**Chapter 9 – Solutions to Instructor Reserve Problems**

**I-1 Instructor**

**a.** Searching along the 126.16o line (10% overshoot, ** = 0.59), find the operating point at

-1.1207 + j1.5336 with *K* = 27.9948. Hence, *Kv* = .

**b.** A 3.0006 x improvement will yield *Kv* = 4. Use a lag compensator, *Gc(s)* = .

**c.**

**Program:**

K=17.5

G=zpk([],[0,-3,-5],K)

Gc=zpk([-0.3429],[-0.1],1)

Ge=G\*Gc;

T1=feedback(G,1);

T2=feedback(Ge,1);

T3=tf(1,[1,0]); %Form 1/s to integrate step input

T4=T1\*T3;

T5=T2\*T3;

t=0:0.1:20;

step(T3,T4,T5,t) %Show input ramp and ramp responses

**Computer response:**

K =

27.9948

Zero/pole/gain:

27.9948

-------------

s (s+3) (s+7)

Zero/pole/gain:

(s+0.3001)

----------

(s+0.1)



**I-2 Instructor**

**Program:**

clf

'Uncompensated System'

numg=[1 6];

deng=poly([-2 -3 -5]);

'G(s)'

G=tf(numg,deng);

Gzpk=zpk(G)

rlocus(G,0:1:100)

z=0.707;

pos=exp(-pi\*z/sqrt(1-z^2))\*100;

sgrid(z,0)

title(['Uncompensated Root Locus with ' , num2str(z), ' Damping Ratio Line'])

[K,p]=rlocfind(G); %Allows input by selecting point on graphic

'Closed-loop poles = '

p

i=input('Give pole number that is operating point ');

'Summary of estimated specifications'

operatingpoint=p(i)

gain=K

estimated\_settling\_time=4/abs(real(p(i)))

estimated\_peak\_time=pi/abs(imag(p(i)))

estimated\_percent\_overshoot=pos

estimated\_damping\_ratio=z

estimated\_natural\_frequency=sqrt(real(p(i))^2+imag(p(i))^2)

Kp=dcgain(K\*G)

'T(s)'

T=feedback(K\*G,1)

'Press any key to continue and obtain the step response'

pause

step(T)

title(['Step Response for Uncompensated System with ' , num2str(z),...

' Damping Ratio'])

'Press any key to go to PD compensation'

pause

'Compensated system'

done=1;

while done>0

a=input('Enter a Test PD Compensator, (s+a). a = ')

numc=[1 a];

'Gc(s)'

GGc=tf(conv(numg,numc),deng);

GGczpk=zpk(GGc)

wn=4/[(estimated\_settling\_time/2)\*z];

rlocus(GGc)

sgrid(z,wn)

title(['PD Compensated Root Locus with ' , num2str(z),...

' Damping Ratio Line', 'PD Zero at ', num2str(a), ', and Required Wn'])

done=input('Are you done? (y=0,n=1) ');

end

[K,p]=rlocfind(GGc); %Allows input by selecting point on graphic

'Closed-loop poles = '

p

i=input('Give pole number that is operating point ');

'Summary of estimated specifications'

operatingpoint=p(i)

gain=K

estimated\_settling\_time=4/abs(real(p(i)))

estimated\_peak\_time=pi/abs(imag(p(i)))

estimated\_percent\_overshoot=pos

estimated\_damping\_ratio=z

estimated\_natural\_frequency=sqrt(real(p(i))^2+imag(p(i))^2)

Kp=dcgain(K\*GGc)

'T(s)'

T=feedback(K\*GGc,1)

'Press any key to continue and obtain the step response'

pause

step(T)

title(['Step Response for Compensated System with ' , num2str(z),...

' Damping Ratio'])

**Computer response:**

ans =

Uncompensated System

ans =

G(s)

Zero/pole/gain:

(s+6)

-----------------

(s+5) (s+3) (s+2)

Select a point in the graphics window

selected\_point =

-2.3104 + 2.2826i

ans =

Closed-loop poles =

p =

-5.3603

-2.3199 + 2.2835i

-2.3199 - 2.2835i

Give pole number that is operating point 2

ans =

Summary of estimated specifications

operatingpoint =

-2.3199 + 2.2835i

gain =

4.4662

estimated\_settling\_time =

1.7242

estimated\_peak\_time =

1.3758

estimated\_percent\_overshoot =

4.3255

estimated\_damping\_ratio =

0.7070

estimated\_natural\_frequency =

3.2552

Kp =

0.8932

ans =

T(s)

Transfer function:

4.466 s + 26.8

-----------------------------

s^3 + 10 s^2 + 35.47 s + 56.8

ans =

Press any key to continue and obtain the step response

ans =

Press any key to go to PD compensation

ans =

Compensated system

Enter a Test PD Compensator, (s+a). a = 6

a =

6

ans =

Gc(s)

Zero/pole/gain:

(s+6)^2

-----------------

(s+5) (s+3) (s+2)

