EE3025 IDP Assignment-1

Dasari Shree Ujjwal - EE18BTECH11010

Download all python codes from

https://github.com/dsujjwal/IDP_DSP/tree/main/Assignment1/Codes

and latex-tikz codes from

https://github.com/dsujjwal/IDP_DSP/tree/main/Assignment1

1 Question

1.1. Let

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

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

Compute

$$X(k) \triangleq \sum_{n=0}^{N-1} x(n)e^{-j2\pi kn/N}, \quad k = 0, 1, \dots, N-1 \tag{1.1.3}$$

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

2 Solution

2.1. For any LTI System, if the Unit Impulse signal is given as the input, then its output is the Impulse response of the system. Now, from equation (1.1.2) we get the Impulse response of the system as :

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

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

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

Similarly, DFT of Impulse Response $h(n)$ is,

$$H(k) \triangleq \sum_{n=0}^{N-1} h(n)e^{-j2\pi kn/N}, \quad k = 0, 1, \dots, N-1 \tag{2.2.2}$$

The following python code computes the DFT of x(n) and h(n).

https://github.com/dsujjwal/IDP_DSP/blob/main/Assignment1/Codes/ee18btech11010.py

The plots are in

https://github.com/dsujjwal/IDP_DSP/tree/main/Assignment1/figs

2.3. DFT of x(n) and h(n) using Matrix multiplication method :

$$X(k) \triangleq \sum_{n=0}^{N-1} x(n)e^{-j2\pi kn/N}, \quad k = 0, 1, \dots, N-1 \tag{2.3.1}$$

Let us define a matrix 'W', a NxN matrix, such that,

$$DFT(x) = X = W.x \tag{2.3.2}$$

Here (.) represents the multiplication of Matrix W and vector x.
Every entry of the matrix W is defined as,

$$W_N^{nk} = e^{-j2\pi kn/N} \tag{2.3.3}$$

$$\tag{2.3.4}$$

Here n and k are column and row indices of the matrix respectively.

Now from equation: (2.3.2), DFT(x) using the DFT matrix is given by,

$$\begin{bmatrix} X(0) \\ X(1) \\ X(2) \\ \vdots \\ X(N-1) \end{bmatrix} = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ 1 & W_N^1 & \cdots & W_N^{N-1} \\ 1 & W_N^2 & \cdots & W_N^{2(N-1)} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & W_N^{N-1} & \cdots & W_N^{(N-1)(N-1)} \end{bmatrix} \begin{bmatrix} x(0) \\ x(1) \\ x(2) \\ \vdots \\ x(N-1) \end{bmatrix} \tag{2.3.5}$$

Given, x(n) = $\left\{\underset{\uparrow}{1}, 2, 3, 4, 2, 1\right\}$ and N=6 we have,

$$\begin{bmatrix} X(0) \\ X(1) \\ X(2) \\ X(3) \\ X(4) \\ X(5) \end{bmatrix} = \begin{bmatrix} 1+2+3+4+2+1 \\ 1+(2)e^{-j\pi/3}+...+(1)e^{-j5\pi/3} \\ 1+(2)e^{-2j\pi/3}+...+(1)(e^{-2j5\pi/3} \\ 1+(2)e^{-3j\pi/3}+...+(1)e^{-3j5\pi/3} \\ 1+(2)e^{-4j\pi/3}+...+(1)e^{-4j5\pi/3} \\ 1+(2)e^{-5j\pi/3}+...+(1)e^{-5j5\pi/3} \end{bmatrix} \tag{2.3.6}$$

On Solving we get,

$$X(0) = 13 + 0j \tag{2.3.7}$$
$$X(1) = -4 - 1.732j \tag{2.3.8}$$
$$X(2) = 1 + 0j \tag{2.3.9}$$
$$X(3) = -1 + 0j \tag{2.3.10}$$
$$X(4) = 1 + 0j \tag{2.3.11}$$
$$X(5) = -4 + 1.732j \tag{2.3.12}$$

