Ryerson University

Department of Electrical, Computer, & Biomedical Engineering
Faculty of Engineering & Architectural Science

Course Title:	ELE
Course Number:	532
Semester/Year (e.g.F2016)	F 2021

Instructor:	Javad Alirezaie

Assignment/Lab Number:	4
Assignment/Lab Title:	The Fourier Transform: Properties and Applications

Submission Date :	Dec 5th /2021
Due Date:	Dec 5th /2021

Student LAST Name	**Student FIRST Name**	**Student Number**	**Section**	**Signature***
Youssef	Kirolos	500968175	01	K. Y
Miceli	Julian	500984182	01	J. M

Table of Contents

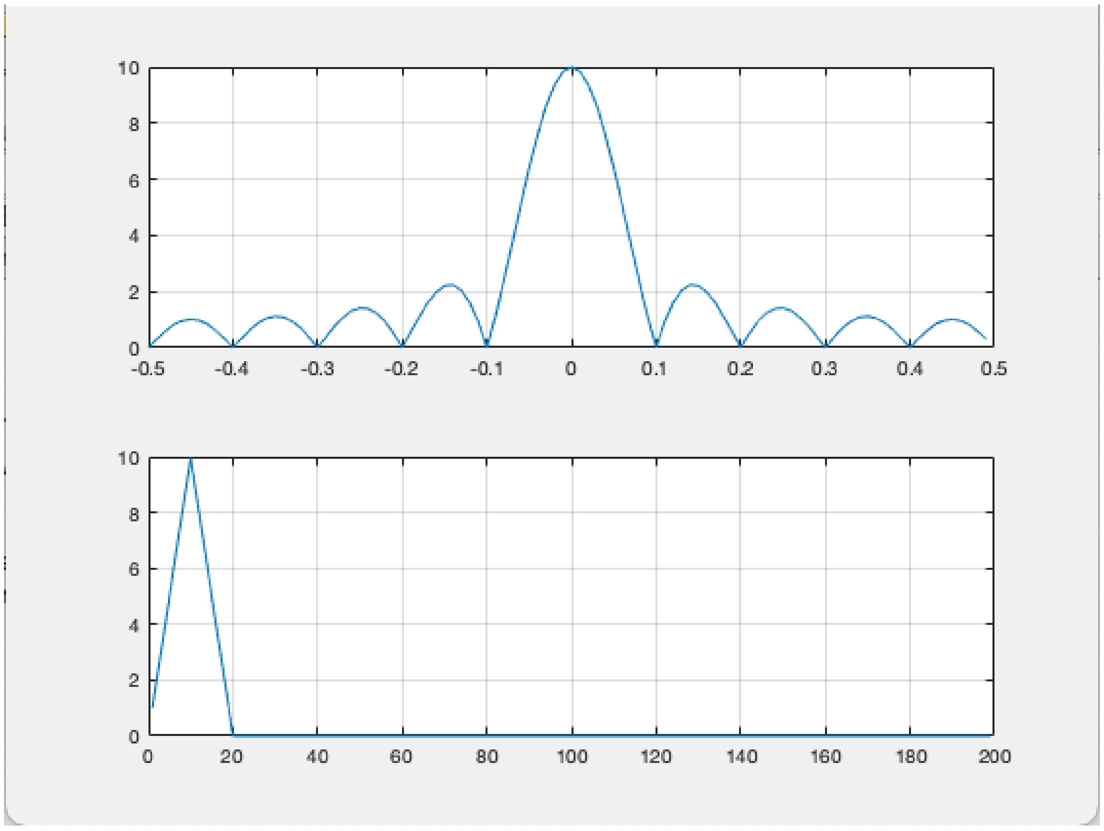
Objective:

The objective of this lab is to analyze the role of fourier transform on different systems and its applications in signal processing.

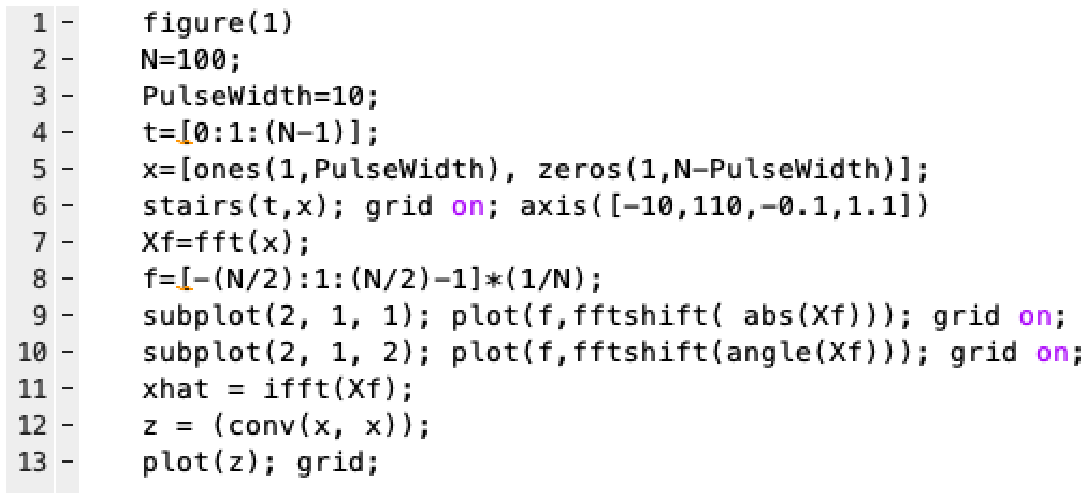
Experiment:

A1:

Figure A1:



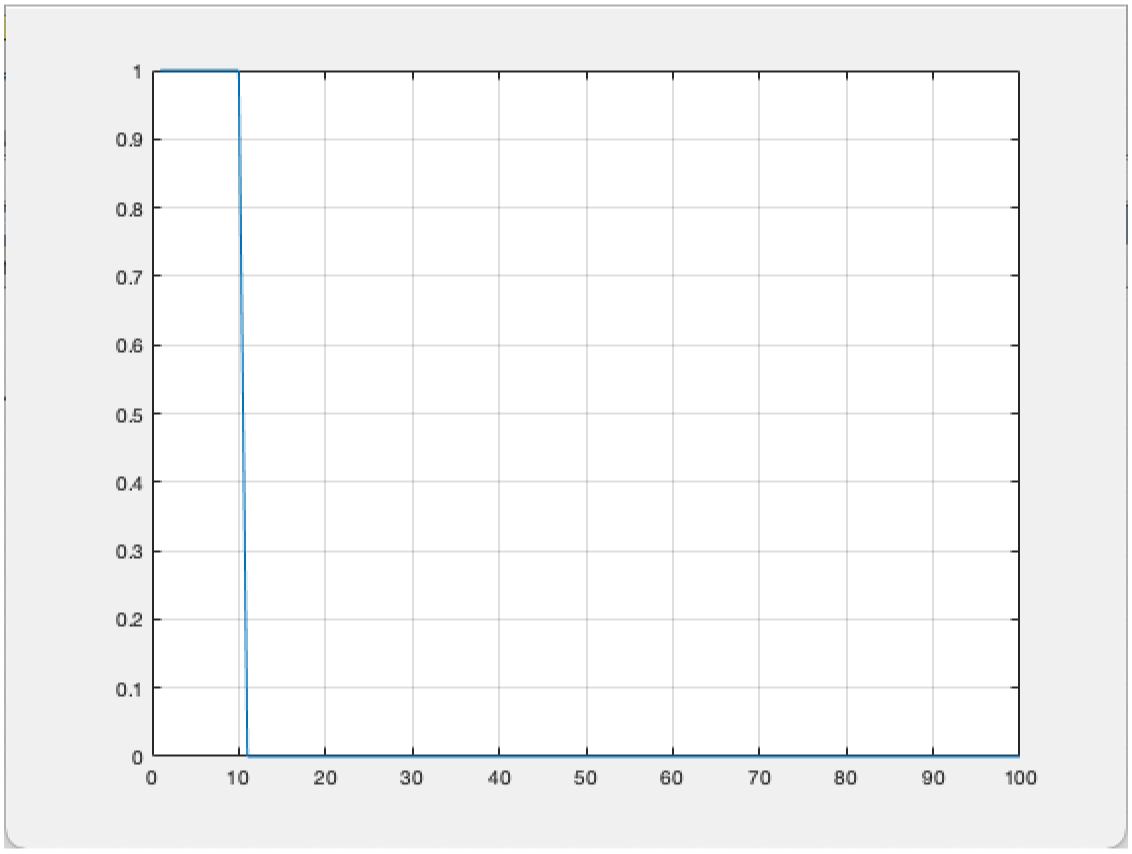
Matlab code used to plot Figure A1:



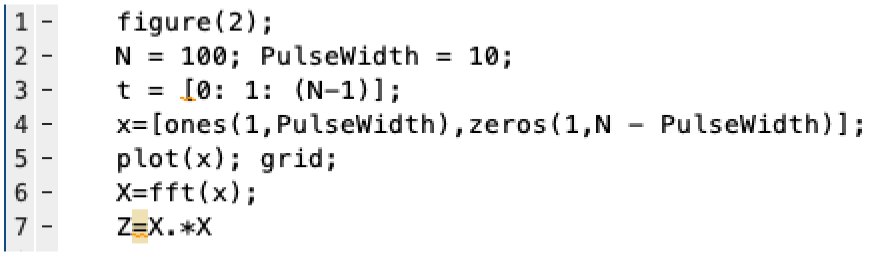
```
figure(1)
N=100;
PulseWidth=10;
t=[0:1:(N-1)];
x=[ones(1,PulseWidth), zeros(1,N-PulseWidth)];
stairs(t,x); grid on; axis([-10,110,-0.1,1.1])
Xf=fft(x);
f=[-(N/2):1:(N/2)-1]*(1/N);
subplot(2, 1, 1); plot(f,fftshift( abs(Xf))); grid on;
subplot(2, 1, 2); plot(f,fftshift(angle(Xf))); grid on;
xhat = ifft(Xf);
z = (conv(x, x));
plot(z); grid;
```

