T H I R T E E N

Digital Control Systems

SOLUTIONS TO CASE STUDIES CHALLENGES

Antenna Control: Transient Design via Gain

a. From the answer to the antenna control challenge in Chapter 5, the equivalent forward transfer function found by neglecting the dynamics of the power amplifier, replacing the pots with unity gain, and including the integration in the sample-and-hold is

$$G_e(s) = \frac{0.16K}{s^2(s+1.32)}$$

But,

$$\frac{1}{s^2(s+1.32)} = -0.57392\frac{1}{s} + 0.57392\frac{1}{s+1.32} + 0.75758\frac{1}{s^2}$$

$$G_z = -0.57392\frac{z}{z-1} + 0.57392\frac{z}{z-e^{-1.32T}} + 0.75758\frac{Tz}{(z-1)^2}$$

$$T = 0.1$$

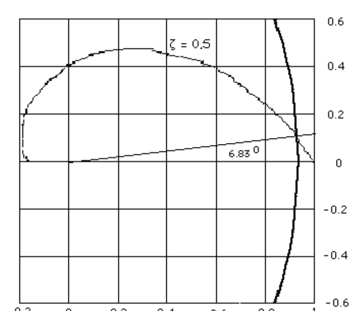
$$G_z = -0.57392\frac{z}{z-1} + 0.57392\frac{z}{z-e^{-0.132}} + 0.75758\frac{0.1z}{(z-1)^2}$$

$$G_z = 0.0047871\frac{(z+0.95696)z}{(z-1)^2(z-0.87634)}$$

Thus, $G_e(z) = 0.16K\frac{z-1}{z}G_z$, or,

$$G_e(z) = 7.659\times10^{-4}K\frac{(z+0.95696)}{(z-1)(z-0.87634)}$$

b. Draw the root locus and overlay it over the $\zeta = 0.5$ (i.e. 16.3% overshoot) curve.



We find that the root locus crosses at approximately 0.93 ± j0.11 with $7.659\text{x}10^{-4}\text{K} = 8.63\text{x}10^{-3}$.

Hence, K = 11.268.

c.

$$K_v = \frac{1}{T}\lim_{z\to 1}(z-1)G_e(z) = \frac{(7.659\text{x}10^{-4}\text{K})(1.95696)}{0.12366} = 0.1366;$$

$$e(\infty) = \frac{1}{K_v} = 7.321$$

d.

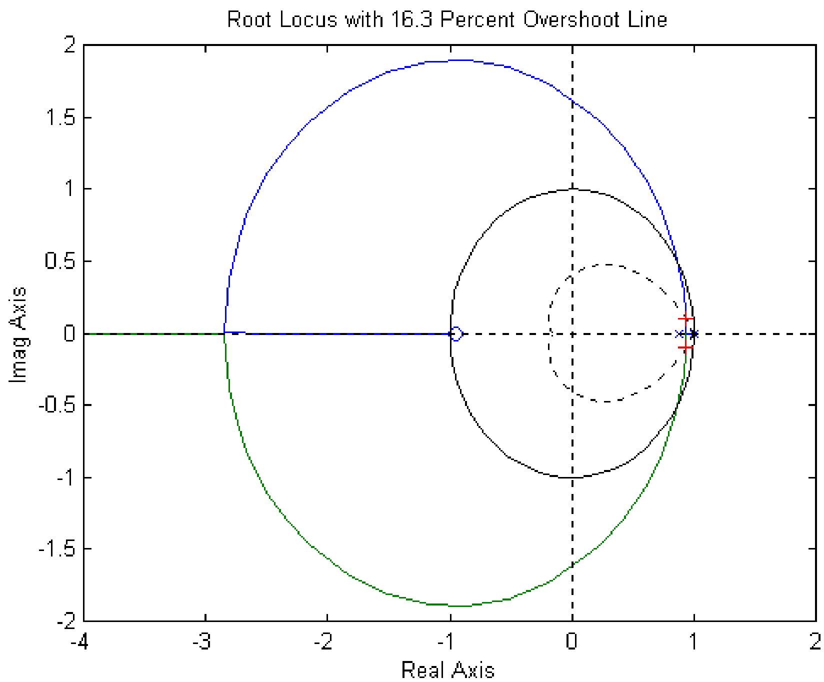
Program:

```
T=0.1;                       %Input sampling time
numf=0.16;                   %Numerator of F(s)
denf=[1 1.32 0 0];           %Denominator of F(s)
'F(s)'                       %Display label
F=tf(numf,denf)              %Display F(s)
numc=conv([1 0],numf);       %Differentiate F(s) to compensate
                             %for c2dm which assumes series zoh
denc=denf;                   %Denominator of continuous system
                             %same as denominator of F(s)
C=tf(numc,denc);             %Form continuous system, C(s)
C=minreal(C,1e-10);          %Cancel common poles and zeros
D=c2d(C,T,'zoh');            %Convert to z assuming zoh
'F(z)'
D=minreal(D,1e-10)           %Cancel common poles and zeros and display
rlocus(D)
pos=(16.3);
z=-log(pos/100)/sqrt(pi^2+[log(pos/100)]^2);
zgrid(z,0)
title(['Root Locus with ' , num2str(pos), ' Percent Overshoot Line'])
[K,p]=rlocfind(D)            %Allows input by selecting point on
                             %graphic
```

Computer response:

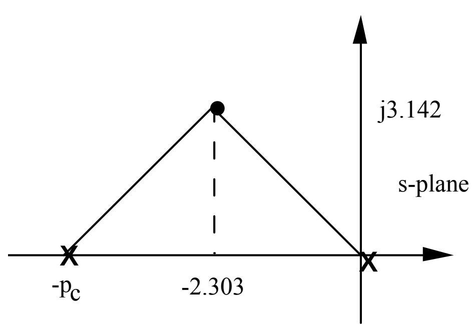
```
ans =

F(s)

Transfer function:
     0.16
--------------
s^3 + 1.32 s^2

ans =

F(z)

Transfer function:
0.0007659 z + 0.000733
----------------------
z^2 - 1.876 z + 0.8763

Sampling time: 0.1
Select a point in the graphics window

selected_point =

  9.2969e-001 +1.0219e-001i

K =

  9.8808e+000
```

```
p =

  9.3439e-001 +1.0250e-001i
  9.3439e-001 -1.0250e-001i
```



Antenna Control: Digital Cascade Compensator Design

a. Let the compensator be $KG_c(s)$ and the plant be $G_p(s) = \dfrac{0.16}{s(s+1.32)}$. For 10% overshoot and a peak time of 1 second, $\zeta = 0.591$ and $\omega_n = 3.895$, which places the dominant poles at $-2.303 \pm j3.142$. If we place the compensator zero at -1.32 to cancel the plant's pole, then the following geometry results.



Hence, $p_c = 4.606$. Thus, $G_c(s) = \dfrac{K(s+1.32)}{(s+4.606)}$ and $G_c(s)G_p(s) = \dfrac{0.16K}{s(s+4.606)}$. Using the product of pole lengths to find the gain, $0.16K = (3.896)^2$, or $K = 94.87$. Hence,

$G_c(s) = \dfrac{94.87(s+1.32)}{(s+4.606)}$. Using a sampling interval of 0.01 s, the Tustin transformation of $G_c(s)$ is $G_c(z) = \dfrac{93.35(z-0.9869)}{(z-0.955)} = \dfrac{93.35z-92.12}{z-0.955}$.

b. Cross multiplying**,**

$$(z - 0.955)X(z) = (93.35z - 92.12)E(z)$$

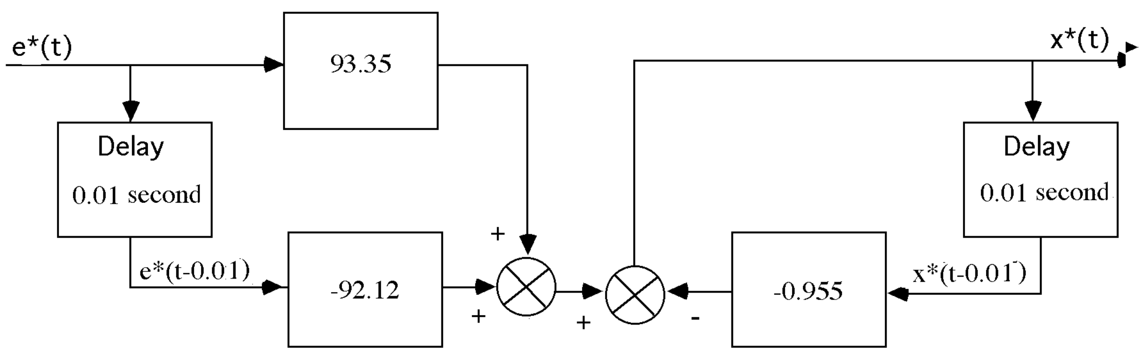
Solving for the highest power of z operating on *X(z)*,

$$zX(z) = (93.35z - 92.12)E(z) + 0.955X(z)$$

Solving for *X(z)*,

$$X(z) = (93.35 - 92.12z^{-1})E(z) + 0.955z^{-1}X(z)$$

Implementing this equation as a flowchart yields the following diagram



c.

Program:

```
's-plane lead design for Challenge - Lead Comp'
clf                             %Clear graph on screen.
'Uncompensated System'          %Display label.
numg=0.16;                      %Generate numerator of G(s).
deng=poly([0 -1.32]);           %Generate denominator of G(s).
'G(s)'                          %Display label.
G=tf(numg,deng);                %Create G(s).
Gzpk=zpk(G)                     %Display G(s).
pos=input('Type desired percent overshoot ');
                                %Input desired percent overshoot.
z=-log(pos/100)/sqrt(pi^2+[log(pos/100)]^2);
                                %Calculate damping ratio.
Tp=input('Type Desired Peak Time ');
                                %Input desired peak time.
wn=pi/(Tp*sqrt(1-z^2));         %Evaluate desired natural frequency.
b=input('Type Lead Compensator Zero, (s+b). b=  ');
                                %Input lead compensator zero.
done=1;                         %Set loop flag.

while done==1                   %Start loop for trying lead
                                %compensator pole.
a=input('Enter a Test Lead Compensator Pole, (s+a). a =     ');
                                %Enter test lead compensator pole.
numge=conv(numg,[1 b]);         %Generate numerator of Gc(s)G(s).
denge=conv([1 a],deng);         %Generate denominator of Gc(s)G(s).
Ge=tf(numge,denge);             %Create Ge(s)=Gc(s)G(s).
clf                             %Clear graph on screen.
```

```
rlocus(Ge)                      %Plot compensated root locus with
                                %test lead compensator pole.
axis([-5 2 -8 8]);              %Change axes ranges.
sgrid(z,wn)                     %Overlay grid on lead-compensated
                                %root locus.
title(['Lead-Compensated Root Locus with ' , num2str(pos),...
'%  Overshoot Line, Lead Pole at  ', num2str(-a),...
' and Required Wn'])            %Add title to lead-compensated root
                                %locus.
done=input('Are you done? (y=0,n=1)  ');
                                %Set loop flag.
end                             %End loop for trying compensator
                                %pole.
[K,p]=rlocfind(Ge);             %Generate gain, K, and closed-loop
                                %poles, p, for point selected
                                %interactively on the root locus.
'Gc(s)'                         %Display label.
Gc=K*tf([1 b],[1 a])            %Display lead compensator.
'Gc(s)G(s)'                     %Display label.
Ge                              %Display Gc(s)G(s).
'Closed-loop poles = '          %Display label.
p                               %Display lead-compensated system's
                                %closed-loop poles.
f=input('Give pole number that is operating point   ');
                                %Choose lead-compensated system
                                %dominant pole.
'Summary of estimated specifications for selected point on lead'
'compensated root locus'        %Display label.
operatingpoint=p(f)             %Display lead-compensated dominant
                                %pole.
gain=K                          %Display lead-compensated gain.
estimated_settling_time=4/abs(real(p(f)))
                                %Display lead-compensated settling
                                %time.
estimated_peak_time=pi/abs(imag(p(f)))
                                %Display lead-compensated peak time.
estimated_percent_overshoot=pos %Display lead-compensated percent
                                %overshoot.
estimated_damping_ratio=z       %Display lead-compensated damping
                                %ratio.
estimated_natural_frequency=sqrt(real(p(f))^2+imag(p(f))^2)
                                %Display lead-compensated natural
                                %frequency.
s=tf([1 0],1);                  %Create transfer function, "s".
sGe=s*Ge;                       %Create sGe(s) to evaluate Kv.
sGe=minreal(sGe);               %Cancel common poles and zeros.
Kv=dcgain(K*sGe)                %Display lead-compensated Kv.
ess=1/Kv                        %Display lead-compensated steady-
                                %state error for unit ramp input.
'T(s)'                          %Display label.
T=feedback(K*Ge,1)              %Create and display lead-compensated
                                %T(s).
'Press any key to continue and obtain the lead-compensated step'
'response'                      %Display label
pause
step(T)                         %Plot step response for lead
                                %compensated system.
title(['Lead-Compensated System with  ' ,num2str(pos),'% Overshoot'])
                                %Add title to step response of PD
                                %compensated system.
pause

'z-plane conversion for Challenge - Lead Comp'
clf                             %Clear graph.
'Gc(s) in polynomial form'      %Print label.
Gcs=Gc                          %Create Gc(s) in polynomial form.
'Gc(s) in polynomial form'      %Print label.
```

```
Gcszpk=zpk(Gcs)                  %Create Gc(s) in factored form.
'Gc(z) in polynomial form via Tustin Transformation'
                                 %Print label.
Gcz=c2d(Gcs,1/100,'tustin')      %Form Gc(z) via Tustin
                                 %transformation.
'Gc(z) in factored form via Tustin Transformation'
                                 %Print label.
Gczzpk=zpk(Gcz)                  %Show Gc(z) in factored form.
'Gp(s) in polynomial form'       %Print label.
Gps=G                            %Create Gp(s) in polynomial form.
'Gp(s) in factored form'         %Print label.
Gpszpk=zpk(Gps)                  %Create Gp(s) in factored form.
'Gp(z) in polynomial form'       %Print label.
Gpz=c2d(Gps,1/100,'zoh')         %Form Gp(z) via zoh transformation.
'Gp(z) in factored form'         %Print label.
Gpzzpk=zpk(Gpz)                  %Form Gp(z) in factored form.
pole(Gpz)                        %Find poles.
Gez=Gcz*Gpz;                     %Form Ge(z) = Gc(z)Gp(z).
'Ge(z) = Gc(z)Gp(z) in factored form'
                                 %Print label.
Gezzpk=zpk(Gez)                  %Form Ge(z) in factored form.
'z-1'                            %Print label.
zm1=tf([1 -1],1,1/100)           %Form z-1.
zm1Gez=minreal(zm1*Gez,.00001);
'(z-1)Ge(z)'                     %Print label.
zm1Gezzpk=zpk(zm1Gez)
pole(zm1Gez)
Kv=300*dcgain(zm1Gez)
Tz=feedback(Gez,1)
step(Tz)
title('Closed-Loop Digital Step Response')
                                 %Add title to step response.
```

Computer response:

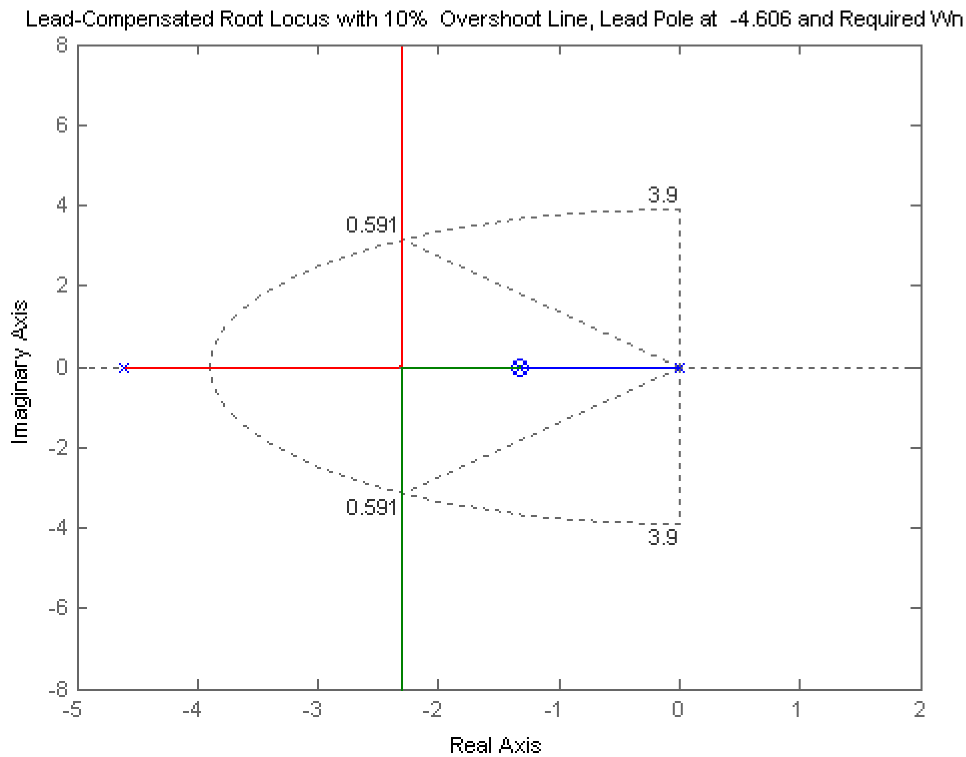
```
ans =

s-plane lead design for Challenge - Lead Comp


ans =

Uncompensated System


ans =

G(s)


Zero/pole/gain:
   0.16
----------
s (s+1.32)

Type desired percent overshoot 10
Type Desired Peak Time 1
Type Lead Compensator Zero, (s+b). b=  1.32
Enter a Test Lead Compensator Pole, (s+a). a =     4.606
Are you done? (y=0,n=1)  0
Select a point in the graphics window

selected_point =

  -2.3045 + 3.1056i


ans =
```

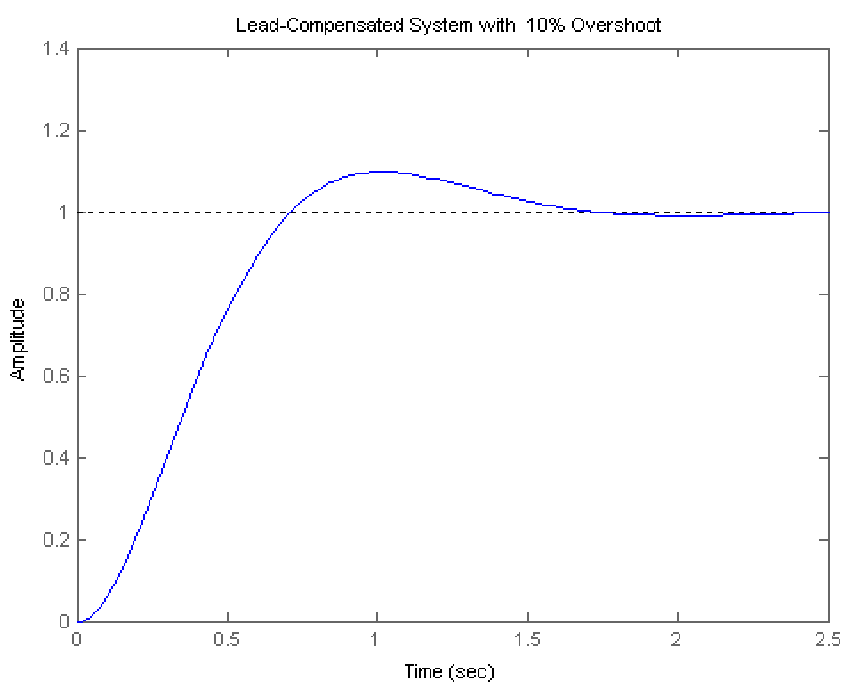
```
Gc(s)

Transfer function:
93.43 s + 123.3
---------------
   s + 4.606

ans =

Gc(s)G(s)

Transfer function:
    0.16 s + 0.2112
------------------------
s^3 + 5.926 s^2 + 6.08 s

ans =

Closed-loop poles =

p =

  -2.3030 + 3.1056i
  -2.3030 - 3.1056i
  -1.3200

Give pole number that is operating point   1

ans =

Summary of estimated specifications for selected point on lead

ans =

compensated root locus

operatingpoint =

  -2.3030 + 3.1056i

gain =

   93.4281

estimated_settling_time =

    1.7369

estimated_peak_time =

    1.0116

estimated_percent_overshoot =

    10
```

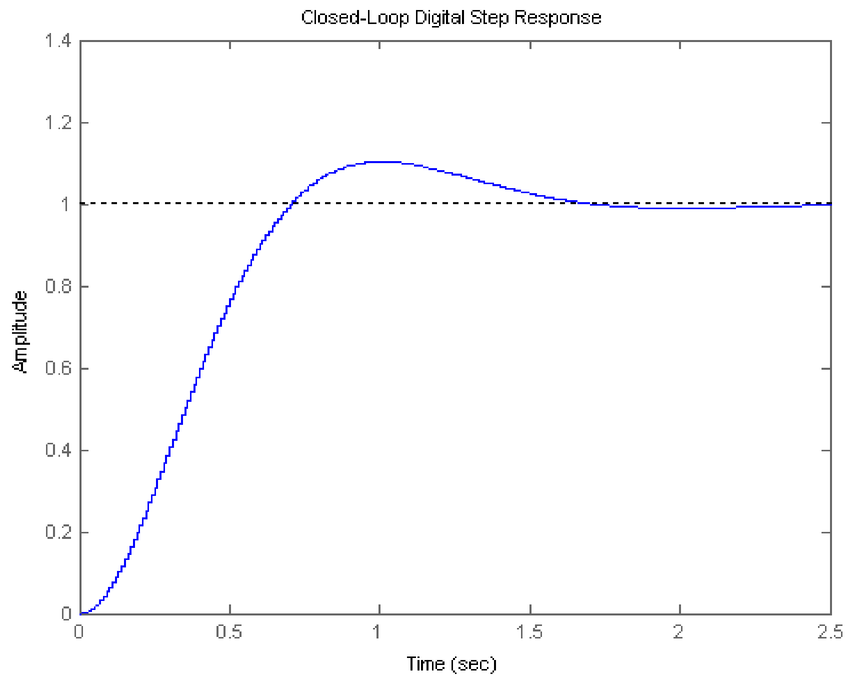
```
estimated_damping_ratio =

    0.5912


estimated_natural_frequency =

    3.8663


Kv =

    3.2454


ess =

    0.3081


ans =

T(s)


Transfer function:
         14.95 s + 19.73
---------------------------------
s^3 + 5.926 s^2 + 21.03 s + 19.73


ans =

Press any key to continue and obtain the lead-compensated step


ans =

response


ans =

z-plane conversion for Challenge - Lead Comp


ans =

Gc(s) in polynomial form


Transfer function:
93.43 s + 123.3
---------------
   s + 4.606


ans =

Gc(s) in polynomial form


Zero/pole/gain:
93.4281 (s+1.32)
----------------
```

```
   (s+4.606)

ans =

Gc(z) in polynomial form via Tustin Transformation


Transfer function:
91.93 z - 90.72
---------------
   z - 0.955

Sampling time: 0.01

ans =

Gc(z) in factored form via Tustin Transformation


Zero/pole/gain:
91.9277 (z-0.9869)
------------------
    (z-0.955)

Sampling time: 0.01

ans =

Gp(s) in polynomial form


Transfer function:
    0.16
------------
s^2 + 1.32 s


ans =

Gp(s) in factored form


Zero/pole/gain:
   0.16
----------
s (s+1.32)


ans =

Gp(z) in polynomial form


Transfer function:
7.965e-006 z + 7.93e-006
------------------------
 z^2 - 1.987 z + 0.9869

Sampling time: 0.01

ans =

Gp(z) in factored form


Zero/pole/gain:
```

```
7.9649e-006 (z+0.9956)
----------------------
   (z-1) (z-0.9869)

Sampling time: 0.01

ans =

    1.0000
    0.9869


ans =

Ge(z) = Gc(z)Gp(z) in factored form


Zero/pole/gain:
0.0007322 (z+0.9956) (z-0.9869)
-------------------------------
  (z-1) (z-0.9869) (z-0.955)

Sampling time: 0.01

ans =

z-1


Transfer function:
z - 1

Sampling time: 0.01

ans =

(z-1)Ge(z)


Zero/pole/gain:
0.0007322 (z+0.9956)
--------------------
     (z-0.955)

Sampling time: 0.01

ans =

    0.9550


Kv =

    9.7362


Transfer function:
0.0007322 z^2 + 6.387e-006 z - 0.0007194
----------------------------------------
   z^3 - 2.941 z^2 + 2.884 z - 0.9432

Sampling time: 0.01
```


Lead-Compensated Root Locus with 10% Overshoot Line, Lead Pole at -4.606 and Required Wn
Imaginary Axis
Real Axis
0.591
0.591
3.9
3.9
8
6
4
2
0
-2
-4
-6
-8
-5
-4
-3
-2
-1
0
1
2


Lead-Compensated System with 10% Overshoot
Amplitude
Time (sec)
1.4
1.2
1
0.8
0.6
0.4
0.2
0
0
0.5
1
1.5
2
2.5


Closed-Loop Digital Step Response
Amplitude
Time (sec)
0
0.2
0.4
0.6
0.8
1
1.2
1.4
0
0.5
1
1.5
2
2.5

ANSWERS TO REVIEW QUESTIONS

1. (1) Supervisory functions external to the loop; (2) controller functions in the loop

2. (1) Control of multiple loops by the same hardware; (2) modifications made with software, not hardware; (3) more noise immunity (4) large gains usually not required

3. Quantization error; conversion time

4. An ideal sampler followed by a sample-and-hold

5. $z = e^{sT}$

6. The value of the time waveform only at the sampling instants

7. Partial fraction expansion; division to yield power series

8. Partial fraction

9. Division to yield power series

10. The input must be sampled; the output must be either sampled or thought of as sampled.

11. c(t) is c*(t) = c(kT), i.e. the output only at the sampling instants.

12. No; the waveform is only valid at the sampling instants. Instability may be apparent if one could only see between the sampling instants. The roots of the denominator of G(z) must be checked to see that they are within the unit circle.

13. A sample-and-hold must be present between the cascaded systems.

14. Inside the unit circle

15. Raible table; Jury's stability test

16. z=+1

17. There is no difference.

18. Map the point back to the s-plane. Since $z = e^{sT}$, s = (1/T) ln z. Thus, σ = (1/T) ln (Re z), and ω = (1/T) ln (Im z).

19. Determine the point on the s-plane and use $z = e^{sT}$. Thus, Re z = $e^{\sigma T}$ cos ω, and Im z = $e^{\sigma T}$ sin ω.

20. Use the techniques described in Chapters 9 and 11 and then convert the design to a digital compensator using the Tustin transformation.

21. Both compensators yield the same output at the sampling instants.

SOLUTIONS TO PROBLEMS

1.

a. $f(t) = e^{-at}$; $f^*(t) = \sum_{k=0}^{\infty} e^{-akT}\delta(t-kT)$; $F^*(s) = \sum_{k=0}^{\infty} e^{-akT} e^{-kTs} = 1 + e^{-aT} e^{-Ts} + e^{-a2T} e^{-2Ts} + \ldots$ Thus,

$F(z) = 1 + e^{-aT} z^{-1} + e^{-a2T} z^{-2} + \ldots = 1 + x^{-1} + x^{-2} + \ldots$ where $x = e^{-aT} z^{-1}$.

But, $F(z) = \dfrac{1}{1 - x^{-1}} = \dfrac{1}{1 - e^{-aT} z^{-1}} = \dfrac{z}{z - e^{-aT}}$.

b. $f(t) = u(t)$; $f^*(t) = \sum_{k=0}^{\infty} \delta(t-kT)$; $F^*(s) = \sum_{k=0}^{\infty} e^{-kT s} = 1 + e^{-Ts} + e^{-2Ts} + \ldots$

Thus, $F(z) = 1 + z^{-1} + z^{-2} + ..$ Since $\dfrac{1}{1 - z^{-1}} = 1 + z^{-1} + z^{-2} + z^{-3}$, $F(z) = \sum_{k=0}^{\infty} z^{-k} = \dfrac{1}{1 - z^{-1}} =$

$\dfrac{z}{z - 1}$.

c. $f(t) = t^2 e^{-at}$; $f^*(t) = \sum_{k=0}^{\infty} (kT)^2 e^{-akT}\delta(t-kT)$; $F^*(s) = T^2 \sum_{k=0}^{\infty} k^2 e^{-akT} e^{-kTs}$

$= T^2 \sum_{k=0}^{\infty} k^2 (e^{-(s+a)T})^k = T^2 \sum_{k=0}^{\infty} k^2 x^k = T^2(x + 4x^2 + 9x^3 + 16x^4 + \ldots)$, where $x = e^{-(s+a)T}$.

