EE324 Control Systems Lab

Problem Sheet 9

Param Rathour | 190070049

Autumn Semester 2021-22

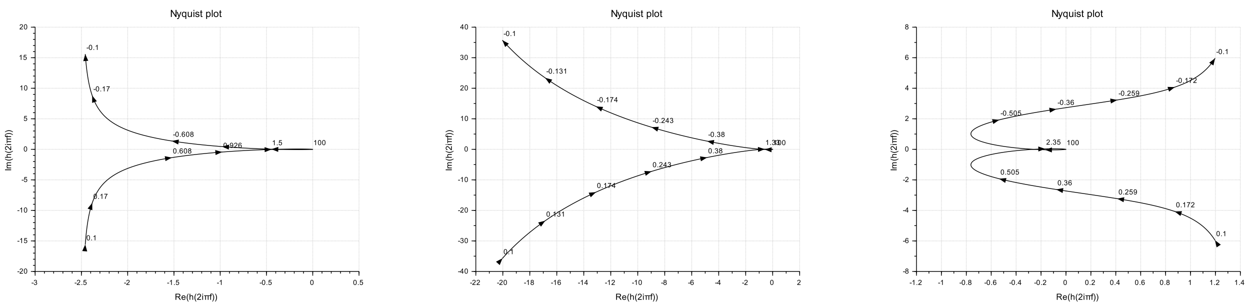
Contents

1 Q1

Open Loop Transfer Function is

$$G_1(s) = \frac{10}{\left(s \cdot \left(\frac{s}{5}+1\right) \cdot \left(\frac{s}{20}+1\right)\right)}$$

The corresponding Nyquist plot is shown in Figure 1a.



(a) $G_1(s)$ (b) $G_2(s)$ (c) $G_3(s)$

Figure 1: Nyquist Plots

Margins

System	Gain Margin (in dB)	Phase Margin (in °)
$G_1(s)$	7.9588002	22.535942
$G_2(s)$	2.0762546	4.0247332
$G_3(s)$	11.759539	43.173118

a)

$$\text{Lag Compensator} = \frac{(s+3)}{(s+1)} \rightarrow G_2(s) = C(s)\cdot G_1(s) = \frac{(s+3)}{(s+1)} \cdot \frac{10}{\left(s\cdot\left(\frac{s}{5}+1\right)\cdot\left(\frac{s}{20}+1\right)\right)}$$

The Nyquist plot is shown in Figure 1b. The Lag Compensator decreases both Gain Margin and Phase Margin.

b)

$$\text{Lead Compensator} = \frac{(s+1)}{(s+3)} \rightarrow G_3(s) = C(s)\cdot G_1(s) = \frac{(s+1)}{(s+3)} \cdot \frac{10}{\left(s\cdot\left(\frac{s}{5}+1\right)\cdot\left(\frac{s}{20}+1\right)\right)}$$

The Nyquist plot is shown in Figure 1c. The Lead Compensator increases both Gain Margin and Phase Margin.

Code to obtain Figure 1 is given below

```
s = %s;
fMin = 1e-1;
fMax = 1e2;
G1 = 10/(s*(s/5+1)*(s/20+1));
G1 = syslin('c', G1);
lag = (s+3)/(s+1);
lead = (s+1)/(s+3);
G2 = lag * G1;
G3 = lead * G1;
disp(g_margin(G1), p_margin(G1));
--> 7.9588002
--> 22.535942
nyquist(G1, (fMin,fMax));
xs2pdf(0, 'Q1a');
disp(g_margin(G2), p_margin(G2));
--> 2.0762546
--> 4.0247332
nyquist(G2, (fMin,fMax));
xs2pdf(0, 'Q1b');
disp(g_margin(G3), p_margin(G3));
--> 11.759539
--> 43.173118
nyquist(G3, (fMin,fMax));
xs2pdf(0, 'Q1c');
```

2 Q2

a)

Ideally, a Notch Filter annihilates a single frequency. Such Filter are not possible to realise in practice (with infinite slope). Instead, a simple Notch Filter (approximate) can be used. It's transfer function is given by

$$H(s) = \frac{s^2+\omega_z^2}{s^2+\frac{\omega_p}{Q}s+\omega_p^2}$$

Where, Q denotes Q-factor, ω_z is zero circular frequency (cutoff frequency) and ω_p is pole circular frequency. We take $\omega_z = \omega_p$ for a standard notch filter.

