Faculty of Engineering, Architecture and Science

Department of Electrical and Computer Engineering

Course Number	ELE632
Course Title	Signals and Systems II
Semester/Year	Winter 2022

Instructor	Dr. Dimitri Androutsos

ASSIGNMENT No.	**5**

Assignment Title	Sampling and Discrete Fourier Transform

Submission Date	April 8, 2022
Due Date	April 10, 2022

Student Name	Reza Aablue
Student ID	500966944
Signature*	R.A.

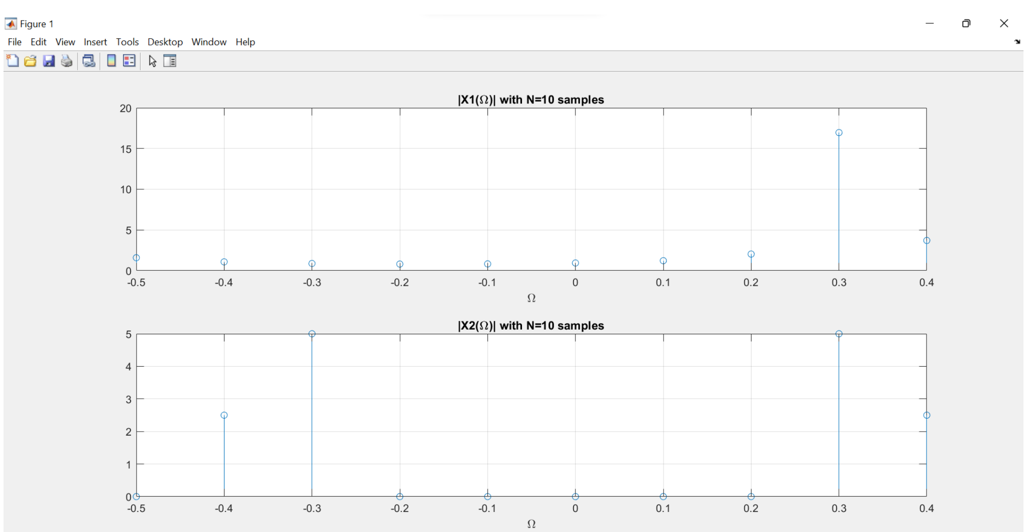
Problem A.1

A.1-i) |X2(w)| has a symmetric spectrum as the signal x2[n] is a periodic signal.

A.1-ii) It is possible to distinguish both frequency components in the plots since x1[n] is not a periodic signal (implies that |X1(w)| will have only one peak unlike |X2(w)| which has multiple peaks).

A.1-iii) There is an exponential trend from w=0 up to w=0.3, but as the sampling rate is a small number, the frequency component is spread out at other frequencies that haven't been taken into account.

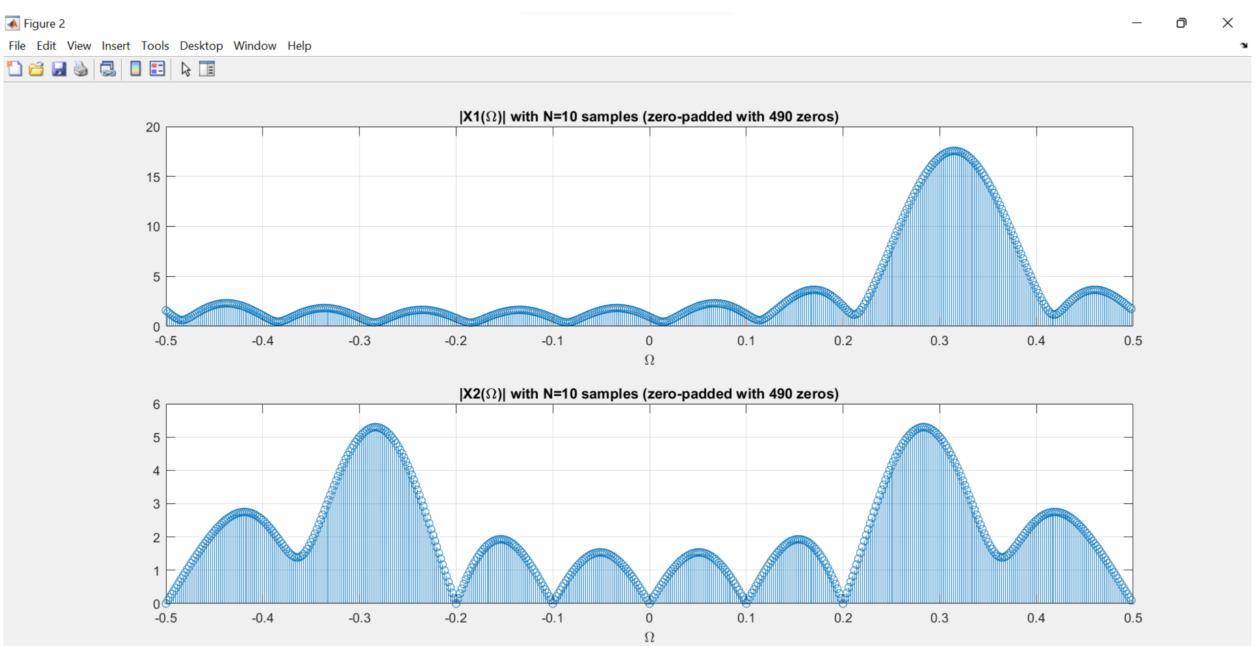
```matlab
% Reza Aablue
% 500966944
% Section 05

% Problem A.1
n1 = 0:9; % 10 samples will be taken.
fr = linspace (-0.5, 0.5-1/10, 10); % Frequency fr.
N1 = length (n1); % Length of samples is 10.

x1 = exp(1i*2*pi*n1*(30/100)) + exp(1i*2*pi*n1*(33/100)); % x1[n] signal.
x2 = cos(2*pi*n1*(30/100)) + 0.5*cos(2*pi*n1*(40/100)); % x2[n] signal.

X1 = fft (x1,N1); X2 = fft (x2, N1);

figure (1);
subplot (2,1,1); stem (fr, abs(fftshift(X1)));
title ('|X1(\Omega)| with N=10 samples'); xlabel ('\Omega'); grid;
subplot (2,1,2); stem (fr, abs(fftshift(X2)));
title ('|X2(\Omega)| with N=10 samples'); xlabel ('\Omega'); grid;
```



Problem A.2

A.2) There is noticeable improvement in the DFT as when both signals are zero-padded, a more clear image of the frequency spectrum is obtained, which makes it appear to be closer to a continuous-time definition.

