Namal University, Mianwali

Department of Electrical Engineering

EE 345 (L) – Digital Signal Processing (Lab)

Lab – 5

Analysis of Signals using DFT and IDFT in MATLAB

Student Name	Student ID
Fahim Ur Rehman Shah	NIM-BSEE-2021-24

Instructor: Zulaikha Kiran

Lab Engineer: Faizan Ahmad

Version 1.0 – Muhammad Usman – 15/04/2021
Version 1.1 – Zulaikha Kiran – 05/04/2022
Version 1.2 – Husnain, Saiqa Dilawaiz – 06/04/2023
Version 1.3 – Faizan Ahmad – 26/02/2024

Introduction

The purpose of this lab is to revise some MATLAB concepts and implementations including Discrete Fourier Transform (DFT), verify DFT properties, and Inverse DFT.

Course Learning Outcomes

CLO1: Develop algorithms to perform signal processing techniques on digital signals using MATLAB and DSP Kit DSK6713

CLO3: Deliver a report/lab notes/presentation/viva, effectively communicating the design and analysis of the given problem

Equipment

- Software
  - MATLAB

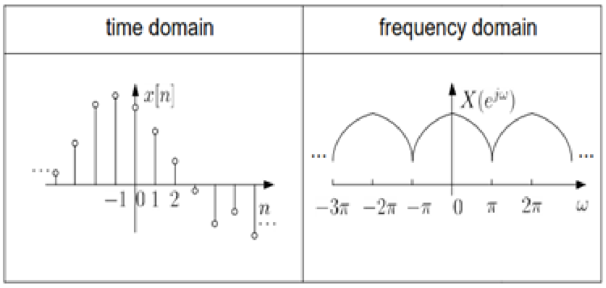
Instructions

1. This is an individual lab. You will perform the tasks individually and submit a report.
2. Some of these tasks are for practice purposes only while others (marked as ‘Exercise’) have to be answered in the report.
3. When asked to display an image/ graph in the exercise either save it as jpeg or take a screenshot, in order to insert it in the report.
4. The report should be submitted on the given template, including:
   a. Code (copy and pasted, NOT a screenshot)
   b. Output values (from command window, can be a screenshot)
   c. Output figure/graph (as instructed in 3)
   d. Explanation where required
5. The report should be properly formatted, with easy to read code and easy to see figures.
6. Plagiarism or any hint thereof will be dealt with strictly. Any incident where plagiarism is caught, both (or all) students involved will be given zero marks, regardless of who copied whom. Multiple such incidents will result in disciplinary action being taken.

<u>Discrete Time Fourier Transform:</u>

DTFT is a frequency analysis of discrete signals. The DTFT of x[n] has been derived using below equation

$$\boldsymbol{X(e^{jw}) = \sum_{n=-inf}^{inf} x[n]e^{-jwn}}$$



DTFT have Discrete time and Countinous frequency domain. DTFT cannot be implemented practically because of continous frequency domain.

