ES 155 Homework 7

1. When deriving the transfer function $G_{a\to b}$, assume all other inputs (besides a and b) are 0.

   (a)

$$\begin{aligned} G_{r\to e} &= \frac{E(s)}{R(s)} \\ E(s) &= R(s) - H(s) \\ &= R(s) - P(s)C(s)E(s) \\ &= \frac{1}{1+P(s)C(s)}R(s) \\ G_{r\to e} &= \frac{1}{1+P(s)C(s)} \end{aligned}$$

   (b)

$$\begin{aligned} G_{d\to e} &= \frac{E(s)}{D(s)} \\ E(s) &= -Y(s) = -P(s)\big(D(s) + C(s)E(s)\big) \\ &= \frac{P(s)C(s)}{1+P(s)C(s)}D(s) \\ G_{d\to e} &= \frac{P(s)C(s)}{1+P(s)C(s)} \end{aligned}$$

   (c)

$$\begin{aligned} G_{n\to e} &= \frac{E(s)}{N(s)} \\ E(s) &= -Y(s) = -\big(N(s) + P(s)C(s)E(s)\big) \\ &= \frac{-N(s)}{1+P(s)C(s)} \\ G_{n\to e} &= -\frac{1}{1+P(s)C(s)} \end{aligned}$$

   (d)

$$\begin{aligned} G_{r\to y} &= \frac{Y(s)}{R(s)} \\ Y(s) &= P(s)C(s)E(s) = P(s)C(s)\big(R(s) - Y(s)\big) \\ &= \frac{P(s)C(s)}{1+P(s)C(s)}R(s) \\ G_{r\to y} &= \frac{P(s)C(s)}{1+P(s)C(s)} \end{aligned}$$

(e)

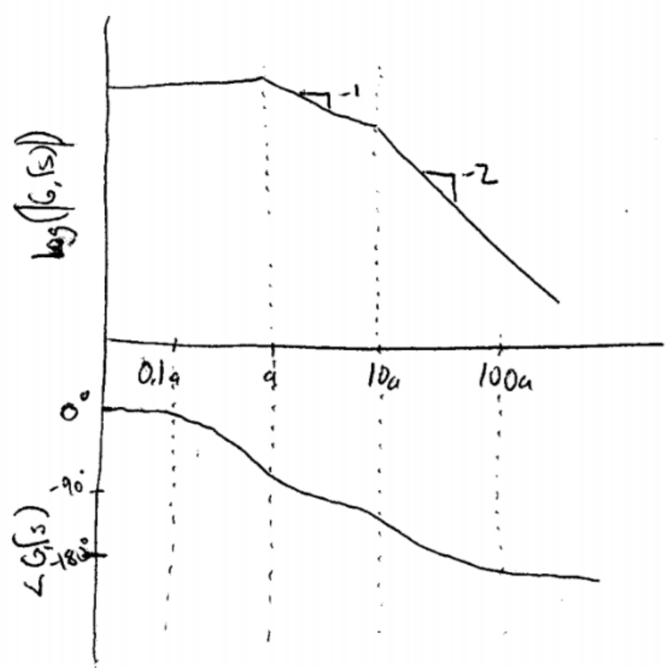
$$
\begin{aligned}
G_{d\to y} &= \frac{Y(s)}{D(s)} \\
Y(s) &= P(s)\big(D(s) + C(s)E(s)\big) \\
&= P(s)\big(D(s) - C(s)Y(s)\big) \\
&= \frac{P(s)}{1+P(s)C(s)}D(s) \\
G_{d\to y} &= \frac{P(s)}{1+P(s)C(s)}
\end{aligned}
$$

(f)

$$
\begin{aligned}
G_{n\to y} &= \frac{Y(s)}{N(s)} \\
Y(s) &= N(s) + P(s)C(s)(-Y(s)) \\
Y(s) &= \frac{1}{1+P(s)C(s)}N(s) \\
G_{n\to y} &= \frac{1}{1+P(s)C(s)}
\end{aligned}
$$

2. (a)

