EE3025 Assignment-1

Adithya Vardhan - EE18BTECH11008

Download all python codes from

https://github.com/Adithya–Vardhan/HW1/tree/main/codes

and latex-tikz codes from

https://github.com/Adithya–Vardhan/HW1

1 Problem

Compute

$$X(k) \triangleq \sum_{n=0}^{N-1} x(n)e^{-j2\pi kn/N}, \quad k = 0, 1, \ldots, N-1 \tag{1.0.1}$$

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

2 Solution

Let

$$x(n) = \left\{ \underset{\uparrow}{1}, 2, 3, 4, 2, 1 \right\} \tag{2.0.1}$$

and the given difference equation is

$$y(n) + \frac{1}{2}y(n-1) = x(n) + x(n-2) \tag{2.0.2}$$

By applying Z-transform to the above equation we get,

$$Y(z) + \frac{1}{2}z^{-1}Y(z) = X(z) + z^{-2}X(z) \tag{2.0.3}$$

$$Y(z) = \frac{2(z^2+1)}{z(2z+1)}X(z) \tag{2.0.4}$$

Therefore H(z) is

$$H(z) = \frac{2(z^2+1)}{z(2z+1)} \tag{2.0.5}$$

$$H(z) = \frac{1+z^{-2}}{1+\frac{1}{2}z^{-1}} \tag{2.0.6}$$

$$H(z) = z^{-1}\left[\frac{1}{1+\frac{1}{2}z^{-1}} + \frac{z^{-2}}{1+\frac{1}{2}z^{-1}}\right] \tag{2.0.7}$$

By applying inverse z-transform we get,

$$h(n) = \left[\frac{-1}{2}\right]^n u(n) + \left[\frac{-1}{2}\right]^{n-2} u(n-2) \tag{2.0.8}$$

By using equation 1.0.1

$$X(k) = \sum_{n=0}^{N-1} x(n)e^{-j2\pi kn/N}, \quad k = 0, 1, \ldots, N-1 \tag{2.0.9}$$

the above equation can be written as,where $W = e^{\frac{-j2\pi}{N}}$

$$\begin{pmatrix} X(0) \\ X(1) \\ X(2) \\ \vdots \\ X(N-1) \end{pmatrix} = \begin{pmatrix} 1 & 1 & \cdots & 1 \\ 1 & W & \cdots & W^{N-1} \\ 1 & W^2 & \cdots & W^{2(N-1)} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & W^{N-1} & \cdots & W^{(N-1)(N-1)} \end{pmatrix} \begin{pmatrix} x(0) \\ x(1) \\ x(2) \\ \vdots \\ x(N-1) \end{pmatrix} \tag{2.0.10}$$

and

$$H(k) = \sum_{n=0}^{N-1} h(n)e^{-j2\pi kn/N}, \quad k = 0, 1, \ldots, N-1 \tag{2.0.11}$$

the above equation can be written as,

$$\begin{pmatrix} H(0) \\ H(1) \\ H(2) \\ \vdots \\ H(N-1) \end{pmatrix} = \begin{pmatrix} 1 & 1 & \cdots & 1 \\ 1 & W & \cdots & W^{N-1} \\ 1 & W^2 & \cdots & W^{2(N-1)} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & W^{N-1} & \cdots & W^{(N-1)(N-1)} \end{pmatrix} \begin{pmatrix} h(0) \\ h(1) \\ h(2) \\ \vdots \\ h(N-1) \end{pmatrix} \tag{2.0.12}$$

from above mentioned python codes we get the following plots

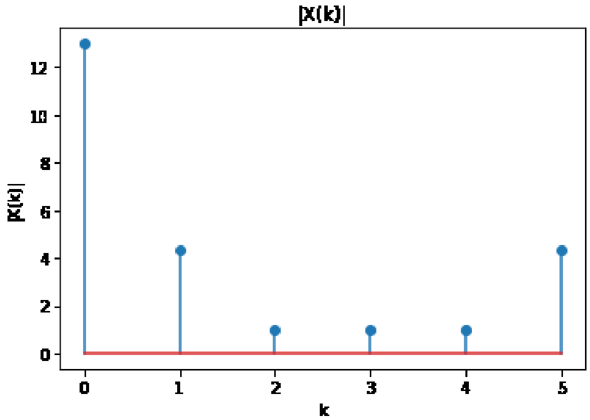

Fig. 0: Magnitude of X(k)

