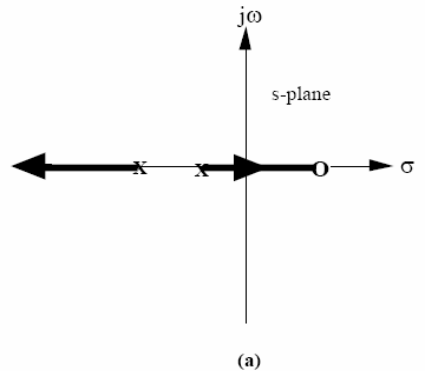
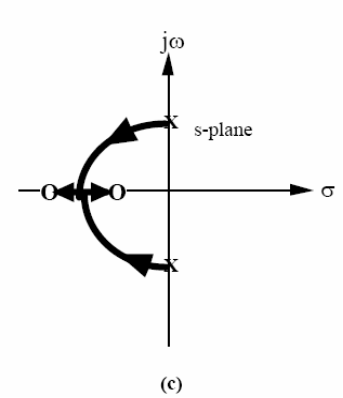
EME172
Homework #5 Solution

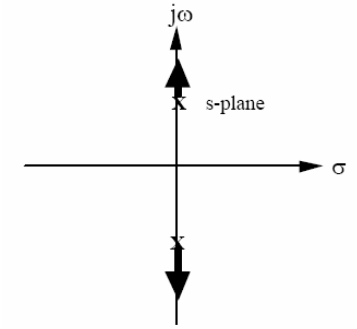
2. Problem 8-2 (a, c, d, f):



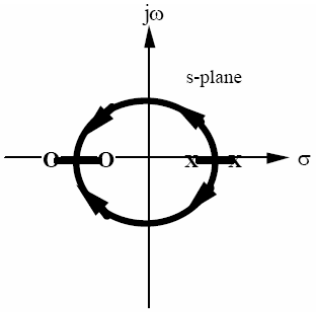
(a)



(c)

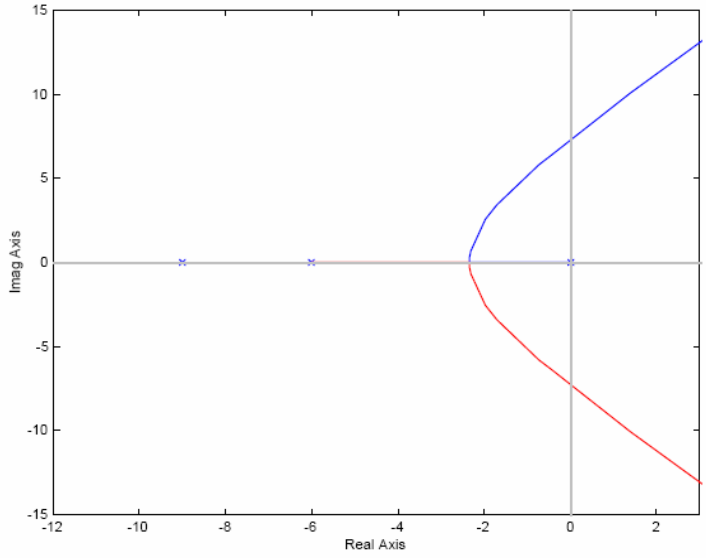


(d)



(f)

3. Problem 8-14:



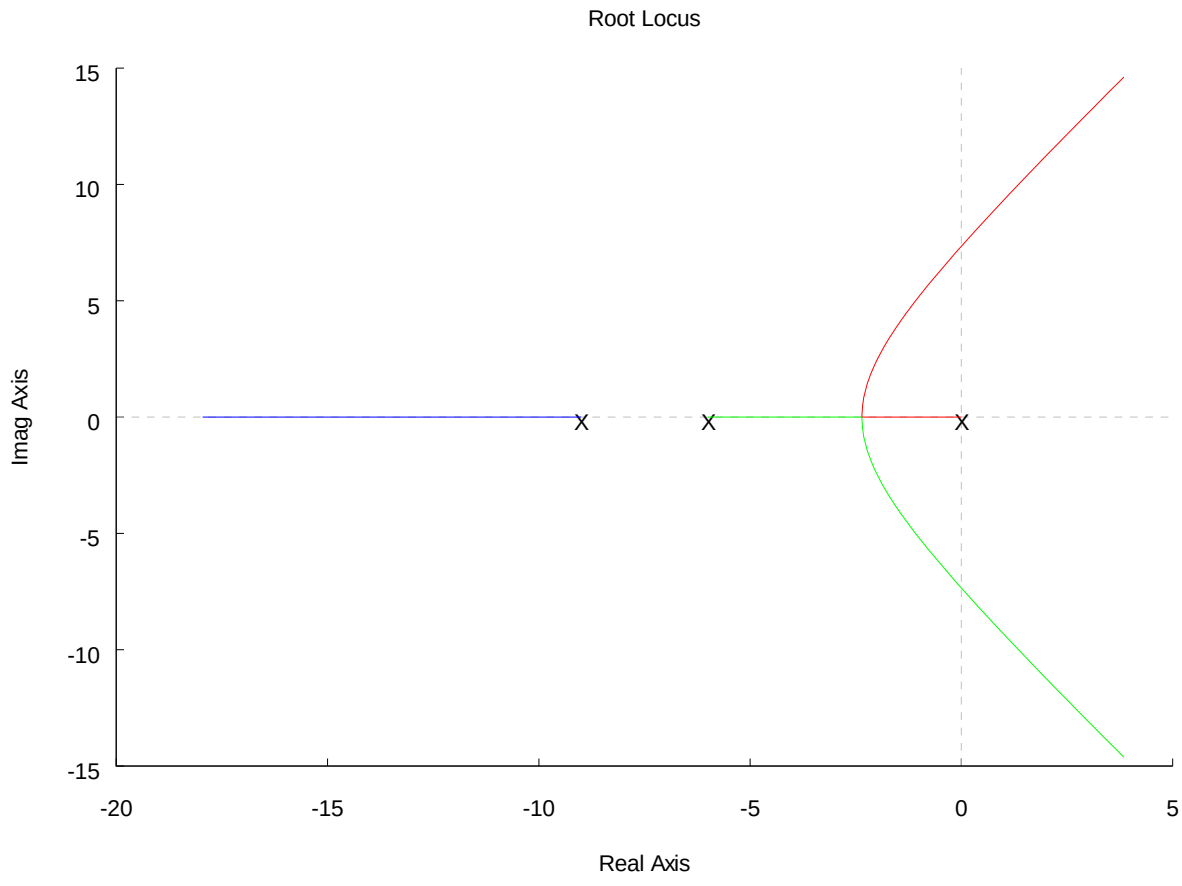
Root locus crosses the imaginary axis at j7.348 with a gain of 810. Real axis breakaway is at –2.333 at a gain of 57.04. Real axis intercept for the asymptotes is $\frac{-15}{3} = -5$. The angle of the asymptotes is $= \frac{\pi}{3}, \pi, \frac{5\pi}{3}$.