A2

Figure A2:



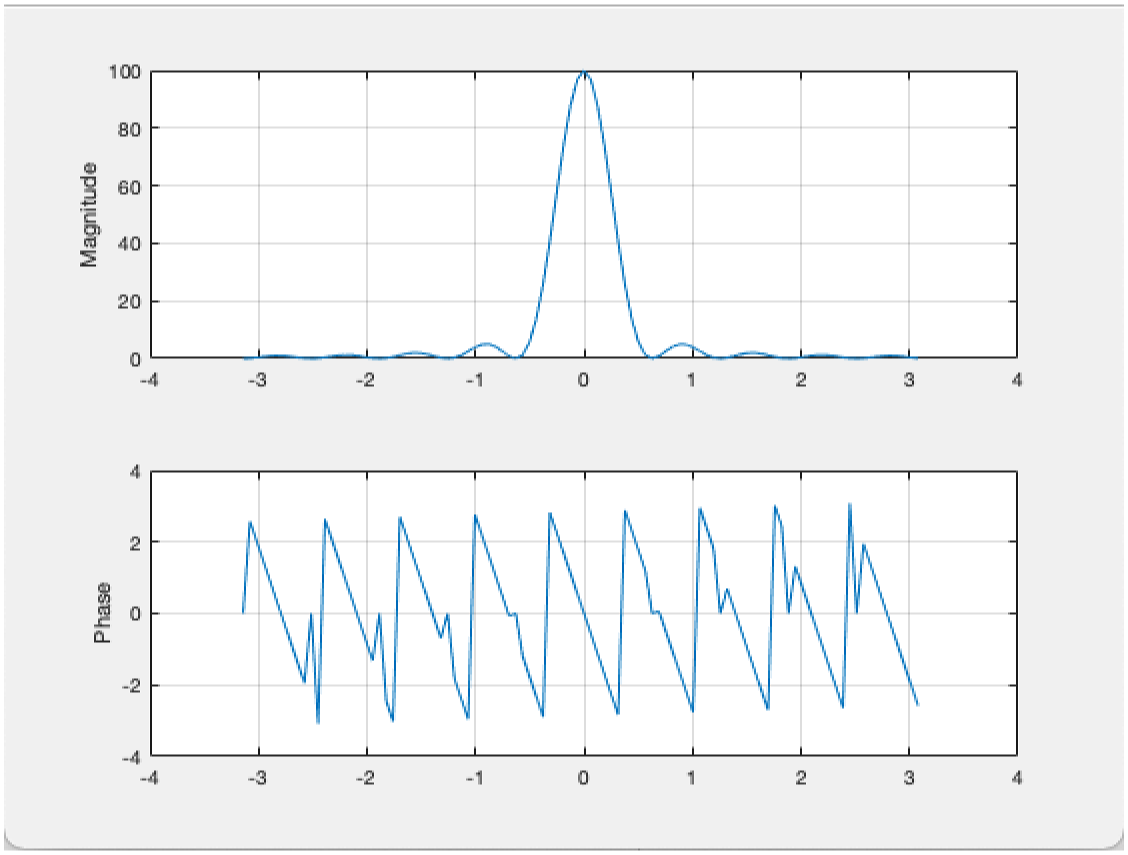
Matlab code used to plot Figure A2:



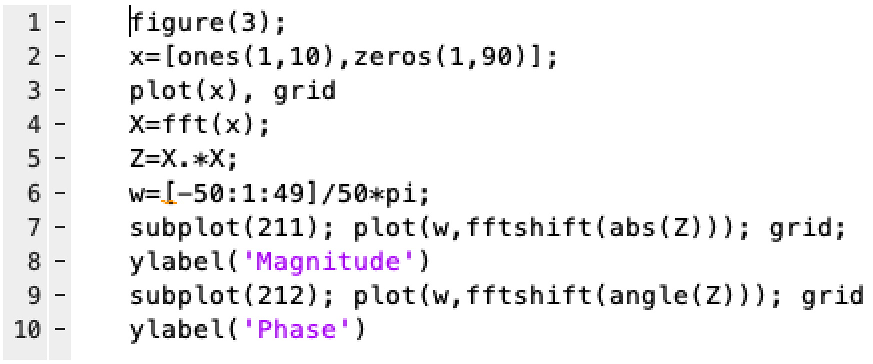
```
figure(2);
N = 100; PulseWidth = 10;
t = [0: 1: (N-1)];
x=[ones(1,PulseWidth),zeros(1,N - PulseWidth)];
plot(x); grid;
X=fft(x);
Z=X.*X
```

A3

Figure A3:

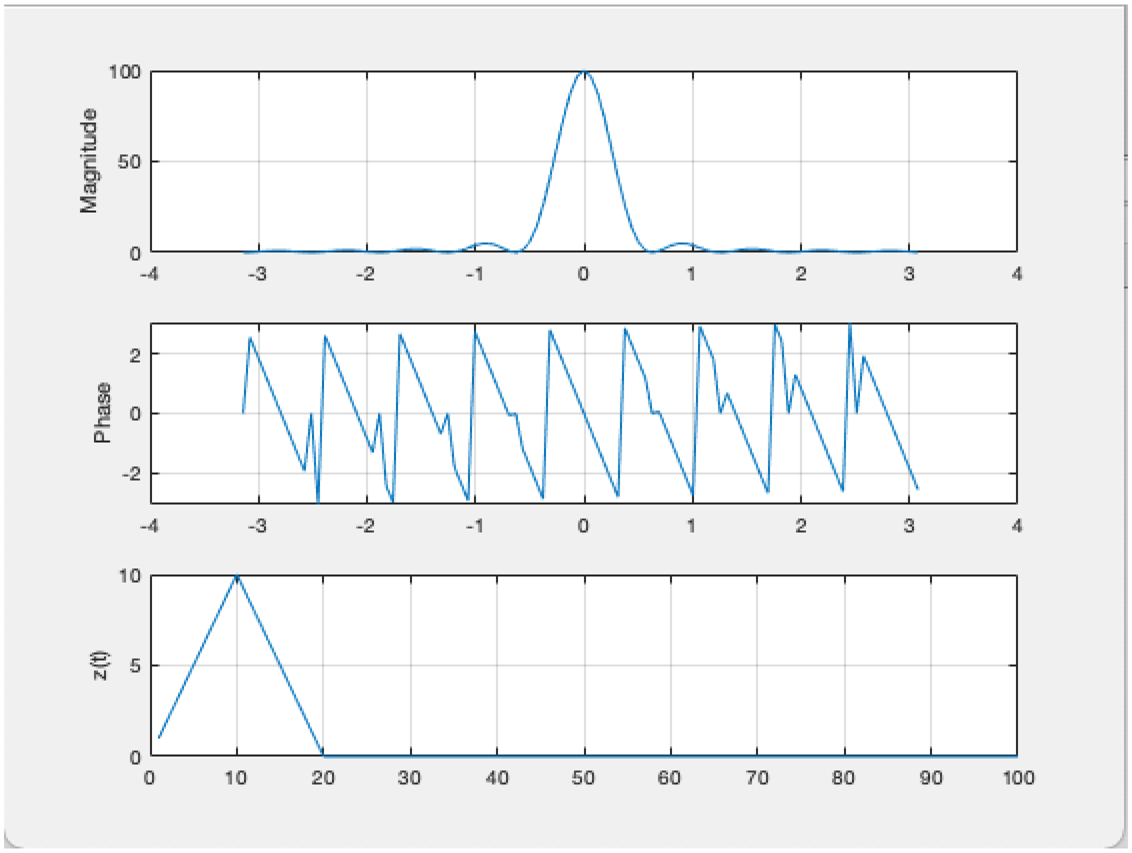


Matlab code used to plot Figure A3:

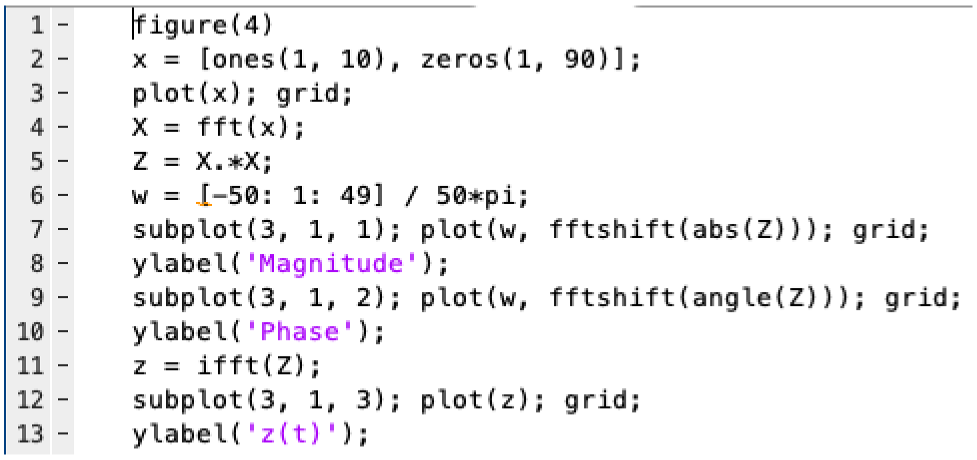

```
figure(3);
x=[ones(1,10),zeros(1,90)];
plot(x), grid
X=fft(x);
Z=X.*X;
w=[-50:1:49]/50*pi;
subplot(211); plot(w,fftshift(abs(Z))); grid;
ylabel('Magnitude')
subplot(212); plot(w,fftshift(angle(Z))); grid
ylabel('Phase')
```

A4

Figure A4:



Matlab code used to plot Figure A4:



```
figure(4)
x = [ones(1, 10), zeros(1, 90)];
plot(x); grid;
X = fft(x);
Z = X.*X;
w = [-50: 1: 49] / 50*pi;
subplot(3, 1, 1); plot(w, fftshift(abs(Z))); grid;
ylabel('Magnitude');
subplot(3, 1, 2); plot(w, fftshift(angle(Z))); grid;
ylabel('Phase');
z = ifft(Z);
subplot(3, 1, 3); plot(z); grid;
ylabel('z(t)');
```

A4 Explanation:

A1 and A4 graphs were identical due to the Fourier transform property which states that; the product of two functions in the frequency domain is equal to the convolution of two functions in the time domain. f(t) -> F(w) & g(t) -> G(w) then f(t)*g(t) -> F(w)*G(w). In A4, the product in the frequency domain was generated. -> F(w)XG(w) In A1, the convolution of two functions in the time domain was generated. -> f(t)*g(t) Which resulted in the same graph, proving the Fourier Transform property.

