EE324 Control Systems Lab

Problem Sheet 7

Param Rathour | 190070049

Autumn Semester 2021-22

Contents

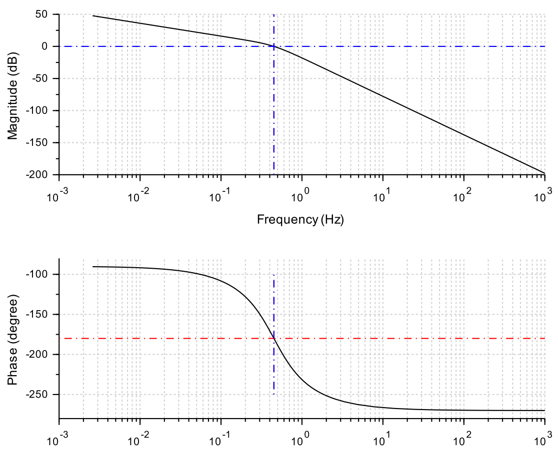
1 Q1

Open Loop Transfer Function is

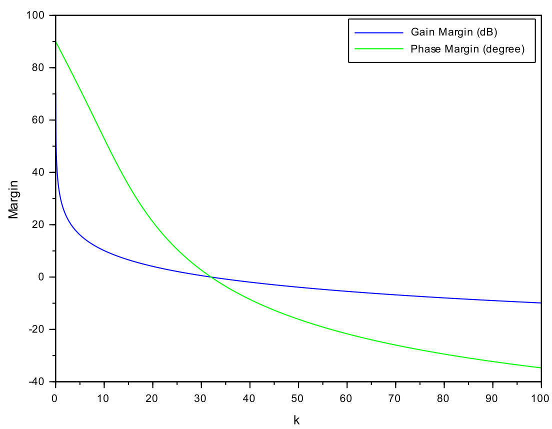
$$\mathrm{G}(s) = \frac{1}{s(s^2 + 4s + 8)}$$

a)

The value of K for which the closed-loop characteristic equation has Gain margin and Phase margin zero is 32.



(a) Bode Plot $G(s)$



(b) Margins vs Gain

Code to obtain K and Figure 1.1a is given below

```
s = %s;
kMax = 100;
kStep = 0.01;
kValues = kStep:kStep:100;
TF = 1/(s*(s^2+4*s+8));
TF = syslin('c', TF);
gainMargin = [];
phaseMargin = [];
for i = 1:length(kValues)
    k = kValues(i);
```

```
    G = k * TF;
    G = syslin('c', G);
    gainMargin($+1) = g_margin(G);
    phaseMargin($+1) = p_margin(G);
    if (abs(gainMargin($)) < 1e-6) and (abs(phaseMargin($)) < 1e-6)
        disp(k);
        show_margins(32 * TF);
        xs2pdf(0,'Q1a');
    end
end
--> ans = 32
```

b)

It is not possible to have gain margin non-zero & phase margin = 0 and vice versa.
As can be seen in Figure 1.1b, both margins are strictly decreasing. So they can be zero only once and that happens when both gain margin and phase margin are zero.
Code to obtain Figure 1.1b is given below

```
plot(kValues', gainMargin, 'b'); xlabel("k"); ylabel("Margin");
plot(kValues', phaseMargin, 'g'); xlabel("k"); ylabel("Margin");
legend(['Gain Margin (dB)';'Phase Margin (degree)']);
ax = gca();
ax.data_bounds=[0 -40;100 100];
xs2pdf(0,'Q1ae');
```

c)

At $K = 32$, the system is marginally stable as seen from the Bode Plot 1.1a.
Alternatively, it has closed loop poles at imaginary axis and LHP (-4 and $\pm 2\sqrt{2}\iota$).

```
--> [z,p,k] = tf2zp(32*TF/(1+32*TF))
z  =
    []
p  =
  -4.         + 0.i
  -6.939D-16 + 2.8284271i
  -6.939D-16 - 2.8284271i
k  =
   32.
```

2 Q2

Open Loop Transfer Function is

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

a)

We know that for % OS = 10,

$$\zeta = -\frac{\ln(\%OS/100)}{\sqrt{\pi^2 + (\ln(\%OS/100))^2}} \quad \Rightarrow \quad \zeta = 0.5911550338$$

$$\text{slope} = -\frac{\sqrt{1-\zeta^2}}{\zeta} = -1.3643763538$$

The root locus intersects this line at $-1.5 + 2.047\iota$, giving $K = 4.439$.

