Sessions 5 & 6 - Introduction to Control - 2023/2024

Authors:
  Lourenço Faria
  Gustavo Toste
  Adriano Cardoso

Date:
  18/12/2023

Contents


```
clear all; close all; clc;
```

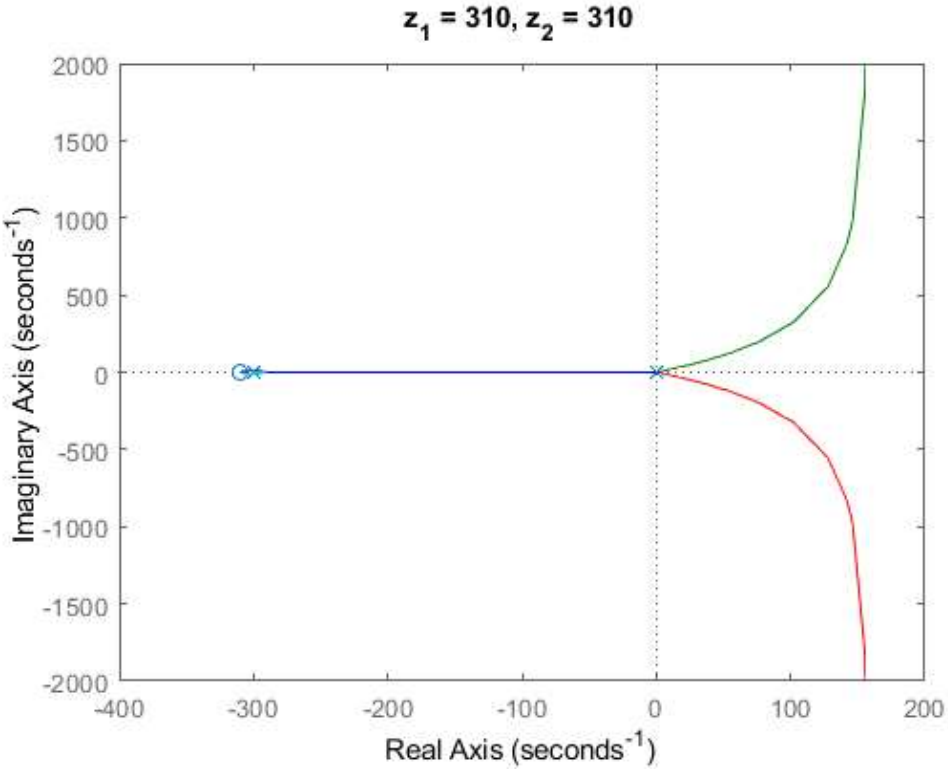
Definition of variables

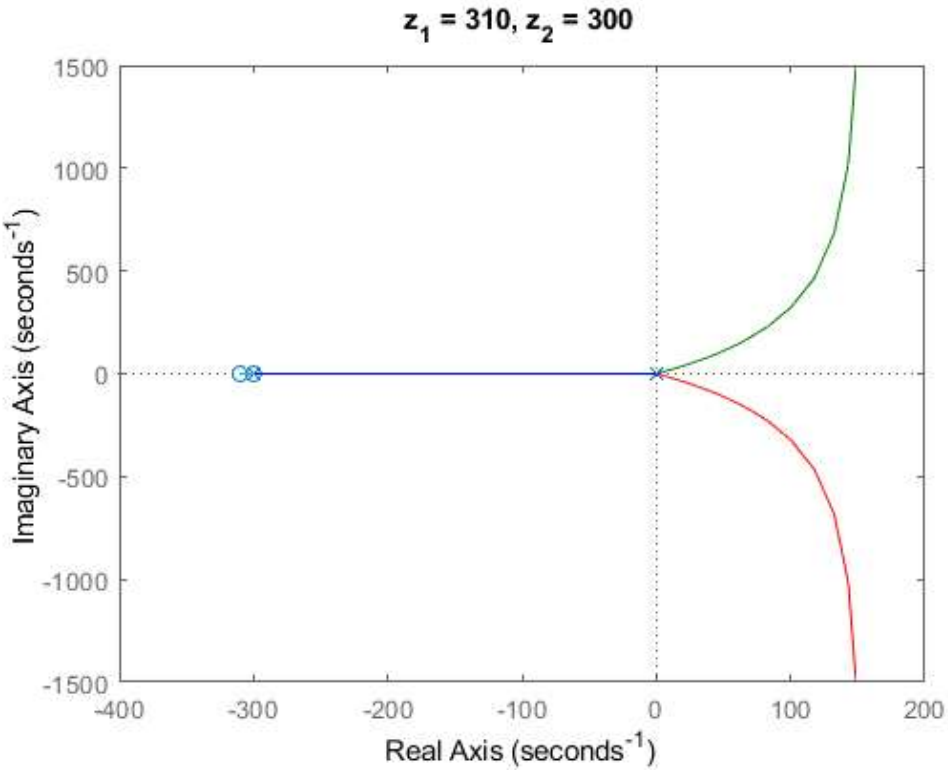
```
clear all;

G = 9.8; %m s^-2
M = 1; %kg
K_t = 3.575E-5; % N s^2 rad^-2
Z0 = 2; %m
omega0 = sqrt(G*M/K_t); %rad s^-1
```

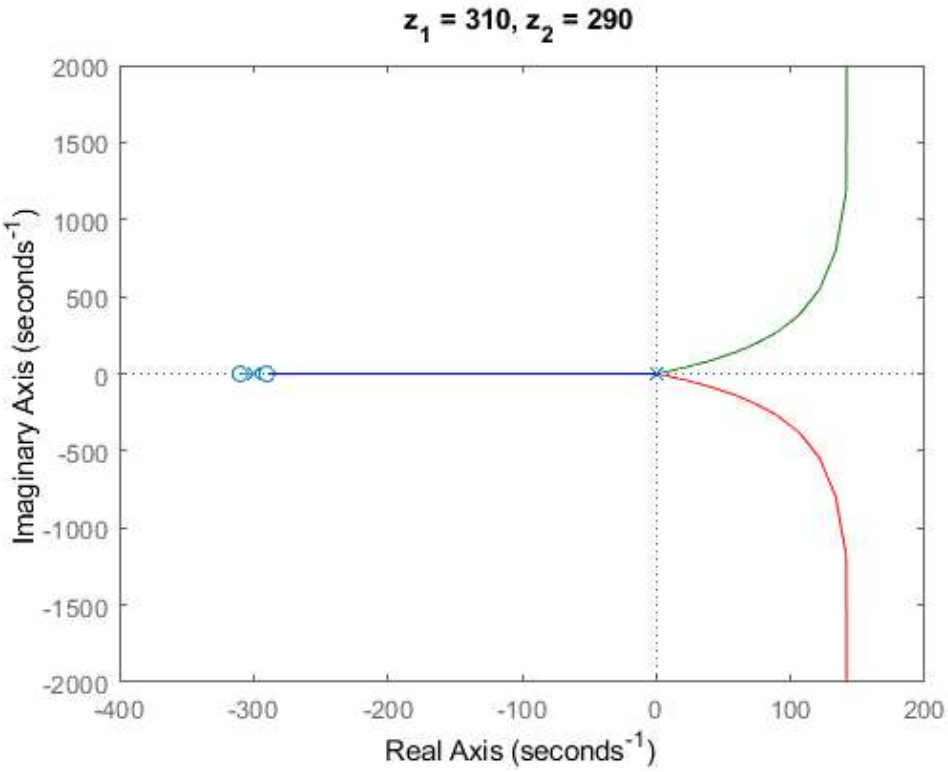
Q4.2

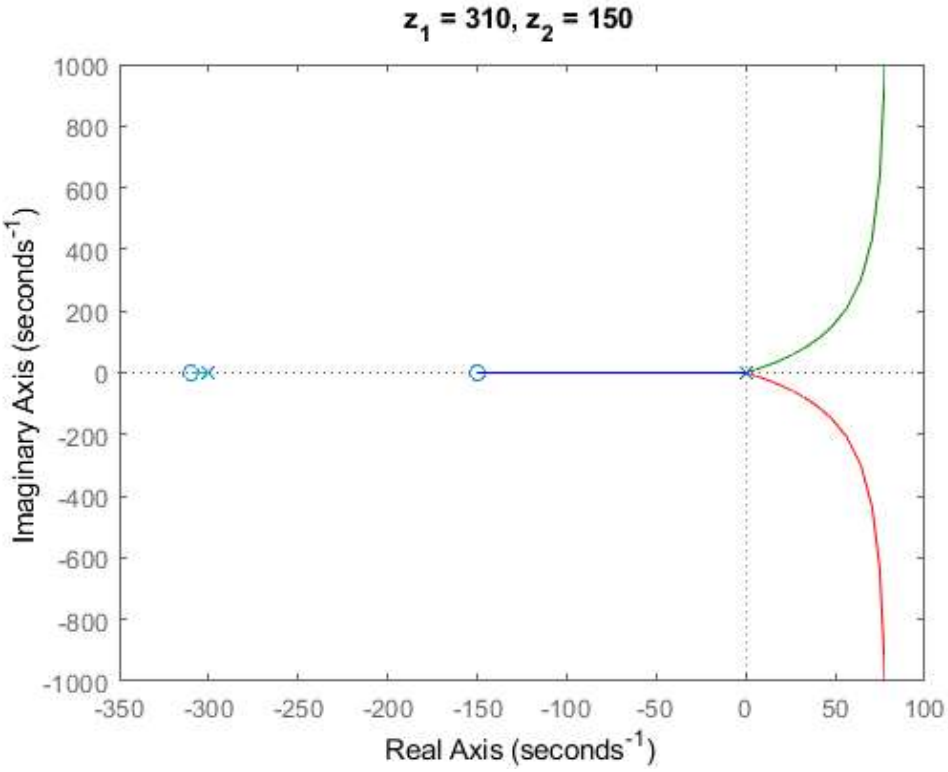
```
z_1 = 310;  %left and near
z_2 = 300;  %root mult. 1
z_3 = 290;  %between and near
z_4 = 150;  %between
z_5 = 100/3;%between
z_6 = 10;   %between and near
z_7 = 0;    %root mult. 2
z_8 = -10;  %right and near

z_array = [z_1, z_2, z_3, z_4, z_5, z_6, z_7, z_8];

for i=1:length(z_array)
    for j=i:length(z_array)
        figure;
        z1 = z_array(i);
        z2 = z_array(j);
        rlocus(tf([1 (z1+z2) z1*z2],[1 300 0 0 0]));
        title("z_1 = " + num2str(z1) + ", z_2 = " + num2str(z2));
        hold off;
```

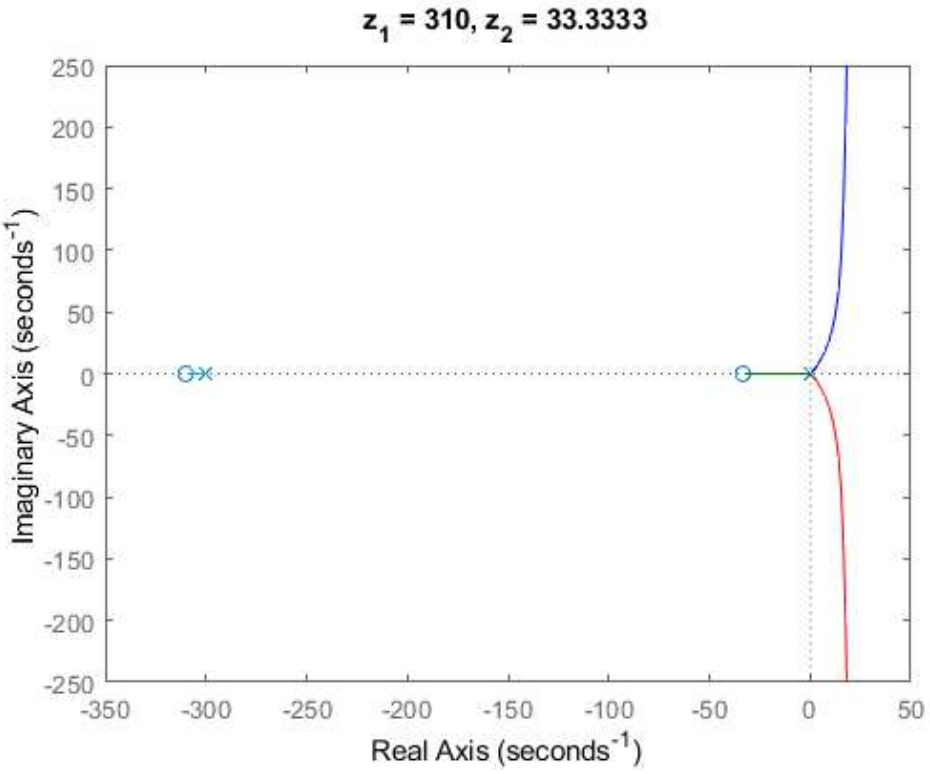
```
    end
end
```

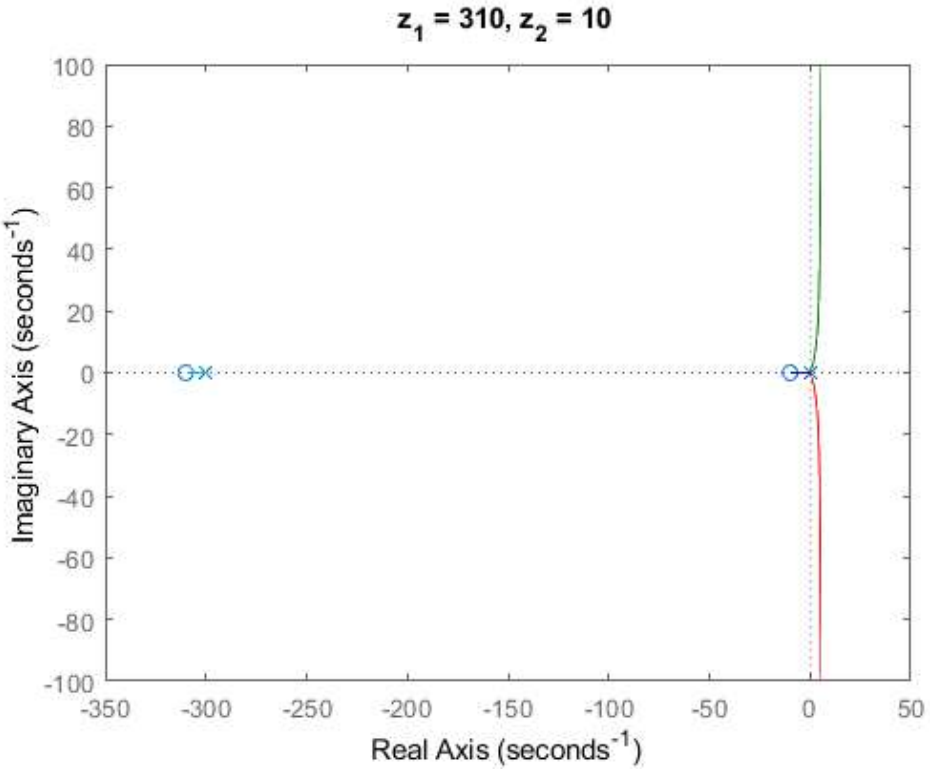

$z_1 = 310, z_2 = 310$
Imaginary Axis (seconds^{-1})
Real Axis (seconds^{-1})
2000
1500
1000
500
0
-500
-1000
-1500
-2000
-400
-300
-200
-100
0
100
200

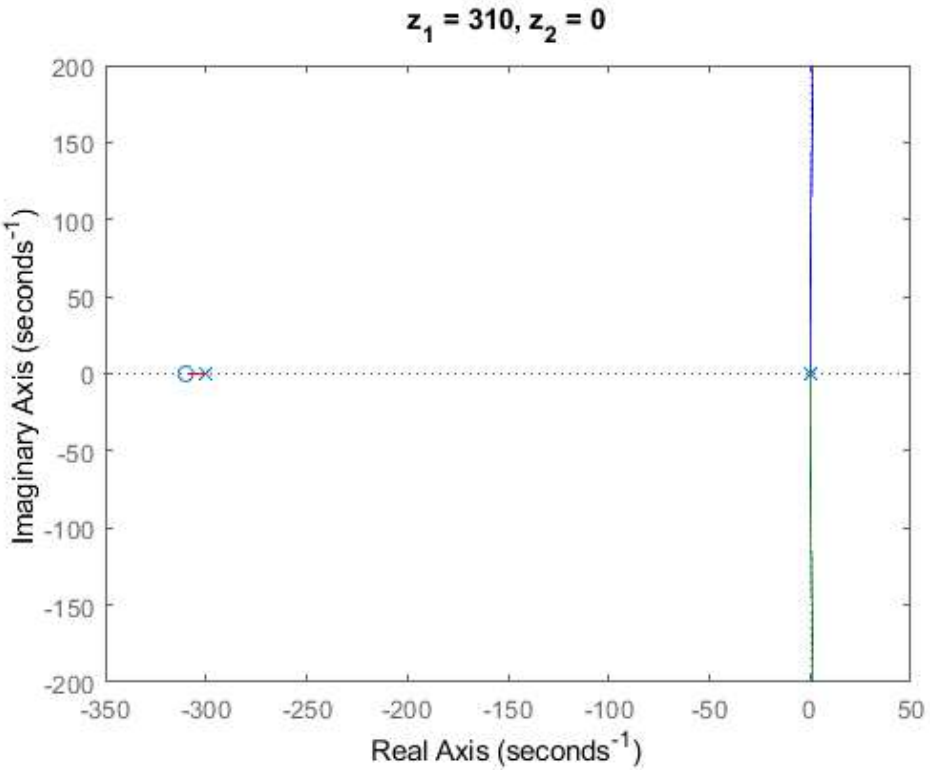

$z_1 = 310, z_2 = 300$
Imaginary Axis (seconds^{-1})
Real Axis (seconds^{-1})
1500
1000
500
0
-500
-1000
-1500
-400
-300
-200
-100
0
100
200

