OF TECHNOLOGY
MANIPAL
(A constituent unit of MAHE, Manipal)

CONTROL SYSTEM LABORATORY MANUAL

ICE 3262, VI SEM, B.TECH.

NAME: ______________________________

REG NO: ____________________________

SECTION: ___________________________

DEPARTMENT OF INSTRUMENTATION AND CONTROL ENGINEERING

February – 2023

MANIPAL INSTITUTE OF TECHNOLOGY
MANIPAL
A Constituent Unit of MAHE, Manipal.

CERTIFICATE

This is to certify that the Laboratory Manual for lab titled CONTROL SYSTEM LABAROTARY (ICE 3262) from Mr./Ms. ____________________ with Reg. No: ____________________of sixth semester of B.Tech Electronics & Instrumentation Engineering for the academic year 2022-2023 has been submitted as per laboratory course requirements, which has been evaluated and duly certified.

Place:

Date: Lab In-Charge

DEPARTMENT OF INSTRUMENTATION & CONTROL ENGINEERING

February 2023

MANIPAL INSTITUTE OF TECHNOLOGY
MANIPAL
A Constituent Unit of MAHE, Manipal.

CONTROL SYSTEM LABORATORY (ICE 3262)

Exp No.	Title	Faculty Signature
	Module -1 MATLAB based experiments	
L0	Familiarization with MATLAB	
L1	a) Block diagram reduction b) Time domain Specifications and steady state errors c)Frequency response and frequency domain Specifications	
L2	Stability analysis-Root locus, Bode Plot, Nyquist Plot	
L3	State space analysis, Controllability, observability and Pole placement	
L4	Proportional and Lag compensator design	
L5	LEAD and LAG LEAD Compensator Design	
	Module 2 Hardware based experiments	
L6	DC motor System Identification and Speed control Using DC motor trainer kit	
L7	DC Motor System Identification and Position control using DC Motor trainer kit	
L8	LAG, Lead Compensator Design	
L9	Performance characteristics of PID controller	
L10	Temperature control using PID controller	

Evaluation Plan:

Continuous Evaluation : **60%** (Preparation, Lab performance, Journal, Assignments and Regularity)

End Semester Lab Exam : **40%**

L0 - Familiarization with MATLAB Control system toolbox

AIM: To enter, display and transform to different forms of transfer function in continuous time

Problem-1

Consider a plant transfer function $G(s) = \dfrac{s^3 + 5s^2 + 4s + 6}{4s^3 + 7s^2 + 12s + 9}$

Open the editor window and a new file. Type the following MATLAB code.

%(a) To enter a transfer function

```
num=[1 5 4 6];
den=[4 7 12 9];
G1=tf(num,den)
```

Then save the file as *.m file (* is any file alphanumeric name, must be other than inbuilt function/library function name). Now open command window. Type the file name (without extension) at the command prompt and enter. File will compile and run if there is no error. If there is some error debug the code in editor and reexecute the file. To suppress execution use **%** mark in the beginning and to suppress display of result in command window put **;** at the end of the line. Continue appending following in the same file.

```
%Alternate method
s=tf('s');
G2=( s^3+5*s^2+4*s+6)/( 4*s^3+7*s^2+12*s+9)
```

%(b)Transfer function to pole-zero conversion

```
[z,p,k]=tf2zp(num,den) %z =num roots, p=den roots, k=ratio of coefficient of highest
%powered term of s in num and den that is gain of TF with normalized num and den
%polynomial.
```

%(c)To draw the pole zero plot

```
pzmap(num,den)
```

%(d)zp2tf to zero pole to transfer function

```
[num1,den1]=zp2tf(z,p,k)
```

%(e) To find the partial fraction expansion of the transfer function

```
[r,p,k]=residue(num,den)% r=residue that is constant multiplier for each factor with roots p
%and k is gain of transfer function with normalization of num and den polynomial
```

%(f) r,p,k to transfer function

```
[num2,den2]=residue(r,p,k)
```

% (g) Representation of Transfer function of 2 o/p 1 i/p system)(2X1 matrix)

```
 G11G21= tf( {1 ; [1 2 3]} , {[1 2] ; [1 3 6 11 0]}) % specifies the two-output, one-input
%transfer function
```

%Result: Transfer function from output 1 to input1 G11 and output 2 to input1 G21 are %respectively

```
       1
 #1:  -----
     s + 2

          s^2 + 2 s + 3
 #2:  ---------------------------
     s^4 + 3 s^3 + 6 s^2 + 11 s
```

Problem 2

Consider a controller $G_1(s) = \dfrac{s+1}{s+2}$ and plant $G_2(s) = \dfrac{1}{500s^2}$ as two system transfer functions. Obtain the overall transfer function of the system when they are connected in a) cascade b) parallel c) feedback

% (a)cascaded system when both controller and plant are in series in the forward path

```
numg1=[1];deng1=[500 0 0];numg2=[1 1];deng2=[1 2];
[nums,dens]=series(numg1,deng1,numg2,deng2)
printsys(nums,dens)
```

%Alternatively

```
s=tf('s');
G1=1/(500*s^2);G2=(s+1)/(s+2);
GC=G1*G2
```

%Alternatively

```
[nums]=conv(numg1,numg2)
[dens]=conv(deng1,deng2)
syss=tf(nums,dens)
```

%(b)Parallel system when plant and controller are in parallel forward paths

```
[nump,denp]=parallel(numg1,deng1,numg2,deng2)
printsys(nump,denp)
```

%Alternatively

```
GP=G1+G2
```

%(ci)-ve unity feedback systems with G1 and G2 are in series in forward path

```
[ncl,dcl]=feedback(nums,dens,1,1,-1)
printsys(ncl,dcl)
%Alternatively
[ncl,dcl]=cloop(nums,dens)
printsys(ncl,dcl)
```

%(cii) +ve unity feedback systems with plant and controller are in series in forward path

```
[nclp,dclp]=feedback(numc,denc,1,1,+1)
printsys(nclp,dclp)
```

%Alternatively

```
[nclp,dclp]=cloop(numc,denc,+1)
printsys(nclp,dclp)
```

% (ciii) feedback system with G2 in forward path and G1 in feedback path with
%-ve feedback

```
[nclgh,dclgh]=feedback(numg2,deng2,numg1,deng1,-1)
printsys(nclgh,dclgh)
```

Exercise 1:

Consider the transfer function G(s)=(6s^2+1)/(s^3+3s^2+3s+1) and H(s)=(s+1)(s+2)/(s+2i)(s-2i)(s+3). Perform the following (i) Compute the poles and zeros of G(s) (ii) Express H(s) as a ratio of two polynomials in s (iii) Obtain G(s)/H(s), also find its pole zero plot.

```
% (i) compute the poles and zeros of G(s)
numg=[6 0 1];deng=[1 3 3 1];
Z=roots(numg); P=roots(deng)%z are zeroes and P are poles , here k=6.

%(ii)Express H(s) as a function of two polynomials
n1=[1 1]; n2=[1 2]; d1=[1 2*i]; d2=[1 -2*i]; d3=[1 3];
numh=conv(n1,n2);
denh=conv(conv(d1,d2),d3);
H=tf(numh,denh)

%(iii)Obtain G(s)/H(s),also find its pole zero plot.
num=conv(numg,denh);
den=conv(deng,numh)
printsys(num,den)
pzmap(num,den)
```

Additional Exercise: Solve the following using appropriate Matlab code.

Exercise 2: Consider the two polynomials $p(s)=s^2+2s+1$ and $q(s)=s+1$, compute the following using MATLAB.

