

EE3025 Assignment-1

Kartikeya Jaiswal - EE18BTECH11025

Download all python codes from

<https://github.com/kartikeyajaiswal/EE3025/tree/main/codes>

and latex-tikz codes from

<https://github.com/kartikeyajaiswal/EE3025>

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 $X(k)$, $H(k)$ and $y(n)$ using FFT and IFFT methods.

2 SOLUTION

2.1. To compute $h(n)$, find the $Y(z)$ by applying Z-transform on equation i.e.,

$$Y(z) + \frac{1}{2}z^{-1}Y(z) = X(z) + z^{-2}X(z) \quad (2.1.1)$$

$$\Rightarrow Y(z) = \frac{2(z^2 + 1)}{z(2z + 1)}X(z) \quad (2.1.2)$$

Now $H(z)$

$$H(z) = \frac{Y(z)}{X(z)} \quad (2.1.3)$$

$$H(z) = \frac{2(z^2 + 1)}{z(2z + 1)} \quad (2.1.4)$$

$$H(z) = \frac{1 + z^{-2}}{1 + \frac{1}{2}z^{-1}} \quad (2.1.5)$$

applying inverse Z-transform to compute $h(n)$

$$h(n) = Z^{-1} \left[\frac{1}{1 + \frac{1}{2}z^{-1}} + \frac{z^{-2}}{1 + \frac{1}{2}z^{-1}} \right] \quad (2.1.6)$$

$$h(n) = \left[\frac{-1}{2} \right]^n u(n) + \left[\frac{-1}{2} \right]^{n-2} u(n-2) \quad (2.1.7)$$

2.2. X can be expressed as Matrix Multiplication of DFT Matrix and x .

$$X(k) = \left[e^{-j2\pi kn/N} \right]_{1 \times N} x, \quad n = 0, 1, \dots, N-1 \quad (2.2.1)$$

i.e.

$$X = \left[e^{-j2\pi nk/N} \right]_{N \times N} x, \quad n, k = 0, 1, \dots, N-1 \quad (2.2.2)$$

H can be calculated in a similar manner; also

$$Y(k) = X(k)H(k) \quad (2.2.3)$$

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

2.4. Consider:

$$X(k) = \sum_{n=0}^{N-1} x(n)e^{-j2\pi kn/N}, \quad k = 0, 1, \dots, N-1 \quad (2.4.1)$$

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

Now, using the following properties of W_N ,

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

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

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

to compute FFT from DFT:

$$\begin{aligned}
X(k) &= \sum_{n=\text{even}} x(n)W_N^{kn} + \sum_{n=\text{odd}} x(n)W_N^{kn} \quad (2.4.3) \\
&= \sum_{m=0}^2 x(2m)W_N^{2mk} + \sum_{m=0}^2 x(2m+1)W_N^{(2m+1)k} \quad (2.4.4)
\end{aligned}$$

Let first term of the above be $X_e(k)$ and the second be $X_o(k)$, which are basically DFTs of $x(2m)$ and $x(2m+1)$ for $m=0,1,2$.

2.5. X_e and X_o can be written as

$$X_e(k) = \sum_{m=0}^2 x(2m)W_3^{mk} \quad (2.5.1)$$

$$X_o(k) = \sum_{m=0}^2 x(2m+1)W_3^{mk} \quad (2.5.2)$$

Here, $N=3 \therefore m$ takes three values

written in matrix form,

$$\begin{bmatrix} X_e(0) \\ X_e(1) \\ X_e(2) \end{bmatrix} = \begin{bmatrix} W_3^0 & W_3^0 & W_3^0 \\ W_3^1 & W_3^1 & W_3^1 \\ W_3^2 & W_3^2 & W_3^2 \end{bmatrix} \begin{bmatrix} x(0) \\ x(2) \\ x(4) \end{bmatrix} \quad (2.5.3)$$

and

$$\begin{bmatrix} X_o(0) \\ X_o(1) \\ X_o(2) \end{bmatrix} = \begin{bmatrix} W_3^0 & W_3^0 & W_3^0 \\ W_3^1 & W_3^1 & W_3^1 \\ W_3^2 & W_3^2 & W_3^2 \end{bmatrix} \begin{bmatrix} x(1) \\ x(3) \\ x(5) \end{bmatrix} \quad (2.5.4)$$

combining the two:

$$\begin{bmatrix} X_e(0) \\ X_e(1) \\ X_e(2) \\ X_o(0) \\ X_o(1) \\ X_o(2) \end{bmatrix} = \begin{bmatrix} W_3^0 & W_3^0 & W_3^0 & 0 & 0 & 0 \\ W_3^1 & W_3^1 & W_3^1 & 0 & 0 & 0 \\ W_3^2 & W_3^2 & W_3^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & W_3^0 & W_3^0 & W_3^0 \\ 0 & 0 & 0 & W_3^1 & W_3^1 & W_3^1 \\ 0 & 0 & 0 & W_3^2 & W_3^2 & W_3^2 \end{bmatrix} \begin{bmatrix} x(0) \\ x(2) \\ x(4) \\ x(1) \\ x(3) \\ x(5) \end{bmatrix} \quad (2.5.5)$$

