AUDIO FILTERING ASSIGNMENT

EE23BTECH11011- Batchu Ishitha*

I. DIGITAL FILTER

I1. The sound file used for this code can be obtained from the following link.

```
https://github.com/BATCHUISHITHA/EE−1205/blob/main/audio_filtering/codes/ishitha.wav
```

I2. Python code for removal of out of band noise:

```
import soundfile as sf
from scipy import signal

# read.wavfile
input_signal,fs=sf.read('ishitha.wav')

print("",fs)

#sampling frequency of input signal
sampl_freq=fs

#order of the filter
order=4

#cutoff frequency
cutoff_freq=1000.0


#digital frequency
Wn=2*cutoff_freq/sampl_freq

#b and a are numerator and denominator polynomials respectively
b,a=signal.butter(order,Wn,'low')
print("",a)
print("",b)
#filter the input signal with butterworth filter
output_signal=signal.filtfilt(b,a,input_signal, padlen=1)

#output_signal=signal.lfilt(b,a,input_signal)

#write the output signal into .wav file
sf.write('ishithareducednoise.wav', output_signal,fs)
```

I3. Analysis of sound file before and after removal of noise using spectrogram ie: https://academo.org/demos/spectrum-analyzer.
Solution: The darker areas are those where the frequencies have very low intensities, and the orange and the yellow areas represent the frequencies that have high intensities in sound.

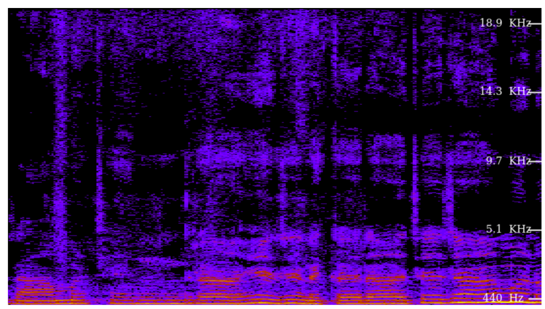

Fig. I.3. Spectrogram of the audio file before Filtering

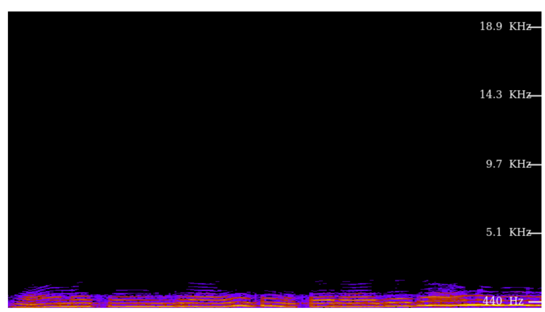

Fig. I.3. Spectrogram of the audio file after Filtering

II. DIFFERENCE EQUATION

II1. Let

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

Sketch $x(n)$.

II2. Let

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

Solution: C code for generating values of y(n):

```
https://github.com/BATCHUISHITHA/EE-1205/
    blob/main/audio_filtering/codes/2.2.c
```

Python code for plotting x(n) and y(n):

```
https://github.com/BATCHUISHITHA/EE-1205/
    blob/main/audio_filtering/codes/2.2.py
```