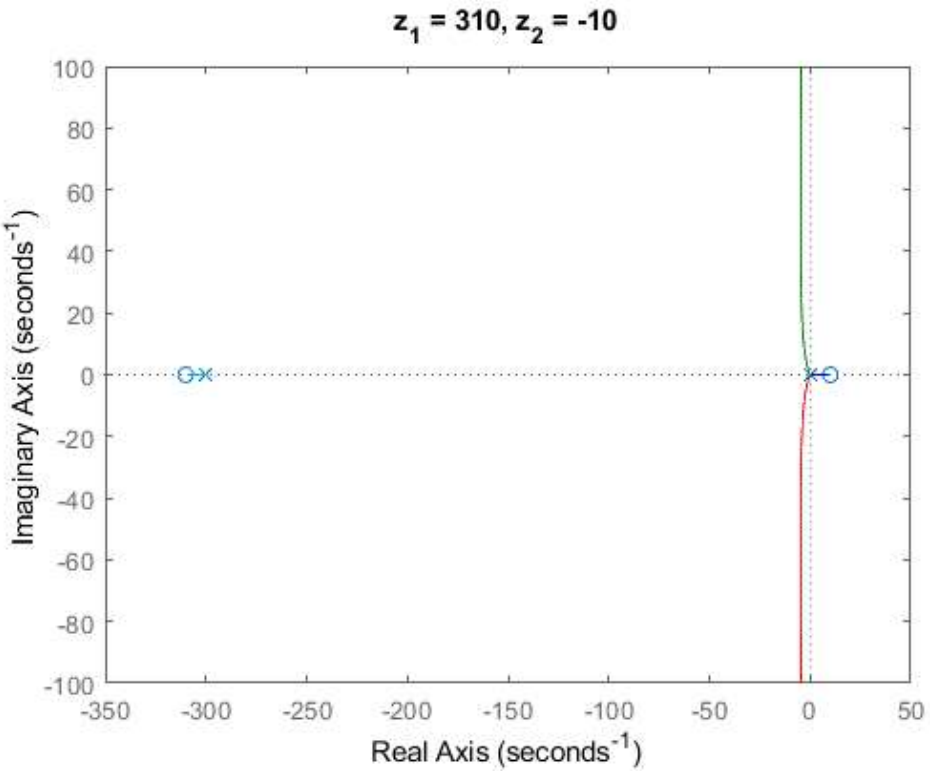

$z_1 = 310, z_2 = 290$
2000
1500
1000
500
0
-500
-1000
-1500
-2000
-400
-300
-200
-100
0
100
200
Imaginary Axis (seconds^{-1})
Real Axis (seconds^{-1})

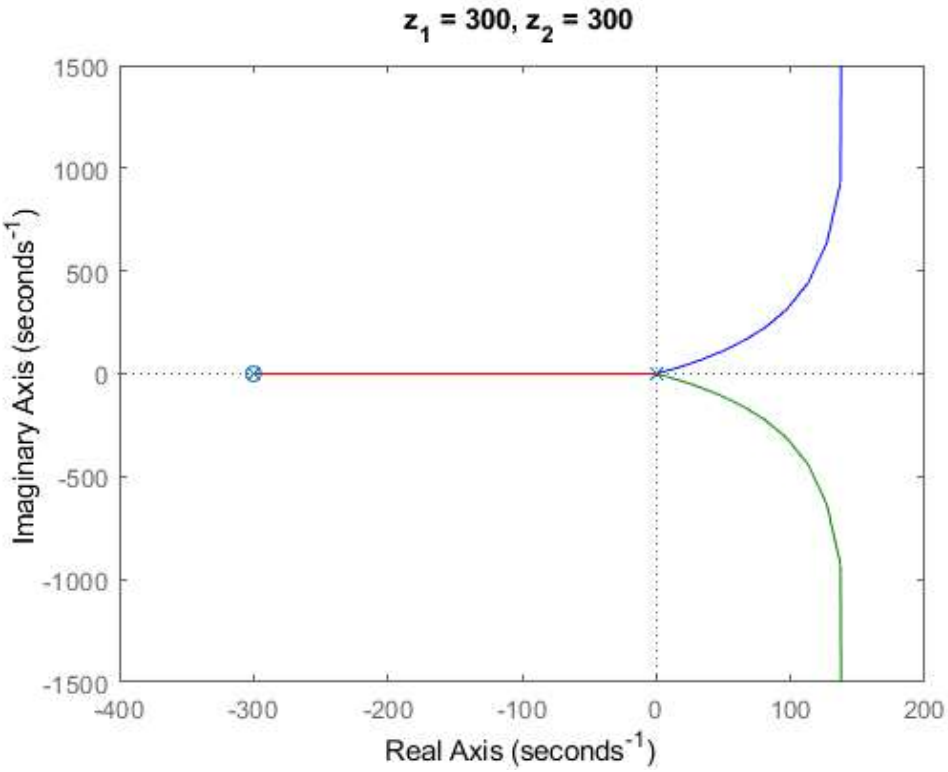

$z_1 = 310, z_2 = 150$
1000
800
600
400
200
0
-200
-400
-600
-800
-1000
-350
-300
-250
-200
-150
-100
-50
0
50
100
Imaginary Axis (seconds^{-1})
Real Axis (seconds^{-1})

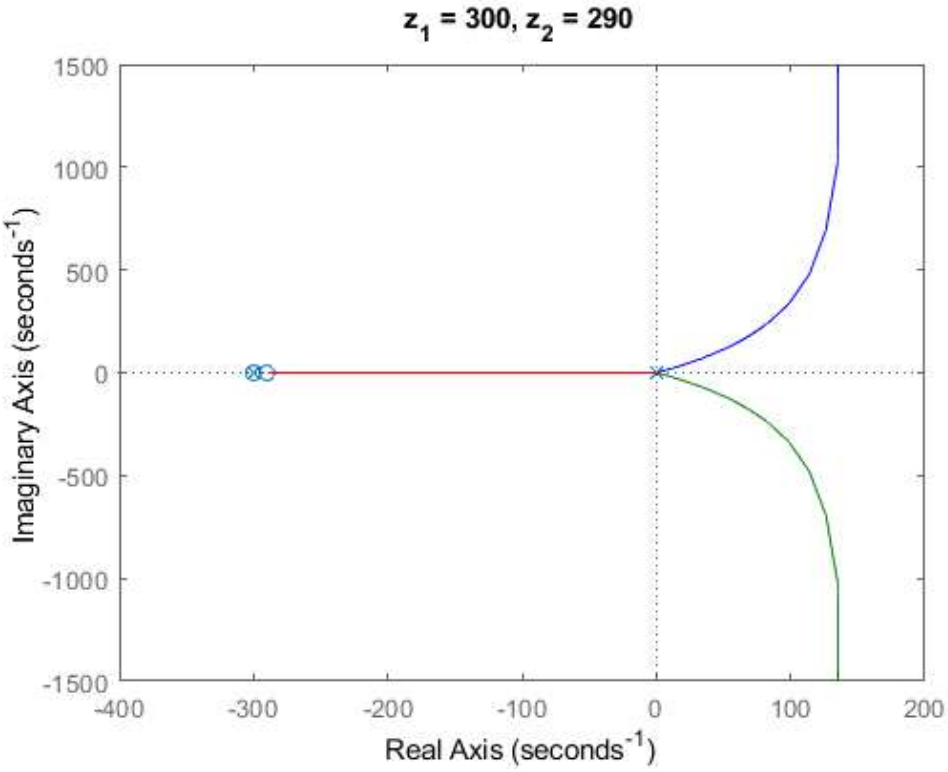

$z_1 = 310, z_2 = 33.3333$
Imaginary Axis (seconds^{-1})
Real Axis (seconds^{-1})
250
200
150
100
50
0
-50
-100
-150
-200
-250
-350
-300
-250
-200
-150
-100
-50
0
50

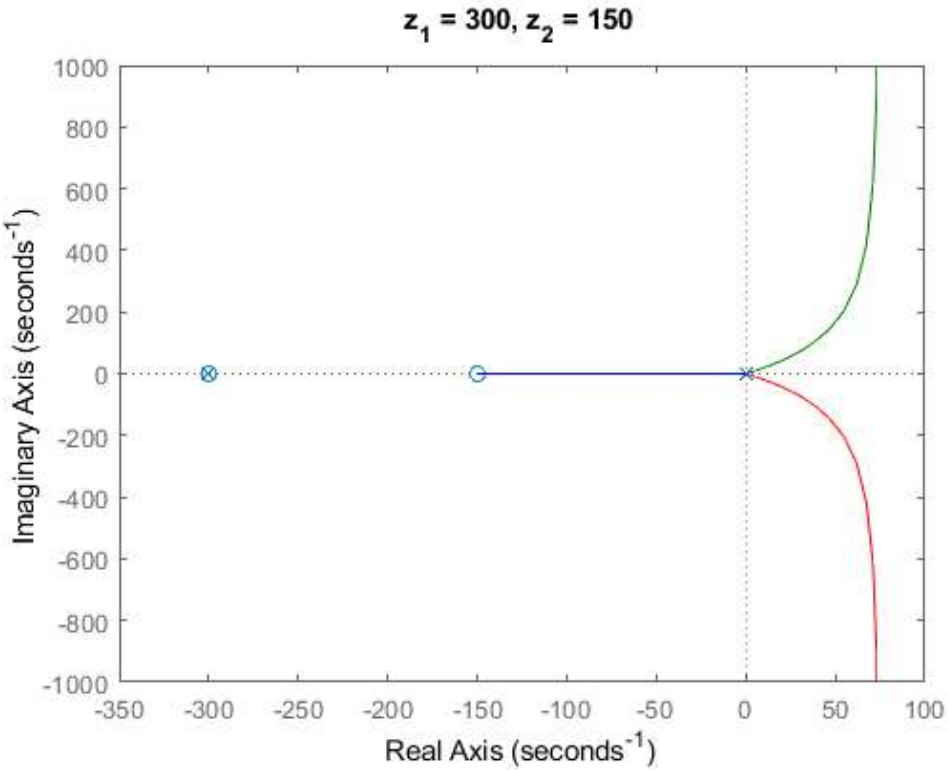

$z_1 = 310, z_2 = 10$
Imaginary Axis (seconds^{-1})
Real Axis (seconds^{-1})
100
80
60
40
20
0
-20
-40
-60
-80
-100
-350
-300
-250
-200
-150
-100
-50
0
50

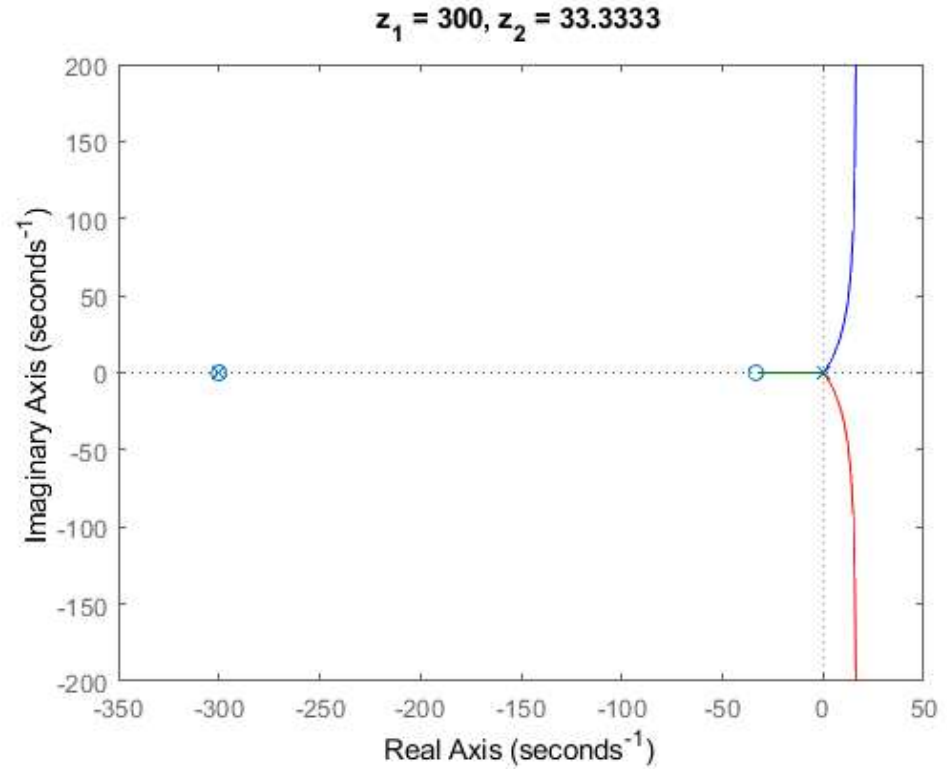

$z_1 = 310, z_2 = 0$
200
150
100
50
0
-50
-100
-150
-200
-350
-300
-250
-200
-150
-100
-50
0
50
Imaginary Axis (seconds^{-1})
Real Axis (seconds^{-1})

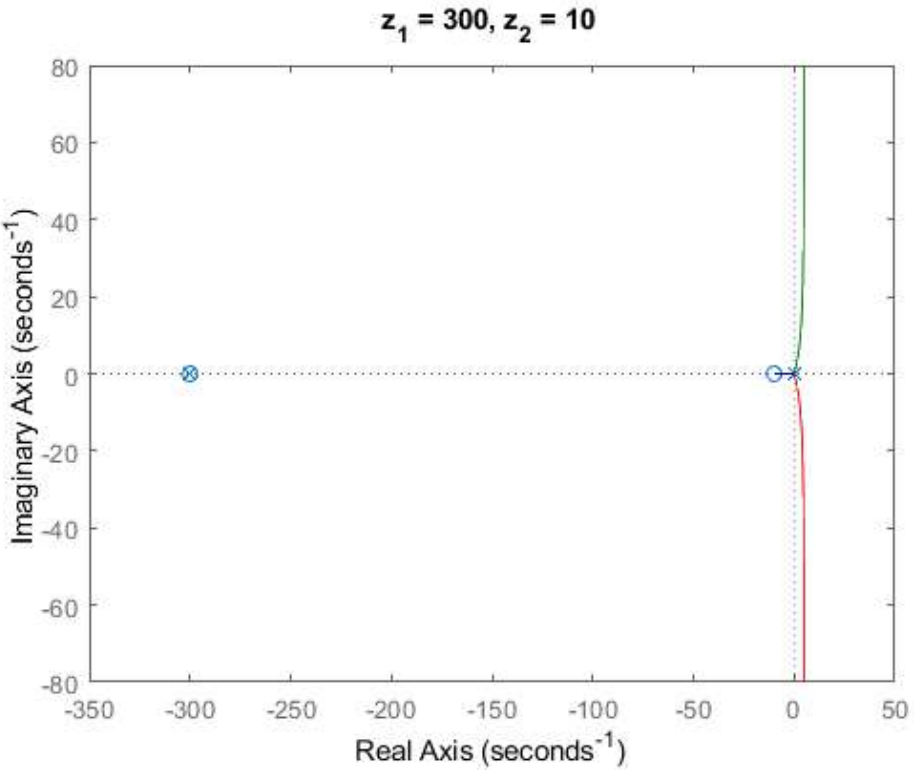

$z_1 = 310, z_2 = -10$
100
80
60
40
20
0
-20
-40
-60
-80
-100
-350
-300
-250
-200
-150
-100
-50
0
50
Imaginary Axis (seconds^{-1})
Real Axis (seconds^{-1})