Similarly, the DFT of h(n) can be defined as,

$$DFT(h) = H = W.h \tag{2.3.13}$$

So, we have,

$$\begin{bmatrix} H(0) \\ H(1) \\ H(2) \\ \vdots \\ H(N-1) \end{bmatrix} = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ 1 & W_N^1 & \cdots & W_N^{N-1} \\ 1 & W_N^2 & \cdots & W_N^{2(N-1)} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & W_N^{N-1} & \cdots & W_N^{(N-1)(N-1)} \end{bmatrix} \begin{bmatrix} h(0) \\ h(1) \\ h(2) \\ \vdots \\ h(N-1) \end{bmatrix} \tag{2.3.14}$$

$$\begin{bmatrix} H(0) \\ H(1) \\ H(2) \\ H(3) \\ H(4) \\ H(5) \end{bmatrix} = \begin{bmatrix} h(0)+h(1)+h(2)+h(3)+h(4)+h(5) \\ h(0)+h(1)e^{-j\pi/3}+...+h(5)e^{-j5\pi/3} \\ h(0)+h(1)e^{-2j\pi/3}+...+h(5)e^{-2j5\pi/3} \\ h(0)+h(1)e^{-3j\pi/3}+...+h(5)e^{-3j5\pi/3} \\ h(0)+h(1)e^{-4j\pi/3}+...+h(5)e^{-4j5\pi/3} \\ h(0)+h(1)e^{-5j\pi/3}+...+h(5)e^{-5j5\pi/3} \end{bmatrix} \tag{2.3.15}$$

We know from above that,

$$h(n) = \delta(n) + \delta(n-2) - \frac{1}{2}h(n-1) \tag{2.3.16}$$

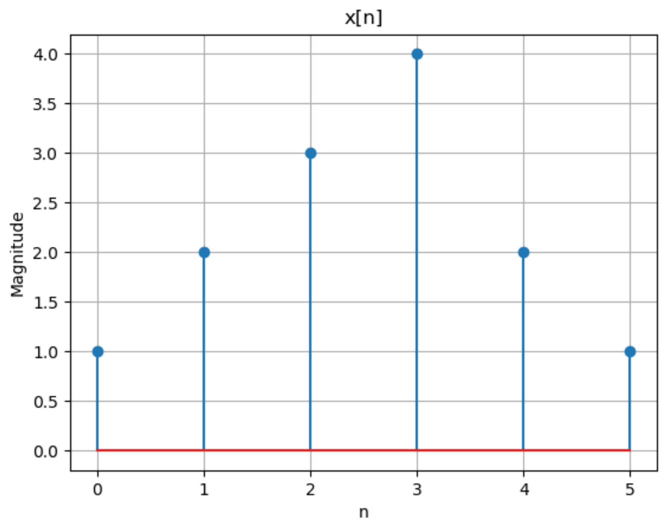
for N=6,

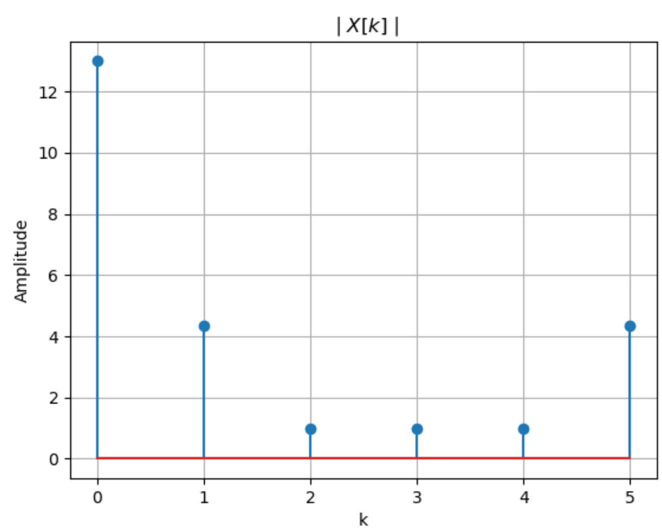
$$h(n) = \left\{ \underset{\uparrow}{1}, -0.5, 1.25, -0.625, 0.3125, -0.15625 \right\} \tag{2.3.17}$$