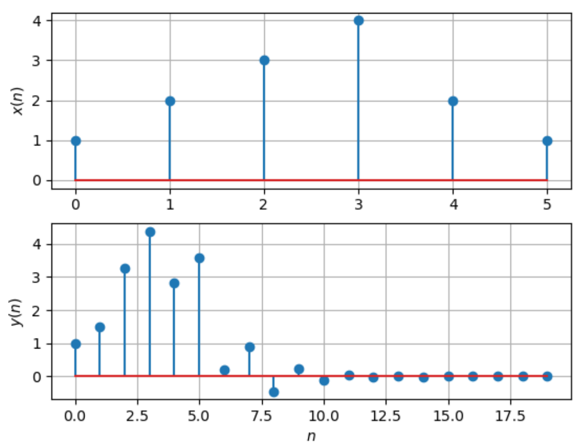

Fig. 2. Plot of $x(n)$ and $y(n)$

III. Z-Transform

III.1 The Z-transform of $x(n)$ is defined as

$$X(z) = \mathcal{Z}\{x(n)\} = \sum_{n=-\infty}^{\infty} x(n)z^{-n} \tag{3}$$

Show that

$$\mathcal{Z}\{x(n-1)\} = z^{-1}X(z) \tag{4}$$

and find

$$\mathcal{Z}\{x(n-k)\}. \tag{5}$$

Solution: Let

$$y(n) = x(n-k) \tag{6}$$

Taking z-transform

$$\mathcal{Z}(y(n)) = \mathcal{Z}(x(n-k)) \tag{7}$$

Simplifying LHS

$$Y(z) = \sum_{n=-\infty}^{\infty} y(n)z^{-n} \tag{8}$$

From (6)

$$Y(z) = \sum_{n=-\infty}^{\infty} x(n-k)z^{-n} \tag{9}$$

Let

$$n - k = s \tag{10}$$
$$\implies n = s + k \tag{11}$$

From (9) and (11)

$$Y(z) = \sum_{s=-\infty}^{\infty} x(s)z^{-(s+k)} \tag{12}$$
$$= z^{-k} \sum_{s=-\infty}^{\infty} x(s)z^{-s} \tag{13}$$

As variable in Z-transform is dummy, on replacing it, we get

$$Y(z) = z^{-k} \sum_{n=-\infty}^{\infty} x(n)z^{-n} \tag{14}$$
$$= z^{-k}X(z) \tag{15}$$

From (7) and (15)

$$\mathcal{Z}(x(n-k)) = z^{-k}X(z) \tag{16}$$

Put $k = 1$ resulting in (4)
Hence proved

III.2 Find

$$H(z) = \frac{Y(z)}{X(z)} \tag{17}$$

from (2) assuming that the Z-transform is a linear operation.
Solution: Applying Z-transform on both sides of (2)

$$Y(z) + \frac{1}{2}z^{-1}Y(z) = X(z) + z^{-2}X(z) \tag{18}$$
$$\implies H(z) = \frac{Y(z)}{X(z)} = \frac{1 + z^{-2}}{1 + \frac{1}{2}z^{-1}} \tag{19}$$

III.3 Find the Z transform of

$$\delta(n) = \begin{cases} 1 & n = 0 \\ 0 & \text{otherwise} \end{cases} \tag{20}$$

and show that the Z-transform of

$$u(n) = \begin{cases} 1 & n \geq 0 \\ 0 & \text{otherwise} \end{cases} \tag{21}$$

is

$$U(z) = \frac{1}{1 - z^{-1}}, \quad |z| > 1 \tag{22}$$

Solution:

$$Z\{\delta[n]\} = \sum_{n=-\infty}^{\infty} \delta[n] \cdot z^{-n} \tag{23}$$
$$= z^0 \tag{24}$$
$$= 1 \tag{25}$$

and from (21),

$$U(z) = \sum_{n=0}^{\infty} z^{-n} \tag{26}$$
$$= \frac{1}{1 - z^{-1}}, \quad |z| > 1 \tag{27}$$

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

III.4 Show that

$$a^n u(n) \overset{\mathcal{Z}}{\longleftrightarrow} \frac{1}{1 - az^{-1}} \quad |z| > |a| \tag{28}$$

Solution:

$$\mathcal{Z}\{a^n u(n)\} = \sum_{n=0}^{\infty} \left(az^{-1}\right)^n \tag{29}$$
$$= \frac{1}{1 - az^{-1}} \quad |z| > |a| \tag{30}$$