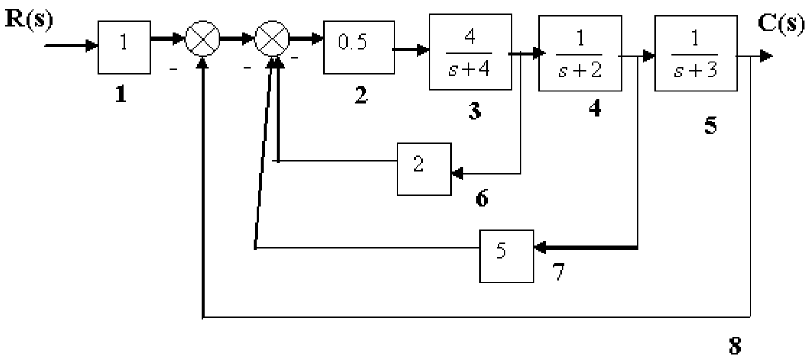
i) $p(s)q(s)$ ii) poles and zeros of $G(s)=\dfrac{q(s)}{p(s)}$ iii) $p(-1)$

Exercise 3: Obtain the factored form and pole zero plot of the following transfer function

$$G(s)=\frac{s^2+6s+8}{s^4+8s^3+12s^2+16s+20}$$

L1a - Block diagram reduction

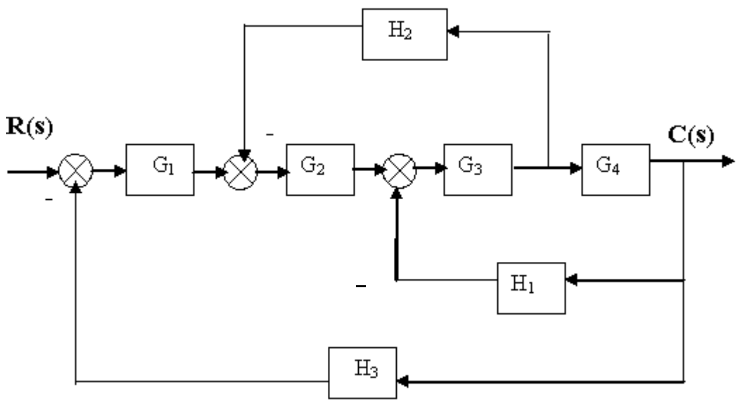
Exercise 1
% Find the overall transfer function of the block diagram shown below.




```
n1=1;d1=1;n2=0.5;d2=1;n3=4;d3=[1  4];n4=1;d4=[1 2];n5=1;d5=[1 3];
n6=[2];d6=1;n7=5;d7=1;n8=1;d8=1 ;
nblocks=8
blkbuild
q=[1 0 0 0 0; 2 1 -6 -7 -8; 3 2 0 0 0; 4 3 0 0 0; 5 4 0 0 0; 6 3 0 0 0; 7 4 0 0 0; 8 5 0 0 0];
 iu=[1]; iy=[5]; [A B C D]=connect(a,b,c,d,q,iu,iy);
 [num,den]=ss2tf(A,B,C,D);
[ncl,dcl]=minreal(num,den)
 sys1=tf(ncl,dcl)
```

% EXERCISE: **By block diagram reduction technique obtain the overall transfer function of the system**

$G_1(s)=\dfrac{1}{s+10}$; $G_2(s)=\dfrac{1}{s+1}$; $G_3(s)=\dfrac{s^2+1}{s^2+4s+4}$; $G_4(s)=\dfrac{s+1}{s+6}$; $H_1(s)=\dfrac{s+1}{s+2}$;
$H_2(s)=2$; $H_3(s)=1$.




```
%MATLAB CODE
n1=1;d1=1;n2=[1];d2=[1 10];n3=[1];d3=[1 1];n4=[1 0 1];d4=[1 4 4];n5=[1 1];d5=[1 6];
n6=[1 1];d6=[1 2];n7=[2];d7=[1];n8=[1];d8=[1];
nblocks=8;
```

```
blkbuild;
q=[1 0 0;2 1 -8;3 2 -7;4 3 -6;5 4 0;6 5 0;7 4 0;8 5 0]; %Max. connection to a block is 2
iu=1;iy=5; %iu=input to block, iy=output from block
[A B C D]=connect(a,b,c,d,q,iu,iy);
 [num,den]=ss2tf(A,B,C,D);
cltf=tf(num,den)
```

L1b: Time domain specifications and steady state errors

Problem 1:

Obtain the unit step response of the system whose closed loop transfer function is given by $\frac{C(s)}{R(s)} = \frac{25}{s^2 + 4s + 25}$

Also obtain the time domain specifications i) Maximum overshoot, ii) peak time, iii) rise time, iv) delay time, v) settling time and verify result theoretically.

```
%step response of a system
num=[0 0 25]; den=[1 4 25];
figure(1);
step(num,den)%plots step response of the system
title('unit step response of G(s)=25/s^2+4s+25)')
xlabel('t Sec'); ylabel('output')
grid
%On plot right click mouse and select characteristics option to see time response
%specifications
```

Problem 2

Obtain the impulse response of the closed loop transfer function $\frac{C(s)}{R(s)} = \frac{5s^2 + 3s + 6}{s^3 + 6s^2 + 11s + 6}$.

```
% Impulse response
num=[5 3 6];
den=[1 6 11 6];
figure(1)
impulse(num,den)% impulse response of original system
grid; title('unit impulse response')
xlabel('t sec')
ylabel('output response')
```

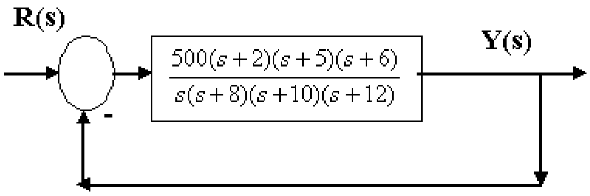
Problem 3:

% Compute the ramp response of unity negative feedback closed loop system with forward path transfer functions $G_c = \frac{s+0.6}{s+1}$ and $G=\frac{70}{s(s^2+7s+10)}$. Also calculate the steady state error.

```
t=[0:0.01:25];u=t; %u is defined as unit ramp signal
numgc=[1 0.6];dengc=[1 1];
numg=[70];deng=[1 7 10 0];
```

```
[numa,dena]=series(numgc,dengc,numg,deng);
[num,den]=cloop(numa,dena);%closed loop transfer function with unity –ve feedback
[y,x]=lsim(num,den,u,t);%computes time response for a given input  and given t.
plot(t,y,t,u)
grid; xlabel('Time[Sec]'),ylabel('C(t)')
kv=dcgain(conv([1 0],numa),dena);%Velocity(ramp) error coefficient
ess=1/kv %zoom on;
%[a b]=ginput(2)%place and click mouse on same time point on r and y plot respectively
%ess=b(2)-b(1) %difference between y and r at a given time
```

Problem 4



%Find the steady state errors

```
numg=500*poly([-2 -5 -6])
deng=poly([0 -8 -10 -12])
G=tf(numg,deng);
```

%check stability

```
T=feedback(G,1);
poles=pole(T)
```

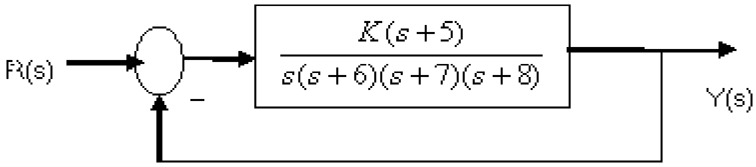
%step input

```
kp=dcgain(G)%Position(step) error coefficient
ess=1/(1+kp)
%ramp input
numsg=conv([1 0],numg);
sG=tf(numsg,deng)
sG=minreal(sG)
Kv=dcgain(sG)%Velocity error coefficient
ess=1/Kv
```

%parabolic input

```
nums2g=conv([1 0 0],numg)
s2G=tf(nums2g,deng)
s2G=minreal(s2G)
Ka=dcgain(s2G)%acceleration(parabolic) error coefficient
ess=1/Ka
```

Problem 5: %Gain design to meet a steady state error specification




```
%consider with K=1
numg=[1 5];
deng=poly([0 -6 -7 -8]);
G=tf(numg,deng);
numgkv=conv([1 0],numg);
GKv=tf(numgkv,deng)
GKv=minreal(GKv);
Kvi=dcgain(GKv)
ess=0.1%desired steady state velocity error
K=1/(ess*Kvi)
T=feedback(K*G,1)
poles=pole(T)
```

Exercise 1:

Obtain the unit step responses of the following systems whose closed loop transfer function are given by $\frac{C(s)}{R(s)} = \frac{1}{s^2 + 0.5s + 1}$ and $\frac{C(s)}{R(s)} = \frac{1}{s^2 + 0.5s + 4}$. Plot their step responses curves in one figure window.

%program

```
clc; figure(1)
t=0:0.1:20
num1=[0 0 1]; den1=[1 0.5 1]; num2=[0 0 1]; den2=[1 0.5 4];
[y1,x1,t]=step(num1,den1,t)
[y2,x2,t]=step(num2,den2,t)
plot(t,y1,'g',t,y2,'r')
xlabel('t Sec')
ylabel('outputs y1 and y2')
title('Step response of two systems')
```

Exercise 2: For the following closed loop transfer functions obtain the step/impulse/ramp responses of the given system. ii) And also obtain the time domain specification from step response and verify result theoretically.

$$\frac{C(s)}{R(s)} = \frac{w_n^{\ 2}}{s^2 + 2\xi w_n s + w_n^{\ 2}}$$,where $\xi = 0.6$ and $w_n = 8$.

