Xi'an Jiaotong-Liverpool University

西交利物浦大学

Department of Electrical and Electronic Engineering

EEE220 Instrumentation and Control System

2018-2019, Semester 2

Experiment 2:

Control System CAD and CAS using Matlab

Name: Kai-Yu Lu

Student ID: 1614649

Date: 2019.5.18

Problem 1

Consider the differential equation

$$\ddot{y} + 3\dot{y} + 2y = u$$

Where $y(0) = \dot{y}(0) = 0$ and $u(t)$ is a unit step.

Determine the solution $y(t)$ analytically:

Laplace transform of first-order derivatives:

$$\mathcal{L}\{f'(t)\} = s\,\mathcal{L}\{f(t)\} - f(0)$$

Laplace transform of second-order derivatives:

$$\mathcal{L}\{f''(t)\} = s^2\mathcal{L}\{f(t)\} - s\,f(0) - f'(0)$$

Take Laplace transform on both sides of the equation, it becomes

$$[s^2Y(s) - sy(0) - y'(0)] + 3[sY(s) - y(0)] + 2Y(s) = \frac{1}{s}$$

$$(s^2 + 3s + 2)Y(s) = \frac{1}{s}$$

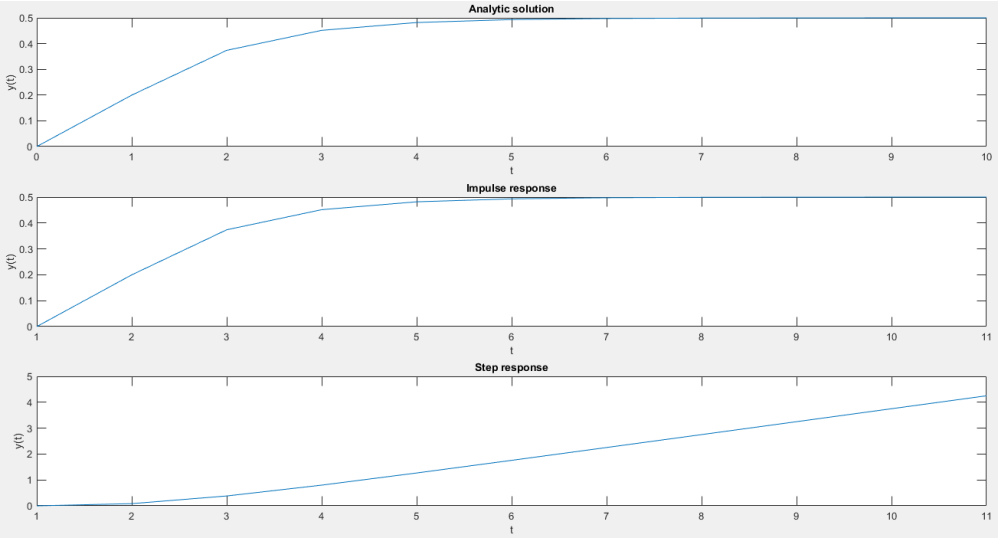
$$Y(s) = \frac{\frac{1}{2}}{s} + \frac{-1}{s+1} + \frac{\frac{1}{2}}{s+2}$$

Below is the Matlab code for co-plotting the analytic solution and the step and impulse response obtained with the step and impulse function.

Codes:

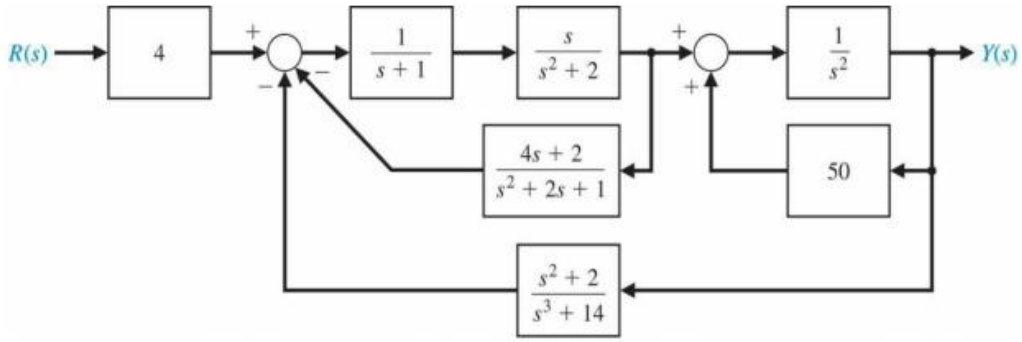
```
GA=tf(1, [1 3 2 0]);
t = 0:1:10;
y = 0.5+-exp(-t)+0.5*exp(-2*t);

subplot(3,1,1),plot(t,y);xlabel('t'),ylabel('y(t)'),title('Analytic solution')
subplot(3,1,2),plot(impulse(GA,t));xlabel('t'),ylabel('y(t)'),title('Impulse response')
subplot(3,1,3),plot(step(GA,t));xlabel('t'),ylabel('y(t)'),title('Step response')
```

Results:



Problem 2

(a) Use an m-file to reduce the block diagram in Fig. P2, and compute the closed-loop transfer function



m-file code:

```
G1=tf(1, [1 1]);
G2=tf([1 0], [1 0 2]);
G3=tf(1, [1 0 0]);
G4=tf([4 2], [1 2 1]);
G5=tf([1 0 2], [1 0 0 14]);
H1=tf(4, 1);
H2=tf(50, 1);

G12=series(G1,G2);
GA=feedback(G12,G4,-1);

GB=feedback(G3,H2,1);
GC=series(GA,GB);
GD=feedback(GC,G5,-1);
GE=series(H1,GD);
```

Computed close-loop transfer function by m-file:

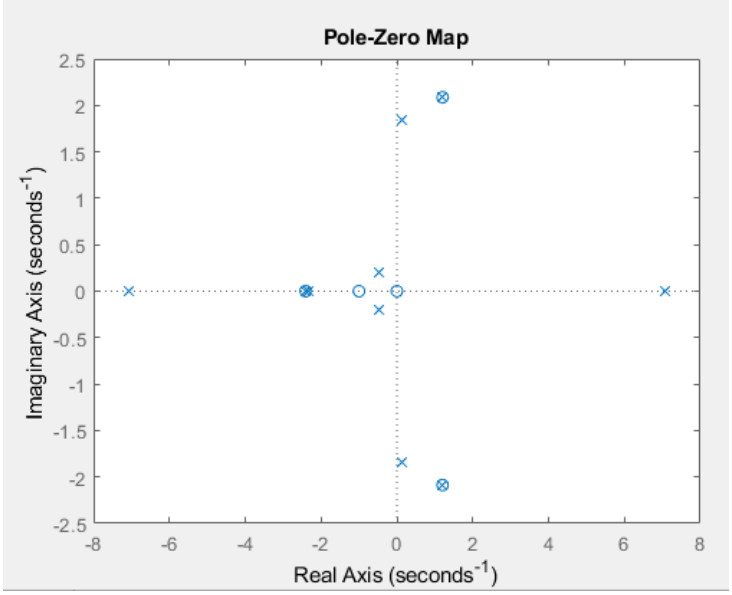
```
GE =

                          4 s^6 + 8 s^5 + 4 s^4 + 56 s^3 + 112 s^2 + 56 s
  ---------------------------------------------------------------------------------------------------
  s^10 + 3 s^9 - 45 s^8 - 125 s^7 - 200 s^6 - 1177 s^5 - 2344 s^4 - 3485 s^3 - 7668 s^2 - 5598 s - 1400

Continuous-time transfer function.
```

(b) Generate a pole-zero map of the closed-loop transfer function in graphical form using the pzmap function.

Code:

```
G1=tf(1, [1 1]);
G2=tf([1 0], [1 0 2]);
G3=tf(1, [1 0 0]);
G4=tf([4 2], [1 2 1]);
G5=tf([1 0 2], [1 0 0 14]);
H1=tf(4, 1);
H2=tf(50, 1);

G12=series(G1,G2);
GA=feedback(G12,G4,-1);

GB=feedback(G3,H2,1);
GC=series(GA,GB);
GD=feedback(GC,G5,-1);
GE=series(H1,GD);

pzmap(GE)
```

Result:



(c) Determine explicitly the poles and zeros of the closed-loop transfer function using the pole and zero functions and correlate the results with the pole-zero map in part (b).

