EE324: CONTROL SYSTEMS LAB
PROBLEM SHEET 5

VINIT AWALE, 18D070067

February 21, 2021

Contents

1 Root Locus and Behaviour of closed loop poles

1.1 Problem a

The given closed loop transfer function of a system with unity negative feedback is as follows:

$$T(s) = \frac{10}{s^3 + 4s^2 + 5s + 10}$$

To find the open loop transfer function we apply a unity gain positive feedback to the system. Hence the open loop transfer function of the system can be written as

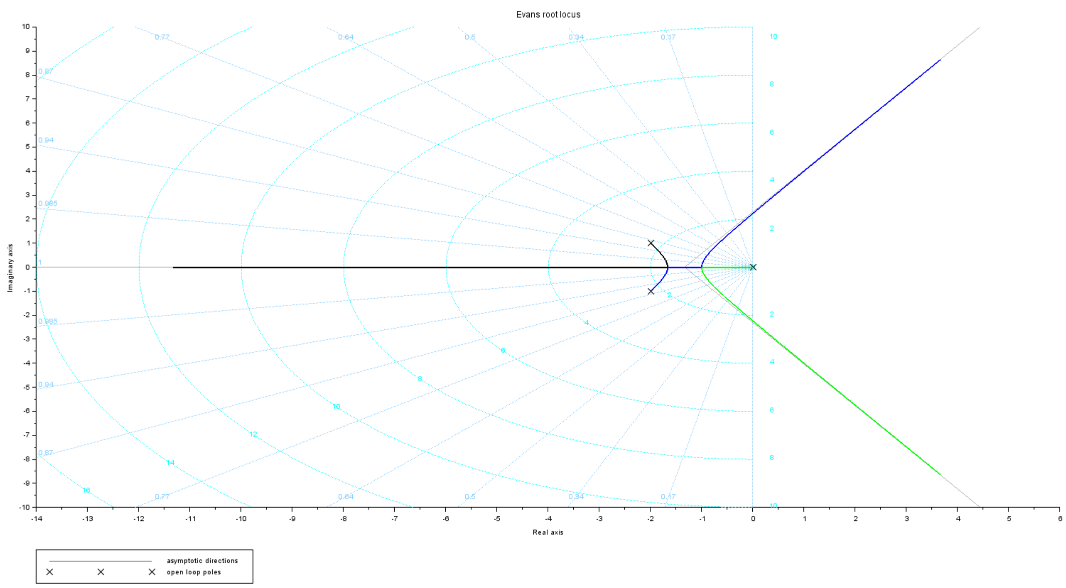
$$G(s) = \frac{T(s)}{1 - T(s)}$$

We find the root locus of this system using the following Scilab code.

Scilab code

```
clear
close
clc

s = poly(0,'s');
T = 10/(s^3 + 4*s^2 + 5*s + 10);
G = T/(1-T);
Glin = syslin('c',G);
clf();
evans(Glin,100);
sgrid();

// Post-tuning graphical elements
ch = gca().children;
curves = ch(2).children;
curves.thickness = 2;
asymptotes = ch(ch.type=="Segs");
asymptotes.segs_color = color("grey70");
```

Output



Observation

We can easily see from the root locus plot that the closed loop poles begin from finite values and move to infinite values.

1.2 Problem b

The given open loop transfer function is:

$$G(s) = \frac{s+1}{s^2 * (s+3.6)}$$

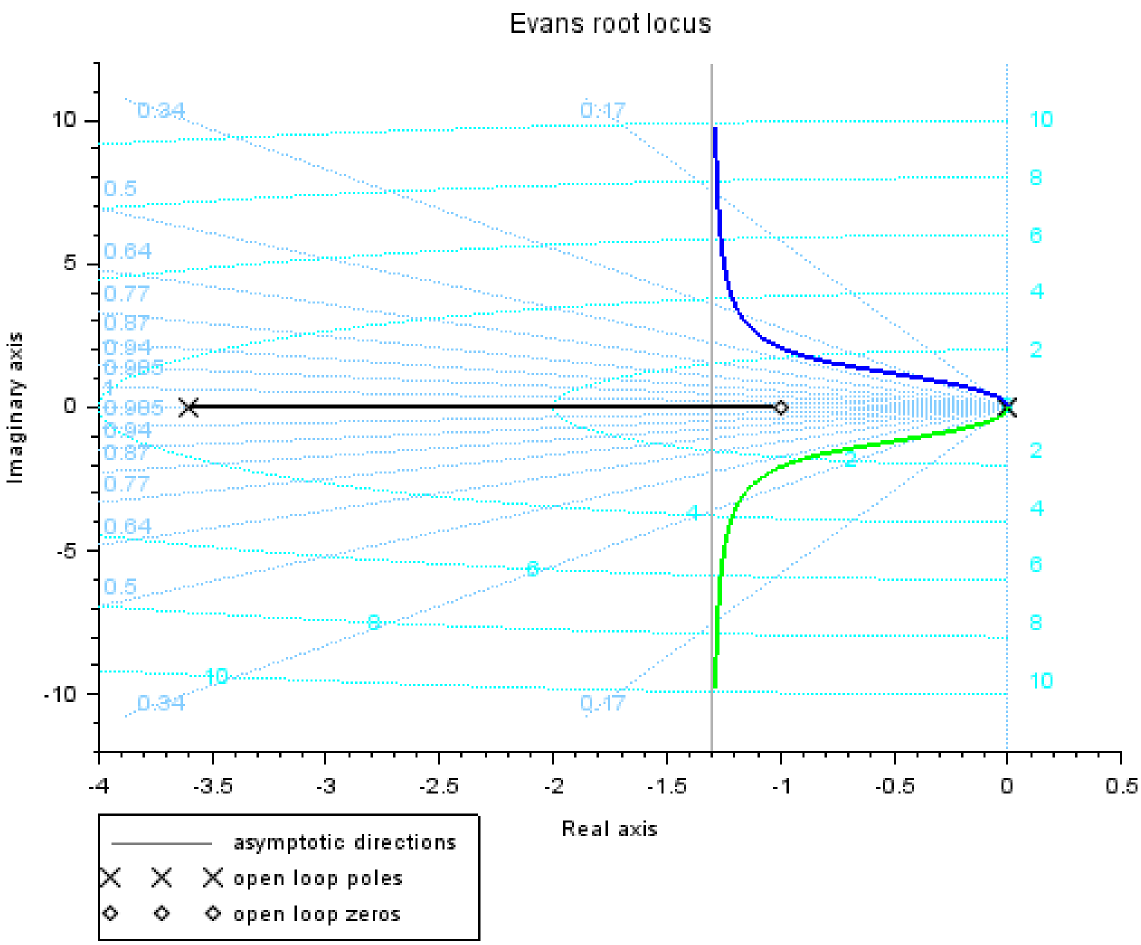
We find the root locus of this system using the following Scilab code.