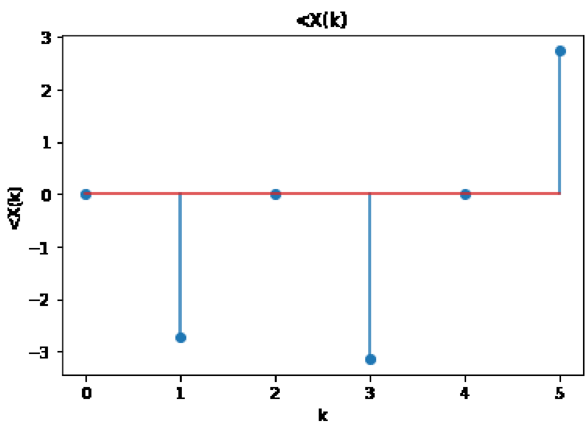

Fig. 0: Phase of X(k)

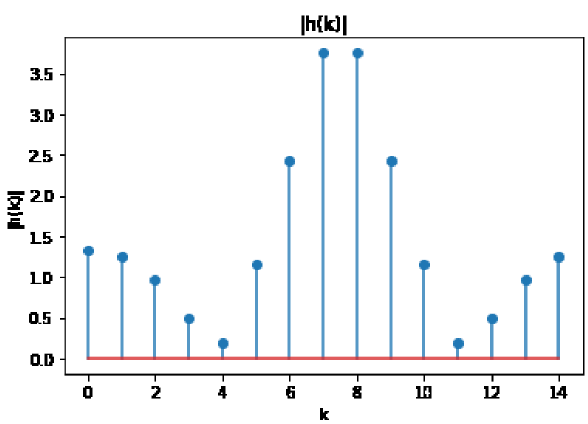

Fig. 0: Magnitude of h(k)

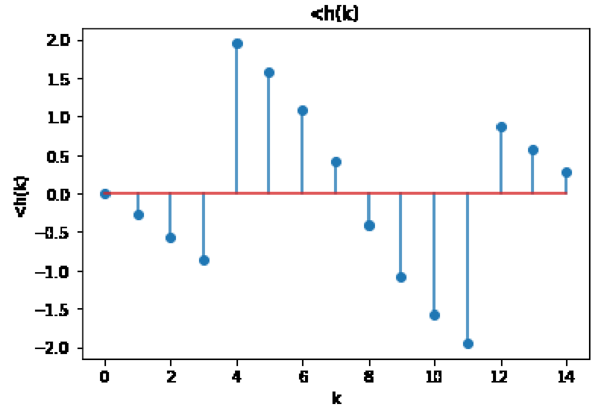

Fig. 0: Phase of h(k)

consider

$$x(n) = \left\{\underset{\uparrow}{1}, 2, 3, 4, 4, 3, 2, 1\right\} \tag{2.0.13}$$

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

$$F_N = \begin{bmatrix} I_{N/2} & D_{N/2} \\ I_{N/2} & -D_{N/2} \end{bmatrix} \begin{bmatrix} F_{N/2} & 0 \\ 0 & F_{N/2} \end{bmatrix} P_N \tag{2.0.14}$$