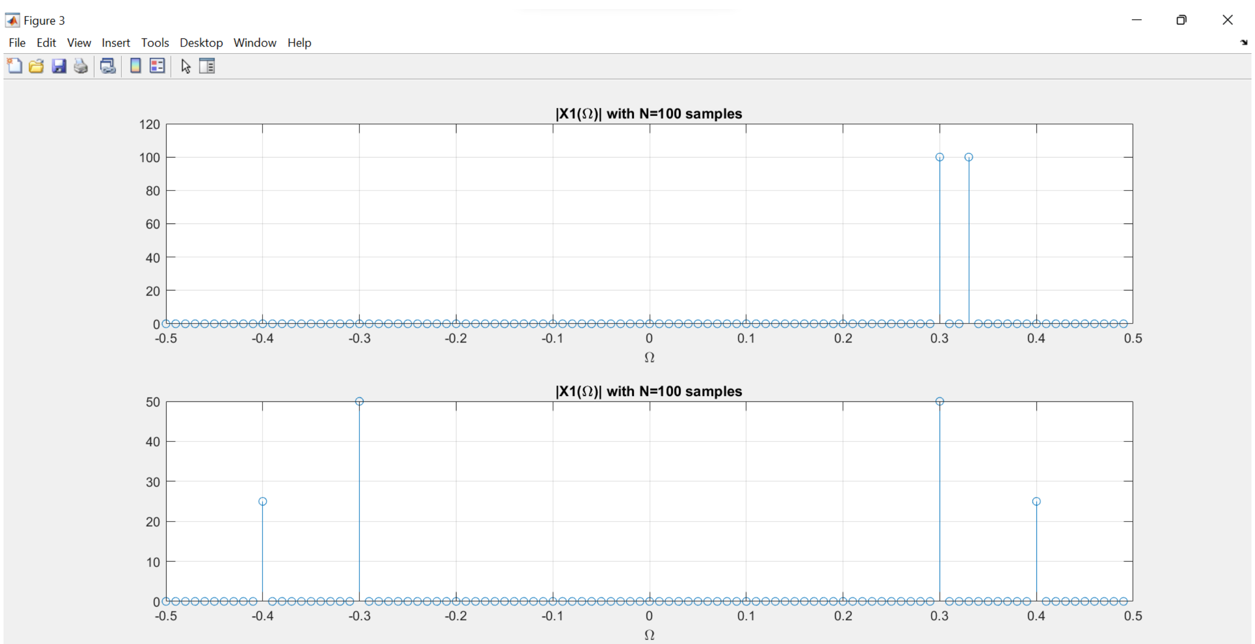
```
% Problem A.2
n2 = 0:499; % 500 samples will be taken.
N2 = length (n2);
fr2 = linspace (-0.5, 0.5-1/10, 500); % Frequency fr.

x1ofn = [zeros(1,245),x1,zeros(1,245)];
X1ofF = fft(x1ofn, N2);
x2ofn = [zeros(1,245),x2,zeros(1,245)];
X2ofF = fft(x2ofn, N2);

figure (2);
subplot (2,1,1); stem (fr2, abs(fftshift(X1ofF)));
title ('|X1(\Omega)| with N=10 samples (zero-padded with 490 zeros)');
xlabel ('\Omega'); grid;
subplot (2,1,2); stem (fr2, abs(fftshift(X2ofF)));
title ('|X2(\Omega)| with N=10 samples (zero-padded with 490 zeros)');
xlabel ('\Omega'); grid;
```



Problem A.3

A.3) |X2(w)| has a symmetric spectrum as x2[n] is a sum of cosine (periodic) signals, whose fundamental frequencies can be used to prove that x2[n] itself is also periodic.

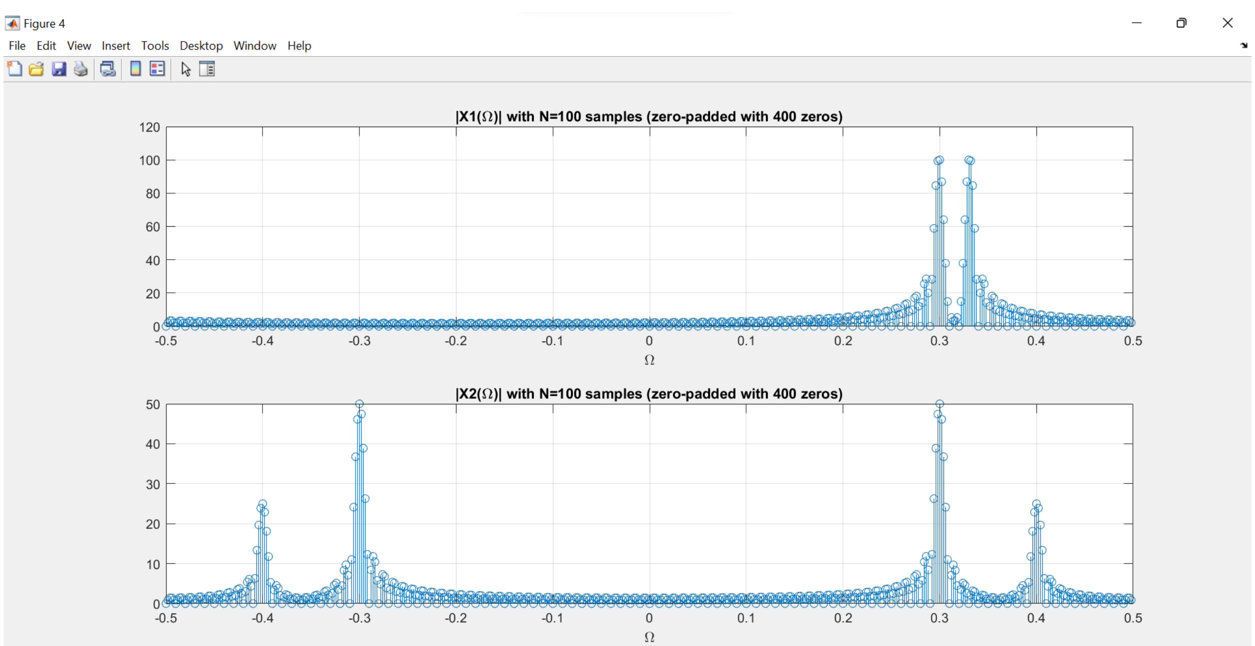
```
% Problem A.3
n3 = 0:99; % 100 samples will be taken
N3 = length(n3);
fr3 = linspace (-0.5, 0.5-1/10, 100); % Frequency fr.

x_1 = exp(1i*2*pi*n3*(30/100)) + exp(1i*2*pi*n3*(33/100)); % x1[n] signal.
x_2 = cos(2*pi*n3*(30/100)) + 0.5*cos(2*pi*n3*(40/100)); % x2[n] signal.

X_1 = fft (x_1,N3); X_2 = fft (x_2, N3);

figure (3);
subplot (2,1,1); stem (fr3, abs(fftshift(X_1)));
title ('|X1(\Omega)| with N=100 samples'); xlabel ('\Omega'); grid;
subplot (2,1,2); stem (fr3, abs(fftshift(X_2)));
title ('|X1(\Omega)| with N=100 samples'); xlabel ('\Omega'); grid;
```



<u>Problem A.4</u>

A.4) With more samples taken and zero-padding, there appears to be a much more accurate frequency-domain representation of the signal with the DFT.

```
% Problem A.4
n4 = 0:499; % 500 samples will be taken.
N4 = length(n4);
fr4 = linspace (-0.5, 0.5-1/500, 500); % Frequency fr.

x1_n = [zeros(1,200),x_1,zeros(1,200)];
X1_F = fft(x1_n, N4);
x2_n = [zeros(1,200),x_2,zeros(1,200)];
X2_F = fft(x2_n, N4);

figure (4);
subplot (2,1,1); stem (fr4, abs(fftshift(X1_F)));
title ('|X1(\Omega)| with N=100 samples (zero-padded with 400 zeros)');
xlabel ('\Omega'); grid;
subplot (2,1,2); stem (fr4, abs(fftshift(X2_F)));
title ('|X2(\Omega)| with N=100 samples (zero-padded with 400 zeros)');
xlabel ('\Omega'); grid;
```