Scilab code

```
clear
close
clc

s = poly(0,’s’);
G = (s+1)/(s^2*(s+3.6));
Glin = syslin(’c’,G);
clf();
evans(Glin,100);
sgrid();

// Post-tuning graphical elements
ch = gca().children;
curves = ch(2).children;
```

```
curves.thickness = 2;
asymptotes = ch(ch.type=="Segs");
asymptotes.segs_color = color("grey70");
```

Output



1.3 Problem c

The given open loop transfer function is:

$$G(s) = \frac{s+0.4}{s^2*(s+3.6)}$$

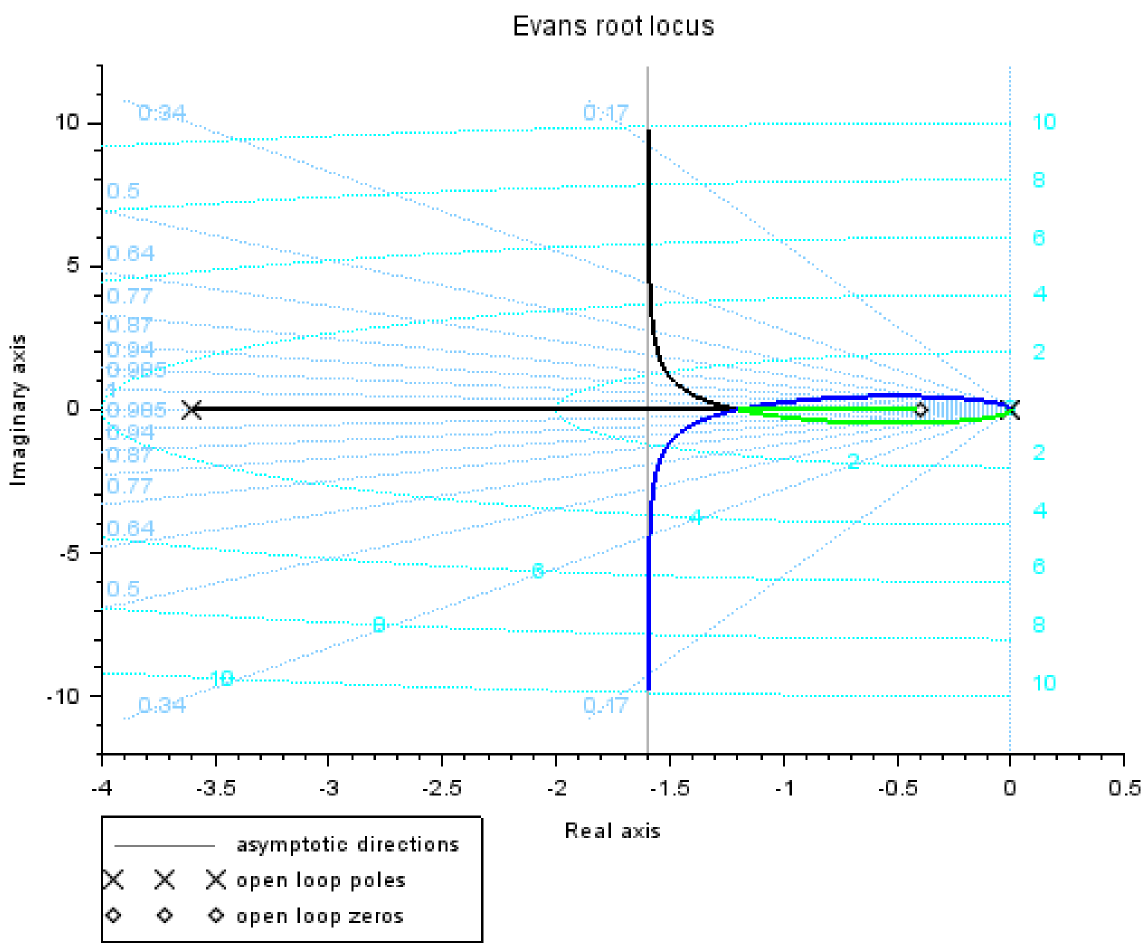
We find the root locus of this system using the following Scilab code.