Let the above 6x6 matrix be Z_1 and 6x1 be X_f . Now, a matrix P can be found such that,

$$P \begin{bmatrix} x(0) \\ x(1) \\ x(2) \\ x(3) \\ x(4) \\ x(5) \end{bmatrix} = \begin{bmatrix} x(0) \\ x(2) \\ x(4) \\ x(1) \\ x(3) \\ x(5) \end{bmatrix} \quad (2.5.6)$$

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

Then, the equation can be written as,

$$X_f = Z_1 P x \quad (2.5.7)$$

2.6. To compute X

$$X(k) = X_e(k) + W_N^k X_o(k) \quad (2.6.1)$$

$$\begin{bmatrix} X(0) \\ X(1) \\ X(2) \\ X(3) \\ X(4) \\ X(5) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & W_6^0 & 0 & 0 \\ 0 & 1 & 0 & 0 & W_6^1 & 0 \\ 0 & 0 & 1 & 0 & 0 & W_6^2 \\ 1 & 0 & 0 & W_6^3 & 0 & 0 \\ 0 & 1 & 0 & 0 & W_6^4 & 0 \\ 0 & 0 & 1 & 0 & 0 & W_6^5 \end{bmatrix} \begin{bmatrix} X_e(0) \\ X_e(1) \\ X_e(2) \\ X_o(0) \\ X_o(1) \\ X_o(2) \end{bmatrix} \quad (2.6.2)$$

$$X = \begin{bmatrix} X(0) \\ X(1) \\ X(2) \\ X(3) \\ X(4) \\ X(5) \end{bmatrix} = Z_2 \begin{bmatrix} X_1(0) \\ X_1(1) \\ X_1(2) \\ X_2(0) \\ X_2(1) \\ X_2(2) \end{bmatrix} \quad (2.6.3)$$

using equation (2.5.7), we get

$$X = Z_2 Z_1 P x \quad (2.6.4)$$

H can be calculated using the same formula as above,

$$\mathcal{H} = Z_2 Z_1 P h \quad (2.6.5)$$

$$H = \begin{bmatrix} H(0) \\ H(1) \\ H(2) \\ H(3) \\ H(4) \\ H(5) \end{bmatrix} = \begin{bmatrix} 1.28125 \\ 0.515625 - j0.5142 \\ -0.078125 + j1.1095 \\ 3.84375 + j4.97 \times 10^{-16} \\ -0.078125 - j1.10959 \\ 0.515625 + j0.5142 \end{bmatrix} \quad (2.6.6)$$

2.7. To compute Y , we can do elementwise multiplication $Y = H.X$

$$Y = \begin{bmatrix} Y(0) \\ Y(1) \\ Y(2) \\ Y(3) \\ Y(4) \\ Y(5) \end{bmatrix} = \begin{bmatrix} 16.65625 \\ -2.95312 + j1.1637 \\ -0.07812 + j1.10959 \\ -3.8437 - j1.0953 \times 10^{-14} \\ -0.078125 - j1.10959 \\ -2.953125 - j1.16372 \end{bmatrix} \quad (2.7.1)$$

2.8. Now IFFT can be computed as;

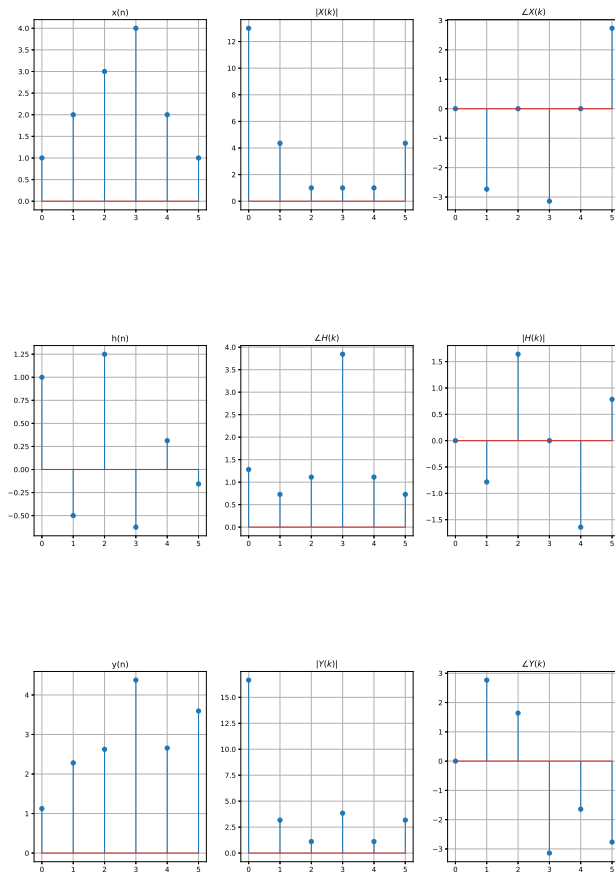
$$y = \frac{1}{N}(Z_1 Z_2 P)^H Y \quad (2.8.1)$$

where H denotes hermitian of a matrix. The above can be used to calculate y .

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

<https://github.com/kartikeyajaiswal/EE3025/tree/main/codes>

2.10. The following plots are obtained



2.11. Benefits of FFT:

- The DFT would have n^2 operations for the computation, whereas in the modified algorithm (FFT) the number of operations are $2(\frac{n}{2})^2$ since two smaller matrices are combined instead of one large matrix.
- Also there is a computational benefit since the matrices are sparse (most elements are zeros or ones).
- If the above is recursively performed, the complexity of the algorithm will be $\frac{n}{2} \log n$.