EE324 Control Systems Lab

Problem Sheet 8

Prakhar Diwan, Roll no. 180100083

1. Problem 1

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

1.1. Part (a)

By the use of scilab defined functions, we obtain the following plot for the open loop transfer function $G(s)$:

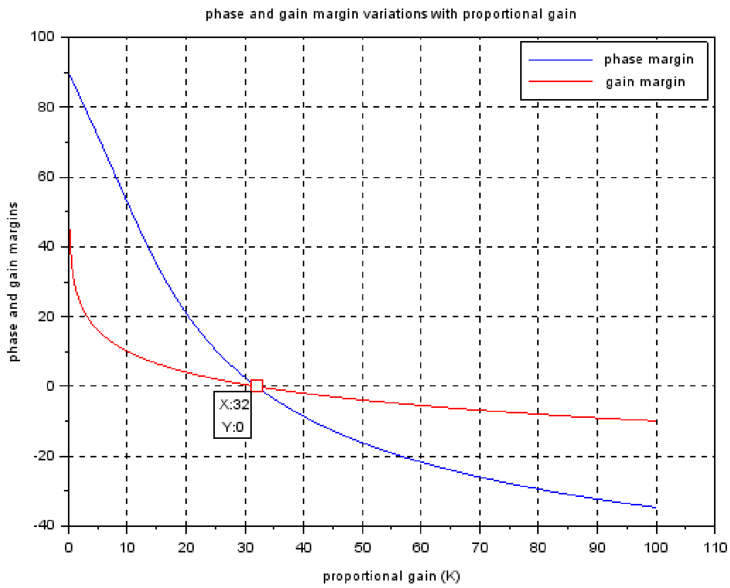

Figure 1: Phase Margin and Gain Margin vs Proportional Gain (K)

As is visible from the above plot, the value of K (gain) for which the gain and phase margin are both equal to zero is 32.

1.2. Part (b)

No, as is visible from the figure 1, so we cannot have a gain (K) for which gain margin is 0 but phase margin is non-zero or vice-versa. Both of them cross the x-axis only once and that too together.

1.3. Part (c)

Poles of closed loop system when $K = 32$ are $-4,\ \pm 2.8284i$
Since we have poles of the closed loop system on the imaginary axis, the system is marginally stable.
The bode plot for the above system is shown below:

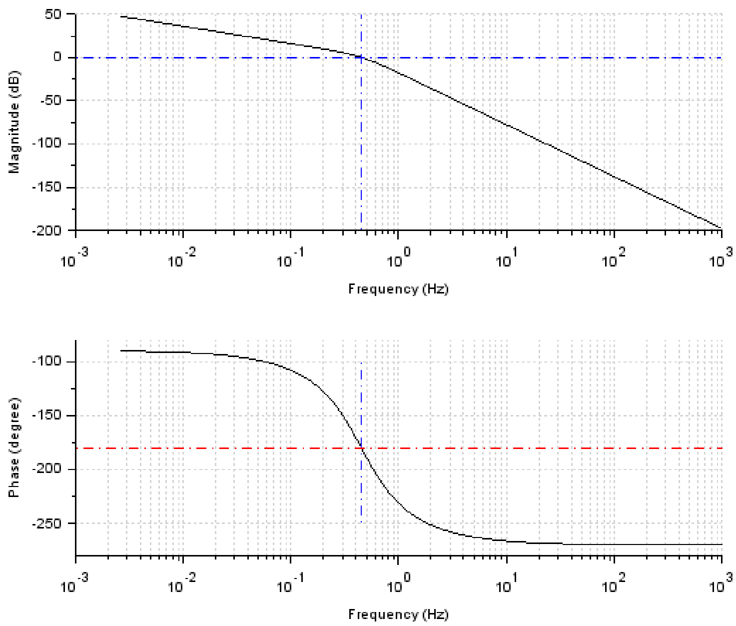

Figure 2: Bode plot of system for K = 32

Scilab Code for Problem 1

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

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);
frp = zeros(length(K), 1);
frg = zeros(length(K), 1);

i = 1;
for  k = K
    sys1 = (k*sys);
    [phm(i),frp(i)] = p_margin(sys1);
    [gm(i), frg(i)] = g_margin(sys1);
    i = i + 1;
end

scf();
```

```
plot(K,phm,'b');
plot(K,gm,'r');
xtitle("phase and gain margin variations with proportional
gain","proportional gain (K)", "phase and gain margins");
xgrid();
legend(["phase margin", "gain margin"]);