Code:

```
G1=tf(1, [1 1]);
G2=tf([1 0], [1 0 2]);
G3=tf(1, [1 0 0]);
G4=tf([4 2], [1 2 1]);
G5=tf([1 0 2], [1 0 0 14]);
H1=tf(4, 1);
H2=tf(50, 1);

G12=series(G1,G2);
GA=feedback(G12,G4,-1);

GB=feedback(G3,H2,1);
GC=series(GA,GB);
GD=feedback(GC,G5,-1);
GE=series(H1,GD);

pzmap(GE)
pole(GE)
zero(GE)
```

Result:

For poles:

```
ans =

   7.0709 + 0.0000i
  -7.0713 + 0.0000i
   1.2051 + 2.0863i
   1.2051 - 2.0863i
   0.1219 + 1.8374i
   0.1219 - 1.8374i
  -2.3933 + 0.0000i
  -2.3333 + 0.0000i
  -0.4635 + 0.1997i
  -0.4635 - 0.1997i
```

For zeroes:

```
ans =

   0.0000 + 0.0000i
   1.2051 + 2.0872i
   1.2051 - 2.0872i
  -2.4101 + 0.0000i
  -1.0000 + 0.0000i
  -1.0000 - 0.0000i
```

Comparing the results with the pzmap function in part(b) with using the pole and zero functions in part(c), the results are the same correspondingly. pzmap function is a function combing pole function with zero function and plot the zeros and poles together in a map.

Problem 3: Determine a state variable representation for the following transfer functions using the ss function.

(a) $\mathbf{G(s)} = \frac{\mathbf{1}}{\boldsymbol{s+12}}$

Code:

```
GA=tf(1, [1 12]);

ss(GA)
```

Result:

```
ans =

  A =
       x1
   x1 -12

  B =
       u1
   x1   1

  C =
       x1
   y1   1

  D =
       u1
   y1   0

Continuous-time state-space model.
```

(b) $\mathbf{G(s)} = \frac{s^2+5s+3}{s^3+8s+5}$

Code:

```
GB=tf([1 5 3], [1 0 8 5]);
ss(GB)
```

Result:

```
ans =

  A =
        x1     x2     x3
  x1     0     -2  -1.25
  x2     4      0      0
  x3     0      1      0

  B =
      u1
  x1   2
  x2   0
  x3   0

  C =
        x1     x2     x3
  y1   0.5  0.625  0.375

  D =
      u1
  y1   0

Continuous-time state-space model.
```

(c) $\mathbf{G(s)} = \frac{s^2+4s+1}{s^4+3s^2+3s+1}$

Code:

```
2 -     GC=tf([1 4 1], [1 0 3 3 1]);
3
4 -     ss(GC)
```

Result:

```
ans =

  A =
        x1    x2    x3    x4
   x1    0  -1.5  -1.5  -0.5
   x2    2     0     0     0
   x3    0     1     0     0
   x4    0     0     1     0

  B =
       u1
   x1   2
   x2   0
   x3   0
   x4   0

  C =
        x1    x2    x3    x4
   y1    0  0.25     1  0.25

  D =
       u1
   y1   0

Continuous-time state-space model.
```

Problem 4

(a) Using the tf function, determine the transfer function Y(s)/U(s).

Code:

```
A=[0 1 0;0 0 1;-3 -2 -5];
B=[1;0;1];
C=[1 0 0];
D=0;

sys=ss(A,B,C,D);

tf(sys)
```

Result:

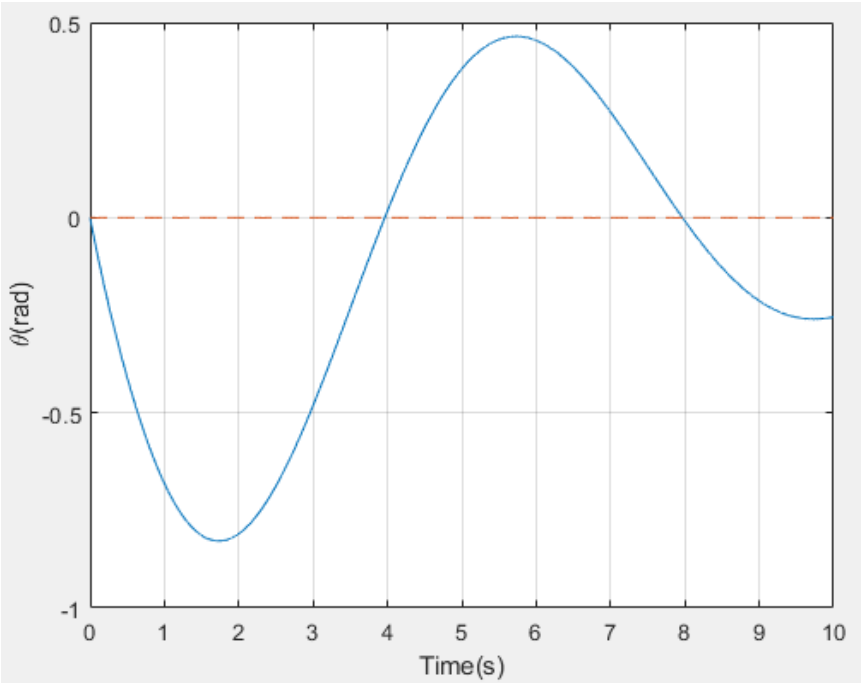
```
ans =

     s^2 + 5 s + 3
  ---------------------
  s^3 + 5 s^2 + 2 s + 3
```

(b) Plot the response of the system to the initial condition $x(0) = [0 \quad -1 \quad 1]^T$ for 0 ≤ t ≤ 10.

Code:

```
A=[0 1 0;0 0 1;-3 -2 -5];
B=[1;0;1];
C=[1 0 0];
D=0;

sys=ss(A,B,C,D);

t = 0:0.001:10;
u = 0*t;
x0=[0;-1;1];
[y,T]=lsim(sys,u,t,x0);
plot(T,y,t,u,'--');
xlabel('Time(s)'),ylabel('\theta(rad)'),grid;
```

Result:



(c) Compute the state transition matrix using the expm function, and determine $x(t)$ at t = 10 for the initial condition given in part (b).

Code:

```
A=[0 1 0;0 0 1;-3 -2 -5];
B=[1;0;1];
C=[1 0 0];
D=0;

sys=ss(A,B,C,D);

t = 0:0.001:10;
u = 0*t;
x0=[0;-1;1];
[y,T]=lsim(sys,u,t,x0);
plot(T,y,t,u,'--');
xlabel('Time(s)'),ylabel('\theta(rad)'),grid;

dt=10;
Phi = expm(A*dt);
x=Phi*x0;
```

Result:

```
x =

   -0.2545
    0.0418
    0.1500
```

$x(t)$ at t = 10

```
Phi =

    0.0853    0.3183    0.0637
   -0.1911   -0.0421   -0.0003
    0.0010   -0.1905   -0.0405
```

State transition matrix

Problem 5

Consider the closed loop transfer function

$$\mathrm{T(s)} = \frac{1}{s^5 + 2s^4 + 2s^3 + 4s^2 + s + 2}$$

(a) Routh-Hurwitz method:

$$\begin{array}{c|ccc} s^5 & 1 & 2 & 1 \\ s^4 & 2 & 4 & 2 \\ s^3 & \varepsilon & 0 & 0 \\ s^2 & c & 2 & 0 \\ s^1 & d & 0 & 0 \\ s^0 & 2 & 0 & 0 \end{array}$$

Where $\varepsilon = -\frac{1}{2}\begin{vmatrix} 1 & 2 \\ 2 & 4 \end{vmatrix} = 0$, $c = -\frac{1}{\varepsilon}\begin{vmatrix} 2 & 4 \\ \varepsilon & 0 \end{vmatrix} = 4$, $d = -\frac{1}{c}\begin{vmatrix} \varepsilon & 0 \\ c & 2 \end{vmatrix} = -\frac{2\varepsilon}{c} = 0$

Because $\varepsilon = d = 0$, there are two elements in the first column are zeroes. Therefore, it is not stable.

$$T(s) = \frac{1}{(s+2)(s+j)(s-j)(s+j)(s-j)}$$

There are two poles in the right-half plane, they are two pole: $(0,\ j)$ and $(0, -j)$.

In conclusion, the two changes in sign in the first column indicate the presence of two roots in the right-hand plane, and the system is unstable.

(b) Compute the poles of $T(s)$ and then verify the result in part(a).

Code:

```
GA = tf(1, [1 2 2 4 1 2]);
pole(GA)
```

Result:

```
ans =

  -2.0000 + 0.0000i
  -0.0000 + 1.0000i
  -0.0000 - 1.0000i
   0.0000 + 1.0000i
   0.0000 - 1.0000i
```

From this simulated result, we can rewrite as

$$\mathrm{T}(s) = \frac{1}{(s+2)(s+j)(s-j)(s+j)(s-j)}$$

We can see the simulated T(s) is the same as the result in part (a).

