EE 324: Problem Sheet-5

Abhilaksh Kumar, 18D070035

February 22, 2021

1 Problem-1

(a)

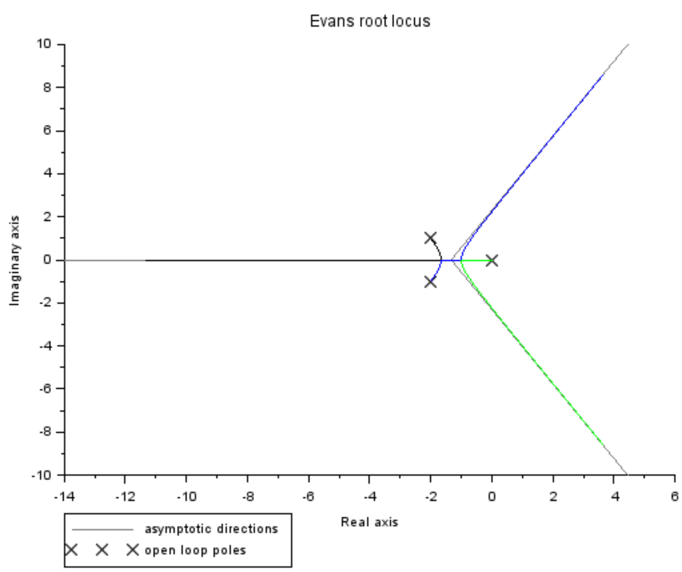
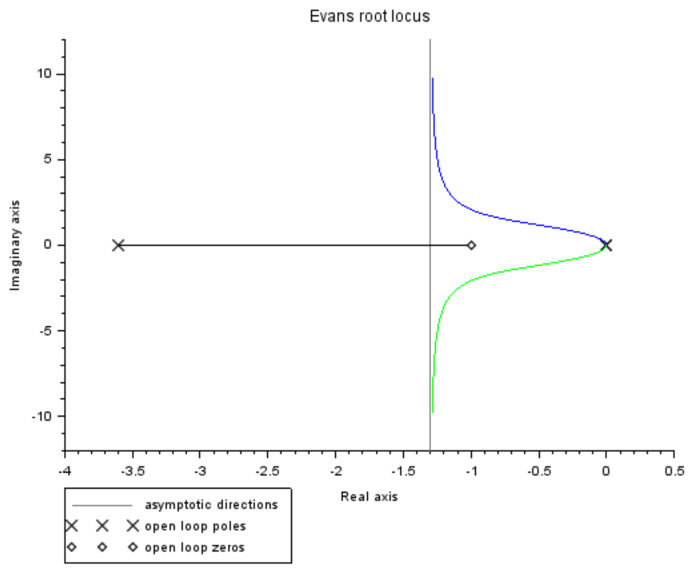


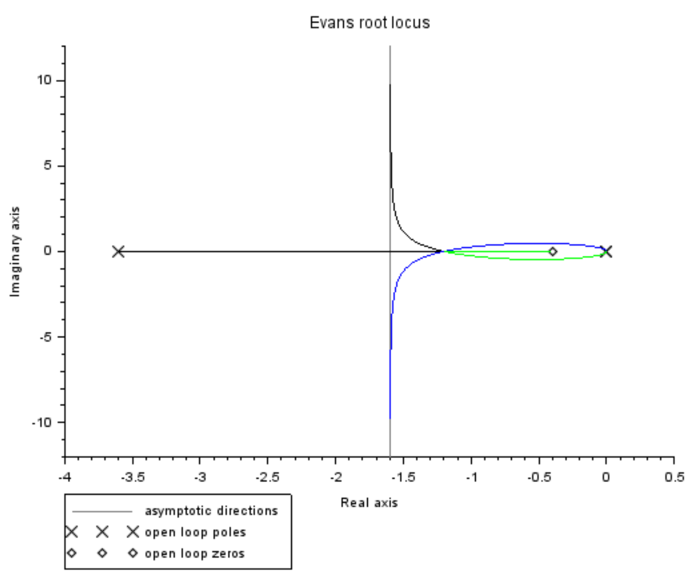
Figure 1: The Root Locus Plot for 1(a)

(b)



