Second project report

Name	ID
Mostafa Mohamed Abdel-Azeem Hassanen	20011950
Mazen Mohamed Hassanen	20011161
Mohamed Ashraf Elsayed Mahmoud	20011488
Ahmed Abdel-Hakem Abdel-Salam Ali	20010124

Section:	4

Department:	Electronics and Communication

The code

```
clear all;
clc;
%1)Transmitter

%Playing Sound or Music
Input=input('Enter the number the you want to play : \n 1)Sound  \n 2)Music \n');
if Input==1
   [x,f_s] = audioread('85._45.mp3');
elseif Input==2
   [x,f_s] = audioread('gamed.mp3');
end
%Play the sound.
sound(x,f_s);

%Time representation.
N = length(x);  %Length of x.
t=linspace(0,N/f_s,N); %Time.
subplot(3,1,1)
plot(t,x); %Origional sound in time.
xlabel('Time');
ylabel('sound');
title('Signal in Time representation')

%Frequency Representation.
X=fftshift(fft(x));
X_magnitude= abs(X);
X_phase = angle(X);
fvec=linspace(-f_s/2,f_s/2,length(X));
subplot(3,1,2)
plot(fvec,X_magnitude); %Origional sound in frequency.
xlabel('Frequency');
ylabel('sound');
title('Signal Frequency magnitude representation');
subplot(3,1,3)
plot(fvec,X_phase); %Origional sound in phase.
xlabel('Frequency ');
ylabel('sound');
title('Signal Frequency angle representation');

status = "stop";
forcestop = "mkmkm ";
while ~(strcmp(status,forcestop))
    forcestop = input('Type stop to stop the sound : ','s');
end
```

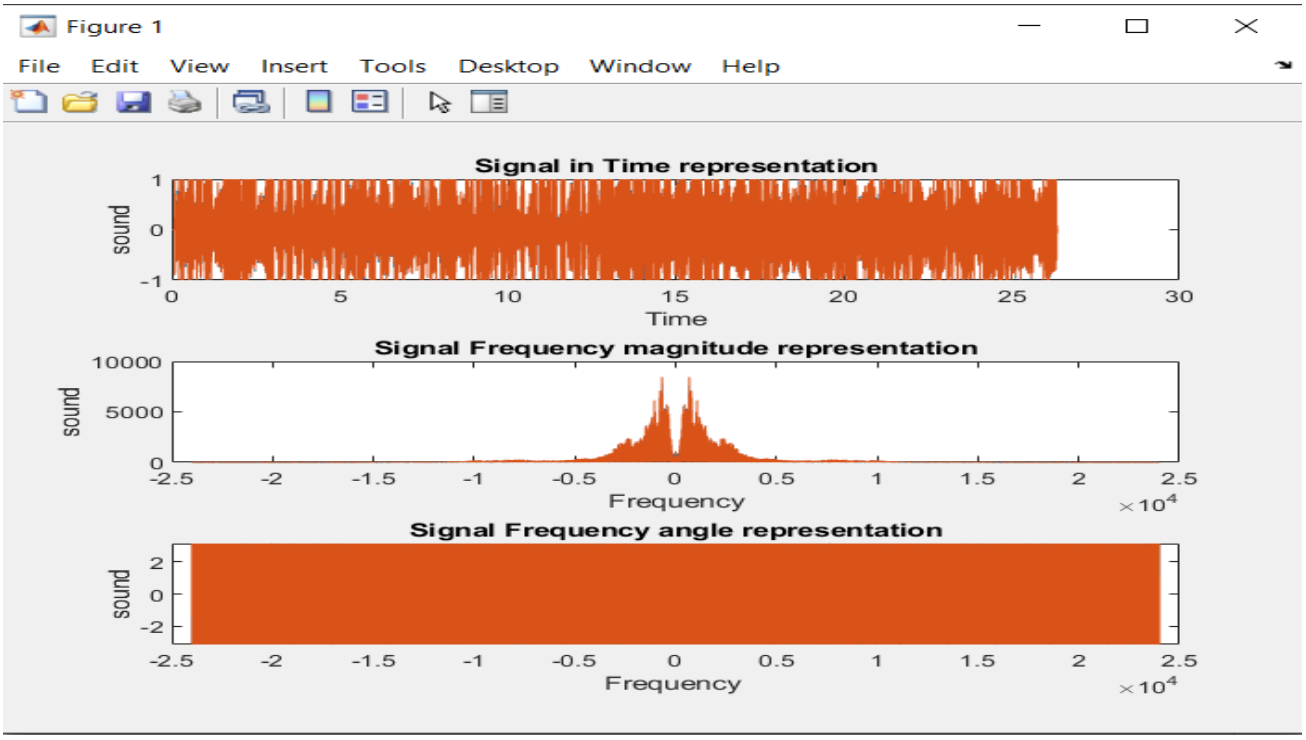
```
clear sound;%Stop the sound (Can be replaced by typing this in command window to stop
%it whenever you want).
%*************************************************************************************************************************************************
%2)Channels

% Define the impulse responses
h1 = [1 zeros(1, N-1)]; % Delta function
h2 = exp(-2*pi*5000*t); % exp(-2pi*5000t)
h3 = exp(-2*pi*1000*t); % exp(-2pi*1000t)
h4 = zeros(size(t));
h4(t == 0) = 2;
h4(t == 1) = 0.5;
%Implementation of the Channels
figure;
subplot(2,2,1);
plot(h1);
xlabel('time');
ylabel('channel 1');
subplot(2,2,2);
plot(h2);
xlabel('time');
ylabel('channel 2');
subplot(2,2,3);
plot(h3);
xlabel('time');
ylabel('channel 3');
subplot(2,2,4);
plot(h4);
xlabel('time');
ylabel('channel 4');

%Taking the channel to be performed on the signal
channels=input('Enter the number of the channel you want to perform on the signal : \n 1)Delta function  \n 2)exp(-2pi*5000t) \n 3)exp(-2pi*1000t)\n 4)impulse response \n');

if channels==1
    y = conv(x(:)', h1(:));
elseif channels==2
    y = conv(x(:)', h2(:));
elseif channels==3
    y = conv(x(:)', h3(:));
elseif channels==4
    y = conv(x(:)', h4(:));
end

```

```
% Plot the result
figure;
subplot(2,1,1);
plot(t, x);
title('Original signal');
xlabel('Time (s)');
ylabel('Amplitude');

t_conv=linspace(0,length(y)/f_s,length(y));
subplot(2,1,2);
plot(t_conv, y);
title('Convolved signal');
xlabel('Time (s)');
ylabel('Amplitude');
%*********************************************************************************
%3)Adding Noise.

%Taking Sigma.
Sigma=input('Enter the sigama (Noise) to be introduced to the channel: ');
%Introduce noise( Gaussian Distribution noise with zero mean and standard
%deviation = sigma ).
Noise = Sigma * randn(size(y));
%Nosied signal.
y= y + Noise;
%Play the sound after adding noise.
sound(y,f_s);

%Plot the noised signal in time domain.
New_N = length(y);  %Length of y
New_t=linspace(0,New_N/f_s,New_N); %Time
figure;
subplot(3,1,1)
plot(New_t,y);
xlabel('Time');
ylabel('Noised sound');
title('Time representation of Noised signal.')

%Plot the noised signal in Frequency domain.
Noised=fftshift(fft(y));
Noised_magnitude= abs(Noised);
Noised_phase = angle(Noised);
NoisedFreqVec=linspace(-f_s/2,f_s/2,length(Noised));
subplot(3,1,2)
plot(NoisedFreqVec,Noised_magnitude);
```

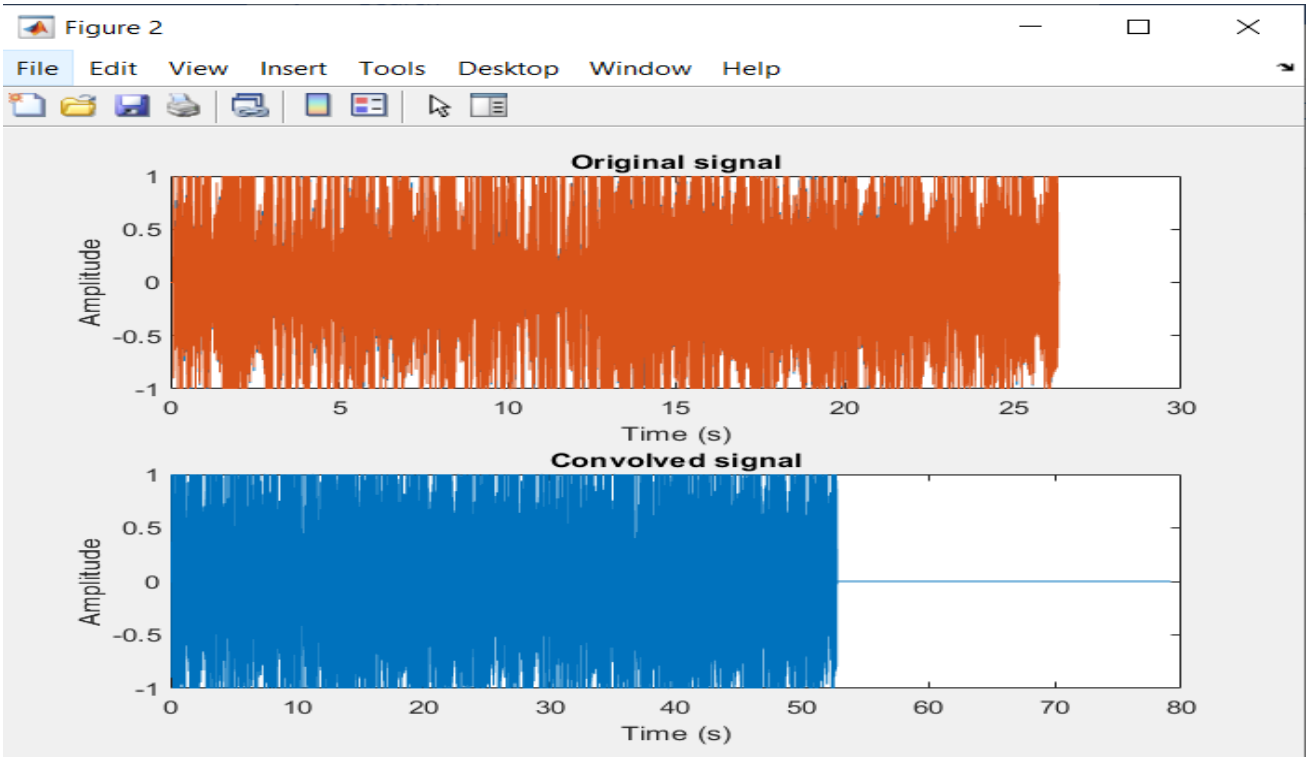
```
xlabel('Frequency');
ylabel('Noised Signal');
title('Frequency magnitude of Noised signal.');
subplot(3,1,3)
plot(NoisedFreqVec,Noised_phase); %Origional sound in phase.
xlabel('Frequency ');
ylabel('sound');
title('Frequency angle of Noised signal.');

status2 = "stop";
forcestop2 = "anything other than the word play";
while ~(strcmp(status2,forcestop2))
    forcestop2 = input('Type stop to stop the sound : ','s');
end
clear sound;%Stop the sound (Can be replaced by typing this in command window to stop
%it whenever you want).

%*************************************************************************************************
%RECIEVER
samplePerHz = New_N/f_s;

freqDiff = f_s/2 - 3400;
samplesFiltered1 = uint32(samplePerHz * freqDiff);
samplesFiltered2 = uint32(length(Noised) - samplesFiltered1 + 1);
Noised([1:samplesFiltered1 samplesFiltered2:end])=0;
NoisedMagAfterFilter=abs(Noised);
NoisedAfterFilterTime=ifft(ifftshift(Noised));
sound(NoisedAfterFilterTime,f_s);
figure;
subplot(2,1,1);
plot(NoisedFreqVec,NoisedMagAfterFilter);
title('filtered signal in frequency domain');
subplot(2,1,2);
plot(New_t,NoisedAfterFilterTime);
title('filtered signal in time domain');

status3 = "stop";
forcestop3 = "anything other than the word play";
while ~(strcmp(status3,forcestop3))
    forcestop3 = input('Type stop to stop the sound : ','s');
end
clear sound;%Stop the sound (Can be replaced by typing this in command window to stop
%it whenever you want).
```

