EE324 Lab 8

Shubham Kar, 180070058

March 2021

1 Question 1

Assuming that the closed loop characteristic equation is:

$$G_{charac}(s) \ = \ G(s) * K \ + \ 1 \ = \ 1 \ + \ \frac{K}{s(s^2 + 4s + 8)} \tag{1}$$

Then, we get that the proportional gain at which the closed loop characteristic has gain and phase margin as 0, as **64**. The corresponding equations and poles of the closed loop system for $K \ = \ 64$ is shown in Figure 1.

```
Gain at which Gain and phase margins are equal is: 64.00
Phase margin at K = 64.0 is: 0.00
Gain margin at K = 64.0 is: -0.00
Gain Crossover Frquency is: 0.45
Phase Crossover Frquency is: 0.45
Closed loop system:

         64
   -----------------
              2   3
   64 + 8s + 4s + s
Poles of the closed loop system:
  -4.9766045
   0.4883022 + 3.5527081i
   0.4883022 - 3.5527081i
Therefore, closed loop system is not stable
```

Figure 1: Closed loop characteristics for $K = 64$

The bode plot of the closed loop characteristic equation is shown in Figure 2.
The gain and the phase margins varying with K are plotted and shown in Figure 3. Here, we see that the gain and the phase margin plots intersect at 2 points among which, $K = 64$ is of the lines. Both the margins are 0 there and therefore, we cannot have a condition that gain margin is 0 and phase margin is non zero as gain margin is monotonic in nature. Same goes for the condition that gain margin is non-zero when phase margin is zero as the phase margin plot is also monotonic.
Since the dominant poles of the closed loop system for the K when the closed

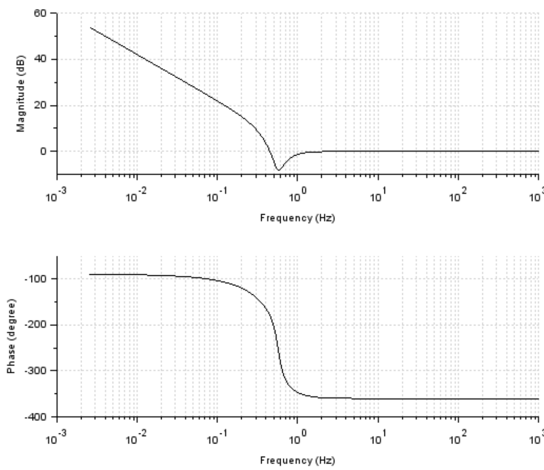


Figure 2: Bode plot of the Closed loop characteristic equation for $K = 64$

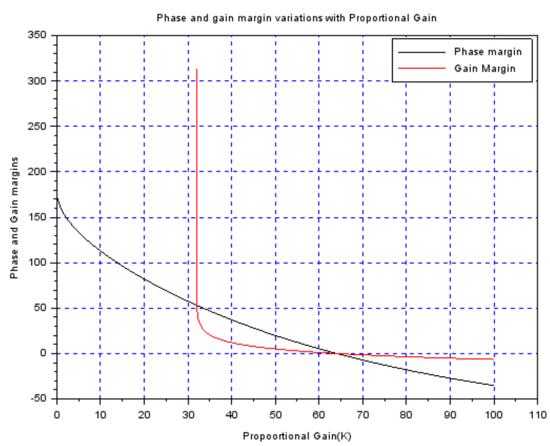


Figure 3: Gain and Phase margin plot with variation in K

loop characteristic equation has 0 phase and gain margins lie in the Right Half Plane, we observe that the system is unstable for the K in part a i.e. $K = 64$. The code is shown in the below listing:

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

// Part a
G = 1/(s*(s^2 + 4*s + 8));
sys = syslin('c', G);
K = 0.01:0.01:100;
phm = zeros(length(K), 1);
gm = zeros(length(K), 1);
fr_p = zeros(length(K), 1);
fr_g = zeros(length(K), 1);
```