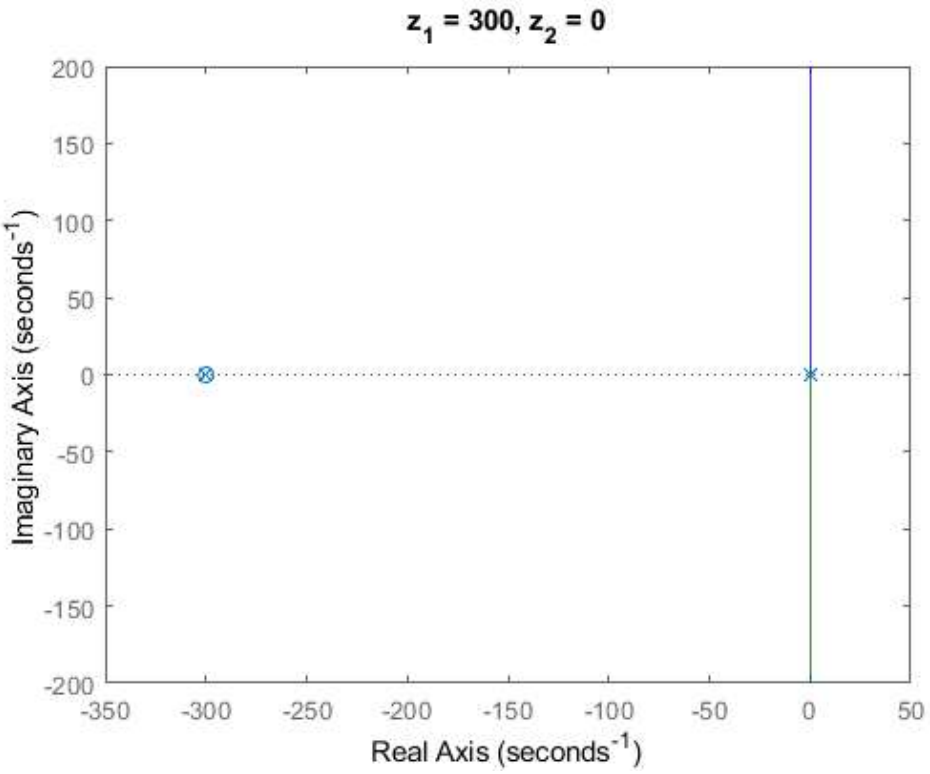

$z_1 = 300, z_2 = 300$
1500
1000
500
0
-500
-1000
-1500
-400
-300
-200
-100
0
100
200
Imaginary Axis (seconds^{-1})
Real Axis (seconds^{-1})

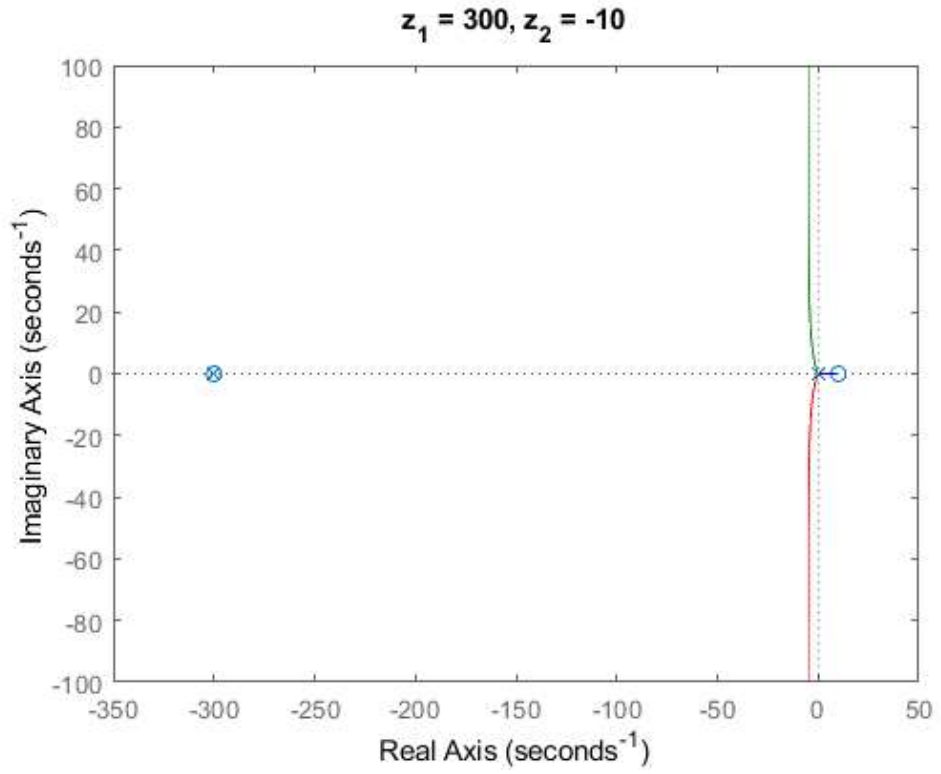

$z_1 = 300, z_2 = 290$
1500
1000
500
0
-500
-1000
-1500
-400
-300
-200
-100
0
100
200
Imaginary Axis (seconds^{-1})
Real Axis (seconds^{-1})

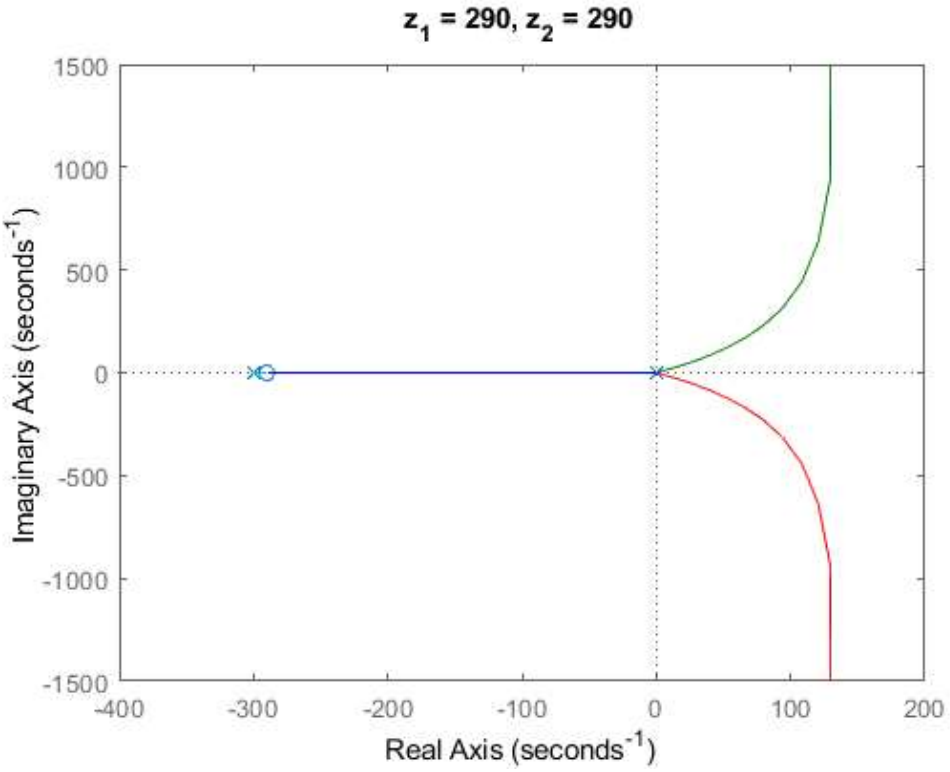

$z_1 = 300, z_2 = 150$
1000
800
600
400
200
0
-200
-400
-600
-800
-1000
-350
-300
-250
-200
-150
-100
-50
0
50
100
Imaginary Axis (seconds^{-1})
Real Axis (seconds^{-1})

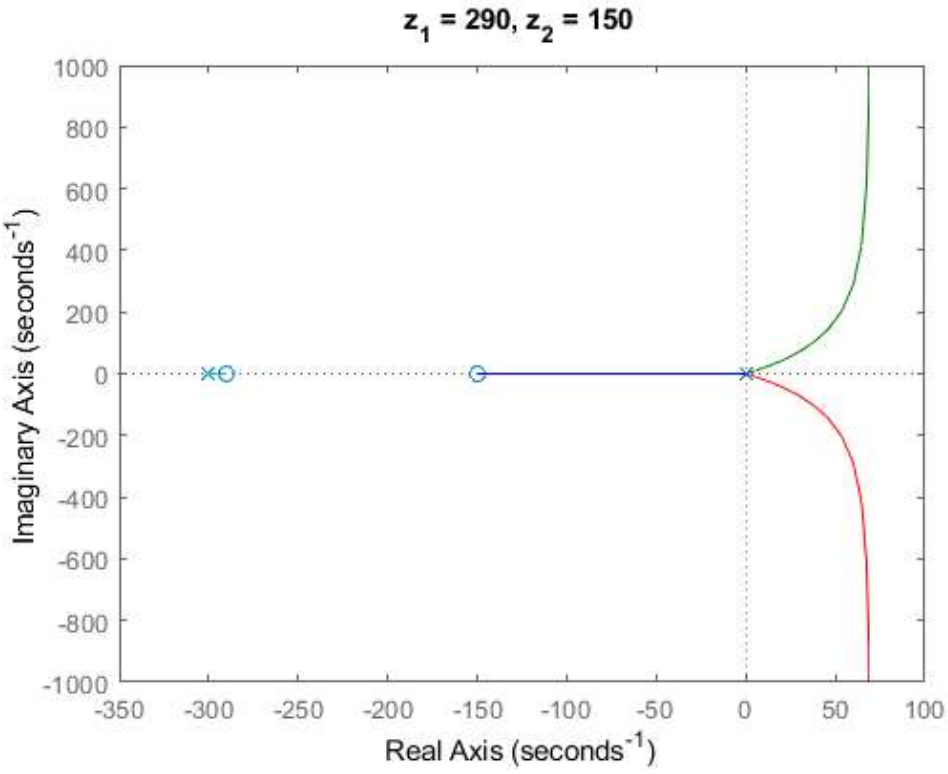

$z_1 = 300, z_2 = 33.3333$
200
150
100
50
0
-50
-100
-150
-200
-350
-300
-250
-200
-150
-100
-50
0
50
Imaginary Axis (seconds^{-1})
Real Axis (seconds^{-1})

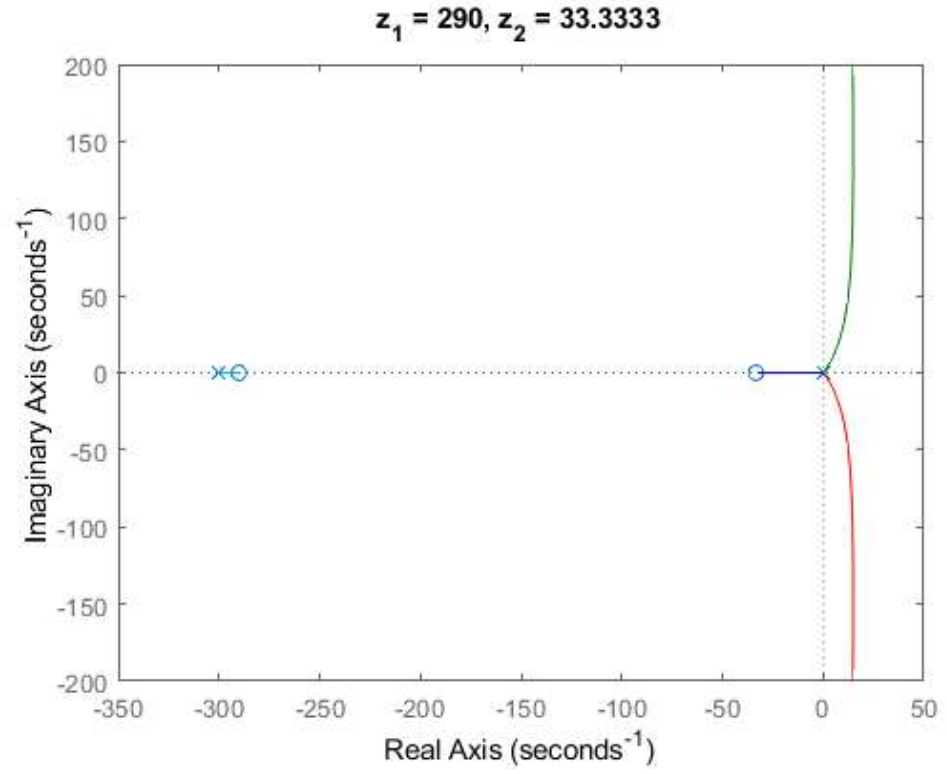

$z_1 = 300, z_2 = 10$
80
60
40
20
0
-20
-40
-60
-80
Imaginary Axis (seconds^{-1})
-350
-300
-250
-200
-150
-100
-50
0
50
Real Axis (seconds^{-1})

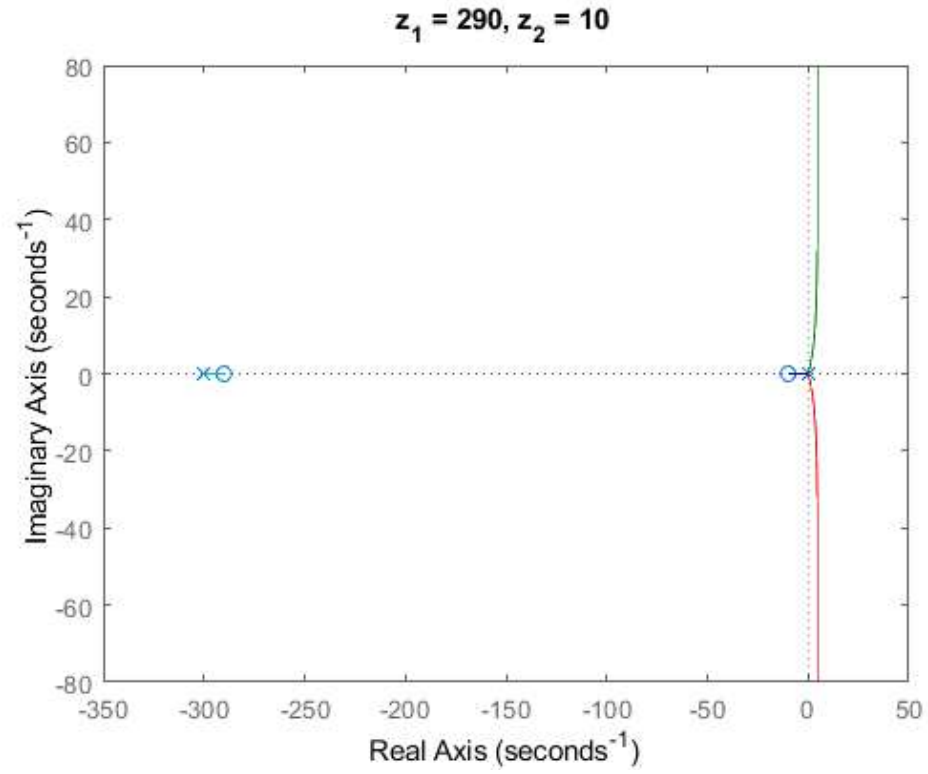

$z_1 = 300, z_2 = 0$
200
150
100
50
0
-50
-100
-150
-200
Imaginary Axis (seconds^{-1})
-350
-300
-250
-200
-150
-100
-50
0
50
Real Axis (seconds^{-1})

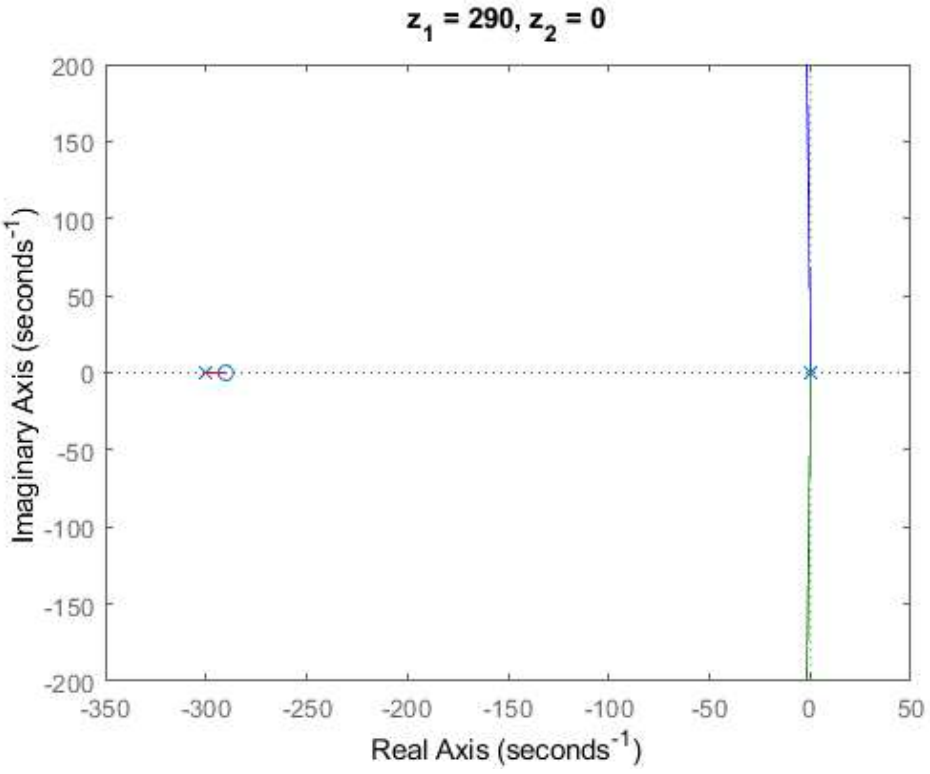

$z_1 = 300, z_2 = -10$
100
80
60
40
20
0
-20
-40
-60
-80
-100
-350
-300
-250
-200
-150
-100
-50
0
50
Imaginary Axis (seconds^{-1})
Real Axis (seconds^{-1})

