EE 324: Problem Sheet-9

Abhilaksh Kumar, 18D070035

April 4, 2021

1 Problem-1

The Nyquist plots for the given transfer functions are:-

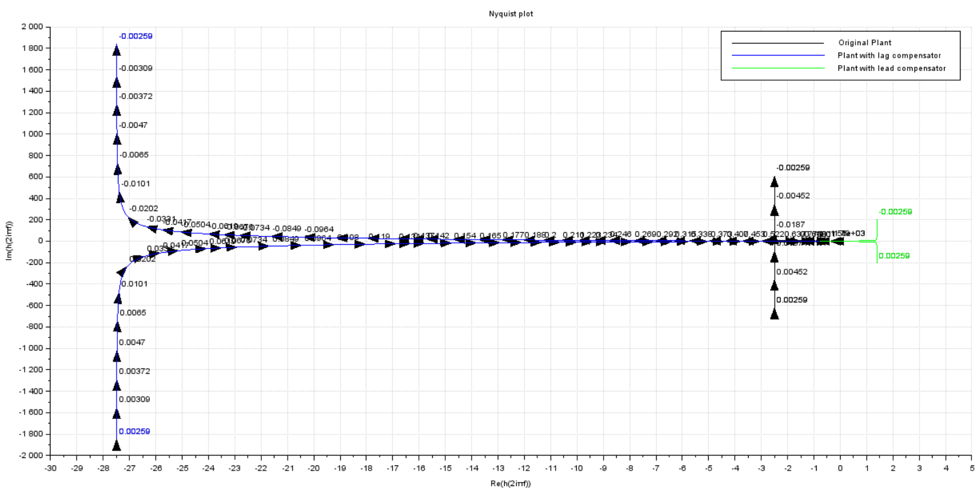


Figure 1: The Nyquist Plots

Gain margin is smallest for the lag compensator and is the largest for the lead compensator.
The Phase Margin is also the smallest for the lag compensator and the largest for the lead compensator.

Code

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

G = syslin('c', 10/(s*(s/5+1)*(s/20+1)));

G1 = syslin('c', (s+3)/(s+1)*(10/(s*(s/5+1)*(s/20+1))));          // with lag

G2 = syslin('c', (s+1)/(s+3)*(10/(s*(s/5+1)*(s/20+1))));        // with lead c

nyquist(G)
nyquist(G1)
nyquist(G2)

nyquist([G;G1;G2],["Original Plant";"Plant with lag compensator"; "Plant with

[gm ,w_gcf]=g_margin(G);
[gm1 ,w_gcf1]=g_margin(G1);
[gm2 ,w_gcf2]=g_margin(G2);


