Digital Signal Processing

G V V Sharma*

Contents

***Abstract*—This manual provides a simple introduction to digital signal processing.**

1 Software Installation

Run the following commands

```
sudo apt−get update
sudo apt−get install libffi−dev libsndfile1 python3
    −scipy python3−numpy python3−matplotlib
sudo pip install cffi pysoundfile
```

2 Digital Filter

2.1 Download the sound file from

```
wget https://raw.githubusercontent.com/
    gadepall/
EE1310/master/filter/codes/Sound_Noise.wav
```

2.2 You will find a spectrogram at https://academo.org/demos/spectrum-analyzer.. Upload the sound file that you downloaded in Problem 2.1 in the spectrogram and play. Observe the spectrogram. What do you find?
Solution: There are a lot of yellow lines between 440 Hz to 5.1 KHz. These represent the synthesizer key tones. Also, the key strokes are audible along with background noise.

2.3 Write the python code for removal of out of band noise and execute the code.
Solution:

```
import soundfile as sf
from scipy import signal

#read .wav file
input_signal,fs = sf.read('Sound_Noise.wav'
    )

#sampling frequency of Input signal
sampl_freq=fs

#order of the filter
order=4

#cutoff frquency 4kHz
cutoff_freq=4000.0

#digital frequency
Wn=2*cutoff_freq/sampl_freq

# b and a are numerator and denominator
    polynomials respectively
b, a = signal.butter(order,Wn, 'low')

#filter the input signal with butterworth filter
output_signal = signal.filtfilt(b, a,
    input_signal)
#output_signal = signal.lfilter(b, a,
    input_signal)

#write the output signal into .wav file
sf.write('Sound_With_ReducedNoise.wav',
    output_signal, fs)
```

2.4 The output of the python script

*The author is with the Department of Electrical Engineering, Indian Institute of Technology, Hyderabad 502285 India e-mail: gadepall@iith.ac.in.

in Problem 2.3 is the audio file Sound_With_ReducedNoise.wav. Play the file in the spectrogram in Problem 2.2. What do you observe?
Solution: The key strokes as well as background noise is subdued in the audio. Also, the signal is blank for frequencies above 5.1 kHz.

3 Difference Equation

3.1 Let

$$x(n) = \left\{\underset{\uparrow}{1}, 2, 3, 4, 2, 1\right\} \tag{3.1}$$

Sketch $x(n)$.

3.2 Let

$$y(n) + \frac{1}{2}y(n-1) = x(n) + x(n-2), \quad y(n) = 0, n < 0 \tag{3.2}$$

Sketch $y(n)$.
Solution: The following code yields Fig. 3.2.

```
wget https://github.com/gadepall/EE1310/raw/
    master/filter/codes/xnyn.py
```

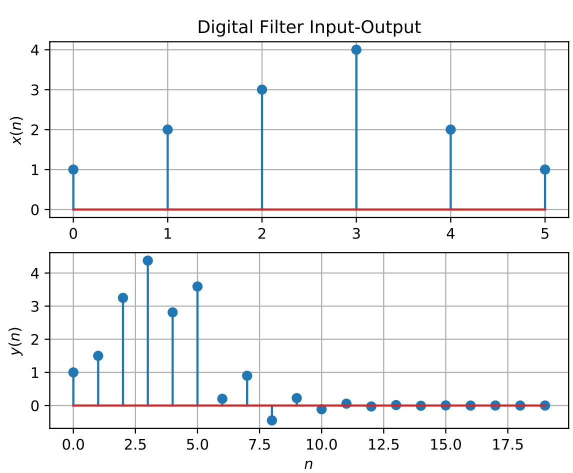

Fig. 3.2

3.3 Repeat the above exercise using a C code.

4 Z-transform

4.1 The Z-transform of $x(n)$ is defined as

$$X(z) = \mathcal{Z}\{x(n)\} = \sum_{n=-\infty}^{\infty} x(n)z^{-n} \tag{4.1}$$

Show that

$$\mathcal{Z}\{x(n-1)\} = z^{-1}X(z) \tag{4.2}$$

and find

$$\mathcal{Z}\{x(n-k)\} \tag{4.3}$$

Solution: From (4.1),

$$\mathcal{Z}\{x(n-k)\} = \sum_{n=-\infty}^{\infty} x(n-1)z^{-n} \tag{4.4}$$

$$= \sum_{n=-\infty}^{\infty} x(n)z^{-n-1} = z^{-1}\sum_{n=-\infty}^{\infty} x(n)z^{-n} \tag{4.5}$$

resulting in (4.2). Similarly, it can be shown that

$$\mathcal{Z}\{x(n-k)\} = z^{-k}X(z) \tag{4.6}$$

4.2 Obtain $X(z)$ for $x(n)$ defined in problem 3.1.
Solution: From (4.1)

$$X(z) = \sum_{n=0}^{5} x(n)z^{-n} \tag{4.7}$$

$$= x(0) + x(1)z^{-1} + \cdots + x(5)z^{-5} \tag{4.8}$$

$$= 1 + 2z^{-1} + 3z^{-2} + 4z^{-3} + 2z^{-4} + z^{-5} \tag{4.9}$$

4.3 Find

$$H(z) = \frac{Y(z)}{X(z)} \tag{4.10}$$

from (3.2) assuming that the Z-transform is a linear operation.
Solution: Applying (4.6) in (3.2),

$$Y(z) + \frac{1}{2}z^{-1}Y(z) = X(z) + z^{-2}X(z) \tag{4.11}$$

$$\implies \frac{Y(z)}{X(z)} = \frac{1 + z^{-2}}{1 + \frac{1}{2}z^{-1}} \tag{4.12}$$

4.4 Find the Z transform of

$$\delta(n) = \begin{cases} 1 & n = 0 \\ 0 & \text{otherwise} \end{cases} \tag{4.13}$$

and show that the Z-transform of

$$u(n) = \begin{cases} 1 & n \geq 0 \\ 0 & \text{otherwise} \end{cases} \tag{4.14}$$

is

$$U(z) = \frac{1}{1 - z^{-1}}, \quad |z| > 1 \tag{4.15}$$

Solution: It is easy to show that

$$\delta(n) \overset{\mathcal{Z}}{\rightleftharpoons} 1 \tag{4.16}$$

and from (4.14),

$$U(z) = \sum_{n=0}^{\infty} z^{-n} \tag{4.17}$$

$$= \frac{1}{1 - z^{-1}}, \quad |z| > 1 \tag{4.18}$$

using the formula for the sum of an infinite geometric progression.

4.5 Show that

$$a^n u(n) \overset{\mathcal{Z}}{\rightleftharpoons} \frac{1}{1 - az^{-1}} \quad |z| > |a| \tag{4.19}$$

Solution: From (4.19) and (4.14)

$$U_a(z) = \sum_{n=-\infty}^{\infty} u(n)\, a^n z^{-n} \tag{4.20}$$

$$= \sum_{n=0}^{\infty} a^n z^{-n} \tag{4.21}$$

$$= \frac{1}{1 - az^{-1}}, \quad \left|\frac{z}{a}\right| > 1 \tag{4.22}$$

using the formula for the sum of an infinite geometric progression.

4.6 Let

$$H\left(e^{\jmath\omega}\right) = H\left(z = e^{\jmath\omega}\right). \tag{4.23}$$

Plot $|H\left(e^{\jmath\omega}\right)|$. Is it periodic? If so, find the period. $H(e^{\jmath\omega})$ is known as the *Discret Time Fourier Transform* (DTFT) of $h(n)$.