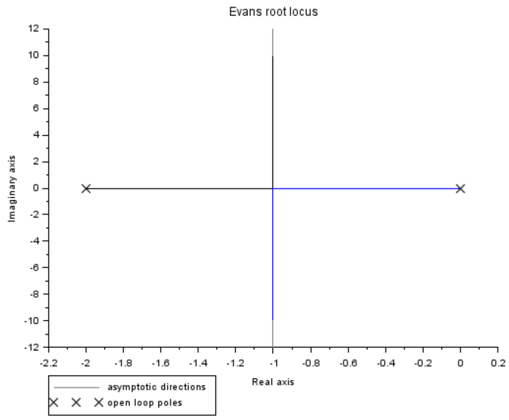
Evans root locus
Imaginary axis
Real axis
asymptotic directions
open loop poles
open loop zeros

(c)

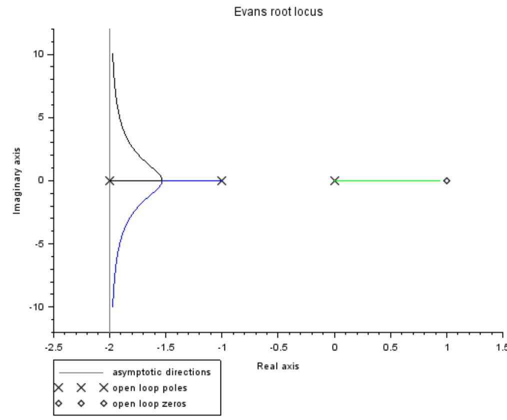


Evans root locus
Imaginary axis
Real axis
asymptotic directions
open loop poles
open loop zeros

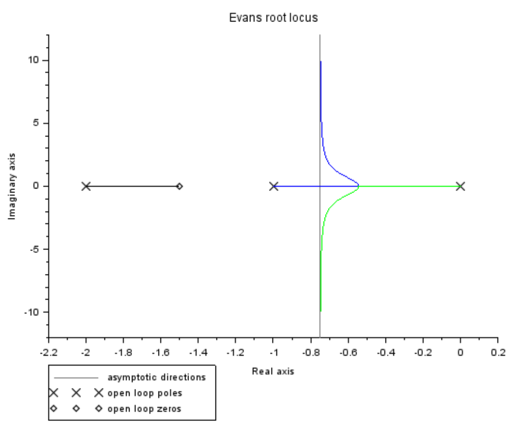
(d)