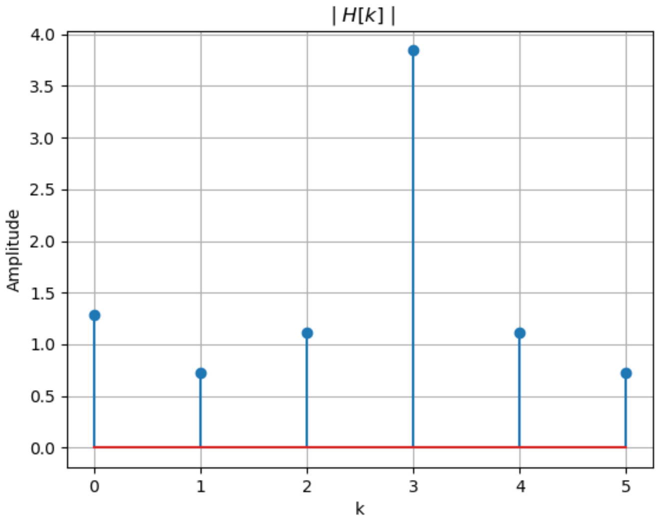
Therefore, on solving we get,

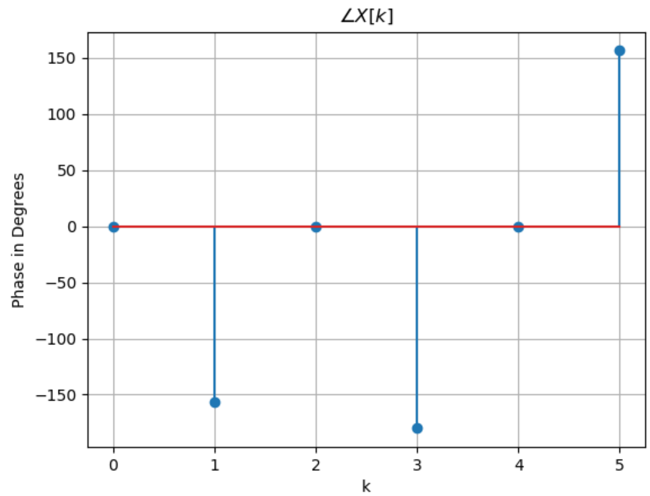
$$H(0) = 1.28125 + 0j, \tag{2.3.18}$$
$$H(1) = 0.515625 - 0.5142026j, \tag{2.3.19}$$
$$H(2) = -0.078125 + 1.109595j, \tag{2.3.20}$$
$$H(3) = 3.84375 + 0j, \tag{2.3.21}$$
$$H(4) = -0.078125 - 1.109595j. \tag{2.3.22}$$
$$H(5) = 0.515625 + 0.51420256j \tag{2.3.23}$$

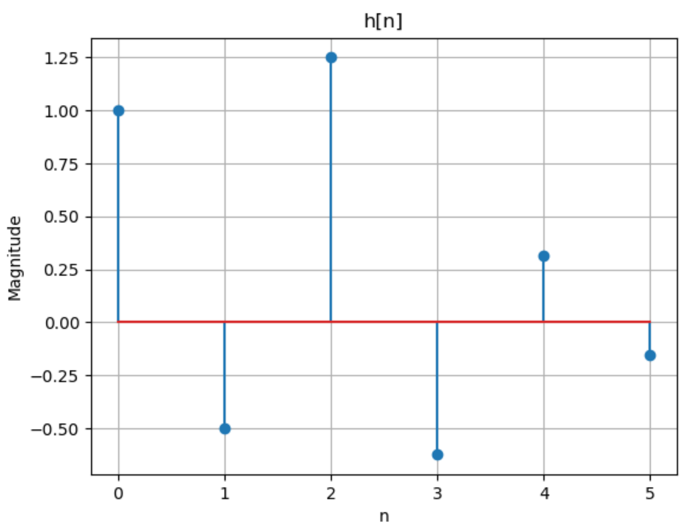
We can observe that the values of X and H are same as the values that we obtained from the plots.

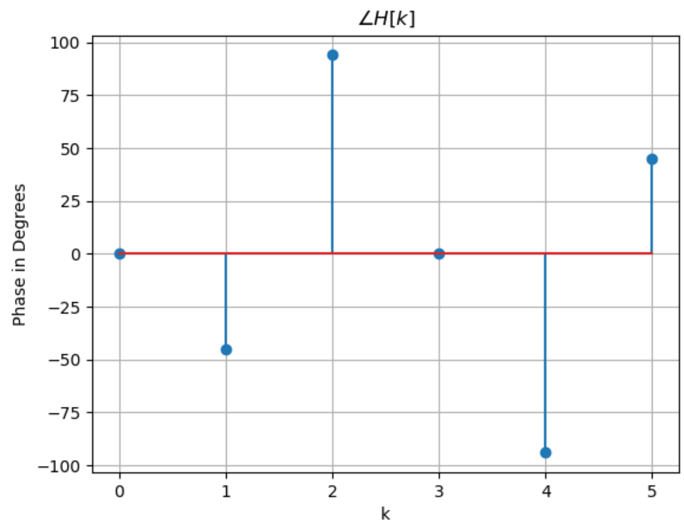












3 8 Point FFT

3.1. Properties of W_N

a) Property-1 :