Let $s_1 = x + 4x^2 + 9x^3 + 16x^4 + \ldots$ Thus, $xs_1 = x^2 + 4x^3 + 9x^4 + 16x^5 + \ldots$

Let $s_2 = s_1 - xs_1 = x + 3x^2 + 5x^3 + 7x^4 + \ldots$Thus, $xs_2 = x^2 + 4x^3 + 9x^4 + 16x^3 + \ldots$

Let $s_3 = s_2 - xs_2 = x + 2x^2 + 2x^3 + 2x^4 + \ldots$ Thus $xs_3 = x^2 + 2x^3 + 2x^4 + 2x^3 + \ldots$

Let $s_4 = s_3 - xs_3 = x + x^2$.

Solving for s_3,

$$s_3 = \frac{x + x^2}{1 - x}$$

and

$$s_2 = \frac{s_3}{1 - x} = \frac{x + x^2}{(1 - x)^2}$$

and

$$s_1 = \frac{s_2}{1 - x} = \frac{x + x^2}{(1 - x)^3}$$

Thus

$$F^*(s) = T^2 s_1 = T^2 \frac{x + x^2}{(1 - x)^3} = T^2 \frac{(e^{-(s+a)T} + e^{-2(s+a)T})}{(1 - e^{-(s+a)T})^3} =$$

$$\frac{T^2[z^{-1}e^{-aT}+z^{-2}e^{-2aT}]}{z^{-3}(z-e^{-aT})^3}=\frac{T^2ze^{-aT}[z+e^{-aT}]}{(z-e^{-aT})^3}$$

d. $f(t) = \cos(\omega kT)$; $f^*(t) = \sum_{k=0}^{\infty}\cos(\omega kT)\delta(t-kT)$; $F^*(s) = \sum_{k=0}^{\infty}\cos(\omega kT)e^{-kTs}$

$$= \sum_{k=0}^{\infty}\frac{(e^{j\omega kT}+e^{-j\omega kT})e^{-kTs}}{2} = \frac{1}{2}\sum_{k=0}^{\infty}(e^{T(s-j\omega)})^{-k}+(e^{T(s+j\omega)})^{-k}$$

But,

$$\sum_{k=0}^{\infty}x^{-k} = \frac{1}{1 - x^{-1}}\ .$$

Thus,

$$F^*(s) = \frac{1}{2}\left[\frac{1}{1-e^{-T(s-j\omega)}}+\frac{1}{1-e^{-T(s+j\omega)}}\right]= \frac{1}{2}\left[\frac{2-e^{-Ts}(e^{j\omega T}+e^{-j\omega T})}{1-e^{-T(s-j\omega)}-e^{-T(s+j\omega)})+e^{-T(s-j\omega)}e^{-T(s+j\omega)}}\right]$$

$$= \frac{1}{2}\left[\frac{2-e^{-Ts}(2\cos(\omega T)}{1-e^{-Ts}(e^{j\omega T}+e^{-j\omega T})+e^{-2Ts}}\right]=\frac{1-z^{-1}\cos(\omega T)}{1-2z^{-1}\cos(\omega T)+z^{-2}}$$

Therefore,

$$F(z)=\frac{z(z-\cos(\omega T))}{z^2-2z\cos(\omega T)+1}$$

2.

Program:

```
syms T a w n                %Construct symbolic objects for
                            %'T', 'a','w', and 'n'.
'(a)'                       %Display label.
'f(kT)'                     %Display label.
f=exp(-a*n*T);              %Define f(kT).
pretty(f)                   %Pretty print f(kT)
'F(z)'                      %Display label.
F=ztrans(f);                %Find z-transform, F(z).
pretty(F)                   %Pretty print F(z).

'(b)'                       %Display label.
'f(kT)'                     %Display label.
f=exp(-0*n*T);              %Define f(kT)
pretty(f)                   %Pretty print f(kT)
'F(z)'                      %Display label.
F=ztrans(f);                %Find z-transform, F(z).
pretty(F)                   %Pretty print F(z).

'(c)'                       %Display label.
'f(kT)'                     %Display label.
f=(n*T)^2*exp(-a*n*T);      %Define f(kT)
pretty(f)                   %Pretty print f(kT)
'F(z)'                      %Display label.
F=ztrans(f);                %Find z-transform, F(z).
pretty(F)                   %Pretty print F(z).

'(d)'                       %Display label.
'f(kT)'                     %Display label.
f=cos(w*n*T);               %Define f(kT)
pretty(f)                   %Pretty print f(kT)
'F(z)'                      %Display label.
```

```
F=ztrans(f);                   %Find z-transform, F(z).
pretty(F)                      %Pretty print F(z).
```

Computer response:

```
ans =

(a)

ans =

f(kT)


                                  exp(-a n T)
ans =

F(z)

                                       z
                          -------------------------
                                    /     z         \
                          exp(-a T) |--------- - 1|
                                    \exp(-a T)     /
ans =

(b)

ans =

f(kT)

                                       1
ans =



F(z)

                                       z
                                     -----
                                     z - 1
ans =

(c)

ans =

f(kT)

                                 2  2
                                n  T  exp(-a n T)
ans =

F(z)

                         2
                        T  z exp(-a T) (z + exp(-a T))
                        ------------------------------
                                                  3
                               (z - exp(-a T))
ans =

(d)

ans =
```

```
f(kT)

                              cos(w n T)

ans =

F(z)

                         (z - cos(w T)) z
                       ---------------------
                        2
                       z  - 2 z cos(w T) + 1
```

3.

a.

$$F(z)=\frac{z(z+3)(z+5)}{(z-0.4)(z-0.6)(z-0.8)}$$

$$\frac{F(z)}{z}=\frac{229.5}{z-0.4}-\frac{504}{z-0.6}+\frac{275.5}{z-0.8}$$

$$F(z)=\frac{229.5z}{z-0.4}-\frac{504z}{z-0.6}+\frac{275.5z}{z-0.8}$$

$$f(kT)=229.5(0.4)^k-504(0.6)^k+275.5(0.8)^k,\ k=0,\ 1,\ 2,\ 3,\ldots$$

b.

$$F(z)=\frac{(z+0.2)(z+0.4)}{(z-0.1)(z-0.5)(z-0.9)}$$

$$\frac{F(z)}{z}=-\frac{1.778}{z}+\frac{4.6875}{z-0.1}-\frac{7.875}{z-0.5}+\frac{4.9653}{z-0.9}$$

$$F(z)=-1.778+\frac{4.6875z}{z-0.1}-\frac{7.875z}{z-0.5}+\frac{4.9653z}{z-0.9}$$

$$f(kT)=4.6875(0.1)^k-7.875(0.5)^k+4.9653(0.9)^k,\ k=1,2,\ 3,\ldots$$

c.

$$F(z)=\frac{(z+1)(z+0.3)(z+0.4)}{z(z-0.2)(z-0.5)(z-0.7)}$$

$$\frac{F(z)}{z}=\frac{(z+1)(z+0.3)(z+0.4)}{z^2(z-0.2)(z-0.5)(z-0.7)}$$

$$=\frac{38.1633}{z-0.7}-\frac{72}{z-0.5}+\frac{60}{z-0.2}-\frac{26.1633}{z}-\frac{1.7143}{z^2}$$

$$F(z)=\frac{38.1633z}{z-0.7}-\frac{72z}{z-0.5}+\frac{60z}{z-0.2}-26.1633-\frac{1.7143}{z}$$

$$F=38.1633(0.7)^k-72(0.5)^k+60(0.2)^k \quad \text{for } k=2,3,4,\ldots$$

$$=1 \text{ for } k=1$$

$$=0 \text{ for } k=0$$

4.

Program:

```
'(a)'
syms z k
F=vpa(z*(z+3)*(z+5)/((z-0.4)*(z-0.6)*(z-0.8)),4);
pretty(F)
f=vpa(iztrans(F),4);
pretty(f)
'(b)'
syms z k
F=vpa((z+0.2)*(z+0.4)/((z-0.1)*(z-0.5)*(z-0.9)),4);
pretty(F)
f=vpa(iztrans(F),4);
pretty(f)
'(c)'
syms z k
F=vpa((z+1)*(z+0.3)*(z+0.4)/(z*(z-0.2)*(z-0.5)*(z-0.7)),4);
pretty(F)
f=vpa(iztrans(F),4);
pretty(f)
```

Computer response:

```
ans =

(a)


                z (z + 3.) (z + 5.)
        -----------------------------------
        (z - .4000) (z - .6000) (z - .8000)

             n              n              n
    275.5 .8000  - 504.0 .6000  + 229.5 .4000

ans =

(b)


               (z + .2000) (z + .4000)
        -----------------------------------
        (z - .1000) (z - .5000) (z - .9000)

                        n              n              n
```

```
       -1.778 charfcn[0](n) + 4.965 .9000  - 7.875 .5000  + 4.688 .1000

ans =

(c)

                    (z + 1.) (z + .3000) (z + .4000)
                  ------------------------------------
                  z (z - .2000) (z - .5000) (z - .7000)

                                                                  n              n
  -1.714 charfcn[1](n) - 26.16 charfcn[0](n) + 38.16 .7000  - 72.00 .5000

                      n
         + 60.00 .2000
```

5.

a.

	By division	**By Formula**	
Instant	**Value**	**k**	**Value**
0	1	0	1
1	9.8	1	9.8
2	31.6	2	31.6
3	46.88	3	46.88
4	53.4016	4	53.4016
5	53.43488	5	53.43488
6	49.64608	6	49.64608
7	44.043776	7	44.043776
8	37.90637056	8	37.90637056
9	31.95798733	9	31.95798733
10	26.5581568	10	26.5581568
11	21.84639857	11	21.84639857
12	17.83896791	12	17.83896791
13	14.48905384	13	14.48905384
14	11.72227881	14	11.72227881
15	9.456567702	15	9.456567702
16	7.612550239	16	7.612550239
17	6.118437551	17	6.118437551
18	4.911796342	18	4.911796342
19	3.939668009	19	3.939668009
20	3.15787423	20	3.15787423
21	2.529983782	21	2.529983782
22	2.026197867	22	2.026197867
23	1.622284879	23	1.622284879
24	1.298623886	24	1.298623886
25	1.039376712	25	1.039376712
26	0.831787937	26	0.831787937
27	0.665602292	27	0.665602292
28	0.532584999	28	0.532584999
29	0.4261299	29	0.4261299
30	0.34094106	30	0.34094106

b.

By division		By Formula	
Instant	**Value**	**k**	**Value**
1	1	1	1.00002
2	2.1	2	2.100018
3	2.64	3	2.6400162
4	2.766	4	2.76601458
5	2.6859	5	2.685913122
6	2.51571	6	2.51572181
7	2.313354	7	2.313364629
8	2.1066276	8	2.106637166
9	1.90826949	9	1.908278099
10	1.723594881	10	1.723602629
11	1.554311564	11	1.554318538
12	1.400418494	12	1.40042477
13	1.261145687	13	1.261151336
14	1.13541564	14	1.135420724
15	1.022066337	15	1.022070912
16	0.919955834	16	0.919959951
17	0.828008315	17	0.828012021
18	0.745231516	18	0.745234852
19	0.670720381	19	0.670723383
20	0.603654351	20	0.603657053
21	0.54329192	21	0.543294352
22	0.48896423	22	0.488966419
23	0.440068558	23	0.440070528
24	0.396062078	24	0.39606385
25	0.356456058	25	0.356457653
26	0.320810546	26	0.320811982
27	0.288729538	27	0.28873083
28	0.259856608	28	0.259857771
29	0.233870959	29	0.233872006
30	0.210483869	30	0.210484811
31	0.189435485	31	0.189436333

c.

Instant	via Division	via Closed Form Expression
0		0
1	1	1
2	3.1	3.100017
3	4.57	4.5700119
4	4.759	4.75900833
5	4.1833	4.183305831
6	3.36871	3.368714082
7	2.581177	2.581179857
8	1.9189399	1.9189419
9	1.39943113	1.39943253
10	1.007711431	1.007712411
11	0.71945743	0.719458116
12	0.510650836	0.510651317
13	0.360971088	0.360971424
14	0.254437549	0.254437785
15	0.178985186	0.178985351
16	0.125729082	0.125729198
17	0.088230084	0.088230165
18	0.061870922	0.061870978
19	0.043364577	0.043364617
20	0.03038267	0.030382697
21	0.021281602	0.021281621
22	0.014903988	0.014904001
23	0.010436225	0.010436234
24	0.007307074	0.00730708

6.

a.

$$G(s)=\frac{(s+4)}{(s+2)(s+5)}=\frac{0.6667}{s+2}+\frac{0.3333}{s+5}$$

$$G(z)=\frac{0.6667z}{z-e^{-2T}}+\frac{0.3333z}{z-e^{-5T}}$$

For $T = 0.5$ s,

$$G(z)=\frac{0.6667z}{z-0.3679}+\frac{0.3333z}{z-0.082085}=\frac{z(z-0.1774)}{(z-0.3679)(z-0.082085)}$$

b.

$$G(s)=\frac{(s+1)(s+2)}{s(s+3)(s+4)}=\frac{0.1667}{s}-\frac{0.6667}{s+3}+\frac{1.5}{s+4}$$

$$G(z)=\frac{0.1667z}{z-1}-\frac{0.6667z}{z-e^{-3T}}+\frac{1.5z}{z-e^{-4T}}$$

For $T = 0.5$ s,

$$G(z)=\frac{0.1667z}{z-1}-\frac{0.6667z}{z-0.22313}+\frac{1.5z}{z-0.13534}=\frac{z(z-0.29675)(z-0.8408)}{(z-1)(z-0.22313)(z-0.13534)}$$

c.

$$G(s)=\frac{20}{(s+3)(s^2+6s+25)}=\frac{1.25}{s+3}-\frac{1.25s+3.57}{s^2+6s+25}=\frac{1.25}{s+3}-\frac{1.25(s+3)}{(s+3)^2+4^2}$$

$$G(z)=-1.25\frac{z}{z-e^{-aT}}-1.25\frac{z^2-zae^{-aT}\cos\omega T}{z^2-2ze^{-aT}\cos\omega T+e^{-2aT}}$$

For $a=3;\ \omega=4;\ T=0.5$,

$$G(z)=-1.25\frac{z}{z-0.2231}-1.25\frac{z^2+0.0929z}{z^2+0.1857z+0.0498}$$
$$=0.395\frac{z(z+0.2232)}{(z-0.2231)(z^2+0.1857z+0.0498)}$$

d.

$$G(s)=\frac{15}{s(s+1)(s^2+10s+81)}=\frac{0.1852}{s}-\frac{0.2083}{s+1}+0.02314\frac{s+0.9978}{s^2+10s+81}$$
$$=\frac{0.1852}{s}-\frac{0.2083}{s+1}+0.02314\frac{(s+5)-0.5348\sqrt{56}}{(s+5)^2+56}$$

$$G(z)=0.1852\frac{z}{z-1}-0.2083\frac{z}{z-e^{\beta T}}+0.02314\frac{z^2-zae^{-aT}\cos\omega T}{z^2-2ze^{-aT}\cos\omega T+e^{-2aT}}-0.0124\frac{ze^{-aT}\sin\omega T}{z^2-2ze^{-aT}\cos\omega T+e^{-2aT}}$$

For $a=5;\ \beta=1;\ \omega=\sqrt{56};\ T=0.5$,

$$G(z)=0.1852\frac{z}{z-1}-0.2083\frac{z}{z-0.6065}+0.02314\frac{z^2+0.0678z}{z^2+0.1355z+0.006738}+0.0005748\frac{z}{z^2+0.1355z+0.006738}$$

$$=\frac{0.00004z^4+0.05781z^3+0.02344z^2+0.001946z}{(z-1)(z-0.6065)\left(z^2+0.1355z+0.006738\right)}\approx\frac{0.05781z^3+0.02344z^2+0.001946z}{(z-1)(z-0.6065)\left(z^2+0.1355z+0.006738\right)}$$

$$=0.05781\frac{z^3+0.4055z^2+0.0337z}{(z-1)(z-0.6065)\left(z^2+0.1355z+0.006738\right)}=0.05781\frac{z(z+0.2888)(z+0.1167)}{(z-1)(z-0.6065)\left(z^2+0.1355z+0.006738\right)}$$

7.

Program:

```
'(a)'
syms s z n T                  %Construct symbolic objects for
                              %'s', 'z', 'n',and 'T'.
Gs=(s+4)/((s+2)*(s+5));       %Form G(s).
'G(s)'                        %Display label.
pretty(Gs)                    %Pretty print G(s).
%'g(t)'                       %Display label.
gt=ilaplace(Gs);              %Find g(t).
%pretty(gt)                   %Pretty print g(t).
gnT=compose(gt,n*T);          %Find g(nT).
%'g(kT)'                      %Display label.
%pretty(gnT)                  %Pretty print g(nT).
```

```
Gz=ztrans(gnT);                %Find G(z).
Gz=simplify(Gz);               %Simplify G(z).
%'G(z)'                        %Display label.
%pretty(Gz)                    %Pretty print G(z).
Gz=subs(Gz,T,0.5);             %Let T = 0.5 in G(z).
Gz=vpa(simplify(Gz),6);        %Simplify G(z) and evaluate numerical
                               %values to 6 places.
Gz=vpa(factor(Gz),6);          %Factor G(z).

'G(z) evaluated for T=0.5'     %Display label.
pretty(Gz)                     %Pretty print G(z) with numerical
                               %values.
'(b)'
Gs=(s+1)*(s+2)/(s*(s+3)*(s+4));
                               %Form G(s) = G(s).
'G(s)'                         %Display label.
pretty(Gs)                     %Pretty print G(s).
%'g(t)'                        %Display label.
gt=ilaplace(Gs);               %Find g(t).
%pretty(gt)                    %Pretty print g(t).
gnT=compose(gt,n*T);           %Find g(nT).
%'g(kT)'                       %Display label.
%pretty(gnT)                   %Pretty print g(nT).
Gz=ztrans(gnT);                %Find G(z).
Gz=simplify(Gz);               %Simplify G(z).
%'G(z)'                        %Display label.
%pretty(Gz)                    %Pretty print G(z).
Gz=subs(Gz,T,0.5);             %Let T = 0.5 in G(z).
Gz=vpa(simplify(Gz),6);        %Simplify G(z) and evaluate numerical
                               %values to 6 places.
Gz=vpa(factor(Gz),6);          %Factor G(z).
'G(z) evaluated for T=0.5'     %Display label.
pretty(Gz)                     %Pretty print G(z) with numerical
                               %values.
'(c)'
Gs=20/((s+3)*(s^2+6*s+25));    %Form G(s) = G(s).
'G(s)'                         %Display label.
pretty(Gs)                     %Pretty print G(s).
%'g(t)'                        %Display label.
gt=ilaplace(Gs);               %Find g(t).
%pretty(gt)                    %Pretty print g(t).
gnT=compose(gt,n*T);           %Find g(nT).
%'g(kT)'                       %Display label.
%pretty(gnT)                   %Pretty print g(nT).
Gz=ztrans(gnT);                %Find G(z).
Gz=simplify(Gz);               %Simplify G(z).
%'G(z)'                        %Display label.
%pretty(Gz)                    %Pretty print G(z).
Gz=subs(Gz,T,0.5);             %Let T = 0.5 in G(z).
Gz=vpa(simplify(Gz),6);        %Simplify G(z) and evaluate numerical
                               %values to 6 places.
Gz=vpa(factor(Gz),6);          %Factor G(z).
'G(z) evaluated for T=0.5'     %Display label.
pretty(Gz)                     %Pretty print G(z) with numerical
                               %values.
'(d)'
Gs=15/(s*(s+1)*(s^2+10*s+81));
                               %Form G(s) = G(s).
'G(s)'                         %Display label.
pretty(Gs)                     %Pretty print G(s).
%'g(t)'                        %Display label.
gt=ilaplace(Gs);               %Find g(t).
%pretty(gt)                    %Pretty print g(t).
gnT=compose(gt,n*T);           %Find g(nT).
%'g(kT)'                       %Display label.
%pretty(gnT)                   %Pretty print g(nT).
Gz=ztrans(gnT);                %Find G(z).
```

```
Gz=simplify(Gz);              %Simplify G(z).
%'G(z)'                       %Display label.
%pretty(Gz)                   %Pretty print G(z).
Gz=subs(Gz,T,0.5);            %Let T = 0.5 in G(z).
Gz=vpa(simplify(Gz),6);       %Simplify G(z) and evaluate numerical
                              %values to 6 places.
Gz=vpa(factor(Gz),6);         %Factor G(z).
'G(z) evaluated for T=0.5'    %Display label.
pretty(Gz)                    %Pretty print G(z) with numerical
                              %values.
```

Computer response:

```
ans =

(a)

ans =

G(s)

                                 s + 4
                            ---------------
                            (s + 2) (s + 5)
ans =

G(z) evaluated for T=0.5

                               z (z - .177350)
                      1.00000 ----------------------------
                              (z - .0820850) (z - .367880)
ans =

(b)

ans =

G(s)

                            (s + 1) (s + 2)
                           -----------------
                           s (s + 3) (s + 4)
ans =

G(z) evaluated for T=0.5

                          z (z - .296742) (z - .840812)
                  1.00000 ------------------------------------
                          (z - .135335) (z - .223130) (z - 1.)
ans =

ans =

(c)


ans =

G(s)


                                  20
                         -----------------------
                                   2
                         (s + 3) (s  + 6 s + 25)

ans =
```

G(z) evaluated for T=0.5

$$.394980\,\frac{(z + .223130)\,z}{(z - .223135)\,(z^2 + .185705\,z + .0497861)}$$

ans =

(d)

ans =

G(s)

$$\frac{15}{s\,(s + 1)\,(s^2 + 10\,s + 81)}$$

ans =

G(z) evaluated for T=0.5

$$.0578297\,\frac{(z + .289175)\,(z + .116364)\,z}{(z - .606535)\,(z - .999995)\,(z^2 + .135489\,z + .00673794)}$$

8.

a.

$$G2(s) = G(s)/s = \frac{20}{s^2\,(s + 5)} = \frac{4}{s^2} - 4/5\ 1/s + 4/5\,\frac{1}{s + 5}$$

Thus,

$$g2t = 4\ k\ T - 4/5 + 4/5\ \exp(-5\ k\ T)$$

Hence,

$$G(z) = (1 - 1/z)\left(4\,\frac{T\,z}{(z - 1)^2} - 4/5\,\frac{z}{z - 1} + 4/5\,\frac{z}{\exp(-5\ T)\left(\dfrac{z}{\exp(-5\ T)} - 1\right)}\right)$$

Letting $T = 0.3$,

$$G(z) = 4000\,\frac{6.482\,z + 3.964}{(z - 1.)\,(4.482\,z - 1.)}$$

b.

$$G2(s) = G(s)/s = \frac{20}{s^2\,(s+5)\,(s+3)} = 4/3\,\frac{1}{s^2} - \frac{32}{45}\,1/s + 10/9\,\frac{1}{s+3} - 2/5\,\frac{1}{s+5}$$

Thus,

$$g2t = 4/3\,k\,T - \frac{32}{45} + 10/9\,\exp(-3\,k\,T) - 2/5\,\exp(-5\,k\,T)$$

Hence,

$$G(z) = (1 - 1/z)\left(4/3\,\frac{T\,z}{(z-1)^2} - \frac{32}{45}\,\frac{z}{z-1} + 10/9\,\frac{z}{\exp(-3\,T)\left(\dfrac{z}{\exp(-3\,T)} - 1\right)} - 2/5\,\frac{z}{\exp(-5\,T)\left(\dfrac{z}{\exp(-5\,T)} - 1\right)}\right)$$

Letting $T = 0.3$,

$$G(z) = .04444\,\frac{12.82\,z^2 + 29.1\,z + 3.84}{(z - 1.)\,(2.460\,z - 1.)\,(4.482\,z - 1.)}$$

c.

$$G_e(z) = G_a(z)G(z)$$

where $G_a(z)$ is the answer to part (a) and $G(z)$, the pulse transfer function for $\frac{1}{s+3}$ in cascade with a zero-order-hold will now be found:

$$G2(s) = G(s)/s = \frac{1}{s\,(s+3)} = 1/s - 1/3\,\frac{1}{s+3}$$

Thus,

$$g2t = 1/3 - 1/3\,\exp(-3\,k\,T)$$

Hence,

```
                    /      z                        z              \
G(z) = (1 - 1/z) |1/3 ----- - 1/3 -------------------------|
                 |    z - 1              /    z         \|
                 |               exp(-3 T) |--------- - 1||
                 \                         \exp(-3 T)    //
```

Letting $T = 0.3$,

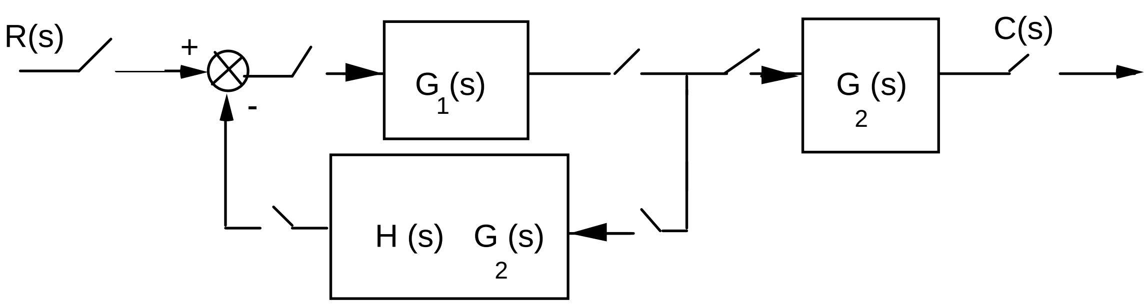
```
              .4866
G(z) =  ------------
         2.460 z - 1.
```

Thus,

$$G_e(z) = G_a(z)G(z) = 0.19464\frac{6.482z + 3.964}{(z-1)(4.482z-1)(2.46z-1)}$$

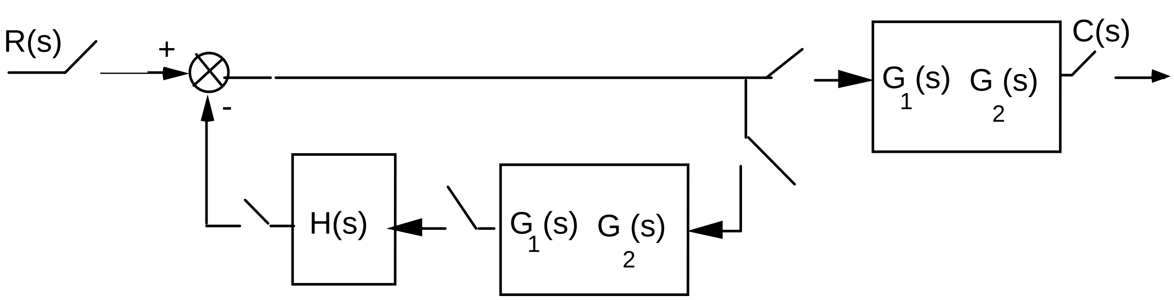
9.

a. Add phantom samplers at the input, output, and feedback path after H(s). Push G_2 (s) and its input sampler to the right past the pickoff point. Add a phantom sampler after G_1(s). Hence,



From this block diagram, $T(z) = \dfrac{G_1(z)G_2(z)}{1 + G_1(z)HG_2(z)}$.