Scilab code

```
clear
close
clc

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

```
G = (s+0.4)/(s^2*(s+3.6));
Glin = syslin('c',G);
clf();
evans(Glin,100);
sgrid();

// Post-tuning graphical elements
ch = gca().children;
curves = ch(2).children;
curves.thickness = 2;
asymptotes = ch(ch.type=="Segs");
asymptotes.segs_color = color("grey70");
```

Output

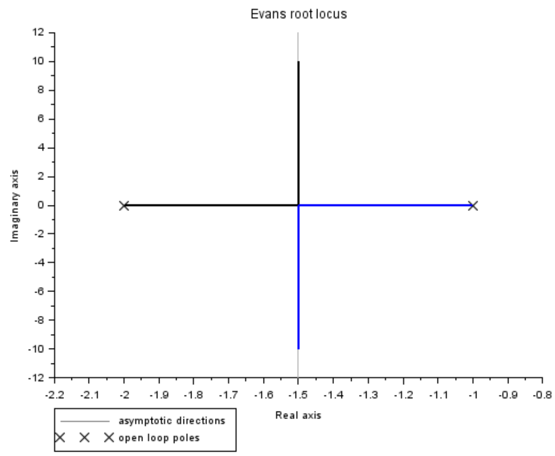


1.4 Problem d

Here we have been given a function having a parameter 'p' and are asked to comment on the stability of the system, as we vary this free parameter. $G(s) = \frac{(s+p)}{(s(s+1)(s+2))}$ Below shown are different plots of root locus for different values of p.

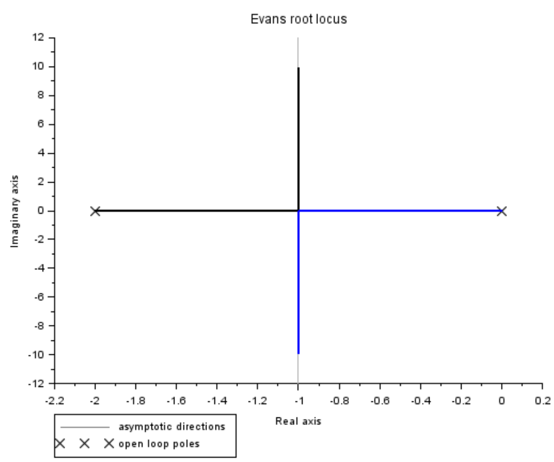
I have plotted differently plots for different p as it is hard to track them if plotted in same plot.

1.4.1 p=0



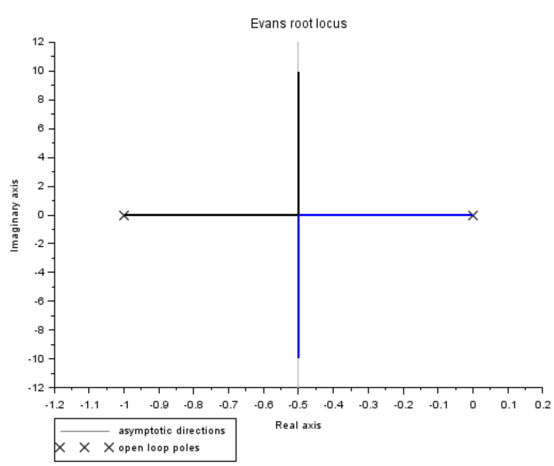
Root Locus for p = 0

1.4.2 p=1



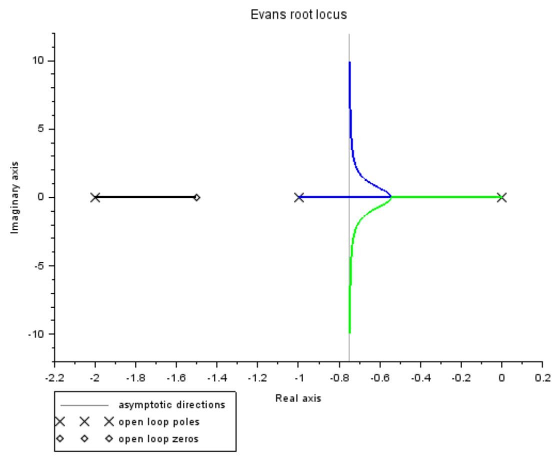
Root Locus for p = 1

1.4.3 p=2



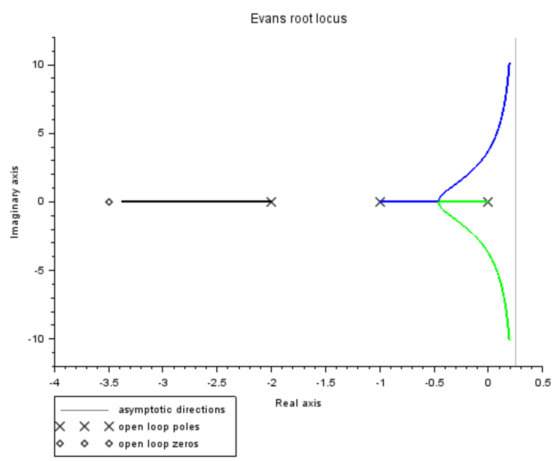
Root Locus for p = 2

1.4.4 For any intermediate value in (0,2) - 1 taken p = 1.5



