Experiment No: 6 Date:29/08/2024

DFT AND IDFT

Aim

To perform DFT, IDFT and magnitude and phase plot of DFT.

Theory

The Discrete Fourier Transform (DFT) and its inverse (IDFT) are fundamental tools in signal processing, used to convert a discrete-time signal into its frequency-domain representation and vice versa.

The DFT transforms a sequence of N complex numbers, x[n], into a sequence of N complex numbers, X[k], where X[k] represents the amplitude and phase of the k-th frequency component in the signal. The DFT is defined by the following equation:

$$X[k] = \sum_{n=0}^{N-1} x[n] \cdot e^{-j\frac{2\pi}{N}kn} \quad \text{for} \quad k = 0, 1, 2, \ldots, N-1$$

- X[k] is the k-th DFT coefficient
- x[n] is the n-th sample of the input signal
- N is the length of the input signal
- j is the imaginary unit (√-1)

The IDFT, which is the inverse of the DFT, transforms the frequency-domain representation back into the time-domain. It is defined as:

$$x[n] = \frac{1}{N} \sum_{k=0}^{N-1} X[k] \cdot e^{j\frac{2\pi}{N}kn} \quad \text{for} \quad n = 0, 1, 2, \ldots, N-1$$

In the IDFT:

- X[k] represents the frequency-domain data.
- x[n] is the reconstructed time-domain signal.
- The term 1/N ensures that the energy of the signal is correctly scaled after transformation.

OBSERVATION

OUTPUT

a)DFT

```
2.0000 + 0.0000i

1.0000 - 1.0000i

0.0000 + 0.0000i

1.0000 + 1.0000i

2.0000 + 0.0000i 1.0000 - 1.0000i 0.0000 + 0.0000i 1.0000 + 1.0000i
```

b)IDFT

```
 1

 1

 0

 0

1 1 0 0
```

PROGRAM

a)DFT

```
clc;
clear;
 close all;
 x=[1 1 0 0];
 N=length(x);
 X=zeros(4,1);
 for k=0:N-1
    for n =0:N-1
        X(k+1)=X(k+1)+x(n+1)*exp(-i*2*pi*n*k/N);
    end
 end
 disp(round(X));
  disp(fft(x));
```

b)IDFT

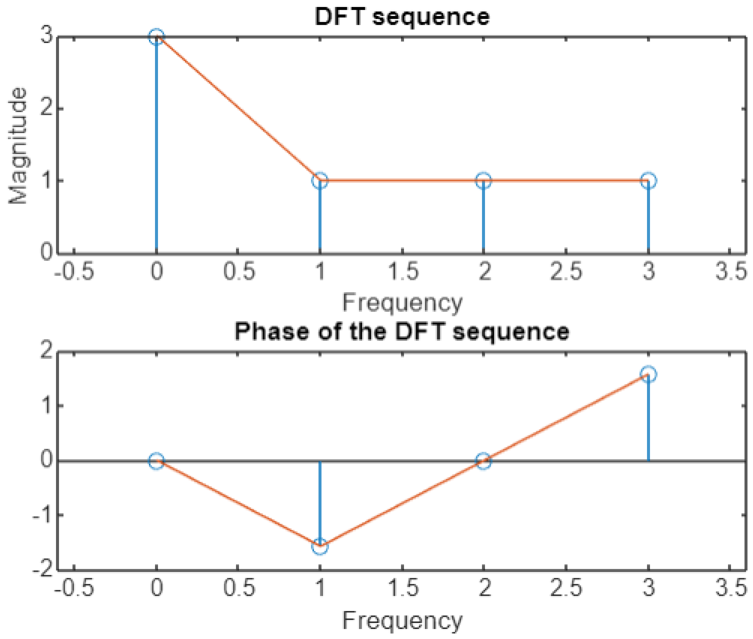
```
clc;
clear all;
 close all;
 X=[2 1-i 0 1+i];
 N=length(X);
 x=zeros(4,1);
 for n=0:N-1
    for k =0:N-1
        x(n+1)=(x(n+1)+X(k+1)*exp(i*2*pi*n*k/N))
     end
 end
x=x/N;
 disp(round(x));
```

OBSERVATION

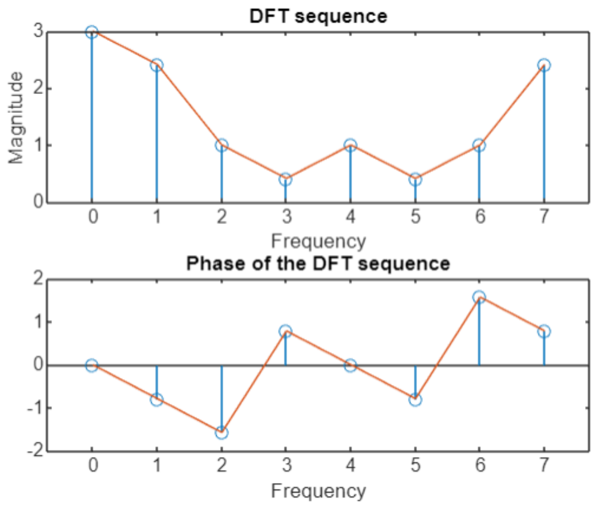
OUTPUT

c) Magnitude and Phase plot of DFT

```
Enter the value for N:4
   3.0000 + 0.0000i
   0.0000 - 1.0000i
   1.0000 + 0.0000i
   0.0000 + 1.0000i
```



```
Enter the value for N:8
   3.0000 + 0.0000i
   2.0000 - 2.0000i
   0.0000 - 1.0000i
   0.0000 + 0.0000i
   1.0000 + 0.0000i
   0.0000 + 0.0000i
   0.0000 + 1.0000i
   2.0000 + 2.0000i
```

