Random Numbers

JARPULA BHANU PRASAD - AI21BTECH11015

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

1 SOFTWARE INSTALLATION

1.1 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://github.com/jarpula−Bhanu/
    EE3900/blob/main/Ass/soundfiles/
    Sound_Noise.wav
```

2.2 You will find a spectogram 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 spectogram. What do you find?
Solution: There are a lot of yellow lines betweeen 440Hz to 5.1KHz. 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 nosie and execute the code.
Solution: Download and run the following code.

```
wget https://github.com/jarpula−Bhanu/
    EE3900/blob/main/Ass/Codes/2.3_noise.
    py
```

run the above code using the command

```
python3 2.3_noise.py
```

2.4 The output of the python scripy in problem 2.3 is the audio file Sound_With_ReducedNoise.wav. Play the file in the spectogram 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 5.1KHz.

3 DIFFERENCE EQUATION

3.1 Let

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

Sketch x(n).
Solution: Download and run the following code.Below code plots fig(3.1)

```
wget https://github.com/jarpula−Bhanu/
    EE3900/blob/main/Ass/Codes/3.1.py
```
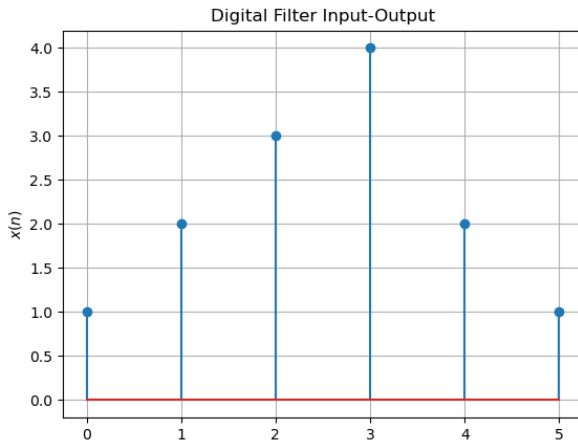
run the above code using the command