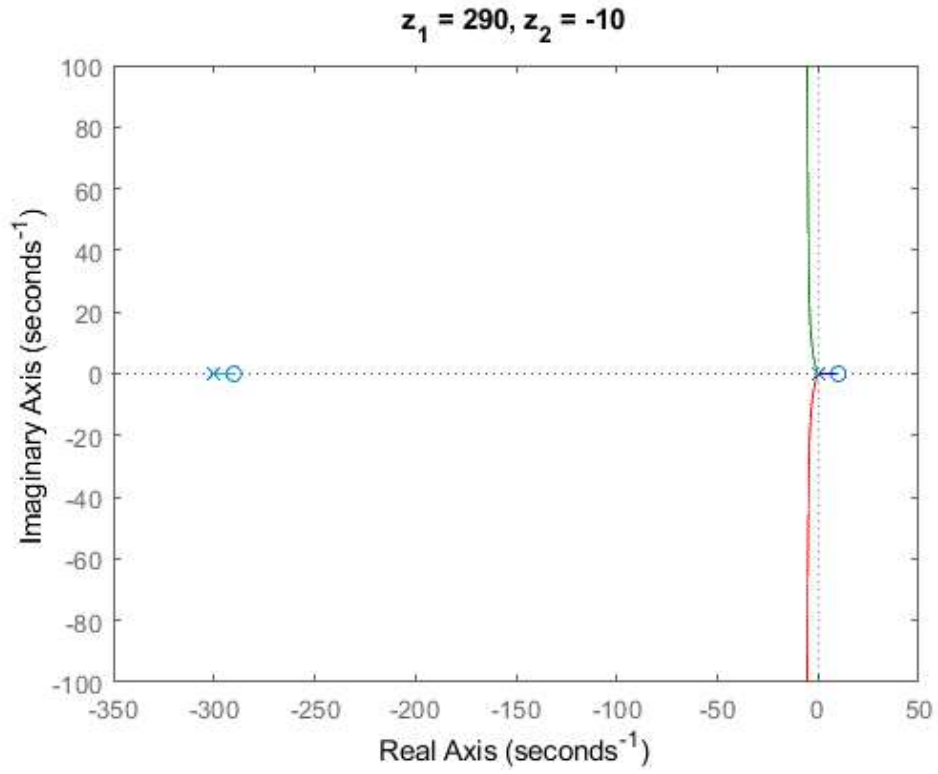

$z_1 = 290, z_2 = 290$
1500
1000
500
0
-500
-1000
-1500
-400
-300
-200
-100
0
100
200
Imaginary Axis (seconds^{-1})
Real Axis (seconds^{-1})

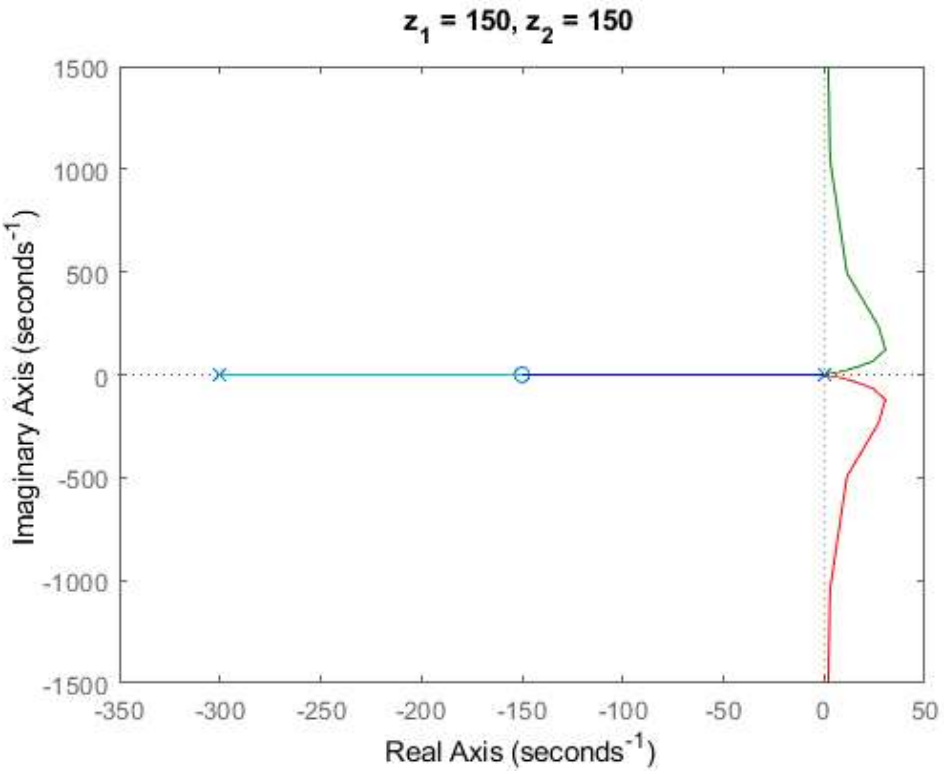

$z_1 = 290, z_2 = 150$
1000
800
600
400
200
0
-200
-400
-600
-800
-1000
-350
-300
-250
-200
-150
-100
-50
0
50
100
Imaginary Axis (seconds^{-1})
Real Axis (seconds^{-1})

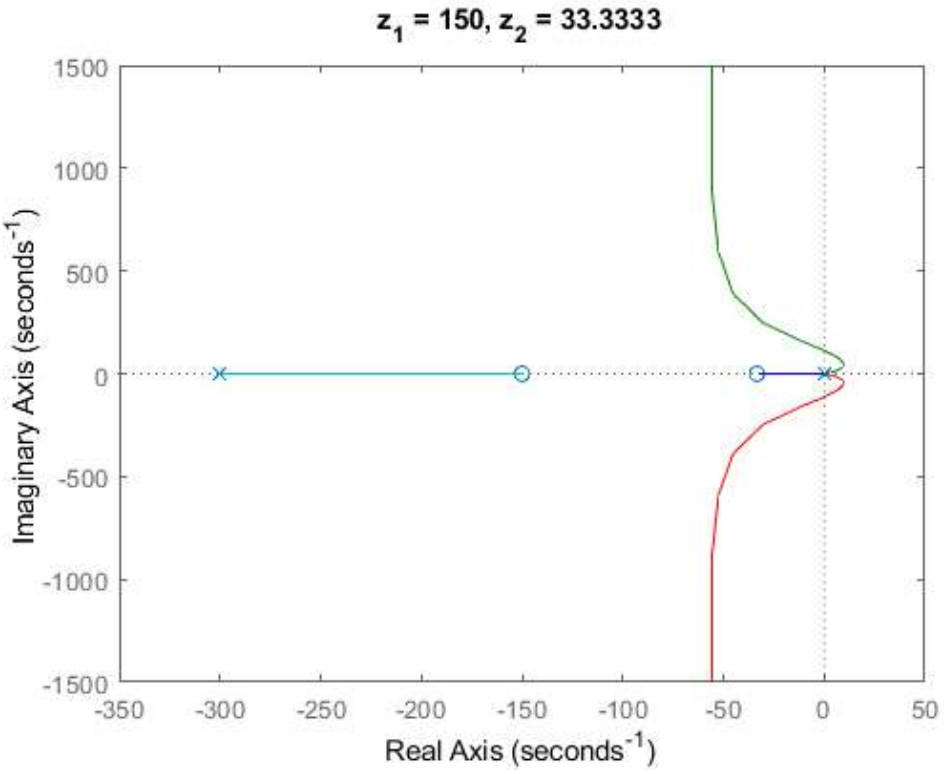

$z_1 = 290, z_2 = 33.3333$
200
150
100
50
0
-50
-100
-150
-200
-350
-300
-250
-200
-150
-100
-50
0
50
Imaginary Axis (seconds^{-1})
Real Axis (seconds^{-1})

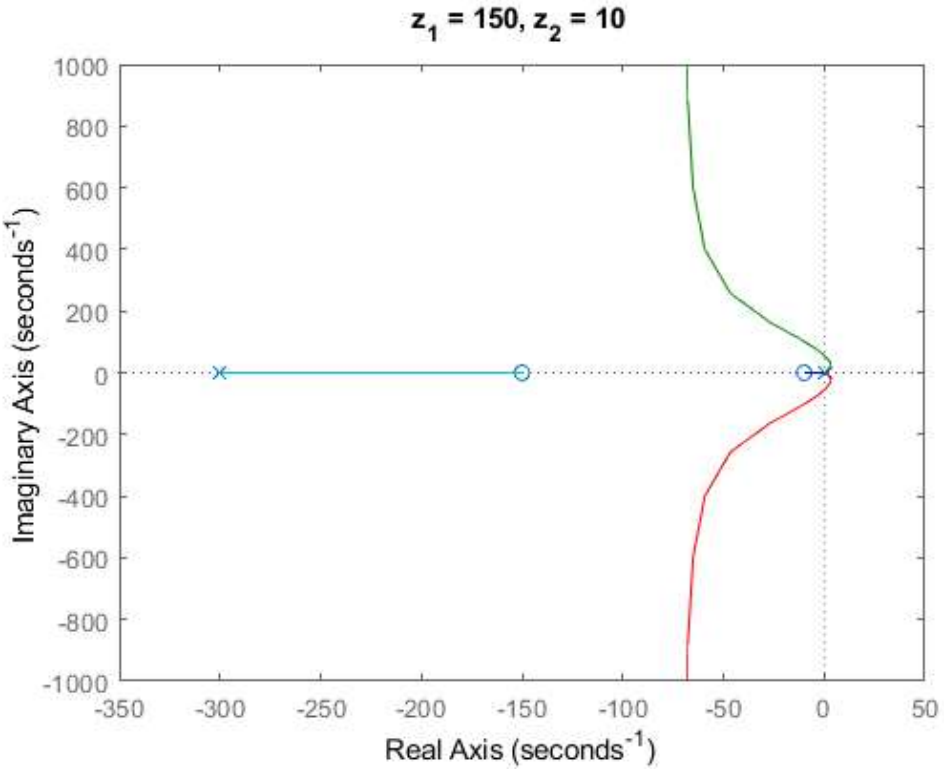

$z_1 = 290, z_2 = 10$
Imaginary Axis (seconds^{-1})
Real Axis (seconds^{-1})
80
60
40
20
0
-20
-40
-60
-80
-350
-300
-250
-200
-150
-100
-50
0
50

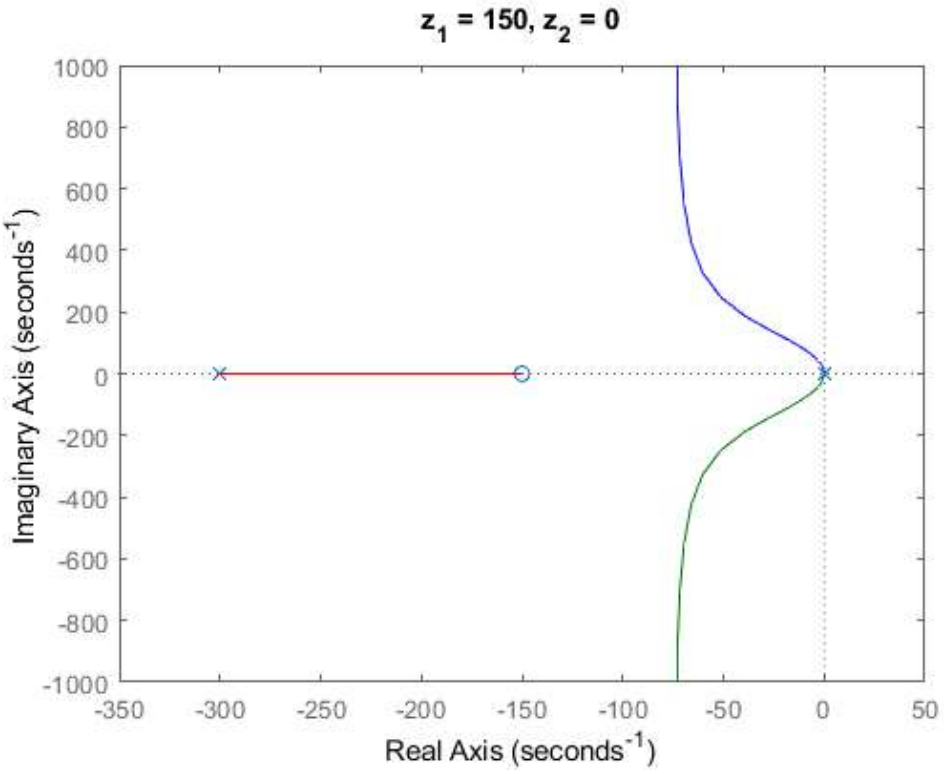

$z_1 = 290, z_2 = 0$
Imaginary Axis (seconds^{-1})
Real Axis (seconds^{-1})
200
150
100
50
0
-50
-100
-150
-200
-350
-300
-250
-200
-150
-100
-50
0
50

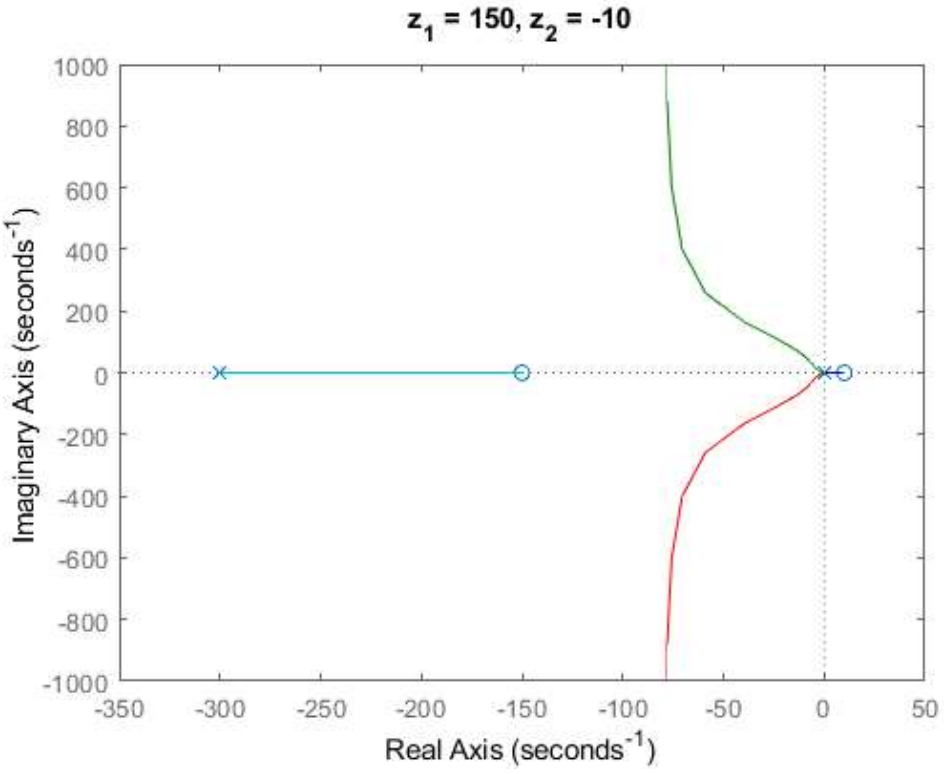

$z_1 = 290, z_2 = -10$
100
80
60
40
20
0
-20
-40
-60
-80
-100
-350
-300
-250
-200
-150
-100
-50
0
50
Imaginary Axis (seconds^{-1})
Real Axis (seconds^{-1})

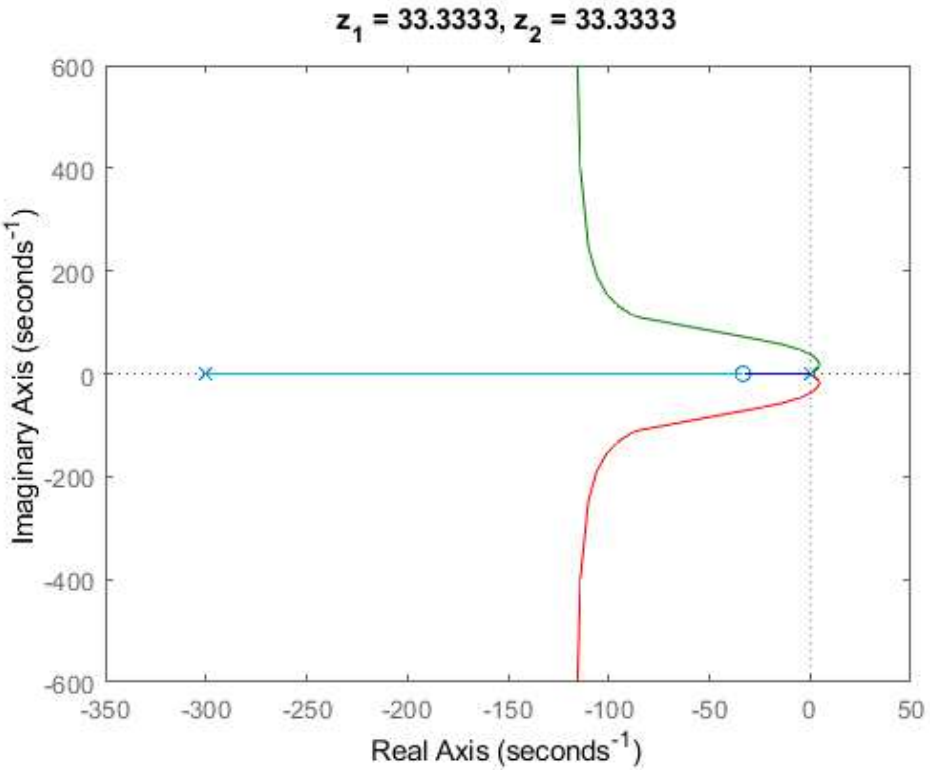

$z_1 = 150, z_2 = 150$
1500
1000
500
0
-500
-1000
-1500
-350
-300
-250
-200
-150
-100
-50
0
50
Imaginary Axis (seconds^{-1})
Real Axis (seconds^{-1})