[pm ,w_pcf]=p_margin(G);
[pm1 ,w_pcf1]=p_margin(G1);
[pm2 ,w_pcf2]=p_margin(G2);
```

2 Problem-2

For this filter, we want a minima at 50Hz and then the magnitude part of the bode plot graph rise to a constant value. For this, we fist define a numerator which has a minima at 50Hz and is constant before that:-

$$N^r(s) = (s^2 + 50^2);$$

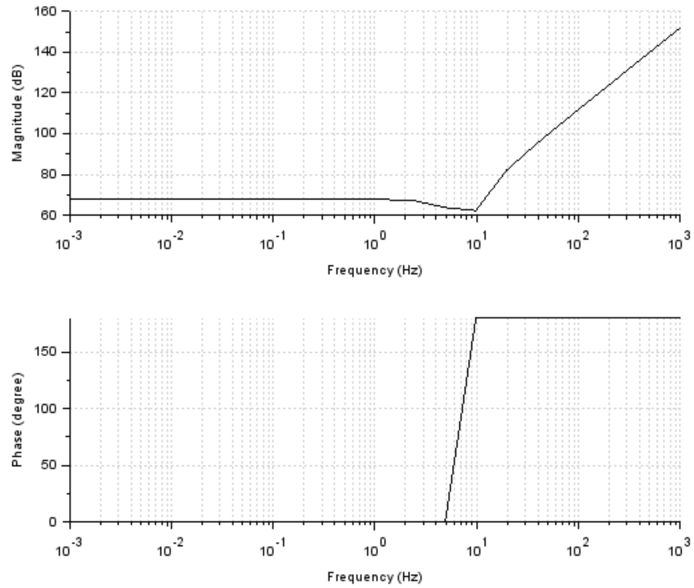


Figure 2: Bode Plot of the Numerator

Now, in the denominator we need a function that has the same value of slope for frequencies>>50Hz. The function plot should also be a constant for frequencies<<50Hz and shouldn't reach a minimum at 50Hz, or it would just end up cancelling the original minima. Thus, for the denominator we use:-

$$D^r(s) = (s^2 + 90s + 50^2);$$