Solution: The following code plots Fig. 4.6.

$$|H\left(e^{\jmath\omega}\right)| = \left|\frac{1 + e^{-2\jmath\omega}}{1 + \frac{1}{2}e^{-\jmath\omega}}\right| \tag{4.24}$$

$$= \sqrt{\frac{(1 + \cos 2\omega)^2 + (\sin 2\omega)^2}{\left(1 + \frac{1}{2}\cos\omega\right)^2 + \left(\frac{1}{2}\sin\omega\right)^2}} \tag{4.25}$$

$$= \sqrt{\frac{2\left(1 + \cos 2\omega\right)}{\frac{5}{4} + \cos\omega}} \tag{4.26}$$

$$= \sqrt{\frac{2\left(2\cos^2\omega\right)}{\frac{5}{4} + \cos\omega}} \tag{4.27}$$

$$= \frac{4|\cos\omega|}{\sqrt{5 + 4\cos\omega}} \tag{4.28}$$

Thus,

$$H\left(e^{\jmath(\omega+2\pi)}\right) = \frac{4|\cos(\omega + 2\pi)|}{\sqrt{5 + 4\cos(\omega + 2\pi)}} \tag{4.29}$$

$$= \frac{4|\cos\omega|}{\sqrt{5 + 4\cos\omega}} \tag{4.30}$$

$$= H\left(e^{\jmath\omega}\right) \tag{4.31}$$

So, the fundamental period is 2π.

```
wget https://raw.githubusercontent.com/
    gadepall/EE1310/master/filter/codes/dtft.
    py
```

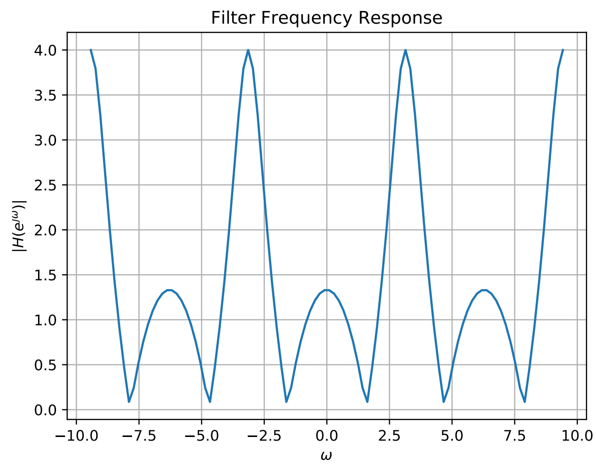

Fig. 4.6: $|H\left(e^{\jmath\omega}\right)|$

4.7 Express $h(n)$ in terms of $H\left(e^{\jmath\omega}\right)$.

Solution: We have,

$$H(e^{\jmath\omega}) = \sum_{k=-\infty}^{\infty} h(k) e^{-\jmath\omega k} \tag{4.32}$$

$$\int_{-\pi}^{\pi} e^{\jmath\omega(n-k)} d\omega = \begin{cases} 2\pi & n = k \\ 0 & \text{otherwise} \end{cases} \tag{4.33}$$

From the above equations, we have

$$\frac{1}{2\pi}\int_{-\pi}^{\pi} H(e^{\jmath\omega}) e^{j\omega n} d\omega \tag{4.34}$$

$$= \frac{1}{2\pi}\sum_{k=-\infty}^{\infty}\int_{-\pi}^{\pi} h(k) e^{\jmath\omega(n-k)} d\omega \tag{4.35}$$

$$= \frac{1}{2\pi} 2\pi h(n) = h(n) \tag{4.36}$$

which is known as the Inverse Discrete Fourier

Transform.

$$h(n) = \frac{1}{2\pi}\int_{-\pi}^{\pi} H(e^{\jmath\omega})e^{\jmath\omega n}d\omega \tag{4.37}$$

5 Impulse Response

5.1 Using long division, find

$$h(n), \quad n < 5 \tag{5.1}$$

for H(z) in (4.12).
Solution: Using the substitution $x := z^{-1}$, we perform long division.

$$\begin{array}{r} 2x-4 \\ \frac{1}{2}x+1 \big) \overline{\; x^2 \qquad\quad +1} \\ \underline{-\,x^2-2x} \\ -\,2x+1 \\ \underline{2x+4} \\ 5 \end{array}$$

Thus,

$$H(z) = -4 + 2z^{-1} + \frac{5}{1+\frac{1}{2}z^{-1}} \tag{5.2}$$

$$= -4 + 2z^{-1} + 5\sum_{n=0}^{\infty}\left(-\frac{1}{2}\right)^n z^{-n} \tag{5.3}$$

$$= 1 - \frac{1}{2}z^{-1} + 5\sum_{n=2}^{\infty}\left(-\frac{1}{2}\right)^n z^{-n} \tag{5.4}$$

$$= \sum_{n=0}^{\infty}\left(-\frac{1}{2}\right)^n z^{-n} + 4\sum_{n=2}^{\infty}\left(-\frac{1}{2}\right)^n z^{-n} \tag{5.5}$$

$$= \sum_{n=-\infty}^{\infty} u(n)\left(-\frac{1}{2}\right)^n z^{-n} + \sum_{n=-\infty}^{\infty} u(n-2)\left(-\frac{1}{2}\right)^{n-2} z^{-n} \tag{5.6}$$

Therefore, from (4.1),

$$h(n) = \left(-\frac{1}{2}\right)^n u(n) + \left(-\frac{1}{2}\right)^{n-2} u(n-2) \tag{5.7}$$

5.2 Find an expression for $h(n)$ using $H(z)$, given that

$$h(n) \overset{\mathcal{Z}}{\rightleftharpoons} H(z) \tag{5.8}$$

and there is a one to one relationship between $h(n)$ and $H(z)$. $h(n)$ is known as the *impulse response* of the system defined by (3.2).

Solution: From (4.12),

$$H(z) = \frac{1}{1+\frac{1}{2}z^{-1}} + \frac{z^{-2}}{1+\frac{1}{2}z^{-1}} \tag{5.9}$$

$$\implies h(n) = \left(-\frac{1}{2}\right)^n u(n) + \left(-\frac{1}{2}\right)^{n-2} u(n-2) \tag{5.10}$$

using (4.19) and (4.6).

5.3 Sketch $h(n)$. Is it bounded? Justify theoretically.
Solution: The following code plots Fig. 5.3.

```
wget https://raw.githubusercontent.com/
    gadepall/EE1310/master/filter/codes/hn.py
```

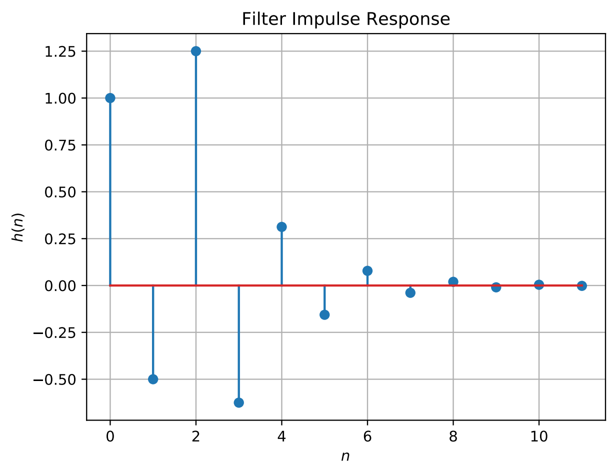

Fig. 5.3: $h(n)$ as the inverse of $H(z)$

5.4 Convergent? Justify using the ratio test. **Solution:** For large n, we see that

$$h(n) = \left(-\frac{1}{2}\right)^n + \left(-\frac{1}{2}\right)^{n-2} \tag{5.11}$$

$$= \left(-\frac{1}{2}\right)^n (4+1) = 5\left(-\frac{1}{2}\right)^n \tag{5.12}$$

$$\implies \left|\frac{h(n+1)}{h(n)}\right| = \frac{1}{2} \tag{5.13}$$

and therefore, $\lim_{n\to\infty}\left|\frac{h(n+1)}{h(n)}\right| = \frac{1}{2} < 1$. Hence, we see that $h(n)$ converges.

