Digital Signal Processing

Sumanth N R *

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

Github Repo Link

```
https://github.com/Sigma1084/EE3900
```

2.1 Download the sound file from

```
wget https://raw.githubusercontent.com/
    Sigma1084/EE3900/master/filter/code/
    Sound_Noise.wav
```

*The author is with the Department of Electrical Engineering, Indian Institute of Technology, Hyderabad 502285 India e-mail: ma20btech@iith.ac.in.

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
sample_freq = fs

# order of the filter
order = 4

# cutoff frequency 4kHz
cutoff_freq = 4000.0

# digital frequency
Wn = 2 * cutoff_freq / sample_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
```

2.4 The output of the python script 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

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

Sketch $x(n)$.

3.2 Let

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

$$y(n) = 0, n < 0 \quad (3.2)$$

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

wget https://github.com/Sigma1084/EE3900/
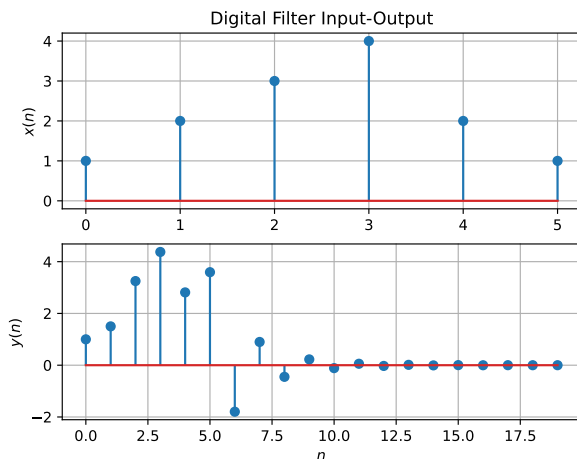   raw/master/**filter**/code/Ex3_xnyn.py

Fig. 3.2

3.3 Repeat the above exercise using a C code.
Solution:

wget https://github.com/Sigma1084/EE3900/
    raw/master/filer/code/Ex3_xnyn.c

4 Z-TRANSFORM

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

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

Show that

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

and find

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

Solution: From (4.1),

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

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

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

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

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

$$X(z) = \sum_{n=-\infty}^{\infty} x(n)z^{-n}$$ (4.7)

Here, we have $x(n) = 0 \ \forall \ n < 0$ and $n \geq 6$

$$X(z) = \sum_{n=0}^{5} x(n)z^{-n}$$ (4.8)

$$\implies X(z) = 1 + \frac{2}{z} + \frac{3}{z^2} + \frac{4}{z^3} + \frac{2}{z^4} + \frac{1}{z^5}$$ (4.9)

4.3 Find

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

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

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

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

4.4 Find the Z transform of

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

and show that the Z-transform of

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

is

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

Solution: It is easy to show that

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

and from (4.14),

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

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

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

4.5 Show that

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

Solution:

Let $u_a(n) := a^n u(n) = \begin{cases} a^n & n \geq 0 \\ 0 & \text{otherwise} \end{cases}$ (4.20)

$$\implies U_a(z) = \sum_{n=0}^{\infty} z^{-n} \cdot a^n \quad (4.21)$$

$$= \frac{1}{1 - az^{-1}}, \quad |z| > |a| \quad (4.22)$$

using the formula for the sum of infinite geometric progression with r = az^{-1} and works since $|r| < 1$

4.6 Let

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

Plot $|H\left(e^{\jmath\omega}\right)|$. Comment. $H(e^{\jmath\omega})$ is known as the *Discret Time Fourier Transform* (DTFT) of $x(n)$.

Solution: The following code plots Fig. 4.6.