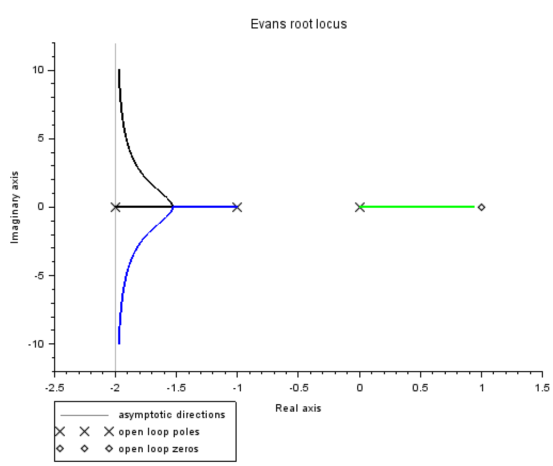
Root Locus for p = 1.5

1.4.5 Plot for p > 2



Root Locus for p = 3.5

1.4.6 Plot for p < 0



Root Locus for p < 0

Observation

So, here we observe that for $p \in [0,2]$ the system with any loop gain will always be stable. Whereas for values of p outside this range we can see from the above plots that the poles go in the right half plane for some value of k. Hence, the system for this p will not be stable for all values of k.

2 Special Root Locus

2.1 Problem a

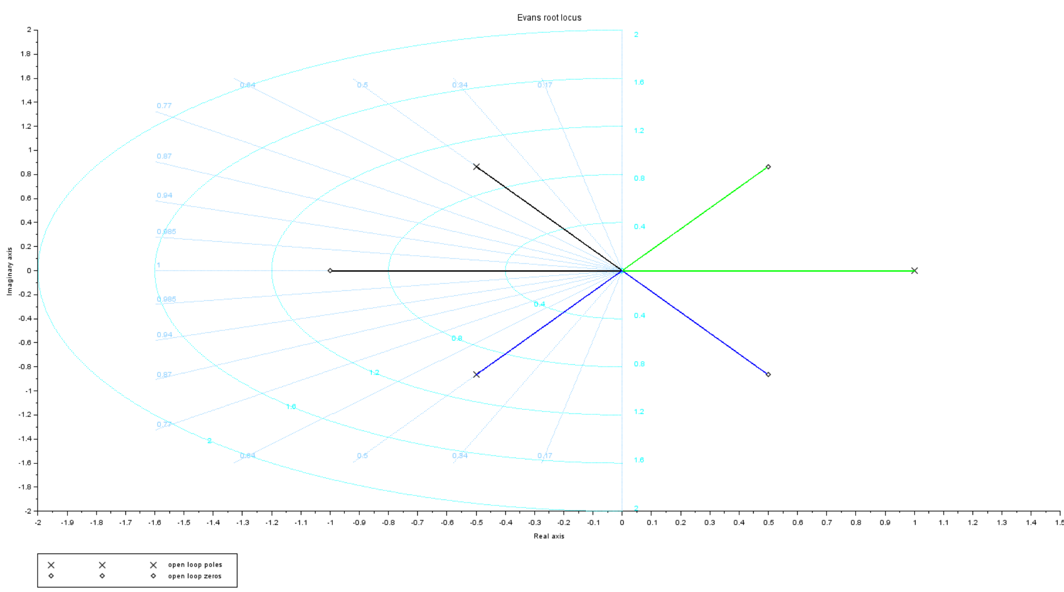
Here, we keep the roots and zeros of the open loop on a unit circle. Let the zeros be cube roots of unity and zeros be rotated by 60 degrees to the zeros, so we can get a real polynomial for both of them.

Below is the open loop transfer function used-

$$G(s) = \frac{(s+1)((s-1/2)^2+3/4)}{((s-1)((s+1/2)^2+3/4))}$$

Below shown of the plot for the root locus of the same.

Output



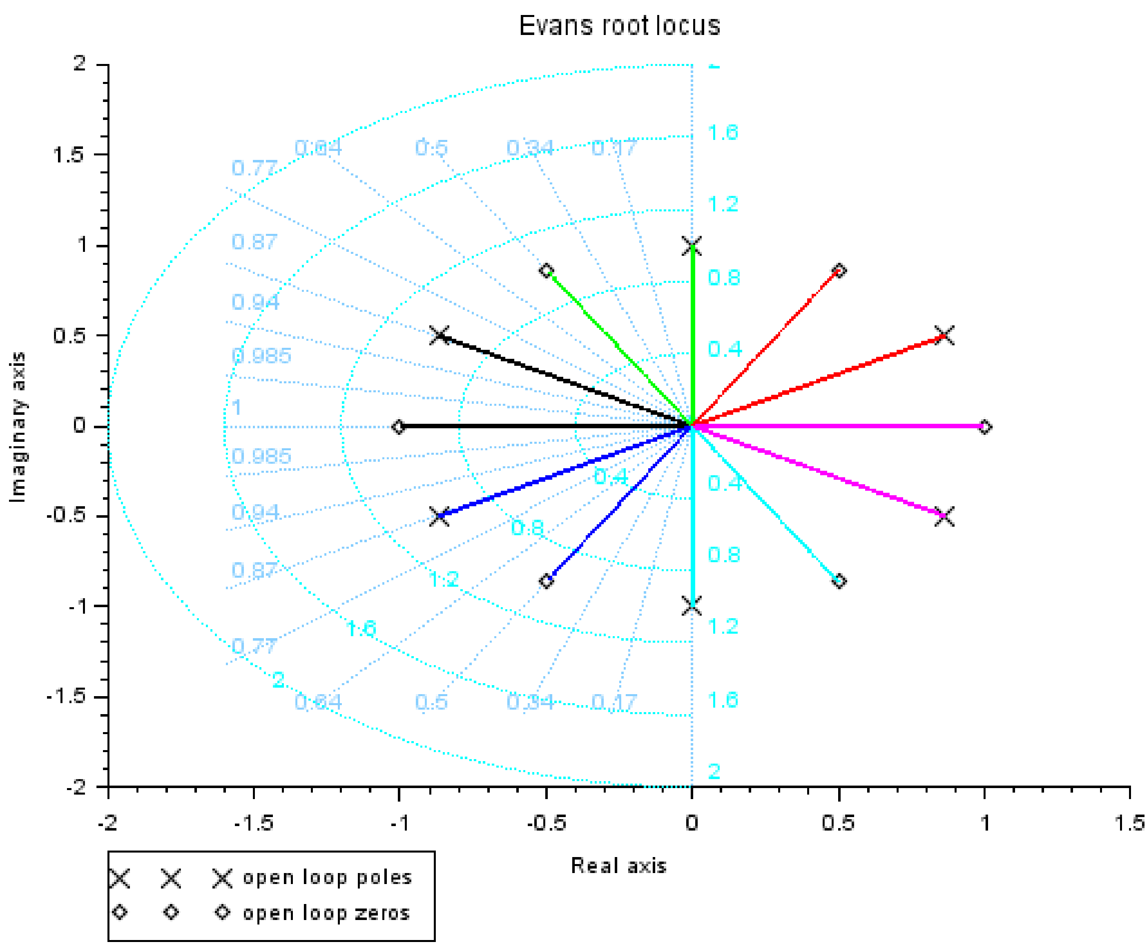
2.2 Problem b