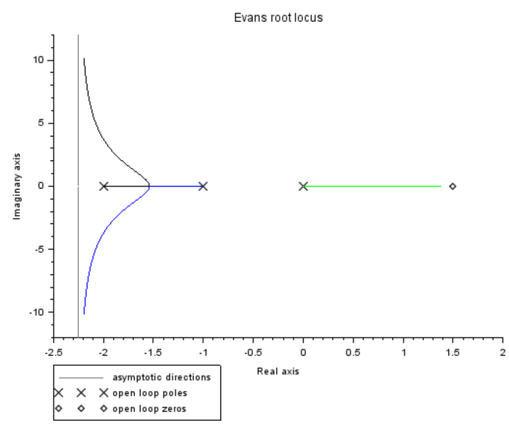
(a) p = 1



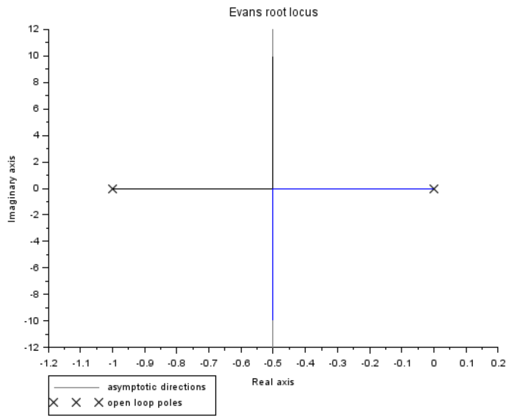
(b) p = -1



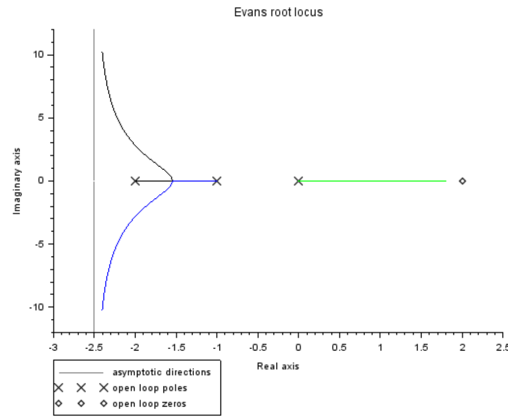
(a) p = 1.5



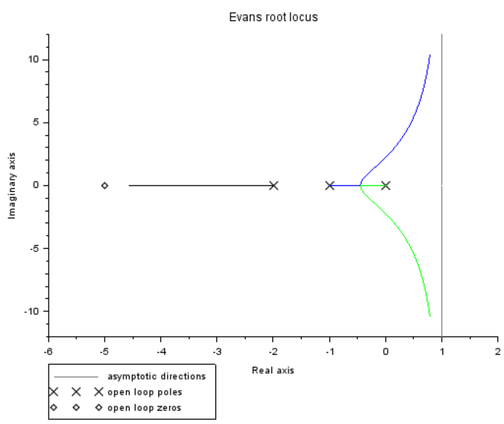
(b) p = -1.5



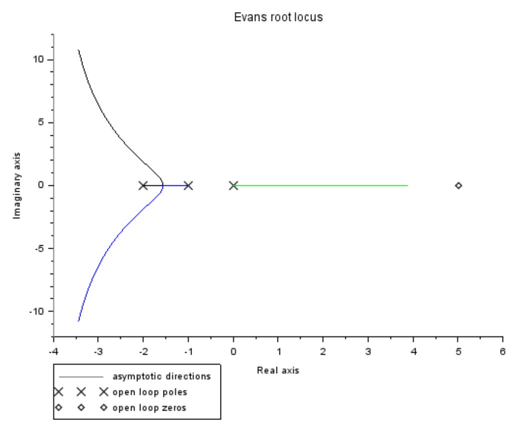
(a) p = 2



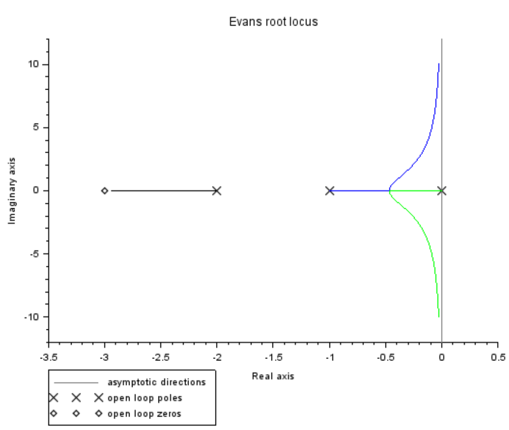
(b) p = -2



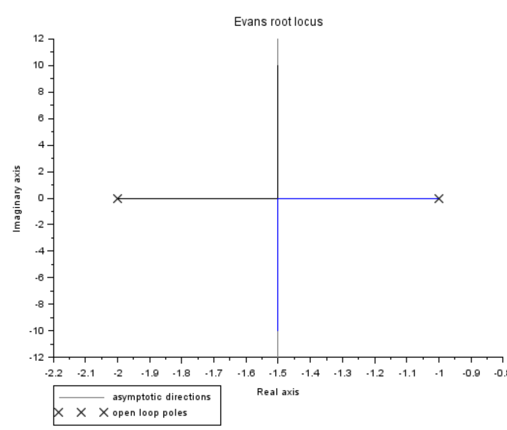
(a) p = 5



(b) p = -5



(a) p = 3



(b) p = 0

Code

```
s = poly(0,'s');
 // Part-A

// G/G+1 = given value
G = 10/(s*(s^2+4*s+5));
G = syslin('c',G);
evans(G,100);
show_window(1)

// Part-B
G = syslin('c', (s+1)/(s^2*(s+3.6)));
evans(G,100);
show_window(2)

// Part-C
G = syslin('c', (s+.4)/(s^2*(s+3.6)));
evans(G,100);
show_window(3)

// Part-D
ps = [-5,-2,-1.5,-1,0,1,1.5,2,3,4,5]
for i = 1:length(ps)
    p = ps(i)
    G = syslin('c', (s+p)/(s*(s+1)*(s+2)));
    scf(i+3);
    evans(G,100);
    xs2png(i+3, 'Q1d'+  string(10*p)+ '.png');
end
evans(G,100);
show_window(4)
```