III.5 Let

$$H(e^{j\omega}) = H(z = e^{j\omega}). \tag{31}$$

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

Solution: Substituting $z = e^{j\omega}$ in (19),

$$H(e^{j\omega}) = \frac{1 + e^{-2j\omega}}{1 + \frac{1}{2}e^{-j\omega}} \tag{32}$$
$$\left|H(e^{j\omega})\right| = \left|\frac{1 + \cos 2\omega - j\sin 2\omega}{1 + \frac{1}{2}(\cos\omega - j\sin\omega)}\right| \tag{33}$$
$$= \sqrt{\frac{(1 + \cos 2\omega)^2 + (\sin 2\omega)^2}{\left(1 + \frac{1}{2}\cos\omega\right)^2 + \left(\frac{1}{2}\sin\omega\right)^2}} \tag{34}$$
$$= \frac{4\left|\cos\omega\right|}{\sqrt{5 + 4\cos\omega}} \tag{35}$$
$$\left|H(e^{j(\omega+2\pi)})\right| = \left|\frac{1 + e^{-2j(\omega+2\pi)}}{1 + \frac{1}{2}e^{-j(\omega+2\pi)}}\right| \tag{36}$$
$$= \frac{4\left|\cos\omega\right|}{\sqrt{5 + 4\cos\omega}} \tag{37}$$
$$= \left|H(e^{j\omega})\right| \tag{38}$$

Therefore, the fundamental period of $H(e^{j\omega})$ is 2π.

$\implies$ DTFT of a signal is always periodic.

The following code plots (III.5):

```
https://github.com/BATCHUISHITHA/EE
    −1205/blob/main/audio_filtering/codes
    /3.5.py
```

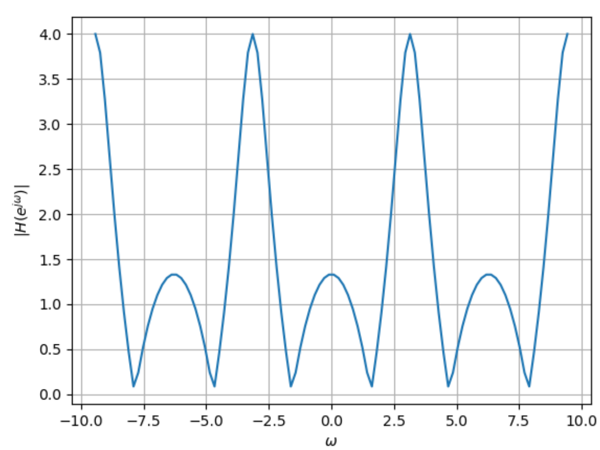

Fig. III.5. $\left|H\left(e^{j\omega}\right)\right|$

IV. Impulse Response

IV.1 Find an expression for $h(n)$ using $H(z)$, given that

$$h(n) \overset{\mathcal{Z}}{\longleftrightarrow} H(z) \tag{39}$$

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 (2).
Solution: From (19),

$$H(z) = \frac{1}{1 + \frac{1}{2}z^{-1}} + \frac{z^{-2}}{1 + \frac{1}{2}z^{-1}} \tag{40}$$

$$\implies h(n) = \left(-\frac{1}{2}\right)^n u(n) + \left(-\frac{1}{2}\right)^{n-2} u(n-2) \tag{41}$$

from (30) and (16).

IV.2 Sketch $h(n)$. Is it bounded?Convergent?
Solution: Yes, h(n) is convergent and bounded. The following code plots h(n) vs n.

```
https://github.com/BATCHUISHITHA/EE
    −1205/blob/main/audio_filtering/codes
    /4.2.py
```

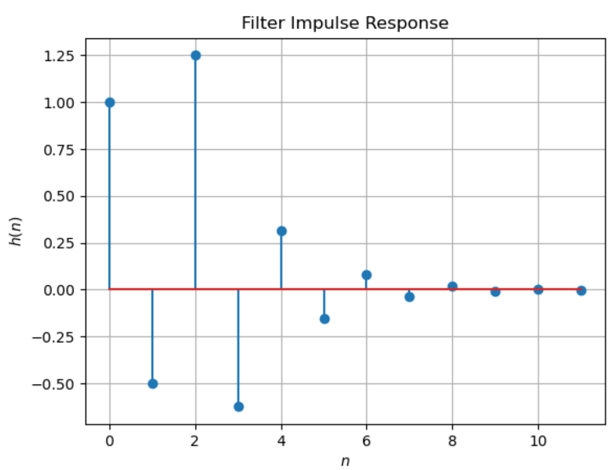

Fig. IV.2. $h(n)$ as the inverse of $H(z)$

IV.3 The system with h(n) is defined to be stable if

$$\sum_{n=-\infty}^{\infty} h(n) < \infty \tag{42}$$

Is the system defined by (2) stable for impulse response in (39)?
Solution: For stable system (42) must be converging.
For n $\to \infty$,