Are you done? (y=0,n=1) 1

Enter a Test PD Compensator, (s+a). a = 7.1

a =

7.1000

ans =

Gc(s)

Zero/pole/gain:

(s+7.1) (s+6)

-----------------

(s+5) (s+3) (s+2)

Are you done? (y=0,n=1) 0

Select a point in the graphics window

selected\_point =

-4.6607 + 4.5423i

ans =

Closed-loop poles =

p =

-4.6381 + 4.5755i

-4.6381 - 4.5755i

-5.4735

Give pole number that is operating point 1

ans =

Summary of estimated specifications

operatingpoint =

-4.6381 + 4.5755i

gain =

4.7496

estimated\_settling\_time =

0.8624

estimated\_peak\_time =

0.6866

estimated\_percent\_overshoot =

4.3255

estimated\_damping\_ratio =

0.7070

estimated\_natural\_frequency =

6.5151

Kp =

6.7444

ans =

T(s)

Transfer function:

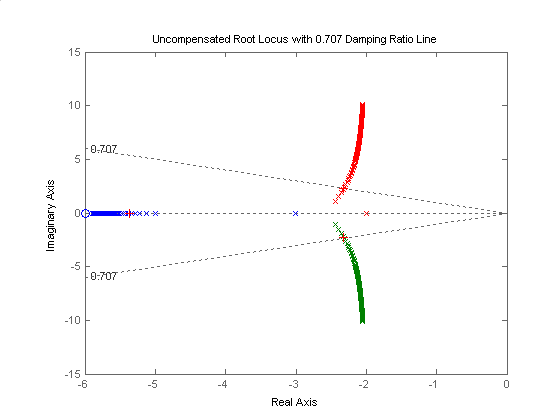
4.75 s^2 + 62.22 s + 202.3

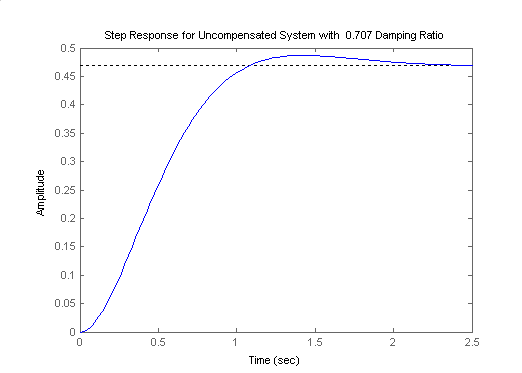
---------------------------------

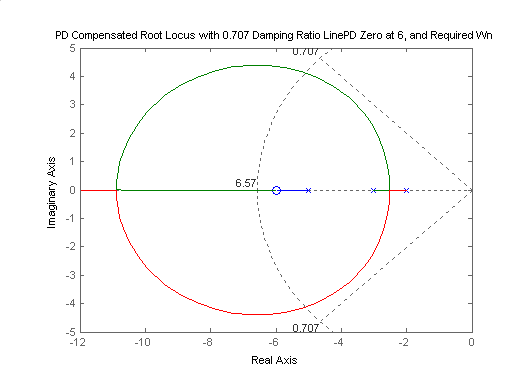
s^3 + 14.75 s^2 + 93.22 s + 232.3

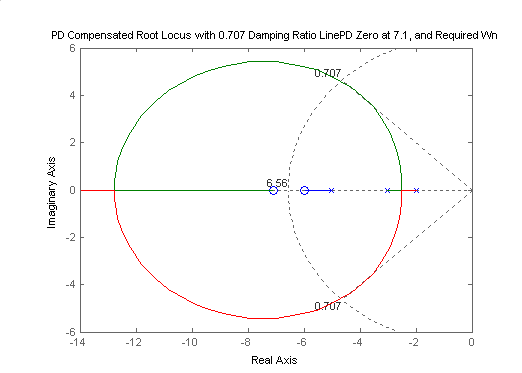
ans =

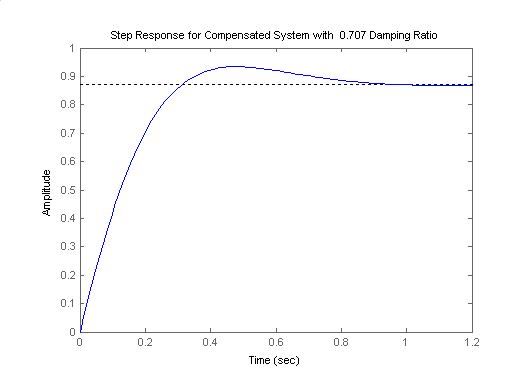
Press any key to continue and obtain the step response











**I-3 Instructor**

The damping ratio corresponding to 20.5% O.S. is ** = 0.45. Theuncompensated system’s step response (at that value of **) was obtained using MATLAB and its performance is summarized in Table P-9.8.