2 Problem-2

(a)

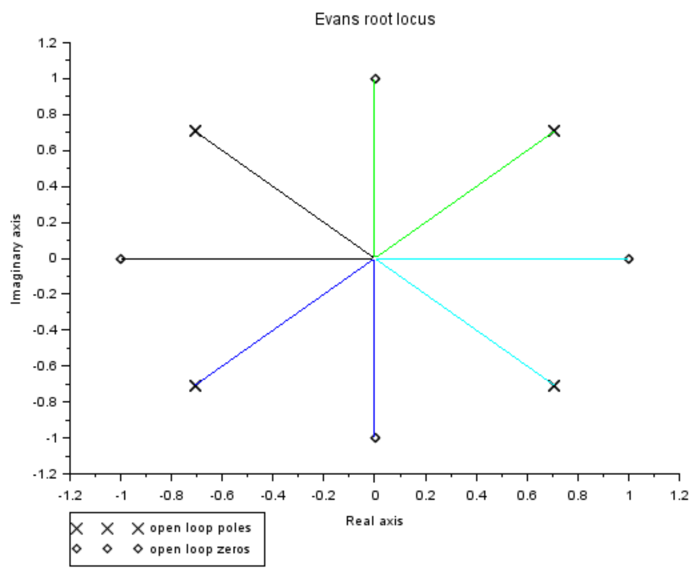


Figure 7: The Root Locus for 2(a)

(b)

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

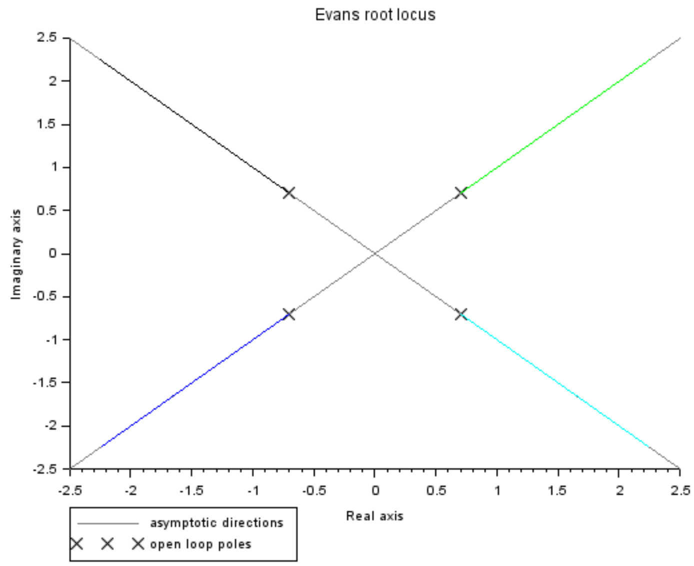


Figure 8: The Root Locus for 2(b)

(c)

$$G(s) = \frac{1}{s^4 - 1}$$

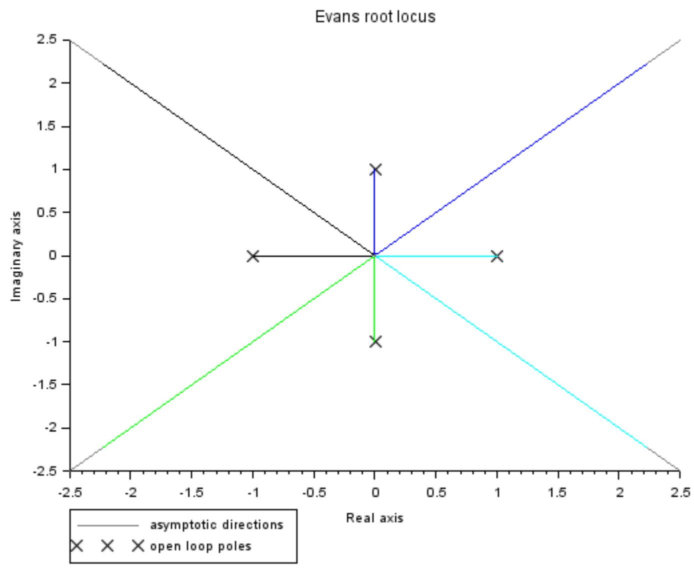


Figure 9: The Root Locus for 2(c)