Here also can place the poles and zeros on the unit circle. If the poles and zeros are placed symmetrically then the breakin/breakaway point is the origin. So number of branches coming to the break-in point in this case is equal to number of pole/zeros. So we make a system having 6 poles and zeros placed symmetrically on the unit circle. Below is the transfer function and graph.

$$G(s) = \frac{(s^6-1)}{(s^6+1)}$$

Below shown of the plot for the root locus of the same.

Output



2.3 Problem c

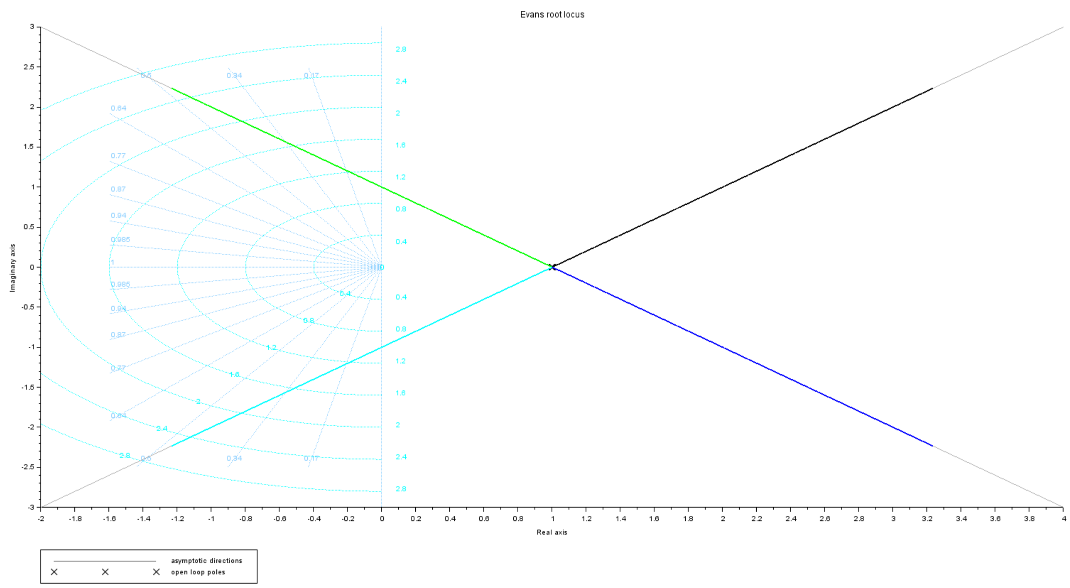
Here if we have a repeated pole and no zero then we will be having branches coming out linearly and thus will go to infinity without any deviation and thus we will be having asymptotes and branches of the root locus coinciding.

The transfer function used was

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

Below shown of the plot for the root locus of the same.

Output



2.4 Problem d

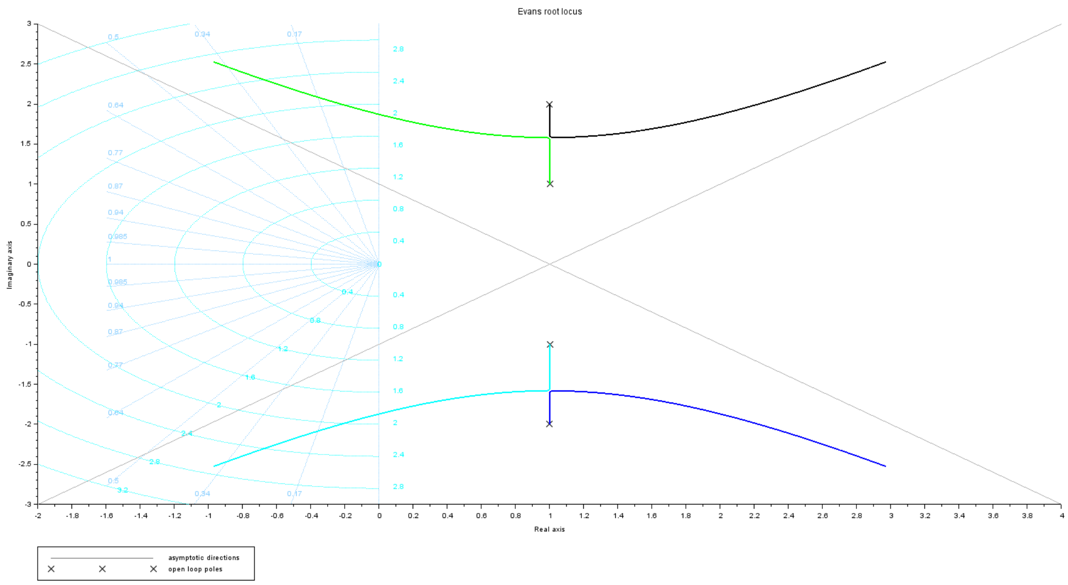
Here we will just follow the steps given-

So finally we get the following transfer function

$$G(s) = \frac{1}{((-(s-1)^2-1)(-(s-1)^2-4))}$$

Below shown of the plot for the root locus of the same.

