3TR4: Communication Systems
Lab 1

Instructor: Dr. Chen

Tamer Rafidi – L09 – 400333527
Samer Rafidi – L09 – 400333524

Design and Simulation

Important Information given from the appendix

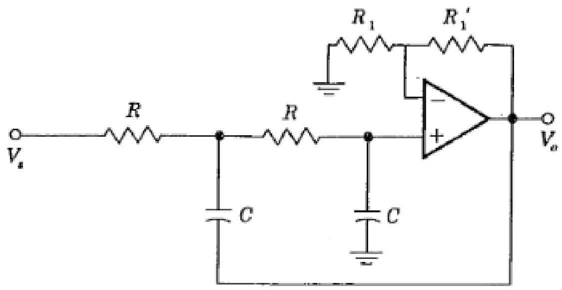


Figure 1. Second-order low-pass filter

TABLE 16-1 Normalized Butterworth polynominals

n	*Factors of polynomial $P_n(s)$*
1	$(s+1)$
2	$(s^2+1.414s+1)$

$$\frac{1}{(s/\omega_o)^2 + 2k(s/\omega_o) + 1}$$

Since we chose to create a 2nd order filter we use this equation = $\frac{1}{s^2+1.414s+1} = \frac{1}{s^2+2ks+1}$ From this. We can conclude that 2k = 1.414. We assume $w_o = 1\frac{rad}{s}$ which is why it is removed from the equation.

$$A_{Vo} = 3 - 2k$$

$$A_{vo} = 3 - 1.414 = 1.586$$

We can use the equation below to calculate the transfer function of the filter.

$$A_V(s) = \frac{V_o}{V_s} = \frac{A_{Vo}Z_3Z_4}{Z_3(Z_1+Z_2+Z_3)+Z_1Z_2+Z_1Z_4(1-A_{Vo})} = A_{Vo}\frac{(1/RC)^2}{s^2+\left(\frac{3-A_{Vo}}{RC}\right)s+\left(\frac{1}{RC}\right)^2}$$

The following equation is the transfer function of the filter. This is important to know because it can help us solve the filter frequency:

$$H(s) = 1.586 - \frac{\left(\frac{1}{RC}\right)^2}{s^2 + \frac{1.414}{RC} + \left(\frac{1}{RC}\right)^2}$$

To calculate the cut-off frequency using a resistor of 1KΩ and a capacitor of 0.1uF:

$$F_c = \frac{1}{2\pi RC} = \frac{1}{2\pi(1)(0.1)} \approx 1.59 \approx 1.6kHz$$

Both capacitators must be the same value and both must have a value of less than 1uF in order to work for a second-order low pass Butterworth filter.

