Assignment-1

Ganraj Borade - EE18BTECH11016

Download all python codes from

```
https://github.com/ganrajborade/EE3025/tree/main/A1/codes
```

and latex-tikz codes from

```
https://github.com/ganrajborade/EE3025/tree/main/A1
```

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

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

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

1.3. Compute

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

2 Solution

2.1. The Impulse Response of the LTI system is the output of the system when Unit Impulse Signal is given as input to the system. Impulse response h(n) can be found from given difference equation as follows (h(n) is IIR Filter)

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

2.2. DFT of a Input Signal $x(n)$ is

$$X(k) \triangleq \sum_{n=0}^{N-1} x(n)e^{-j2\pi kn/N}, \quad k = 0, 1, \dots, N-1 \tag{2.2.1}$$

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

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

$$X = \left[W_N^{ij}\right]_{N\times N} x, \quad i, j = 0, 1, \dots, N-1 \tag{2.3.1}$$

2.4. Using the property of complex exponentials :

$$W_N^2 = W_{N/2} \tag{2.4.1}$$

$$\mathcal{X}(k) = \sum_{n=0}^{N-1} x(n)W_N^{kn}, \quad k = 0, 1, \dots, N-1 \tag{2.4.2}$$

$$= \sum_{n=even} x(n)W_N^{kn} + \sum_{n=odd} x(n)W_N^{kn} \tag{2.4.3}$$

$$= \sum_{m=0}^{2} x(2m)W_N^{2mk} + \sum_{m=0}^{2} x(2m+1)W_N^{(2m+1)k} \tag{2.4.4}$$

using property above property, we get,

$$\mathcal{X}(k) = \sum_{m=0}^{2} x(2m)W_{N/2}^{mk} + W_N^k \sum_{m=0}^{2} x(2m+1)W_{N/2}^{mk} \tag{2.4.5}$$

$$= X_1(k) + W_N^k X_2(k) \tag{2.4.6}$$

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

Taking N = 6 and expressing the even odd DFT's $X_1(k)$, $X_2(k)$ interms of matrices we get,

2.5. Let F_N be the N-point DFT Matrix.

Using the property of Complex Exponentials we can express F_N in terms of $F_{N/2}$

$$F_N = \begin{bmatrix} I_{N/2} & D_{N/2} \\ I_{N/2} & -D_{N/2} \end{bmatrix} \begin{bmatrix} F_{N/2} & 0 \\ 0 & F_{N/2} \end{bmatrix} P_N \tag{2.5.1}$$

For N = 6

$$\implies F_6 = \begin{bmatrix} I_3 & D_3 \\ I_3 & -D_3 \end{bmatrix} \begin{bmatrix} F_3 & 0 \\ 0 & F_3 \end{bmatrix} P_6 \tag{2.5.2}$$

where I_3 is the 3x3 identity matrix. Writing some matrices in block form :

$$D_3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & W_3^1 & 0 \\ 0 & 0 & W_3^2 \end{bmatrix} \tag{2.5.3}$$

$$P_6 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.5.4}$$

$$\implies P_6 \begin{bmatrix} x(0) \\ x(1) \\ x(2) \\ x(3) \\ x(4) \\ x(5) \end{bmatrix} = \begin{bmatrix} x(0) \\ x(2) \\ x(4) \\ x(1) \\ x(3) \\ x(5) \end{bmatrix} \tag{2.5.5}$$

Let

$$\begin{bmatrix} X_1(0) \\ X_1(1) \\ X_1(2) \end{bmatrix} = F_{N/2} \begin{bmatrix} x(0) \\ x(2) \\ x(4) \end{bmatrix} \tag{2.5.6}$$

$$\begin{bmatrix} X_2(0) \\ X_2(1) \\ X_2(2) \end{bmatrix} = F_{N/2} \begin{bmatrix} x(1) \\ x(3) \\ x(5) \end{bmatrix} \tag{2.5.7}$$

be the N/2 point DFTs.