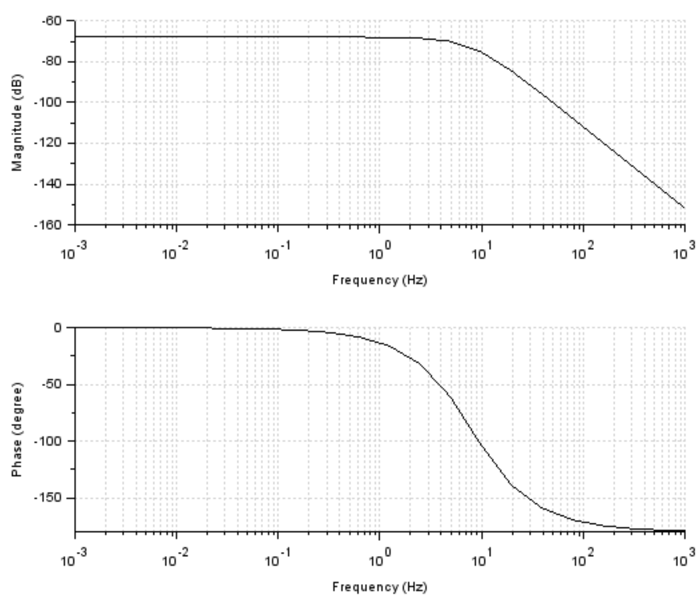


Figure 3: Bode Plot of the 1/Denominator

Thus, the transfer function of our notch filter becomes:-

$$\boxed{G(s) = \frac{s^2 + 50^2}{s^2 + 90s + 50^2}}$$

The bode plot obtained for the notch filter is as follows:-

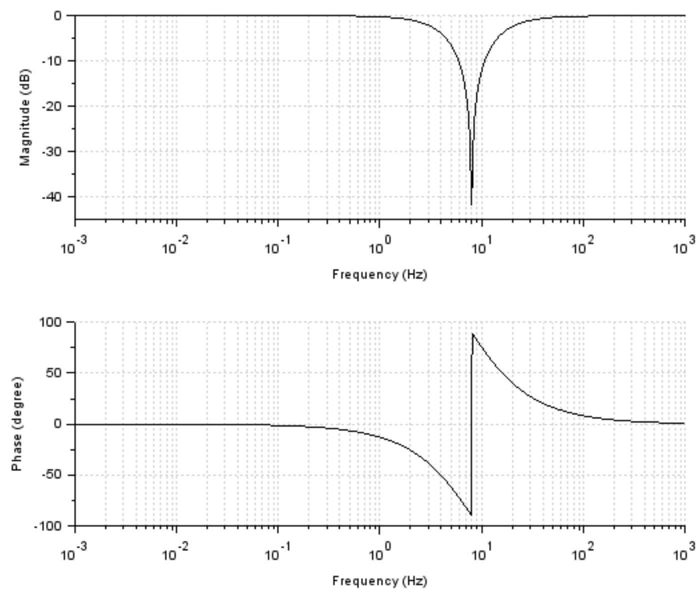


Figure 4: Bode Plot of The Notch Filter

The steepness of the slope can be increased by decreasing the value of ζ(the coeffecient of s) in the denominator. As an example, the bode plot of the transfer function:-

$$G(s) = \frac{s^2 + 50^2}{s^2 + 30s + 50^2}$$

The bode plot obtained for the notch filter is as follows:-

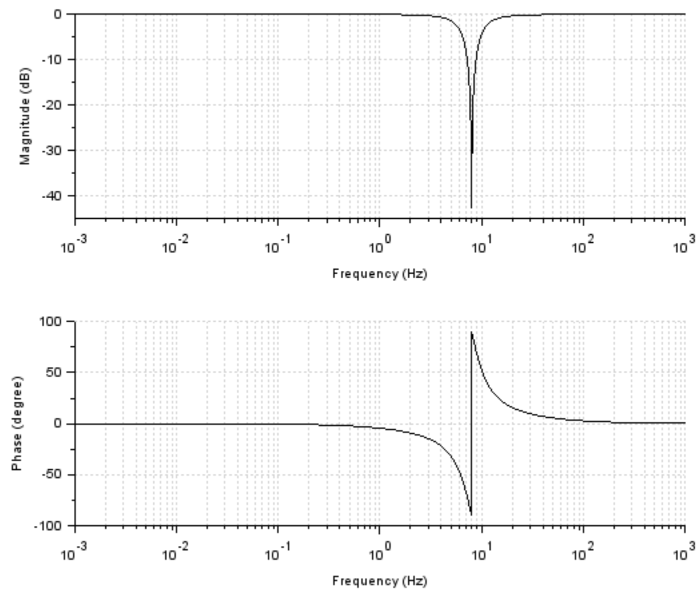


Figure 5: Comparative Bode Plot of The Notch Filter

Code