A5

Figure A5a:

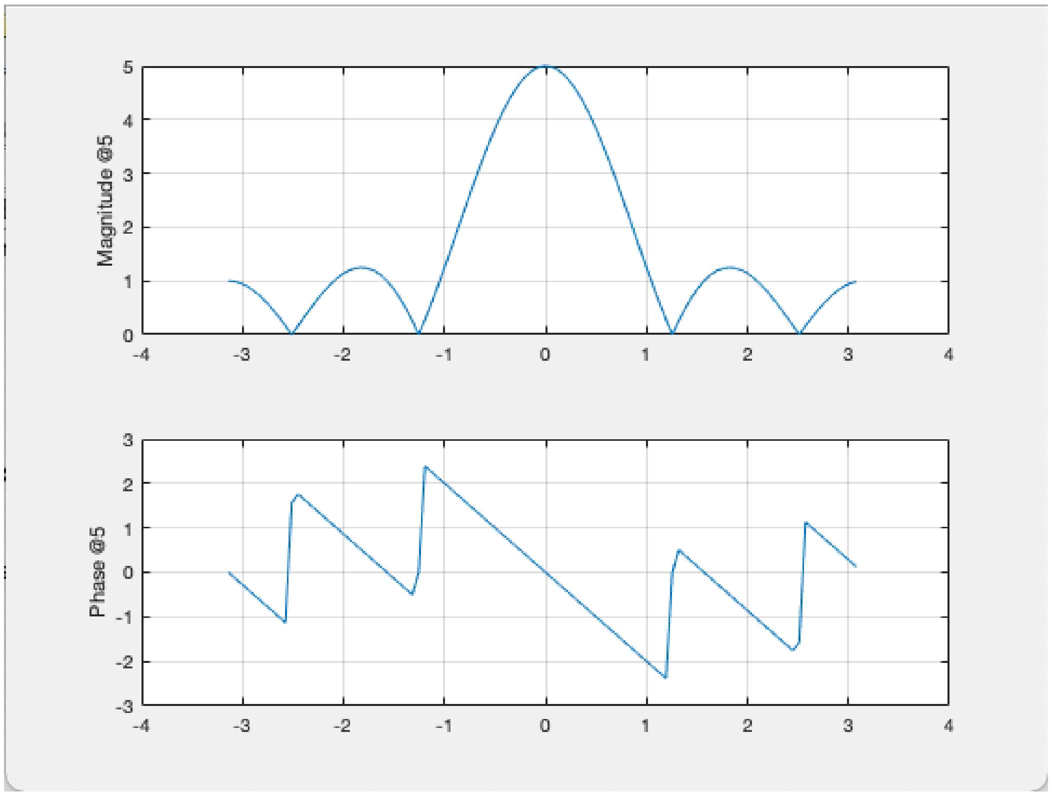


Figure A5b:

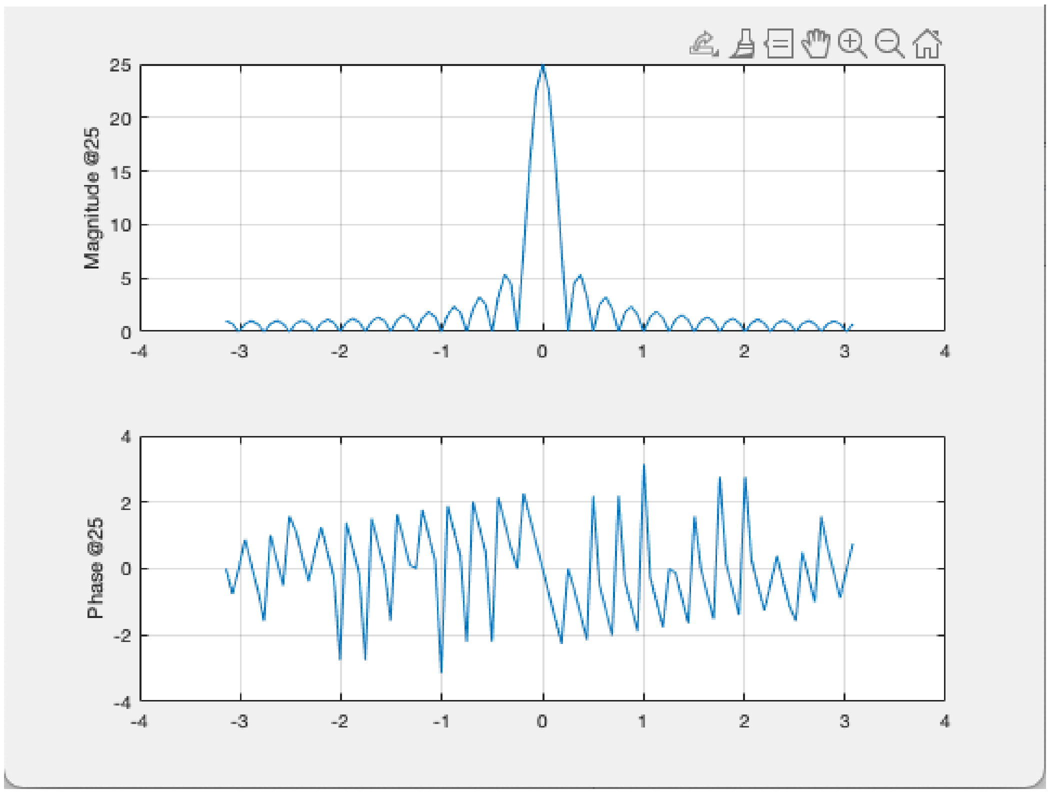


Figure A5c:

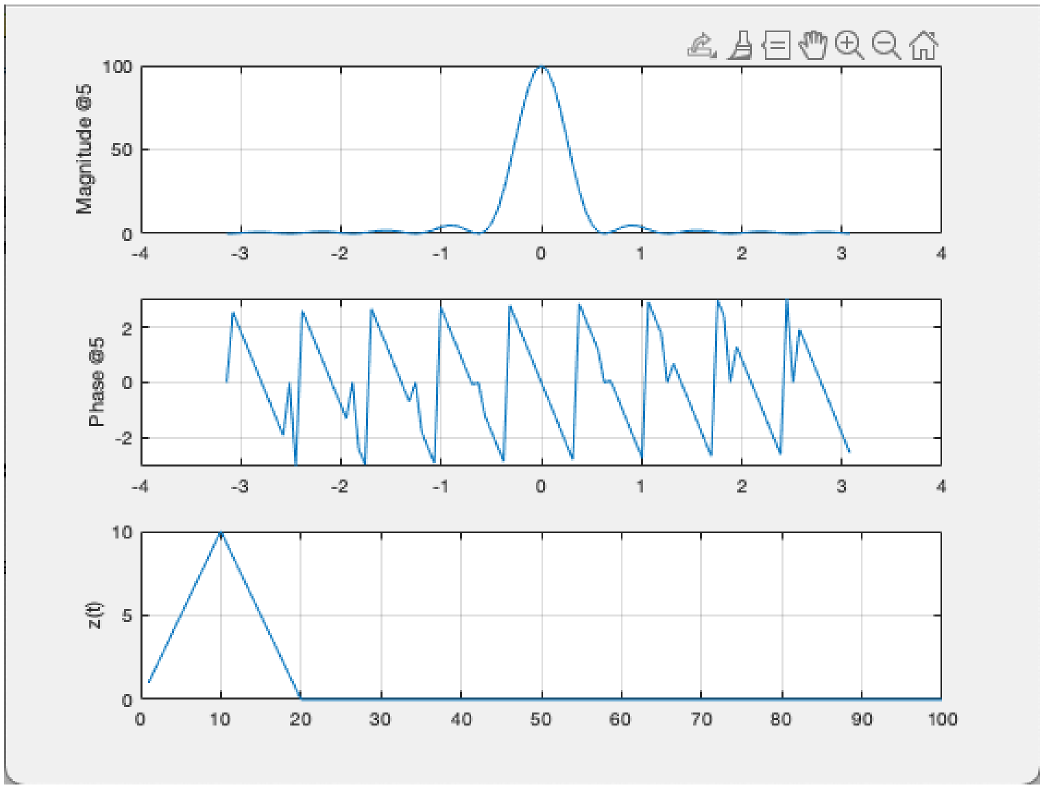
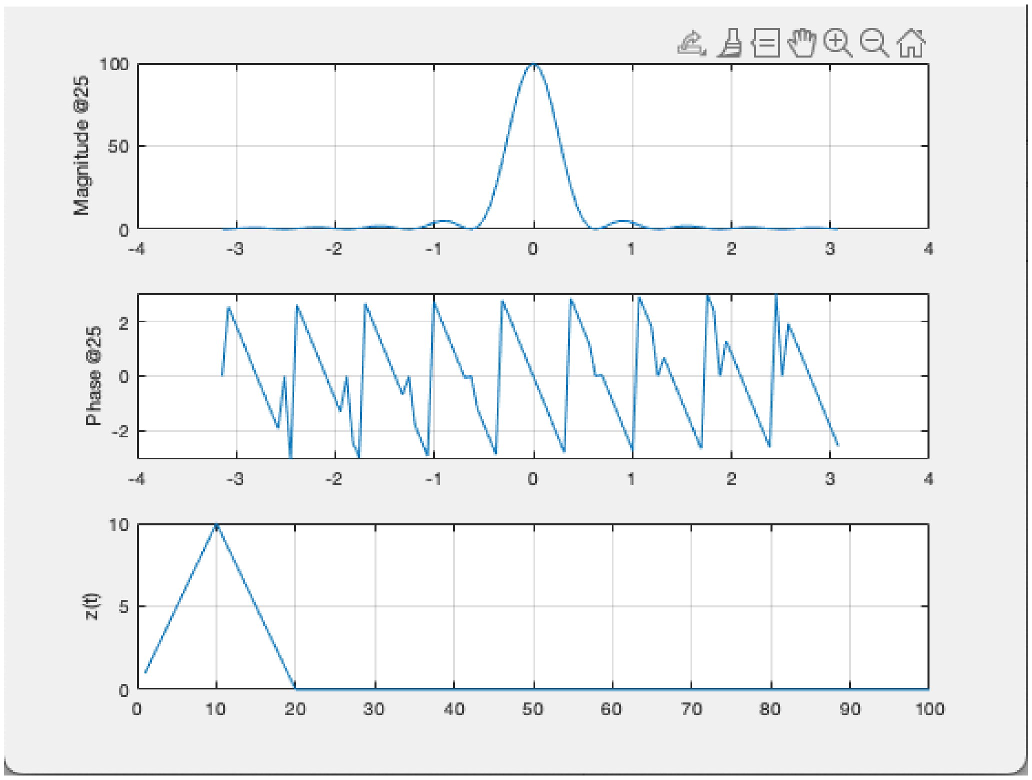
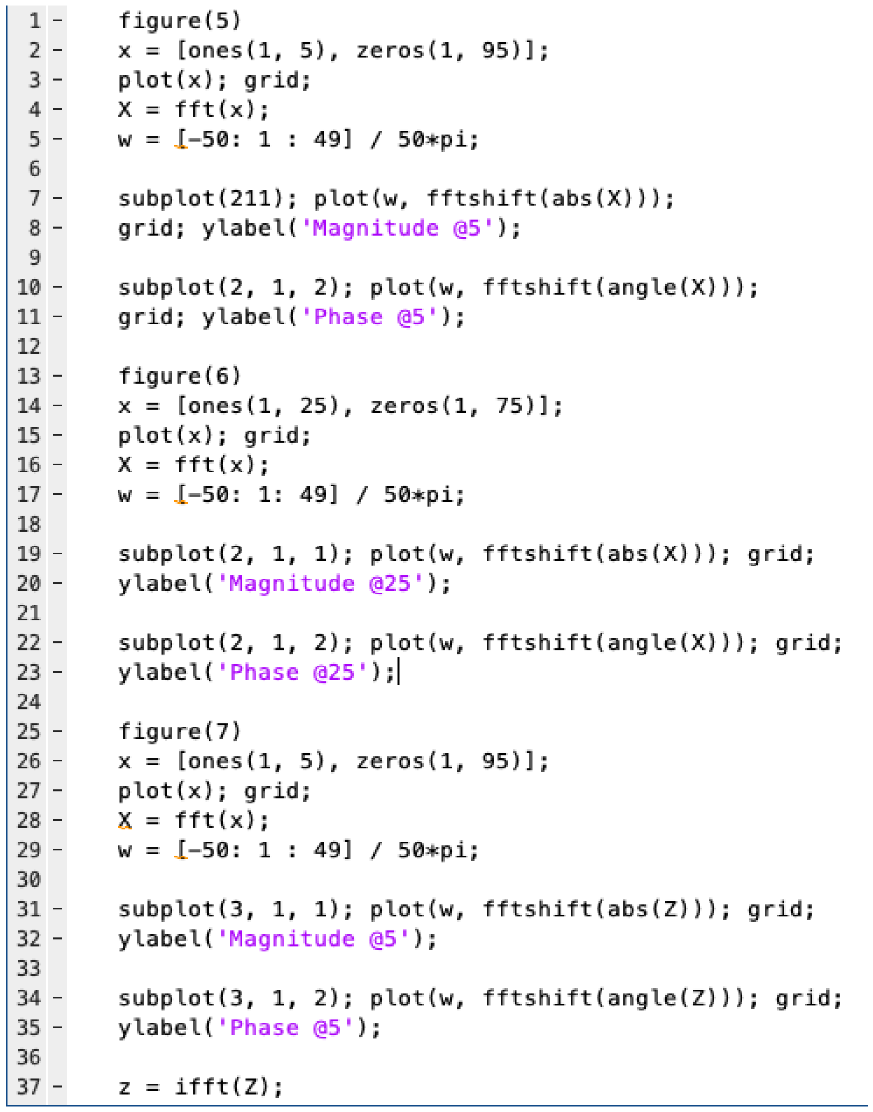


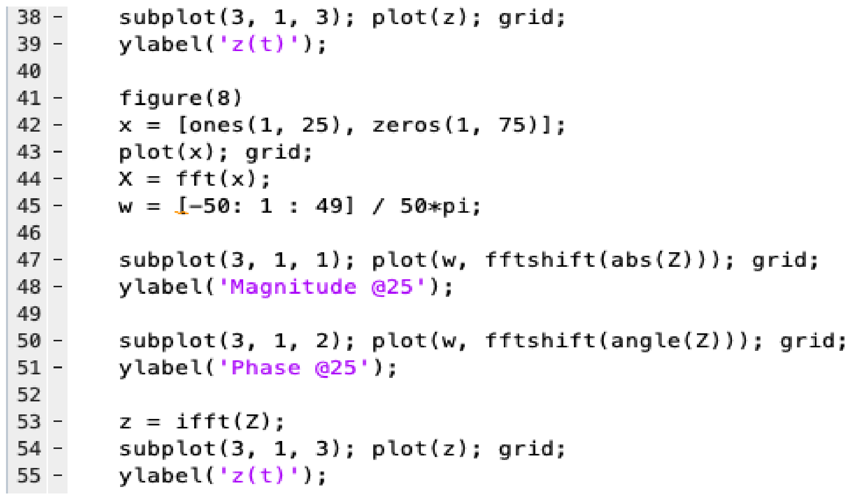
Figure A5d:



Matlab code used to plot Figure A5a - A5d:



```matlab
figure(5)
x = [ones(1, 5), zeros(1, 95)];
plot(x); grid;
X = fft(x);
w = [-50: 1 : 49] / 50*pi;

subplot(211); plot(w, fftshift(abs(X)));
grid; ylabel('Magnitude @5');

subplot(2, 1, 2); plot(w, fftshift(angle(X)));
grid; ylabel('Phase @5');

figure(6)
x = [ones(1, 25), zeros(1, 75)];
plot(x); grid;
X = fft(x);
w = [-50: 1: 49] / 50*pi;

subplot(2, 1, 1); plot(w, fftshift(abs(X))); grid;
ylabel('Magnitude @25');

subplot(2, 1, 2); plot(w, fftshift(angle(X))); grid;
ylabel('Phase @25');

figure(7)
x = [ones(1, 5), zeros(1, 95)];
plot(x); grid;
X = fft(x);
w = [-50: 1 : 49] / 50*pi;

subplot(3, 1, 1); plot(w, fftshift(abs(Z))); grid;
ylabel('Magnitude @5');

subplot(3, 1, 2); plot(w, fftshift(angle(Z))); grid;
ylabel('Phase @5');

z = ifft(Z);
```



```matlab
subplot(3, 1, 3); plot(z); grid;
ylabel('z(t)');

figure(8)
x = [ones(1, 25), zeros(1, 75)];
plot(x); grid;
X = fft(x);
w = [-50: 1 : 49] / 50*pi;

subplot(3, 1, 1); plot(w, fftshift(abs(Z))); grid;
ylabel('Magnitude @25');

subplot(3, 1, 2); plot(w, fftshift(angle(Z))); grid;
ylabel('Phase @25');

z = ifft(Z);
subplot(3, 1, 3); plot(z); grid;
ylabel('z(t)');
```

A5 Explanation:

The sinc function was altered in each pulse with different pulse-widths. This is due to the Time Scaling property of Fourier Transform. According to this property, the vertical and horizontal scaling of the graphs generated by the pulse indicated are altered. For pulses 5 and 25, the width and amplitude are changing. The graphs demonstrate this property at values of 5 and 25. For pulse width of 5, the sinc function had the amplitude of 5 and the horizontal values were compressed by a factor of 5. For pulse width of 25, the amplitude of the sinc function was 25 and the horizontal values were compressed by a factor of 25.

