# Digital Signal Processing

Gunjit Mittal (AI21BTECH11011)

## Contents


**1 Software Installation** 1

**2 Digital Filter** 1

**3 Difference Equation** 1

**4 $Z$-transform** 2

**5 Impulse Response** 3

**6 DFT and FFT** 4

**7 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 Problem 2.1 in the spectrogram and play. Observe the spectrogram. What do you find?
**Solution:** By observing spectrogram, it clearly shows that tonal frequency is under 4kHz. And above 4kHz only noise is present.

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
print(input_signal, fs)
# order of the filter
order = 4
# cutoff frquency 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
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 audio is subdued and the higher frequenices are just blank.

## 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

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

Sketch $y(n)$.

**Solution:** The following code yields Fig. 3.2.

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

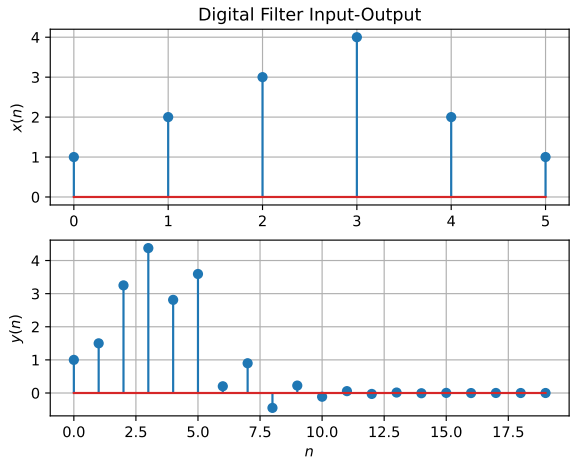


Fig. 3.2. figure

## 4 $Z$-TRANSFORM

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

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

Show that

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

and find

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

**Solution:** From (4.1),

$$\mathcal{Z}\{x(n-1)\} = \sum_{n=-\infty}^{\infty} x(n-1)z^{-n} \quad (4.4)$$
$$= \sum_{n=-\infty}^{\infty} x(n)z^{-n-1} \quad (4.5)$$
$$= z^{-1}\sum_{n=-\infty}^{\infty} x(n)z^{-n} \quad (4.6)$$

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

$$\mathcal{Z}\{x(n-k)\} = \sum_{n=-\infty}^{\infty} x(n-k)z^{-n} \quad (4.7)$$
$$= \sum_{n=-\infty}^{\infty} x(n)z^{-n-k} \quad (4.8)$$
$$= z^{-k}\sum_{n=-\infty}^{\infty} x(n)z^{-n} \quad (4.9)$$

4.2 Find

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

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

**Solution:** We have

$$y(n) + \frac{1}{2}y(n-1) = x(n) + x(n-2) \quad (4.11)$$

Finding the Z-transform

$$\mathcal{Z}\{y(n) + \frac{1}{2}y(n-1)\} = \mathcal{Z}\{x(n) + x(n-2)\} \quad (4.12)$$

Because Z-transform is a linear function

$$\mathcal{Z}y(n) + \frac{1}{2}\mathcal{Z}y(n-1) = \mathcal{Z}x(n) + \mathcal{Z}x(n-2) \quad (4.13)$$

Using (4.9)

$$Y(z) + \frac{1}{2}z^{-1}Y(z) = X(z) + z^{-2}X(z) \quad (4.14)$$

$$\implies \frac{Y(z)}{X(z)} = \frac{1+z^{-2}}{1+\frac{1}{2}z^{-1}} \quad (4.15)$$

4.3 Find the Z transform of

$$\delta(n) = \begin{cases} 1 & n = 0 \\ 0 & \text{otherwise} \end{cases} \quad (4.16)$$

and show that the $Z$-transform of

$$u(n) = \begin{cases} 1 & n \geq 0 \\ 0 & \text{otherwise} \end{cases} \quad (4.17)$$

is

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

**Solution:**

$$\delta(n) \overset{\mathcal{Z}}{\rightleftharpoons} \sum_{n=-\infty}^{\infty} \delta(n) z^{-n} \tag{4.19}$$

$$= \sum_{n=0} z^{-n} = 1 \tag{4.20}$$

and from (4.17),

$$u(n) \overset{\mathcal{Z}}{\rightleftharpoons} U(z) = \sum_{n=-\infty}^{\infty} u(n) z^{-n} \tag{4.21}$$

$$U(z) = \sum_{n=0}^{\infty} z^{-n} \tag{4.22}$$

$$= \frac{1}{1-z^{-1}}, \quad |z| > 1 \tag{4.23}$$

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

4.4 Show that

$$a^n u(n) \overset{\mathcal{Z}}{\rightleftharpoons} \frac{1}{1-az^{-1}} \quad |z| > |a| \tag{4.24}$$

