الجامعة الإسلامية العالمية شيتاغونغ

International Islamic University Chittagong

Department of Computer and Communication Engineering

LAB REPORT

Experiment No : 03

Experiment Name: Discrete Time Fourier Transform (DTFT) computation.

Course Title : Digital Signal Processing Sessional

Course Code : CCE-3602

Submitted By

Name : Ahsanul Karim Tanim

ID No. : E221013

Semester : 6th

Section : A

Date of Experiment:

Date of Submission:

Remark

Submitted To

Dr. Engr. Abdul Gafur

Associate Professor, Dept. of ETE, IIUC

Experiment No: 03
Experiment Name: Discrete Time Fourier Transform (DTFT) computation.
Apparatus:

- ❖ Computer with MATLAB software.

Theory: The DTFT values of a real sequence, represented as a rational function, can be determined using the following software. The input data required by the application includes the number of frequency points k at which the DTFT will be calculated, the vectors **NUM** and **DEN** containing the coefficients of the numerator and denominator, respectively, with the denominator expressed in ascending powers of z−1. These vectors must be enclosed in square brackets. After computing the DTFT at the specified frequency points, the software displays the real and imaginary parts, along with the magnitude and phase spectra. It is important to note that the DTFT of a real sequence can only be evaluated at k evenly spaced frequency values due to its symmetry properties.

$$X\left(e^{j\omega}\right)=\frac{0.008-0.033e^{-j\omega}+0.05e^{-j2\omega}-0.033e^{-j3\omega}+0.008e^{-j4\omega}}{1+2.37e^{-j\omega}+2.7e^{-j2\omega}+1.6e^{-j3\omega}+0.41e^{-j4\omega}}$$

Program:

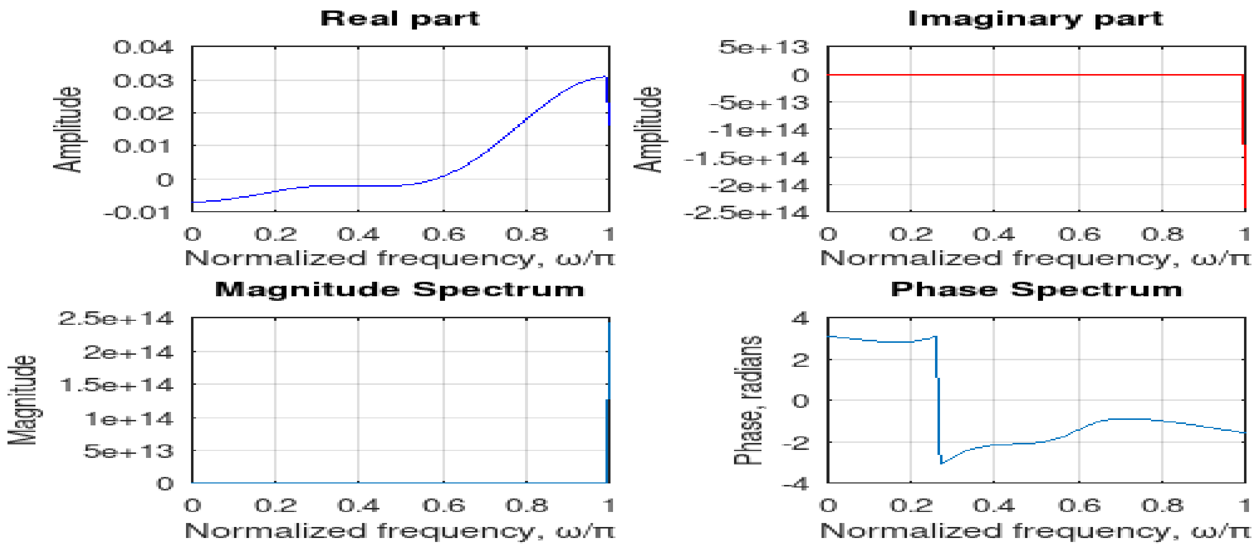
```
clc;
close all;
clear all;
k = input('Number of frequency points = ');
num = input('Numerator coefficients = ');
den = input('Denominator coefficients = ');
w = 0:pi/k:pi;
h = freqz(num, den, w);

subplot(2,2,1);
plot(w/pi, real(h), 'b');  % 'b' for blue color
grid on;
title('Real part');
xlabel('Normalized frequency, \omega/\pi');
ylabel('Amplitude');

subplot(2,2,2);
plot(w/pi, imag(h), 'r');  % 'r' for red color
grid on;
title('Imaginary part');
xlabel('Normalized frequency, \omega/\pi');
ylabel('Amplitude');

subplot(2,2,3);
plot(w/pi, abs(h));
grid on;
title('Magnitude Spectrum');
xlabel('Normalized frequency, \omega/\pi');
ylabel('Magnitude');

subplot(2,2,4);
plot(w/pi, angle(h));
```

```
grid on;
title('Phase Spectrum');
xlabel('Normalized frequency, \omega/\pi');
ylabel('Phase, radians');
```