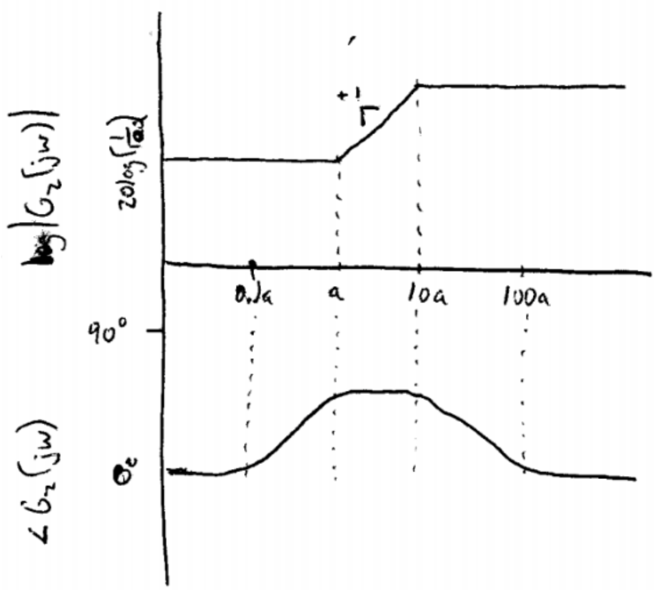
$$G_1(s) = \frac{1}{(s+a)(s+10a)}$$



(b)

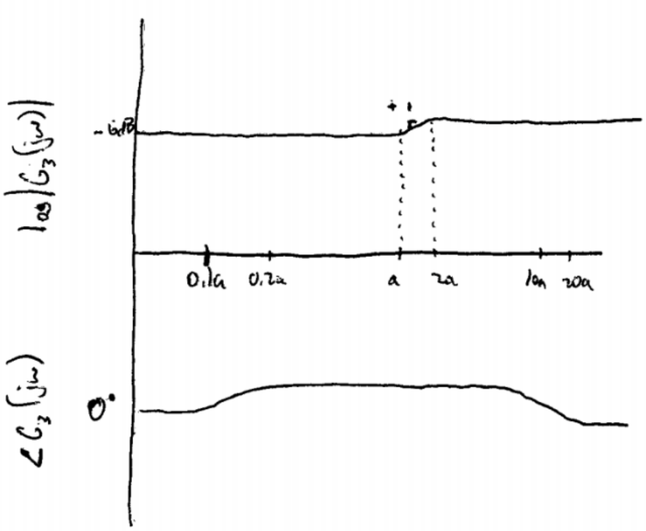
$$G_2(s) = \frac{1 + s/a}{s + 10a} = \left(\frac{1}{a}\right)\frac{s+a}{s+10a}$$

The DC gain is $20\log\frac{1}{10a}$.



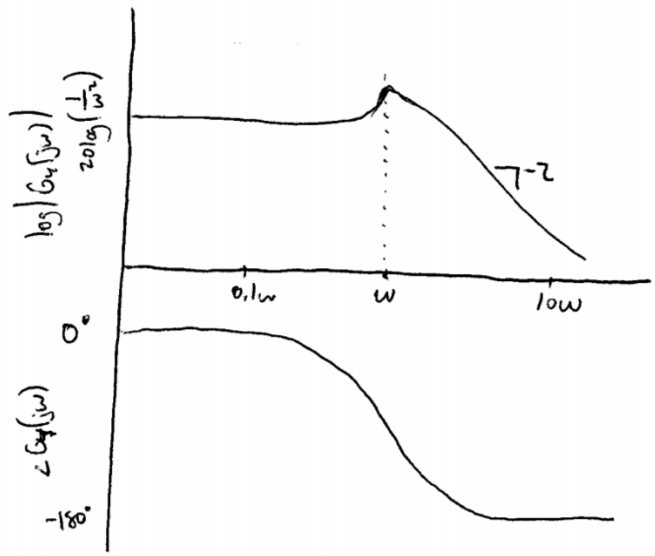
(c)

$$G_3(s) = \frac{s+a}{s+2a}$$



(d)

$$G_4(s) = \frac{1}{s^2 + 2\zeta\omega s + \omega^2}$$



3. The open loop transfer function is $L(s) = P(s)C(s)$. This transfer function can be used directly in MATLAB to plot the Bode and Nyquist plots. See attached MATLAB code for the Bode and Nyquist plots, as well as the computation of the gain and phase margins. Figure 1 shows the Bode and Nyquist plots from MATLAB for the disk drive transfer function $P(s)C(s)$ where

$$P(s) = \frac{1}{s^3 + 10s^2 + 3s + 10} \quad \text{and} \quad C(s) = 1000\frac{s+1}{s+10}$$