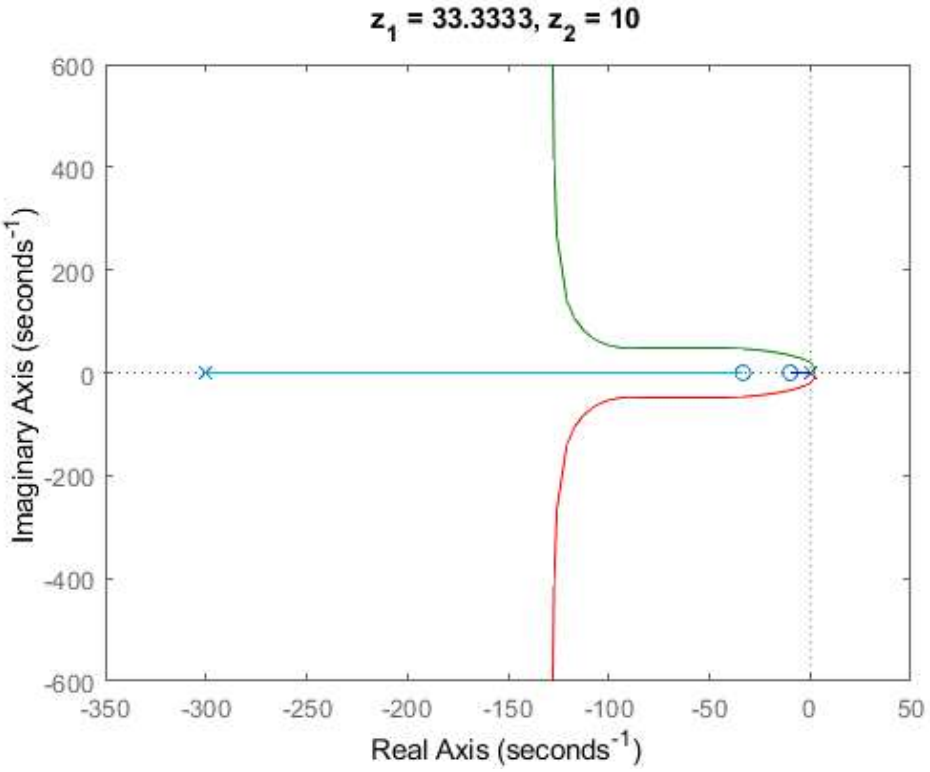

$z_1 = 150, z_2 = 33.3333$
1500
1000
500
0
-500
-1000
-1500
-350
-300
-250
-200
-150
-100
-50
0
50
Imaginary Axis (seconds^{-1})
Real Axis (seconds^{-1})

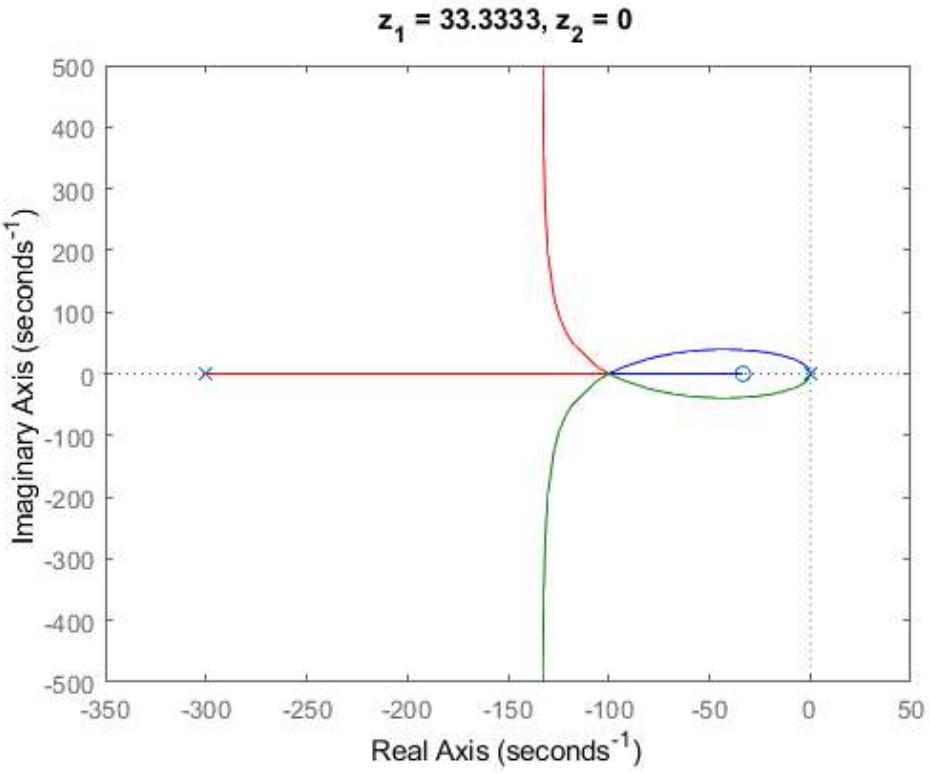

$z_1 = 150, z_2 = 10$
1000
800
600
400
200
0
-200
-400
-600
-800
-1000
-350
-300
-250
-200
-150
-100
-50
0
50
Imaginary Axis (seconds^{-1})
Real Axis (seconds^{-1})

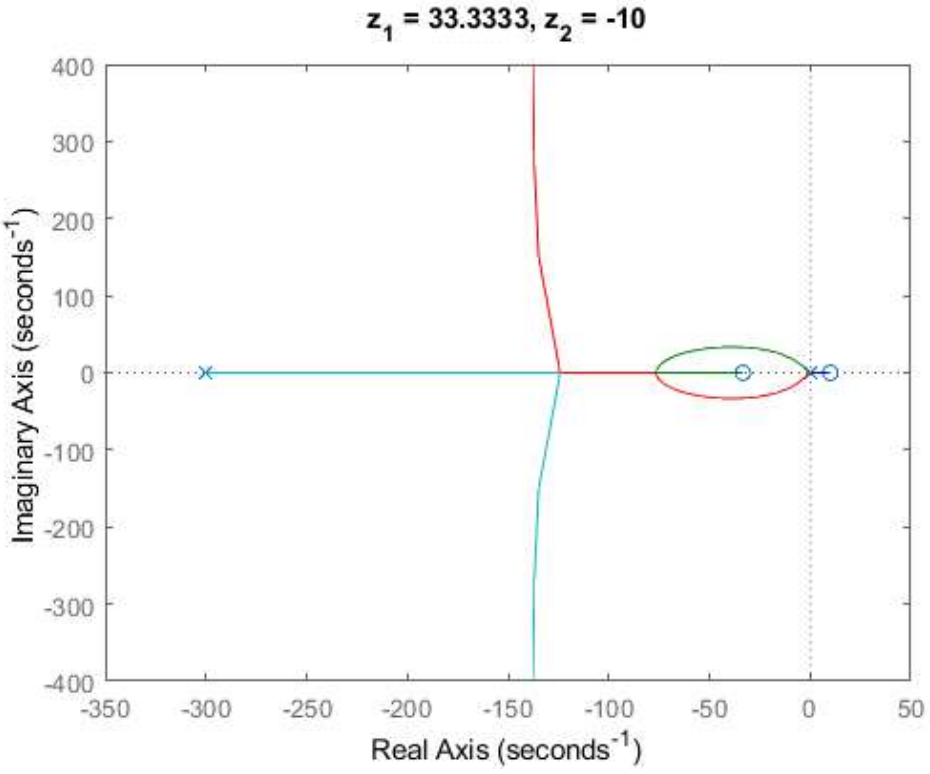

$z_1 = 150, z_2 = 0$
1000
800
600
400
200
0
-200
-400
-600
-800
-1000
-350
-300
-250
-200
-150
-100
-50
0
50
Imaginary Axis (seconds^{-1})
Real Axis (seconds^{-1})

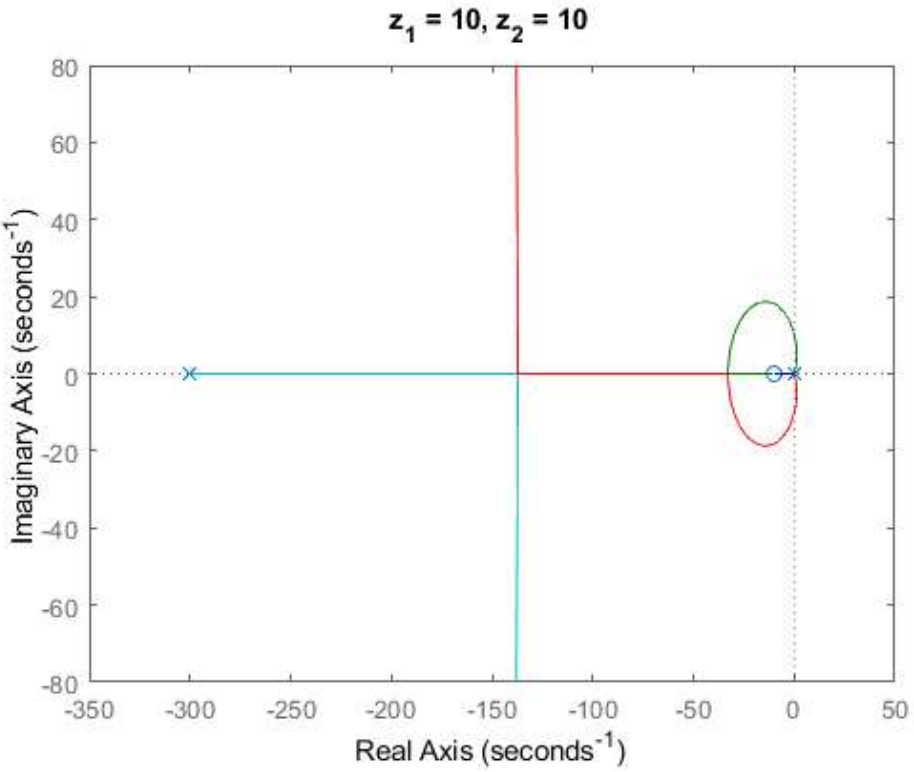

$z_1 = 150, z_2 = -10$
1000
800
600
400
200
0
-200
-400
-600
-800
-1000
-350
-300
-250
-200
-150
-100
-50
0
50
Imaginary Axis (seconds^{-1})
Real Axis (seconds^{-1})

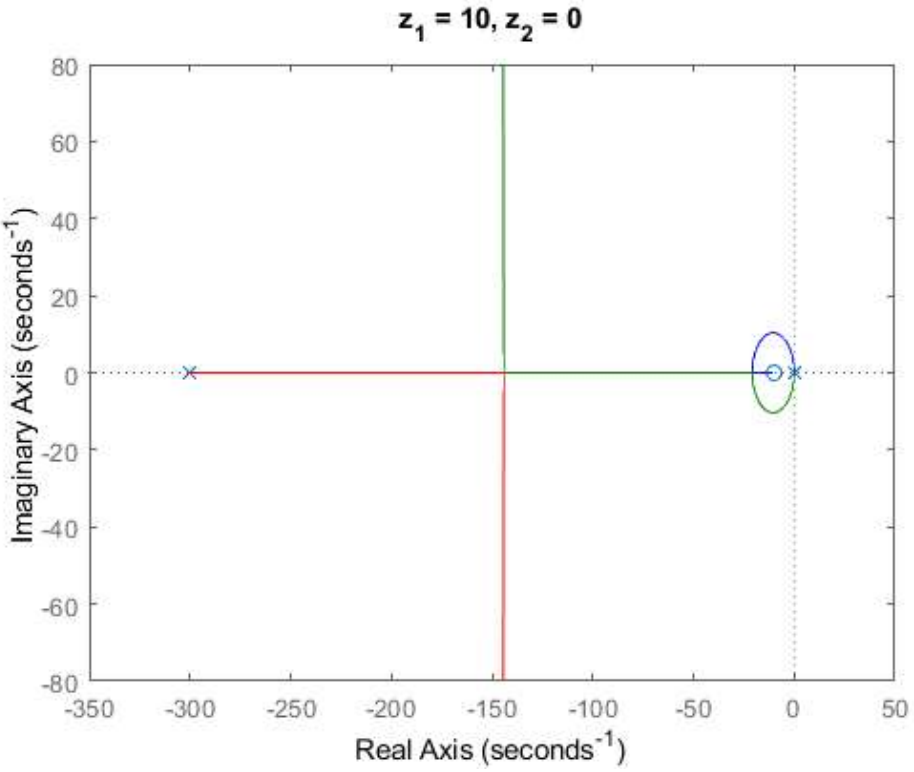

$z_1 = 33.3333, z_2 = 33.3333$
600
400
200
0
-200
-400
-600
-350
-300
-250
-200
-150
-100
-50
0
50
Imaginary Axis (seconds^{-1})
Real Axis (seconds^{-1})

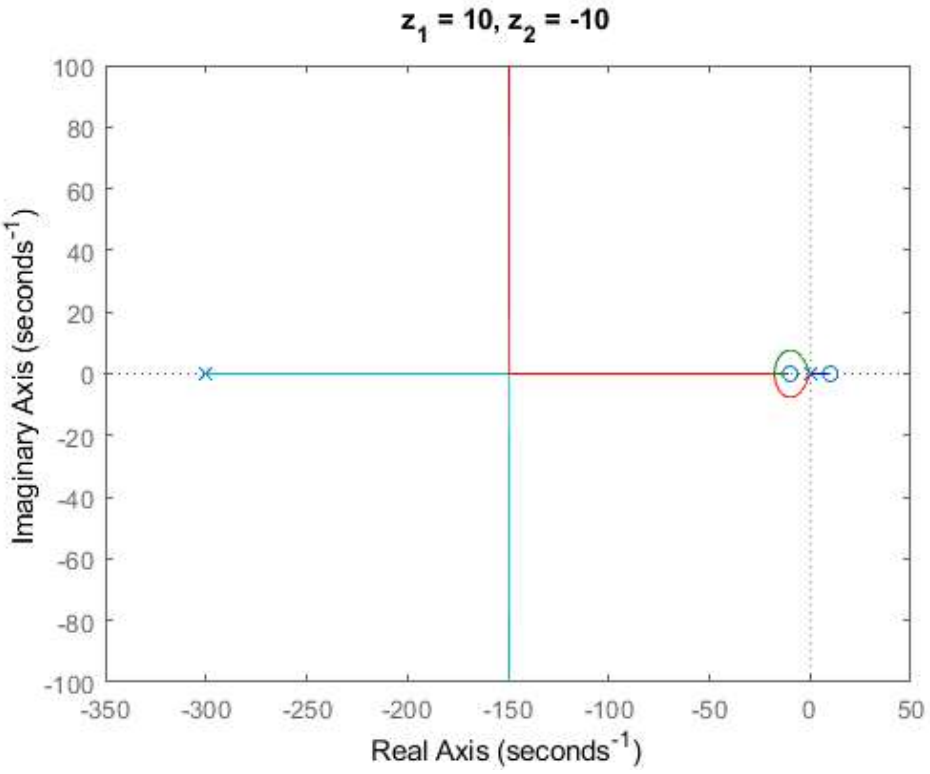

$z_1 = 33.3333, z_2 = 10$
600
400
200
0
-200
-400
-600
-350
-300
-250
-200
-150
-100
-50
0
50
Imaginary Axis (seconds^{-1})
Real Axis (seconds^{-1})

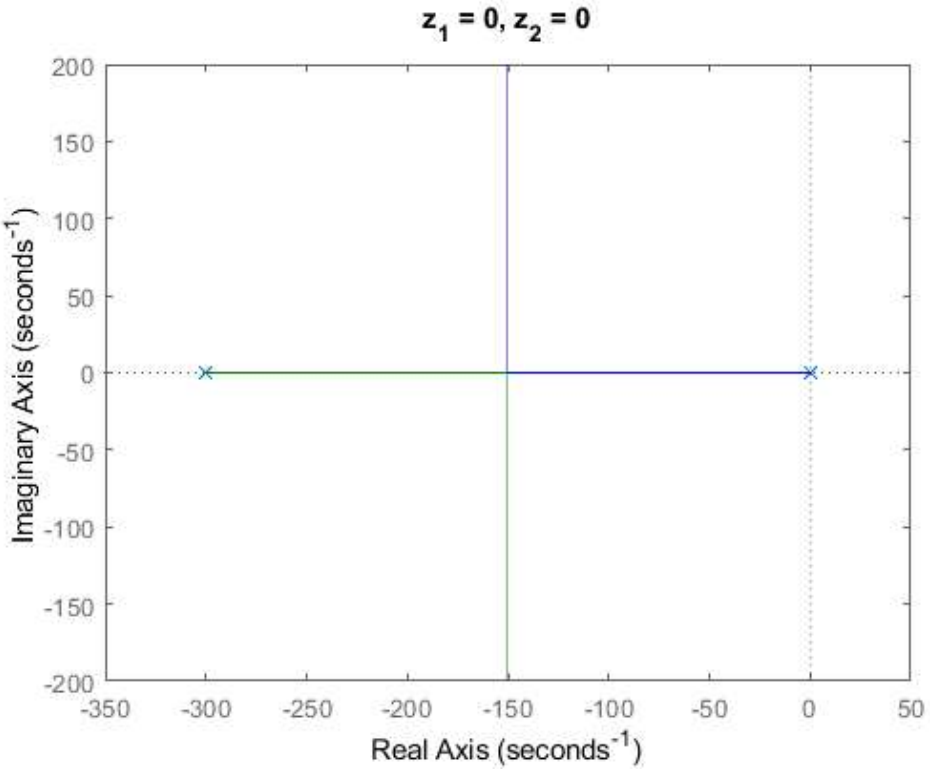

$z_1 = 33.3333, z_2 = 0$
500
400
300
200
100
0
-100
-200
-300
-400
-500
-350
-300
-250
-200
-150
-100
-50
0
50
Imaginary Axis (seconds^{-1})
Real Axis (seconds^{-1})