Command Window

Number of frequency points = 100
Numerator coefficients = [-0.01, 0.005, -0.008, 0.003, -0.004]
Denominator coefficients = [1, 1]

Output



Discussion: In this lab, we computed and visualized the Discrete-Time Fourier Transform (DTFT) of a real sequence represented as a rational function. The software allowed us to calculate the DTFT at multiple frequency points, based on user-defined numerator and denominator coefficients. We visualized the real and imaginary components, as well as the magnitude and phase spectra. These plots provided insight into the frequency characteristics of the sequence. The lab emphasized the importance of symmetry in the DTFT of real sequences and demonstrated how to analyze and interpret the frequency content of discrete-time signals.

International Islamic University Chittagong

Department of Computer and Communication Engineering

LAB REPORT

Experiment No : 04

Experiment Name: Discrete Fourier Transform (DFT) & Inverse DFT (IDFT) Computation.

Course Title : Digital Signal Processing Sessional

Course Code : CCE-3602

Submitted By

Name : Ahsanul Karim Tanim

ID No. : E221013

Semester : 6th

Section : A

Date of Experiment:

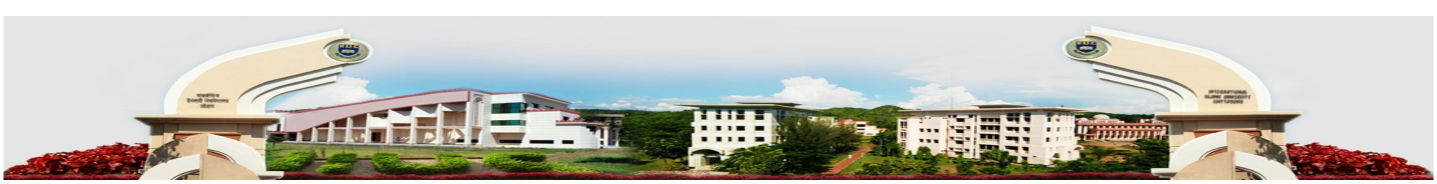
Date of Submission:

Submitted To

Dr. Engr. Abdul Gafur

Associate Professor, Dept. of ETE, IIUC

Remark



Experiment No: 04
Experiment Name: Discrete Fourier Transform (DFT) & Inverse DFT (IDFT) Computation.

Apparatus:

- Computer with MATLAB software

Theory : A time-domain sequence can be transformed into a continuous function of a frequency variable using the discrete-time Fourier transform (DTFT). The parent discrete-time sequence may be easily determined by computing its Fourier series representation due to the periodicity of the DTFT. The frequency-domain representation of a Length-N sequence may be described by N evenly-spaced samples of its DTFT. From these N frequency samples, the original N samples of the discrete-time sequence can be retrieved via a straightforward inverse procedure. The discrete Fourier transform (DFT) of a Length-N sequence is made up of these N frequency samples. The simplest relation between a finite length sequence x[n], defined for $0 \leq n \leq N-1$, and its DTFT X (−) is obtained by uniformly sampling X (−) on the ω- axis between $0 \leq \omega \leq 2\pi$ at ωk 2π

$$X[k] = X(e^{j\omega})\Big|_{\omega=2\pi k/N} = \sum_{n=0}^{N-1} x[n]e^{-2\pi nk/N} \qquad k = 0,1,\ldots,N-1$$