4.

Program:

```
/***********************
 * File name: p8-14.ch
 ***********************/

#include <control.h>

int main() {
    double k = 1;
    array double complex p[3] = {complex(0, 0),
                                 complex(-6, 0),
                                 complex(-9, 0)};
    CControl sys;
    CPlot plot;

    sys.model("zpk", NULL, p, k);
    sys.rlocus(&plot, NULL, NULL);

    return 0;
}
```

Output:



Program:

```
/************************
 * File name: p8-14c.ch
 ************************/

#include <control.h>

int main() {
    // default system gain
    double dk = 1;

    // open-loop poles
    array double complex op[3] = {complex(0, 0),
                                  complex(-6, 0),
                                  complex(-9, 0)};

    // dominant closed-loop pole selected from the root locus
    array double complex dp[1] = {complex(0, 7.348)};

    // system gain when one of the dominant closed-loop poles is selected
    array double k[1];

    CControl sys;

    sys.model("zpk", NULL, op, dk);
    sys.rlocfind(k, NULL, dp);
    printf("k: %f\n", k);

    return 0;
}
```

Output:

```
k: 809.896562
```

Program:

```
/************************
 * File name: p8-14d.ch
 ************************/

#include <control.h>

int main() {
    // default system gain
    double dk = 1;

    // open-loop poles
    array double complex op[3] = {complex(0, 0),
                                  complex(-6, 0),
                                  complex(-9, 0)};

    // dominant closed-loop pole selected from the root locus
    array double complex dp[1] = {complex(-2.333, 0)};

    // system gain when one of the dominant closed-loop poles is selected
    array double k[1];

    CControl sys;

    sys.model("zpk", NULL, op, dk);
    sys.rlocfind(k, NULL, dp);
    printf("k: %f\n", k);

    return 0;
}
```

