Digital Signal Processing

G V V Sharma*

(:	O	Tν	\mathbf{E}	VΊ	ГS

1	Software Installation	1
2	Digital Filter	1
3	Difference Equation	2
4	Z-transform	2
5	Impulse Response	3
6	DFT	4
7	FFT	5
8	Exercises	6

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

*The author is with the Department of Electrical Engineering, Indian Institute of Technology, Hyderabad 502285 India e-mail: gadepall@iith.ac.in. All content in the manuscript is released under GNU GPL. Free to use for anything.

Problem 2.1 in the spectrogram and play. Observe the spectrogram. What do you find? **Solution:** There are a lot of yellow lines be-

tween 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.way'
#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

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

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

Sketch x(n).

3.2 Let

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

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

Sketch y(n).

Solution: The following code yields Fig. 3.2.

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

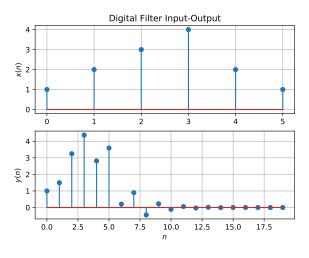


Fig. 3.2

3.3 Repeat the above exercise using a C code.

4 Z-TRANSFORM

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

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

Show that

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

and find

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

Solution: From (4.1),

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

$$= \sum_{n=-\infty}^{\infty} x(n)z^{-n-1} = z^{-1} \sum_{n=-\infty}^{\infty} x(n)z^{-n}$$
(4.4)
$$(4.5)$$

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.

4.3 Find

$$H(z) = \frac{Y(z)}{X(z)} \tag{4.7}$$

from (3.2) assuming that the Z-transform is a linear operation.

Solution: Applying (4.6) in (3.2),

$$Y(z) + \frac{1}{2}z^{-1}Y(z) = X(z) + z^{-2}X(z)$$
 (4.8)

$$\implies \frac{Y(z)}{X(z)} = \frac{1+z^{-2}}{1+\frac{1}{2}z^{-1}} \tag{4.9}$$

4.4 Find the Z transform of

$$\delta(n) = \begin{cases} 1 & n = 0 \\ 0 & \text{otherwise} \end{cases}$$
 (4.10)

and show that the Z-transform of

$$u(n) = \begin{cases} 1 & n \ge 0 \\ 0 & \text{otherwise} \end{cases}$$
 (4.11)

is

$$U(z) = \frac{1}{1 - z^{-1}}, \quad |z| > 1$$
 (4.12)

Solution: It is easy to show that

$$\delta(n) \stackrel{\mathcal{Z}}{\rightleftharpoons} 1 \tag{4.13}$$

and from (4.11),

$$U(z) = \sum_{n=0}^{\infty} z^{-n}$$
 (4.14)

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

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

4.5 Show that

$$a^n u(n) \stackrel{\mathcal{Z}}{\rightleftharpoons} \frac{1}{1 - az^{-1}} \quad |z| > |a| \tag{4.16}$$

4.6 Let

$$H(e^{j\omega}) = H(z = e^{j\omega}).$$
 (4.17)

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

Solution: The following code plots Fig. 4.6.

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

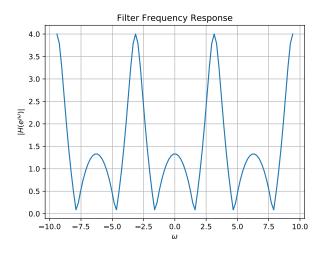


Fig. 4.6: $|H(e^{j\omega})|$

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

5 Impulse Response

5.1 Using long division, find

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

for H(z) in (4.9).

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

$$h(n) \stackrel{\mathcal{Z}}{\rightleftharpoons} H(z) \tag{5.2}$$

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 (4.9),

$$H(z) = \frac{1}{1 + \frac{1}{2}z^{-1}} + \frac{z^{-2}}{1 + \frac{1}{2}z^{-1}}$$
 (5.3)

$$\implies h(n) = \left(-\frac{1}{2}\right)^n u(n) + \left(-\frac{1}{2}\right)^{n-2} u(n-2) \tag{5.4}$$

using (4.16) 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

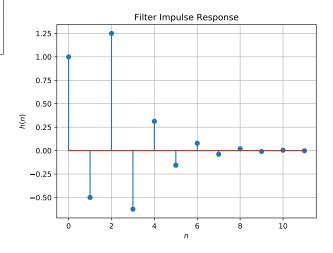


Fig. 5.3: h(n) as the inverse of H(z)

- 5.4 Convergent? Justify using the ratio test.
- 5.5 The system with h(n) is defined to be stable if

$$\sum_{n=-\infty}^{\infty} h(n) < \infty \tag{5.5}$$

Is the system defined by (3.2) stable for the impulse response in (5.2)?