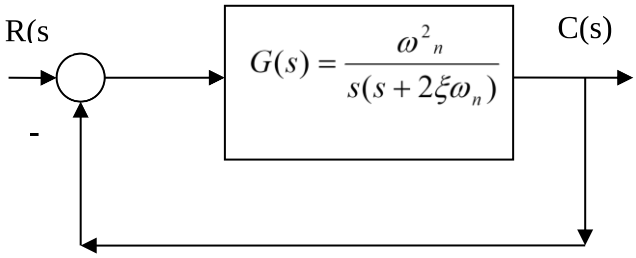
Exercise 3: Determine the step, ramp, and parabolic error constants and the steady state error for the unity negative feedback control system with following open loop transfer function. Also plot the respective time domain responses.

$$G(s) = \frac{5s^2 + 2s + 1}{s(s+2)(s+8)(s+45)}$$

L1c: Frequency response and Frequency domain analysis

Aim: To determine the frequency response of a second-order system and evaluate the frequency domain specifications.

Theory:



Consider the unity negative feedback closed loop system having open loop transfer function G(s)= $\frac{\omega_n^2}{s(s+2\xi\omega_n)}$.

Choose appropriate value of ω_n and ξ. Choose ξ to be <0.707.Obtain the closed loop transfer function $\frac{C(s)}{R(s)} = \frac{G(s)}{1+G(s)} = \frac{\omega_n^2}{s^2+2\xi\omega_n s+\omega_n^2}$. This is a typical second order system.

To obtain the frequency response of the above system and to obtain the frequency response specifications, following Matlab Code may be used.

%Matlab Code:

```
clear all
close all
wn=input('enter the natural frequency of oscillation wn=');
zeta=input('enter the damping ratio zeta='); %zeta <0.707
n=wn^2;
d=[1    2*zeta*wn      wn^2];
sys=tf(n,d);%given closed loop second order transfer function is defined
w=logspace(-2,2,50);% frequency range between 0.001 & 100 with 50 logarithmically spaced
%points.
h=freqresp(sys,w);%evaluates complex value of system at every frequency point
reh=real(h(:));%extracts real part at every frequency
imh=imag(h(:));%extracts imaginary part at every frequency
magh=sqrt(reh.^2+imh.^2);%magnitude in absolute value
%Alternatively
mh=abs(h)%
maga=mh(:);%magh=maga
magdb=20*log10(magh);
phh=atan2(imh,reh);%quadrantwise angle computed in radians
%alternatively
ph=angle(h);
ph2=ph(:);%ph2=phh
phd=phh*180/pi;
C=[w',magdb,phd];
```

```
display(C)
subplot(2,1,1),semilogx(w,magdb);
grid on,
zoom on
subplot(2,1,2),semilogx(w,phd);
grid on,
zoom on
%To read the value of resonant frequency and resonant peak from plot use
[wr,mr]=ginput(1)
%To read bandwidth
[wb,mb]=ginput(1)
%theoretical Verification
```

%Mr= 20*log10($\frac{1}{2\xi\sqrt{1-\xi^2}}$); $\omega_r = \omega_n\sqrt{1-2\xi^2}$;ω_b=ω_n*sqrt((1-2*ξ^2)+sqrt(4*ξ^4-4*ξ^2+2))

Result: From the plot obtain the resonance peak M_r, resonance frequency ω_r and the bandwidth. Compare these values with the theoretically obtained values.

Resonace Peak=M_r= db

Resonance frequency=$\omega_r =$ rad/sec

Bandwidth= ω_b= rad/sec

Exercise 1:

Using Matlab command find the magnitude and phase angle of the system with transfer function $G(s) = \frac{5}{s+2}$ at $\omega = 3 rad/\sec$

```
%Matlab code
GNUM=[5];GDEN=[1 2]; s=3*j;
Gj3=polyval(GNUM,s)/polyval(GDEN,s)%complex function evaluation at a given complex
%value
magGj3=abs(Gj3)
phaseGj3=angle(Gj3)*180/pi
```

Exercise 2

Consider the standard second order transfer function $M(s) = \frac{\omega_n^2}{s^2 + 2\xi\omega_n s + \omega_n^2}$**. By defining the normalized frequency** $\omega_v = \frac{\omega}{\omega_n}$**, plot the frequency response for various values of** $\xi =$ **0.25, 0.5, 0.707,1.**

%Matlab Code

```
wv=0:0.05:3;
z=[0.25 0.5 0.707 1];
for k=1:4
   mnum=[0 0 1];mden=[1 2*z(k) 1];
```

```
    mjomega=freqs(mnum,mden,wv);
    mmag=abs(mjomega);
    plot(wv,mmag)
gtext(‘z(k)’)
    title('frequency response of M(S)')
    xlabel('omegav');
    ylabel('|M(jwv)|');
    hold on
end
gtext('\zeta=0.25')%click on the plot to label plot with zeta=0.25
gtext('\zeta=0.5')
gtext('\zeta=0.707')
gtext('\zeta=1')
grid;
```

L2: STABILITY ANALYSIS

Theory: For Minimum phase system, both gain margin and phase margin should be positive for closed loop stability.

Nyquist stability criteria:
Using Polar plot: -1+j0 point in GH plane is not enclosed by polar plot, then the closed loop system is stable.
Using Nyquist plot: If the minimum phase system has no open loop poles on RHS of s-plane, then the number of encirclements of -1+j0 point should be zero.
If the minimum phase open loop system has p number of open loop poles on RHS of s-plane, then the number of encirclements of -1+j0 point in GH plane should be p in clock wise direction(+). That is, z=p-n=0, implying there is no RHS zero of characteristic equation in RHS of s plane. Else, that is if p is not equat to zero and n=0, or n<p, or if p=0 and n is in counter clock wise direction, the closed loop system is unstable.
Using Root locus: The closed loop system is stable for a range of open loop gain, for which the closed loop poles all lie on LHS of s-plane. The value of K corresponding to imaginary axis crossing of root locus, gives marginal stability and corresponding imaginary axis crossisng frequency, gives frequency of sustained oscillation. If K is set above this critical value, closed loop system becomes unstable.

Problem 1
Determine the stability of the negative unity feedback system whose open loop transfer function is given by $G(s)=\dfrac{1.25}{s(s+1)(0.5s+1)}$

```
%Stability analysis
a=[1.25]; b=[1 1 0]; c=[0.5 1];
d=conv(b,c)
figure(1); bode(a,d)
margin(a,d)
figure(2); nyquist(a,d)
figure(3); rlocus(a,d)
```

Result: Using Bode plot:
Gain cross over frequency
Phase crossover frequency
Gain at phase cross over frequency
Gain margin
Phase at Gain cross over frequency
Phase margin:
Comment on closed loop stability…

Using Nyquist plot:

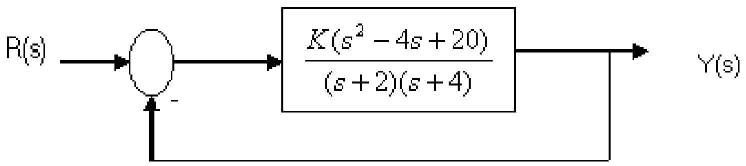
Using Root locus:
Break away point:

Gain at Break away point
Imaginary axis corssing point= Frequency of sustained oscillation=
Gain at imaginary axis crossing point=

Comment on stability and relative stability:

Problem 2
Sketch the root locus for the system shown in figure and find the following

a) The exact point and gain where the locus crosses the 0.45 damping ratio line.
b) The exact point and gain where the locus crosses the imaginary axis.
c) The breakaway point on the real axis.
d) The range of K within which the system is stable.




```
numgh=[1 -4 20];
dengh=poly([-2 -4]);
GH=tf(numgh,dengh)
figure(1); rlocus(GH)
z=0.2:0.05:0.5;
wn=0:1:10
sgrid(z,wn)
figure(2); rlocus(GH)
axis([-3 1 -4 4])
z=0.45;wn=0;sgrid(z,wn)
for k=1:3
[k,p]=rlocfind(GH)
end
```

Program 3 : Draw Bode diagram for TF of G(s)= $\frac{e^{-2s}}{s(s+1)}$ **and determine the stability**

```
nn=30; a=-1;b=1;
w=logspace(a,b,nn)
Gnum=[0 0 1];Gden=[1 1 0];
[mag,phase,w]=bode(Gnum,Gden,w);
db=20*log10(mag);
phased1=(-2)*57.296*w;
phase=phase+phased1;
subplot(211), semilogx(w,db)
title('BODE DIAGRAM')
xlabel('frequency');
ylabel('db');grid
subplot(212),semilogx(w,phase)
xlabel('frequency');ylabel('phase');grid
```