// Verifying that system is marginally stable
K = 32;
sysC = (K*sys);
charC = 1 + sysC;
poles = roots(charC.num);
disp(poles);
show_margins(sysC);
// bode(sysC);
```

2. Problem 2

2.1. Part (a)

$$G(s) = \frac{s+K_1}{s+K_2}$$
$$K_1 = 5K_2$$
$$\therefore\ G(s) = \frac{s+5K_2}{s+K_2}$$

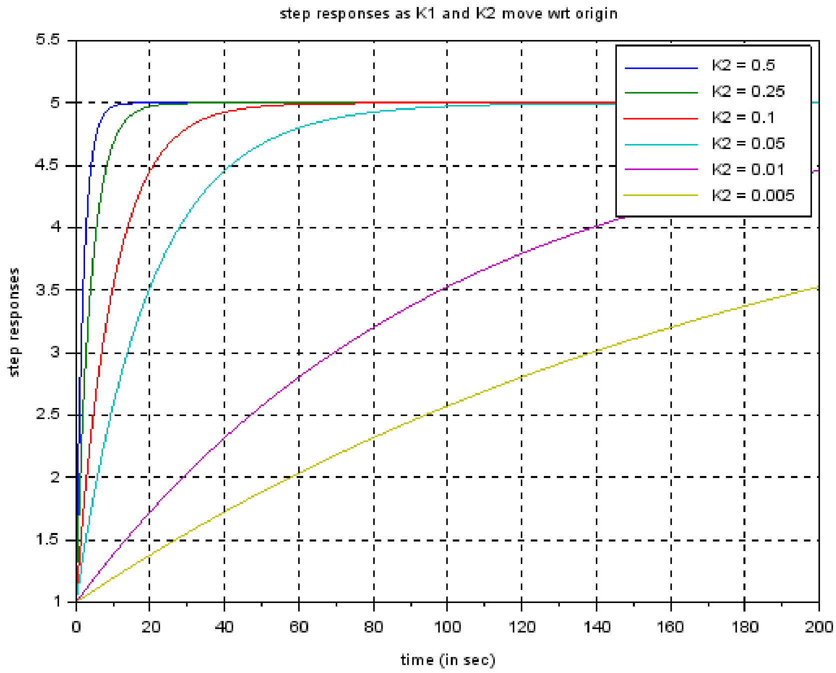

Figure 3: Step responses as pole and zero move wrt origin

We observe that as K_2 increases, the rise time of the step response decreases, that is the transient response kicks in earlier as we've increased the magnitude of the pole, something which is not desirable from a lag compensator as we need it to just affect the steady state value and not the immediate transient response.

2.2. Part (b)

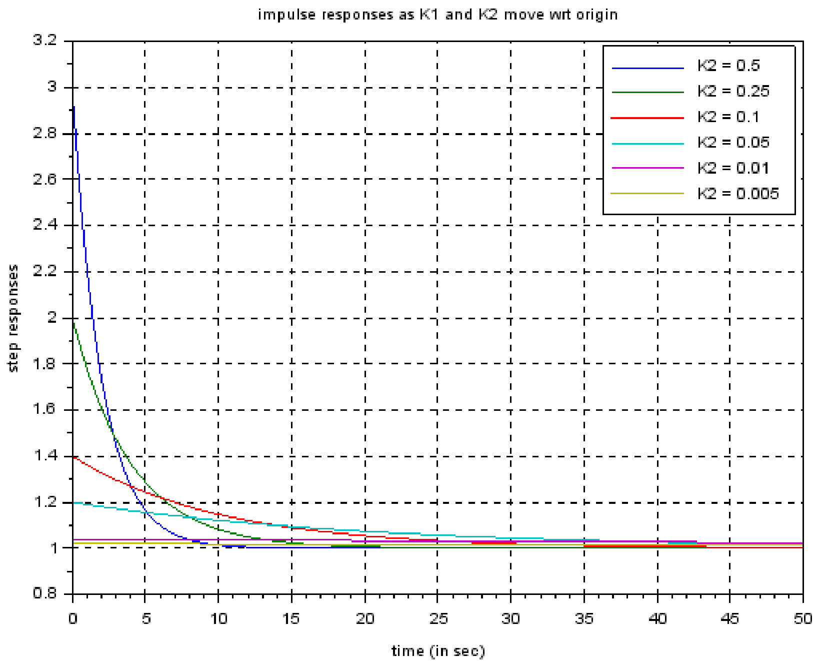

Figure 4: Impulse responses as pole and zero move wrt origin

The impulse response of the system decays much faster and the initial value of response (@ t = 0) also increases as we increase the value of K_2.

Scilab Code for Problem 2