```
python3 3.1.py
```

3.2 Let

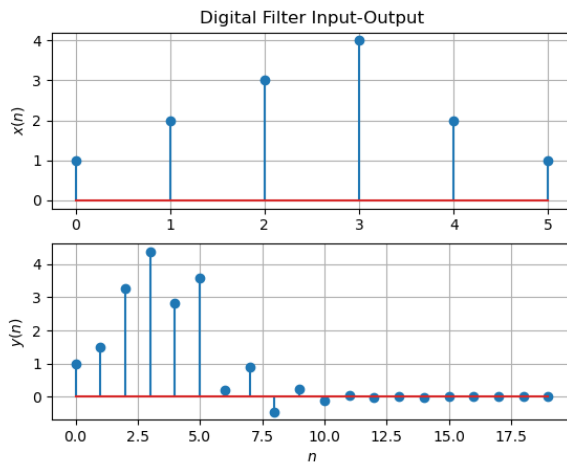$$y(n) + \frac{1}{2}y(n-1) = x(n) + x(n-2), y(n) = 0, n < 0 \tag{3.2}$$

Sketech y(n).
Solution: Download and run the following code.Below code plots fig(3.2)

Fig. 3.1: Sketch of $x(n)$

wget https://github.com/jarpula−Bhanu/
EE3900/blob/main/Ass/Codes/3.2.py

run the above code using the command

python3 3.2.py

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

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)

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

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

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

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

4.2 Find

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

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

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

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

4.3 Find the Z transform of

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

and show that the Z-transform of

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

is

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

Solution: It is easy to show that

$$\delta(n) \overset{\mathcal{Z}}{\rightleftharpoons} 1 \quad (4.13)$$

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 fomula for the sum of an infinite

geometric progression.

4.4 Show that

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

Solution:

$$a^n u(n) \overset{\mathcal{Z}}{\rightleftharpoons} \sum_{n=0}^{\infty} (az^{-1})^n \qquad (4.17)$$

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

4.5 Let

$$H(e^{j\omega}) = H(z = e^{j\omega}). \qquad (4.19)$$

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

Solution: Download and run the following code. The following code plots Fig. 4.5.

wget https://github.com/jarpula−Bhanu/
    EE3900/blob/main/Ass/Codes/4.5.py

run the above code using the command

python3 4.5.py

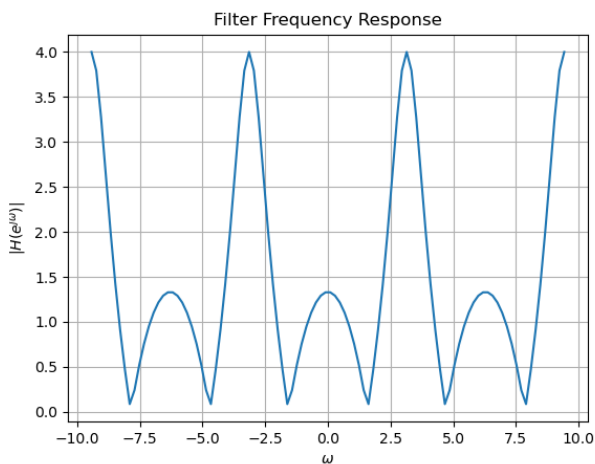We observe that $|H(e^{j\omega})|$ is periodic with fundamental period 2π.

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

5 IMPULSE RESPONSE

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

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

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),

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

$$\implies h(n) = \left(-\frac{1}{2}\right)^n u(n) + \left(-\frac{1}{2}\right)^{n-2} u(n-2) \qquad (5.3)$$
