EXPERIMENT NO: 03

CIRCULAR CONVOLUTION USING DFT-IDFT METHOD.

AIM: To compute circular convolution using DFT - IDFT method and understand the difference between linear and circular convolution.

SOFTWARE USED: Spyder 3: Python 3.8.

THEORY: — Property of Multiplication of two N-point DFTs.

The property says that the circular convolution of two N point sequence in time domain, results in the multiplication of their DFTs in frequency domain.

Let $x_1(n)$ be an N point sequence.

and $x_1(n) \xleftrightarrow[\text{DFT}]{\text{N pt}} X_1(k)$.

$x_2(n)$ be another N point sequence.

$x_2(n) \xleftrightarrow[\text{DFT}]{\text{N-pt}} X_2(k)$.

Then ~~Let~~ $X_3(k) = X_1(k) \cdot X_2(k)$.

and $x_3(n) \xleftrightarrow{\text{IDFT}} X_3(k)$.

ie $x_3(n) = x_1(n)$ Ⓝ $x_2(n)$.

Computation of circular convolution of two sequences $x_1(n) = [1, 2, 3, 4]$ and $x_2(n) = [2, 4]$

length of $x_1 = N_1 = 4$.

length of $x_2 = N_2 = 2$.

length of Circular Convolution $= \max(N_1, N_2)$

$\therefore N = \max(4, 2)$

$$\boxed{N = 4}$$

Adding trailing zeros to x_2

$\therefore x_2$ becomes $[2, 4, 0, 0]$

$$X_{1K} = W_N \cdot x_{1N}$$

$$\begin{bmatrix} X_1(0) \\ X_1(1) \\ X_1(2) \\ X_1(3) \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix} \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \end{bmatrix} = \begin{bmatrix} 1+2+3+4 \\ 1-2j-3+4j \\ 1-2+3-4 \\ 1+2j-3-4j \end{bmatrix}$$

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

$$X_1(k) = [10, -2+2j, -2, -2-2j]$$

$$X_{2k} = W_N \cdot x_{2N}$$

$$\begin{bmatrix} X_2(0) \\ X_2(1) \\ X_2(2) \\ X_2(3) \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix} \begin{bmatrix} 2 \\ 4 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 2+4 \\ 2-4j \\ 2-4 \\ 2+4j \end{bmatrix}$$

$$= \begin{bmatrix} 6 \\ 2-4j \\ -2 \\ 2+4j \end{bmatrix}$$

$$X_2(k) = [6, 2-4j, -2, 2+4j]$$

$$X_3(k) = X_1(k) \cdot X_2(k)$$

$$= [60, 4+12j, 4, 4-12j]$$

$$x_3(n) = \frac{1}{N} W_N^* X_3(k)$$

$$\begin{bmatrix} x_3(0) \\ x_3(1) \\ x_3(2) \\ x_3(3) \end{bmatrix} = \frac{1}{4} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix} \begin{bmatrix} 60 \\ 4+12j \\ 4 \\ 4-12j \end{bmatrix} = \frac{1}{4} \begin{bmatrix} 60+4+12j+4-12j \\ 60+4j-12-4-4j-12 \\ 60-4-12j+4-4+12j \\ 60-4j+12-4+4j+12 \end{bmatrix}$$

$$= \frac{1}{4}\begin{bmatrix} 72 \\ 32 \\ 56 \\ 80 \end{bmatrix} = \begin{bmatrix} 18 \\ 8 \\ 14 \\ 20 \end{bmatrix}$$

$$x_3(n) = [18, 8, 14, 20]$$

Verifying above problem using matrix method.

$$x_3(n) = x_1(n) \; (N) \; x_2(n)$$

$$\begin{bmatrix} 1 & 4 & 3 & 2 \\ 2 & 1 & 4 & 3 \\ 3 & 2 & 1 & 4 \\ 4 & 3 & 2 & 1 \end{bmatrix}_{\substack{N \times N \\ = 4 \times 4}} \begin{bmatrix} 2 \\ 4 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 2+16 \\ 4+4 \\ 6+8 \\ 8+12 \end{bmatrix} = \begin{bmatrix} 18 \\ 8 \\ 14 \\ 20 \end{bmatrix}$$

$$x_3(n) = [18, 8, 14, 20]$$

$\therefore$ The problem of circular convolution using DFT-IDFT is correctly verified.

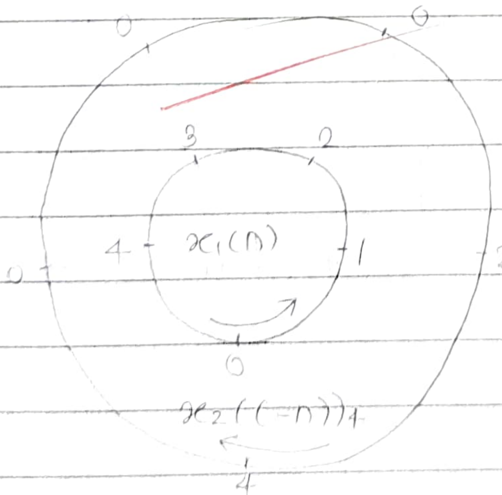
Linear Convolution

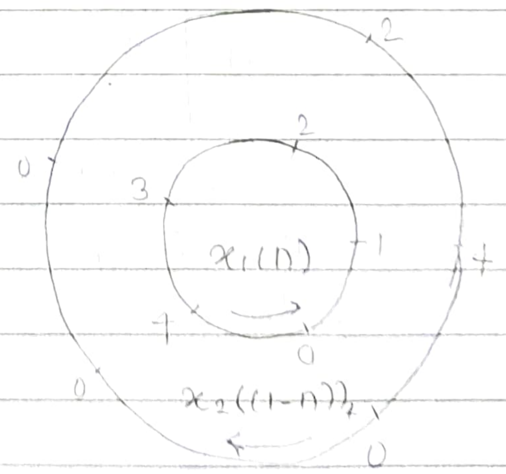
$x_1(n) = [1, 2, 3, 4] \Rightarrow N_1 = 4$