A6

Figure A6a:

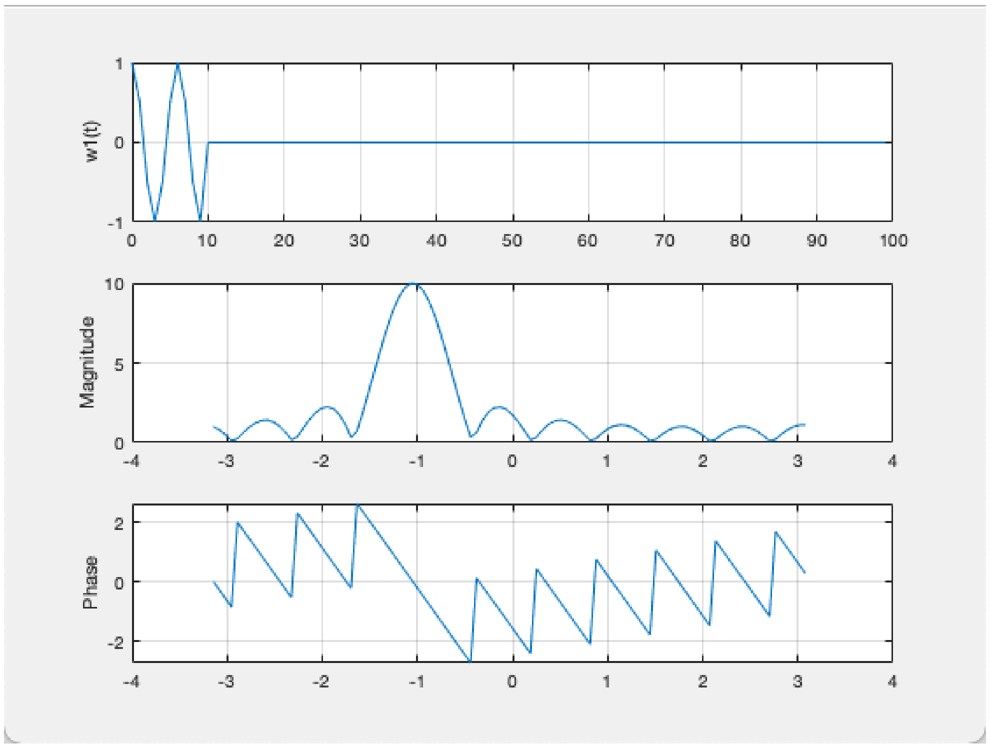


Figure A6b:

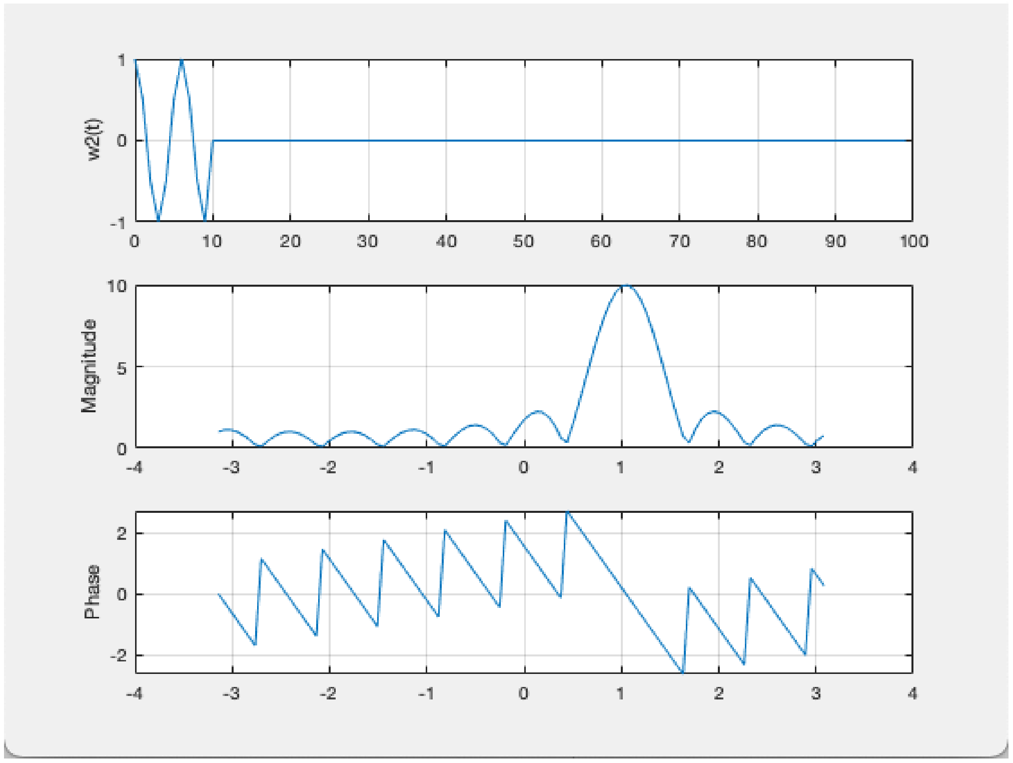
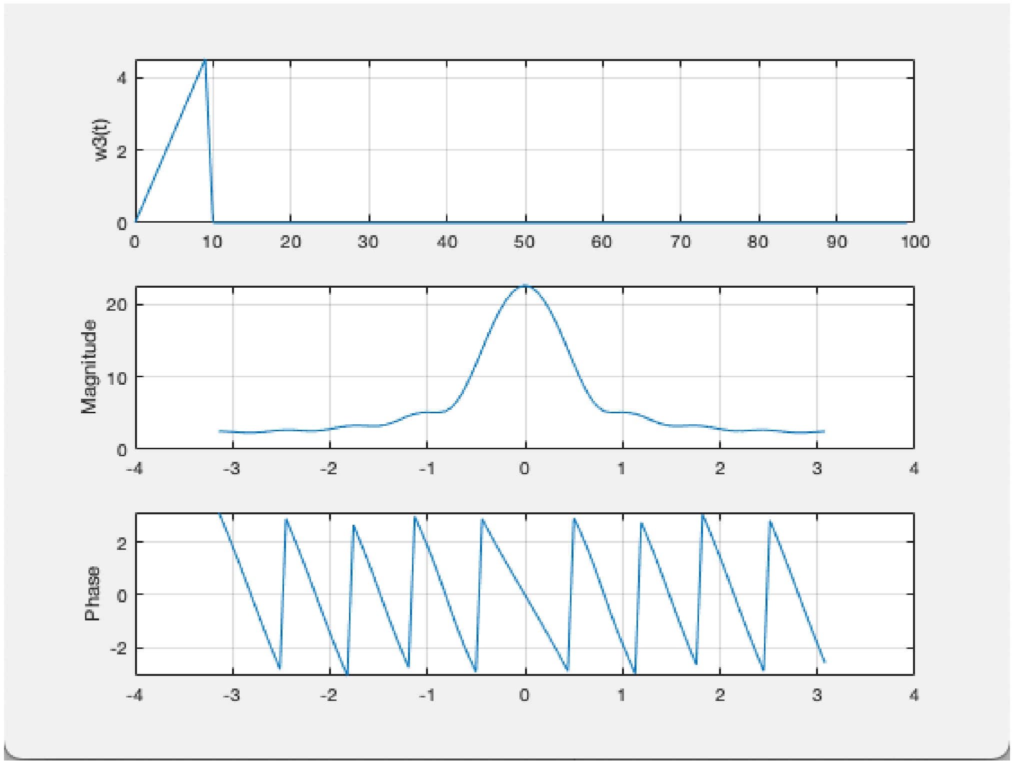
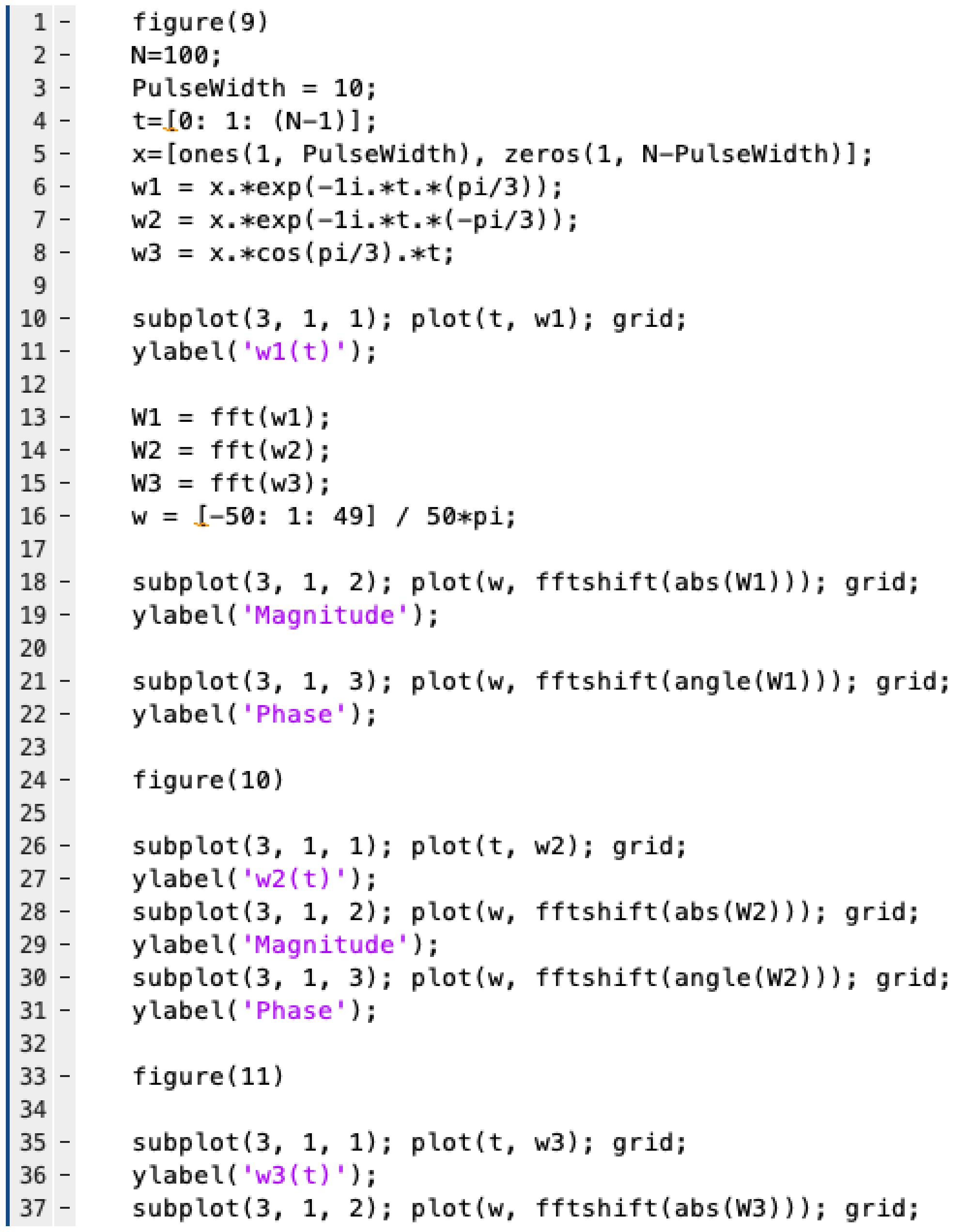