Problem B.1-B.8

B.6) The signal was subsampled by a factor of 2 and on the frequency spectrum, the frequency was halved.

```
% Reza Aablue
% 500966944
% Section 05

load chirp.mat
filename = 'chirp.wav';
audiowrite (filename,y,Fs);
clear y Fs
[y,fs] = audioread ('chirp.wav');

% Problem B.1
No = length (y) % Number of samples.
To = No/fs % Period of signal.
T = 1/fs % Sampling interval.

% Problem B.2
t = linspace (0,To,No);
figure (1); plot (t,y); title ('Plot of y with respect to time'); xlabel ('t'); grid;

% Problem B.3
w = linspace (-(fs/2),(fs/2),No);
Y = fft (y);
figure (2); plot (w,fftshift(abs(Y)));
title ('Y(\Omega): DTF of y(t)'); xlabel('\Omega'); grid;

% Problem B.4
y1 = y(1:2:No);
N1 = length (y1)
T01 = N1/fs
T1 = 2*fs

% Problem B.5
t1 = t(1:2:No);
figure (3); plot (t1,y1); title ('Plot of y1 with respect to time'); grid;

% Problem B.6
w1 = linspace (-(fs/4),(fs/4),N1);
Y1 = fft (y1);
figure (4); plot (w1, fftshift(abs(Y1)));
title ('Y1(\Omega): DFT of y1'); xlabel ('\Omega'); grid;
```

```
% Problem B.7
% Commands to use: sound (y,fs), sound (y1,fs)

% Problem B.8
% Command to use: sound (y5, fs)
y5 = y(1:5:No);
N5 = length (y5);
T05 = N5/fs;
T5 = 2*fs;
t5 = t(1:5:No);

w5 = linspace(-(fs/10),(fs/10),N5);
Y5 = fft (y5);

figure (5); plot (t5, y5);
title ('Plot of y5 with respect to time'); grid;

figure (6); plot (w5,fftshift(abs(Y5)));
title ('Y5(\Omega): DFT of y5'); xlabel ('\Omega'); grid;
```

Command Window

```
>> ProblemB

No =

       13129


To =

    1.6027

T =

   1.2207e-04


N1 =

        6565
```

```
T01 =

    0.8014


T1 =

       16384
```

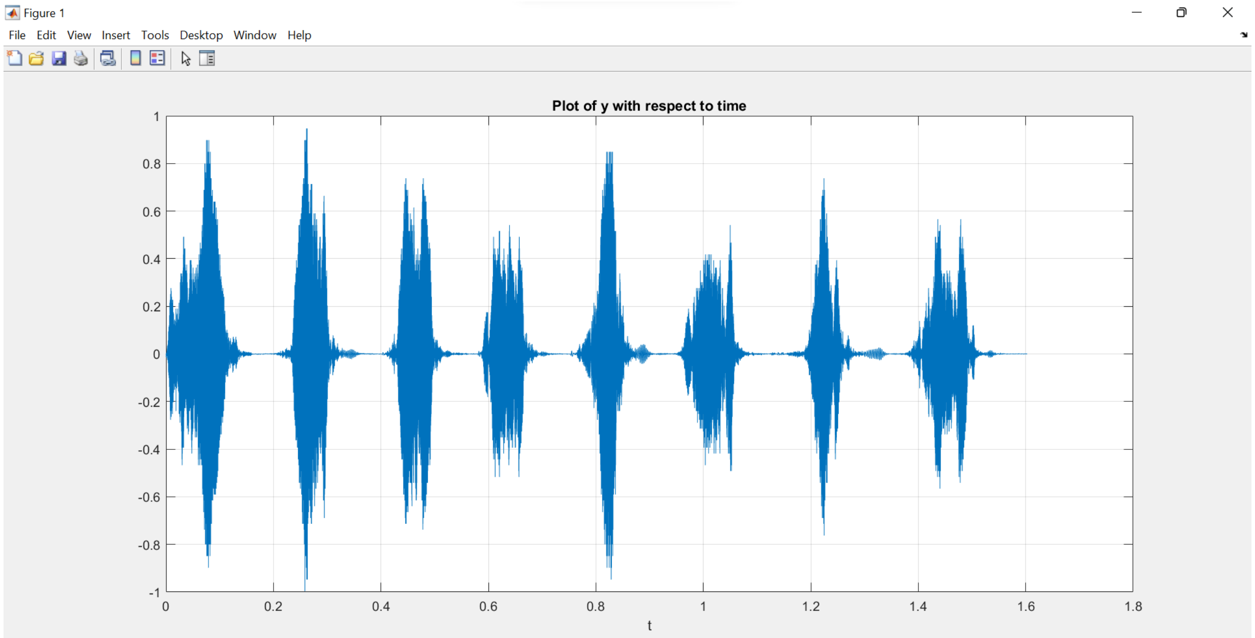


Figure 1
File Edit View Insert Tools Desktop Window Help
Plot of y with respect to time
t