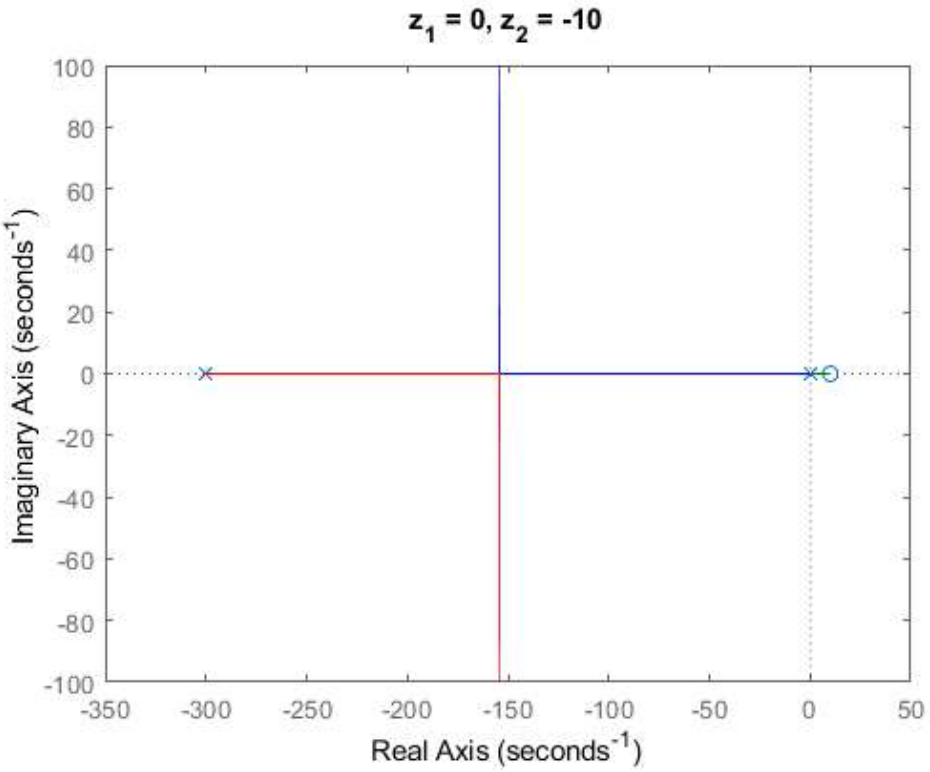

$z_1 = 33.3333, z_2 = -10$
400
300
200
100
0
-100
-200
-300
-400
-350
-300
-250
-200
-150
-100
-50
0
50
Imaginary Axis (seconds^{-1})
Real Axis (seconds^{-1})

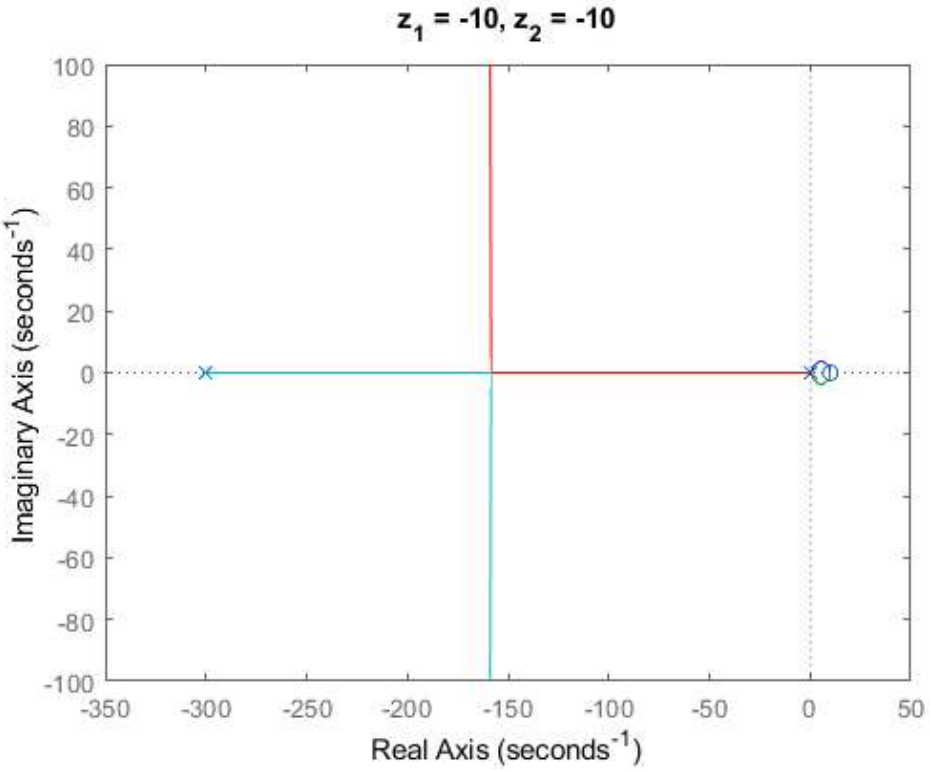

$z_1 = 10, z_2 = 10$
Imaginary Axis (seconds^{-1})
Real Axis (seconds^{-1})
80
60
40
20
0
-20
-40
-60
-80
-350
-300
-250
-200
-150
-100
-50
0
50


$z_1 = 10, z_2 = 0$
Imaginary Axis (seconds^{-1})
Real Axis (seconds^{-1})
80
60
40
20
0
-20
-40
-60
-80
-350
-300
-250
-200
-150
-100
-50
0
50


$z_1 = 10, z_2 = -10$
Imaginary Axis (seconds^{-1})
Real Axis (seconds^{-1})
100
80
60
40
20
0
-20
-40
-60
-80
-100
-350
-300
-250
-200
-150
-100
-50
0
50


$z_1 = 0, z_2 = 0$
Imaginary Axis (seconds^{-1})
Real Axis (seconds^{-1})
200
150
100
50
0
-50
-100
-150
-200
-350
-300
-250
-200
-150
-100
-50
0
50





It can be observed that whenever one of the zeros is on the right real plane or is less than -300, there is always a positive pole. This means that all of these systems will not converge to any meaningful solution.

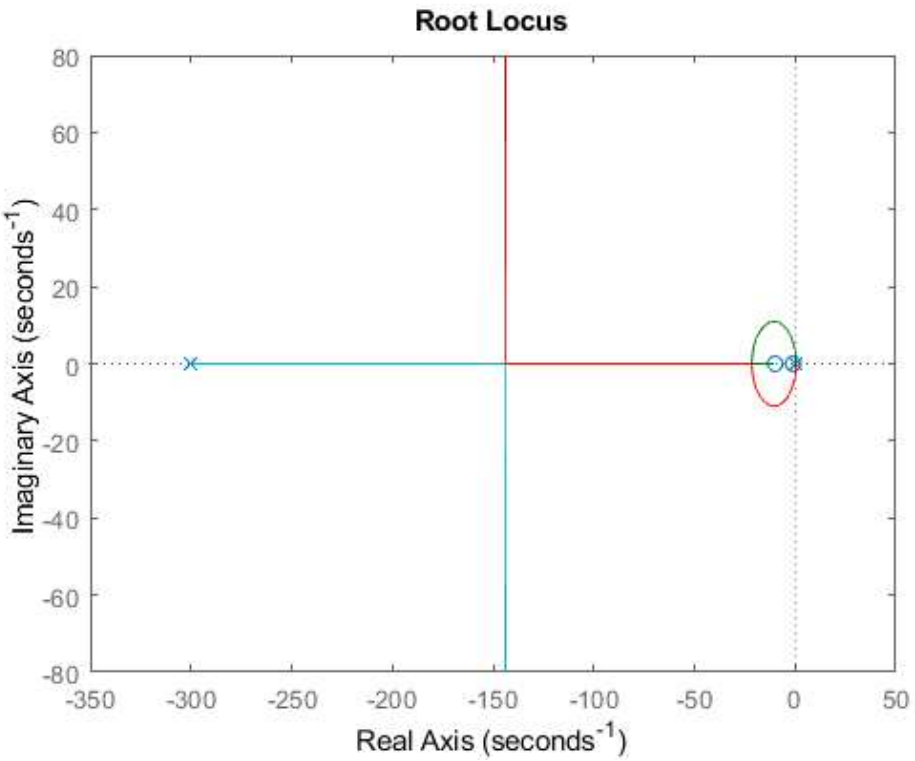
Additionaly, it can be concluded that the system only converges to a solution of medium to high values of a gain when one of the zeros is close to the origin. Otherwise, if the gain is too small, the poles will have their real part in the right real plane.

It may be possible for the system to reach a solution without oscilation, being z1 = 0 & z2 = 0 the best example of such case.

One last thing to notice is that for zeros 0 and -300, the system will behave like a PD controller. This happens because the transfer function will have equal scalar terms in its numerator and denominator.

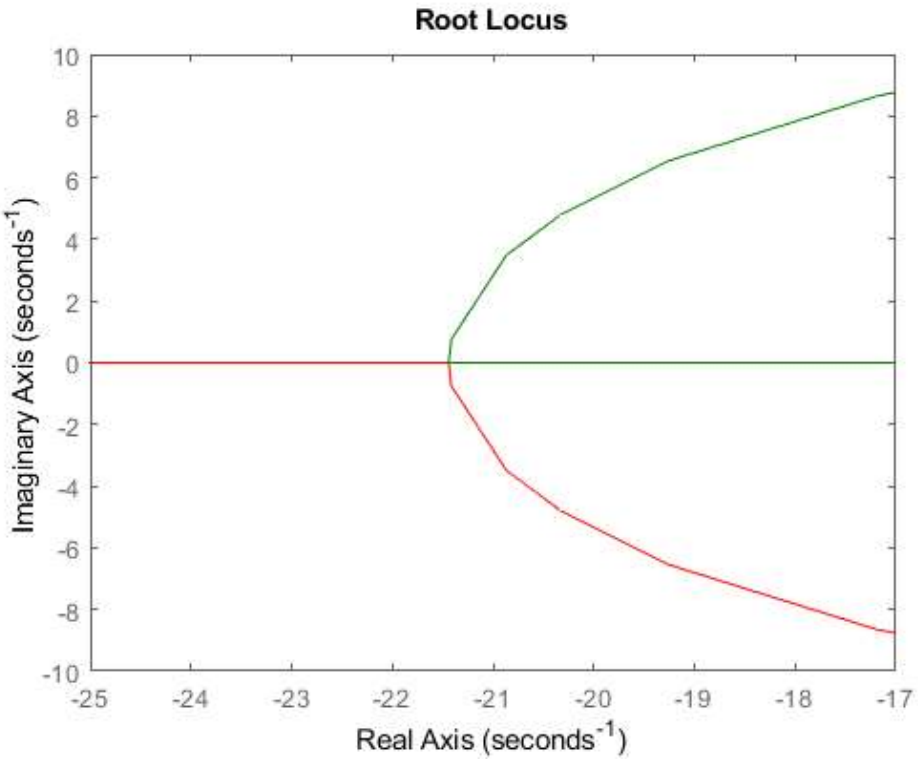
Q4.3

```
z1 = 1;
z2 = 10;

rlocus(tf([1 (z1+z2) z1*z2],[1 300 0 0 0]));
```



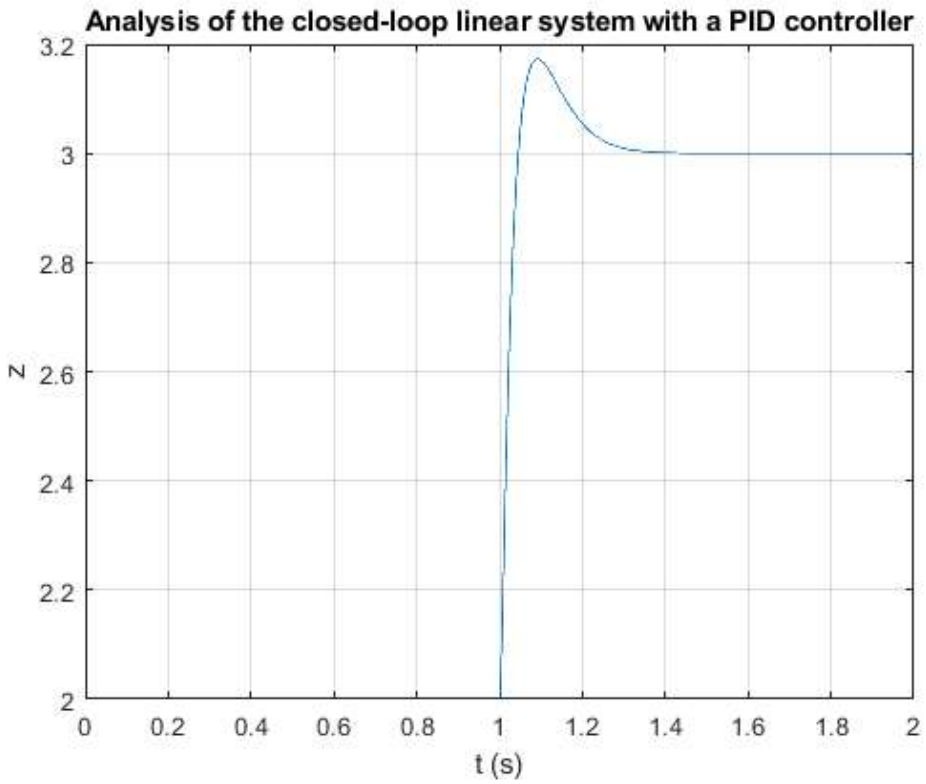
The desired gain can easily be obtained by clicking on the intersection between the poles trajectory around $s = -21$:

```
rlocus(tf([1 (z1+z2) z1*z2],[1 300 0 0 0]));

xlim([-25 -17]);
ylim([-10 10]);

hold off;
```



We get $K = 1.17E4$.

```
K = 1.17E4;
z1 = 1;
z2 = 10;

K_d = (K*M)/(600*K_t*omega0);
K_p = K_d*(z1 + z2);
K_i = K_d*z1*z2;
SimTime = 2;
delta_zr = 1;
StepSize = 1E-4;

        figure;
        sim('lab_4.slx');
        time_array(:, 1) = out.time;
        pid_values_array(:, 1) = out.signals.values(:,1);
        hold on; title("Analysis of the closed-loop linear system with a PID controller");
        plot(time_array, pid_values_array(:, 1));
        xlabel('t (s)');
        ylabel('z');
        grid on;
        box on;

hold off;
```