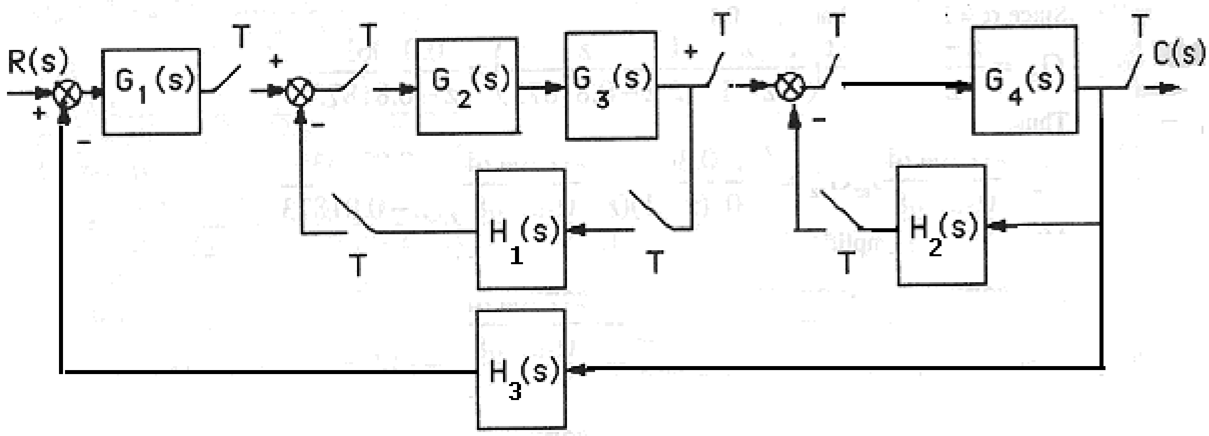
b. Add phantom samplers to the input, output, and the output of H(s). Push $G_1(s)G_2(s)$ and its input sampler to the right past the pickoff point. Add a phantom sampler at the output.



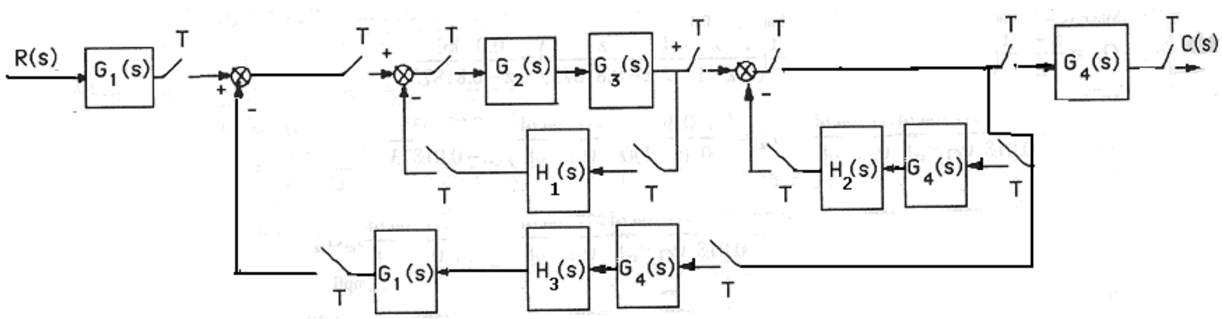
From this block diagram, $T(z) = \dfrac{G_1G_2(z)}{1 + G_1G_2(z)H(z)}$.

10.

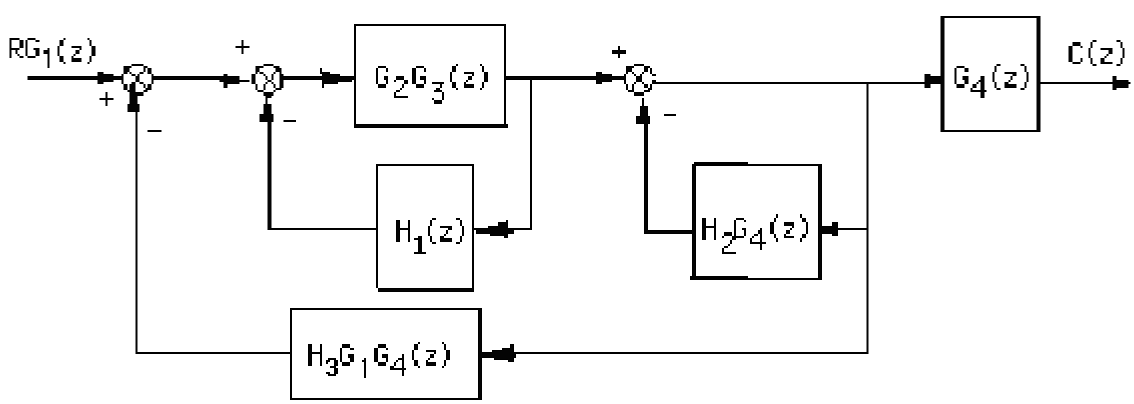
Add phantom samplers after $G_1(s)$, $G_3(s)$, $G_4(s)$, $H_1(s)$, and $H_2(s)$.



Push $G_1(s)$ and its sampler to the left past the summing junction. Also, push $G_4(s)$ and its input sampler to the right past the pickoff point. The resulting block diagram is,



Converting to z transforms,

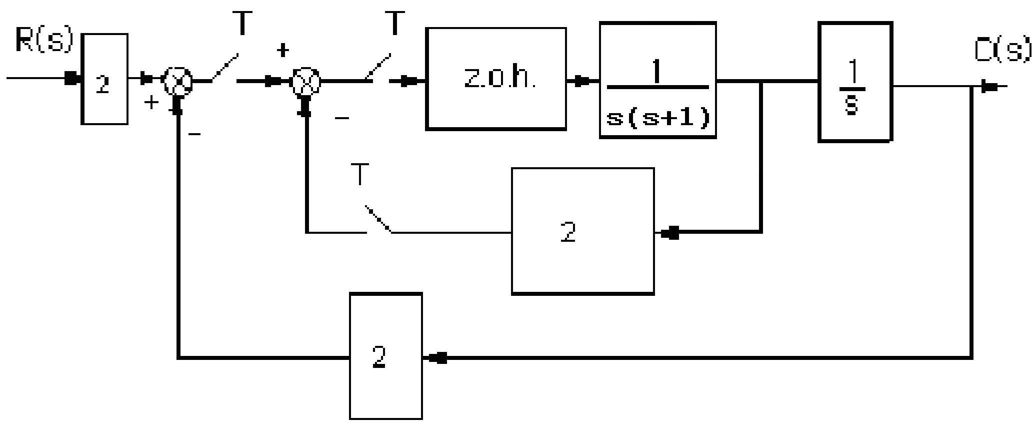


$$C(s)=RG_1(z)G_4(z)\left[\frac{\dfrac{G_2G_3(z)}{(1+G_2G_3(z)H_1(z))}*\dfrac{1}{(1+H_2G_4(z))}}{1+\dfrac{G_2G_3(z)}{(1+G_2G_3(z)H_1(z))}*\dfrac{1}{(1+H_2G_4(z))}H_3G_1G_4(z)}\right]$$

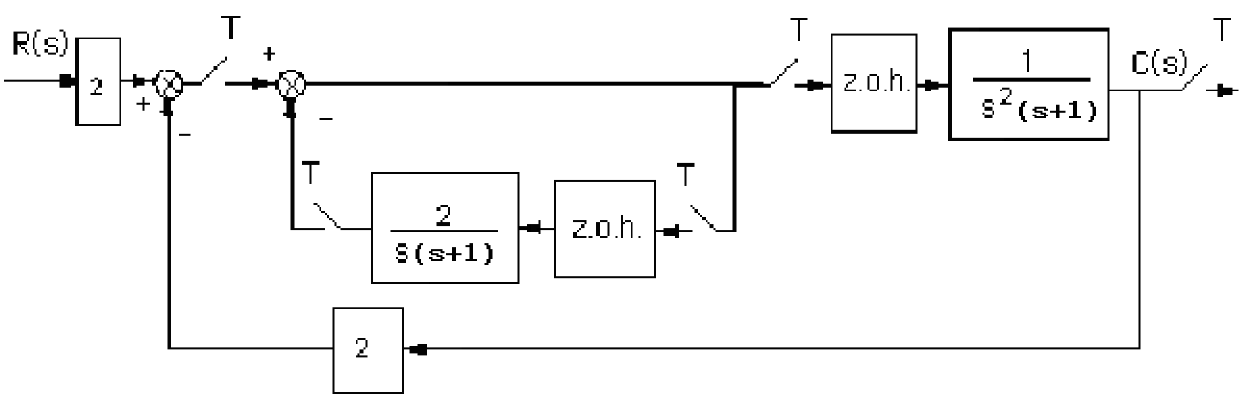
$$=\frac{RG_1(z)G_4(z)G_2G_3(z)}{(1+G_2G_3(z)H_1(z))(1+H_2G_4(z))+G_2G_3(z)H_3G_1G_4(z)}$$

11.

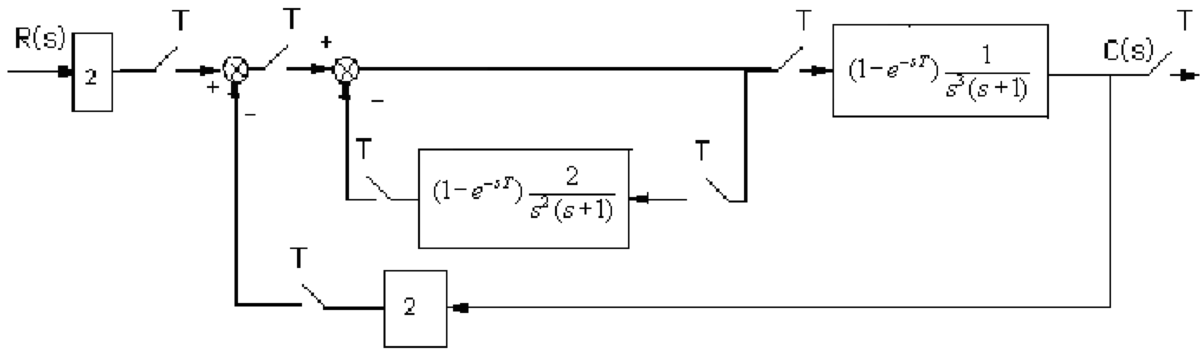
Push gain of 2 to the left past the summing junction and add phantom samplers as shown.



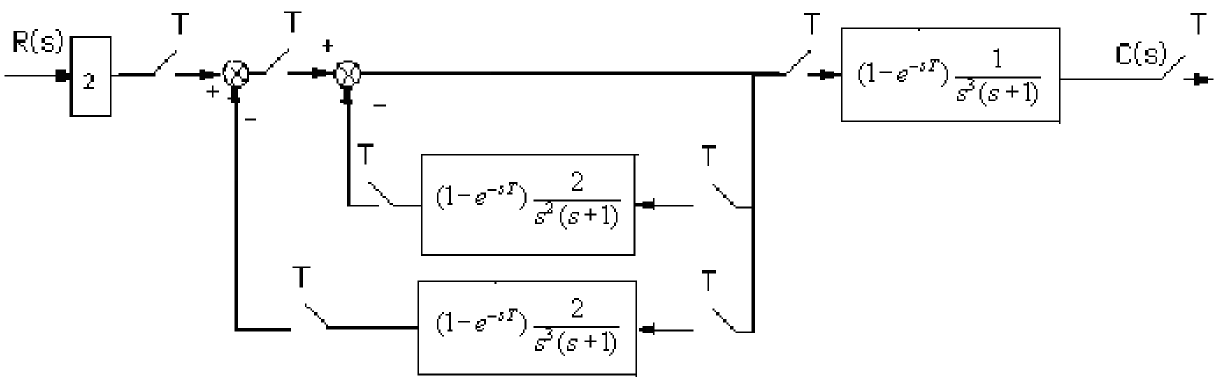
Push the z.o.h. and $\dfrac{1}{s(s+1)}$ to the right past the pickoff point. Also, add a phantom sampler at the output.



Add phantom samplers after the gain of 2 at the input and in the feedback. Also, represent the z.o.h. as Laplace transforms.



Push the last block to the right past the pickoff point and get,



Find the z transform for each transfer function.

$G_1(s) = 2$

transforms into

$G_1(z) = 2$.

$$H_1(s) = (1-e^{-sT})\frac{2}{s^2(s+1)} = (1-e^{-sT})\left[\frac{2}{s^2} - \frac{2}{s} + \frac{2}{s+1}\right]$$

transforms into

$$H_1(z) = \frac{z-1}{z}\left[2\frac{Tz}{(z-1)^2} - 2\frac{z}{z-1} + 2\frac{z}{z-e^{-T}}\right] = 2\frac{Tz - Te^{-T} + ze^{-T} - z - e^{-T} + 1}{(z-1)(z-e^{-T})}$$

$$H_2(s) = (1-e^{-sT})\frac{2}{s^3(s+1)} = (1-e^{-sT})\left[\frac{2}{s} - \frac{2}{s+1} - \frac{2}{s^2} + \frac{2}{s^3}\right]$$

transforms into

$$H_2(z) = \frac{z-1}{z}\left[\frac{2z}{z-1} - \frac{2z}{z-e^{-T}} - \frac{2Tz}{(z-1)^2} + \frac{T^2z(z+1)}{(z-1)^3}\right]$$

$$= \frac{(T^2 - 2e^{-T} + 2 - 2T)z^2 + (4e^{-T} - 4 + 2Te^{-T} + 2T + T^2 - T^2e^{-T})z + (2 - 2e^{-T} - 2Te^{-T} - T^2e^{-T})}{(z-1)^2(z-e^{-T})}$$

$$G_2(s) = (1 - e^{-sT})\frac{1}{s^3(s+1)}$$

transforms into

$$\frac{1}{2}H_2(z)$$

Thus, the closed-loop transfer function is

$$T(z) = G_1(z)G_2(z)\left[\frac{1}{1 + H_1(z) + H_2(z)}\right]$$

12.

$$G(z) = \frac{z - 1}{z}\ Z\left\{\frac{1}{s^2(s+1)}\right\}.$$

Using Eq. (13.49)

$$G(z) = \frac{T}{z - 1} - \frac{(1 - e^{-T})}{z - e^{-T}} = \frac{(T-1+e^{-T})z + (1-e^{-T}-Te^{-T})}{(z-1)(z-e^{-T})}$$

But,

$$T(z) = \frac{G(z)}{1 + G(z)} = \frac{(T-1+e^{-T})\,z + (1-e^{-T}-Te^{-T})}{z^2 + (T-2)\,z + (1-Te^{-T})}$$

The roots of the denominator are inside the unit circle for $0<T<3.923$.

13.

Program:

```
numg1=10*[1 7];
deng1=poly([-1 -3 -4 -5]);
G1=tf(numg1,deng1);
for T=5:-.01:0;
Gz=c2d(G1,T,'zoh');
Tz=feedback(Gz,1);
r=pole(Tz);
rm=max(abs(r));
if rm<=1;
break;
end;
end;
T
r
rm
```

Computer response:

```
T =

    3.3600


r =

   -0.9990
   -0.0461
   -0.0001
```

```
   -0.0000


rm =

    0.9990

>>
T =

    3.3600


r =

   -0.9990
   -0.0461
   -0.0001
   -0.0000


rm =

    0.9990
```

14.

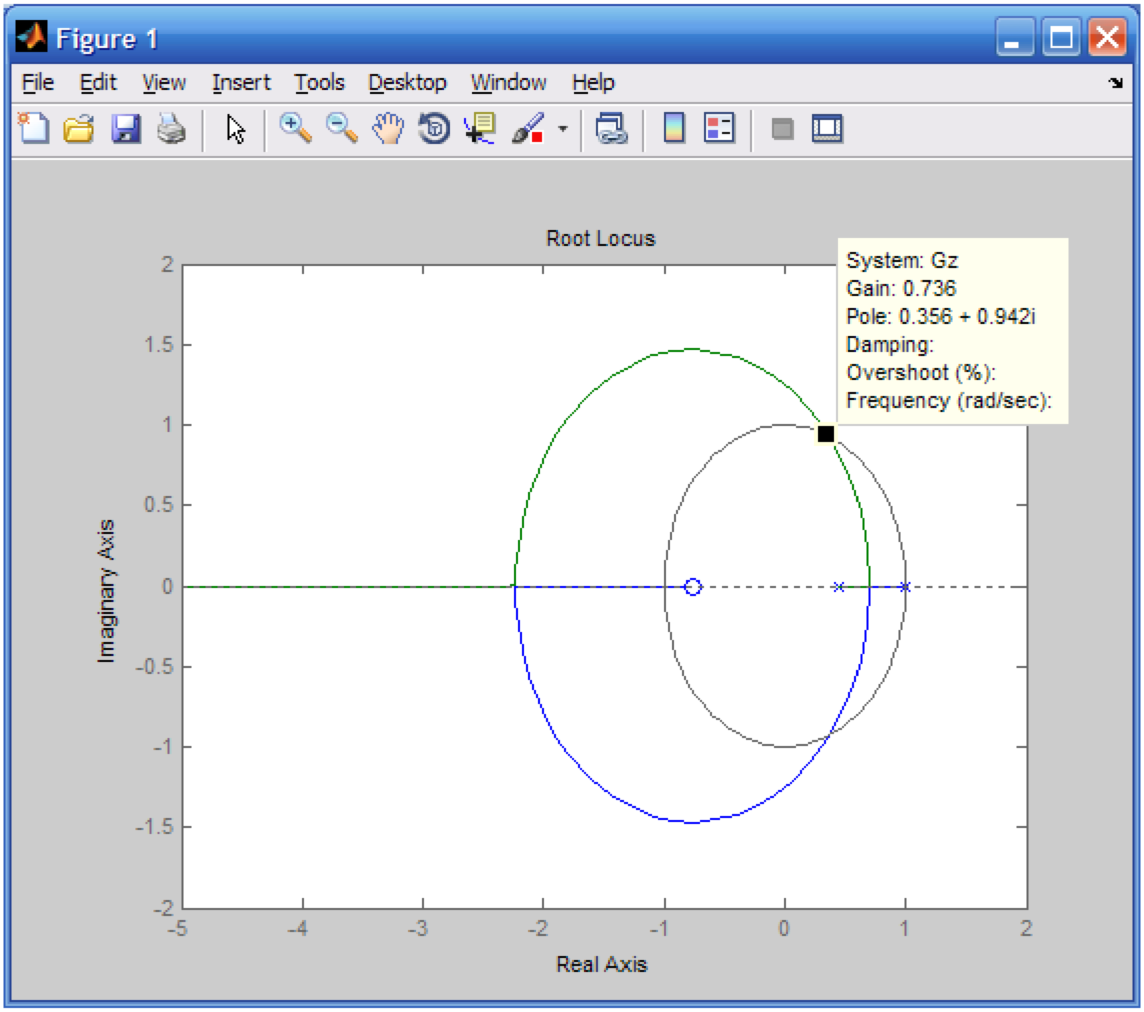
$$G(s) = K(1-e^{-sT})\frac{3}{s^2(s+4)}$$

$$G(s) = K(1-e^{-sT})\left[\frac{0.1875}{s+4} - \frac{0.1875}{s} + \frac{0.75}{s^2}\right]$$

$$G(z) = K\left[\frac{z-1}{z}\left\{0.1875\frac{z}{z-e^{-4(0.2)}} - 0.1875\frac{z}{z-1} + 0.75\frac{0.2z}{(z-1)^2}\right\}\right]$$

$$G(z) = K\frac{0.0467(z+0.767)}{(z-0.4493)(z-1)}$$

K = 0.736/0.0467 = 15.76



15.

a.

$$G_s = \left(1 - e^{-Ts}\right)\frac{1}{s(s+\alpha)}$$

$$G_s = \left(1 - e^{-Ts}\right)\left(-\frac{1}{\alpha[s+\alpha]} + \frac{1}{\alpha s}\right)$$

$$G_z = \frac{z-1}{z}\left(-\frac{1}{\alpha}\frac{z}{z - e^{-\alpha T}} + \frac{1}{\alpha}\frac{z}{z-1}\right)$$

$$G_z = \frac{-\frac{1}{e^{T\alpha}} + 1}{\alpha\left(z - \frac{1}{e^{T\alpha}}\right)}$$

$$\alpha = 2$$

$$T = 0.5$$

$$G_z = 0.31606\frac{1}{z - 0.36788}$$

First, check to see that the system is stable.

$$T_z = \frac{G_z}{1+G_z}$$
$$T_z = 0.31606\,\frac{1}{z-0.051819}$$

Since the closed-loop poles are inside the unit circle, the system is stable. Next, evaluate the static error constants and the steady-state error.

$$K_p = \lim_{z\to 1} G(z) = 0.5 \qquad e^*(\infty) = \frac{1}{1+K_p} = \frac{2}{3}$$

$$K_v = \frac{1}{T}\lim_{z\to 1}(z-1)G(z) = 0 \qquad e^*(\infty) = \frac{1}{K_v} = \infty$$

$$K_a = \frac{1}{T^2}\lim_{z\to 1}(z-1)^2 G(z) = 0 \qquad e^*(\infty) = \frac{1}{K_a} = \infty$$

b.

$$G_s = \left(1-e^{-Ts}\right)\frac{K\alpha}{s^2(s+\alpha)}$$

From Equation 13.48

$$G_z = K\,\frac{\alpha T\left(z-e^{-\alpha T}\right)-(z-1)\left(1-e^{-\alpha T}\right)}{\alpha(z-1)\left(z-e^{-\alpha T}\right)}$$

$K = 10$
$\alpha = 2$
$T = 0.1$

$$G_z = 5\,\frac{0.018731\,(z+0.93553)}{(z-1)(z-0.81873)}$$

First, test stability.

$$T_z = \frac{G_z}{1+G_z}$$

$$T_z = 0.093654\,\frac{z+0.93553}{(z-0.86254+0.40296i)(z-0.86254-0.40296i)}$$

The system is stable. The closed-loop poles are inside the unit circle. Now find the static error constants and the steady-state errors.

$$K_p = \lim_{z\to 1} G(z) = \infty \qquad e^*(\infty) = \frac{1}{1+K_p} = 0$$

$$K_v = \frac{1}{T}\lim_{z\to 1}(z-1)G(z) = 10 \qquad e^*(\infty) = \frac{1}{K_v} = 0.1$$

$$K_a = \frac{1}{T^2}\lim_{z\to 1}(z-1)^2 G(z) = 0 \qquad e^*(\infty) = \frac{1}{K_a} = \infty$$

c.

$$G_z = \frac{1.28}{z - 0.37}$$

First, test stability.

$$T_z = \frac{G_z}{1 + G_z}$$

$$T_z = 1.28 \frac{1}{z + 0.91}$$

The system is stable. The closed-loop pole is inside the unit circle. Now find the static error constants and the steady-state errors.

$$K_p = \lim_{z \to 1} G(z) = 2.03 \qquad e^*(\infty) = \frac{1}{1 + K_p} = 0.33$$

$$K_v = \frac{1}{T} \lim_{z \to 1} (z - 1) G(z) = 0 \qquad e^*(\infty) = \frac{1}{K_v} = \infty$$

$$K_a = \frac{1}{T^2} \lim_{z \to 1} (z - 1)^2 G(z) = 0 \qquad e^*(\infty) = \frac{1}{K_a} = \infty$$

d.

$$G_z = \frac{0.13\,(z + 1)}{(z - 1)\,(z - 0.74)}$$

First, test stability.

$$T_z = \frac{G_z}{1 + G_z}$$

$$T_z = 0.13 \frac{z + 1}{z^2 - 1.61\,z + 0.87}$$

$$T_z = 0.13 \frac{z + 1}{(z + [-0.805 + 0.47114i])\,(z + [-0.805 - 0.47114i])}$$

The system is stable. The closed-loop pole is inside the unit circle. Now find the static error constants and the steady-state errors.

$$K_p = \lim_{z \to 1} G(z) = \infty \qquad e^*(\infty) = \frac{1}{1 + K_p} = 0$$

$$K_v = \frac{1}{T} \lim_{z \to 1} (z - 1) G(z) = 10 \qquad e^*(\infty) = \frac{1}{K_v} = 0.1$$

$$K_a = \frac{1}{T^2} \lim_{z \to 1} (z - 1)^2 G(z) = 0 \qquad e^*(\infty) = \frac{1}{K_a} = \infty$$

16.

Program:

```
T=0.1;
numgz=[0.04406 -0.03624 -0.03284 0.02857];
dengz=[1 -3.394 +4.29 -2.393 +0.4966];
'G(z)'
Gz=tf(numgz,dengz,0.1)
'Zeros of G(z)'
zeros=roots(numgz)
'Poles of G(z)'
poles=roots(dengz)
%Check stability
Tz=feedback(Gz,1);
'Closed-Loop Poles'
r=pole(Tz)
M=abs(r)
pause
'Find Kp'
Gz=minreal(Gz,.00001);
Kp=dcgain(Gz)
'Find Kv'
factorkv=tf([1 -1],[1 0],0.1);    %Makes transfer function
                                  %proper and yields same Kv
Gzkv=factorkv*Gz;

Gzkv=minreal(Gzkv,.00001);        %Cancel common poles and
                                  %zeros
Kv=(1/T)*dcgain(Gzkv)
'Find Ka'
factorka=tf([1 -2 1],[1 0 0],0.1);%Makes transfer function
                                  %proper and yields same Ka
Gzka=factorka*Gz;

Gzka=minreal(Gzka,.00001);        %Cancel common poles and
                                  %zeros
Ka=(1/T)^2*dcgain(Gzka)
```

Computer response:

```
ans =

G(z)


Transfer function:
0.04406 z^3 - 0.03624 z^2 - 0.03284 z + 0.02857
-----------------------------------------------
 z^4 - 3.394 z^3 + 4.29 z^2 - 2.393 z + 0.4966

Sampling time: 0.1

ans =

Zeros of G(z)


zeros =

  -0.8753
   0.8489 + 0.1419i
   0.8489 - 0.1419i


ans =

Poles of G(z)


poles =
```

```
   1.0392
   0.8496 + 0.0839i
   0.8496 - 0.0839i
   0.6557


ans =

Closed-Loop Poles


r =

   0.9176 + 0.1699i
   0.9176 - 0.1699i
   0.7573 + 0.1716i
   0.7573 - 0.1716i


M =

    0.9332
    0.9332
    0.7765
    0.7765


ans =

Find Kp


Kp =

   -8.8750


ans =

Find Kv


Kv =

     0


ans =

Find Ka


Ka =

     0
```

17.