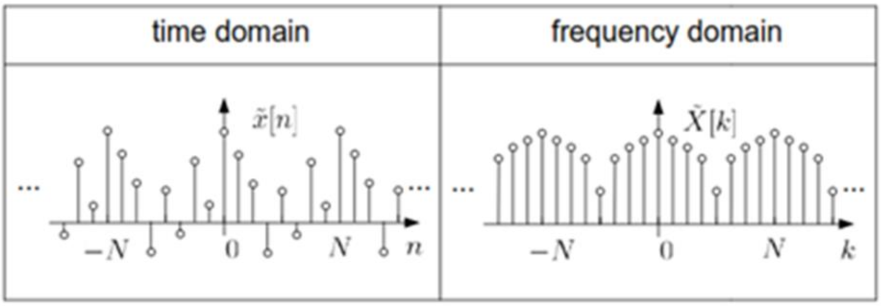
<u>Discrete Fourier Transform:</u>

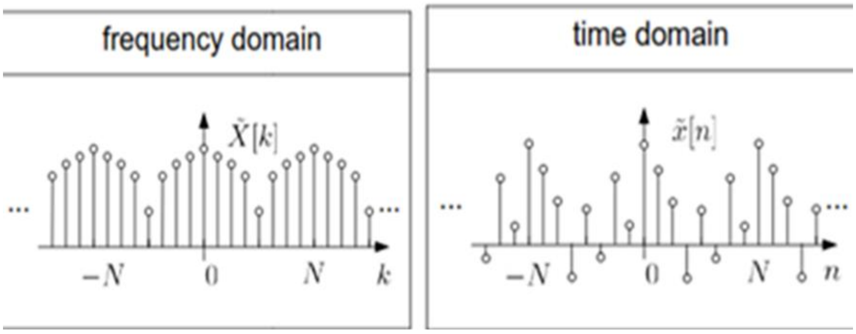
Discrete Fourier Transform (DFT) have Discrete time and Discrete in frequency domain. It can be implemented practically because of discrete frequency domain. DFT is derived from DTFT. DFTs are mainly used in computer based analysis as computers store data in discrete sequences with finite length. You can find the DFT using below formula.

$$\boldsymbol{X[k] = \sum_{n=0}^{N-1} x[n]e^{-j\frac{2\pi}{N}kn}}$$

We can find the inverse fourier transform using this formula:

$$\boldsymbol{X[n] = \frac{1}{N}\sum_{n=0}^{N-1} x[k]e^{j\frac{2\pi}{N}kn}}$$





Lab Exercise

Perform the following tasks.

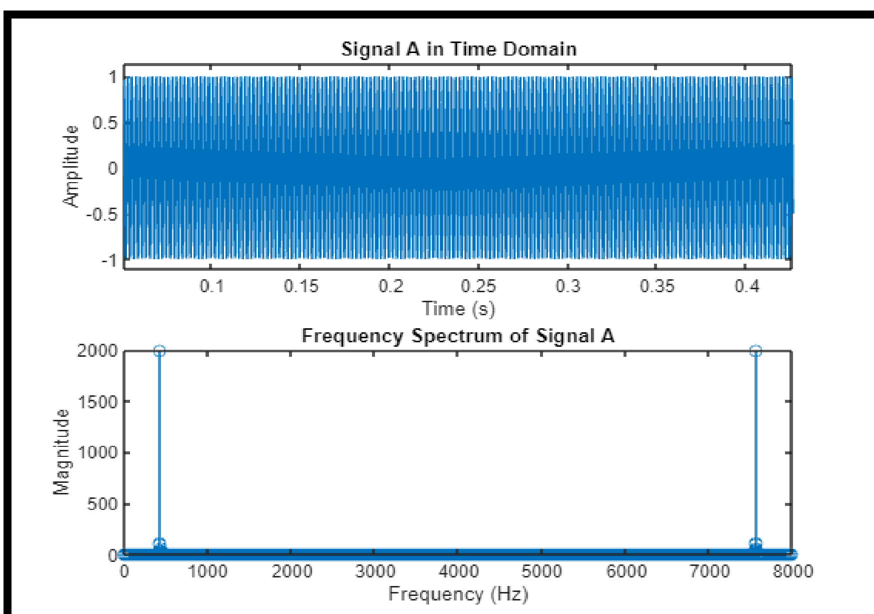
Task 1

Observe the frequency spectrum of the signal A?

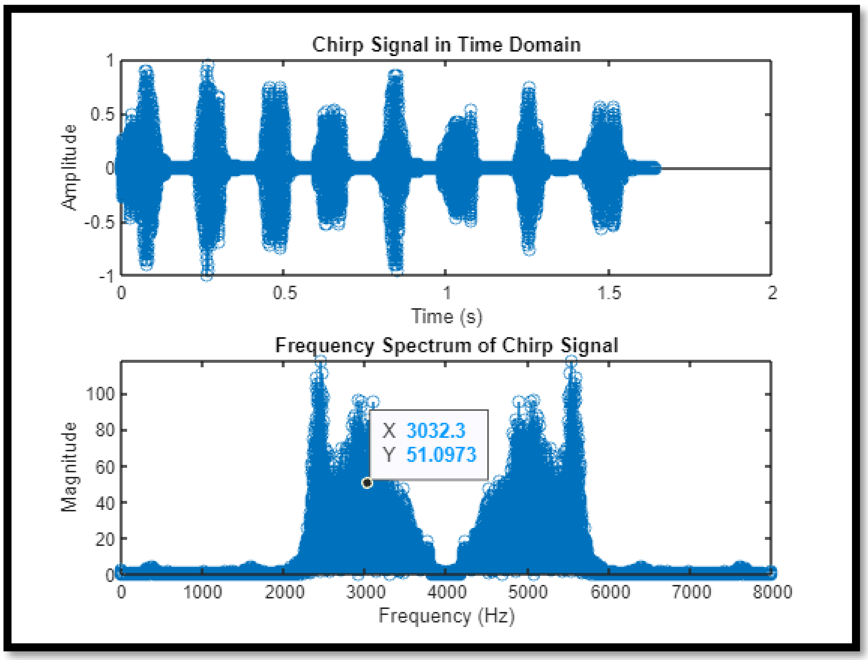
- **t=0:1/8000:0.5;**
- **A=sin(2*pi*440*(t));**