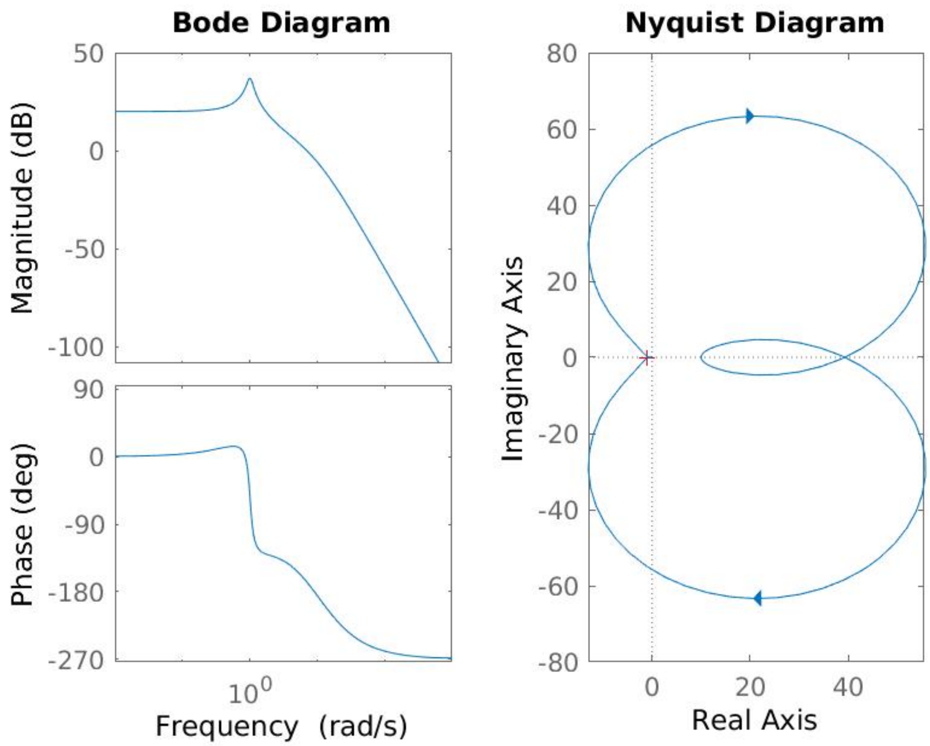


Figure 1: Disk drive transfer function. The gain margin is 1.6047 and the phase margin is 12.9616.

Figure 2 shows the Bode and Nyquist plots from MATLAB for the second order PD transfer function $P(s)C(s)$ where

$$P(s) = \frac{100}{(100s+1)(s+1)} \quad \text{and} \quad C = s + 10$$

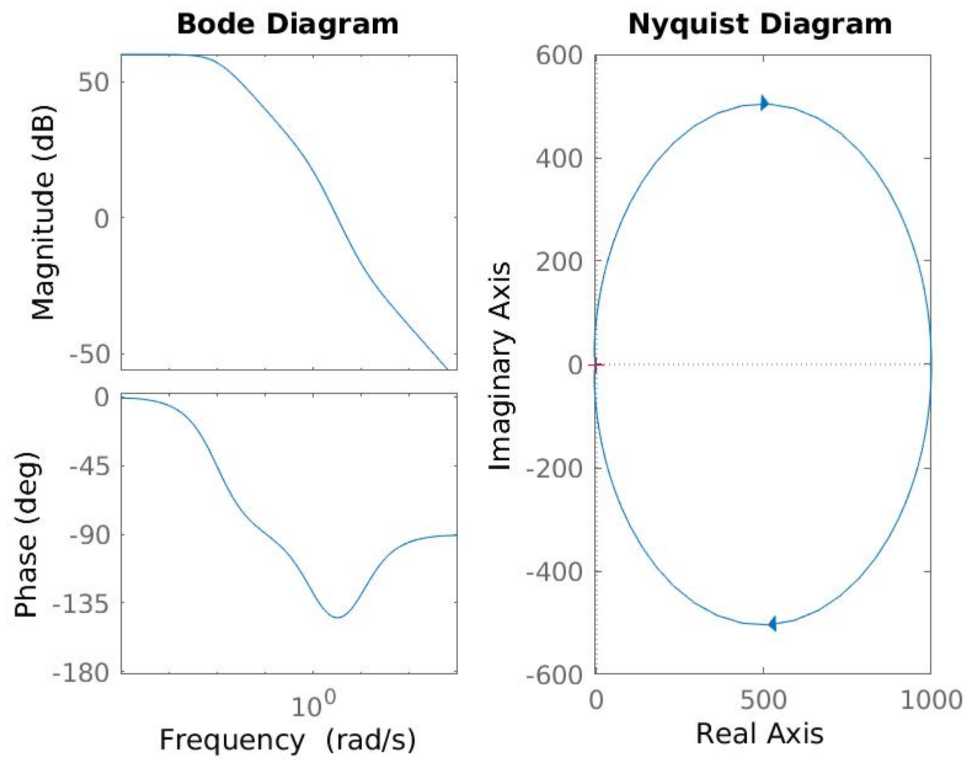


Figure 2: PD transfer function. The gain margin reported by MATLAB is ∞ while the phase margin is 35.2780. This might not be an error, as the system has two poles and one zero, which means the phase at high frequencies is -90dB.

