Digital Signal Processing

sumeeth kumar

Contents

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

1 Software Installation

Run the following commands (commands may change depending on Linux distro)

```
$ sudo apt update && sudo apt upgrade
$ sudo apt install libffi−dev libsndfile1 python3−
    scipy python3−numpy python3−matplotlib
$ pip3 install cffi pysoundfile
```

2 Digital Filter

2.1 Download the sound file using

```
$https://github.com/sumeethkumar12/signal−
    processing/blob/main/
    Sound_With_ReducedNoise.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: Download the source code using

```
$ https://github.com/sumeethkumar12/signal−
    processing/blob/main/codes/cancel_noise.
    py
```

and execute it using

```
$ python3 cancel_noise.py
```

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

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

Sketch $x(n)$.

3.2 Let

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

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

```
$ https://github.com/sumeethkumar12/signal−
    processing/blob/main/codes/xnyn.py
```

and execute it using

```
$ python3 xnyn.py
```

4 Z-transform

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

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

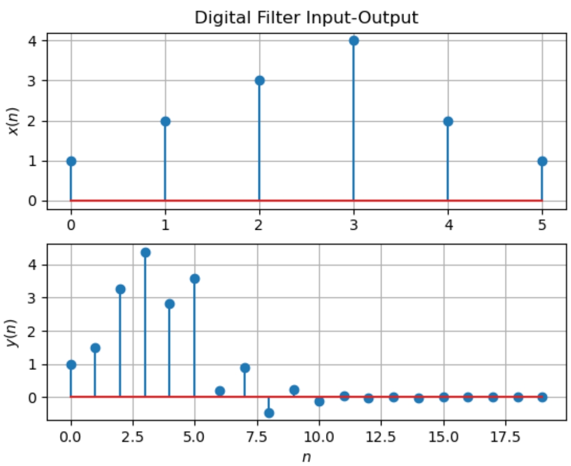

Fig. 3.2: Plot of $x(n)$ and $y(n)$

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),

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

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

$$= z^{-k}\sum_{n=-\infty}^{\infty} x(n)z^{-n} \tag{4.6}$$

$$= z^{-k}X(n) \tag{4.7}$$

Putting $k = 1$ gives (4.2).

4.2 Find

$$H(z) = \frac{Y(z)}{X(z)} \tag{4.8}$$

from (3.2) assuming that the Z-transform is a linear operation. **Solution:** Applying (4.7) in (3.2),

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

$$\implies \frac{Y(z)}{X(z)} = \frac{1+z^{-2}}{1+\frac{1}{2}z^{-1}} \tag{4.10}$$

4.3 Find the Z transform of

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

and show that the Z-transform of

$$u(n) = \begin{cases} 1 & n \geq 0 \\ 0 & \text{otherwise} \end{cases} \tag{4.12}$$

is

$$U(z) = \frac{1}{1-z^{-1}}, \quad |z| > 1 \tag{4.13}$$

Solution: We see using (4.11) that

$$\mathcal{Z}\{\delta(n)\} = \delta(0) = 1 \tag{4.14}$$

and from (4.12),

$$U(z) = \sum_{n=0}^{\infty} z^{-n} \tag{4.15}$$

$$= \frac{1}{1-z^{-1}}, \quad |z| > 1 \tag{4.16}$$

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

4.4 Show that

$$a^n u(n) \overset{\mathcal{Z}}{\rightleftharpoons} \frac{1}{1-az^{-1}} \quad |z| > |a| \tag{4.17}$$

Solution:

$$a^n u(n) \overset{\mathcal{Z}}{\rightleftharpoons} \sum_{n=0}^{\infty} \left(az^{-1}\right)^n \tag{4.18}$$

$$= \frac{1}{1-az^{-1}} \quad |z| > |a| \tag{4.19}$$

4.5 Let

$$H\left(e^{\jmath\omega}\right) = H\left(z = e^{\jmath\omega}\right). \tag{4.20}$$

Plot $|H(e^{\jmath\omega})|$. Comment. $H(e^{\jmath\omega})$ is known as the *Discret Time Fourier Transform* (DTFT) of $x(n)$. **Solution:** The following code plots Fig. (4.5).

```
$ https://github.com/sumeethkumar12/signal−
    processing/blob/main/codes/dtft.py
```

The figure can be generated using