5.5 The system with $h(n)$ is defined to be stable if

$$\sum_{n=-\infty}^{\infty} h(n) < \infty \tag{5.14}$$

Is the system defined by (3.2) stable for the impulse response in (5.8)?

Solution: : From (5.10),

$$h(n) = \left(-\frac{1}{2}\right)^n u(n) + \left(-\frac{1}{2}\right)^{n-2} u(n-2) \quad (5.15)$$

then

$$\sum_{n=-\infty}^{\infty} h(n) = \sum_{n=0}^{\infty} \left(-\frac{1}{2}\right)^n + \sum_{n=2}^{\infty} \left(-\frac{1}{2}\right)^{n-2} \quad (5.16)$$

$$\sum_{n=-\infty}^{\infty} h(n) = \frac{4}{3} \quad (5.17)$$

since

$$\sum_{n=-\infty}^{\infty} h(n) < \infty \quad (5.18)$$

$h(n)$ is stable.

5.6 Verify the above result using a python code.

5.7 Compute and sketch $h(n)$ using

$$h(n) + \frac{1}{2}h(n-1) = \delta(n) + \delta(n-2), \quad (5.19)$$

This is the definition of $h(n)$.

Solution: The following code plots Fig. 5.7. Note that this is the same as Fig. 5.3.

```
wget https://raw.githubusercontent.com/
    gadepall/EE1310/master/filter/codes/hndef
    .py
```

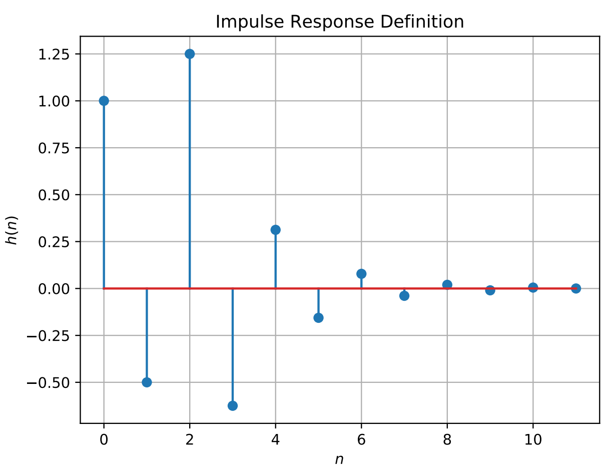

Fig. 5.7: $h(n)$ from the definition

5.8 Compute

$$y(n) = x(n) * h(n) = \sum_{n=-\infty}^{\infty} x(k)h(n-k) \quad (5.20)$$

Comment. The operation in (5.20) is known as *convolution*.

Solution: The following code plots Fig. 5.8. Note that this is the same as $y(n)$ in Fig. 3.2.

```
wget https://raw.githubusercontent.com/
    DeepakReddyVelagala/EE1310/main/filter
    /codes/ynconv.py
```

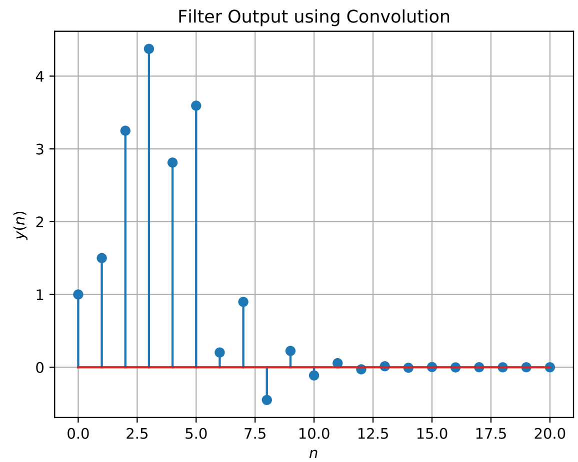

Fig. 5.8: $y(n)$ from the definition of convolution

5.9 Express the above convolution using a Teoplitz matrix.

Solution: We use Toeplitz matrices for convolution

$$\mathbf{y} = \mathbf{x} \circledast \mathbf{h} \quad (5.21)$$

$$\mathbf{y} = \begin{pmatrix} h_1 & 0 & . & . & . & 0 \\ h_2 & h_1 & . & . & . & 0 \\ h_3 & h_2 & h_1 & . & . & 0 \\ . & . & . & . & . & . \\ 0 & . & . & h_3 & h_2 & h_1 \\ 0 & . & . & . & h_2 & h_1 \\ 0 & . & . & . & 0 & h_1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} \quad (5.22)$$

5.10 Show that

$$y(n) = \sum_{n=-\infty}^{\infty} x(n-k)h(k) \quad (5.23)$$

Solution: From (5.20), we substitute $k := n-k$

to get

$$y(n) = \sum_{k=-\infty}^{\infty} x(k)\,h(n-k) \tag{5.24}$$
$$= \sum_{n-k=-\infty}^{\infty} x(n-k)\,h(k) \tag{5.25}$$
$$= \sum_{k=-\infty}^{\infty} x(n-k)\,h(k) \tag{5.26}$$

6 DFT and FFT

6.1 Compute

$$X(k) \triangleq \sum_{n=0}^{N-1} x(n)e^{-\jmath 2\pi kn/N}, \quad k = 0, 1, \dots, N-1 \tag{6.1}$$

and $H(k)$ using $h(n)$.
Solution: The following code plots $X(k)$ and $H(k)$.

```
https://github.com/DeepakReddyVelagala/
    EE3900/blob/main/EE3900−2022−
    main/filter/codes/XkHk_dft.py
```

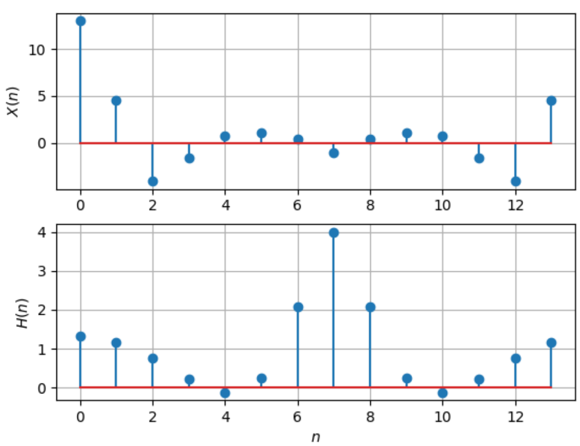

Fig. 6.1

6.2 Compute

$$Y(k) = X(k)H(k) \tag{6.2}$$

Solution: The following code plots $Y(k)$.

```
https://github.com/DeepakReddyVelagala/
    EE3900/blob/main/EE3900−2022−
    main/filter/codes/Yk.py
```