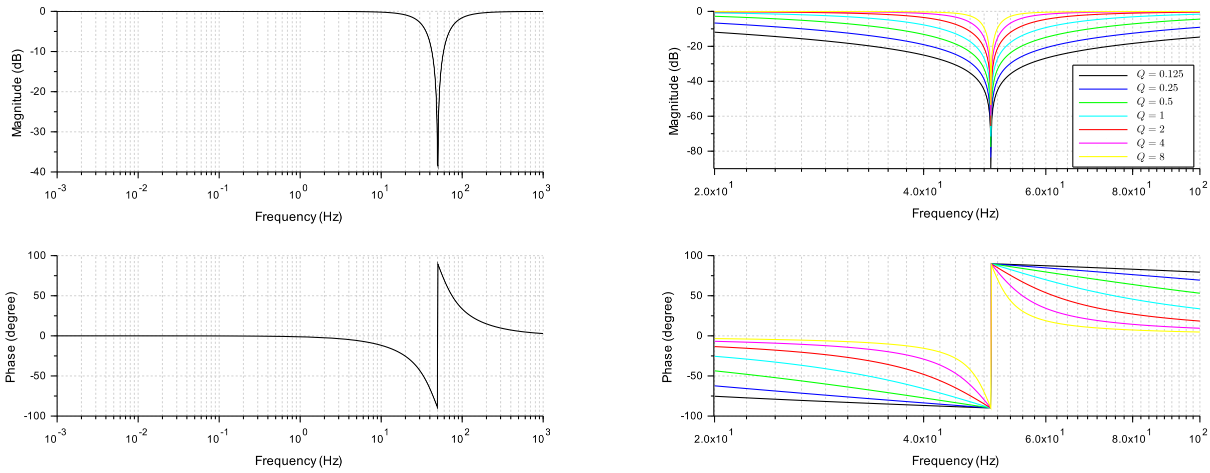
Now, the transfer function of Notch Filter that rejects 50Hz is given by ($\omega_z = \omega_p = 2\pi \cdot 50$ rad/s and let $Q = 1$)

$$H(s) = \frac{s^2+(100\pi)^2}{s^2+(100\pi)s+(100\pi)^2}$$

The frequency response of this filter is shown in Figure 2a

b)

Now, we can adjust the steepness of the magnitude plot for the notch filter by varying Q, that is Quality Factor. As can be seen from Figure 2b, the steepness increases (decreases) as Q increases (decreases).
An interesting thing to note is that as Q increases, the poles of $H(s)$ becomes closer (in horizontal direction).



(a) Frequency Response of $H(s)$ (b) Variation of Frequency Response $H(s)$ with Q

Figure 2: Bode Plots

Code to obtain Figure 2 is given below

```
s = %s;
omegaZ = 50 * 2 * %pi;
omegaP = omegaZ;
Q = 1;
H = (s^2+omegaZ^2)/(s^2+(omegaP/Q)*s+omegaP^2)
H = syslin('c', H);
-->      98696.044 +s^2
   -------------------------
   98696.044 +314.15927s +s^2
bode(H);
xs2pdf(0, 'Q2a');

y = [];
QValues = [1/8 1/4 1/2 1 2 4 8];
legendValues = [];
dims = 1;
for i = 1:length(QValues)
    Q = QValues(i);
    str = "$Q = " + string(Q) + "$" ;
    legendValues = cat(dims, legendValues, str);
    H = (s^2+omegaZ^2)/(s^2+(omegaP/Q)*s+omegaP^2)
    H = syslin('c', H);
    y = cat(dims, y, H);
end
fMin = 2e1;
fMax = 1e2;
bode(y, fMin, fMax);
legend(legendValues,opt=4);
xs2pdf(0, 'Q2b');
```

3 Q3

Open Loop Transfer Function is

$$C(s) = \frac{100}{(s+30)}$$

For this system, the gain margin turned out to be ∞ and the phase margin is $107.45760°$. Hence,

$$T = \text{Minimum delay} = \frac{\text{Phase Margin (in rad)}}{\text{Gain Crossover Frequency (in rad/s)}} = 0.0196605$$