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

b) Property-2 :

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

c) Property-3 :

$$W_{N/2}^{k+N/2} = W_{N/2}^k$$

3.2. a N-point DFT is computed as,

$$X(k) = \sum_{n=0}^{N-1} x(n)W_N^{nk}, \quad k = 0, 1, \ldots, N-1 \tag{3.2.1}$$

For easier calculations we can Divide the inputs into even and odd indices and using

Property-1 (above),

$$X(k) = \sum_{n=even} x(n)W_N^{kn} + \sum_{n=odd} x(n)W_N^{kn} \quad (3.2.2)$$

$$= \sum_{m=0}^{N/2-1} x(2m)W_N^{2mk} + \sum_{m=0}^{N/2-1} x(2m+1)W_N^{(2m+1)k} \quad (3.2.3)$$

$$= \sum_{m=0}^{N/2-1} x(2m)W_{N/2}^{mk} + W_N^k \sum_{m=0}^{N/2-1} x(2m+1)W_{N/2}^{mk} \quad (3.2.4)$$

Observing,

$$X(k) = X_e(k) + W_N^k X_o(k) \quad (3.2.5)$$

Now for the given question N = 6, we can express the even odd DFT's $X_e(k)$, $X_o(k)$ in terms of block matrices,obtaining,

$$\begin{bmatrix} X_e(0)\\ X_e(1)\\ X_e(2)\\ X_o(0)\\ X_o(1)\\ X_o(2) \end{bmatrix} = \begin{bmatrix} W_3^0 & W_3^0 & W_3^0 & 0 & 0 & 0\\ W_3^0 & W_3^1 & W_3^2 & 0 & 0 & 0\\ W_3^0 & W_3^2 & W_3^4 & 0 & 0 & 0\\ 0 & 0 & 0 & W_3^0 & W_3^0 & W_3^0\\ 0 & 0 & 0 & W_3^0 & W_3^1 & W_3^2\\ 0 & 0 & 0 & W_3^0 & W_3^2 & W_3^4 \end{bmatrix} \begin{bmatrix} x(0)\\ x(2)\\ x(4)\\ x(1)\\ x(3)\\ x(5) \end{bmatrix} \quad (3.2.6)$$

Lets take , F_N as a N-Point DFT Matrix and P_N as an odd-even Permutation Matrix, we can write the above equation as

$$\begin{bmatrix} X_e(0)\\ X_e(1)\\ X_e(2)\\ X_o(0)\\ X_o(1)\\ X_o(2) \end{bmatrix} = \begin{bmatrix} F_3 & 0\\ 0 & F_3 \end{bmatrix} P_6 x \quad (3.2.7)$$

where

$$P_6 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0\\ 0 & 0 & 1 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 1 & 0\\ 0 & 1 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 1 & 0 & 0\\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.2.8)$$

and

$$P_6 x = P_6 \begin{bmatrix} x(0)\\ x(1)\\ x(2)\\ x(3)\\ x(4)\\ x(5) \end{bmatrix} = \begin{bmatrix} x(0)\\ x(2)\\ x(4)\\ x(1)\\ x(3)\\ x(5) \end{bmatrix} \quad (3.2.9)$$

Now,using (3.2.5) we can formulate $X(k)$ in-terms of $X_e(k)$ and $X_o(k)$

$$\begin{bmatrix} X(0)\\ X(1)\\ X(2)\\ X(3)\\ X(4)\\ X(5) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & W_6^0 & 0 & 0\\ 0 & 1 & 0 & 0 & W_6^1 & 0\\ 0 & 0 & 1 & 0 & 0 & W_6^2\\ 1 & 0 & 0 & W_6^3 & 0 & 0\\ 0 & 1 & 0 & 0 & W_6^4 & 0\\ 0 & 0 & 1 & 0 & 0 & W_6^5 \end{bmatrix} \begin{bmatrix} X_e(0)\\ X_e(1)\\ X_e(2)\\ X_o(0)\\ X_o(1)\\ X_o(2) \end{bmatrix} \quad (3.2.10)$$