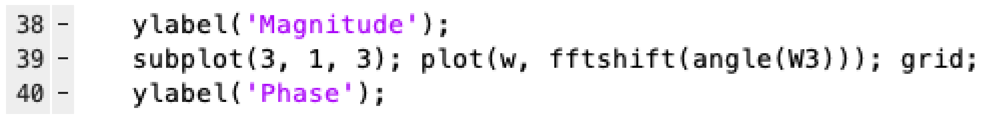
Figure A6c:


w3(t)
Magnitude
Phase
4
2
0
0
10
20
30
40
50
60
70
80
90
100
20
10
0
-4
-3
-2
-1
0
1
2
3
4
2
0
-2
-4
-3
-2
-1
0
1
2
3
4

Matlab code used to plot Figure A6a - A6c:



```
figure(9)
N=100;
PulseWidth = 10;
t=[0: 1: (N-1)];
x=[ones(1, PulseWidth), zeros(1, N-PulseWidth)];
w1 = x.*exp(-1i.*t.*(pi/3));
w2 = x.*exp(-1i.*t.*(-pi/3));
w3 = x.*cos(pi/3).*t;

subplot(3, 1, 1); plot(t, w1); grid;
ylabel('w1(t)');

W1 = fft(w1);
W2 = fft(w2);
W3 = fft(w3);
w = [-50: 1: 49] / 50*pi;

subplot(3, 1, 2); plot(w, fftshift(abs(W1))); grid;
ylabel('Magnitude');

subplot(3, 1, 3); plot(w, fftshift(angle(W1))); grid;
ylabel('Phase');

figure(10)

subplot(3, 1, 1); plot(t, w2); grid;
ylabel('w2(t)');
subplot(3, 1, 2); plot(w, fftshift(abs(W2))); grid;
ylabel('Magnitude');
subplot(3, 1, 3); plot(w, fftshift(angle(W2))); grid;
ylabel('Phase');

figure(11)

subplot(3, 1, 1); plot(t, w3); grid;
ylabel('w3(t)');
subplot(3, 1, 2); plot(w, fftshift(abs(W3))); grid;
```



```
ylabel('Magnitude');
subplot(3, 1, 3); plot(w, fftshift(angle(W3))); grid;
ylabel('Phase');
```

A6 Explanation:

The graphs generated for Parts W + and - have shifted versions of the graph generated in Problem A.3. The property of Fourier Transform that was used is the pertaining to multiplying any function with an exponential. In this case, the first exponential was e^j(pi/3)t where the fourier transform shifts the function by (pi/3)t. The second exponential was e^j-(pi/3)t where the fourier transform shifts the function by -(pi/3)t. The graph generated in Part c, has the frequency shifted version of A.3. Since the cosine function is composed of two specific Dn, which produce two spirals according to Euler's formula, the graph in Part c has two peaks at the negative and positive natural friencies representing the two Fourier coefficients of the cosine function.

B1

Figure 1:

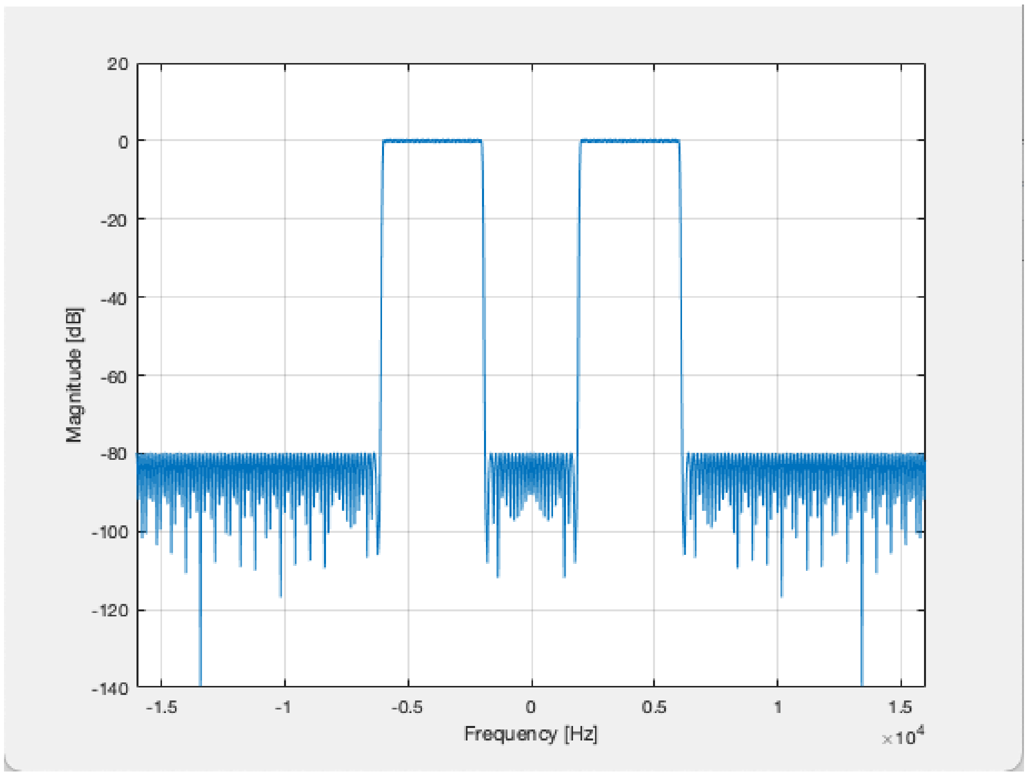

20
0
-20
-40
-60
-80
-100
-120
-140
Magnitude [dB]
-1.5
-1
-0.5
0
0.5
1
1.5
Frequency [Hz]
$\times 10^4$

Figure 2:

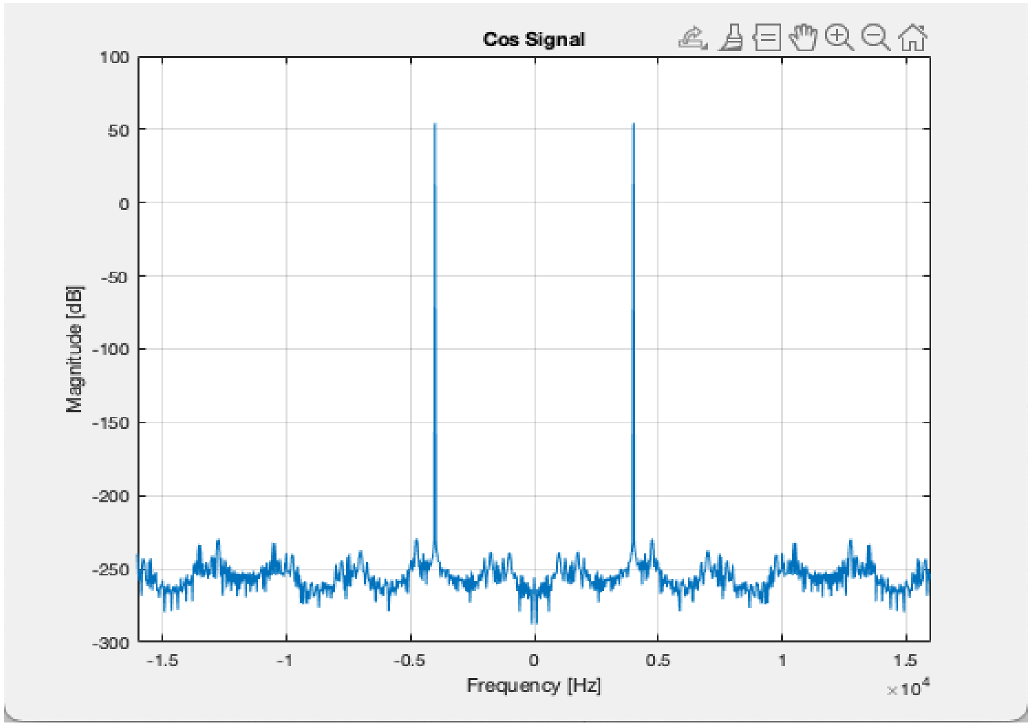


Figure 3:

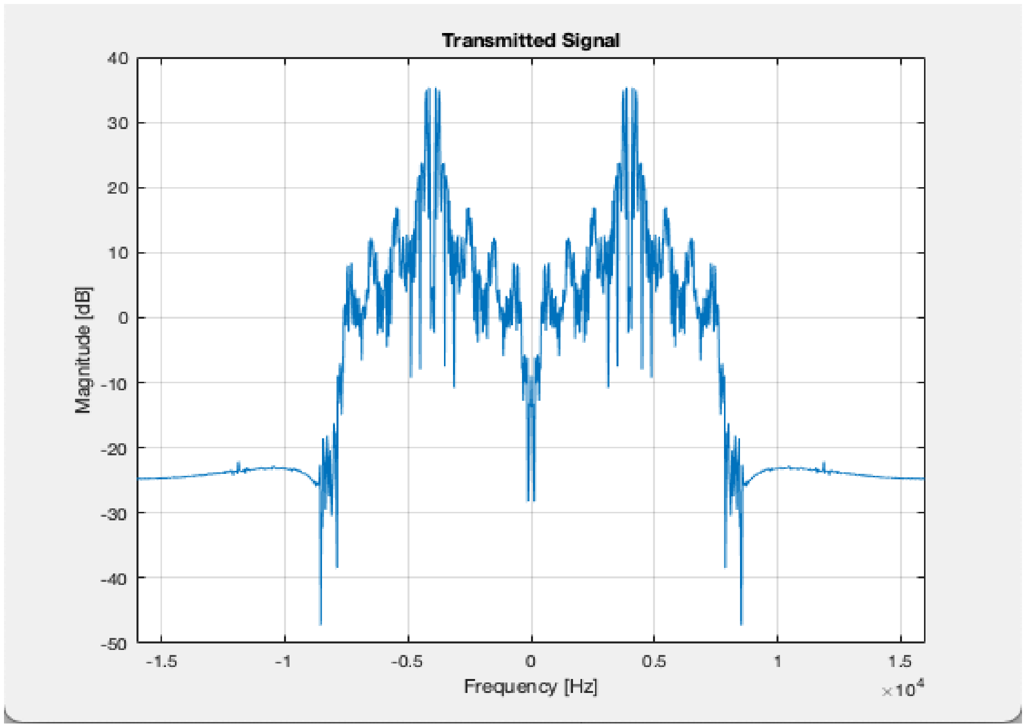


Figure 4:

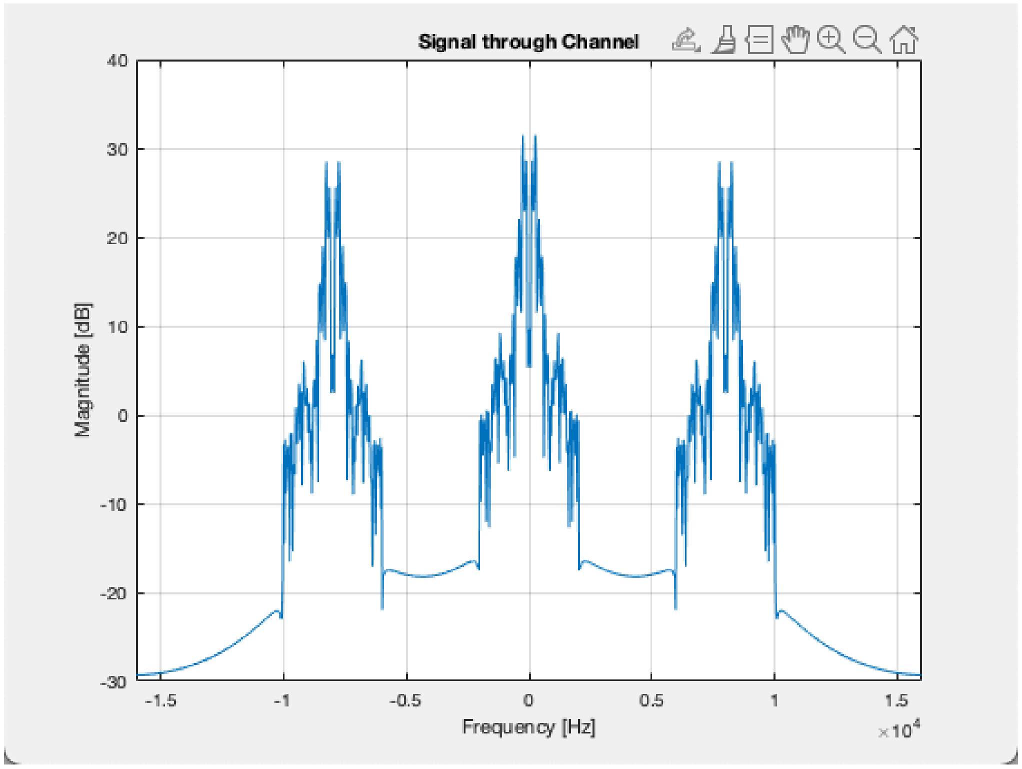


Figure 5:

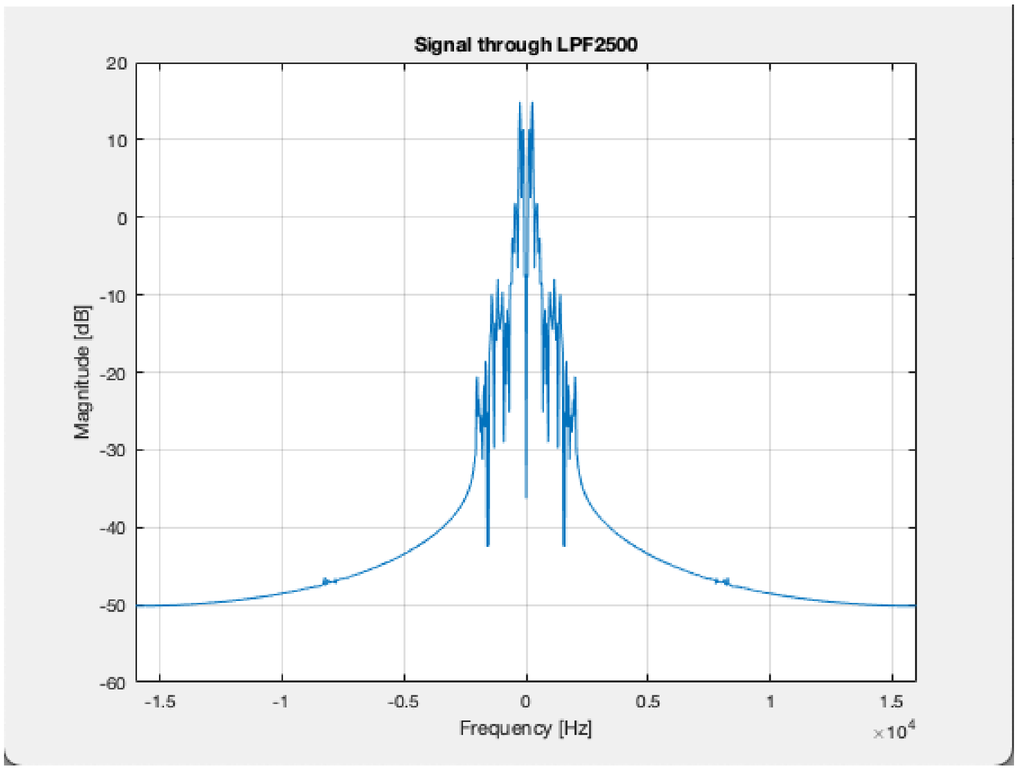
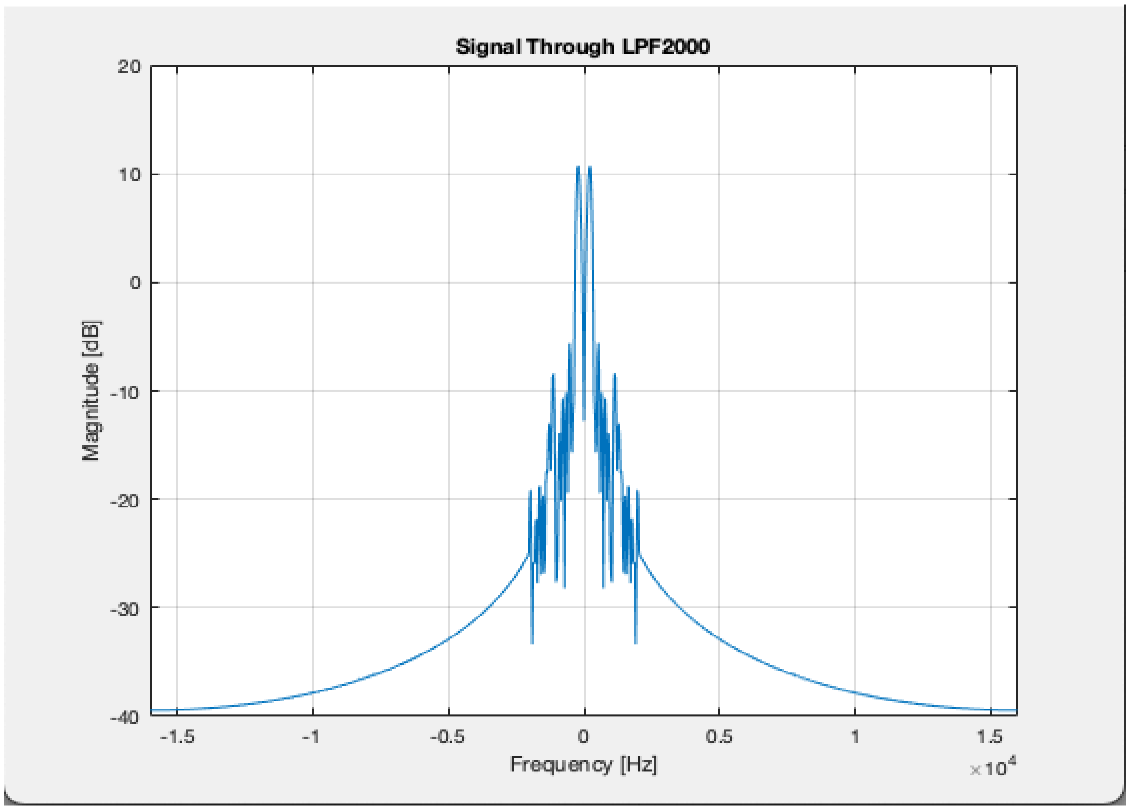
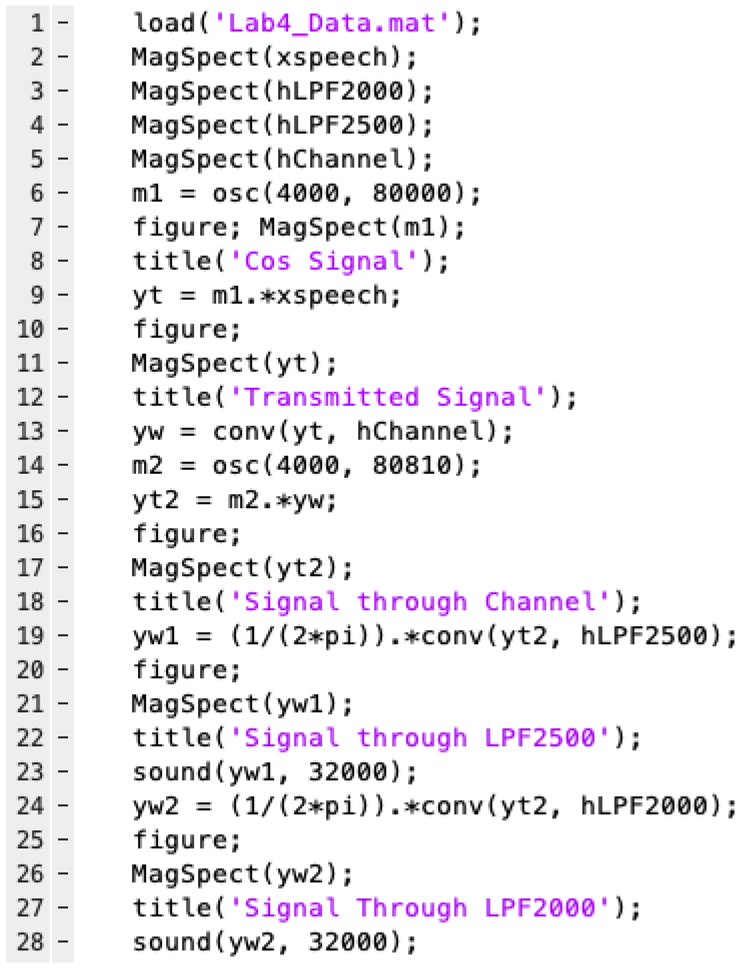


