Digital Signal Processing Lab

EEL-325

Lab Journal: 04

Name: M. Mobashir Hassan
Class: BCE – 06(A)
Enrollment No: 01-132182-022

Lab # 04

Fourier Transform and Spectral Analysis in MATLAB

Objective:

- We will be able to know about FFT functions computes the Fourier transform using a fast Fourier Transform (FFT) algorithm.
- We will be able to find the fundamental frequency of audio signal.

Introduction:

Fourier Transform is a mathematical function that takes a time-based pattern as input and determines the overall cycle offset, rotation speed and strength for every possible cycle in the given pattern. The Fourier transform is applied to waveforms which are basically a function of time, space or some other variable. The Fourier transform decomposes a waveform into a sinusoid and thus provides another way to represent a waveform.

Spectral analysis studies the frequency spectrum contained in discrete, uniformly sampled data. The Fourier transform is a tool that reveals frequency components of a time- or space-based signal by representing it in frequency space.

Procedure:

- Open MATLAB Software.
- We create the new project.
- Make a new script and name it on the name of your lab.
- Write the code in main file and burn the code using run command.
- In the function file write the functionality of the code and code in main file.
- The function file and the script main file must be in the same folder.
- Run the mail file and get the output of the given input and observe the result.

Problem No. 01:

Task 1: Create an analog signal with component frequency at 15 Hz and 40 Hz with fs=200.

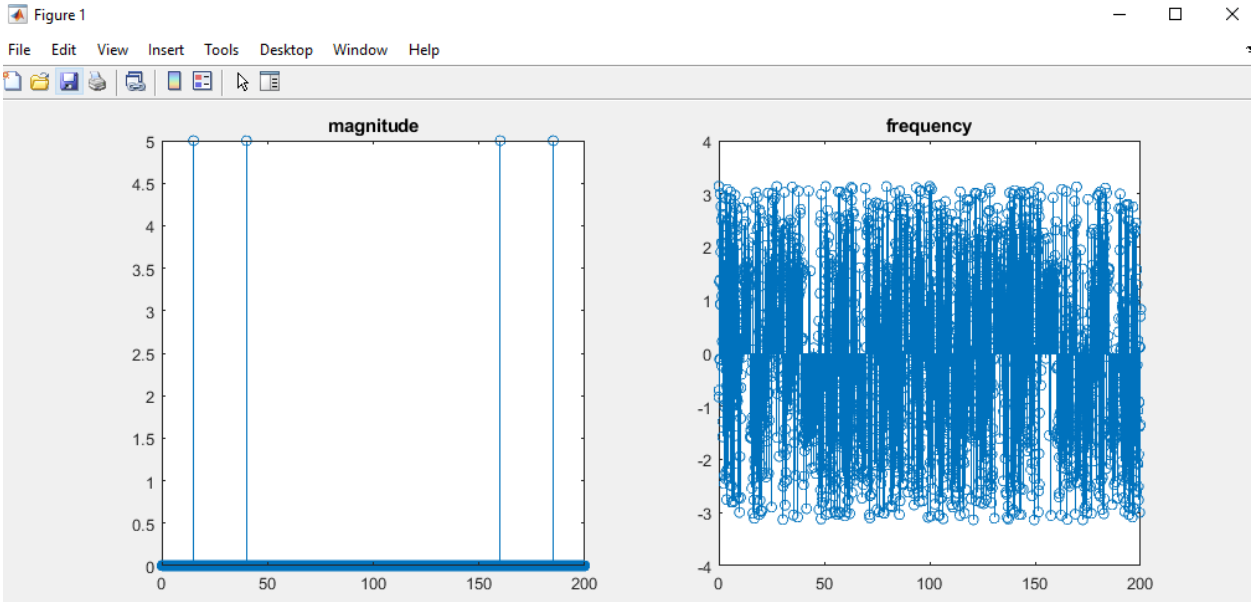
i.e. Analog signal (y) = 5*Sin (2π.f.t)

Using FFT analysis show its magnitude and frequencies in graph.