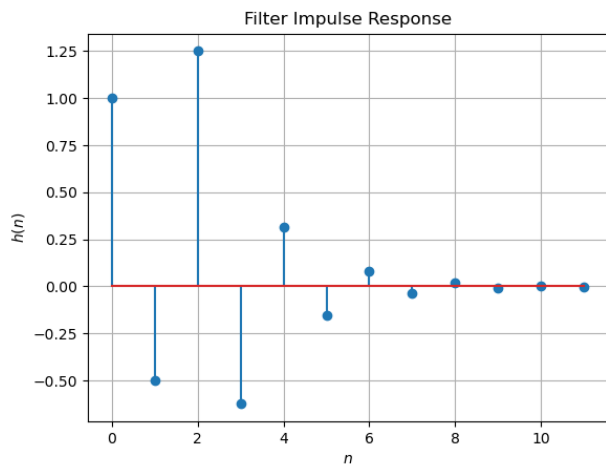
using (4.16) and (4.6).

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

wget https://github.com/jarpula−Bhanu/
    EE3900/blob/main/Ass/Codes/5.2.py

run the above code using the command.

python3 5.2.py

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

$h(n)$ is bounded and convergent.

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

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

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

Solution: Note that

$$\sum_{n=-\infty}^{\infty} h(n) = \sum_{n=-\infty}^{\infty} \left(-\frac{1}{2}\right)^n u(n) + \left(-\frac{1}{2}\right)^{n-2} u(n-2) \tag{5.5}$$

$$= 2\left(\frac{1}{1+\frac{1}{2}}\right) = \frac{4}{3} \tag{5.6}$$

Thus, the given system is stable.

5.4 Compute and sketch $h(n)$ using

$$h(n) + \frac{1}{2}h(n-1) = \delta(n) + \delta(n-2), \tag{5.7}$$

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://github.com/jarpula−Bhanu/
EE3900/blob/main/Ass/Codes/5.4.py
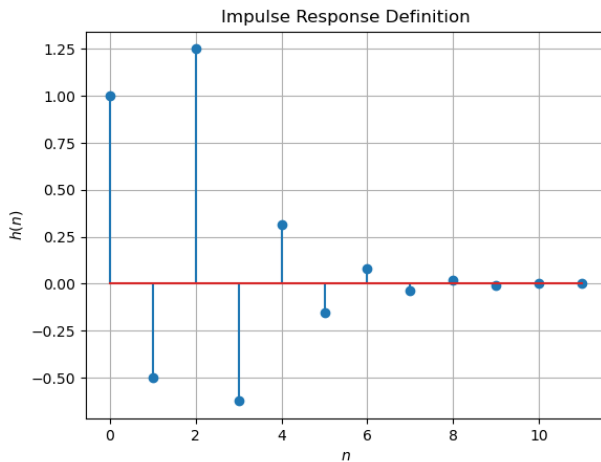
run the above code using the command.

python3 5.4.py

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

5.5 Compute

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

Comment. The operation in (5.8) 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/jarpula−Bhanu/
EE3900/blob/main/Ass/Codes/5.5.py
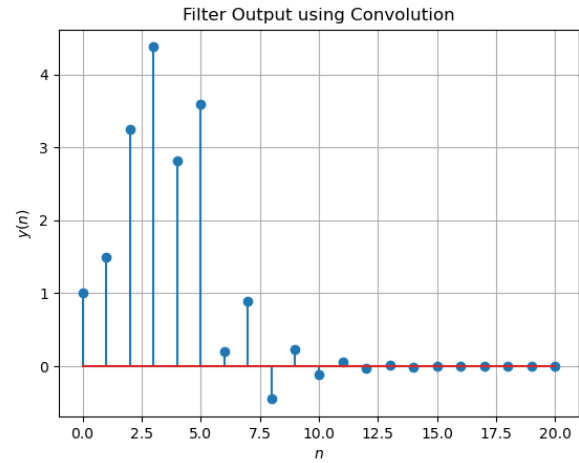
run the above code using the command.

python3 5.5.py

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

5.6 Show that

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

Solution: from 5.8, we substitute $k := n - k$ to get

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

$$= \sum_{n-k=-\infty}^{\infty} x(n-k)h(k) \tag{5.11}$$

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

6 DFT and FFT

6.1 Compute

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

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

Solution: The following code plots Fig. 6.1.

wget https://github.com/jarpula−Bhanu/
EE3900/blob/main/Ass/Codes/6.1.py

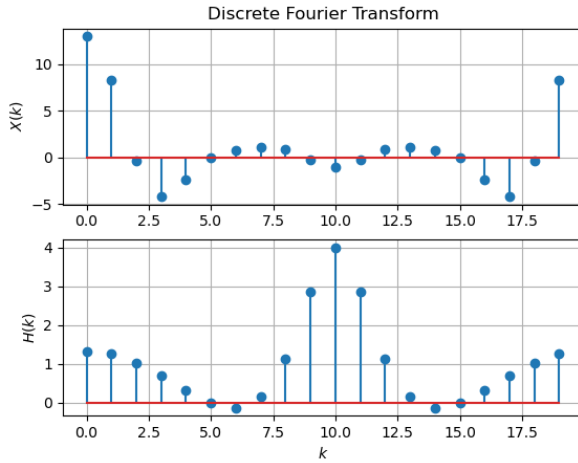run the above code using the command.

python3 6.1.py

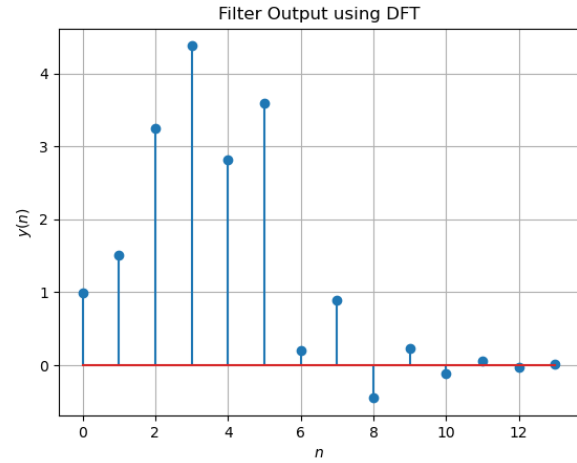Fig. 6.1: Plots of the real parts of the DFT of $x(n)$ and $h(n)$

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

6.2 Compute

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

Solution: Download and run the following code.

