

Contents

1 Problem 1

1.0.1 A. Design a Butterworth LOW PASS filter

1. Specifications:

- DC Gain = 0 dB
- Amax = 1.0 dB
- Amin = 30 dB
- $p = 2000 \text{ rad/sec}$
- $s = 4700 \text{ rad/sec}$

Use Case 2 Sallen-Key circuit(s) (Figure 4.8) with $R = 10k$

2. My Work

Case 2 Sallen Key:

Formulas:

$$\frac{Z}{R C_c} = \frac{\omega_c}{Q} \quad \left. \vphantom{\frac{Z}{R C_c}} \right\} Q = \frac{\omega_c R C_c}{Z}$$

$$\frac{V_{out}}{V_{in}} = \frac{\omega_c^2}{s^2 + \left(\frac{2}{RC_c}\right)s + \omega_c^2}$$

$$\therefore \omega_c = \frac{1}{\sqrt{R^2 C_c}} \quad Q = \frac{RC_c}{2} \cdot \frac{\sqrt{C_c}}{C_c}$$

$$C_c/C_1 = 4Q^2 \quad RC_c = 1/\omega_c \quad C_c = 2Q C_1 \quad C_1 = C_2/2Q$$

$DC_{gain} = A = 0 \quad R = 10k\Omega \quad A_{min} = 30dB$
 $\omega_p = 2000 \text{ rad/s} \quad \omega_s = 4700 \text{ rad/s} \quad A_{max} = 1.0dB$

$\omega_0 = \frac{\omega_p}{(10^{\frac{A_{max}-A_{min}}{20}} - 1)^{1/2}} = \frac{\omega_s}{(10^{\frac{A_{max}-A_{min}}{20}} - 1)^{1/2}}$
 $Q = \frac{A}{10^{\frac{A_{max}-A_{min}}{20}} - 1}$

$\omega_0 = \frac{2000}{(10^{\frac{1-30}{20}} - 1)^{1/2}} = 2289.37$
 $Q = \frac{1}{10^{\frac{1-30}{20}} - 1} = 4.83$

Q is for 5th order = 0.5, 0.618, 0.707, 1.618
 * diff 4th probably 1/2 grounding error. They're pretty close, so all I care is the AVE.

$RC_c = 1/\omega_0$
 $10000(C_c) = 1/2289.37 = 43nF$

Stage 2: $C_{c1} = 4Q^2 = 1/C_{c2} = \frac{4 \times 10^{-9}}{2(0.618)} = 34.7 \times 10^{-9}$
 $C_{c2} = 4 \times 10^{-9} \times (0.618) = 2.47 \times 10^{-9}$

Stage 3: $C_{c1} = 4 \times 10^{-9} / 2(1.618) = 1.24 \times 10^{-9}$
 $C_{c2} = 4 \times 10^{-9} \times (1.618) = 6.47 \times 10^{-9}$

Stage 1: $C_{c1} = 4 \times 10^{-9} / 2(0.5) = 4 \times 10^{-9}$
 $C_{c2} = 4 \times 10^{-9} \times (0.5) = 2 \times 10^{-9}$

full circuit:

1.0.2 B. Matlab Plot

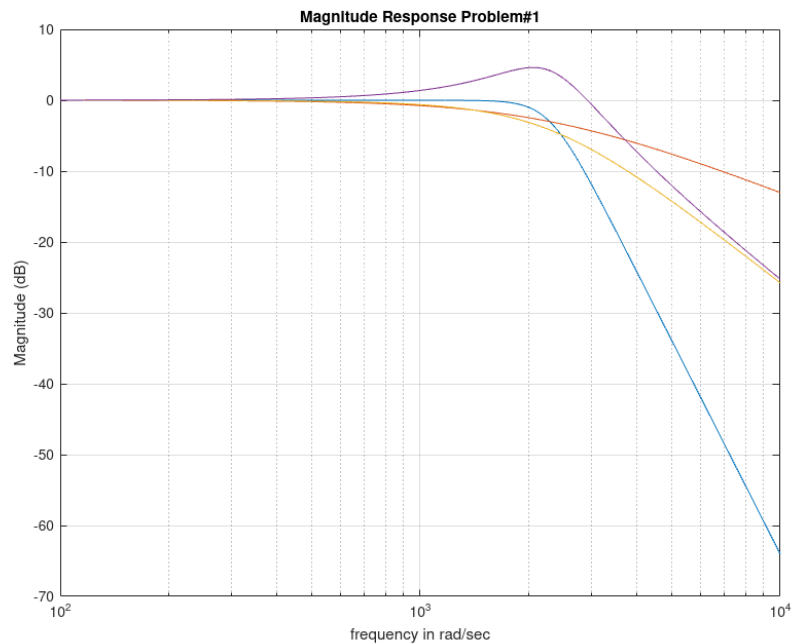
Plot the MAGNITUDE response for each section of your design and the overall MAGNITUDE response.

```
%Problem#1
w01=2289.37; Q1=0.5;
w02=2289.37; Q2=0.618;
w03=2289.37; Q3=1.618;
Num1=w01;
Num2=w02^2;
Num3=w03^2;
Den1=[1 w01];
```

```

Den2=[1 wo2/Q2 wo2^2];
Den3=[1 wo3/Q3 wo3^2];
w=logspace(2,4,1000);
H1=freqs(Num1,Den1,w);
H2=freqs(Num2,Den2,w);
H3=freqs(Num3,Den3,w);
H=H1.*H2.*H3;
H1db=20*log10(abs(H1));
H2db=20*log10(abs(H2));
H3db=20*log10(abs(H3));
Hdb=20*log10(abs(H));
semilogx(w,Hdb,w,H1db,w,H2db,w,H3db);
grid;xlabel('frequency in rad/sec');ylabel('Magnitude (dB)')
title('Magnitude Response Problem#1');

```



2 Problem 2

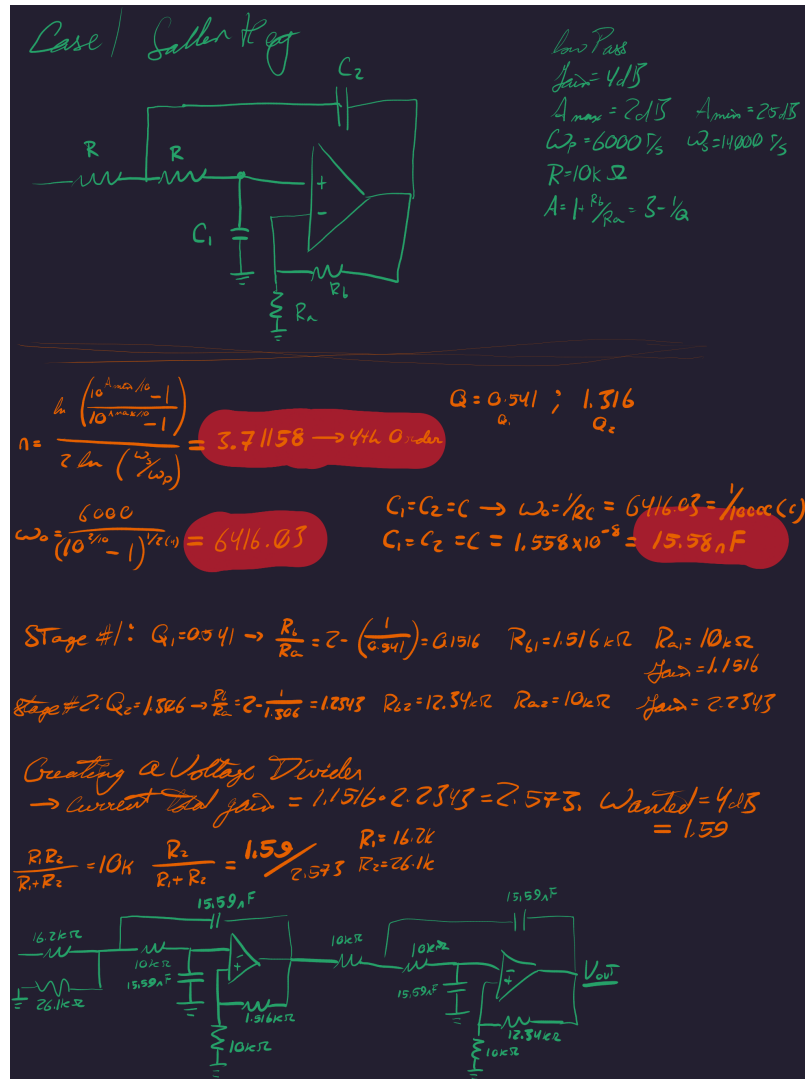
2.0.1 A. Design a Butterworth LOW PASS filter

