Faculty of Engineering, Architecture and Science

Department of Electrical and Computer Engineering

Course Number	**ELE532**
Course Title	**Signals and Systems I**
Semester/Year	Fall 2020

Instructor	**Javad Alirezaie**

ASSIGNMENT No. 4

Assignment Title	*Fourier Transform: Properties and Applications*

Submission Date	**November 29th, 2020**
Due Date	**November 29th, 2020**

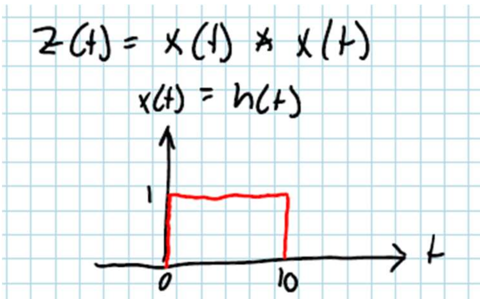
Student Name	**Nathan Agnew**
Student ID	500819685
Signature*	Nathan Agnew
Student Name	**Ahmad Fahmy**
Student ID	500913092
Signature*	**AF**

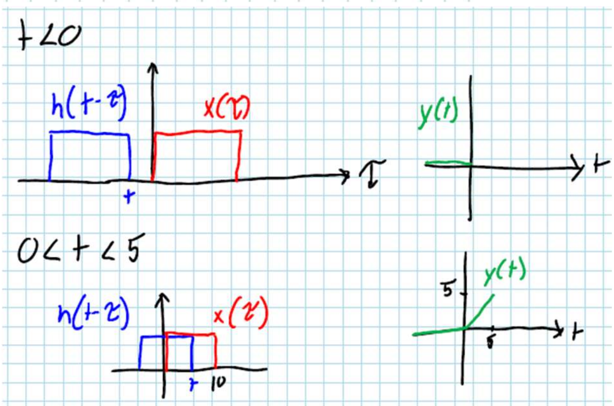
**By signing above you attest that you have contributed to this written lab report and confirm that all work you have contributed to this lab report is your own work. Any suspicion of copying or plagiarism in this work will result in an investigation of Academic Misconduct and may result in a "0" on the work, an "F" in the course, or possibly more severe penalties, as well as a Disciplinary Notice on your academic record under the Student Code of Academic Conduct, which can be found online at: www.ryerson.ca/senate/current/pol60.pdf.*

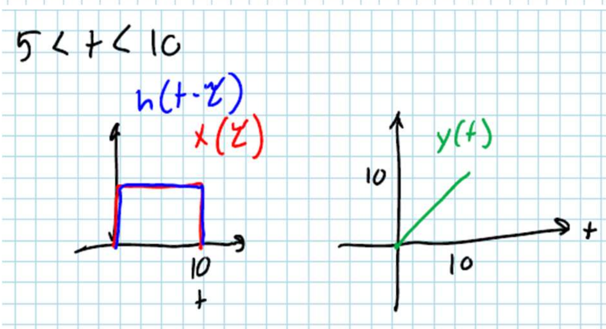
Table of Contents

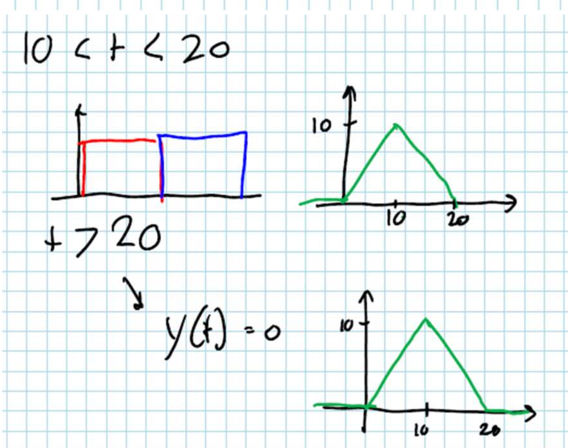
A. Fourier Transform and its Properties

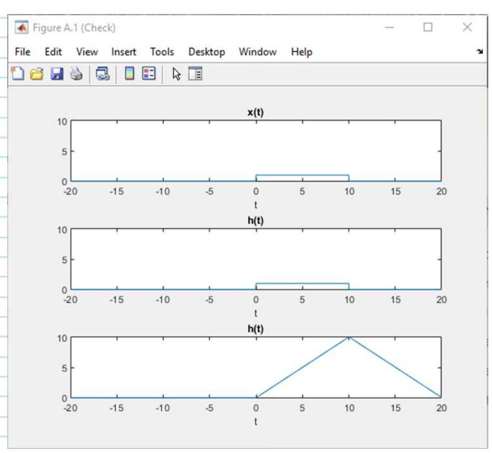
Problem A.1 [2 Marks] For the signal $x(t)$ shown in Figure (1), compute and plot $z(t) = x(t) * x(t)$.











Problem A.2 [2 Marks] Using MATLAB, calculate $Z(\omega) = X(\omega)X(\omega)$.