$$u(n) = u(n-2) = 1 \tag{43}$$

$$\implies h(n) = \left(-\frac{1}{2}\right)^n + \left(-\frac{1}{2}\right)^{n-2} \tag{44}$$

Since, both terms of h(n) tends to 0 as n$\to \infty$, h(n)$\to$ 0.
$\implies$ output remains bounded for bounded inputs,ie: $h(n)$ is stable.

IV.4 Compute and sketch h(n) using

$$h(n) + \frac{1}{2}h(n-1) = \delta(n) + \delta(n-2) \tag{45}$$

This is the definition of h(n).
Solution: The following code plots (IV.4) . Note that this is same as (IV.2).

```
https://github.com/BATCHUISHITHA/EE
    −1205/blob/main/audio_filtering/codes
    /4.4.py
```

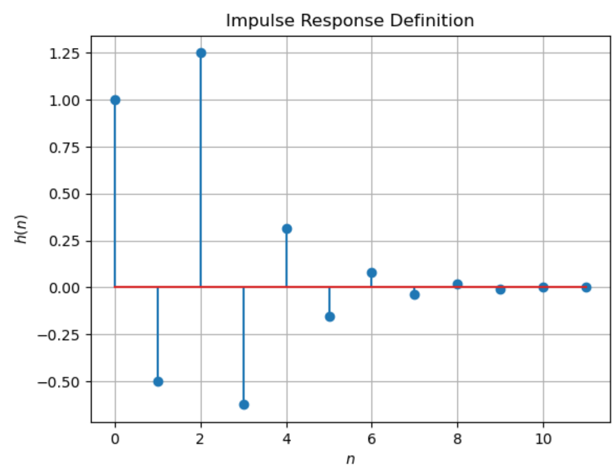

Fig. IV.4. $h(n)$ from the definition

IV.5 Compute

$$y(n) = x(n) * h(n) = \sum_{k=-\infty}^{\infty} x(k)h(n-k) \tag{46}$$

Comment. The operation in (46) is known as *convolution*.
Solution: The following code plots Fig. IV.5. Note that this is the same as $y(n)$ in Fig:2.

```
https://github.com/BATCHUISHITHA/EE
    −1205/blob/main/audio_filtering/codes
    /4.5.py
```

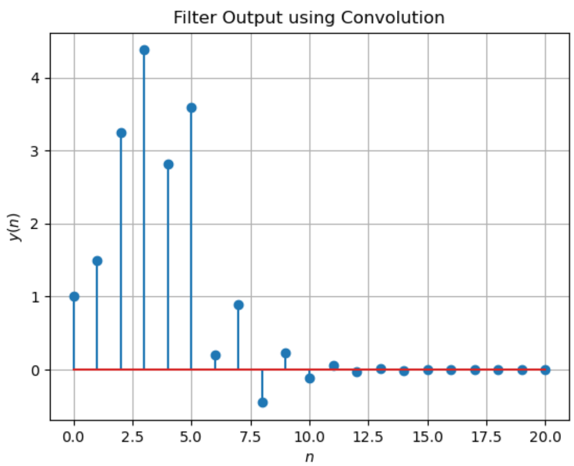

Fig. IV.5. $y(n)$ from the definition of convolution

IV.6 Show that

$$y(n) = \sum_{k=-\infty}^{\infty} x(n-k)h(k) \tag{47}$$

Solution: In (46), replacing k by $n-m$

$$y(n) = \sum_{n-m=-\infty}^{\infty} x(n-m)h(n-(n-m)) \tag{48}$$

$$= \sum_{n-m=-\infty}^{\infty} x(n-m)h(m) \tag{49}$$

Since, the variable in summation is dummy, replacing m by k,

$$= \sum_{k=-\infty}^{\infty} x(n-k)h(k) \tag{50}$$

V. DFT AND FFT

V.1 Compute

$$X(k) \triangleq \sum_{n=0}^{N-1} x(n)e^{\frac{-2\pi jkn}{N}}, \quad k = 0,1,\ldots,N-1 \tag{51}$$