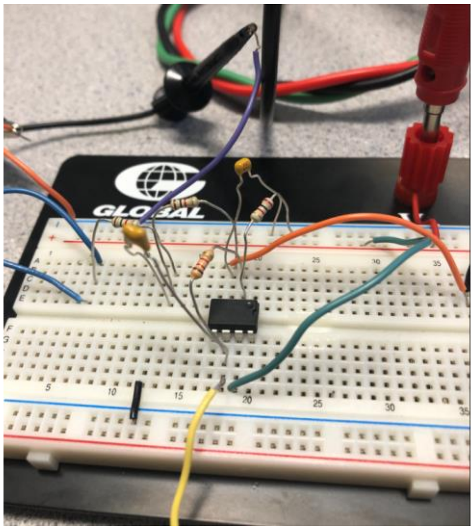


Figure 2. Physical Circuit used in-lab

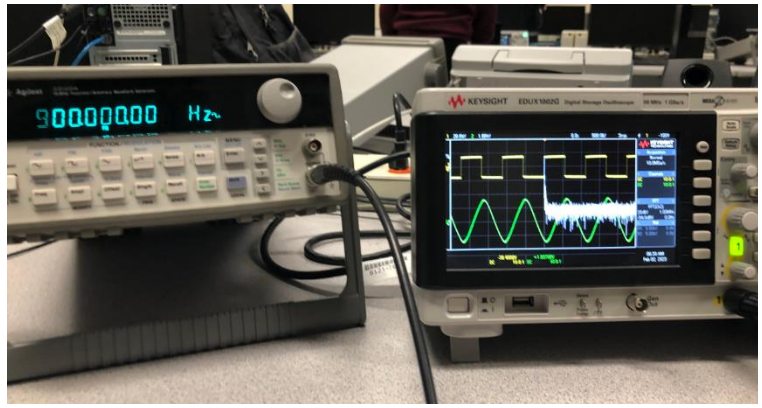
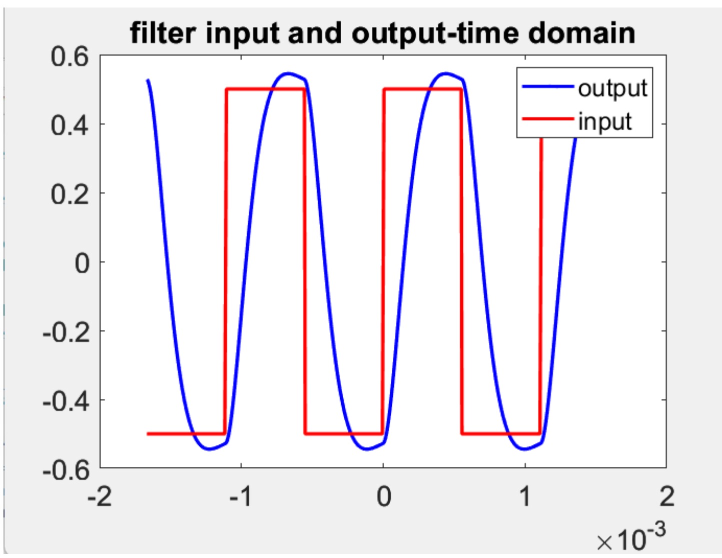
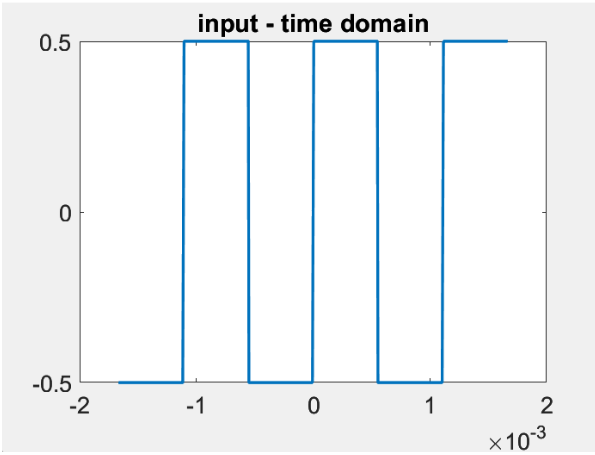


Figure 3. Oscilloscope from in-lab testing



We can compare the MATLAB plot with the experimental plot above. Both plots have similarities when looking at the shapes of both waves. We see a square wave and a sinusoidal wave in both the MATLAB and experimental plots.



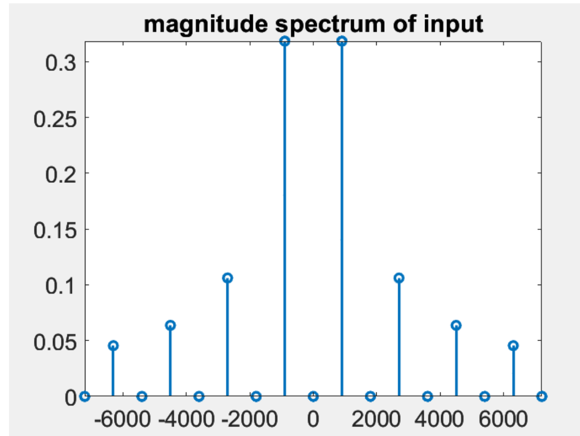
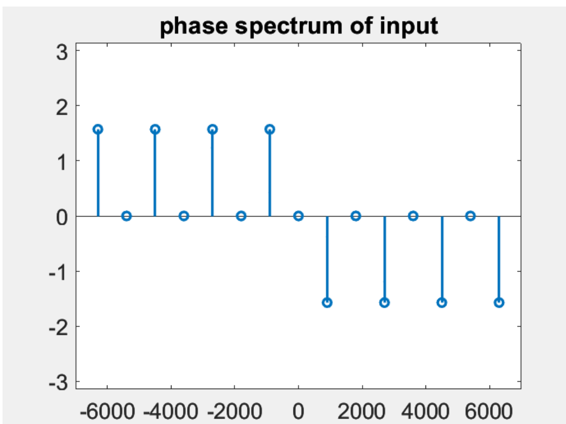


Figure 3.Matlab input with a frequency of 900Hz



Part 1: Butterworth Filter Experiments

We can analyze the details on the testing of the filter by applying a square wave input. Looking at the given notes, we can see that the Fourier series representation is given as

$$C_n = \frac{1}{T_0}\int_{-\frac{T_0}{2}}^{\frac{T_0}{2}} g(t)e^{-j2\pi n f_o t}dt$$

$$C_n = \frac{1}{T_0}\int_{-\frac{T_0}{2}}^{0}(-1)e^{-j2\pi n f_o t}dt + \int_{-0}^{\frac{T_0}{2}}(1)e^{-j2\pi n f_o t}dt$$

$$C_n = \frac{1}{Jn\pi}(1 - cosn\pi)$$

These values are only important when:

$$C_n = \begin{cases} \frac{2}{jn\pi} & n = odd \\ 0 & otherwise \end{cases}$$

Using the information above, we can find $\frac{C3}{C1}$

$$= \frac{\left(\frac{1}{j3\pi}\right)}{\frac{1}{jn}} = -10dB\ (before)$$

However, the information given in the appendix gives us a better representation of the changes between different frequency components.