First find G(z)

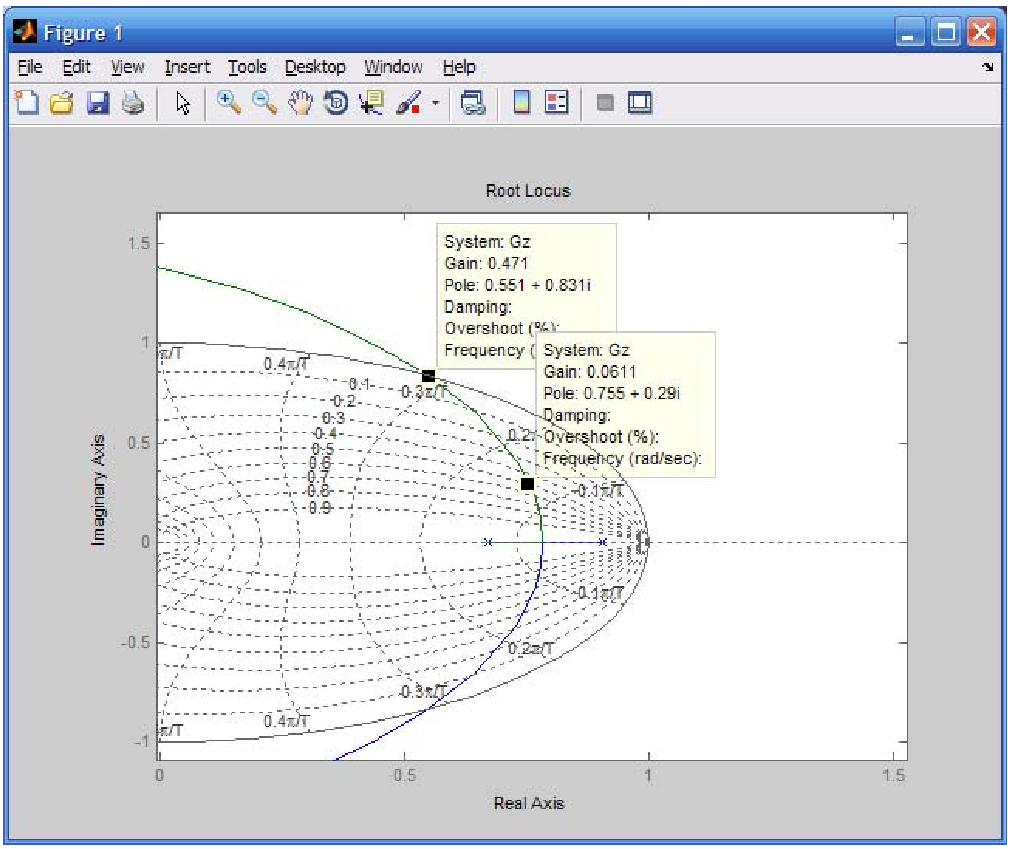
$$G(s) = K(1-e^{-sT})\frac{1}{s(s+1)(s+4)}$$

$$G(s) = K(1-e^{-sT})\left[\frac{1/4}{s} - \frac{1/3}{s+1} + \frac{0.0833}{s+4}\right]$$

$$G(z) = K\left[\frac{z-1}{z}\left\{\frac{1}{4}\frac{z}{z-1} - \frac{1}{3}\frac{z}{z-e^{-(0.1)}} - +0.0833\frac{z}{z-e^{-4(0.1)}}\right\}\right]$$

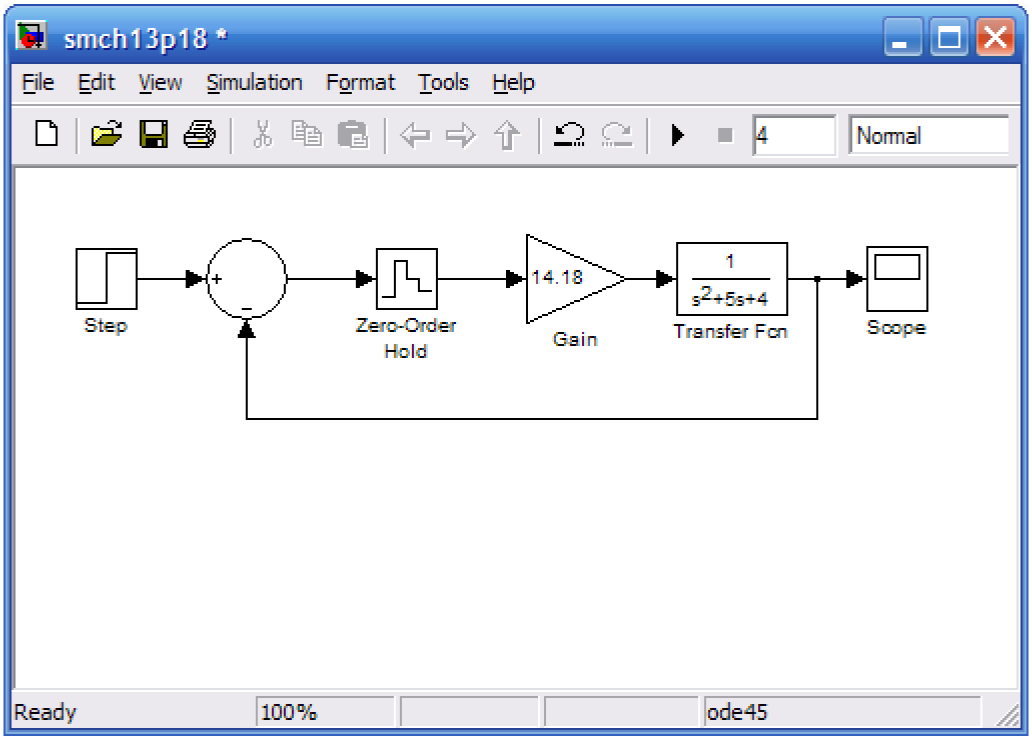
$$G(z) = K\frac{0.00431(z+0.8283)}{(z-0.90484)(z-0.6703)}$$

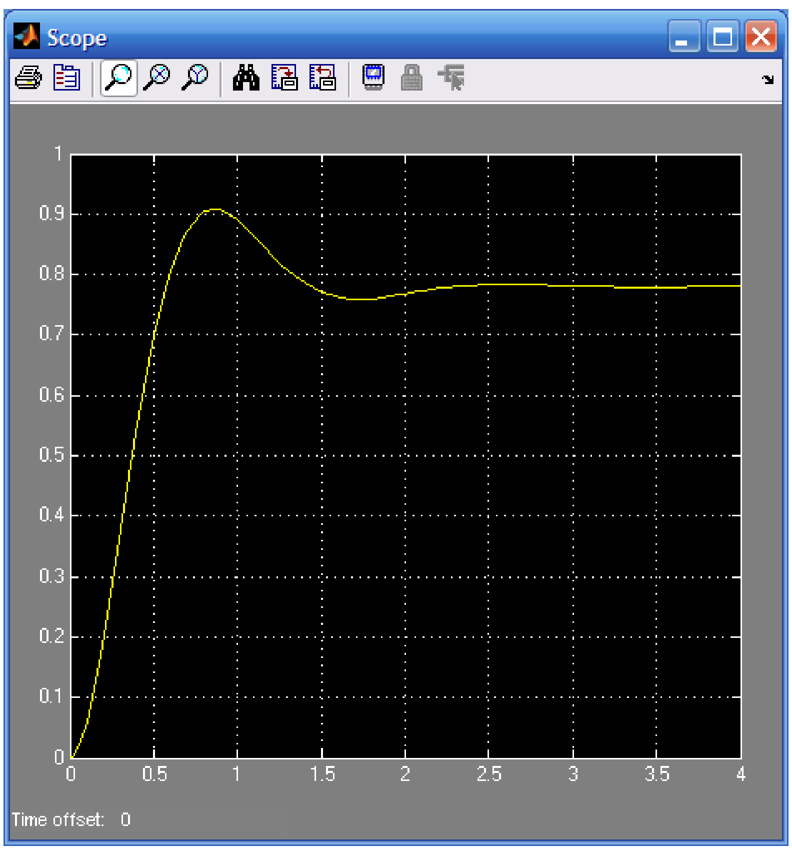
Next, plot the root locus.



Root locus intersects 0.5 damping ratio for 0.00431K = 0.0611. Thus, K = 14.18 for 16.3% overshoot. Root locus intersects the unit circle for 0.00431K = 0.47. Thus 0 < K < 109.28 for stability.

18.


smch13p18 *
File Edit View Simulation Format Tools Help
4
Normal
Step
Zero-Order Hold
14.18
Gain
1
s2+5s+4
Transfer Fcn
Scope
Ready
100%
ode45


Scope
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0 0.5 1 1.5 2 2.5 3 3.5 4
Time offset: 0

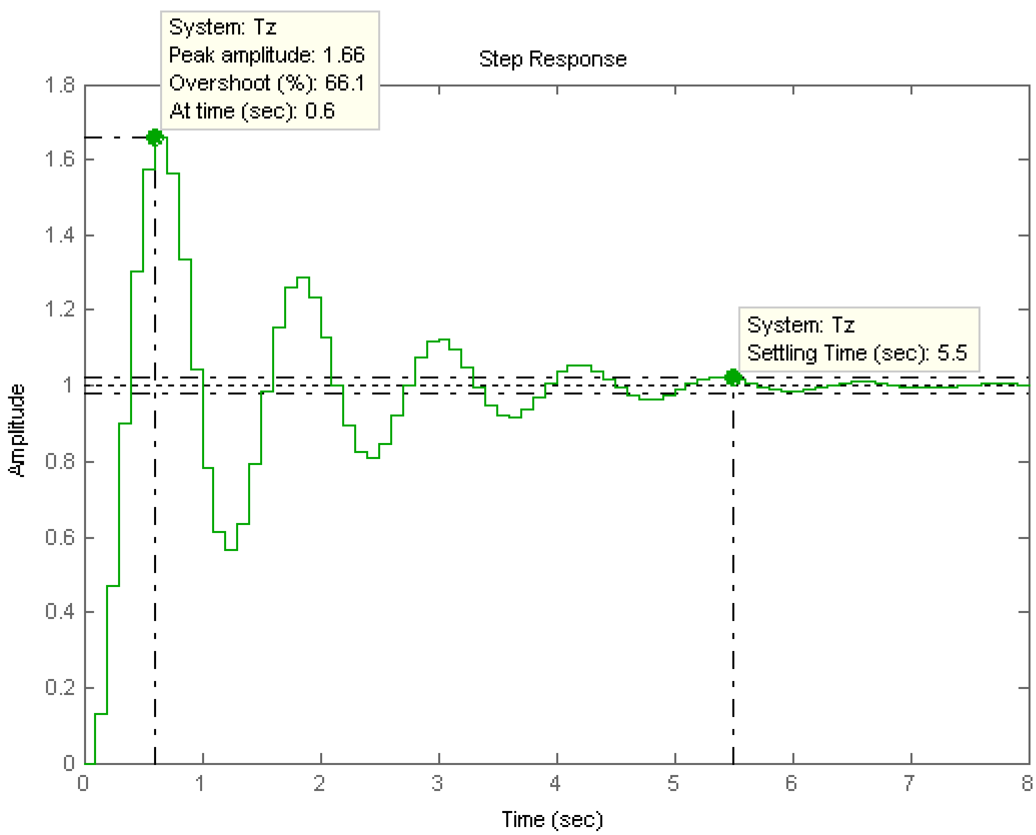
19.

Program:

```
numgz=0.13*[1 1];
dengz=poly([1 0.74]);
Gz=tf(numgz,dengz,0.1)
Gzpkz=zpk(Gz)
Tz=feedback(Gz,1)
Ltiview
```

Computer response:

```
Transfer function:
   0.13 z + 0.13
--------------------
z^2 - 1.74 z + 0.74

Sampling time: 0.1


Zero/pole/gain:
  0.13 (z+1)
--------------
(z-1) (z-0.74)

Sampling time: 0.1

Transfer function:
   0.13 z + 0.13
--------------------
z^2 - 1.61 z + 0.87

Sampling time: 0.1
```



20.

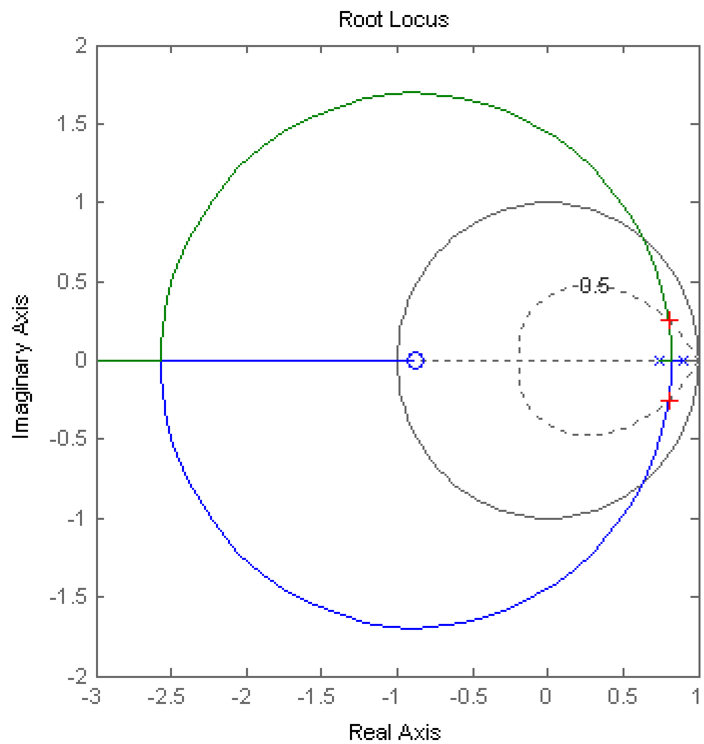
Program:

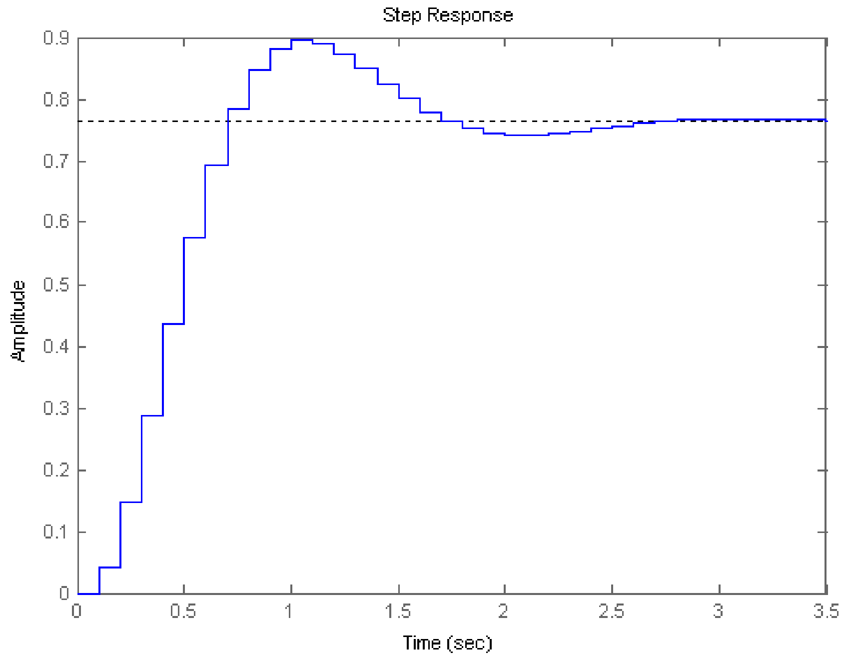
```
%Digitize G1(s) preceded by a sample and hold
numg1=1;
deng1=poly([-1  -3]);
'G1(s)'
G1s=tf(numg1,deng1)
'G(z)'
Gz=c2d(G1s,0.1,'zoh')
%Input transient response specifications
Po=input('Type %OS  ');
%Determine damping ratio
z=(-log(Po/100))/(sqrt(pi^2+log(Po/100)^2));
%Plot root locus
rlocus(Gz)
zgrid(z,0)
title(['Root Locus'])
[K,p]=rlocfind(Gz)  %Allows input by selecting point on graphic
pause
'T(z)'
Tz=feedback(K*Gz,1)
step(Tz)
```

Computer response:

```
ans =

G1(s)


Transfer function:
      1
-------------
s^2 + 4 s + 3


ans =

G(z)


Transfer function:
0.004384 z + 0.003837
---------------------
z^2 - 1.646 z + 0.6703

Sampling time: 0.1
Type %OS  16.3
Select a point in the graphics window

selected_point =

   0.8016 + 0.2553i


K =

    9.7200


p =

   0.8015 + 0.2553i
   0.8015 - 0.2553i


ans =
```

```
T(z)

Transfer function:
  0.04262 z + 0.0373
----------------------
z^2 - 1.603 z + 0.7076

Sampling time: 0.1
```





21.

Using the result from Problem 13.12

$$G_z = \frac{\left(T-1+e^{-T}\right)z+\left(1-e^{-T}-Te^{-T}\right)}{(z-1)\left(z-e^{-T}\right)}K$$

Letting T=0.1,

$$G_z = \frac{(0.0048374\,(z+0.96722))\,K}{(z-1)\,(z-0.90484)}$$

For Tp = 2 seconds,

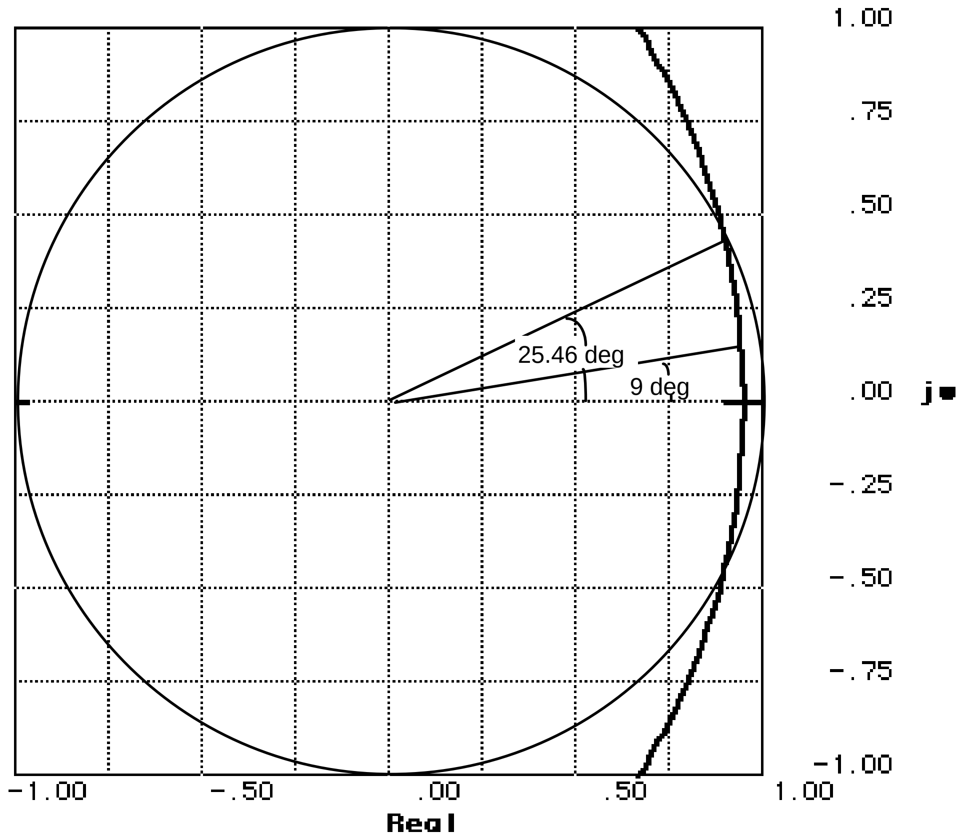
$$\frac{T_p}{T} = 20$$

Hence,

$$\frac{\pi}{\theta_1} = 20$$

Or,

$$\theta_1 = 9^0$$



The root locus intersects the Tp/T curve at 0.958<9 deg. with a gain of 0.0129. Hence, 4.837E-3 K = 0.0129, or K=2.67.
To determine stability, we see that the root locus intersects the 0 damping ratio curve at 1<25.4 deg. with a gain of 0.0983. Hence, 4.837E-3 K = 0.0983, or K=20.32.

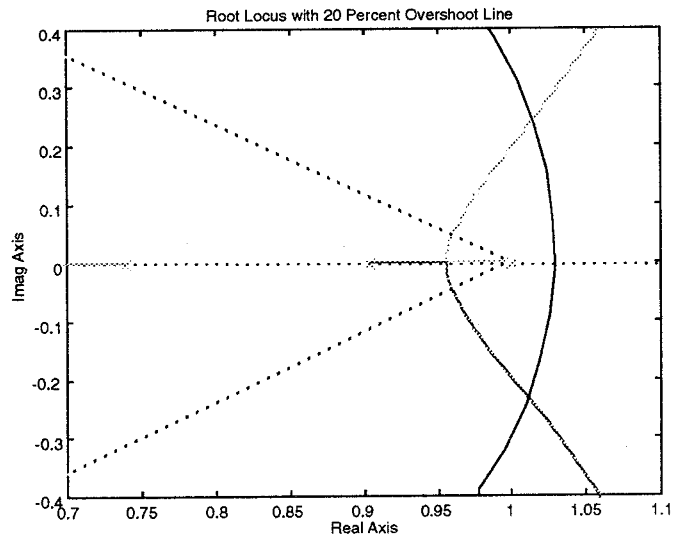
22.

First find G(z). $G(z) = K\frac{z-1}{z}\ z\{\frac{1}{s^2(s+1)(s+3)}\} = K\frac{z-1}{z}$

$z\{-\frac{1}{18}\frac{1}{s+3}+\frac{1}{2}\frac{1}{s+1}-\frac{4}{9}\frac{1}{s}+\frac{1}{3}\frac{1}{s^2}\}$

For T=0.1, $G(z) = K\frac{z-1}{z}\left(\frac{-\frac{1}{18}z}{z-0.74082}+\frac{\frac{1}{2}z}{z-0.90484}-\frac{\frac{4}{9}z}{z-1}+\frac{\frac{1}{30}z}{[z-1]^2}\right)$

$= 0.00015103K\frac{(z+0.24204)(z+3.3828)}{(z-1)(z-0.74082)(z-0.90484)}$. Plotting the root locus and overlaying the 20% overshoot curve, we select the point of intersection and calculate the gain: 0.00015103K = 1.789. Thus, K = 11845.33. Finding the intersection with the unit circle yields 0.00015103K = 9.85. Thus, 0 < K < 65218.83 for stability.

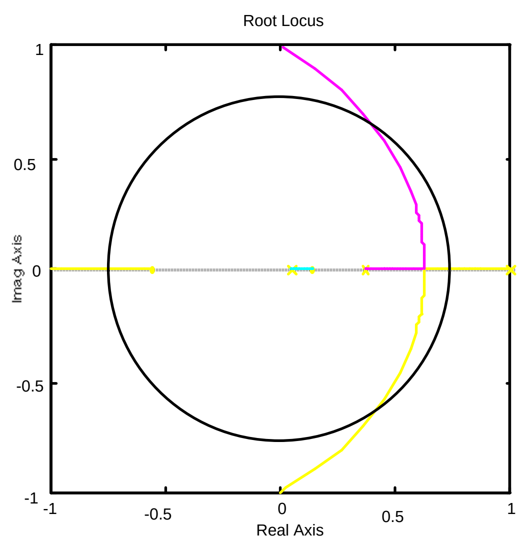


23.

First find G(z). $G(z) = K\frac{z-1}{z}\, z\{\frac{(s+2)}{s^2(s+1)(s+3)}\} = K\frac{z-1}{z}\, z\{\frac{1}{18}\frac{1}{s+3} + \frac{1}{2}\frac{1}{s+1} - \frac{5}{9}\frac{1}{s} + \frac{2}{3}\frac{1}{s^2}\} =$

For T=1, $G(z) = K\frac{z-1}{z}\left(\frac{\frac{1}{18}z}{z-0.049787} + \frac{\frac{1}{2}z}{z-0.36788} - \frac{\frac{5}{9}z}{z-1} + \frac{\frac{2}{3}z}{[z-1]^2}\right)$

$= 0.29782K\,\frac{(z-0.13774)(z+0.55935)}{(z-1)(z-0.049787)(z-0.36788)}$. Plotting the root locus and overlaying the $T_s = 15$ second circle, we select the point of intersection (0.4 + j0.63) and calculate the gain: 0.29782K = 1.6881. Thus, K = 5.668. Finding the intersection with the unit circle yields 0.29782K = 4.4. Thus, 0 < K < 14.77 for stability.



24.

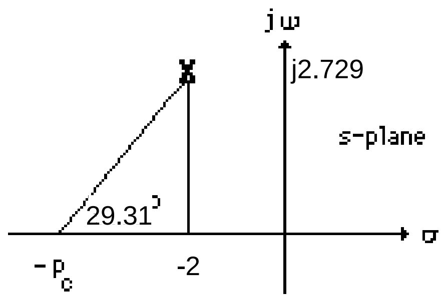
Substituting Eq. (13.88) into $G_c(s)$ and letting T = 0.01 yields

$$G_c(z)=\frac{1180z^2-1839z+661.1}{z^2-1}=1180\frac{(z-0.9959)(z-0.5625)}{(z+1)(z-1)}$$

25.

Since %OS = 10%, $\zeta=\frac{-\ln(\frac{\%OS}{100})}{\sqrt{\pi^2+\ln^2(\frac{\%OS}{100})}}=0.591$. Since $T_s=\frac{4}{\zeta\omega_n}=2$ seconds,

ω_n = 3.383 rad/s. Hence, the location of the closed-loop poles must be –2 ± j2.729. The summation of angles from open-loop poles to –2 ± j2.729 is –192.99°. Therefore, the design point is not on the root locus. A compensator whose angular contribution is 192.99° - 180° = 12.99° is required. Assume a compensator zero at -5 canceling the pole at -5. Adding the compensator zero at –5 to the plant's poles yields –150.69° at to –2 ± j2.729. Thus, the compensator's pole must contribute 180° – 150.69 = 29.31°. The following geometry results.



Thus,

$$\frac{2.729}{p_c-2}=\tan(29.31^0)$$

Hence, p_c = 6.86.Adding the compensator pole and zero to the system poles, the gain at the design point is found to be 124.3. Summarizing the results: $G_c(s)=\frac{124.3(s+5)}{(s+6.86)}$. Substituting Eq. (13.88) into $G_c(s)$ and letting T = 0.01 yields

$$G_c(z)=\frac{123.2z-117.2}{z-0.9337}=\frac{123.2(z-0.9512)}{(z-0.9337)}$$

26.