2.6. By replacing the above results in the equation $X = F_N x$, we get

$$\begin{bmatrix} X(0) \\ X(1) \\ X(2) \\ X(3) \\ X(4) \\ X(5) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & W_6^0 & 0 & 0 \\ 0 & 1 & 0 & 0 & W_6^1 & 0 \\ 0 & 0 & 1 & 0 & 0 & W_6^2 \\ 1 & 0 & 0 & -W_6^0 & 0 & 0 \\ 0 & 1 & 0 & 0 & -W_6^1 & 0 \\ 0 & 0 & 1 & 0 & 0 & -W_6^2 \end{bmatrix} \begin{bmatrix} X_1(0) \\ X_1(1) \\ X_1(2) \\ X_2(0) \\ X_2(1) \\ X_2(2) \end{bmatrix} \tag{2.6.1}$$

Broke N-point DFT into 2 N/2-point DFTs using above method

$$\begin{bmatrix} X(0) \\ X(1) \\ X(2) \end{bmatrix} = \begin{bmatrix} X_1(0) \\ X_1(1) \\ X_1(2) \end{bmatrix} + \begin{bmatrix} W_6^0 & 0 & 0 \\ 0 & W_6^1 & 0 \\ 0 & 0 & W_6^2 \end{bmatrix} \begin{bmatrix} X_2(0) \\ X_2(1) \\ X_2(2) \end{bmatrix} \tag{2.6.2}$$

$$\begin{bmatrix} X(3) \\ X(4) \\ X(5) \end{bmatrix} = \begin{bmatrix} X_1(0) \\ X_1(1) \\ X_1(2) \end{bmatrix} - \begin{bmatrix} W_6^0 & 0 & 0 \\ 0 & W_6^1 & 0 \\ 0 & 0 & W_6^2 \end{bmatrix} \begin{bmatrix} X_2(0) \\ X_2(1) \\ X_2(2) \end{bmatrix} \tag{2.6.3}$$

We can reduce our time complexity from O(N^2) to O(NlogN) by doing this.

2.7. Now, if $N = 2^M$ where $M \in \mathbb{Z}^+$ then we can recursively breakdown N/2 point DFT Matrix to N/4 point DFT Matrix ..so on till we reach 2-point DFT Matrix.So for N = 8, we can write,

$$F_8 = \begin{bmatrix} I_4 & D_4 \\ I_4 & -D_4 \end{bmatrix} \begin{bmatrix} F_4 & 0 \\ 0 & F_4 \end{bmatrix} P_8 \tag{2.7.1}$$

$$F_4 = \begin{bmatrix} I_2 & D_2 \\ I_2 & -D_2 \end{bmatrix} \begin{bmatrix} F_2 & 0 \\ 0 & F_2 \end{bmatrix} P_4 \tag{2.7.2}$$

Finally, the 2-point DFT Matrix is the base case

$$F_2 \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} x_1 + x_2 \\ x_1 - x_2 \end{bmatrix} \tag{2.7.3}$$

2.8. Step by Step visualization of computing 8-Point DFT recursively using 4-point DFT's and 2-point DFT's.Expressing 8-point DFT's in terms of 4-point DFT's.

$$\begin{bmatrix} X(0) \\ X(1) \\ X(2) \\ X(3) \end{bmatrix} = \begin{bmatrix} X_e(0) \\ X_e(1) \\ X_e(2) \\ X_e(3) \end{bmatrix} + \begin{bmatrix} W_8^0 & 0 & 0 & 0 \\ 0 & W_8^1 & 0 & 0 \\ 0 & 0 & W_8^2 & 0 \\ 0 & 0 & 0 & W_8^3 \end{bmatrix} \begin{bmatrix} X_o(0) \\ X_o(1) \\ X_o(2) \\ X_o(3) \end{bmatrix} \tag{2.8.1}$$

$$\begin{bmatrix} X(4) \\ X(5) \\ X(6) \\ X(7) \end{bmatrix} = \begin{bmatrix} X_e(0) \\ X_e(1) \\ X_e(2) \\ X_e(3) \end{bmatrix} - \begin{bmatrix} W_8^0 & 0 & 0 & 0 \\ 0 & W_8^1 & 0 & 0 \\ 0 & 0 & W_8^2 & 0 \\ 0 & 0 & 0 & W_8^3 \end{bmatrix} \begin{bmatrix} X_o(0) \\ X_o(1) \\ X_o(2) \\ X_o(3) \end{bmatrix} \tag{2.8.2}$$