MATLAB Code:

```
%fs = 200
%Component Freq = 15Hz, 40Hz
fs=200;
t=0:1/fs:10-1/fs;
x=5*sin(2*pi*15*t)+5*sin(2*pi*40*t);
N=length(x);
x=2/N*x;
y=fft(x);
f=linspace(0,fs,N);
subplot(1,2,1);
stem(f,abs(y));
title('magnitude');
subplot(1,2,2);
stem(f,angle(y));
title('frequency');
```

Output:



Task 2: Create an analog signal with component frequency at 20 Hz, 35Hz and 40 Hz with fs=200.And Inject random Gaussian noise. Then compute the frequency components and its magnitude

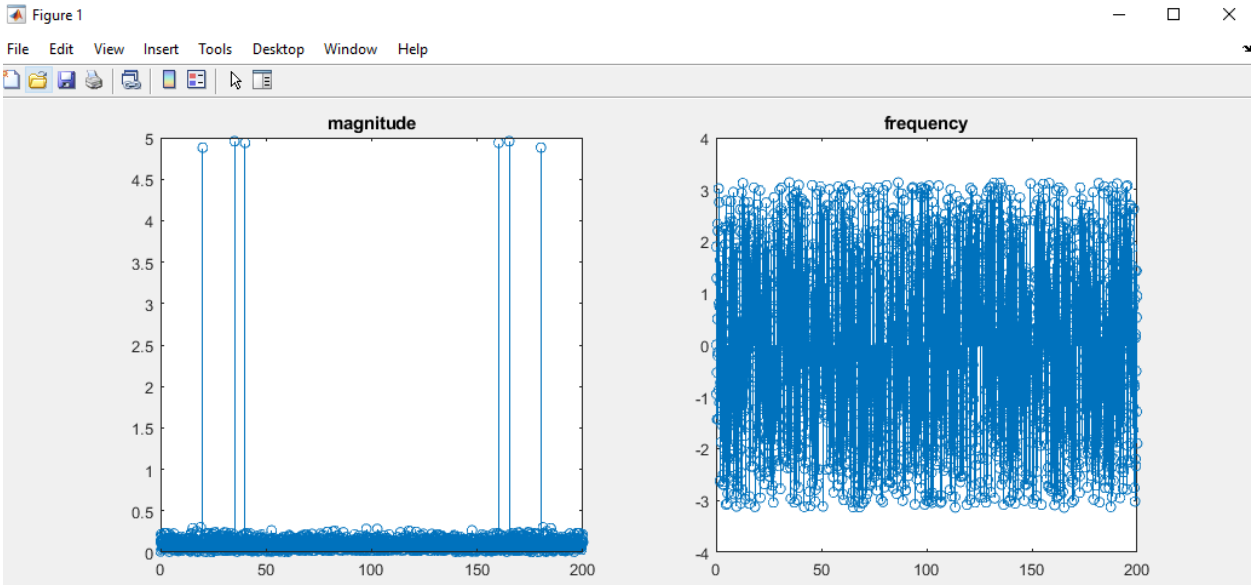
MATLAB Code:

```
%fs = 200
%Component Freq = 20Hz, 35Hz, 40Hz

fs=200;
t=0:1/fs:10-1/fs;
x=5*sin(2*pi*20*t)+5*sin(2*pi*35*t)+5*sin(2*pi*40*t)+2.5*randn(size(t));
N=length(x);
x=2/N*x;
y=fft(x);
f=linspace(0,fs,N);
subplot(1,2,1);
stem(f,abs(y));
title('magnitude');
subplot(1,2,2);
stem(f,angle(y));
```

title('frequency');

Output:



Problem No. 02:

Task 1: Find DFT of the following signal and plot the result a = [1 1 2 2] where a is a discrete time signal. By Taking IFFT verify if 'a' is regenerated or not?

MATLAB Code:

#Function

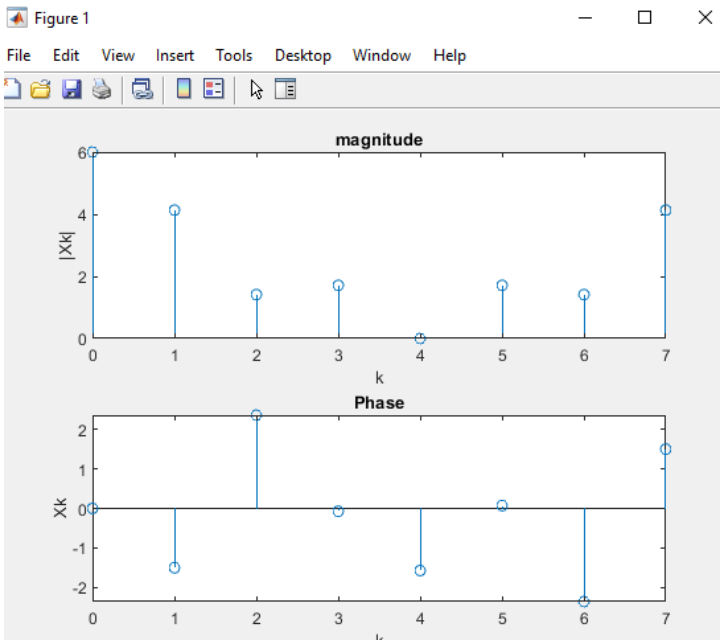
```
function Xk=dft_fun(xn, N)
L=length(xn);
if(N<L)
   error('N should be greater than or equal to L')
end
xn=[xn zeros(1,N-L)];
for k=0:N-1
   for  n=0:N-1
      Wn=exp(-j*2*pi*k*n/N);
      X1(k+1, n+1)=Wn;
   end
```

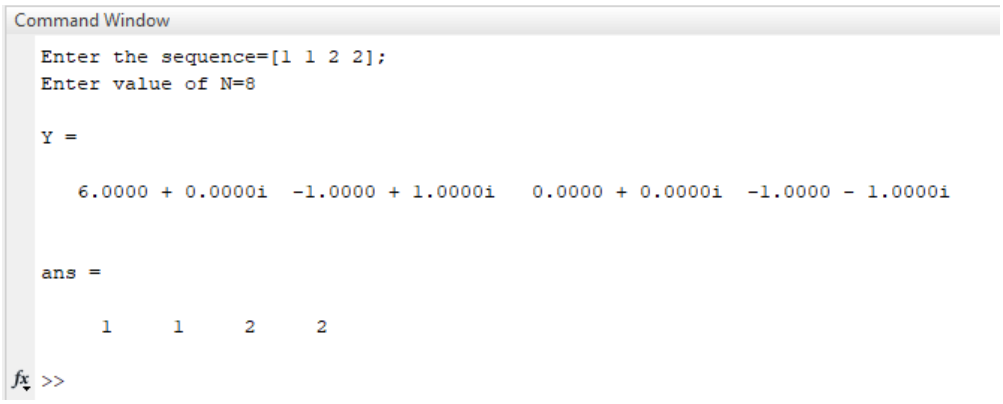
```
end
Xk=X1*xn';

clc;
clear all;
close all;
xn=input('Enter the sequence=');
N=input('Enter value of N=');
Xk=dft_fun(xn, N);
k=0:N-1;
subplot(2,1,1)
stem(k, abs(Xk))
xlabel('k')
ylabel('|Xk|')
title('magnitude')

subplot(2,1,2)
stem(k, angle(Xk))
xlabel('k')
ylabel('Xk')
title('Phase')
Y = fft(xn)
ifft(Y)
```

Output:





<u>Task 2</u>: Use audio signal of your own choice and find the fundamental frequency of that signal.

- **DSP Lab two code**
- **Take FFT (only magnitude) of my recording. Than plot.**
- **Take IFFT (only magnitude) of my recording for original recording. Then plot.**
- **Use fs=200, fftshift command, proper labeling and find power. (P=abs(y). ^2/N).**