Editor - C:\Users\ahmad\OneDrive\Surface Desktop\ELE lab4\AssignmentA.m

osc.m | MagSpect.m | AssignmentA.m | AssignmentA4.m

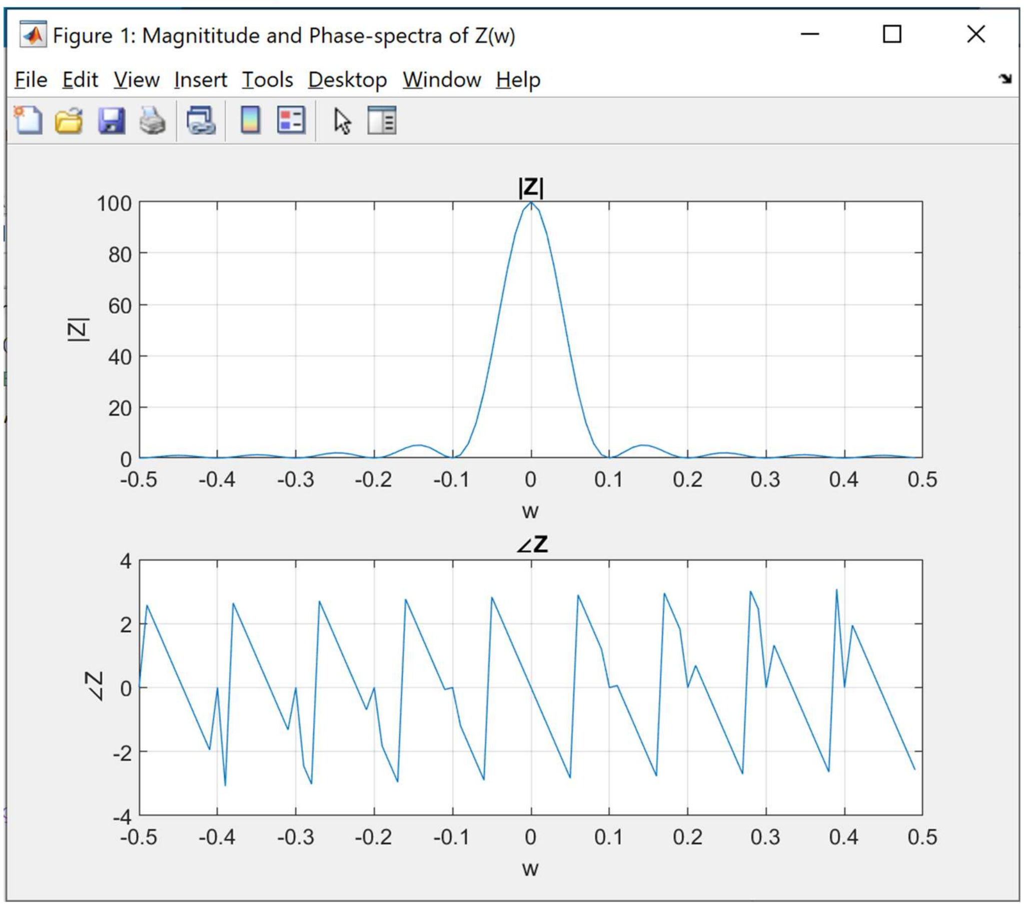
```
N = 100; PulseWidth = 10;
t = [0:1:(N-1)]; %#ok<NBRAK>
x = [ones(1,PulseWidth), zeros(1, N-PulseWidth)];
z = conv(x, x);

Xf = fft(x);

%A.2
Zf = Xf.*Xf;
```

Zf

1x100 complex double

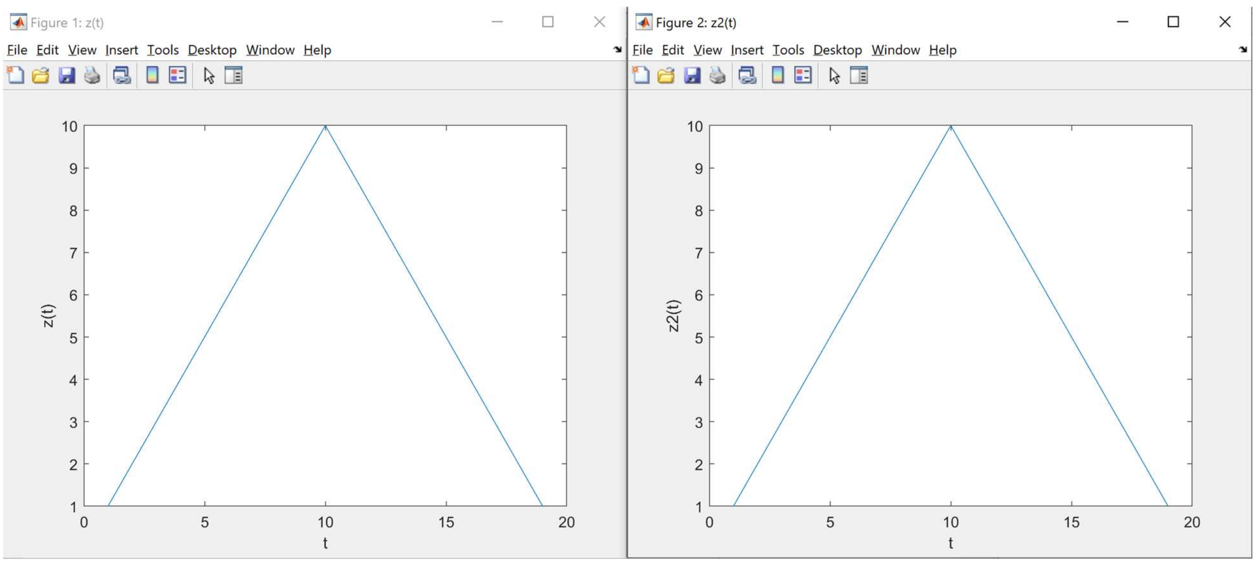
	1	2	3	4	5	6	7	8	9	10	11	12	
1	1.0000e+0...	81.7182 - 5...	37.3107 - 7...	-9.2624 - 7...	-36.7036 - ...	-38.8635 - ...	-24.9515 +...	-9.4153 + ...	-1.0468 + ...	0.4516 + 1....	0.0000 + 0....	0.8306 + 0....	2.23

Problem A.3 [3 Marks] Plot the magnitude- and phase-spectra of $z(t)$.



Problem A.4 [3 Marks] Compute $z(t)$ using **time-domain and frequency-domain operations** implemented in MATLAB. Plot both results and compare with the analytic result you determined in Problem A.1. Determine the appropriate time indices for proper labelling of the time-domain plots of $z(t)$. How do the results you generated in MATLAB using time- and frequency-domain operations compare with the analytic result you computed in Problem A.1? Explain which property of the Fourier transform you have demonstrated.