Program 4

Obtain the polar and Nyquist plot of the system having its open loop transfer function $G(s)H(s)=\frac{k}{(s)(s+0.5)(s+2)}$; take k=1, 10. Also determine the gain cross over frequency, phase margin, phase cross over frequency and gain margin. Verify the Nyquist theorem and validate the result by determining the roots of the characteristic equation using MATLAB.

```
%to obtain polar/Nyquist plot
close all; clear all;
k=input('enter the value of open loop gain k=')
pol=input('enter three open loop poles [p1,p2,p3]')
n=k; d=poly(pol);
sol=tf(n,d); %open loop transfer function
[re,im,w]=nyquist(sol);
mag=sqrt(re(:).^2+im(:).^2)
magdb=20*log10(mag);
phd=(180/pi)*atan2(im(:),re(:));
c=[w,magdb,phd];
display(c)
plot(re(:),im(:));%plots polar plot
figure(2), plot(re(:), im(:), re(:), -im(:));%plots Nyquist plot
[x,y]=scircle1(0,0,1);
hold on, plot(x,y) %plots unit circle
%to plot Nyquist plot
figure(3), nyquist(n,d)
oltf=tf(n,d);
cltf=feedback(oltf,1); %to get closed loop transfer function
[ncl,dcl]=tfdata(cltf,1);
clpoles=roots(dcl);
```

Specimen Calculation: Using [regc,imgc]=ginput(1) and [repc,impc]=ginput(1) on command window, calculate phase angle and gain at gain cross over and phase cross over frequencies and hence the margins. Or using the matrix 'c' frequency at cross over and gain and phase at respective points can be calculated. Note: If the angle is positive at gain cross over frequency, subtract 360 from displayed angle to obtain actual angle.

Additional Exercises:

Exercise 1: Obtain the Nyquist plot and determine the stability of the following unity feedback systems with open loop transfer functions are i) $G(s)=\frac{15(s+5)}{s(s+2)(s^2+6s+15)}$ ii) $G(s)=\frac{12(s+2)}{s^2-6s+10}$

Exercise 2: Obtain the root locus plot for K>0, the following unity feedback systems with open loop transfer functions are i) $G(s)=\frac{K}{s(s+4)(s+8)}$ ii) $G(s)=\frac{K(s+7)}{(s+2)(s^2+2s+3)}$. Also find the range of K for which the systems are stable. Also find the break-away points.

L3a: State space analysis

Problem 1

Aim: To represent transfer function in state space form and vice versa. Also to obtain step response of SISO and MIMO systems

Consider a plant transfer function $G(s) = \dfrac{s^3 + 5s^2 + 4s + 6}{4s^3 + 7s^2 + 12s + 9}$

% transfer function to state space conversion

```
num1=[1 5 4 6];
den1=[4 7 12 9];
[A,B,C,D]=tf2ss(num1,den1)
```

%state space to transfer function

```
[num3,den3]=ss2tf(A,B,C,D)
%step response from state space model
figure(1)
step(A,B,C,D)

%To Obtain the transfer function of a MIMO (2 o/p, 2 i/p) system 2X2 matrix
A=[0 1;-25 -4]; B=[1 1;0 1]; C=[1 0;0 1] D=[0 0;0 0];
[num1,den1]=ss2tf(A,B,C,D,1)% gives y1(s)/u1(s) and y2(s)/u1(s)  with u2 zero
[num2,den2]=ss2tf(A,B,C,D,2)%Gives y1(s)/u2(s) and y2(s)/u2(s) with u1 zero
```

%Step Response curves of a MIMO system.

```
A=[-1 -1;6.5 0];
B=[1 1;1 0];
C=[1 0;0 1];
D=[0 0;0 0]
figure(2)
[y1 y2 t]=step(A,B,C,D,1);
plot(t,y1,t,y2)
grid
title('step Response plots:Input=u1(u2=0)')
text(3.4,-0.06,'Y1')
text(3.4,1.4,'Y2')
figure(3)
[y1 y2 t]=step(A,B,C,D,2);
plot(t,y1,t,y2)
grid
title('step Response plots:Input=u2(u1=0)')
text(3,0.14,'Y1')
text(2.8,1.1,'Y2')
```

% To find the eigen value

```
eig(A)% Roots of denominator polynomial of transfer function. All –ve system stable.
```

Problem 2) Diagonalization of A matrix in Controllable canonical form

A linear time invariant system is described by the following constant matrices

```
A=[0 1 0;0 0 1;-6 -11 -6];
B=[0;0;2];C=[1 0 0];
```

D=[0];
Determine the following using matlab commands.

(i) Transform the above state model in to Jordan canonical form $\hat{A}, \hat{B}, \hat{C}, \hat{D}$

(ii) Show that eigen values of A are equal to $\hat{A}$

(iii) Determinant of A is equal to determinant of $\hat{A}$

(iv) The trace of A is equal to trace of $\hat{A}$

(v) The transfer functions of A,B,C,D = transfer functions of $\hat{A}, \hat{B}, \hat{C}, \hat{D}$

MATLAB Code:

```
L=eig(A)
P=[1 1 1;L(1) L(2) L(3);L(1)^2 L(2)^2 L(3)^2];%Van der monde transformation matrix for
%transformation of A in controllable canonical form, to diagonal form for distinct root case
PI=inv(P);
ASTILDA=PI*A*P
BSTILDA=PI*B
CSTILDA=C*P
DSTILDA=D
L1=eig(ASTILDA)
DETA=det(A)
detstilda=det(ASTILDA)
TRA=trace(A)
TRASTLIDA=trace(ASTILDA)
[num den]=ss2tf(A,B,C,D)
S1=tf(num,den)
[num1 den1]=ss2tf(ASTILDA,BSTILDA,CSTILDA,DSTILDA)
S2=tf(num1,den1)
```

L3b: Controllability, observability and Pole placement

Problem 1 : To Check the controllability and observability of the control system.

```
A=[0 1 0 ;0 0 1;-6 -11 -6 ];
B=[0;0;1];C=[4 5 1];
Ob = obsv(A,C)
no=size(A);
if rank(Ob)==min(no)
display('system is observable')
else
display('system is not observable')
end
CO = ctrb(A,B)
if rank(CO)==min(no)
display('system is controllable')
else
display('system is not controllable')
end
```

Problem 2

%(1) Pole placement by Ackermann’s Formula

Theory: For a system described by state model, A, B, C, D, and (A,B) pair is controllable, then arbitrary placement of closed loop poles can be done by a state feedback control law u=-Kx, given K=[0 0 1]*inv(M)*phi(A), where M is controllability matrix, phi(A) is the matrix polynomial of desired characteristic equation. (no of elements in K equals n, order of system which all except last entry zero and last entry is 1

%Illustrative example 1

```
A=[-1 0 0;1 -2 0;2 1 -3];B=[10;1;0];%Third order system example
eig(A)%poles of open loop system A B C D
M=[B A*B A^2*B];%Controllability matrix
rank(M);
nc=size(A);
if rank(M)==nc
display('system is controllable and pole placement is possible')
else
display('system is not controllable and pole placement is not possible')
exit
end
J=[-1+i*2 0 0;0 -1-i*2 0;0 0 -6]%Canonical representation with desired closed loop poles
poly(J)%desired characteristic polynomial coefficients
Phi=polyvalm(poly(J),A);%desired characteristic polynomial formed with matrix A;
%A^3+alpha1*A^2+alpha2*A+alpha3
K=[0 0 1]*(inv(M))*Phi%Ackerman's formula
Acl=A-B*K
eig(Acl)%poles of closed loop system with u=-Kx.
```

%(2) pole placement by controllable canonical form transformation method 1

%Note: System order considered as 3, for change in dimension of A matrix(system order), %appropriate change need to be made in ai’s, alphai’s,w matrix,j vector, k vector

```
A=[-1 0 0;1 -2 0;2 1 -3];B=[10;1;0];
eig(A)%open loop poles
no=size(A);
Co=ctrb(A,B);
rank(Co)
if rank(CO)==min(no)
display('system is controllable and arbitray pole placement can be done')
else
display('system is not controllable and arbitrary pole placement is not possible')
end
ja=poly(A)%open loop system characteristic polynomial
a1=ja(2);a2=ja(3);a3=ja(4);%extraction of characteristic equation polynomial
w=[a2 a1 1;a1 1 0;1 0 0];%weighing matrix
t=M*w%transformation matrix to convert A to controllable canonical form with last row as
%characteristic equation coefficients with negative sign in reverse order
j=[-1+i*2 0 0;0 -1-i*2 0;0 0 -6];%desired pole location for closed loop system with state
%feedback u=-kx
jj=poly(j);%desired characteristic polynomial
alpha1=jj(2);alpha2=jj(3);alpha3=jj(4);%extraction of closed loop characteristic polynomial
%coefficients
k=[alpha3-a3 alpha2-a2 alpha1-a1]*inv(t)%feedback gain matrix
```

```
Acl=A-B*k;
eig(Acl)%final closed loop poles
```