Load chirp.mat and observe the frequency spectrum?

```
% Task 1: Observe the frequency spectrum of signal A
t = 0:1/8000:0.5;
A = sin(2*pi*440*t);

% Plot the time-domain signal
subplot(2, 1, 1);
plot(t, A);
ylim([-1.5 1.5])
title('Signal A in Time Domain');
xlabel('Time (s)');
ylabel('Amplitude');

% Compute and plot the frequency spectrum
A_fft = fft(A);
frequencies = linspace(0, 8000, length(A_fft));
subplot(2, 1, 2);
stem(frequencies, abs(A_fft));
title('Frequency Spectrum of Signal A');
xlabel('Frequency (Hz)');
ylabel('Magnitude');
```



```
% Load chirp.mat and observe its frequency spectrum
load('chirp.mat');
dt = t(2) - t(1); % Define the sampling interval
t_chirp = 0:dt:(length(y)-1)*dt;
chirp_fft = fft(y);
frequencies_chirp = linspace(0, 1/dt, length(chirp_fft));
figure; % Open a new figure window
subplot(2, 1, 1);
stem(t_chirp, y);
title('Chirp Signal in Time Domain');
xlabel('Time (s)');
ylabel('Amplitude');
subplot(2, 1, 2);
stem(frequencies_chirp, abs(chirp_fft));
title('Frequency Spectrum of Chirp Signal');
xlabel('Frequency (Hz)');
ylabel('Magnitude');
```



Explanation:

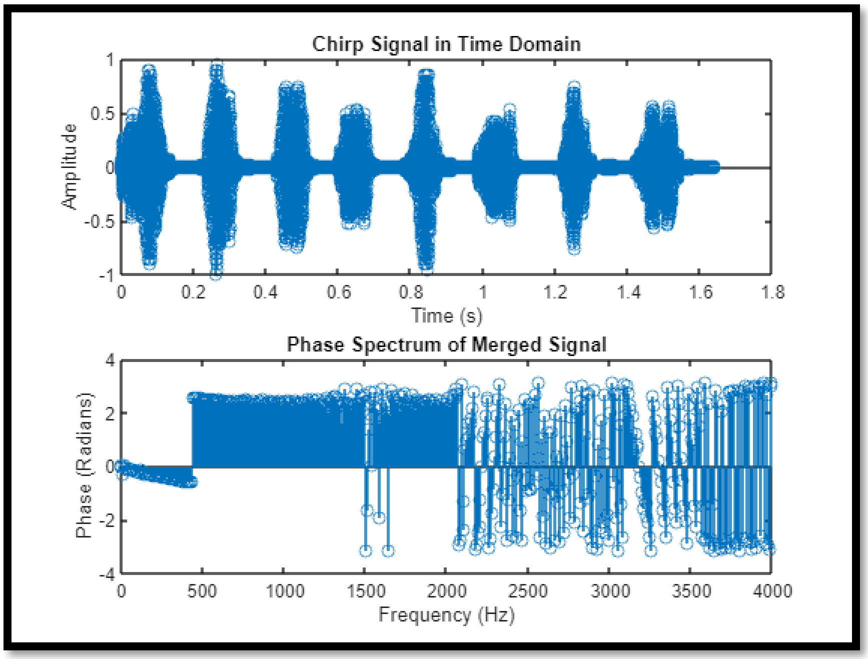
In our digital signal processing lab, I performed the Fourier transform on a signal. This involved converting a time-domain signal into its frequency components to understand its frequency content. I computed the magnitude and phase of the frequency components to analyze the signal's characteristics in the frequency domain.