Program:

```
'Design of digital lead compensation'
clf                                    %Clear graph on screen.
'Uncompensated System'                 %Display label.
numg=1;                                %Generate numerator of G(s).
deng=poly([0 -5 -8]);                  %Generate denominator of G(s).
'G(s)'                                 %Display label.
G=tf(numg,deng)                        %Create and display G(s).
pos=input('Type desired percent overshoot ');
                                       %Input desired percent overshoot.
z=-log(pos/100)/sqrt(pi^2+[log(pos/100)]^2);
                                       %Calculate damping ratio.
rlocus(G)                              %Plot uncompensated root locus.
sgrid(z,0)                             %Overlay desired percent overshoot
                                       %line.
title(['Uncompensated Root Locus with ' , num2str(pos),...
```

```
'% Overshoot Line'])                %Title uncompensated root locus.
[K,p]=rlocfind(G);                  %Generate gain, K, and closed-loop
                                    %poles, p, for point selected
                                    %interactively on the root locus.
'Closed-loop poles = '              %Display label.
p                                   %Display closed-loop poles.
f=input('Give pole number that is operating point   ');
                                    %Choose uncompensated system
                                    %dominant pole.
'Summary of estimated specifications for selected point on'
'uncompensated root locus'          %Display label.
operatingpoint=p(f)                 %Display uncompensated dominant
                                    %pole.
gain=K                              %Display uncompensated gain.
estimated_settling_time=4/abs(real(p(f)))
                                    %Display uncompensated settling
                                    %time.
estimated_peak_time=pi/abs(imag(p(f)))
                                    %Display uncompensated peak time.
estimated_percent_overshoot=pos     %Display uncompensated percent
                                    %overshoot.
estimated_damping_ratio=z           %Display uncompensated damping
                                    %ratio.
estimated_natural_frequency=sqrt(real(p(f))^2+imag(p(f))^2)
                                    %Display uncompensated natural
                                    %frequency.
numkv=conv([1 0],numg);             %Set up numerator to evaluate Kv.
denkv=deng;                         %Set up denominator to evaluate Kv.
sG=tf(numkv,denkv);                 %Create sG(s).
sG=minreal(sG);                     %Cancel common poles and zeros.
Kv=dcgain(K*sG)                     %Display uncompensated Kv.
ess=1/Kv                            %Display uncompensated steady-state
                                    %error for unit ramp input.
'T(s)'                              %Display label.
T=feedback(K*G,1)                   %Create and display T(s).
step(T)                             %Plot step response of uncompensated
                                    %system.
title(['Uncompensated System with  ' ,num2str(pos),'% Overshoot'])
                                    %Add title to uncompensated step
                                    %response.
‘Press any key to go to lead compensation'
                                    %Display label.
pause
Ts=input('Type Desired Settling Time ');
                                    %Input desired settling time.
b=input('Type Lead Compensator Zero, (s+b). b=  ');
                                    %Input lead compensator zero.
done=1;                             %Set loop flag.
while done==1                       %Start loop for trying lead
                                    %compensator pole.
a=input('Enter a Test Lead Compensator Pole, (s+a). a =     ');
                                    %Enter test lead compensator pole.
numge=conv(numg,[1 b]);             %Generate numerator of Gc(s)G(s).
denge=conv([1 a],deng);             %Generate denominator of Gc(s)G(s).
Ge=tf(numge,denge);                 %Create Ge(s)=Gc(s)G(s).
wn=4/(Ts*z);                        %Evaluate desired natural frequency.
clf                                 %Clear graph on screen.
rlocus(Ge)                          %Plot compensated root locus with
                                    %test lead compensator pole.
axis([-10,10,-10,10])               %Change lead-compensated root locus
                                    %axes.
sgrid(z,wn)                         %Overlay grid on lead-compensated
                                    %root locus.
title(['Lead-Compensated Root Locus with ' , num2str(pos),...
'%  Overshoot Line, Lead Pole at  ', num2str(-a),...
' and Required Wn'])                %Add title to lead-compensated root
                                    %locus.
```

```
done=input('Are you done? (y=0,n=1)  ');
                                    %Set loop flag.
end                                 %End loop for trying compensator
                                    %pole.
[K,p]=rlocfind(Ge);                 %Generate gain, K, and closed-loop
                                    %poles, p, for point selected
                                    %interactively on the root locus.
'Gc(s)'                             %Display label.
Gc=tf([1 b],[1 a])                  %Display lead compensator.
'Gc(s)G(s)'                         %Display label.
Ge                                  %Display Gc(s)G(s).
'Closed-loop poles = '              %Display label.
p                                   %Display lead-compensated system's
                                    %closed-loop poles.
f=input('Give pole number that is operating point   ');
                                    %Choose lead-compensated system
                                    %dominant pole.
'Summary of estimated specifications for selected point on lead'
'compensated root locus'            %Display label.
operatingpoint=p(f)                 %Display lead-compensated dominant
                                    %pole.
gain=K                              %Display lead-compensated gain.
estimated_settling_time=4/abs(real(p(f)))
                                    %Display lead-compensated settling
                                    %time.
estimated_peak_time=pi/abs(imag(p(f)))
                                    %Display lead-compensated peak time.
estimated_percent_overshoot=pos     %Display lead-compensated percent
                                    %overshoot.
estimated_damping_ratio=z           %Display lead-compensated damping
                                    %ratio.
estimated_natural_frequency=sqrt(real(p(f))^2+imag(p(f))^2)
                                    %Display lead-compensated natural
                                    %frequency.
s=tf([1 0],1);                      %Create transfer function, "s".
sGe=s*Ge;                           %Create sGe(s) to evaluate Kv.
sGe=minreal(sGe);                   %Cancel common poles and zeros.
Kv=dcgain(K*sGe)                    %Display lead-compensated Kv.
ess=1/Kv                            %Display lead-compensated steady-
                                    %state error for unit ramp input.
'T(s)'                              %Display label.
T=feedback(K*Ge,1)                  %Create and display lead-compensated
                                    %T(s).
'Press any key to continue and obtain the lead-compensated step'
'response'                          %Display label
pause
step(T)                             %Plot step response for lead
                                    %compensated system.
title(['Lead-Compensated System with  ' ,num2str(pos),'% Overshoot'])
                                    %Add title to step response of PD
                                    %compensated system.
pause
'Digital design'                    %Print label.
T=0.01                              %Define sampling interval.
clf                                 %Clear graph.
'Gc(s) in polynomial form'          %Print label.
Gcs=K*Gc                            %Create Gc(s) in polynomial form.
'Gc(s) in polynomial form'          %Print label.
Gcszpk=zpk(Gcs)                     %Create Gc(s) in factored form.
'Gc(z) in polynomial form via Tustin Transformation'
                                    %Print label.
Gcz=c2d(Gcs,T,'tustin')             %Form Gc(z) via Tustin transformation.
'Gc(z) in factored form via Tustin Transformation'
                                    %Print label.
Gczzpk=zpk(Gcz)                     %Show Gc(z) in factored form.
'Gp(s) in polynomial form'          %Print label.
Gps=G                               %Create Gp(s) in polynomial form.
```

```
'Gp(s) in factored form'            %Print label.
Gpszpk=zpk(Gps)                     %Create Gp(s) in factored form.
'Gp(z) in polynomial form'          %Print label.
Gpz=c2d(Gps,T,'zoh')                %Form Gp(z) via zoh transformation.
'Gp(z) in factored form'            %Print label.
Gpzzpk=zpk(Gpz)                     %Form Gp(z) in faactored form.
pole(Gpz)                           %Find poles of Gp(z).
Gez=Gcz*Gpz;                        %Form Ge(z) = Gc(z)Gp(z).
'Ge(z) = Gc(z)Gp(z) in factored form'
                                    %Print label.
Gezzpk=zpk(Gez)                     %Form Ge(z) in factored form.
'z-1'                               %Print label.
zm1=tf([1 -1],1,T)                  %Form z-1.
zm1Gez=minreal(zm1*Gez,.00001);     %Cancel common factors.
'(z-1)Ge(z)'                        %Print label.
zm1Gezzpk=zpk(zm1Gez)               %Form & display (z-1)Ge(z) in
                                    %factored form.
pole(zm1Gez)                        %Find poles of (z-1)Ge(z).
Kv=10*dcgain(zm1Gez)                %Find Kv.
Tz=feedback(Gez,1)                  %Find closed-loop
                                    %transfer function, T(z)
step(Tz)                            %Find step reponse.
title('Closed-Loop Digital Step Response')
                                    %Add title to step response.
```

Computer response:

```
ans =

Design of digital lead compensation


ans =

Uncompensated System


ans =

G(s)


Transfer function:
         1
-------------------
s^3 + 13 s^2 + 40 s

Type desired percent overshoot 10
Select a point in the graphics window

selected_point =

  -1.6435 + 2.2437i


ans =

Closed-loop poles =


p =

  -9.6740
  -1.6630 + 2.2492i
  -1.6630 - 2.2492i

Give pole number that is operating point   2

ans =
```

```
Summary of estimated specifications for selected point on

ans =

uncompensated root locus

operatingpoint =

  -1.6630 + 2.2492i

gain =

   75.6925

estimated_settling_time =

    2.4053

estimated_peak_time =

    1.3968

estimated_percent_overshoot =

    10

estimated_damping_ratio =

    0.5912

estimated_natural_frequency =

    2.7972

Kv =

    1.8923

ess =

    0.5285

ans =

T(s)

Transfer function:
           75.69
---------------------------
s^3 + 13 s^2 + 40 s + 75.69

ans =
```

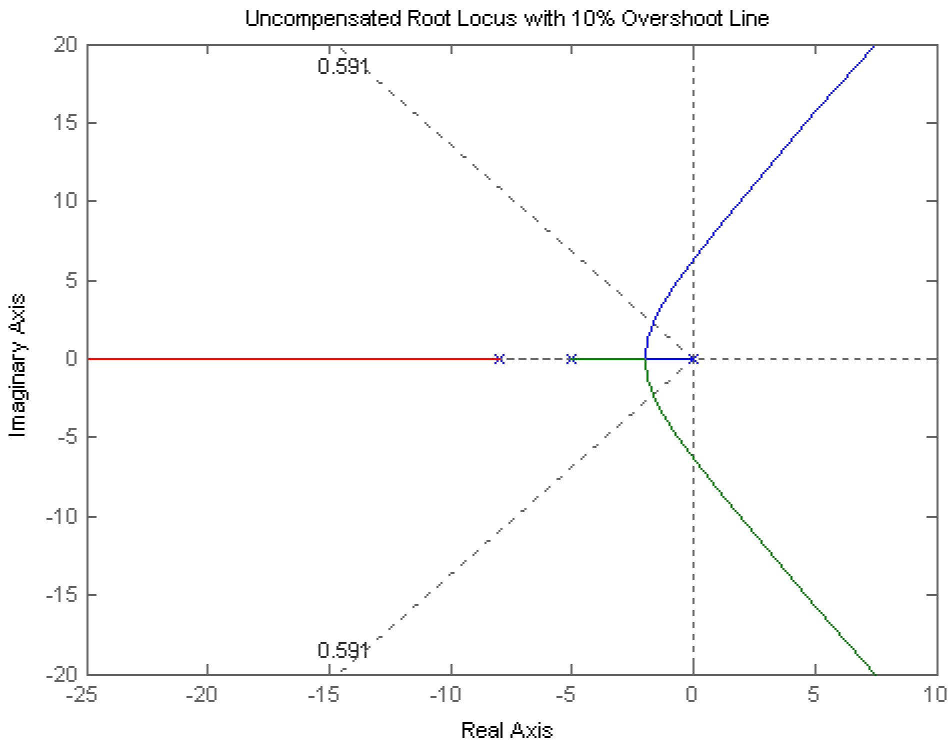
```
Press any key to go to lead compensation

Type Desired Settling Time 2
Type Lead Compensator Zero, (s+b). b=  5
Enter a Test Lead Compensator Pole, (s+a). a =     6.8
Are you done? (y=0,n=1)  0
Select a point in the graphics window

selected_point =

  -1.9709 + 2.6692i


ans =

Gc(s)


Transfer function:
 s + 5
-------
s + 6.8


ans =

Gc(s)G(s)


Transfer function:
              s + 5
----------------------------------
s^4 + 19.8 s^3 + 128.4 s^2 + 272 s


ans =

Closed-loop poles =


p =

 -10.7971
  -5.0000
  -2.0015 + 2.6785i
  -2.0015 - 2.6785i

Give pole number that is operating point   3

ans =

Summary of estimated specifications for selected point on lead


ans =

compensated root locus


operatingpoint =

  -2.0015 + 2.6785i


gain =

  120.7142
```

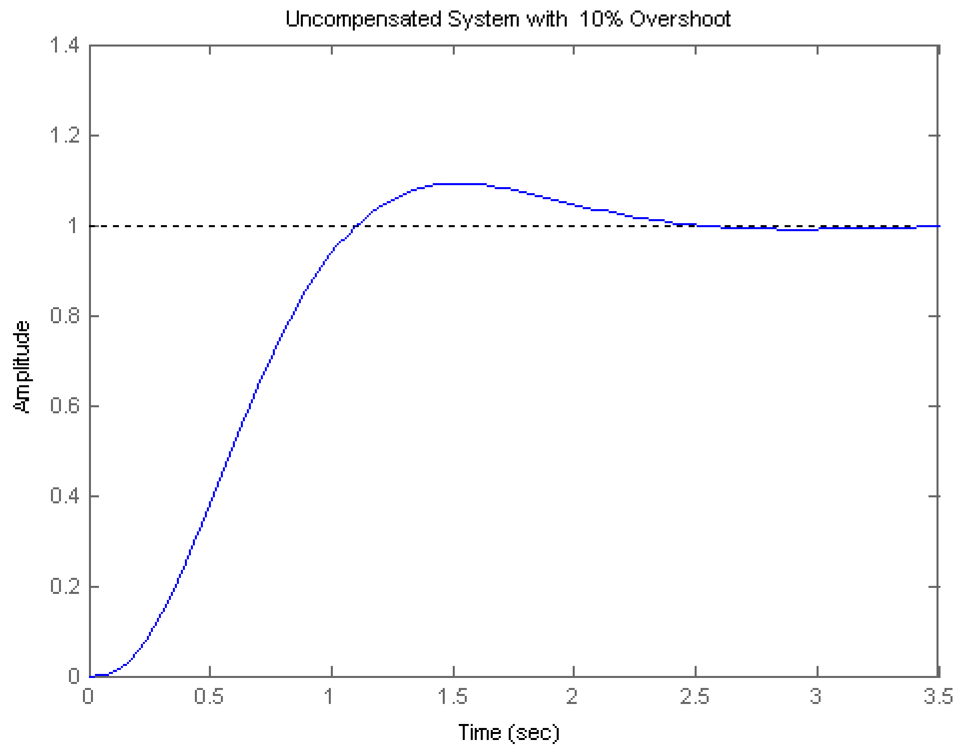
```
estimated_settling_time =

    1.9985

estimated_peak_time =

    1.1729

estimated_percent_overshoot =

    10

estimated_damping_ratio =

    0.5912

estimated_natural_frequency =

    3.3437

Kv =

    2.2190

ess =

    0.4507

ans =

T(s)

Transfer function:
              120.7 s + 603.6
--------------------------------------------
s^4 + 19.8 s^3 + 128.4 s^2 + 392.7 s + 603.6

ans =

Press any key to continue and obtain the lead-compensated step

ans =

response

ans =

Digital design

T =

    0.0100
```

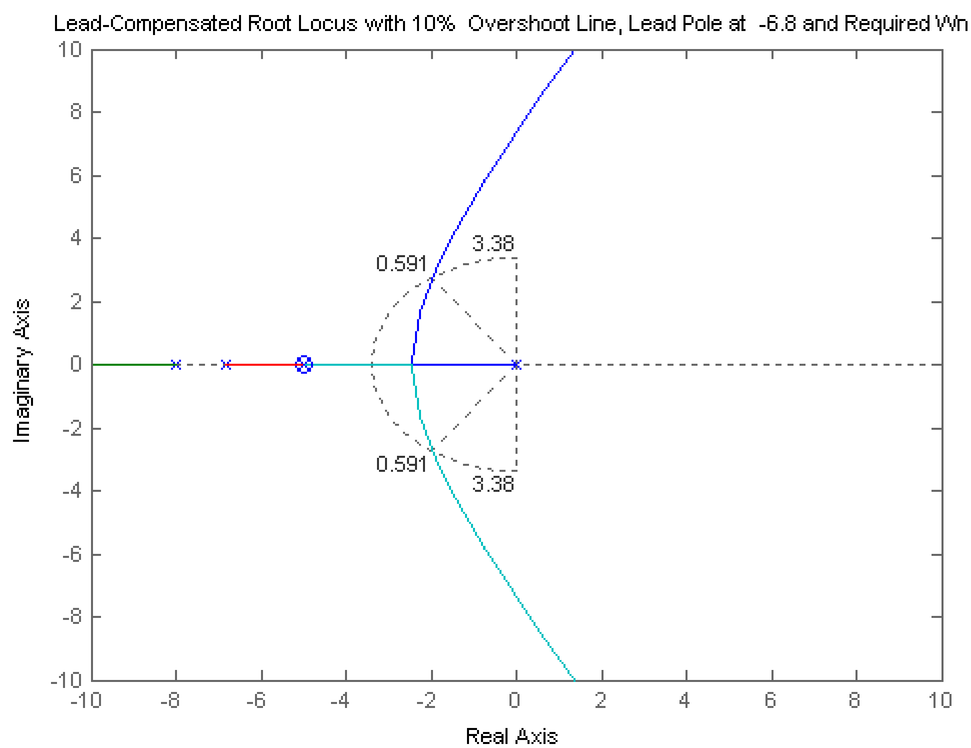
```
ans =

Gc(s) in polynomial form


Transfer function:
120.7 s + 603.6
---------------
    s + 6.8


ans =

Gc(s) in polynomial form


Zero/pole/gain:
120.7142 (s+5)
--------------
   (s+6.8)


ans =

Gc(z) in polynomial form via Tustin Transformation


Transfer function:
119.7 z - 113.8
---------------
  z - 0.9342

Sampling time: 0.01

ans =

Gc(z) in factored form via Tustin Transformation


Zero/pole/gain:
119.6635 (z-0.9512)
-------------------
    (z-0.9342)

Sampling time: 0.01

ans =

Gp(s) in polynomial form


Transfer function:
         1
-------------------
s^3 + 13 s^2 + 40 s


ans =

Gp(s) in factored form


Zero/pole/gain:
      1
-------------
s (s+8) (s+5)
```

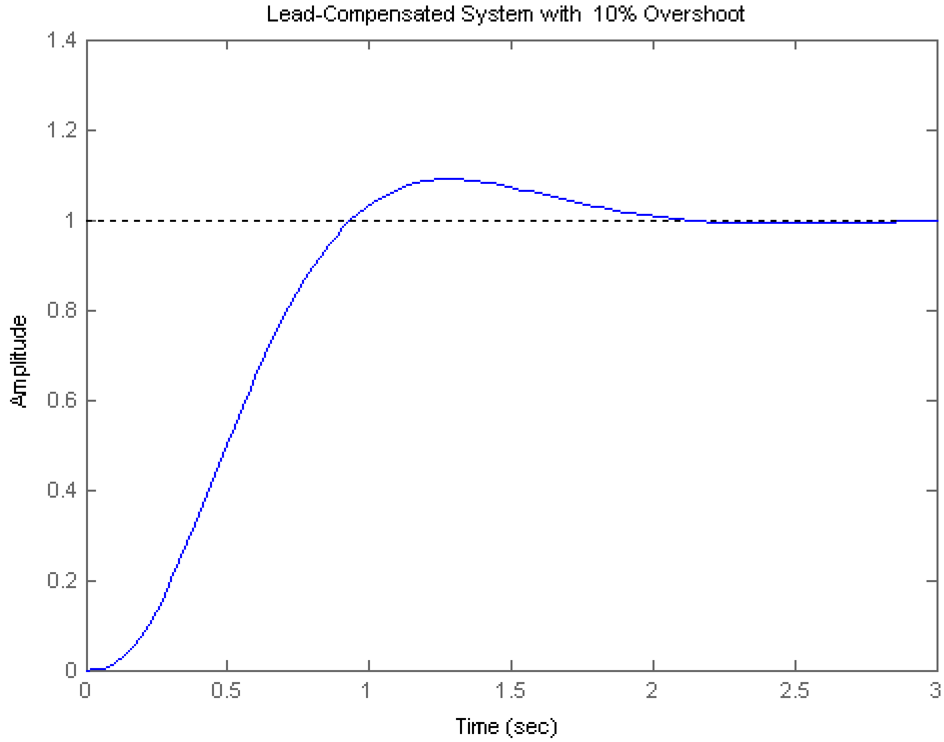
```
ans =

Gp(z) in polynomial form


Transfer function:
1.614e-007 z^2 + 6.249e-007 z + 1.512e-007
------------------------------------------
    z^3 - 2.874 z^2 + 2.752 z - 0.8781

Sampling time: 0.01

ans =

Gp(z) in factored form


Zero/pole/gain:
1.6136e-007 (z+3.613) (z+0.2593)
--------------------------------
  (z-1) (z-0.9512) (z-0.9231)

Sampling time: 0.01

ans =

    1.0000
    0.9512
    0.9231


ans =

Ge(z) = Gc(z)Gp(z) in factored form


Zero/pole/gain:
1.9308e-005 (z+3.613) (z-0.9512) (z+0.2593)
-------------------------------------------
  (z-1) (z-0.9512) (z-0.9342) (z-0.9231)

Sampling time: 0.01

ans =

z-1


Transfer function:
z - 1

Sampling time: 0.01

ans =

(z-1)Ge(z)


Zero/pole/gain:
1.9308e-005 (z+3.613) (z+0.2593)
--------------------------------
     (z-0.9342) (z-0.9231)

Sampling time: 0.01
```

```
ans =

    0.9342
    0.9231


Kv =

    0.2219


Transfer function:
1.931e-005 z^3 + 5.641e-005 z^2 - 5.303e-005 z - 1.721e-005
-----------------------------------------------------------
       z^4 - 3.809 z^3 + 5.438 z^2 - 3.45 z + 0.8203

Sampling time: 0.01
```

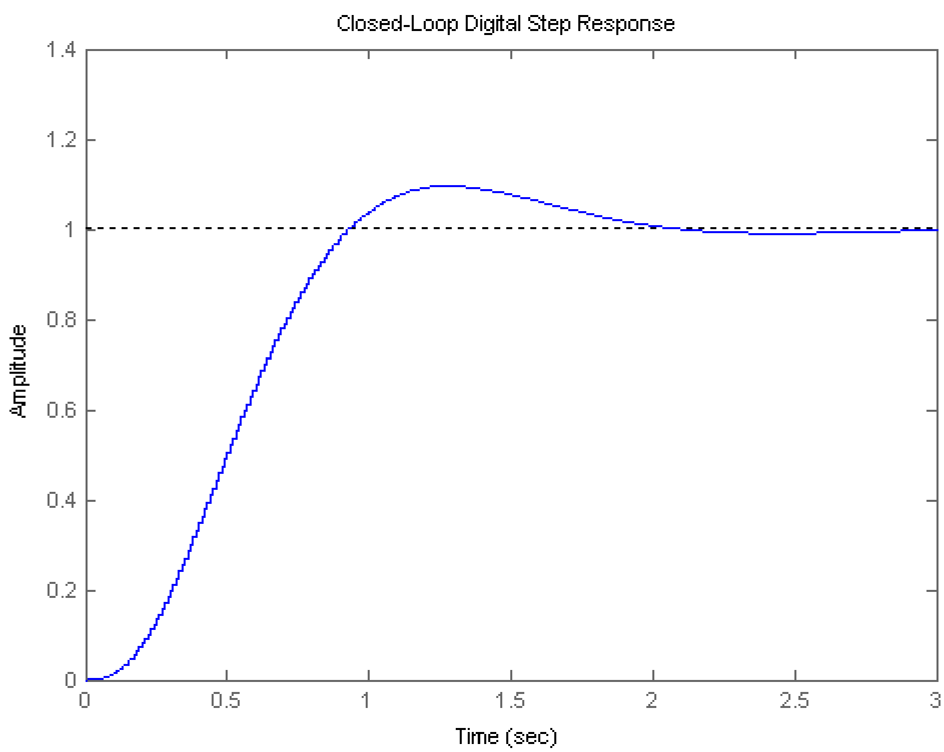



Uncompensated System with 10% Overshoot
Amplitude
Time (sec)
0
0.2
0.4
0.6
0.8
1
1.2
1.4
0
0.5
1
1.5
2
2.5
3
3.5


Lead-Compensated Root Locus with 10% Overshoot Line, Lead Pole at -6.8 and Required Wn
Imaginary Axis
Real Axis
0.591
3.38
0.591
3.38
10
8
6
4
2
0
-2
-4
-6
-8
-10
-10
-8
-6
-4
-2
0
2
4
6
8
10



Lead-Compensated System with 10% Overshoot
Amplitude
Time (sec)
1.4
1.2
1
0.8
0.6
0.4
0.2
0
0.5
1.5
2
2.5
3

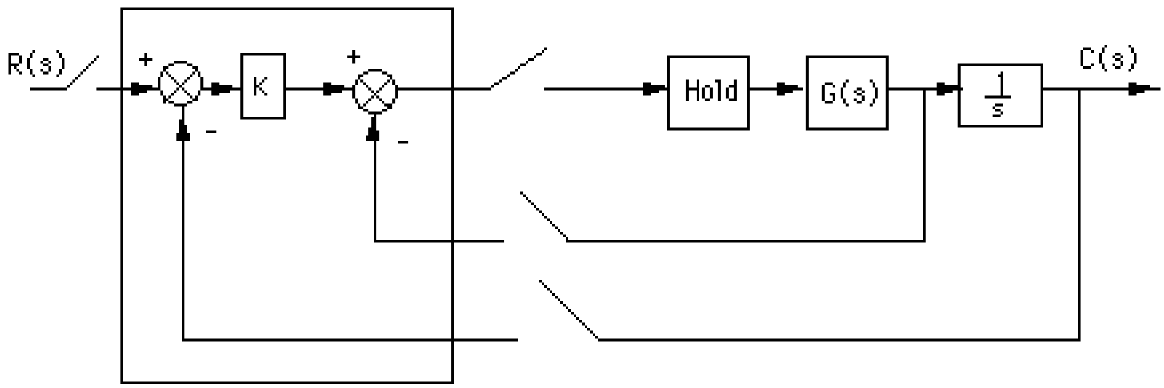


Closed-Loop Digital Step Response
Amplitude
Time (sec)
1.4
1.2
1
0.8
0.6
0.4
0.2
0
0.5
1.5
2
2.5
3

SOLUTIONS TO DESIGN PROBLEMS

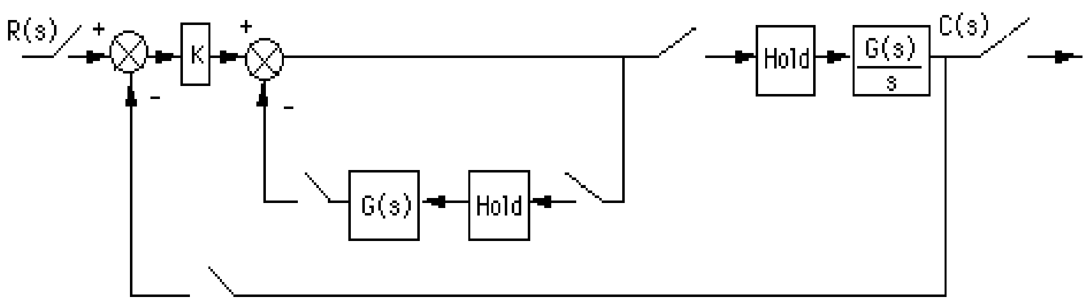
27.

a. Push negative sign from vehicle dynamics to the left past the summing junction. The computer will be the area inside the large box with the inputs and outputs shown sampled. G(s) is the combined rudder actuator and vehicle dynamics. Also, the yaw rate sensor is shown equivalently before the integrator with unity feedback.

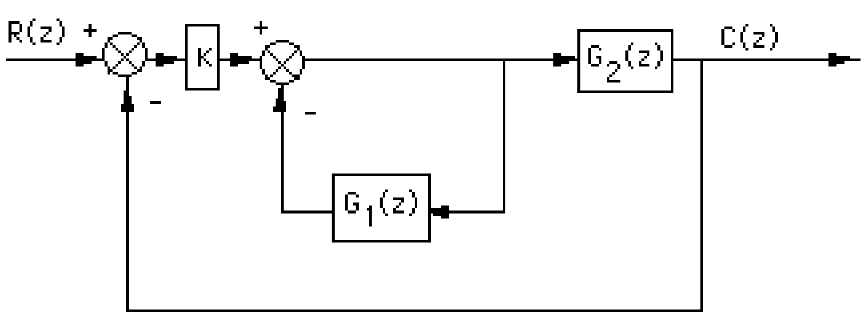


where $G(s) = \dfrac{0.25(s+0.437)}{(s+2)(s+1.29)(s+0.193)}$.

b. Add a phantom sampler at the output and push G(s) along with its sample and hold to the right past the pickoff point.



Move the outer-loop sampler to the output of $\dfrac{G(s)}{s}$ and write the z transforms of the transfer functions.



where

$$G_1(s) = (1 - e^{-Ts})\ \frac{0.25(s+0.437)}{s(s+2)(s+1.29)(s+0.193)}$$

and

$$G_2(s) = (1 - e^{-Ts})\ \frac{0.25(s+0.437)}{s^2(s+2)(s+1.29)(s+0.193)}\ .$$

Now find the z transforms of $G_1(s)$ and $G_2(s)$. For $G_1(z)$.