**Solution:**

$$a^n u(n) \overset{\mathcal{Z}}{\rightleftharpoons} \sum_{n=-\infty}^{\infty} a^n u(n) z^{-n} \tag{4.25}$$

$$= \sum_{n=0}^{\infty} (az^{-1})^{-n} \tag{4.26}$$

$$= \frac{1}{1-az^{-1}}, \quad |z| > |a| \tag{4.27}$$

4.5 Let

$$H\left(e^{\mathrm{J}\omega}\right) = H\left(z = e^{\mathrm{J}\omega}\right). \tag{4.28}$$

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

**Solution:** The following code plots Fig. 4.5.

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

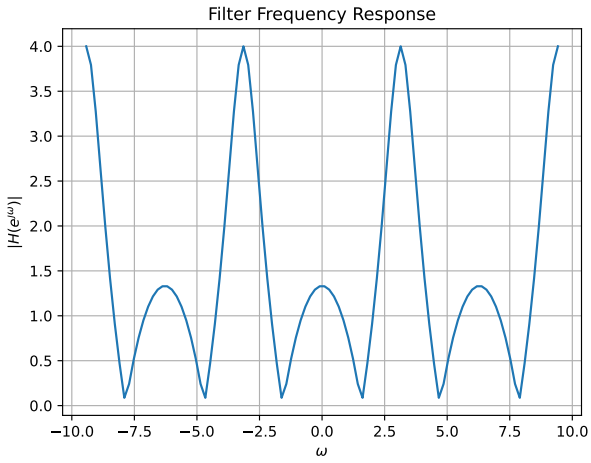


Fig. 4.5. $\left|H\left(e^{\mathrm{J}\omega}\right)\right|$

## 5 Impulse Response

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

$$h(n) \overset{\mathcal{Z}}{\rightleftharpoons} H(z) \tag{5.1}$$

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 (4.15),

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

$$\implies h(n) = \left(-\frac{1}{2}\right)^n u(n) + \left(-\frac{1}{2}\right)^{n-2} u(n-2) \tag{5.3}$$

using (4.24) and (4.9).

5.2 Sketch $h(n)$. Is it bounded? Convergent?

**Solution:** The following code plots Fig. 5.2.

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

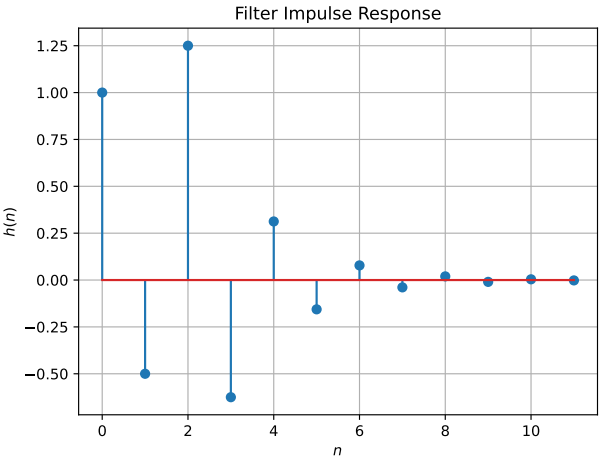


Fig. 5.2. $h(n)$ as the inverse of $H(z)$

As we can see from the plot $h(n)$ is both bounded and convergent.

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

$$\sum_{n=-\infty}^{\infty} h(n) < \infty \tag{5.4}$$

Is the system defined by (3.2) stable for the impulse response in (5.1)?

**Solution:** We can caluclate the sum of the values and it comes out to be 1.33203125 which is clearly less than $\infty$