```
wget https://github.com/jarpula−Bhanu/
    EE3900/blob/main/Ass/Codes/6.2.py
```

run the above code using the command.

```
python3 6.2.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. 5.5. Note that this is the same as $y(n)$ in Fig. 3.2.

```
wget https://github.com/jarpula−Bhanu/
    EE3900/blob/main/Ass/Codes/6.3.py
```

run the above code using the command.

```
python3 6.3.py
```

6.4 Repeat the previous exercise by computing $X(k), H(k)$ and $y(n)$ through FFT and IFFT.
Solution: Download the code from

```
wget https://github.com/jarpula−Bhanu/
    EE3900/blob/main/Ass/Codes/6.4.py
%
```
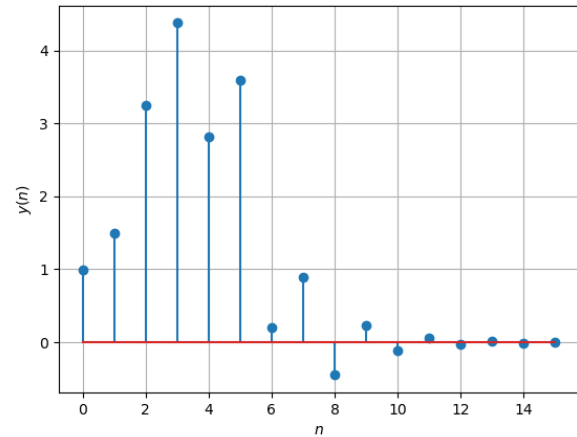

Fig. 6.4: $y(n)$ using FFT and IFFT

and execute it using

```
$ python3 6.4.py
```

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

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

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

i.e. $W_{jk} = \omega^{jk}$, $0 \le j, k < N$. Hence, we can

write any DFT equation as

$$\mathbf{X} = \mathbf{Wx} = \mathbf{xW} \qquad (6.5)$$

where

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

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

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

$$\implies \mathbf{W}^{-1} = \frac{1}{N}\mathbf{W}^{\mathbf{H}} \qquad (6.8)$$

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

$$\mathbf{Y} = \mathbf{H} \cdot \mathbf{X} = (\mathbf{Wh}) \cdot (\mathbf{Wx}) \qquad (6.9)$$

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

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

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. **Solution: Solution:** Download the code from

```
wget https://github.com/jarpula−Bhanu/
    EE3900/blob/main/Ass/Codes/7.1.py
```

and execute it using

```
$ python3 7.1.py
```

7.2 Repeat all the exercises in the previous sections for the above a and b.
Solution: The filter frequency response is plotted at

```
wget https://github.com/jarpula−Bhanu/
    EE3900/blob/main/Ass/Codes/7.2.1.py
```

The impulse response function is plotted at

```
wget https://github.com/jarpula−Bhanu/
    EE3900/blob/main/Ass/Codes/7.2.2.py
```

We see that $h(n)$ is bounded and convergent.Also,since 1 is not a pole of the transfer function, the system is stable.
run the above codes using

```
        python3 7.2.1.py
        python3 7.2.2.py
```
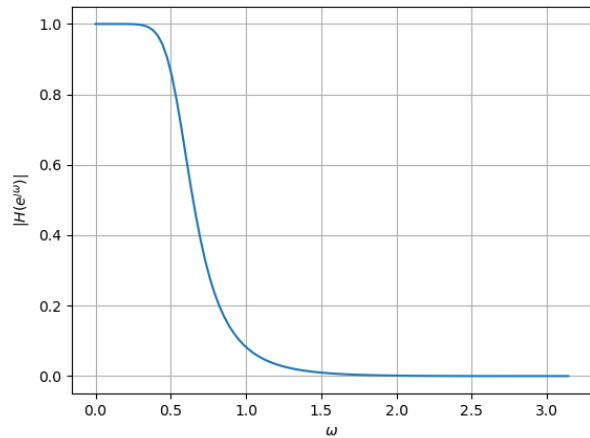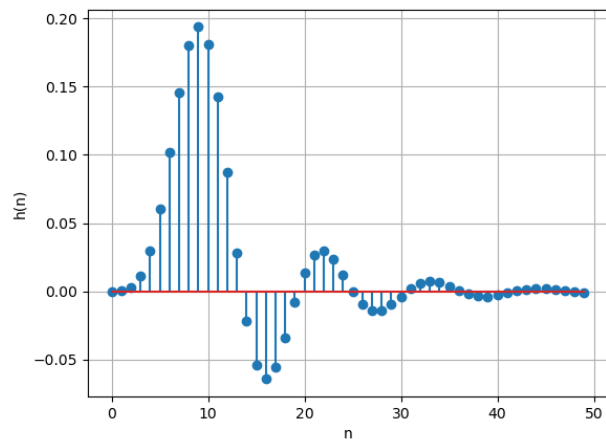

Fig. 7.2: Filter frequency response

Fig. 7.2: Plot of $h(n)$

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=4 and cutoff-frequency=4kHz.

7.5 Modifying the code with different input parameters and to get the best possible output.
Solution: A better filtering was found on setting te order of the filter to be 7