Experiment No: 6 Date: 29/09/2024

DFT and IDFT

Aim

To compute the DFT and IDFT of a signal using inbuilt function and manual method

Theory

Discrete Fourier Transform is the transformation used to represent the finite duration frequencies. DFT of a discrete sequence x(n) is obtained by performing sampling operations in both time domain and frequency domain. It is the frequency domain representation of a discrete digital signal.

The DFT of a sequence x (n) of length N is given by the following equation,

$$x(k) = \sum_{n=0}^{N-1} x(n)e^{\frac{-j2\pi kn}{N}} \; ; 0 \leq k \leq N-1$$

IDFT performs the reverse operation of DFT, to obtain the time domain sequence x(n) from frequency domain sequence X(k). IDFT of the sequence is given as,

$$x(n) = \frac{1}{N}$$

Program

➔ DFT

```
% DFT
clc;
clear;
close all;
x = input("Enter the Sequence");
N = input("Enter the value for N point DFT");
L = length(x);
if N>=L
    x_new = [x, zeros(1,N-L)]; %zero padding
X = [zeros(1,N)]; %Creating N columns of 0 for initialising

for k = 0:N-1
    for n = 0:N-1
        X(k+1) = X(k+1) + x_new(n+1).*exp(-1i*2*pi*k*n/N); %Equation
for DFT
    end
end
```

```
disp("DFT without inbuilt function:");
disp(round(X));
disp("DFT using FFT:");
y = fft(x_new, N);
disp(round(y));

mag = abs(X);
subplot(2,1,1);
stem(0:N-1, mag);
title('Magnitude Spectrum');
xlabel('Frequency');
ylabel('Magnitude');

phase = angle(X);
subplot(2,1,2);
stem(0:N-1, phase);
title('Phase Spectrum');
xlabel('Frequency');
ylabel('Phase')

else
    disp('DFT cannot be calculated');
end
```

➔ IDFT

```
clc;
clear all;
close all;
X=input("Enter DFT sequence: ");
L=length(X);
N=input("Enter the value of N for N-point IDFT:");
if N>=L
 Xn=[X zeros(1,N-L)];
x=zeros(1,N);
for n=0:N-1
 for k=0:N-1
 x(n+1)=x(n+1)+((Xn(k+1).*exp(1i*2*pi*n*k/N))/N);
 end
end
disp("IDFT without using inbuilt function:");
disp(round(x,5));
y=round(ifft(Xn,N),5);
disp("IDFT using ifft:");
disp(y);
else
 disp("N-point IDFT cannot be found!")
end
```

➔ DFT using Twiddle Factor

```
clc;
clear;
close all;
x = input("Enter the Sequence");
N = input("Enter the value for N point DFT");
L = length(x);
twiddle_factors = zeros(N, N);
if N>=L
    x_new = [x, zeros(1,N-L)]; %zero padding
X = [zeros(1,N)]; %Creating N columns of 0 for initialising
for k = 0:N-1
    for n = 0:N-1
        twiddle = exp(-2*pi*1i*k*n/N);
        twiddle_factors(k+1, n+1) = twiddle;
        X(k+1) = X(k+1) + x(n+1) * twiddle;
    end
end
disp("DFT:");
disp(X);
else
    disp("DFT cannot be calculated")
end
```

➔ IDFT using Twiddle Factor

```
clc;
clear;
close all;
X = input("Enter DFT sequence: ");
L = length(X);
N = input("Enter the value of N for N-point IDFT:");
if N>=L
 Xn=[X zeros(1,N-L)];
x=zeros(1,N);
twiddle_factors = zeros(N,N);
for n=0:N-1
    for k=0:N-1
        twiddle =exp(2*1i*pi*k*n/N);
        twiddle_factors(n+1,k+1) = twiddle;
        x(n+1)=x(n+1)+X(k+1)*twiddle;
    end
end
disp("IDFT:");
x=x/N;
disp(x)
else
    disp("IDFT cannot be calculated");
end
```