Since

$$\frac{0.25\,(s+0.437)}{s\,(s+2)\,(s+1.29)\,(s+0.193)} = 0.15228\,\frac{1}{s+2} - 0.15944\,\frac{1}{s+0.193} - 0.21224\,\frac{1}{s+1.29} + 0.2194\,\frac{1}{s}$$

$$G_1(z) = \frac{z-1}{z}\left(0.15228\,\frac{z}{z-e^{-2T}} - 0.15944\,\frac{z}{z-e^{-0.193\,T}} - 0.21224\,\frac{z}{z-e^{-1.29\,T}} + 0.2194\,\frac{z}{z-1}\right)$$

$T = 0.1$

$$G_1(z) = \frac{0.0011305\,z^2 - 6.0812\times10^{-5}\,z - 0.00097764}{(z-0.81873)\,(z-0.87897)\,(z-0.98089)}$$

For $G_2(z)$:

Since

$$\frac{0.25\,(s+0.437)}{s^2\,(s+2)\,(s+1.29)\,(s+0.193)} = -0.076142\,\frac{1}{s+2} + 0.82613\,\frac{1}{s+0.193} + 0.16453\,\frac{1}{s+1.29} - 0.91452\,\frac{1}{s} + 0.2194\,\frac{1}{s^2}$$

$$G_2(z) = \frac{z-1}{z}\left(-0.076142\,\frac{z}{z-e^{-2T}} + 0.82613\,\frac{z}{z-e^{-0.193\,T}} + 0.16453\,\frac{z}{z-e^{-1.29\,T}} - 0.91452\,\frac{z}{z-1} + 0.2194\,\frac{Tz}{[z-1]^2}\right)$$

$T = 0.1$

$$G_2(z) = \frac{3.8642\times10^{-5}\,z^3 + 0.00010636\,z^2 - 0.00010404\,z - 3.1765\times10^{-5}}{(z-1)\,(z-0.81873)\,(z-0.87897)\,(z-0.98089)}$$

Now find the closed-loop transfer function. First find the equivalent forward transfer function.

$$G_e(z) = K\,\frac{G_2(z)}{1+G_1(z)}$$

$$G_e = 3.8642\times10^{-5}\,\frac{(z+0.24807)\,(z-0.95724)\,(z+3.4616)\,K}{(z-1)\,(z-0.75327)\,(z^2-1.9253\,z+0.93574)}$$

Thus,

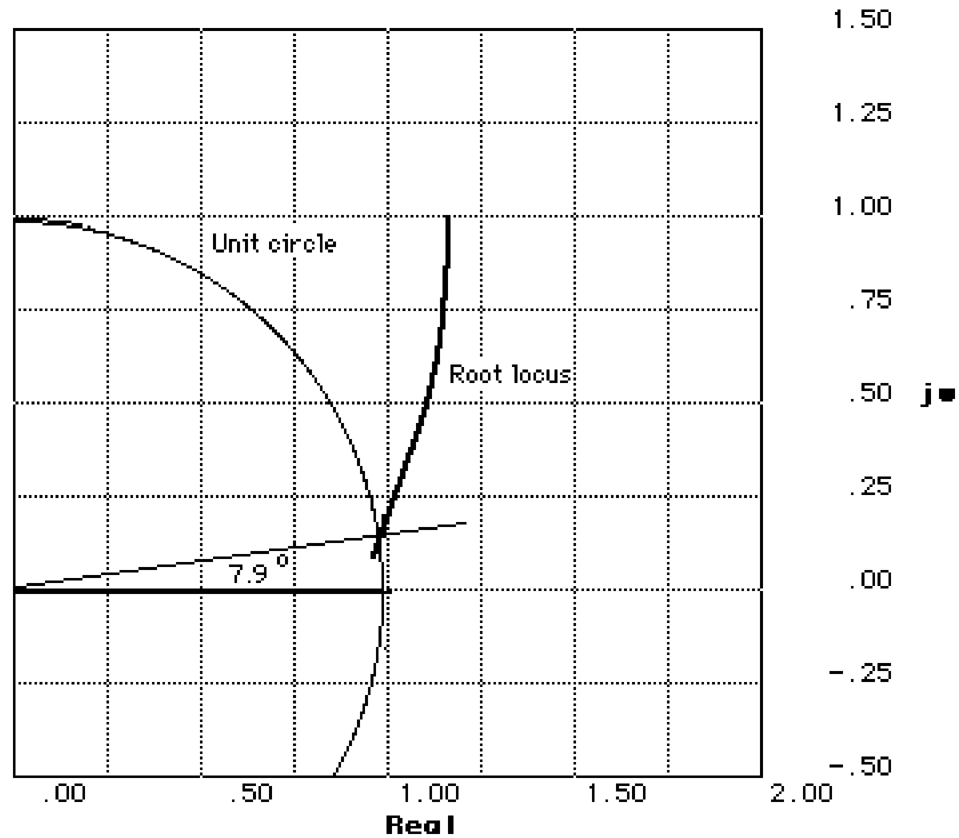
$$T(z) = \frac{G_e(z)}{1+G_e(z)}$$

Substituting values,

$$T = 3.8642\times10^{-5}\,\frac{(z+0.24807)\,(z-0.95724)\,(z+3.4616)\,K}{z^4 + (3.8642\times10^{-5}\,K - 3.6786)\,z^3 + (0.00010636\,K + 5.0646)\,z^2 - (0.00010404\,K + 3.0909)\,z + (-3.1765\times10^{-5}\,K + 0.70487)}$$

c. Using $G_e(z)$, plot the root locus and see where it crosses the unit circle.

$$G_e = 3.8642\times10^{-5}\frac{(z+0.24807)(z-0.95724)(z+3.4616)K}{(z-1)([z-0.75327][z-0.96266+0.095008i][z-0.96266-0.095008i])}$$



The root locus crosses the unit circle when $3.8642\times10^{-5}K = 5.797\times10^{-4}$, or K = 15.

28.

a. First find G(z).

$$G(z) = K\frac{z-1}{z}\ z\{\frac{1}{s^2(s^2+7s+1220)}\}$$

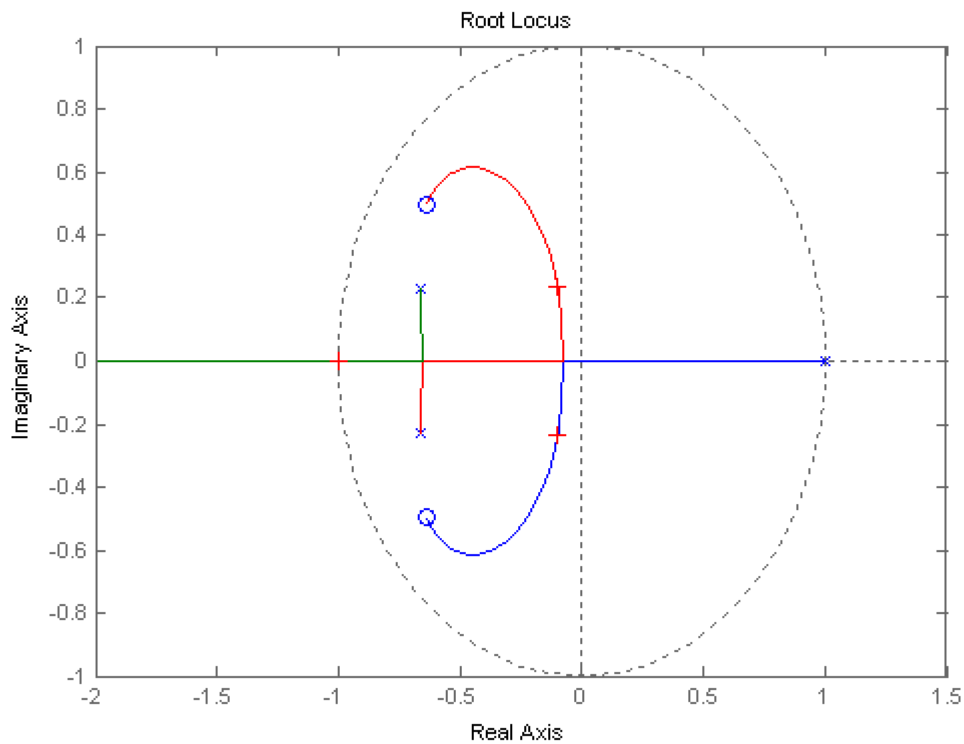
$$= K\frac{z-1}{z}\ z\{6.7186\times10^{-7}(\frac{7(s+3.5)-34.4\sqrt{1207.8}}{(s+3.5)^2+1207.8}-7\frac{1}{s}+1220\frac{1}{s^2})\}$$

For T = 0.1,

$$= K\frac{z-1}{z}\ \{6.7186\times10^{7}(7\frac{z^2+0.66582z}{z^2+1.3316z+0.49659}+7.8472\frac{z}{z^2+1.3316z+0.49659}-7\frac{z}{z-1}+122\frac{z}{(z-1)^2})\}$$

$$G(z) = K\ 7.9405\times10^{-5}\frac{(z+0.63582+0.49355i)(z+0.63582-0.49355i)}{(z-1)([z+0.66582+0.2308i][z+0.66582-0.2308i])}$$

b.



c. The root locus intersects the unit circle at -1 with a gain, $7.9405 \times 10^{-5} K = 10866$, or

$0 < K < 136.84 \times 10^{6}$.

d.

Program:

```
%Digitize G1(s) preceded by a sample and hold
numg1=1;
deng1=[1 7 1220 0];
'G1(s)'
G1s=tf(numg1,deng1)
'G(z)'
Gz=c2d(G1s,0.1,'zoh')
[numgz,dengz]=tfdata(Gz,'v');
'Zeros of G(z)'
roots(numgz)
'Poles of G(z)'
roots(dengz)
%Plot root locus
rlocus(Gz)
title(['Root Locus'])
[K,p]=rlocfind(Gz)
```

Computer response:

```
ans =

G1(s)


Transfer function:
         1
--------------------
s^3 + 7 s^2 + 1220 s
```

```
ans =

G(z)


Transfer function:
7.947e-005 z^2 + 0.0001008 z + 5.15e-005
----------------------------------------
  z^3 + 0.3316 z^2 - 0.8351 z - 0.4966

Sampling time: 0.1

ans =

Zeros of G(z)


ans =

  -0.6345 + 0.4955i
  -0.6345 - 0.4955i


ans =

Poles of G(z)


ans =

   1.0000
  -0.6658 + 0.2308i
  -0.6658 - 0.2308i

Select a point in the graphics window

selected_point =

   -0.9977


K =

  1.0885e+004


p =

  -0.9977
  -0.0995 + 0.2330i
  -0.0995 - 0.2330i
```

See part (b) for root locus plot.

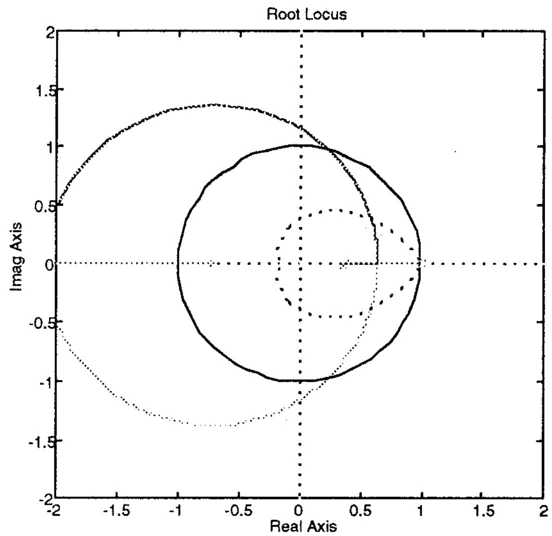
29.

a. First find G(z). $G(z) = K\frac{z-1}{z}\ z\{\frac{20000}{s^2(s+100)}\} = K\frac{z-1}{z}\ z\{2\frac{1}{s+100} - 2\frac{1}{s} + 200\frac{1}{s^2}\}$

For T = 0.01, $G(z) = K\frac{z-1}{z}\left(-2\frac{z}{z-1} + 2\frac{z}{[z-1]^2} + 2\frac{z}{z-0.36788}\right)$

$= 0.73576K\frac{z+0.71828}{(z-1)(z-0.36788)}$.

b. Plotting the root locus. Finding the intersection with the unit circle yields 0.73576K = 1.178. Thus, 0 < K < 1.601 for stability.



c. Using the root locus, we find the intersection with the 15% overshoot curve (ζ = 0.517) at 0.5955 + j0.3747 with 0.73576K = 0.24. Thus K = 0.326.

d.

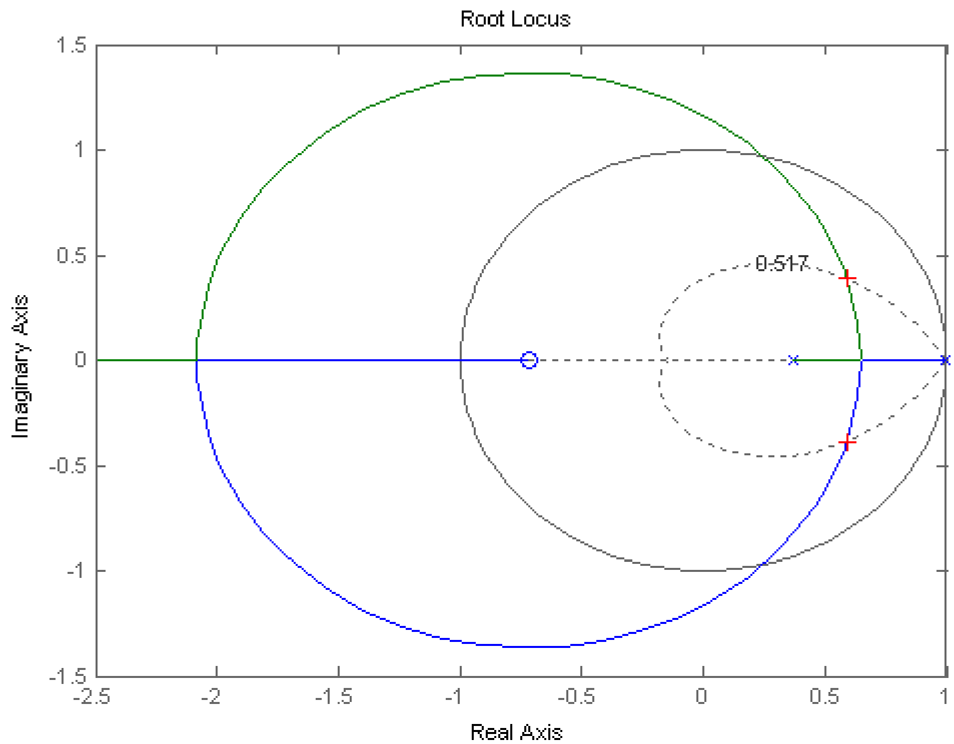
Program:

```
%Digitize G1(s) preceded by a sample and hold
numg1=20000;
deng1=[1 100 0];
'G1(s)'
G1s=tf(numg1,deng1)
'G(z)'
Gz=c2d(G1s,0.01,'zoh')
[numgz,dengz]=tfdata(Gz,'v');
'Zeros of G(z)'
roots(numgz)
'Poles of G(z)'
roots(dengz)
%Input transient response specifications
Po=input('Type %OS  ');
%Determine damping ratio
z=(-log(Po/100))/(sqrt(pi^2+log(Po/100)^2))
%Plot root locus
rlocus(Gz)
zgrid(z,0)
title(['Root Locus'])
[K,p]=rlocfind(Gz)  %Allows input by selecting point on graphic.
```

Computer response:

```
ans =

G1(s)


Transfer function:
   20000
-----------
s^2 + 100 s
```

```
ans =

G(z)


Transfer function:
  0.7358 z + 0.5285
----------------------
z^2 - 1.368 z + 0.3679

Sampling time: 0.01

ans =

Zeros of G(z)


ans =

   -0.7183


ans =

Poles of G(z)


ans =

    1.0000
    0.3679

Type %OS  15

z =

    0.5169

Select a point in the graphics window

selected_point =

   0.5949 + 0.3888i


K =

    0.2509


p =

   0.5917 + 0.3878i
   0.5917 - 0.3878
```



30.

The open loop transmission with the sample and hold is $L(s) = \frac{1-e^{-sT}}{s}\frac{20000}{s}$, so

$$L(z) = \frac{z-1}{z}\left[\frac{20000Tz}{(z-1)^2}\right] = \frac{20000T}{z-1}$$

The system's characteristic equation is $1 + L(z) = 1 + \frac{20000T}{z-1} = 0$ or

$z - 1 + 20000T = 0$. So the system has one closed loop pole at $z = 1 - 20000T$. For stability it is required to have $|1-20000T| < 1$, or $-1 < 1 - 20000T < 1$ or $-2 < -20000T < 0$ or $0 < T < 1m\sec$.

31.

With the zero order hold the open loop transfer function is

$$G_1(s) = \frac{(1-e^{-s})0.0187K}{s(s^2+0.237s+0.00908)} = \frac{(1-e^{-s})0.0187K}{s(s+0.0481)(s+0.1889)}$$

$$= (1-e^{-s})K\left[\frac{2.06}{s} - \frac{2.76}{s+0.0481} + \frac{0.7031}{s+0.1889}\right]$$

$$G(z) = K\frac{z-1}{z}\left[\frac{2.06z}{z-1} - \frac{2.76z}{z-e^{-0.0481}} + \frac{0.7031z}{z-e^{-0.1889}}\right] = \frac{(0.0031z^2+0.0031z+0.0104)K}{(z-0.953)(z-0.8279)}$$

The characteristic equation for this system is:

$$1+G(z)=1+\frac{(0.0031z^2+0.0031z+0.0104)K}{(z-0.953)(z-0.8279)}=0 \text{ or}$$

$$(1+0.0031K)z^2+(0.0031K-1.7809)z+(0.789+0.0104K)=0$$

We use now the bilinear transformation by substituting $z=\frac{s'+1}{s'-1}$

$$(1+0.0031K)\frac{(s'+1)^2}{(s'-1)^2}+(0.0031K-1.7809)\frac{s'+1}{s'-1}+(0.789+0.0104K)=0$$

giving

$$(0.0081+0.0166K)s'^2+(0.422-0.0146K)s'+(3.5699+0.0104K)=0$$

The Routh array is

s'^2	$0.0081+0.0166K$	$3.5699+0.0104K$
s'	$0.422-0.0146K$	
1	$3.5699+0.0104K$	

The first column of the array will be >0 when K>0 and from the second row

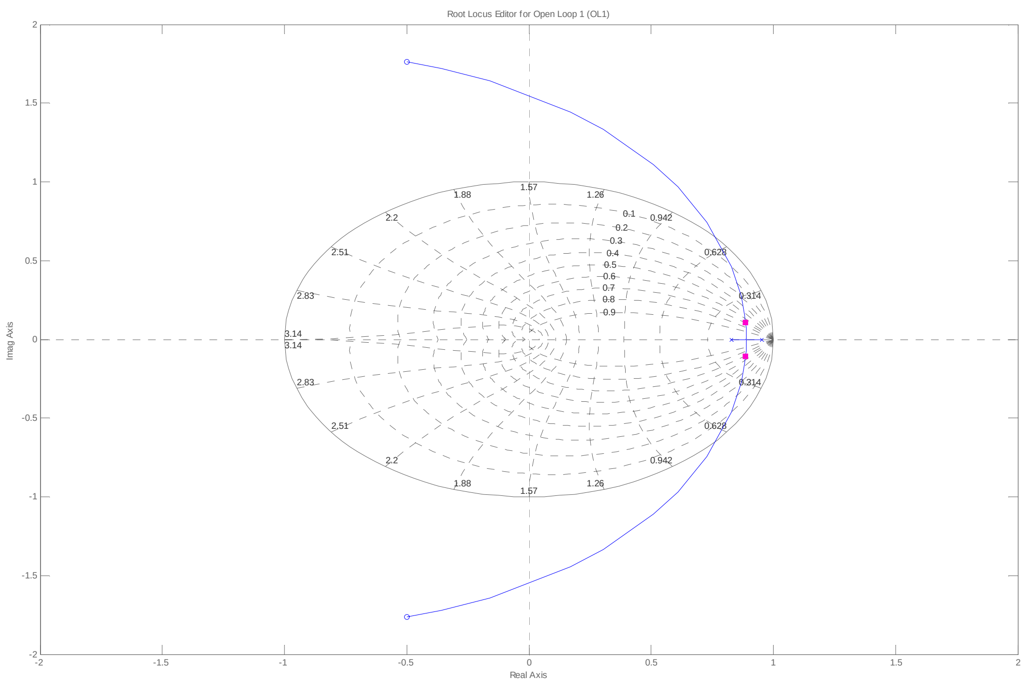
$$K<\frac{0.422}{0.0146}=28.9041$$

So the system is closed loop stable when $0<K<28.9041$.

32.

a. In Problem 31 we found that for this system with T=1sec, $G(z)=\frac{K(0.0031z^2+0.0031z+0.0104)}{(z-0.953)(z-0.8279)}$

In MATLAB this system is defined as G=tf(0.0031*[1 1 3.3548],conv([1 -0.953],[1 -0.8279]),1). Invoking SISOTOOL one gets



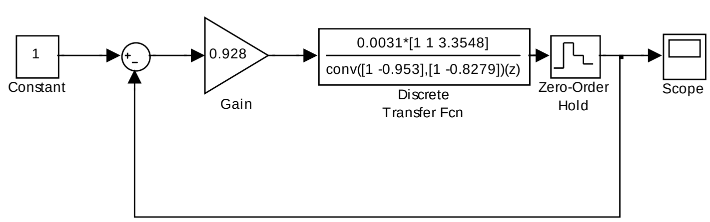
b. $\zeta = 0.7$ is achieved when K=0.928

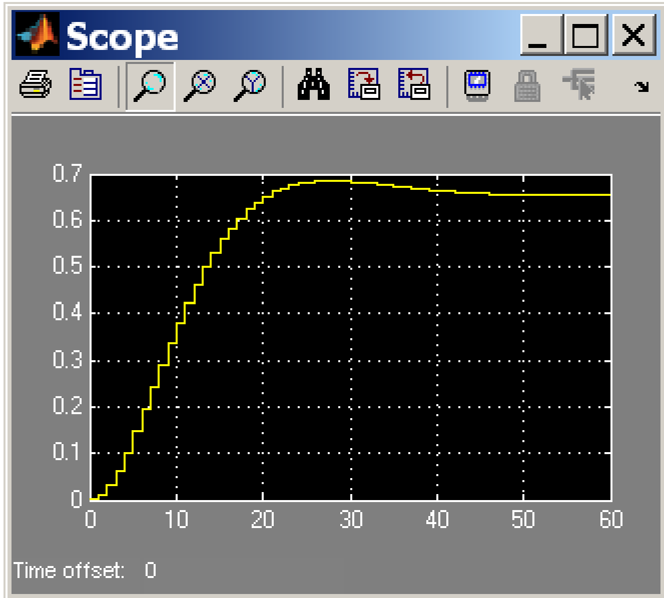
c. The closed loop poles are located at 0.866±j0.103. The radial distance from the origin is $r = \sqrt{0.866^2 + 0.103^2} = 0.872$, so $T_s = \dfrac{-4T}{\ln(r)} = 29.2\sec$. The radial angle from the origin is

$$\theta_1 = \tan^{-1}\frac{0.103}{0.866} = 6.8^\circ = 0.12rad$$, so $\dfrac{T_p}{T} = \dfrac{\pi}{\theta_1} = \dfrac{\pi}{0.12} = 26.54$ or $T_p = 26.54\sec$

d. We have that $K_p = \lim\limits_{z\to 1} G(z) = \lim\limits_{z\to 1} \dfrac{(0.928)(0.0031)(z^2 + z + 3.3548)}{(z-0.953)(z-0.8279)} = 1.905$. The steady state error is $e_{ss} = \dfrac{1}{1+K_p} = 0.344$. Then $c(\infty) = 1 - 0.344 = 0.655$

e.





33.

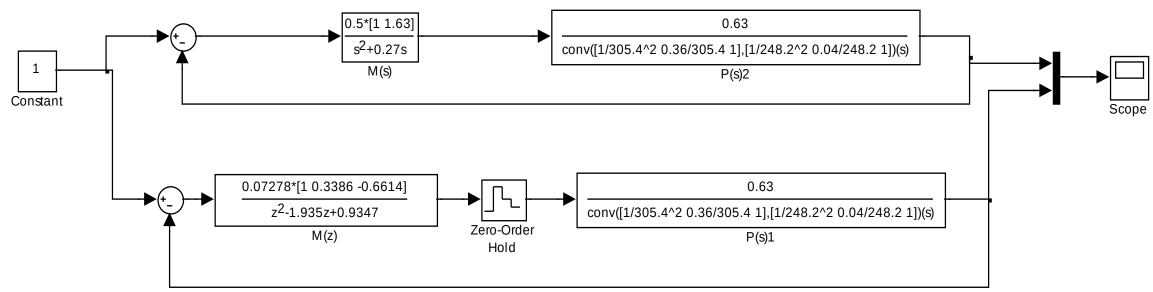
a. A bode plot of the open loop transmission $L(s) = M(s)P(s)$ shows that the crossover frequency for this system is $\omega_c = 0.726 \frac{rad}{\sec}$. The recommended range for the sampling frequency is:

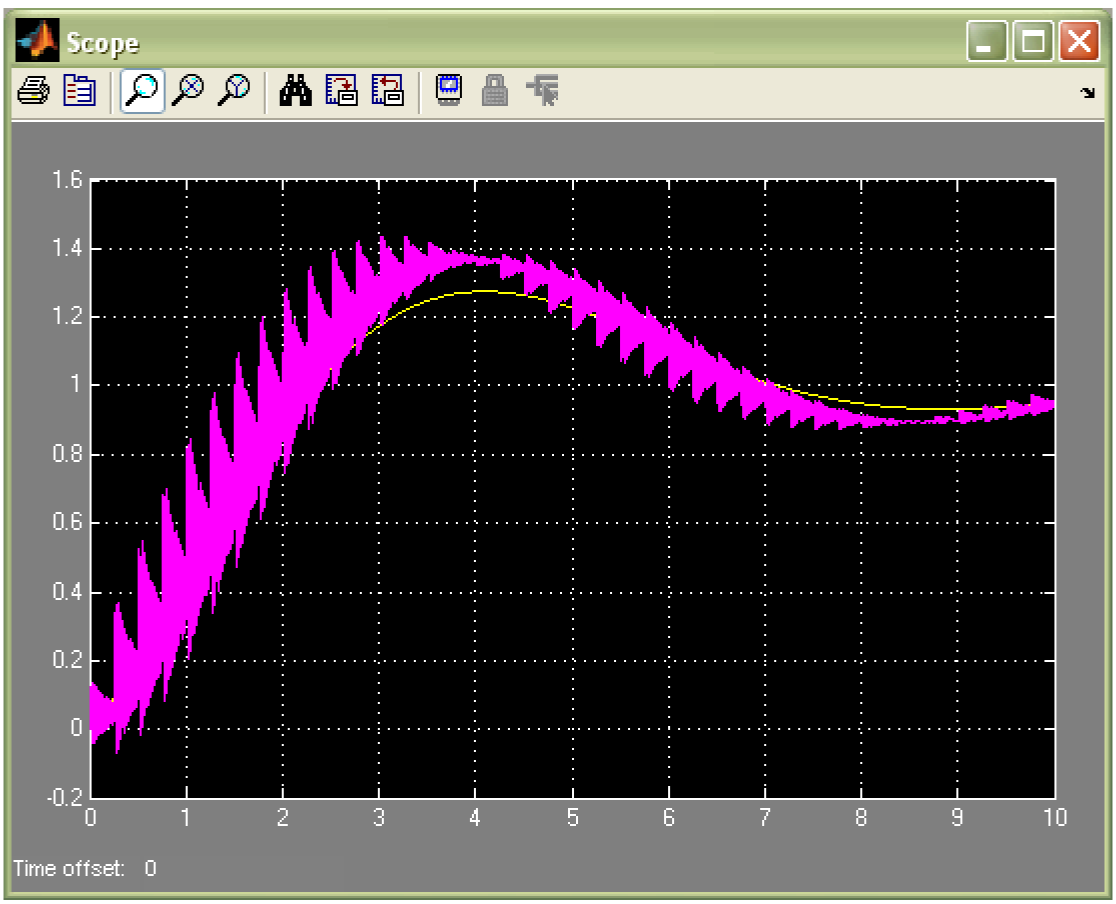
$0.2066 \sec = \frac{0.15}{\omega_c} < T < \frac{0.5}{\omega_c} = 0.6887 \sec$. Arbitrarilly we choose $T = 0.25 \sec$.

b. Substituting $s = 8\frac{z-1}{z+1}$ into $M(s)$, after algebraic manipulations we get

$$M(z) = \frac{0.07278(z^2 + 0.3386z - 0.6614)}{z^2 - 1.935z + 0.9347}$$

c.


1
Constant
0.5*[1 1.63]
s2+0.27s
M(s)
0.63
conv([1/305.4^2 0.36/305.4 1],[1/248.2^2 0.04/248.2 1])(s)
P(s)2
Scope
0.07278*[1 0.3386 -0.6614]
z2-1.935z+0.9347
M(z)
Zero-Order
Hold
P(s)1


Scope
1.6
1.4
1.2
1
0.8
0.6
0.4
0.2
0
-0.2
0 1 2 3 4 5 6 7 8 9 10
Time offset: 0

The output of the system exhibits oscillations caused by a plant high frequency resonance. This problem can be improved by using other discretization techniques.

34.