To improve the settling time by 4, the dominant poles need to be at -11.60 ± j23.24. Summing the angles from the open-loop poles to the design point, yields 84.6o. Thus, the zero must contribute 180o - 84.6o = 95.4o. Using the geometry below,

**X**

**23.24**

**95.4oOo**

**σ**

**- ZC**

**j**

**-11.6**

 Thus, *ZC* = - 9.40.

The step response of the compensated system was plotted using MATLAB and the results obtained are also summarized below in Table P-9.8.

|  |  |  |
| --- | --- | --- |
|  | Uncompensated | Compensated |
| Plant and Compensator |  |  |
| Dominant poles | -2.90 ± j5.81 | - 11.60 ± j23.24 |
| K | 1146.5 | 691 |
| Ζ | 0.45 | 0.45 |
| ωn | 6.3 | 25.7 |
| %O.S. | 20.5 | 20.6 |
| Ts | 1.32 | 0.321 |
| Tp | 0.587 | 0.135 |
| Kv | 5.73 | 32.5 |
| e(∞) (ramp) | 0.174 | 0.0308 |
| Other poles | -27.1969 | -9.78 |
| Zero | None | -9.4 |
| Comments | Second-order approx. OK | Simulate to be sure of pole/zero cancellation |

**I-4 Instructor**

**a.** Searching the 15% overshoot line (121.127o) for 180o yields -0.372 + j0.615. Hence, Ts = = = 10.75 seconds.

**b.** For 7 seconds settling time, d = = = 0.571. d = 0.571 tan (180o - 121.127o) = 0.946. Therefore, the design point is -0.571 + j0.946. Summing the angles of the uncompensated system's poles as well as the compensator pole at -15 yields -213.493o. Therefore, the compensator zero must contribute (213.493o - 180o) = 33.493o. Using the geometry below,



= tan (33.493o) . Hence, zc = 2. The compensated open-loop transfer function is

. Evaluating the gain for this function at the point, -0.571 + j0.946 yields K = 207.512.

**c.**

**Program:**

numg= 207.512\*[1 2];

r=roots([1,10,26]);

deng=poly([0 ,-1, r(1),r(2),-15]);

'G(s)'

G=tf(numg,deng);

Gzpk=zpk(G)

T=feedback(G,1);

step(T)

title(['Step Response for Design of Ts = 7, %OS = 15'])

**Computer response:**

ans =

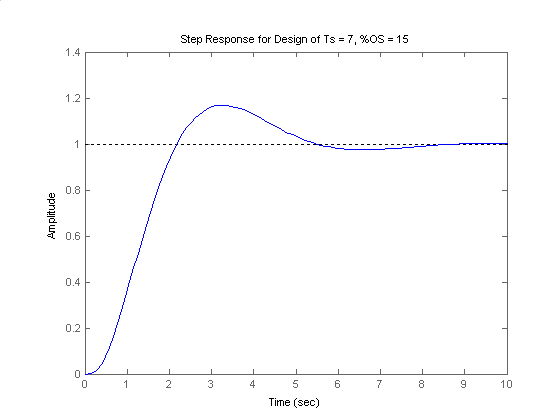
G(s)

Zero/pole/gain:

207.512 (s+2)

-------------------------------

s (s+15) (s+1) (s^2 + 10s + 26)



**I-5 Instructor**

**Program:**

numg=1;

deng=poly([-2 -4 -6 -8]);

'G(s)'

G=tf(numg,deng);

Gzpk=zpk(G)

rlocus(G,0:5:500)

z=0.5;

pos=exp(-pi\*z/sqrt(1-z^2))\*100;

sgrid(z,0)

title(['Uncompensated Root Locus with ' , num2str(z), ' Damping Ratio Line'])

[K,p]=rlocfind(G); %Allows input by selecting point on graphic

'Closed-loop poles = '

p

i=input('Give pole number that is operating point ');

'Summary of estimated specifications for uncompensated system'

operatingpoint=p(i)

gain=K

estimated\_settling\_time=4/abs(real(p(i)))

estimated\_peak\_time=pi/abs(imag(p(i)))

estimated\_percent\_overshoot=pos

estimated\_damping\_ratio=z

estimated\_natural\_frequency=sqrt(real(p(i))^2+imag(p(i))^2)

Kpo=dcgain(K\*G)

T=feedback(K\*G,1);

'Press any key to continue and obtain the step response'

pause

step(T)

whitebg('w')

title(['Step Response for Uncompensated System with ' , num2str(z),...

' Damping Ratio'],'color','black')

'Press any key to go to Lead compensation'

pause

'Compensated system'

b=5;

'Lead Zero at -b '

done=1;

while done>0

a=input('Enter a Test Lead Compensator Pole, (s+a). a = ');

numgglead=[1 b];

dengglead=conv([1 a],poly([-2 -4 -6 -8]));

'G(s)Glead(s)'

GGlead=tf(numgglead,dengglead);

GGleadzpk=zpk(GGlead)

wn=4/((estimated\_settling\_time-0.5)\*z);

clf

rlocus(GGlead,0:10:2000)

sgrid(z,wn)

axis([-10 0 -5 5])

title(['Lead Compensated Root Locus with ' , num2str(z),...

