Experiment No. 4

Aim: Write a MATLAB Program to find DFT of the sequence and Plot Magnitude & Phase Plot.

Tool: Matlab

Theory:

Discrete Fourier Transform (DFT) converts a finite sequence of equally-spaced samples of a function into a same-length sequence of equally-spaced samples of the discrete-time Fourier transform (DTFT), which is a complex-valued function of frequency. The interval at which the DTFT is sampled is the reciprocal of the duration of the input sequence. An inverse DFT is a Fourier series, using the DTFT samples as coefficients of complex sinusoids at the corresponding DTFT frequencies. It has the same sample-values as the original input sequence. The DFT is therefore said to be a frequency domain representation of the original input sequence. If the original sequence spans all the non-zero values of a function, its DTFT is continuous (and periodic), and the DFT provides discrete samples of one cycle. If the original sequence is one cycle of a periodic function, the DFT provides all the non-zero values of one DTFT cycle.

The DFT is the most important discrete transform, used to perform Fourier analysis in many practical applications. In digital signal processing, the function is any quantity or signal that varies over time, such as the pressure of a sound wave, a radio signal, or daily temperature readings, sampled over a finite time interval (often defined by a window function). In image processing, the samples can be the values of pixels along a row or column of a raster image. The DFT is also used to efficiently solve partial differential equations, and to perform other operations such as convolutions or multiplying large integers.

Since it deals with a finite amount of data, it can be implemented in computers by numerical algorithms or even dedicated hardware. These implementations usually employ efficient fast Fourier transform (FFT) algorithms; so much so that the terms "FFT" and "DFT" are often used interchangeably. Prior to its current usage, the "FFT" initialism may have also been used for the ambiguous term "finite Fourier transform".

Input Parameter:

Enter the Sequence for which DFT is to be calculated [1 2 3 4 5]

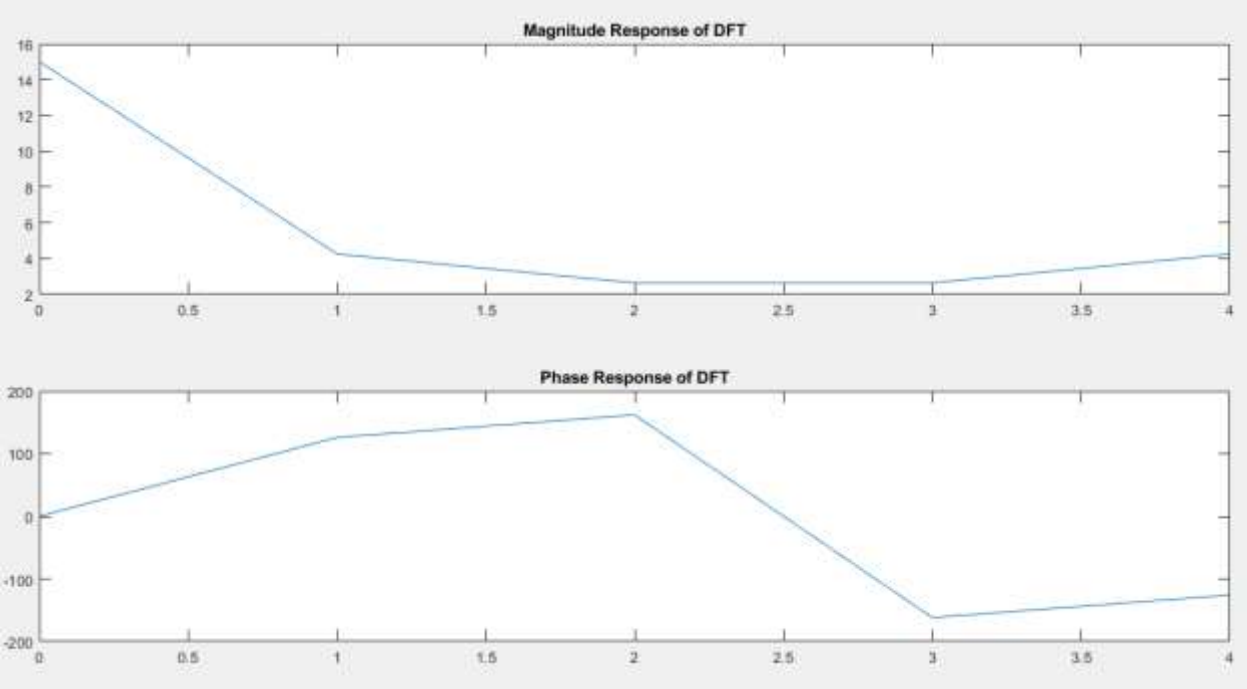
```
  15.0000 + 0.0000i  -2.5000 + 3.4410i  -2.5000 + 0.8123i  -2.5000 - 0.8123i  -2.5000 - 3.4410i
```