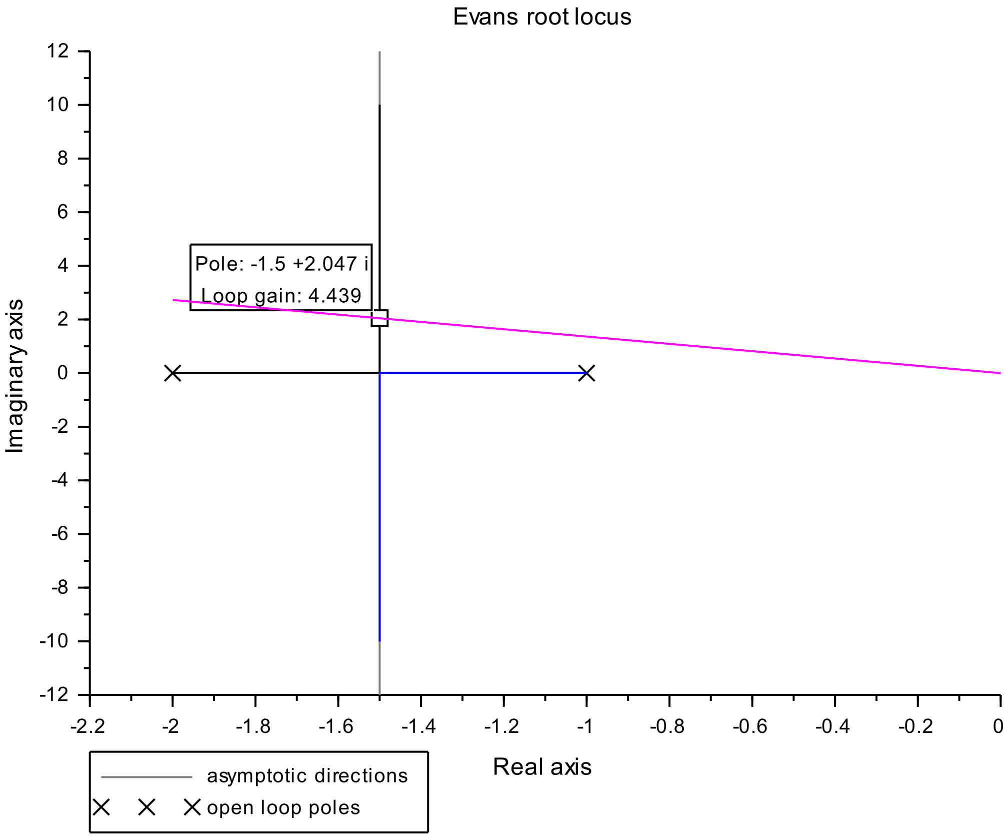


Figure 2.1: Root Locus $G(s)$

Code to obtain Figure 2.1

```
s = %s;
G = 1/(s^2+3*s+2);
OSp = 10
zeta = -log(OSp/100)/sqrt(%pi^2+(log(OSp/100))^2)
slope = -sqrt(1-zeta*zeta)/zeta;
evans(G, 100);
x = -2;
plot([0 x], [0 slope*x], 'm');
xs2pdf(0,'Q2a');
```

b)

$$\text{SSE} = \lim_{s\to 0} \frac{sR(s)}{1+K_p\cdot G(s)} = \left(R(s) = \frac{1}{s}\right)$$

Therefore,

$$\text{SSE} = \lim_{s\to 0} \frac{1}{1+K_p\cdot G(s)} = \frac{1}{1+4.439\cdot\frac{1}{2}} = 0.3106072$$

$$\text{New SSE} = \lim_{s\to 0} \frac{1}{1+K_p\cdot G(s)} = \frac{1}{1+4.439\cdot 20\cdot\frac{1}{2}} = 0.0220312844$$

