Institute of Information Technology (IIT)

Jahangirnagar University

Lab Report: 04

Course Code: ICT-4104

<u>Submitted by:</u>

Name: Md. Shakil Hossain

Roll No: 2023

Lab Date: 10.08.2023

Submission Date: 14.08.2023

EXPERIMENT NO: 04

NAME OF EXPERIMENT:
To find the FFT of a given sequence

AIM
To find the FFT of a given sequence.

APPARATUS
Software: MATLAB

THEORY
DFT of a sequence

$$X_k = \sum_{n=0}^{N-1} x_n \cdot e^{-\frac{i2\pi}{N}kn}$$

Where N= Length of sequence.

K= Frequency Coefficient.

n = Samples in time domain.

FFT: -Fast Fourier transform.

There are two methods.

1. Decimation in time (DIT) FFT.
2. Decimation in Frequency (DIF) FFT.

The number of multiplications in DFT $=N^2$. The number of Additions in DFT = N (N-1).

On the other hand, The no of multiplication in FFT= $N/2 \log_2 N$ and The number of additions = $N \log_2 N$.

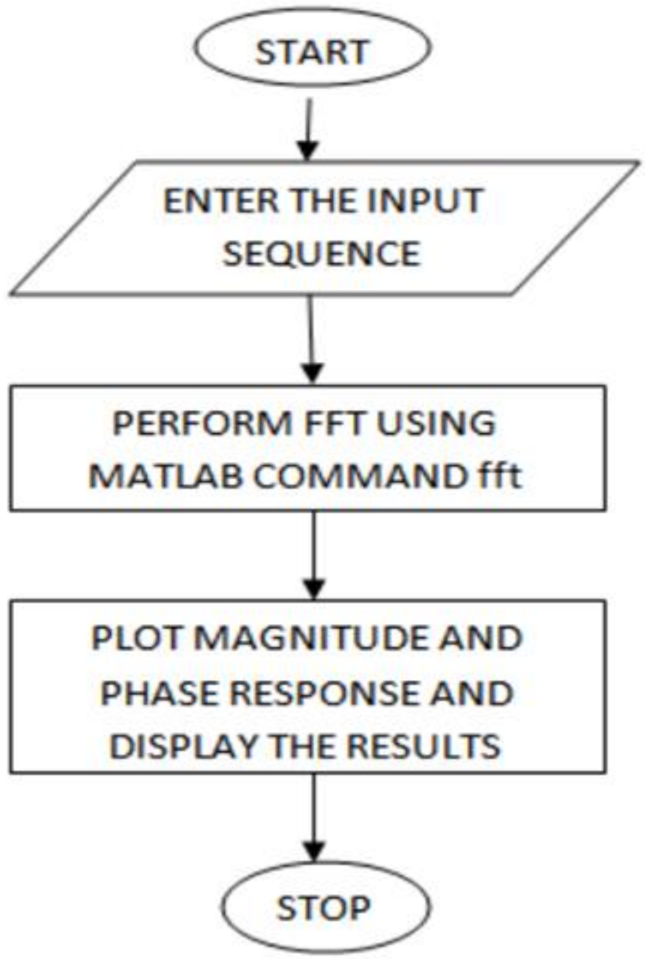
To reduce complexity and calculation we prefer FFT rather than DFT

METHODOLOGY

Algorithm:

Step I: Give input sequence x[n].
Step II: Find the length of the input sequence using length command.
Step III: Find FFT and IFFT using matlab commands FFT and IFFT.
Step IV: Plot magnitude and phase response
Step V: Display the results

Flow Chart:

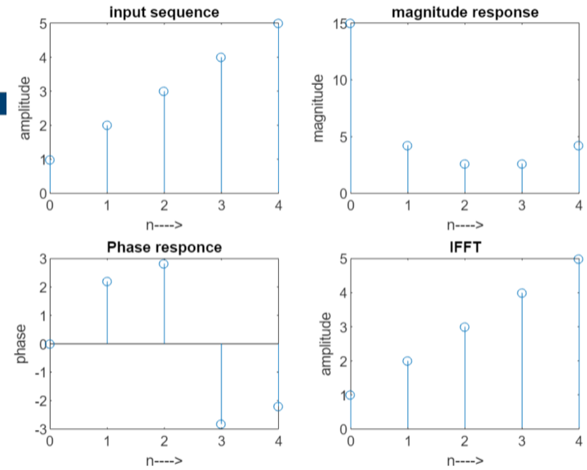


Program:

```
clc;
clear;
close all;
x=input('Enter the sequence : ');
N=length(x);
xK=fft(x,N);
xn=ifft(xK);
n=0:N-1;

subplot (2,2,1);
stem(n,x);
xlabel('n---->');
ylabel('amplitude');
title('input sequence');

subplot (2,2,2);
stem(n,abs(xK));
xlabel('n---->');
ylabel('magnitude');
title('magnitude response');

subplot (2,2,3);
stem(n,angle(xK));
xlabel('n---->');
ylabel('phase');
title('Phase responce');

subplot (2,2,4);
stem(n,xn);
xlabel('n---->');
ylabel('amplitude');
title('IFFT');

disp('FFT Sequence (xK):');
disp(xK);
disp('IFFT Sequence (xn):');
disp(xn);
```

Output:

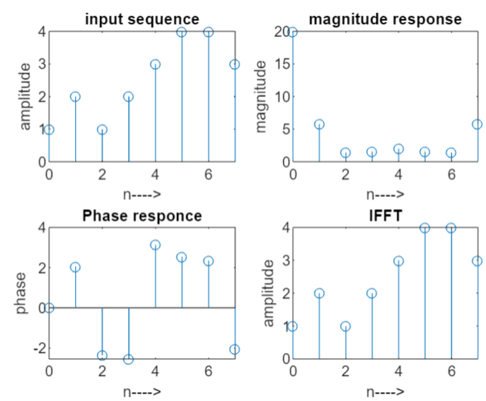
```
Command Window
Enter the sequence :
[1 2 3 4 5]

x =

     1     2     3     4     5


N =

     5


xK =

  15.0000 + 0.0000i  -2.5000 + 3.4410i  -2.5000 + 0.8123i  -2.5000 - 0.8123i  -2.5000 - 3.4410i


xn =

     1     2     3     4     5

>>
```



Exercise 1:1. Find 8-point DFT of sequence x(n)=[1 2 1 2 3 4 4 3] using the FFT algorithm.

Output:

```
Enter the sequence :
[1 2 1 2 3 4 4 3]

x =

     1     2     1     2     3     4     4     3


N =

     8


xK =

  Columns 1 through 6

  20.0000 + 0.0000i  -2.7071 + 5.1213i  -1.0000 - 1.0000i  -1.2929 - 0.8787i  -2.0000 + 0.0000i  -1.2929 + 0.8787i

  Columns 7 through 8

  -1.0000 + 1.0000i  -2.7071 - 5.1213i


xn =

    1.0000    2.0000    1.0000    2.0000    3.0000    4.0000    4.0000    3.0000
```



DISCUSSION

Hand calculation:

Lab problem

Given sequence,

$$x(n) = [1 \quad 2 \quad 3 \quad 4 \quad 5]$$

DFT expression is,

$$X(k) = \sum_{n=0}^{N-1} x(n)\, e^{-j2\pi nk/N}$$

when,

$$k=0, \quad X(0) = \sum_{n=0}^{4} x(n)\, e^{0}$$

$$= x(0) + x(1) + x(2) + x(3) + x(4)$$

$$= 1+2+3+4+5 = 15$$

$$k=1, \quad X(1) = \sum_{n=0}^{4} x(n)\, e^{-j2\pi n/5}$$

$$= x(0) + x(1)\cdot e^{-j2\pi/5} + x(2)\, e^{-j4\pi/5} + x(3)\, e^{-j6\pi/5} + x(4)\, e^{-j8\pi/5}$$