We determine the M-point DFT U[k] of the following N-point sequence

Program:

```
clc;
close all;
clear all;

% Input length of the time-domain sequence
and DFT length
N = input('Length of the sequence = ');
M = input('Length of the DFT = ');

% Define the time-domain sequence (unit step
sequence)
u = ones(1, N); % Sequence of ones of length N

% Compute the DFT using fft() and zero-
padding to M points
U = fft(u, M);

% Time index for the original sequence
t = 0:1:N-1;

% Plot the original time-domain sequence
subplot(2,2,1);
stem(t, u);
axis([-1 N 0 1.2]);
title('Original Time-Domain Sequence');
xlabel('Time index n');
ylabel('Amplitude');
```

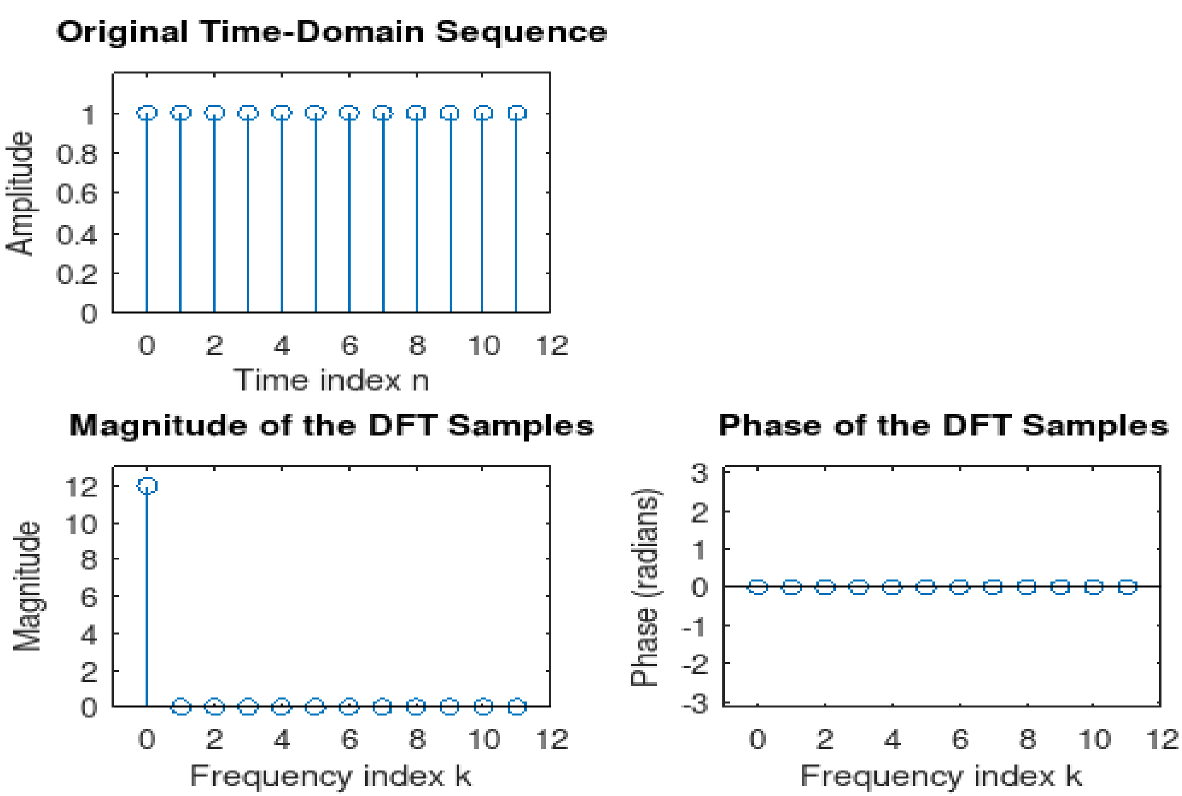
```
% Frequency index for the DFT samples
k = 0:1:M-1;

% Plot the magnitude of the DFT samples
subplot(2,2,3);
stem(k, abs(U));
axis([-1 M 0 max(abs(U)) + 1]);
title('Magnitude of the DFT Samples');
xlabel('Frequency index k');
ylabel('Magnitude');

% Plot the phase of the DFT samples
subplot(2,2,4);
stem(k, angle(U));
axis([-1 M -pi pi]);
title('Phase of the DFT Samples');
xlabel('Frequency index k');
ylabel('Phase (radians)');
```