Output:

```
k: 57.036925
```

5. Problem 9-5:

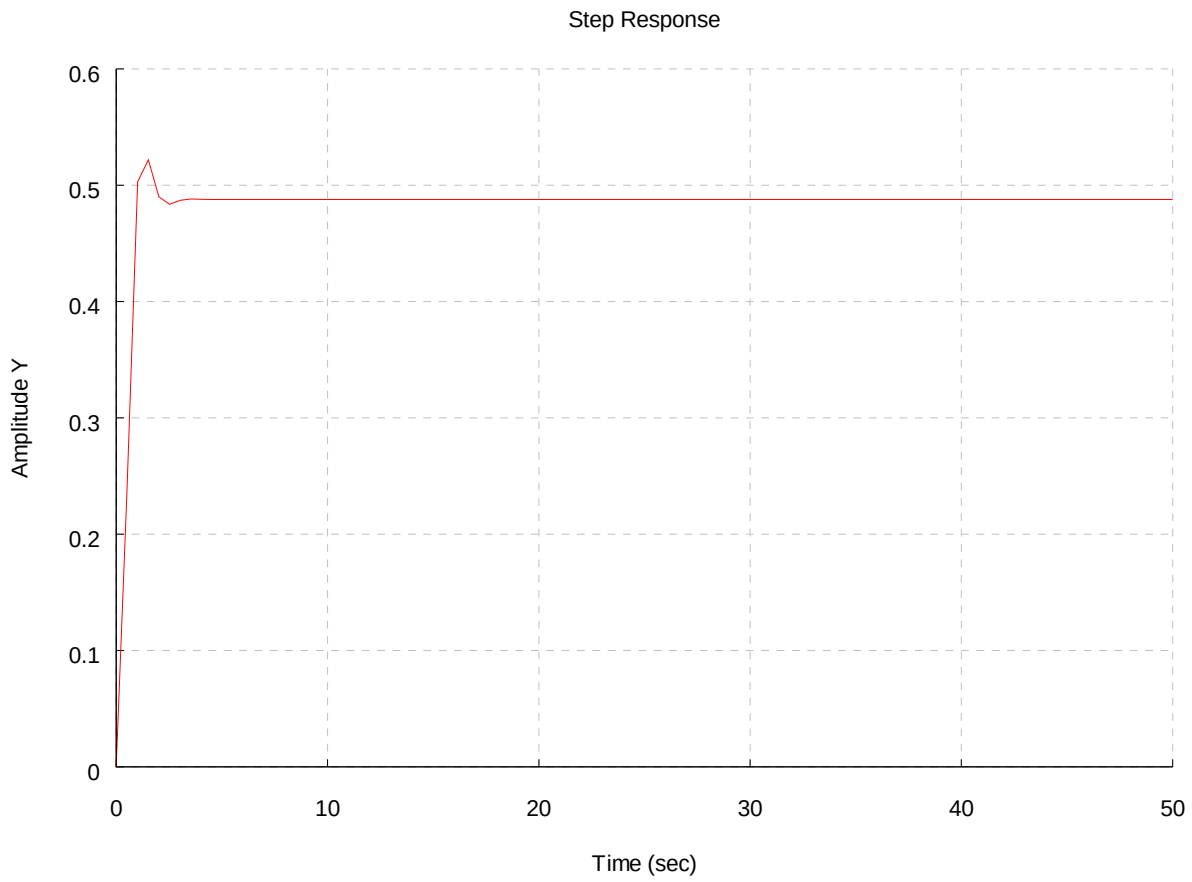
a. Uncompensated: Searching along the 126.16° line (10% overshoot, $\zeta = 0.59$), find the operating point at -2.03 + j2.77 with K = 45.72. Hence, $K_p = \frac{45.72}{2 \times 4 \times 6} = 0.9525$. An improvement of $\frac{20}{0.9525} = 20.1$ is required. Let $G_c(s) = \frac{0.201}{0.01}$. Compensated: Searching along the 126.16° line (10% overshoot, $\zeta = 0.59$), find the operating point at - 1.99+j2.72 with K = 46.05. Hence, $K_p = \frac{46.05 \times 0.201}{2 \times 4 \times 6 \times 0.01} = 19.28$.

b.

Program:

```
/*****************************
 * File name: p9-5b_uncomp.ch
 *****************************/

#include <control.h>

int main() {
    // system gain
    double k = 45.72;

    // open-loop poles
    array double complex op[3] = {complex(-2, 0),
                                  complex(-4, 0),
                                  complex(-6, 0)};
    array double num[1] = {1};
    array double den[1] = {1};
    double tf = 50;

    CControl sys1, sys2, *sys3;
    CPlot plot;

    sys1.model("zpk", NULL, op, k);
    sys2.model("tf", num, den);
    sys3 = sys1.feedback(&sys2);
    sys3->grid(1);
    sys3->step(&plot, NULL, NULL, NULL, tf);

    return 0;
}
```

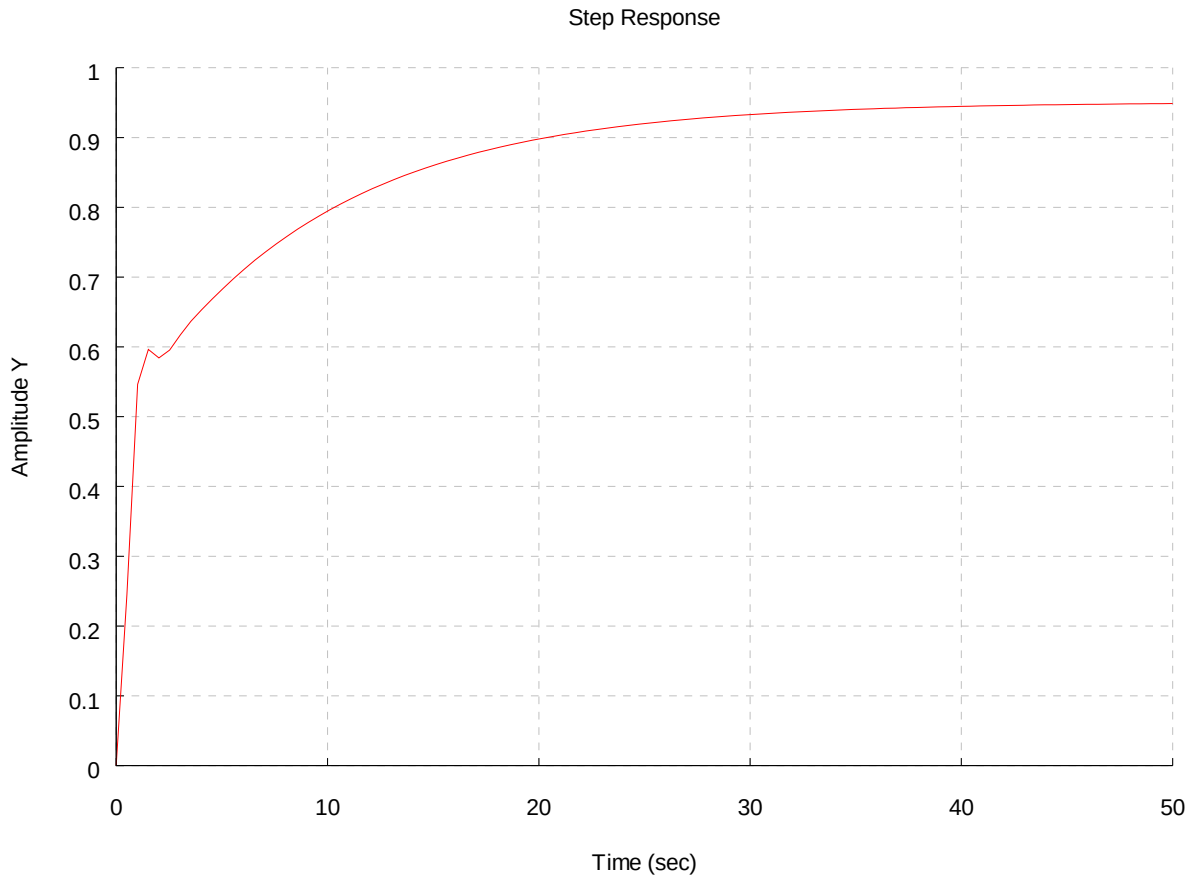
Output:


Step Response
Amplitude Y
Time (sec)
0
0.1
0.2
0.3
0.4
0.5
0.6
0
10
20
30
40
50

Program

```
/*****************************
 * File name: p9-5b_comp.ch
 *****************************/

#include <control.h>

int main() {
    // system gain
    double k = 46.05;

    // open-loop zero
    array double complex oz[1] = {complex(-0.201, 0)};

    // open-loop poles
    array double complex op[4] = {complex(-0.01, 0),
                                  complex(-2, 0),
                                  complex(-4, 0),
                                  complex(-6, 0)};
    array double num[1] = {1};
    array double den[1] = {1};
    double tf = 50;

    CControl sys1, sys2, *sys3;
    CPlot plot;

    sys1.model("zpk", oz, op, k);
    sys2.model("tf", num, den);
    sys3 = sys1.feedback(&sys2);
    sys3->grid(1);
    sys3->step(&plot, NULL, NULL, NULL, tf);

    return 0;
}
```

Output:



c. From (b), about 28 seconds

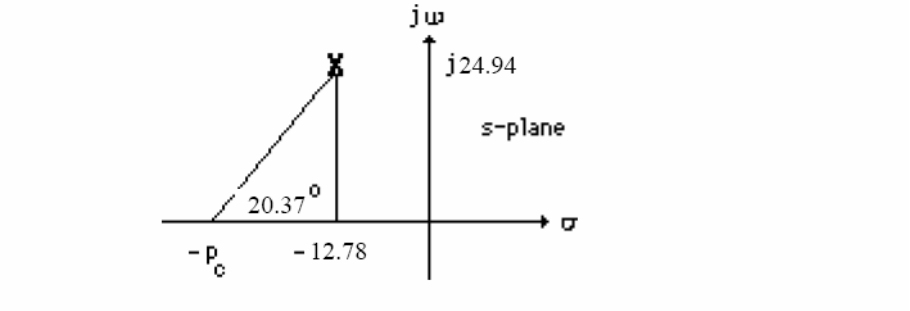
6. Problem 9-13:

a. Searching along the 117.13o line (%OS = 20%; $\zeta = 0.456$), find the operating point at

-6.39 + j12.47 with K = 9273. Searching along the real axis for K = 9273, we find a higher-order pole at –47.22. Thus, $T_s = \frac{4}{\zeta\omega_n} = \frac{4}{6.39} = 0.626$ second.

b. For the settling time to decrease by a factor of 2, Re = $-\zeta\omega_n$ = -6.39 x 2 = -12.78. The imaginary part is Im = -12.78 tan 117.13° = 24.94. Hence, the compensated closed-loop poles are -12.78 ± j24.94. A settling time of 0.313 second would result.

c. Assume a compensator zero at -20. Using the uncompensated system's poles along with the compensator zero, the summation of angles to the design point, -12.78 ± j24.94 is –159.63°. Thus, the contribution of the compensator pole must be 159.63°-180° = -20.37°. Using the following geometry, $\frac{24.94}{p_c - 12.78} = \tan 20.37°$, or $p_c = 79.95$.



