Assignment-1

Bhanu prakash - EE18BTECH11022

1 Problem

1.1. Let

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

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

1.2. Compute

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

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

1.3. Compute

$$Y(k) = X(k)H(k) \tag{1.3.1}$$

2 Solution

2.1. Given difference equation is:

$$h(n) + \frac{1}{2}h(n-1) = \delta(n) + \delta(n-2) \tag{2.1.1}$$

2.2. Let $W_N = e^{-j2\pi/N}$
We can express X as Matrix Multiplication of DFT Matrix and x.

$$X = \left[W_N^{ij}\right]_{N\times N} x, \quad i, j = 0, 1, \ldots, N-1 \tag{2.2.1}$$

2.3. For this given x(n),we have N = 6

$$\implies W_6 = e^{-j2\pi/6} = \frac{1}{2} - \frac{\sqrt{3}}{2}j \tag{2.3.1}$$

$$\begin{bmatrix} X(0) \\ X(1) \\ X(2) \\ X(3) \\ X(4) \\ X(5) \end{bmatrix} = \begin{bmatrix} W_6^0 & W_6^0 & W_6^0 & W_6^0 & W_6^0 & W_6^0 \\ W_6^0 & W_6^1 & W_6^2 & W_6^3 & W_6^4 & W_6^5 \\ W_6^0 & W_6^2 & W_6^4 & W_6^6 & W_6^8 & W_6^{10} \\ W_6^0 & W_6^3 & W_6^6 & W_6^9 & W_6^{12} & W_6^{15} \\ W_6^0 & W_6^4 & W_6^8 & W_6^{12} & W_6^{16} & W_6^{20} \\ W_6^0 & W_6^5 & W_6^{10} & W_6^{15} & W_6^{20} & W_6^{25} \end{bmatrix} \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 2 \\ 1 \end{bmatrix} \tag{2.3.2}$$

$$\implies \begin{bmatrix} X(0) \\ X(1) \\ X(2) \\ X(3) \\ X(4) \\ X(5) \end{bmatrix} = \begin{bmatrix} 13 \\ -4 - \sqrt{3}j \\ 1 \\ -1 \\ 1 \\ -4 + \sqrt{3}j \end{bmatrix} \tag{2.3.3}$$

2.4.

$$\begin{bmatrix} H(0) \\ H(1) \\ H(2) \\ H(3) \\ H(4) \\ H(5) \end{bmatrix} = \begin{bmatrix} W_6^0 & W_6^0 & W_6^0 & W_6^0 & W_6^0 & W_6^0 \\ W_6^0 & W_6^1 & W_6^2 & W_6^3 & W_6^4 & W_6^5 \\ W_6^0 & W_6^2 & W_6^4 & W_6^6 & W_6^8 & W_6^{10} \\ W_6^0 & W_6^3 & W_6^6 & W_6^9 & W_6^{12} & W_6^{15} \\ W_6^0 & W_6^4 & W_6^8 & W_6^{12} & W_6^{16} & W_6^{20} \\ W_6^0 & W_6^5 & W_6^{10} & W_6^{15} & W_6^{20} & W_6^{25} \end{bmatrix} \begin{bmatrix} 1 \\ -0.5 \\ 1.25 \\ -0.625 \\ 0.3125 \\ -0.15625 \end{bmatrix} \tag{2.4.1}$$

$$\implies \begin{bmatrix} H(0) \\ H(1) \\ H(2) \\ H(3) \\ H(4) \\ H(5) \end{bmatrix} = \begin{bmatrix} 1.28125 \\ 0.51625 - 0.5142j \\ -0.07813 + 1.1096j \\ 3.84375 \\ -0.07183 - 1.1096j \\ 0.51625 + 0.5142j \end{bmatrix} \tag{2.4.2}$$