```
i = 1;
for k = K
    //sys1 = (sys*k)/.syslin('c', 1, 1);
    sys1 = (1 + k*sys);
    [phm(i), fr_p(i)] = p_margin(sys1);
    [gm(i), fr_g(i)] = g_margin(sys1);
    i = i + 1;
end
temp = find(phm >= 0);
mprintf(sprintf("Gain at which Gain and phase margins are equal is: %0.2f", K(temp(length(temp)))));
mprintf("\n");
mprintf("Phase margin at K = %0.1f is: %0.2f", K(temp(length(temp))), phm(temp(length(temp))));
mprintf("\n");
mprintf("Gain margin at K = %0.1f is: %0.2f", K(temp(length(temp))), gm(temp(length(temp))));
mprintf("\n");
mprintf("Gain Crossover Frquency is: %0.2f", fr_p(temp(length(temp))));
mprintf("\n");
mprintf("Phase Crossover Frquency is: %0.2f", fr_g(temp(length(temp))));
mprintf("\n");
//sys1 = (sys*K(temp(length(temp))))/.syslin('c', 1, 1);
sys1 = (sys*K(temp(length(temp))))/.syslin('c', 1, 1);
mprintf("Closed loop system:")
disp(sys1);
mprintf("Poles of the closed loop system:");
disp(roots(sys1.den));
mprintf("Therefore, closed loop system is not stable");
mprintf("\n");
sys2 = 1 + K(temp(length(temp)))*sys;
scf();
clf();
bode(sys2);
xs2png(gcf(), "Q1a.png");
scf();
clf();
plot(K, phm, 'k');
plot(K, gm, 'r');
xlabel("Propoortional Gain(K)");
ylabel("Phase and Gain margins");
title("Phase and gain margin variations with Proportional Gain");
xgrid(2);
legend(["Phase margin", "Gain Margin"]);
xs2png(gcf(), "Q1b.png");
```

2 Question 2

2.1 Part A

The step response plots with varying positions of K_1 and K_2 are shown in Figure 4.

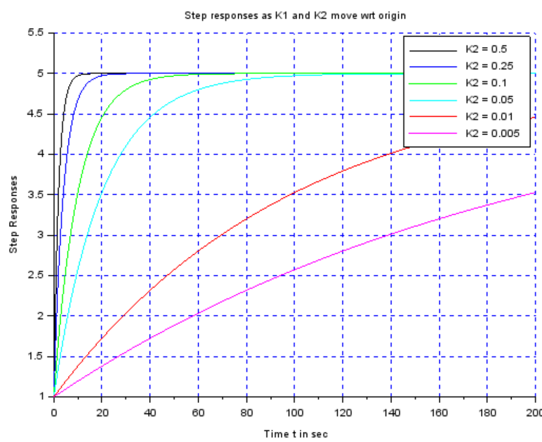


Figure 4: comparison of step responses as position of K_1 and K_2 from the origin varies

.

The step response can be given by:

$$\frac{G(s)}{s} = \frac{s+K_1}{s(s+K_2)} = \frac{K_1}{K_2 s} - \frac{K_1 - K_2}{K_2(s+K_2)} \tag{2}$$

$$\implies y(t) = \frac{K_1}{K_2}u(t) - \frac{K_1 - K_2}{K_2}e^{-K_2 t} \tag{3}$$

So, the further away K_2 is from the origin, the faster will the transient decay to give 1 as the steady state output. This is shown in Figure 4, where the output settles to the steady state value of 1 faster when K_2 is farther from the origin.

2.2 Part B

The impulse response plots with varying positions of K_1 and K_2 are shown in Figure 5.
The step response can be given by:

$$G(s) = \frac{s+K_1}{s+K_2} = 1 + \frac{K_1 - K_2}{s+K_2} \tag{4}$$

$$\implies y(t) = \delta(t) + (K_1 - K_2)e^{-K_2 t} \tag{5}$$

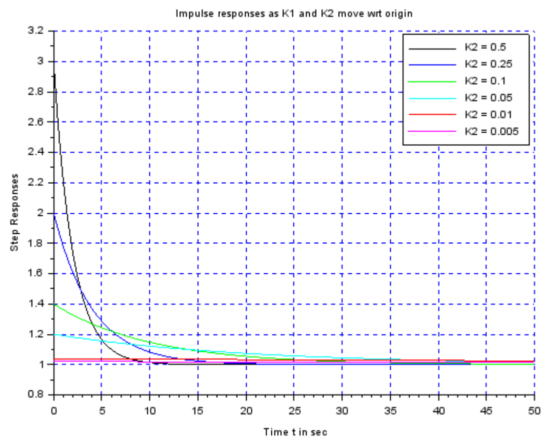


Figure 5: comparison of impulse responses as position of K_1 and K_2 from the origin varies

.

Here, we see that theoretically, we should obtain the steady state of 0 faster if K_2 is farther from the origin as the transient behaviour is shown only due to the second term of the output in Equation 5. This is totally justified by the comparison of the impulse responses we see in Figure 5.
The code for the same is shown in the below listing:

```
clear();
s = poly(0, 's');
K2 = [0.5, 0.25, 0.1, 0.05, 0.01, 0.005];
K1 = K2.*5;

t1 = 0:0.01:200;
t2 = 0:0.001:50;
y1 = zeros(length(t1), length(K2));
y2 = zeros(length(t2), length(K2));
for i = 1:length(K2)
    sys = syslin('c', (s + K1(i)), (s + K2(i)));
    y1(:, i) = csim('step', t1, sys);
    y2(:, i) = csim('impulse', t2, sys);
end
scf();
clf();
xgrid(2);
plot2d(t1, y1, 1:length(K2));
xlabel("Time t in sec");
ylabel("Step Responses");
title("Step responses as K1 and K2 move wrt origin");
legend(["K2 = 0.5", "K2 = 0.25", "K2 = 0.1", "K2 = 0.05", "K2 = 0.01", "K2 = 0.005"
    ])
xs2png(gcf(), "Q2a.png");
scf();
clf();
xgrid(2);
plot2d(t2, y2, 1:length(K2));
xlabel("Time t in sec");
ylabel("Step Responses");
title("Impulse responses as K1 and K2 move wrt origin");
legend(["K2 = 0.5", "K2 = 0.25", "K2 = 0.1", "K2 = 0.05", "K2 = 0.01", "K2 = 0.005"
    ])
xs2png(gcf(), "Q2b.png");
```

3 Question 3

The root locus of the non-shifted transfer function with no zeroes and 5 poles at -1, $\pm 2i$ and $\pm i$ are shown in Figure 6.

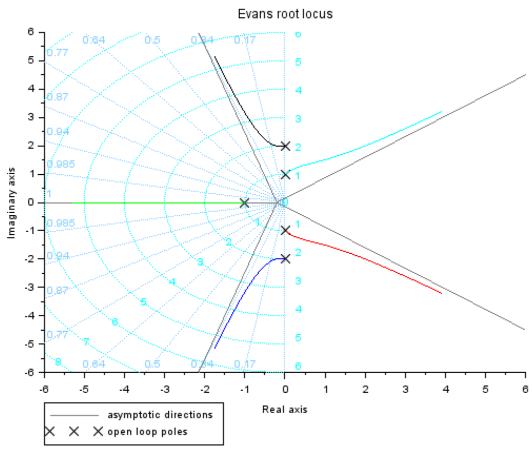
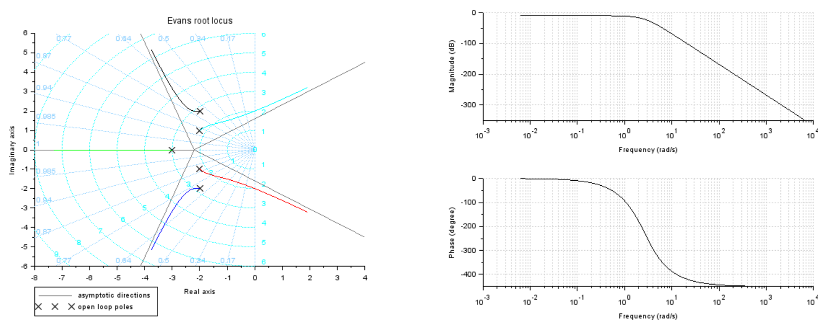


Figure 6: Root locus of base transfer function

.

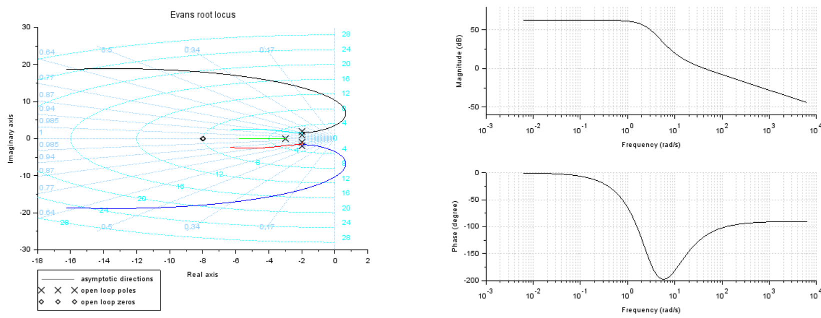
We shift the root locus by 2 to the left. The new root locus and the bode plot of the new transfer function is shown in Figure 7. Now, we see that we have 5



(a) Root locus of shifted transfer function (b) Bode plot of shifted transfer function

Figure 7: Shifted transfer function

poles and we attain an asymptote of -450^0 as frequency increases for the phase plot. We already have a phase crossover frequency by default. We need to add another. For doing this, we need to add a peak to the phase plot. This can be done by adding zeroes to the system closer to the origin than the poles. Adding 4 zeroes would nullify the contribution of 4 poles and we finally get phase plot and the root locus as shown in Figure 8.



(a) Root locus of zero added transfer function

(b) Bode plot of zero added transfer function

Figure 8: Zero added transfer function

we observe in the root locus that the new system has a root locus which intersects the imaginary axis at two points. The zeroes added were all at -8, 4 in total. The code is shown in the below listing:

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

G = 40/((s + 1)*(s^2 + 4)*(s^2 + 1));
sys = syslin('c', G);
scf();
clf();
evans(sys, 100);
sgrid();
xs2png(gcf(), "Q3a.png");

k = 2;
G1 = 40/(((s+k) + 1)*((s+k)^2 + 4)*((s+k)^2 + 1));
sys1 = syslin('c', G1);
scf();
clf();
evans(sys1, 100);
sgrid();
xs2png(gcf(), "Q3b.png");
scf();
clf();
bode(sys1, 'rad');
xs2png(gcf(), "Q3b_bode.png");
[phm, fr] = p_margin(sys1);

sys2 = sys1*((s+8)^4);
scf();
clf();
bode(sys2, 'rad');
[phm, fr] = p_margin(sys2);
xs2png(gcf(), "Q3c_bode.png");
scf();
clf();
evans(sys2);
sgrid();
xs2png(gcf(), "Q3c.png");
```