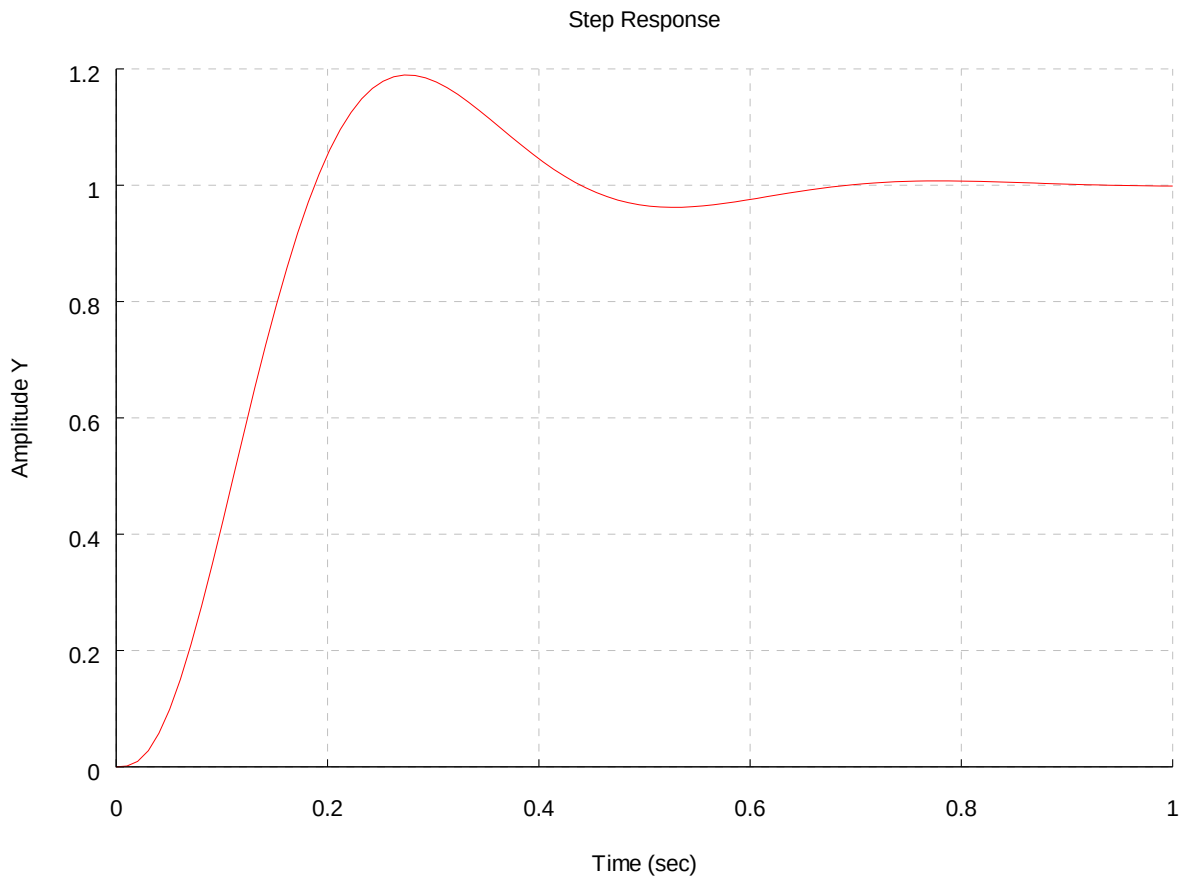
Adding the compensator pole and using --12.78 ± j24.94 as the test point, K = 74130.

d.

Program:

```
/*****************************
 * File name: p9-13d_uncomp.ch
 *****************************/

#include <control.h>

int main() {
    // system gain
    double k = 9273;

    // open-loop poles
    array double complex op[3] = {complex(0, 0),
                                  complex(-20, 0),
                                  complex(-40, 0)};
    array double num[1] = {1};
    array double den[1] = {1};
    double tf = 1;

    CControl sys1, sys2, *sys3;
    CPlot plot;

    sys1.model("zpk", NULL, op, k);
    sys2.model("tf", num, den);
    sys3 = sys1.feedback(&sys2);
    sys3->grid(1);
    sys3->step(&plot, NULL, NULL, NULL, tf);

    return 0;
}
```