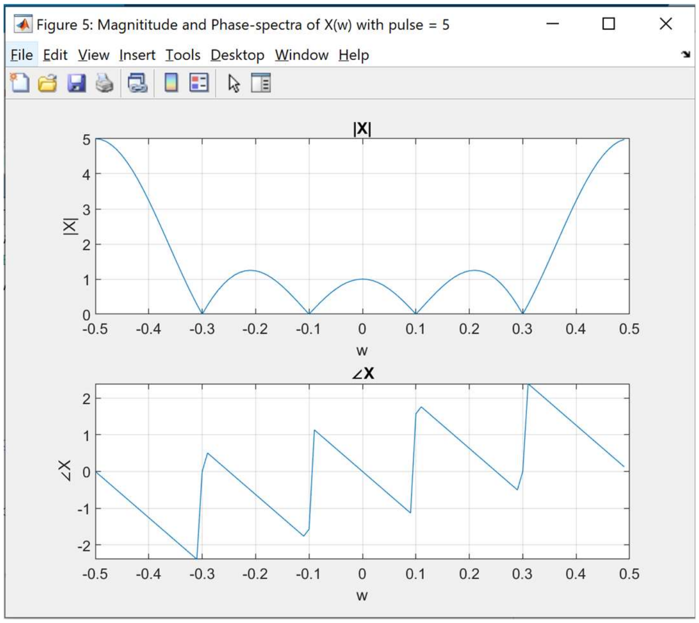
The results we generated in MATLAB were the same as the analytical result we calculated in A.1. The property of the Fourier Transform demonstrated is that convolution in the time-domain of two functions is equal to the product in the frequency-domain of the Fourier Transforms of those two functions.

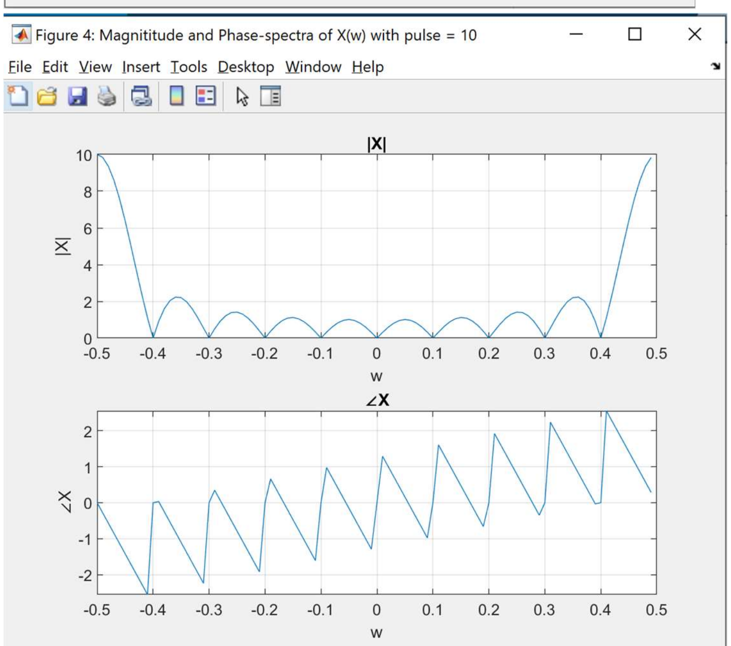


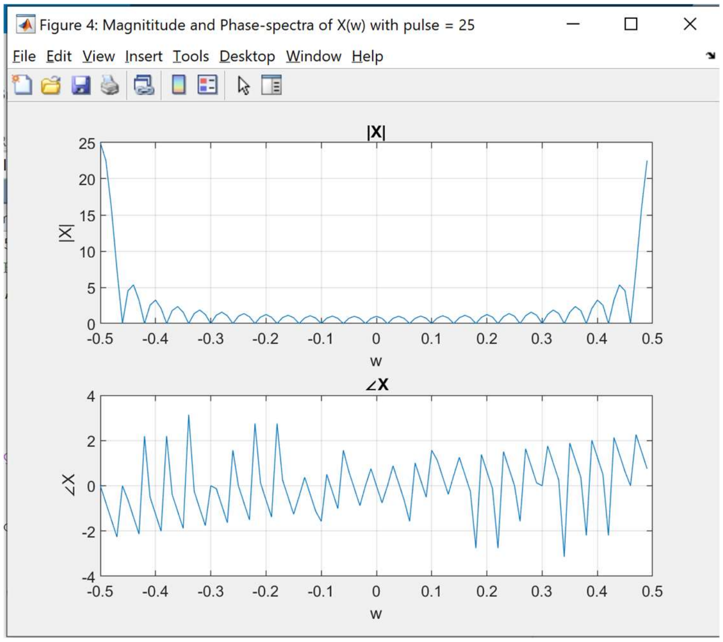

```
N = 10; PulseWidth = 10;
t = [0:1:(N-1)]; %#ok<NBRAK>
x = [ones(1,PulseWidth), zeros(1, N-PulseWidth)];

z = conv(x, x);


Xf = fft(x);
Zf = Xf.*Xf;
z2 = ifft(Zf);


figure('name', ['', 'z(t)']);
plot(z);
xlabel('t');ylabel('z(t)');

figure('name', ['', 'z2(t)']);
plot(z2);
xlabel('t');ylabel('z2(t)');
```

Problem A.5 [3 Marks] Change the width of the pulse $x(t)$ to 5 while keeping the total length at $N = 100$. Compute the Fourier transform of the narrower pulse and plot the corresponding magnitude- and phase-spectra. Repeat for a pulse width of 25. Explain the observed differences from the comparison of the frequency spectra generated by the three pulses with different pulse-widths. Explain which property of the Fourier transform you have demonstrated.

As we transition from pulse width 5 to 10 to 25, the magnitude and frequency spectra become more frequent and as a result tighter/closer together, the property demonstrated is scaling which demonstrates the behavior described.



