

Assignment-1

Krutik Mehta - EE18BTECH11027

Download all python codes from

https://github.com/Krutikmehta/IDP6/tree/main/Assignment_1/codes

and latex-tikz codes from

https://github.com/Krutikmehta/IDP6/tree/main/Assignment_1

1 PROBLEM

1.1. Let

$$x(n) = \left\{ \underset{\uparrow}{1}, 2, 3, 4, 2, 1 \right\} \quad (1.1.1)$$

$$y(n) + \frac{1}{2}y(n-1) = x(n) + x(n-2) \quad (1.1.2)$$

1.2. Compute

$$X(k) \triangleq \sum_{n=0}^{N-1} x(n)e^{-j2\pi kn/N}, \quad k = 0, 1, \dots, N-1 \quad (1.2.1)$$

and $H(k)$ using $h(n)$.

1.3. Compute

$$Y(k) = X(k)H(k) \quad (1.3.1)$$

2 SOLUTION

2.1. Let $W_N = e^{-j2\pi/N}$

We can express X as Matrix Multiplication of DFT Matrix and x.

$$X = \left[W_N^{ij} \right]_{N \times N} x, \quad i, j = 0, 1, \dots, N-1 \quad (2.1.1)$$

2.2. Properties :

a) symmetry property :

$$W_N^{k+N/2} = -W_N^k$$

b) Periodicity property :

$$W_N^{k+N} = W_N^k$$

c)

$$W_N^2 = W_{N/2}$$

2.3. Using properties to derive FFT from DFT :

$$X(k) = \sum_{n=0}^{N-1} x(n)W_N^{kn}, \quad k = 0, 1, \dots, N-1 \quad (2.3.1)$$

$$= \sum_{n=\text{even}} x(n)W_N^{kn} + \sum_{n=\text{odd}} x(n)W_N^{kn} \quad (2.3.2)$$

$$= \sum_{m=0}^2 x(2m)W_N^{2mk} + \sum_{m=0}^2 x(2m+1)W_N^{(2m+1)k} \quad (2.3.3)$$

using property c, we get,

$$X(k) = \sum_{m=0}^2 x(2m)W_{N/2}^{mk} + W_N^k \sum_{m=0}^2 x(2m+1)W_{N/2}^{mk} \quad (2.3.4)$$

$$= X_1(k) + W_N^k X_2(k) \quad (2.3.5)$$

- $X_1(k)$ and $X_2(k)$ are 3 point DFTs of $x(2m)$ and $x(2m+1)$, $m=0,1,2$.
- $X_1(k)$ and $X_2(k)$ are periodic, Hence $X_1(k+3) = X_1(k)$ and $X_2(k+3) = X_2(k)$.

2.4. Calculating X_1 and X_2

$$X_1(k) = \sum_{m=0}^2 x(2m)W_3^{mk} \quad (2.4.1)$$

$$X_2(k) = \sum_{m=0}^2 x(2m+1)W_3^{mk} \quad (2.4.2)$$

$$\begin{bmatrix} X_1(0) \\ X_1(1) \\ X_1(2) \\ X_2(0) \\ X_2(1) \\ X_2(2) \end{bmatrix} = \begin{bmatrix} W_3^0 & W_3^0 & W_3^0 & 0 & 0 & 0 \\ W_3^1 & W_3^1 & W_3^2 & 0 & 0 & 0 \\ W_3^2 & W_3^2 & W_3^4 & 0 & 0 & 0 \\ 0 & 0 & 0 & W_3^0 & W_3^1 & W_3^2 \\ 0 & 0 & 0 & W_3^1 & W_3^2 & W_3^4 \\ 0 & 0 & 0 & W_3^2 & W_3^4 & W_3^1 \end{bmatrix} \begin{bmatrix} x(0) \\ x(2) \\ x(4) \\ x(1) \\ x(3) \\ x(5) \end{bmatrix} \quad (2.4.3)$$

$$\hat{X} = \begin{bmatrix} X_1(0) \\ X_1(1) \\ X_1(2) \\ X_2(0) \\ X_2(1) \\ X_2(2) \end{bmatrix} = B_1 \begin{bmatrix} x(0) \\ x(2) \\ x(4) \\ x(1) \\ x(3) \\ x(5) \end{bmatrix} \quad (2.4.4)$$

$$\hat{X} = \begin{bmatrix} X_1(0) \\ X_1(1) \\ X_1(2) \\ X_2(0) \\ X_2(1) \\ X_2(2) \end{bmatrix} = B_1 P \begin{bmatrix} x(0) \\ x(1) \\ x(2) \\ x(3) \\ x(4) \\ x(5) \end{bmatrix} \quad (2.4.5)$$

$$\text{where } P = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

P is an odd-even permutation matrix.