```
wget https://raw.githubusercontent.com/
    Sigma1084/EE3900/master/filter/code/
    Ex4_dtft.py
```

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

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

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

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

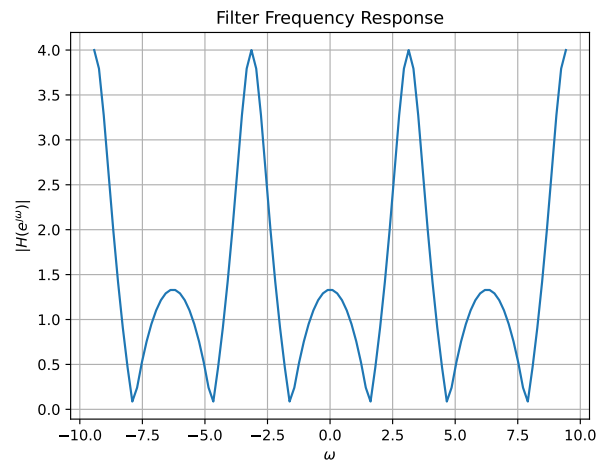$$= \frac{4|\cos\omega|}{\sqrt{5 + 4\cos\omega}}$$

Fundamental Period is clearly 2π.

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

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

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

We know that,

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

Consider the Inverse Discrete Fourier Transform

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

$$= \frac{1}{2\pi} \int_{-\pi}^{\pi} \sum_{m=-\infty}^{\infty} h(m) e^{j\omega(n-m)} d\omega$$

$$= \frac{1}{2\pi} \sum_{m=-\infty}^{\infty} \int_{-\pi}^{\pi} h(m) e^{j\omega(n-m)} d\omega$$

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

Thus,

$$h(n) = \frac{1}{2\pi} \int_{-\pi}^{\pi} H(e^{j\omega}) e^{j\omega n} d\omega \qquad (4.24)$$

Using (5.1),

$$h(n) = \frac{1}{2\pi} \int_{-\pi}^{\pi} \frac{1 + e^{-2j\omega}}{1 + \frac{1}{2} e^{-j\omega}} e^{j\omega n} d\omega$$

This is solved and expressed as a function of n in 5.1

5 IMPULSE RESPONSE

5.1 Using long division, compute $h(n)$ for $n < 5$ from $H(z)$.
Solution: We substitute $x := z^{-1}$, and perform the long division.

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

Thus,

$$H(z) = -4 + 2z^{-1} + \frac{5}{1 + \frac{1}{2}z^{-1}} \qquad (5.1)$$

$$= -4 + 2z^{-1} + 5 \sum_{n=0}^{\infty} \left(-\frac{1}{2}\right)^n z^{-n} \qquad (5.2)$$

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

$$= \sum_{n=0}^{\infty} \left(-\frac{1}{2}\right)^n z^{-n} + 4 \sum_{n=2}^{\infty} \left(-\frac{1}{2}\right)^n z^{-n} \qquad (5.4)$$

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

$$\sum_{n=-\infty}^{\infty} u(n-2) \left(-\frac{1}{2}\right)^{n-2} z^{-n} \qquad (5.5)$$

Therefore, from (4.1),

$$h(n) = \left(-\frac{1}{2}\right)^n u(n) + \left(-\frac{1}{2}\right)^{n-2} u(n-2) \qquad (5.6)$$

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

$$h(n) \overset{\mathcal{Z}}{\rightleftharpoons} H(z) \qquad (5.7)$$

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),

$$H(z) = \frac{1}{1 + \frac{1}{2}z^{-1}} + \frac{z^{-2}}{1 + \frac{1}{2}z^{-1}} \qquad (5.8)$$

$$\implies h(n) = \left(-\frac{1}{2}\right)^n u(n) + \left(-\frac{1}{2}\right)^{n-2} u(n-2) \qquad (5.9)$$

using (4.19) and (4.6).

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