%(2)Pole placement by controllable canonical transformation method 2

```
A=[-1 0 0;1 -2 0;2 1 -3];B=[10;1;0];
eig(A)%open loop poles
no=size(A);
Co=ctrb(A,B);
rank(Co)
if rank(CO)==min(no)
display('system is controllable and arbitray pole placement can be done')
else
display('system is not controllable and arbitrary pole placement is not possible')
end
ja=poly(A)%open loop system characteristic polynomial
a1=ja(2);a2=ja(3);a3=ja(4);%extraction of characteristic equation polynomial
   invcon=inv(Co)
   p1=invcon(3,:)%Last row of inverse of controllability matrix extracted

V=[p1;p1*A;p1*A*A]% Transformation matrix to transform A to controllable canonical form
AC=V*A*inv(V)%controllable canonical form transformation of A
BC=V*B;
alpha=AC(3,:)%Extraction of last row of Controllable canonical form of A
P=[-1+i*2  -1-i*2  -6]%Desired closed loop poles vector
lamba=poly(P)%Desired characteristic polynomial coefficients
lambda=lamba(1,2:4)%extraction of coefficients except first one.
K1=[alpha(1)+lambda(3) alpha(2)+lambda(2) alpha(3)+lambda(1)]%SFB gain matrix of A in
%CCF
%If A itself is controllable comment following line
K=K1*V%SFB gain matrix for A
eig(A-B*K)
eig(AC-BC*K1)
%Alternate method by using Ackerman formula
Kacker=[0 0 1]*inv(Co)*[A*A*A+lambda(1)*A*A+lambda(2)*A+lambda(3)*eye(3)]
%Alternate method using Acker command
Kacker1=acker(A,B,P)
```

%(4)Using place command

Illustrative example 2: Find the control gain F such that when for the system,

$$A = \begin{bmatrix} -2 & -2.5 & -0.5 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \quad ; \quad B = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

is controlled by u(t) = -Fx(t), the closed loop poles are at s = -1, -2 and -3, verify the control gain by finding the eigen values of A-BF.

Program:

```
% Pole placement design.
```

```
A=[-2 -2.5 -0.5; 1 0 0; 0 1 0]
B=[1; 0; 0]
P= [-1 -2 -3]%Vector of desired closed loop poles
F=place(A,B,P)%Feedback gain matrix, u=-Fx
A_cl=A-B*F
eig(A_cl)%Verification of closed loop poles
```

Exercise

1. For the state model given below obtain the state models (i)) Jordan canonical form (ii) Plot the pole zero plot of the system.

$$\dot{X} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -2 & 1 & -3 \end{bmatrix} \mathbf{X} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \mathbf{U} \quad ; \quad Y = [4 \;\; 1 \;\; 0]$$

2. Determine system controllability and observability.

i) $$A = \begin{bmatrix} -2 & -2.5 & -0.5 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \quad ; \quad B = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \quad ; \quad C = \begin{bmatrix} 1 & 4 & 3.5 \end{bmatrix}$$

3. Place the poles of the system in illustrative example 2 at s=-4, -8, -10.

4. Place the poles of the system.

$$A = \begin{bmatrix} -0.1 & 5 & 0.1 \\ -5 & -0.1 & 5 \\ 0 & 0 & -10 \end{bmatrix} \quad ; \quad B = \begin{bmatrix} 0 \\ 0 \\ 10 \end{bmatrix};$$

at (i) s= -1+j5, -1-j5 and -10 (ii) -5, -60 and -70, find and compare the norms of F.

6. A SISO system represented by the state equation

$$A = \begin{bmatrix} -1 & 0 & 0 \\ 0 & -2 & -2 \\ -1 & 0 & -3 \end{bmatrix} \text{ and B matrix is given by 2 cases } B1 = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \text{ and } B2 = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}$$

For each B i)determine controllability

ii) Find eigen values

iii) Find transformation matrix V that transforms the state equation to controllable canonical form

iv) Determine state feedback matrix K required to assign eigen values to {-2 -4 -6}

L4: Proportional and Lag compensator design

Design -1: Proportional controller (Gain adjustment) design using Bode plot:
For the control system shown in figure (a), find the value of gain K to yield 9.5% overshoot in the transient response for a step input.

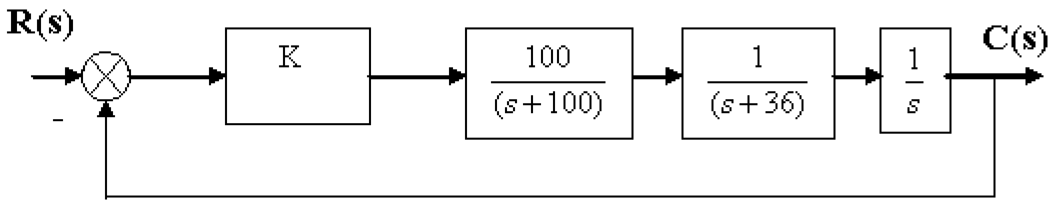


Figure (a)

Procedure
For the given overshoot, find the required damping ratio and hence obtain the required phase margin.
%design1
%Gain adjustment; enter the specifications
OS=9.5; numg=[100]; deng=poly([0 -36 -100]);
G=tf(numg,deng);
%TO FIND REQUIRED DAMPING RATIO
z=(-log(OS/100))/(sqrt(pi^2+log(OS/100)^2));
%To find the rquired phase margin
pmreq=atan(2*z/(sqrt(-2*z^2+sqrt(1+4*z^4))))*(180/pi);

- **Find the phase margin of the given system**
  w=logspace(-2,3,1000);
  margin(G,w);

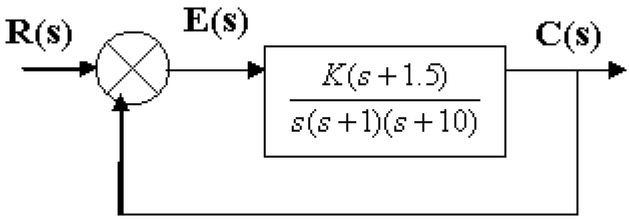
- **Find the additional gain needed to produce the required phase margin**

Calculation: After obtaining Bode plot perform following calculation.
From the plot determine the frequency at which desired phase margin angle is satisfied.
That is mark wgcn as that frequency where phase angle is -180+pmreq
Determine the gain at this frequency from the graph. Let it be M db.
To make this as gain cross over frequency the magnitude plot should cross zero dB at this frequency.
Hence gain required is K=10^(-M/20)

- **Verify the result from closed loop step response.**

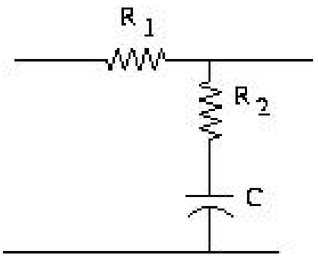
% verify the design by plotting
numgn=K*numg
hold on
bode(numgn,deng)
%To verify closed loop step response with designed K
T=feedback(K*G,1);
step(T)
CGC=K*G;

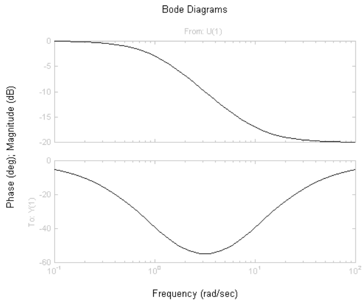
Design 2: Using Root locus
For the system shown below, design the value of gain K to yield 1.52% overshoot. Also estimate the settling time, peak time and steady state error.




```
%design 2
% Root locus method (DO NOT TYPE THE FULL PROGRAM)
%For the system given design the value of gain, to yeild 1.52%overshoot.
%also estimate settling time, peak time and steady state error.
%Enter the system transfer function
clear;clc;clf;
num=[1 1.5];
den=poly([0 -1 -10]);
G=tf(num,den);
%TO OBTAIN THE ROOT LOCUS PLOT
rlocus(G);
title('original root locus plot');
pause
%enter the desired %overshoot
OS=1.52;
%obtain the damping ratio
z=(-log(OS/100))/(sqrt(pi^2+log(OS/100)^2));
%to mark desired damping ratio
sgrid(z,0);
title([' root locus plot with ',num2str(OS),'%overshoot line']);
%to obtain gain K and closed loop poles for point selected, (zoom the figure and
%observe step by step.
[K,P]=rlocfind(G);
%obtain closed loop transfer function for the given K
T=feedback(K*G,1);
pause
step(T);
%verify the time domain specification using graph and equations
TS=-4/real(P(1))
TP=pi/imag(P(1));
s=tf([1 0],[0 1]);
KV=dcgain(s*G);
```

THEORY: LAG compensator design using Frequency response:





TF= $\frac{1}{\beta} * \frac{\left(s+\frac{1}{\tau}\right)}{\left(s+\frac{1}{\beta\tau}\right)}$; $\tau = R_2 C$; $\beta = (R_1 + R_2)/R_2$

NOTE: This form useful for root locus based design, where amplifier gain can be used to to cancel $\frac{1}{\beta}$, and get desired steady state error requirement met by multiplying K to meet desired pole location with β to meet the steady state error requirements.

