EE324 Control Systems Lab

Problem Sheet 6

Sheel Shah — 19D070052

September 25, 2021

1 Question 1

We are given $G(s) = \dfrac{1}{(s+3)(s+4)(s+12)}$

1.1 Part A

We need steady state error of 0.489.

$$\therefore \frac{1}{1+KG(0)} = 0.489$$
$$\therefore \frac{1}{1+K/144} = 0.489$$
$$\therefore 1 + K/144 = 1/0.489$$
$$\therefore K = 144 * 1.045$$
$$\therefore K = 150.48$$

1.2 Part B code

We need $\zeta = 0.35$. Hence OS is constant.

$$tan(\alpha) = -\frac{\sqrt{1-\zeta^2}}{\zeta}$$

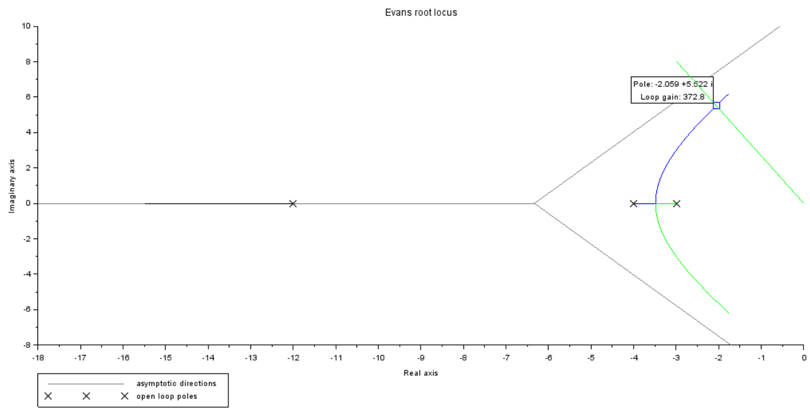


Figure 1: Plot indicating K is 372

1.3 Part C

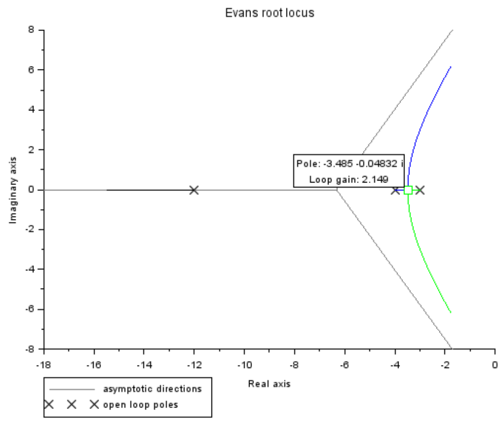


Figure 2: Plot indicating K is 2.15

1.4 Part D

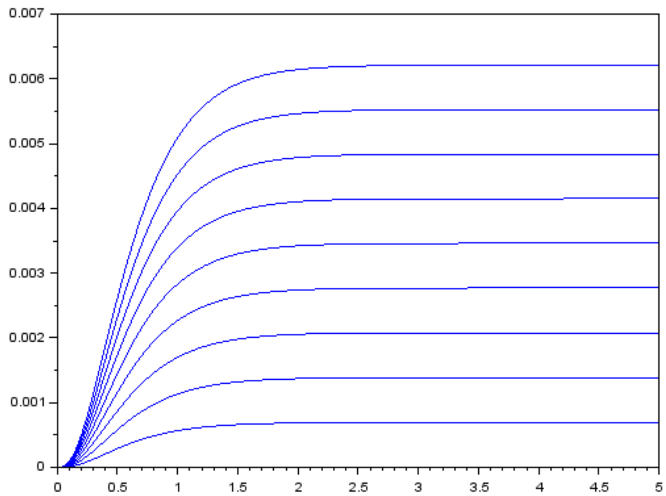


Figure 3: Step responses for K in (0, 1]

Conclusions:

1. Closed loop poles are all purely real. This follows from the observation that there is no overshoot.

2. Hence this is the horizontal portion of the root locus before break-away.

3. Steady state error decreases as K increases.

1.5 Part E

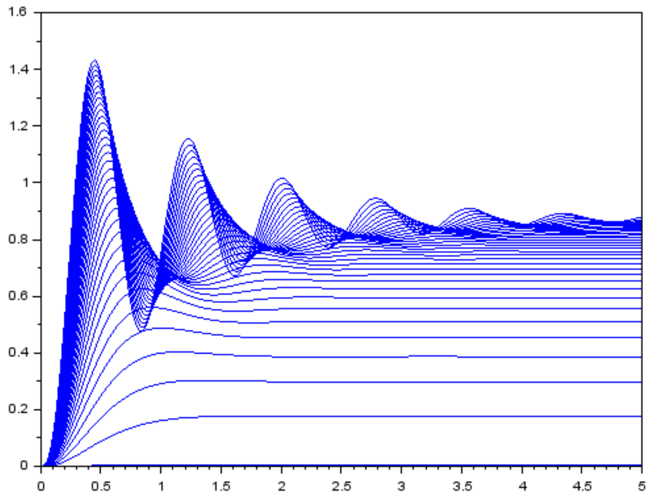


Figure 4: Step responses for K in [1, 1000]