1. Specifications:

- DC Gain = 4 dB
- Amax = 2 dB
- Amin = 25 dB
- p = 6000 rad/sec
- s = 14000 rad/sec

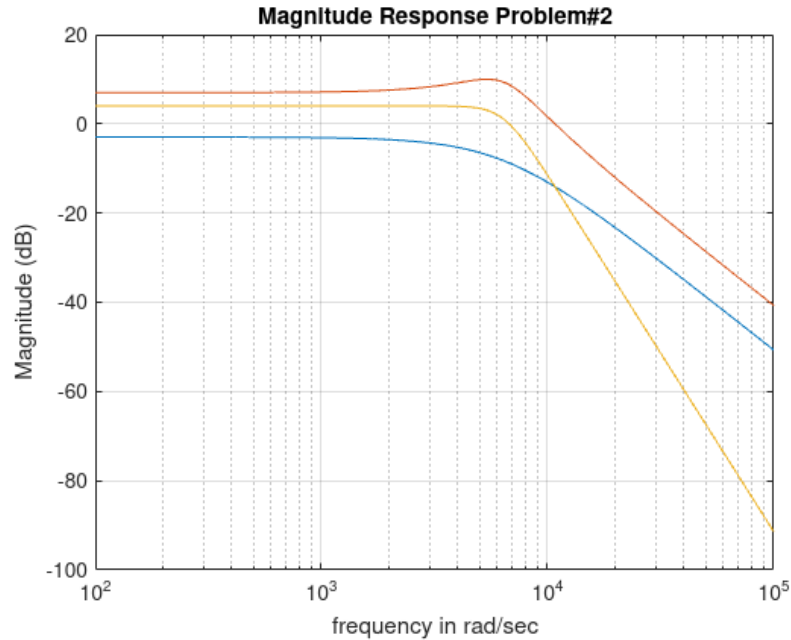
Use Case 1 Sallen-Key circuit(s) and a voltage divider with $R = 10k$
NOTE each stage will have a gain of $A = 1 + \frac{R_b}{R_a} = 3 - \frac{1}{Q}$, which means we need a voltage divider. (Probably only one).

2. My work



2.0.2 B. Matlab Plot

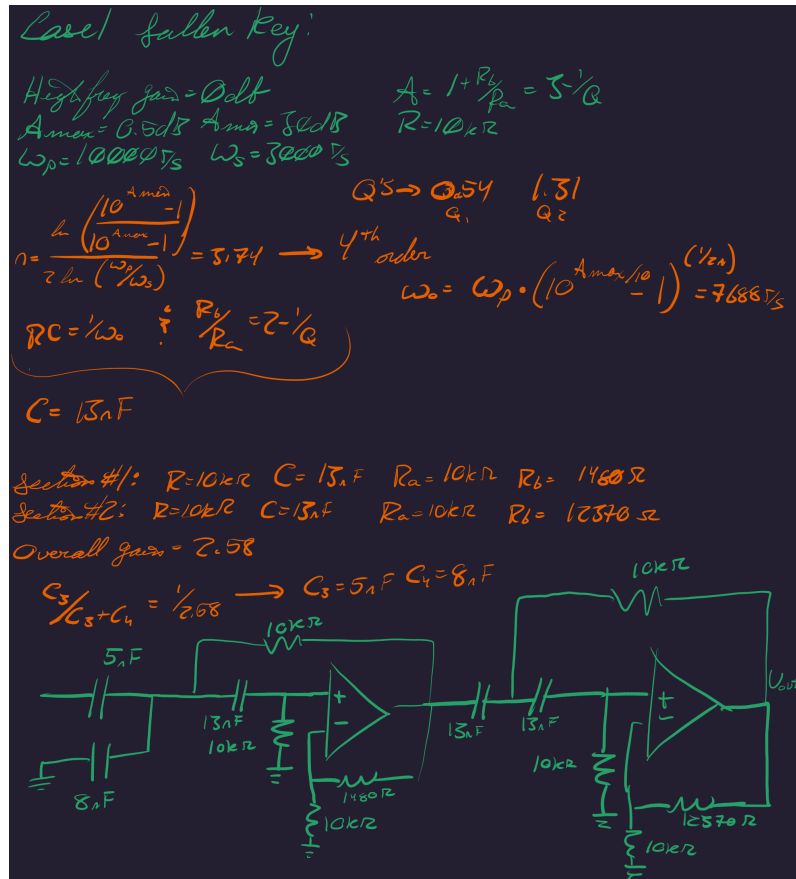
Plot the MAGNITUDE response for each section of your design and the overall MAGNITUDE response.