High frequency attenuation=20*log10 (1/β) db

For Frequency domain design the form $\frac{\tau s+1}{\beta\tau s+1}$ is useful.

Lag Compensator design:
Design a Lag compensator for a given OLTF for achieving a specified phase margin of 30 degrees.

Design -3

For a unity feedback control system with $G(s) = \frac{1}{s(s+1)(0.5s+1)}$, design a phase lag compensator such that the closed loop system will satisfy the following requirements. K_V=5/s, Phase margin=40^0, gain margin >10db.

Procedure:
1. Bode plot of the uncompensated system is plotted after setting the value of the gain constant from the steady state requirement.

```
%lag compensator design using Bbode plots
%enter  the design specifications
KV=5;PM=40;GM=10;
%ENTER THE TRANSFER FUNCTION OF THE UN COMPENSATED SYSTEM
numg=[1];deng=conv(conv([1 0],[1 1]),[0.5 1]);
G=tf(numg,deng);
%TO FIND k, to meet steady state error criteria, use sG(s)H(s)
```

K=KV/(dcgain(conv([1 0],numg),deng));
2. Gain margin and Phase margin of Gain adjusted uncompensated system is measured.

```
w=logspace(-1,2,100);
[mag,ang]=bode(K*numg,deng,w);
[gm,pm,wcg,wcp]=margin(mag,ang,w);
```

Calculation:

3. The frequency corresponding to the required phase margin+allowance(~8 to 12 deg (to cancel additional phase lag due to lag network at new gain cross over frequency) is determined from the Phase curve. This gives ω_m . That is, from the Bode plot, identify the frequency ω_m as where phase angle is -180+PM+allowance.

4. To achieve the specified phase margin at this frequency, the gain curve should pass through 0 db at this frequency. This is achieved by setting the maximum attenuation of the lag-compensating network to be equal to the gain of the uncompensated system at this frequency. i.e., gain of the uncompensated system at new Gain Cross over frequency ω_m , M dB is measured from the magnitude plot and is set equal to -20*log(1/β). Thus, $\beta > 1$ is determined.

That is Measure gain in dB at ω_m from magnitude plot. Compute β using relation

$20 \log_{10} \beta = M$ and β =10^(M/20).

5. To minimize the phase contribution of LAG network around the new gain cross over frequency, the upper corner frequency of the LAG compensating network is chosen to be at least 1 decade less than new gain crossover. Hence the TF of the LAG compensating network given by $G_c(s)= \dfrac{1}{\beta} \dfrac{\left(s+\frac{1}{\tau}\right)}{\left(s+\frac{1}{\beta\tau}\right)} = \dfrac{(\tau s+1)}{(\beta\tau s+1)}$

is designed as, $1/\tau < \omega_m /10$

Hence, the lower corner frequency of lag network is $1/\beta\tau$.

6. Obtain the overall TF using "series" command in MATLAB. Obtain the Bode plot of the overall TF. Hence determine the new gain and phase margin.

%Lag compensator transfer function:

wh= ω_m /10; w1=wh/β; kc=w1/wh;

```
lagc=tf(kc*[1 wh],[1 w1]);
%compensated system
CGCS=K*G*lagc;
S=tf([1 0],[0 1]);
KVF=dcgain(S*CGCS);
CLS=feedback(CGCS,1);
margin(CGCS); pause; step(CLS)
```

Result: Observe and plot the Bode plot of uncompensated with gain compensated system, compensated system, and closed loop step response. Verify the performance.

LAG Compensator using Root locus:

Gain of the system is increased as much as possible without appreciably changing root locus in the vicinity of dominant poles of the closed loop system. This is achieved by restricting the angle contribution of lag network to be negligible say less than 5 deg. Hence the poles and zeros of lag network are placed close to each other and nearer to the origin. The pole and zero of lag compensator is determined as follows:
Open loop poles and zeros of uncompensated system are located in s plane and the dominant closed loop poles are identified to satisfy the given transient response specification. This point should satisfy and phase angle condition of root locus as well as magnitude condition. Applying magnitude condition the open loop gain of uncompensated system is determined as K_u at the dominant pole. Now with the specified steady state requirements, the required open loop gain K to meet the error requirement is determined. The lag network to be introduced in cascade in the forward path has transfer function $G_c(s)= \frac{1}{\beta}\frac{(s+\frac{1}{\tau})}{(s+\frac{1}{\beta\tau})}$. With the introduction of the network the overall forward path gain now becomes $K_u\beta$, after using an amplifier with gain β. To meet the steady state error requirement, this should be equal to K. Hence β is determined as $\frac{K}{K_u}$. However to compensate for the change in magnitude condition due to the introduction of compensator, this gain is compensated by a multiplying with a tolerance value of 1.1. The compensator zero $-\frac{1}{\tau}$ is placed at about 0.1 of second real pole from origin of uncompensated system. With this zero $-\frac{1}{\tau}$ and the value of β determined, the pole can be determined. With this the open loop transfer function of compensated system is written as 1.1KβG_cG and overall root locus is plotted and the specifications are verified. If not satisfied, the procedure is repeated with slightly modifying the position of zero and pole of compensator.

Design Example 4:

For a unity feedback control system with $G(s)=\frac{1}{(s+1)(s+2)(s+10)}$, design a lag compensator to improve the steady state error by a factor of 10 if the system is operating with a damping ratio of 0.174.

Program

```
clear all; close all; clc
s=tf('s')
g1=1/((s+1)*(s+2)*(s+10))
rlocus(g1)
%design a lag compensator to improve steady state error by a factor of 10
z=0.174;
sgrid(z,0); %Marks on rootlocus {\zeta}=0.174 line
[k,p]=rlocfind(g1)%Place cursor on intersection point of {\zeta} line and rootlocus and
%measure gain and poles corresponding to desired damping ratio
```

```
%for type 0 systems
kp=dcgain(k*g1)% Find position error coefficient with gain corresponding to desired DR
ess=1/(1+kp);%steady state for type 0 unity feedback ol system
%to find beta which reduces steady state error by 10
kp1=9+10*kp; %desired position error coefficient to meet specification
beta=(kp1/kp)*1.1;%HF attenuation to be provided by lag network to meet desired spec
%without changing DR
%zero of lag network placed close to origin or abouot 1/10 of second dominant real pole
zlag=0.1;
plag=zlag/beta;
gclag=(s+zlag)/(s+plag);%Lag network transfer function  used with UF OLTF
cgc= g1*gclag%effective UF OLTF
zpk(cgc)
rlocus(cgc)
sgrid(z,0)
[k1,p]=rlocfind(cgc)%Find gain to meet desired DR by placing cursor at intersection
%type 0 systems
kpf=dcgain(k1*cgc)
ess1=1/(1+kpf)
ess/ess1
%Comparison of closed loop response of uncompensated and compensated system
cltfu=feedback(k*g1,1)
step(cltfu)
hold on
cltfc=feedback(kpf*cgc,1)
step(cltfc)
```

Example 5: Lag compensator design using Root locus to get desired velocity error coefficient as kvf=5, z=0.5 and settling time as 10s.

```
clear all; close all; clc
s=tf('s')
%type 1 system
g1=1/(s*(s+1)*(s+4))
rlocus(g1)
z=0.5;TS=10;kvf=5
wn=4/(z*TS);
s1=-z*wn+j*wn*sqrt(1-z*z);%desired closed loop dominant poles
 [k,p]=rlocfind(g1,s1)%finding gain to meet desired poles
%static velocity error constant
d1=s
kv=dcgain(d1*k*g1)%type 1 system
ess=1/kv;
%desired gain
k=kvf/kv;% gain to be provided by lag network to meet steady state error
beta=k*1.1;
zlag=0.2;
plag=zlag/beta;
gclag=(s+zlag)/(s+plag);
cgc=g1*gclag
zpk(cgc)
```