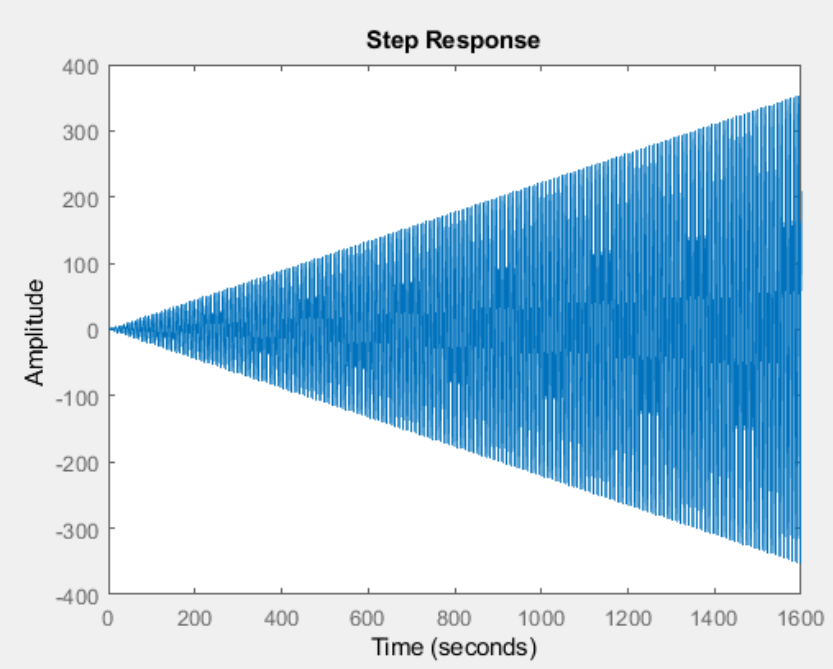
Therefore, it is correct.

(c) Plot the unit step response, and discuss the results.

Code:

```
GA = tf(1, [1 2 2 4 1 2]);

step(GA)
```

Result:



From the result, we could observe that amplitude is continuously increasing, which means it does not have bounded output. Therefore, the system is unstable, which is the same as the theory in part (a).

Problem 6

A control system is shown in Fig.P6. Sketch the root locus, and select K so that the step response of the system has an overshoot of less than 10 % and the settling time (with 2% criterion) is less than 4 seconds.

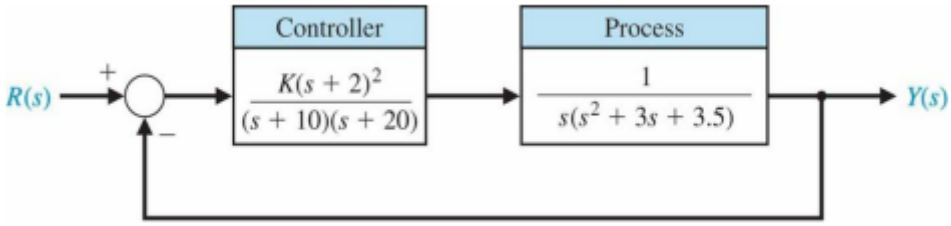
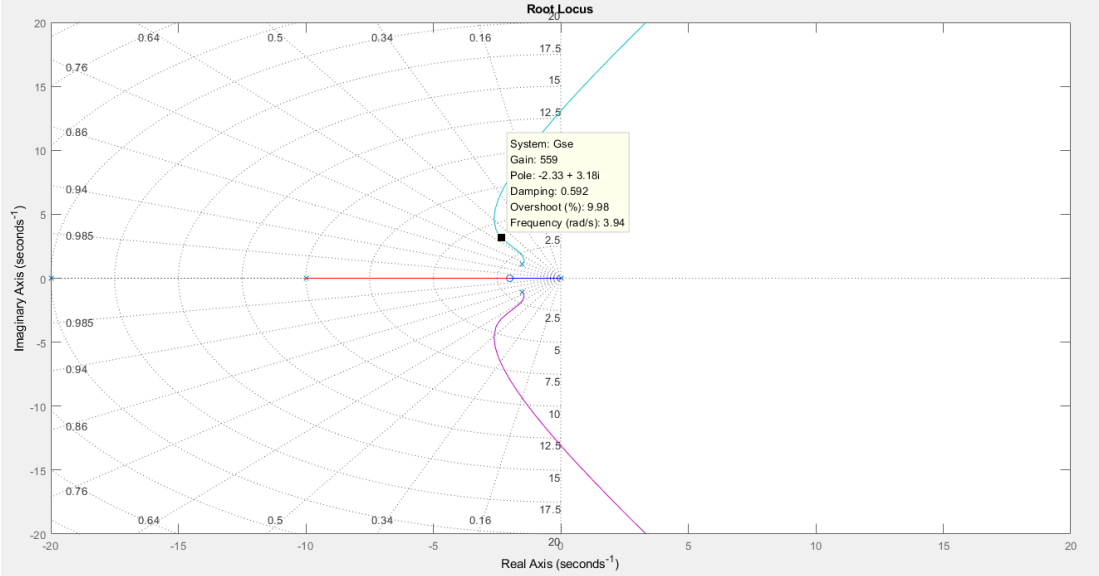


Fig.P6

Code for sketching the root locus and find K

```
GC = tf([1 4 4],[1 30 200]);
GS = tf(1,[1 3 3.5 0]);
Gse = series(GC,GS);
rlocus(Gse)
grid on
xlim([-20,20])
ylim([-20,20])
```

Result:



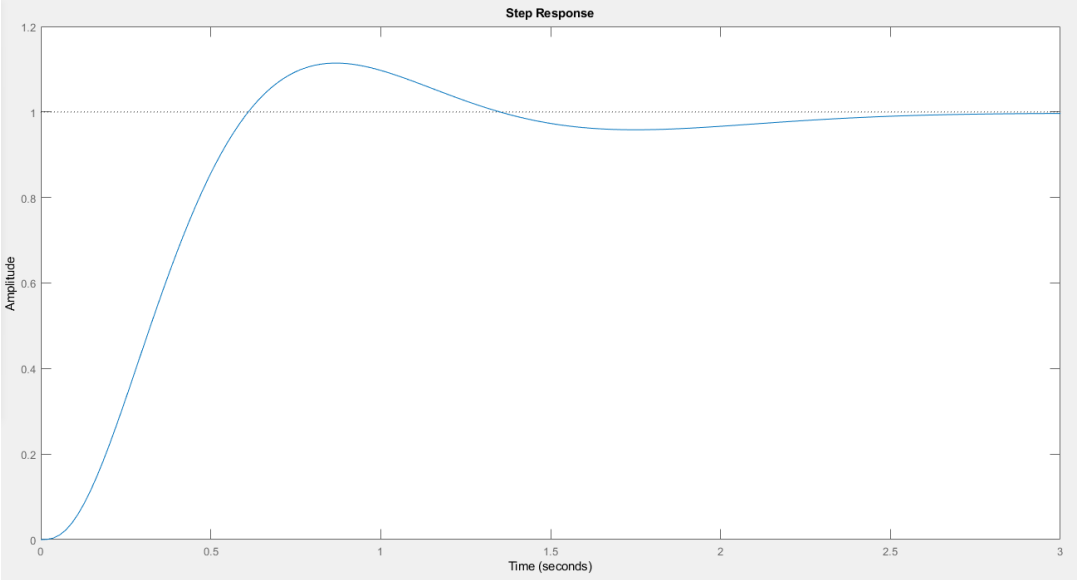
So K is selected as 599

Code for testing K so that the step response of the system has settling time (with 2% criterion) is less than 4 seconds

Code:

```
GC = tf([1 4 4],[1 30 200]);
GS = tf(1,[1 3 3.5 0]);
Gse = series(GC,GS);

sys=feedback(series(559,Gse),1);
step(sys)
stepinfo(sys)
```

Result:

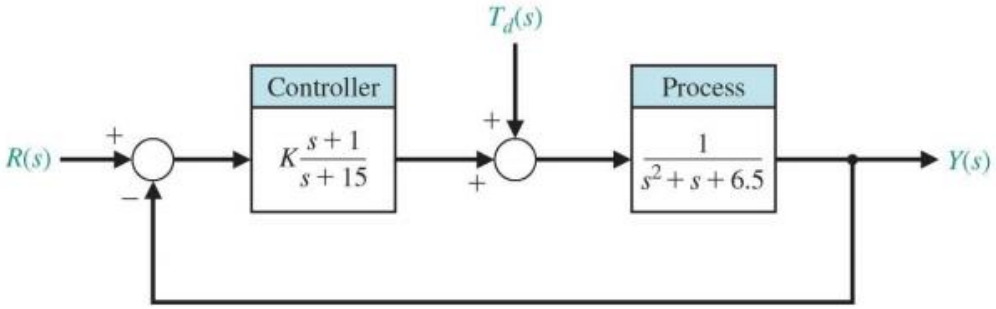


struct with fields:

```
     RiseTime: 0.3907
  SettlingTime: 2.2464
   SettlingMin: 0.9048
   SettlingMax: 1.1147
     Overshoot: 11.4730
    Undershoot: 0
          Peak: 1.1147
      PeakTime: 0.8608
```