' Damping Ratio Line, Lead Pole at ', num2str(-a), ', and Required Wn'])

done=input('Are you done? (y=0,n=1) ');

end

[K,p]=rlocfind(GGlead); %Allows input by selecting point on graphic

'Closed-loop poles = '

p

i=input('Give pole number that is operating point ');

'Summary of estimated specifications for lead-compensated system'

operatingpoint=p(i)

gain=K

estimated\_settling\_time=4/abs(real(p(i)))

estimated\_peak\_time=pi/abs(imag(p(i)))

estimated\_percent\_overshoot=pos

estimated\_damping\_ratio=z

estimated\_natural\_frequency=sqrt(real(p(i))^2+imag(p(i))^2)

Kplead=dcgain(K\*GGlead)

T=feedback(K\*GGlead,1);

'Press any key to continue and obtain the step response'

pause

step(T)

whitebg('w')

title(['Step Response for Lead Compensated System with ' , num2str(z),...

' Damping Ratio'],'color','black')

'Press any key to continue and design lag compensation'

pause

'Improvement in steady-state error with lead compensator is'

error\_improvement=(1+Kplead)/(1+Kpo)

additional\_error\_improvement=30/error\_improvement

Kpnn=additional\_error\_improvement\*(1+Kplead)-1

pc=0.001

zc=pc\*(Kpnn/Kplead)

numggleadlag=conv(numgglead,[1 zc]);

denggleadlag=conv(dengglead,[1 pc]);

'G(s)Glead(s)Glag(s)'

GGleadGlag=tf(numggleadlag,denggleadlag);

GGleadGlagzpk=zpk(GGleadGlag)

rlocus(GGleadGlag,0:10:2000)

z=0.5;

pos=exp(-pi\*z/sqrt(1-z^2))\*100;

sgrid(z,0)

title(['Lag-Lead Compensated Root Locus with ' , num2str(z), ...

' Damping Ratio Line and Lag Pole at ',num2str(-pc)])

[K,p]=rlocfind(GGleadGlag); %Allows input by selecting point on graphic

'Closed-loop poles = '

p

i=input('Give pole number that is operating point ');

'Summary of estimated specifications for lag-lead compensated system'

operatingpoint=p(i)

gain=K

estimated\_settling\_time=4/abs(real(p(i)))

estimated\_peak\_time=pi/abs(imag(p(i)))

estimated\_percent\_overshoot=pos

estimated\_damping\_ratio=z

estimated\_natural\_frequency=sqrt(real(p(i))^2+imag(p(i))^2)

Kpleadlag=dcgain(K\*GGleadGlag)

T=feedback(K\*GGleadGlag,1);

'Press any key to continue and obtain the step response'

pause

step(T)

whitebg('w')

title(['Step Response for Lag-Lead Compensated System with ', num2str(z),...

' Damping Ratio and Lag Pole at ',num2str(-pc)],'color','black')

**Computer response:**

ans =

G(s)

Zero/pole/gain:

1

-----------------------

(s+8) (s+6) (s+4) (s+2)

Select a point in the graphics window

selected\_point =

-1.5036 + 2.6553i

ans =

Closed-loop poles =

p =

-8.4807 + 2.6674i

-8.4807 - 2.6674i

-1.5193 + 2.6674i

-1.5193 - 2.6674i

Give pole number that is operating point 3

ans =

Summary of estimated specifications for uncompensated system

operatingpoint =

-1.5193 + 2.6674i

gain =

360.8014

estimated\_settling\_time =

2.6328

estimated\_peak\_time =

1.1778

estimated\_percent\_overshoot =

16.3034

estimated\_damping\_ratio =

0.5000

estimated\_natural\_frequency =

3.0698

Kpo =

0.9396

ans =

Press any key to continue and obtain the step response

ans =

Press any key to go to Lead compensation

ans =

Compensated system

ans =

Lead Zero at -b

Enter a Test Lead Compensator Pole, (s+a). a = 10

ans =

G(s)Glead(s)

Zero/pole/gain:

(s+5)

------------------------------

(s+10) (s+8) (s+6) (s+4) (s+2)

Are you done? (y=0,n=1) 1

Enter a Test Lead Compensator Pole, (s+a). a = 15

ans =

G(s)Glead(s)

Zero/pole/gain:

(s+5)

------------------------------

(s+15) (s+8) (s+6) (s+4) (s+2)

Are you done? (y=0,n=1) 0

Select a point in the graphics window

selected\_point =

-1.9076 + 3.2453i

ans =

Closed-loop poles =

p =

-13.0497 + 1.9313i

-13.0497 - 1.9313i

-5.0654

-1.9176 + 3.2514i

-1.9176 - 3.2514i

Give pole number that is operating point 4

ans =

Summary of estimated specifications for lead-compensated system

operatingpoint =

-1.9176 + 3.2514i

gain =

1.3601e+003

estimated\_settling\_time =

2.0860

estimated\_peak\_time =

0.9662

estimated\_percent\_overshoot =

16.3034

estimated\_damping\_ratio =

0.5000

estimated\_natural\_frequency =

3.7747

Kplead =

1.1806

ans =

Press any key to continue and obtain the step response

ans =

Press any key to continue and design lag compensation

ans =

Improvement in steady-state error with lead compensator is

error\_improvement =

1.1243

additional\_error\_improvement =

26.6842

Kpnn =

57.1876

pc =

0.0010

zc =

0.0484

ans =

G(s)Glead(s)Glag(s)