4. The dynamics for the system are

$$P(s) = \frac{Tba/m}{(s+a)(s+c/m)} = \frac{Tba}{(s+a)(ms+c)} = \frac{Tba}{ms^2+(am+c)s+ac}$$

This plant transfer function is modelled in MATLAB.

(a) The proportional integral controller

$$C(s) = k_p + \frac{k_i}{s} = \frac{k_p s + k_i}{s}$$

can also be modelled in MATLAB. The phase and gain margins, rise time, and overshoot are computed via `stepinfo`() and plotted in the table.

k_p	k_i	Stable	g_m	φ_m	SS_{err}	T_r	M_p
0.5000	0.1000	1	∞	118.1865	0.0950	24.2034	0
0.0500	1.0000	1	∞	54.7945	0.0070	1.6819	14.2429
0.0500	0.0010	1	∞	∞	0.9095	149.6098	0
0.0050	0.0010	1	∞	∞	0.9091	200.4907	0

(b) Figure 3 shows the zero-pole diagrams and step responses for the closed loop systems.

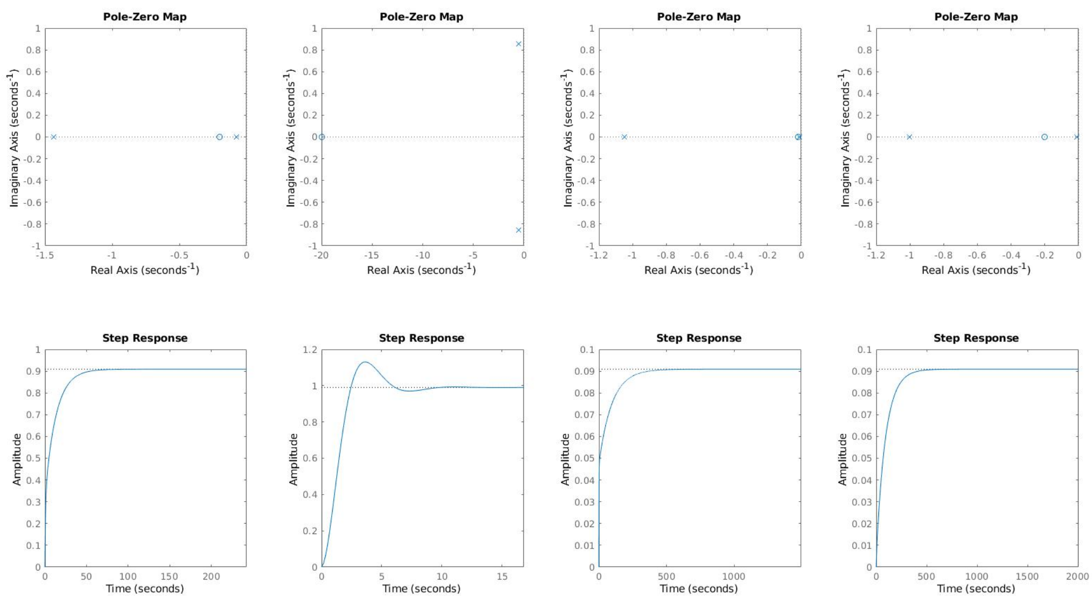


Figure 3: Top: Pole-Zero diagrams. Bottom: Step Responses. From left to right, the k_p and k_i parameters are (0.5, 0.1), (0.05, 1), (0.05, 0.001), and (0.005, 0.001).

(c) The table shows that all of the closed loop systems are stable, as evidenced by the Nyquist plots in Figure 4, shows the corresponding Bode and Nyquist plots for the same closed loop systems. None of the plots encircles (-1,0), so the systems are stable. This is reflected in the step responses shown in Figure 3. Their gain margins are all ∞, as any amount of gain applied will never result in the Nyquist plot encircling (-1, 0), as none of them cross the real axis left of the origin. The error was relatively low for the first and second systems, which had relatively stronger parameters than the latter ones. Once the controllers' strength decreased too much, they were unable to correct the system, and it sat around 1/10 the desired value. For these later systems, the rise time was very long as well, because of these weak controls. The second system had a very strong integral term which resulted in a damped oscillatory response, allowing it to have a much shorter rise time than the first system. However, this also resulted in overshoot, which none of the other systems exhibit.

