EEE 391

Matlab Assignment 2

İlke Doğan 21702215

Section 1

20/11/2021

TABLE OF CONTENTS

Introduction

This report is about where the concepts learned in the signals and systems course appear in real life and their real-time applications.

1. DTMF Signal and Transceiver

Part 1.0

Code: Whole Code of Question 1

```
%% Part 1.2
Number = [0 5 0 7 6 5 6 6 5 6 7];
x = DTMFTRA(Number);
soundsc(x,16384);

%% Part 1.3
Number = [0 2 2 1 5];
x = DTMFTRA(Number);
X = FT(x);
omega=linspace(-16384*pi,16384*pi,16384*2.5+1);
omega=omega(1:end-1);

plot(omega,abs(X)), title("Fourier Transform");
xlabel("Frequency");
ylabel("Magnitude");

%% Part 1.4 & 1.5 & 1.6
% index 3
rectangle3 = [zeros(1, 16384) ones(1, 0.5*16384), zeros(1,16384)];
x3 = x.*rectangle3;
X = FT(x3);
omega=linspace(-16384*pi,16384*pi,16384*2.5+1);
omega=omega(1:end-1);
plot(omega,abs(X));

% index 2
rectangle2 = [zeros(1, 0.5*16384) ones(1, 0.5*16384), zeros(1,1.5*16384)];
x2 = x.*rectangle2;
X = FT(x2);
omega=linspace(-16384*pi,16384*pi,16384*2.5+1);
omega=omega(1:end-1);
plot(omega,abs(X));

% index 1
rectangle1 = [ones(1, 0.5*16384), zeros(1,2*16384)];
x1 = x.*rectangle1;
X = FT(x1);
omega=linspace(-16384*pi,16384*pi,16384*2.5+1);
omega=omega(1:end-1);
plot(omega,abs(X));


% index 4
rectangle4 = [zeros(1, 1.5*16384) ones(1, 0.5*16384) zeros(1,0.5*16384)];
x4 = x.*rectangle4;
X = FT(x4);
omega=linspace(-16384*pi,16384*pi,16384*2.5+1);
omega=omega(1:end-1);
plot(omega,abs(X));

% index 5
rectangle5 = [zeros(1, 2*16384) ones(1, 0.5*16384)];
x5 = x.*rectangle5;
X = FT(x5);
omega=linspace(-16384*pi,16384*pi,16384*2.5+1);
omega=omega(1:end-1);
plot(omega,abs(X));

```

```
%% Part 1.1 - General view of the Question1
function [x] = DTMFTRA(Number)
Fr=[697 770 852 941];
Fc=[1209 1336 1477 1633];
numLength = length(Number);
x = [];
t = 0;
index = 1;
while (t < 0.5 * numLength)
    toAdd = 0;
    switch Number(index)
        case 0
            toAdd = cos(2 * pi * Fr(4) * t) + cos(2 * pi * Fc(2) * t);
        case 1
            toAdd = cos(2 * pi * Fr(1) * t) + cos(2 * pi * Fc(1) * t);
        case 2
            toAdd = cos(2 * pi * Fr(1) * t) + cos(2 * pi * Fc(2) * t);
        case 3
            toAdd = cos(2 * pi * Fr(1) * t) + cos(2 * pi * Fc(3) * t);
        case 4
            toAdd = cos(2 * pi * Fr(2) * t) + cos(2 * pi * Fc(1) * t);
        case 5
            toAdd = cos(2 * pi * Fr(2) * t) + cos(2 * pi * Fc(2) * t);
        case 6
            toAdd = cos(2 * pi * Fr(2) * t) + cos(2 * pi * Fc(3) * t);
        case 7
            toAdd = cos(2 * pi * Fr(3) * t) + cos(2 * pi * Fc(1) * t);
        case 8
            toAdd = cos(2 * pi * Fr(3) * t) + cos(2 * pi * Fc(2) * t);
        case 9
            toAdd = cos(2 * pi * Fr(3) * t) + cos(2 * pi * Fc(3) * t);
        otherwise
            disp('Enter in the range 0 to 9 only')
    end
```

```
    x = [x, toAdd];
    if(t > index/2)
        index = index + 1;
    end
    t = t + 1/16384;
end
end
```

Part 1.1

In this part, it was first asked to create a sound with signals. in the given keyboard in the matlab assignment 2 manual.

Code

```
%% Part 1.1 - General view of the Question1
function [x] = DTMFTRA(Number)
Fr=[697 770 852 941];
Fc=[1209 1336 1477 1633];
numLength = length(Number);
x = [];
t = 0;
index = 1;
while (t < 0.5 * numLength)
    toAdd = 0;
    switch Number(index)
        case 0
            toAdd = cos(2 * pi * Fr(4) * t) + cos(2 * pi * Fc(2) * t);
        case 1
            toAdd = cos(2 * pi * Fr(1) * t) + cos(2 * pi * Fc(1) * t);
        case 2
            toAdd = cos(2 * pi * Fr(1) * t) + cos(2 * pi * Fc(2) * t);
        case 3
            toAdd = cos(2 * pi * Fr(1) * t) + cos(2 * pi * Fc(3) * t);
        case 4
            toAdd = cos(2 * pi * Fr(2) * t) + cos(2 * pi * Fc(1) * t);
        case 5
            toAdd = cos(2 * pi * Fr(2) * t) + cos(2 * pi * Fc(2) * t);
        case 6
            toAdd = cos(2 * pi * Fr(2) * t) + cos(2 * pi * Fc(3) * t);
        case 7
            toAdd = cos(2 * pi * Fr(3) * t) + cos(2 * pi * Fc(1) * t);
        case 8
            toAdd = cos(2 * pi * Fr(3) * t) + cos(2 * pi * Fc(2) * t);
        case 9
            toAdd = cos(2 * pi * Fr(3) * t) + cos(2 * pi * Fc(3) * t);
        otherwise
            disp('Enter in the range 0 to 9 only')
    end
```

```
    x = [x, toAdd];
    if(t > index/2)
        index = index + 1;
    end
    t = t + 1/16384;
end
end
```

Part 1.2

Code

```
%% Part 1.2
Number = [0 5 0 7 6 5 6 6 5 6 7];
x = DTMFTRA(Number);
soundsc(x,16384);
```

In part 1.2, the signal code was created in part 1.1. With these signals, a test was requested with the person's own phone number. On my phone number 0 5 0 7 6 5 6 6 5 6 7, the sound made during dialing on the phone and the sounds I received in this application matched. Clearly, it sounds like typing on the keyboard of my phone.