Let the pole of Lag Compensator be at 0.01.

$$\text{Lag Compensator} = \frac{K(s+z)}{(s+p)} = 4.439\frac{(s+0.4)}{(s+0.02)}$$

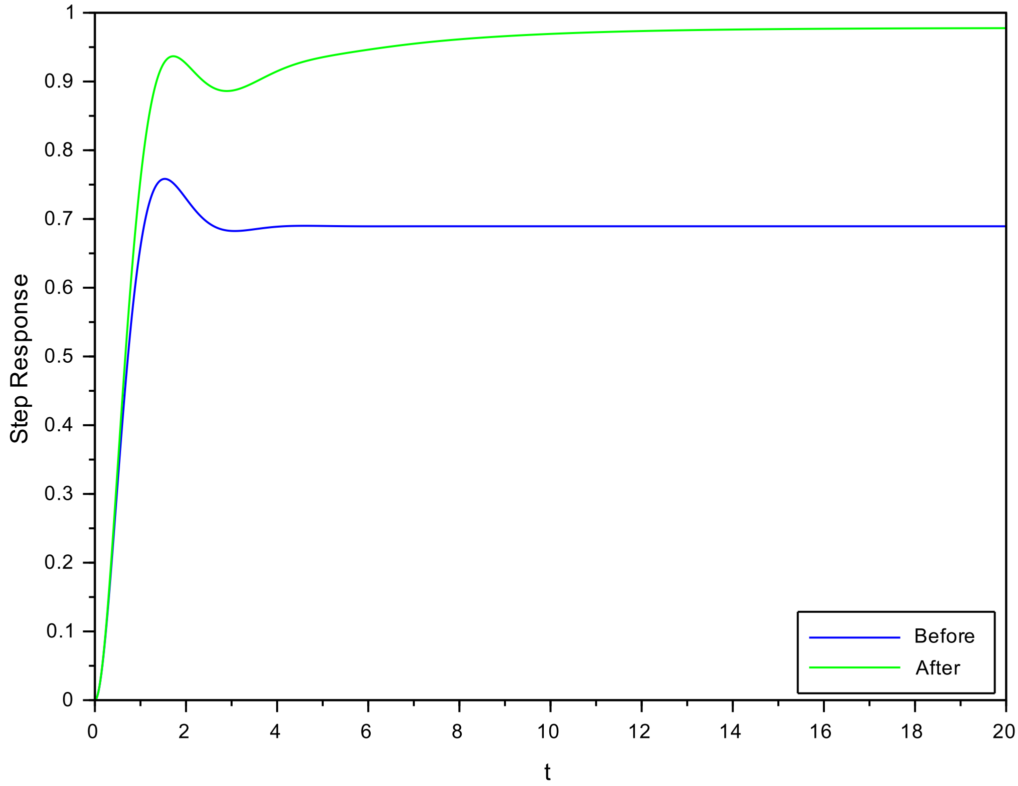


Figure 2.2: Step Response before and after the addition of the Lag Compensator

Code to obtain Figure 2.2

```
s = %s
tMax = 20;
tStep = 0.01;
t = 0:tStep:tMax;
p = 0.02;
z = 20 * p;
PI = (s+z)/(s+p);
K = 4.439;
G = K/(s^2+3*s+2);
TF = G * PI;
G = syslin('c', G);
TF = syslin('c', TF);
G = csim('step', t, G/(1+G));
TF = csim('step', t, TF/(1+TF));
plot(t,G,'b'); xlabel("t" ); ylabel("Step Response");
plot(t,TF,'g'); xlabel("t" ); ylabel("Step Response");
legend(['Before';'aAfter'], opt = 4);
xs2pdf(0,'Q2b');
```

c)

When p is very less ($<$ 0.01), the settling time is very high. As p increases, the settling time decreases until overshoot happens. Then the %OS and settling time increases and finally after a value (p$\approx$ 0.26) the system become unstable.