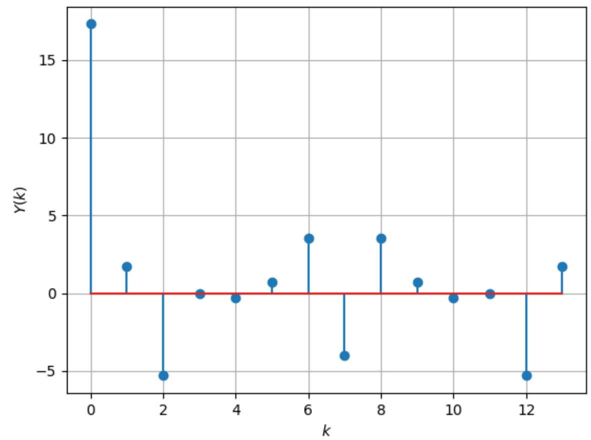

Fig. 6.2

6.3 Compute

$$y(n) = \frac{1}{N}\sum_{k=0}^{N-1} Y(k)\cdot e^{\jmath 2\pi kn/N}, \quad n = 0, 1, \dots, N-1 \tag{6.3}$$

Solution: The following code plots Fig. 5.8. Note that this is the same as $y(n)$ in Fig. 3.2.

```
wget https://raw.githubusercontent.com/
    gadepall/EE1310/master/filter/codes/yndft.
    py
```

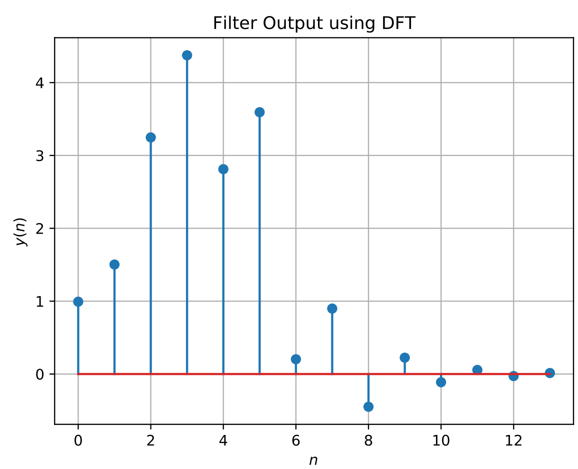

Fig. 6.3: $y(n)$ from the DFT

6.4 Repeat the previous exercise by computing $X(k), H(k)$ and $y(n)$ through FFT and IFFT.
Solution: The following code plots $X(k)$, $H(k)$ and $y(n)$ by fft.

https://github.com/DeepakReddyVelagala/EE3900/blob/main/EE3900−2022−main/**filter**/codes/Xk_Hk_fft.

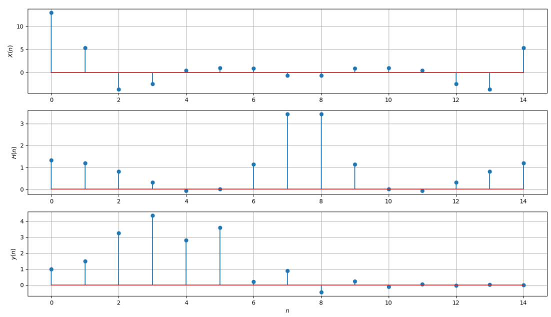

Fig. 6.4: $X(k)$, $H(k)$ and $y(n)$ from fft and IFFT

7 FFT

7.1 The DFT of $x(n)$ is given by

$$X(k) \triangleq \sum_{n=0}^{N-1} x(n) e^{-j2\pi kn/N}, \quad k = 0, 1, \dots, N-1 \tag{7.1}$$

7.2 Let

$$W_N = e^{-j2\pi/N} \tag{7.2}$$

Then the N-point *DFT matrix* is defined as

$$\mathbf{F}_N = [W_N^{mn}] \tag{7.3}$$

where W_N^{mn} are the elements of $\mathbf{F}_N$.

7.3 Let

$$\mathbf{I}_4 = \begin{pmatrix} \mathbf{e}_4^1 & \mathbf{e}_4^2 & \mathbf{e}_4^3 & \mathbf{e}_4^4 \end{pmatrix} \tag{7.4}$$

be the 4×4 identity matrix. Then the 4 point *DFT permutation matrix* is defined as

$$\mathbf{P}_4 = \begin{pmatrix} \mathbf{e}_4^1 & \mathbf{e}_4^3 & \mathbf{e}_4^2 & \mathbf{e}_4^4 \end{pmatrix} \tag{7.5}$$

7.4 The 4 point *DFT diagonal matrix* is defined as

$$\mathbf{D}_4 = diag \begin{pmatrix} W_N^0 & W_N^1 & W_N^2 & W_N^3 \end{pmatrix} \tag{7.6}$$

7.5 Show that

$$W_N^2 = W_{N/2} \tag{7.7}$$

Solution: We know that.

$$W_N = e^{-j2\pi/N} \tag{7.8}$$

Then

$$W_{N/2} = e^{-2*j2\pi/N} \tag{7.9}$$

$$W_{N/2} = {W_N}^2 \tag{7.10}$$

Hence Proved.

7.6 Find $\mathbf{P}_6$.

Solution:

$$\mathbf{I}_6 = \begin{pmatrix} \mathbf{e}_6^1 & \mathbf{e}_6^2 & \mathbf{e}_6^3 & \mathbf{e}_6^4 & \mathbf{e}_6^5 & \mathbf{e}_6^6 \end{pmatrix} \tag{7.11}$$

Now to find the $\mathbf{p}_6$ we have to seperate odd columns and even columns Then

$$\mathbf{P}_6 = \begin{pmatrix} \mathbf{e}_6^1 & \mathbf{e}_6^3 & \mathbf{e}_6^5 & \mathbf{e}_6^2 & \mathbf{e}_6^4 & \mathbf{e}_6^6 \end{pmatrix} \tag{7.12}$$

$$\mathbf{P}_6 = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \tag{7.13}$$

7.7 Find $\mathbf{D}_3$.

Solution:

$$\mathbf{D}_3 = diag \begin{pmatrix} W_3^0 & W_3^1 & W_3^2 \end{pmatrix} \tag{7.14}$$

$$\mathbf{D}_3 = \begin{pmatrix} W_3^0 & 0 & 0 \\ 0 & W_3^1 & 0 \\ 0 & 0 & W_3^2 \end{pmatrix} \tag{7.15}$$

7.8 Show that

$$\mathbf{F}_4 = \begin{bmatrix} \mathbf{I}_2 & \mathbf{D}_2 \\ \mathbf{I}_2 & -\mathbf{D}_2 \end{bmatrix} \begin{bmatrix} \mathbf{F}_2 & 0 \\ 0 & \mathbf{F}_2 \end{bmatrix} \mathbf{P}_4 \tag{7.16}$$