2.5. We can find Y using,

$$Y(k) = X(k)H(k) \tag{2.5.1}$$

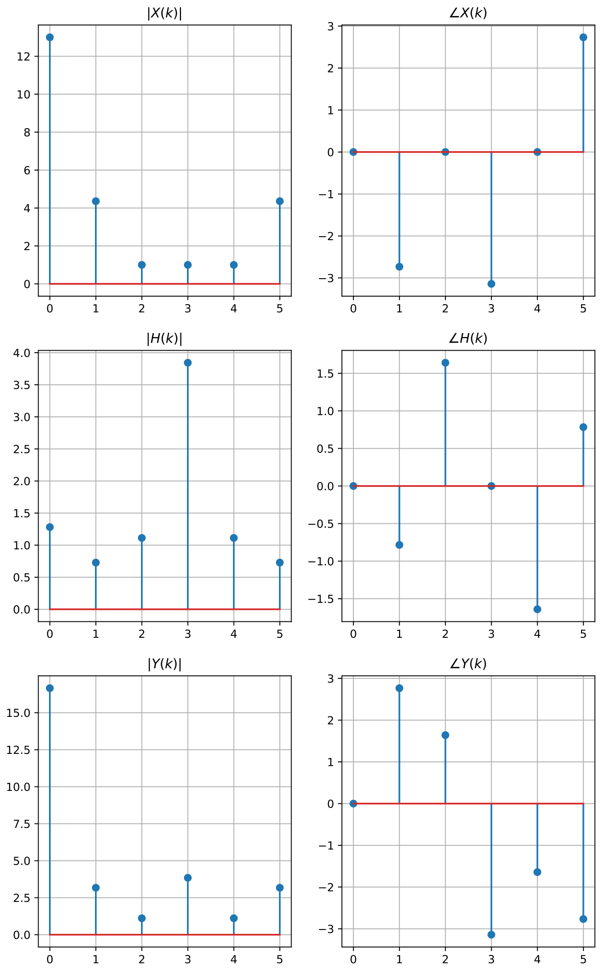
$$\begin{bmatrix} Y(0) \\ Y(1) \\ Y(2) \\ Y(3) \\ Y(4) \\ Y(5) \end{bmatrix} = \begin{bmatrix} X(0) \\ X(1) \\ X(2) \\ X(3) \\ X(4) \\ X(5) \end{bmatrix} \times \begin{bmatrix} H(0) \\ H(1) \\ H(2) \\ H(3) \\ H(4) \\ H(5) \end{bmatrix} \tag{2.5.2}$$

$$\implies \begin{bmatrix} Y(0) \\ Y(1) \\ Y(2) \\ Y(3) \\ Y(4) \\ Y(5) \end{bmatrix} = \begin{bmatrix} 16.65625 \\ -2.95312 + 1.16372j \\ -0.07813 + 1.1096j \\ -3.84375 \\ -0.07813 - 1.1096j \\ -2.95312 - 1.16372j \end{bmatrix} \tag{2.5.3}$$

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

```
https://github.com/Bhanuprakash072/dsp/tree/
    main/code/ee18btech11022_1.py
```

2.7. The following plots are obtained



2.8. Lets now look at one of the property of W_N and how can it be used to reduce time complexity of computation.

2.9. *Properties*

a) Symmetric Property :

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

b) Periodic Property :

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

c) Square Property :

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

Lets look at this property in particular

$$W_N^2 = W_{N/2} \tag{2.9.1}$$

2.10. F_N is the N-point DFT Matrix.
Using the property of Complex Exponentials we can express F_N in terms of $F_{N/2}$

$$F_N = \begin{bmatrix} I_{N/2} & D_{N/2} \\ I_{N/2} & -D_{N/2} \end{bmatrix} \begin{bmatrix} F_{N/2} & 0 \\ 0 & F_{N/2} \end{bmatrix} P_N \tag{2.10.1}$$

For N = 6

$$\implies F_6 = \begin{bmatrix} I_3 & D_3 \\ I_3 & -D_3 \end{bmatrix} \begin{bmatrix} F_3 & 0 \\ 0 & F_3 \end{bmatrix} P_6 \tag{2.10.2}$$

where I_3 is the 3x3 identity matrix

$$\tag{2.10.3}$$

$$D_3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & W_3^1 & 0 \\ 0 & 0 & W_3^2 \end{bmatrix} \tag{2.10.4}$$

$$P_6 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.10.5}$$

$$\implies P_6 \begin{bmatrix} x(0) \\ x(1) \\ x(2) \\ x(3) \\ x(4) \\ x(5) \end{bmatrix} = \begin{bmatrix} x(0) \\ x(2) \\ x(4) \\ x(1) \\ x(3) \\ x(5) \end{bmatrix} \tag{2.10.6}$$

Let

$$\begin{bmatrix} X_1(0) \\ X_1(1) \\ X_1(2) \end{bmatrix} = F_{N/2} \begin{bmatrix} x(0) \\ x(2) \\ x(4) \end{bmatrix} \tag{2.10.7}$$

$$\begin{bmatrix} X_2(0) \\ X_2(1) \\ X_2(2) \end{bmatrix} = F_{N/2} \begin{bmatrix} x(1) \\ x(3) \\ x(5) \end{bmatrix} \tag{2.10.8}$$

be the N/2 point DFTs.

