Indian Institute of Technology Bombay

EE 324: Control Systems Lab

Problem Sheet 8 Report

March 20, 2021

Student Name	*Roll Number*
Mantri Krishna Sri Ipsit	180070032

Contents

1 Question 1

The open loop transfer function given is

$$G(s) = \frac{1}{s(s^2 + 4s + 8)}$$

1.1 Part a

Assuming only a proportional controller, the closed loop characteristic equation will be

$$T = 1 + KG(s) = 1 + \frac{K}{s(s^2 + 4s + 8)}$$

Using Scilab, the value of gain K for which the closed loop characteristic equation has gain and phase margin equal to zero is $K = 64$.

1.2 Part b

The variation of both gain and phase margin of the closed loop characteristic equation is given by

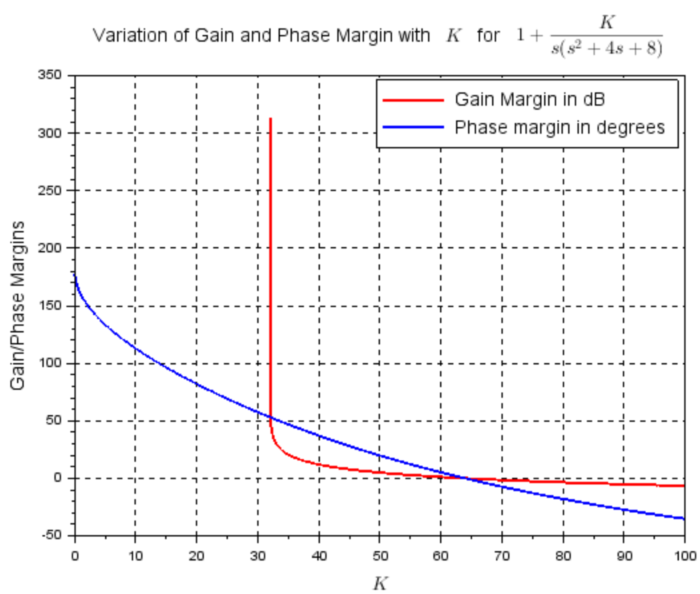

Figure 1: Gain and Phase margin vs K

We cannot have gain margin 0 and phase margin non-zero and vice versa because both are monotonic functions and intersect at only 2 points, outof which, at one intersection they are 0. At the other intersection, they are positive. Hence it is not possible for one to them to be zero and at the same time the other one to be non-zero.

1.3 Part c

The closed loop system for $K = 64$ is not stable. This is concluded by looking at zeros of the closed loop characteristic equation at $K = 64$ which come out as

$$-4.9766045, 0.4883022 \pm j3.5527081$$

The following code has been used to get the results:

```
clc; clear;
s = poly(0, 's');
g = 1 / (s * (s^2 + 4*s + 8));
// --------------------------------------------
// Part a
K = 0:0.01:100;
k_req = 0;
gms = [];
pms = [];
for i=1:size(K, 2)
    k = K(i);
    tf = 1 + k * g;
    T = syslin('c', tf);
    gm = g_margin(T);
    pm = p_margin(T);
    gms = [gms, gm];
    pms = [pms, pm];
    if abs(gm - pm) <= 1e-8 then
        k_req = k;
     end
end
scf();
plot(K, gms, 'r-', 'LineWidth', 2);
plot(K(2:size(K, 2)), pms, 'b-', 'LineWidth', 2);
xgrid();
xlabel("$K$", 'fontsize', 3);
ylabel("Gain/Phase Margins", 'fontsize', 3);
title(["Variation of Gain and Phase Margin with", "$K$", "for",...
"$1 +\frac{K}{s(s^2 + 4s + 8)}$"], "fontsize", 3);
L = legend(["Gain Margin in dB", "Phase margin in degrees"]);
L.font_size = 3;
disp(sprintf("The value of K for which the gain and the phase margins become zero is %.4f",k_req));
// --------------------------------------------
// Part c
tf = 1 + (k_req * g);
T = syslin('c', tf);
[z, p, g] = tf2zp(T);
disp("The closed loop poles at K = 64 are");
disp(z);
```