Solution: Observe that for $n \in \mathbb{N}$, $W_4^{4n} = 1$ and $W_4^{4n+2} = -1$. Using (7.7),

$$\mathbf{D}_2\mathbf{F}_2 = \begin{bmatrix} W_4^0 & 0 \\ 0 & W_4^1 \end{bmatrix} \begin{bmatrix} W_2^0 & W_2^0 \\ W_2^0 & W_2^1 \end{bmatrix} \tag{7.17}$$

$$= \begin{bmatrix} W_4^0 & 0 \\ 0 & W_4^1 \end{bmatrix} \begin{bmatrix} W_4^0 & W_4^0 \\ W_4^0 & W_4^2 \end{bmatrix} \tag{7.18}$$

$$= \begin{bmatrix} W_4^0 & W_4^0 \\ W_4^1 & W_4^3 \end{bmatrix} \tag{7.19}$$

$$\implies -\mathbf{D}_2\mathbf{F}_2 = \begin{bmatrix} W_4^2 & W_4^6 \\ W_4^3 & W_4^9 \end{bmatrix} \tag{7.20}$$

and

$$\mathbf{F}_2 = \begin{pmatrix} W_2^0 & W_2^0 \\ W_2^0 & W_2^1 \end{pmatrix} \tag{7.21}$$

$$= \begin{pmatrix} W_4^0 & W_4^0 \\ W_4^0 & W_4^2 \end{pmatrix} \tag{7.22}$$

Hence,

$$\mathbf{W}_4 = \begin{pmatrix} W_4^0 & W_4^0 & W_4^0 & W_4^0 \\ W_4^0 & W_4^2 & W_4^1 & W_4^3 \\ W_4^0 & W_4^4 & W_4^2 & W_4^6 \\ W_4^0 & W_4^6 & W_4^3 & W_4^9 \end{pmatrix} \tag{7.23}$$

$$= \begin{bmatrix} \mathbf{I}_2\mathbf{F}_2 & \mathbf{D}_2 F_2 \\ \mathbf{I}_2\mathbf{F}_2 & -\mathbf{D}_2 F_2 \end{bmatrix} \tag{7.24}$$

$$= \begin{bmatrix} \mathbf{I}_2 & \mathbf{D}_2 \\ \mathbf{I}_2 & \mathbf{D}_2 \end{bmatrix} \begin{bmatrix} \mathbf{F}_2 & 0 \\ 0 & \mathbf{F}_2 \end{bmatrix} \tag{7.25}$$

Multiplying (7.25) by $\mathbf{P}_4$ on both sides, and noting that $\mathbf{W}_4\mathbf{P}_4 = \mathbf{F}_4$ gives us (7.16).

7.9 Show that

$$\mathbf{F}_N = \begin{bmatrix} \mathbf{I}_{N/2} & \mathbf{D}_{N/2} \\ \mathbf{I}_{N/2} & -\mathbf{D}_{N/2} \end{bmatrix} \begin{bmatrix} \mathbf{F}_{N/2} & 0 \\ 0 & \mathbf{F}_{N/2} \end{bmatrix} \mathbf{P}_N \tag{7.26}$$

Solution: Observe that for even N and letting $\mathbf{f}_N^i$ denote the i^{th} column of $\mathbf{F}_N$, from (7.19) and (7.20),

$$\begin{pmatrix} \mathbf{D}_{N/2}\mathbf{F}_{N/2} \\ -\mathbf{D}_{N/2}\mathbf{F}_{N/2} \end{pmatrix} = \begin{pmatrix} \mathbf{f}_N^2 & \mathbf{f}_N^4 & \dots & \mathbf{f}_N^N \end{pmatrix} \tag{7.27}$$

and

$$\begin{pmatrix} \mathbf{I}_{N/2}\mathbf{F}_{N/2} \\ \mathbf{I}_{N/2}\mathbf{F}_{N/2} \end{pmatrix} = \begin{pmatrix} \mathbf{f}_N^1 & \mathbf{f}_N^3 & \dots & \mathbf{f}_N^{N-1} \end{pmatrix} \tag{7.28}$$

Thus,

$$\begin{bmatrix} \mathbf{I}_2\mathbf{F}_2 & \mathbf{D}_2\mathbf{F}_2 \\ \mathbf{I}_2\mathbf{F}_2 & -\mathbf{D}_2\mathbf{F}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{I}_{N/2} & \mathbf{D}_{N/2} \\ \mathbf{I}_{N/2} & -\mathbf{D}_{N/2} \end{bmatrix} \begin{bmatrix} \mathbf{F}_{N/2} & 0 \\ 0 & \mathbf{F}_{N/2} \end{bmatrix}$$

$$= \begin{pmatrix} \mathbf{f}_N^1 & \dots & \mathbf{f}_N^{N-1} & \mathbf{f}_N^2 & \dots & \mathbf{f}_N^N \end{pmatrix} \tag{7.29}$$

and so,

$$\begin{bmatrix} \mathbf{I}_{N/2} & \mathbf{D}_{N/2} \\ \mathbf{I}_{N/2} & -\mathbf{D}_{N/2} \end{bmatrix} \begin{bmatrix} \mathbf{F}_{N/2} & 0 \\ 0 & \mathbf{F}_{N/2} \end{bmatrix} \mathbf{P}_N$$

$$= \begin{pmatrix} \mathbf{f}_N^1 & \mathbf{f}_N^2 & \dots & \mathbf{f}_N^N \end{pmatrix} = \mathbf{F}_N \tag{7.30}$$

7.10 Find

$$\mathbf{P}_4\mathbf{x} \tag{7.31}$$

Solution: We have,

$$\mathbf{P}_4\mathbf{x} = \begin{pmatrix} \mathbf{e}_4^1 & \mathbf{e}_4^3 & \mathbf{e}_4^2 & \mathbf{e}_4^4 \end{pmatrix} \begin{pmatrix} x(0) \\ x(1) \\ x(2) \\ x(3) \end{pmatrix} = \begin{pmatrix} x(0) \\ x(2) \\ x(1) \\ x(3) \end{pmatrix} \tag{7.32}$$

7.11 Show that

$$\mathbf{X} = \mathbf{F}_N\mathbf{x} \tag{7.33}$$

where $\mathbf{x}, \mathbf{X}$ are the vector representations of $x(n), X(k)$ respectively.

Solution:

$$X(k) \triangleq \sum_{n=0}^{N-1} x(n) e^{-j2\pi kn/N}, \quad k = 0, 1, \dots, N-1 \tag{7.34}$$

This can be written as product of two matrices such as

$$\begin{pmatrix} W_N^0 & W_N^1 & \dots & W_N^{N-1} \\ W_N^0 & W_N^{2*1} & \dots & W_N^{2*(N-1)} \\ . & & & \\ . & & & \\ . & & & \\ W_N^{(N-1)*0} & W_N^{(N-1)*1} & \dots & W_N^{(N-1)*(N-1)} \end{pmatrix} * \begin{pmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 2 \\ 1 \end{pmatrix} \tag{7.35}$$

The first matrix is $\mathbf{F_N}$ and the second is $\mathbf{x}$. Hence

$$\mathbf{X} = \mathbf{F}_N\mathbf{x} \tag{7.36}$$

Hence Proved

7.12 Derive the following Step-by-step visualisation of 8-point FFTs into 4-point FFTs and so on

$$\begin{bmatrix} X(0) \\ X(1) \\ X(2) \\ X(3) \end{bmatrix} = \begin{bmatrix} X_1(0) \\ X_1(1) \\ X_1(2) \\ X_1(3) \end{bmatrix} + \begin{bmatrix} W_8^0 & 0 & 0 & 0 \\ 0 & W_8^1 & 0 & 0 \\ 0 & 0 & W_8^2 & 0 \\ 0 & 0 & 0 & W_8^3 \end{bmatrix} \begin{bmatrix} X_2(0) \\ X_2(1) \\ X_2(2) \\ X_2(3) \end{bmatrix} \tag{7.37}$$

$$\begin{bmatrix} X(4) \\ X(5) \\ X(6) \\ X(7) \end{bmatrix} = \begin{bmatrix} X_1(0) \\ X_1(1) \\ X_1(2) \\ X_1(3) \end{bmatrix} - \begin{bmatrix} W_8^0 & 0 & 0 & 0 \\ 0 & W_8^1 & 0 & 0 \\ 0 & 0 & W_8^2 & 0 \\ 0 & 0 & 0 & W_8^3 \end{bmatrix} \begin{bmatrix} X_2(0) \\ X_2(1) \\ X_2(2) \\ X_2(3) \end{bmatrix} \tag{7.38}$$