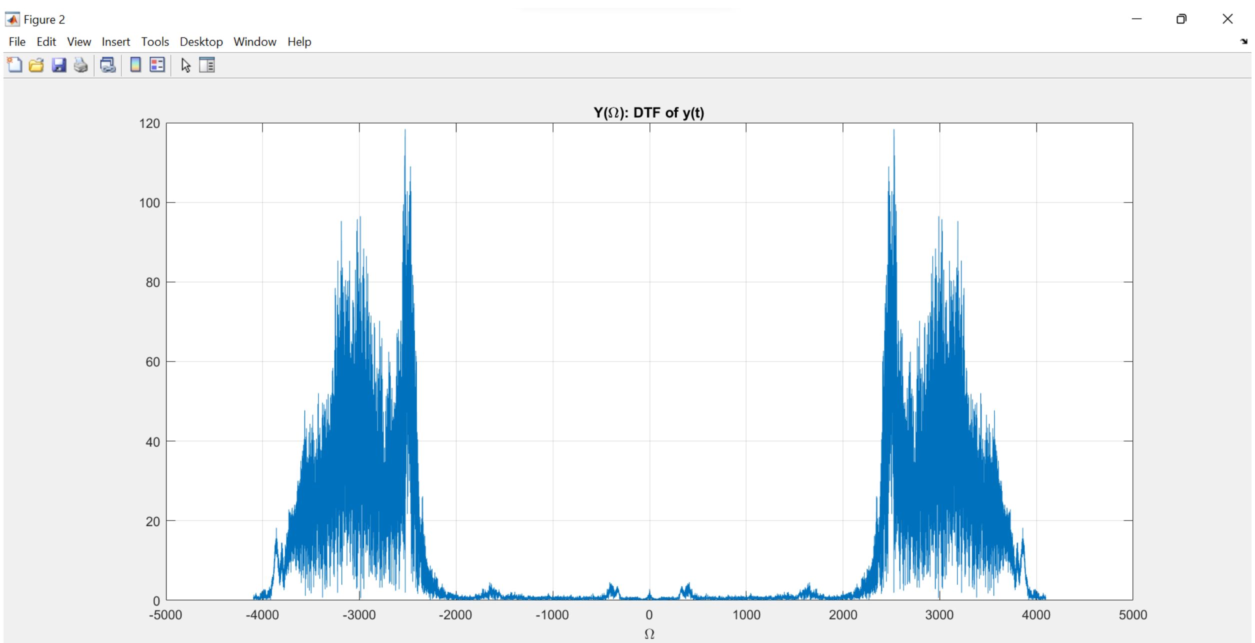


Figure 2
File Edit View Insert Tools Desktop Window Help
Y(Ω): DTF of y(t)
Ω

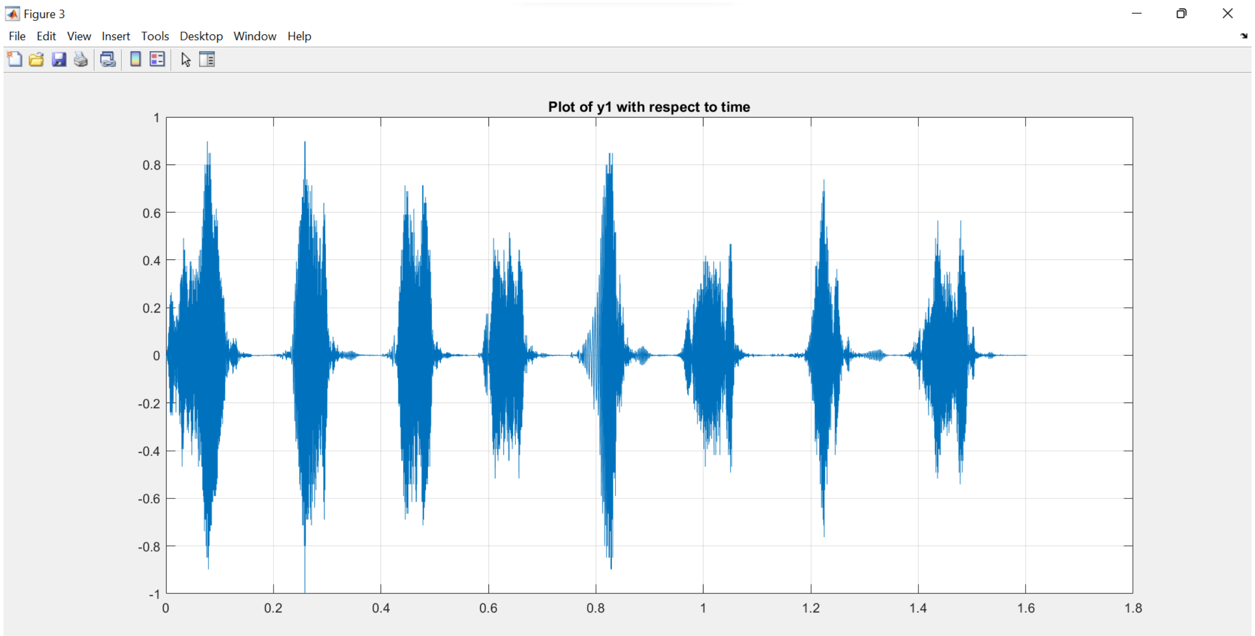


Figure 3
File Edit View Insert Tools Desktop Window Help
Plot of y1 with respect to time
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
0 0.2 0.4 0.6 0.8 1 1.2 1.4 1.6 1.8

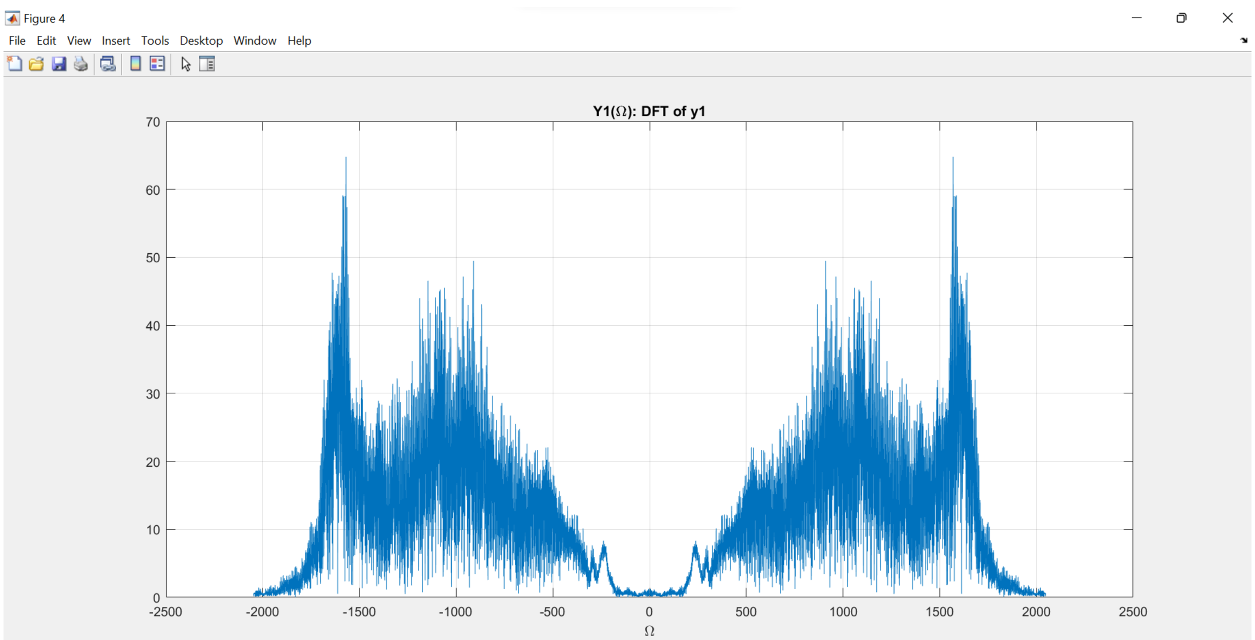


Figure 4
File Edit View Insert Tools Desktop Window Help
Y1(Ω): DFT of y1
70
60
50
40
30
20
10
0
-2500 -2000 -1500 -1000 -500 0 500 1000 1500 2000 2500
Ω