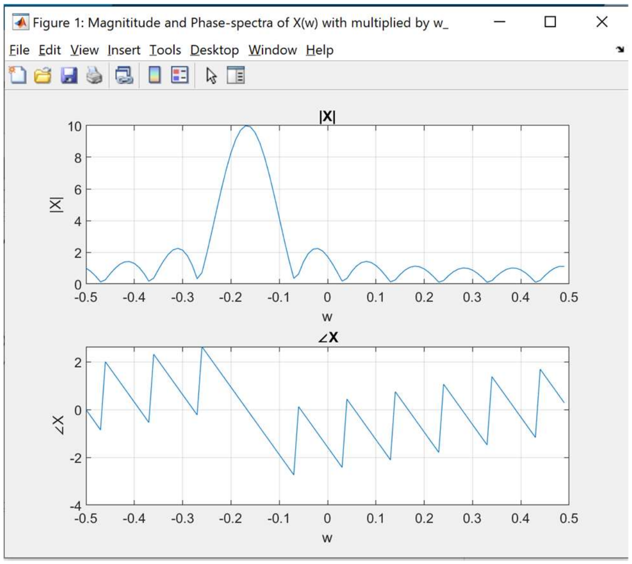





```
N = 100; PulseWidth = 5;
t = [0:1:(N-1)]; %#ok<NBRAK>
x = [ones(1,PulseWidth), zeros(1, N-PulseWidth)];

f = [-(N/2):1:(N/2)-1]*(1/N);

Xf = fft(x);

%A.5
figure('name', ['', 'Magnititude and Phase-spectra

subplot(2,1,1);
plot(f, (abs(Xf))); grid on;
%stem(abs(Xf));
title('|X|')
xlabel('w');ylabel('|X|');


subplot(2,1,2);
plot(f, (angle(Xf))); grid on;
%stem(angle(Xf));
title('∠X')
xlabel('w');ylabel('∠X');
```

Problem A.6 [3 Marks] Let $w_+(t) = x(t)e^{j(\pi/3)t}$ where $x(t)$ is the original pulse of pulse-width 10 shown in Figure (1). Using MATLAB compute and plot the magnitude- and phase-spectra of $w_+(t)$. Compare the frequency spectra result with those you generated in Problem A.3 and comment on the observed differences. Repeat for $w_-(t) = x(t)e^{-j(\pi/3)t}$ and $w_c(t) = x(t)\cos(\pi/3)t$. Explain which property of the Fourier transform you have demonstrated.

The property demonstrated is Frequency Shift, which shows that a shift in the frequency domain can be caused by multiplying the time-domain function by an exponential function, since cosines can be represented in Euler form the above statement holds true.



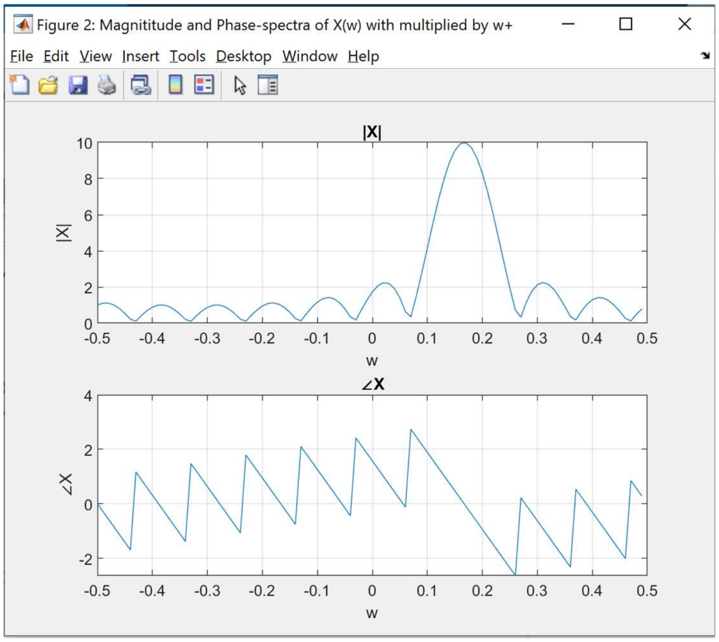


Figure 2: Magnititude and Phase-spectra of X(w) with multiplied by w+
File Edit View Insert Tools Desktop Window Help
|X|
∠X
w
-0.5 -0.4 -0.3 -0.2 -0.1 0 0.1 0.2 0.3 0.4 0.5

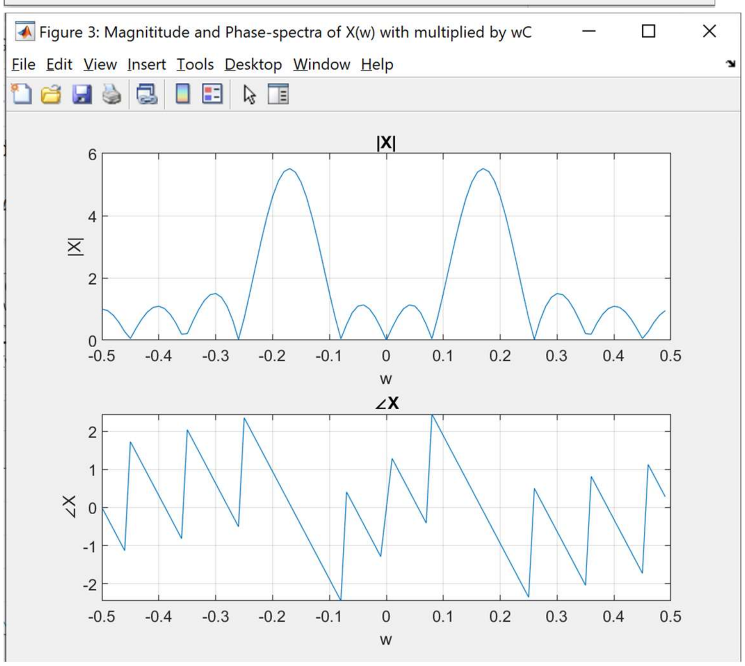


Figure 3: Magnititude and Phase-spectra of X(w) with multiplied by wC
File Edit View Insert Tools Desktop Window Help
|X|
∠X
w
-0.5 -0.4 -0.3 -0.2 -0.1 0 0.1 0.2 0.3 0.4 0.5