```
wget https://raw.githubusercontent.com/
    Sigma1084/EE3900/master/filter/code/
    Ex5_hn.py
```

We see that $h(n)$ is bounded. For large n,

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

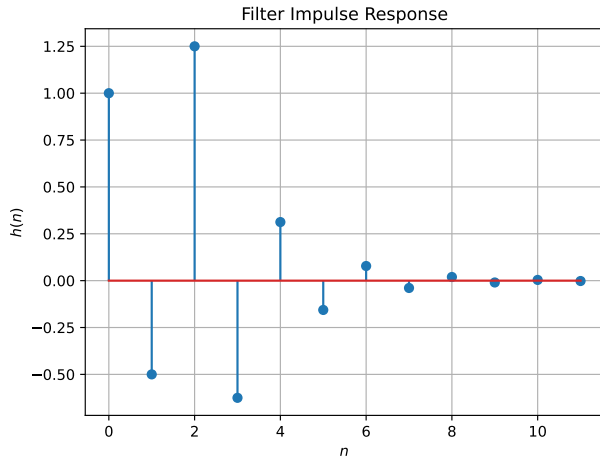$$= \left(-\frac{1}{2}\right)^n (4 + 1) = 5 \left(-\frac{1}{2}\right)^n$$

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

5.4 Convergent? Justify using ratio test.
Solution:
$\lim_{n \to \infty} \left| \frac{h(n+1)}{h(n)} \right| = \frac{1}{2} < 1$ Therefore, $h(n)$ converges.

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

$$\sum_{n=-\infty}^{\infty} h(n) < \infty \qquad (5.10)$$

Is the system defined by (3.2) stable for the impulse response in (5.7)?
Solution:

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

$$= 2 \left(\frac{1}{1 + \frac{1}{2}} \right) = \frac{4}{3} < \infty$$

Thus, the given system is stable.

5.6 Verify the above result using a python code.
Solution: The below code calculates around 100000 terms of $h(n)$ for an approximate since we know $h(n)$ is convergent

wget https://raw.githubusercontent.com/
Sigma1084/EE3900/master/**filter**/code/
Ex5_verify_hn.py

5.7 Compute and sketch $h(n)$ using

$$h(n) + \frac{1}{2}h(n-1) = \delta(n) + \delta(n-2), \quad (5.11)$$

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/
Sigma1084/EE3900/master/**filter**/code/
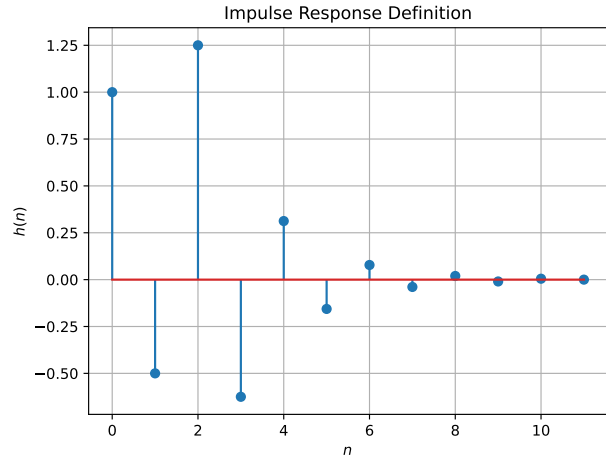Ex5_hndef.py

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

5.8 Compute

$$y(n) = x(n) * h(n) = \sum_{k=-\infty}^{\infty} x(k)h(n-k) \quad (5.12)$$

Comment. The operation in (5.12) 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/
Sigma1084/EE3900/master/**filter**/code/
Ex5_ynconv.py

5.9 Express the above convolution using a Teoplitz matrix.

Solution:

$$\mathbf{y} = \mathbf{x} \circledast \mathbf{h}$$

$$\mathbf{y} = \begin{pmatrix} h_1 & 0 & . & . & . & 0 \\ h_2 & h_1 & . & . & . & 0 \\ h_3 & h_2 & h_1 & . & . & 0 \\ . & . & . & . & . & . \\ 0 & . & . & h_3 & h_2 & h_1 \\ 0 & . & . & . & h_2 & h_1 \\ 0 & . & . & . & 0 & h_1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix}$$
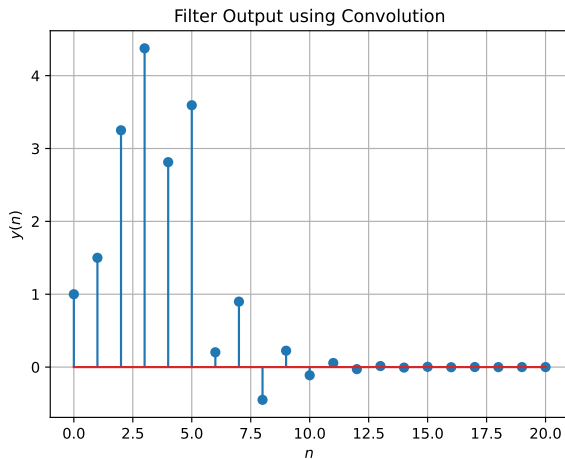
Fig. 5.8: $y(n)$ from the definition of convolution

5.10 Show that