4-point FFTs into 2-point FFTs

$$\begin{bmatrix} X_1(0) \\ X_1(1) \end{bmatrix} = \begin{bmatrix} X_3(0) \\ X_3(1) \end{bmatrix} + \begin{bmatrix} W_4^0 & 0 \\ 0 & W_4^1 \end{bmatrix} \begin{bmatrix} X_4(0) \\ X_4(1) \end{bmatrix} \tag{7.39}$$

$$\begin{bmatrix} X_1(2) \\ X_1(3) \end{bmatrix} = \begin{bmatrix} X_3(0) \\ X_3(1) \end{bmatrix} - \begin{bmatrix} W_4^0 & 0 \\ 0 & W_4^1 \end{bmatrix} \begin{bmatrix} X_4(0) \\ X_4(1) \end{bmatrix} \tag{7.40}$$

$$\begin{bmatrix} X_2(0) \\ X_2(1) \end{bmatrix} = \begin{bmatrix} X_5(0) \\ X_5(1) \end{bmatrix} + \begin{bmatrix} W_4^0 & 0 \\ 0 & W_4^1 \end{bmatrix} \begin{bmatrix} X_6(0) \\ X_6(1) \end{bmatrix} \quad (7.41)$$

$$\begin{bmatrix} X_2(2) \\ X_2(3) \end{bmatrix} = \begin{bmatrix} X_5(0) \\ X_5(1) \end{bmatrix} - \begin{bmatrix} W_4^0 & 0 \\ 0 & W_4^1 \end{bmatrix} \begin{bmatrix} X_6(0) \\ X_6(1) \end{bmatrix} \quad (7.42)$$

$$P_8 \begin{bmatrix} x(0) \\ x(1) \\ x(2) \\ x(3) \\ x(4) \\ x(5) \\ x(6) \\ x(7) \end{bmatrix} = \begin{bmatrix} x(0) \\ x(2) \\ x(4) \\ x(6) \\ x(1) \\ x(3) \\ x(5) \\ x(7) \end{bmatrix} \quad (7.43)$$

$$P_4 \begin{bmatrix} x(0) \\ x(2) \\ x(4) \\ x(6) \end{bmatrix} = \begin{bmatrix} x(0) \\ x(4) \\ x(2) \\ x(6) \end{bmatrix} \quad (7.44)$$

$$P_4 \begin{bmatrix} x(1) \\ x(3) \\ x(5) \\ x(7) \end{bmatrix} = \begin{bmatrix} x(1) \\ x(5) \\ x(3) \\ x(7) \end{bmatrix} \quad (7.45)$$

Therefore,

$$\begin{bmatrix} X_3(0) \\ X_3(1) \end{bmatrix} = F_2 \begin{bmatrix} x(0) \\ x(4) \end{bmatrix} \quad (7.46)$$

$$\begin{bmatrix} X_4(0) \\ X_4(1) \end{bmatrix} = F_2 \begin{bmatrix} x(2) \\ x(6) \end{bmatrix} \quad (7.47)$$

$$\begin{bmatrix} X_5(0) \\ X_5(1) \end{bmatrix} = F_2 \begin{bmatrix} x(1) \\ x(5) \end{bmatrix} \quad (7.48)$$

$$\begin{bmatrix} X_6(0) \\ X_6(1) \end{bmatrix} = F_2 \begin{bmatrix} x(3) \\ x(7) \end{bmatrix} \quad (7.49)$$

Solution: We write out the values of performing an 8-point FFT on $\mathbf{x}$ as follows.

$$X(k) = \sum_{n=0}^{7} x(n) e^{-\frac{j2kn\pi}{8}} \quad (7.50)$$

$$= \sum_{n=0}^{3} \left(x(2n) e^{-\frac{j2kn\pi}{4}} + e^{-\frac{j2k\pi}{8}} x(2n+1) e^{-\frac{j2kn\pi}{4}} \right) \quad (7.51)$$

$$= X_1(k) + e^{-\frac{j2k\pi}{4}} X_2(k) \quad (7.52)$$

where $\mathbf{X}_1$ is the 4-point FFT of the even-numbered terms and $\mathbf{X}_2$ is the 4-point FFT of the odd numbered terms. Noticing that for $k \geq 4$,

$$X_1(k) = X_1(k-4) \quad (7.53)$$

$$e^{-\frac{j2k\pi}{8}} = -e^{-\frac{j2(k-4)\pi}{8}} \quad (7.54)$$

we can now write out $X(k)$ in matrix form as in (7.37) and (7.38). We also need to solve the two 4-point FFT terms so formed.

$$X_1(k) = \sum_{n=0}^{3} x_1(n) e^{-\frac{j2kn\pi}{8}} \quad (7.55)$$

$$= \sum_{n=0}^{1} \left(x_1(2n) e^{-\frac{j2kn\pi}{4}} + e^{-\frac{j2k\pi}{8}} x_2(2n+1) e^{-\frac{j2kn\pi}{4}} \right) \quad (7.56)$$

$$= X_3(k) + e^{-\frac{j2k\pi}{4}} X_4(k) \quad (7.57)$$

using $x_1(n) = x(2n)$ and $x_2(n) = x(2n+1)$. Thus we can write the 2-point FFTs

$$\begin{bmatrix} X_3(0) \\ X_3(1) \end{bmatrix} = F_2 \begin{bmatrix} x(0) \\ x(4) \end{bmatrix} \quad (7.58)$$

$$\begin{bmatrix} X_4(0) \\ X_4(1) \end{bmatrix} = F_2 \begin{bmatrix} x(2) \\ x(6) \end{bmatrix} \quad (7.59)$$

Using a similar idea for the terms X_2,

$$\begin{bmatrix} X_5(0) \\ X_5(1) \end{bmatrix} = F_2 \begin{bmatrix} x(1) \\ x(5) \end{bmatrix} \quad (7.60)$$

$$\begin{bmatrix} X_6(0) \\ X_6(1) \end{bmatrix} = F_2 \begin{bmatrix} x(3) \\ x(7) \end{bmatrix} \quad (7.61)$$

But observe that from (7.32),

$$\mathbf{P}_8 \mathbf{x} = \begin{pmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{pmatrix} \quad (7.62)$$

$$\mathbf{P}_4 \mathbf{x}_1 = \begin{pmatrix} \mathbf{x}_3 \\ \mathbf{x}_4 \end{pmatrix} \quad (7.63)$$

$$\mathbf{P}_4 \mathbf{x}_2 = \begin{pmatrix} \mathbf{x}_5 \\ \mathbf{x}_6 \end{pmatrix} \quad (7.64)$$

where we define $x_3(k) = x(4k)$, $x_4(k) = x(4k+2)$, $x_5(k) = x(4k+1)$, and $x_6(k) = x(4k+3)$ for $k = 0, 1$.