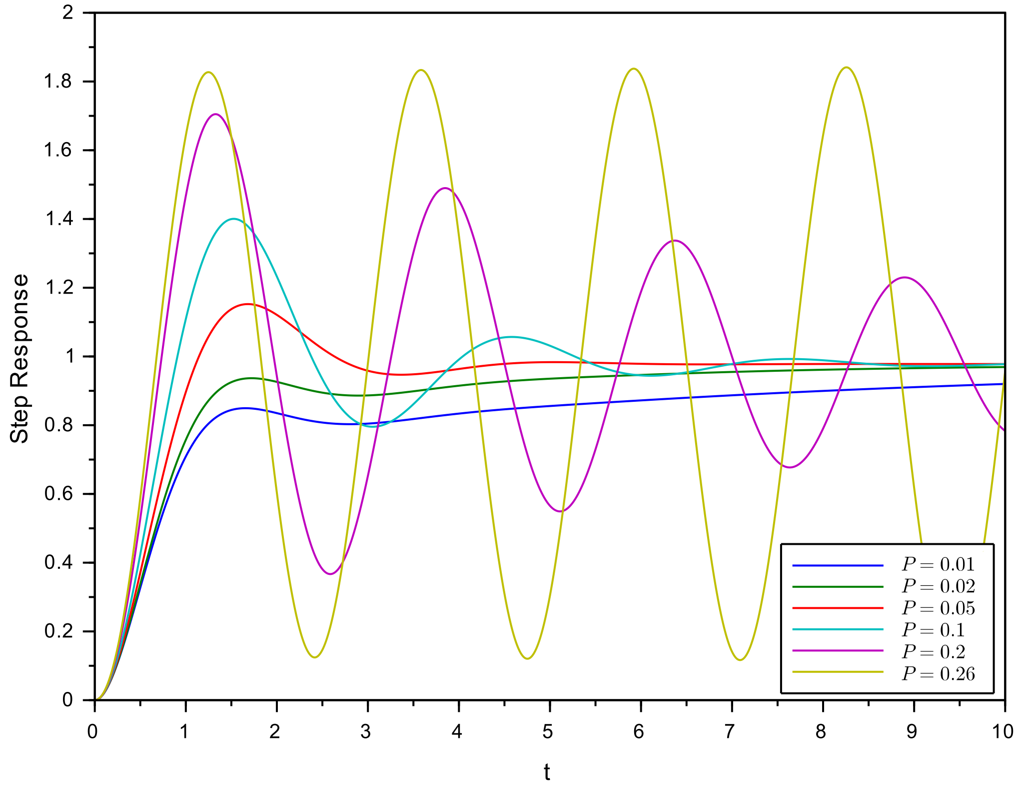


Figure 2.3: Variation in Step Response for different pole values

Code to obtain Figure 2.3

```
s = %s;
tMax = 10;
tStep = 0.01;
t = 0:tStep:tMax;
TF = K/(s^2+3*s+2);
y = [];
pValues = [0.01 0.02 0.05 0.1 0.2 0.26]
legendValues = [];
dims = 1;
for i = 1:length(pValues)
    p = pValues(i);
    z = 20 * p;
    str = "$P = " + string(p) + "$";
    legendValues = cat(dims, legendValues, str);
    PI = (s+z)/(s+p);
    sys = syslin('c', TF*PI/(1+TF*PI));
    gp = csim('step', t, sys);
    y = cat(dims, y, csim('step', t, sys));
end
plot(t',y'); xlabel(" t" ); ylabel(" Step Response" );
legend(legendValues,opt=4);
xs2pdf(0,'Q2c');
```

3 Q3

2% Settling Time

```
function Ts = settlingTime(gp, tStep)
    finalValue = gp($);
    Ts = 0;
    for x = length(gp):-1:1
        if abs(gp(x) - finalValue) > (0.02 * finalValue)
            Ts = x;
        break
        end
    end
    Ts = Ts * tStep;
endfunction
```

Open Loop Transfer Function is

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

a)