Part 1.3

Code

```
%% Part 1.3
Number = [0 2 2 1 5];
x = DTMFTRA(Number);
X = FT(x);
omega=linspace(-16384*pi,16384*pi,16384*2.5+1);
omega=omega(1:end-1);

plot(omega,abs(X)), title("Fourier Transform");
xlabel("Frequency");
ylabel("Magnitude");

```

In part 1.3, the signal code was created in part 1.1. With these signals, a test was requested with the last 5 digits of students' Bilkent University ID. My ID ends with ***"0 2 2 1 5"***. In this part, the frequencies are viewed by using DTMF. It can be seen from the plot given in Figure 1 and peak numbered plot in Figure 2.

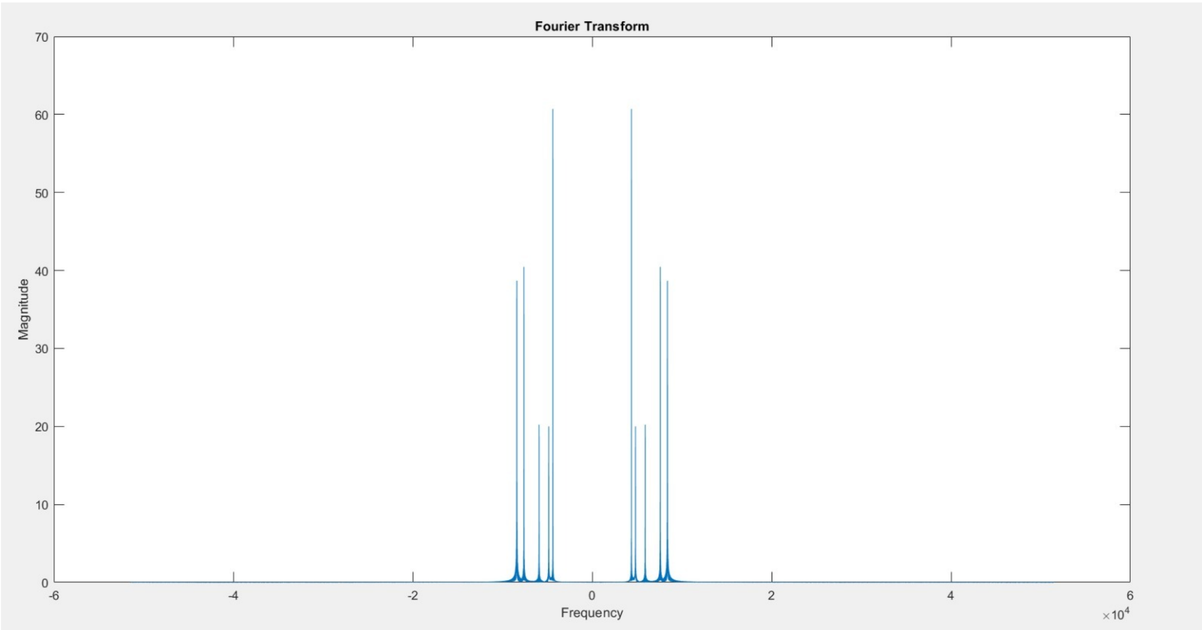

Figure.1. - Fourier Transform

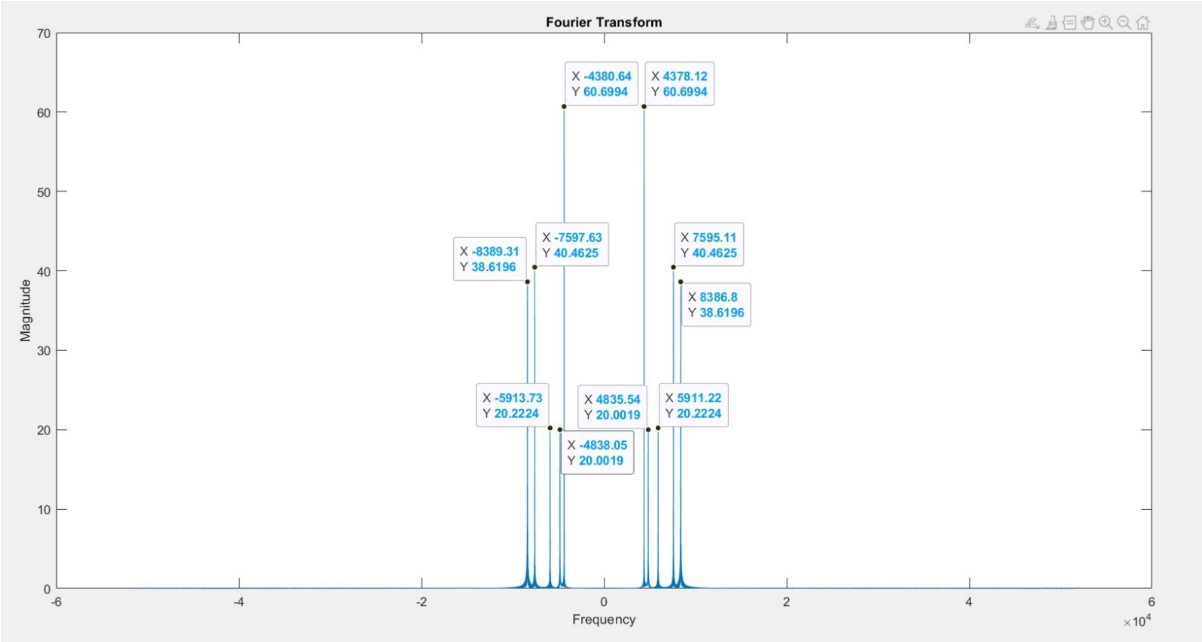

Figure.2. - Fourier Transform With Peak Numbers

Part 1.4 & 1.5 & 1.6

Code

```
% index 3
rectangle3 = [zeros(1, 16384) ones(1, 0.5*16384), zeros(1,16384)];
x3 = x.*rectangle3;
X = FT(x3);
omega=linspace(-16384*pi,16384*pi,16384*2.5+1);
omega=omega(1:end-1);