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

V.2 Compute

$$Y(k) = X(k)H(k) \tag{52}$$

V.3 Compute

$$y(n) = \frac{1}{N}\sum_{k=0}^{N-1} Y(k)\cdot e^{\frac{-2\pi jkn}{N}}, \quad n = 0,1,\ldots,N-1 \tag{53}$$

Solution: The above three questions are solved using the code below.

```
https://github.com/BATCHUISHITHA/EE
    -1205/blob/main/audio_filtering/codes/5.
    py
```

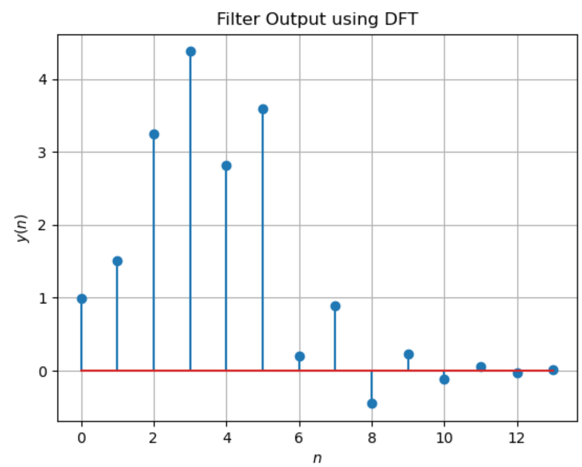

Fig. V.3. $y(n)$ from the DFT

V.4 Repeat the previous exercise by computing $X(k), H(k)$ and $y(n)$ through FFT and IFFT.
Solution: This code verifies the result by plotting the result obtained from DFT,IDFT and the result obtained from FFT,IFFT.

```
https://github.com/BATCHUISHITHA/EE
    -1205/blob/main/audio_filtering/codes
    /5.4.py
```

V.5 Wherever possible, express all the above equations as matrix equations.
Solution: The DFT matrix is defined as :

$$\mathbf{W} = \begin{pmatrix} \omega^0 & \omega^0 & \ldots & \omega^0 \\ \omega^0 & \omega^1 & \ldots & \omega^{N-1} \\ \vdots & \vdots & \ddots & \vdots \\ \omega^0 & \omega^{N-1} & \ldots & \omega^{(N-1)(N-1)} \end{pmatrix} \tag{54}$$

where $\omega = e^{-\frac{j2\pi}{N}}$. Now any DFT equation can be written as

$$\mathbf{X} = \mathbf{W}\mathbf{x} \tag{55}$$

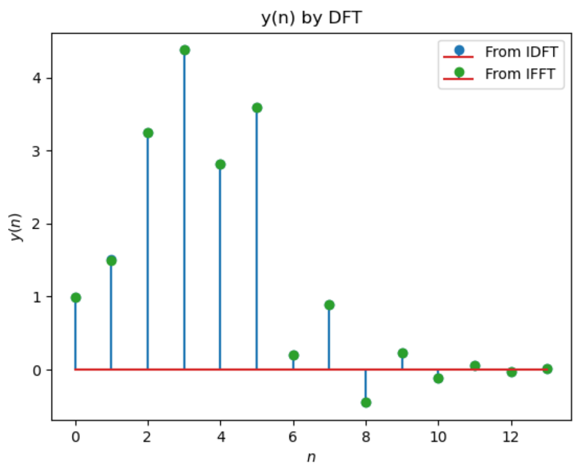

Fig. V.4. $y(n)$ from the DFT, IDFT and from the FFT,IFFT are plotted and verified