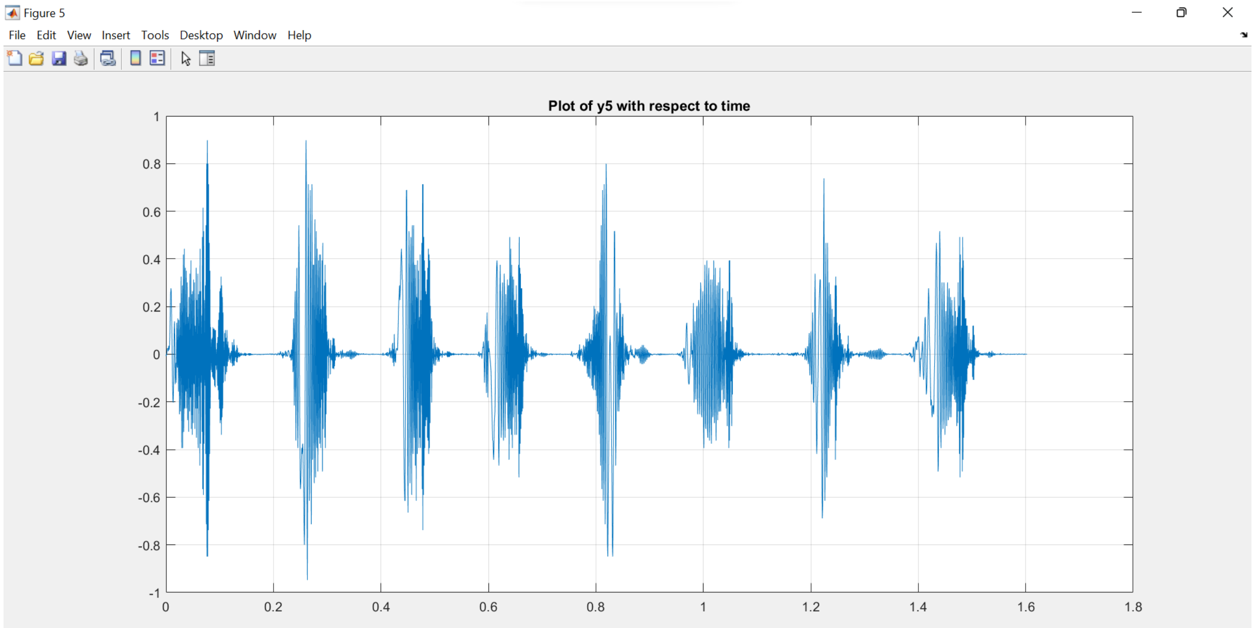


Figure 5
File Edit View Insert Tools Desktop Window Help
Plot of y5 with respect to time

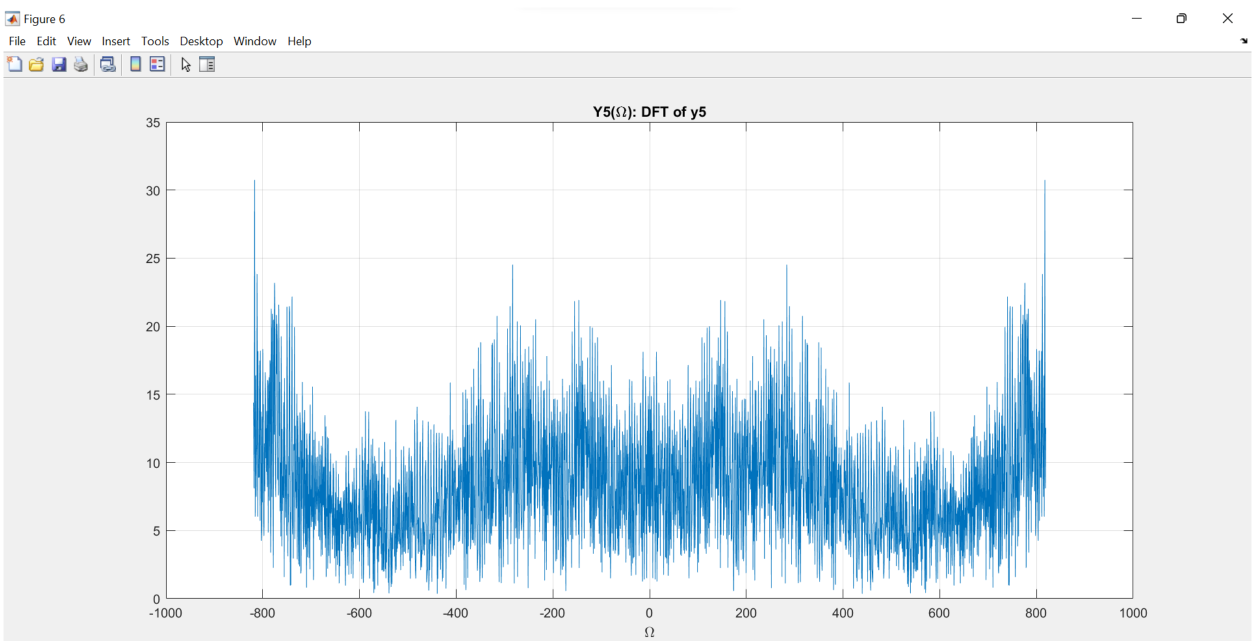


Figure 6
File Edit View Insert Tools Desktop Window Help
Y5(Ω): DFT of y5
Ω

Problem C.1-C.5

C.2) The frequency components beyond the +/-2 kHz range are removed, and so, parts of the audio are not silent due to the use of the low pass filter.

C.3) The bass sounds (lower frequencies) are removed and replaced with silence.

C.4) The higher frequency components of the audio are amplified and are now louder to one's ears.