First: the sound file

Output:



First Channel:



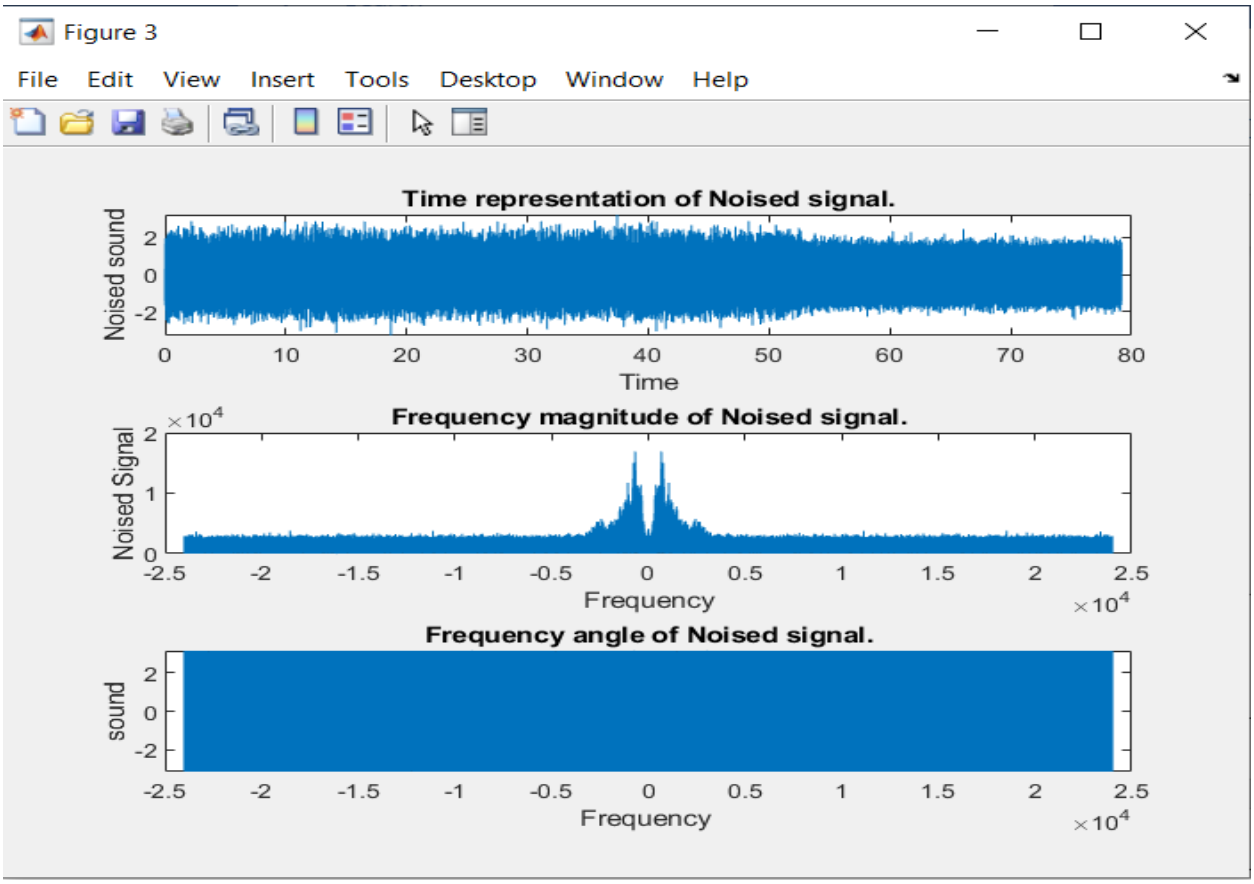


Figure 3
File Edit View Insert Tools Desktop Window Help
Time representation of Noised signal.
Noised sound
Time
Frequency magnitude of Noised signal.
Noised Signal
Frequency
Frequency angle of Noised signal.
sound
Frequency

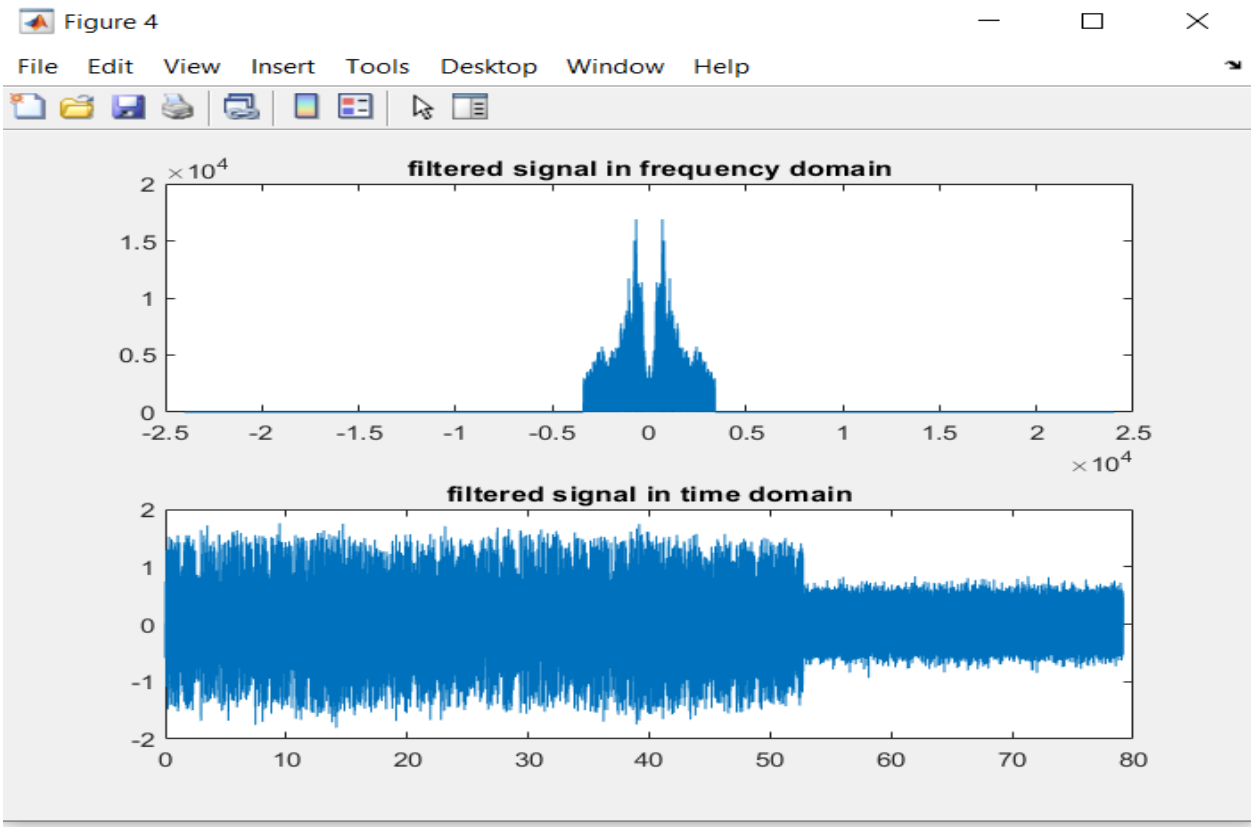
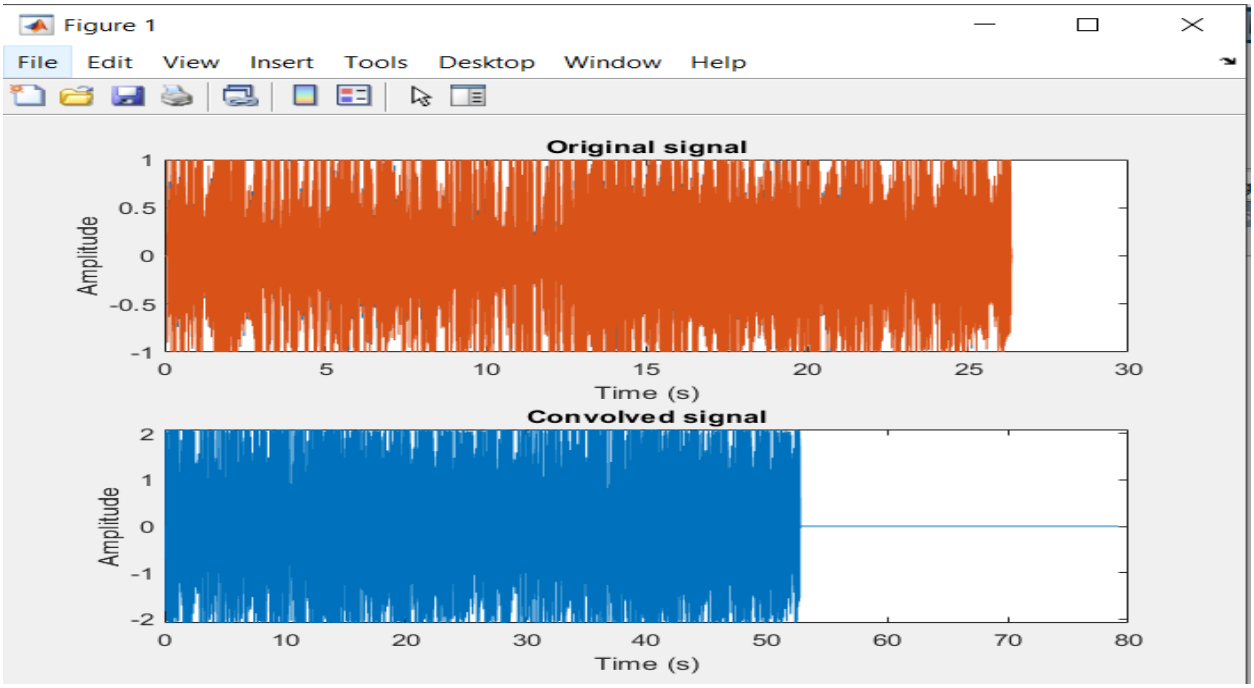
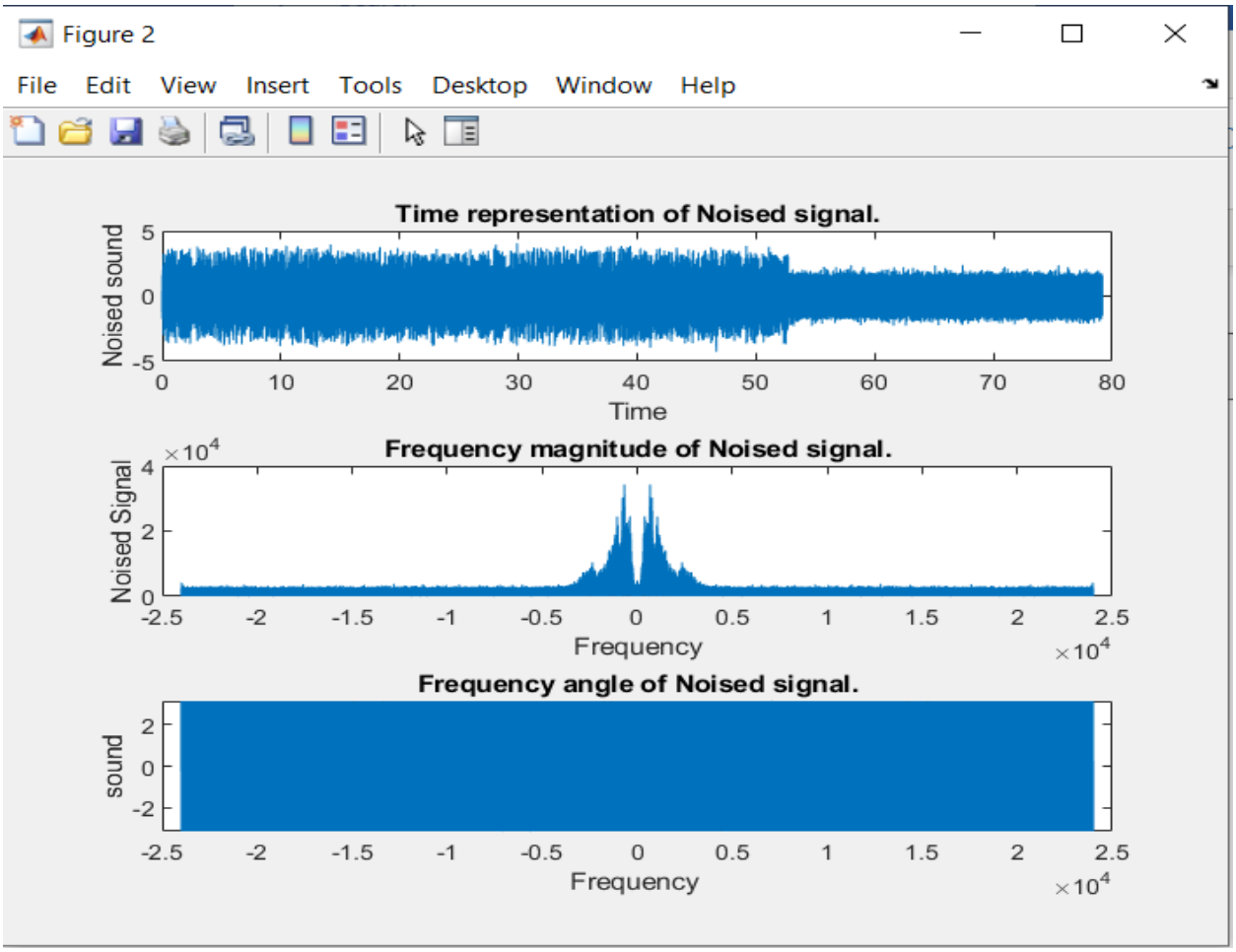
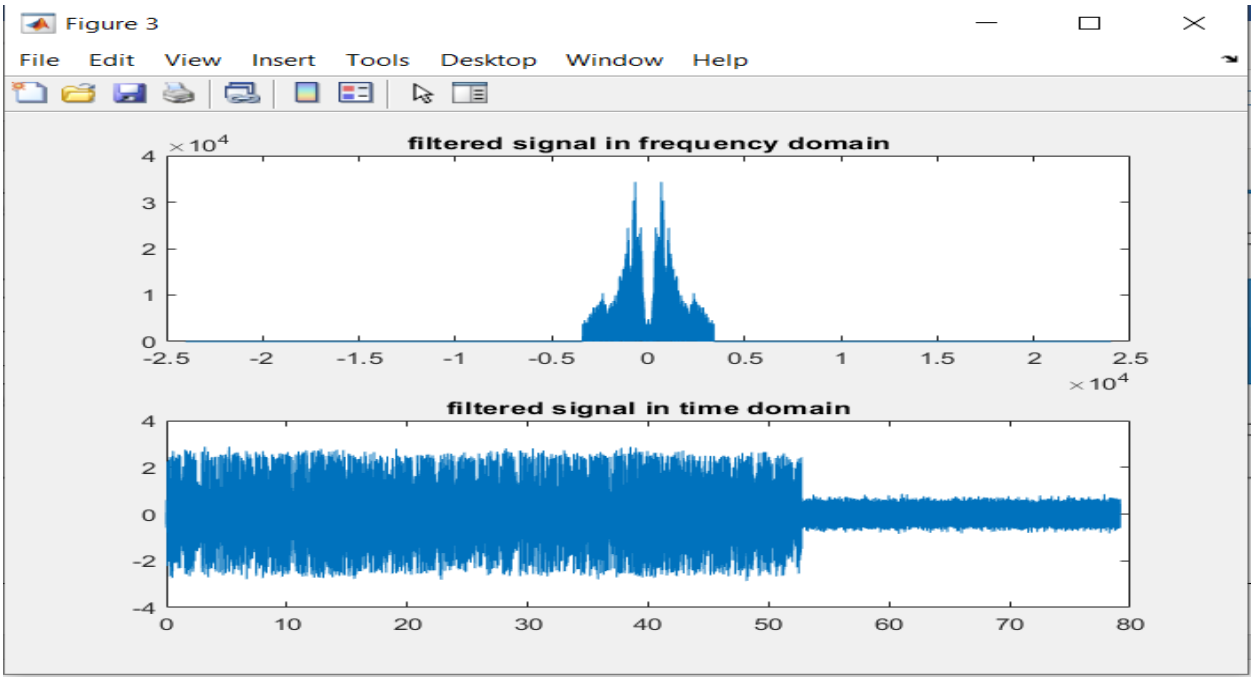