Zero/pole/gain:

(s+5) (s+0.04844)

----------------------------------------

(s+15) (s+8) (s+6) (s+4) (s+2) (s+0.001)

Select a point in the graphics window

selected\_point =

-1.8306 + 3.2919i

ans =

Closed-loop poles =

p =

-13.0938 + 2.0650i

-13.0938 - 2.0650i

-5.0623

-1.8617 + 3.3112i

-1.8617 - 3.3112i

-0.0277

Give pole number that is operating point 4

ans =

Summary of estimated specifications for lag-lead compensated system

operatingpoint =

-1.8617 + 3.3112i

gain =

1.4428e+003

estimated\_settling\_time =

2.1486

estimated\_peak\_time =

0.9488

estimated\_percent\_overshoot =

16.3034

estimated\_damping\_ratio =

0.5000

estimated\_natural\_frequency =

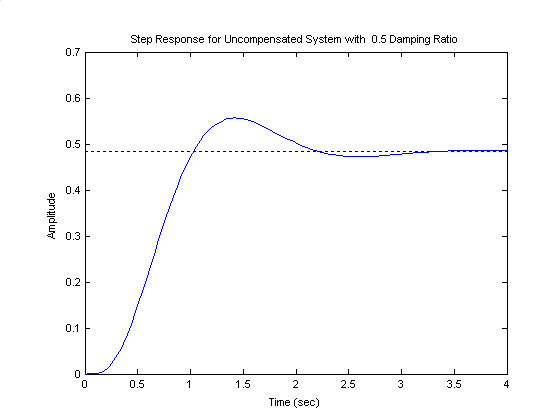
3.7987

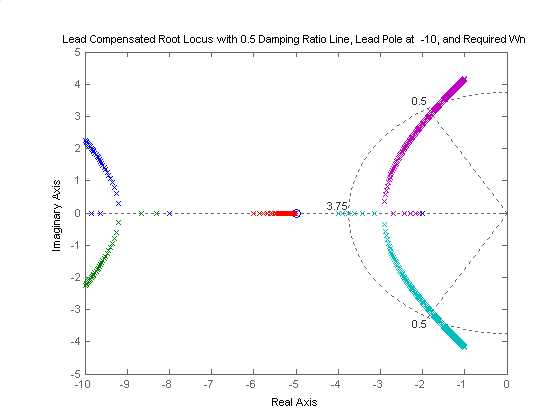
Kpleadlag =

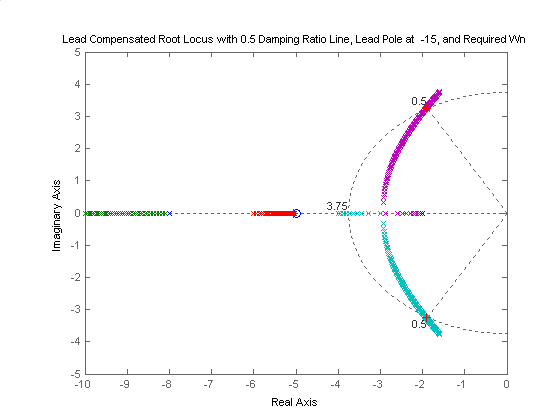
60.6673

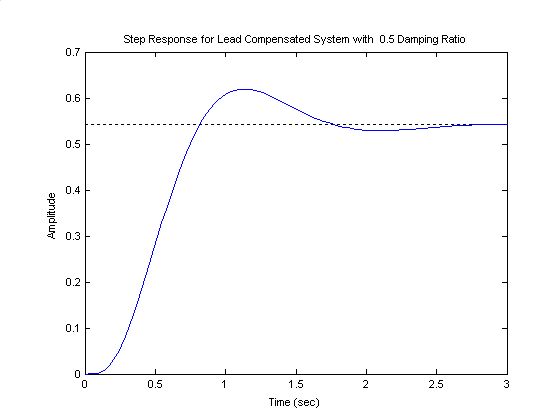
ans =

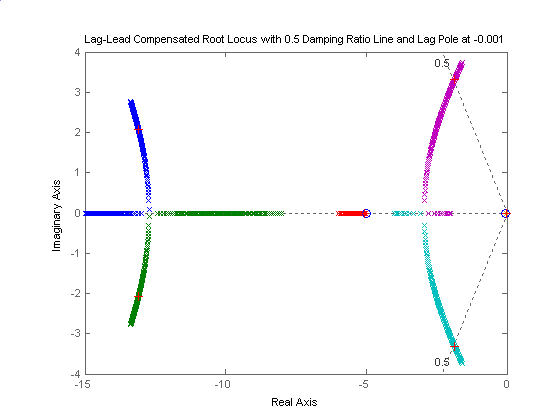
Press any key to continue and obtain the step response