Conclusions:

1. Closed loop poles are no longer purely real. This follows from the observation that there is overshoot.

2. Hence this is the curved portion of the root locus after break-away.

3. Steady state error decreases as K increases.

4. The settling time increases as K increases.

5. The system is stable as the poles still have negative real part, but if K were to increase further, the system will become unstable.

1.6 Code Used

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

g = 1 / ((s + 3) * (s + 4) * (s + 12))

G = syslin('c', g)


// part B

scf()

evans(G, 500)

x = -3:0.01:0

m = sqrt(1 - 0.35*0.35) / -0.35

y = m .* x

plot(x, y, "g-")


// part C

scf()

evans(G, 500)


// part D

scf()

K = %eps : 0.1: 1

t = 0 : 0.01: 5

for k = K

    G_temp = k * G / (1 + k * G)

    step_resp = csim("step", t, G_temp)

    plot(t, step_resp)

end
```

```
// part E

scf()

K = 1 : 30: 1000

t = 0 : 0.01: 5

for k = K

    G_temp = k * G / (1 + k * G)

    step_resp = csim("step", t, G_temp)

    plot(t, step_resp)

end
```

2 Question 2

All sub questions have been answered in the code/plots.

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

g = 1 / ((s + 3) * (s + 4) * (s + 12))

G = syslin('c', g)


// part A

// similar to question 1B

scf()

c = (s + 0.01) / s

C = syslin('c', c)

evans(C*G, 1000)

x = -3:0.01:0

m = sqrt(1 - 0.2*0.2) / -0.2

y = m .* x

plot(x, y, "g-")


// part B

// fixed w_n => fixed length of pole

scf()

c = (s + 0.01) / s

C = syslin('c', c)

evans(C*G, 2000)

x = -8:0.1:8
```

```
y = sqrt(64 - x^2)
plot(x, y, "g-")


x = -8:0.1:8
y = sqrt(81 - x^2)
plot(x, y, "r-")


// part C
Z = 0.01:0.3:1.5
for z=Z
    c = (s + z) / s
    C = syslin('c', c)
    scf()
    evans(C*G, 2000)
end
// I noticed that as z is increased, the curved part becomes flatter
// also, the branches are closer in
// implying the asymptotes are reached at hgher K


// part D
// yeds we can change the pole location
// as z increases, the pole moves slightly inwards
// therefore for he same damping ratio, we can have different poles
```