$$y(n) = \sum_{k=-\infty}^{\infty} x(n-k)h(k) \qquad (5.13)$$

Solution: Substituting k = n - k in (5.13), we get

$$y(n) = \sum_{n-k=-\infty}^{\infty} x(k)h(n-k) \qquad (5.14)$$

This is clearly equivalent to (5.12)

6 DFT

6.1 Compute

$$X(k) \triangleq \sum_{n=0}^{N-1} x(n)e^{-j2\pi kn/N}, \quad k = 0, 1, \ldots, N-1 \qquad (6.1)$$

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

wget https://raw.githubusercontent.com/
    Sigma1084/EE3900/master/**filter**/code/
    Ex6_XkHk.py

6.2 Compute

$$Y(k) = X(k)H(k) \qquad (6.2)$$

Solution: The following code yields Yk

wget https://raw.githubusercontent.com/
    Sigma1084/EE3900/master/**filter**/code/
    Ex6_Yk.py

6.3 Compute

$$y(n) = \frac{1}{N} \sum_{k=0}^{N-1} Y(k) \cdot e^{j2\pi kn/N}, \quad n = 0, 1, \ldots, N-1 \qquad (6.3)$$

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/
    Sigma1084/EE3900/master/**filter**/code/
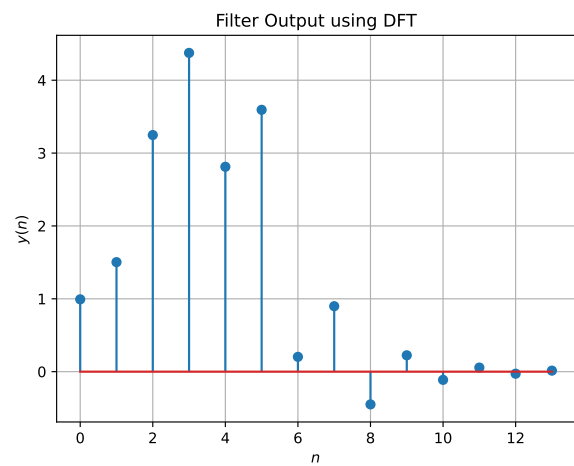   Ex6_yndft.py

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 Fig. 6.4. Note that this is the same as $y(n)$ in Fig. 3.2

$ wget https://raw.githubusercontent.com/
    Sigma1084/EE3900/master/**filter**/code/
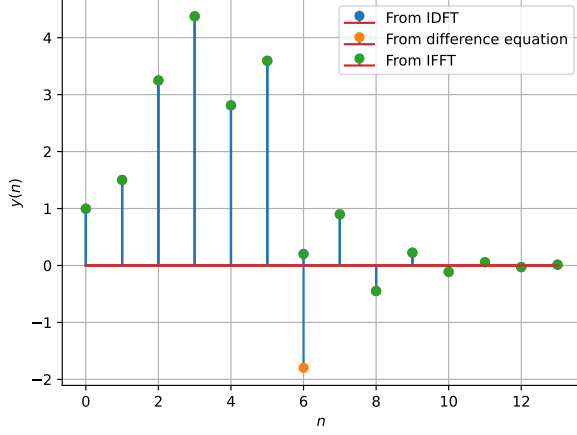   Ex6_ynfft.py

Fig. 6.4: $y(n)$ using FFT and IFFT

7 FFT

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

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

7.2 Let

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

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

$$\mathbf{F}_N = [W_N^{mn}], \quad 0 \le m, n \le N-1$$
(7.3)

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

7.3 Let

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

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

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

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

$$\mathbf{D}_4 = diag \begin{pmatrix} W_8^0 & W_8^1 & W_8^2 & W_8^3 \end{pmatrix}$$
(7.6)

7.5 Show that

$$W_N^2 = W_{N/2}$$
(7.7)

Solution:

$$W_N^2 = \left(e^{-\frac{j2\pi}{N}} \right)^2 = e^{-\frac{j2\pi}{N/2}} = W_{N/2}$$
(7.8)

7.6 Show that

$$\mathbf{F}_4 = \begin{bmatrix} \mathbf{I}_2 & \mathbf{D}_2 \\ \mathbf{I}_2 & -\mathbf{D}_2 \end{bmatrix} \begin{bmatrix} \mathbf{F}_2 & 0 \\ 0 & \mathbf{F}_2 \end{bmatrix} \mathbf{P}_4$$