Butterworth Filter[6] A common approximation of Eq. (16-17) uses the Butterworth polynomials $B_n(s)$, where

$$A_V(s) = \frac{A_{V_o}}{B_n(s)} \qquad (16\text{-}18)$$

and with $s = j\omega$,

$$|A_V(s)|^2 = |A_V(s)|\,|A_V(-s)| = \frac{A_{V_o}{}^2}{1 + (\omega/\omega_o)^{2n}} \qquad (16\text{-}19)$$

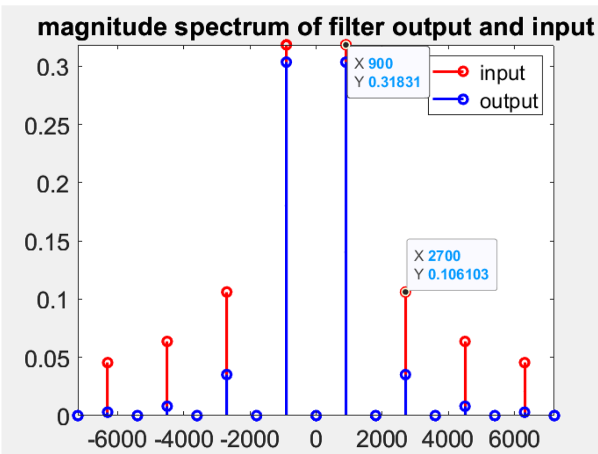
From Eqs. (16-18) and (16-19) we note that the magnitude of $B_n(\omega)$ is given by

$$|B_n(\omega)| = \sqrt{1 + \left(\frac{\omega}{\omega_o}\right)^{2n}} \qquad (16\text{-}20)$$

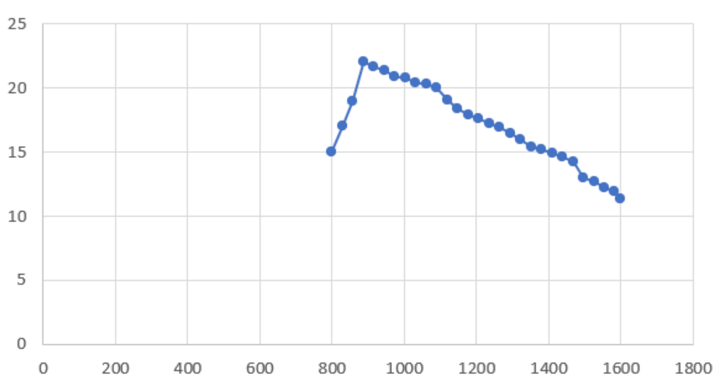
Using these equations above, we can try out the same ratio above. N = 2 as we are in the 2nd order. Also, we consider the input frequency at ½ of the cut off frequency as shown earlier.

$$\left(\frac{1}{3}\right)\left(\frac{\left(\frac{A_v^2}{\sqrt{1+\left(\frac{3}{2}\right)^4}}\right)}{\left(\frac{A_v^2}{\sqrt{1+\left(\frac{1}{2}\right)^4}}\right)}\right) = \left(\frac{1}{3}\right)\left(\frac{\sqrt{1+\left(\frac{1}{2}\right)^4}}{\sqrt{1+\left(\frac{3}{2}\right)^4}}\right) \approx 0.1395$$

$20\log(0.1395) = -17.1dB$ (after)



The graph above is the most important graph as we can see the input and output magnitudes. As we can see, the 3rd harmonic is 0.106103 and the 1st harmonic is 0.31831. The ratio between these two values are $\frac{C3}{C1} = \frac{0.106103}{0.31831} \approx 0.33$. When looking at the 3rd harmonic. We can do the same equation as earlier to compare those values. $20\log(0.106103) = -19.485dB$. The value above was -17.1dB



This graph shows the shape that resembles the low-pass filter. There are some weird data points, however, that is expected as there will always be some type of error in labs if they are not virtual. These data points were found by changing the frequency from 800Hz to our cut off (1.6kHz) and seeing the peak of the graph.