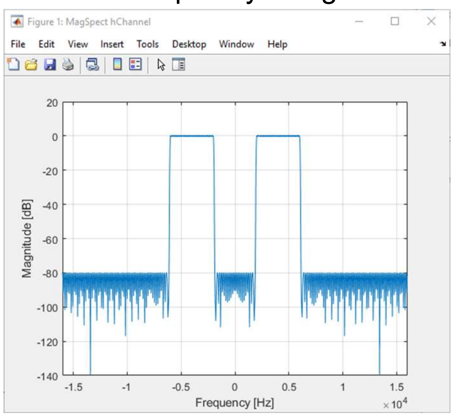
```
f = [-(N/2):1:(N/2)-1]*(1/N);


wplus = x.*exp(j*t*pi/3);
wminus = x.*exp(-j*t*pi/3);
wC = x.*cos(t*pi/3);

Xw = fft(wC);

figure('name', ['', 'Magnititude and Phase-s

subplot(2,1,1);
plot(f, fftshift(abs(Xw))); grid on;
title('|X|')
xlabel('w');ylabel('|X|');


subplot(2,1,2);
plot(f, fftshift(angle(Xw))); grid on;
%stem(angle(Xw));
title('∠X')
xlabel('w');ylabel('∠X');
```

B. Application of the Fourier Transform

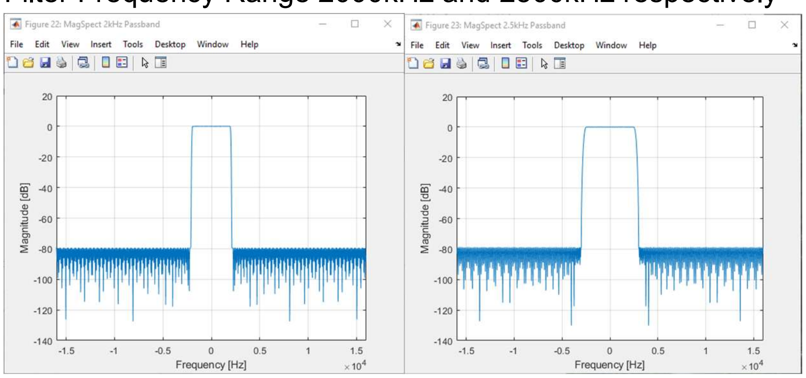
Problem B.1 [4 Marks]

Study the frequency-domain (magnitude only) characteristics of the signal **xspeech**, the lowpass filters defined by **hLPF2000** and **hLPF2000**, and the transmission channel defined by **hChannel**. Design a coding system that will allow the transmission of the signal **xspeech** over the channel. Also design the corresponding decoder to recover **xspeech** from the channel output. Implement both the coder and the decoder in MATLAB. Provide block diagrams of the coder and the decoder you designed and explain the rationale of your design.

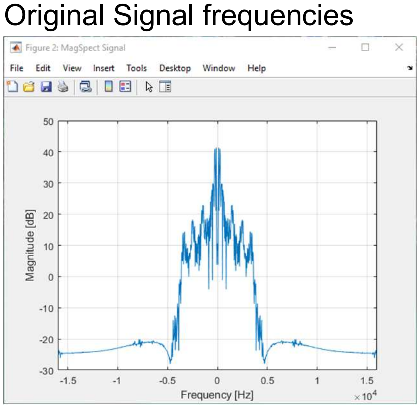
Channel Frequency Range



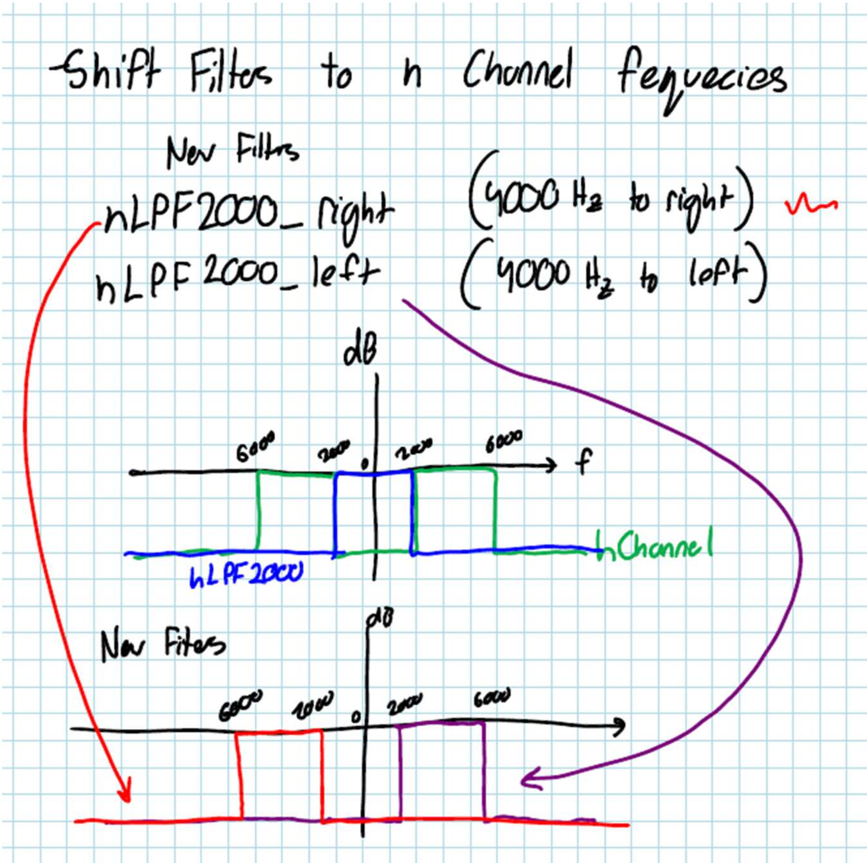
Filter Frequency Range 2000kHz and 2500kHz respectively