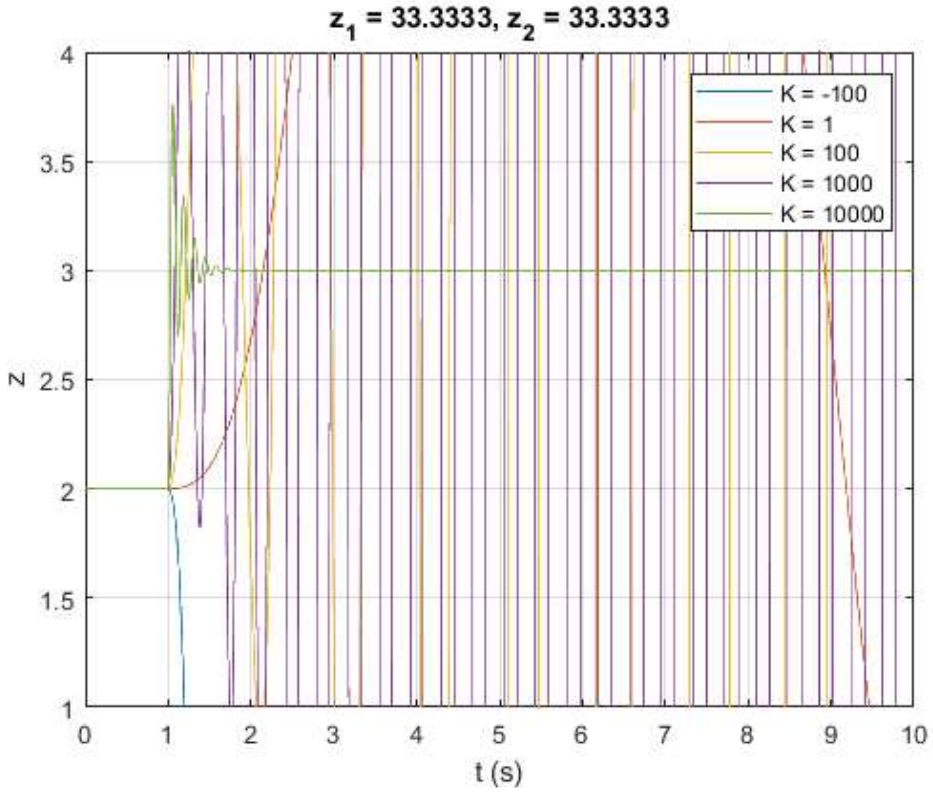


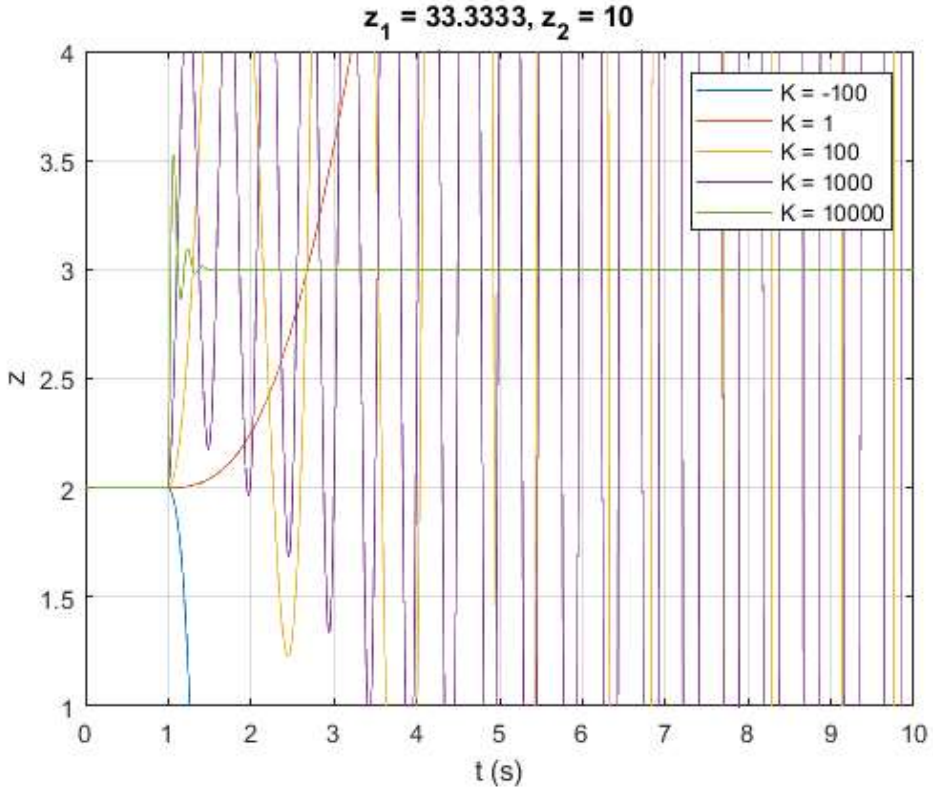
We can see an overshoot happen. This is a result of the integration that characterizes PID controllers. Eitherway, the system is able to converge to a solution. The transient is quickly removed, showing that the system is stable.

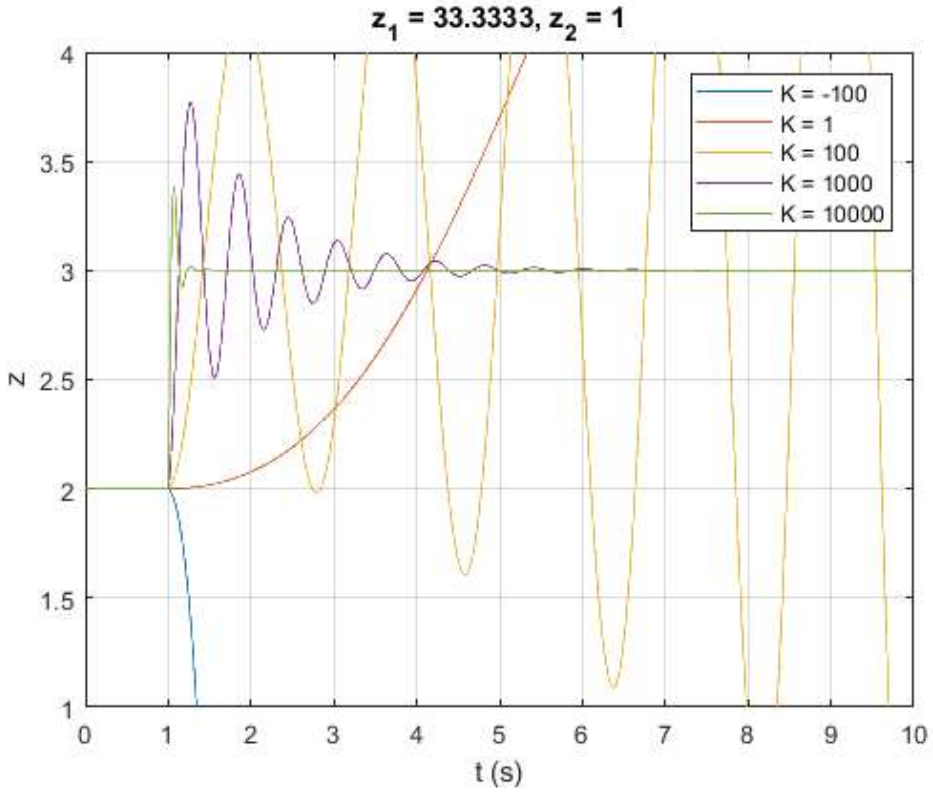
Q4.4

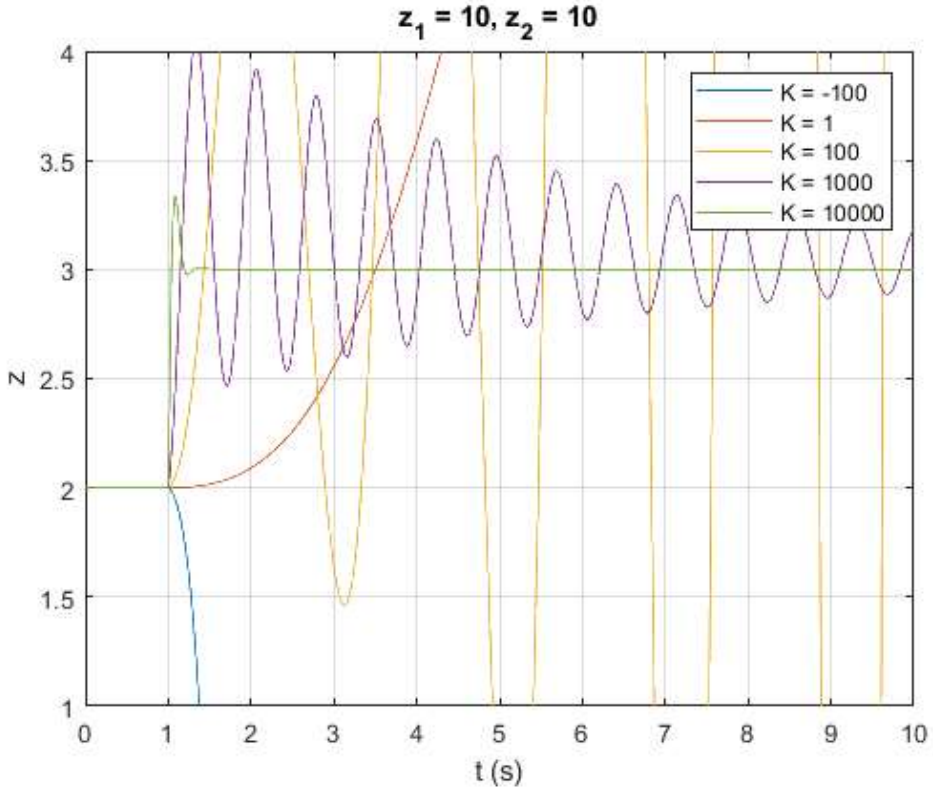
```
clear all; hold off;

G = 9.8; %m s^-2
M = 1; %kg
K_t = 3.575E-5; % N s^2 rad^-2
Z0 = 2; %m
omega0 = sqrt(G*M/K_t); %rad s^-1

SimTime = 10;
delta_zr = 1;
StepSize = 1E-4;

z_4 = 150;  %between
z_5 = 100/3;%between
z_6 = 10;   %between and near
z_7 = 1;    %root mult. 2

z_array = [z_5, z_6, z_7];
K_array = [-100, 1, 100, 1000, 10000];

for i=1:length(z_array)
    for j=i:length(z_array)
        figure;
        for k=1:length(K_array)
            z1 = z_array(i);
            z2 = z_array(j);
            K = K_array(k);
            K_d = (K*M)/(600*K_t*omega0);
            K_p = K_d*(z1 + z2);
```

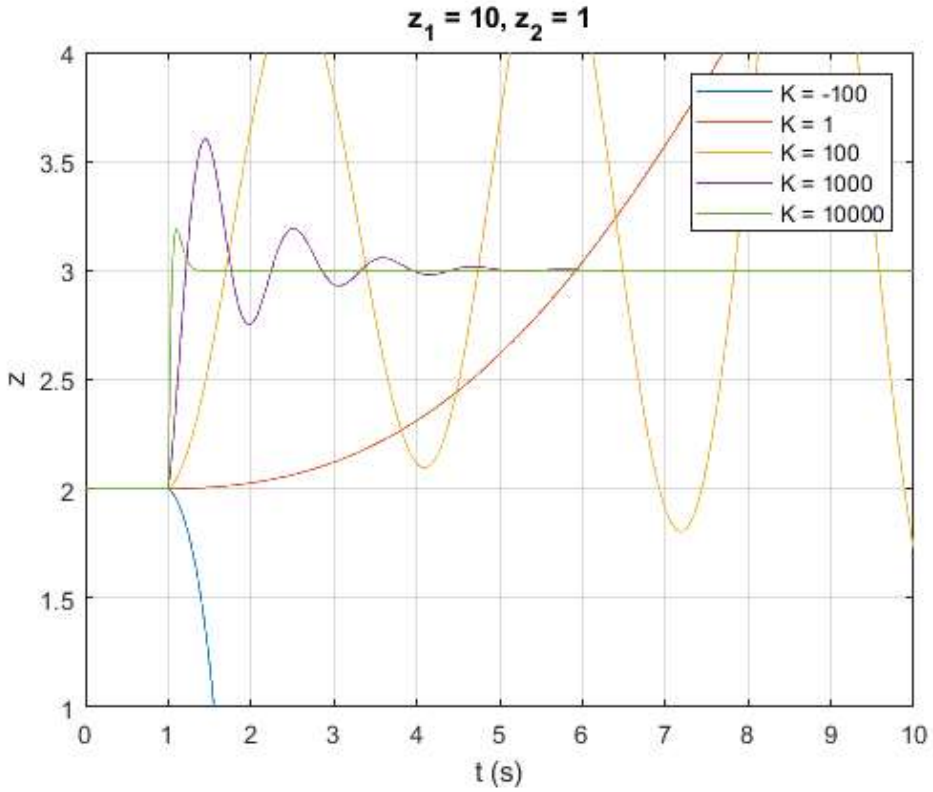
```
            K_i = K_d*z1*z2;
            sim('lab_4.slx');
            time_array(:, 1) = out.time;
            pid_values_array(:, 1) = out.signals.values(:,1);
            hold on; title("z_1 = " + num2str(z1) + ", z_2 = " + num2str(z2));
            xlim([0 SimTime]);
            ylim([1 4]);
            plot(time_array, pid_values_array(:, 1));
            xlabel('t (s)');
            ylabel('z');
            grid on;
            box on;
        end
        legend('K = -100', 'K = 1', 'K = 100', 'K = 1000', 'K = 10000');
    end
end
```

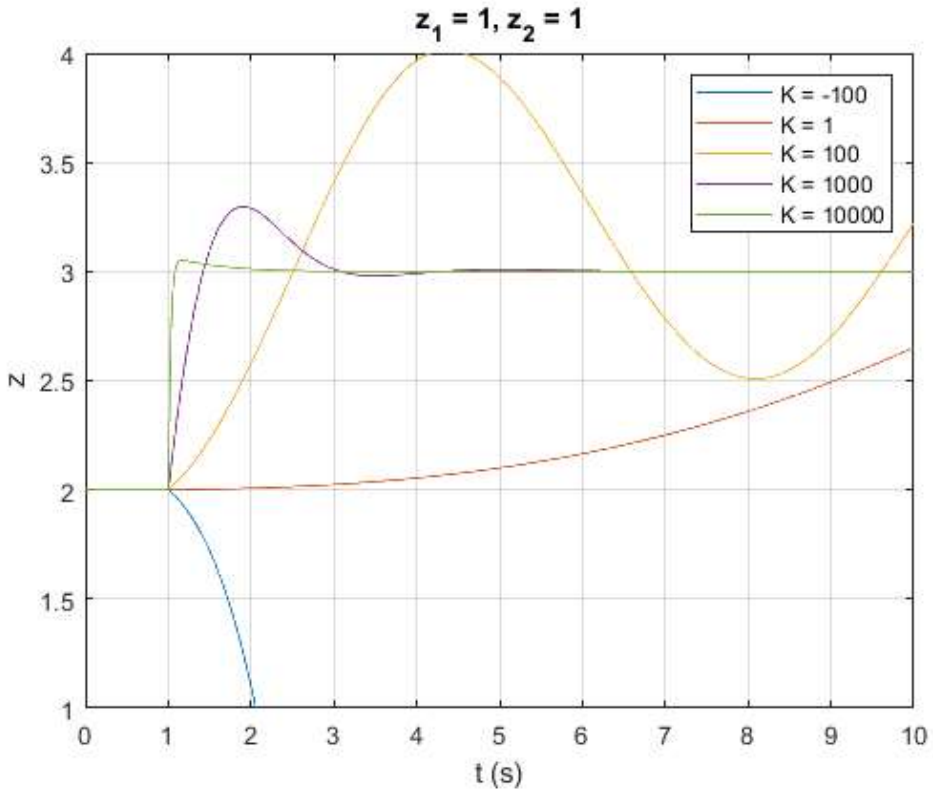

$z_1 = 33.3333, z_2 = 33.3333$
K = -100
K = 1
K = 100
K = 1000
K = 10000
z
t (s)


$z_1 = 33.3333, z_2 = 10$
K = -100
K = 1
K = 100
K = 1000
K = 10000
z
t (s)


$z_1 = 33.3333, z_2 = 1$
K = -100
K = 1
K = 100
K = 1000
K = 10000
z
t (s)


$z_1 = 10, z_2 = 10$
K = -100
K = 1
K = 100
K = 1000
K = 10000
z
t (s)



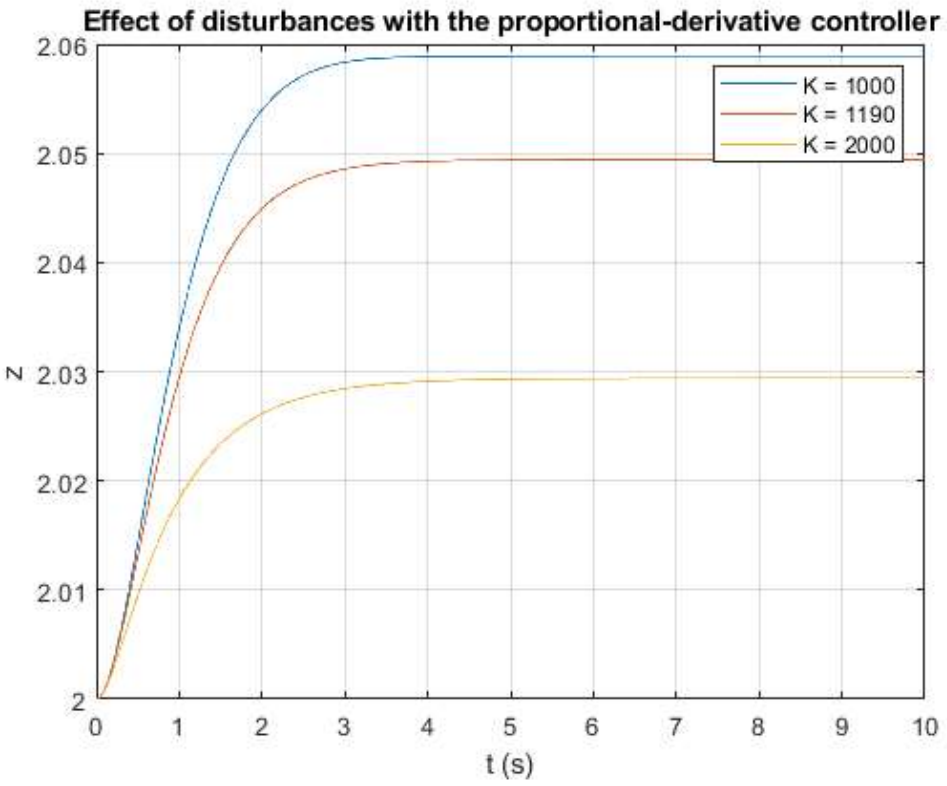


Looking back at what was discussed in section 4.2, it can be seen that the inital predictions are true. In some of the step responses, convergence is only obtained for high values of gain. Small values lead to divergence of the system. For zeros very distant from 0, none of the systems converge and, for small values, either gain, as long as positive, allows convergence.