```
clear();
s = %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();
xgrid();
plot(t1, y1, 1:length(K2));
```

```
xtitle("step responses as K1 and K2 move wrt origin", "time (in sec)",
 "step responses");
legend(["K2 = 0.5", "K2 = 0.25", "K2 = 0.1", "K2 = 0.05", "K2 = 0.01",
 "K2 = 0.005"])

scf();
xgrid();
plot(t2, y2, 1:length(K2));
xtitle("impulse responses as K1 and K2 move wrt origin", "time (in
sec)", "step responses");
legend(["K2 = 0.5", "K2 = 0.25", "K2 = 0.1", "K2 = 0.05", "K2 = 0.01",
 "K2 = 0.005"])
```

3. Problem 3

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

3.1. Part (a)

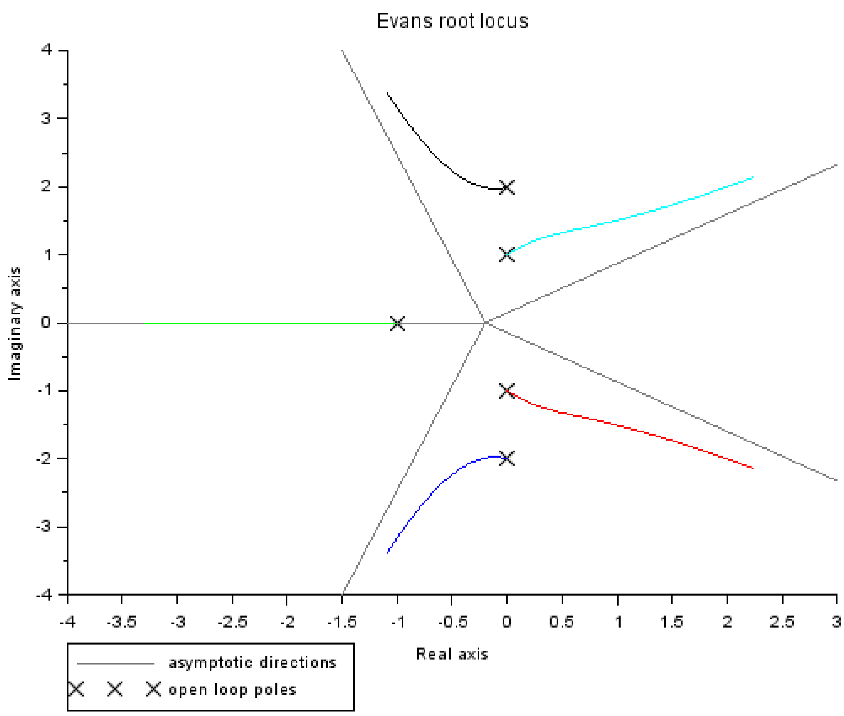

Figure 5: Root Locus for G(s)

3.2. Part (b)

The transfer function becomes:

$$G_2(s) = \frac{4}{(s+4)((s+3)^2+1)((s+3)^2+4)}$$

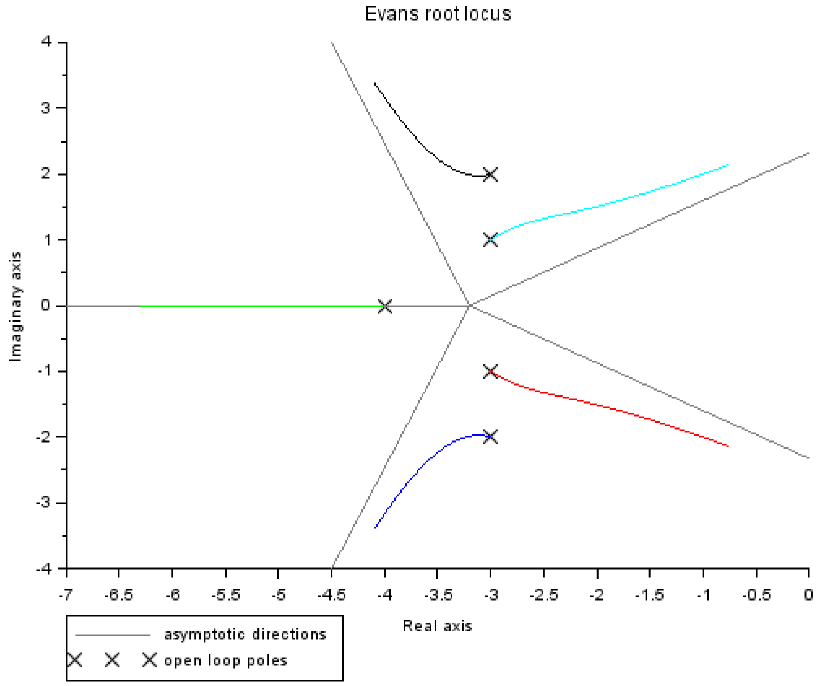

Figure 6: Root locus for 3b

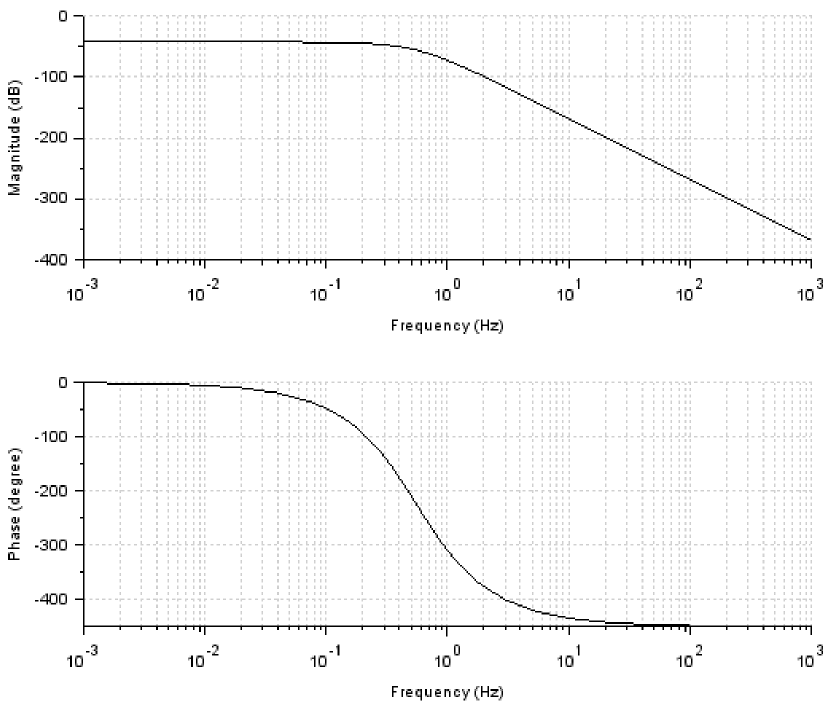

Figure 7: Bode plot for 3b

3.3. Part (c)

For 2 phase crossover frequencies we use the following transfer function:

$$G_3(s) = \frac{4(s+16)^4}{(s+4)((s+3)^2+1)((s+3)^2+4)}$$

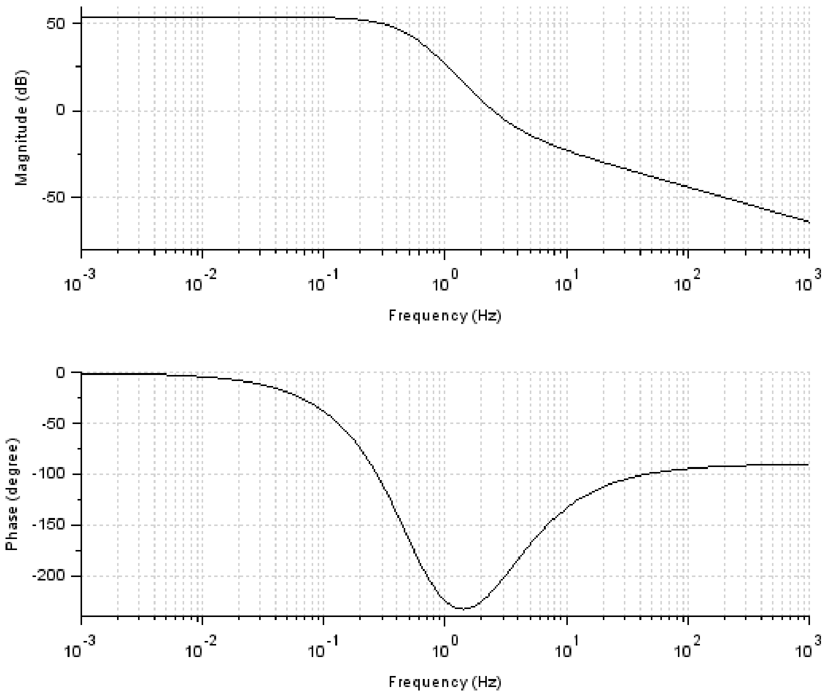

Figure 8: 2 phase crossover frequencies for $G_3(s)$

3.4. Part (d)

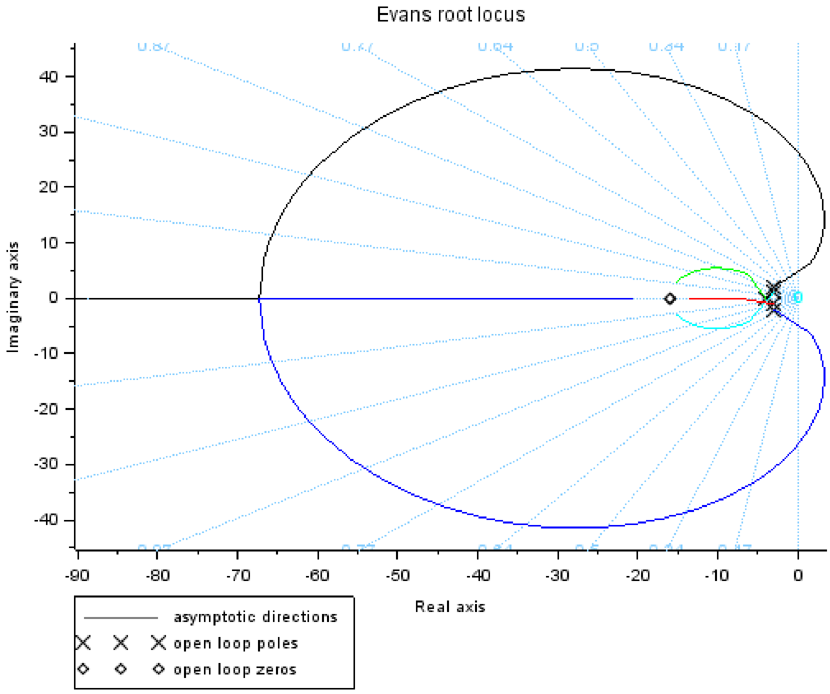

Figure 9: Root locus for $G_3(s)$

We can see that there are indeed 2 positions wherein a single branch of the root locus cuts the imaginary axis, hence there are 2 frequencies which result in phase reversal.

Scilab Code for Problem 3

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

G = 4/((s+1)*(s^2 + 4)*(s^2 + 1));
sys = syslin('c', G);

scf();
evans(sys, 100);

k = 3;
G1 = 4/(((s+k)+1)*((s+k)^2+4)*((s+k)^2+1));
sys1 = syslin('c', G1);

scf();
evans(sys1, 100);

scf();
bode(sys1);
```

```
sys2 = sys1*((s+16)^4);

scf();
bode(sys2);

scf();
evans(sys2,1000);
sgrid();
```