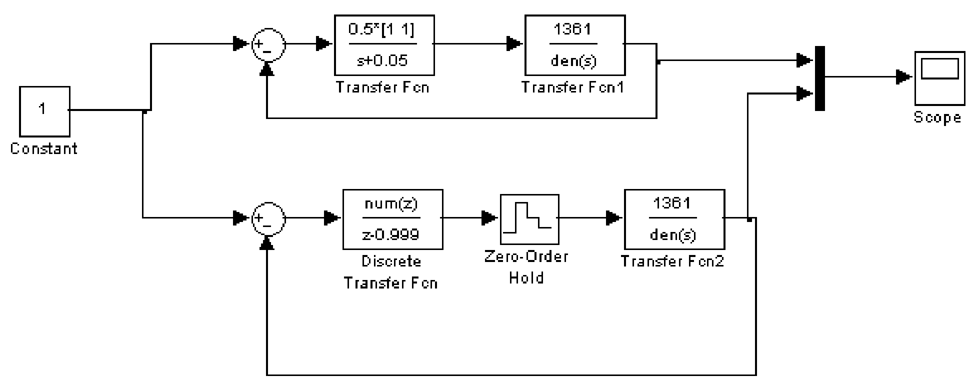
a. A bode plot of the open loop transmission $L(s) = G_c(s)G(s)$ shows that the crossover frequency for this system is $\omega_c = 9.9\frac{rad}{\sec}$. The recommended range for the sampling frequency is:

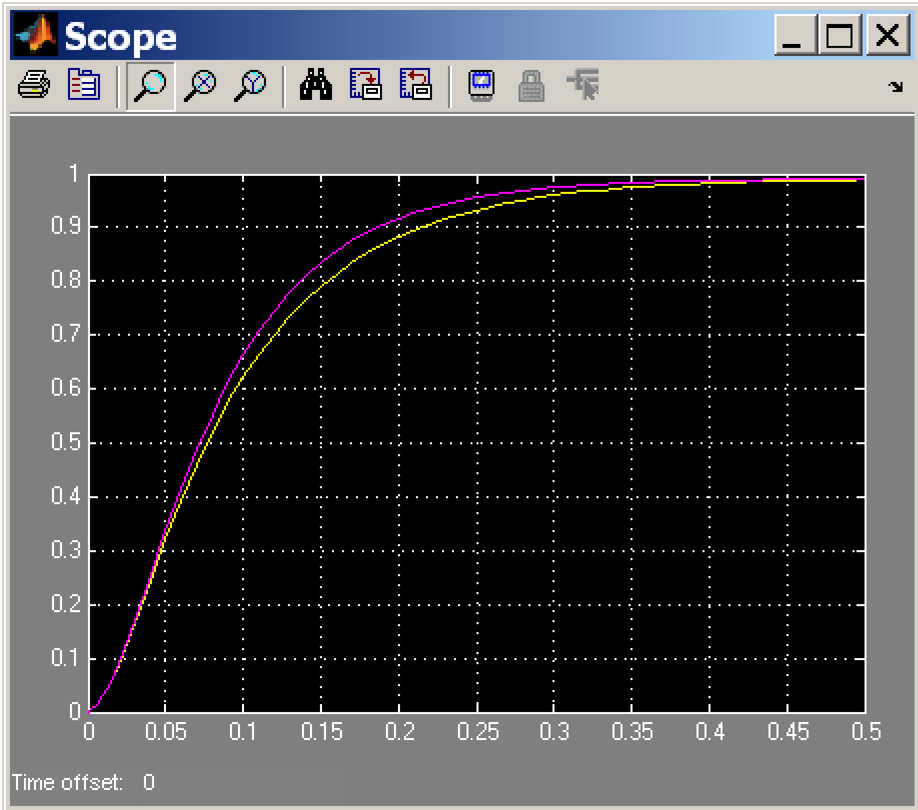
$0.015\sec = \frac{0.15}{\omega_c} < T < \frac{0.5}{\omega_c} = 0.051\sec$. Arbitrarilly we choose $T = 0.02\sec$.

b. Substituting $s = 8\frac{z-1}{z+1}$ into $G_C(s)$, after algebraic manipulations we get

$$G_c(z) = \frac{0.5047z - 0.4948}{z - 0.999}$$

c.





35.

a. With the zero order hold the open loop transfer function is

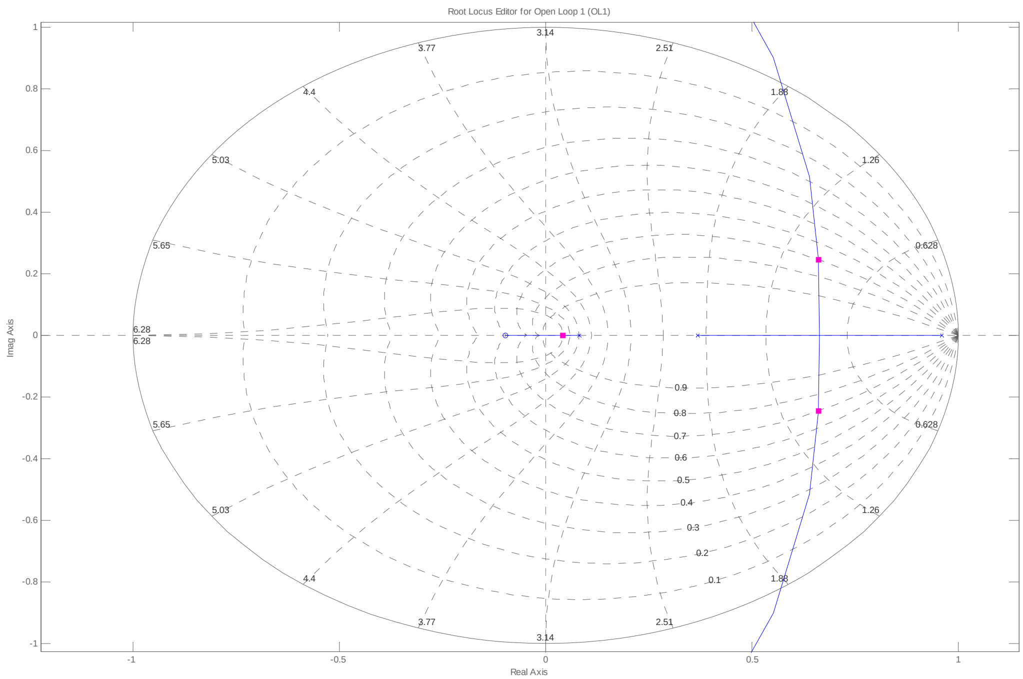
$$G_1(s) = \frac{(1-e^{-0.5s})K}{s(s+0.08)(s+2)(s+5)}$$

$$= (1-e^{-0.5s})K\left[\frac{1.25}{s} - \frac{1.3233}{s+0.08} + \frac{0.0868}{s+2} - \frac{0.0136}{s+5}\right]$$

$$G(z) = K\frac{z-1}{z}\left[\frac{1.25z}{z-1} - \frac{1.3233z}{z-0.961} + \frac{0.0868z}{z-0.368} - \frac{0.0136z}{z-0.0821}\right]$$

$$= \frac{-1\times10^{-4}K(z^3-94z^2-174z-16)}{(z-0.0821)(z-0.368)(z-0.961)} == \frac{-1\times10^{-4}K(z^3-94z^2-174z-16)}{z^3-1.411z^2+0.4628z-0.02903}$$

In MATLAB this system is defined as G=tf(-1e-4*[1 -94 -174 -16],[1 -1.411 0.4628 -0.02903]),0.5). Invoking SISOTOOL one gets



b. $\zeta = 0.7$ is achieved when K=5.15

c. The closed loop poles are located at 0.661±j0.247. The radial distance from the origin is

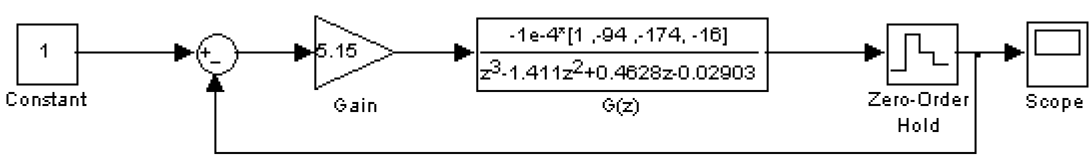
$r = \sqrt{0.661^2 + 0.247^2} = 0.706$, so $T_s = \dfrac{-4T}{\ln(r)} = 5.75\sec$. The radial angle from the origin is

$\theta_1 = \tan^{-1}\dfrac{0.247}{0.661} = 20.49^\circ = 0.36rad$, so $\dfrac{T_p}{T} = \dfrac{\pi}{\theta_1} = \dfrac{\pi}{0.36} = 8.8$ or $T_p = 4.4\sec$

d. We have that $K_p = \lim_{z\to 1} G(z) = \dfrac{(5.15)(-1\times10^{-4})(1-94-174-16)}{(1-1.411+0.4628-0.02903)} = 6.4$. The steady state error is

$e_{ss} = \dfrac{1}{1+K_p} = 0.135$. Then $c(\infty) = 1 - 0.135 = 0.865$

e.



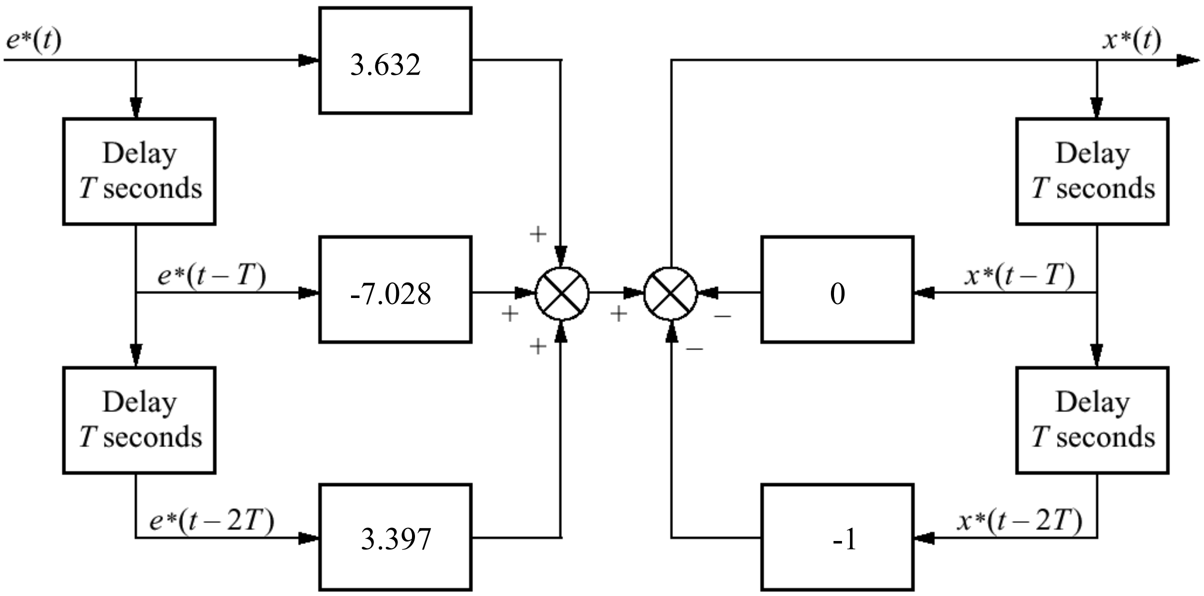
36.

$$G_{PID}(s) = \frac{0.5857(s+0.19)(s+0.01)}{s}.$$

Substituting Eq. (13.88) with T = 1/3 second,

$$G_c(z) = \frac{3.632z^2 - 7.028z + 3.397}{z^2 - 1} = \frac{3.632(z-0.9967)(z-0.9386)}{(z+1)(z-1)}$$

Drawing the flow diagram yields



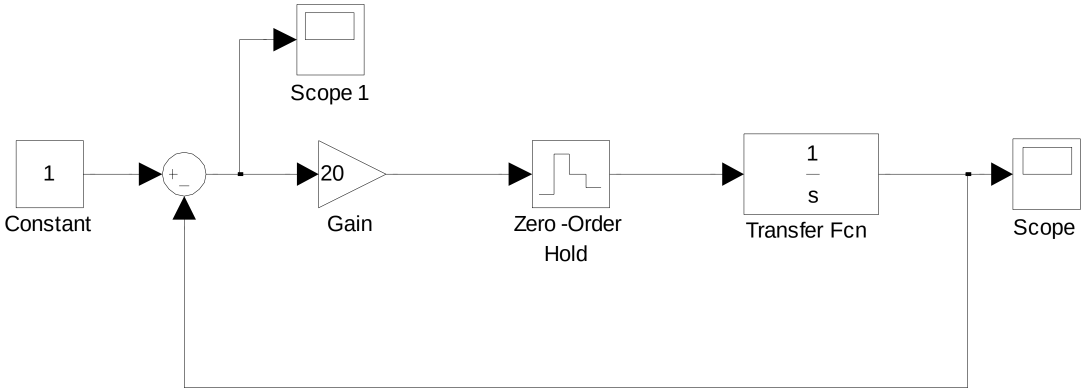
T = 1/3 second

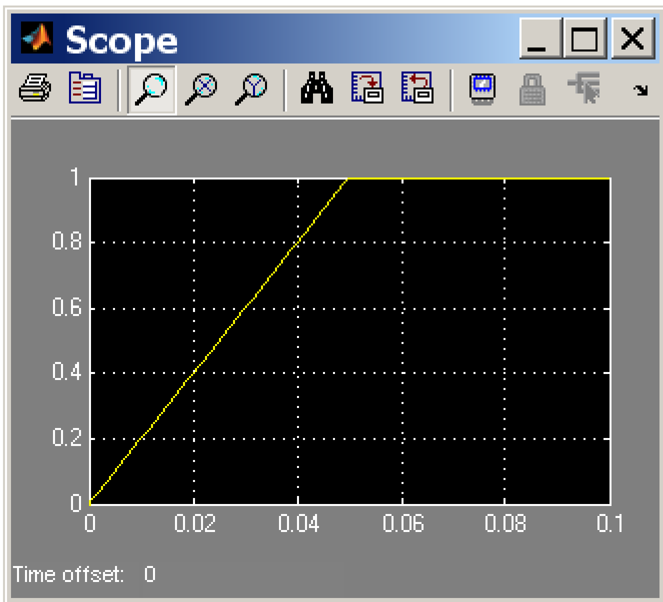
37.

a. The pulse transfer function for the plant is $G_p(z) = \frac{z-1}{z} Z\left\{\frac{1}{s^2}\right\} = \frac{T}{z-1}$. The desired $T(z) = z^{-1}$, so the compensator is $G_c(z) = \frac{1}{G_p(z)} \frac{T(z)}{1-T(z)} = \frac{1}{T}$

b. The steady state error constant is given by $K_v = \frac{1}{T} \lim_{z-1} (z-1) \frac{1}{z-1} = \frac{1}{T}$. So the steady state error for a ramp input is $e(\infty) = \frac{1}{K_p} = T$

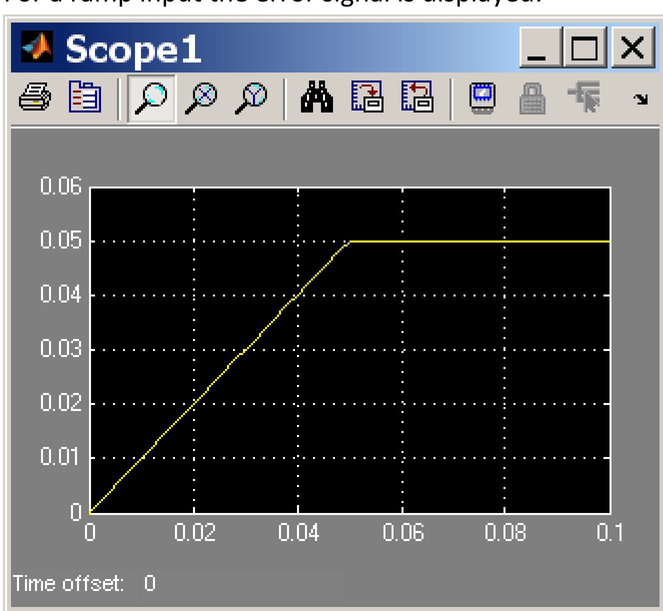
c. The simulations diagram is shown next





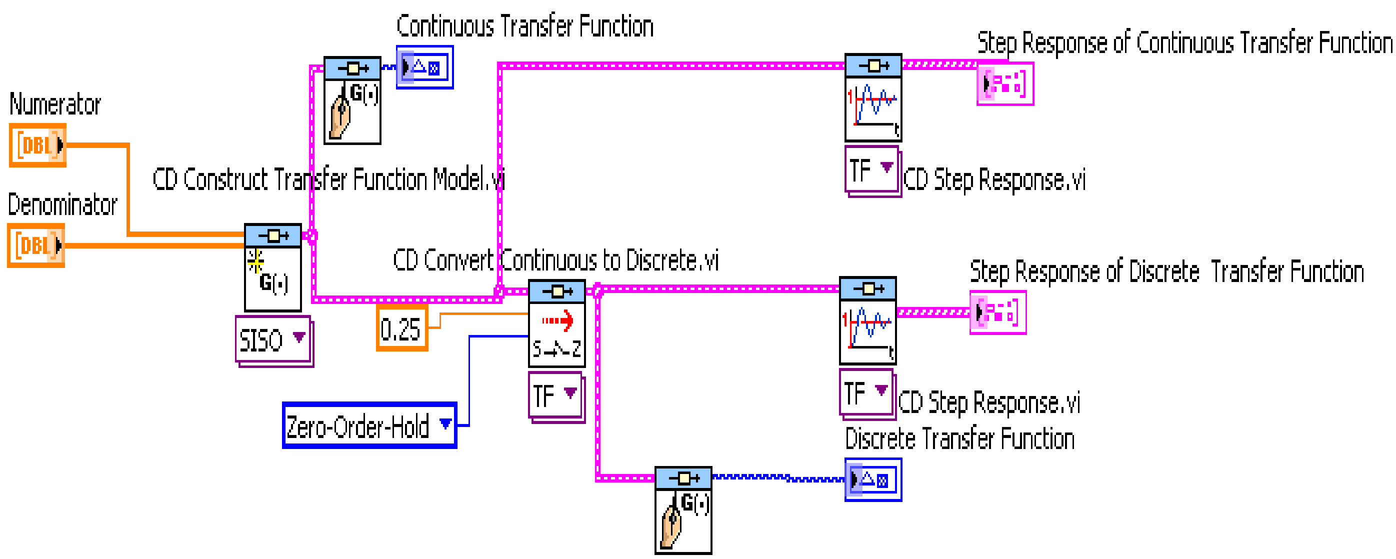
For a step input, the system reaches steady state within one sample.

For a ramp input the error signal is displayed:

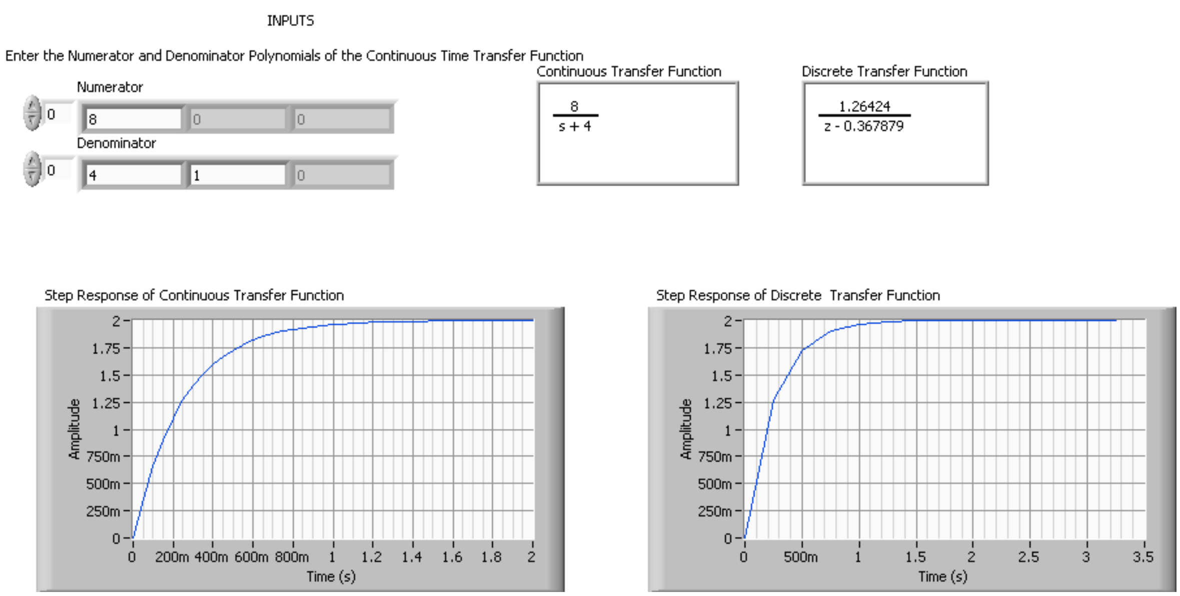


38.

Block Diagram

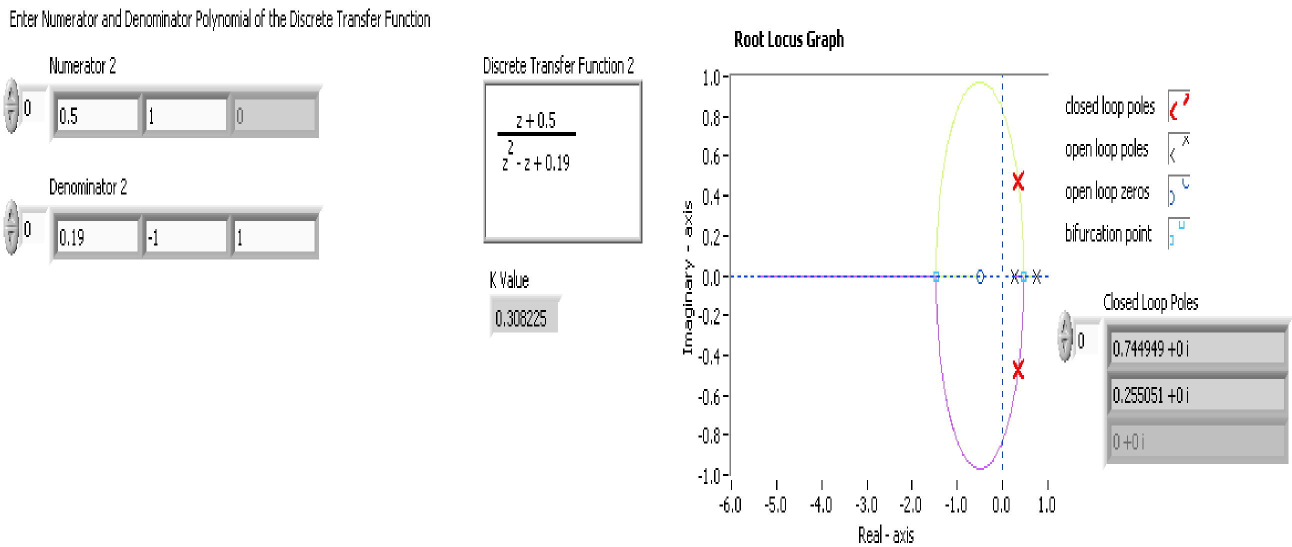


Front Panel

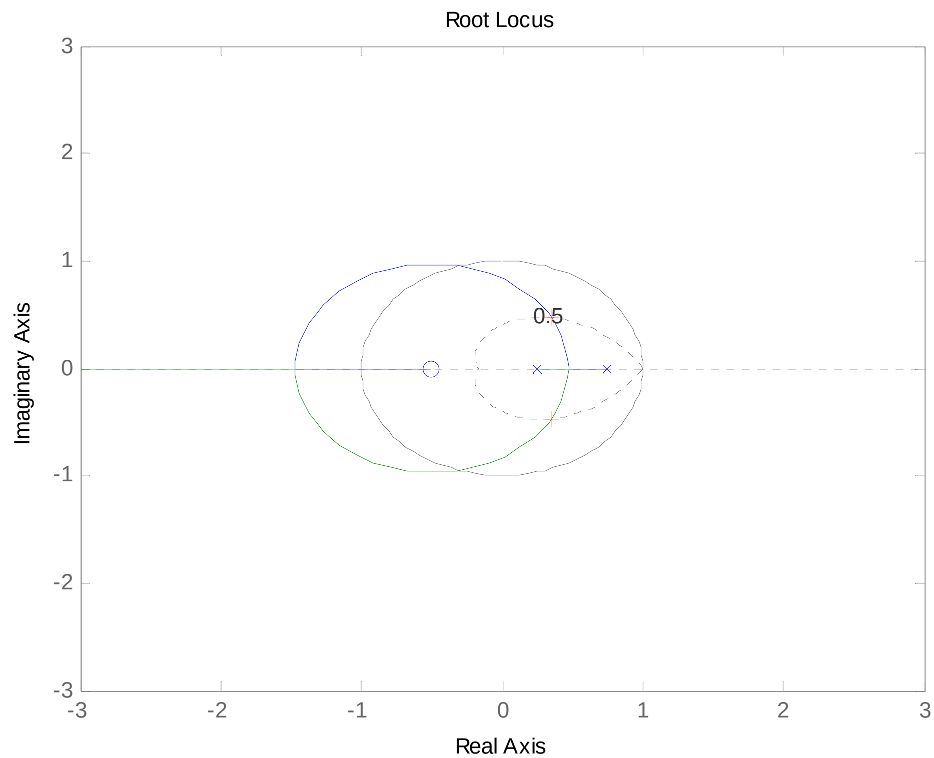


39.

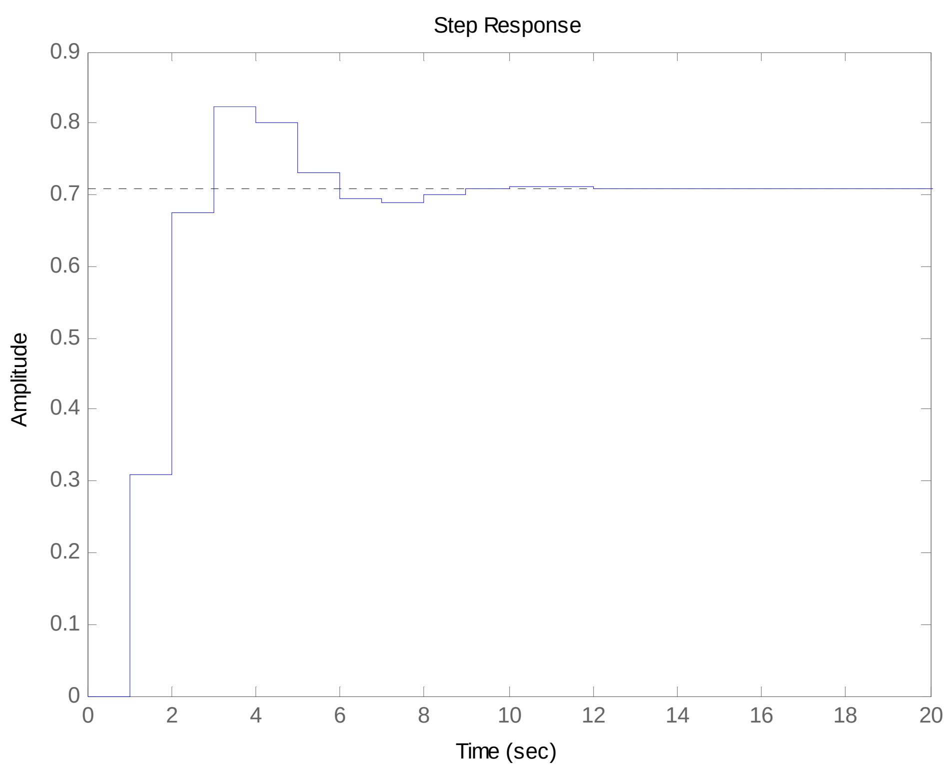
Front panel



Note: K=0.3082 coincides with the answer for Skill-Assessment Exercise 13.8.

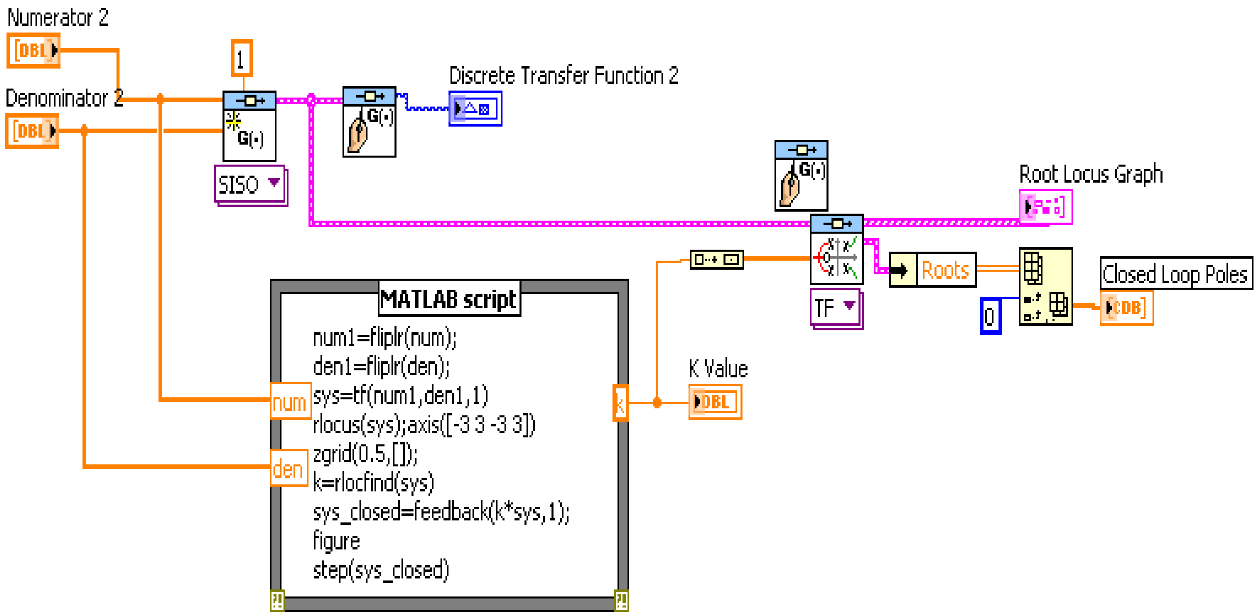


Root Locus Interactive. A trace of z=0.5 was made to guide the user's selection of poles.