The 2% Settling Time was found to be 2.34s. We require $T_{s'} = \frac{T_s}{2}$

$$T_s = -\frac{\ln(0.02\sqrt{1-\zeta^2})}{\zeta\omega_n} \Rightarrow \text{Real Part} = \zeta\omega_n = -\frac{\ln(0.02\sqrt{1-\zeta^2})}{T_{s'}} = -3.5273525951$$

$$\text{Imaginary Part} = \text{slope} \cdot \text{Real Part} = 1.3643763538 \cdot 3.5273525951 = 4.8126364724$$

Hence, a required closed loop pole on root locus is

$$-3.5273525951 + 4.8126364724\imath$$

Let's assume zero at -10, the corresponding pole location which cancels the angle contribution of this zero is

$$\frac{4.8126364724}{p_c - 3.5273525951} = \tan(\theta) \Rightarrow p_c = 3.5273525951 + \frac{4.8126364724}{\tan(\theta)}$$

Where,

$$\arctan(\text{slope}) = \text{Angle with zero} - \text{Angle with Pole 1} - \text{Angle with Pole 2} - \theta$$

```
s = %s;
tMax = 10;
tStep = 0.01;
t = 0:tStep:tMax;
K = 4.439;
G = K/(s^2+3*s+2);
G = syslin('c', G);
gp = csim('step', t, G/(1+G));
Ts = settlingTime(gp, tStep)
--> 2.34
```

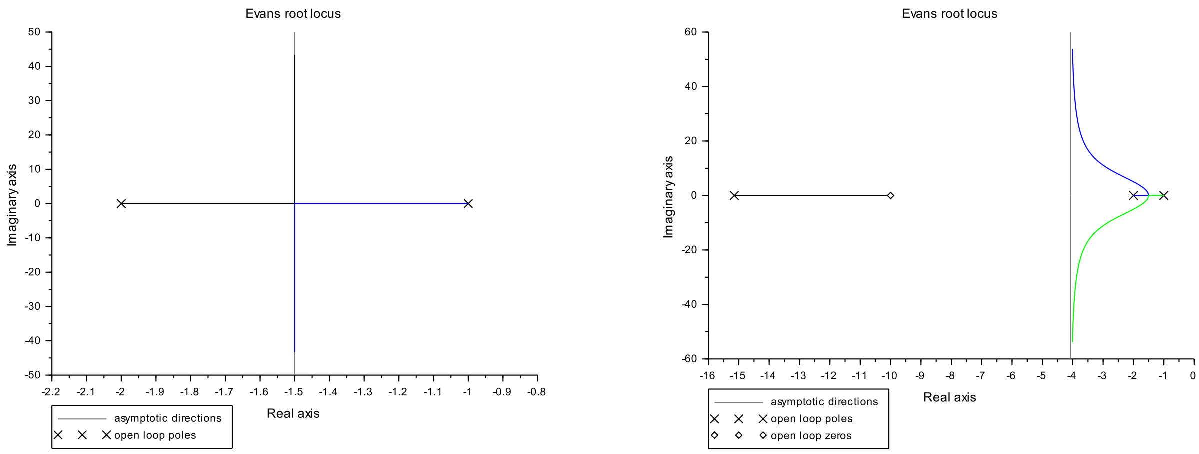
Corresponding pole calculation

```
OSp = 10;
zeta = -log(OSp/100)/sqrt(%pi^2+(log(OSp/100))^2);
slope = -sqrt(1-zeta*zeta)/zeta;
Real = -log(0.02*sqrt(1-zeta^2))/(Ts/2);
z = Real + %i*slope*Real;
p1 = -1;
p2 = -2;
zc = -10;
theta = atan(slope)*180/%pi - atan(imag(z-p1),real(z-p1))*180/%pi
```

```
      - atan(imag(z-p2),real(z-p2))*180/%pi + atan(imag(z-zc),real(z-zc))*180/%pi
pc = Real + slope*Real/tan(theta*%pi/180)
--> -15.148441
```

Hence, place pole at -15.148441 (this also satisfies 2nd order approximation) and the transfer function is From root locus, Gain at required pole is approximately 5

```
--> Gn = G * (s-zc)/(s-pc)
                  44.39 +4.439s
   --------------------------------------
   30.296883 +47.445324s +18.148441s^2 +s^3
```



(a) Root Locus Before

(b) Root Locus After

Figure 3.1: fig:Q3a1

Settling times haven't improved much.