plot(omega,abs(X));

% index 2
rectangle2 = [zeros(1, 0.5*16384) ones(1, 0.5*16384), zeros(1,1.5*16384)];
x2 = x.*rectangle2;
X = FT(x2);
omega=linspace(-16384*pi,16384*pi,16384*2.5+1);
omega=omega(1:end-1);
plot(omega,abs(X));

% index 1
rectangle1 = [ones(1, 0.5*16384), zeros(1,2*16384)];
x1 = x.*rectangle1;
X = FT(x1);
omega=linspace(-16384*pi,16384*pi,16384*2.5+1);
omega=omega(1:end-1);
plot(omega,abs(X));

% index 4
rectangle4 = [zeros(1, 1.5*16384) ones(1, 0.5*16384) zeros(1,0.5*16384)];
x4 = x.*rectangle4;
X = FT(x4);
omega=linspace(-16384*pi,16384*pi,16384*2.5+1);
omega=omega(1:end-1);
plot(omega,abs(X));

% index 5
rectangle5 = [zeros(1, 2*16384) ones(1, 0.5*16384)];
x5 = x.*rectangle5;
X = FT(x5);
omega=linspace(-16384*pi,16384*pi,16384*2.5+1);
omega=omega(1:end-1);
plot(omega,abs(X));
```

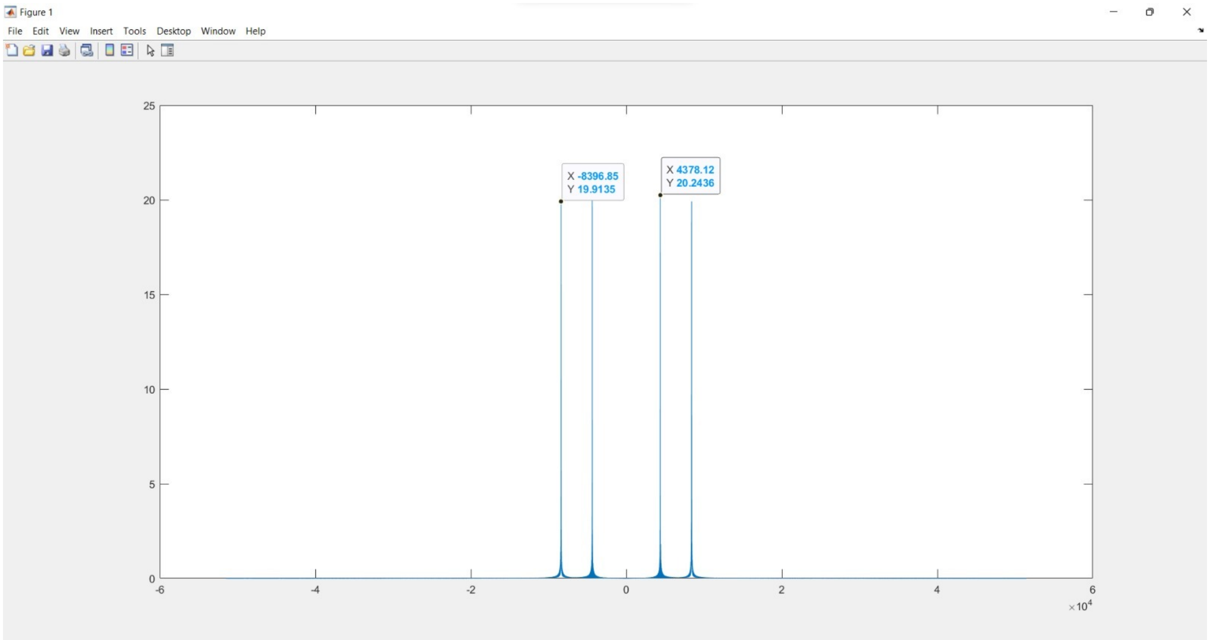

Figure.3. Index(3)

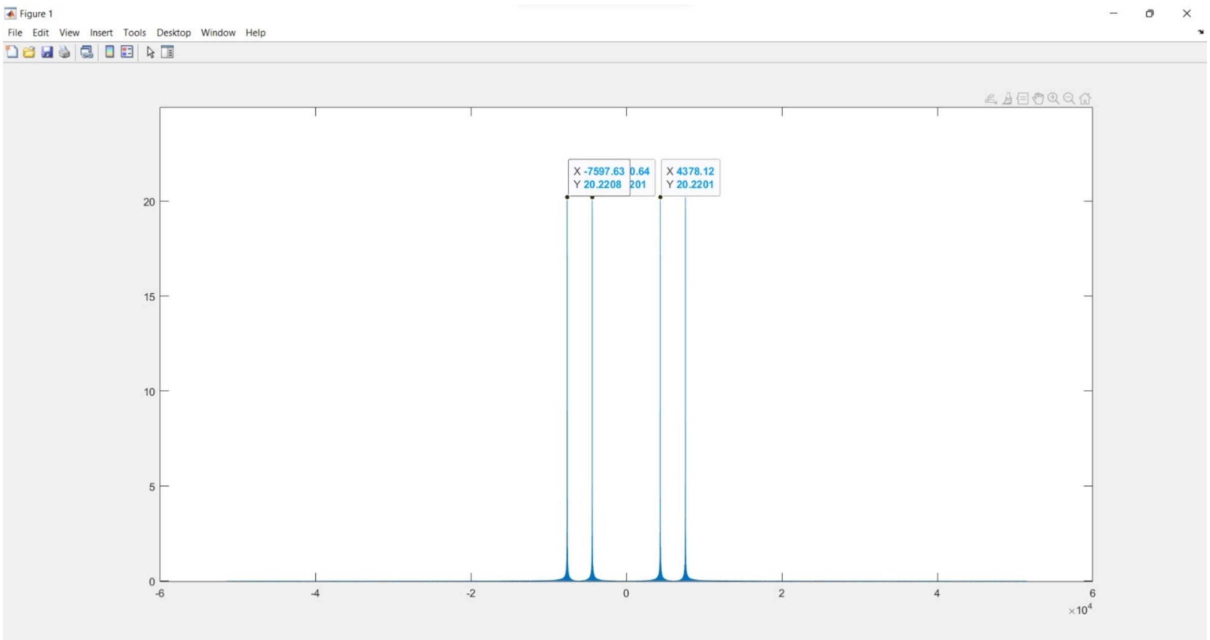

Figure.4. Index(2)

