Tribhuvan University

Institute of Engineering

Pulchowk Campus

Digital Signal Analysis and Processing

Lab 4

DFT and FFT

Submitted by:

Bishal Katuwal

075BCT028

Submitted to:

Department of Electronics and Computer Engineering

Pulchowk Campus

Submitted on:

17th August, 2022

Title
DFT and FFT

Background Theory
The transformation of time-domain signals to frequency domain signals are the key part of Digital Signal Processing. This process of transformation includes various tools such as DTFT, DFT, FFT etc. As time passes, the evolution of processes takes place. The FFT is the updated version or way of implementation of the DFT that takes less computational time and more efficient results than that of ordinary DFTs.

DFT
The signals found in nature are basically analog type of signals. But the digital computers that are used for the analysis of the signals can work only with the information that is discrete in nature and finite in length. Hence, the digitization of signals is performed. The Fourier transform of a signal within a finite range is called Discrete Fourier Transform. The mathematics and the algorithms of the Fourier transform are the heart of the DFT.
The Discrete Fourier Transform of a signal x(n) is mathematically expressed as:

$$\begin{gathered} X(k) = \sum_{n=0}^{N-1} x(n) e^{-\frac{j2\pi kn}{N}} \\ where\ k = 0, 1, 2, ..., N-1 \\ and\ e^{-\frac{j2\pi}{N}}\ is\ Nth\ root\ of\ 1. \end{gathered} \tag{1}$$

The signals obtained in the discrete Fourier transform are discrete and periodic in nature. DFT is able to calculate the frequency spectrum of a signal. The frequency response of a signal from its impulse response can be obtained from the DFT of the required signal. The DFT allows the frequency domain analysis of the signal and examines the information encoded in the frequency, phase and amplitude of the signal.

FFT
The Fast Fourier Transform (FFT) is nothing but an implementation of DFT. The FFT provides a more efficient result than DFT. The computational time required for a signal in the case of FFT is much lesser than that of DFT. Hence, it is called Fast Fourier Transform which is a collection of various fast DFT computation techniques. The FFT works with some algorithms that are used for computation.
In MATLAB DFT is done via FFT.

```
fft(x)
ifft(y)
```

The inverse process of DFT and FFT are called IDFT and IFFT respectively.

Activity

1. Sample DFT: x[n] = {0, 1, 2, 3}

```
x = [0,1,2,3]
y =fft(x)

subplot(2,1,1)
stem(real(y))
title('Real part of y')

subplot(2,1,2)
stem(imag(y))
title('Imaginary part of y')
```

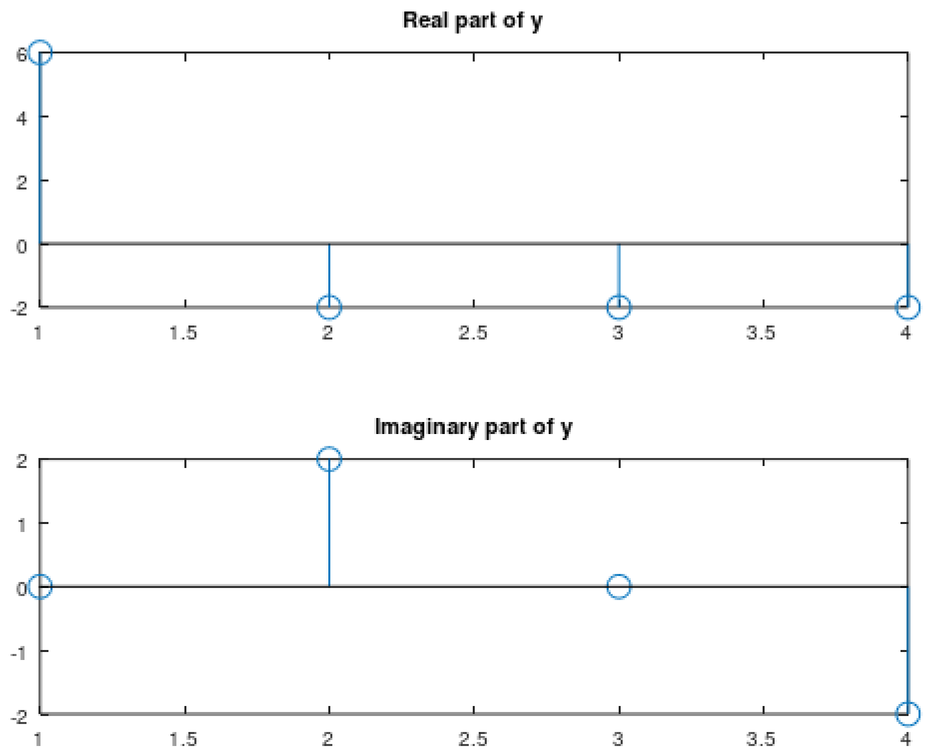

Figure 1: Sample DFT

2. Sample IDFT: IDFT of Question 1

```
x = [0,1,2,3]
subplot(4,1,1)
stem(x)
title('x')

y =fft(x)

subplot(4,1,2)
```

```
stem(real(y))
title('Real part of y')

subplot(4,1,3)
stem(imag(y))
title('Imaginary part of y')

x = ifft(y)
subplot(4,1,4)
stem(x)
title('Inverse FFT of y')
```