Figure 6:


Signal Through LPF2000
Magnitude [dB]
20
10
0
-10
-20
-30
-40
-1.5
-1
-0.5
0
0.5
1
1.5
Frequency [Hz]
$\times 10^4$

Matlab code used to plot Figure 1 - Figure 6:



```
load('Lab4_Data.mat');
MagSpect(xspeech);
MagSpect(hLPF2000);
MagSpect(hLPF2500);
MagSpect(hChannel);
m1 = osc(4000, 80000);
figure; MagSpect(m1);
title('Cos Signal');
yt = m1.*xspeech;
figure;
MagSpect(yt);
title('Transmitted Signal');
yw = conv(yt, hChannel);
m2 = osc(4000, 80810);
yt2 = m2.*yw;
figure;
MagSpect(yt2);
title('Signal through Channel');
yw1 = (1/(2*pi)).*conv(yt2, hLPF2500);
figure;
MagSpect(yw1);
title('Signal through LPF2500');
sound(yw1, 32000);
yw2 = (1/(2*pi)).*conv(yt2, hLPF2000);
figure;
MagSpect(yw2);
title('Signal Through LPF2000');
sound(yw2, 32000);
```

Demo Appendix:

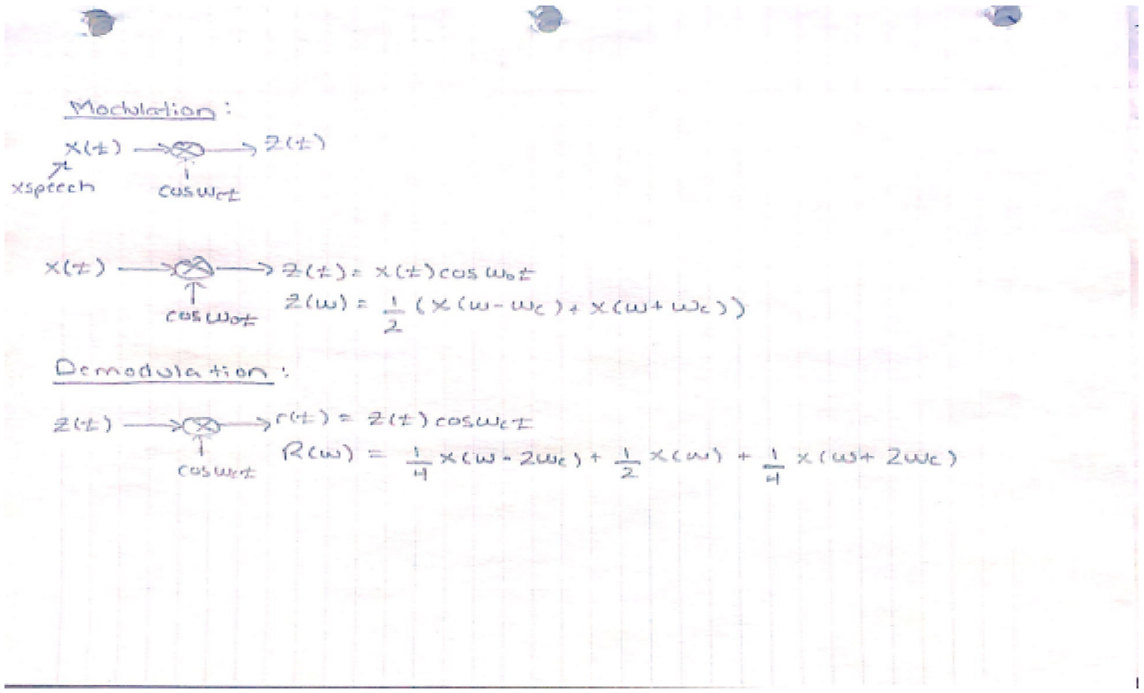
Above is the code for Lab 5 part B. In order to run the code the TA should do the following:

1. Download the following files: osc.m, MagSpect.m, Lab4_Data.mat and Lab4_B1.m file. Please make sure to save the files in your respective MatLab file directory in order to open them in MatLab.
2. Compile the osc.m and MagSpect.m files provided to check for errors.
3. If the osc.m and MagSpect.m files throw no errors, compile the Lab4_B1.m file to generate the following graphs: **Figure 2** Cosine Signal, **Figure 3** Transmitted Signal, **Figure 4** Signal through Channel, **Figure 5** Signal through LPF2500 and **Figure 6** Signal through LPF2000.

How the Code works:

Upon compiling the Lab4_B1 file, the Lab4_Data.mat is loaded and all the information contained in the file is retrieved. The MagSpect command is a part of the MagSpect.m file and simplifies the process of generating and plotting the fourier transform in a similar fashion as the fft() command used in Part A. With this command we generated and plotted the magnitude spectra of xspeech, hLPF250, hLPF2000 and hChannel **(Figure 1)**. Next, the osc function from the osc.m was called for the purpose of generating user specified number of samples of the sinusoidal waveform $cos(2F_o t)$. In the case of this lab, 80000 samples at a user defined frequency of 4000 were generated based on the specified waveform and stored in the variable m1. Using MagSpect(), the magnitude spectra of m1 was displayed **(Figure 2)**. The output, yt, was then determined and MagSpect() was called to display the magnitude spectra **(Figure 3).** This output was then taken, and the convolution between yt and hChannel was taken and stored in ytw. The function osc was used again to generate 80810 samples at a user defined frequency of 4000 of the sinusoidal waveform and stored in the variable m2. The output, yt2, was then determined and MagSpect was called to display the magnitude spectra **(Figure 4).** Next the signal yw1 and yw2 were determined by taking the convolution between (yt2, hLPF2500) and (yt2, hLPF2000) respectively. MagSpect was then called and the magnitude spectra of yw1 was displayed **(Figure 5)** and the magnitude spectra of yw2 was displayed **(Figure 6).**

Block Diagram:



Modulation:

$$\begin{tikzcd}
x(t) \arrow[r] & \otimes \arrow[r] & z(t) \\
\text{xspeech} \arrow[u] & \cos w_c t \arrow[u] &
\end{tikzcd}$$

$$\begin{tikzcd}
x(t) \arrow[r] & \otimes \arrow[r] & z(t) = x(t)\cos w_0 t \\
& \cos w_0 t \arrow[u] &
\end{tikzcd}$$

$$Z(w) = \frac{1}{2}\left(X(w - w_c) + X(w + w_c)\right)$$

Demodulation:

$$\begin{tikzcd}
z(t) \arrow[r] & \otimes \arrow[r] & r(t) = z(t)\cos w_c t \\
& \cos w_c t \arrow[u] &
\end{tikzcd}$$

$$R(w) = \frac{1}{4}X(w - 2w_c) + \frac{1}{2}X(w) + \frac{1}{4}X(w + 2w_c)$$