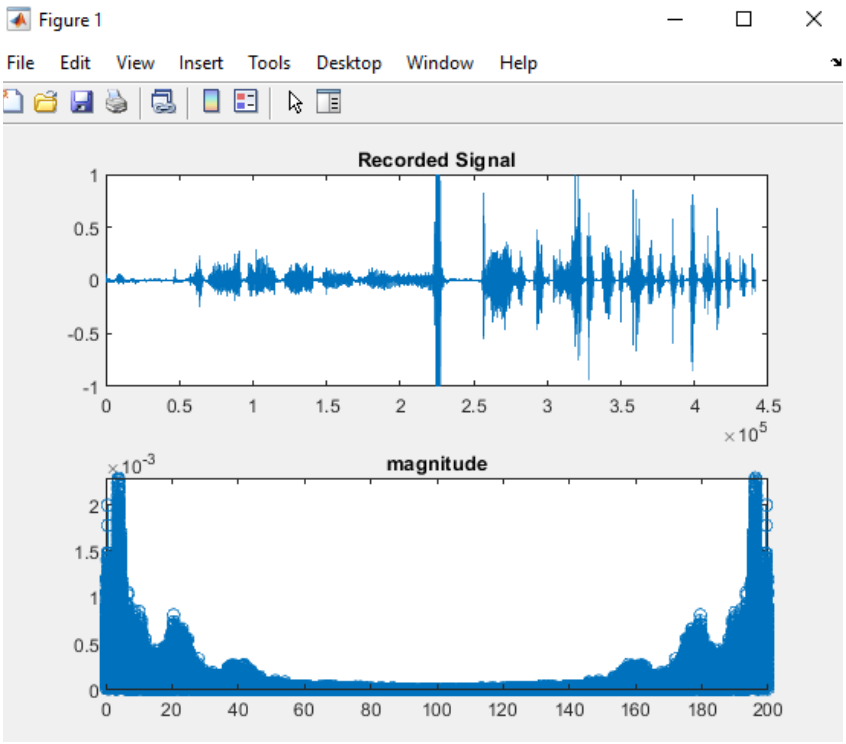
DSP Lab Two Code (Take FFT (only magnitude) of my recording. Then plot)

<u>MATLAB Code</u>:

```
recordVoice = audiorecorder(44100,8,1);
disp('Recording Started, please speak')
recordblocking(recordVoice,10);
disp('Recording Ended');
play(recordVoice);
recording = getaudiodata(recordVoice);
subplot(2,1,1);
plot(recording);
title('Recorded Signal');
fs=200;
t=0:1/fs:10-1/fs;
N=length(recording);
recording=2/N*recording;
```

```
y=fft(recording);
f1=linspace(0,fs,N);
subplot(2,1,2)
stem(f1,abs(y))
title('magnitude');
```

Output:



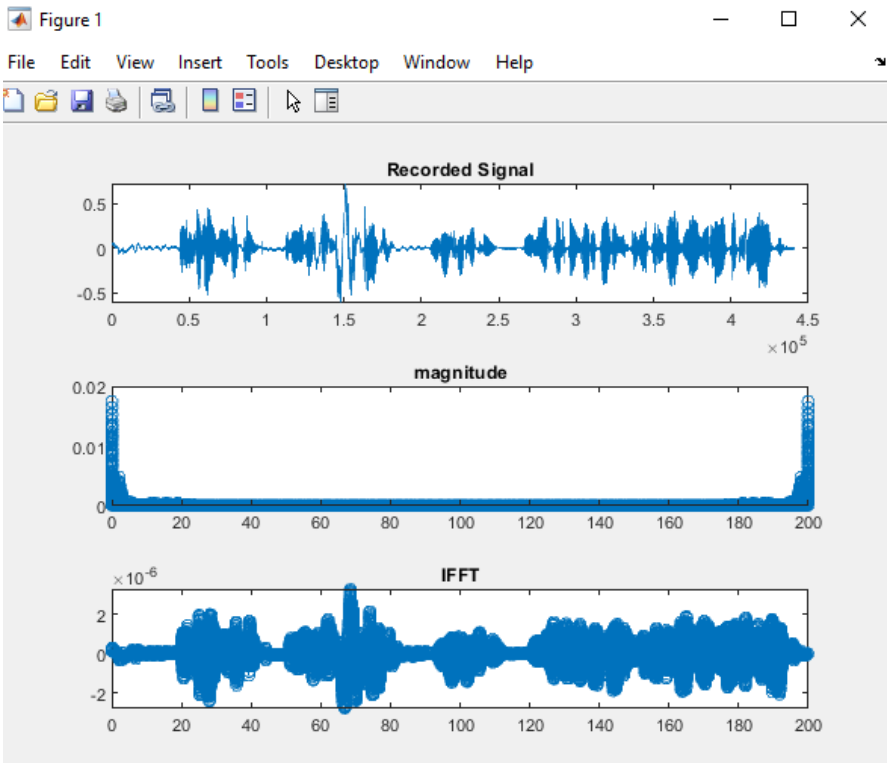
Take IFFT (only magnitude) of my recording for original recording. Then plot.