Part 2: Audio Signal Filtering

Low-Pass Filter:

The low pass filter sounds quieter than the original sound while also giving a muffled sound. It is still audible, but it is close to being hard to understand what is being said in this audio.

Band-Pass Filter:

The band pass filter sounds quieter than the original sound while also giving a muffled sound. It is harder to understand than the low-pass filter, but it is still audible. I realized that the sound also changes. At first it is more muffled, but it becomes less muffled and easier to understand near the end.

High-Pass Filter:

The high-pass filter sounds a bit louder than the band-pass and low-pass filter however it is much harder to understand. There is a very faint buzzing sound which make it hard to understand If I didn't the original sound. I would not understand what the words said other than the "it's easy to". The sentence "tell the depth of the whale" is very hard to understand.

Notch Filter:

The original sound had a very high tone on top of the audio. The notch filter does a very good job in removing that high tone pitch and keeping the words sounding very clear. However, I hear a weird bubbling effect like the sound is going into a cup.

Matlab

```
%% Square wave generator
clc
clear all
hold off

f0=900;     %fundamental freq of input triangular wave.
T0 = 1/f0;  %period
tstep = 0.005*T0;
no_sample = 3*T0/tstep + 1; %no. of samples  within  3*T0
no_sample1 = T0/tstep + 1; %no. of samples  within  T0
tt = -1.5*T0:tstep:1.5*T0;

tt1 = -0.5*T0:tstep:0.5*T0; % time vector for the period -0.5T0 to 0.5T0
gp1 = 0.5*sign(2*pi*f0*tt1);
gp_in = [gp1 gp1(2:no_sample1-1) gp1]; %3 cycles of the triangular wave
figure(1)
Hp1 = plot(tt,gp_in);
set(Hp1,'LineWidth',2)
Ha = gca;
set(Ha,'Fontsize',16)
title('input - time domain')
pause

%% Fourier series representation of signal (Amplitude Spectrum)
K=1/(2);
N=100; %no. of harmonics
nvec = -N:N;
c_in = zeros(size(nvec));
for n = nvec
    m = n+N+1;
    %% This equation was found during in lab
    c_in(m) = 0.5*(1-cos(n*pi))/(1i*n*pi);

    if (n == 0)
      c_in(m) = 0.0;
    end
end
```

```
f = nvec*f0; %frequency vector
figure(2)
Hp1=stem(f,abs(c_in));
axis([-8*f0 8*f0 0 max(abs(c_in))])
set(Hp1,'LineWidth',2)
Ha = gca;
set(Ha,'Fontsize',16)
title('magnitude spectrum of input')
pause

%% Fourier series representation of signal (Phase Spectrum)

figure(3)
Hp1=stem(f,angle(c_in));
set(Hp1,'LineWidth',2)
Ha = gca;
set(Ha,'Fontsize',16)
axis([-0.7e4 0.7e4 -pi pi])
title('phase spectrum of input')
pause

%% Designing the 2nd order Butterworth filter

%R=1e3;
%C=0.1e-6;
%fc=1/(2*pi*R*C)     %cutoff freq of filter
fc = 1600;

Hf = 1 ./(1+ 1.414*(1i*f/fc) + (1i*f/fc).^2) ;%filter transfer function
c_out = c_in .* Hf; %Fourier coefficients of the filter output

figure(4)
stem(f,abs(c_in),'r','LineWidth',2);
hold on
stem(f,abs(c_out),'b','LineWidth',2);
```

```
hold off
axis([-8*f0 8*f0 0 max(abs(c_in))])
Ha = gca;
set(Ha,'Fontsize',16)
title('magnitude spectrum of filter output and input')
Ha = gca;
set(Ha,'Fontsize',16)
legend('input','output')
pause

% hold off
% Hp1=plot(f,angle(c_out))
% set(Hp1,'LineWidth',2)
% Ha = gca;
% set(Ha,'Fontsize',16)
% title('phase spectrum of output')
% axis([-0.1e4 0.1e4 -pi pi])
% pause
% hold on
% Hp1=plot(f,angle(c_in),'r')
% set(Hp1,'LineWidth',2)
% Ha = gca;
% set(Ha,'Fontsize',16)
% pause
% hold off

%% Construct the output signal from the Cout Fourier coefficients

A = zeros(2*N+1,ceil(no_sample));
for n = nvec
    m=n+N+1;
    A(m,:) = c_out(m) .* exp(1i*2*pi*n*f0*tt);
end
gp_out = sum(A);
figure(5)
Hp1 = plot(tt,real(gp_out),'b',tt,gp_in,'r');
set(Hp1,'LineWidth',2)
Ha = gca;
set(Ha,'Fontsize',16)
title('filter input and output-time domain')
set(Ha,'Fontsize',16)
legend('output','input')
```