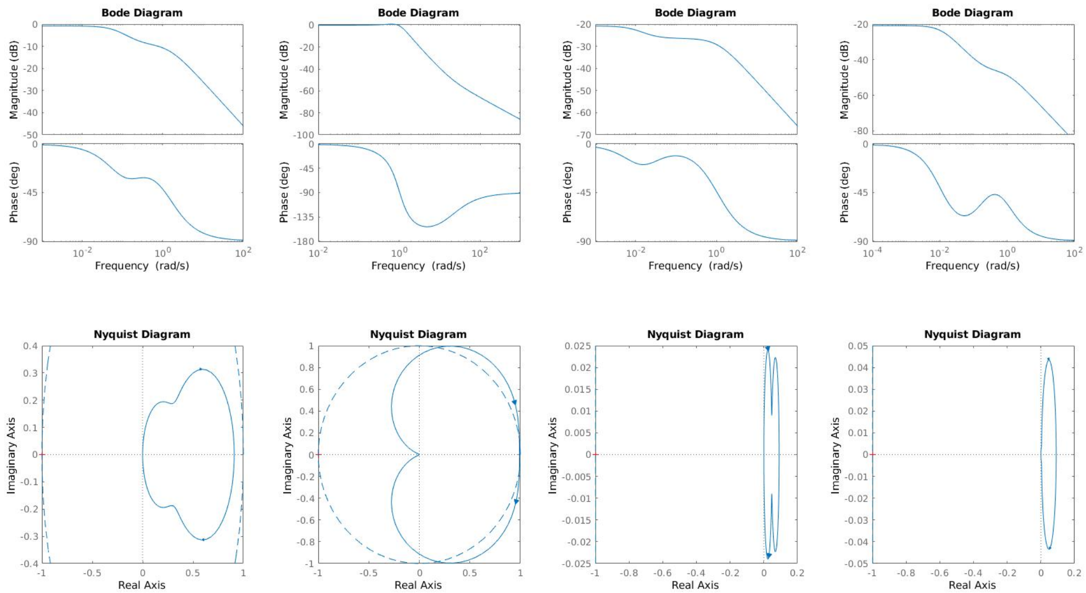


Figure 4: Top: Bode diagrams. Bottom: Nyquist plots. From left to right, the k_p and k_i parameters are (0.5, 0.1), (0.05, 1), (0.05, 0.001), and (0.005, 0.001).

ES155 P7

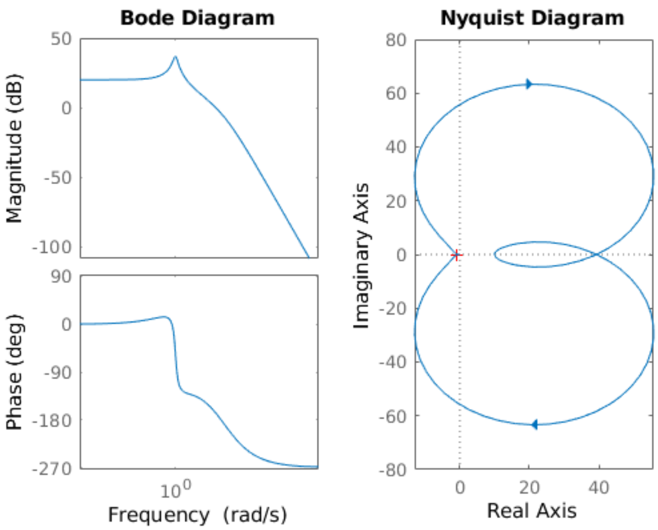
Contents

Problem 3

3.a

```
P = tf(1, [1 10 3 10])
S = 1000 * tf([1 1], [1 10])

L = P*S

figure(1); clf;
subplot(1,2,1)
bode(L)
subplot(1,2,2)
nyquist(L)

saveas(gcf, "ES155P7_3a.jpg")

pole(L)
[GainMargin, PhaseMargin, Wcg, Wcp] = margin(L)
```

```
P =

             1
  ----------------------
  s^3 + 10 s^2 + 3 s + 10

Continuous-time transfer function.


S =

  1000 s + 1000
  -------------
     s + 10

Continuous-time transfer function.


L =

             1000 s + 1000
  ----------------------------------
  s^4 + 20 s^3 + 103 s^2 + 40 s + 100

Continuous-time transfer function.


ans =

 -10.0000 + 0.0000i
  -9.7980 + 0.0000i
  -0.1010 + 1.0052i
  -0.1010 - 1.0052i


GainMargin =

    1.6047


PhaseMargin =

   12.9616


Wcg =

    9.0682


Wcp =

    7.0090
```