Task 2

Can we add signal A and chirp signal of Task 2, if yes then see the frequency spectrum of merged signal?

```
% Define the time vector for signal A
t = 0:1/8000:0.5;

% Create signal A
A = sin(2*pi*440*t);

% Load the chirp signal from chirp.mat
load chirp.mat; % Assuming 'y' is the chirp signal

% Convert 'y' from column to row vector if necessary
Z = y';

% Pad the shorter signal with zeros to make both signals the same length
if length(A) > length(Z)
    Z = [Z, zeros(1, length(A) - length(Z))];
elseif length(Z) > length(A)
    A = [A, zeros(1, length(Z) - length(A))];
end

% Add the two signals
mergedSignal = A + Z;
sound(mergedSignal, 8000);

% Define length of the FFT (nfft)
nfft = 1024;

% Compute the Fast Fourier Transform (FFT) of the merged signal
MergedX = fft(mergedSignal, nfft);

% Extract the first half of MergedX for positive frequencies
MergedX = MergedX(1:nfft/2+1);

% Calculate the magnitude of the FFT
mag_MergedX = abs(MergedX);

% Calculate the phase of the FFT
phase_MergedX = angle(MergedX);

% Frequency vector for plotting
f = (0:nfft/2)*8000/nfft;

% Plot the magnitude of the FFT
figure(1);
stem(f, mag_MergedX);
title('Magnitude Spectrum of Merged Signal');
xlabel('Frequency (Hz)');
ylabel('Magnitude');
```

```
% Plot the phase of the FFT
figure(2);
stem(f, phase_MergedX);
title('Phase Spectrum of Merged Signal');
xlabel('Frequency (Hz)');
ylabel('Phase (Radians)');
```



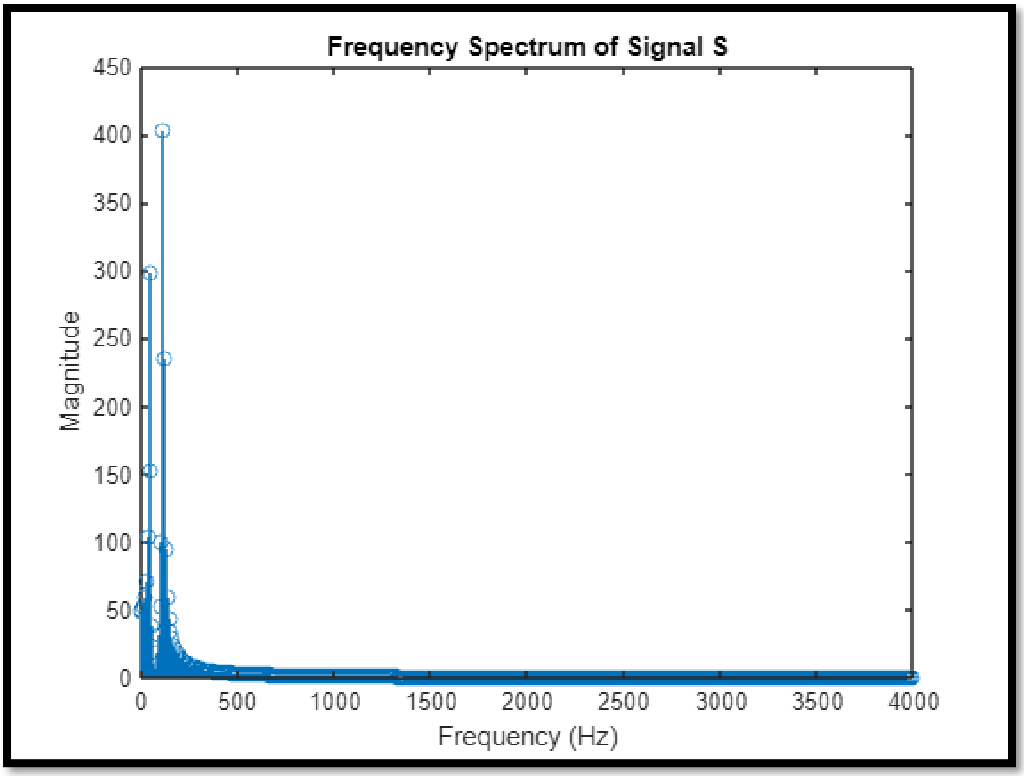
Explanation:

During the lab session, I also worked on converting a frequency-domain signal back to the time domain using the inverse Fourier transform. This allowed me to reconstruct the original time-domain signal from its frequency components. By comparing the original and reconstructed signals, I was able to verify the accuracy of the inverse transformation.

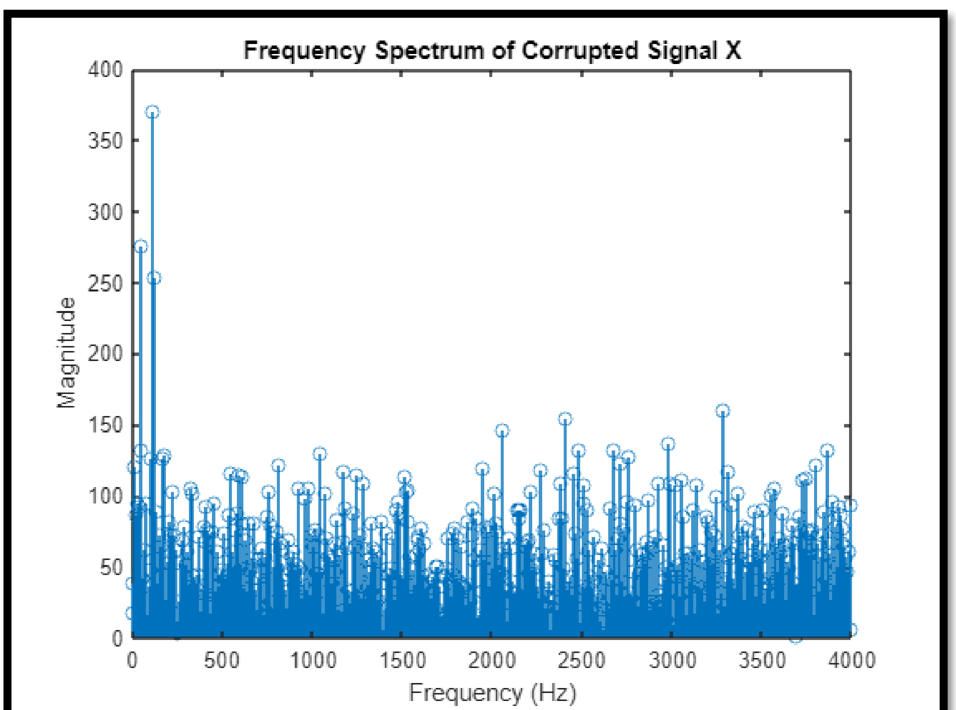
Task 3:

Form a signal S containing a 50 Hz sinusoid of amplitude 0.7 and a 120 Hz sinusoid of amplitude 1. Corrupt the signal by adding noise. X = S + 2*randn(size(t)); Listen both signals and observe the frequency spectrum of both?

```
% Define the time vector for the signals
t = 0:1/8000:1;  % Assuming a duration of 1 second

% Form the signal S containing a 50 Hz sinusoid of amplitude 0.7 and a 120 Hz sinusoid
of amplitude 1
S = 0.7 * sin(2*pi*50*t) + sin(2*pi*120*t);

% Corrupt the signal by adding noise
X = S + 2*randn(size(t));

% Listen to the original signal S
sound(S, 8000);

% Listen to the corrupted signal X
sound(X, 8000);

% Compute the Fast Fourier Transform (FFT) of the original signal S
nfft = 1024;
SX = fft(S, nfft);
mag_SX = abs(SX);
f_SX = (0:nfft/2)*8000/nfft;

% Compute the Fast Fourier Transform (FFT) of the corrupted signal X
X = X(:);  % Ensure X is a column vector
nfft = 1024;
X_fft = fft(X, nfft);
mag_X = abs(X_fft);
f_X = (0:nfft/2)*8000/nfft;

% Plot the frequency spectrum of the original signal S
figure;
stem(f_SX, mag_SX(1:nfft/2+1));
title('Frequency Spectrum of Signal S');
xlabel('Frequency (Hz)');
ylabel('Magnitude');
```




```
% Plot the frequency spectrum of the corrupted signal X
figure;
stem(f_X, mag_X(1:nfft/2+1));
title('Frequency Spectrum of Corrupted Signal X');
xlabel('Frequency (Hz)');
ylabel('Magnitude');
```