2.11. By replacing the above results in the equation

$X = F_N x$, we get

$$\begin{bmatrix} X(0)\\ X(1)\\ X(2)\\ X(3)\\ X(4)\\ X(5) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & W_6^0 & 0 & 0\\ 0 & 1 & 0 & 0 & W_6^1 & 0\\ 0 & 0 & 1 & 0 & 0 & W_6^2\\ 1 & 0 & 0 & -W_6^0 & 0 & 0\\ 0 & 1 & 0 & 0 & -W_6^1 & 0\\ 0 & 0 & 1 & 0 & 0 & -W_6^2 \end{bmatrix} \begin{bmatrix} X_1(0)\\ X_1(1)\\ X_1(2)\\ X_2(0)\\ X_2(1)\\ X_2(2) \end{bmatrix} \tag{2.11.1}$$

2.12. Using the above method we have broken down an N-point DFT into 2 N/2-point DFTs

$$\begin{bmatrix} X(0)\\ X(1)\\ X(2) \end{bmatrix} = \begin{bmatrix} X_1(0)\\ X_1(1)\\ X_1(2) \end{bmatrix} + \begin{bmatrix} W_6^0 & 0 & 0\\ 0 & W_6^1 & 0\\ 0 & 0 & W_6^2 \end{bmatrix} \begin{bmatrix} X_2(0)\\ X_2(1)\\ X_2(2) \end{bmatrix} \tag{2.12.1}$$

$$\begin{bmatrix} X(3)\\ X(4)\\ X(5) \end{bmatrix} = \begin{bmatrix} X_1(0)\\ X_1(1)\\ X_1(2) \end{bmatrix} - \begin{bmatrix} W_6^0 & 0 & 0\\ 0 & W_6^1 & 0\\ 0 & 0 & W_6^2 \end{bmatrix} \begin{bmatrix} X_2(0)\\ X_2(1)\\ X_2(2) \end{bmatrix} \tag{2.12.2}$$

By doing this recursively,we can reduce our time complexity from O(N^2) to O(NlogN)

Now, say

$$N = 2^3 \tag{2.12.3}$$

2.13. Let x(n) = $\left\{\underset{\uparrow}{1}, 2, 3, 4, 2, 1, 0, 1\right\}$

2.14. The below code computes Y, for above mentioned x(n) and generates magnitude and phase plots of X, H, Y

https://github.com/Bhanuprakash072/dsp/tree/main/code/ee18btech11022_2.py

2.15. The following plots are obtained Solving it manually, Recursively breaking down,

$$F_8 = \begin{bmatrix} I_4 & D_4\\ I_4 & -D_4 \end{bmatrix} \begin{bmatrix} F_4 & 0\\ 0 & F_4 \end{bmatrix} P_8 \tag{2.15.1}$$