(7.9)

Solution:

$$\begin{bmatrix} \mathbf{I}_2 & \mathbf{D}_2 \\ \mathbf{I}_2 & -\mathbf{D}_2 \end{bmatrix} \begin{bmatrix} \mathbf{F}_2 & 0 \\ 0 & \mathbf{F}_2 \end{bmatrix}$$
(7.10)

$$= \begin{bmatrix} \mathbf{F}_2 & \mathbf{D}_2\mathbf{F}_2 \\ \mathbf{F}_2 & -\mathbf{D}_2\mathbf{F}_2 \end{bmatrix}$$
(7.11)

$$= \begin{bmatrix} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} & \begin{pmatrix} 1 & 0 \\ 0 & -j \end{pmatrix}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \\ \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} & -\begin{pmatrix} 1 & 0 \\ 0 & -j \end{pmatrix}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \end{bmatrix}$$
(7.12)

$$= \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & -j & j \\ 1 & 1 & -1 & -1 \\ 1 & -1 & j & -j \end{bmatrix}$$
(7.13)

$$\mathbf{P}_4 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$
(7.14)

$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & -j & j \\ 1 & 1 & -1 & -1 \\ 1 & -1 & j & -j \end{bmatrix}\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$
(7.15)

$$= \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix}$$
(7.16)

$$= \begin{bmatrix} W_4^0 & W_4^0 & W_4^0 & W_4^0 \\ W_4^0 & W_4^1 & W_4^2 & W_4^3 \\ W_4^0 & W_4^2 & W_4^4 & W_4^6 \\ W_4^0 & W_4^3 & W_4^6 & W_4^9 \end{bmatrix}$$
(7.17)

$$= \mathbf{F}_4$$
(7.18)

7.7 Show that

$$\mathbf{F}_N = \begin{bmatrix} \mathbf{I}_{N/2} & \mathbf{D}_{N/2} \\ \mathbf{I}_{N/2} & -\mathbf{D}_{N/2} \end{bmatrix} \begin{bmatrix} \mathbf{F}_{N/2} & 0 \\ 0 & \mathbf{F}_{N/2} \end{bmatrix} \mathbf{P}_N$$
(7.19)

Solution: Observe that for even N and letting $\mathbf{f}_N^i$ denote the ith column of $\mathbf{F}_N$,

$$\begin{pmatrix} \mathbf{D}_{N/2}\mathbf{F}_{N/2} \\ -\mathbf{D}_{N/2}\mathbf{F}_{N/2} \end{pmatrix} = \begin{pmatrix} \mathbf{f}_N^2 & \mathbf{f}_N^4 & \cdots & \mathbf{f}_N^N \end{pmatrix}$$
(7.20)

and

$$\begin{pmatrix} \mathbf{I}_{N/2}\mathbf{F}_{N/2} \\ \mathbf{I}_{N/2}\mathbf{F}_{N/2} \end{pmatrix} = \begin{pmatrix} \mathbf{f}_N^1 & \mathbf{f}_N^3 & \cdots & \mathbf{f}_N^{N-1} \end{pmatrix} \qquad (7.21)$$

Thus,

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