```
s = poly(0,'s');
G = syslin('c', (s^2+50^2)/(s^2+90*s+50^2));
scf(0);
bode(G);
xs2png(0, 'Q2.png')

// Numerator
scf(1);
N = syslin('c', (s^2+50^2),1);
bode(N);
xs2png(1, 'Q2_NUM.png')

// Denominator
scf(2);
D = 1/(s^2+90*s+50^2);
bode(syslin('c', D));
xs2png(2, 'Q2_DEN.png')

// Steeper Bode Plot
scf(3);
G_s = syslin('c',(s^2+50^2)/(s^2+30*s+50^2));
```

```
bode(G_s);
xs2png(3,  'Q2_comp.png')
```

3 Problem-3

(a)

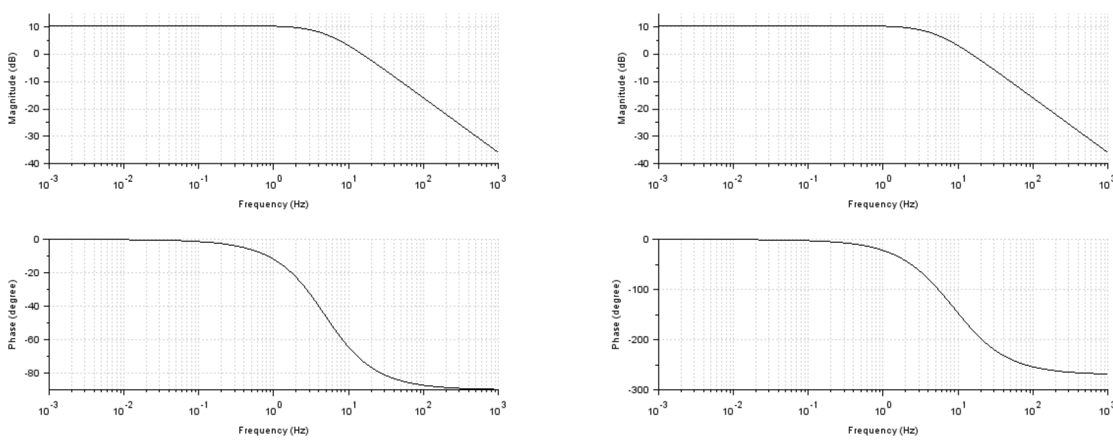
The minimum delay required to destabilize the transfer function can be found as:-

$\frac{PM}{\omega_{gcf}}$ where ω_{gcf} and PM are the Gain crossover Frequency and Phase Margin for $C(s)$

Using this, the obtained time delay is $\boxed{t_d = 0.019s}$.

We'll use pade's approximation because scilab doesn't have exponential function for which we get the minimum Time delay to be **0.0286 sec**.

The comparative bode plots for both the systems is:-



(a) Bode plot of $C(s)$ (b) Bode plot of $C(s)G(s)$

Figure 6: Bode plot comparison of original and delayed version of open loop transfer function

Code

