EXPERIMENT No:1 COMPUTATION OF DFT AND IDFT

AIM:— To compute DFT and IDFT of given sequence.

SOFTWARE USED:— Spyder 3: Python 3.8.

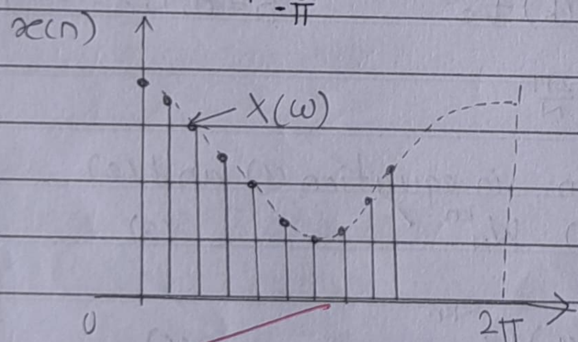
THEORY:—

DFT (Discrete Fourier Transform) is computed by discretization of DTFT (Discrete Time Fourier Transform)

Here the DTFT signal is sampled to N equidistant points of $x(\omega)$ where $N = 2^n = 2, 4, 6, 8, 16, 32$

$$X(\omega) = \sum_{n=\infty}^{\infty} x(n)\, e^{-j\omega n}$$

$$x(n) = \frac{1}{2\pi} \int_{-\pi}^{\pi} X(\omega)\, e^{j\omega n}\, d\omega.$$



$$X(\omega) \Big|_{\omega = \frac{2\pi}{N} k}$$

where,
N = Equidistant Samples.
k = Discrete frequency index.
n = discrete time index.

Equation of DFT

$$X(\omega) = \sum_{n=-\infty}^{\infty} x(n)\, e^{-j\omega n}$$

with $\omega \Big|_{2\pi \frac{k}{N}}$ and $N \rightarrow$ finite

$$X\left(\frac{2\pi}{N}\right) = X(k) = \sum_{n=0}^{N-1} x(n)\, e^{-j\frac{2\pi}{N}kn}.$$

$$X(k) = \sum_{n=0}^{N-1} x(n)\, e^{-j\frac{2\pi}{N}kn}.$$

where k = discrete frequency index $(0, 1, 2, \ldots, N-1)$.
n = discrete time index.

- <u>In Linear Transformation it is represented as.</u>

Let $W_N = e^{-j2\pi/N}$ ..... W_N is twiddle factor or phase factor

N point DFT of sequence $x(n)$ is given as.

$$X(k) = \sum_{n=0}^{N-1} x(n)\, e^{-j\frac{2\pi}{N}kn} \qquad k = 0, 1, 2, \ldots, N-1. \qquad \text{(1)}.$$

IDFT equation is given by.

$$x(n) = \frac{1}{N}\sum_{k=0}^{N-1} X(k)\, e^{j\frac{2\pi}{N}kn} \qquad n = 0, 1, \ldots N-1 \qquad \text{(2)}.$$

Let $W_n = e^{-j\frac{2\pi}{N}}$

put value of W_n in equation (1) and (2).

$$X(k) = \sum_{n=0}^{N-1} x(n)\, W_N^{kn} \qquad \text{(3)}$$

$$x(n) = \frac{1}{N}\sum_{k=0}^{N-1} X(k)\, W_N^{-kn} \qquad \text{(4)}$$

Equation (3) ~~in~~ and (4) in matrix form is written as.

$$X_K = W_N \cdot x_N \qquad \text{(5)}$$

$$x_N = \frac{1}{N} W_N^{*} X_K \qquad \text{(6)}$$

where, $W_N \Rightarrow N \times N$ Matrix (Twiddle).
$X_K \rightarrow N \times 1 \Longrightarrow$ DFT points.
$x_N \Rightarrow N \times 1 \Longrightarrow$ IDFT points.

STEPS OF PROGRAM :-

(1) Compute DFT and IDFT using inbuilt numpy function (fft, ifft) in python.

step 1 : Import fft, ifft packages from the numpy library.
step 2 : Define N point input sequence.
step 3 : Take the user input for required number of DFT points.
step 4 : Calculate the number of trailing zeros required and add them to the input sequence by zero padding.
step 5 : Find DFT of the sequence using fft() function and print the answer.
step 6 : Use IFFT () function to find the original input sequence. Round up the real parts by using np. round () function and print the input sequence x(n).