Result

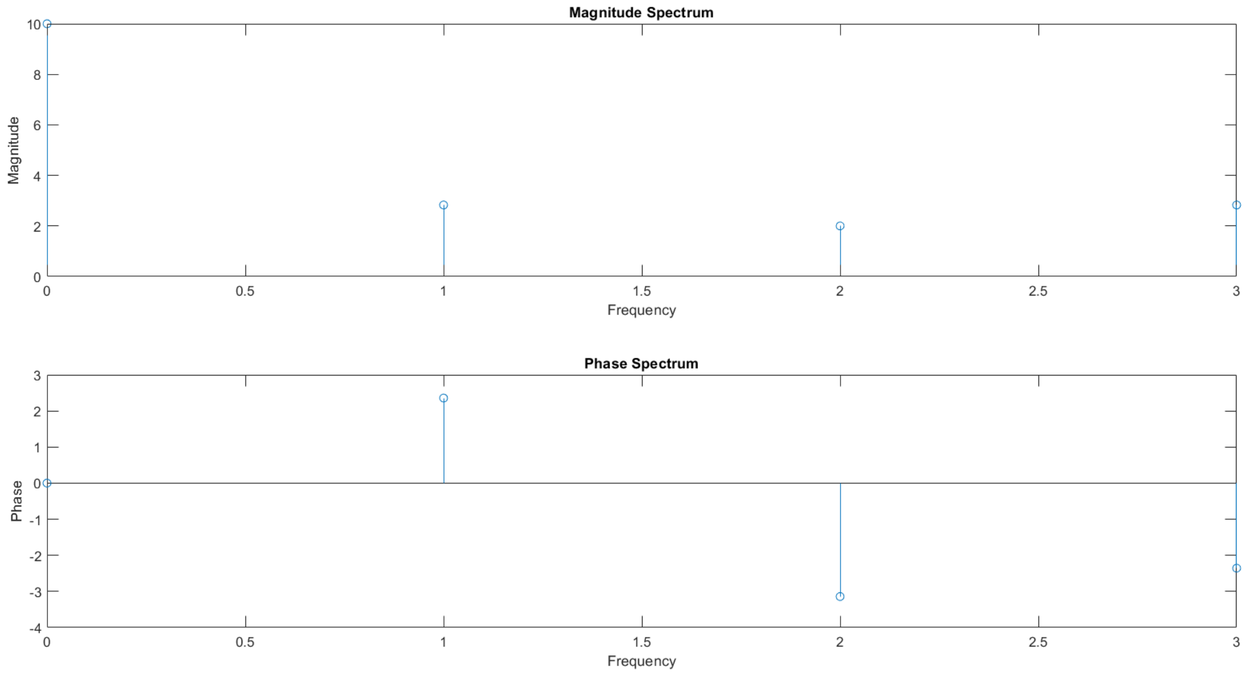
Simulated and computed the DFT and IDFT of a signal using inbuilt function and manual method.