$$\Rightarrow \hat{X} = B_1 P x \quad (2.4.6)$$

2.5. Calculating X

$$X(k) = X_1(k) + W_N^k X_2(k) \quad (2.5.1)$$

$$\begin{bmatrix} X(0) \\ X(1) \\ X(2) \\ X(3) \\ X(4) \\ X(5) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & W_6^0 & 0 & 0 \\ 0 & 1 & 0 & 0 & W_6^1 & 0 \\ 0 & 0 & 1 & 0 & 0 & W_6^2 \\ 1 & 0 & 0 & W_6^3 & 0 & 0 \\ 0 & 1 & 0 & 0 & W_6^4 & 0 \\ 0 & 0 & 1 & 0 & 0 & W_6^5 \end{bmatrix} \begin{bmatrix} X_1(0) \\ X_1(1) \\ X_1(2) \\ X_2(0) \\ X_2(1) \\ X_2(2) \end{bmatrix} \quad (2.5.2)$$

$$X = \begin{bmatrix} X(0) \\ X(1) \\ X(2) \\ X(3) \\ X(4) \\ X(5) \end{bmatrix} = B_2 \begin{bmatrix} X_1(0) \\ X_1(1) \\ X_1(2) \\ X_2(0) \\ X_2(1) \\ X_2(2) \end{bmatrix} \quad (2.5.3)$$

$$X = B_2 \hat{X} \quad (2.5.4)$$

$$X = B_2 B_1 P x \quad (2.5.5)$$

2.6. From the above exercise we have factorised the W matrix

$$W = B_2 B_1 P \quad (2.6.1)$$

2.7. Hence IFFT can be computed as;

$$x = \frac{1}{N} W^H X \quad (2.7.1)$$

$$x = \frac{1}{6} P^H B_1^H B_2^H X$$

2.8. The following code computes Y and generates magnitude and phase plots of X, H, Y

$$X = B_2 B_1 P x$$

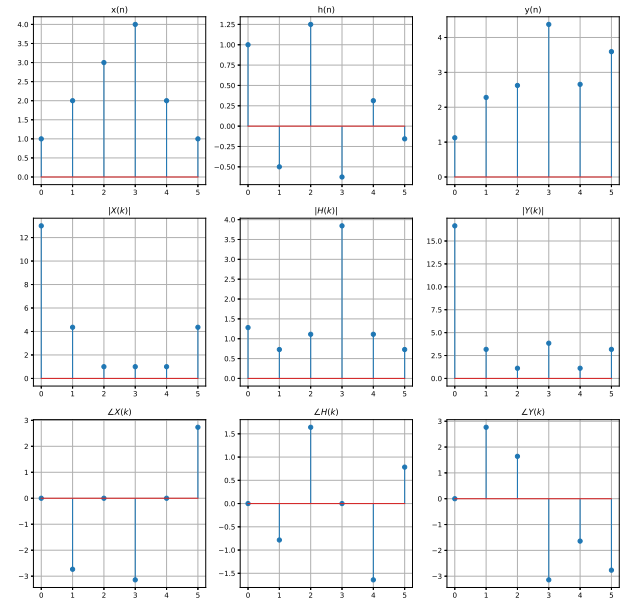
$$H = B_2 B_1 P h$$

$$Y = H X \quad (2.8.1)$$

$$y = \frac{1}{6} P^H B_1^H B_2^H y$$

https://github.com/Krutikmehta/IDP6/tree/main/Assignment_1/codes/EE18BTECH11027.py

2.9. The following plots are obtained



2.10. Benefits of FFT

- The DFT which would require n^2 operations for computation, but in the modified version, as most of the elements are zeros or ones, there are $(\frac{n}{2})^2$ operations.
- If we recursively perform this decomposition, it would result in $\frac{n}{2} \log n$ operation