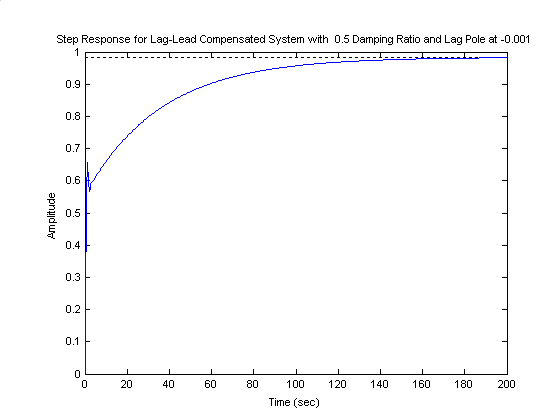












**I-6 Instructor**

**a**.Searching along the 10% overshootline (** = 0.591) the operating point is found to be –1.851 + j2.539 with *K* = 21.71. A third pole is at –10.298. Thus, the estimated performance before compensation is: 10% overshoot,  seconds, and .

**b**. **Lead design:** Place compensator zero at –3. The desired operating point is found from the desired specifications.  and . Thus, . Hence the design point is –4 + j5.46. The angular contribution of the system poles and compensator zero at the design point is –166.960. Thus, the compensator pole must contribute –1800 + 166.960 = -13.040. Using the geometry below, = tan (13.04o). Hence, *pc* = 27.57. The compensated open-loop transfer function is . Evaluating the gain for this function at the point -4 + j5.46 yields K = 1092 with higher-order poles at –4.055 and –29.52.

s-plane

j





j5.46

-4

-p

c

13.04

o

X

Thus, with lead-compensation (see the step-response shown below) we have: 22% overshoot; settling time, Ts = 0.943 seconds; and .



**Lag design:** For the lead-compensated system, Kp = 1.485. Thus, we need an improvement of  times. Thus, the transfer function of the lag compensator to be added should be: , where .

**c**. MATLAB was used to simulate the lag-lead compensated system. After a few attempts (starting from *plag* = 0.01) we found that *plag* = 0.3 would lead to a response, which satisfies all of the desired specifications. The corresponding equivalent forward-path transfer function is:

.

As could be seen from the step response shown below:

The overshoot = 21.4% < 22%,

The settling time = 1.56 sec < 1.6 seconds,

The steady-state error, ** = 0.062 < .



**I-7 Instructor**

We first find the operating point. The imaginary part of the compensated closed-loop poles will be  . Since  . Hence the design point is -1.732+j3. Assume a PI controller  , to reduce the steady state error to zero. Using the system’s poles and the pole at zero and the zero od the ideal integral compensator, the summation of angles to the design point is

-218.077°. Hence the ideal derivative compensator must contribute 218.077°-180°=38.077°. Using the geometry below  .

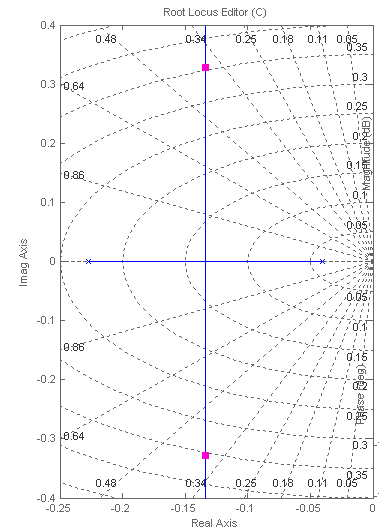


The PID controller is thus  . Using all poles and zeros of the system and PID controller, the gain at the design point is  . Searching the real axis segment , a higher order pole is found at -0.694. A simulation of the system shows that the actual obtained  sec and the  (compared to the  specified by  ).

**SOLUTIONS TO DESIGN PROBLEMS**

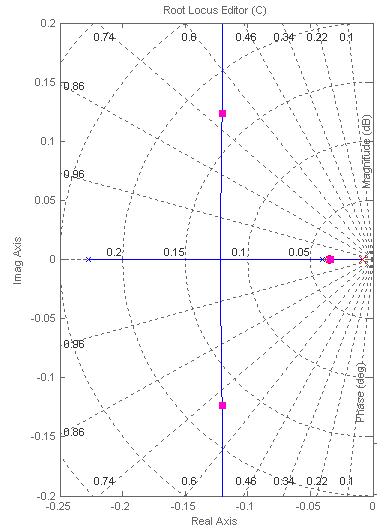
**I-9 Instructor**

With , **** giving. The root locus is:



Note that the damping factor 

We start by calculating the required from the steady state error requirement or . So . Arbitrarily let  which gives . Then the loop gain is adjusted to obtain , but did not change much in this design. We have . The root locus is:



The time domain simulation can be done as follows:

>> syms s

>> s=tf('s');

>> G=0.0187/(s^2+0.267\*s+0.00908);

>> Gc=(s+0.035)/(s+0.008);

>> T1=G/(1+G);

>> T2=G\*C/(1+G\*C);

>> step(T1,T2)



**I-10 Instructor**

**a**.  ;  ,  ,  sec,  , 

**b**.  ;  ,  , A 15% overshoot corresponds to  . Since the specification requires  , we have that  ,  and since  we get  . We also have  or  .