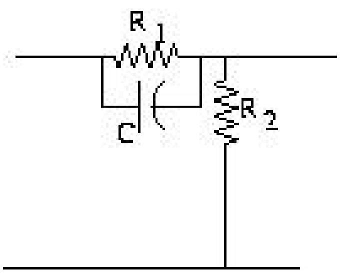
```
rlocus(cgc)
[k1,p1]=rlocfind(cgc,s1)%gain of compensated system which places roots and meet SS error
kvf=dcgain(d1*cgc*k1)% compare with desired value

%Comparison of closed loop response of uncompensated and compensated system
cltfu=feedback(k*g1,1)
figure(2)
step(cltfu)
hold on
cltfc=feedback(k1*cgc,1)
step(cltfc)
```

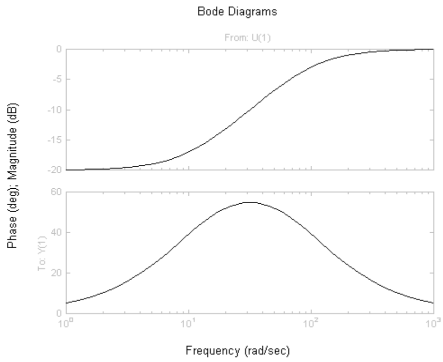
Exercise 1

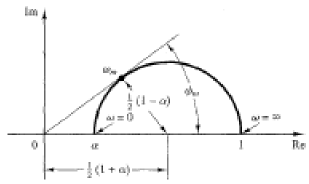
For a unity feedback control system with $G(s)=\dfrac{1}{(s+1)(s+2)(s+10)}$**, design a lag compensator to improve the steady state error by a factor of 10 if the system is operating with a damping ratio of 0.174 and settling time 1s.**

L5: LEAD and LAG-LEAD Compensator design

Theory of Lead compensator design using frequency response.







TF= $\dfrac{\left(s+\dfrac{1}{\tau}\right)}{\left(s+\dfrac{1}{\alpha\tau}\right)}$; $\tau = R_1 C$; $\alpha = R_2 /(R_1 + R_2)$

This network is having low frequency attenuation property. That is low frequency gain α is less than one, ($20\log_{10}(\alpha)$db, which is negative). Hence in the design of compensator an amplifier with a gain of $\dfrac{1}{\alpha}$ is to be cascaded with the lead network in the frequency domain design. That is effectively using lead network transfer function as $\dfrac{(\tau s+1)}{(\alpha\tau s+1)}$.

The design value of α is obtained from the maximum phase lead provided by the network. From the polar plot of lead network shown in figure, by drawing a tangent to the polar plot origin, we get the maximum phase lead angle φ_m provided by the network. By drawing the normal to this to the center of the semicircle, we can relate maximum phase lead angle φ_m to α as $\sin\phi_m = \dfrac{1-\alpha}{1+\alpha}$. Using the phase angle expression φ_m= atan($\omega_m\tau$)-atan($\alpha\omega_m\tau$) at frequency ω_m, and relationship for tan φ_m=$\dfrac{1-\alpha}{2\sqrt{\alpha}}$, from the polar plot, we get ω_m=$\dfrac{1}{\tau\sqrt{\alpha}}$

From magnitude condition, the magnitude of lead network at ω_m= $10\log_{10}(\alpha)$db (negative)(or $\sqrt{\alpha}$ in abs). After cascading the amplifier, this will become -10log10(α)db (positive) or($\dfrac{1}{\sqrt{\alpha}}$ in abs).

Characteristics

Lead network has high pass filter characteristics. Compensator zero is closer to the origin than pole. The reshaping of root locus/frequency response is achieved through the phase lead characteristics. It improves damping ratio and hence speed of response, ie, rise time and settling time reduces. Gain cross over frequency increase increasing bandwidth and speed of closed loop response. It also increases phase margin. Additional amplifier gain is essential to overcome low frequency attenuation. There is no much improvement in steady state performance.

Main drawback of lead compensator is if the system is prone to the effect of noise, which is of high frequency signal, then this may amplify the noise and performance may degrade.

Design 1:

For a unity feedback control system with $G(s) = \frac{4K}{s(s+2)}$**, design a phase lead compensator such that the closed loop system will satisfy the following requirements. K_V=20/s, Phase margin=50^0, gain margin >10db.**

Procedure: Design a Phase ***lead*** compensator for the given OLTF
***Specification*:** Phase margin at least 50 degrees, Velocity error coefficient is 20/s

1. To evaluate K of given OLTF:
   Given K_v=20.
   Where $K_v = \lim_{s\to 0}$ sGH(s), which gives K as G(s) is type 1 system.

Draw Bode plot using Matlab.

```
%enter the transfer function of uncompensated system
numg=[4];
deng=[1 2 0];
G=tf(numg,deng);
%To solve for K satisfying KV=20
K=20/(dcgain(conv([1 0],numg),deng));
```

1. With this value of K, obtain Bode plot of OLTF, from which determine Phase margin pm

```
%to obtain the factored form of G *K
        G1=zpk(G*K);
        w=logspace(-1,2,100);
        bode(G1,w);
```

Design of Lead network:

To compute the max phase lead angle considering 8 to 12deg additional for tolerance

Measure actual phase margin as pm from graph
Compute desired max phase lead ϕ_m as PC=(PM+10)-pm;

Design of lead compensator by computing alpha
alpha=(1-sin(PC*pi/180))/(1+sin(PC*pi/180)); %(α =<1)

and corresponding magnitude in dB contributed by lead network at max phase lead angle frequency
magpc=20*log10(1/sqrt(alpha)); (Positive)(after using amplifier)

To obtain the frequency corresponds to the magnitude magpc fom graph
From the Bode plot of gain compensated system measure frequency at which gain is –magpc dB(negative)

Note this frequency as wmax. Compute zero of lead network as

2. Determine time constant of compensating network $\tau = \frac{1}{(\sqrt{\alpha})\omega_m}$ or

   zc=wmax*sqrt(alpha);
3. Pole of lead network as pc=zc/alpha;

4. Obtain the TF of compensating network $G_c(s) = \frac{1}{\alpha}\frac{\left(s+\frac{1}{\tau}\right)}{\left(s+\frac{1}{\alpha\tau}\right)}$

Verification using MATLAB:

```
    kc=1/alpha;
    GC=tf(kc*[1 zc],[1 pc]);%Lead network transfer function with amplifier included

%compensated system
CGC=G1*GC;%Overall forward path transfer function

margin(CGC);%measure compensated system Phase margin, gain margin
pause
S=tf([1 0],[0 1]);
KV=dcgain(S*CGC)%Verification of velocity error coefficient
%CLOSED LOOP SYSTEM
CLGC=feedback(CGC,1);
step(CLGC) %closed loop step response of compensated unity feedback system
%factorise the compensated system
CGC1=zpk(CGC)
```

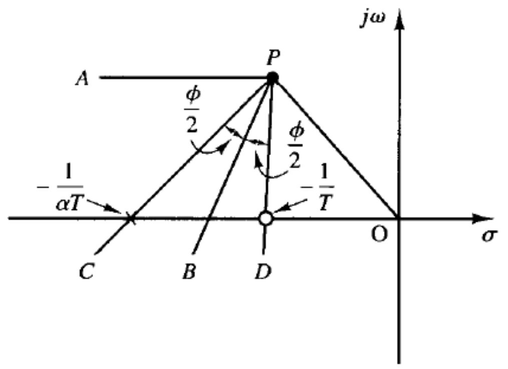
LEAD Compensator design using Root locus

<u>Procedure</u>

From the performance specifications, determine the desired location for the dominant closed loop poles. After drawing root locus ascertain whether by gain adjustment alone yields the desired roots. If not find the sum of angles at one of the desired closed loop poles with the open loop poles and zeros of the original system, an determine the angle of deficiency φ_m to be added so that the total sum of the angles is equal to $\pm 180^{o+}(2k+1)$. The lead compensator must contribute this angle φ_m to make the desired closed loop poles pass through the root locus. The lead compensator $G_c(s)$ has the form

$$G_c(s) = K_c\alpha\frac{Ts+1}{\alpha Ts+1} = K_c\frac{s+\frac{1}{T}}{s+\frac{1}{\alpha T}}, \qquad (0<\alpha<1)$$

, where K_c is the amplifier gain required to compensate low frequency attenuation, α and T are determined from angle deficiency. K_c is determined from the requirement of open loop gain.



First draw a horizontal line passing through point P, the desired location for one of the dominant closed loop poles. This is shown as line PA. Draw also a line connecting point P and the origin. Bisect the angle between the lines PA and PO. Draw two lines PC and PD that make angles $\pm\varphi_m/2$ with the bisector PB. The intersections of PC and PD with the negative real axis give necessary location for the pole and zero of the lead network. The compensator thus designed will make point P a point on the root locus of the compensated system.