- 5.6 Verify the above result using a python code.
- 5.7 Compute and sketch h(n) using

$$h(n) + \frac{1}{2}h(n-1) = \delta(n) + \delta(n-2), \quad (5.6)$$

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

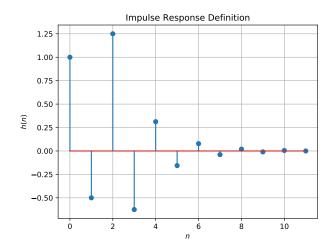


Fig. 5.7: h(n) from the definition

5.8 Compute

$$y(n) = x(n) * h(n) = \sum_{n=-\infty}^{\infty} x(k)h(n-k)$$
 (5.7)

Comment. The operation in (5.7) 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/gadepall/EE1310/master/filter/codes/ynconv.py

- 5.9 Express the above convolution using a Teoplitz matrix.
- 5.10 Show that

$$y(n) = \sum_{n = -\infty}^{\infty} x(n - k)h(k)$$
 (5.8)

6 DFT

6.1 Compute

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

and H(k) using h(n).

6.2 Compute

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

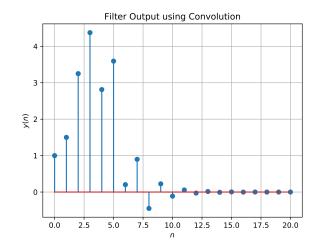


Fig. 5.8: y(n) from the definition of convolution

6.3 Compute

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

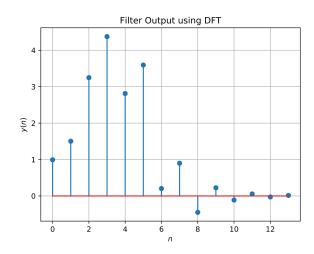


Fig. 6.3: y(n) from the DFT

6.4 Repeat the previous exercise by computing X(k), H(k) and y(n) through FFT and IFFT.

7 FFT

1. The DFT of x(n) is given by

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

2. Let

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

Then the N-point DFT matrix is defined as

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

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

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

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

4. The 4 point DFT diagonal matrix is defined as

$$\mathbf{D}_4 = diag \begin{pmatrix} W_4^0 & W_N^1 & W_N^2 & W_N^3 \end{pmatrix}$$
 (7.6)

5. Show that

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

6. Show that

$$\mathbf{F}_4 = \begin{bmatrix} \mathbf{I}_2 & \mathbf{D}_2 \\ \mathbf{I}_2 & -\mathbf{D}_2 \end{bmatrix} \begin{bmatrix} \mathbf{F}_2 & 0 \\ 0 & \mathbf{F}_2 \end{bmatrix} \mathbf{P}_4 \tag{7.8}$$

7. Show that

$$\mathbf{F}_{N} = \begin{bmatrix} \mathbf{I}_{N/2} & \mathbf{D}_{N/2} \\ \mathbf{I}_{N/2} & -\mathbf{D}_{N/2} \end{bmatrix} \begin{bmatrix} \mathbf{F}_{N/2} & 0 \\ 0 & \mathbf{F}_{N/2} \end{bmatrix} \mathbf{P}_{N} \quad (7.9)$$

8. Find

$$\mathbf{P}_4\mathbf{x} \tag{7.10}$$

9. Show that

$$\mathbf{X} = \mathbf{F}_N \mathbf{x} \tag{7.11}$$

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

10. Derive the following Step-by-step visualisation

of 8-point FFTs into 4-point FFTs and so on

$$\begin{bmatrix} X(0) \\ X(1) \\ X(2) \\ X(3) \end{bmatrix} = \begin{bmatrix} X_1(0) \\ X_1(1) \\ X_1(2) \\ X_1(3) \end{bmatrix} + \begin{bmatrix} W_8^0 & 0 & 0 & 0 \\ 0 & W_8^1 & 0 & 0 \\ 0 & 0 & W_8^2 & 0 \\ 0 & 0 & 0 & W_8^3 \end{bmatrix} \begin{bmatrix} X_2(0) \\ X_2(1) \\ X_2(2) \\ X_2(3) \end{bmatrix}$$
(7.12)