Observation

➔ DFT

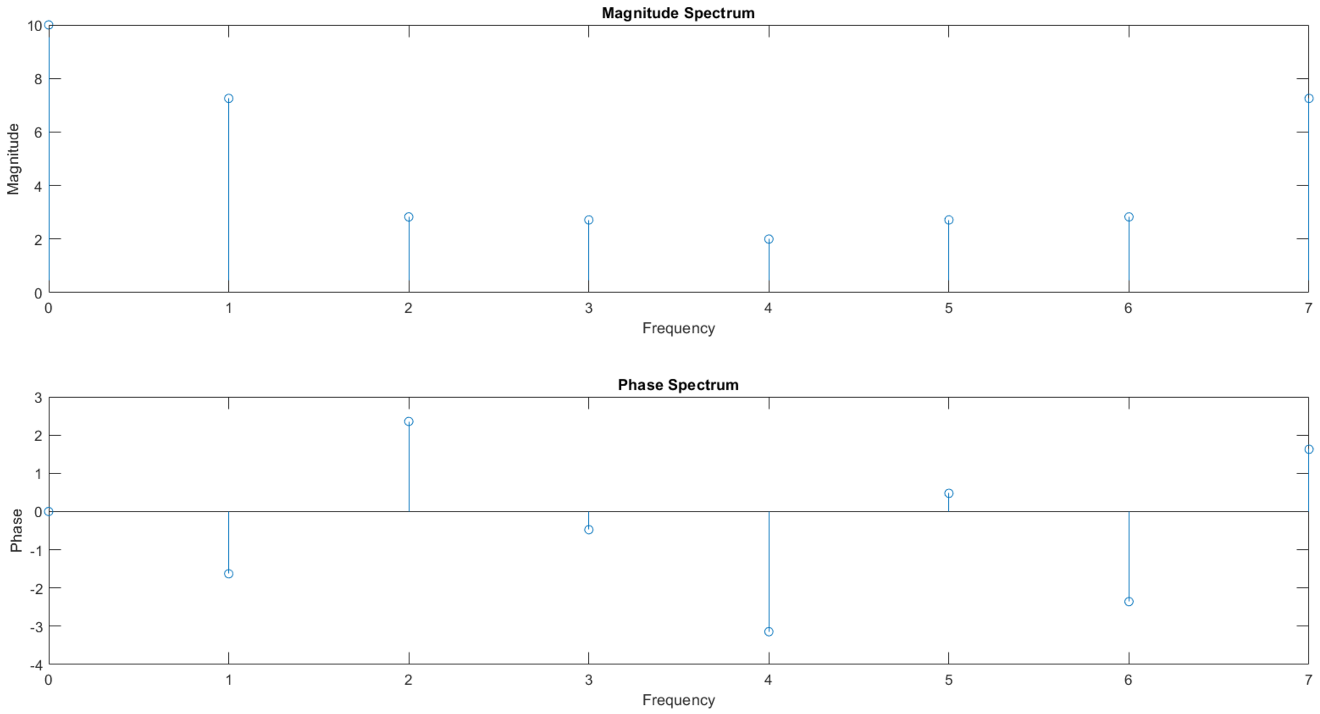
➔ N = 4

```
Enter the Sequence [1 2 3 4]
Enter the value for N point DFT 4
DFT without inbuilt function:
  10.0000 + 0.0000i  -2.0000 + 2.0000i  -2.0000 + 0.0000i  -2.0000 -
2.0000i

DFT using FFT:
  10.0000 + 0.0000i  -2.0000 + 2.0000i  -2.0000 + 0.0000i  -2.0000 -
2.0000i
```