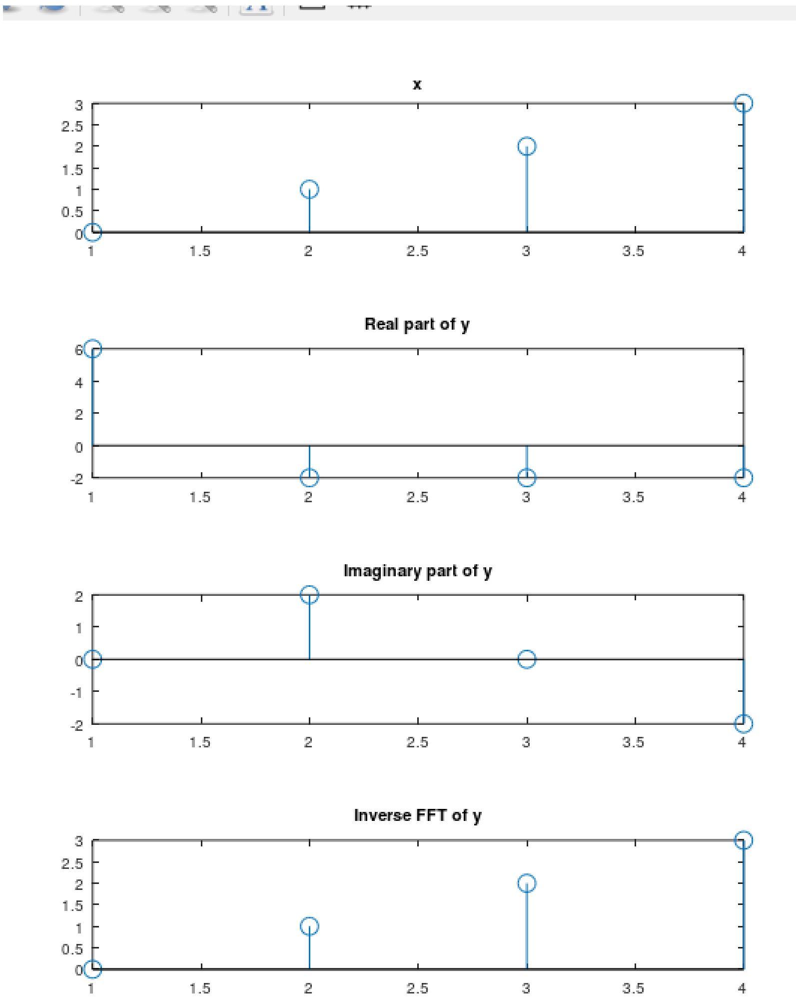

Figure 2: Sample IDFT

3. For

```
x = [1. 2, 3.7, 0.6, 1. 3, 2]
```

- Find DFT using FFT.
- Find x[n] from the obtained FFT.

```
x = [1. 2, 3.7, 0.6, 1. 3, 2]
subplot(4,1,1)
stem(x)
title('x')

y =fft(x)

subplot(4,1,2)
stem(real(y))
title('Real part of y')

subplot(4,1,3)
stem(imag(y))
title('Imaginary part of y')

x = ifft(y)
subplot(4,1,4)
stem(x)
title('Inverse FFT of y')
```