$$\begin{bmatrix} X(4) \\ X(5) \\ X(6) \\ X(7) \end{bmatrix} = \begin{bmatrix} X_1(0) \\ X_1(1) \\ X_1(2) \\ X_1(3) \end{bmatrix} - \begin{bmatrix} W_8^0 & 0 & 0 & 0 \\ 0 & W_8^1 & 0 & 0 \\ 0 & 0 & W_8^2 & 0 \\ 0 & 0 & 0 & W_8^3 \end{bmatrix} \begin{bmatrix} X_2(0) \\ X_2(1) \\ X_2(2) \\ X_2(3) \end{bmatrix}$$
 (7.13)

4-point FFTs into 2-point FFTs

$$\begin{bmatrix} X_1(0) \\ X_1(1) \end{bmatrix} = \begin{bmatrix} X_3(0) \\ X_3(1) \end{bmatrix} + \begin{bmatrix} W_4^0 & 0 \\ 0 & W_4^1 \end{bmatrix} \begin{bmatrix} X_4(0) \\ X_4(1) \end{bmatrix}$$
(7.14)

$$\begin{bmatrix} X_1(2) \\ X_1(3) \end{bmatrix} = \begin{bmatrix} X_3(0) \\ X_3(1) \end{bmatrix} - \begin{bmatrix} W_4^0 & 0 \\ 0 & W_4^1 \end{bmatrix} \begin{bmatrix} X_4(0) \\ X_4(1) \end{bmatrix}$$
(7.15)

$$\begin{bmatrix} X_2(0) \\ X_2(1) \end{bmatrix} = \begin{bmatrix} X_5(0) \\ X_5(1) \end{bmatrix} + \begin{bmatrix} W_4^0 & 0 \\ 0 & W_4^1 \end{bmatrix} \begin{bmatrix} X_6(0) \\ X_6(1) \end{bmatrix}$$
 (7.16)

$$P_{8} \begin{bmatrix} x(0) \\ x(1) \\ x(2) \\ x(3) \\ x(4) \\ x(5) \\ x(6) \\ x(7) \end{bmatrix} = \begin{bmatrix} x(0) \\ x(2) \\ x(4) \\ x(6) \\ x(1) \\ x(3) \\ x(5) \\ x(7) \end{bmatrix}$$
 (7.18)

$$P_{4} \begin{bmatrix} x(0) \\ x(2) \\ x(4) \\ x(6) \end{bmatrix} = \begin{bmatrix} x(0) \\ x(4) \\ x(2) \\ x(6) \end{bmatrix}$$
 (7.19)

$$P_{4} \begin{bmatrix} x(1) \\ x(3) \\ x(5) \\ x(7) \end{bmatrix} = \begin{bmatrix} x(1) \\ x(5) \\ x(3) \\ x(7) \end{bmatrix}$$
 (7.20)

Therefore,

$$\begin{bmatrix} X_3(0) \\ X_3(1) \end{bmatrix} = F_2 \begin{bmatrix} x(0) \\ x(4) \end{bmatrix}$$
 (7.21)

$$\begin{bmatrix} X_4(0) \\ X_4(1) \end{bmatrix} = F_2 \begin{bmatrix} x(2) \\ x(6) \end{bmatrix}$$
 (7.22)

$$\begin{bmatrix} X_5(0) \\ X_5(1) \end{bmatrix} = F_2 \begin{bmatrix} x(1) \\ x(5) \end{bmatrix}$$
 (7.23)

$$\begin{bmatrix} X_6(0) \\ X_6(1) \end{bmatrix} = F_2 \begin{bmatrix} x(3) \\ x(7) \end{bmatrix}$$
 (7.24)

11. For

$$\mathbf{x} = \begin{pmatrix} 1\\2\\3\\4\\2\\1 \end{pmatrix} \tag{7.25}$$

compte the DFT using (7.11)

- 12. Repeat the above exercise using the FFT after zero padding **x**.
- 13. Write a C program to compute the 8-point FFT.

8 Exercises

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

8.1 The command

in Problem 2.3 is executed through the following difference equation

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