Problem 7

Consider the feedback system:



(a) Determine the closed-loop transfer function $T(s) = Y(s)/U(s)$.

$$\mathrm{T(s)} = \frac{Y(s)}{R(s)} = \frac{G_C G}{1 + G_C G} = \frac{K\frac{s+1}{s+15} \; X \frac{1}{s^2+s+6.5}}{1 + K\frac{s+1}{s+15} \; X \frac{1}{s^2+s+6.5}}$$

$$\mathrm{T(s)} = \frac{K(s+1)}{s^3 + 16s^2 + (21.5+K)s + (97.5+K)}$$

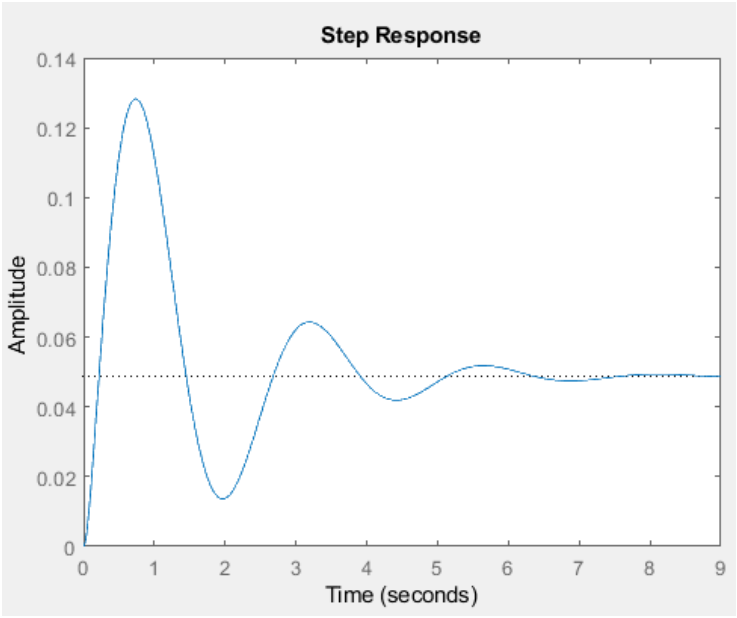
(b) Plot the response of the closed-loop system for K = 5, 10, and 50 to unit step input.

When K=5

Code:

```
k = 5;
GU = tf([k k],[1 16 21.5+k 97.5+k]);
step(GU);
```

Result:



Transfer function:

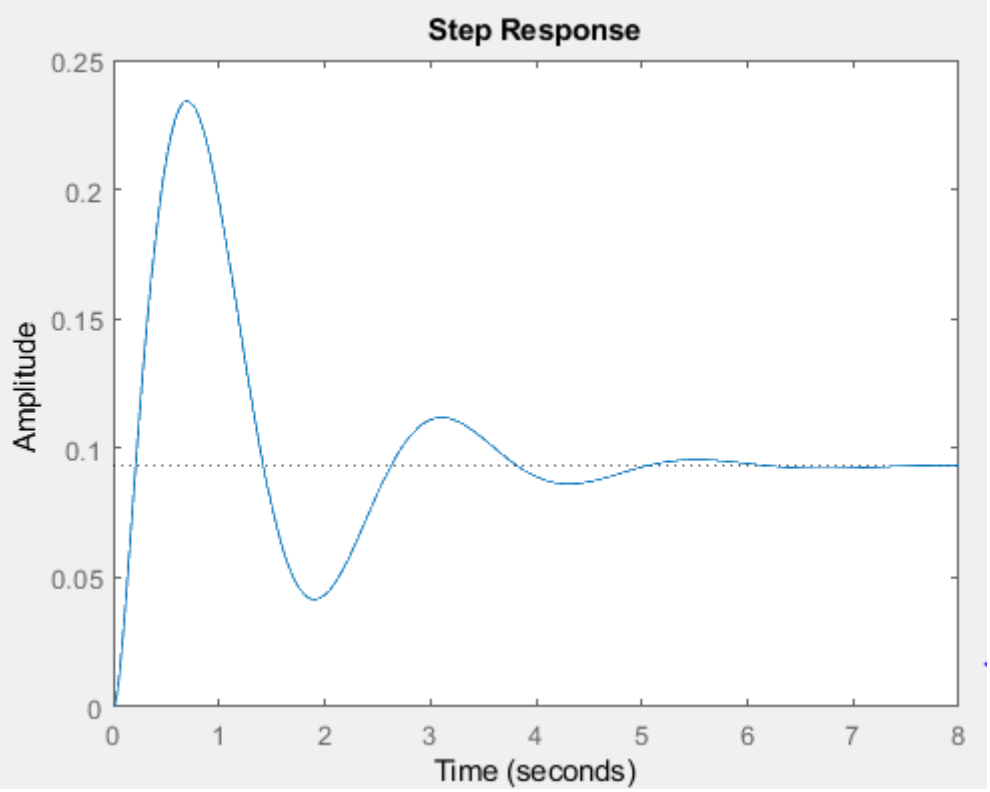
```
GU =

             5 s + 5
  ------------------------------
  s^3 + 16 s^2 + 26.5 s + 102.5

Continuous-time transfer function.
```

When K=10

Code:

```
1 -    k = 10;
2 -    GU = tf([k k],[1 16 21.5+k 97.5+k]);
3 -    step(GU);
```

Result:



Transfer function:

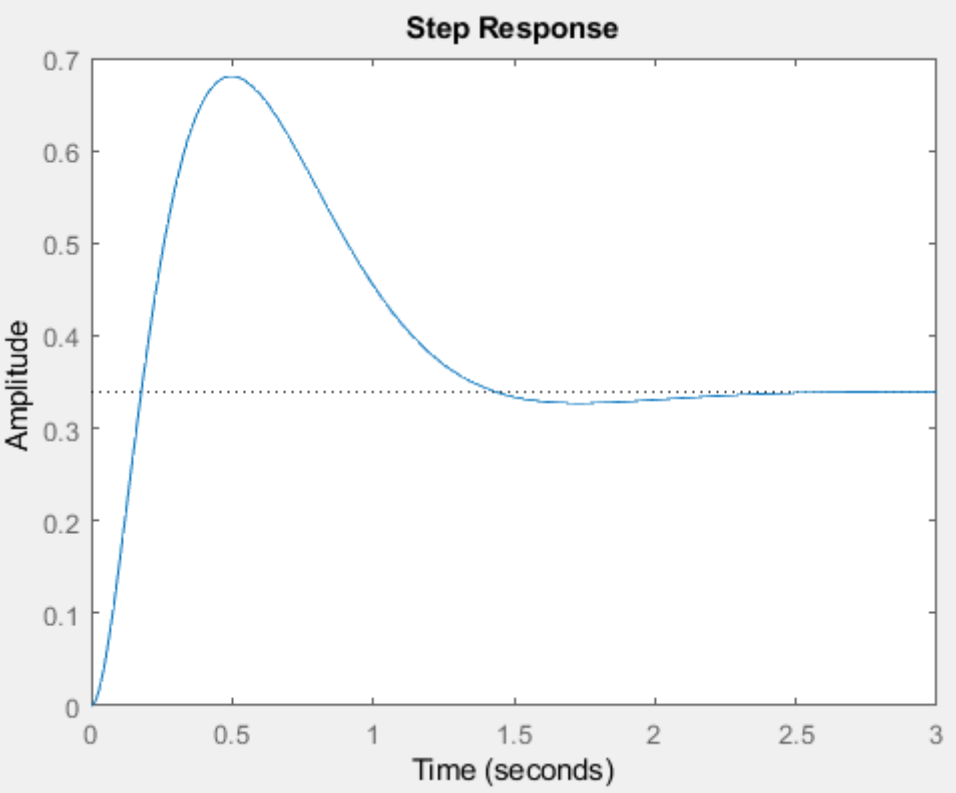
```
GU =

            10 s + 10
  ----------------------------
  s^3 + 16 s^2 + 31.5 s + 107.5

Continuous-time transfer function.
```

When K=50

Code:

```
1 -    k = 50;
2 -    GU = tf([k k],[1 16 21.5+k 97.5+k]);
3 -    step(GU);
```

Result:



Transfer function:

```
GU =

                50 s + 50
  ----------------------------
  s^3 + 16 s^2 + 71.5 s + 147.5

Continuous-time transfer function.
```