3 Problem 3

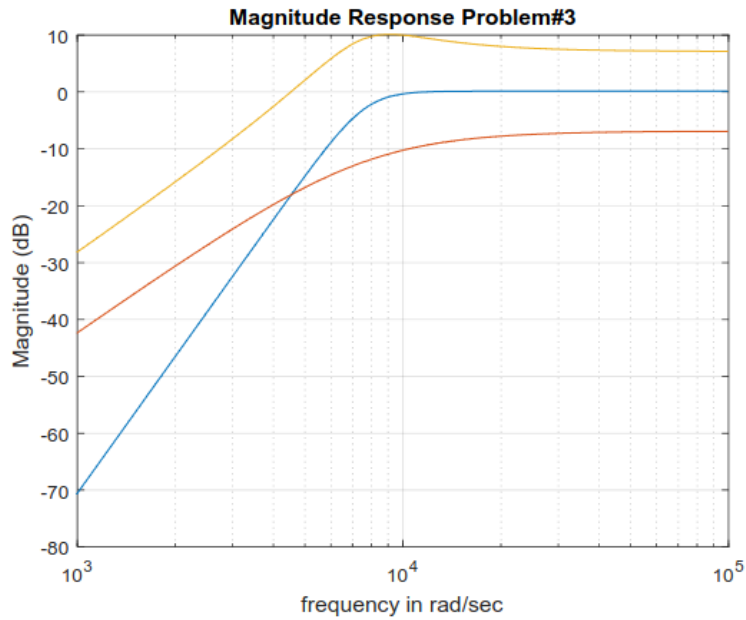
3.0.1 A. Design a Butterworth HIGH PASS filter

1. Specifications: Design for Problem 4.17 in your textbook Use Case 1 Sallen-Key circuit and design equations (4.53) and (4.54) You should require a 4th order filter to meet this specification Use ONE voltage divider (a capacitive voltage divider!) at the beginning of the 1st stage to set your overall DC Gain to 0dB. Use $R = 10k$ where possible in your design.
2. My work



3.0.2 B. Matlab Plot

Plot the MAGNITUDE response for each section of your design and the overall MAGNITUDE response.



4 Problem 4

Write a MATLAB function to calculate the pole locations for a Butterworth low pass filter, given filter specifications.

```
function[poles] = CJ_Butterpoles(n,w)
    k=0:(2*n-1);
    if(mod(n,2)==1)
        poles=w*exp(j*2*pi*k/(2*n));
    else
        poles=w*exp(j*(2*k+1)*pi/(2*n));
    end
    poles=poles(real(poles)<0);
    return
```

5 Problem 5

Write a MATLAB function to display W_o 's and Q 's associated with a set of poles.


```

function[] = CJ_QWgraph(poles)
% Function Name: Display_Wos_and_Qs
%% Parameters: Poles (get passed in)
% Returns: dummy (i.e. nothing)
% Description: Displays Wo's and Q's for any set of poles
    Q = (1./(2*cos(angle(poles))));
    numpoles = length(poles);
    k = 0:numpoles-1;
    wo = zeros(size(k));
    if (mod(numpoles,2)==1)
        for i = 1:length(k)
            wo(i) = poles(i)/exp(1j*k(i)*pi/numpoles)
        end
    else
        for i = 1:length(k)
            wo(i) = poles(i)/exp((1j*(2*k(i)+1)*pi)/(2*numpoles))
        end
    end
    polarplot(poles,'o');
    return;
end

```