$$= \begin{pmatrix} \mathbf{f}_N^1 & \cdots & \mathbf{f}_N^{N-1} & \mathbf{f}_N^2 & \cdots & \mathbf{f}_N^N \end{pmatrix} \qquad (7.22)$$

and so,

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

$$= \begin{pmatrix} \mathbf{f}_N^1 & \mathbf{f}_N^2 & \cdots & \mathbf{f}_N^N \end{pmatrix} = \mathbf{F}_N \qquad (7.23)$$

7.8 Find

$$\mathbf{P}_4\mathbf{x} \qquad (7.24)$$

Solution:

$$\mathbf{P}_4\mathbf{x} = \begin{pmatrix} \mathbf{e}_4^1 & \mathbf{e}_4^3 & \mathbf{e}_4^2 & \mathbf{e}_4^4 \end{pmatrix} \begin{pmatrix} x(0) \\ x(1) \\ x(2) \\ x(3) \end{pmatrix} = \begin{pmatrix} x(0) \\ x(2) \\ x(1) \\ x(3) \end{pmatrix} \qquad (7.25)$$

7.9 Show that

$$\mathbf{X} = \mathbf{F}_N\mathbf{x} \qquad (7.26)$$

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

Solution: Writing the terms of X,

$$X(0) = x(0) + x(1) + \ldots + x(N-1) \qquad (7.27)$$

$$X(1) = x(0) + x(1)e^{-\frac{j2\pi}{N}} + \ldots + $$
$$+ x(N-1)e^{-\frac{j2(N-1)\pi}{N}} \qquad (7.28)$$

$$\vdots$$

$$X(N-1) = x(0) + x(1)e^{-\frac{j2(N-1)\pi}{N}} + \ldots + $$
$$+ x(N-1)e^{-\frac{j2(N-1)(N-1)\pi}{N}} \qquad (7.29)$$

Clearly, the term in the mth row and n^{th} column is given by ($0 \le m \le N-1$ and $0 \le n \le N-1$)

$$T_{mn} = (x(n)e^{-\frac{j2n\pi}{N}})e^{-\frac{j2m\pi}{N}} \qquad (7.30)$$

$$= x(n)e^{-\frac{j2mn\pi}{N}} \qquad (7.31)$$

$$\implies \mathbf{X} = \mathbf{F}_N\mathbf{x} \qquad (7.32)$$

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

$$\begin{bmatrix} X(0) \\ X(1) \\ X(2) \\ X(3) \end{bmatrix} = \begin{bmatrix} X_1(0) \\ X_1(1) \\ X_1(2) \\ X_1(3) \end{bmatrix} + \begin{bmatrix} W_8^0 & 0 & 0 & 0 \\ 0 & W_8^1 & 0 & 0 \\ 0 & 0 & W_8^2 & 0 \\ 0 & 0 & 0 & W_8^3 \end{bmatrix} \begin{bmatrix} X_2(0) \\ X_2(1) \\ X_2(2) \\ X_2(3) \end{bmatrix}$$
$$(7.33)$$

$$\begin{bmatrix} X(4) \\ X(5) \\ X(6) \\ X(7) \end{bmatrix} = \begin{bmatrix} X_1(0) \\ X_1(1) \\ X_1(2) \\ X_1(3) \end{bmatrix} - \begin{bmatrix} W_8^0 & 0 & 0 & 0 \\ 0 & W_8^1 & 0 & 0 \\ 0 & 0 & W_8^2 & 0 \\ 0 & 0 & 0 & W_8^3 \end{bmatrix} \begin{bmatrix} X_2(0) \\ X_2(1) \\ X_2(2) \\ X_2(3) \end{bmatrix}$$
$$(7.34)$$

4-point FFTs into 2-point FFTs

$$\begin{bmatrix} X_1(0) \\ X_1(1) \end{bmatrix} = \begin{bmatrix} X_3(0) \\ X_3(1) \end{bmatrix} + \begin{bmatrix} W_4^0 & 0 \\ 0 & W_4^1 \end{bmatrix} \begin{bmatrix} X_4(0) \\ X_4(1) \end{bmatrix} \qquad (7.35)$$