(c) **When the controller gain is K = 10, determine the steady-state value of $y(t)$ when the disturbance is a unit step, that is, when** $\mathrm{T}_d = \frac{1}{s}$ **and $R(s)$ = 0.**

$$E_a(s) = R(s) - Y(s)$$

$$Y(s) = (E_a(s) + \mathrm{T}_d(s))\mathrm{G}$$

$$Y(s) = \frac{G_c(s)G(s)}{1 + G_c(s)G(s)} R(s) + \frac{G(s)}{1 + G_c(s)G(s)} \mathrm{T}_d(s)$$

So

$$Y(s) = \frac{10(s+1)}{s^3 + 16s^2 + 31.5s + 107.5} R(s) + \frac{s + 15}{s^3 + 16s^2 + 31.5s + 107.5} \mathrm{T}_d(s)$$

Where $\mathrm{T}_d = \frac{1}{s}$ and $R(s) = 0$.

$$\lim_{t \to \infty} y(t) = \lim_{s \to 0} sY(s) = \frac{15}{107.5} = \frac{6}{43} \approx 0.14$$

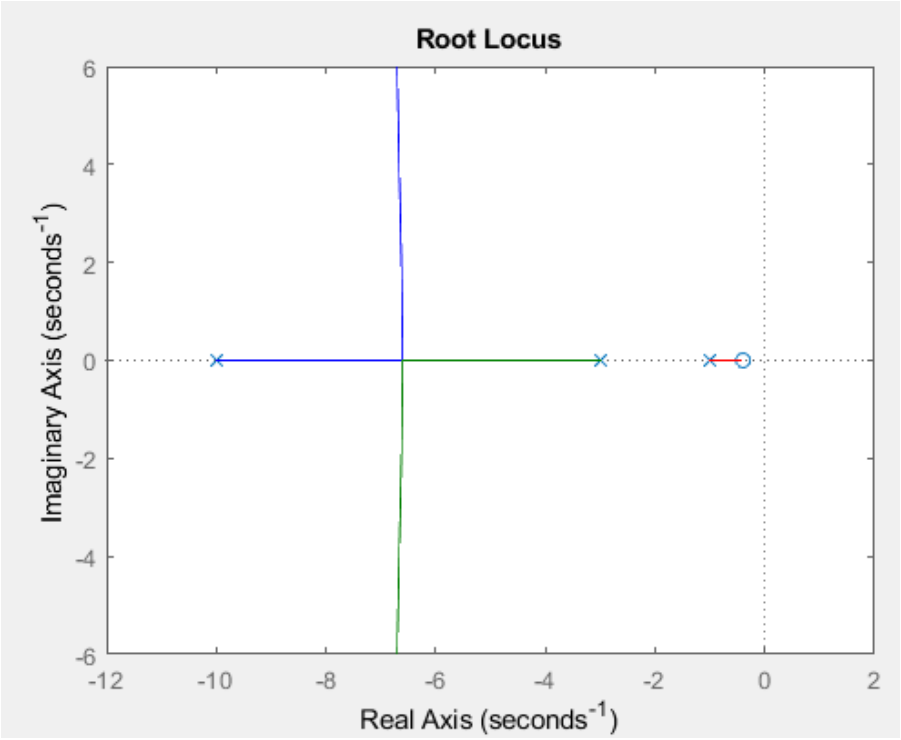
Problem 8

(a) $\boldsymbol{G(s) = \frac{30s+12}{s^3+14s^2+43s+30}}$

Code:

```
1 -    GA=tf([30 12],[1 14 43 30]);
2
3 -    rlocus(GA);
```

Result:

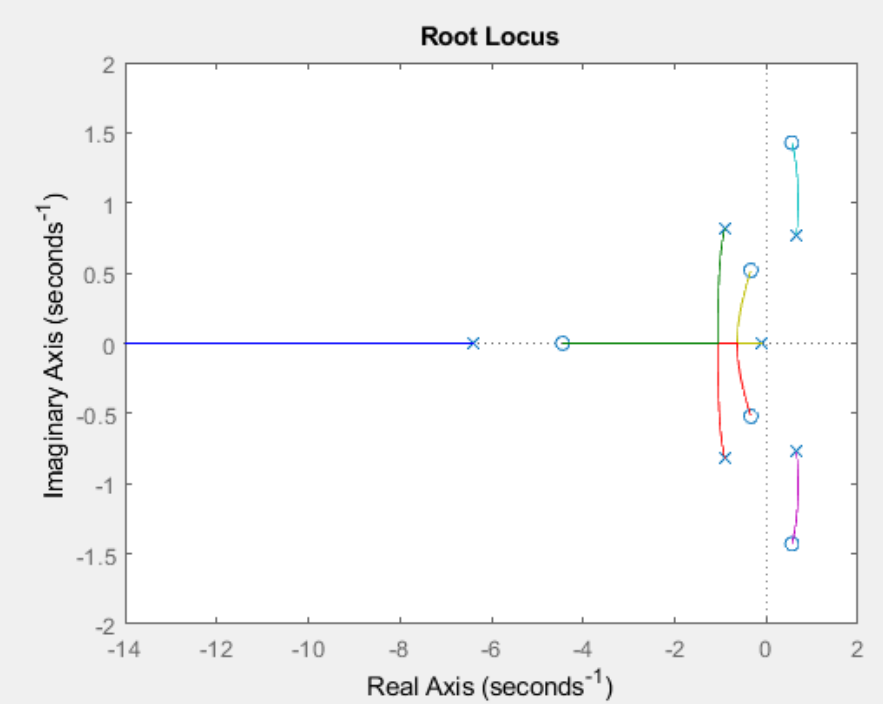


(b) $G(s) = \frac{s^5+4s^4+10s^2+6s+4}{s^6+7s^5+4s^4+s^3+s^2+10s+1}$

Code:

```
2 -    GB=tf([1 4 0 10 6 4],[1 7 4 1 1 10 1]);
3 -    rlocus(GB);
```

Result:



Problem 9

(a) Suppose the controller is a constant gain controller given by $G_C(s) = 2$. Using the lsim function, compute and plot the ramp response for $\theta_d(t) = at$ where a = 0.5 °/s. Determine the attitude error after 10 seconds.

Code:

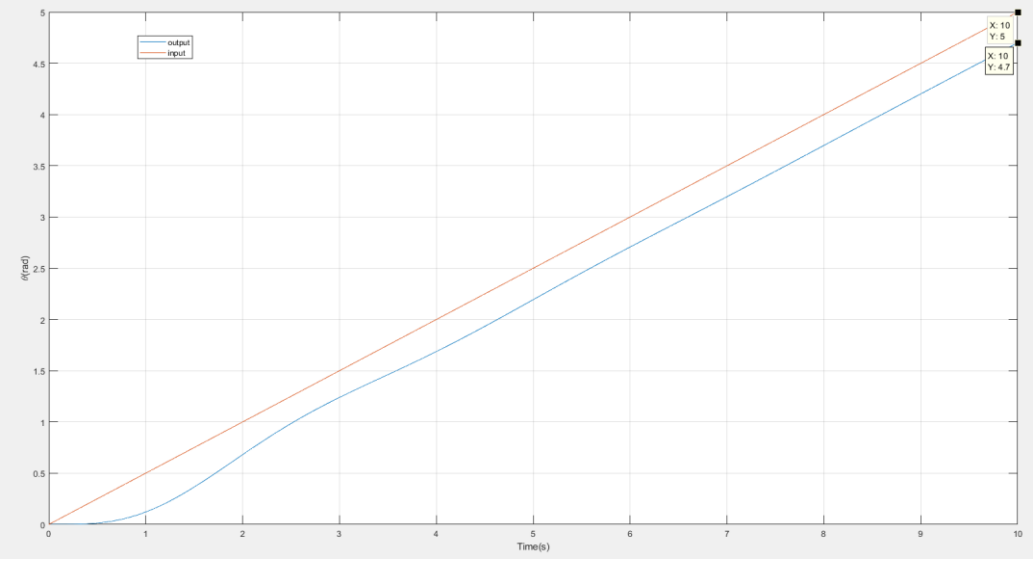
```
GC = tf(2,1);
GE = tf(-10,[1 10]);
GA = tf([-1 -5], [1 3.5 6 0]);

G1 = series(GC,GE);
G2 = series(G1,GA);
sys = feedback(G2,1);

t = 0:0.01:10;
theta = 0.5*t;

[y,T]=lsim(sys,theta,t);
plot(T,y,t,theta),xlabel('Time(s)'),ylabel('\theta(rad)'),grid on;
legend('output','input');
```

Result:

Transfer function:

```
sys =

                20 s + 100
  ---------------------------------
  s^4 + 13.5 s^3 + 41 s^2 + 80 s + 100

Continuous-time transfer function.
```

Ramp response:



Therefore, the attitude error after 10 seconds can be calculated from the result figure, which is 5-4.7=0.3°.