Output



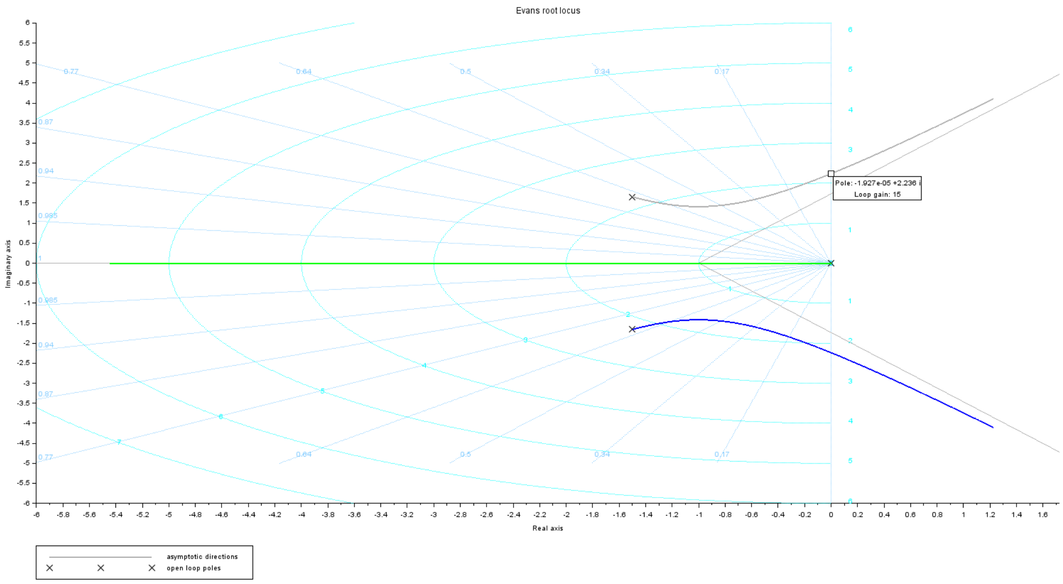
3 Design of Proportional Controller - I

The given third order system is:

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

We want the closed-loop system to have time domain specification of 1.5 seconds as the rise time, on giving the step input.

First, we draw the root locus of the given system



In the above plot we can easily see that the system becomes marginally stable for a loop gain of 15. For higher loop gains the system is unstable. Hence, we need to check the condition only till the loop gain of 15.

By following the general method, we tried to approximate the system as a second order system so we can use the method of intersection of the root locus with the circle of radius 1.8/rise time.

But when checked we can see that the circle does not intersect with the root locus. But by this we cannot comment upon whether the system is able to achieve rise time as 1.5 secs or not.

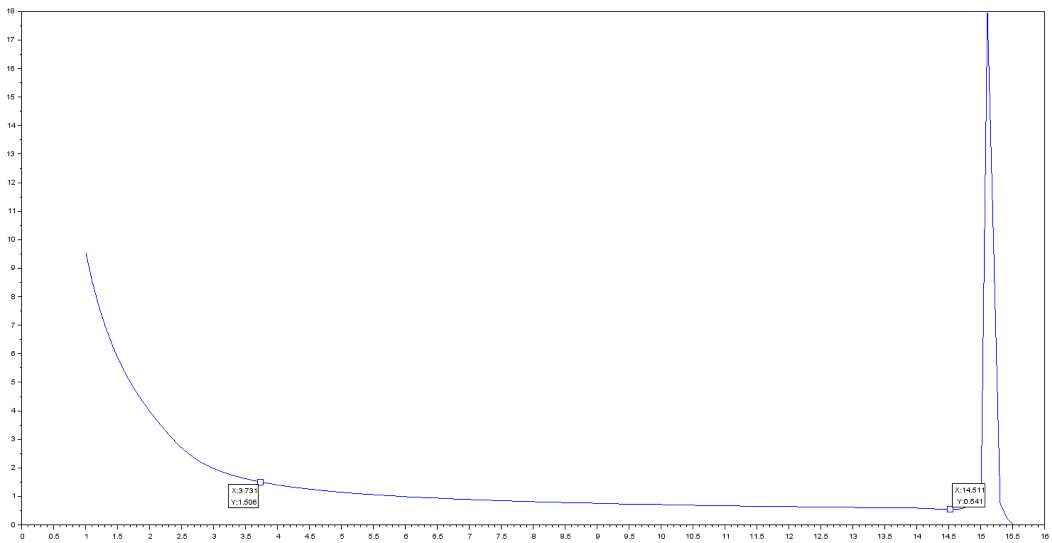
So, we need to do this without any approximation.

Scilab Code