It can also be seen that, in case of cnvergence, the higher the gain, the faster the system converges to a solution and, if its the case, the higher the gain will also imply a smaller overshoot.

Q5.4

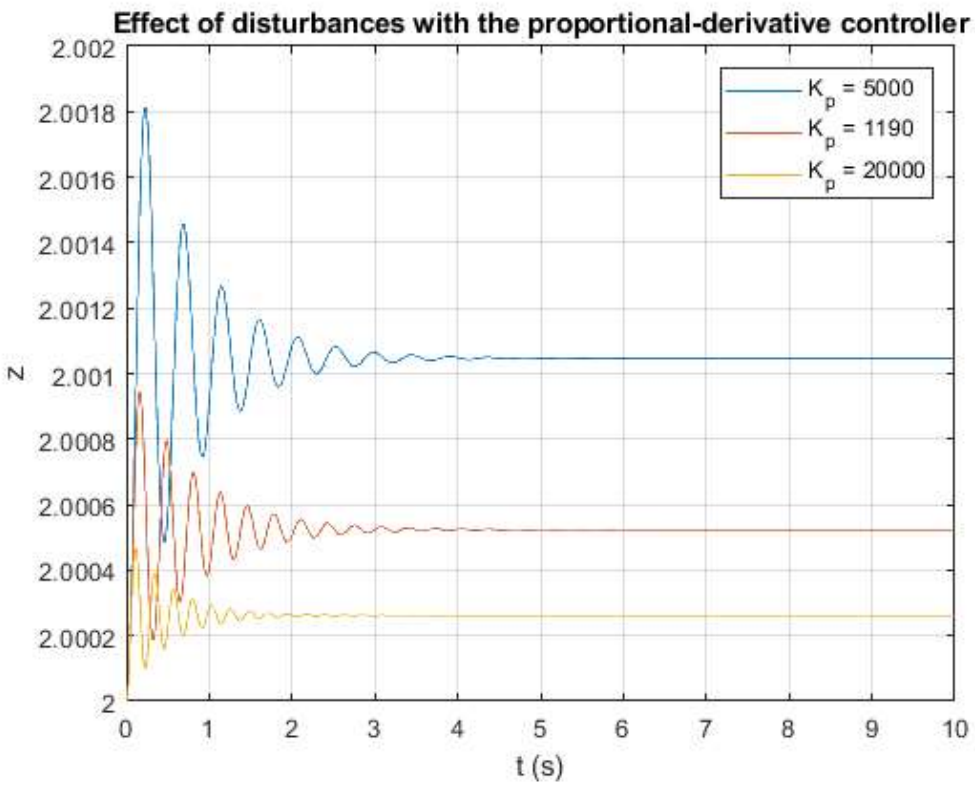
```
w_d = 50 * 2 * pi / 60;
delta_zr = 0;
K_array = [1000, 1.19E3, 2000];
z = 1;
SimTime = 10;
StepSize = 1E-4;

figure; hold on, title("Effect of disturbances with the proportional-derivative controller");
for i=1:length(K_array)
        K = K_array(i);
        K_d = (K*M)/(600*K_t*omega0);
        K_p = z * K_d;
        sim('PD.slx');
        time_array(:, 1) = out.time;
        pd_values_array(:, 1) = out.signals.values(:,1);
        hold on;
        plot(time_array, pd_values_array(:, 1));
        xlabel('t (s)');
        ylabel('z');
        grid on;
        box on;
end

legend("K = " + num2str(K_array(1)), "K = " + num2str(K_array(2)), "K = " + num2str(K_array(3)));

hold on;
```



As we can see, the plotted result shows that the PD system converges to a solution that is different from the initial condition when there is a constant disturbance. This means that, for a constant disturbance, the PD is unable to ignore it and, although it finds a solution, it isn't optimal.

The value to which the system converges was the one obtained in question $Q5.3$ for $K = 1190$.

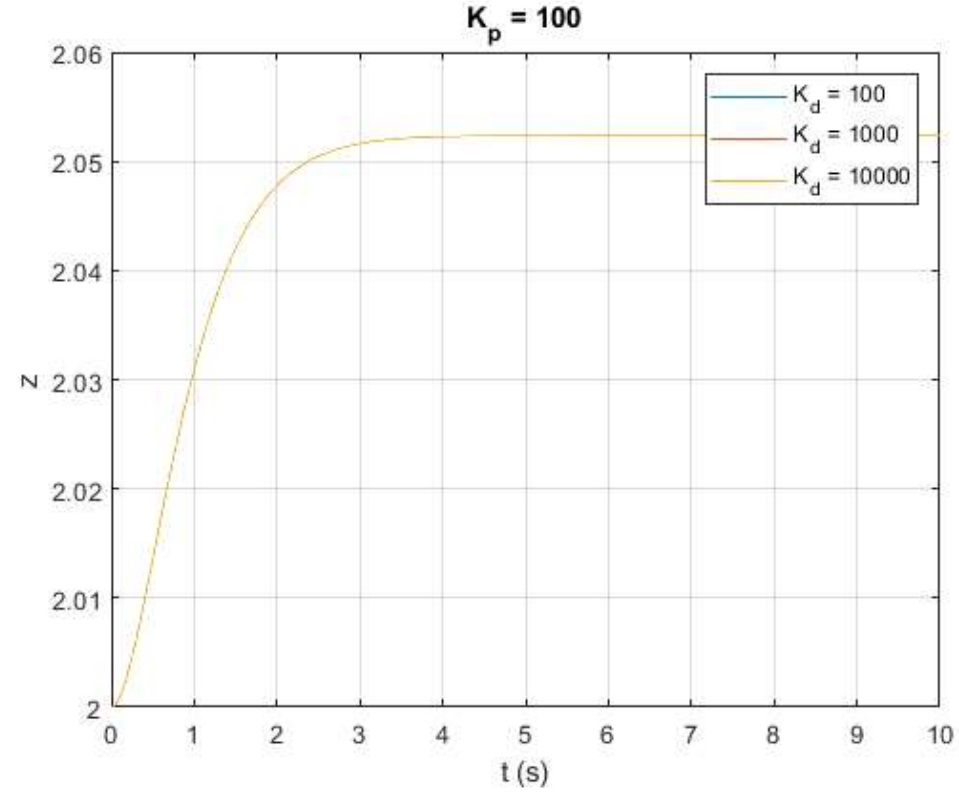
```
w_d = 50 * 2 * pi / 60;
delta_zr = 0;
K = 1.19E3;
K_p_array = [5000 10000 20000];
z = 1;
SimTime = 10;
StepSize = 1E-4;

figure; hold on, title("Effect of disturbances with the proportional-derivative controller");
for i=1:length(K_array)
        K = K_array(i);
        K_d = (K*M)/(600*K_t*omega0);
        K_p = K_p_array(i);
        sim('PD.slx');
        time_array(:, 1) = out.time;
        pd_values_array(:, 1) = out.signals.values(:,1);
        hold on;
        plot(time_array, pd_values_array(:, 1));
        xlabel('t (s)');
        ylabel('z');
        grid on;
        box on;
end

legend("K_p = " + num2str(K_p_array(1)), "K_p = " + num2str(K_array(2)), "K_p = " + num2str(K_p_array(3)));

hold on;
```

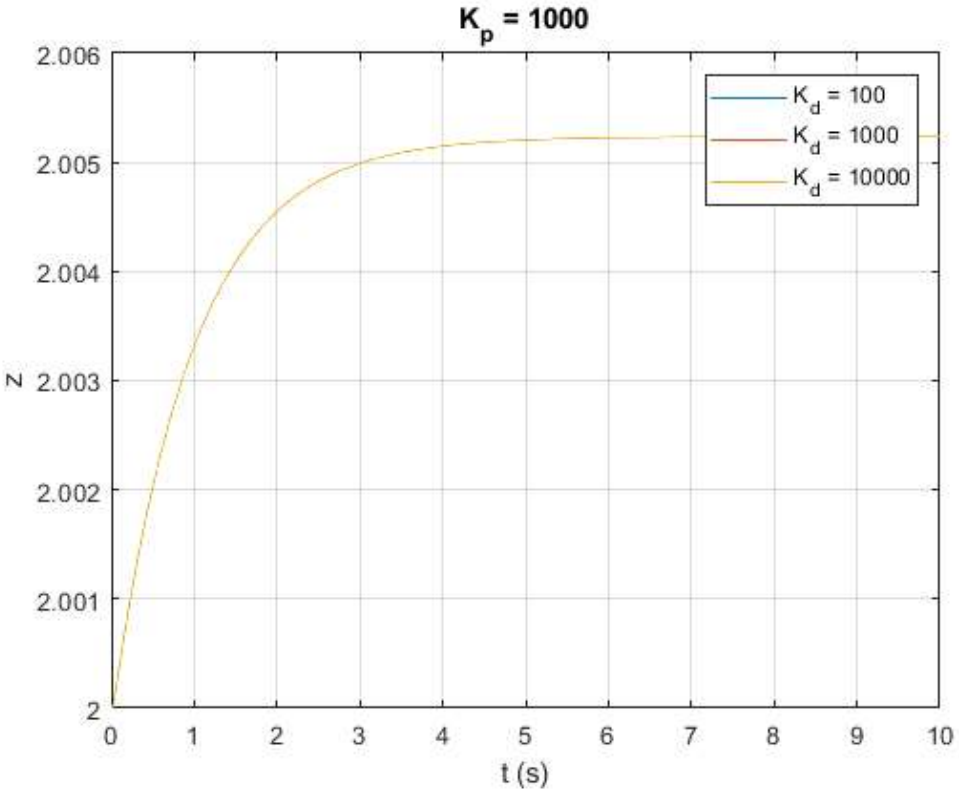


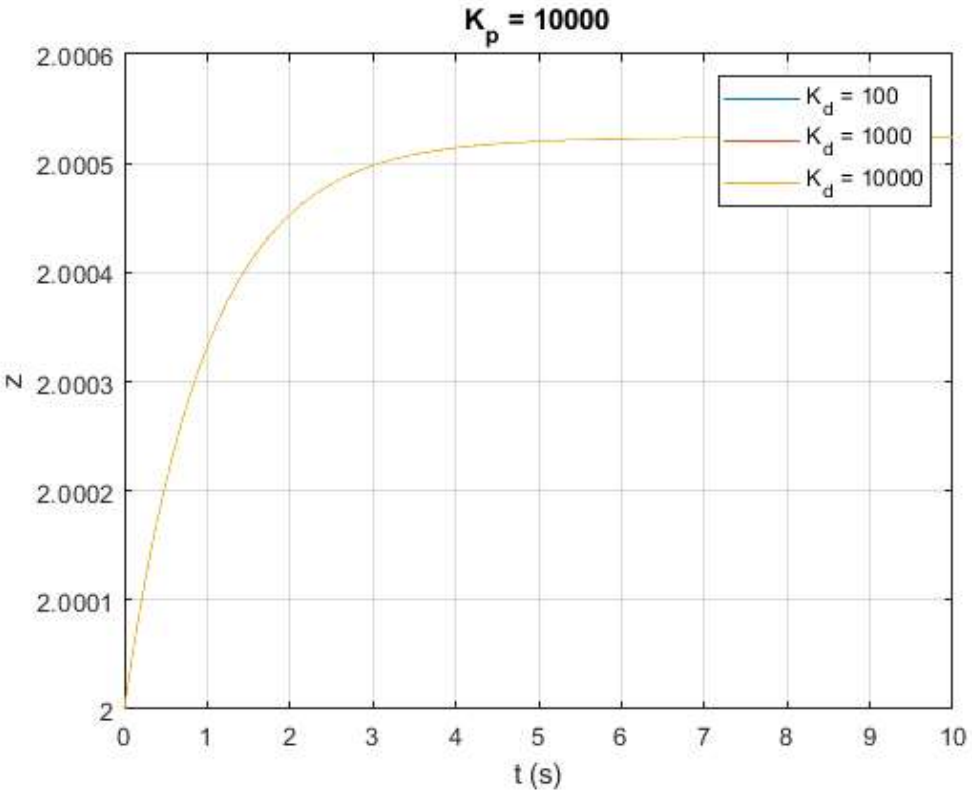
Additionally, we can see that the error in the convergence of the system is inversely proportional to K_p.

```
w_d = 50 * 2 * pi / 60;
delta_zr = 0;
```

```
SimTime = 10;
StepSize = 1E-4;

K_p_array = [100 1000 10000];
K_d_array = [100 1000 10000];

for i=1:length(K_p_array)
    figure; hold on; title("K_p = " + num2str(K_p_array(i)));
    for j=1:length(K_d_array)
        K_p = K_p_array(i);
        K_d = K_d_array(i);
        sim('PD.slx');
        time_array(:, 1) = out.time;
        pd_values_array(:, 1) = out.signals.values(:,1);
        hold on;
        plot(time_array, pd_values_array(:, 1));
        xlabel('t (s)');
        ylabel('z');
        grid on;
        box on;
    end
    legend("K_d = " + num2str(K_d_array(1)), "K_d = " + num2str(K_d_array(2)), "K_d = " + num2str(K_d_array(3)));
    hold off;
end
```


K_p = 100
2.06
2.05
2.04
2.03
2.02
2.01
2
z
0
1
2
3
4
5
6
7
8
9
10
t (s)
K_d = 100
K_d = 1000
K_d = 10000

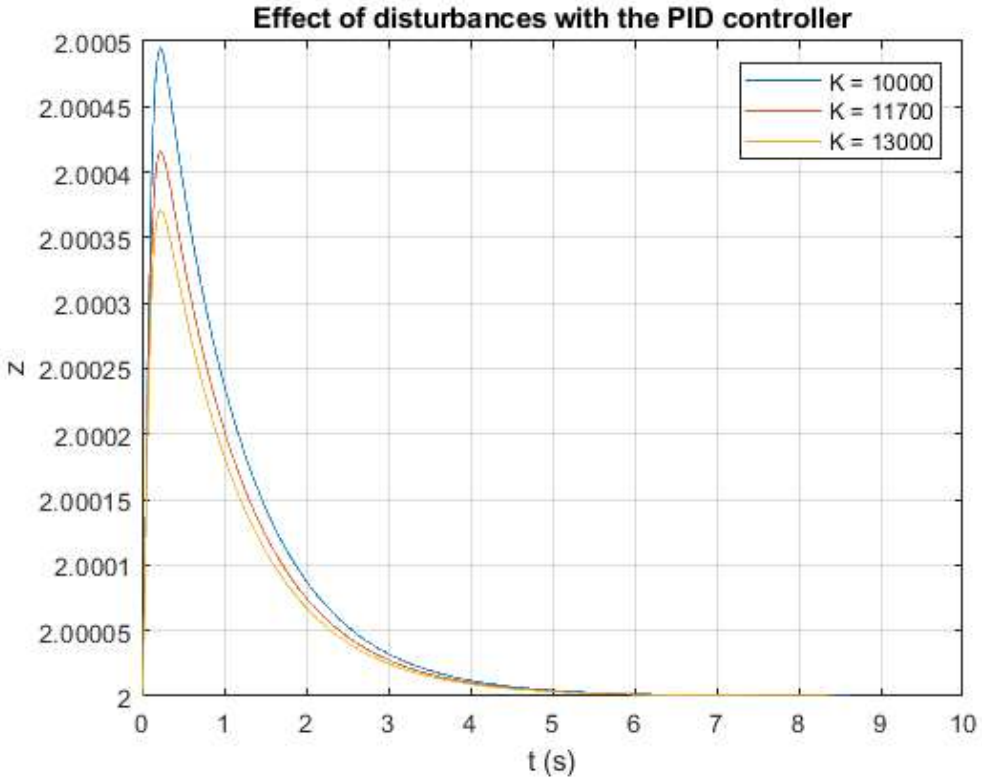

K_p = 1000
2.006
2.005
2.004
2.003
2.002
2.001
2
z
0
1
2
3
4
5
6
7
8
9
10
t (s)
K_d = 100
K_d = 1000
K_d = 10000



What can be concluded here is that for a fixed K_p, different values for K_d give the same response. Therefore, K_d does not alter the step response of the system.

Q5.6

```
w_d = 50 * 2 * pi / 60;
delta_zr = 0;
K_array = [10000, 1.17E4, 13000];
z1 = 1;
z2 = 10;
SimTime = 10;
StepSize = 1E-4;

figure; hold on; title("Effect of disturbances with the PID controller");
for i=1:length(K_array)
        K = K_array(i);
        K_d = (K*M)/(600*K_t*omega0);
        K_p = K_d*(z1 + z2);
        K_i = K_d*z1*z2;
        sim('PID.slx');
        time_array(:, 1) = out.time;
        pid_values_array(:, 1) = out.signals.values(:,1);
        plot(time_array, pid_values_array(:, 1));
        xlabel('t (s)');
        ylabel('z');
        grid on;
        box on;
end

legend("K = " + num2str(K_array(1)), "K = " + num2str(K_array(2)), "K = " + num2str(K_array(3)));

hold off;
```



Here something different happens when compared to the PD controller. The PID is able to suppress the constant disturbance being put into the system and the system converges to the initial condition. In other words, the PID controller is able to ignore constant perturbations.

It can also be seen that the higher the gain, the smaller the transient.

Published with MATLAB® R2023a