Step response of the close loop discrete system

Block Diagram



40.

a. From Chapter 9, the plant without the pots and unity gain power amplifier is

$$G_p(s) = \frac{64.88\ (s+53.85)}{(s^2 + 8.119s + 376.3)\ (s^2 + 15.47s + 9283)}$$

The PID controller and notch filter with gain adjusted for replacement of pots (i.e. divided by 100) was

$$G_c(s) = \frac{26.82\ (s+24.1)\ (s+0.1)\ (s^2 + 16.s + 9200)}{s\ (s+60)^2}$$

Thus, Ge(s) = Gp(s)G_c(s) is

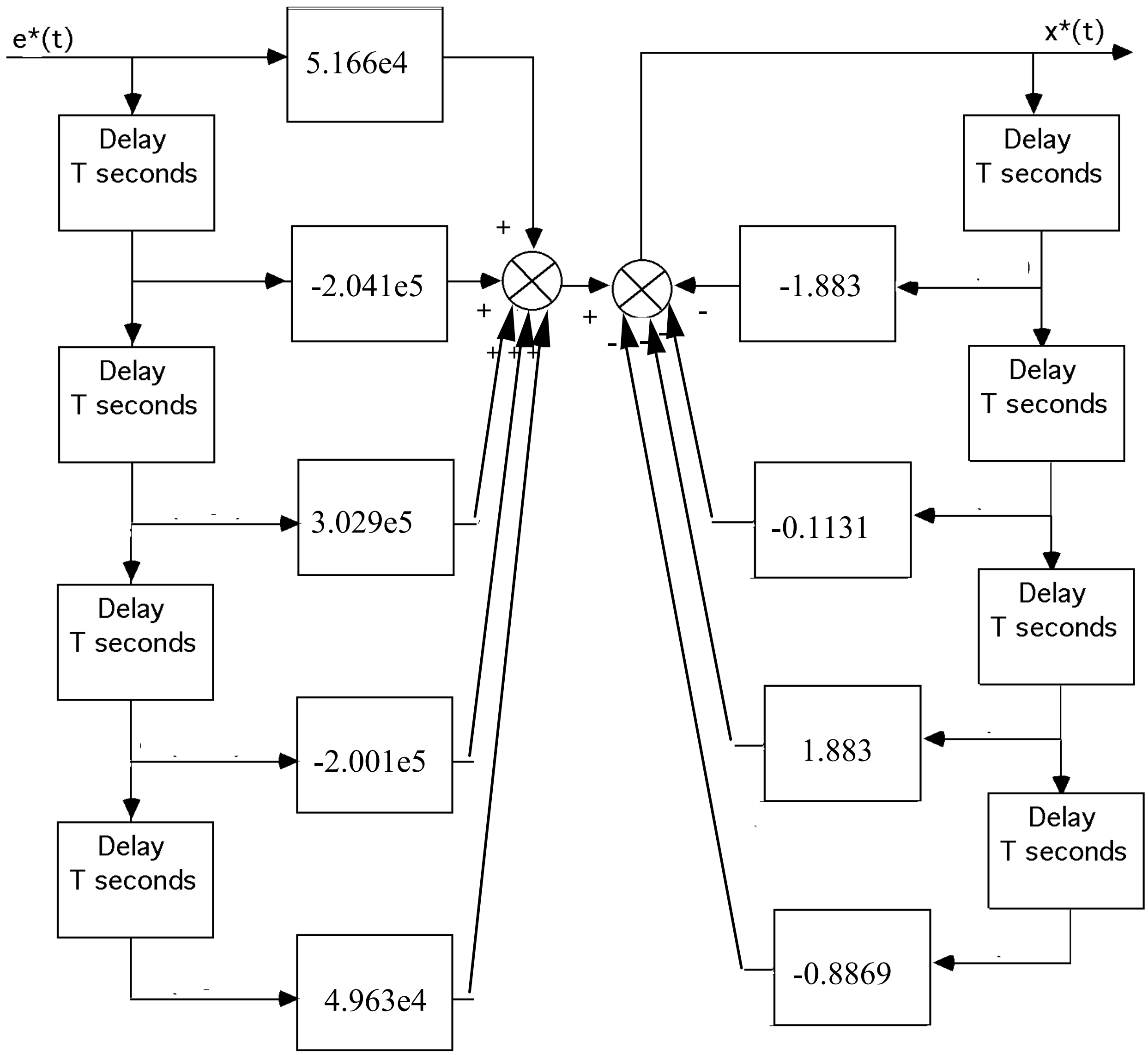
$$G_{et}(s) = \frac{1740.0816\ (s+53.85)(s^2 + 16s + 9200)(s+24.09)(s+0.1)}{s\ (s^2 + 8.119s + 376.3)\ (s^2 + 15.47s + 9283)(s+60)^2}$$

A Bode magnitude plot of $G_e(s)$ shows $\omega_c = 36.375$ rad/s. Thus, the maximum T should be in the range $0.15/\omega_c$ to $0.5/\omega_c$ or 4.1237e-03 to 1.3746e-02. Let us select T = 0.001.

Performing a Tustin transformation on $G_c(s)$ yields

$$G_c(z) = \frac{5.166e04\ z^4 - 2.041e05\ z^3 + 3.029e05\ z^2 - 2.001e05\ z + 4.963e04}{Z^4 - 1.883\ z^3 - 0.1131\ z^2 + 1.883\ z - 0.8869}$$

b. Drawing the flowchart



T = 0.001

c.

Program:

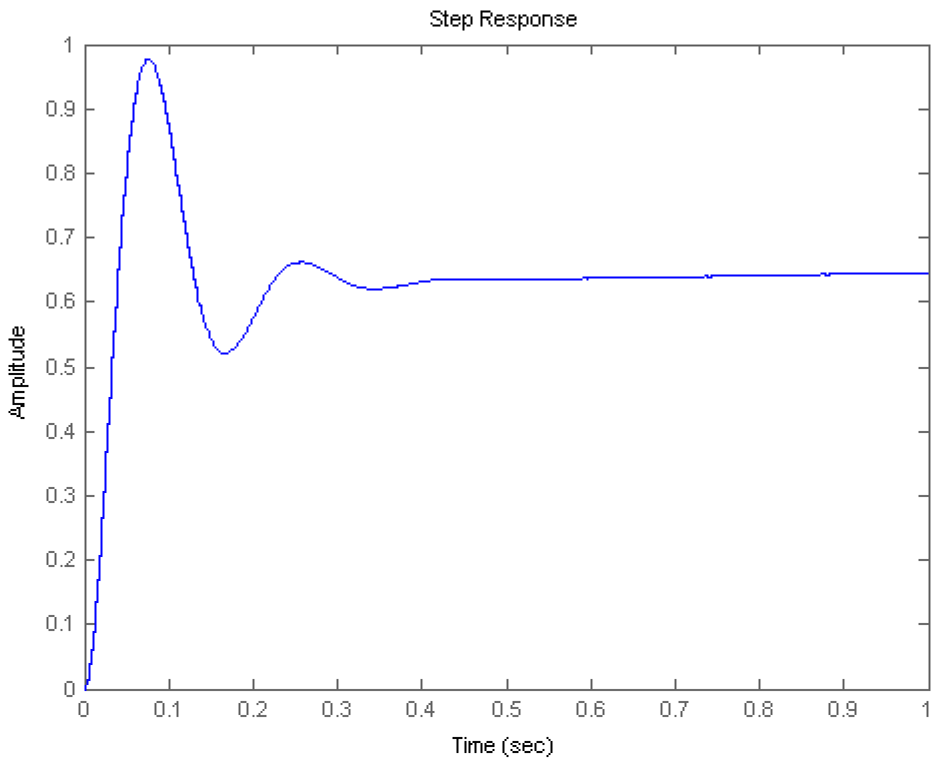
```
syms s
'Compensator from Chapter 9'
T=.001
Gc=26.82*(s^2+16*s+9200)*(s+24.09)*(s+.1)/(s*((s+60)^2));
Gc=vpa(Gc,4);
[numgc,dengc]=numden(Gc);
numgc=sym2poly(numgc);
dengc=sym2poly(dengc);
```

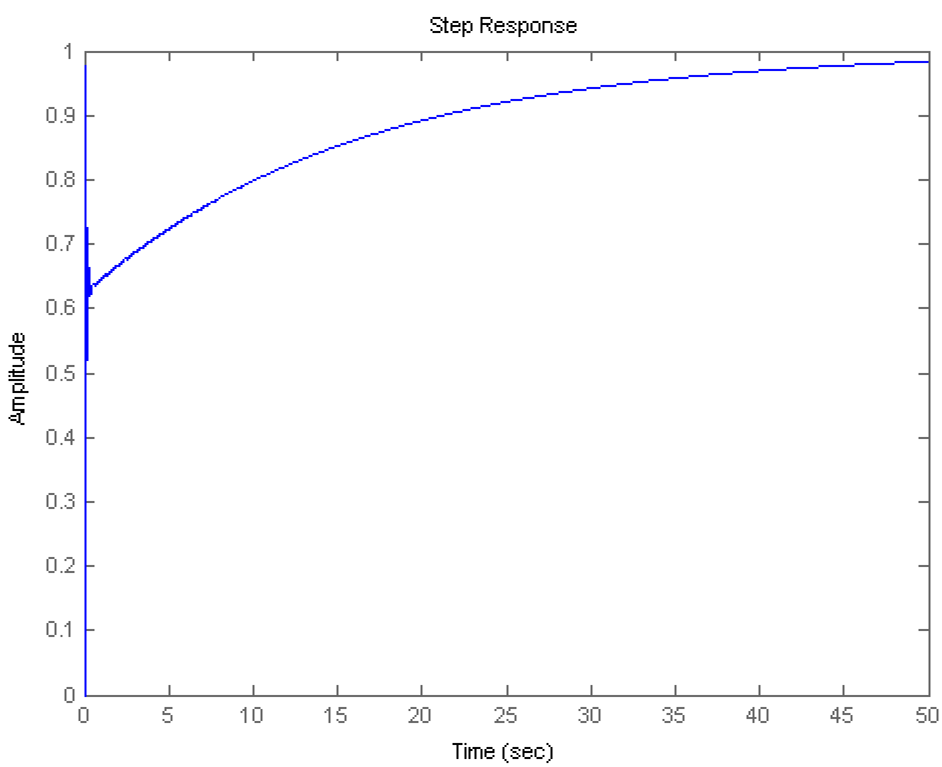
```
Gc=tf(numgc,dengc);
'Gc(s)'
Gczpk=zpk(Gc)
'Gc(z)'
Gcz=c2d(Gc,T,'tustin')
'Gc(z)'
Gczzpk=zpk(Gcz)
'Plant from Chapter 9'
Gp=64.88*(s+53.85)/[(s^2+15.47*s+9283)*(s^2+8.119*s+376.3)];
Gp=vpa(Gp,4);
[numgp,dengp]=numden(Gp);
numgp=sym2poly(numgp);
dengp=sym2poly(dengp);
'Gp(s)'
Gp=tf(numgp,dengp)
'Gp(s)'
Gpzpk=zpk(Gp)
'Gp(z)'
Gpz=c2d(Gp,T,'zoh')
'Gez=Gcz*Gpz'
Gez=Gcz*Gpz
Tz=feedback(Gez,1);
t=0:T:1;
step(Tz,t)
pause
t=0:T:50;
step(Tz,t)
```

Computer response:

```
ans =

Compensator from Chapter 9


T =

    0.0010


ans =

Gc(s)


Zero/pole/gain:
26.82 (s+24.09) (s+0.1) (s^2  + 16s + 9198)
-------------------------------------------
                s (s+60)^2


ans =

Gc(z)


Transfer function:
5.17e004 z^4 - 2.043e005 z^3 + 3.031e005 z^2 - 2.002e005 z + 4.966e004
----------------------------------------------------------------------
           z^4 - 1.883 z^3 - 0.1131 z^2 + 1.883 z - 0.8869

Sampling time: 0.001

ans =

Gc(z)
```

```
Zero/pole/gain:
51699.4442 (z-1) (z-0.9762) (z^2  - 1.975z + 0.9842)
----------------------------------------------------
              (z+1) (z-1) (z-0.9417)^2

Sampling time: 0.001

ans =

Plant from Chapter 9


ans =

Gp(s)


Transfer function:
                   64.88 s + 3494
----------------------------------------------------
s^4 + 23.59 s^3 + 9785 s^2 + 8.119e004 s + 3.493e006


ans =

Gp(s)


Zero/pole/gain:
               64.88 (s+53.85)
----------------------------------------------
(s^2  + 8.119s + 376.3) (s^2  + 15.47s + 9283)


ans =

Gp(z)


Transfer function:
1.089e-008 z^3 + 3.355e-008 z^2 - 3.051e-008 z - 1.048e-008
-----------------------------------------------------------
       z^4 - 3.967 z^3 + 5.911 z^2 - 3.92 z + 0.9767

Sampling time: 0.001

ans =

Gez=Gcz*Gpz


Transfer function:

0.000563 z^7 - 0.0004901 z^6 - 0.005129 z^5 + 0.01368 z^4 - 0.01328 z^3

                                 + 0.004599 z^2 + 0.0005822 z - 0.0005203

--------------------------------------------------------------------------
z^8 - 5.85 z^7 + 13.27 z^6 - 12.72 z^5 - 0.6664 z^4 + 13.25 z^3

                                           - 12.74 z^2 + 5.317 z - 0.8662


Sampling time: 0.001
```


Step Response
Amplitude
Time (sec)
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1


Step Response
Amplitude
Time (sec)
0
5
10
15
20
25
30
35
40
45
50

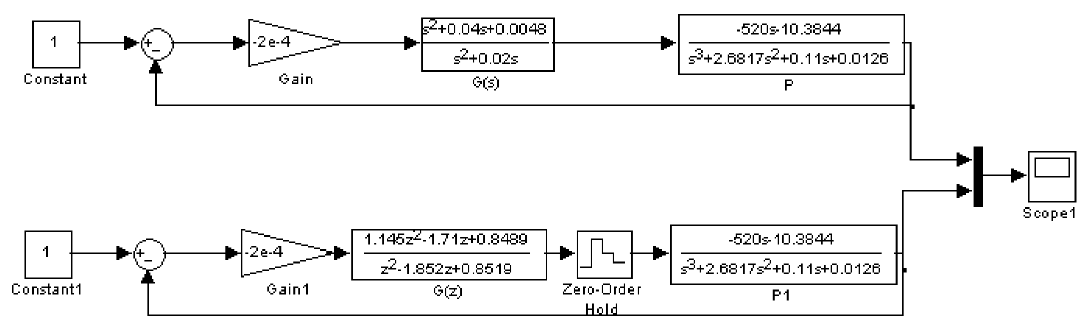
41.

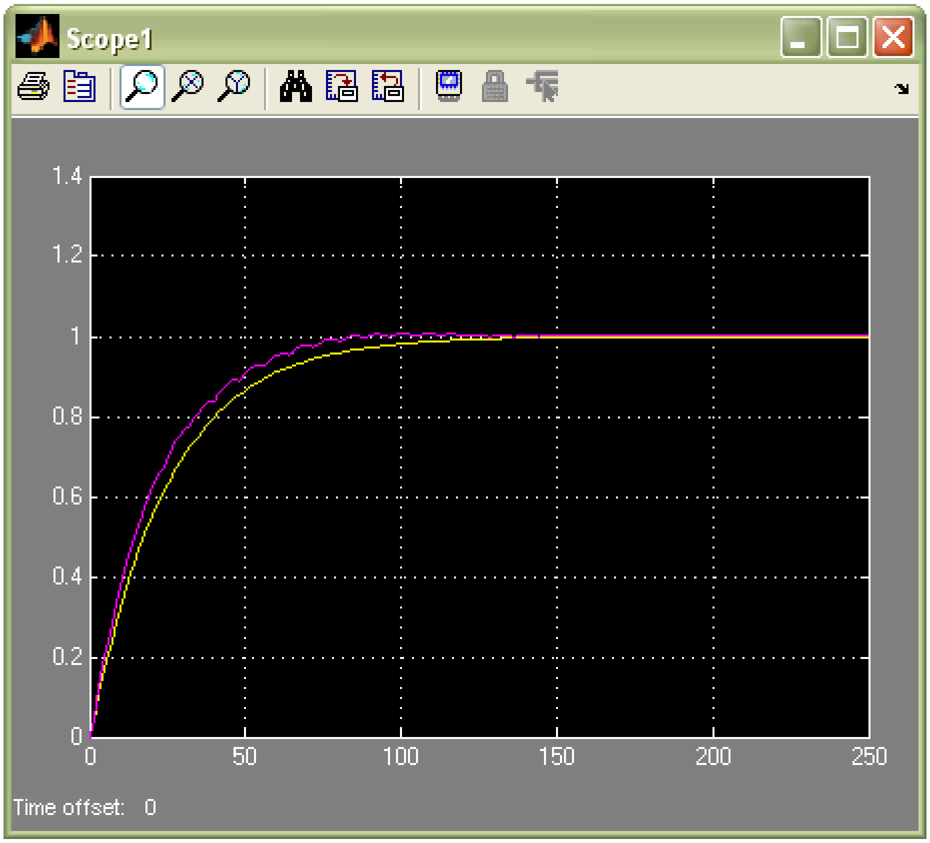
a. A bode plot of the open loop transmission $G_c(s)P(s)$ shows that the open loop transfer function has a crossover frequency of $\omega_c = 0.04\dfrac{rad}{day}$. A convenient range for sampling periods is $3.75day = \dfrac{0.15}{\omega_c} < T < \dfrac{0.5}{\omega_c} = 12.5day$. T=8 days fall within range.

b. We substitute $s = \dfrac{1}{4}\dfrac{z-1}{z+1}$ into $G_c(s)$ we get

$$G_c(z) = \frac{-2\times10^{-4}(1.145z^2 - 1.71z + 0.8489)}{z^2 - 1.852z + 0.8519}$$

c.





42.

a. The following MATLAB M-file was developed

```
%Digitize G1(s) preceded by a sample and hold
%Input transient response specifications
Po=input('Type %OS ');
K = input('Type Proportional Gain of PI controller  ');

    numg1 = K*poly([-0.01304 -0.6]);
    deng1 = poly([0 -0.01631 -0.5858]);
    G1 = tf(numg1,deng1);
       for T=5:-.01:0;
       Gz=c2d(G1,T,'zoh');
```

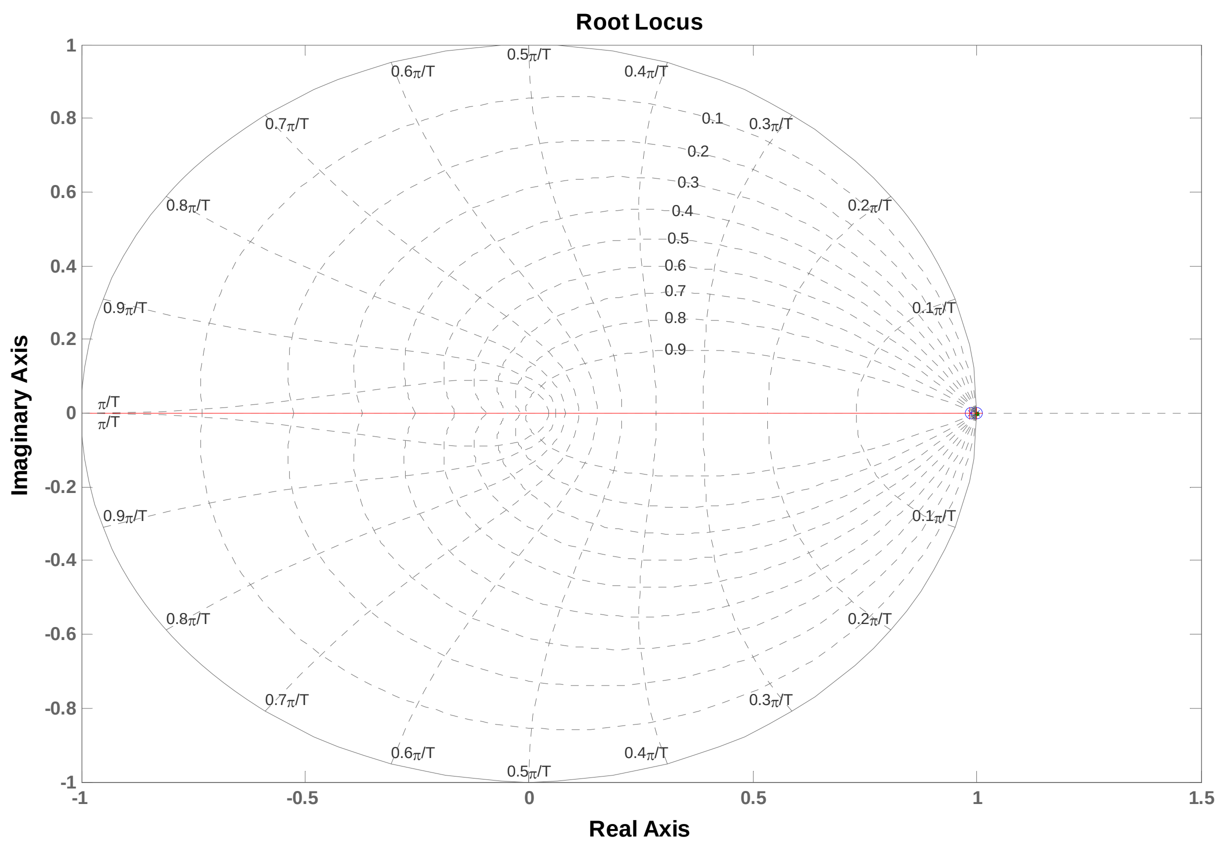
```
        Tz=feedback(Gz,1);
          r=pole(Tz);
          rm=max(abs(r));
          if rm<=1;
          break;
          end;
        end;
'G1(s)';
G1s=tf(numg1,deng1);
'G(z)';
Gz=c2d(G1s,0.75*T,'zoh');
%Determine damping ratio
z=(-log(Po/100))/(sqrt(pi^2+log(Po/100)^2));
%Plot root locus
rlocus(Gz)
zgrid(z,0)
title('Root Locus')
[K,p]=rlocfind(Gz); %Allows input by selecting point on graphic
pause
'T(z)';
Tz=feedback(K*Gz,1);
step(Tz)
```

b. As the M-file developed in (a) was run and the values of the desired percent overshoot, *%O.S.* = 0, and PI speed controller's proportional gain, $K = 61$ were entered, the root locus, shown below, was obtained.

c. A point was selected on the root locus such that is inside the unit circle. That point is: 0.9837 + 0.0000i

d. The sampled data transfer functions, G_z and T_z, obtained at a Sampling time, $T = 0.0225 = 0.75*0.03$ are:

$$G_z = \frac{1.373\,z^2 - 2.727\,z + 1.354}{z^3 - 2.987\,z^2 + 2.973\,z - 0.9865}$$



and

$$T_z = \frac{0.018\,z^2 - 0.03575\,z + 0.01775}{z^3 - 2.969\,z^2 + 2.973\,z - 0.9688}$$

The poles, r, of the closed-loop transfer function, T_z, are:

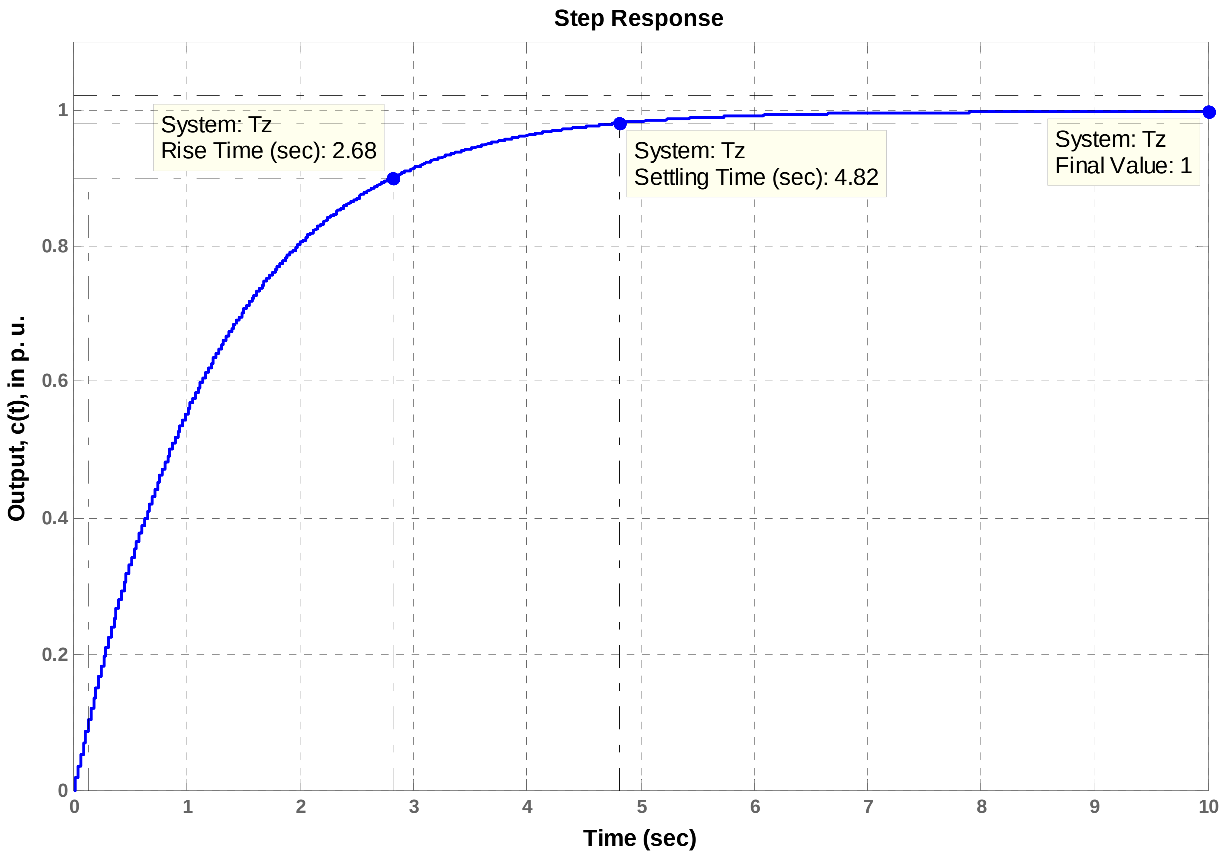
$$-0.8300,\ 0.9996,\ 0.9822$$

e. The step response of this digital system, T_z, is shown below with the main transient response characteristics (the final value, rise time, and settling time) displayed on the graph. These are:

Final Value = $c(\infty)$ = 1 p. u.;

Rise Time = T_r = 2.68 seconds;

Settling Time = T_s = 4.82 seconds.


Step Response
System: Tz
Rise Time (sec): 2.68
System: Tz
Settling Time (sec): 4.82
System: Tz
Final Value: 1
Output, c(t), in p. u.
0
0.2
0.4
0.6
0.8
1
0
1
2
3
4
5
6
7
8
9
10
Time (sec)

Founded in 1807, John Wiley & Sons, Inc. has been a valued source of knowledge and understanding for more than 200 years, helping people around the world meet their needs and fulfill their aspirations. Our company is built on a foundation of principles that include responsibility to the communities we serve and where we live and work. In 2008, we launched a Corporate Citizenship Initiative, a global effort to address the environmental, social, economic, and ethical challenges we face in our business. Among the issues we are addressing are carbon impact, paper specifications and procurement, ethical conduct within our business and among our vendors, and community and charitable support. For more information, please visit our website: www.wiley.com/go/citizenship.

ISBN 13 978-0470-54756-4