

G V V Sharma\*

## CONTENTS

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

**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
```

\*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.

- 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 frequency 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

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

Sketch  $x(n)$ .

3.2 Let

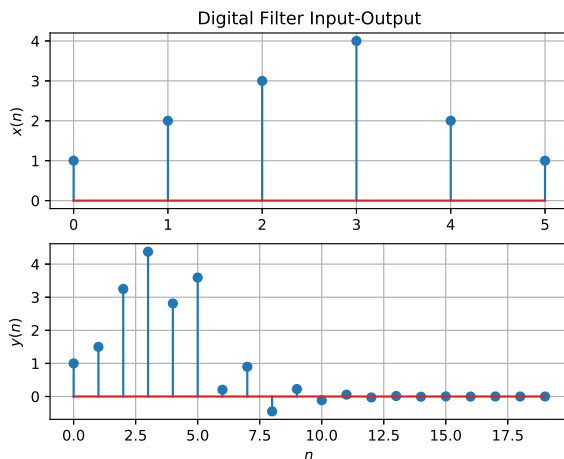
$$y(n) + \frac{1}{2}y(n-1) = x(n) + x(n-2),$$

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

Sketch  $y(n)$ .

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

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



and from (4.11),

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

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

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

4.4 Show that

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

4.5 Let

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

Plot  $|H(e^{j\omega})|$ . Comment.  $H(e^{j\omega})$  is known as the *Discrete 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
```

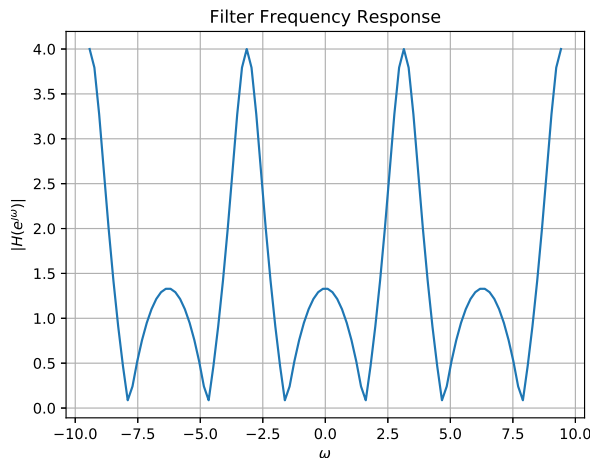


Fig. 4.5:  $|H(e^{j\omega})|$

## 5 IMPULSE RESPONSE

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

$$h(n) \stackrel{Z}{\rightleftharpoons} H(z) \quad (5.1)$$

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 (4.9),

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

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

using (4.16) and (4.6).

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
```

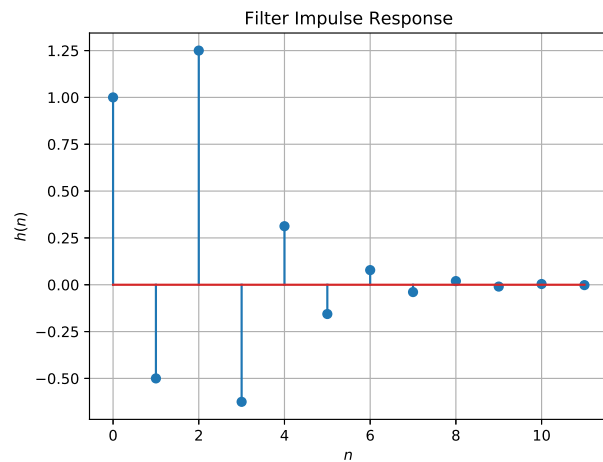


Fig. 5.2:  $h(n)$  as the inverse of  $H(z)$

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

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

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

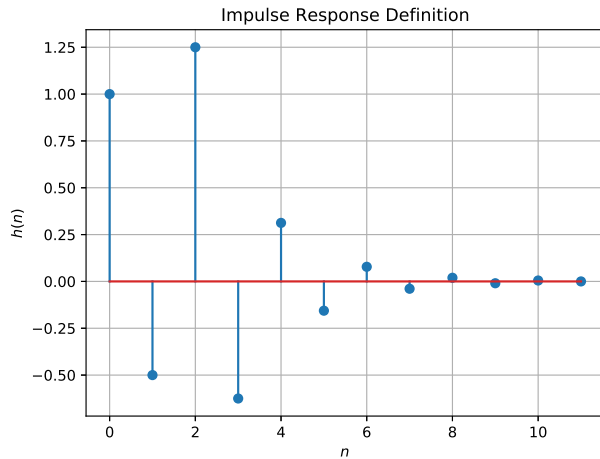
5.4 Compute and sketch  $h(n)$  using

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

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

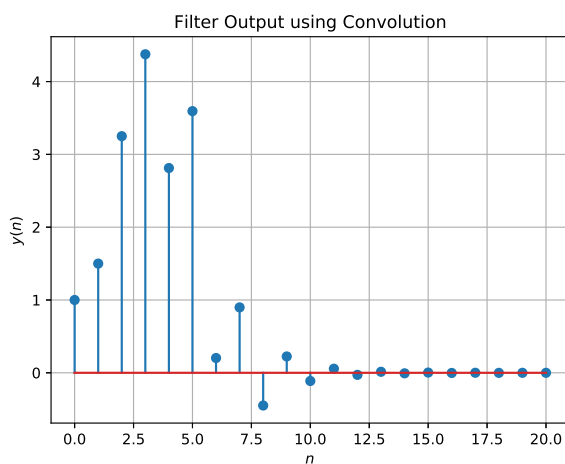
## 5.5 Compute

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

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

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

## 6 DFT AND FFT

## 6.1 Compute

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

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

## 6.2 Compute

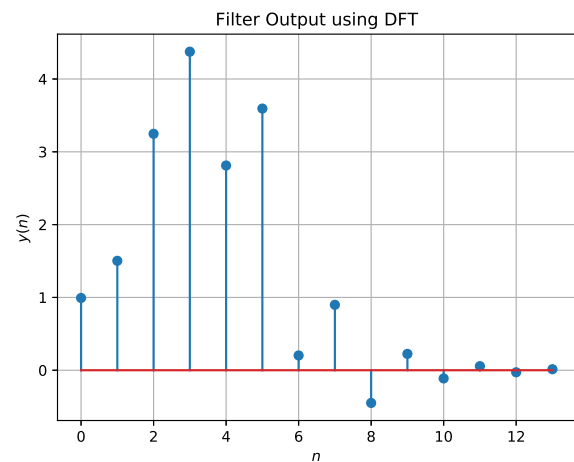
$$Y(k) = X(k)H(k) \quad (6.2)$$

## 6.3 Compute

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

**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.

6.5 Wherever possible, express all the above equations as matrix equations.

## 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

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

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.