7.13 For

$$\mathbf{x} = \begin{pmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 2 \\ 1 \end{pmatrix} \quad (7.65)$$

compute the DFT using (7.33)
Solution:

$$\mathbf{X} = \mathbf{F_6 x} \tag{7.66}$$

$$\begin{pmatrix} W_N^0 & W_N^0 & \dots & W_N^0 \\ W_N^0 & W_N^{1*1} & \dots & W_N^{1*(N-1)} \\ . & & & \\ . & & & \\ . & & & \\ W_N^{(N-1)*0} & W_N^{(N-1)*1} & \dots & W_N^{(N-1)*(N-1)} \end{pmatrix} * \begin{pmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 2 \\ 1 \end{pmatrix} \tag{7.67}$$

This will turn out to be.

$$\mathbf{X} = \begin{pmatrix} W_6^0 + 2W_6^0 + 3W_6^0 + 4W_6^0 + 2W_6^0 + W_6^0 \\ W_6^0 + 2W_6^1 + 3W_6^2 + 4W_6^3 + 2W_6^4 + W_6^5 \\ W_6^0 + 2W_6^2 + 3W_6^4 + 4W_6^6 + 2W_6^2 + W_6^4 \\ W_6^0 + 2W_6^3 + 3W_6^6 + 4W_6^3 + 2W_6^0 + W_6^3 \\ W_6^0 + 2W_6^4 + 3W_6^2 + 4W_6^0 + 2W_6^4 + W_6^2 \\ W_6^0 + 2W_6^5 + 3W_6^4 + 4W_6^3 + 2W_6^2 + W_6^1 \end{pmatrix} \tag{7.68}$$

The final result will be.

$$\mathbf{x} = \begin{pmatrix} 13 \\ -4 - \sqrt{3}\jmath \\ 1 \\ -1 \\ 1 \\ -4 + \sqrt{3}\jmath \end{pmatrix} \tag{7.69}$$

7.14 Repeat the above exercise using FFT after zero padding
Solution:

$$\mathbf{F_3} = \begin{pmatrix} W_3^0 & W_3^0 & W_3^0 \\ W_3^0 & W_3^1 & W_3^2 \\ W_3^0 & W_3^2 & W_3^4 \end{pmatrix} \tag{7.70}$$

$$\mathbf{F_3} = \begin{pmatrix} 1 & 1 & 1 \\ 1 & W_3^1 & W_3^2 \\ 1 & W_3^2 & W_3^1 \end{pmatrix} \tag{7.71}$$

$$\begin{pmatrix} X_1(0) \\ X_1(1) \\ X_1(2) \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 \\ 1 & W_3^1 & W_3^2 \\ 1 & W_3^2 & W_3^1 \end{pmatrix} * \begin{pmatrix} 1 \\ 3 \\ 2 \end{pmatrix} \tag{7.72}$$

$$\begin{pmatrix} X_1(0) \\ X_1(1) \\ X_1(2) \end{pmatrix} = \begin{pmatrix} 6 \\ -1.5 - \frac{\sqrt{3}}{2} * \jmath \\ -1.5 + \frac{\sqrt{3}}{2} * \jmath \end{pmatrix} \tag{7.73}$$

Similarly

$$\begin{pmatrix} X_2(0) \\ X_2(1) \\ X_2(2) \end{pmatrix} = \begin{pmatrix} 6 \\ -0.5 - \frac{3\sqrt{3}}{2} * \jmath \\ -0.5 + \frac{3\sqrt{3}}{2} * \jmath \end{pmatrix} \tag{7.74}$$

$$\begin{pmatrix} W_6^0 & 0 & 0 \\ 0 & W_6^1 & 0 \\ 0 & 0 & W_6^2 \end{pmatrix} * \begin{pmatrix} X_2(0) \\ X_2(1) \\ X_2(2) \end{pmatrix} = \begin{pmatrix} X_2(0) \\ W_6^1 * X_2(1) \\ W_6^2 * X_2(2) \end{pmatrix} \tag{7.75}$$

Now

$$\begin{pmatrix} X(0) \\ X(1) \\ X(2) \end{pmatrix} = \begin{pmatrix} X_1(0) \\ X_1(1) \\ X_1(2) \end{pmatrix} + \begin{pmatrix} W_6^0 & 0 & 0 \\ 0 & W_6^1 & 0 \\ 0 & 0 & W_6^2 \end{pmatrix} * \begin{pmatrix} X_2(0) \\ X_2(1) \\ X_2(2) \end{pmatrix} \tag{7.76}$$

$$\begin{pmatrix} X(0) \\ X(1) \\ X(2) \end{pmatrix} = \begin{pmatrix} 13 \\ -4 - \sqrt{3}\jmath \\ 1 \end{pmatrix} \tag{7.77}$$

similarly

$$\begin{pmatrix} X(3) \\ X(4) \\ X(5) \end{pmatrix} = \begin{pmatrix} X_1(0) \\ X_1(1) \\ X_1(2) \end{pmatrix} - \begin{pmatrix} W_6^0 & 0 & 0 \\ 0 & W_6^1 & 0 \\ 0 & 0 & W_6^2 \end{pmatrix} * \begin{pmatrix} X_2(0) \\ X_2(1) \\ X_2(2) \end{pmatrix} \tag{7.78}$$

$$\begin{pmatrix} X(3) \\ X(4) \\ X(5) \end{pmatrix} = \begin{pmatrix} -1 \\ 1 \\ -4 + \sqrt{3}\jmath \end{pmatrix} \tag{7.79}$$

Hence The final result will be.

$$\mathbf{x} = \begin{pmatrix} 13 \\ -4 - \sqrt{3}\jmath \\ 1 \\ -1 \\ 1 \\ -4 + \sqrt{3}\jmath \end{pmatrix} \tag{7.80}$$

7.15 Write a C program to compute the 8-point FFT.
Solution: The following code calculates the 8-point fft of $x(n)$ in 3.1

```
https://github.com/DeepakReddyVelagala/
    EE3900/blob/main/EE3900−2022−
    main/filter/codes/8_point_FFT.c
```

8 Exercises

Answer the following questions by looking at the python code in Problem 2.3.

8.1 The command

```
output_signal = signal.lfilter(b, a,
    input_signal)
```

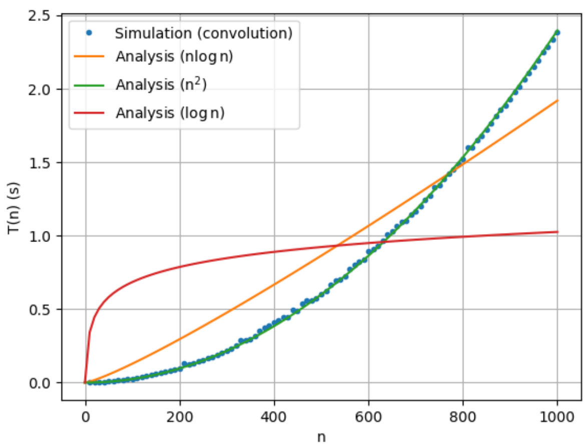

Fig. 7.15: Complexity of convolution

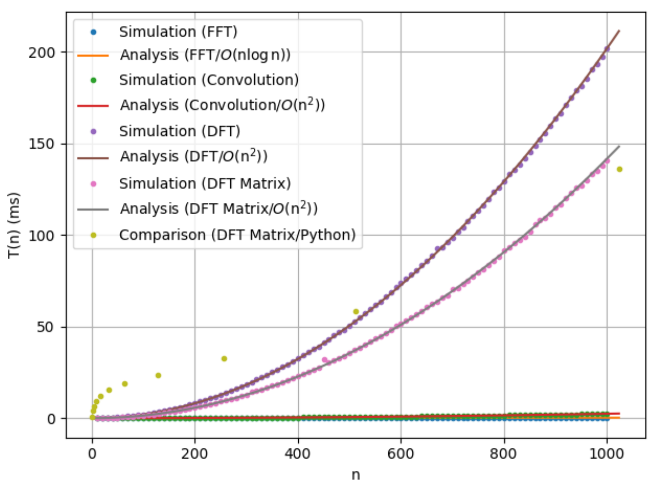

Fig. 7.15: Complexity of FFT

in Problem 2.3 is executed through the following difference equation

$$\sum_{m=0}^{M} a(m)\, y(n-m) = \sum_{k=0}^{N} b(k)\, x(n-k) \quad (8.1)$$

where the input signal is $x(n)$ and the output signal is $y(n)$ with initial values all 0. Replace **signal.filtfilt** with your own routine and verify.
Solution:

https://github.com/DeepakReddyVelagala/EE3900/blob/main/EE3900−2022−main/**filter**/codes/7_1.py