Command Window :

Length of the sequence = 12

Length of the DFT = 12

Output:



Program(IDTF):

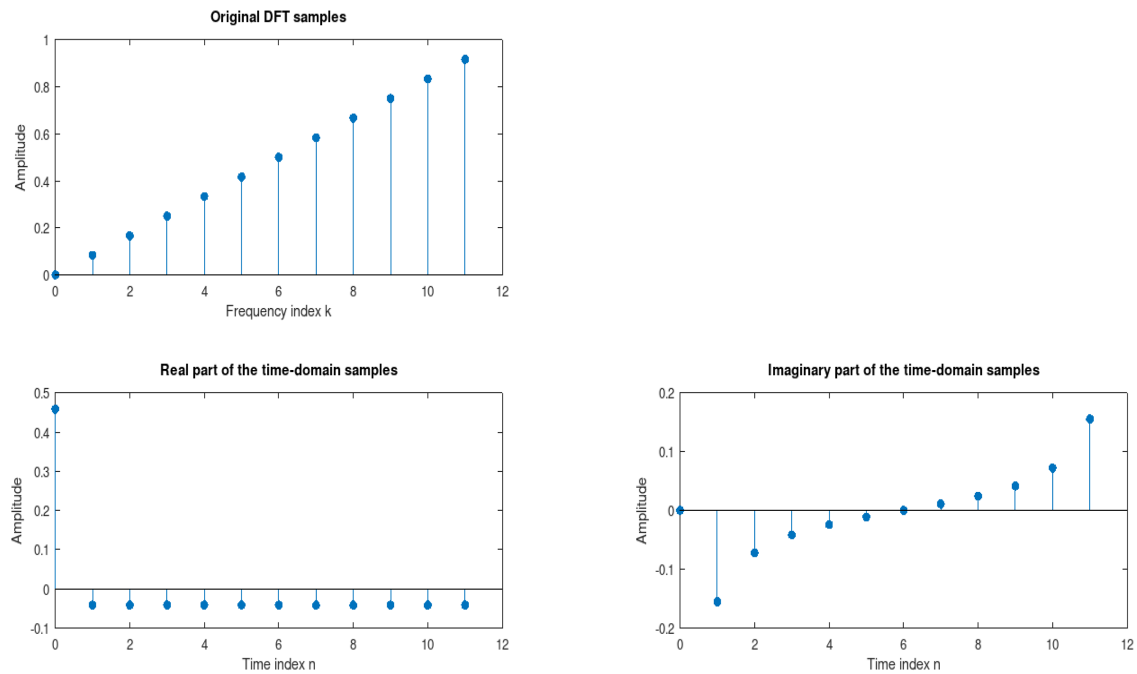
```
clc;
close all;
clear all;
% Input the length of the DFT (K) and the IDFT (N)
K = input('Length of DFT = ');
N = input('Length of IDFT = ');

% Define frequency indices for DFT samples
k = 0:K-1;  % Frequency indices (0 to K-1)
U = (k) / K;  % DFT samples (simple example)

% Compute the Inverse DFT (IDFT) using ifft() function
u = ifft(U, N);

% Plot the DFT samples (real values)
subplot(2,2,1);
stem(k, U, 'filled');
xlabel('Frequency index k');
ylabel('Amplitude');
title('Original DFT samples');

% Plot the real part of the time-domain samples (IDFT result)
subplot(2,2,3);
n = 0:N-1;  % Time index
stem(n, real(u), 'filled');
title('Real part of the time-domain samples');
xlabel('Time index n');
ylabel('Amplitude');

subplot(2,2,4); %Result
stem(n, imag(u), 'filled');
title('Imaginary part of the time-domain samples');
xlabel('Time index n');
ylabel('Amplitude');
```

Command Window :

Length of DFT = 12

Length of IDFT = 12

Output:



Discussion: In this lab, we studied the Discrete Fourier Transform (DFT) and its inverse (IDFT). We computed the IDFT of a set of DFT samples using the ifft() function in GNU Octave. The process involved visualizing both the frequency-domain samples and the reconstructed time-domain sequence. This demonstrated the reversible nature of the DFT, where we can recover the original sequence from its frequency representation. The lab helped us understand the relationship between the time and frequency domains in signal processing.