Hence $C(s) \cdot e^{-Ts}$ will not be a stable system anymore. To calculate e^{-Ts} in Scilab, a transfer function $G(s) \approx e^{-Ts}$ is calculted using Padé approximation, which is given by

$$e^{-Ts} \approx \frac{\sum_{i=0}^{m} p_i (Ts)^i}{\sum_{i=0}^{n} q_i (Ts)^i}$$

When $m = n$, these coefficients are

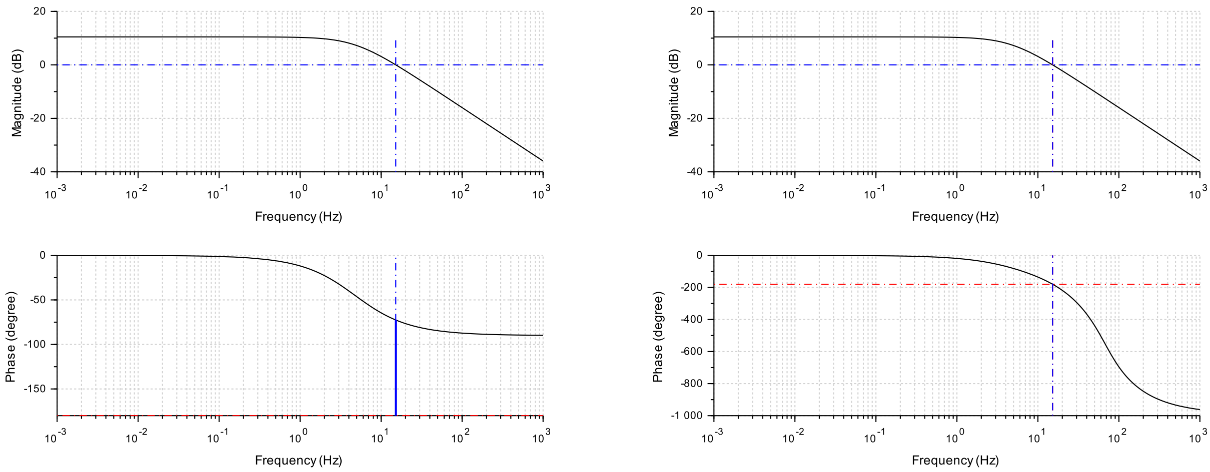
$$p_i = (-1)^i \frac{(2n-i)! \cdot n!}{(2n)! \cdot i! \cdot (n-i)!} \quad q_i = \frac{(2n-i)! \cdot n!}{(2n)! \cdot i! \cdot (n-i)!} \quad i = 0, 1, \ldots, n$$

I made a degree 6 approximation. So,

$$\text{TF} = C(s) \cdot G(s) = \frac{100}{(s+30)} \cdot \frac{\sum_{i=0}^{6} p_i (Ts)^i}{\sum_{i=0}^{6} q_i (Ts)^i}$$

Now, the new gain margin turned out to be -0.1004913dB and the phase margin is $3.3 \cdot 10^{-8°}$.
The gain margin reduced drastically (in fact, the system turned unstable), and the phase margin is almost zero.

Note that these results depend on the transfer function used for approximating e^{-Ts}.
Hence, some disrepencies in new values (ideally, new phase margin should have been 0 and gain margin unchanged).



(a) Frequency Response of $C(s)$ (b) Frequency Response of $C(s) \cdot G(s)$ (delayed)

Figure 3: Bode Plots

Code to obtain Figure 3 is given below

```
s = %s;
C = 100/(s+30);
C = syslin('c', C);
gainMargin = g_margin(C)
--> inf
[phaseMargin, gainCrossoverFrequency] = p_margin(C)
--> 107.45760, 15.182414
timeDelay = ((phaseMargin * %pi / 180) / (2 * %pi)) / gainCrossoverFrequency
--> 0.0196605

n = 6;
fact = [];
product = 1;
for i = 1:2*n+1
    fact($+1) = product;
    product = product * i;
end
function f = Factorial(n)
    f = fact(n+1);
endfunction
t = timeDelay;
numerator = 0;
denominator = 0;
for i = 0:n
    value =  Factorial(2*n-i) * Factorial(n) / (Factorial(2*n) * Factorial(i) * Factorial(n-i));
    numerator = numerator + (-t*s)^i * value;
    denominator = denominator + (t*s)^i * value;
end
G = numerator / denominator;
--> 1 -0.0098302s +0.0000439s^2 -0.0000001s^3 +1.886D-10s^4 -1.854D-13s^5 +8.681D-17s^6
    ------------------------------------------------------------------------------
    1 +0.0098302s +0.0000439s^2 +0.0000001s^3 +1.886D-10s^4 +1.854D-13s^5 +8.681D-17s^6
TF = C*G;
-->     100 -0.9830233s +0.0043924s^2 -0.0000115s^3 +1.886D-08s^4 -1.854D-11s^5 +8.681D-15s^6
   ---------------------------------------------------------------------------------------------
   30 +1.294907s +0.011148s^2 +0.0000474s^3 +0.0000001s^4 +1.942D-10s^5 +1.880D-13s^6 +8.681D-17s^7
gainMargin = g_margin(TF);
--> -0.1004913
[phaseMargin, gainCrossoverFrequency] = p_margin(TF);
--> 3.300D-08, 15.182414
timeDelay = ((phaseMargin * %pi / 180) / (2 * %pi)) / gainCrossoverFrequency;
--> 6.037D-12s

show_margins(C);
xs2pdf(0, 'Q3a');
show_margins(TF);
xs2pdf(0, 'Q3b');
```