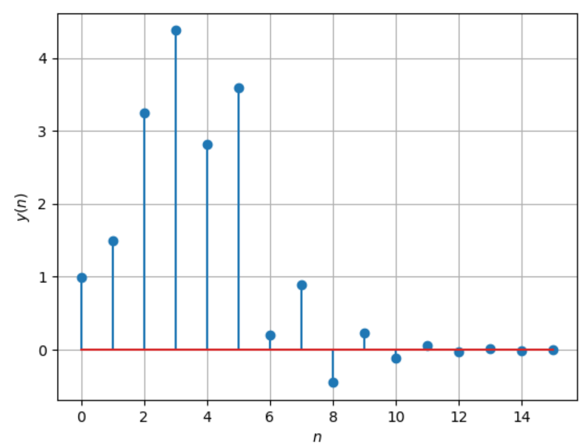

Fig. V.5. $y(n)$ obtained from DFT matrix

where $\mathbf{x}$ is the original signal and $\mathbf{X}$ is the frequency-domain representation.

$$\mathbf{x} = \begin{pmatrix} x(0) \\ x(1) \\ \vdots \\ x(n-1) \end{pmatrix} \tag{56}$$

$$\mathbf{X} = \begin{pmatrix} X(0) \\ X(1) \\ \vdots \\ X(n-1) \end{pmatrix} \tag{57}$$

Thus we can rewrite (52) as:

$$\mathbf{Y} = \mathbf{X} \odot \mathbf{H} = (\mathbf{W}\mathbf{x}) \odot (\mathbf{W}\mathbf{h}) \tag{58}$$

where the $\odot$ represents the Hadamard product which performs element-wise multiplication. This is specifically called "SCHUR PRODUCT" when defined for matrices.

```
https://github.com/BATCHUISHITHA/EE
    −1205/blob/main/audio_filtering/codes
    /5.5.py
```

VI. EXERCISES

Answer the following questions by looking at the python code in Problem:(I.2)

VI.1 The command

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

in Problem:(I.2) is executed through the following difference equation

$$\sum_{m=0}^{M} a(m)\, y(n-m) = \sum_{k=0}^{N} b(k)\, x(n-k) \tag{59}$$

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: The below is the code for output of an audio signal with and without using inbuilt function signal.lfilter

```
https://github.com/BATCHUISHITHA/EE
    −1205/blob/main/audio_filtering/codes
    /6.1.py
```

VI.2 Repeat all the exercises in the previous sections for the above a and b.
Solution: The code in I.2 generates the values of a and b which can be used to generate a difference equation.
And,

$$M = 5 \tag{60}$$
$$N = 5 \tag{61}$$

From 59

$$a(0)\, y(n) + a(1)\, y(n-1) + a(2)\, y(n-2) + a(3) \tag{62}$$
$$y(n-3) + a(4)\, y(n-4) = b(0)\, x(n) + b(1)\, x(n-1)$$
$$+ b(2)\, x(n-2) + b(3)\, x(n-3) + b(4)\, x(n-4)$$

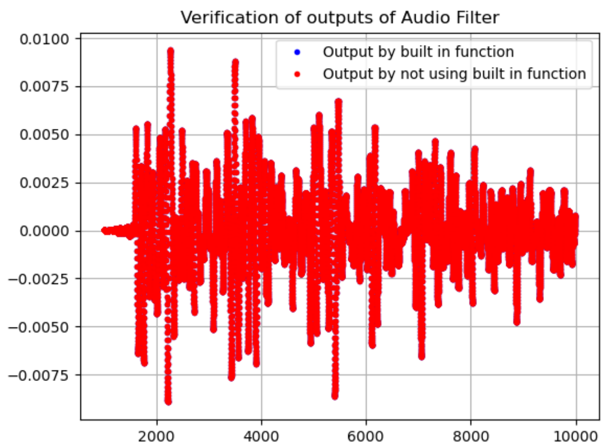

Fig. VI.1. output of an audio signal with and without inbuilt function signal.lfilter are plotted and verified