**c**. Uncompensated:  . so  ,  .

Compensated:  ,  , 

**I-11 Instructor**

Consider only the minor loop. Searching along the 143.13o line ( = 0.8), locate the minor-loop dominant poles at -7.74 ± j5.8 with Kf = 36.71. Searching the real axis segments for Kf = 36.71 locates a higher-order pole at - 0.535. Using the minor-loop poles at -7.74 ± j5.8 and - 0.535 as the open-loop poles (the open-loop zero at the origin is not a closed-loop zero) for the entire system, search along the 135o line ( = 0.707; 4.32% overshoot) and find the dominant second-order poles at

- 4.38 + j4 .38 with K = 227.91. Searching the real axis segment locates a higher-order pole at -7.26.

Uncompensated system performance: Setting Kf = 0 and searching along the 135o line (4.32% overshoot) yields -2.39 + j2.39 as the point on the root locus with K = 78.05. Searching the real axis segments of the root locus for K = 78.05 locates a higher-order pole at -11.2. The following table compares the predicted uncompensated characteristics with the predicted compensated characteristics.

**Uncompensated** **Compensated**

G(s) = G(s) =

Dominant poles: -2.39 + j2.39 Dominant poles: - 4.38 + j4 .38

 = 0.707  = 0.707

= 4.32% = 4.32%

n = = 3.38 rad/s n = = 6.19 rad/s

Ts = = 1.67 seconds Ts = = 0.91 second

Tp = = 1.31 seconds Tp = = 0.72 second

Kp = = 1.56 Kp = = 4.56

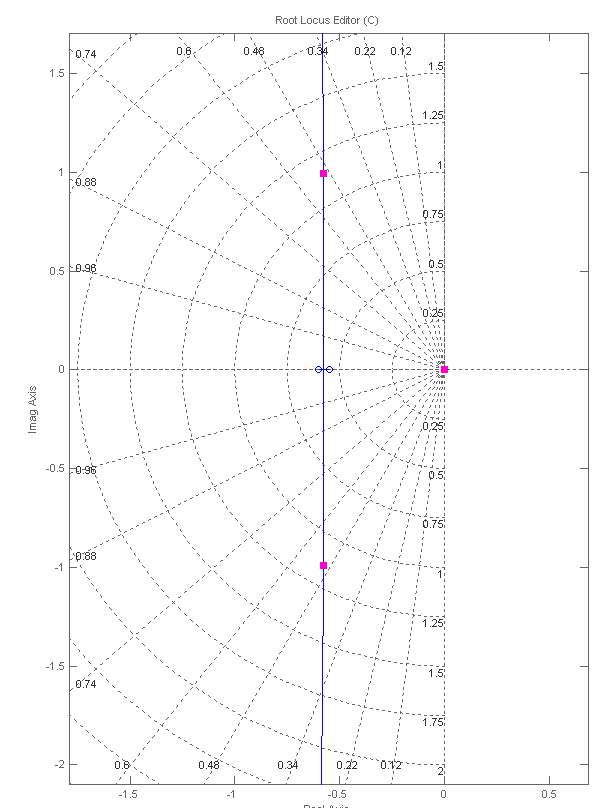
Higher-order pole: -11.22 Higher-order pole: -7.26

Second-order approximation OK Higher-order pole not 5x further from imaginary axis than dominant poles.

Simulate to be sure of the performance.

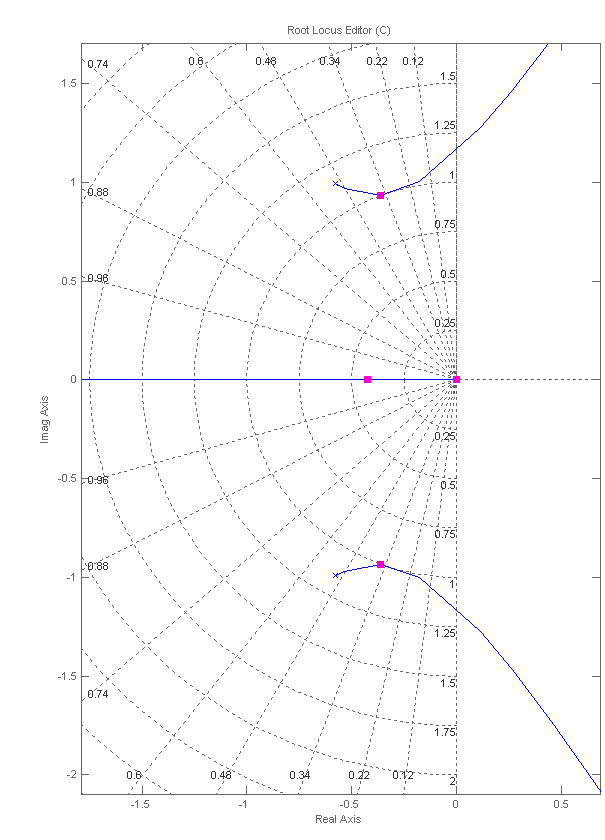
**I-12 Instructor**

a. The open loop transmission for the minor loop is  . The root locus is shown next where K=132results in the desired 