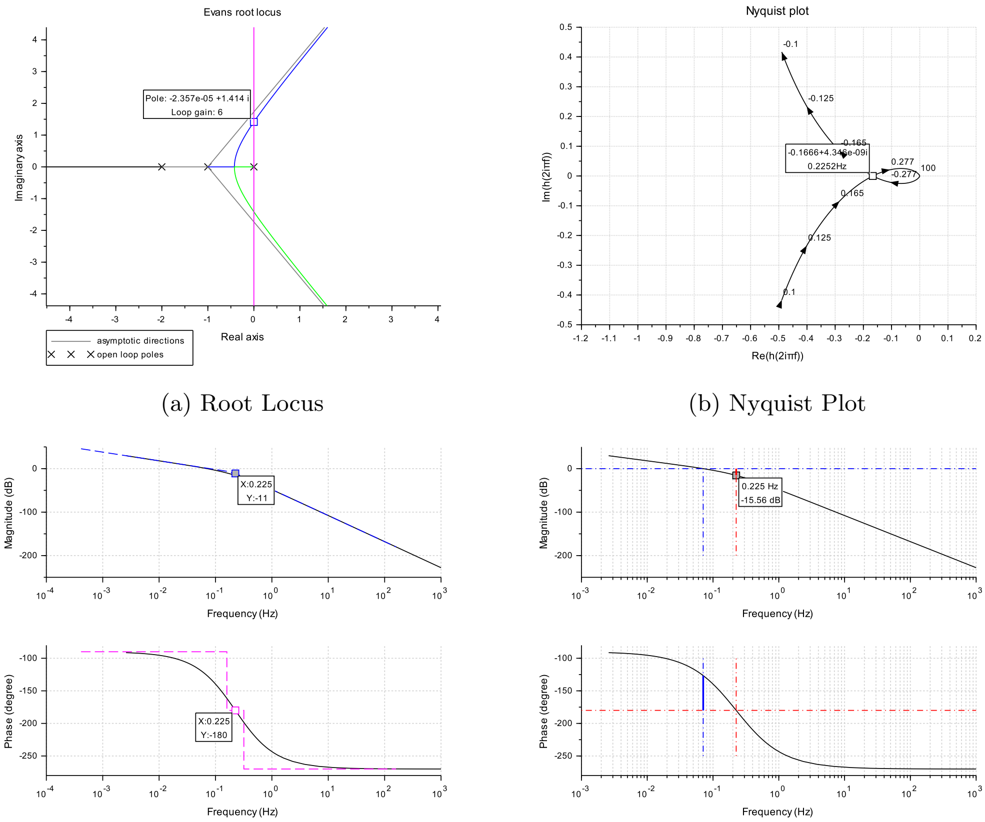
4 Q4

Open Loop Transfer Function is

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

The results are shown in below Table. The values are consistent except for Asymptotic Bode Plot.
Also, For Asymptotic Bode Plot, $f_{\text{gcf}} = \sqrt{f_L \cdot f_H}$, where $f_L = 0.159\text{Hz}$, $f_H = 0.318\text{Hz}$ gives $f_{\text{gcf}} \approx 0.225\text{Hz}$.

Method	Gain Margin (in dB)
Root Locus	15.563025 ($20 \log(6)$)
Nyquist Plot	15.566500 ($20 \log(1/0.1666)$)
Asymptotic Bode Plot	11
Actual Bode Plot	15.56