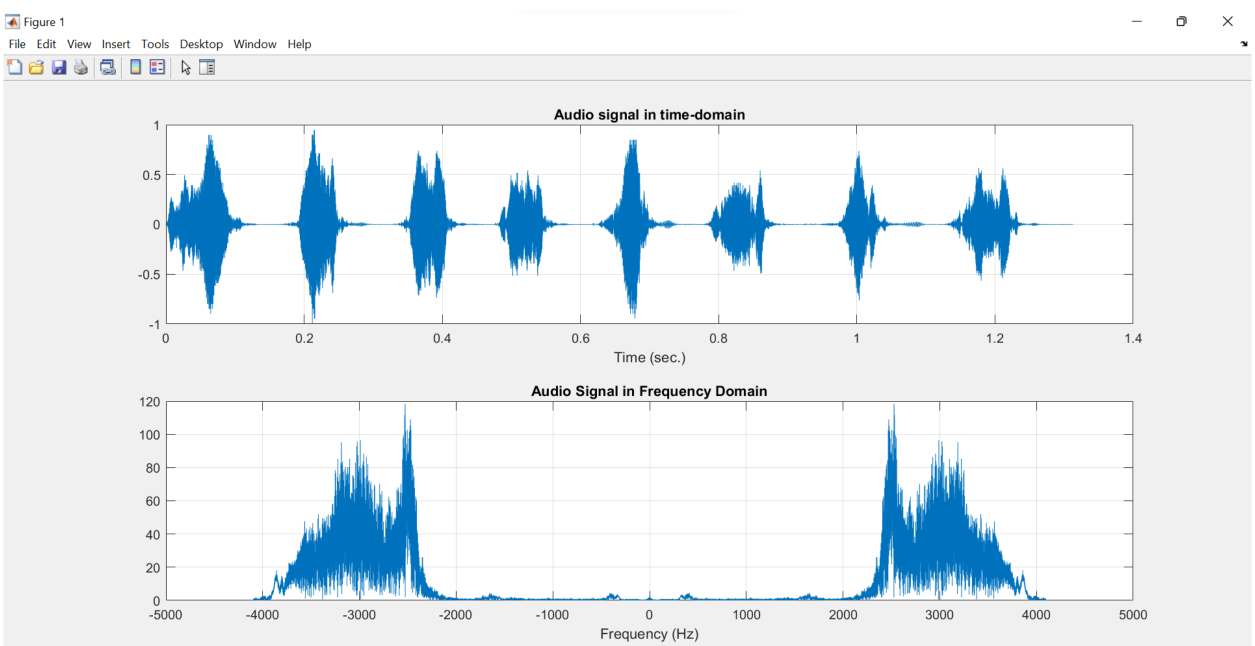
C.5) The linearity property of the DFT is used to ensure that time-domain amplification of the treble frequencies reflects in the frequency domain as well.

```
% Reza Aablue
% 500966944
% Section 05

% Problem C.1
file = 'chirp.wav'
[y,Fs] = audioread(file);
audio = y;
audio_DFT = fftshift(fft(audio)); % DFT of audio signal.
Fs_half = Fs/2;
t = 0:1:length(audio_DFT)-1; t=t/10000;
f = linspace (0,Fs,length(y)); f=f-Fs_half;

% LPF at 2 KHz
H = abs (f) < 2000; H = transpose (H);
filtered_audio = H.*audio_DFT; % Filtered system at 2 KHz.

figure (1); subplot (2,1,1); plot (t, audio);
title ('Audio signal in time-domain'); xlabel ('Time (sec.)'); grid on;
subplot (2,1,2); plot (f,abs(audio_DFT)); title ('Audio Signal in Frequency Domain');
xlabel ('Frequency (Hz)'); grid on;

figure (2); plot (f,abs(H)); title ('Lowpass Filter (2 KHZ)'); xlabel('Frequency (Hz)'); grid on;
```

```
figure (3); subplot (2,1,1); plot (f,abs(filtered_audio));
title ('Audio Signal in Freq. Domain [-2 KHz:2 KHz]'); xlabel ('Frequency (Hz)'); grid on;
subplot (2,1,2); plot (t,real(ifft(fftshift(filtered_audio))));
title ('Filtered Audio - Time Domain'); xlabel ('Time (sec.)'); grid on;

% Problem C.2
sound (real(ifft(fftshift(filtered_audio))),Fs);

% Problem C.3
% Bandpass bass filtering out (16-256 Hz)
H2 = ~(abs(f) >= 16 & abs (f) <= 256);
H2 = transpose (H2);

% Filtering out bass frequencies
filtered_audio2 = audio_DFT.*H2;
figure (4); plot (f,abs(H2));
title ('Bass Filter [16 Hz:256 Hz]'); xlabel ('Frequency (Hz)'); grid on;

figure (5); subplot (2,1,1); plot (f,abs(filtered_audio2));
title ('Filtered Audio - Frequency Domain'); xlabel ('Frequency (Hz)'); grid on;
```

```
subplot (2,1,2); plot (t,real(ifft(fftshift(filtered_audio2))));
title ('Filtered Audio - Time Domain'); xlabel ('Time (sec.)'); grid on;

sound (real(ifft(fftshift(filtered_audio2))),Fs);

% Problem C.4
% Bandpass filter (2048-16384 Hz)
H3 = abs (f) >= 2048 & abs (f) <= 16384;
H3 = transpose (H3);

% Change amplitude of BPF to 0.25 to reduce the frequency components
% passing through it
H3 = H3.*0.25;

% Take the 25% to add to original audio
filtered_audio3 = audio_DFT + (audio_DFT.*H3);
```

```
figure (6); plot (f,real(H3));
title ('Treble Filter [2048 Hz: 16384 Hz]'); xlabel ('Frequency (Hz)');
grid on;

figure (7); subplot (2,1,1); plot (f,abs(filtered_audio3));
title ('Amplified Audio - Frequency Domain'); xlabel ('Frequency (Hz)');
grid on;
subplot (2,1,2); plot (t,real(ifft(filtered_audio3)));
title('Amplified Audio - Time Domain'); xlabel ('Time (sec.)'); grid on;

sound (real(ifft(fftshift(filtered_audio3))),Fs);
```



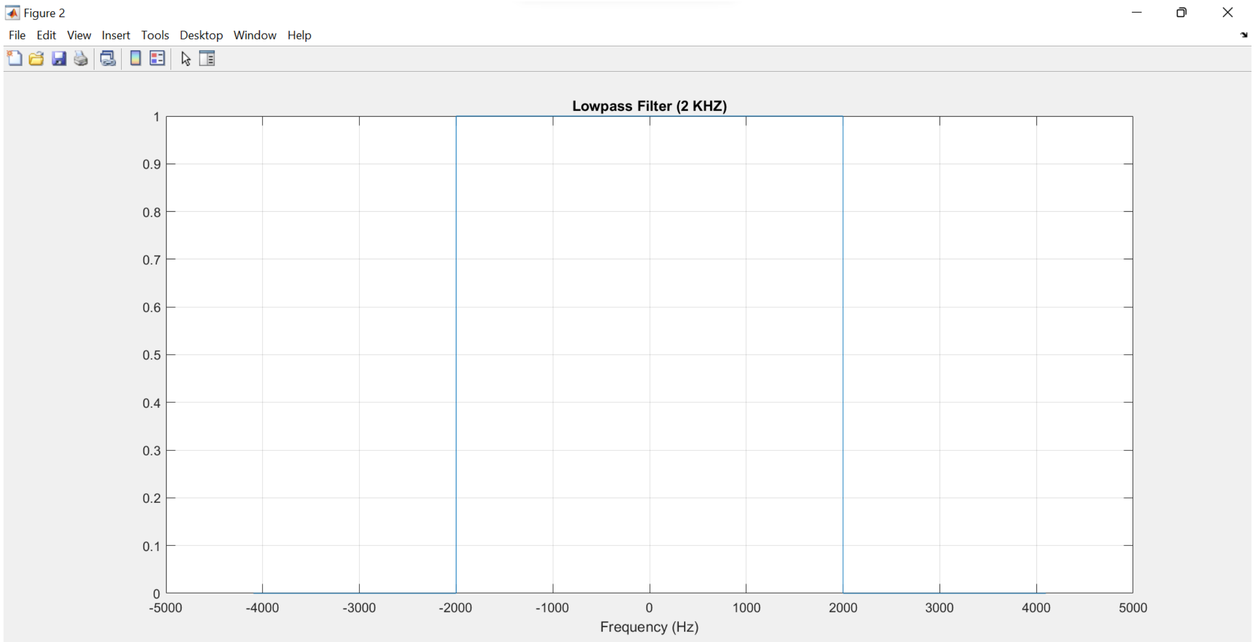


Figure 2
File Edit View Insert Tools Desktop Window Help
Lowpass Filter (2 KHZ)
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
-5000 -4000 -3000 -2000 -1000 0 1000 2000 3000 4000 5000
Frequency (Hz)