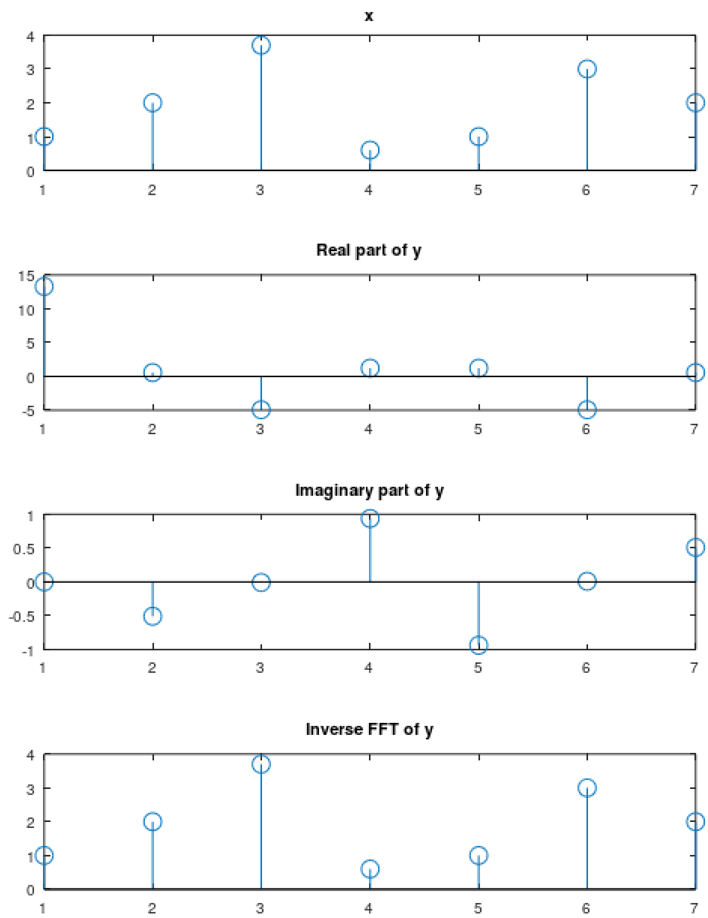

Figure 3: FFT and IFFT

4. For

$$x = 0.5^n u[n]$$

- Find DFT of x[n].
- Find x[n] from the obtained FFT.

```
x = 0.5 .^t(t>0)
subplot(4,1,1)
stem(x)
title('x')

y =fft(x)

subplot(4,1,2)
stem(real(y))
title('Real part of y')

subplot(4,1,3)
stem(imag(y))
title('Imaginary part of y')
```

```
x = ifft(y)
subplot(4,1,4)
stem(x)
title('Inverse FFT of y')
```

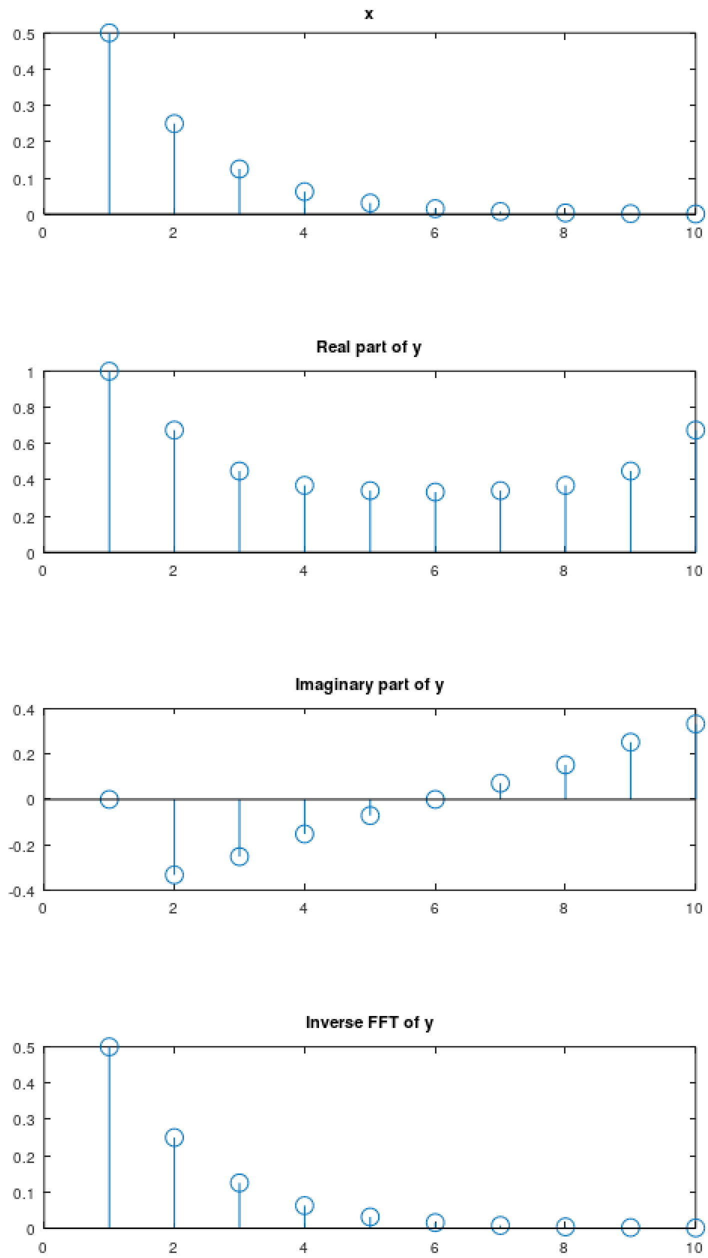

Figure 4: FFT and IFFT

Conclusion

In this way "Lab4 : DFT and FFT" was completed through the use of MATLAB.