Now, 4-point DFT's to 2-point DFT's

$$\begin{bmatrix} X_e(0) \\ X_e(1) \end{bmatrix} = \begin{bmatrix} X_{e_1}(0) \\ X_{e_1}(1) \end{bmatrix} + \begin{bmatrix} W_4^0 & 0 \\ 0 & W_4^1 \end{bmatrix} \begin{bmatrix} X_{o_1}(0) \\ X_{o_1}(1) \end{bmatrix} \tag{2.8.3}$$

$$\begin{bmatrix} X_e(2) \\ X_e(3) \end{bmatrix} = \begin{bmatrix} X_{e_1}(0) \\ X_{e_1}(1) \end{bmatrix} - \begin{bmatrix} W_4^0 & 0 \\ 0 & W_4^1 \end{bmatrix} \begin{bmatrix} X_{o_1}(0) \\ X_{o_1}(1) \end{bmatrix} \tag{2.8.4}$$

$$\begin{bmatrix} X_o(0) \\ X_o(1) \end{bmatrix} = \begin{bmatrix} X_{e_2}(0) \\ X_{e_2}(1) \end{bmatrix} + \begin{bmatrix} W_4^0 & 0 \\ 0 & W_4^1 \end{bmatrix} \begin{bmatrix} X_{o_2}(0) \\ X_{o_2}(1) \end{bmatrix} \tag{2.8.5}$$

$$\begin{bmatrix} X_o(2) \\ X_o(3) \end{bmatrix} = \begin{bmatrix} X_{e_2}(0) \\ X_{e_2}(1) \end{bmatrix} - \begin{bmatrix} W_4^0 & 0 \\ 0 & W_4^1 \end{bmatrix} \begin{bmatrix} X_{o_2}(0) \\ X_{o_2}(1) \end{bmatrix} \tag{2.8.6}$$

$$P_8 \begin{bmatrix} x(0) \\ x(1) \\ x(2) \\ x(3) \\ x(4) \\ x(5) \\ x(6) \\ x(7) \end{bmatrix} = \begin{bmatrix} x(0) \\ x(2) \\ x(4) \\ x(6) \\ x(1) \\ x(3) \\ x(5) \\ x(7) \end{bmatrix} \tag{2.8.7}$$

$$P_4\begin{bmatrix}x(0)\\x(2)\\x(4)\\x(6)\end{bmatrix}=\begin{bmatrix}x(0)\\x(4)\\x(2)\\x(6)\end{bmatrix} \quad (2.8.8)$$

$$P_4\begin{bmatrix}x(1)\\x(3)\\x(5)\\x(7)\end{bmatrix}=\begin{bmatrix}x(1)\\x(5)\\x(3)\\x(7)\end{bmatrix} \quad (2.8.9)$$

Finally,

$$\begin{bmatrix}X_{e_1}(0)\\X_{e_1}(1)\end{bmatrix}=F_2\begin{bmatrix}x(0)\\x(4)\end{bmatrix}=\begin{bmatrix}x(0)+x(4)\\x(0)-x(4)\end{bmatrix} \quad (2.8.10)$$

$$\begin{bmatrix}X_{o_1}(0)\\X_{o_1}(1)\end{bmatrix}=F_2\begin{bmatrix}x(2)\\x(6)\end{bmatrix}=\begin{bmatrix}x(2)+x(6)\\x(2)-x(6)\end{bmatrix} \quad (2.8.11)$$