```
clear
close
clc

s = poly(0,'s');

G = 1/(s*(s^2 + 3*s + 5));
t = 0:0.005:100;

k = 1:0.1:16;
t_rise1 = zeros(length(k));
t_rise2 = zeros(length(k));
t_rise = zeros(length(k));

for j = 1:1:length(k)
    t_sys = k(j)*G/(1 + k(j)*G);
    T = syslin('c',t_sys); //For defining the continuos time LTI system
    T_stp = csim('step' , t , T);
    T_stp_stdy = T_stp(length(t));

    //Finding rise time
    for i = 1:length(t)
```

```
        if (T_stp(i)> 0.1*T_stp_stdy)
            t_rise1(j) = t(i);
            break;
        end
    end

    for i = 1:length(t)
        if (T_stp(i)> 0.9*T_stp_stdy)
            t_rise2(j) = t(i);
            break;
        end
    end
    t_rise(j) = t_rise2(j) - t_rise1(j);
end
plot(k, t_rise)
```

Obtained Plot



Observations

- From the above plot we can easily see that the rise time of 1.5 sec is attained for the loop gain of 3.73.
- The min rise time that can be attained is 0.541 sec attained at the loop gain of 14.511.

4 Design of Proportional Controller - II

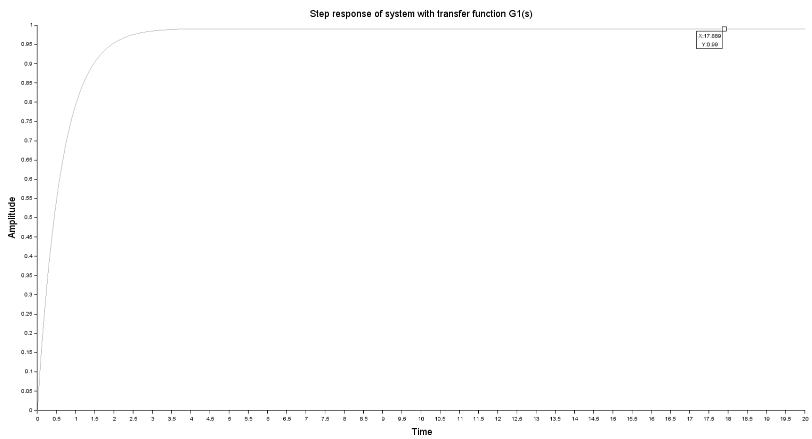
Firstly we will find the closed loop gain in terms of K_p manually.
So,

$$G_{closed} = \frac{KG_{open}}{1 + KG_{open}} = \frac{0.11 * K(s + 0.6)}{K * 0.11(s + 0.6) + (6s^2 + 3.6127s + 0.0572)}$$

So the steady state response to a unit step will be-
steady state response = $\frac{0.11*K*0.6}{K*0.11*0.6+0.0572}$
Now to make it 1 percent of the unit step, i.e.

$$0.99 \leq \frac{0.11+K*0.6}{K*0.11*0.6+0.0572} \implies 86.66 \leq K$$

Hence we can keep k= 86.66 for steady state error as 1%.
This can also be cross checked via Scilab. Below is the step response plot for K = 86.66



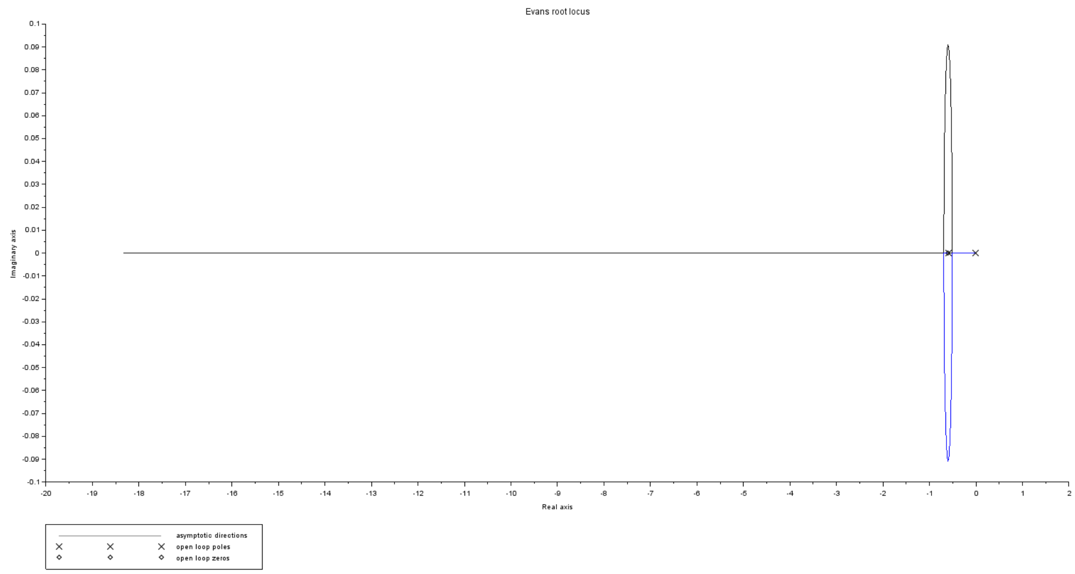
Now, to check for what of K it getting marginally stable we will plot root locus for both positive and negative K's.

Scilab code

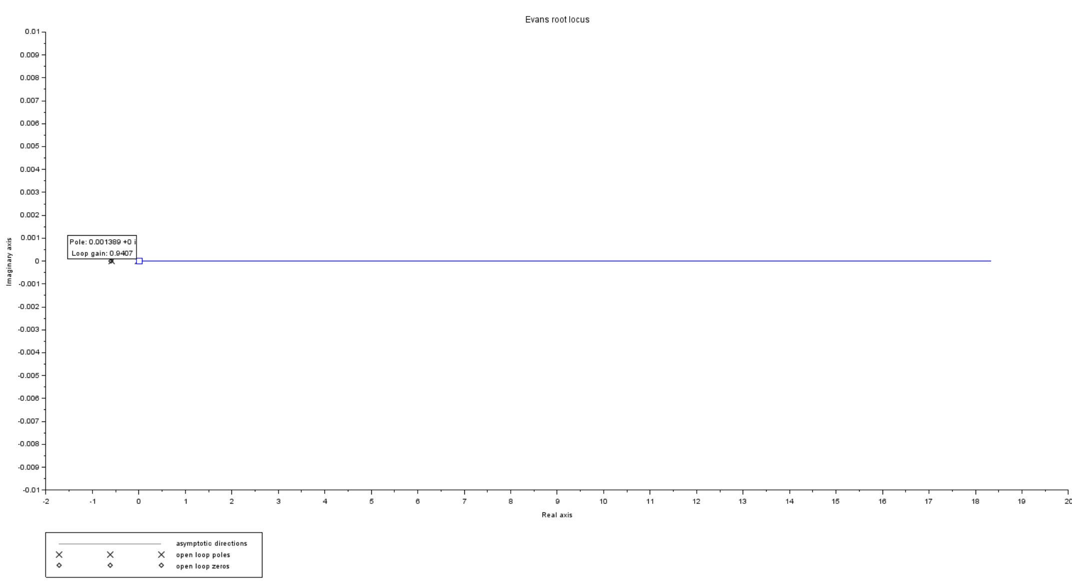
```
s = poly(0,'s');