$x_2(n) = [2, 4] \Rightarrow N_2 = 2.$

$\therefore N = N_1 + N_2 - 1 = 4 + 2 - 1 = \underline{5}$

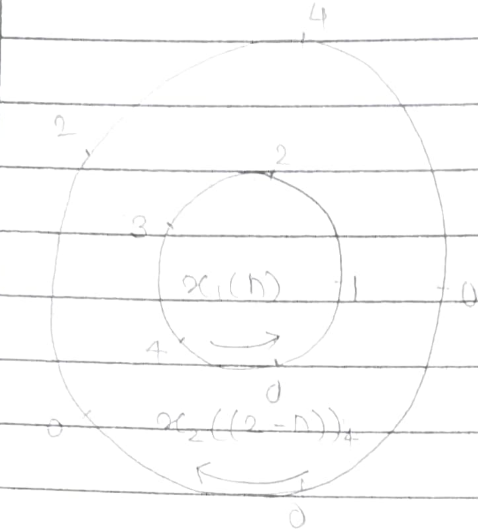
$x_1(n) = [1, 2, 3, 4, 0]$, $x_2(n) = [2, 4, 0, 0, 0]$

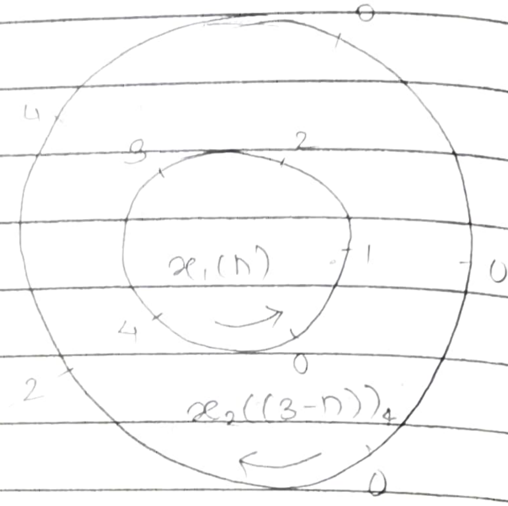




$x_3(0) = \underline{2}$

$x_3(n) = 4 + 4 = \underline{8}$



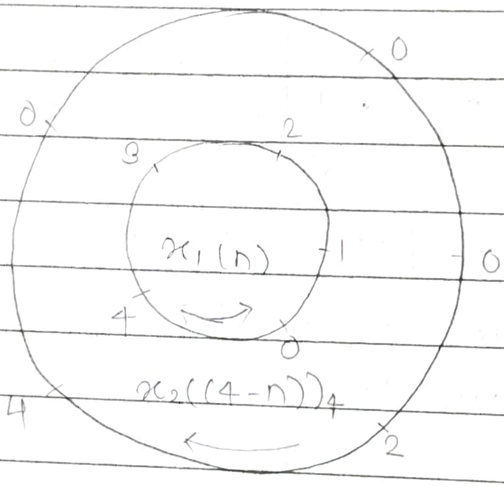


$x_3(2) = 6+8$

$= \underline{\underline{14}}$

$x_3(3) = 8+12$

$= \underline{\underline{20}}$



$x_3(4) = \underline{\underline{16}}$.

$x_3(n) = [2, 8, 14, 20, 16]$

Verification using diagonal addition method.

	1	2	3	4
2	2	4	6	8
4	4	8	12	16

$x_3(n) = [2, 8, 14, 20, 16]$

∴ The Linear Convolution is verified using diagonal addition.

From the above calulations
Circular Convolution is $x_3(n) = [18, 8, 14, 20]$
and linear Convolution is $x_3(n) = [2, 8, 14, 20, 16]$.
Here, in linear convolution the sequence is 5 point sequence and the sum of the first and the last term are alised together to form the sequence of Circular convolution. Therefore, it is proved that circular convolution is alised version of Linear Convolution.

Steps of Program:— Circular Convolution using DFT/IDFT.

step 1: Import Numpy package with fft, ifft.
step 2: Define the input sequences and calculate their length.
step 3: Calculate the length of sequence of circular convolution.
step 4: Add the required number of trailing zeros to the input sequences.
step 5: Compute the DFT for $x_1(n)$ and $x_2(n)$ sequences using the fft function.
step 6: Multiply the two dft sequences.
step 7: Take IDFT of the result to get circular convolution of the two sequences.

Linear Convolution using DFT/IDFT.

step1: Import Numpy package with fft, ifft
step2: Define the input sequences and calculate their length.
step3: Calculate the length of sequence for linear convolution.
step 4: Add the trailing zeros to the shorter input sequences.
step5: Compute $N = N_1 + N_2 - 1$ point DFT for both input sequence
step6: Multiplication of two dft sequences.
step7: Take IDFT of the result to get ~~circular~~ linear convolution of the two sequences.

CONCLUSION:—

(1) Through this experiment, we have learned to compute circular convolution using DFT-IDFT method.

(2) For two sequences of length L and M points respectively, the number of DFT, IDFT points to get the results of circular convolution is $N = \max(L, M)$

(3) For two sequences of length L and M points respectively, the number of DFT, IDFT points to get the result of Linear convolution is $N = L + M - 1$

(4) Circular convolution is aliased version of linear convolution