$$\begin{bmatrix} X_1(2) \\ X_1(3) \end{bmatrix} = \begin{bmatrix} X_3(0) \\ X_3(1) \end{bmatrix} - \begin{bmatrix} W_4^0 & 0 \\ 0 & W_4^1 \end{bmatrix} \begin{bmatrix} X_4(0) \\ X_4(1) \end{bmatrix} \qquad (7.36)$$

$$\begin{bmatrix} X_2(0) \\ X_2(1) \end{bmatrix} = \begin{bmatrix} X_5(0) \\ X_5(1) \end{bmatrix} + \begin{bmatrix} W_4^0 & 0 \\ 0 & W_4^1 \end{bmatrix} \begin{bmatrix} X_6(0) \\ X_6(1) \end{bmatrix} \qquad (7.37)$$

$$\begin{bmatrix} X_2(2) \\ X_2(3) \end{bmatrix} = \begin{bmatrix} X_5(0) \\ X_5(1) \end{bmatrix} - \begin{bmatrix} W_4^0 & 0 \\ 0 & W_4^1 \end{bmatrix} \begin{bmatrix} X_6(0) \\ X_6(1) \end{bmatrix} \qquad (7.38)$$

$$P_8 \begin{bmatrix} x(0) \\ x(1) \\ x(2) \\ x(3) \\ x(4) \\ x(5) \\ x(6) \\ x(7) \end{bmatrix} = \begin{bmatrix} x(0) \\ x(2) \\ x(4) \\ x(6) \\ x(1) \\ x(3) \\ x(5) \\ x(7) \end{bmatrix} \qquad (7.39)$$

$$P_4 \begin{bmatrix} x(0) \\ x(2) \\ x(4) \\ x(6) \end{bmatrix} = \begin{bmatrix} x(0) \\ x(4) \\ x(2) \\ x(6) \end{bmatrix} \qquad (7.40)$$

$$P_4 \begin{bmatrix} x(1) \\ x(3) \\ x(5) \\ x(7) \end{bmatrix} = \begin{bmatrix} x(1) \\ x(5) \\ x(3) \\ x(7) \end{bmatrix} \qquad (7.41)$$

Therefore,

$$\begin{bmatrix} X_3(0) \\ X_3(1) \end{bmatrix} = F_2 \begin{bmatrix} x(0) \\ x(4) \end{bmatrix} \qquad (7.42)$$

$$\begin{bmatrix} X_4(0) \\ X_4(1) \end{bmatrix} = F_2 \begin{bmatrix} x(2) \\ x(6) \end{bmatrix} \qquad (7.43)$$

$$\begin{bmatrix} X_5(0) \\ X_5(1) \end{bmatrix} = F_2 \begin{bmatrix} x(1) \\ x(5) \end{bmatrix} \qquad (7.44)$$

$$\begin{bmatrix} X_6(0) \\ X_6(1) \end{bmatrix} = F_2 \begin{bmatrix} x(3) \\ x(7) \end{bmatrix} \qquad (7.45)$$

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

$$X(k) = \sum_{n=0}^{7} x(n) e^{-\frac{j2kn\pi}{8}} \qquad (7.46)$$

$$= \sum_{n=0}^{3} \left(x(2n) e^{-\frac{j2kn\pi}{4}} + e^{-\frac{j2k\pi}{8}} x(2n+1) e^{-\frac{j2kn\pi}{4}} \right) \qquad (7.47)$$

$$= X_1(k) + e^{-\frac{j2k\pi}{4}} X_2(k) \qquad (7.48)$$

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 $k \geq 4$,

$$X_1(k) = X_1(k-4) \qquad (7.49)$$

$$e^{-\frac{j2k\pi}{8}} = -e^{-\frac{j2(k-4)\pi}{8}} \qquad (7.50)$$

we can now write out $X(k)$ in matrix form.