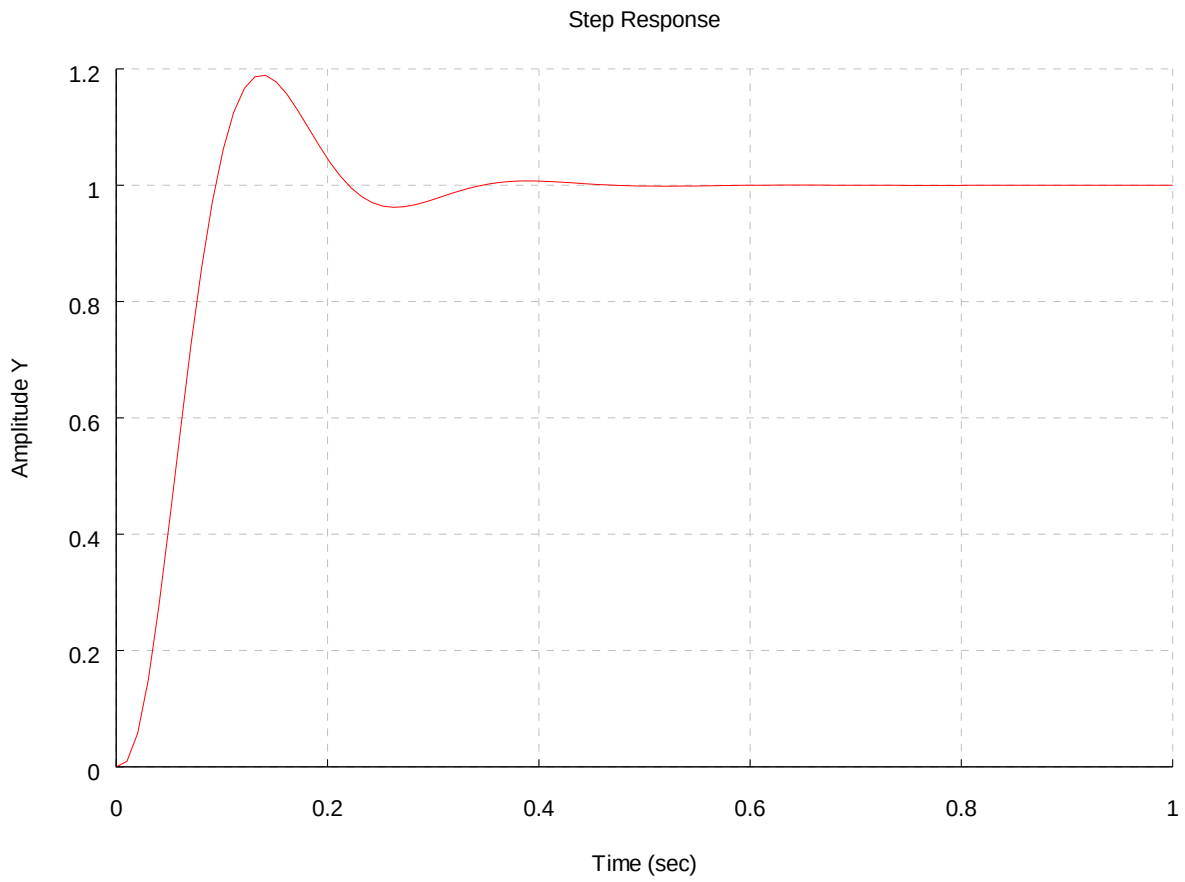
Output:


Step Response
Amplitude Y
Time (sec)
0
0.2
0.4
0.6
0.8
1
1.2
0
0.2
0.4
0.6
0.8
1

Program:

```
/*****************************
 * File name: p9-13d_comp.ch
 *****************************/

#include <control.h>

int main() {
    // system gain
    double k = 74130;

    // open-loop zero
    array double complex oz[1] = {complex(-20, 0)};

    // open-loop poles
    array double complex op[4] = {complex(-79.95, 0),
                                  complex(0, 0),
                                  complex(-20, 0),
                                  complex(-40, 0)};
    array double num[1] = {1};
    array double den[1] = {1};
    double tf = 1;

    CControl sys1, sys2, *sys3;
    CPlot plot;

    sys1.model("zpk", oz, op, k);
    sys2.model("tf", num, den);
    sys3 = sys1.feedback(&sys2);
    sys3->grid(1);
    sys3->step(&plot, NULL, NULL, NULL, tf);

    return 0;
}
```

```
Output:
```