Original Signal frequencies



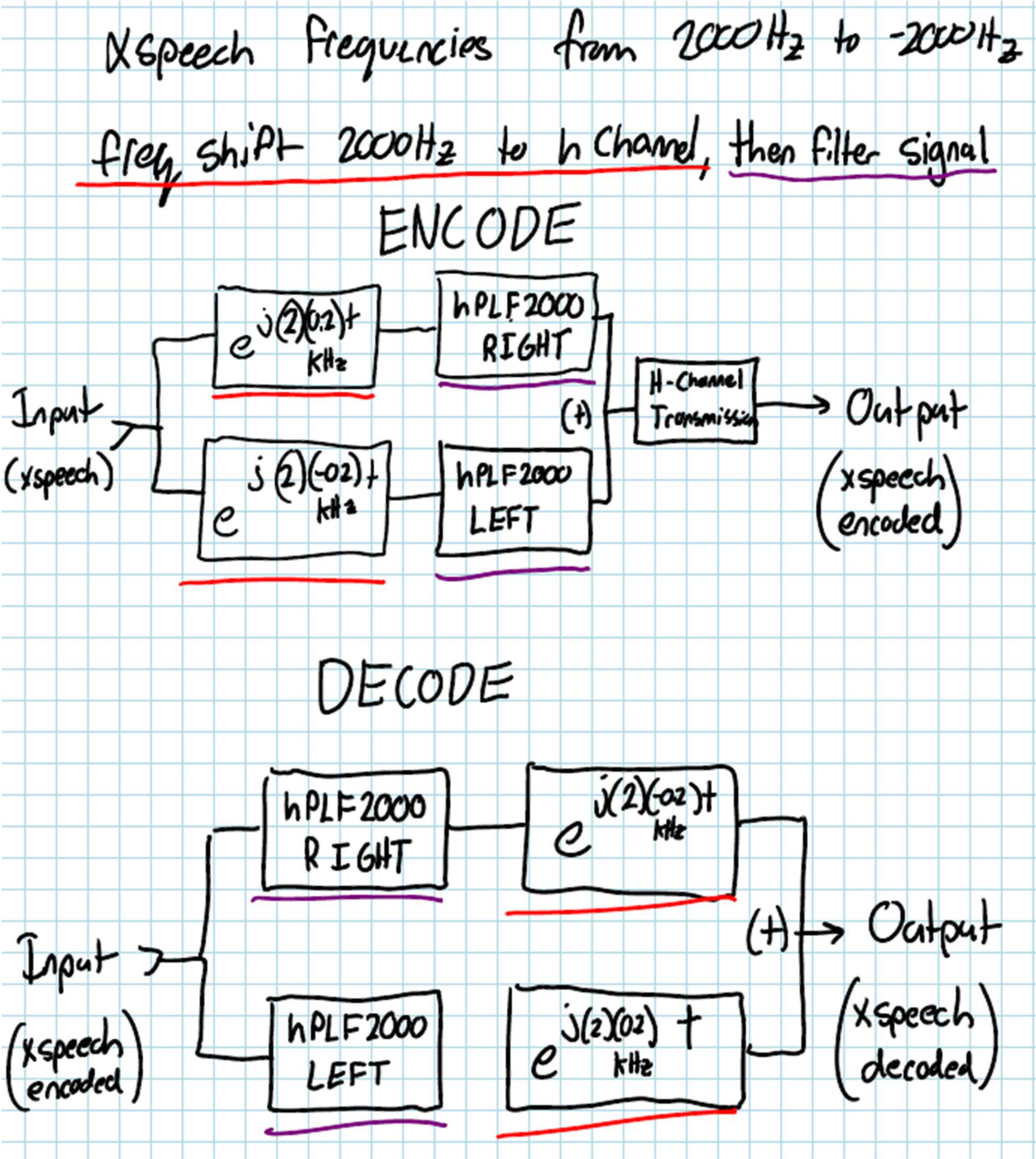
Frequency shifts to filters to align with h channel properties.