For N = 8

$$\implies F_8 = \begin{bmatrix} I_4 & D_4 \\ I_4 & -D_4 \end{bmatrix} \begin{bmatrix} F_4 & 0 \\ 0 & F_4 \end{bmatrix} P_8 \tag{2.0.15}$$

where I_4 is the 4x4 identity matrix

$$D_4 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & W_4^1 & 0 & 0 \\ 0 & 0 & W_4^2 & 0 \\ 0 & 0 & 0 & W_4^3 \end{bmatrix} \tag{2.0.16}$$

$$P_8 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.0.17}$$

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.

$$\begin{bmatrix} X(0) \\ X(1) \\ X(2) \\ X(3) \end{bmatrix} = \begin{bmatrix} X_e(0) \\ X_e(1) \\ X_e(2) \\ X_e(3) \end{bmatrix} + \begin{bmatrix} W_8^0 & 0 & 0 & 0 \\ 0 & W_8^1 & 0 & 0 \\ 0 & 0 & W_8^2 & 0 \\ 0 & 0 & 0 & W_8^3 \end{bmatrix} \begin{bmatrix} X_o(0) \\ X_o(1) \\ X_o(2) \\ X_o(3) \end{bmatrix} \tag{2.0.18}$$

$$\begin{bmatrix} X(4) \\ X(5) \\ X(6) \\ X(7) \end{bmatrix} = \begin{bmatrix} X_e(0) \\ X_e(1) \\ X_e(2) \\ X_e(3) \end{bmatrix} - \begin{bmatrix} W_8^0 & 0 & 0 & 0 \\ 0 & W_8^1 & 0 & 0 \\ 0 & 0 & W_8^2 & 0 \\ 0 & 0 & 0 & W_8^3 \end{bmatrix} \begin{bmatrix} X_o(0) \\ X_o(1) \\ X_o(2) \\ X_o(3) \end{bmatrix} \tag{2.0.19}$$

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

$$\begin{bmatrix} X_e(0) \\ X_e(1) \end{bmatrix} = \begin{bmatrix} X_{e_1}(0) \\ X_{e_1}(1) \end{bmatrix} + \begin{bmatrix} W_4^0 & 0 \\ 0 & W_4^1 \end{bmatrix} \begin{bmatrix} X_{o_1}(0) \\ X_{o_1}(1) \end{bmatrix} \tag{2.0.20}$$

$$\begin{bmatrix} X_e(2) \\ X_e(3) \end{bmatrix} = \begin{bmatrix} X_{e_1}(0) \\ X_{e_1}(1) \end{bmatrix} - \begin{bmatrix} W_4^0 & 0 \\ 0 & W_4^1 \end{bmatrix} \begin{bmatrix} X_{o_1}(0) \\ X_{o_1}(1) \end{bmatrix} \tag{2.0.21}$$

$$\begin{bmatrix} X_o(0) \\ X_o(1) \end{bmatrix} = \begin{bmatrix} X_{e_2}(0) \\ X_{e_2}(1) \end{bmatrix} + \begin{bmatrix} W_4^0 & 0 \\ 0 & W_4^1 \end{bmatrix} \begin{bmatrix} X_{o_2}(0) \\ X_{o_2}(1) \end{bmatrix} \tag{2.0.22}$$

$$\begin{bmatrix} X_o(2) \\ X_o(3) \end{bmatrix} = \begin{bmatrix} X_{e_2}(0) \\ X_{e_2}(1) \end{bmatrix} - \begin{bmatrix} W_4^0 & 0 \\ 0 & W_4^1 \end{bmatrix} \begin{bmatrix} X_{o_2}(0) \\ X_{o_2}(1) \end{bmatrix} \tag{2.0.23}$$

$$P_8 \begin{bmatrix} x(0) \\ x(1) \\ x(2) \\ x(3) \\ x(4) \\ x(5) \\ x(6) \\ x(7) \end{bmatrix} = \begin{bmatrix} x(0) \\ x(2) \\ x(4) \\ x(6) \\ x(1) \\ x(3) \\ x(5) \\ x(7) \end{bmatrix} \tag{2.0.24}$$

$$P_4 \begin{bmatrix} x(0) \\ x(2) \\ x(4) \\ x(6) \end{bmatrix} = \begin{bmatrix} x(0) \\ x(4) \\ x(2) \\ x(6) \end{bmatrix} \tag{2.0.25}$$

$$P_4 \begin{bmatrix} x(1) \\ x(3) \\ x(5) \\ x(7) \end{bmatrix} = \begin{bmatrix} x(1) \\ x(5) \\ x(3) \\ x(7) \end{bmatrix} \tag{2.0.26}$$