(a) Root Locus

(b) Nyquist Plot

(c) Asymptotic Bode Plot

(d) Actual Bode Plot

Figure 4: A comparison of different methods to evaluate gain margin

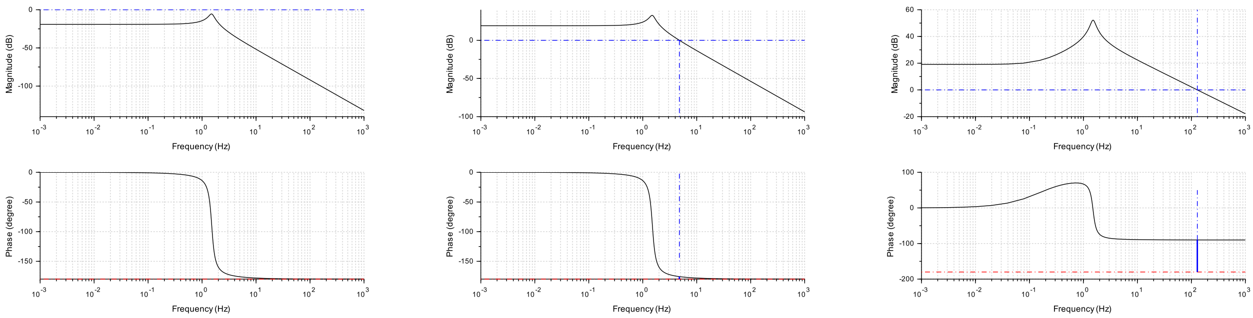
Code to obtain Figure 4 is given below

```
s = %s;
G = 1/(s^3+3*s^2+2*s);
G = syslin('c', G);
evans(G);
x = 5;
plot([0 0], [x -x], 'm');
xs2pdf(0, 'Q4a');
fMin = 1e-1;
fMax = 1e2;
nyquist(G, (fMin,fMax));
xs2pdf(0, 'Q4b');
bode(G);
bode_asymp(G);
fL = 0.159;
fH = 0.318;
fGCF = sqrt(fL * fH);
--> 0.2248600
xs2pdf(0, 'Q4c');
show_margins(G);
xs2pdf(0, 'Q4d');
```

5 Q5

Open Loop Transfer Function is

$$G_1(s) = \frac{10s + 2000}{s^3 + 202s^2 + 490s + 18001}$$



(a) $G_1(s)$ (b) $G_2(s)$ (c) $G_3(s)$

Figure 5: Bode Plots

a)

The Bode Plot is shown in Figure 5a.
The gain margin and the phase margin is ∞.

b)

$$\text{SSE} = 10\% = 0.1 = \lim_{s\to 0} \frac{sR(s)}{1 + K_p \cdot G(s)} \quad \rightarrow \quad K_p = \frac{\frac{1}{0.1} - 1}{\frac{2000}{18001}} = 81.0045 \quad \left(R(s) = \frac{1}{s}\right)$$

Hence,

$$G_2(s) = K \cdot \frac{10s + 2000}{s^3 + 202s^2 + 490s + 18001}$$

The Bode Plot is shown in Figure 5b.

c)

The gain margin is still ∞ (with phase crossover frequency undefined) and the phase margin is 4.2426012° (with gain crossover frequency 4.7688891Hz).

d)

To improve the phase margin of $G_2(s)$, we cascade the system with a zero such that the phase margin $\geq 90°$, without altering the dc gain of the closed-loop system.
Let the zero be at -1

$$G_3(s) = K \cdot \frac{10s + 2000}{s^3 + 202s^2 + 490s + 18001} \cdot (s + 1)$$

The Bode Plot is shown in Figure 5c.

e)

The gain margin is still ∞ (with phase crossover frequency undefined) and the phase margin is 90.070741° (with gain crossover frequency 128.94005Hz).
As gain margin, phase margin are > 0, the system is closed loop stable (as all closed loop poles are in RHP)

Code to obtain Figure 5 is given below

```
s = %s;
G1 = (10*s+2000)/(s^3+202*s^2+490*s+18001);
G1 = syslin('c', G1);
gainMargin = g_margin(G1);
--> Inf
phaseMargin = p_margin(G1);
--> []
show_margins(G1);
xs2pdf(0, 'Q5a');

K = (1/0.1 - 1) / (2000/18001);
--> 81.0045

G2 = K * G1;
--> 162009 +810.045s
    ----------------------
    18001 +490s +202s² +s³
[gainMargin, phaseCrossoverFrequency] = g_margin(G2);
--> Inf, []
[phaseMargin, gainCrossoverFrequency] = p_margin(G2);
--> 4.2426012, 4.7688891
show_margins(G2);
xs2pdf(0, 'Q5b');

G3 = G2 * (s+1);
--> 162009 +162819.05s +810.045s²
    -----------------------------
       18001 +490s +202s² +s³
[gainMargin, phaseCrossoverFrequency] = g_margin(G3);
--> Inf, []
[phaseMargin, gainCrossoverFrequency] = p_margin(G3);
--> 90.070741, 128.94005
show_margins(G3);
xs2pdf(0, 'Q5c');
```

6 References

Notch Filter - Wikipedia

Rational Approximation of Time Delay - Hanta