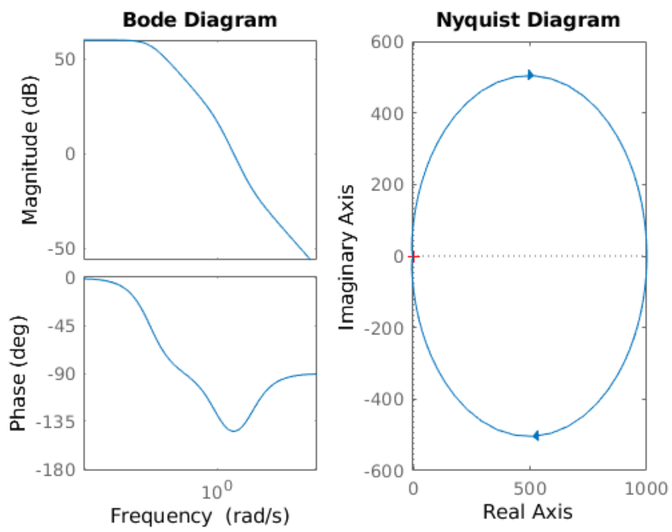


3.b

```
P = tf(100, [100, 101, 1])
S = tf([1 10], 1)

L = P*S

figure(2); clf;
subplot(1,2,1)
bode(L)
subplot(1,2,2)
nyquist(L)

saveas(gcf, "ES155P7_3b.jpg")

pole(L)
[GainMargin, PhaseMargin, Wcg, Wcp] = margin(L)
```

```
P =

          100
  ------------------
  100 s^2 + 101 s + 1

Continuous-time transfer function.


S =

  s + 10

Continuous-time transfer function.


L =

     100 s + 1000
  ------------------
  100 s^2 + 101 s + 1

Continuous-time transfer function.


ans =

   -1.0000
   -0.0100


GainMargin =

   Inf


PhaseMargin =

   35.2780


Wcg =

   NaN
```

```
Wcp =

    3.1623
```



Problem 4

```
% Constants
a = 0.2
b = 25
c = 50
T = 200
m = 1000

P = tf(T*b*a, [m, a*m + c, a*c])
```

```
a =

    0.2000


b =

    25


c =

    50


T =

   200


m =

        1000


P =

          1000
  ---------------------
  1000 s^2 + 250 s + 10

Continuous-time transfer function.
```

4.a

```
table = zeros(4,8);

figure(3); clf;
figure(4); clf;

Kp = [0.5, 0.05, 0.05, 0.005]
```

```
Ki = [0.1, 1, 0.001, 0.001]

for i = 1:4
   figure(3)
   kp = Kp(i)
   ki = Ki(i)

   table(i,1) = kp;
   table(i,2) = ki;

   C = tf([kp, ki], 1)
   G = feedback(P*C, 1)

   L = P*C
   sysL = ss(L)

   sys = ss(G);

   table(i,3) = isstable(sys);

   [GainMargin, PhaseMargin, Wcg, Wcp] = margin(sysL)

   table(i,4) = GainMargin;
   table(i,5) = PhaseMargin;

   [y, t] = step(sys);
   if y(length(t)) == y(length(t-2))
       error_SS = 1 - y(length(t))
   else
       error("Steady State not reached")
   end

   table(i,6) = error_SS;

   S = stepinfo(sys)

   table(i,7) = S.RiseTime;
   table(i,8) = S.Overshoot;

   subplot(2,4,i)
   pzmap(sys)

   subplot(2,4,i+4)
   step(sys, S.RiseTime*10)

   figure(4)
   subplot(2,4,i)
   bode(sys)
   subplot(2,4,i+4)
   nyquist(sys)

   % plot the unit circle on the nyquist plot
   theta = 0:0.1:2*pi;
   x = cos(theta);
   y = sin(theta);
   hold on;
   plot(x,y,'--');
   hold off;
end

table
```

```
Kp =

    0.5000    0.0500    0.0500    0.0050


Ki =

    0.1000    1.0000    0.0010    0.0010


kp =

    0.5000


ki =

    0.1000


C =

  0.5 s + 0.1

Continuous-time transfer function.


G =
```