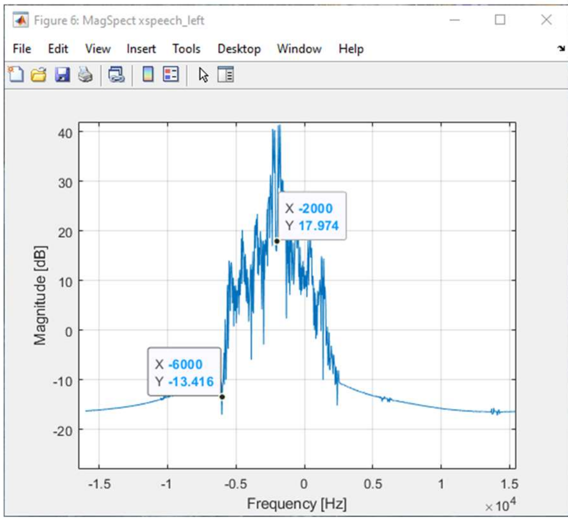
W0 = 2 * 0.4kHz = 0.8

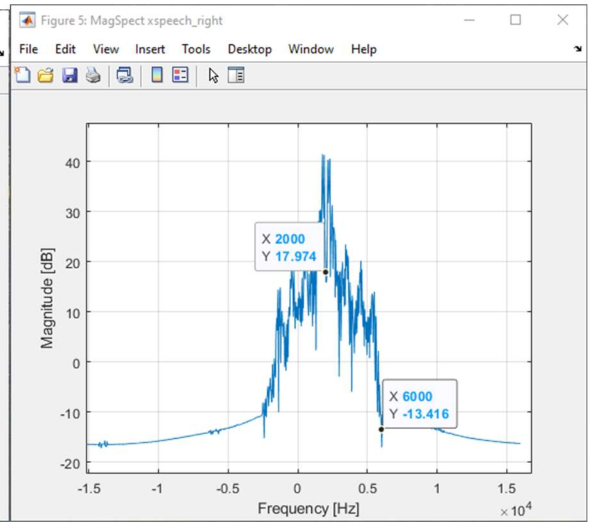
```
%Frequency Shift hLPF2000
w0=.8;
hLPF2000_right=zeros(1,711);
for t=1:711
    hLPF2000_right(1,t)=hLPF2000(1,t)*exp(1i*w0*t);
end
figure('name','MagSpect hLPF2000_right');
MagSpect(hLPF2000_right);

%Frequency Shift hLPF2000
w0=-.8;
hLPF2000_left=zeros(711,1);
for t=1:711
    hLPF2000_left(t,1)=hLPF2000(1,t)*exp(1i*w0*t);
end
 figure('name','MagSpect hLPF2000_left');
 MagSpect(hLPF2000_left);
```

Block diagram of encoder and decoder.



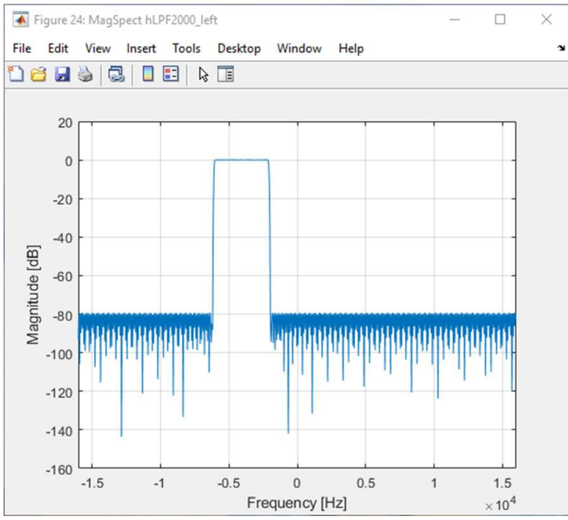
Encoding: Shifted original signal to half respective halves lining up with p channel frequencies.

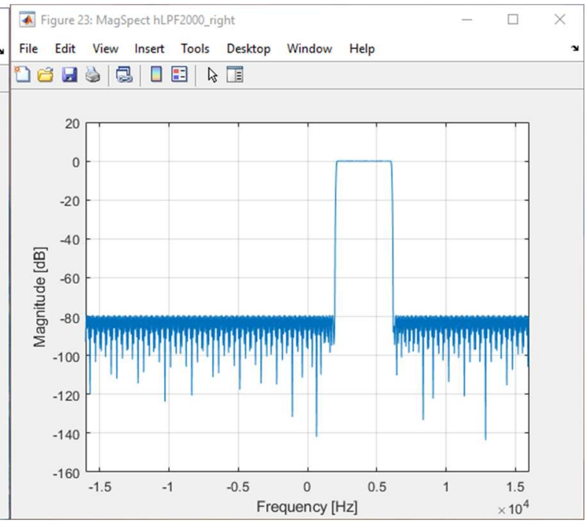





```
t=(1:80000);
%ENCODE: FREQUENCY SHIFT RIGHT XSPEECH
w0=.4;
xspeech_right=zeros(80000,1);
for t=1:80000
    xspeech_right(t,1)=xspeech(t,1)*exp(1i*w0*t);
end
figure('name','MagSpect xspeech_right');
MagSpect(xspeech_right);


%ENCODE: FREQUENCY SHIFT LEFT XSPEECH
w0=-.4;
xspeech_left=zeros(80000,1);
for t=1:80000
    xspeech_left(t,1)=xspeech(t,1)*exp(1i*w0*t);
end
figure('name','MagSpect xspeech_left');
MagSpect(xspeech_left);
```

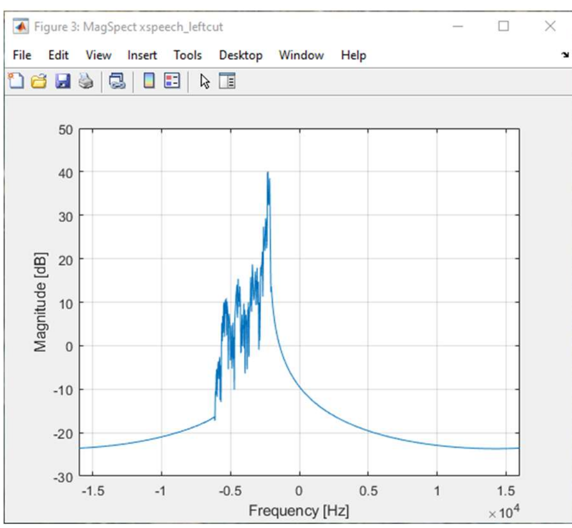
Encoding: The shifted filter to match p-channel. (see previous explanation)