Explanation:

In another part of the lab, I experimented with creating a signal containing multiple frequency components and then visualizing its frequency spectrum. This involved plotting the magnitude and phase spectra of the signal to observe the amplitudes and phase shifts of its frequency components.

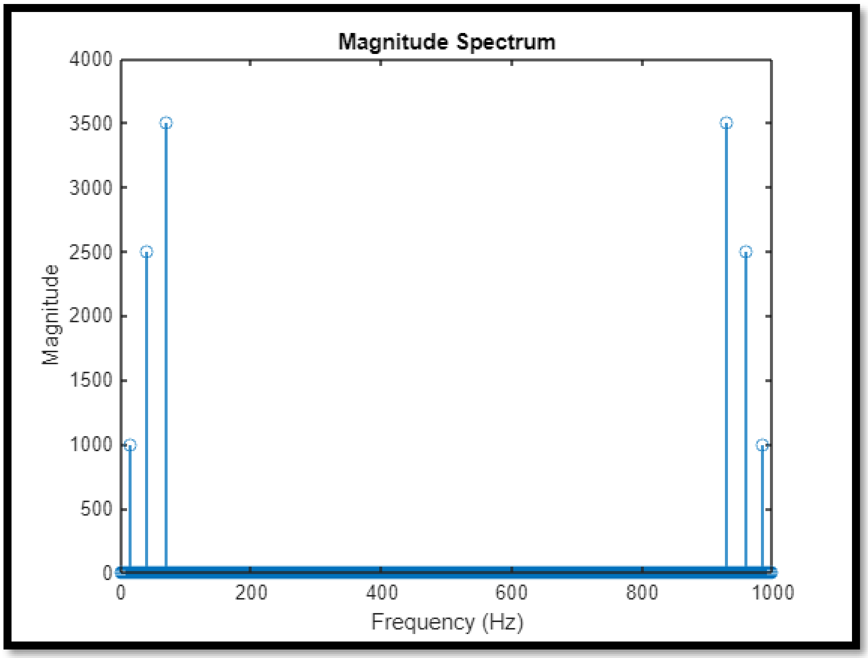
Task 4:

Find the discrete Fourier transform of the following signal (both phase and magnitude plot)?

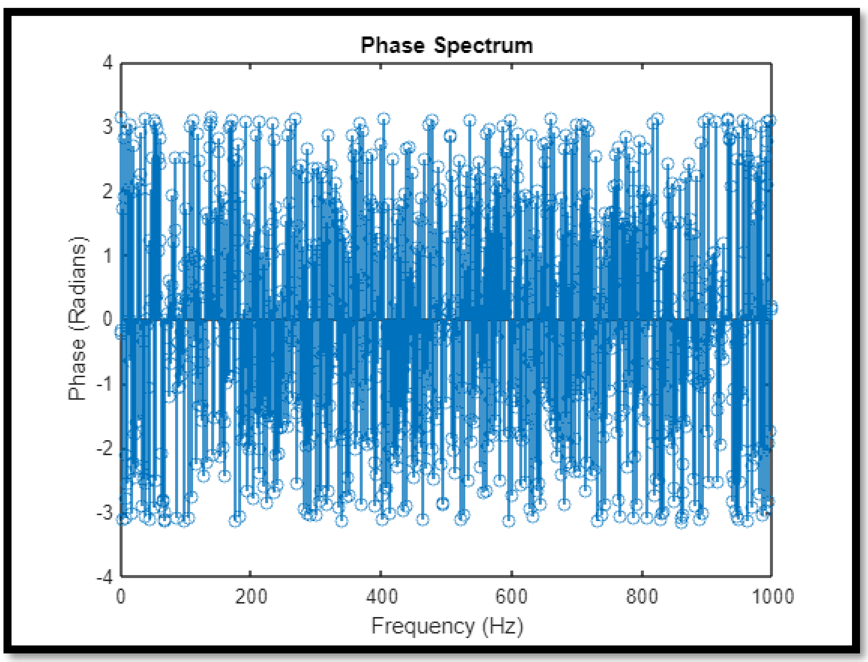
F=2cos(2*pi*15*(t)) + 5cos(2*pi*40*(t)) + 7cos(2*pi*70*(t));