4. Problem 4

For matching the magnitude plot, we can obtain transfer function as follows:

$$G(s) = K \times \frac{s+1}{(s+5)(s+10)(s+100)}$$
$$K = 10^{(-75/20)} \times 5000 \; \textit{chosen for appropriate level}$$

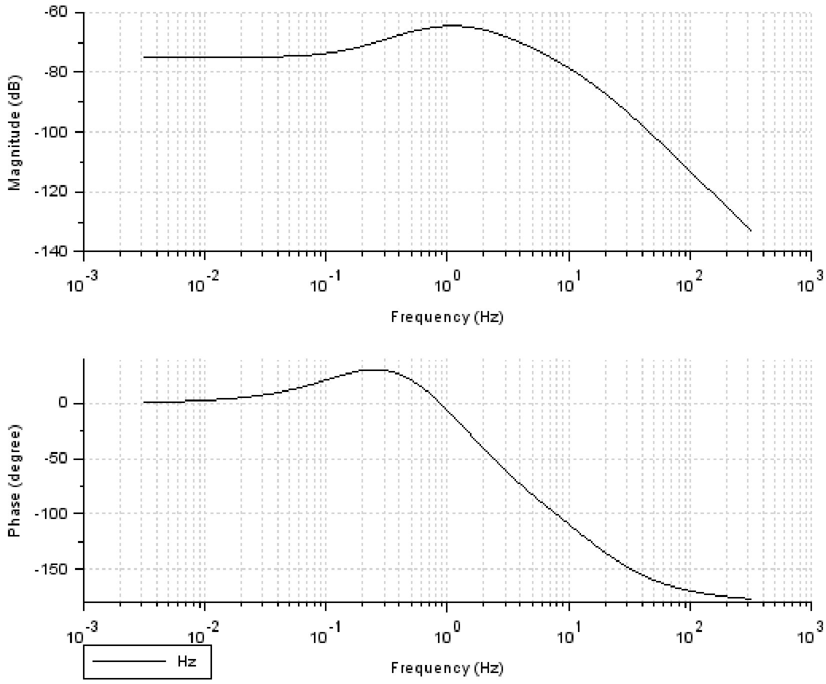

Figure 10: Minimum phase system meeting the magnitude specifications

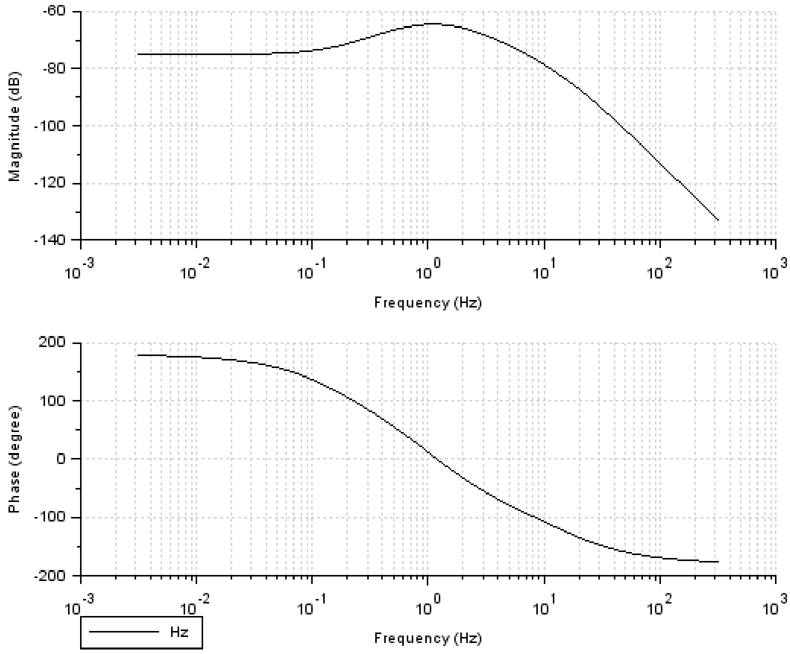

Figure 11: Non-minimum phase system meeting the magnitude specifications

```
clear();
s = poly(0,'s');
k = (10^(-75/20))*5000;
G = k*(s-1)/((s+5)*(s+10)*(s+100)); // s+-1 acc to min or non min
phase
sys = syslin('c', G);

scf();
clf();
bode(sys, 0.0032, 320, 0.0001, 'Hz');
```