Figure 4
File Edit View Insert Tools Desktop Window Help
filtered signal in frequency domain
filtered signal in time domain

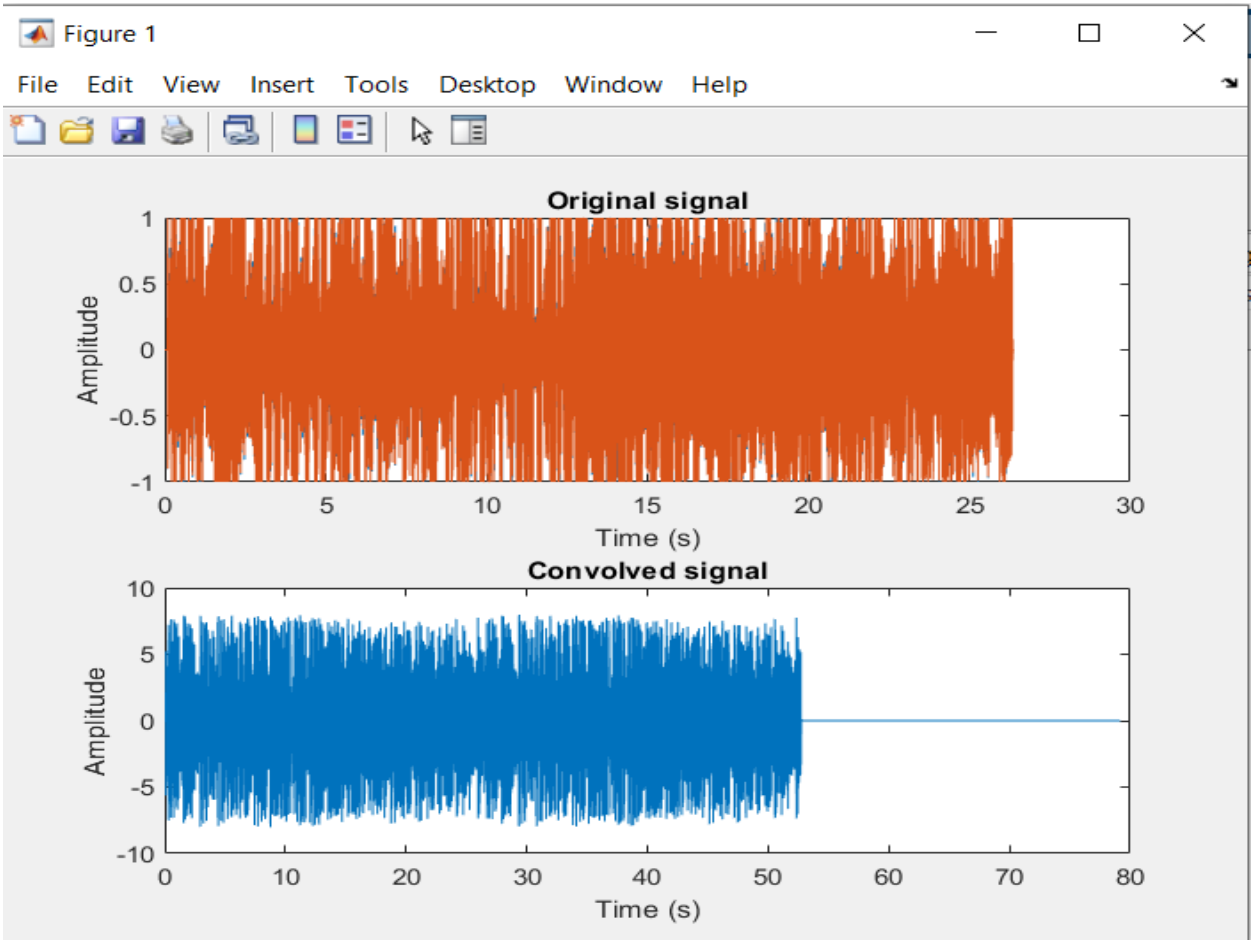
Second Channel:







Third Channel:



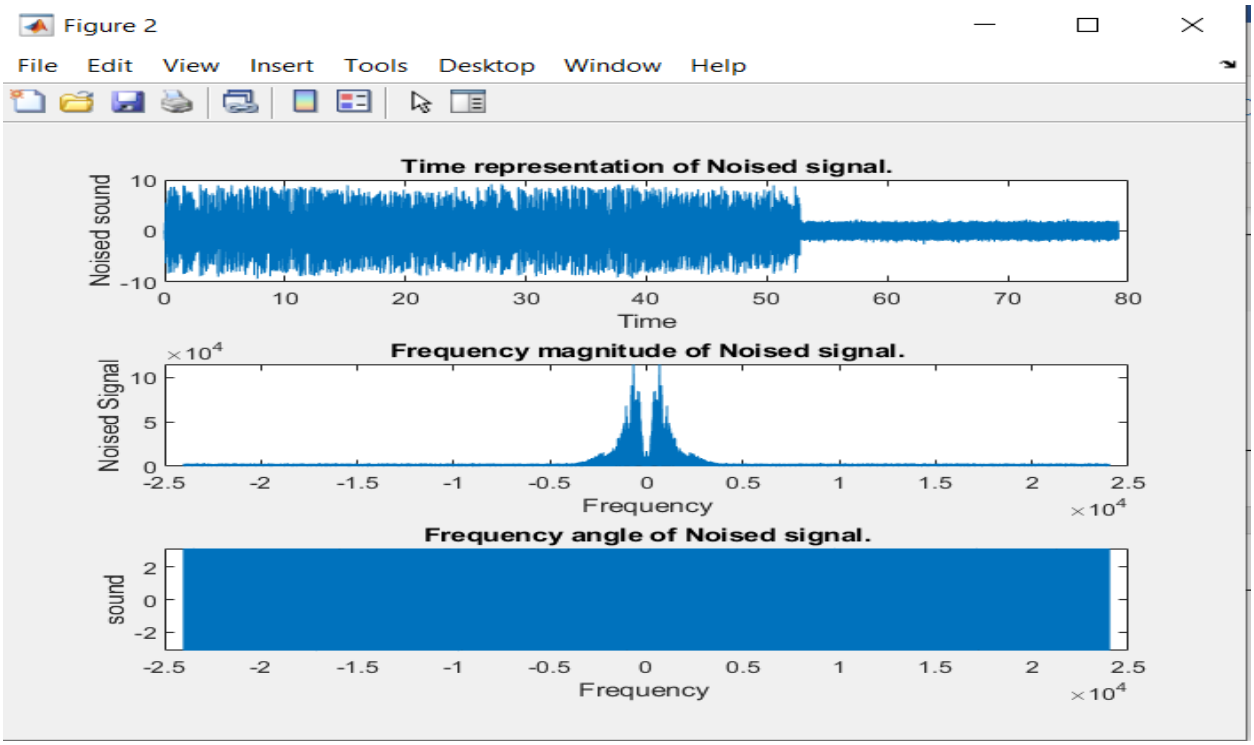


Figure 2
File Edit View Insert Tools Desktop Window Help
Time representation of Noised signal.
Noised sound
Time
Frequency magnitude of Noised signal.
Noised Signal
Frequency
Frequency angle of Noised signal.
sound
Frequency