$$\begin{bmatrix}X_{e_2}(0)\\X_{e_2}(1)\end{bmatrix}=F_2\begin{bmatrix}x(1)\\x(5)\end{bmatrix}=\begin{bmatrix}x(1)+x(5)\\x(1)-x(5)\end{bmatrix} \quad (2.8.12)$$

$$\begin{bmatrix}X_{o_2}(0)\\X_{o_2}(1)\end{bmatrix}=F_2\begin{bmatrix}x(3)\\x(7)\end{bmatrix}=\begin{bmatrix}x(3)+x(7)\\x(3)-x(7)\end{bmatrix} \quad (2.8.13)$$

So, $X_{e_2} \in \text{DFT}\{x(1), x(5)\}$ and $X_{o_2} \in \text{DFT}\{x(3), x(7)\}$ would combine to give X_o . And $X_{e_1} \in \text{DFT}\{x(0), x(4)\}$ and $X_{o_1} \in \text{DFT}\{x(2), x(6)\}$ would combine to give X_e.

2.9. The following C program will compute and print the FFT (N-point where N is of the form 2^n)

```
https://github.com/ganrajborade/EE3025/blob/
main/A1/codes/ee18btech11016_fft.c
```

2.10. *Time Complexity:* Matrix multiplication of NxN matrix with Nx1 vector is there in DFT. Hence it has $O(N^2)$ time complexity which is very slow for high N.
In this recursive approach which is termed as FFT - N-point FFT is broken down recursively into 2 N/2-point FFTs recursively.
Additionally $O(N)$ operation of Vector multiplication is performed on the N/2 point FFTs.

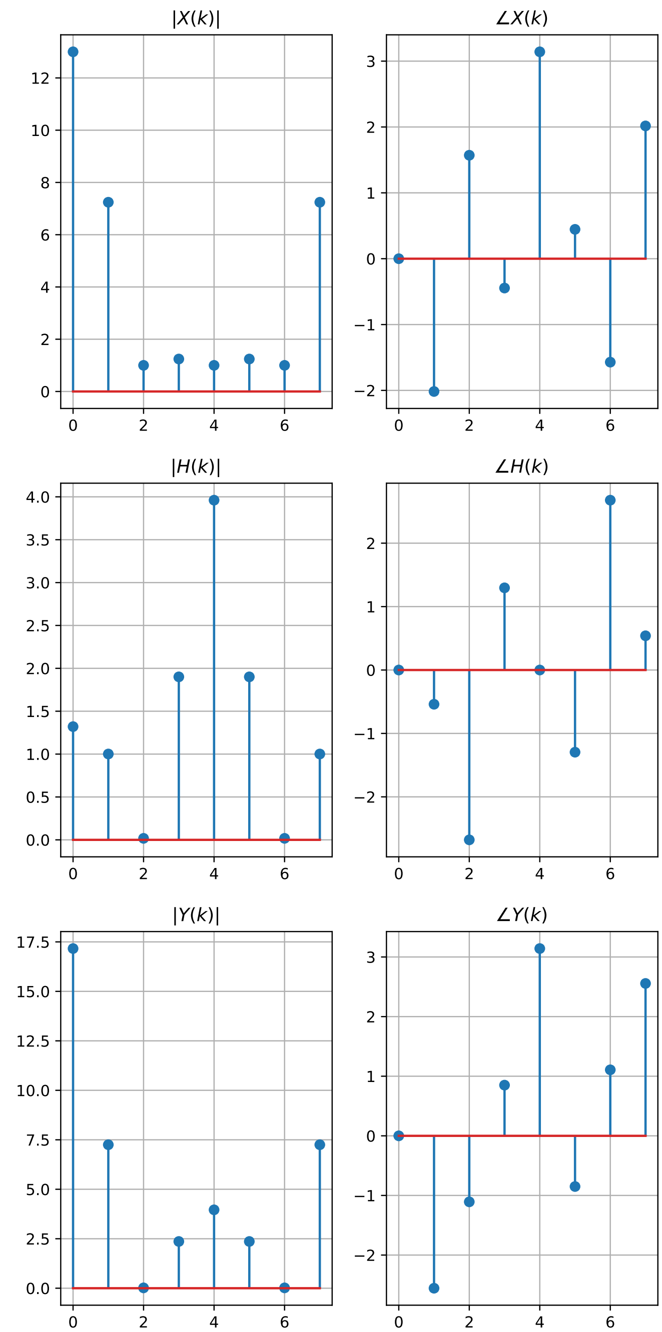
$$T(n) = 2T(n/2) + O(n) \quad (2.10.1)$$