2 Question 2

We have a system with the transfer function

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

2.1 Part a

We keep the ratio $\frac{K_1}{K_2} = 5$ as a constant and then plot the step response by varying the location of the pole. The step resposne of this system in time domain is given by

$$c(t) = \frac{K_1}{K_2} - \frac{K_1 - K_2}{K_2} e^{-K_2 t}$$

By looking at the time domain expression, we can conclude that as the pole moves farther away from the origin, the transients decay faster. This is verified by observing the figure below.

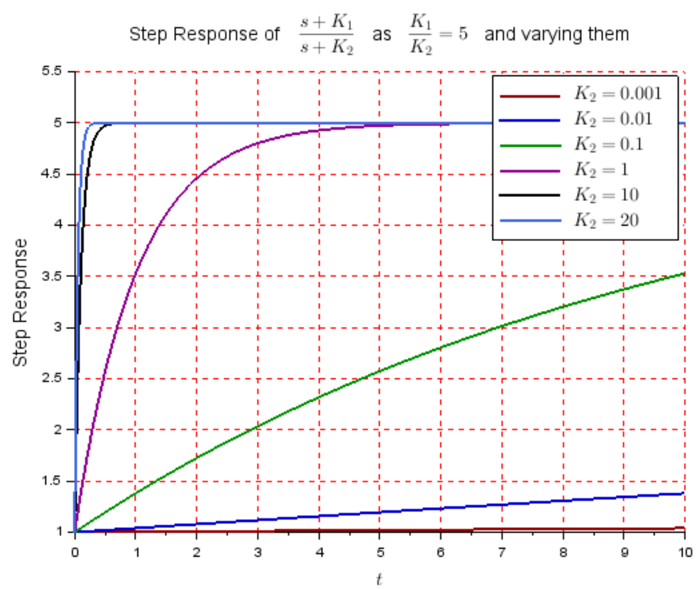

Figure 2: Step response transients

2.2 Part b

Here again we keep the ratio $\frac{K_1}{K_2} = 5$ as a constant and then plot the impulse response by varying the location of the pole. The impulse resposne of this system in time domain is given by

$$h(t) = \delta(t) + (K_1 - K_2)\, e^{-K_2 t}$$

By looking at the time domain expression, we can conclude that as the pole moves farther away from the origin, the transients decay faster. This is verified by observing the figure below.