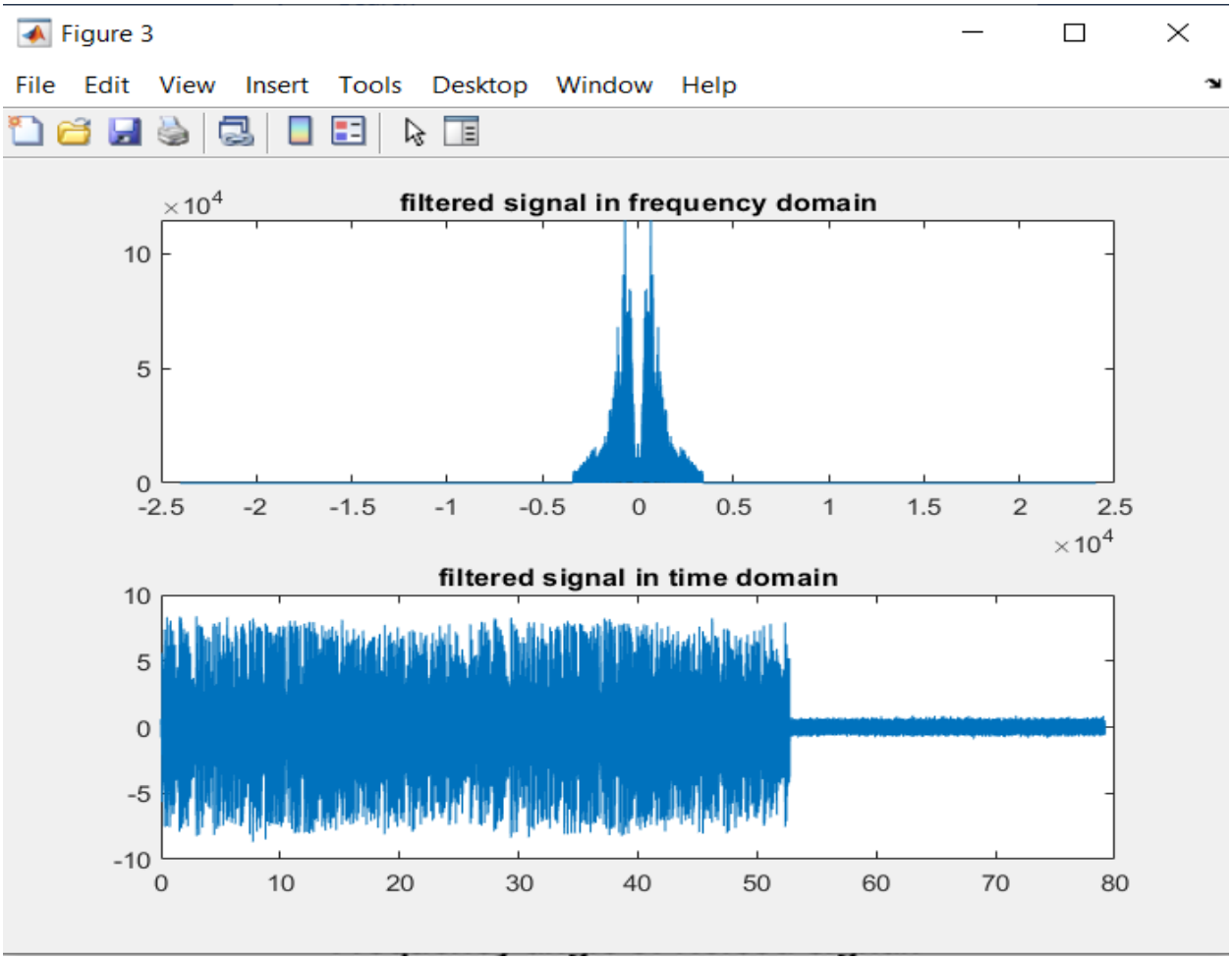
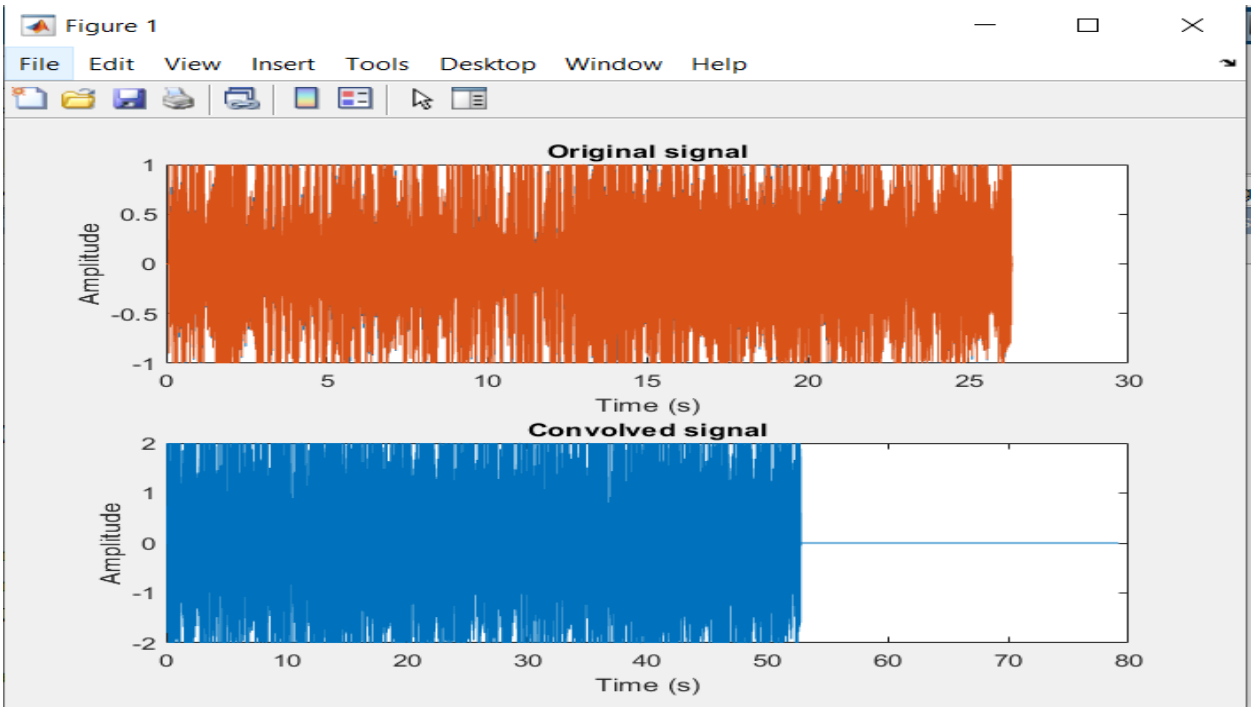
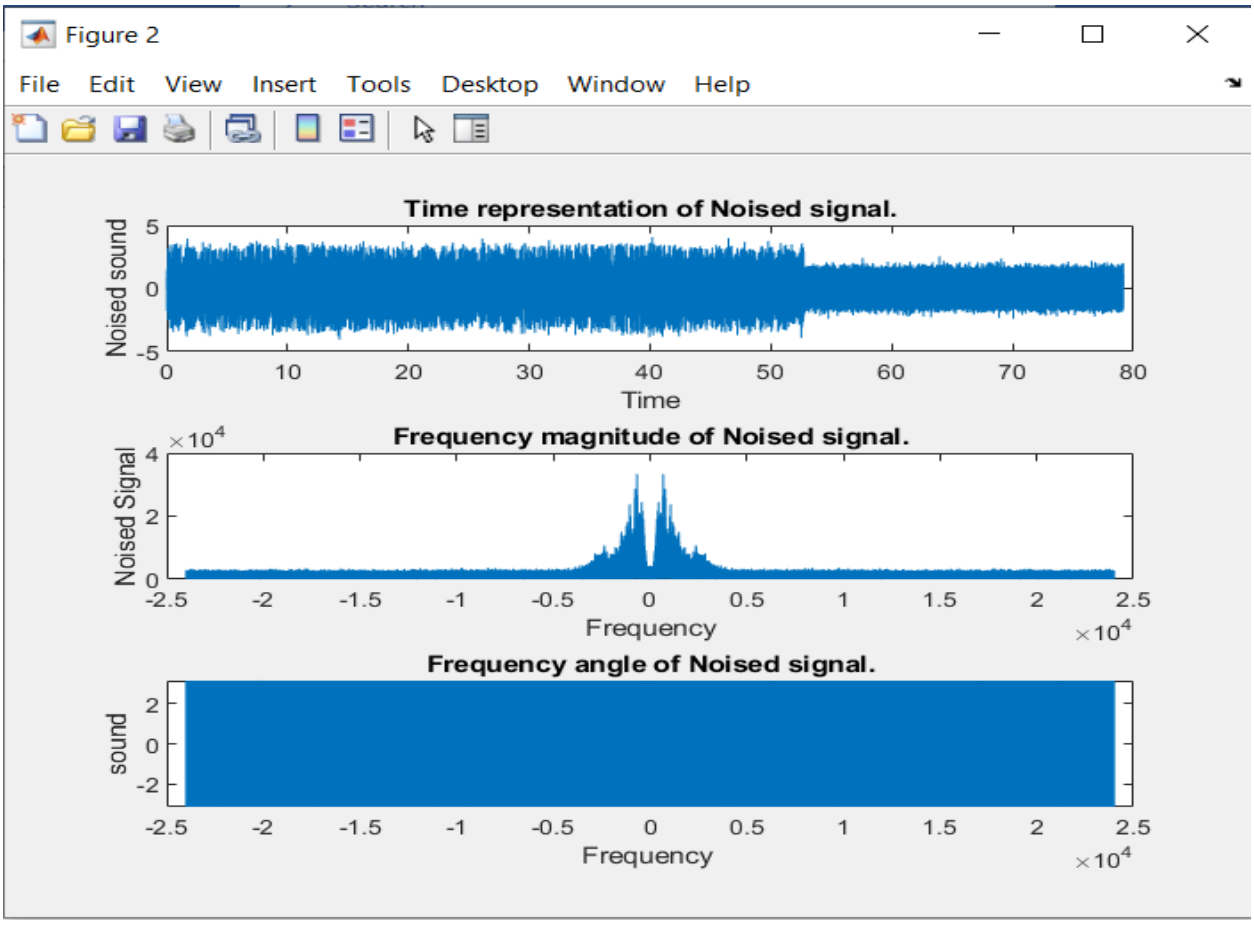


Figure 3
File Edit View Insert Tools Desktop Window Help
filtered signal in frequency domain
filtered signal in time domain

Fourth Channel:





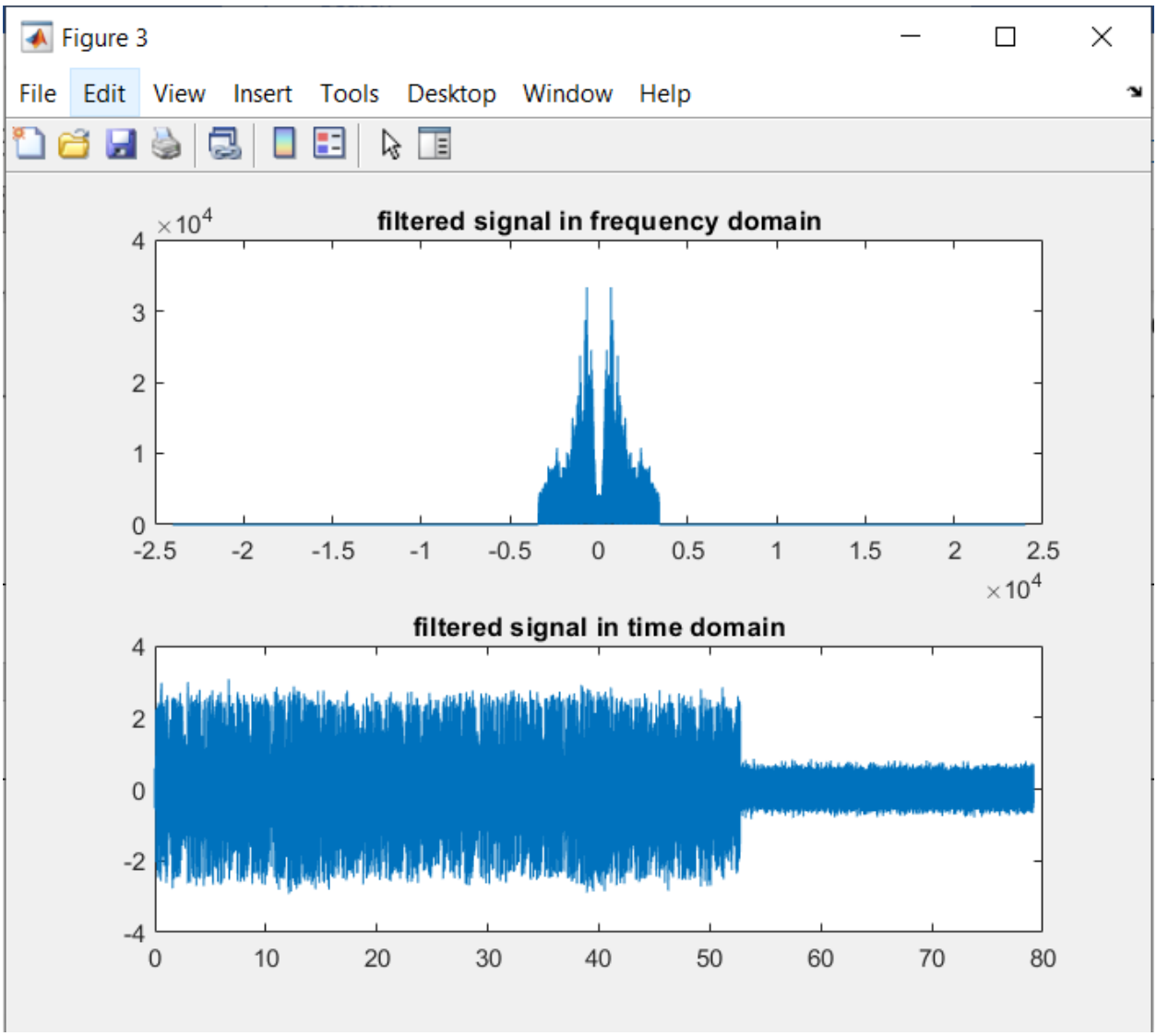
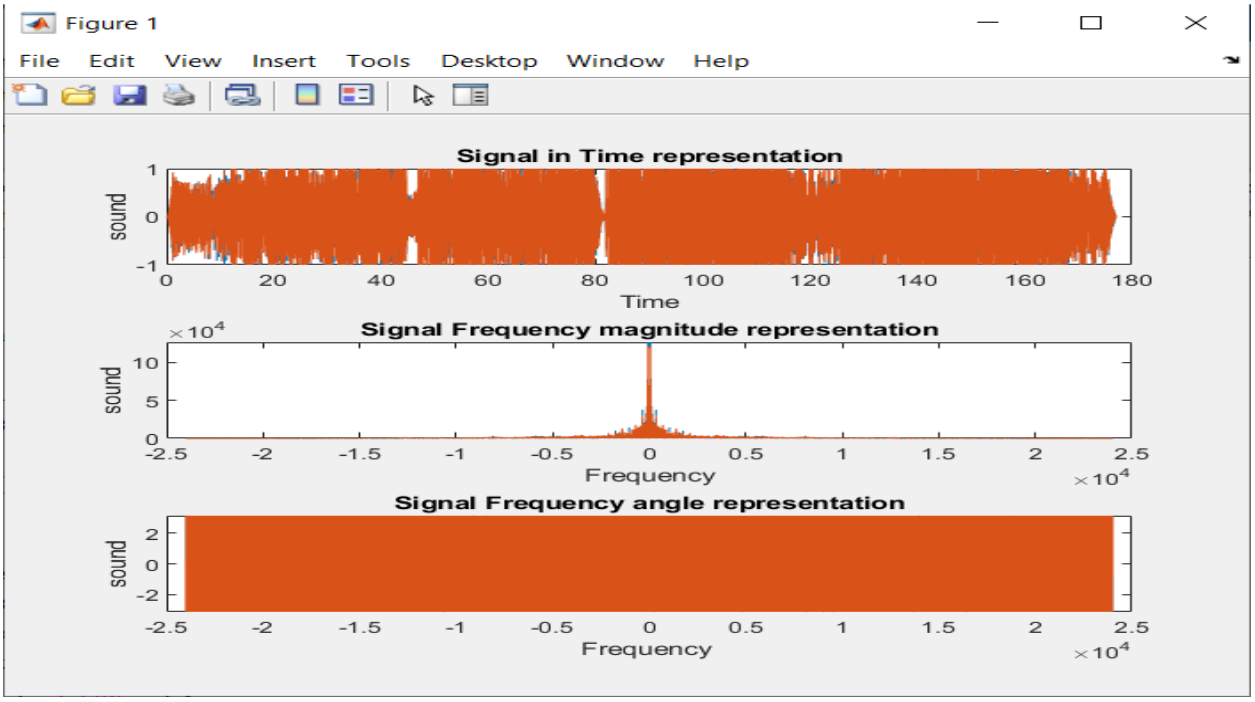


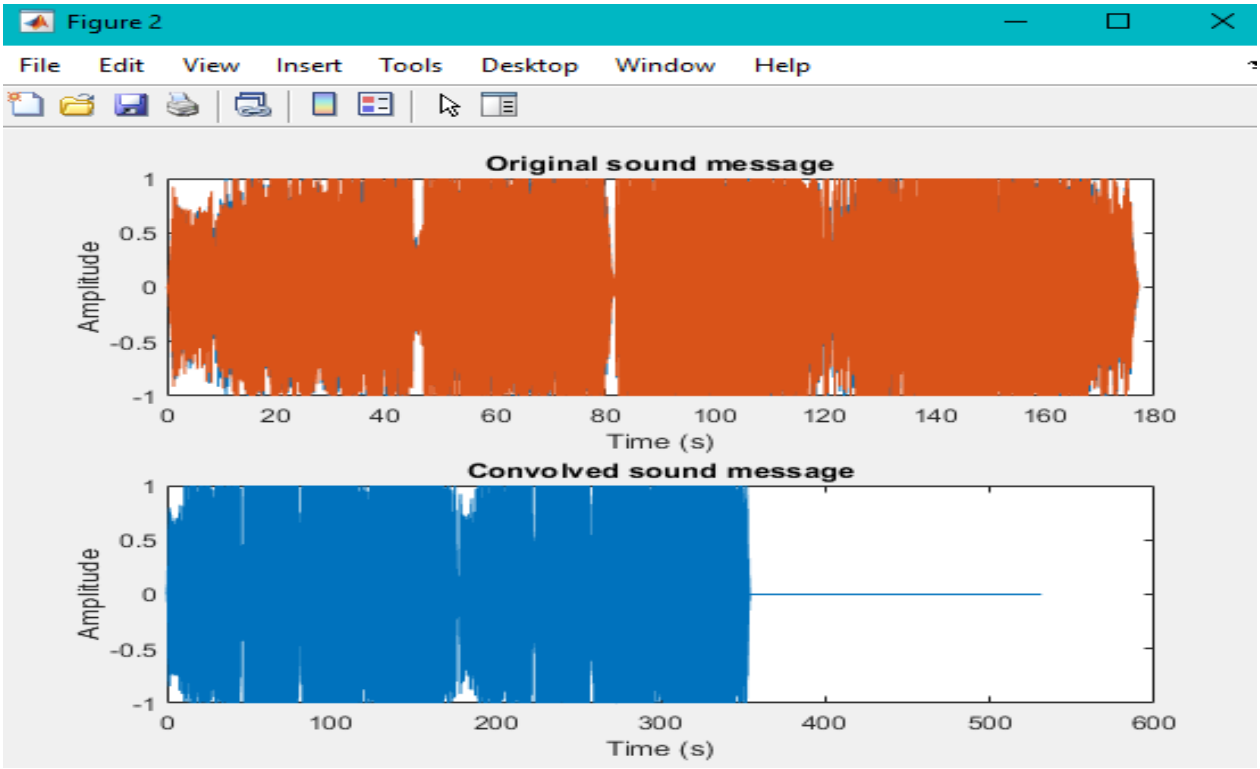
Figure 3
File Edit View Insert Tools Desktop Window Help
filtered signal in frequency domain
×10^4
4
3
2
1
0
-2.5 -2 -1.5 -1 -0.5 0 0.5 1 1.5 2 2.5
×10^4
filtered signal in time domain
4
2
0
-2
-4
0 10 20 30 40 50 60 70 80

second: the music file

Output:



First Channel:



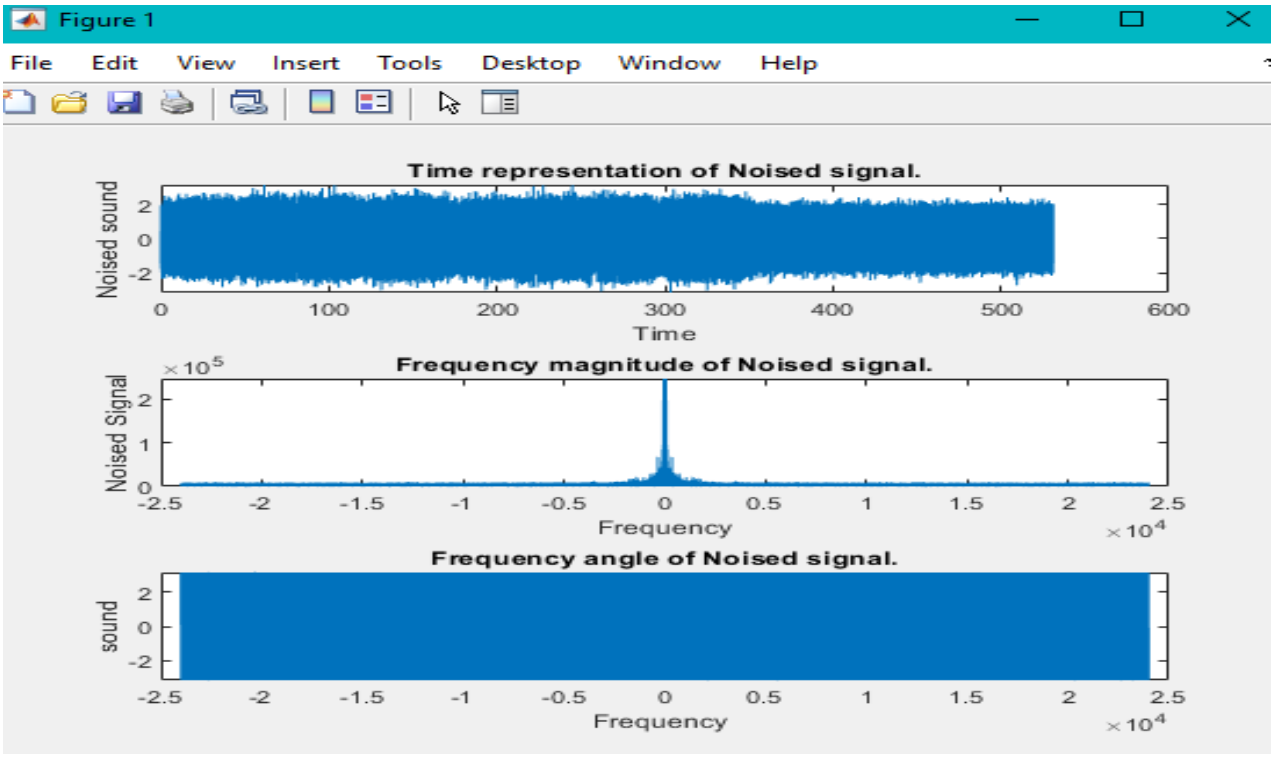


Figure 1
File Edit View Insert Tools Desktop Window Help
Time representation of Noised signal.
Noised sound
Time
Frequency magnitude of Noised signal.
Noised Signal
Frequency
Frequency angle of Noised signal.
sound
Frequency