7. Program 10-37:

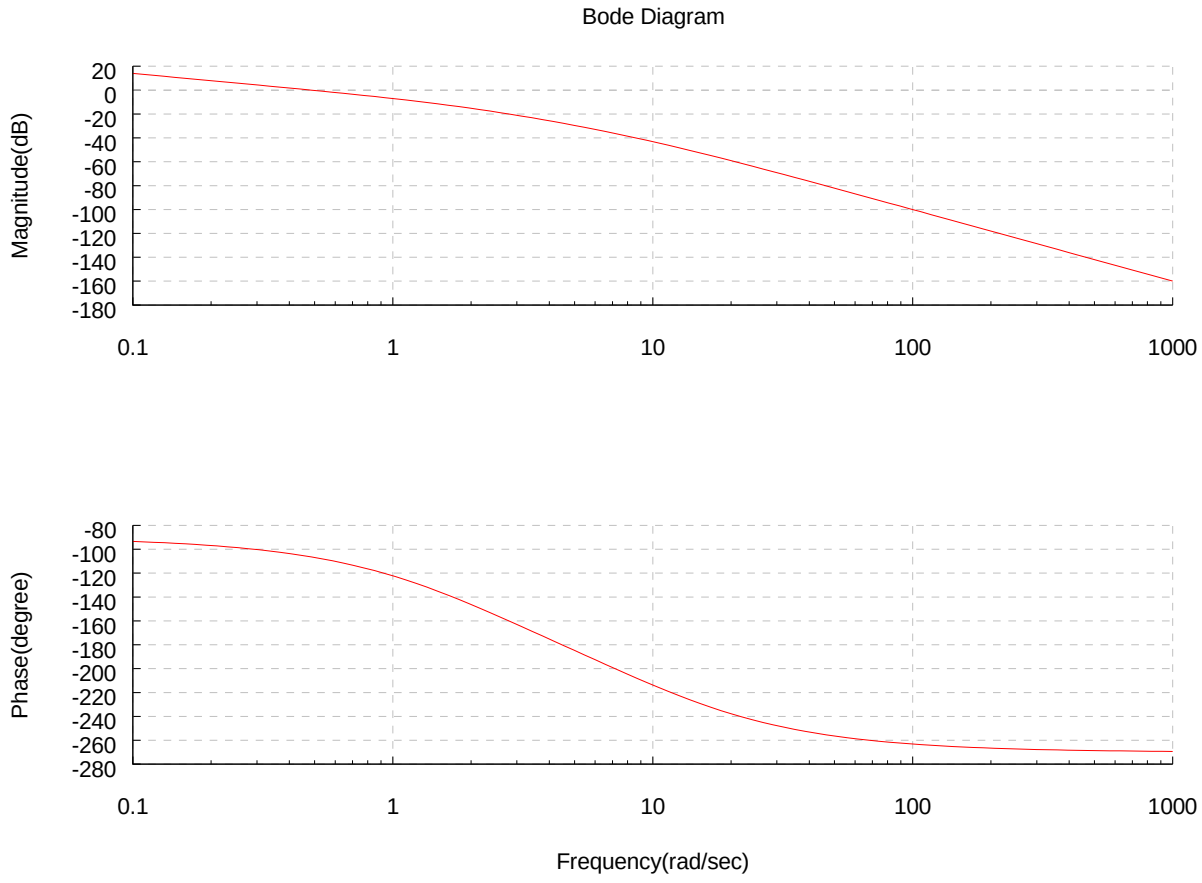
a. See the next page

b. From the Bode plot shown in the next page, we may obtain the gain margin = 27.6 dB and the phase margin = 73.6 deg.

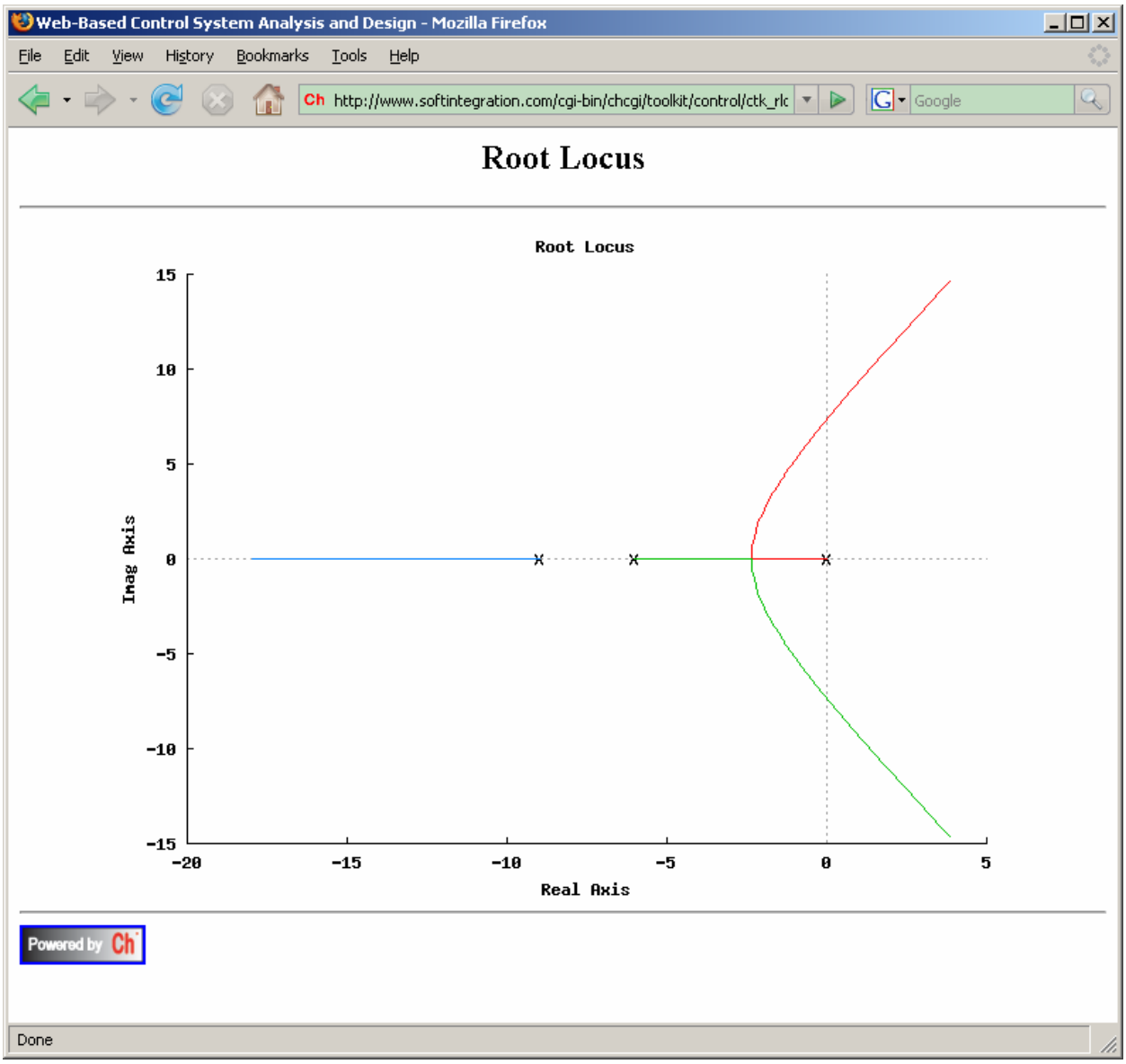
c.
Program:

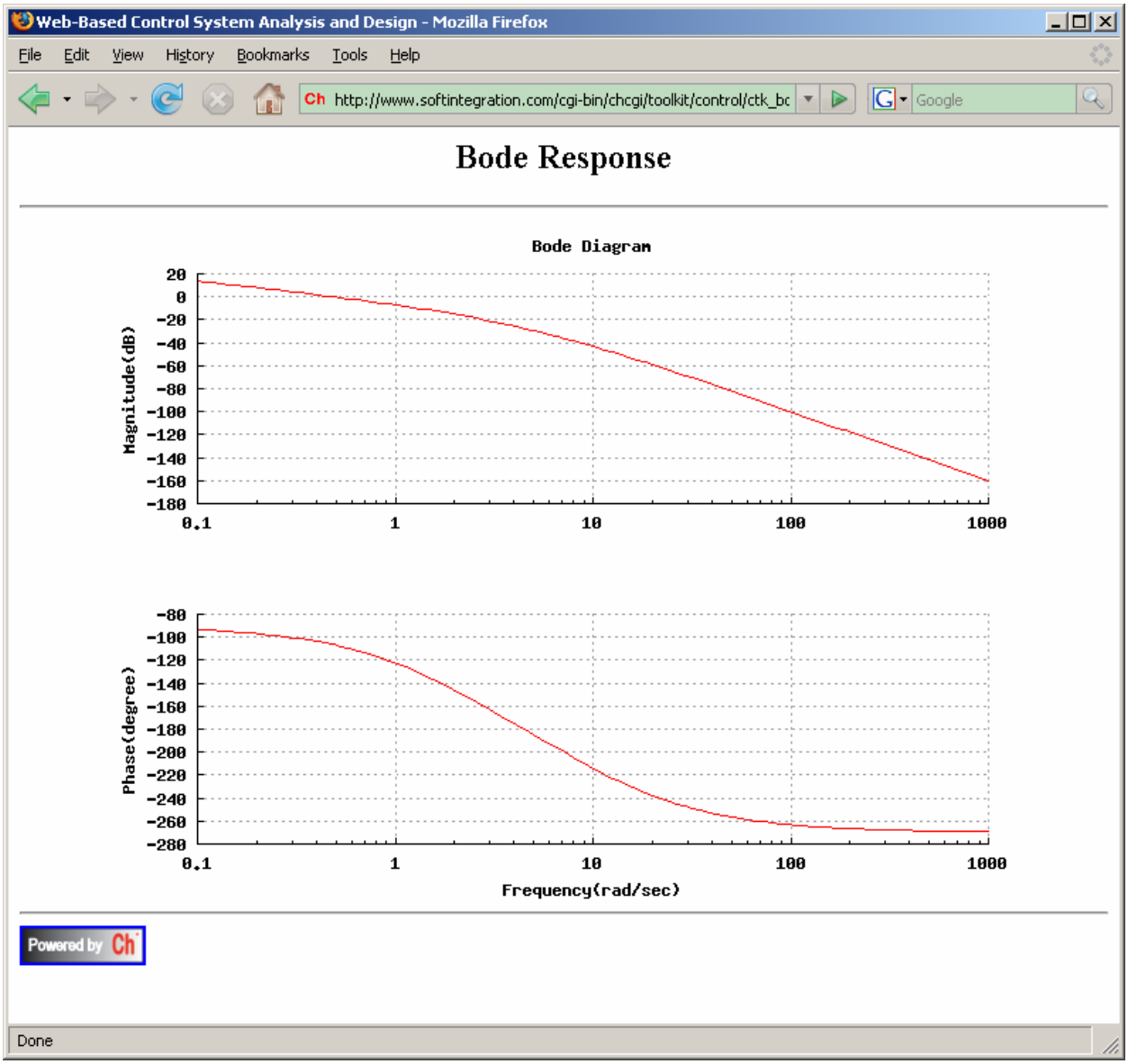
```
/************************
 * File name: p10-37c.ch
 ************************/

#include <control.h>

int main() {
    double k = 10;
    array double complex p[3] = {complex(0, 0),
                                 complex(-2, 0),
                                 complex(-10, 0)};
    CControl sys;
    CPlot plot;

    sys.model("zpk", NULL, p, k);
    sys.grid(1);
    sys.bode(&plot, NULL, NULL, NULL);

    return 0;
}
```

Output:



8.


Web-Based Control System Analysis and Design - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://www.softintegration.com/cgi-bin/chcgi/toolkit/control/ctk_rlc
Google
Root Locus
Root Locus
Imag Axis
Real Axis
15
10
5
0
-5
-10
-15
-20
-15
-10
-5
0
5
Powered by Ch
Done


Web-Based Control System Analysis and Design - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://www.softintegration.com/cgi-bin/chcgi/toolkit/control/ctk_bc
Google
Bode Response
Bode Diagram
Magnitude(dB)
20
0
-20
-40
-60
-80
-100
-120
-140
-160
-180
0.1
1
10
100
1000
Phase(degree)
-80
-100
-120
-140
-160
-180
-200
-220
-240
-260
-280
0.1
1
10
100
1000
Frequency(rad/sec)
Powered by Ch
Done