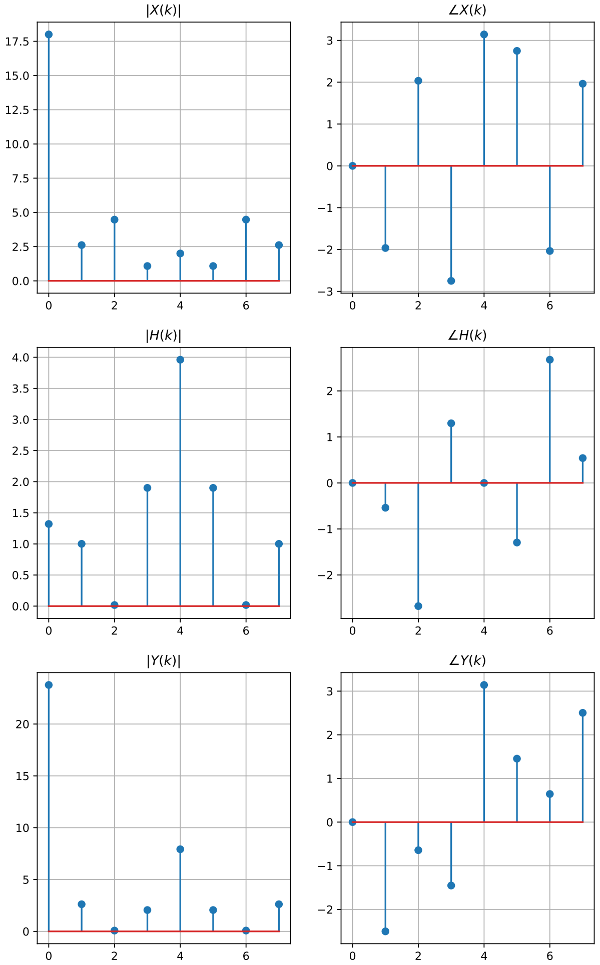
$$F_4 = \begin{bmatrix} I_2 & D_2\\ I_2 & -D_2 \end{bmatrix} \begin{bmatrix} F_2 & 0\\ 0 & F_2 \end{bmatrix} P_4 \tag{2.15.2}$$

F_2 is a base case

$$F_2 \begin{bmatrix} x\\ y \end{bmatrix} = \begin{bmatrix} 1 & 1\\ 1 & -1 \end{bmatrix} \begin{bmatrix} x\\ y \end{bmatrix} = \begin{bmatrix} x+y\\ x-y \end{bmatrix} \tag{2.15.3}$$

2.16. Solving 8-point FFTs into 4-point FFTs

$$\begin{bmatrix} X(0)\\ X(1)\\ X(2)\\ X(3) \end{bmatrix} = \begin{bmatrix} X_1(0)\\ X_1(1)\\ X_1(2)\\ X_1(3) \end{bmatrix} + \begin{bmatrix} W_8^0 & 0 & 0 & 0\\ 0 & W_8^1 & 0 & 0\\ 0 & 0 & W_8^2 & 0\\ 0 & 0 & 0 & W_8^3 \end{bmatrix} \begin{bmatrix} X_2(0)\\ X_2(1)\\ X_2(2)\\ X_2(3) \end{bmatrix} \tag{2.16.1}$$



$$\begin{bmatrix} X(4)\\ X(5)\\ X(6)\\ X(7) \end{bmatrix} = \begin{bmatrix} X_1(0)\\ X_1(1)\\ X_1(2)\\ X_1(3) \end{bmatrix} - \begin{bmatrix} W_8^0 & 0 & 0 & 0\\ 0 & W_8^1 & 0 & 0\\ 0 & 0 & W_8^2 & 0\\ 0 & 0 & 0 & W_8^3 \end{bmatrix} \begin{bmatrix} X_2(0)\\ X_2(1)\\ X_2(2)\\ X_2(3) \end{bmatrix} \tag{2.16.2}$$

4-point FFTs into 2-point FFTs

$$\begin{bmatrix} X_1(0)\\ X_1(1) \end{bmatrix} = \begin{bmatrix} X_3(0)\\ X_3(1) \end{bmatrix} + \begin{bmatrix} W_4^0 & 0\\ 0 & W_4^1 \end{bmatrix} \begin{bmatrix} X_4(0)\\ X_4(1) \end{bmatrix} \tag{2.16.3}$$

$$\begin{bmatrix} X_1(2)\\ X_1(3) \end{bmatrix} = \begin{bmatrix} X_3(0)\\ X_3(1) \end{bmatrix} - \begin{bmatrix} W_4^0 & 0\\ 0 & W_4^1 \end{bmatrix} \begin{bmatrix} X_4(0)\\ X_4(1) \end{bmatrix} \tag{2.16.4}$$

$$\begin{bmatrix} X_2(0)\\ X_2(1) \end{bmatrix} = \begin{bmatrix} X_5(0)\\ X_5(1) \end{bmatrix} + \begin{bmatrix} W_4^0 & 0\\ 0 & W_4^1 \end{bmatrix} \begin{bmatrix} X_6(0)\\ X_6(1) \end{bmatrix} \tag{2.16.5}$$