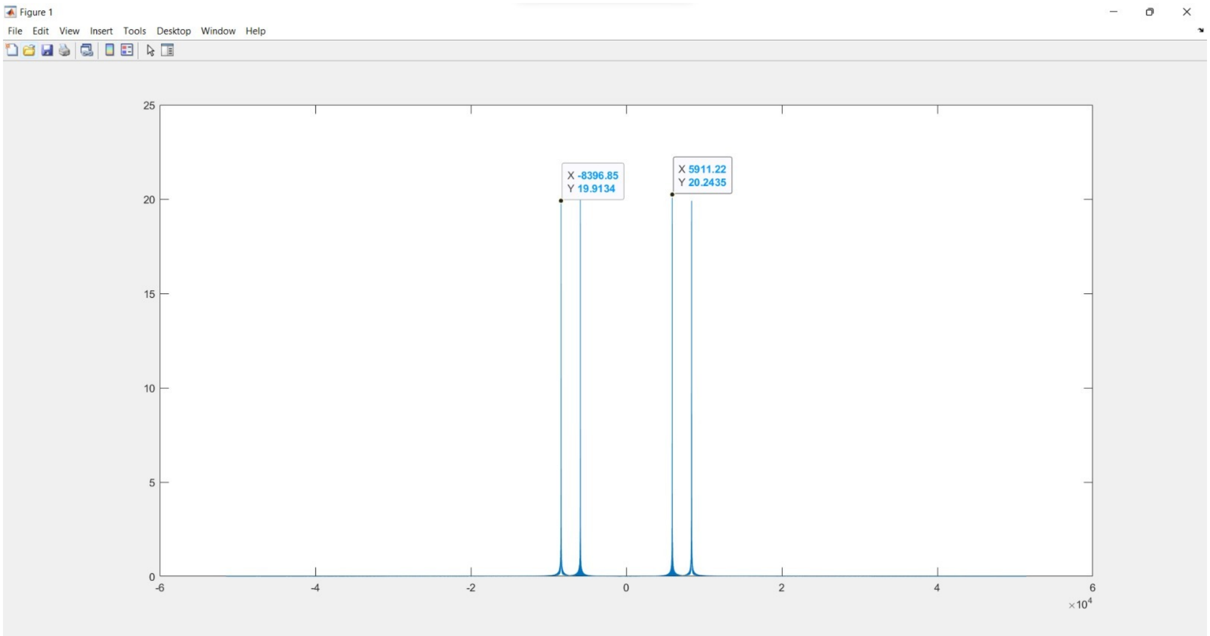

Figure.5 - Index(1)

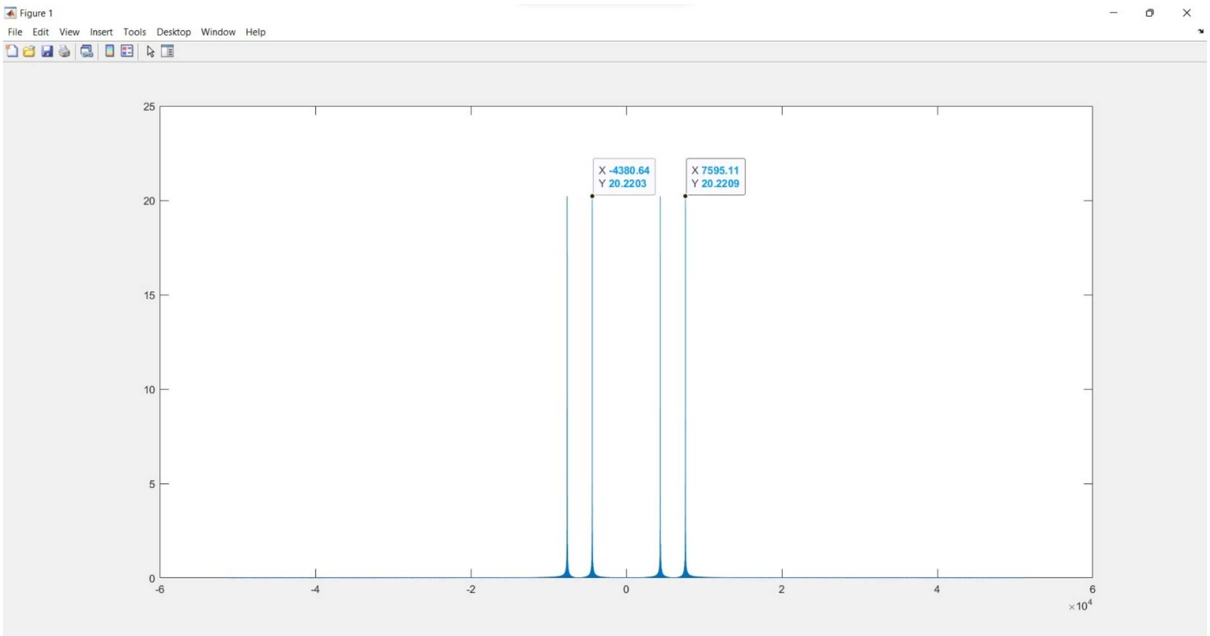

Figure.6. - Index(4)

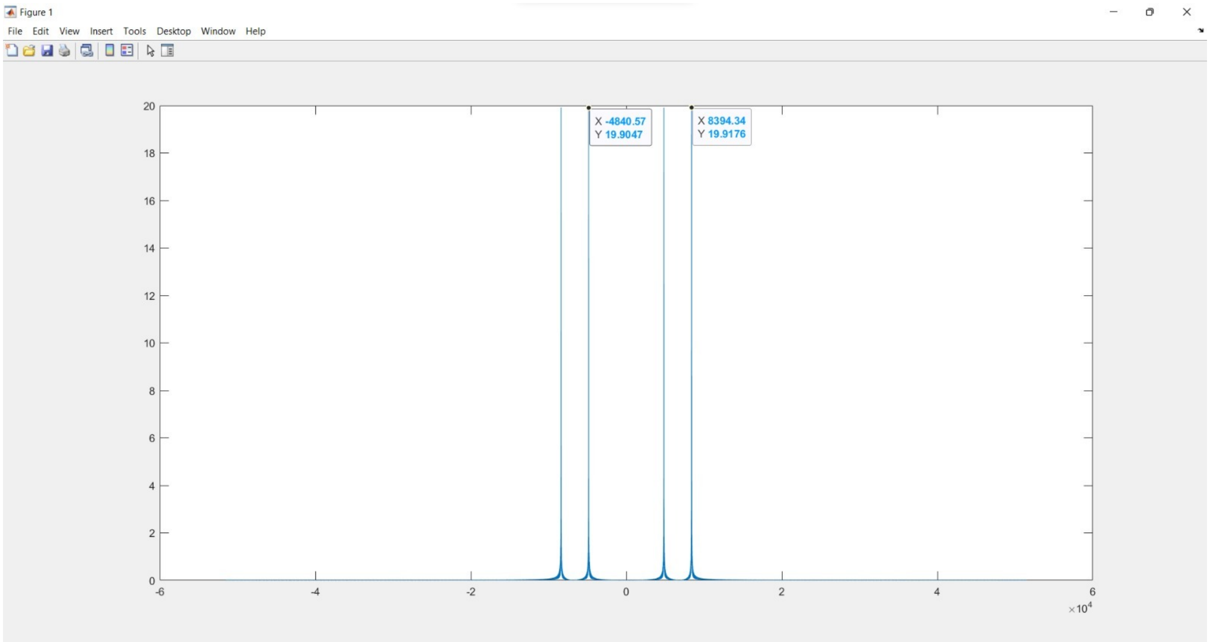

Figure.7. - Index(5)