Difference Equation is given by :

$$y(n) - (3.63)\,y(n-1) + (4.95)\,y(n-2) - (3.01)\,y(n-3) + (0.69)\,y(n-4) = \left(2.15\times10^{-5}\right)x(n) + \left(8.60\times10^{-5}\right)x(n-1) + \left(1.29\times10^{-4}\right)x(n-2) + \left(8.60\times10^{-5}\right)x(n-3) + \left(2.15\times10^{-5}\right)x(n-4) \tag{63}$$

From (59)

$$H(z) = \frac{b_0 + b_1z^{-1} + b_2z^{-2} + \ldots + b_Mz^{-N}}{a_0 + a_1z^{-1} + a_2z^{-2} + \ldots + a_Nz^{-M}} \tag{64}$$

$$H(z) = \frac{\sum_{k=0}^{N} b(k)z^{-k}}{\sum_{k=0}^{M} a(k)z^{-k}} \tag{65}$$

Partial fraction on (65) can be generalised as:

$$H(z) = \sum_i \frac{r(i)}{1 - p(i)z^{-1}} + \sum_j k(j)z^{-j} \tag{66}$$

Now,

$$a^n u(n) \overset{\mathcal{Z}}{\longleftrightarrow} \frac{1}{1 - az^{-1}} \tag{67}$$

$$\delta(n-k) \overset{\mathcal{Z}}{\longleftrightarrow} z^{-k} \tag{68}$$

Taking inverse z transform of (66) by using (67) and (68)

$$h(n) = \sum_i r(i)[p(i)]^n u(n) + \sum_j k(j)\delta(n-j) \tag{69}$$

The below code computes the values of $r(i), p(i), k(i)$ and plots $h(n)$

https://github.com/BATCHUISHITHA/EE
−1205/blob/main/audio_filtering/codes
/6.2.py

$r(i)$	$p(i)$	$k(i)$
$0.06558697 - 0.15997359j$	0.87507075 +0.0480371j	3.1240145×10^{-5}
$0.06558697 + 0.15997359j$	0.87507075 -0.0480371j	–
$-0.06559183 + 0.02744514j$	0.93885135+0.12442455j	–
$-0.06559183 - 0.02744514j$	0.93885135-0.12442455j	–

TABLE 1
VALUES OF $r(i), p(i), k(i)$

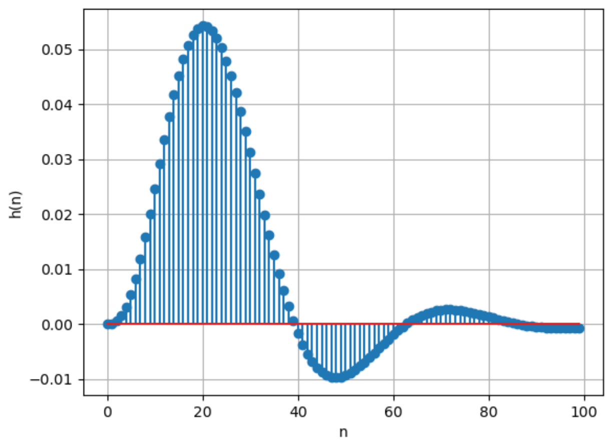

Fig. VI.2. h(n) of Audio Filter

Stability of h(n):
According to (42)

$$H(z) = \sum_{n=0}^{\infty} h(n)\,z^{-n} \tag{70}$$

$$H(1) = \sum_{n=0}^{\infty} h(n) = \frac{\sum_{k=0}^{N} b(k)}{\sum_{k=0}^{M} a(k)} < \infty \tag{71}$$

As both $a(k)$ and $b(k)$ are finite length sequences they converge.
The below code plots Filter frequency response

https://github.com/BATCHUISHITHA/EE
−1205/blob/main/audio_filtering/codes
/6.2.1.py

The below code plots the Butterworth Filter in analog domain by using bilinear transform.