$$X_1(k) = \sum_{n=0}^{3} x_1(n) e^{-\frac{j2kn\pi}{8}} \qquad (7.51)$$

$$= \sum_{n=0}^{1} \left(x_1(2n) e^{-\frac{j2kn\pi}{4}} + e^{-\frac{j2k\pi}{8}} x_2(2n+1) e^{-\frac{j2kn\pi}{4}} \right) \qquad (7.52)$$

$$= X_3(k) + e^{-\frac{j2k\pi}{4}} X_4(k) \qquad (7.53)$$

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

$$\begin{bmatrix} X_3(0) \\ X_3(1) \end{bmatrix} = F_2 \begin{bmatrix} x(0) \\ x(4) \end{bmatrix} \qquad (7.54)$$

$$\begin{bmatrix} X_4(0) \\ X_4(1) \end{bmatrix} = F_2 \begin{bmatrix} x(2) \\ x(6) \end{bmatrix} \qquad (7.55)$$

Using a similar idea for the terms X_2,

$$\begin{bmatrix} X_5(0) \\ X_5(1) \end{bmatrix} = F_2 \begin{bmatrix} x(1) \\ x(5) \end{bmatrix} \qquad (7.56)$$

$$\begin{bmatrix} X_6(0) \\ X_6(1) \end{bmatrix} = F_2 \begin{bmatrix} x(3) \\ x(7) \end{bmatrix} \qquad (7.57)$$

But observe that from (7.25),

$$\mathbf{P}_8 \mathbf{x} = \begin{pmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{pmatrix} \qquad (7.58)$$

$$\mathbf{P}_4 \mathbf{x}_1 = \begin{pmatrix} \mathbf{x}_3 \\ \mathbf{x}_4 \end{pmatrix} \qquad (7.59)$$

$$\mathbf{P}_4 \mathbf{x}_2 = \begin{pmatrix} \mathbf{x}_5 \\ \mathbf{x}_6 \end{pmatrix} \qquad (7.60)$$

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 $k = 0, 1$.

7.11 For

$$\mathbf{x} = \begin{pmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 2 \\ 1 \end{pmatrix} \qquad (7.61)$$

compte the DFT using (7.26)

7.12 Repeat the above exercise using the FFT after zero padding $\mathbf{x}$.

Solution: The following code yields the DFT Matrix of x

```
$ wget https://raw.githubusercontent.com/
    Sigma1084/EE3900/main/filter/codes/
    Ex7_dftmatrix.py
```

7.13 Write a C program to compute the 8-point FFT.
Solution:

```
$ wget https://raw.githubusercontent.com/
    Sigma1084/EE3900/main/filter/codes/
    Ex7_8point.c
```

Run the code using the command below

```
$ gcc Ex7_8point.c −lm −o Ex7.out && ./
    Ex7.out
```

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)
```

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.

8.2 Repeat all the exercises in the previous sections for the above a and b.

8.3 What is the sampling frequency of the input signal?
Solution: 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.