$$\begin{bmatrix} X_2(2) \\ X_2(3) \end{bmatrix} = \begin{bmatrix} X_5(0) \\ X_5(1) \end{bmatrix} - \begin{bmatrix} W_4^0 & 0 \\ 0 & W_4^1 \end{bmatrix} \begin{bmatrix} X_6(0) \\ X_6(1) \end{bmatrix} \tag{2.16.6}$$

$$P_8 \begin{bmatrix} x(0) \\ x(1) \\ x(2) \\ x(3) \\ x(4) \\ x(5) \\ x(6) \\ x(7) \end{bmatrix} = \begin{bmatrix} x(0) \\ x(2) \\ x(4) \\ x(6) \\ x(1) \\ x(3) \\ x(5) \\ x(7) \end{bmatrix} \tag{2.16.7}$$

$$P_4 \begin{bmatrix} x(0) \\ x(2) \\ x(4) \\ x(6) \end{bmatrix} = \begin{bmatrix} x(0) \\ x(4) \\ x(2) \\ x(6) \end{bmatrix} \tag{2.16.8}$$

$$P_4 \begin{bmatrix} x(1) \\ x(3) \\ x(5) \\ x(7) \end{bmatrix} = \begin{bmatrix} x(1) \\ x(5) \\ x(3) \\ x(7) \end{bmatrix} \tag{2.16.9}$$

Therefore,

$$\begin{bmatrix} X_3(0) \\ X_3(1) \end{bmatrix} = F_2 \begin{bmatrix} x(0) \\ x(4) \end{bmatrix} = \begin{bmatrix} x(0) + x(4) \\ x(0) - x(4) \end{bmatrix} \tag{2.16.10}$$

$$\begin{bmatrix} X_4(0) \\ X_4(1) \end{bmatrix} = F_2 \begin{bmatrix} x(2) \\ x(6) \end{bmatrix} = \begin{bmatrix} x(2) + x(6) \\ x(2) - x(6) \end{bmatrix} \tag{2.16.11}$$

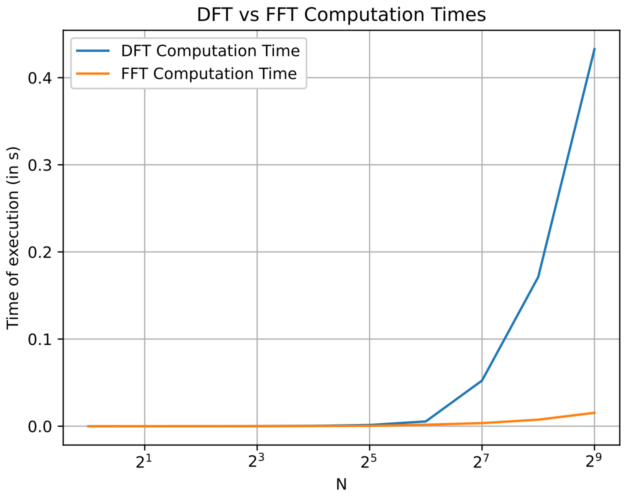
$$\begin{bmatrix} X_5(0) \\ X_5(1) \end{bmatrix} = F_2 \begin{bmatrix} x(1) \\ x(5) \end{bmatrix} = \begin{bmatrix} x(1) + x(5) \\ x(1) - x(5) \end{bmatrix} \tag{2.16.12}$$

$$\begin{bmatrix} X_6(0) \\ X_6(1) \end{bmatrix} = F_2 \begin{bmatrix} x(3) \\ x(7) \end{bmatrix} \tag{2.16.13}$$

2.17. See results in the plot, FFt lot faster

```
https://github.com/Bhanuprakash072/dsp/tree/
    main/code/ee18btech11022_3.py
```

2.18. The following plot is obtained



2.19. Because, In FFT - N-point FFT is broken down recursively into 2 N/2-point FFTs recursively.

$$T(n) = 2T(n/2) + O(n) \tag{2.19.1}$$

Solving this recurrence gives $O(NlogN)$ time complexity.

2.20. The following is the C program for FFT

```
https://github.com/Bhanuprakash072/dsp/tree/
    main/code/ee18btech11022.c
```