Let I_3 to be 3x3 identity matrix and $D_N = diag(1, W_N, W_N^2, ..., W_N^{N-1})$ .accordingly we get $D_{\frac{N}{2}} = diag(1, W_N, W_N^2, ..., W_N^{\frac{N}{2}-1})$.so D_3 will be,

$$D_3 = \begin{bmatrix} 1 & 0 & 0\\ 0 & W_6^1 & 0\\ 0 & 0 & W_6^2 \end{bmatrix} \quad (3.2.11)$$

Using Property-2, Equation (3.2.10) gets expressed in terms of D_3 and I_3 as,

$$X = \begin{bmatrix} X(0)\\ X(1)\\ X(2)\\ X(3)\\ X(4)\\ X(5) \end{bmatrix} = \begin{bmatrix} I_3 & D_3\\ I_3 & -D_3 \end{bmatrix} \begin{bmatrix} X_e(0)\\ X_e(1)\\ X_e(2)\\ X_o(0)\\ X_o(1)\\ X_o(2) \end{bmatrix} \quad (3.2.12)$$

Using Eq (3.2.12) and Eq (3.2.7) we obtain

$$X = \begin{bmatrix} I_3 & D_3\\ I_3 & -D_3 \end{bmatrix} \begin{bmatrix} F_3 & 0\\ 0 & F_3 \end{bmatrix} P_6 x \quad (3.2.13)$$

we know $X = F_6 x$ for N = 6 hence we obtain F_6 as ;

$$\implies F_6 = \begin{bmatrix} I_3 & D_3\\ I_3 & -D_3 \end{bmatrix} \begin{bmatrix} F_3 & 0\\ 0 & F_3 \end{bmatrix} P_6 \quad (3.2.14)$$

above approach can be used for any arbitary N , lets take a N-point DFT Matrix and express it in terms of N/2-point DFT Matrix as

$$F_N = \begin{bmatrix} I_{N/2} & D_{N/2}\\ I_{N/2} & -D_{N/2} \end{bmatrix} \begin{bmatrix} F_{N/2} & 0\\ 0 & F_{N/2} \end{bmatrix} P_N \quad (3.2.15)$$

3.3. Now, for any $N = 2^m$ where $m \in \mathbb{Z}^+$ we can recursively breakdown N/2 point DFT Matrix to N/4 point DFT Matrix . so on until we reach 2-point DFT Matrix.for N = 8, we can break it down using Eq (3.2.15) as follows,

$$F_8 = \begin{bmatrix} I_4 & D_4\\ I_4 & -D_4 \end{bmatrix} \begin{bmatrix} F_4 & 0\\ 0 & F_4 \end{bmatrix} P_8 \quad (3.3.1)$$

$$F_4 = \begin{bmatrix} I_2 & D_2 \\ I_2 & -D_2 \end{bmatrix} \begin{bmatrix} F_2 & 0 \\ 0 & F_2 \end{bmatrix} P_4 \tag{3.3.2}$$

Finally,we reach the 2-point DFT Matrix base case

$$F_2 \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} x_1 + x_2 \\ x_1 - x_2 \end{bmatrix} \tag{3.3.3}$$

3.4. Computing the 8-point DFT using Step by Step visualization of recursion.

$$\begin{bmatrix} X(0) \\ X(1) \\ X(2) \\ X(3) \end{bmatrix} = \begin{bmatrix} X_e(0) \\ X_e(1) \\ X_e(2) \\ X_e(3) \end{bmatrix} + \begin{bmatrix} W_8^0 & 0 & 0 & 0 \\ 0 & W_8^1 & 0 & 0 \\ 0 & 0 & W_8^2 & 0 \\ 0 & 0 & 0 & W_8^3 \end{bmatrix} \begin{bmatrix} X_o(0) \\ X_o(1) \\ X_o(2) \\ X_o(3) \end{bmatrix} \tag{3.4.1}$$

$$\begin{bmatrix} X(4) \\ X(5) \\ X(6) \\ X(7) \end{bmatrix} = \begin{bmatrix} X_e(0) \\ X_e(1) \\ X_e(2) \\ X_e(3) \end{bmatrix} - \begin{bmatrix} W_8^0 & 0 & 0 & 0 \\ 0 & W_8^1 & 0 & 0 \\ 0 & 0 & W_8^2 & 0 \\ 0 & 0 & 0 & W_8^3 \end{bmatrix} \begin{bmatrix} X_o(0) \\ X_o(1) \\ X_o(2) \\ X_o(3) \end{bmatrix} \tag{3.4.2}$$