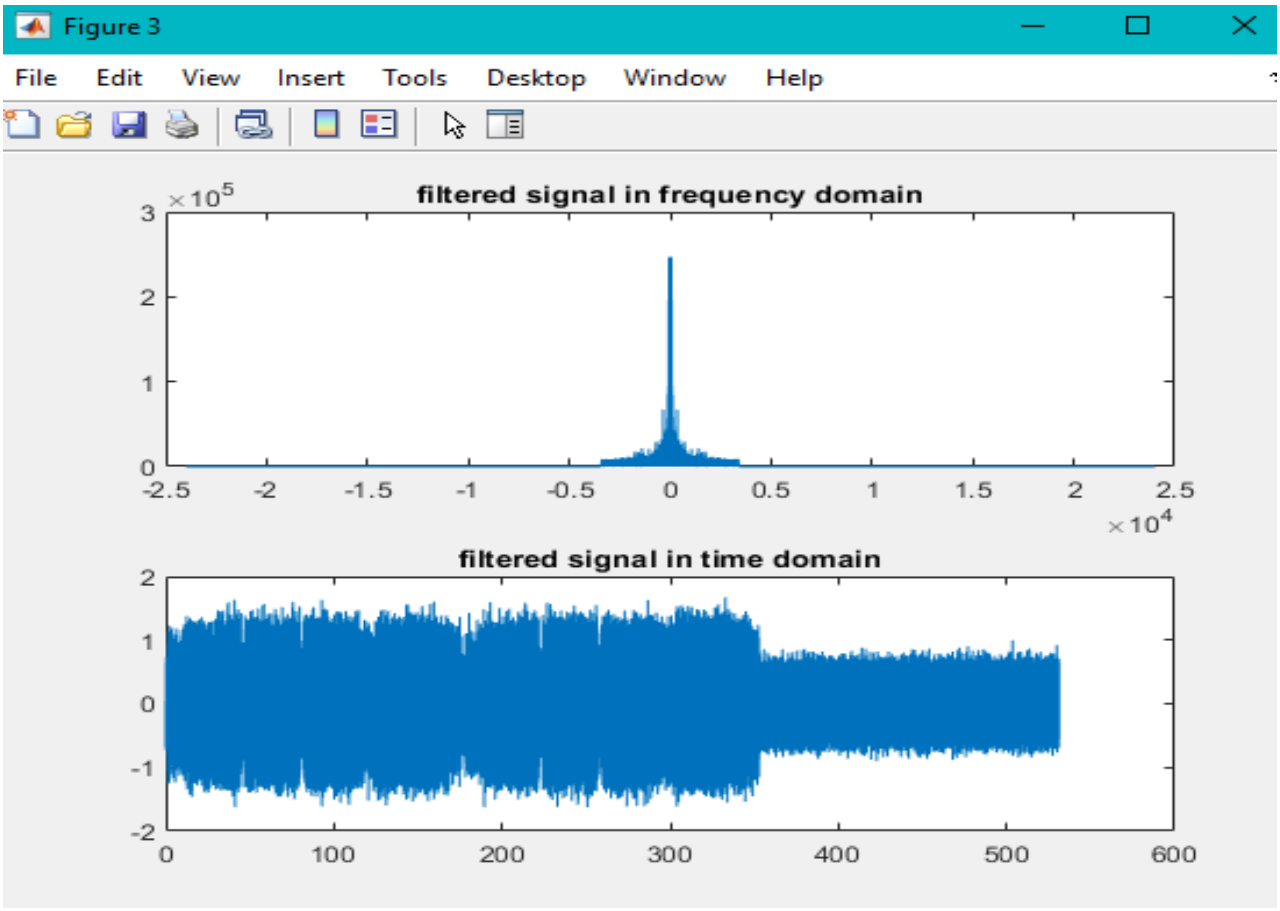
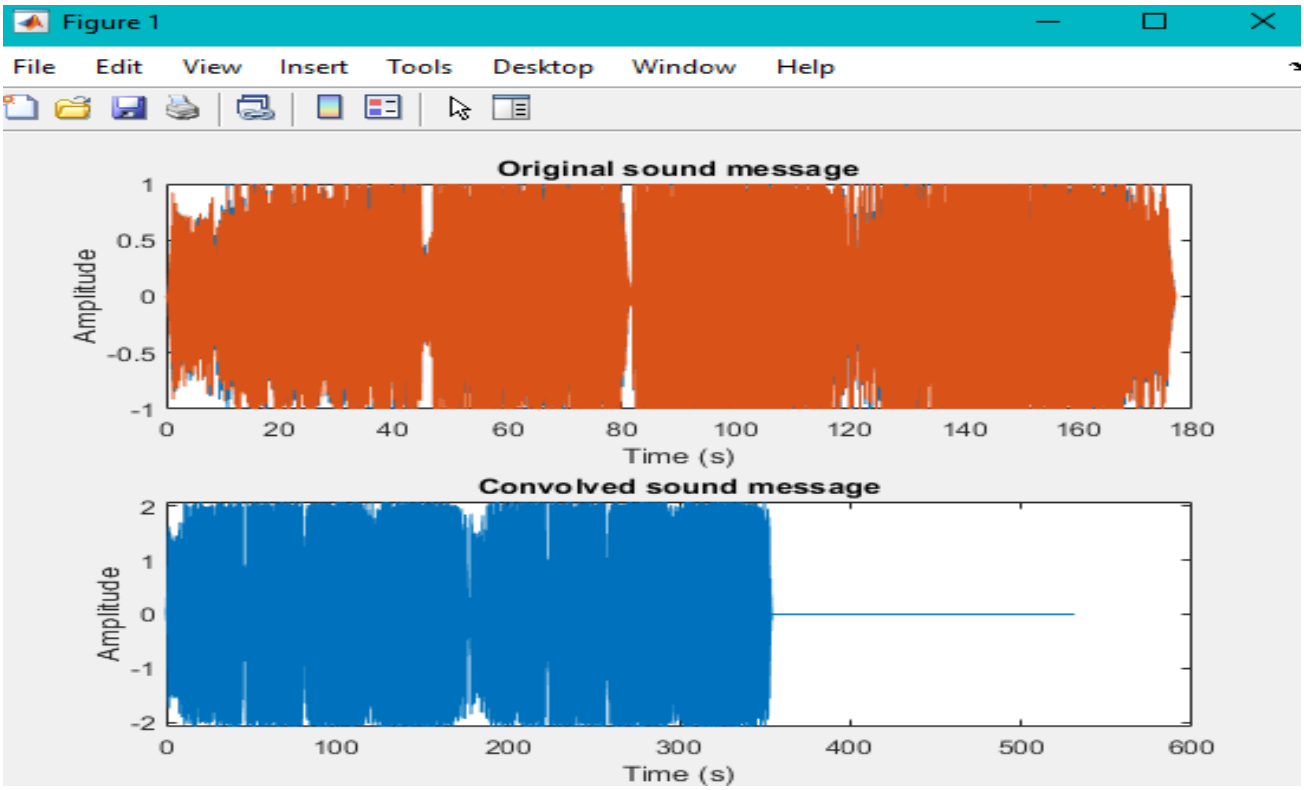
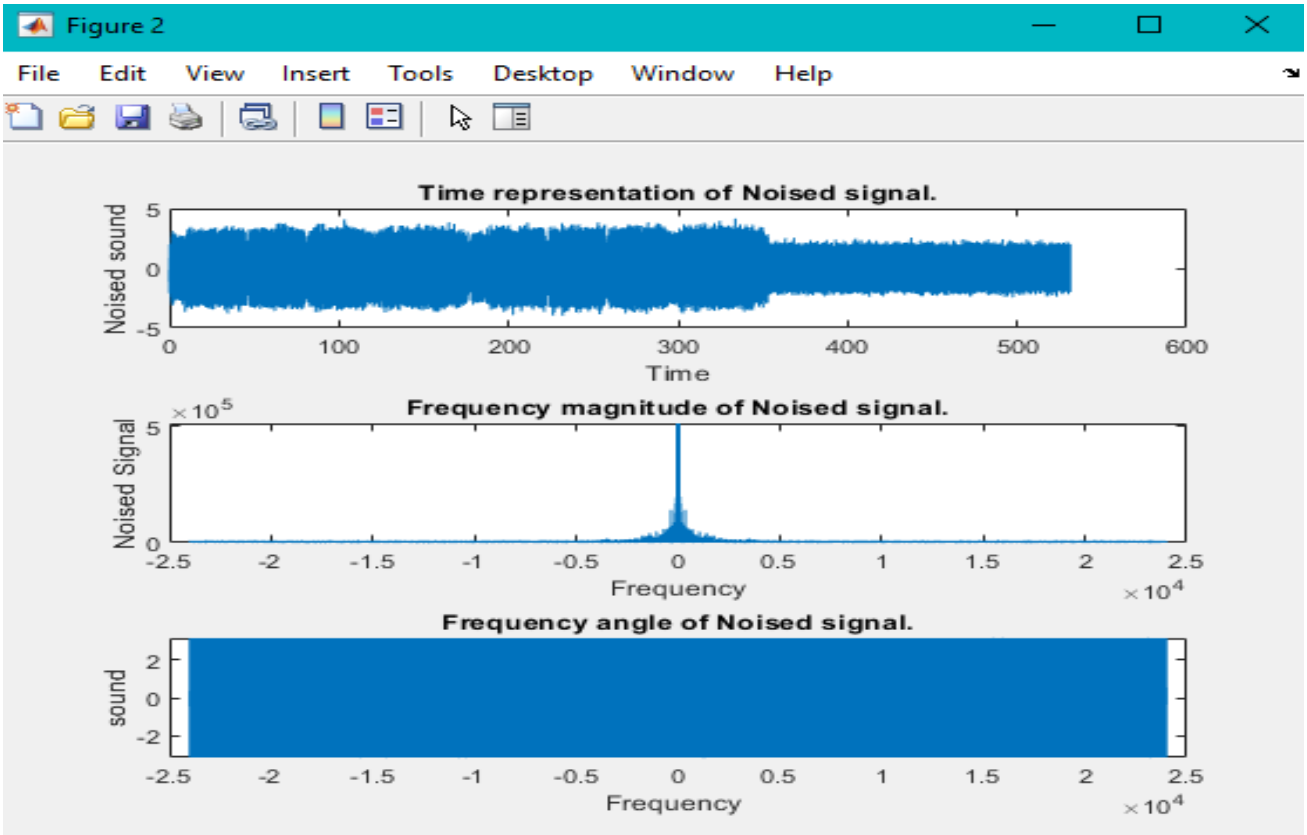
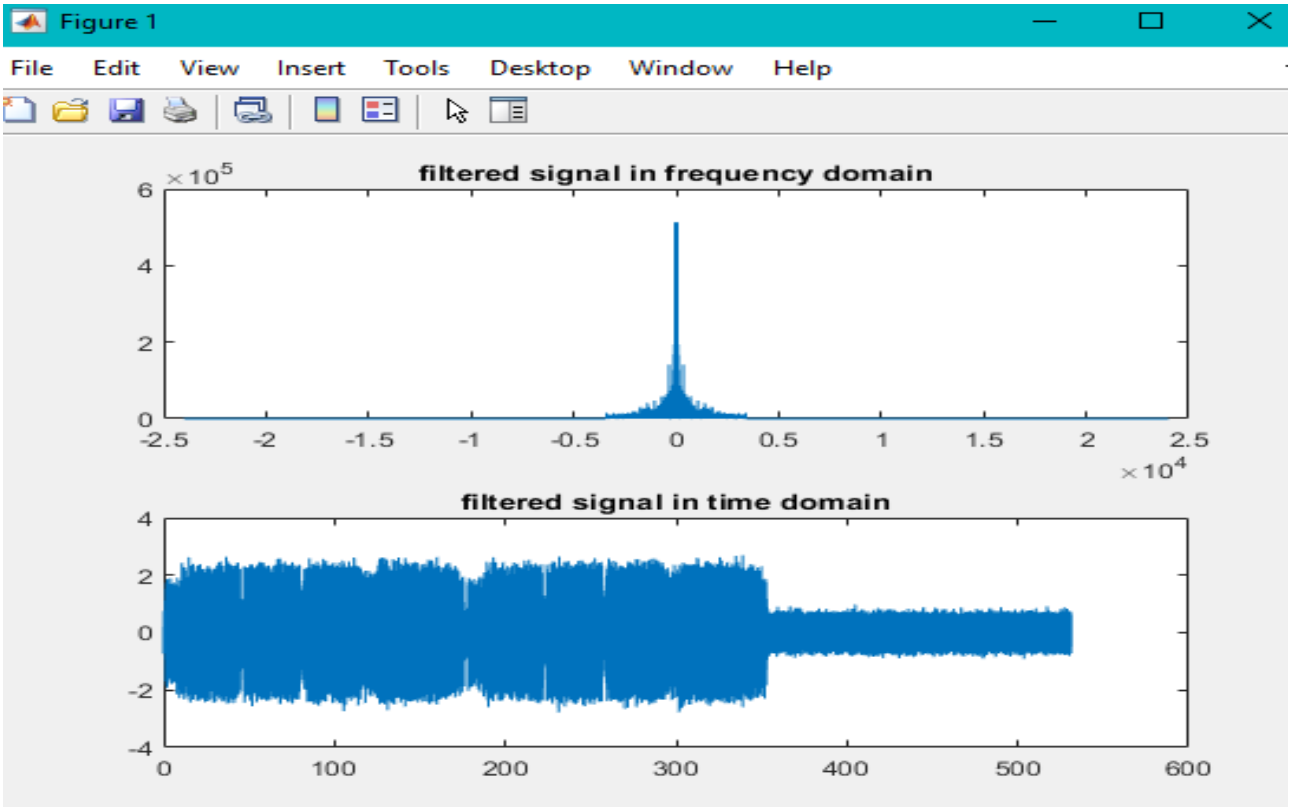