$$= -2.5 + 3.45j$$

$$k=2, \quad X(2) = \sum_{n=0}^{4} x(n)\cdot e^{+4\pi nj/5}$$

$$= x(0) + x(1)\cdot e^{-4\pi j/5} + x(2)\, e^{-8\pi j/5} + x(3)\, e^{-12\pi j/5} + x(4)\, e^{-16\pi j/5}$$

$$= -2.5 + .812j$$

$k=3,\ X(3) = \sum_{n=0}^{4} x(n).e^{-6\pi n j/5} = 1 + 2e^{-j2\pi/N \cdot 3} + 3e^{-j2\pi/N\, 3\times 2} + 4e^{-j2\pi/N\, 9} + 5e^{-j2\pi/N\, 12}$

$= -2.5 - .81j$

$k=4,\ X(4) = 1 + 2e^{-j2\pi/N \cdot 4} + 3e^{-j.2\pi/N \cdot 8} + 4e^{-j.2\pi/N\, 12} + 5e^{-j2\pi/N \cdot 16}$

$= -2.5 - 3.44j$

$\therefore X(k) = \{15,\ -2.5+3.4407j,\ -2.5+.81j,\ -2.5-.81j,\ -2.5-3.44j\}$

<u>Ans</u>

again find Idft for convert frequency to sequence:-

$X(k) = \{15, -2.5+3.45j, -2.5+.81j, -2.5-.81j, -2.5-3.4j\}$

Idft expression is,

$x(n) = 1/N \cdot \sum_{n=0}^{N-1} X(k).e^{j2\pi nk/N}$; $n = 0, 1, \ldots N-1$

$n=0,\ x(0) = 1/5 \sum_{0}^{4} X(k).e^{0}$

$= 1/5 \cdot 5 = 1$

similarly $x(1) = 2,\ x(2) = 3,\ x(3) = 4,\ x(4) = 5$

$\therefore x(n) = [1\ 2\ 3\ 4\ 5]$ <u>Ans</u>

The experiment focused on analyzing the Fast Fourier Transform (FFT) of a given sequence, a fundamental algorithm in signal processing and scientific applications that converts time-domain signals into frequency-domain representations. This discussion highlights key findings: FFT effectively reveals frequency components.

The experiment yielded noteworthy results as the manually calculated Discrete Fourier Transform (DFT) outcomes for the given sequence closely aligned with the results obtained using MATLAB's FFT function. This alignment between hand calculation and computational implementation provides a strong validation of both the DFT methodology and the accuracy of the FFT algorithm.

CONCLUSION

In conclusion, the experiment successfully demonstrated the application of the Fast Fourier Transform in analyzing the frequency content of a given sequence. The FFT proved to be a powerful technique for uncovering the underlying periodicities and oscillations within a signal. The obtained frequency-domain representation has significant implications in signal processing, communications, and various scientific disciplines. Despite its limitations, the FFT remains an essential tool for understanding and manipulating signals in the frequency domain. Further exploration and refinement of the experiment's parameters could provide deeper insights into the capabilities and constraints of the FFT algorithm.

VIVA QUESTIONS

1. Define transform. What is the need for transformation?

In digital signal processing, a transform is a mathematical tool that changes how we view a signal, often making it easier to analyze or achieve specific goals. Transforms are essential because they enable us to understand a signal's frequency content, compress data, remove noise, enhance features, analyze time-frequency patterns, recognize patterns, design filters, and perform efficient calculations.

2. Differentiate Fourier transform and discrete Fourier transform.

Key Differences:

1. Fourier Transform (FT) is for continuous-time signals and provides a continuous frequency spectrum.
2. Discrete Fourier Transform (DFT) is for discrete-time signals (samples) and provides a discrete frequency spectrum.
3. FT uses integrals for continuous functions, while DFT uses summations for discrete data.
4. FT is for analog signals, and DFT is for digital signals.
5. DFT is efficiently calculated using algorithms like FFT for digital processing.