(d)

For this part, I choose the poles to be located at ± 1 and ± 2. Then, our transfer function gets modified step by step as:-

$$G_1 = \frac{1}{(s^2-1)(s^2-4)}$$
$$G_2 = \frac{1}{(-s^2-1)(-s^2-4)} = \frac{1}{(s^2+1)(s^2+4)}$$
$$G_3 = \frac{1}{((s+1)^2+1)((s+1)^2+4)}$$

The plot obtained after step-2 is:-

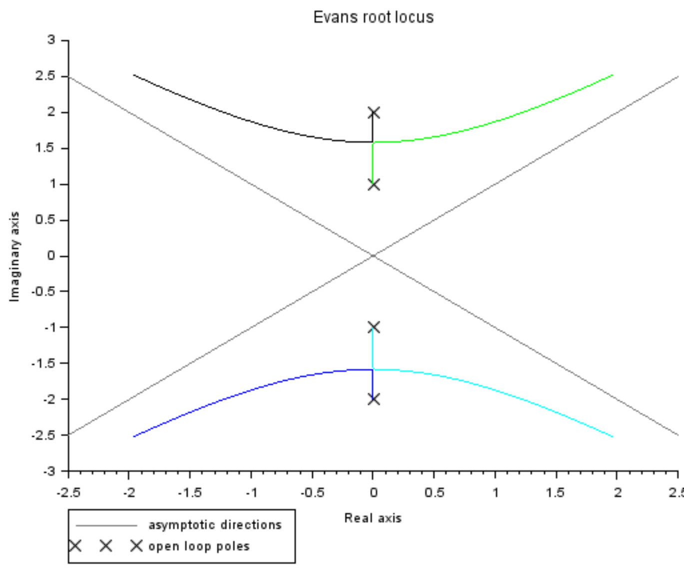


Figure 10: The Root Locus for 2(d) after step-2

The plot obtained after step-3 is:-

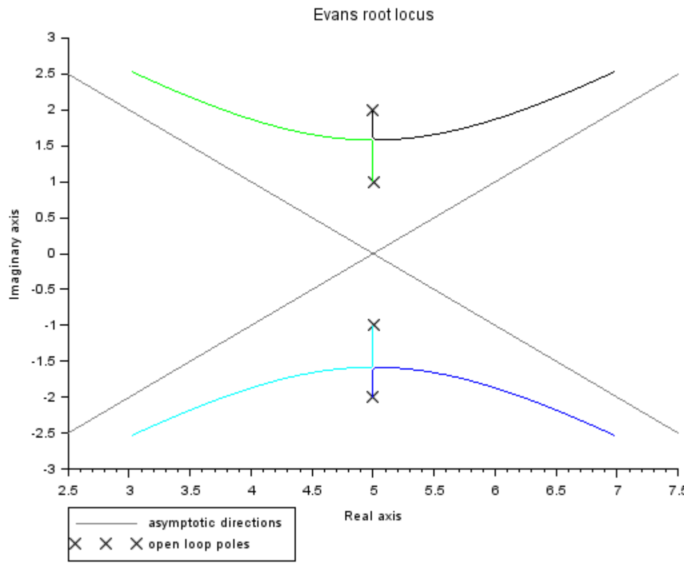


Figure 11: The Root Locus for 2(d) after step-3

Code

```
s = poly(0,'s');

// part a
G = syslin('c', (s^4-1)/(s^4+1));
scf(0);
evans(G,100);
xs2png(0, '2a.png');

// part b
G = syslin('c', 1/(s^4+1));
scf(1);
evans(G,100);
xs2png(1, '2b.png');


// part c
G = syslin('c', 1/(s^4-1));
scf(2);
evans(G,100);
xs2png(2, '2c.png');

// part d
G1 = 1/((s+1)*(s-1)*(s-2)*(s+2));       // chosen tf
G2 = 1/((-s^2-4)*(-s^2-1));
// choosing k = 5
G = syslin('c', 1/((-(s-5)^2-4)*(-(s-5)^2-1)));
scf(3);
evans(G,100);
xs2png(3, '2d.png');
scf(4);
evans(syslin('c',G2),100);
xs2png(4,'2d_1.png');
```