Figure 3
File Edit View Insert Tools Desktop Window Help
filtered signal in frequency domain
filtered signal in time domain

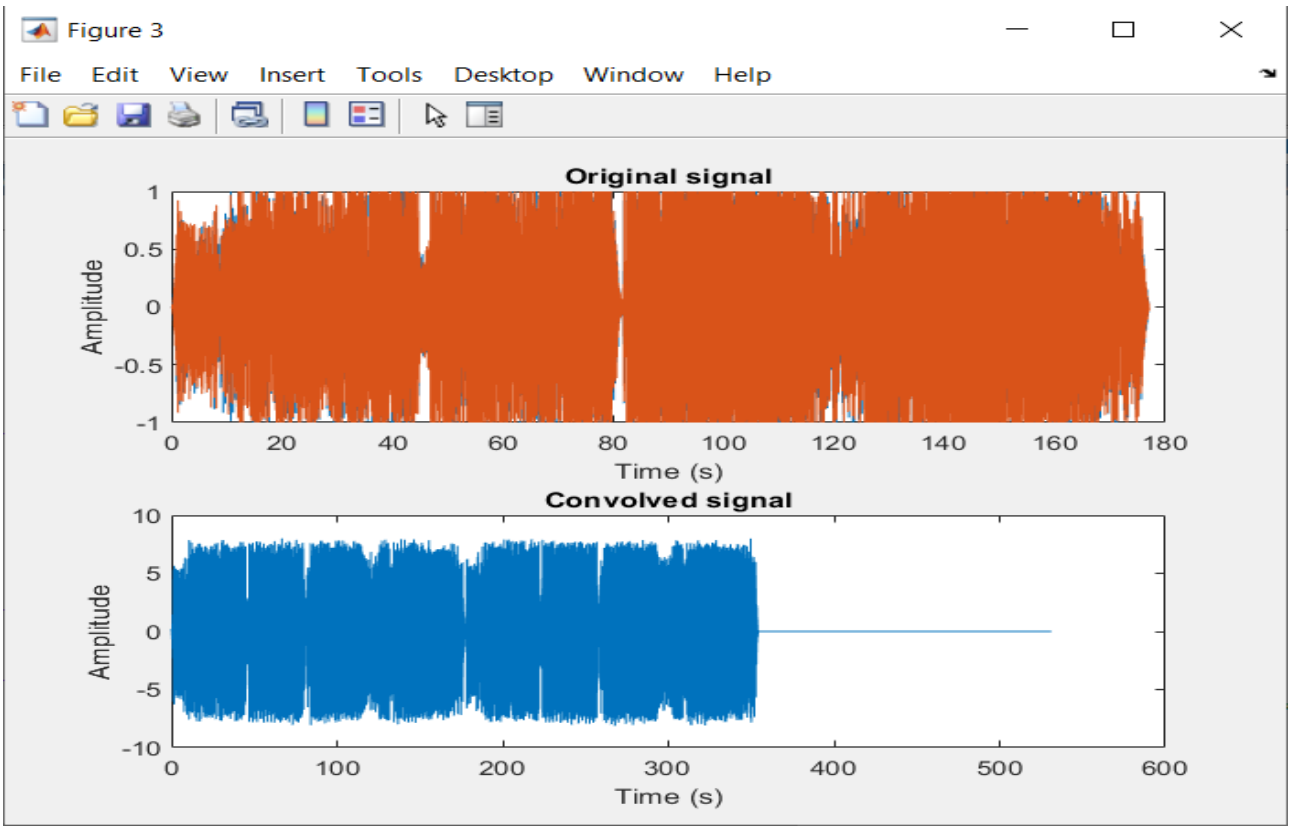
Second Channel:







Third Channel:



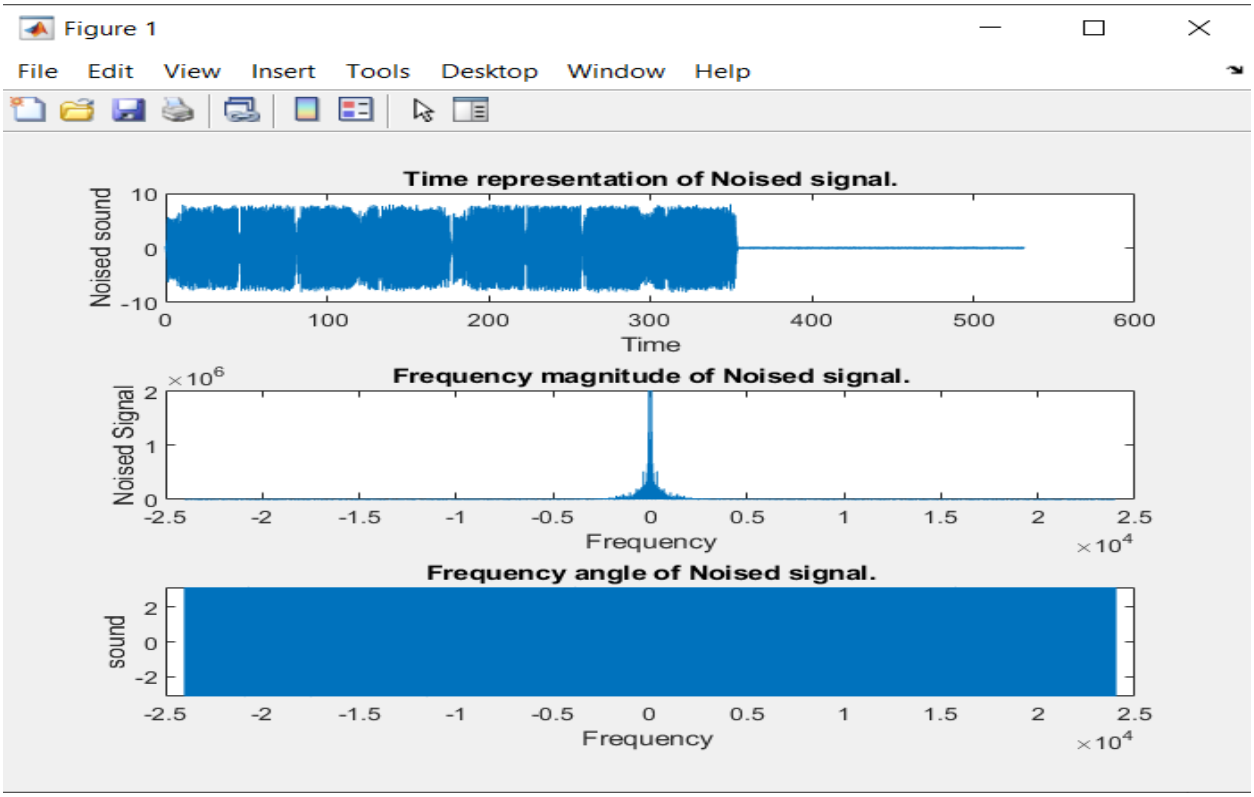


Figure 1
File Edit View Insert Tools Desktop Window Help
Time representation of Noised signal.
Noised sound
Time
Frequency magnitude of Noised signal.
Noised Signal
Frequency
Frequency angle of Noised signal.
sound
Frequency