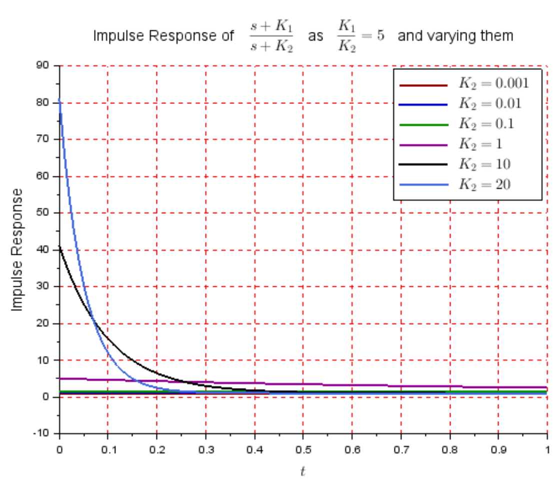

Figure 3: Impulse response transients

The above figures have been generated using the following code:

```
clc; clear;
s = poly(0, 's');
K2 = [0.001, 0.01, 0.1, 1, 10, 20];
K1 = 5 * K2;
t = 0:0.01:10;
plotcolors = ["scilabred4", "scilab blue2", "scilab green4","scilabmagenta4", "black", ...
"royalblue"];
scf();
for i=1:size(K2, 2)
    k2 = K2(i);
    k1 = K1(i);
    g = (s + k1) / (s + k2);
    G = syslin('c', g);
    plot2d(t, csim('step', t, G), style=[color(plotcolors(i))]);
    e = gce();
    e.children.thickness = 2.5;
    xgrid(5);
end
xlabel("$t$", 'fontsize', 3);
ylabel("Step Response", 'fontsize', 3);
title(["Step Response of", "$\frac{s+K_1}{s+K_2}$", "as",...
"$\frac{K_1}{K_2} = 5$", "and varying them"], "fontsize", 3);
L = legend(["$K_2 = 0.001$", "$K_2 = 0.01$", "$K_2 = 0.1$", "$K_2 = 1$",...
"$K_2 = 10$", "$K_2 = 20$"]);
L.font_size = 3;
// ----------------------------------------
// Part b
t = 0:0.01:1;
scf();
for i=1:size(K2, 2)
    k2 = K2(i);
    k1 = K1(i);
    g = (s + k1) / (s + k2);
    G = syslin('c', g);
    plot2d(t, csim('impuls', t, G), style=[color(plotcolors(i))]);
    e = gce();
    e.children.thickness = 2.5;
    xgrid(5);
end
xlabel("$t$", 'fontsize', 3);
ylabel("Impulse Response", 'fontsize', 3);
title(["Impulse Response of", "$\frac{s+K_1}{s+K_2}$", "as",...
"$\frac{K_1}{K_2} = 5$", "and varying them"], "fontsize", 3);
L = legend(["$K_2 = 0.001$", "$K_2 = 0.01$", "$K_2 = 0.1$", "$K_2 = 1$",...
"$K_2 = 10$", "$K_2 = 20$"]);
L.font_size = 3;
h = gca(); // get current axes
h.data_bounds = [0, -0.8 ; 1, 90];
```

3 Question 3

3.1 Part a

The open loop transfer function taken is

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

Its root locus is as follows:

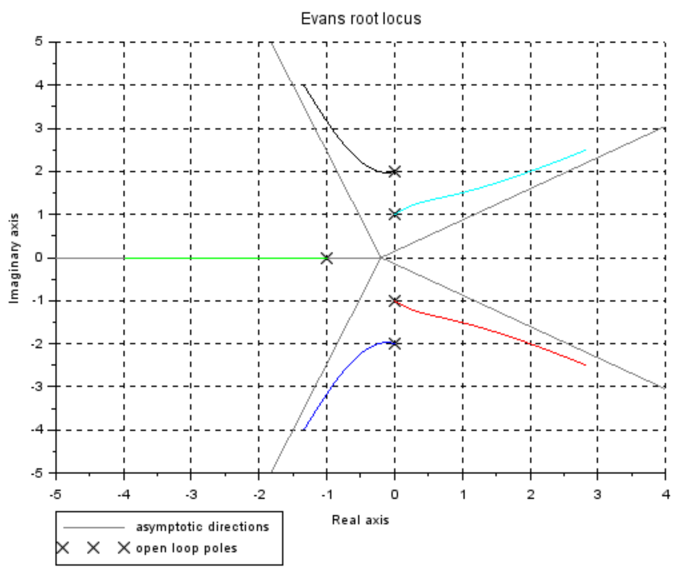

Figure 4: Root Locus

3.2 Part b

After shifting the origin to -5, we have the new transfer function as

$$G_1(s) = \frac{1}{(s+6)((s+5)^2+1)((s+5)^2+4)}$$

Its root locus and bode plots are as follows:

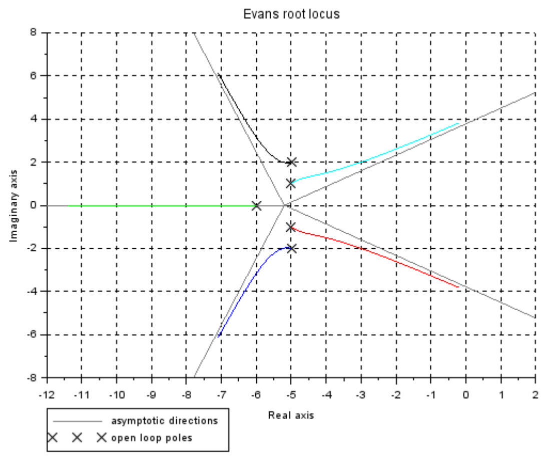

Figure 5: Root Locus

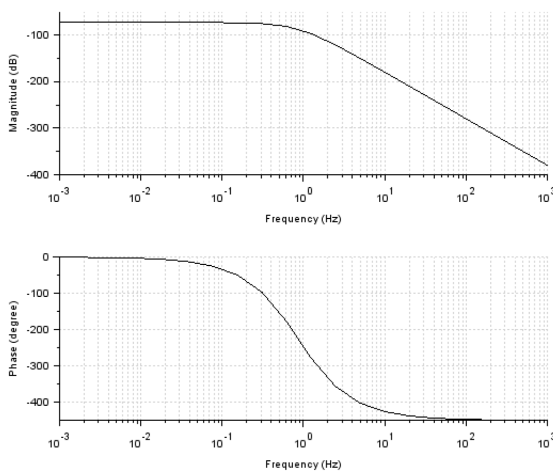

Figure 6: Bode Plot

3.3 Part c

By looking at the above bode plot, we can conclude that we already have one phase crossover. We need one more. For that to happen, we need to add zeros such that the phase response rises after it falls below 180 degrees. This means we have to add 4 zeros atleast (as we have 4 poles) and we have to add the zeros sufficiently left of the farthest pole from the origin. I add 4 zeros located at -20π to the origin-shifted transfer function. The new transfer function is now:

$$G_2(s) = \frac{(s + 20\pi)^4}{(s + 6)((s + 5)^2 + 1)((s + 5)^2 + 4)}$$

The modified bode plot with 2 phase-crossover frequencies is as follows:

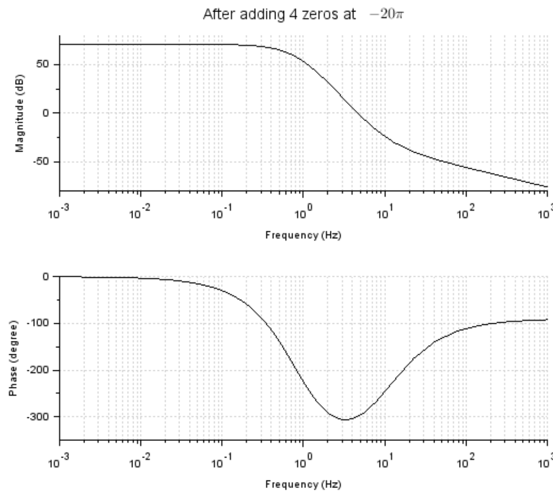

Figure 7: Bode plot After adding 4 zeroes at -20π

3.4 Part d

The root locus of the above transfer function is shown below. By looking at the figure, we can conclude that this new system satisfies the problem statement. All the above plots were made using the following code:

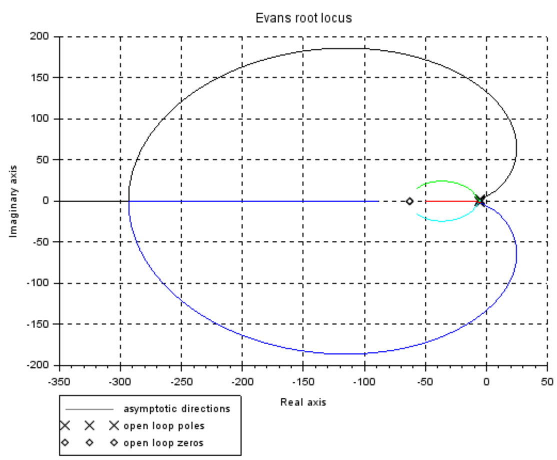

Figure 8: Root Locus After adding 4 zeroes at -20π

```
clc; clear;
s = poly(0, 's');
//----------------------
// Part a
p1 = -1;
p2 = %i;
p3 = -%i;
p4 = 2*%i;
p5 = -2*%i;
g = 1 / ((s-p1)*(s-p2)*(s-p3)*(s-p4)*(s-p5));
G = syslin('c', g);
scf();
evans(G, 1000);
xgrid();
//-----------------------
// Part b
shift = 5;
g_shifted = 1 / ((s-p1 +shift)*(s-p2 +shift)*(s-p3 +shift)*(s-p4 +shift)...
*(s-p5 +shift));
G_shifted = syslin('c', g_shifted);
scf();
evans(G_shifted, 10000);
xgrid();
scf();
bode(G_shifted);
//-------------------------------------------------------------
// Part c
z1 = 10*2*%pi;
g_new = g_shifted * (s + z1)^4;
G_new = syslin('c', g_new);
scf();
bode(G_new);
title(["After adding 4 zeros at", "$-20\pi$"], 'fontsize', 3);
//--------------------------------------------------------------
// Part d
scf();
evans(G_new, 9000);
xgrid();
h = gca(); // get current axes
h.data_bounds = [-350, -200 ; 1, 200];
```

4 Question 4

As per the asymptotic magnitude plot given, we can conclude that there are 4 break frequency points, at 1, 5, 10, 100 respectively. Out of these, at $\omega = 1$, the magnitude response is locally increasing and at other break frequency locations, the magnitude response is locally decreasing. Using this info, we can conclude that the following transfer functions is a good candidate for the given magnitude response:

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

The bode plot of this function is as follows: This was plotted using the following code:

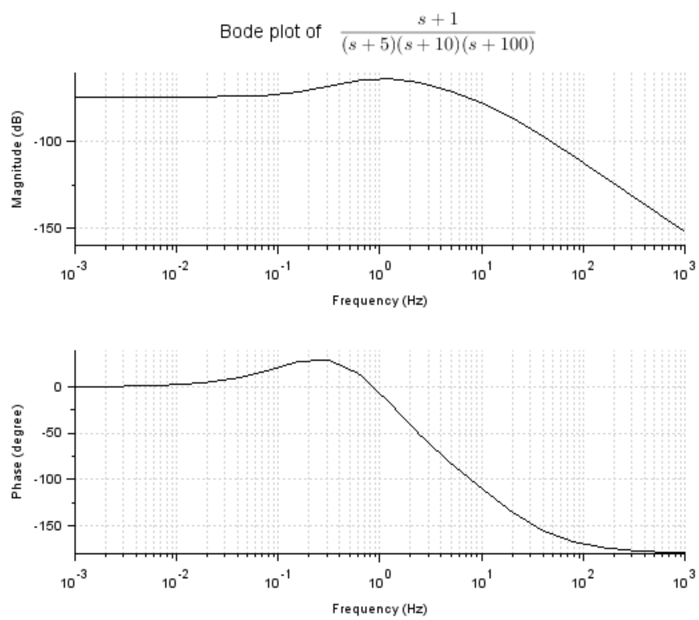

Figure 9: Bode plot

```
clc; clear;
s = poly(0, 's');
z = 1;
p1 = 5;
p2 = 10;
p3 = 100;
g = (s + z) / ((s + p1) * (s + p2) * (s + p3));
G = syslin('c', g);
scf();
bode(G);
title(["Bode plot of", "$\frac{s+1}{(s+5)(s+10)(s+100)}$"], 'fontsize', 3);
```