Code:

```
% To find DFT of the sequence and Plot Magnitude and Phase Response
x=input('Enter the Sequence for which DFT is to be calculated');
N=length(x);
n=[0:1:N-1];
k=[0:1:N-1];
nk=n'*k;
WN=exp(-1j*2*pi/N);
WNnk=WN.^nk;
% DFT of x is Xk
Xk=x*WNnk;
```

```
disp(Xk);
MagX=abs(Xk);
PhaseX=angle(Xk)*180/pi;
subplot(2,1,1);
plot(k,MagX);
title('Magnitude Response of DFT');
subplot(2,1,2);
plot(k,PhaseX);
title('Phase Response of DFT');
```

Result and Observations:



Conclusion: Hence DFT of the given sequence is determined and Magnitude & Phase Plot is plotted.

Experiment No. 5

Aim: Write a MATLAB Program to calculate Circular Convolution of Two Equal Sequence using DFT & IDFT.

Tool: Matlab

Theory:

Circular convolution, also known as cyclic convolution, is a special case of periodic convolution, which is the convolution of two periodic functions that have the same period. Periodic convolution arises, for example, in the context of the discrete-time Fourier transform (DTFT). In particular, the DTFT of the product of two discrete sequences is the periodic convolution of the DTFTs of the individual sequences. And each DTFT is a periodic summation of a continuous Fourier transform function (see DTFT § Definition). Although DTFTs are usually continuous functions of frequency, the concepts of periodic and circular convolution are also directly applicable to discrete sequences of data. In that context, circular convolution plays an important role in maximizing the efficiency of a certain kind of common filtering operation.

A case of great practical interest is illustrated in the figure. The duration of the x sequence is N (or less), and the duration of the h sequence is significantly less. Then many of the values of the circular convolution are identical to values of $x*h$, which is actually the desired result when the h sequence is a finite impulse response (FIR) filter. Furthermore, the circular convolution is very efficient to compute, using a fast Fourier transform (FFT) algorithm and the circular convolution theorem.

There are also methods for dealing with an x sequence that is longer than a practical value for N. The sequence is divided into segments (blocks) and processed piecewise. Then the filtered segments are carefully pieced back together. Edge effects are eliminated by overlapping either the input blocks or the output blocks. To help explain and compare the methods, we discuss them both in the context of an h sequence of length 201 and an FFT size of N = 1024.

Input Parameter:

Enter the First Sequence x1[n] =[1 2 3 4 5]

Enter the Second Sequence x2[n] =[5 4 3 2 1]

45.0000 40.0000 40.0000 45.0000 55.0000

Code:

```
% To calculate Circular Convolution of Two Equal length Sequences
% using DFT and IDFT
x1=input('Enter the First Sequence =');
x2=input('Enter the Second Sequence =');
N=length(x1);
n=[0:1:N-1];
k=[0:1:N-1];
nk=n'*k;
```

```
WN=exp(-1j*2*pi/N);
WNnk=WN.^nk;
% DFT of x1 and x2 is Xk1 and Xk2 respectively
Xk1=x1*WNnk;
Xk2=x2*WNnk;
% Multiplication of Two DFTs
Yk=Xk1.*Xk2;
% IDFT of Yk
WNI=exp(1j*2*pi/N);
WNInk=WNI.^nk;
yc=(Yk*WNInk)/N;
disp(abs(yc));
subplot(2,2,1);
stem(n,x1);
subplot(2,2,2);
stem(n,x2);
subplot(2,2,3);
stem(n,abs(yc));
```

Result and Observations:

Conclusion: Hence the Circular Convolution of Two Equal Sequence is calculated using DFT & IDFT.

Experiment No. 6

Aim: Write a MATLAB Program to Compute DFT & IDFT using Fast Fourier Transform (FFT) Algorithm and Inverse Fast Fourier Transform (IFFT) Algorithm

Tool used: MATLAB

Theory:

A Fast Fourier transform (FFT) is an algorithm that calculates the Discrete Fourier Transform (DFT) of Discrete Periodic Sequence of length N. Discrete Fourier transform is a tool to convert specific types of sequences of functions into other types of representations. Another way to explain discrete Fourier transform is that it transforms the structure of the cycle of a waveform into sine components. Fast Fourier transform is an algorithm to compute DFT with less time complexity hence it can be used predominantly in signal processing. It may be useful in reading things like sound waves, or for any image-processing technologies. Discrete Fourier Transform can be used to solve various types of equations, or show various types of frequency activity in useful ways.