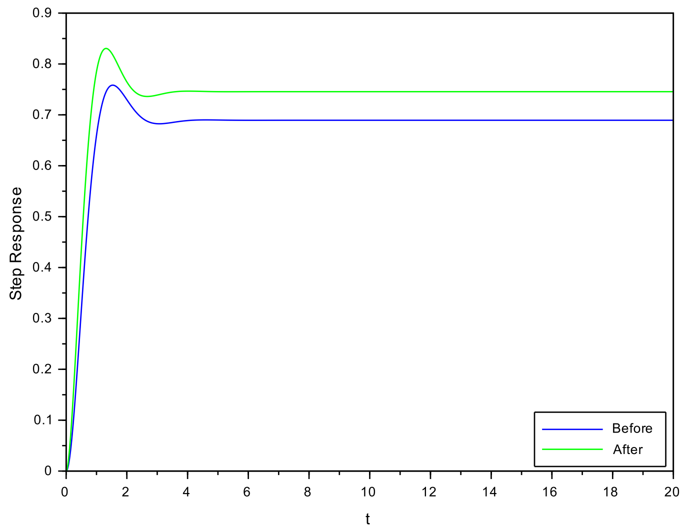


Figure 3.2: Step Response before and after the addition of the Lead Compensator

Code to obtain Figure 3.2

```
s = %s
tMax = 20;
tStep = 0.01;
t = 0:tStep:tMax;
p = 0.02;
z = 20 * p;
PI = (s+z)/(s+p);
K = 4.439;
G = K/(s^2+3*s+2);
TF = G * PI;
G = syslin('c', G);
TF = syslin('c', TF);
G = csim('step', t, G/(1+G));
TF = csim('step', t, TF/(1+TF));
plot(t,G,'b'); xlabel("t" ); ylabel("Step Response");
plot(t,TF,'g'); xlabel("t" ); ylabel("Step Response");
settlingTime(gp1, tStep);
--> 2.34
settlingTime(gp2, tStep);
--> 1.99
legend(['Before';'aAfter'], opt = 4);
xs2pdf(0,'Q3a2');
evans(G);
xs2pdf(0,'Q3a11');
evans(Gn);
xs2pdf(0,'Q3a12');
```

b)

The 2% Settling Time was found to be 2.34s. We require $T_{s'} = \frac{T_s}{2}$ The required closed loop pole on root locus are same as before

$$-3.5273525951 + 4.8126364724\iota$$

Now to calculate the zero, we again use angle contribution which gives

$$\frac{4.8126364724}{3.5273525951 - z_c} = \tan(180 - \theta) \Rightarrow p_c = 3.5273525951 + \frac{4.8126364724}{\tan(\theta)}$$

Where,

$$\arctan(\text{slope}) = \text{Angle with zero}(\theta) - \text{Angle with Pole 1} - \text{Angle with Pole 2}$$

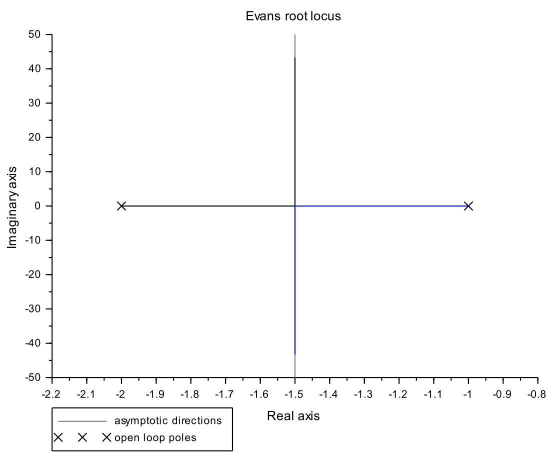
```
s = %s;
tMax = 10;
tStep = 0.01;
t = 0:tStep:tMax;
K = 4.439;
G = K/(s^2+3*s+2);
G = syslin('c', G);
gp = csim('step', t, G/(1+G));
Ts = settlingTime(gp, tStep)
--> 2.34
```