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

$$\begin{bmatrix} X_e(0) \\ X_e(1) \end{bmatrix} = \begin{bmatrix} X_{e_1}(0) \\ X_{e_1}(1) \end{bmatrix} + \begin{bmatrix} W_4^0 & 0 \\ 0 & W_4^1 \end{bmatrix} \begin{bmatrix} X_{o_1}(0) \\ X_{o_1}(1) \end{bmatrix} \tag{3.4.3}$$

$$\begin{bmatrix} X_e(2) \\ X_e(3) \end{bmatrix} = \begin{bmatrix} X_{e_1}(0) \\ X_{e_1}(1) \end{bmatrix} - \begin{bmatrix} W_4^0 & 0 \\ 0 & W_4^1 \end{bmatrix} \begin{bmatrix} X_{o_1}(0) \\ X_{o_1}(1) \end{bmatrix} \tag{3.4.4}$$

$$\begin{bmatrix} X_o(0) \\ X_o(1) \end{bmatrix} = \begin{bmatrix} X_{e_2}(0) \\ X_{e_2}(1) \end{bmatrix} + \begin{bmatrix} W_4^0 & 0 \\ 0 & W_4^1 \end{bmatrix} \begin{bmatrix} X_{o_2}(0) \\ X_{o_2}(1) \end{bmatrix} \tag{3.4.5}$$

$$\begin{bmatrix} X_o(2) \\ X_o(3) \end{bmatrix} = \begin{bmatrix} X_{e_2}(0) \\ X_{e_2}(1) \end{bmatrix} - \begin{bmatrix} W_4^0 & 0 \\ 0 & W_4^1 \end{bmatrix} \begin{bmatrix} X_{o_2}(0) \\ X_{o_2}(1) \end{bmatrix} \tag{3.4.6}$$

$$P_8 \begin{bmatrix} x(0) \\ x(1) \\ x(2) \\ x(3) \\ x(4) \\ x(5) \\ x(6) \\ x(7) \end{bmatrix} = \begin{bmatrix} x(0) \\ x(2) \\ x(4) \\ x(6) \\ x(1) \\ x(3) \\ x(5) \\ x(7) \end{bmatrix} \tag{3.4.7}$$

$$P_4 \begin{bmatrix} x(0) \\ x(2) \\ x(4) \\ x(6) \end{bmatrix} = \begin{bmatrix} x(0) \\ x(4) \\ x(2) \\ x(6) \end{bmatrix} \tag{3.4.8}$$

$$P_4 \begin{bmatrix} x(1) \\ x(3) \\ x(5) \\ x(7) \end{bmatrix} = \begin{bmatrix} x(1) \\ x(5) \\ x(3) \\ x(7) \end{bmatrix} \tag{3.4.9}$$

at last we get,

$$\begin{bmatrix} X_{e_1}(0) \\ X_{e_1}(1) \end{bmatrix} = F_2 \begin{bmatrix} x(0) \\ x(4) \end{bmatrix} = \begin{bmatrix} x(0) + x(4) \\ x(0) - x(4) \end{bmatrix} \tag{3.4.10}$$

$$\begin{bmatrix} X_{o_1}(0) \\ X_{o_1}(1) \end{bmatrix} = F_2 \begin{bmatrix} x(2) \\ x(6) \end{bmatrix} = \begin{bmatrix} x(2) + x(6) \\ x(2) - x(6) \end{bmatrix} \tag{3.4.11}$$

$$\begin{bmatrix} X_{e_2}(0) \\ X_{e_2}(1) \end{bmatrix} = F_2 \begin{bmatrix} x(1) \\ x(5) \end{bmatrix} = \begin{bmatrix} x(1) + x(5) \\ x(1) - x(5) \end{bmatrix} \tag{3.4.12}$$

$$\begin{bmatrix} X_{o_2}(0) \\ X_{o_2}(1) \end{bmatrix} = F_2 \begin{bmatrix} x(3) \\ x(7) \end{bmatrix} = \begin{bmatrix} x(3) + x(7) \\ x(3) - x(7) \end{bmatrix} \tag{3.4.13}$$

So, hence we can find the N point DFT using the above equations.