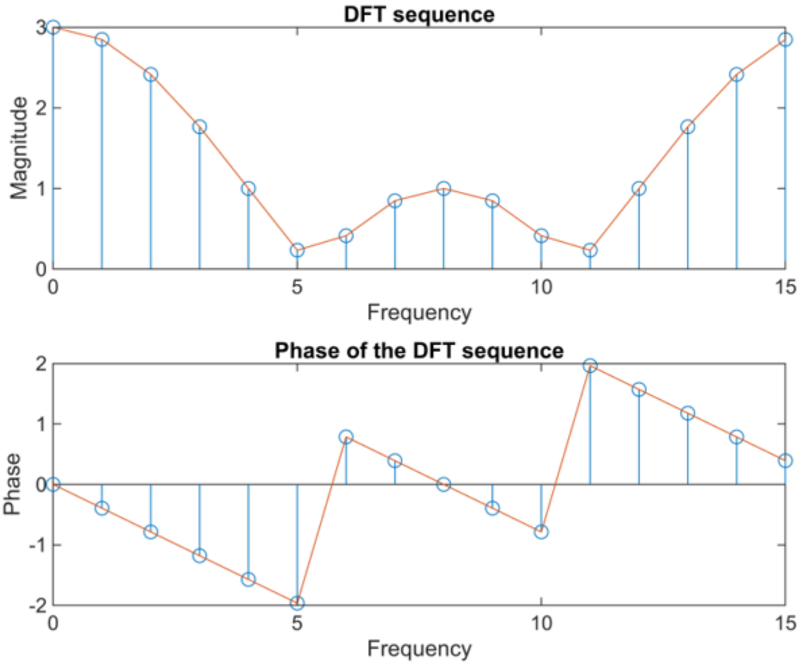


OUTPUT

```
Enter the value for N:16

  3.0000 + 0.0000i
   3.0000 - 1.0000i
   2.0000 - 2.0000i
   1.0000 - 2.0000i
   0.0000 - 1.0000i
   0.0000 + 0.0000i
   0.0000 + 0.0000i
   1.0000 + 0.0000i
   1.0000 + 0.0000i
   1.0000 + 0.0000i
   0.0000 + 0.0000i
   0.0000 + 0.0000i
   0.0000 + 1.0000i
   1.0000 + 2.0000i
   2.0000 + 2.0000i
   3.0000 + 1.0000

3.0000 + 0.0000i   2.6310 - 1.0898i  1.7071 - 1.7071i  0.6756 - 1.6310i  0.0000 - 1.0000i -
0.0898 - 0.2168i
```



```
disp(ifft(X));
```

c) Magnitude and Phase plot of DFT

```
clc;
 clear;
 close all;
xn=[1 1 1];
 N=input(“Enter the value for N:”);
 L=length(xn);
 if(N<L)
    error('N must be greater than or equal to L')
 end
 x=[xn,zeros(1,N-L)];
 N=length(x);
 Xk=zeros(N,1);
 for k=0:N-1
    for n =0:N-1
        Xk(k+1)=Xk(k+1)+x(n+1)*exp(-i*2*pi*n*k/N);
    end
 end
 disp(round(Xk));
  disp(fft(x));
 mgXk=abs(Xk);
 phaseXk=angle(Xk);
 k=0:N-1;
 subplot(2,1,1);
 stem(k,mgXk);
 hold on
 plot(k,mgXk);
```

OBSERVATION

d) DFT using Twiddle Factor

INPUT

```
enter the elements:[1 1 0 0]
```

OUTPUT

```
2.0000 + 0.0000i
 1.0000 - 1.0000i
 0.0000 - 0.0000i
 1.0000 + 1.0000i
```

e) IDFT using Twiddle Factor

OUTPUT

```
 1
 1
 0
 0
 title('DFT sequence');
 xlabel('Frequency');
 ylabel('Magnitude');
 subplot(2,1,2);
 stem(k,phaseXk);
 hold on
 plot(k,phaseXk);
 title('Phase of the DFT sequence');
 xlabel('Frequency');
```

d) DFT using Twiddle Factor

```
clc;
clear all;
 close all;

% Input signal
x = input(“enter the elements:”);
N = length(x);



X = zeros(1, N);
% Compute DFT using the twiddle factor
for k = 0:N-1
    for n = 0:N-1

        W = exp(-1j * 2 * pi * k * n / N);
        X(k+1) = X(k+1) + x(n+1) * W;
    end
end
```

```
% Display the result
disp('DFT of the input signal:');
disp(X);
```

e) IDFT using Twiddle Factor

```
% IDFT using Twiddle Factor
clc;
clear all;
close all;
X = [2,1-j,0,1+j];
N = length(X);
x = zeros(1, N);
% Compute IDFT using the twiddle factor
for n = 0:N-1
    for k = 0:N-1

        W = exp(1j * 2 * pi * k * n / N);
        x(n+1) = x(n+1) + X(k+1) * W;
    end
    % Divide by N as per IDFT formula
    x(n+1) = x(n+1) / N;
end
% Display the result
disp('Reconstructed time-domain signal (IDFT):');
disp(x);
```

Result

Computed DFT and IDFT using inbuilt and manual methods, magnitude and phase of dft and Twiddle factor matrix and verified the output.