If static error coefficients are not specified, the lead compensator pole and zero thus obtained will contribute the necessary angle φ_m. If no other constraints are imposed on the system, make α as large as possible to minimize steady state error. The open loop gain is determined by use of the magnitude condition.

Once the lead compensator is designed, check whether the specifications are met with cascading compensator with the original system. If not satisfactory, repeat the design by changing the position of compensator pole and zero till performance is satisfactory.

Design Example **Program 2**

```
%OGATA pg 486-490 ex 10-1 correct solution

%problem OGATA 4/s(s+2); spec desired clpole wn=4,z=0.5

clear all;close all;clc
%desired zeta and natural frequency
z=0.5; wn=4;
n1=4; d1=conv([1 0],[1 2])
G=tf(n1,d1);
figure(1)
rlocus(G)
%desired closed loop poles
s1=-z*wn+j*wn*sqrt(1-z*z); s2=-z*wn-j*wn*sqrt(1-z*z);
%to determine angle contribution at desired cl poles
th1=(angle(polyval([1 0],s1)))*180/pi
th2=(angle(polyval([1 2],s1)))*180/pi
phd=180-(th1+th2)
%phase angle lead required by lead network is -phd
%graphical construnction
%angle between line joining s1 with origin and horizontal from s1
```

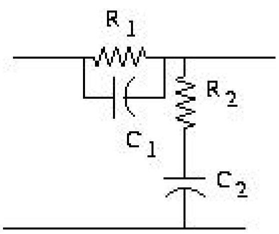
```
phs1=(angle(polyval([1 0],s1)))*180/pi
%zero location bisection of above angle and half of lead contribution cw
angz=(phs1+phd)/2;
%pole location bisection of phs1 and half of lead angle ccw
angp=(phs1-phd)/2;
%to find x coordinate of zero and pole with respect to vertical from s1
zang=(angz-(phs1-90))*pi/180;%angle of zero from vertical from s1
pang=(angp-(phs1-90))*pi/180;%angle of pole from vertical from s1
%zero and pole loaction with respect to real part of s1
x1=imag(s1)*tan(zang)%zero location with respect to vertical from s1
x2=imag(s1)*tan(pang)%pole position wrt vertical from s1
ze=-(real(s1)-x1)
pe=-(real(s1)-x2)
%LEAD COMPENSATOR
nc=[1 ze]; dc=[1 pe];
gc=tf(nc,dc);%
cgc=G*gc;
zpk(cgc)
% to find gain
[k, p]=rlocfind(cgc,s1)
cgcs=k*cgc
figure(2)
rlocus(cgc)
T=feedback(cgc*k,1)
figure(3)
step(T)
d=tf([1 0],[0 1])
kv=dcgain(d*cgcs)
%comparision of equivalent second order system with dominant poles s1,s2
dcl=[1 -(s1+s2) s1*s2]; ncl=s1*s2;
hold on
tfcl=tf(ncl,dcl)
step(tfcl)
```

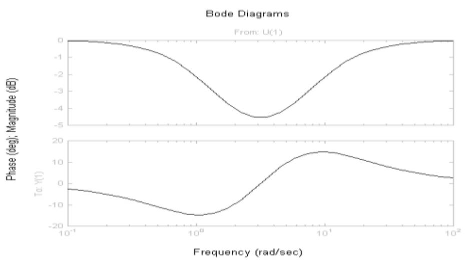
%Problem 3: when rise time, overshoot and wn is specified

```
%problem pillai 100/s(s+8); spec OS 9.5% wn 12r/s
clear all;close all;clc
n1=[100]; d1=[1 8 0];
G=tf(n1,d1);
figure(1)
rlocus(G)
OS=9.5;wn=12;
%desired zeta
z=(-log(OS/100))/(sqrt(pi*pi+log(OS/100)^2));
%desired closed loop poles
s1=-z*wn+j*wn*sqrt(1-z*z);
s2=-z*wn-j*wn*sqrt(1-z*z);
%to determine angle contribution at desired cl poles
th1=(angle(polyval([1 0],s1)))*180/pi
```

```
th2=(angle(polyval([1 8],s1)))*180/pi
phd=180-(th1+th2)
%phase angle lead required by lead network is -phd
%graphical construnction
%angle between line joining s1 with origin and horizontal from s1
phs1=(angle(polyval([1 0],s1)))*180/pi
%zero location bisection of above angle and half of lead contribution cw
angz=(phs1+phd)/2;
%pole location bisection of phs1 and half of lead angle ccw
angp=(phs1-phd)/2;
%to find x coordinate of zero and pole with respect to vertical from s1
zang=(angz-(phs1-90))*pi/180;%angle of zero from vertical from s1
pang=(angp-(phs1-90))*pi/180;%angle of pole from vertical from s1
%zero and pole loaction with respect to real part of s1
x1=imag(s1)*tan(zang)%zero location with respect to vertical from s1
x2=imag(s1)*tan(pang)%pole position wrt vertical from s1
ze=-(real(s1)-x1)
pe=-(real(s1)-x2)
%LEAD COMPENSATOR
nc=[1 ze];
dc=[1 pe];
gc=tf(nc,dc);%
cgc=G*gc;
zpk(cgc)
% to find gain
[k, p]=rlocfind(cgc,s1)
cgcs=k*cgc
figure(2)
rlocus(cgc)
T=feedback(cgc*k,1)
figure(3)
step(T)
d=tf([1 0],[0 1])
kv=dcgain(d*cgcs)
%comparision of equivalent second order system with dominant poles s1,s2
dcl=[1 -(s1+s2) s1*s2]; ncl=s1*s2;
hold on
tfcl=tf(ncl,dcl)
step(tfcl)
```

Lag-Lead network





TF= $\dfrac{\left(s+\dfrac{1}{\tau_1}\right)}{\left(s+\dfrac{1}{\beta\tau_1}\right)}\dfrac{\left(s+\dfrac{1}{\tau_2}\right)}{\left(s+\dfrac{1}{\alpha\tau_2}\right)}$ $\alpha\beta=1, \alpha<1, \beta>1$, $R_1C_1=\tau_1, R_2C_2=\tau_2$

Lag Lead

Lag-lead compensator: Used to achieve performance of both Lag compensator and lead compensator in overall system.

Design -4

For a unity feedback control system with $G(s)=\dfrac{K}{s(s+8)(s+30)}$**, design a lag - lead compensator such that the closed loop system will satisfy the following requirements. K_V=20/s, peak time=0.6sec and% overshoot 15.**

```
%Design of Lag -LEAD compensator using Frequency domain(bode plot) method.
%Enter the uncompensated system transfer function
numg=[0 0 0 1];deng=conv(conv([1 0],[1 8]),[1 30]);
G=tf(numg,deng);
%enter the design specifications(overshoot, peak time and KV)
TP=0.6;KV=20;OS=15;
%TO FIND k,use sG(s)H(s)
K=KV/(dcgain(conv([1 0],numg),deng));
%To find the gain margin, phase margin, gain cross over frequency
%and phase cross over frequency
%To find the desired bandwidth
z=(-log(OS/100))/(sqrt(pi^2+log(OS/100)^2));
wn=pi/(sqrt(1-z^2));
wb=wn*sqrt((1-2*z^2)+sqrt(4*z^4-4*z^2+2));
wb1=wb*0.8;
pmreq=atan(2*z/(sqrt(-2*z^2+sqrt(1+4*z^4))))*(180/pi);
w=logspace(-1,2,100);
[m,P]=bode(K*G,wb*0.8);
pmreqc=pmreq-180+P;
beta=(1-sin(pmreqc*pi/180))/(1+sin(pmreqc*pi/180));
%design og lag compensator
%adjust for design ,first take 10 and then adjust for design
W1=wb1/8;
W2=W1*beta;
kclag=beta;
%lag compensator is
LAGC=tf(kclag*[1 W1],[1 W2]);
%margin(LAGC)
%TO design the lead portion of the compensator
%Lead compensator transfer function
P1=wb1*sqrt(beta);
P2=P1/beta;
```

```
kclead=1/beta;
LEADC=tf(kclead*[1 P1],[1 P2]);
%margin(LEADC)
%TO obtain the transfer function of compensated system
C=K*G*LEADC*LAGC;
pause
margin(C)
%TO check  KV,
D=tf([1 0],[0 1]);
C1=C*D; KV=dcgain(C1)
pause
%to obtain the closed loop step response
C2=feedback(C,1);
step(C2)
```

Exercise

1. For a unity feedback control system with $G(s) = \dfrac{K}{s^3 + 5s^2 + 4s}$, design a lag - lead compensator such that the closed loop system will satisfy the following requirements. K_V=10/s, phase margin=50^0, gain margin >10db.