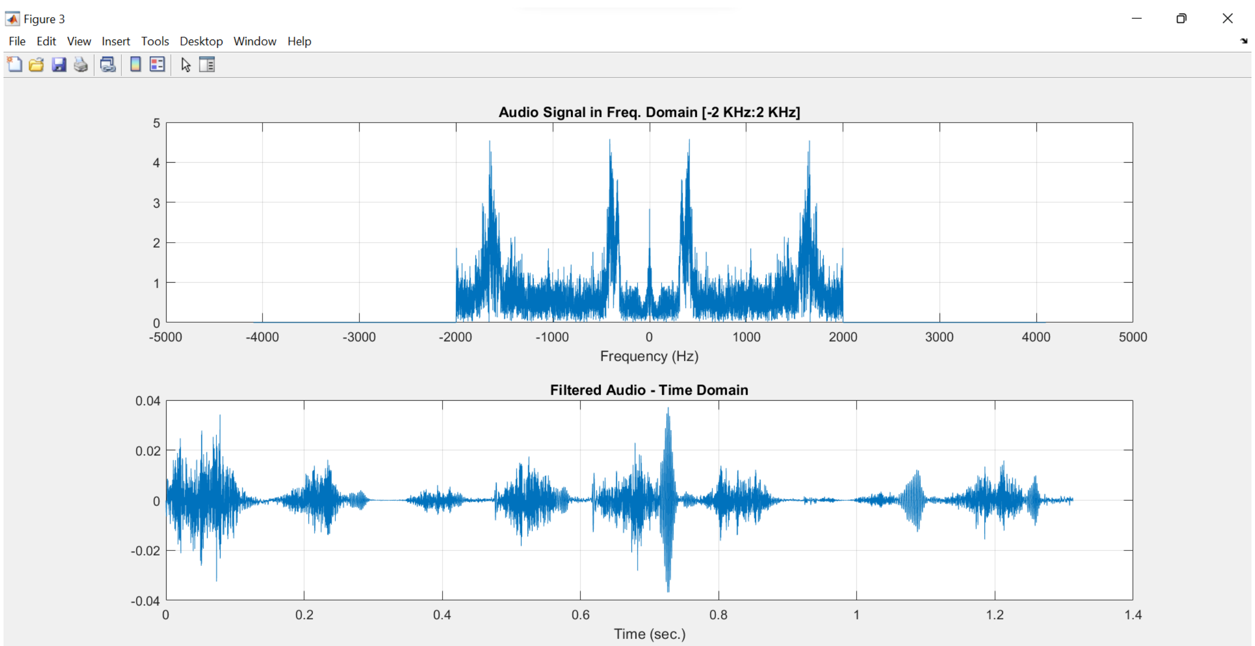


Figure 3
File Edit View Insert Tools Desktop Window Help
Audio Signal in Freq. Domain [-2 KHz:2 KHz]
5
4
3
2
1
0
-5000 -4000 -3000 -2000 -1000 0 1000 2000 3000 4000 5000
Frequency (Hz)
Filtered Audio - Time Domain
0.04
0.02
0
-0.02
-0.04
0 0.2 0.4 0.6 0.8 1 1.2 1.4
Time (sec.)

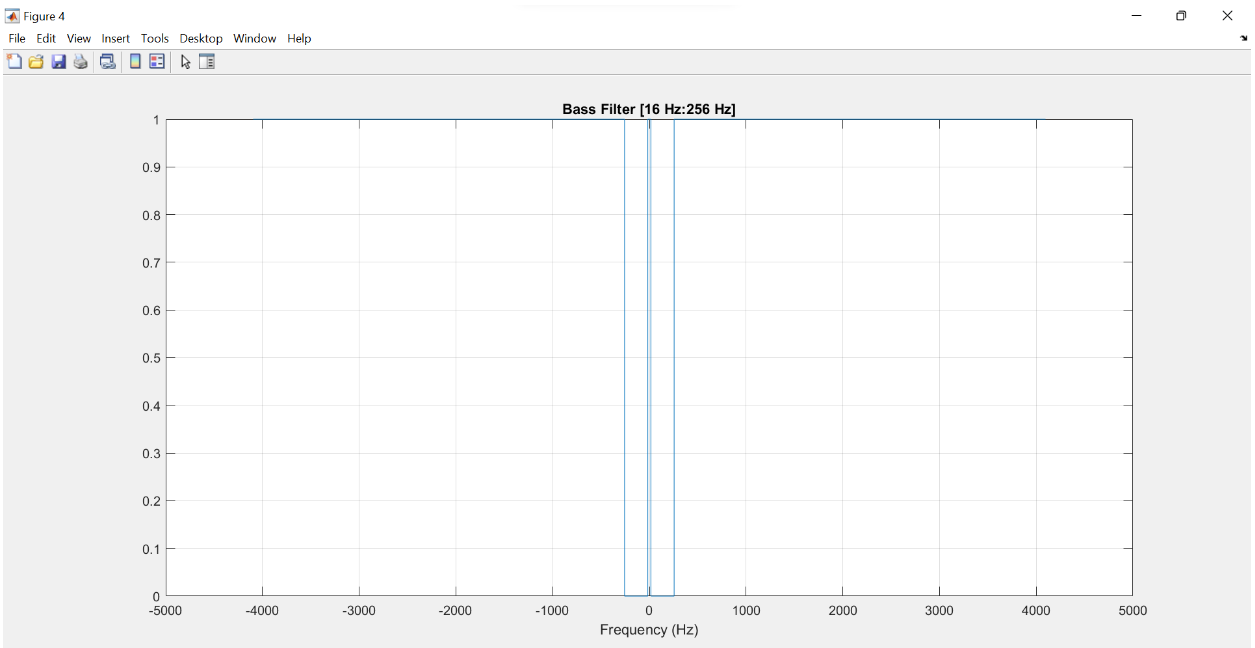


Figure 4
File Edit View Insert Tools Desktop Window Help
Bass Filter [16 Hz:256 Hz]
Frequency (Hz)