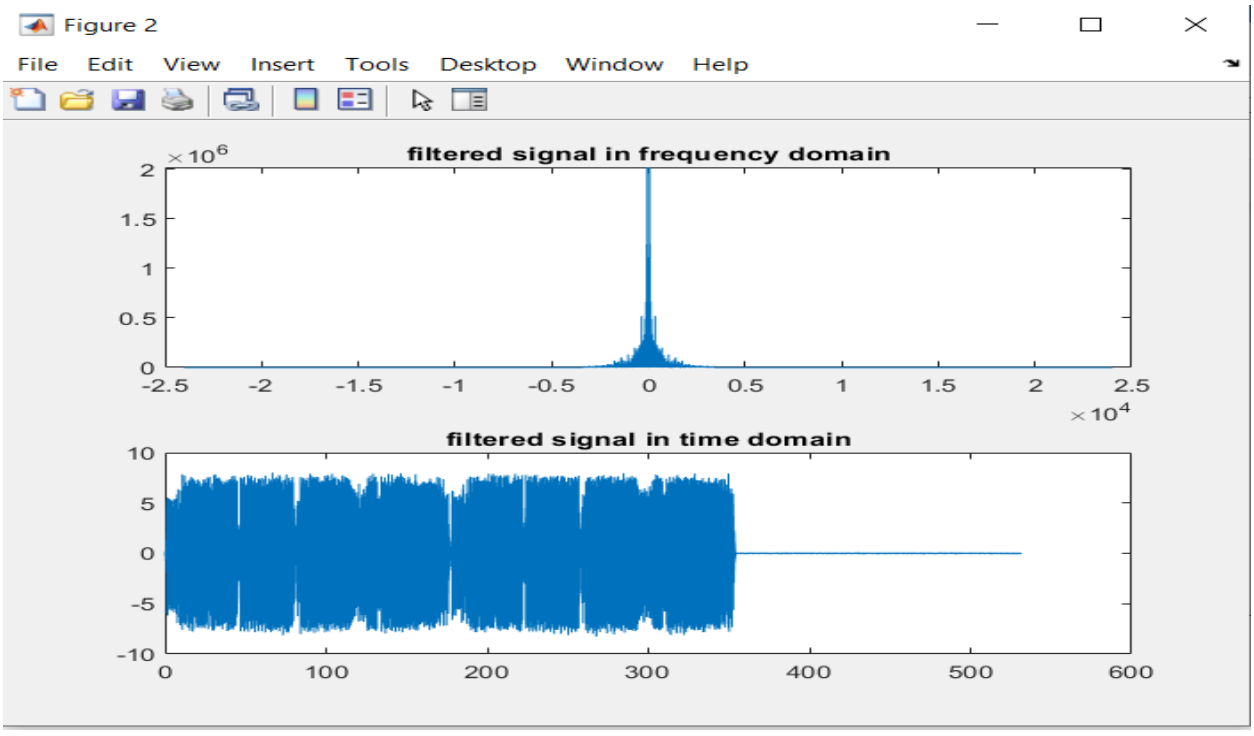
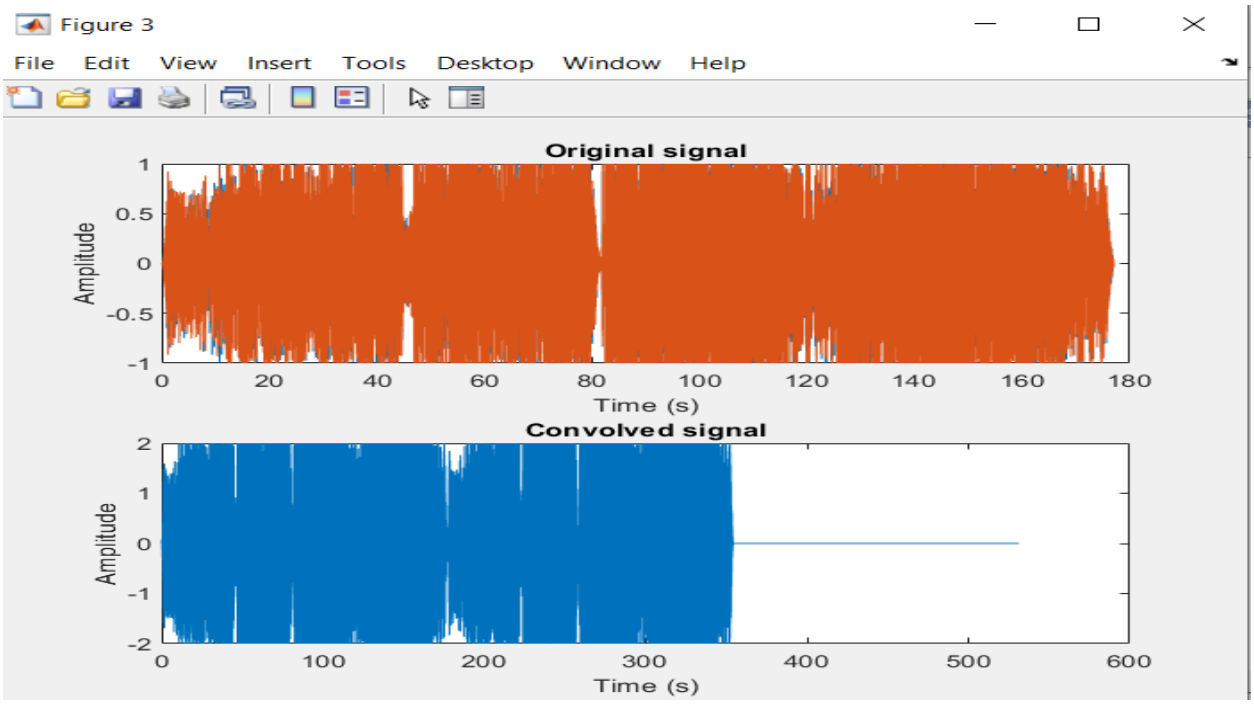
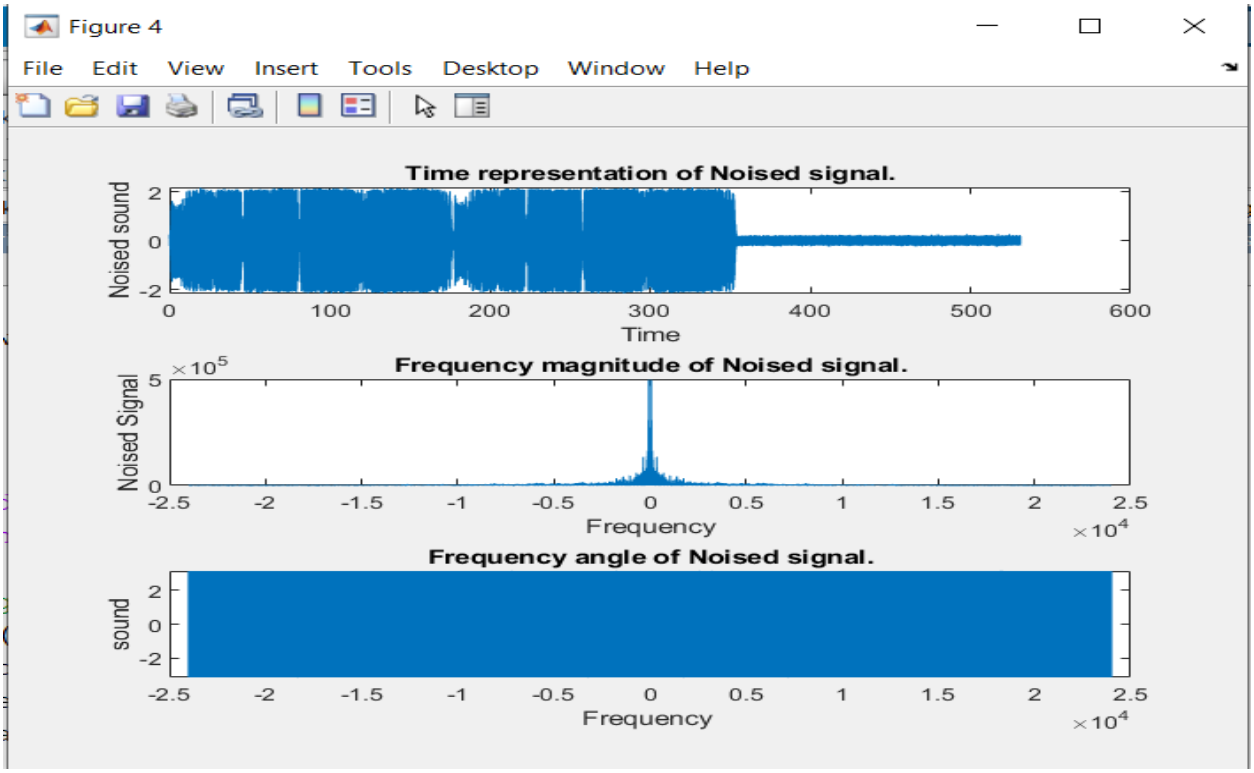
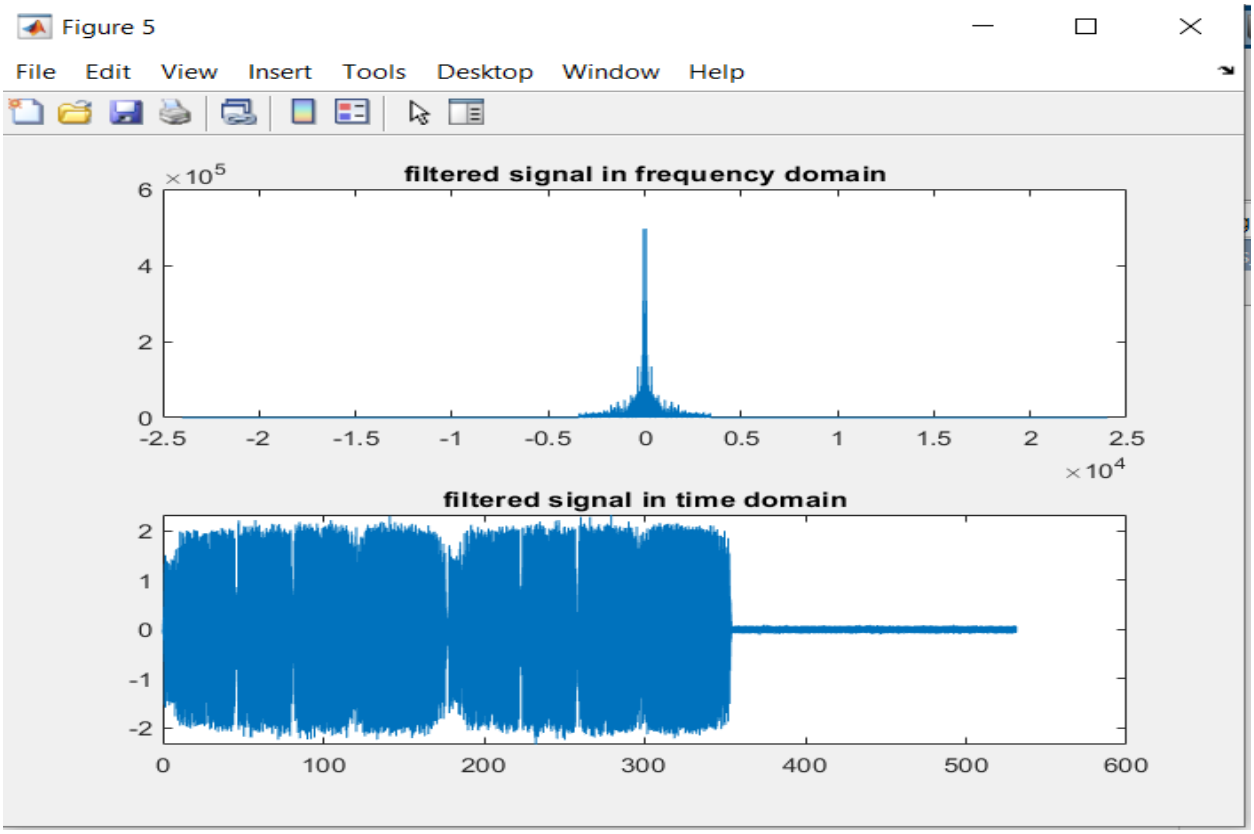


Figure 2
File Edit View Insert Tools Desktop Window Help
filtered signal in frequency domain
filtered signal in time domain

Fourth Channel:







Comment:

After comparing sound and music, it became clear that the sound was not affected much, and it largely preserved the information it contains, and it can be heard clearly to some extent, but in the case of music, and as it is known, music contains many instruments, and if any instrument departs from the specified frequency range It cannot be received theoretically as shown by last graph, but practically what we noticed is that the music appears as if it is not clear and cannot be heard clearly, as if the music is muffled and unclear or some information has been lost, unlike the sound