$\therefore$ The system is stable

5.4 Compute and sketch $h(n)$ using

$$h(n) + \frac{1}{2}h(n-1) = \delta(n) + \delta(n-2), \tag{5.5}$$

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

**Solution:** The following code plots Fig. 5.4. Note that this is the same as Fig. 5.2.

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

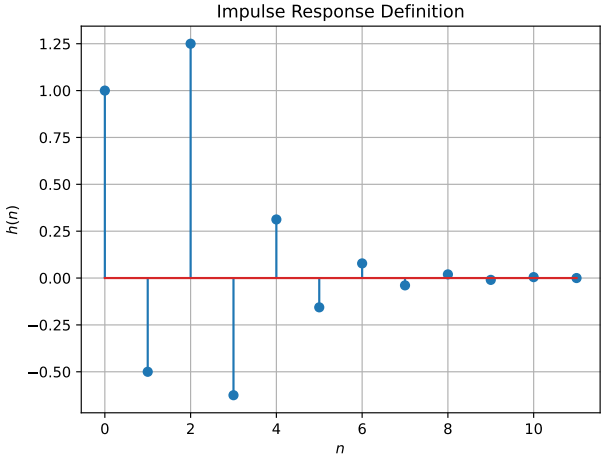


Fig. 5.4. $h(n)$ from the definition

5.5 Compute

$$y(n) = x(n) * h(n) = \sum_{k=-\infty}^{\infty} x(k)h(n-k) \tag{5.6}$$

Comment. The operation in (5.6) is known as *convolution*.