(b) If we increase the complexity of the controller, we can reduce the steady-state tracking error. Suppose we replace the constant gain controller with the more sophisticated PI controller

$$\mathbf{G}_C(s) = \mathbf{K}_1 + \frac{\mathbf{K}_2}{s} = 2 + \frac{1}{s}$$

Repeat the simulation in part (a), and compare the steady-state error obtained.

Code:

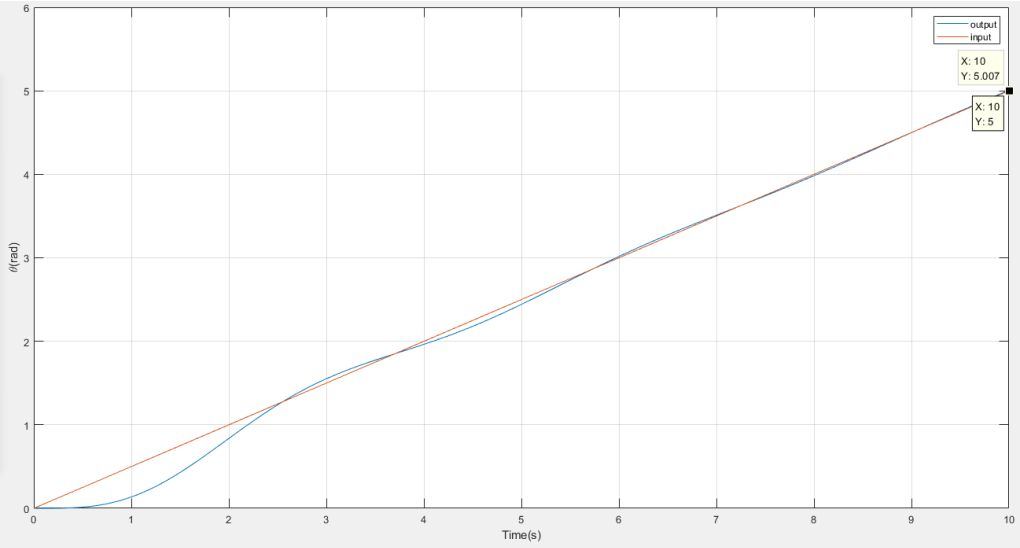
```
GC = tf([2 1],[1 0]);
GE = tf(-10,[1 10]);
GA = tf([-1 -5], [1 3.5 6 0]);

G1 = series(GC,GE);
G2 = series(G1,GA);
sys = feedback(G2,1);

t = 0:0.01:10;
theta = 0.5*t;

[y,T]=lsim(sys,theta,t);
plot(T,y,t,theta),xlabel('Time(s)'),ylabel('\theta(rad)'),grid on
legend('output','input')
```

Result:

Transfer function:

```
sys =

                  20 s^2 + 110 s + 50
  ---------------------------------------------
  s^5 + 13.5 s^4 + 41 s^3 + 80 s^2 + 110 s + 50

Continuous-time transfer function.
```

Ramp response:



From the simulated result, we can calculate the attitude error after 10 seconds is $5^\circ - 5.007^\circ = -0.007^\circ$

In order to calculate the steady-state error, we have

$$\mathrm{E(s)} = \theta_d(s) - \theta(\mathrm{s}) = (1 - \mathrm{T(s)})\theta_d(s)$$

$$\mathrm{E(s)} = \left(1 - \frac{20s^2 + 110s + 50}{s^5 + 13.5s^4 + 41s^3 + 80s^2 + 110s + 50}\right) x \frac{0.5}{s^2}$$

$$= \left(\frac{s^5 + 13.5s^4 + 41s^3 + 60s^2}{s^5 + 13.5s^4 + 41s^3 + 80s^2 + 110s + 50}\right) x \frac{0.5}{s^2}$$

Therefore, the steady-state error is

$$\lim_{t\to\infty} e(t) = \lim_{s\to 0} sE(s) = \frac{0.5}{s} \times \left(\frac{s^5 + 13.5s^4 + 41s^3 + 60s^2}{s^5 + 13.5s^4 + 41s^3 + 80s^2 + 110s + 50}\right)$$

$$= \lim_{s\to 0} 0.5 \times \left(\frac{s^4 + 13.5s^3 + 41s^2 + 60s}{s^5 + 13.5s^4 + 41s^3 + 80s^2 + 110s + 50}\right) = 0$$

Finally, the calculated steady-state error is 0. Comparing this result with the simulated one, they are approximately equal.

Problem 10

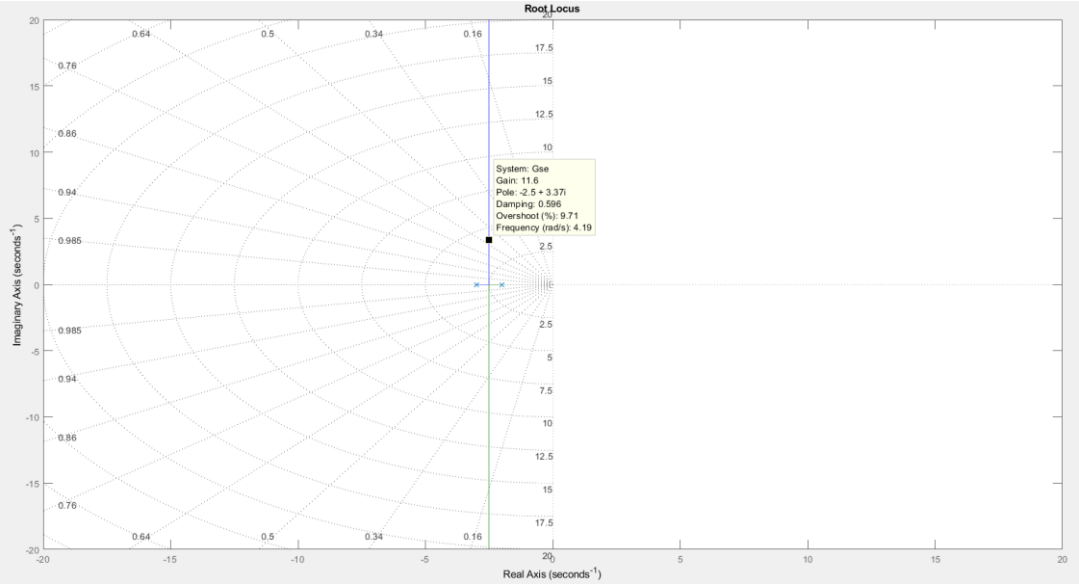
(a) For the proportional controller, develop and m-file to sketch the root locus for $0 \leq K < \infty$, and determine the value of K so that the design specifications are satisfied.

Code:

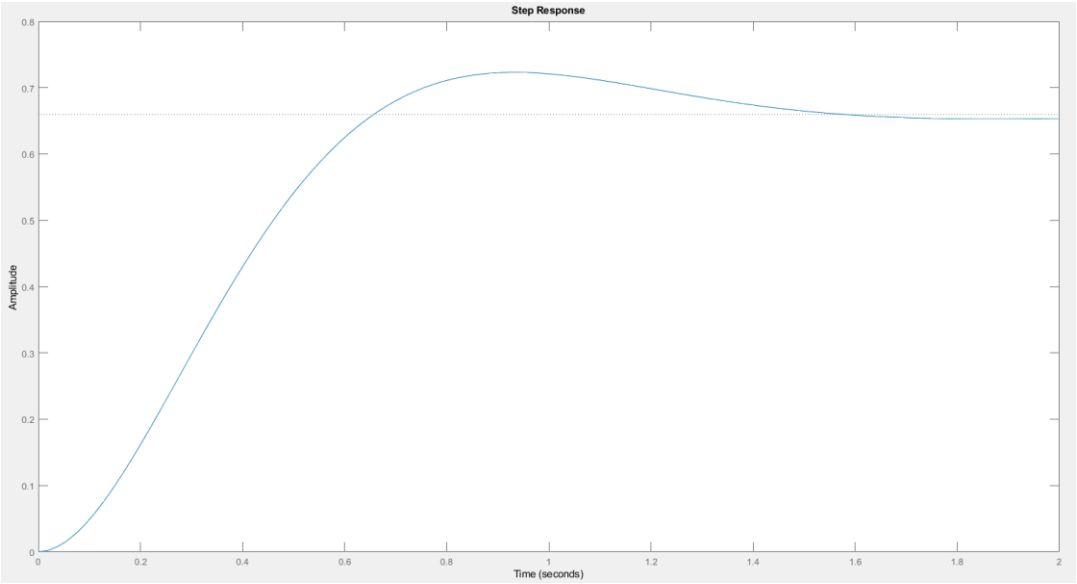
```
GC = tf(1,1);
GS = tf(1,[1 5 6]);
Gse = series(GC,GS);
rlocus(Gse);
grid on;
xlim([-20,20]);
ylim([-20,20]);
```

```
GC = tf(1,1);
GS = tf(1,[1 5 6]);
Gse = series(GC,GS);
sys=feedback(series(11.6,Gse),1);
step(sys);
stepinfo(sys)
```