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

2.11. Computing $X(k)$, $H(k)$ and $Y(k)$ for

$$x(n) = \left\{\underset{\uparrow}{1}, 2, 3, 4, 2, 1, 4, 3\right\} \quad (2.11.1)$$

with N = 8, using above FFT approach. The following plots are obtained from the code given below:



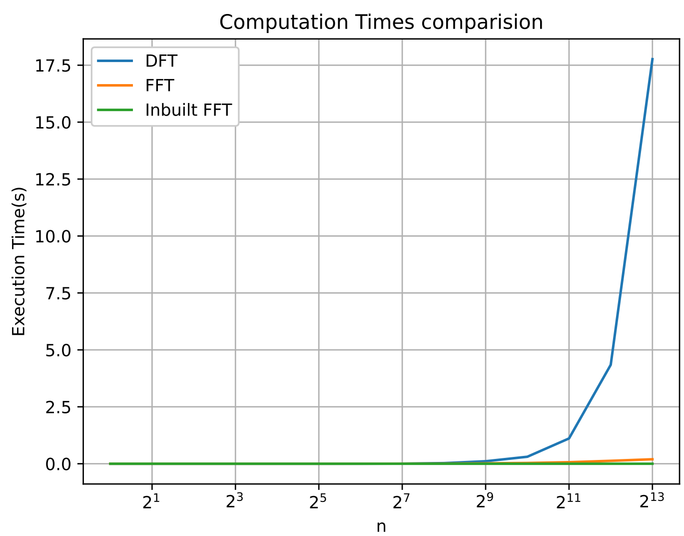
2.12. This code plots above magnitude and phase plots of $X(k)$, $H(k)$ and $Y(k)$.

https://github.com/ganrajborade/EE3025/blob/main/A1/codes/ee18btech11016_1.py

2.13. *Computation Times:* We can compare the computation times for DFT,FFT and Inbuit-FFT algorithms for N = 2^M, for N = 1 (2^0) to 8192 (2^{13}).

2.14. The below code plots the above time comparision plot.

https://github.com/ganrajborade/EE3025/blob/main/A1/codes/ee18btech11016_2.py



2.15. Obtaining 8-Point FFT using DFT Matrix

$$\begin{bmatrix} X(0)\\X(1)\\X(2)\\X(3)\\X(4)\\X(5)\\X(6)\\X(7)\end{bmatrix} = \begin{bmatrix} W_8^0 & W_8^0 & W_8^0 & W_8^0 & W_8^0 & W_8^0 & W_8^0 & W_8^0\\ W_8^0 & W_8^1 & W_8^2 & W_8^3 & W_8^4 & W_8^5 & W_8^6 & W_8^7\\ W_8^0 & W_8^2 & W_8^4 & W_8^6 & W_8^8 & W_8^{10} & W_8^{12} & W_8^{14}\\ W_8^0 & W_8^3 & W_8^6 & W_8^9 & W_8^{12} & W_8^{15} & W_8^{18} & W_8^{21}\\ W_8^0 & W_8^4 & W_8^8 & W_8^{12} & W_8^{16} & W_8^{20} & W_8^{24} & W_8^{28}\\ W_8^0 & W_8^5 & W_8^{10} & W_8^{15} & W_8^{20} & W_8^{25} & W_8^{30} & W_8^{35}\\ W_8^0 & W_8^6 & W_8^{12} & W_8^{18} & W_8^{24} & W_8^{30} & W_8^{36} & W_8^{42}\\ W_8^0 & W_8^7 & W_8^{14} & W_8^{21} & W_8^{28} & W_8^{35} & W_8^{42} & W_8^{49}\end{bmatrix} \begin{bmatrix}1\\2\\3\\4\\2\\1\\0\\0\end{bmatrix} \tag{2.15.1}$$