MATLAB Code:

```
recordVoice = audiorecorder(44100,8,1);
disp('Recording Started, please speak')
recordblocking(recordVoice,10);
disp('Recording Ended');
play(recordVoice);
recording = getaudiodata(recordVoice);
subplot(3,1,1);
plot(recording);
```

```
title('Recorded Signal');
fs=200;
t=0:1/fs:10-1/fs;
N=length(recording);
recording=2/N*recording;
y=fft(recording);
f1=linspace(0,fs,N);
subplot(3,1,2)
stem(f1,abs(y))
title('magnitude');
z=ifft(y);
subplot(3,1,3);
stem(f1,z);
title('IFFT');
```

Output:

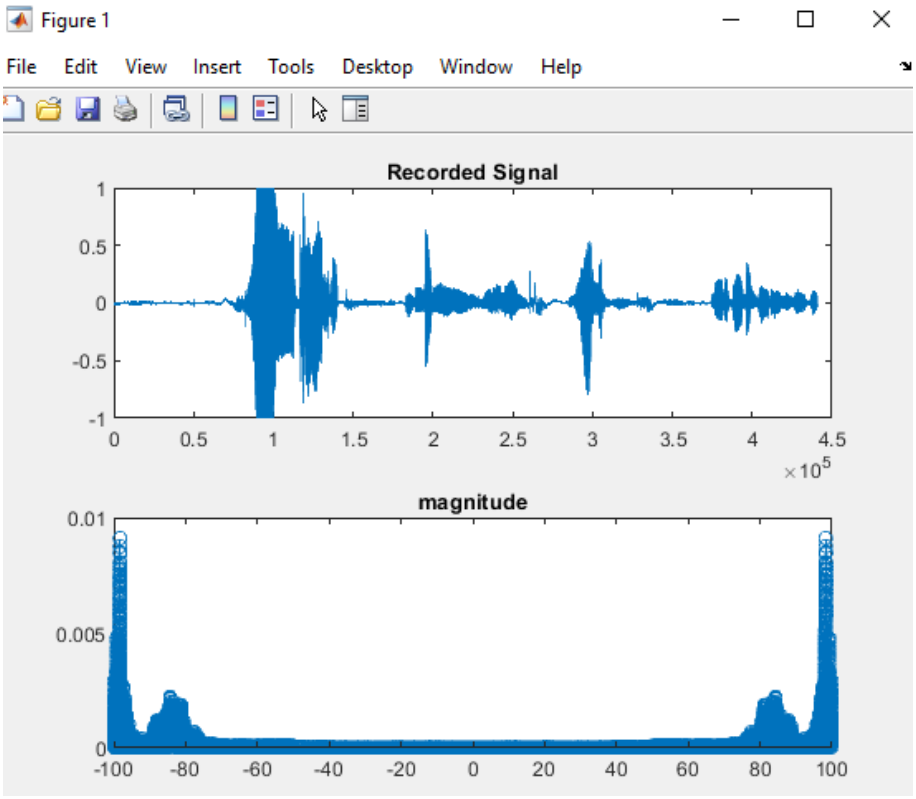


Use fs=200, fftshift command, proper labeling and find power. (P=abs(y).^2/N).

MATLAB Code:

```
recordVoice = audiorecorder(44100,8,1);
disp('Recording Started, please speak')
recordblocking(recordVoice,10);
disp('Recording Ended');
play(recordVoice);
recording = getaudiodata(recordVoice);
subplot(2,1,1);
plot(recording);
title('Recorded Signal');
fs=200;
t=0:1/fs:10-1/fs;
N=length(recording);
recording=2/N*recording;
y=fft(recording);
f1=linspace(-fs/2,fs/2,N);
subplot(2,1,2)
stem(f1,abs(y))
title('magnitude');
y1=fftshift(y);
P=abs(y1).^2/N
```

Output:



Conclusion:

In this lab we were taught about Fourier Transform and Spectral Analysis in MATLAB. Fourier Transform is a mathematical function that takes a time-based pattern as input and determines the overall cycle offset, rotation speed and strength for every possible cycle in the given pattern. We implemented code of the above systems. We performed different functions on the signals. This lab was full of learning.