Result:



K = 11.6




```
ans =

  struct with fields:

        RiseTime: 0.4399
    SettlingTime: 1.4148
     SettlingMin: 0.6037
     SettlingMax: 0.7231
       Overshoot: 9.7132
      Undershoot: 0
            Peak: 0.7231
        PeakTime: 0.9395
```

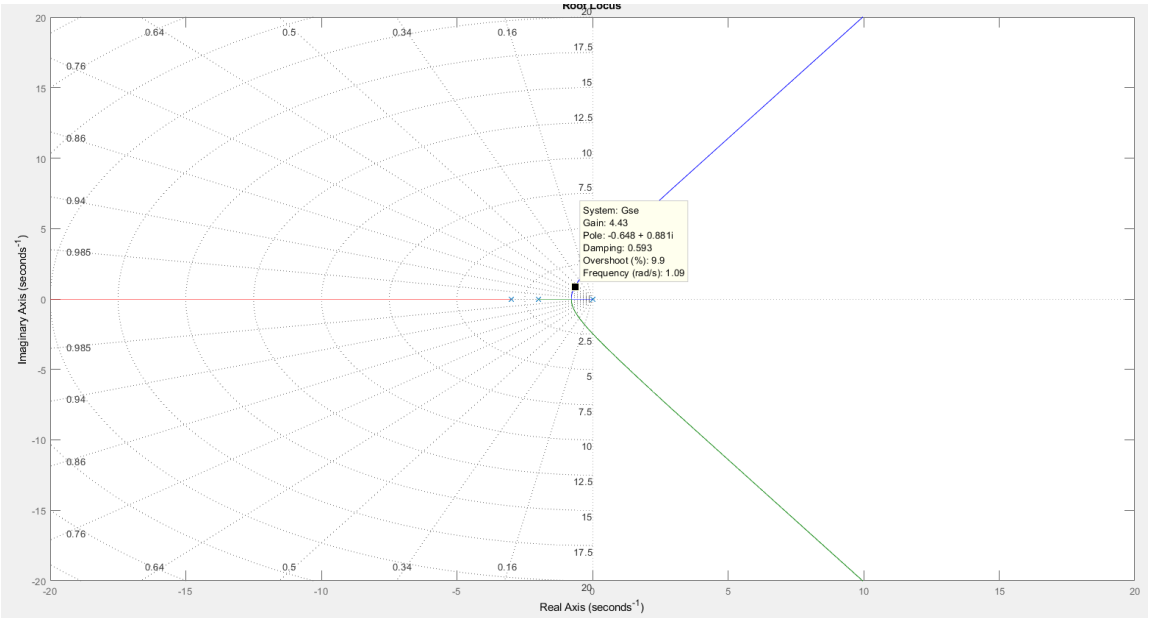
(b) Repeat part (a) for the integral controller

Code:

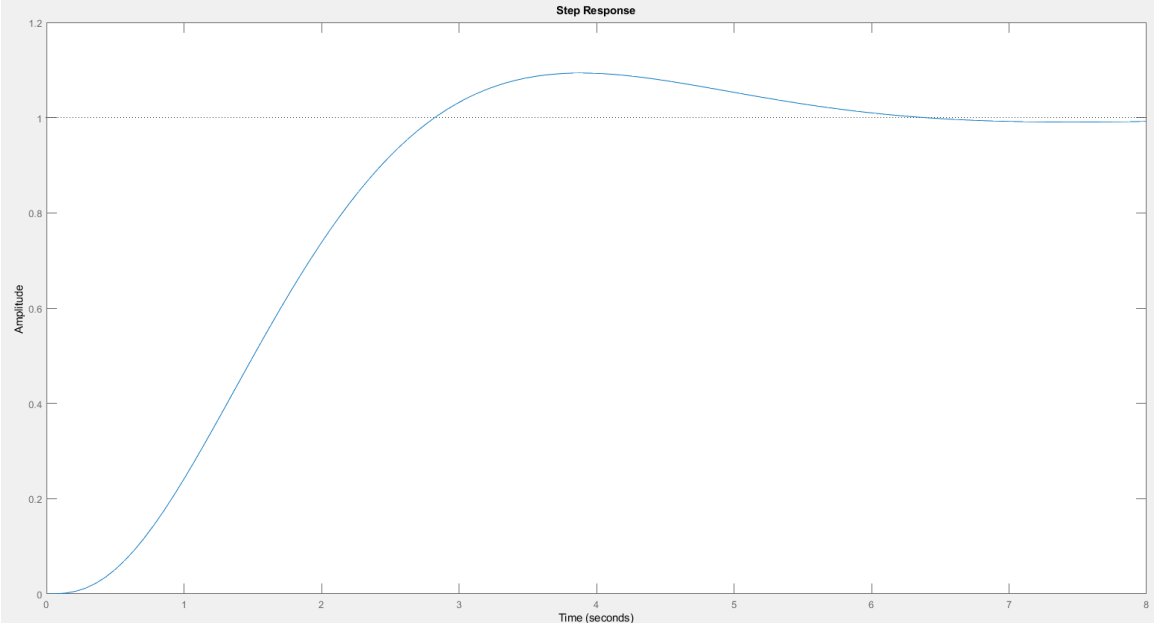
```
GC = tf(1,[1 0]);
GS = tf(1,[1 5 6]);
Gse = series(GC,GS);
rlocus(Gse);
grid on;
xlim([-20,20]);
ylim([-20,20]);
```

```
GC = tf(1,[1 0]);
GS = tf(1,[1 5 6]);
Gse = series(GC,GS);

sys=feedback(series(4.43,Gse),1);
step(sys);
stepinfo(sys)
```

Result:



K=4.43




```
ans =

  struct with fields:

        RiseTime: 1.7733
    SettlingTime: 5.7128
     SettlingMin: 0.9152
     SettlingMax: 1.0937
       Overshoot: 9.3653
      Undershoot: 0
            Peak: 1.0937
        PeakTime: 3.9047
```

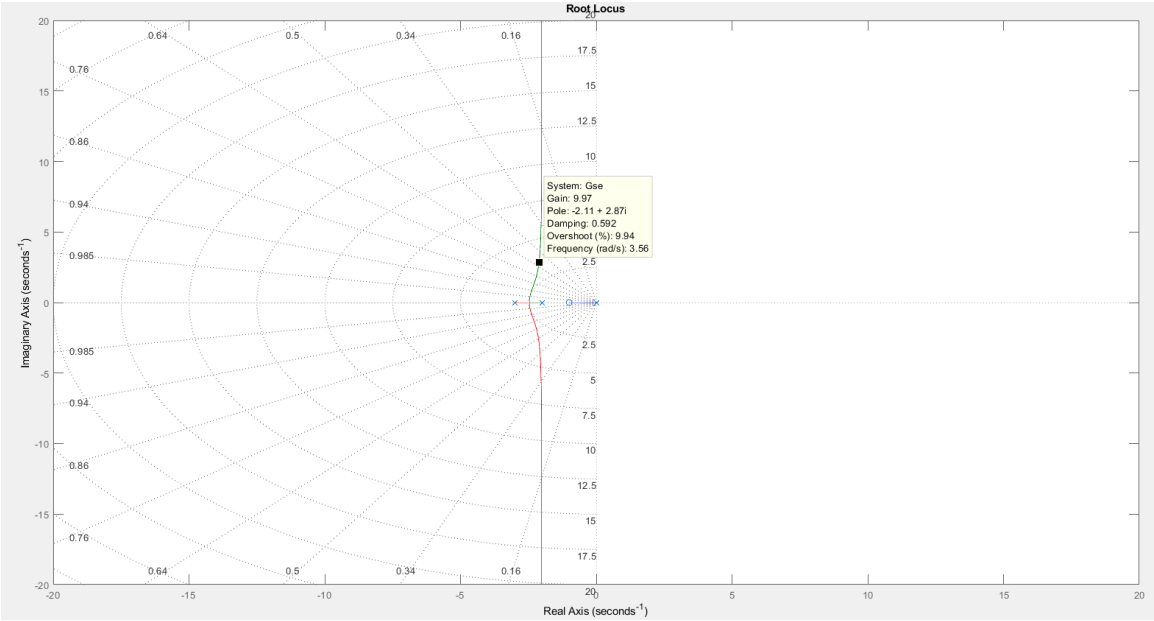
(c) Repeat part (a) for the PI controller