open = -0.11*(s+0.6)/(6*s^2 + 3.6127*s + 0.0572);

evans(open,1000);
```

We observe that the root locus do not cross imaginary axis for positive K's and it does when plotted for negative K's at K = -0.9407.

Plot for positive K's



Plot for negative K's



5 Effect of Poles far from dominant pole

First we consider a third order system

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

As,we can easily see that the pole at s = -100 is very far away from the dominant poles, we can approximate the system by a second order system as follows

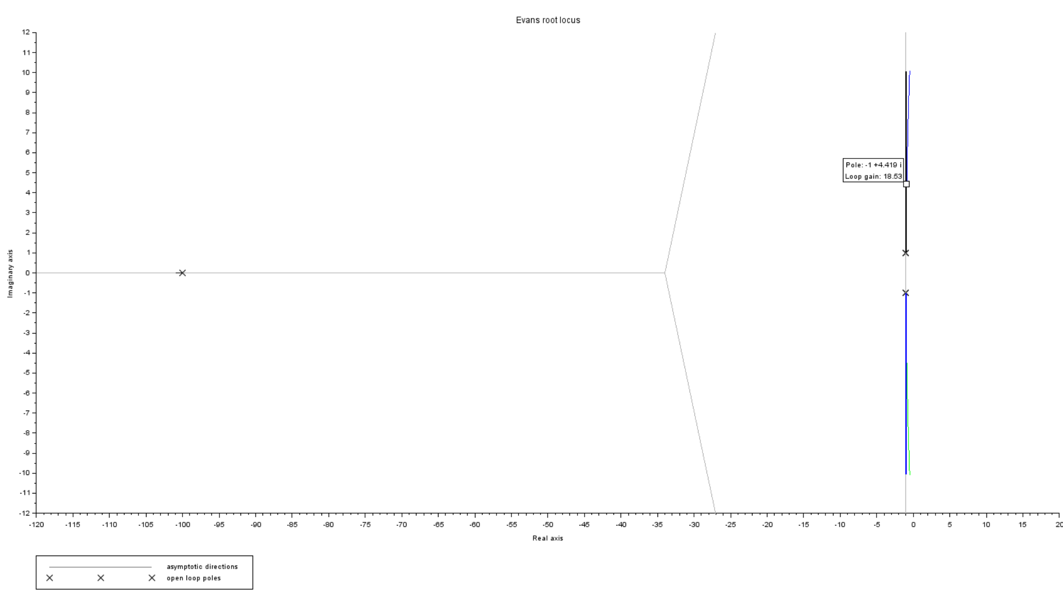
$$G_2(s) = \frac{1}{s^2 + 2s + 2}$$

We plot the Root Locus of both the systems on the same plot.

Scilab Code

```
clear
close
clc

s = poly(0,'s');
G1 = 100/((s+100)*(s^2+2*s+2));
Glin1 = syslin('c',G1);

G2 = 1/((s^2+2*s+2));
Glin2 = syslin('c',G2);

clf();
evans(Glin1,100);
evans(Glin2,100)


// Post-tuning graphical elements
ch = gca().children;
curves = ch(2).children;
curves.thickness = 2;
asymptotes = ch(ch.type=="Segs");
asymptotes.segs_color = color("grey70");
```

Obtained Plot



Observations

From the plot using data tip manager we can easily see that the two root loci are same till almost a loop gain of 18.53. After that the root locus of the G_1 starts to bend along the asymptotes, while that of G_2 keeps moving parallel to the imaginary axis.