As an extremely mathematical part of both computing and electrical engineering, fast Fourier transform and the DFT are largely the province of engineers and mathematicians looking to change or develop elements of various technologies. For example, Discrete Fourier transform might be helpful in sound engineering, seismology or in voltage measurements. The main advantage of having DFT is that through it, we can design the FIR filters. The expression for Discrete Fourier Transform (DFT) is as follows

$$X[k] = \sum_{k=0}^{N-1} x[n] W_N^{nk}$$

Where, Twiddle factor $W_N = e^{-j\frac{2\pi}{N}}$ and N is the Discrete Periodic Time Interval.

The value of N should be taken as power of 2 while calculating the DFT using FFT algorithm.

Inverse Fast Fourier transform (IFFT) is **an algorithm to compute IDFT**. It is also known as backward Fourier transform. It converts a space or time signal to a signal of the frequency domain. The DFT signal is generated by the distribution of value sequences to different frequency components. Working directly to convert on Fourier transform is computationally too expensive. So, Fast Fourier transform is used as it rapidly computes by factorizing the DFT matrix as the product of sparse factors. As a result, it reduces the DFT computation complexity from **$O(N^2)$ to O(N log N)**. And this is a huge difference when working on a large dataset. Also, FFT algorithms are very accurate as compared to the DFT definition directly, in the presence of round-off error.

$$x[n] = \frac{1}{N} \sum_{k=0}^{N-1} X[k] W_N^{-nk}$$

Code:

MATLAB Program:

```
N=input('Enter the value of Time Period N which should be the power of two, N=');
x=input('Enter the Input Sequence of length N in []=');
n=0:1:N-1;
subplot(4,1,1);
stem(n,x);
title('Input Sequence x(n)');
y=fft(x,N);
disp(y);
subplot(4,1,2);
stem(real(y));
title('Real Part of X(k)');
subplot(4,1,3);
stem(imag(y));
title('Imaginary Part of X(k)');
z=ifft(y,N);
disp(z);
subplot(4,1,4);
stem(n,z);
title('IFFT of X(k)');
```

Input Variables:

1. Time Period N (Must be the Power of Two)
2. Input Discrete Sequence x[n]

Results and Observations:

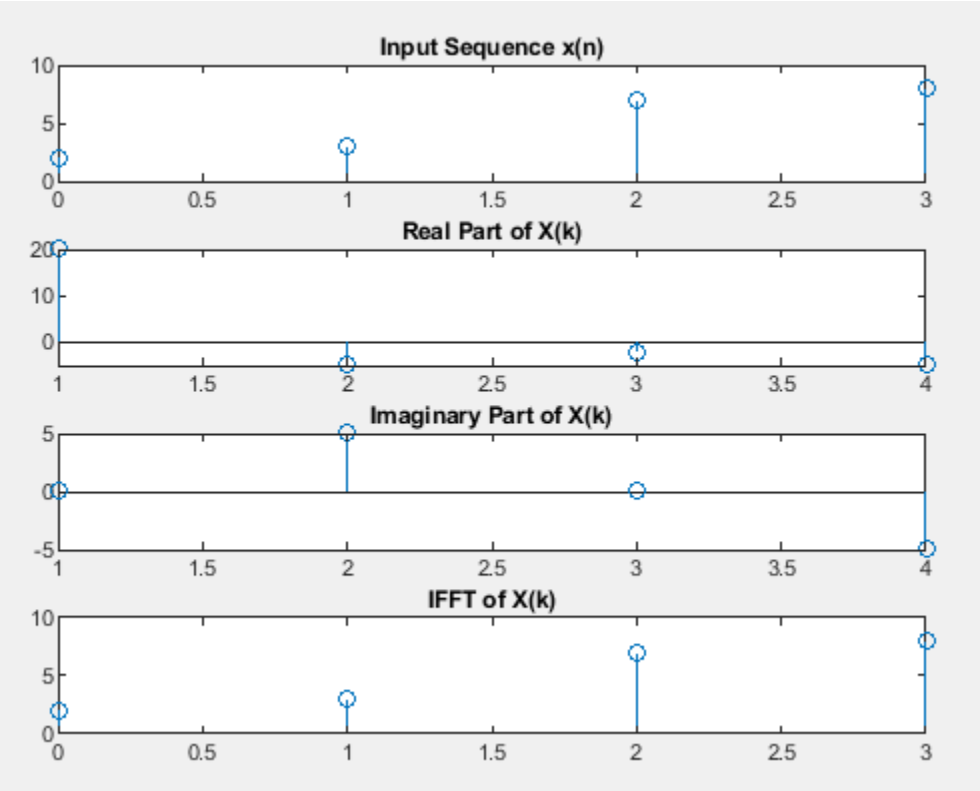


Fig 1 Plots of FFT and IFFT of sequence x[n]=[2 3 7 8]

Conclusion: Thus the plots of Fast Fourier Transform and the Inverse Fast Fourier Transform of the input discrete sequence x[n]=[2 3 7 8] are obtained as shown in Fig 1