By using the implementation reazlied in part 1.3, corresponding Figures 3 to 7 are visualized. The indexes are deduced based on the given numbers at the top of line(peak). The DTMF transceivers show the peak based diagrams that are investigated and each diagram matched with these 5 numbers(last 5 digits of my ID). When I clicked to the top of the line(peak point), it almost proves the correctivity of these numbers . In tle lab manual in 1.4, the given equation can be viewed as a product of x(t) by a rectangular signal. Also, the equivalent of the 3rd digit of the number sequence of $x3(t)$ in the DTMF signal is by writing a rectangle in accordance with the value ranges of the given digits, for example, for 'index 3', "rectangle3 = [zeros(1, 16384) ones(1, 0.5*16384) , zeros(1,16384)];" found. The same process was performed for the other digits

Part 1.7

During the implementation of the second method, I realized that this method promises a much more applicable feature to find and detect all digits in each index. On the other hand, using the first method to be able to identify the call when a call is made will not be efficient because it will involve repetitive frequencies. In short, the second method allows us to understand and find the numbers in the index, as well as to find them in the correct order.

2. Echo cancellation

In part 2, the code is the following. I shared my graph that I reached by the speech of my voice with a sentence ***"Hello I am ilke doğan now it is just fourteen past thirty two and I am doing this homework echo cancellation".***

Part 2.0

Code: Whole Code of Question 2

```
clear
clc
[x,Fs] = audioread("ilke.m4a");
x_ = x(1:length(x)/81920:length(x), :);
figure
plot(x) %my normal speech
legend("normal speech")
xlabel("t")
ylabel("normal speech signal")
figure
plot(x_) %my speech with 8192 sampling
legend("speech with 8192 sampling")
xlabel("t")
ylabel("speech with 8192 sampling signal")
A = [0.8 0.6 0.4 0.2 0.05];
ti = [0.5 1 1.5 2 3];
M = 5;
for i=1:M
  x_(i) = 0;
end
y = zeros(81920,2);
delayed_signal = zeros(81920,2);
for t=1:81920/2-1
  for j=1:M
      y(2*t,:) = x_(2*t,:) + A(j)*x_(abs(2*(t - ti(j)))+1,:);
        delayed_signal(2*t,:) = A(j)*x_(abs(2*(t - ti(j)))+1,:);
  end
end
figure
plot(1:1:81920, x_(:,1))
title("x and Delayed Signal")
legend("x(t)")
xlabel("t")
ylabel("x(t)")
hold on
plot(1:1:81920, delayed_signal)
legend("x(t)", "Delayed signal")
figure
plot(1:1:81920, y(:,1))
title("y and Delayed Signal")
hold on
plot(1:1:81920, delayed_signal)
legend("y(t)", "Delayed signal")
xlabel("t")
```

```
ylabel("y(t)")
omega=linspace(-8192*pi,8192*pi,8192*10+1);
omega=omega(1:end-1);

y = y(1: 10 * 8192);
yw = FT(y);

omega = linspace(-8192 * pi, 8192 * pi, 8192 * 10 + 1);
omega = omega(1:end - 1);

% H is computed
H = 1;
for i = 1:M
    H = H + A(i) * exp(-1i*omega*ti(i));
end
h = IFT(H);
figure(4);
tx = 0:1/8192:10-1/8192;
plot(tx,h), title("h(t) vs t");
ylabel("h(t)");
xlabel("t");

plot(omega, abs(H));
ylabel("| H(w) |"); xlabel("omega");

Xe = yw./H;
xe = IFT(Xe);
plot(tx, xe);
xlabel("t"); ylabel("Xe(t)");
```