```
s = poly(0,'s');
G = 100/(s+30);
T = .028;
pade= (1-s*T/2)/(1+s*T/2);

G_delayed = syslin('c', G*pade);
```

```
G = syslin('c', G);

scf(0);
bode(G)

scf(1);
bode(G_delayed)
```

4 Problem-4

(a)

Using root locus The gain margin thus obtained by root locus is $\boxed{K = 6}$

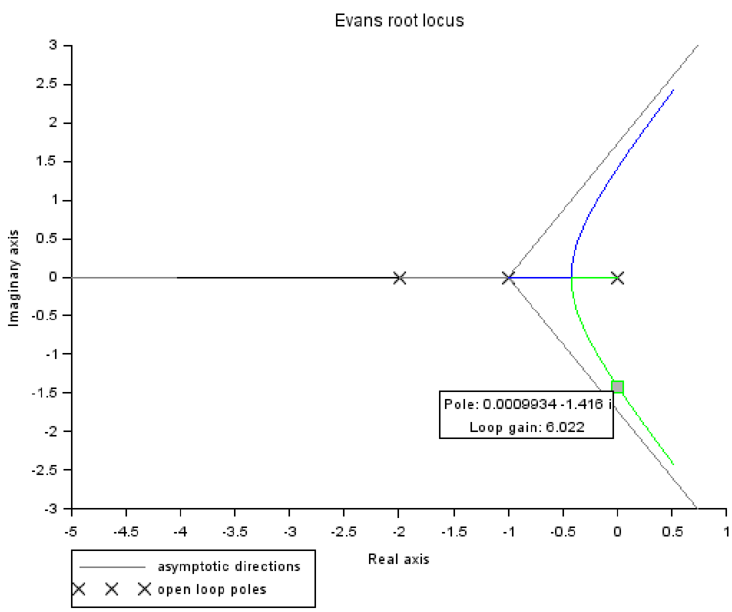


Figure 7: Root Locus of $G(s)$

(b)

(c)

the phase plot is asymptotically at -180^o for the frequency in 1 rad/s to 2 rad/s. Taking the mean, we get the magnitude of $G(s)$ at 1.5 rad/s to be 0.148 implying a gain margin of 12.154 dB

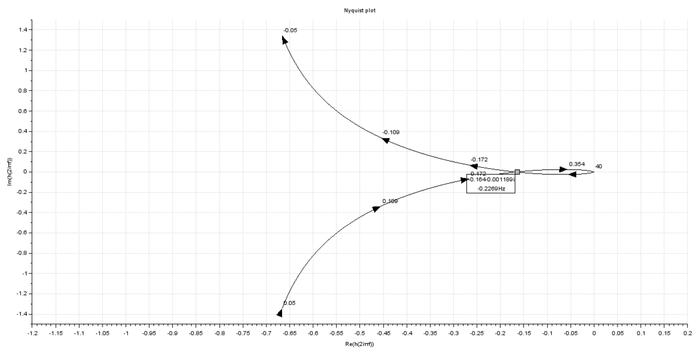


Figure 8: Nyquist plot of $G(s)$

(d)

Using the actual plot, we get gain margin=15.6dB

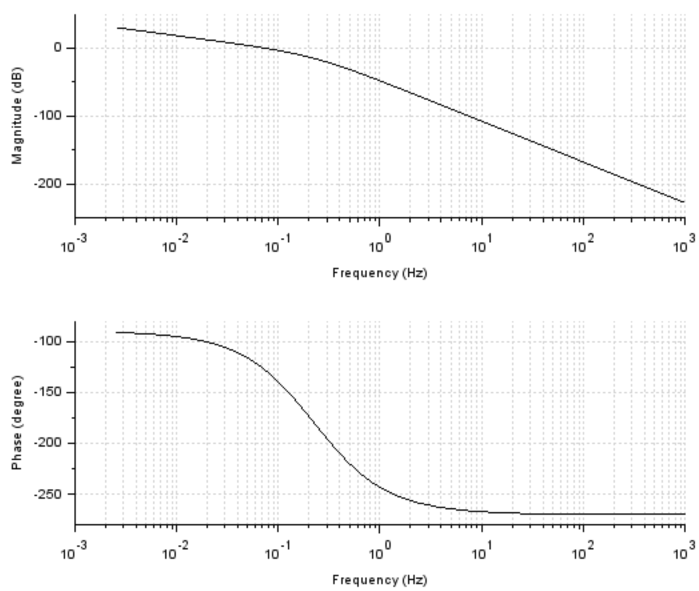


Figure 9: The Obtained Bode Plot

Code