**Solution:** The following code plots Fig. 5.5. 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
```

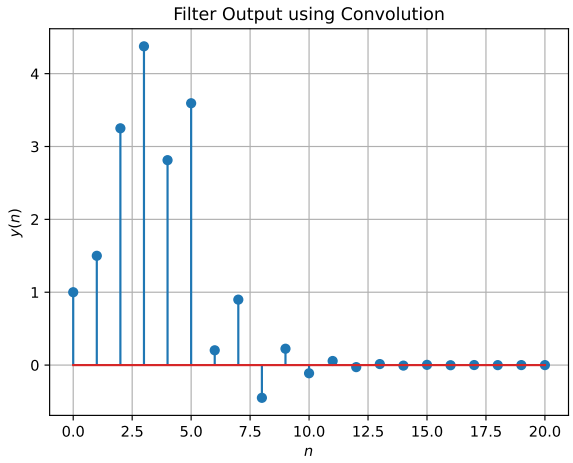


Fig. 5.5. $y(n)$ from the definition of convolution

5.6 Show that

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

**Solution:** From (5.6)

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

Substituting $k$ with $n-k$

$$= \sum_{n-k=-\infty}^{\infty} x(n-k)h(k) \tag{5.9}$$

as $n$ remains constant, we can rewrite it as

$$= \sum_{k=-\infty}^{\infty} x(n-k)h(k) \tag{5.10}$$

## 6 DFT AND FFT

6.1 Compute

$$X(k) \stackrel{\triangle}{=} \sum_{n=0}^{N-1} x(n)e^{-\mathrm{J}2\pi kn/N}, \quad k = 0, 1, \ldots, N-1 \tag{6.1}$$

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

**Solution:** The following code plots Fig. 6.1.

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

6.2 Compute

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

**Solution:** The following code plots Fig. 6.2.

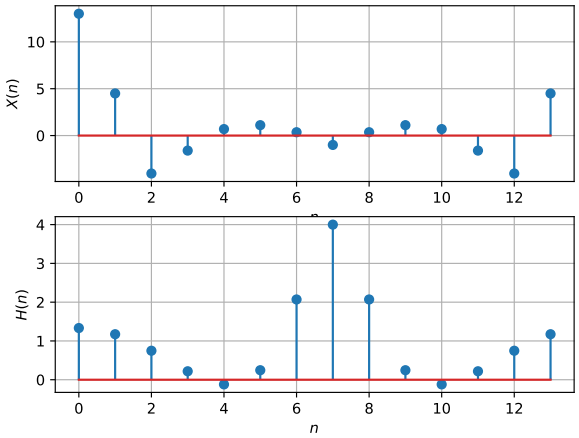


Fig. 6.1. $X(n) and H(n)$

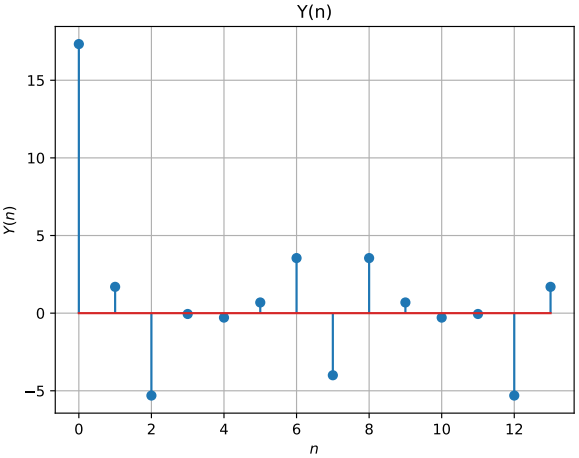


Fig. 6.2. $Y(n)$

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

6.3 Compute

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

**Solution:** The following code plots Fig. 5.5. 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

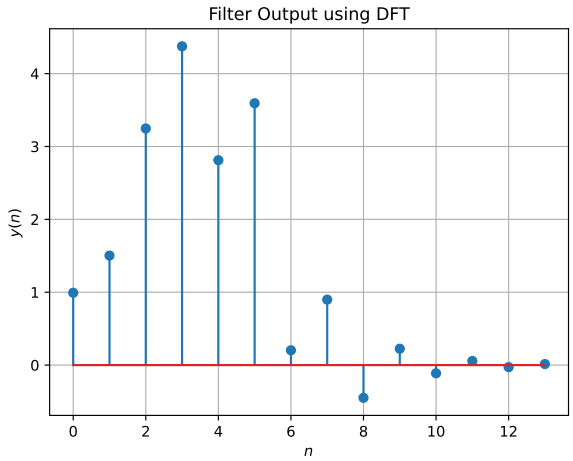


Fig. 6.3. $y(n)$ from the DFT

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

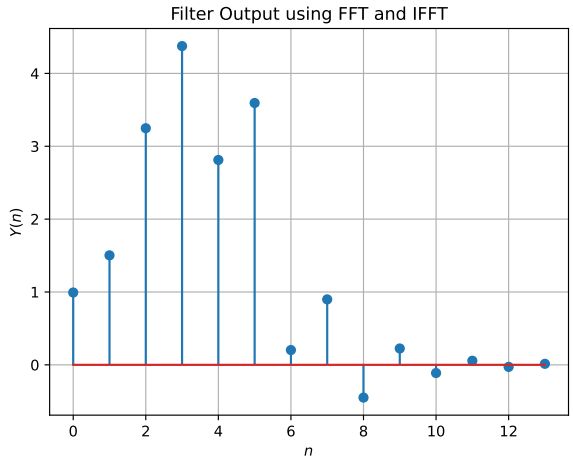


Fig. 6.4. $y(n)$ using FFT and IFFT

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

6.5 Wherever possible, express all the above equations as matrix equations.
**Solution:** We use the DFT Matrix, where $\omega = e^{-\frac{j2k\pi}{N}}$, which is given by

$$\mathbf{W}=\begin{pmatrix}\omega^0 & \omega^0 & \ldots & \omega^0\\ \omega^0 & \omega^1 & \ldots & \omega^{N-1}\\ \vdots & \vdots & \ddots & \vdots\\ \omega^0 & \omega^{N-1} & \ldots & \omega^{(N-1)(N-1)}\end{pmatrix} \tag{6.4}$$

i.e. $W_{jk} = \omega^{jk}$, $0 \leq j, k < N$. Hence, we can write any DFT equation as

$$\mathbf{X} = \mathbf{W}\mathbf{x} = \mathbf{x}\mathbf{W} \tag{6.5}$$

where

$$\mathbf{x} = \begin{pmatrix} x(0) \\ x(1) \\ \vdots \\ x(n-1) \end{pmatrix} \tag{6.6}$$

Using (6.3), the inverse Fourier Transform is given by

$$\mathbf{x} = \mathcal{F}^{-1}(\mathbf{X}) = \mathbf{W}^{-1}\mathbf{X} = \frac{1}{N}\mathbf{W}^{\mathbf{H}}\mathbf{X} = \frac{1}{N}\mathbf{X}\mathbf{W}^{\mathbf{H}} \tag{6.7}$$

$$\implies \mathbf{W}^{-1} = \frac{1}{N}\mathbf{W}^{\mathbf{H}} \tag{6.8}$$

where $H$ denotes hermitian operator. We can rewrite (6.2) using the element-wise multiplication operator as

$$\mathbf{Y} = \mathbf{H} \cdot \mathbf{X} = (\mathbf{W}\mathbf{h}) \cdot (\mathbf{W}\mathbf{x}) \tag{6.9}$$

## 7 Exercises

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

7.1 The command

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

in Problem 2.3 is executed through the following difference equation

$$\sum_{m=0}^{M} a(m)\, y(n-m) = \sum_{k=0}^{N} b(k)\, x(n-k) \tag{7.1}$$

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.

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

7.3 What is the sampling frequency of the input signal?
**Solution:** Sampling frequency(fs)=44.1kHZ.

7.4 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.

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