4 Question 4

The asymptotic points are at 1, 5, 10, and 100. There is a positive slope after 1 and negative/constant slopes after it. Therefore, 5, 10 and 100 form the pole points. Hence, the transfer function comes out to be:

$$G(s) = \frac{s+1}{(s+5)(s+10)(s+100)} \tag{6}$$

The magnitude and phase plot is shown in Figure 9.

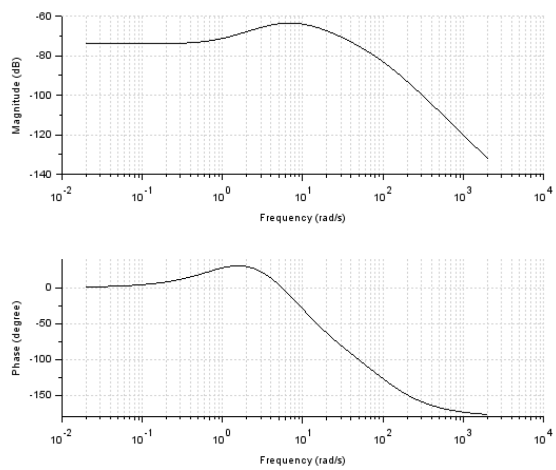


Figure 9: Magnitude and phase plot of $G(s)$

Its observed that the obtained magnitude plot matches well with the required magnitude plot shown in the question.
The code is shown in the below listing:

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

G = (s + 1)/((s + 5)*(s + 10)*(s + 100));
sys = syslin('c', G);

scf();
clf();
bode(sys, 0.0032, 320, 0.0001, 'rad');
xs2png(gcf(), "Q4.png");
```