Then convert Frequency domain signal into time domain signal using inverse fourier transform.

```
% Define the sampling frequency and time vector
Fs = 1000; % Sampling frequency in Hz
t = 0:1/Fs:1-1/Fs; % Time vector for 1 second

% Define the signal F
F = 2*cos(2*pi*15*t) + 5*cos(2*pi*40*t) + 7*cos(2*pi*70*t);

% Compute the DFT of the signal F
nfft = length(F); % Number of points in FFT
F_fft = fft(F, nfft); % DFT of signal F

% Compute the magnitude and phase of the DFT
mag_F_fft = abs(F_fft); % Magnitude
phase_F_fft = angle(F_fft); % Phase

% Frequency vector for plotting
f = (0:nfft-1)*Fs/nfft;

% Plot the magnitude spectrum
figure(1);
stem(f, mag_F_fft);
title('Magnitude Spectrum');
xlabel('Frequency (Hz)');
ylabel('Magnitude');
```

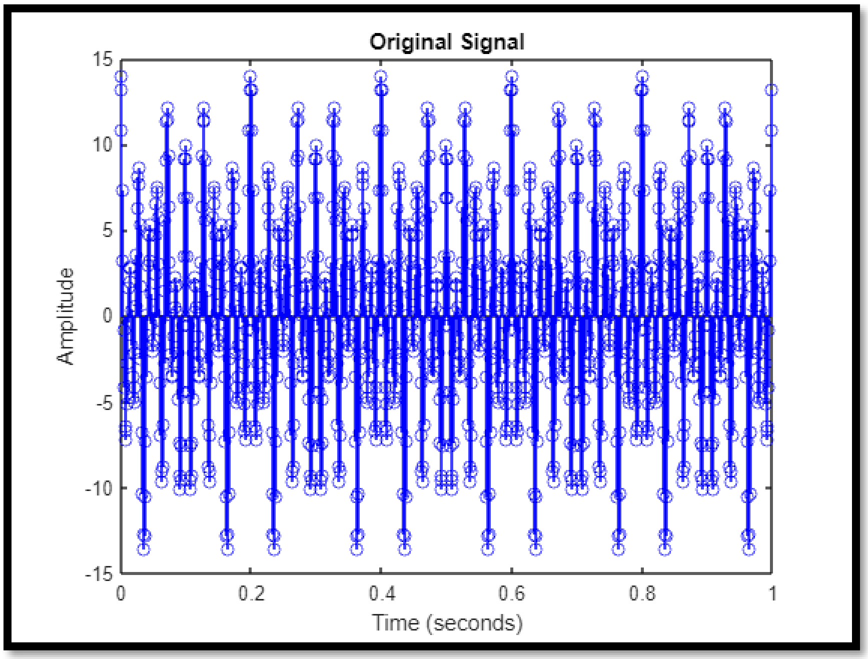



```
% Plot the phase spectrum
figure(2);
stem(f, phase_F_fft);
title('Phase Spectrum');
xlabel('Frequency (Hz)');
ylabel('Phase (Radians)');
```