3. Differentiate DFT and DTFT.

The Discrete Fourier Transform (DFT) and Discrete-Time Fourier Transform (DTFT) are distinct tools in signal processing:

DFT: Analyzes discrete sequences, producing a finite set of frequency components in a periodic spectrum. Useful for digital signals and computations like audio analysis and image processing. FFT is an efficient DFT implementation.

DTFT: Analyzes continuous signals after discrete sampling, yielding a continuous frequency spectrum. It extends infinitely in time, revealing detailed frequency characteristics. Essential for understanding continuous signals in applications like analog-to-digital conversion and signal analysis.

4. What are the advantages of FFT over DFT?

The advantages of the Fast Fourier Transform (FFT) over the Discrete Fourier Transform (DFT) include significantly faster computation, practical real-world application due to speed, lower memory usage, optimized algorithms for efficient processing, and widespread adoption across diverse fields.

5. Differentiate DIT-FFT and DIF-FFT algorithms.

The key difference between DIT-FFT and DIF-FFT lies in the domain where butterfly operations are performed: DIT-FFT uses butterfly operations in the time domain and starts with smaller subsequences, while DIF-FFT uses butterfly operations in the frequency domain and starts with smaller frequency components. Both algorithms aim to optimize the calculations and reduce redundancy, achieving computational efficiency.

6. What is meant by radix?

"Radix" refers to the base or symbols used in a numeral system. It determines how values are represented. In FFT, it can relate to sub-operations in algorithms like radix-2 or radix-4.

7. What is meant by twiddle factor and give its properties?

A twiddle factor is a complex exponential used in the Discrete Fourier Transform and FFT algorithms. Its properties, including symmetry, multiplicative behavior, unity roots, and phase shift, simplify calculations by reducing redundancy. Twiddle factors are precomputed and play a crucial role in optimizing the efficiency of signal transformation between time and frequency domains.

8. How is FFT useful to represent a signal?

FFT is invaluable for signal representation as it dissects signals into their frequency components, offering insights into their composition. The resulting frequency spectrum provides a visual depiction of signal patterns, aids noise identification, and supports feature extraction like dominant frequencies. It efficiently processes signals through quick filtering and mathematical operations in the frequency domain, enhancing computational efficiency. Additionally, FFT contributes to data compression by representing signals succinctly based on their frequency attributes. In diverse fields from audio analysis to image processing, FFT facilitates effective signal understanding, processing, and compact representation, making it a cornerstone of modern signal processing.

9. Compare FFT and DFT with respect to the number of calculations required?

FFT (Fast Fourier Transform) and DFT (Discrete Fourier Transform) differ significantly in calculation complexity:

> FFT: Requires fewer calculations due to optimized algorithms like Cooley-Tukey. Computational complexity is O(N log N), efficient for large datasets.
>
> DFT: Directly calculates all frequency components, leading to O(N^2) complexity. Inefficient for larger data, less practical in real-time or resource-intensive applications.

In summary, FFT's algorithmic optimizations greatly reduce the number of calculations, making it superior for efficient frequency analysis compared to DFT.

10. How is the original signal reconstructed from the FFT of a signal?

The original signal can be reconstructed from the FFT by applying the Inverse Fast Fourier Transform (IFFT). IFFT converts frequency-domain components back to the time domain, recreating the original signal. Proper phase and magnitude information is crucial for accurate reconstruction.

REFERENCE

[1] tutorialspoint,"dsp fast fourier transform". [Online]. Available: https://www.tutorialspoint.com/digital_signal_processing/dsp_fast_fourier_transform.htm [Accessed: august. 13, 2023,09:10am]

[2] "Fast Fourier transform -Algorithms for Competitive Programming," *Cp-algorithms.com*,2022.Available:https://cpalgorithms.com/algebra/fft.html?fbclid=IwAR3nRgPlxN8IhhAcxx6aZ5EkV2nBL4vc7lpPr_byB2deUzn99g0eI4zJqBA#improved-implementation-in-place-computation. [Accessed: Aug. 13, 2023]