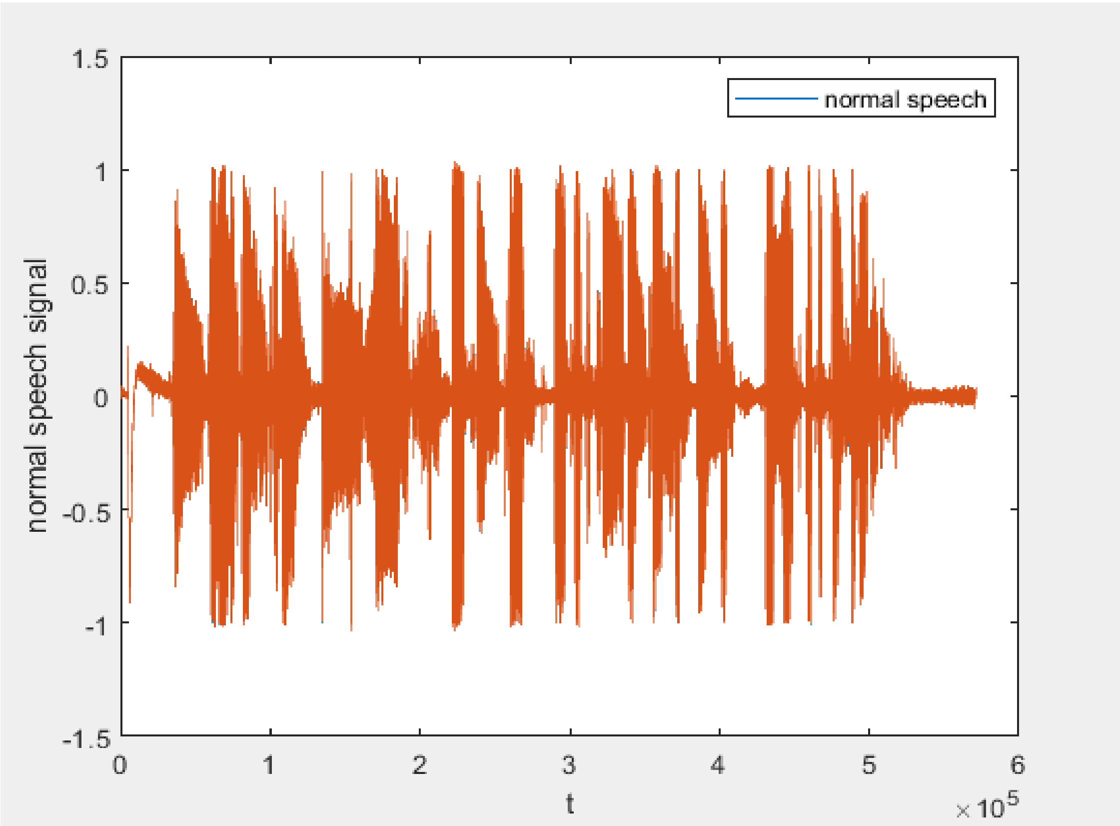

Figure - Normal Speech Signal

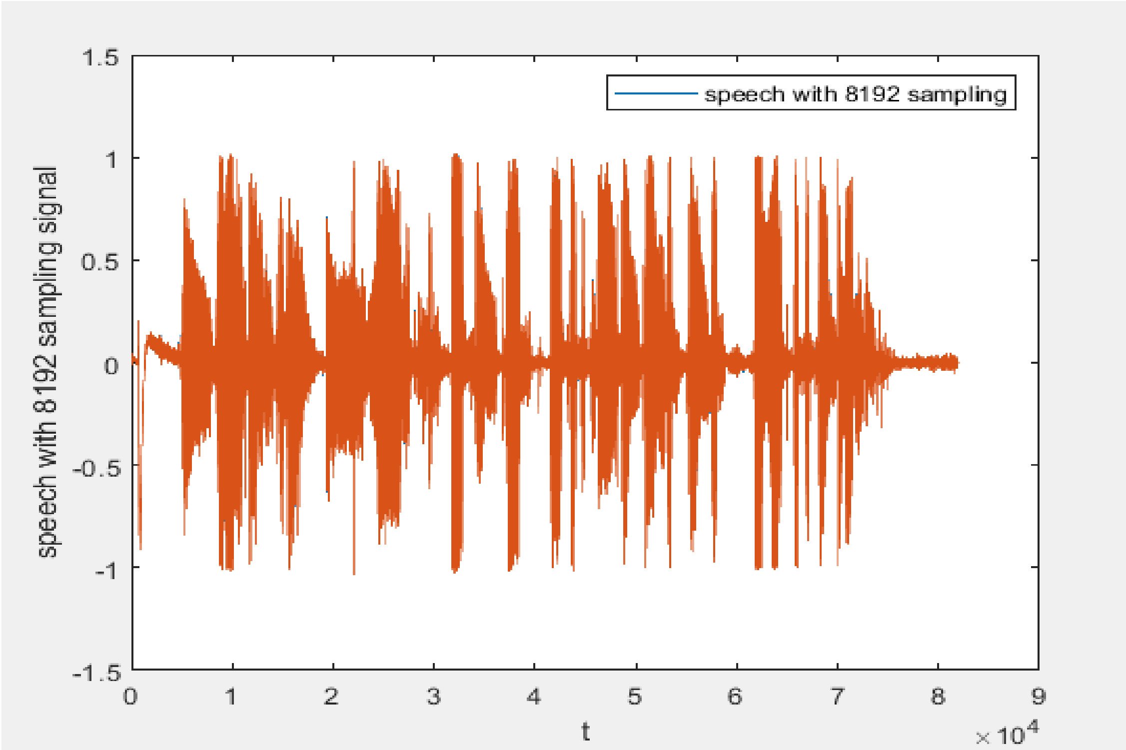

Figure - Speech with 8192 Sampling

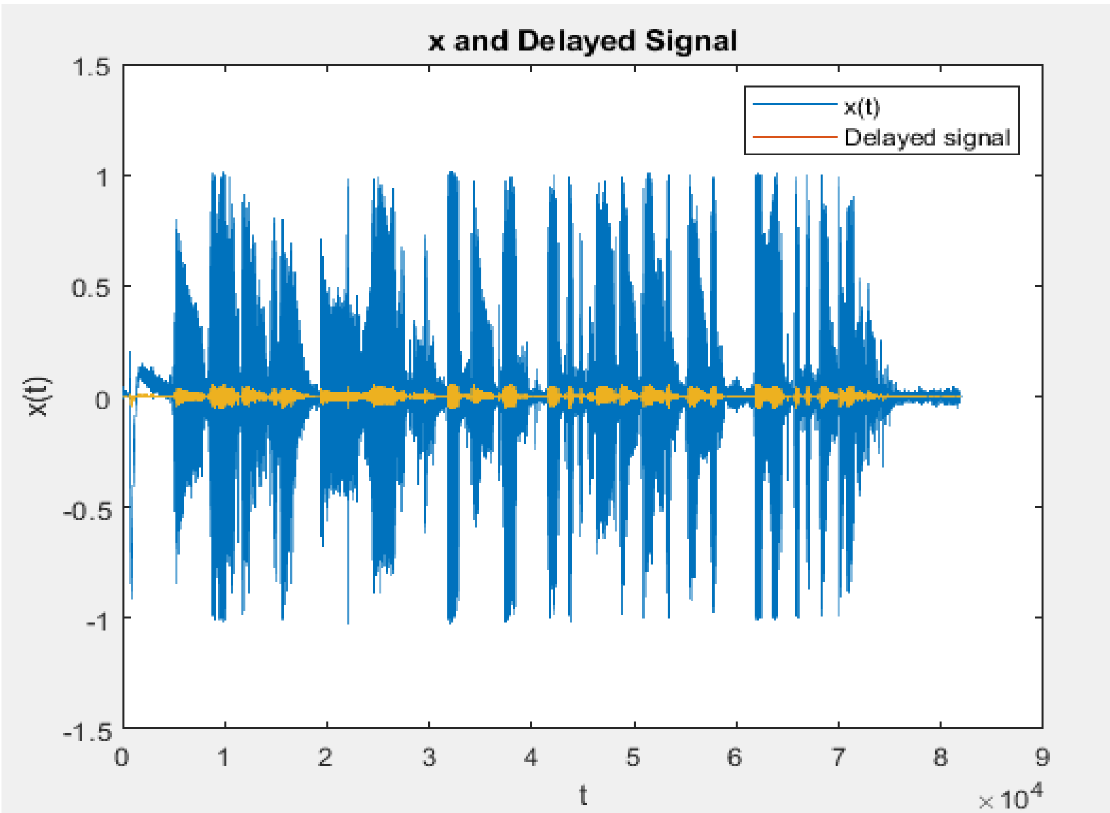

Figure - x and Delayed Signal

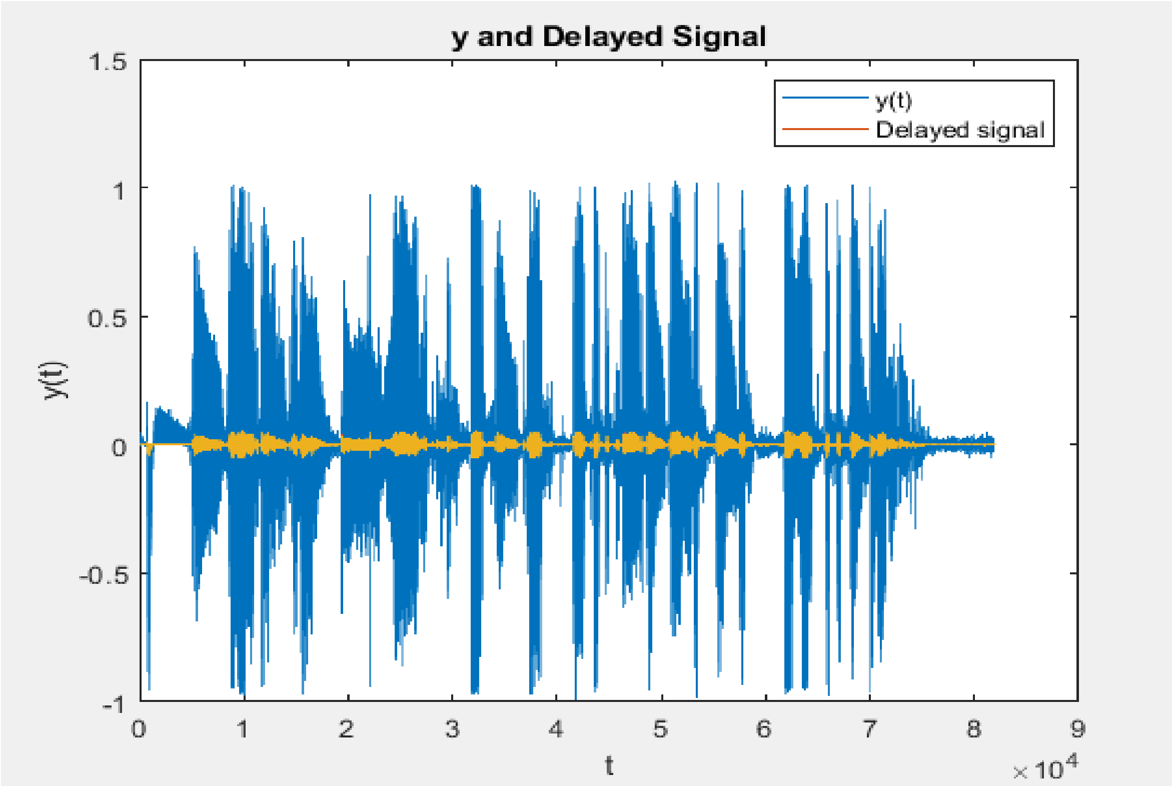

Figure - y and Delayed Signal

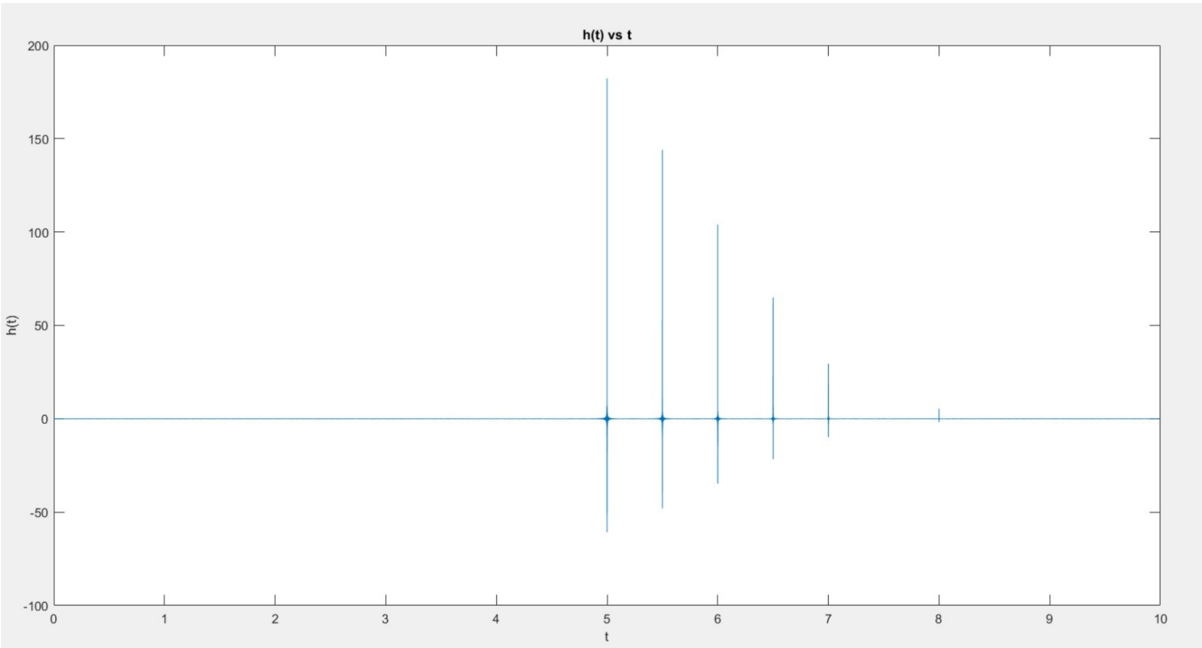

Figure -h(t) versus t in large scope

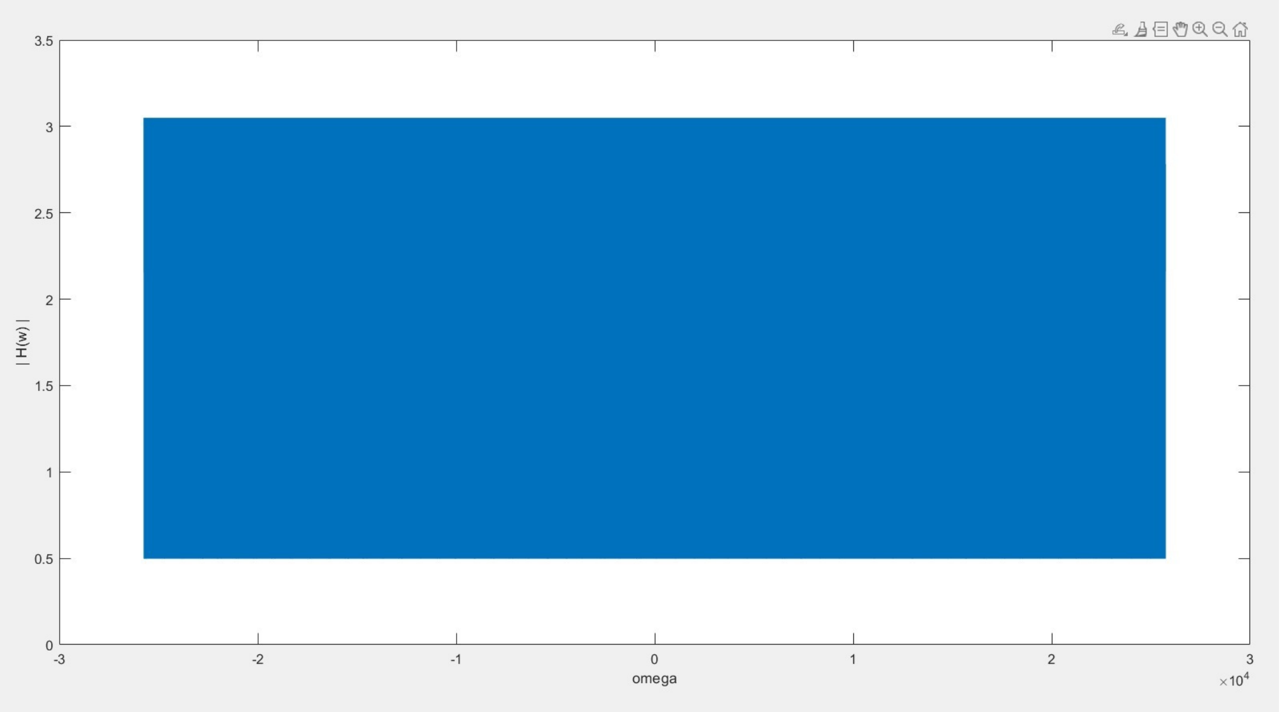

Figure -|H(w)| versus Omega

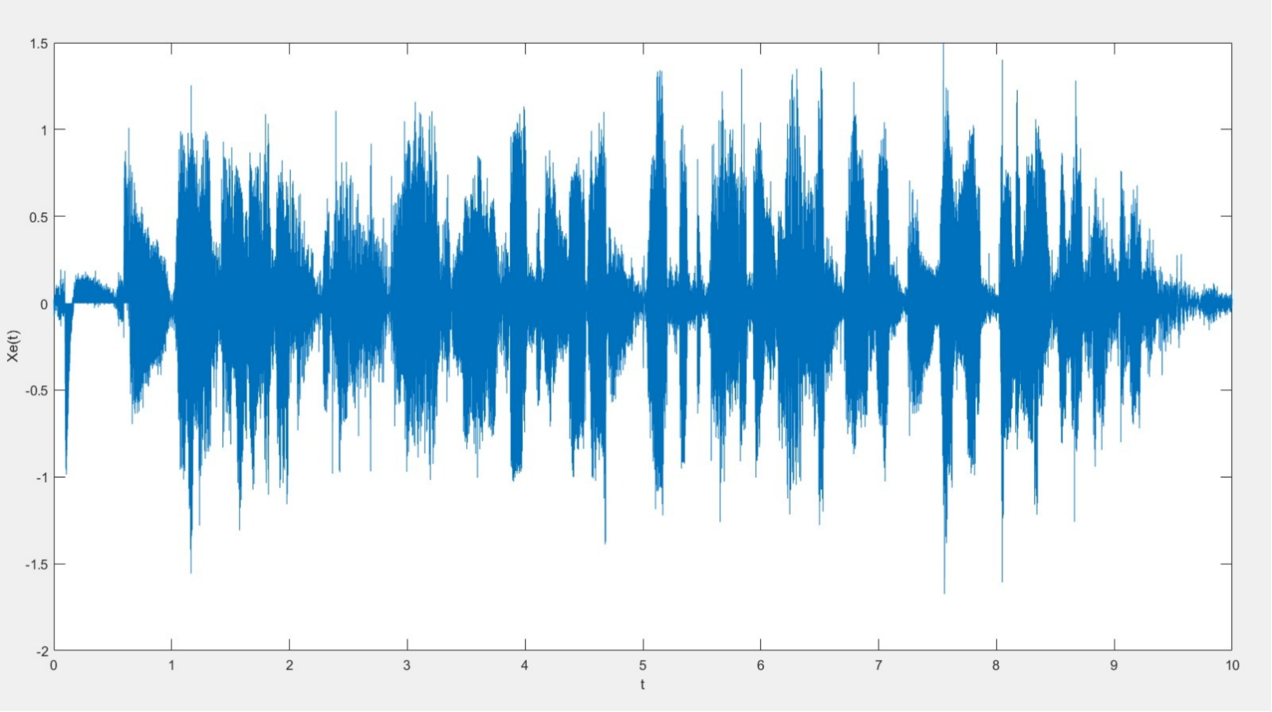

Figure - X_e(t) versus t

- ***I heard Echos for a while during my own talk and these echoes came with a delay and actually maybe because of that their amplitudes were approximately different.***

At the time, I listened to the X_e(t) that is shown in the graphs given above. Nonetheless, while I was listening, because of the echo,I heard several differences in the sound that I am testing, but it was almost the same as the original sound. When I started listening, I heard most of my speech in the sentence I gave the transcript above, but then I noticed the repercussions. Actually, the reason behind this is that they didn't cause me to hear the original signals, as their amplitudes were relatively low. But they are actually the reason why the estimated signal is different from the original signal. Finally, when I was not speaking, I saw on the graph that the estimated signal was not zero, so I thought it was an echo effect again.