```
s = poly(0,'s');
G = 1/(s^3+3*s^2+2*s);
G = syslin('c',G);
```

```
// rlocus
scf(0);
evans(G,25);

kpure(G)

// Nyquist
scf(1);
nyquist(G,.05,40);    //~.16

// Bode actual
scf(2);
bode(G)
[gm,fr] = g_margin(G)          // gm = 15.5dB


// bode asymptodic
scf(3);
bode(G,'rad');
bode_asymp(G)
```

5 Problem-5

(a)

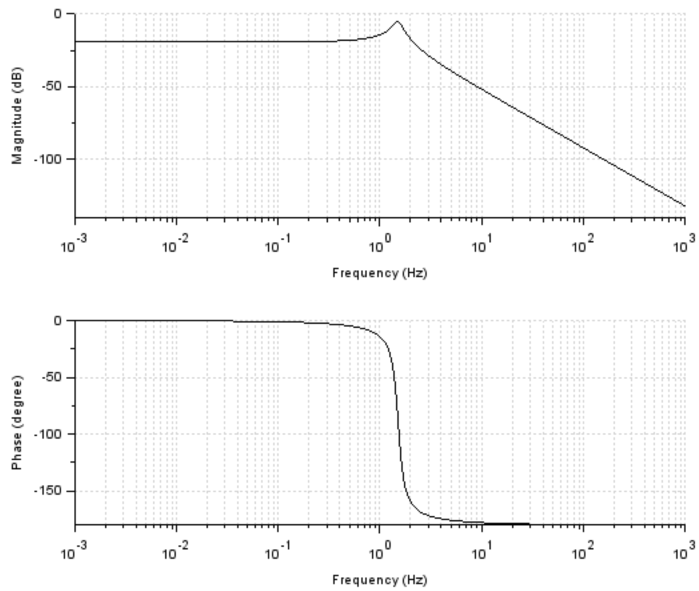


Figure 10: Bode Plot of the Given Transfer Function

Both the Gain and Phase Margins of the system are infinite, since the bode plot never has a magnitude of 1, neither a phase of 180$\circ$

(b)

The current steady state error is:-

$$sse = \frac{1}{1+G(0)} = \frac{1}{1+\frac{2000}{18001}} \approx= \frac{1}{1+\frac{1}{9}} = \frac{9}{10}$$
$$\implies sse' = \frac{9}{100} = \frac{1}{1+\frac{K}{9}} \implies K = 91$$

(c)

The new crossover frequencies and margins can be seen in the following plot:-

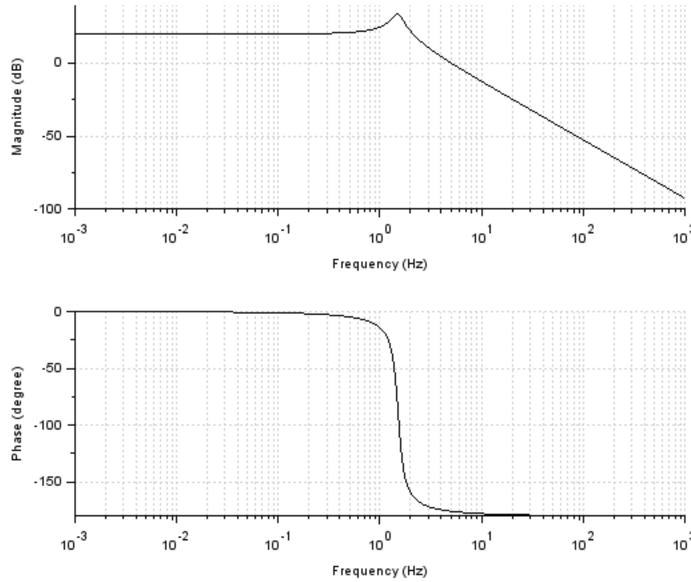


Figure 11: Bode Plot of the New Transfer Function

Since the Magnitude plot shift upwards, we observe a gain crossover frequency, implying a finite phase margin. However, since the phase plot remains the same, the gain margin is still infinity.

(d)

In order to increase the phase margin above 90°, we need to add a zero which shifts the final value to $-90°$, and it should be added well before $\omega = 10$, so that the phase never goes below $-90°$. Thus, adding a zero at -1, we get:-

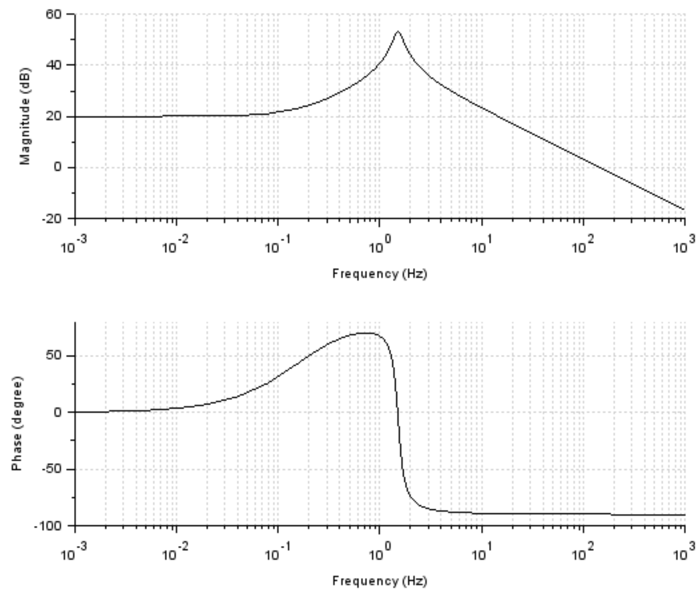


Figure 12: Bode Plot For Zero Added Transfer Function

(e)

Since the system infinite gain margin, its closed loop stable, since infinite gain margin implies the root locus never cuts the imaginary axis, and positive phase margin implies it lies completely in the Open Left Half Plane.

Code

```
s = poly(0,'s');
G = (10*s+2000)/(s^3+202*s^2+490*s+18001);

// a part
scf(0);
bode(syslin('c',G));

// c part
scf(1);
K = 91;
bode(syslin('c',G*K));

// d part
scf(2)
```

```
bode(syslin('c',G*K*(s+1)));
```