Code:

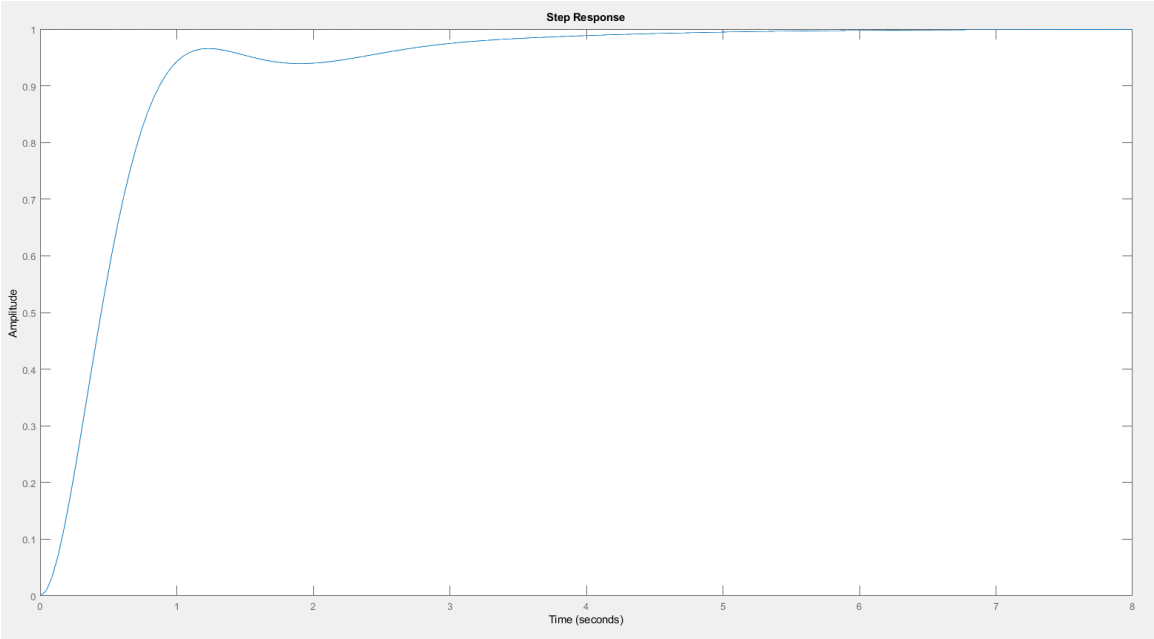
```
GC = tf([1 1],[1 0]);
GS = tf(1,[1 5 6]);
Gse = series(GC,GS);
rlocus(Gse);
grid on;
xlim([-20,20]);
ylim([-20,20]);
```

```
GC = tf([1 1],[1 0]);
GS = tf(1,[1 5 6]);
Gse = series(GC,GS);
sys=feedback(series(9.97,Gse),1);
step(sys);
stepinfo(sys)
```

Result:



K=9.97



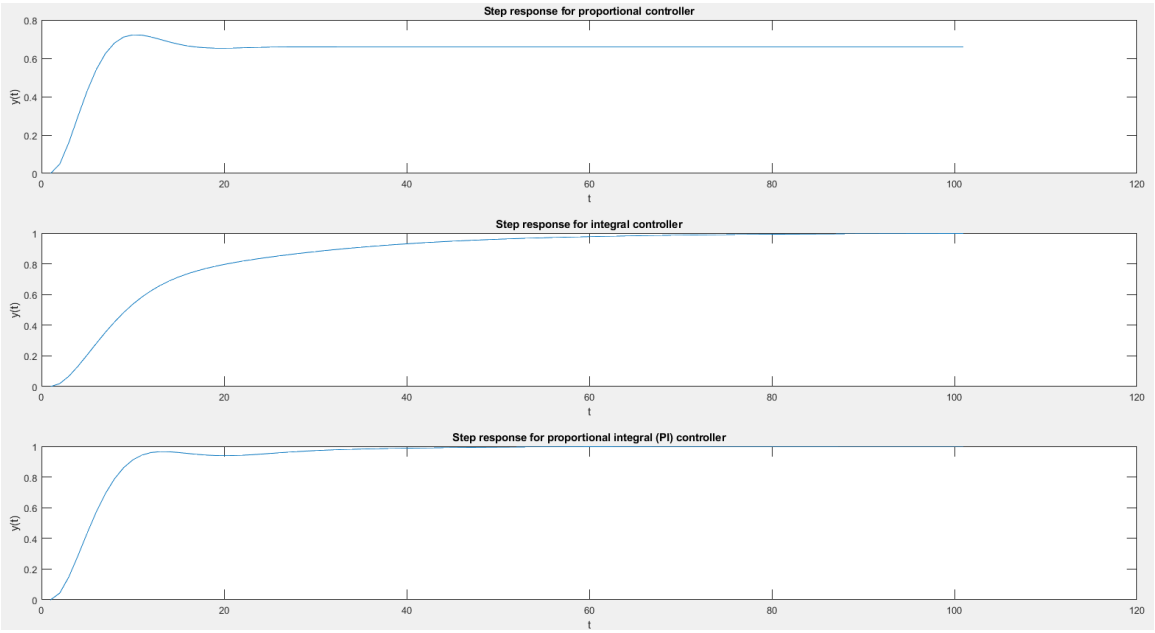

```
ans =

  struct with fields:

        RiseTime: 0.7158
    SettlingTime: 3.2557
     SettlingMin: 0.9007
     SettlingMax: 0.9996
       Overshoot: 0
      Undershoot: 0
            Peak: 0.9996
        PeakTime: 8.2659
```

(d) Compare and contrast the three controllers obtained, concentrating on the steady-state errors and transient performances.

Code:

```
t = 0:0.1:10;

GC1 = tf(1,1);
GS1 = tf(1,[1 5 6]);
Gse1 = series(GC1,GS1);
sys1=feedback(series(11.6,Gse1),1);

GC2 = tf(1,[1 0]);
GS2 = tf(1,[1 5 6]);
Gse2 = series(GC2,GS2);
sys2=feedback(series(4.43,Gse2),1);

GC3 = tf([1 1],[1 0]);
GS3 = tf(1,[1 5 6]);
Gse3 = series(GC3,GS3);
sys3=feedback(series(9.97,Gse3),1);

subplot(3,1,1),plot(step(sys1,t));xlabel('t'),ylabel('y(t)'),title('Step response for proportional controller')
subplot(3,1,2),plot(step(sys2,t));xlabel('t'),ylabel('y(t)'),title('Step response for integral controller')
subplot(3,1,3),plot(step(sys3,t));xlabel('t'),ylabel('y(t)'),title('Step response for proportional integral (PI) controller')
```

Result:



(e) Compare and contrast the three controllers obtained, concentrating on the steady-state errors and transient performances.

$$G_C(s) = K_P + \frac{K_I}{s} + K_D s$$

For (a)

$$G_C(s) = K_P = 11.6,\ K_I = K_D = 0$$

$$\lim_{t\to\infty} e(t) = \lim_{s\to 0} sE(s) = \frac{1}{1+G_C(s)G(s)} = \lim_{s\to 0} \frac{1}{1+11.6 \times \frac{1}{s^2+5s+6}} = 0.341$$

For (b)

$$G_C(\mathrm{s}) = \frac{K_I}{s} = \frac{4.43}{s},\ K_P = K_D = 0$$

$$\lim_{\mathrm{t}\to\infty} e(t) = \lim_{s\to 0} sE(s) = \frac{1}{1+G_C(\mathrm{s})G(s)} = \lim_{s\to 0} \frac{1}{1+\frac{4.43}{s} \times \frac{1}{s^2+5s+6}} = 0$$

For (c)

$$G_C(\mathrm{s}) = K_P + \frac{K_I}{s},\ K_P = K_I = 9.97,\ K_D = 0$$

$$\lim_{\mathrm{t}\to\infty} e(t) = \lim_{s\to 0} sE(s) = \frac{1}{1+G_C(\mathrm{s})G(s)} = \lim_{s\to 0} \frac{1}{1+9.97(1+\frac{1}{s}) \times \frac{1}{s^2+5s+6}} = 0$$

	Proportional controller	Integral controller	Proportional integral (PI)
steady-state errors	0.341	0	0
Overshoot	9.7132	9.3653	0
Settling time	1.4148	5.7128	3.2557

From this Table, we can observe that proportional controller has larger errors than other two's. Comparing integral controller with proportional integral (PI), integral controller has larger overshoot and longer settling time, which means proportional integral is more stable and has shorter settling time to reach the stable value. Therefore, Proportional integral (PI) has better transient performances.