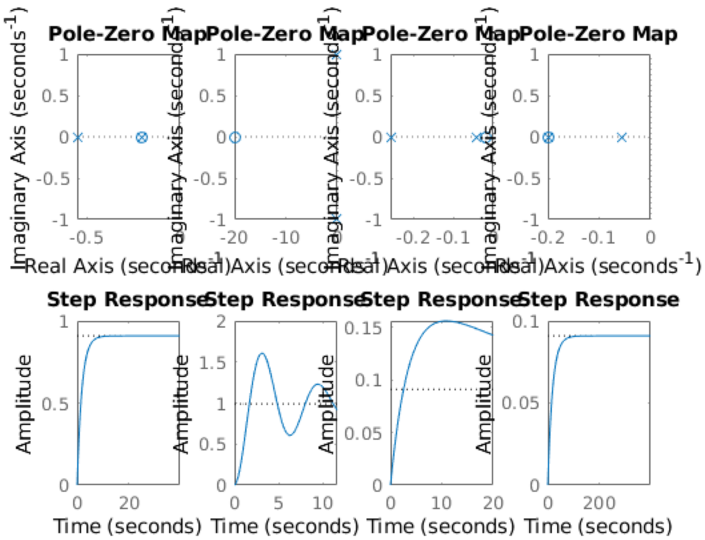
```
      500 s + 100
  ----------------------
  1000 s^2 + 750 s + 110

Continuous-time transfer function.


L =

      500 s + 100
  ---------------------
  1000 s^2 + 250 s + 10

Continuous-time transfer function.


sysL =

  A =
           x1     x2
   x1  -0.25  -0.08
   x2  0.125      0

  B =
       u1
   x1   1
   x2   0

  C =
        x1   x2
   y1  0.5  0.8

  D =
       u1
   y1   0

Continuous-time state-space model.


GainMargin =

   Inf


PhaseMargin =

   95.7406


Wcg =

   NaN


Wcp =

    0.4974


error_SS =

    0.0910


S =

  struct with fields:

        RiseTime: 3.9946
    SettlingTime: 7.1129
     SettlingMin: 0.8223
     SettlingMax: 0.9091
       Overshoot: 0
      Undershoot: 0
            Peak: 0.9091
        PeakTime: 19.1743


kp =

    0.0500


ki =

     1


C =
```

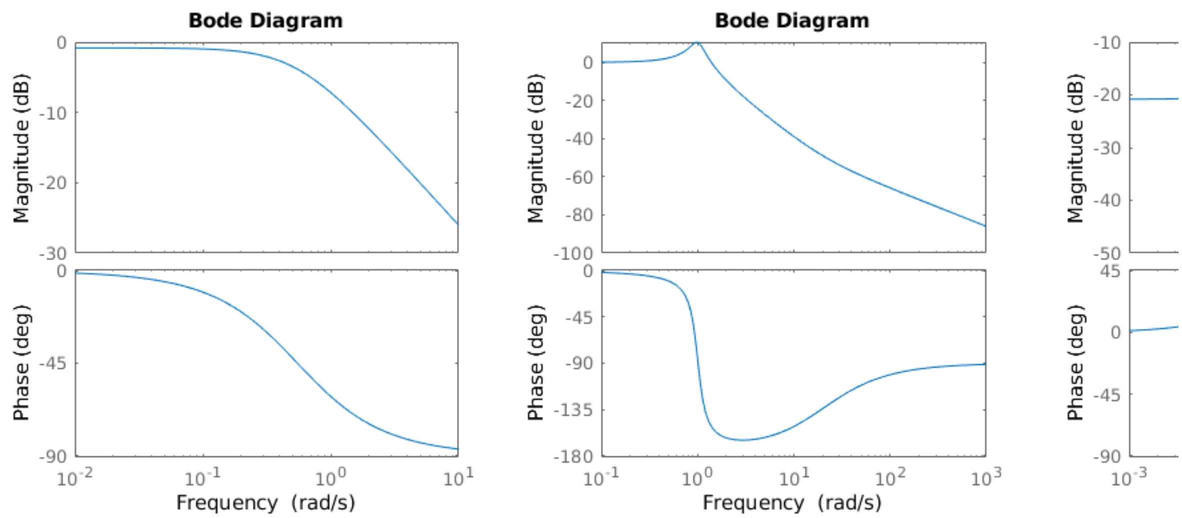
```
  0.05 s + 1

Continuous-time transfer function.


G =

        50 s + 1000
  ----------------------
  1000 s^2 + 300 s + 1010

Continuous-time transfer function.


L =

       50 s + 1000
  ---------------------
  1000 s^2 + 250 s + 10

Continuous-time transfer function.


sysL =

  A =
           x1     x2
   x1  -0.25  -0.08
   x2  0.125      0

  B =
       u1
   x1   4
   x2   0

  C =
            x1      x2
   y1  0.0125       2

  D =
       u1
   y1   0

Continuous-time state-space model.


GainMargin =

   Inf


PhaseMargin =

   17.1460


Wcg =

   NaN


Wcp =

    0.9900


error_SS =

    0.0047


S =

  struct with fields:

        RiseTime: 1.1611
    SettlingTime: 25.7136
     SettlingMin: 0.6095
     SettlingMax: 1.6066
       Overshoot: 62.2638
      Undershoot: 0
            Peak: 1.6066
        PeakTime: 3.0701


kp =

    0.0500


ki =
```