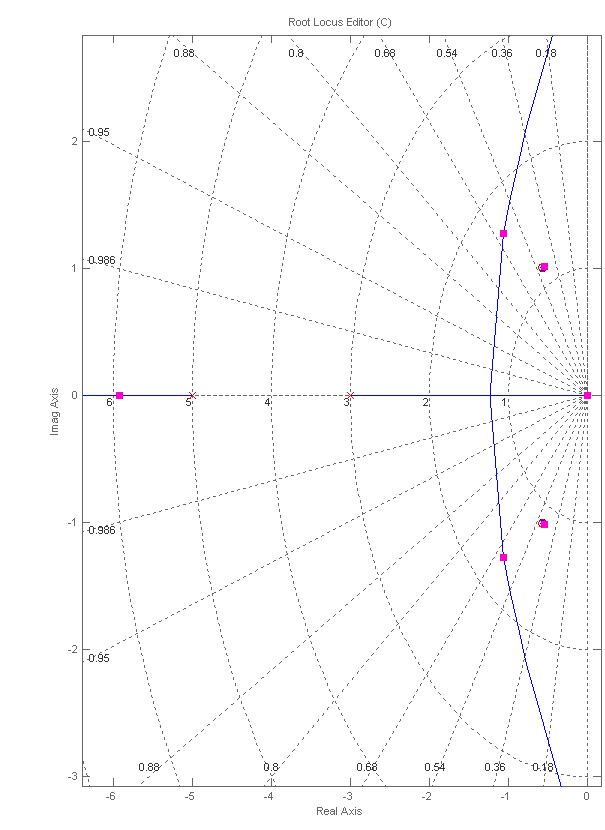


**b.**

With the value of K calculated in part a, the overall open loop transfer function from the main loop point view is  which results in an origin pole zero cancellation and an integrator at the origin. The uncompensated root locus is shown next:



With the two complex conjugate poles it is necessary to use a notch filter to obtain the required . For design the compensator zeros are chosen close to the open loop poles and the poles are placed arbitrarily on the real axis, then the gain is adjusted to get the required damping factor. To satisfy the settling time requirement, the real poles are adjusted interactively in sisotool until their real part is ≈ 1. The resulting compensator is: . The resulting root locus is:



**c.**

>> syms s

>> s=tf('s');

>> G1=574.98/s/(s^2+14.24\*s+3447.91);

>> H=0.046\*s\*(s^2+1.15\*s+0.33);

>> Gml=132\*G1/(1+132\*G1\*H);

>> Gc=0.738\*(s^2+1.15\*s+1.34)/(s+3)/(s+5);

>> T=Gc\*Gml/(1+Gc\*Gml);

>> step(T)

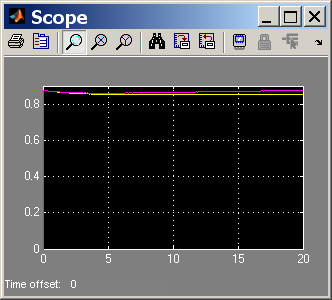


**I-13 Instructor**

1. We have that . The steady state error for a unit step input is
2. The required , which corresponds to a . Following the procedure of Section 9.2, let . Arbitrarily let , so . The resulting compensator is
3. We decided to use SIMULINK for this simulation. The block diagram is given by:



The resulting simulation shows the improvement in steady state error without affecting much the transient response.



**I-14 Instructor**

**a**. Uncompensated System: A 5% overshoot corresponds to a  damping factor. Searching along the line for , we find that the operating point is  with a gain of . The expected  sec.

**b**. PID Compensation: A reduction of 20% is the settling time means that the new  sec . Thus the real part of the new operating point will be , and since ,  . The operating point is 

PD design: The sum of pole angles to the new operating point is  so the PD zero must provide  of phase lead. The location of the zero is obtained by solving  as shown in the geometry below, resulting in .



-zc

Thus  . Using the root locus it is verified that the operating point is achieved with 

PI design: A pole is added at the origin and an arbitrarily placed zero is added at -0.02. Thus  . Searching along the line the new operating point is  achieved when . The design has a new closed loop pole at -0.02. The expected settling time is sec with  .

So 

**c.** A unit step response of the simulation is shown below where it can be seen that  and  sec.