3.5. Example problem: Let x(n) be a input signal,

$$x(n) = \left\{ \underset{\uparrow}{1}, 2, 3, 4, 2, 1 \right\} \tag{3.5.1}$$

Now to compute 8-point DFT, N should be 8.
$\because$ For input signal N=6.
So for N=8 we zero-pad the input signal with 2 zeros.

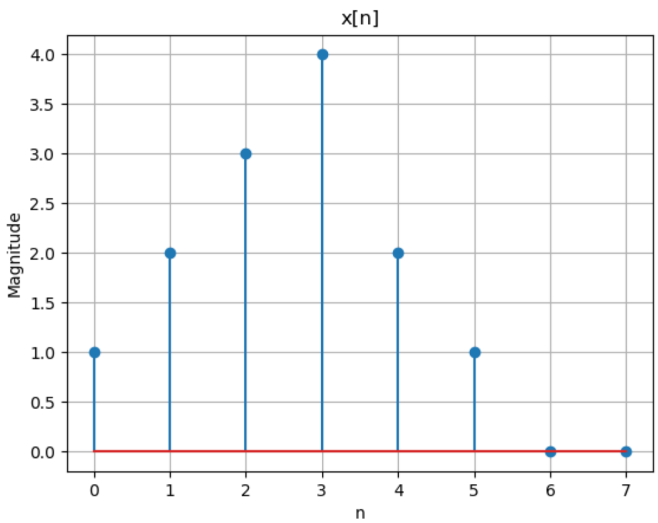
$$\implies x(n) = \left\{ \underset{\uparrow}{1}, 2, 3, 4, 2, 1, 0, 0 \right\} \tag{3.5.2}$$

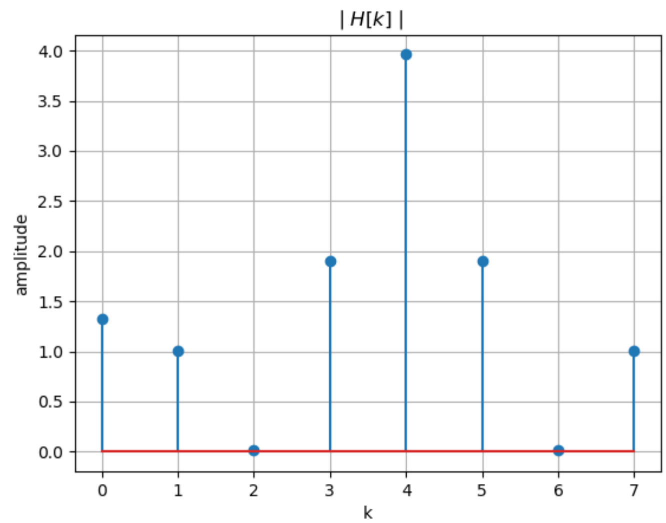
The 8-point FFT algorithm to compute the fourier transform of x(n) and h(n)is in the following python code.

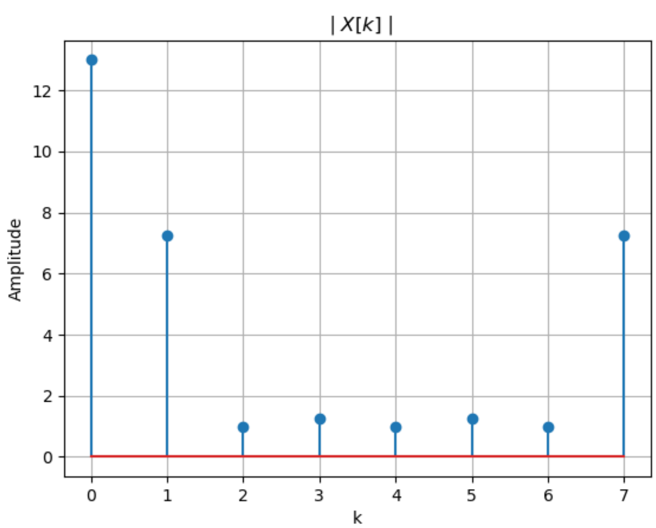
```
https://github.com/dsujjwal/IDP_DSP/blob/
    main/Assignment1/Codes/
    ee18btech11010_fft.py
```