$$z = \frac{1 + sT/2}{1 - sT/2} \tag{72}$$

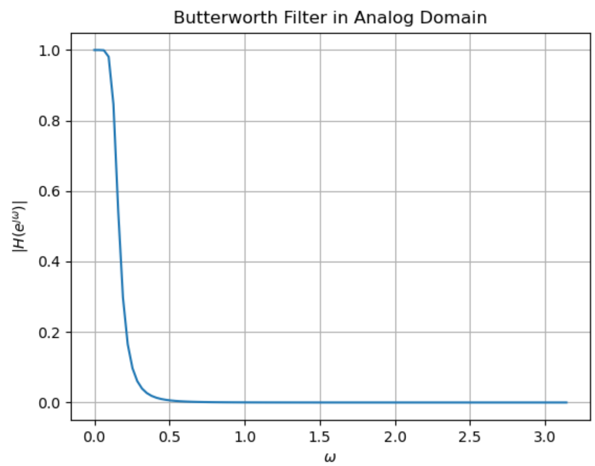

Fig. VI.2. Frequency Response of Audio Filter

https://github.com/BATCHUISHITHA/EE−1205/blob/main/audio_filtering/codes/6.2.2.py

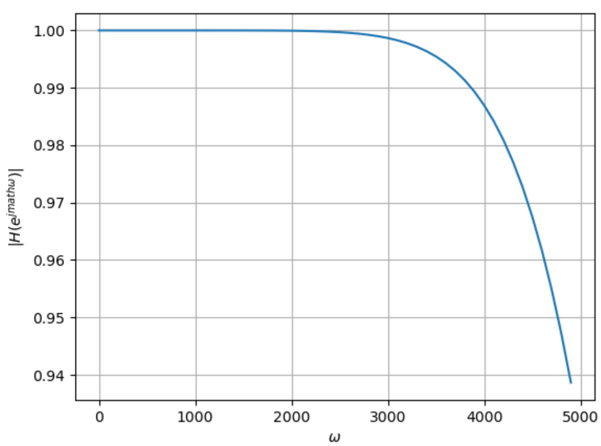

Fig. VI.2. Butterworth Filter Frequency response in analog domain

The below code plots the Pole-Zero Plot of the frequency response.

https://github.com/BATCHUISHITHA/EE−1205/blob/main/audio_filtering/codes/6.2.3.py

VI.3 What is the sampling frequency of the input signal?
Solution: The sampling frequency of the input signal is 44.1kHz.

VI.4 What is type, order and cutoff-frequency of the above butterworth filter

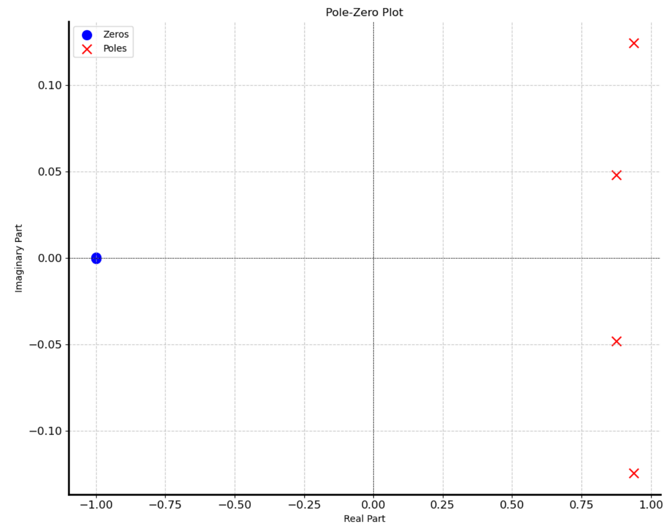

Fig. VI.2. There are complex poles. So $h(n)$ should be damped sinusoid.

Solution: The given butterworth filter is low-pass with order=4 and cutoff-frequency=1kHz.

VI.5 Modifying the code with different input parameters and to get the best possible output.
Solution: A better filtering was found on setting the order of the filter to be 5.