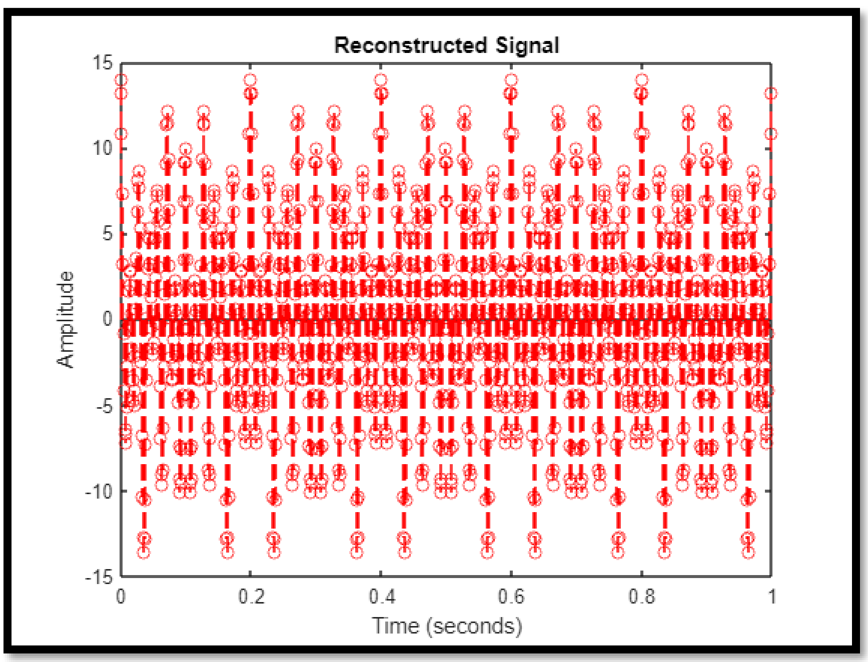



```
% Convert the frequency domain signal back to time domain using IFT
F_ift = ifft(F_fft, nfft);

% Plot the original and reconstructed signals to compare
figure(3);
```

```
stem(t, F, 'b');
title('Original Signal');
xlabel('Time (seconds)');
ylabel('Amplitude');
```




```
figure(4);
stem(t, F_ift, 'r--');
title('Reconstructed Signal');
xlabel('Time (seconds)');
ylabel('Amplitude');
```



Explanation:

In the lab, I dived into the world of Fourier transforms. These magical mathematical tools allowed me to dissect signals, revealing their hidden frequency components. First, I transformed time-domain signals into their frequency-domain counterparts, analyzing magnitudes and phases. Then, I explored the inverse Fourier transform, reconstructing original signals from their frequency components. Visualizing frequency spectra was my next feat-plotting amplitudes and phase shifts.

Conclusion:

Conclusion: Overall, the lab session provided valuable hands-on experience with Fourier transforms and their applications in digital signal processing. I gained a deeper understanding of how signals can be analyzed and manipulated in both the time and frequency domains. By performing these tasks, I developed practical skills in signal processing and gained insight into the importance of understanding the frequency content of signals for various engineering applications.

Evaluation Rubric

- **Method of Evaluation**: In-lab marking by instructors, Report submitted by students
- **Measured Learning Outcomes**:

CLO1: Develop algorithms to perform signal processing techniques on digital signals using MATLAB and DSP Kit DSK6713

CLO3: Deliver a report/lab notes/presentation/viva, effectively communicating the design and analysis of the given problem

	Excellent 10	Good 9-7	Satisfactory 6-4	Unsatisfactory 3-1	Poor 0	Marks Obtained
Tasks (CLO1)	All tasks completed correctly. Correct code with proper comments.	Most tasks completed correctly.	Some tasks completed correctly.	Most tasks incomplete or incorrect.	All tasks incomplete or incorrect.	
Output (CLO1)	Output correctly shown with all Figures/Plots displayed as required and properly labelled	Most Output/Figures/Plots displayed with proper labels	Some Output/Figures/Plots displayed with proper labels OR Most Output/Figures/Plots displayed but without proper labels	Most of the required Output/Figures/Plots not displayed	Output/Figures/Plots not displayed	
Answers (CLO1)	Meaningful answers to all questions. Answers show the understanding of the student.	Meaningful answers to most questions.	Some correct/ meaningful answers with some irrelevant ones	Answers not understandable/ not relevant to questions	Not Written any Answer	
Report (CLO3)	Report submitted with proper grammar and punctuation with proper conclusions drawn and good formatting	Report submitted with proper conclusions drawn with good formatting but some grammar mistakes OR proper grammar but not very good formatting	Some correct/ meaningful conclusions. Some parts of the document not properly formatted or some grammar mistakes	Conclusions not based on results. Bad formatting with no proper grammar/punctuation	Report not submitted	
Total						