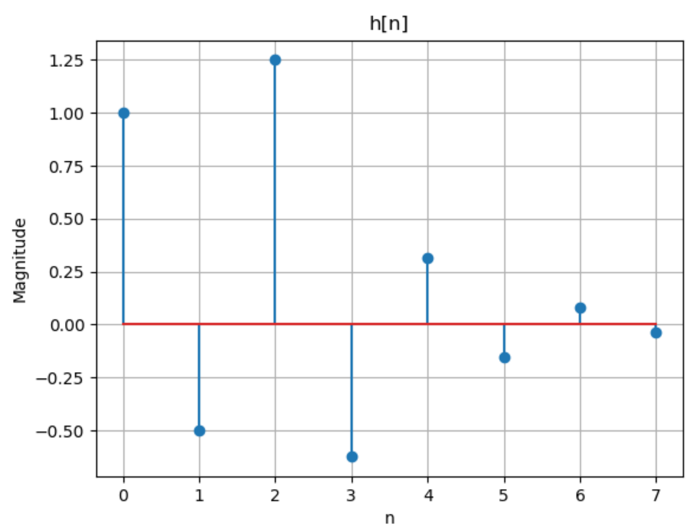
The following C program computes the 8-point FFT.

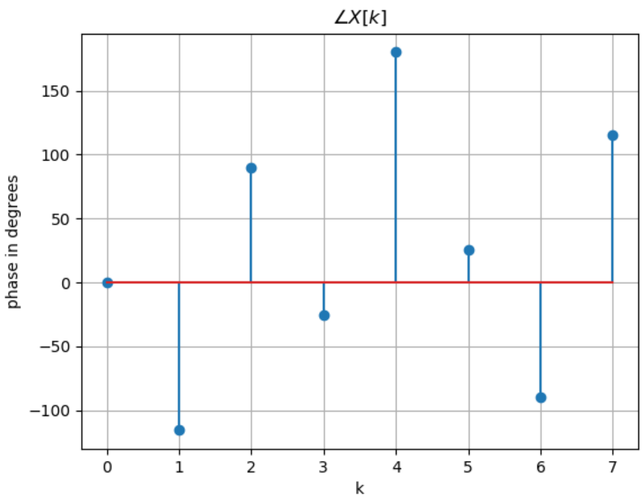
```
https://github.com/dsujjwal/IDP_DSP/blob/
    main/Assignment1/Codes/
    ee18btech11010_fft.c
```

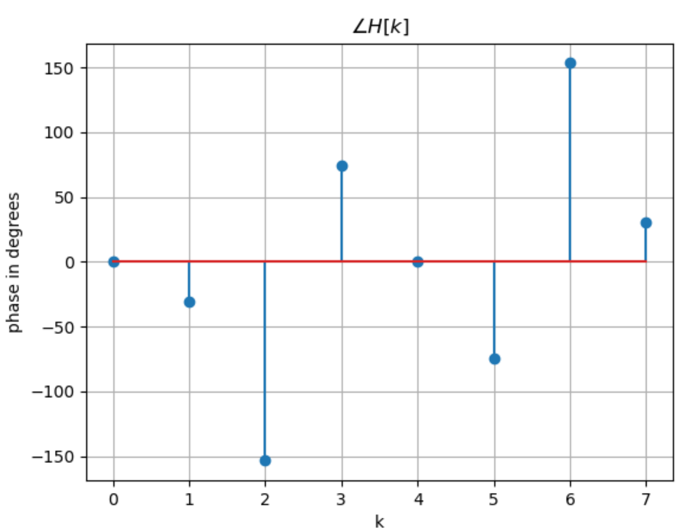

x[n]
Magnitude
n


$|H[k]|$
amplitude
k


$|X[k]|$
Amplitude
k


h[n]
Magnitude
n


$\angle X[k]$
phase in degrees
150
100
50
0
−50
−100
0
1
2
3
4
5
6
7
k


$\angle H[k]$
phase in degrees
150
100
50
0
−50
−100
−150
0
1
2
3
4
5
6
7
k