8.2 Repeat all the exercises in the previous sections for the above a and b. **Solution:** For the given values, the difference equation is

$$\begin{aligned} &y(n) - (4.44)\, y(n-1) + (8.78)\, y(n-2) \\ &- (9.93)\, y(n-3) + (6.90)\, y(n-4) \\ &- (2.93)\, y(n-5) + (0.70)\, y(n-6) \\ &- (0.07)\, y(n-7) = \left(5.02 \times 10^{-5}\right) x(n) \\ &+ \left(3.52 \times 10^{-4}\right) x(n-1) + \left(1.05 \times 10^{-3}\right) x(n-2) \\ &+ \left(1.76 \times 10^{-3}\right) x(n-3) + \left(1.76 \times 10^{-3}\right) x(n-4) \\ &+ \left(1.05 \times 10^{-3}\right) x(n-5) + \left(3.52 \times 10^{-4}\right) x(n-6) \\ &+ \left(5.02 \times 10^{-5}\right) x(n-7) \end{aligned} \quad (8.2)$$

From (8.1), we see that the transfer function can be written as follows

$$H(z) = \frac{\sum_{k=0}^{N} b(k) z^{-k}}{\sum_{k=0}^{M} a(k) z^{-k}} \quad (8.3)$$

$$= \sum_i \frac{r(i)}{1 - p(i) z^{-1}} + \sum_j k(j) z^{-j} \quad (8.4)$$

where $r(i)$, $p(i)$, are called residues and poles respectively of the partial fraction expansion of $H(z)$. $k(i)$ are the coefficients of the direct polynomial terms that might be left over. We can now take the inverse z-transform of (8.4) and get using (4.19),

$$h(n) = \sum_i r(i)[p(i)]^n u(n) + \sum_j k(j)\delta(n-j) \quad (8.5)$$

Substituting the values,

$$\begin{aligned} h(n) = &[(2.76)\,(0.55)^n \\ &+ (-1.05 - 1.84\jmath)\,(0.57 + 0.16\jmath)^n \\ &+ (-1.05 + 1.84\jmath)\,(0.57 - 0.16\jmath)^n \\ &+ (-0.53 + 0.08\jmath)\,(0.63 + 0.32\jmath)^n \\ &+ (-0.53 - 0.08\jmath)\,(0.63 - 0.32\jmath)^n \\ &+ (0.20 + 0.004\jmath)\,(0.75 + 0.47\jmath)^n \\ &+ (0.20 - 0.004\jmath)\,(0.75 - 0.47\jmath)^n]u(n) \\ &+ \left(-6.81 \times 10^{-4}\right) \delta(n) \end{aligned} \quad (8.6)$$

The values $r(i)$, $p(i)$, $k(i)$ and thus the impulse response function are computed and plotted at

https://github.com/DeepakReddyVelagala/EE3900/blob/main/EE3900−2022−main/**filter**/codes/7_2_1.py

The filter frequency response is plotted at https://github.com/DeepakReddyVelagala/EE3900/blob/main/EE3900−2022−main/**filter**/codes/7_2_2.py

Observe that for a series $t_n = r^n$, $\frac{t_{n+1}}{t_n} = r$. By the ratio test, t_n converges if $|r| < 1$. We note that observe that $|p(i)| < 1$ and so, as $h(n)$ is the sum of convergent series, we see that $h(n)$ converges. From Fig. (8.2), it is clear that $h(n)$ is bounded. From (4.1),

$$\sum_{n=0}^{\infty} h(n) = H(1) = 1 < \infty \tag{8.7}$$

Therefore, the system is stable. From Fig. (8.2), $h(n)$ is negligible after $n \geq 64$, and we can apply a 64-bit FFT to get y(n). The following code uses the DFT matrix to generate $y(n)$ in Fig. (8.2).

https://github.com/DeepakReddyVelagala/EE3900/blob/main/EE3900−2022−main/**filter**/codes/7_2_3.py˜

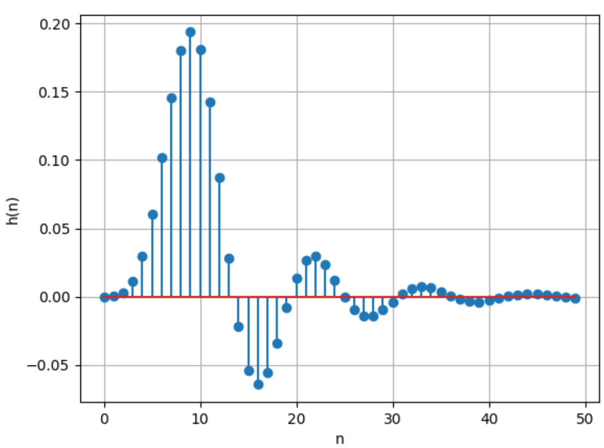

Fig. 8.2: Plot of $h(n)$

8.3 What is the sampling frequency of the input signal?
Solution: run the following code to Sampling frequency(fs)=44.1kHZ.

8.4 What is type, order and cutoff-frequency of the above butterworth filter
Solution: The given butterworth filter is low pass with order=2 and cutoff-frequency=4kHz.

8.5 Modifying the code with different input parameters and to get the best possible output.

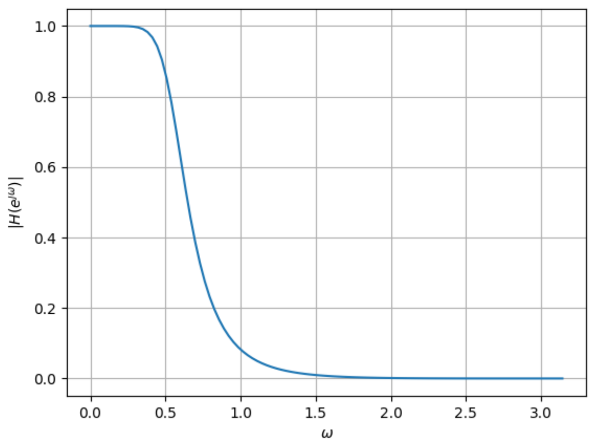

Fig. 8.2: Filter frequency response

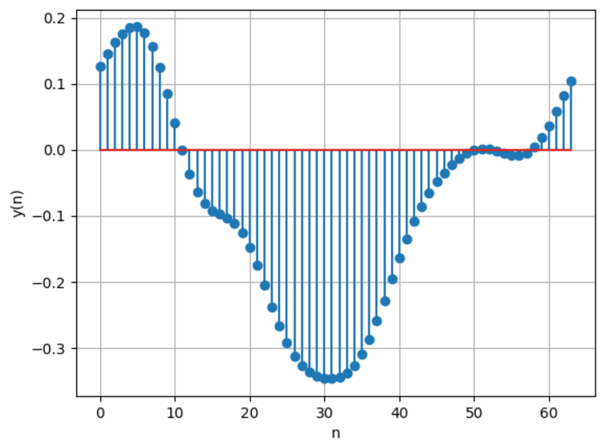

Fig. 8.2: Plot of $y(n)$

Solution: a better filtering was found on changing the order of filter to 7.