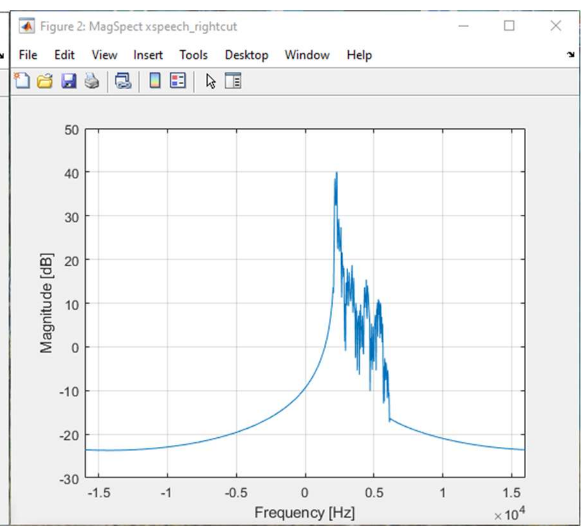




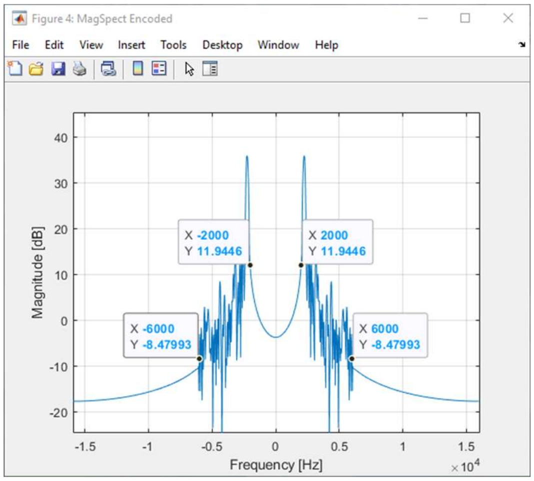
Encoding: The applied filter on both p channel sides.

```
%ENCODE: APPLY FILTERS TO REPECTIVE SIDE OF THE SIGNAL
xspeech_rightcut=conv(xspeech_right,hLPF2000_right);
xspeech_leftcut=conv(xspeech_left,hLPF2000_left);
figure('name','MagSpect xspeech_rightcut');
MagSpect(xspeech_rightcut);
figure('name','MagSpect xspeech_leftcut');
MagSpect(xspeech_leftcut);
```





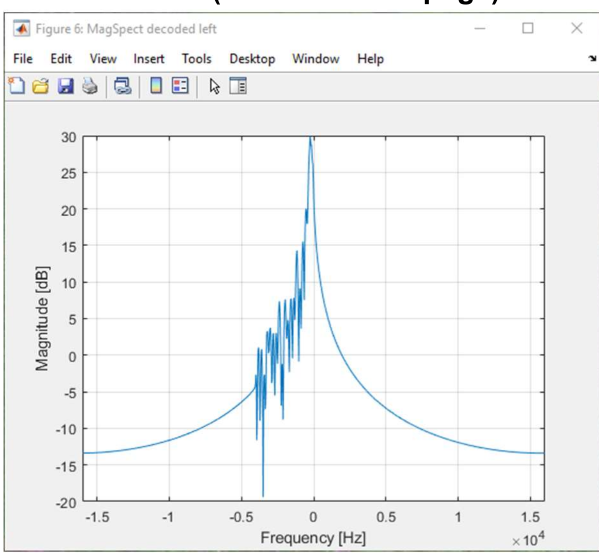
Encoding: Add the two half signals to combine respective encoded halfs

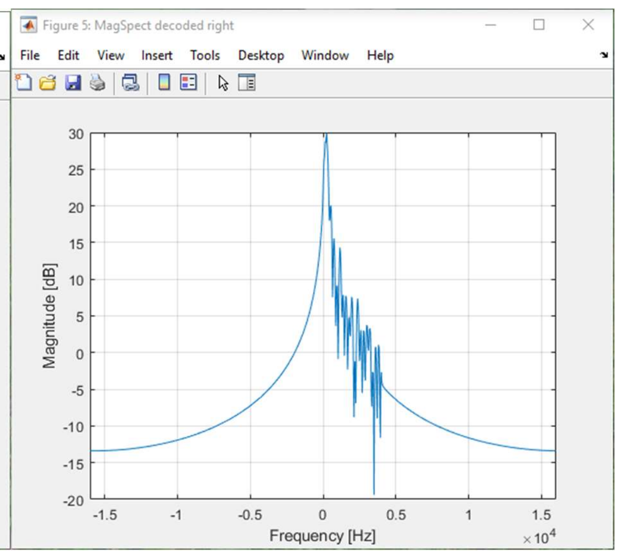



```
%ENCODE: ADD THE TWO HALF SINGALS TO CREATE ENCODED SINGAL
xspeech_new=xspeech_leftcut+xspeech_rightcut;
% figure('name','MagSpect new');
% MagSpect(xspeech_new);
```

```
%TRNASMISSION: CONVOLUDE THE SIGNAL WITH PCHANNEL TO TRANSMIT
xspeech_encoded=conv(xspeech_new,hChannel);
figure('name','MagSpect Encoded');
MagSpect(xspeech_encoded);
```

Decoding: Apply respective filters to isolate encoded sections (convolution), and shift back to center (see code next page)





```
%DECODE: ISOLATE RESPECTIVE HALFS TO DECODE
xspeech_rightcutdec=conv(xspeech_encoded,hLPF2000_right);
xspeech_leftcutdec=conv(xspeech_encoded,hLPF2000_left);

%DECODE: FREQUENCY SHIFT RIGHT XSPEECH
w0=-.4;
xspeech_rightdec=zeros(80000,1);
for t=1:80000
    xspeech_rightdec(t,1)=xspeech_rightcutdec(t,1)*exp(1i*w0*t);
end
figure('name','MagSpect decoded right');
MagSpect(xspeech_rightdec);


%DECODE: FREQUENCY SHIFT LEFT XSPEECH
w0=.4;
xspeech_leftdec=zeros(80000,1);
for t=1:80000
    xspeech_leftdec(t,1)=xspeech_leftcutdec(t,1)*exp(1i*w0*t);
end
figure('name','MagSpect decoded left');
MagSpect(xspeech_leftdec);
```

Decoding: Add the two respective signals to reconstruct the original xspeech

```
%DECODE: ADD HALF SINGALS TO RECREATE ORIGNAL SIGNAL
xspeech_reconstruct=xspeech_leftdec+xspeech_rightdec;
figure('name','MagSpect reconstruct');
MagSpect(xspeech_reconstruct);
```