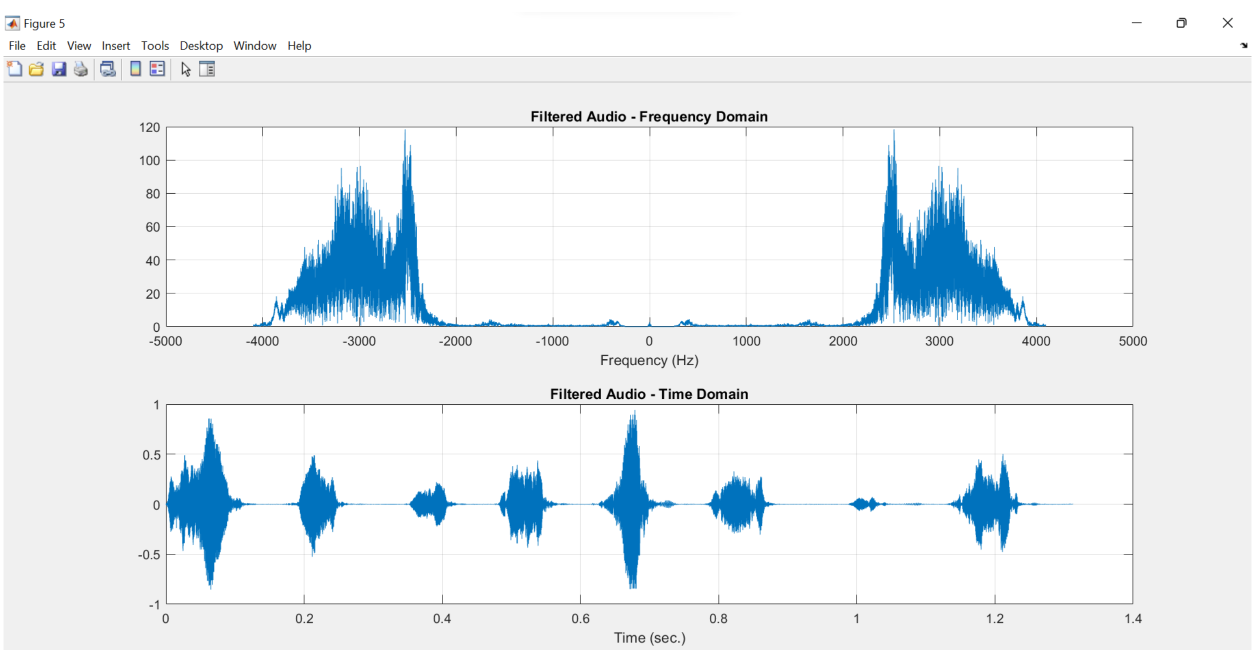


Figure 5
File Edit View Insert Tools Desktop Window Help
Filtered Audio - Frequency Domain
Frequency (Hz)
Filtered Audio - Time Domain
Time (sec.)

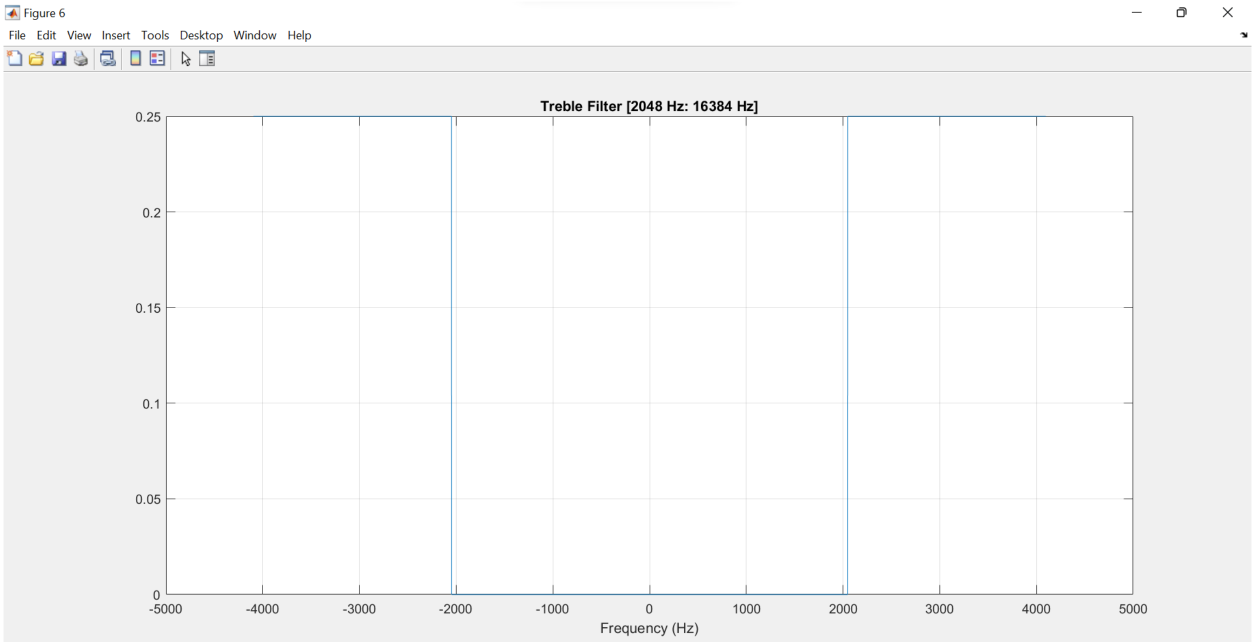


Figure 6
File Edit View Insert Tools Desktop Window Help
Treble Filter [2048 Hz: 16384 Hz]
0.25
0.2
0.15
0.1
0.05
0
-5000 -4000 -3000 -2000 -1000 0 1000 2000 3000 4000 5000
Frequency (Hz)

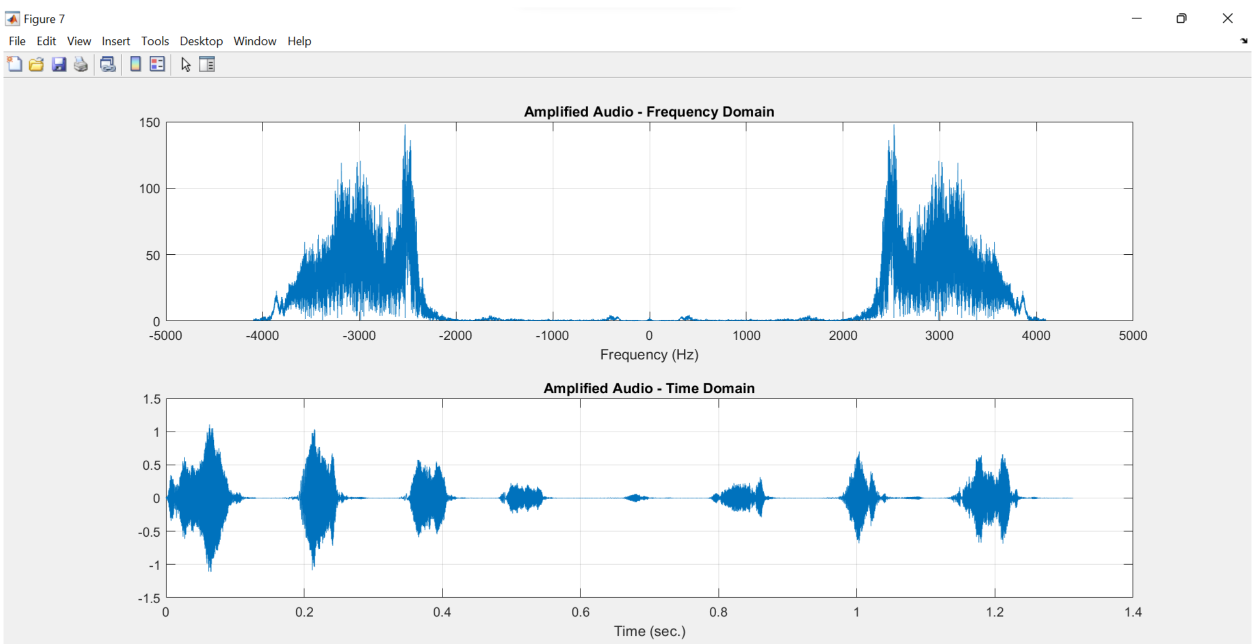


Figure 7
File Edit View Insert Tools Desktop Window Help
Amplified Audio - Frequency Domain
150
100
50
0
-5000 -4000 -3000 -2000 -1000 0 1000 2000 3000 4000 5000
Frequency (Hz)
Amplified Audio - Time Domain
1.5
1
0.5
0
-0.5
-1
-1.5
0 0.2 0.4 0.6 0.8 1 1.2 1.4
Time (sec.)