```
$ python3 dtft.py
```

We observe that $\left|H\left(e^{j\omega}\right)\right|$ is periodic with fundamental period 2π.

5 Impulse Response

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

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

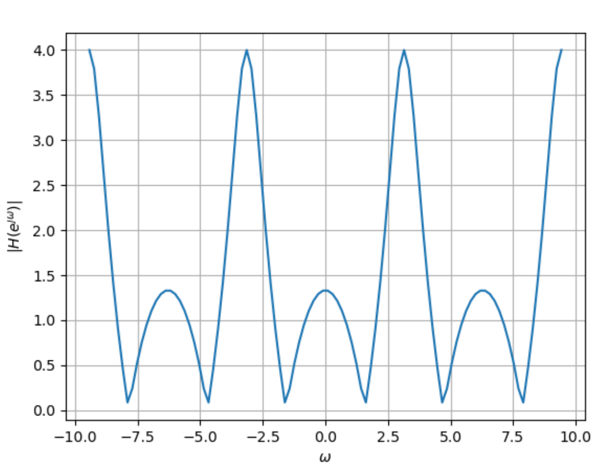

Fig. 4.5: Plot of $\left|H\left(e^{j\omega}\right)\right|$ against ω

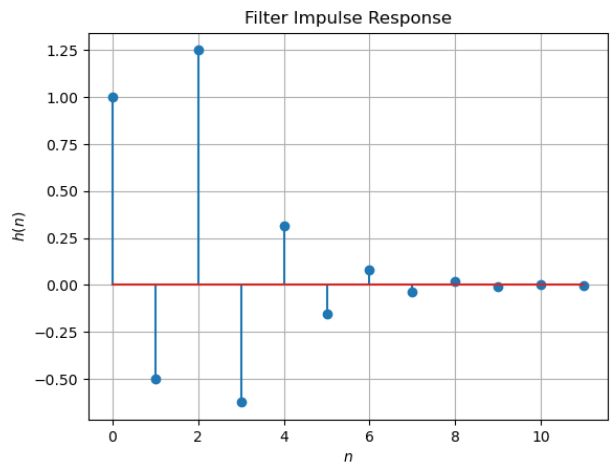

Fig. 5.2: $h(n)$ as the inverse of $H(z)$

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 (4.10),

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

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

using (4.17) and (4.7).

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

```
wget https://github.com/sumeethkumar12/
    signal−processing/blob/main/figs/h_n.png
```

Use the following command in the terminal to run the code

```
python3 h_n.py
```

we can say it is 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:

$$u(n) = \begin{cases} 1 & n \geq 0 \\ 0 & n < 0 \end{cases} \tag{5.5}$$

$$u(n-2) = \begin{cases} 1 & n \geq 2 \\ 0 & n < 2 \end{cases} \tag{5.6}$$

$$\therefore h(n) = \begin{cases} 0 & n < 0 \\ \left(\frac{-1}{2}\right)^n & 0 \leq n < 2 \\ \left(\frac{-1}{2}\right)^n + \left(\frac{-1}{2}\right)^{(n-2)} & n \geq 2 \end{cases} \tag{5.7}$$

$$\therefore \sum_{n=-\infty}^{\infty} h(n) = 0 + 1 + \frac{-1}{2} + \sum_{n=2}^{\infty}\left[\left(\frac{-1}{2}\right)^n + \left(\frac{-1}{2}\right)^{(n-2)}\right] \tag{5.8}$$

$$= \frac{1}{2} + \frac{5}{4} * \left(\frac{2}{3}\right) = \frac{4}{3} < \infty \tag{5.9}$$

$$\tag{5.10}$$

$\therefore$ system defined is stable

5.4 Compute and sketch $h(n)$ using

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

This is the definition of $h(n)$.
Solution: The following code plots Fig. 5.4. Note that this is the same as Fig. **??**.

```
wget https://github.com/sumeethkumar12/
    signal−processing/blob/main/codes/hndef.
    py
```

Use the following command in the terminal to run the code

```
python3 hndef.py
```

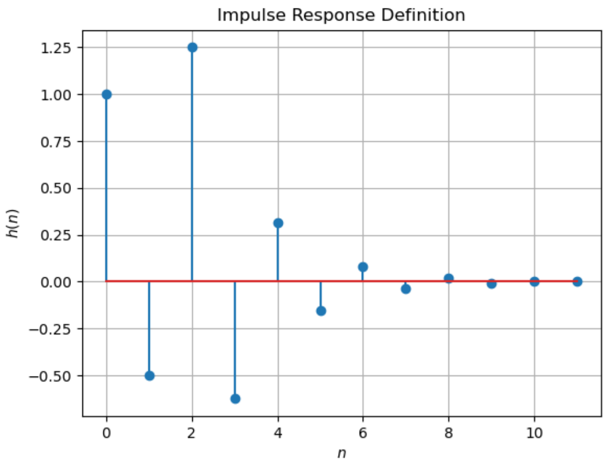

Fig. 5.4: $h(n)$ from the definition

5.5 Compute

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

Comment. The operation in (5.12) 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://github.com/sumeethkumar12/
    signal-processing/blob/main/codes/
    ynconv.py
```

Use the following command in the terminal to run the code

```
python3 ynconv.py
```

5.6 Show that

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

Solution: wkt

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

Replacing k with $n-k$

$$= \sum_{n-k=-\infty}^{\infty} x(n-k)h(k) \therefore y(n) = \sum_{k=-\infty}^{\infty} x(n-k)h(k) \quad (5.15)$$

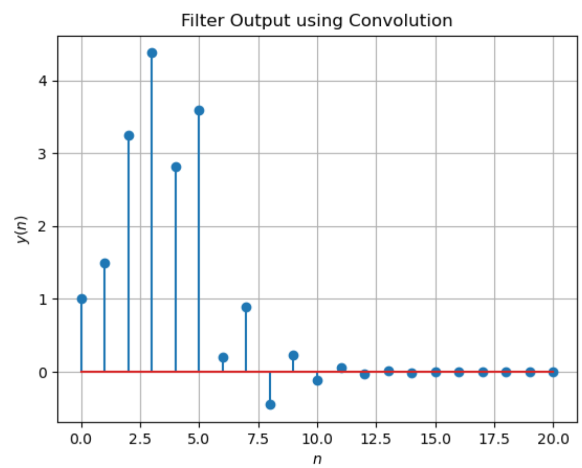

Fig. 5.5: $y(n)$ from the definition of convolution