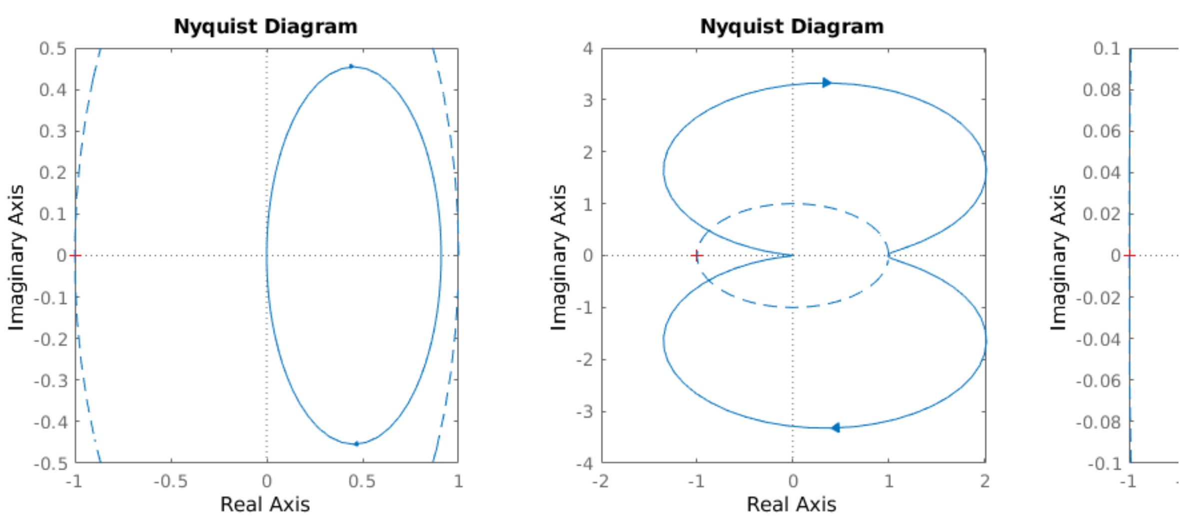
```
   1.0000e-03


C =

  0.05 s + 0.001

Continuous-time transfer function.


G =

        50 s + 1
  ---------------------
  1000 s^2 + 300 s + 11

Continuous-time transfer function.


L =

        50 s + 1
  ---------------------
  1000 s^2 + 250 s + 10

Continuous-time transfer function.


sysL =

  A =
          x1     x2
   x1  -0.25  -0.08
   x2  0.125      0

  B =
         u1
   x1  0.25
   x2     0

  C =
          x1     x2
   y1    0.2  0.032

  D =
       u1
   y1   0

Continuous-time state-space model.


GainMargin =

   Inf


PhaseMargin =

   Inf


Wcg =

   NaN


Wcp =

   NaN


error_SS =

    0.9089


S =

  struct with fields:

        RiseTime: 2.0130
    SettlingTime: 98.7607
     SettlingMin: 0.0896
     SettlingMax: 0.1557
       Overshoot: 71.3097
      Undershoot: 0
            Peak: 0.1557
        PeakTime: 11.0995
```

```
kp =

    0.0050


ki =

   1.0000e-03


C =

  0.005 s + 0.001

Continuous-time transfer function.


G =

         5 s + 1
  ---------------------
  1000 s^2 + 255 s + 11

Continuous-time transfer function.


L =

         5 s + 1
  ---------------------
  1000 s^2 + 250 s + 10

Continuous-time transfer function.


sysL =

  A =
          x1     x2
   x1  -0.25  -0.08
   x2  0.125      0

  B =
          u1
   x1  0.125
   x2      0

  C =
          x1     x2
   y1   0.04  0.064

  D =
       u1
   y1   0

Continuous-time state-space model.


GainMargin =

   Inf


PhaseMargin =

   Inf


Wcg =

   NaN


Wcp =

   NaN


error_SS =

    0.9091


S =

  struct with fields:

        RiseTime: 39.9456
    SettlingTime: 71.1286
     SettlingMin: 0.0822
     SettlingMax: 0.0909
```

```
      Overshoot: 0
     Undershoot: 0
           Peak: 0.0909
       PeakTime: 191.7425


table =

  Columns 1 through 7

    0.5000    0.1000    1.0000       Inf   95.7406    0.0910    3.9946
    0.0500    1.0000    1.0000       Inf   17.1460    0.0047    1.1611
    0.0500    0.0010    1.0000       Inf       Inf    0.9089    2.0130
    0.0050    0.0010    1.0000       Inf       Inf    0.9091   39.9456

  Column 8

         0
   62.2638
   71.3097
         0
```


Pole-Zero Map
Pole-Zero Map
Pole-Zero Map
Pole-Zero Map
Imaginary Axis (seconds^{-1})
Real Axis (seconds^{-1})
Step Response
Step Response
Step Response
Step Response
Amplitude
Time (seconds)


Bode Diagram
Bode Diagram
Magnitude (dB)
Phase (deg)
Frequency (rad/s)


Nyquist Diagram
Nyquist Diagram
Imaginary Axis
Real Axis

Published with MATLAB® R2018a