3 Problem-3

The closed loop tf is

$$G(s) = \frac{K_pG}{1+K_pG} = \frac{K_p}{s^3+3s^2+5s+K_p}$$

R-H table:-

s^3	1	5
s^2	3	K_p
s^1	(15-K_p)/3	0
s^0	K_p	0

Table 1: R-H table for Question-3

For stability, $K_p < 15$ and $K_p > 0$, while varying K, we get the following values of rise time (Value of K for rise time 1.5s is obtained by manually seeing the matrix):-

	Value of K_p	**Rise Time Obtained**
For Minimum	15	0.55
For $T_r = 1.5s$	3.7	1.5s

Table 2: Values Obtained for Problem-3

Code

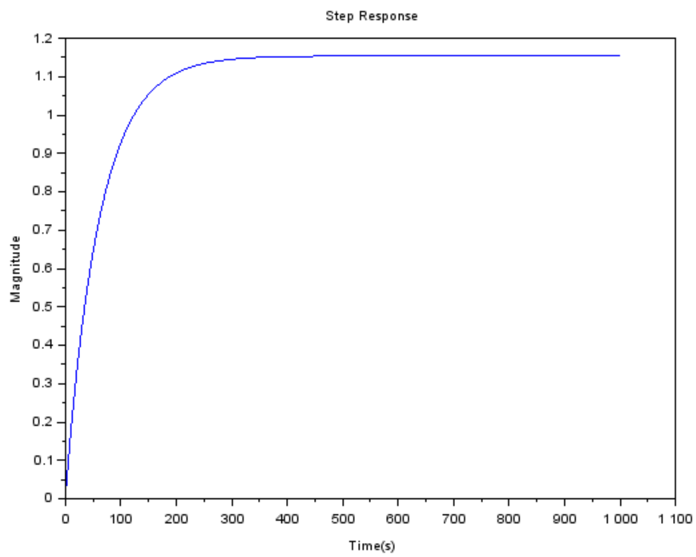
```
s = poly(0,'s');
k_range = .1:.1:15;
t = 0:.015:60;
rise_times = zeros(1,length(k_range));

for i = 1:length(k_range)
    k = k_range(i);
    G = k/(s^3+3*s^2+5*s+k);
    G = syslin('c', G);
    o = csim('step',t,G);
    t1 = t(find(o>.9))(1);
    t2 = t(find(o>.1))(1);
    rise_times(1,i) = t1-t2;
end

tr_min = min(rise_times(1,2:length(k_range)))
k_min = k_range(find(rise_times == tr_min));
```

4 Question 4

Here, we have to design a proportional controller for the following system:

$$G(s) = \frac{0.11(s+0.6)}{6s^2 + 3.6127s + 0.0572}$$

We are to find the gain k_p for which the steady state error for the step response is 1%. The step response of G(s) is as given below:



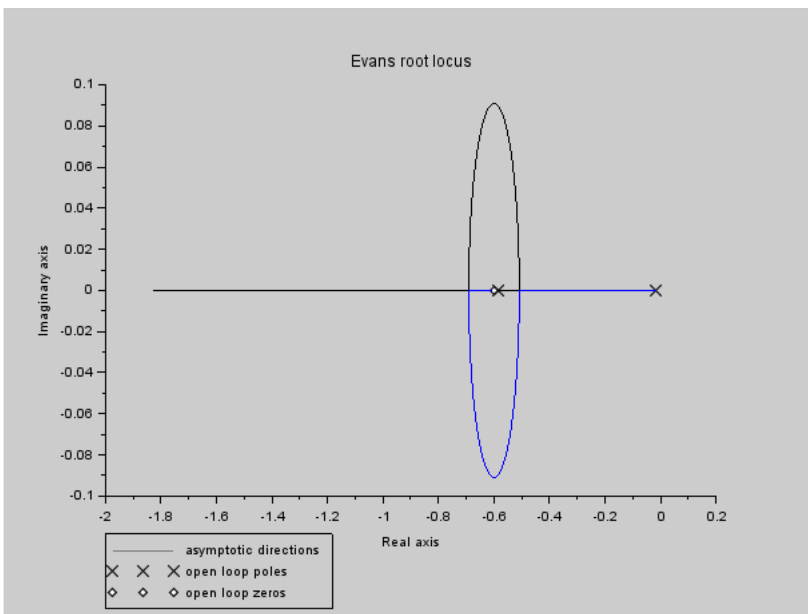


Figure 12: Root Locus

Now for optimising the steady state error, I simply iterated over k from 0.1 to 100 in steps of 0.1, and found that for k = 85.8, we have steady state error exactly = 0.01.

```
For k = 85.00 , Steady State Error = 0.010093
For k = 85.10 , Steady State Error = 0.010081
For k = 85.20 , Steady State Error = 0.010070
For k = 85.30 , Steady State Error = 0.010058
For k = 85.40 , Steady State Error = 0.010046
For k = 85.50 , Steady State Error = 0.010035
For k = 85.60 , Steady State Error = 0.010023
For k = 85.70 , Steady State Error = 0.010012
For k = 85.80 , Steady State Error = 0.010000
For k = 85.90 , Steady State Error = 0.009988
For k = 86.00 , Steady State Error = 0.009977
```

Next up, we had to find K_p for which the system was marginally stable, and this gain was found out to be -0.867.

4.1 Code for Question 4

```
clear;
s = poly(0, 's');
sys = 0.11*(s + 0.6)/(6*s^2 + 3.6127*s + 0.0572);
t = 0.1:0.1:1000;
plot(t, csim('step', t, syslin('c',sys)));
xlabel("Time(s)");
ylabel("Magnitude");
title("Step Response");
k_range = 0.1:0.1:100;
for i = 1:length(k_range)
    e = 100/(1+k_range(i)*.066/.0572);
    disp("K value: " + string(k_range(i)+ " Steady state error percent: "
    +string(e))
end
figure(2)

evans(sys,100)
```

Problem-5

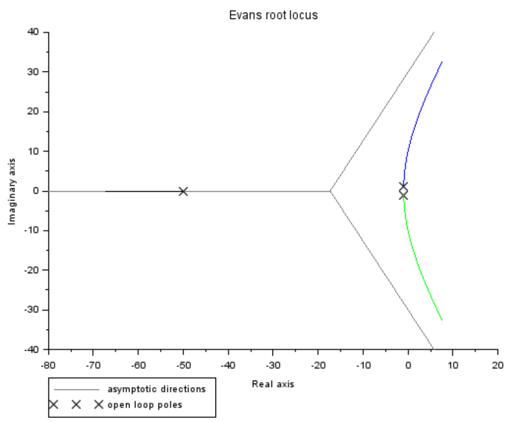
The 3rd order system considered is:-

$$G_1(s) = \frac{100}{((s+1)^2+1)(s+50)}$$

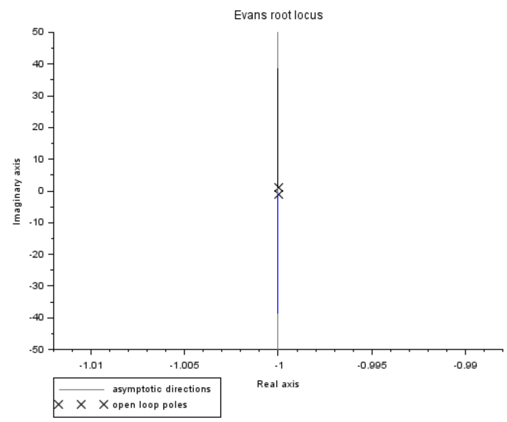
And the second order system is:-

$$G_2(s) = \frac{2}{((s+1)^2+1)}$$

The root locus for both are as shown below:

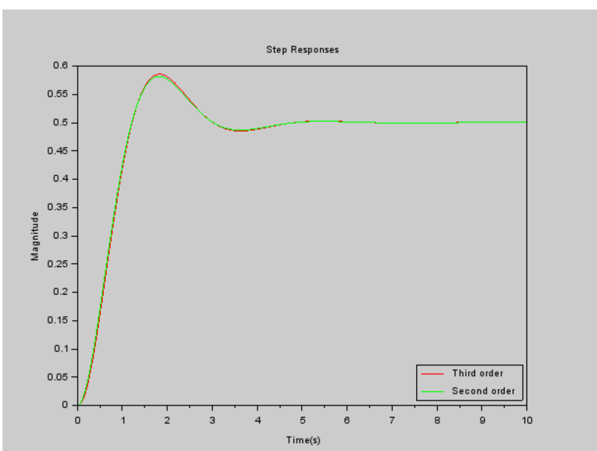


(a) Root Locus of $G(s)$

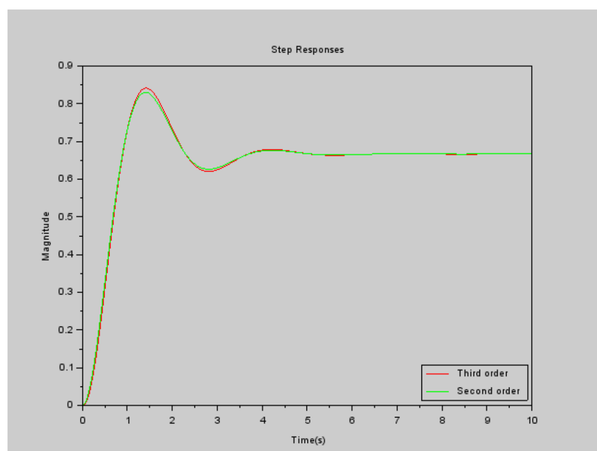


(b) Root Locus of $G_1(s)$

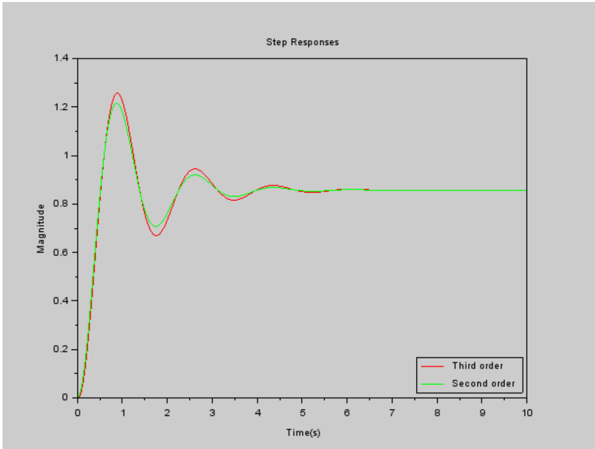
As the third root goes further and further away, G's root locus starts becoming more and more vertical, till it eventually becomes similar to G_1's plot. We then compare the step responses of the closed loop systems by varying the gain factor k:



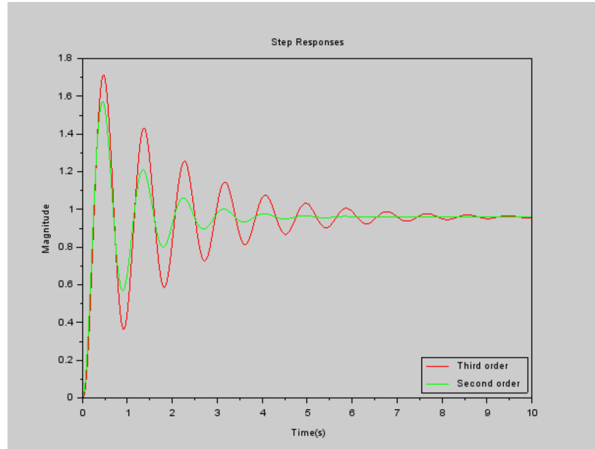
(a) k = 1



(b) k = 2



(a) k = 3



(b) k = 4

Code

```
s = poly(0, 's');
tf = 100/((s^2 + 2*s + 2)*(s+50));
evans(tf);
tf2 = 2/((s^2 + 2*s + 2));
figure(2);
evans(tf2);
t = 0:0.01:10;
k_range = [1,2,3,4];
count = 3;
for k = k_range
tf = tf*k;
tf2 = tf2*k;
y1 = csim('step', t, tf/(1+tf));
y2 = csim('step', t, tf2/(1+tf2));
figure(count)
f.background = 8;
plot(t, y1, 'r');
plot(t, y2, 'g');
xlabel("Time(s)");
ylabel("Magnitude");
title("Step Responses");
legends(['Third order','Second order'],[5,3],opt="lr");
count = count + 1;
end
```