(2) Compute DFT and IDFT with user defined functions.
step 1 : Import numpy library.
step 2 : Define a function DFT using the keyword 'def'
step 3 : Convert list type data into array.
step 4 : Calculate the length of the sequence.
step 5 : Convert the row vector array into column vector using ~~np.~~ reshape function.
step 6 : Declare and define twiddle matrix and round up to 1 decimal point using np. round () function.
step 7 : Print the twiddle matrix and return the value.
step 8 : Define ~~conjugate of twiddle ma~~ a function for IDFT using def() function.

step 9: Repeat steps 4 and 5

step 10: Define conjugate of twiddle matrix and round up upto 1 decimal point using np.round () function.

step 11: Print the matrix and return.

step 12: Define an input sequence.

step 13: Call the function DFT() and IDFT() and print the answers.

PROBLEM:-

Find DFT of the sequence and also find the original signal back.

$$x(n) = [4, 2, 1, 3]$$

We know,

$$X_K = W_N \cdot x_N.$$

$$\begin{bmatrix} X(0) \\ X(1) \\ X(2) \\ X(3) \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix} \begin{bmatrix} 4 \\ 2 \\ 1 \\ 3 \end{bmatrix}$$

$$= \begin{bmatrix} 4+2+1+3 \\ 4-2j-1+3j \\ 4-2+1-3 \\ 4+2j-1-3j \end{bmatrix} = \begin{bmatrix} 10 \\ 3+j \\ 0 \\ 3-j \end{bmatrix}$$

$$X(k) = [10, 3+j, 0, 3-j]$$

IDFT

$$x_N = \frac{1}{N} W_N^* \cdot X_K$$

$$\begin{bmatrix} x(0) \\ x(1) \\ x(2) \\ x(3) \end{bmatrix} = \frac{1}{4} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & +j & -1 & -j \\ 1 & -1 & +1 & -1 \\ 1 & -j & -1 & +j \end{bmatrix} \begin{bmatrix} 10 \\ 3+j \\ 0 \\ 3-j \end{bmatrix}$$

$$= \frac{1}{4} \begin{bmatrix} 10+3+j+3-j \\ 10-3j+3j-1+1 \\ 10-3+j-3+j \\ 10-3j+1+3j-1 \end{bmatrix} = \frac{1}{4} \begin{bmatrix} 16 \\ 8 \\ 4 \\ 2 \end{bmatrix}$$

$$x(n) = [4, 2, 1, 3]$$

Problem of User Defined functions.

$x(n) = [1, 2, 3, 4]$

DFT : $X_K = W_N \cdot x_N$

$$\begin{bmatrix} X(0) \\ X(1) \\ X(2) \\ X(3) \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix} \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \end{bmatrix} = \begin{bmatrix} 1+2+3+4 \\ 1-2j-3+4j \\ 1-2+3-4 \\ 1+2j-3-4j \end{bmatrix}$$

$$= \begin{bmatrix} 10 \\ -2+2j \\ -2 \\ -2-2j \end{bmatrix}$$

$$X(k) = [10, -2+2j, -2, -2-2j]$$

IDFT : $x(n) = \frac{1}{N} W_N^* \cdot X_K$

$$\begin{bmatrix} x(0) \\ x(1) \\ x(2) \\ x(3) \end{bmatrix} = \frac{1}{4} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & j & -1 & -j \\ 1 & -1 & 1 & -1 \\ 1 & -j & -1 & j \end{bmatrix} \begin{bmatrix} 10 \\ -2+2j \\ -2 \\ -2-2j \end{bmatrix}$$

$$= \frac{1}{4} \begin{bmatrix} 4 \\ 8 \\ 12 \\ 16 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \end{bmatrix}$$

$$x(n) = [1, 2, 3, 4]$$

CONCLUSION :—

1. Through this experiment, we have learned the computation of N point DFT and IDFT using linear transformation.
2. For input sequence of L points, minimum L point DFT and IDFT is required to compute for the proper reconstruction of input sequence back.
3. For number of DFT points N, greater than the length of sequence L $(N > L)$, the reconstructed sequence is having $(N-L)$ trailing zeros.
4. N^2 complex multiplications and $(N-1)$ complex additions are required for computing N point DFT using direct DFT.