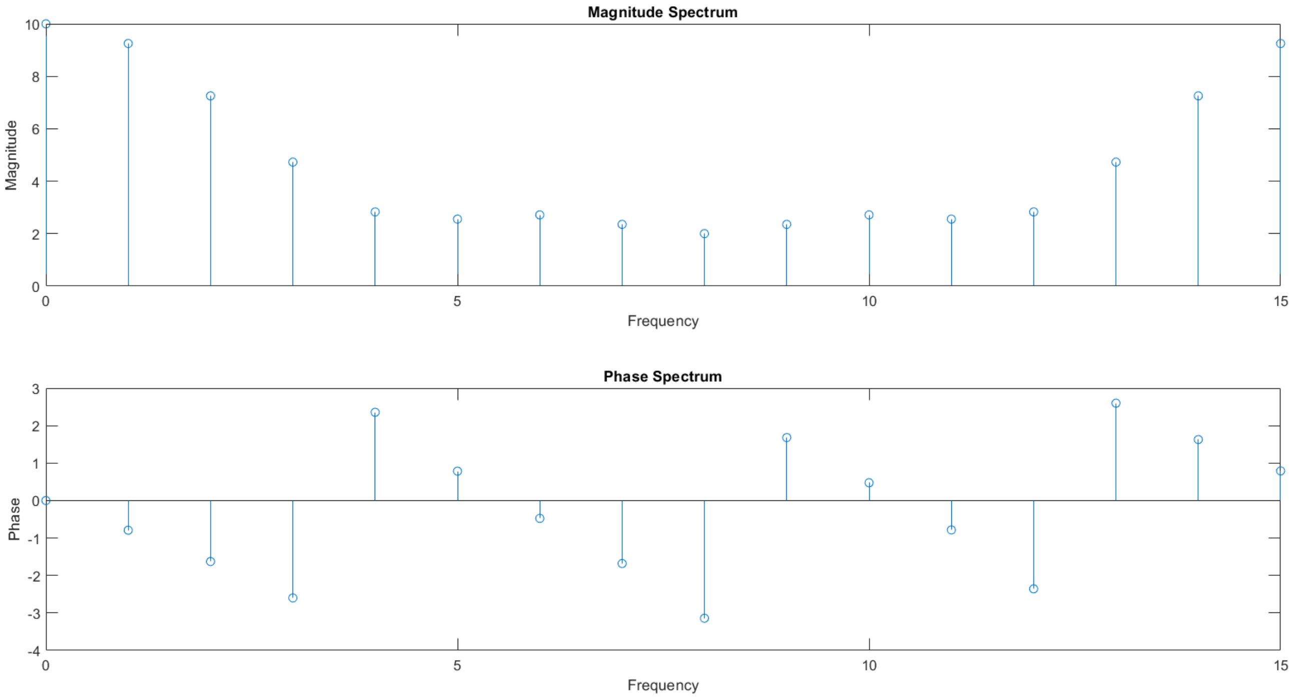
➔ N = 8

```
Enter the Sequence [1 2 3 4]
Enter the value for N point DFT 8
DFT without inbuilt function:
  Columns 1 through 5

  10.0000 + 0.0000i   0.0000 - 7.0000i  -2.0000 + 2.0000i   2.0000 -
1.0000i  -2.0000 + 0.0000i

  Columns 6 through 8

   2.0000 + 1.0000i  -2.0000 - 2.0000i   0.0000 + 7.0000i

DFT using FFT:
  Columns 1 through 5

  10.0000 + 0.0000i   0.0000 - 7.0000i  -2.0000 + 2.0000i   2.0000 -
1.0000i  -2.0000 + 0.0000i

  Columns 6 through 8

   2.0000 + 1.0000i  -2.0000 - 2.0000i   0.0000 + 7.0000i
```



➔ N = 16

```
Enter the Sequence [1 2 3 4]
Enter the value for N point DFT 16
DFT without inbuilt function:
  Columns 1 through 5

  10.0000 + 0.0000i   6.0000 - 7.0000i   0.0000 - 7.0000i  -
4.0000 - 2.0000i  -2.0000 + 2.0000i

  Columns 6 through 10

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

  Columns 11 through 15

   2.0000 + 1.0000i   2.0000 - 2.0000i  -2.0000 - 2.0000i  -
4.0000 + 2.0000i   0.0000 + 7.0000i

  Column 16

   6.0000 + 7.0000i

DFT using FFT:
  Columns 1 through 5

  10.0000 + 0.0000i   6.0000 - 7.0000i   0.0000 - 7.0000i  -
4.0000 - 2.0000i  -2.0000 + 2.0000i

  Columns 6 through 10

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

  Columns 11 through 15

   2.0000 + 1.0000i   2.0000 - 2.0000i  -2.0000 - 2.0000i  -
4.0000 + 2.0000i   0.0000 + 7.0000i

  Column 16

   6.0000 + 7.0000i
```



→ IDFT

```
Enter DFT sequence: [1 0 -1 0]
Enter the value of N for N-point IDFT:4
IDFT without using inbuilt function:
         0    0.5000         0    0.5000

IDFT using ifft:
         0    0.5000         0    0.5000
```

→ DFT using Twiddle Factor

```
Enter the Sequence [ 1 2 3 4]
Enter the value for N point DFT 4
DFT:
  10.0000 + 0.0000i  -2.0000 + 2.0000i  -2.0000 - 0.0000i  -
2.0000 - 2.0000i
```

→ IDFT using Twiddle Factor

```
Enter DFT sequence: [1 0 -1 0]
Enter the value of N for N-point IDFT:4
IDFT:
   0.0000 + 0.0000i   0.5000 - 0.0000i   0.0000 + 0.0000i
0.5000 - 0.0000i
```