Finally,

$$\begin{bmatrix} X_{e_1}(0) \\ X_{e_1}(1) \end{bmatrix} = F_2 \begin{bmatrix} x(0) \\ x(4) \end{bmatrix} = \begin{bmatrix} x(0) + x(4) \\ x(0) - x(4) \end{bmatrix} \tag{2.0.27}$$

$$\begin{bmatrix} X_{o_1}(0) \\ X_{o_1}(1) \end{bmatrix} = F_2 \begin{bmatrix} x(2) \\ x(6) \end{bmatrix} = \begin{bmatrix} x(2) + x(6) \\ x(2) - x(6) \end{bmatrix} \tag{2.0.28}$$

$$\begin{bmatrix} X_{e_2}(0) \\ X_{e_2}(1) \end{bmatrix} = F_2 \begin{bmatrix} x(1) \\ x(5) \end{bmatrix} = \begin{bmatrix} x(1) + x(5) \\ x(1) - x(5) \end{bmatrix} \tag{2.0.29}$$

$$\begin{bmatrix} X_{o_2}(0) \\ X_{o_2}(1) \end{bmatrix} = F_2 \begin{bmatrix} x(3) \\ x(7) \end{bmatrix} = \begin{bmatrix} x(3) + x(7) \\ x(3) - x(7) \end{bmatrix} \tag{2.0.30}$$

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.
from above mentioned python codes we get the following plots

```
https://github.com/Adithya−Vardhan/HW1/tree/
    main/codes
```

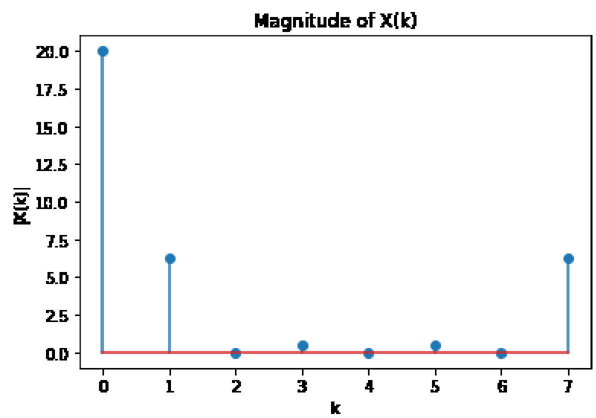

Fig. 0: Magnitude of x(k)

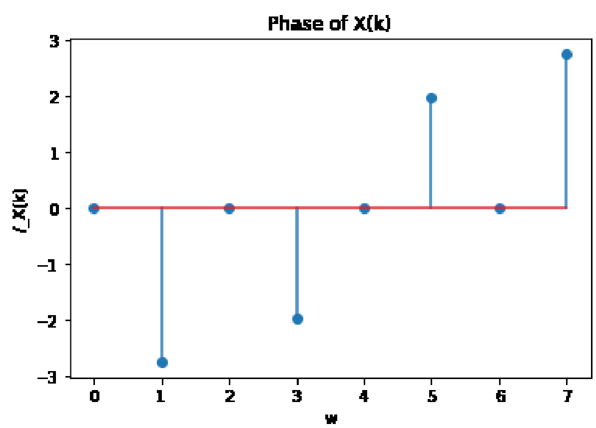

Fig. 0: Phase of x(k)