Zero calculation

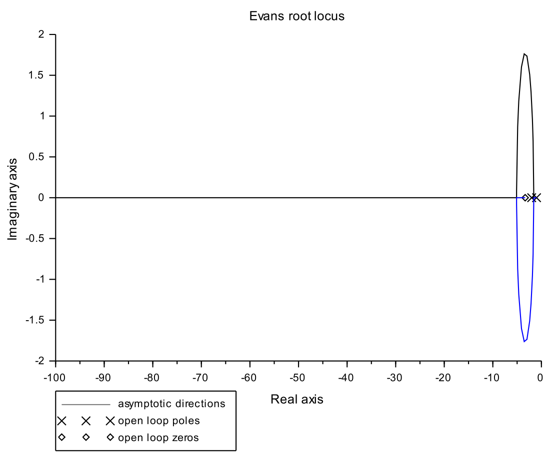
```
OSp = 10;
zeta = -log(OSp/100)/sqrt(%pi^2+(log(OSp/100))^2);
slope = -sqrt(1-zeta*zeta)/zeta;
Real = -log(0.02*sqrt(1-zeta^2))/(Ts/2);
z = Real + %i*slope*Real;
p1 = -1;
p2 = -2;
theta = 360 + atan(imag(z-p1),real(z-p1))*180/%pi + atan(imag(z-p2),real(z-p2))*180/%pi - 180
zc = slope*Real/tan(theta*%pi/180)-Real
--> -3.3420857
```

Hence, place zero at -3.3420857 the transfer function is

```
--> Gn = G * (s-zc)
   14.835518 +4.439s
   -----------------
       2 +3s +s^2
```



(a) Root Locus Before



(b) Root Locus After

Figure 3.3

Here, settling times improved a lot.

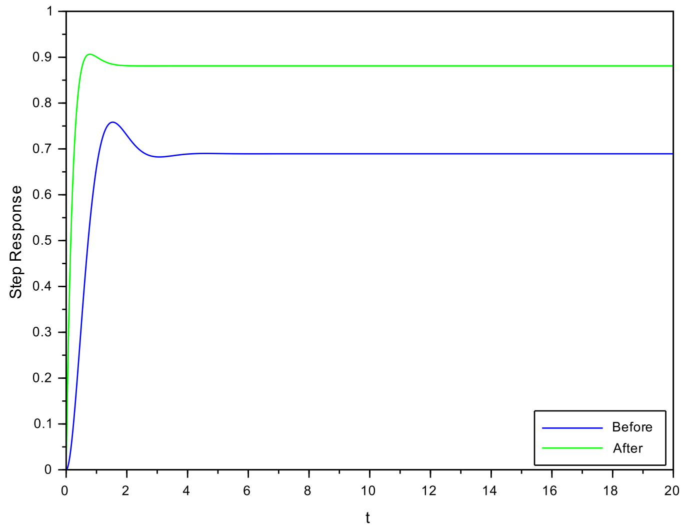


Figure 3.4: Step Response before and after the addition of the Lead Compensator

Code to obtain Figure 3.4

```
s = %s
tMax = 20;
tStep = 0.01;
t = 0:tStep:tMax;
p = 0.02;
z = 20 * p;
PI = (s+z)/(s+p);
K = 4.439;
G = K/(s^2+3*s+2);
TF = G * PI;
G = syslin('c', G);
TF = syslin('c', TF);
G = csim('step', t, G/(1+G));
TF = csim('step', t, TF/(1+TF));
plot(t,G,'b'); xlabel("t" ); ylabel("Step Response");
plot(t,TF,'g'); xlabel("t" ); ylabel("Step Response");
settlingTime(gp1, tStep);
--> 2.34
settlingTime(gp2, tStep);
--> 1.05
legend(['Before';'aAfter'], opt = 4);
xs2pdf(0,'Q3b2');
evans(G);
xs2pdf(0,'Q3b11');
evans(Gn);
xs2pdf(0,'Q3b12');
ax = gca();
ax.data_bounds=[-100 -2;0 2];
```