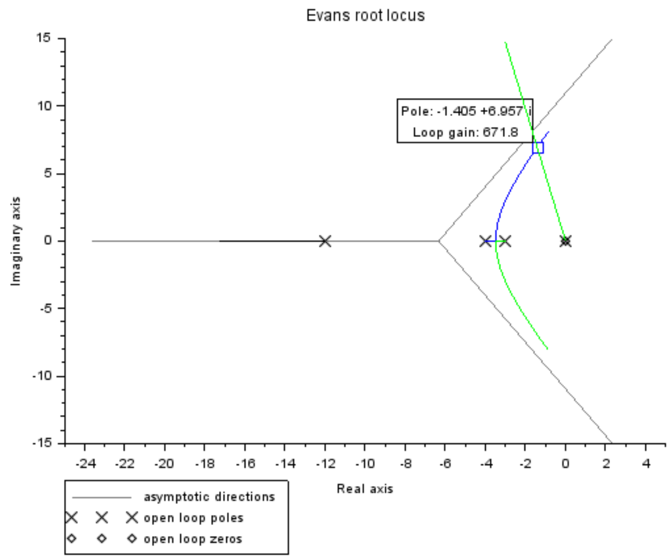


Figure 5: Plot A

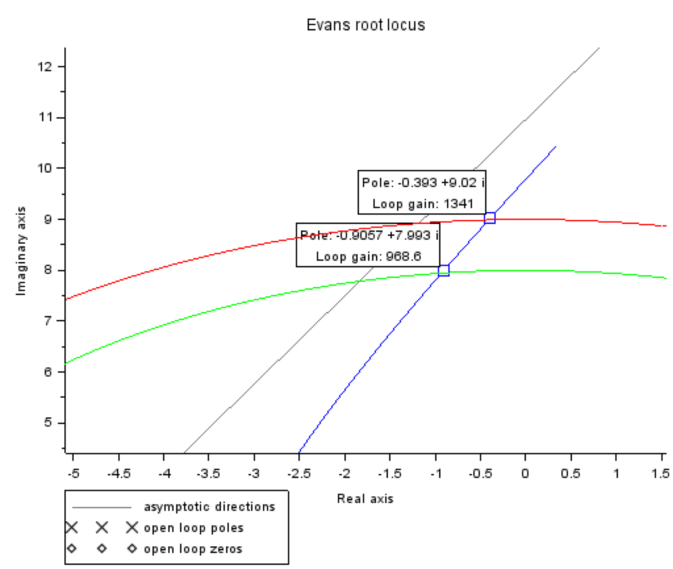


Figure 6: Plot B

3 Question 3

3.1 Part A

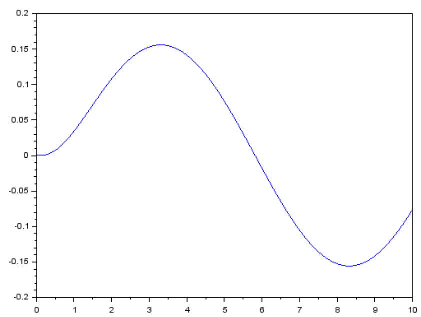
The frequencies chosen were 0.1, 0.5, 1, 2, 5 Hz.

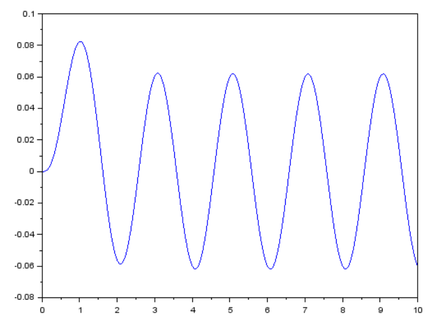
The step responses have been shown below the code.

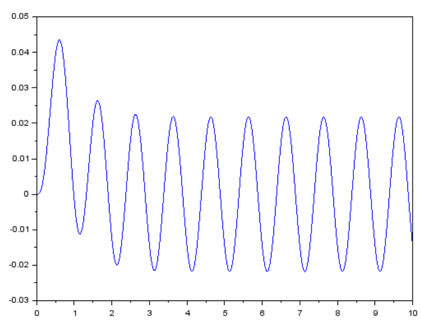
The code also shows theoretic values of gain and phase offset, which match visually with the plots.

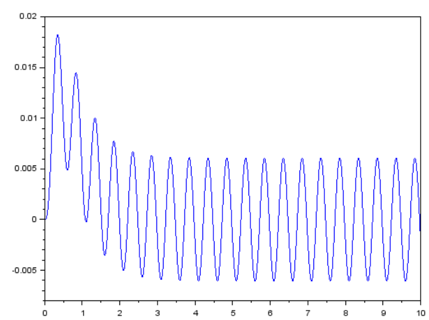
```
s = poly(0, 's')
g = 1 / (s^2 + 5*s + 6)
G = syslin('c', g)

freqs = [0.1, 0.5, 1, 2, 5]
t = 0:0.01:10

for freq = freqs
    input = sin(2 * %pi * freq * t)
    resp = csim(input, t, G)
    scf()
    plot(t, resp)

    g_jw = horner(g, 2 * %pi * freq * %i )
    [radius angle] = polar(g_jw)
    disp(angle, radius)
end
```

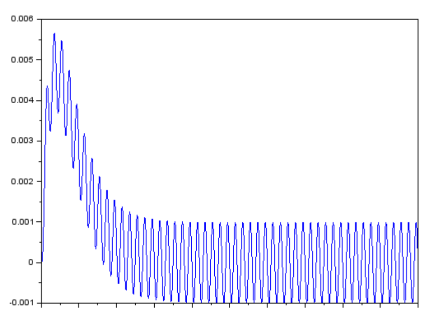
```
// theoretic values:
// 0.15562803, -0.510851115
// 0.06181396, -1.812333614
// 0.021781557, -2.387966089
// 0.006082923, -2.749415376
// 0.001006586, -2.982812192
```











3.2 Part B

We need frequency in rad/s.

3.3 Part C

The phase of G(jw) is independent of the numerator(since the numerator is real).

For phase of π, we want the denominator to be purely imaginary => $w = \sqrt{11}$.

Code similar to part A has been used. PLots are shown below.