$$\implies \begin{bmatrix} X(0)\\X(1)\\X(2)\\X(3)\\X(4)\\X(5)\\X(6)\\X(7)\end{bmatrix} = \begin{bmatrix} 13\\ -3.121-6.535j\\ 1j\\ 1.121-0.535j\\ -1\\ 1.121+0.535j\\ -1j\\ -3.121+6.535j\end{bmatrix} \tag{2.15.2}$$

Similarly,

$$\begin{bmatrix} H(0)\\H(1)\\H(2)\\H(3)\\H(4)\\H(5)\\H(6)\\H(7)\end{bmatrix} = \begin{bmatrix} W_8^0 & W_8^0 & W_8^0 & W_8^0 & W_8^0 & W_8^0 & W_8^0 & W_8^0\\ W_8^0 & W_8^1 & W_8^2 & W_8^3 & W_8^4 & W_8^5 & W_8^6 & W_8^7\\ W_8^0 & W_8^2 & W_8^4 & W_8^6 & W_8^8 & W_8^{10} & W_8^{12} & W_8^{14}\\ W_8^0 & W_8^3 & W_8^6 & W_8^9 & W_8^{12} & W_8^{15} & W_8^{18} & W_8^{21}\\ W_8^0 & W_8^4 & W_8^8 & W_8^{12} & W_8^{16} & W_8^{20} & W_8^{24} & W_8^{28}\\ W_8^0 & W_8^5 & W_8^{10} & W_8^{15} & W_8^{20} & W_8^{25} & W_8^{30} & W_8^{35}\\ W_8^0 & W_8^6 & W_8^{12} & W_8^{18} & W_8^{24} & W_8^{30} & W_8^{36} & W_8^{42}\\ W_8^0 & W_8^7 & W_8^{14} & W_8^{21} & W_8^{28} & W_8^{35} & W_8^{42} & W_8^{49}\end{bmatrix} \tag{2.15.3}$$

$$\implies \begin{bmatrix} H(0)\\H(1)\\H(2)\\H(3)\\H(4)\\H(5)\\H(6)\\H(7)\end{bmatrix} = \begin{bmatrix} 1.32\\ 0.858-0.514j\\ -0.015-0.007j\\ 0.516+1.829j\\ 3.96\\ 0.516-1.829j\\ -0.015+0.007j\\ 0.858+0.514j\end{bmatrix} \tag{2.15.4}$$

So,

$$\begin{bmatrix} Y(0)\\Y(1)\\Y(2)\\Y(3)\\Y(4)\\Y(5)\\Y(6)\\Y(7)\end{bmatrix} = \begin{bmatrix} X(0)\cdot H(0)\\ X(1)\cdot H(1)\\ X(2)\cdot H(2)\\ X(3)\cdot H(3)\\ X(4)\cdot H(4)\\ X(5)\cdot H(5)\\ X(6)\cdot H(6)\\ X(7)\cdot H(7)\end{bmatrix} \tag{2.15.5}$$

$$\implies \begin{bmatrix} Y(0)\\Y(1)\\Y(2)\\Y(3)\\Y(4)\\Y(5)\\Y(6)\\Y(7)\end{bmatrix} = \begin{bmatrix} 17.16\\ -6.04-4j\\ -0.007-0.015j\\ 1.55+1.77j\\ -3.96\\ 1.55-1.77j\\ 0.007+0.015j\\ -6.04+4j\end{bmatrix} \tag{2.15.6}$$