Audio Filter

EE23BTECH11046 - Poluri Hemanth*

I. SOFTWARE INSTALLATION

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
```

II. Digital Filter

2) The audio file with noise to be filtered can be downloaded from

```
$https://github.com/hemanth23ee/
    ASSIGNMENTS/blob/main/Audio%20
    filter%20assignment/codes/Song.wav
```

3) A Python Code is written to achieve Audio Filtering

```
import soundfile as sf
import numpy as np
from scipy import signal
#read .wav file
input_signal,fs = sf.read('Song.wav')

#sampling frequency of Input signal
sampl_freq=fs


#order of the filter
order=4

#cutoff frquency
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.lfilter(b, a,
    input_signal)

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

4) The frequencies of audio with noise are observed with help of spectrogram from https://academo.org/demos/spectrum-analyzer.

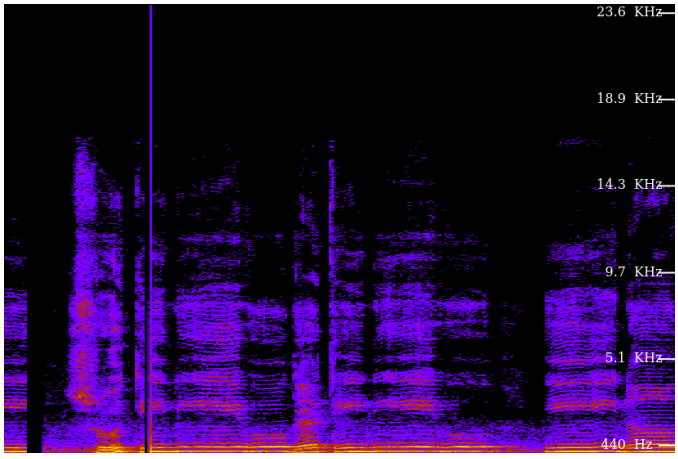

Fig. 1. Input audio to be filtered

The above figure depicts the spectogram of the audio with noise.
The frequencies which have very low intensities are represented as darker areas,and the frequencies that have high intensities in the sound are represented as orange, yellow areas.

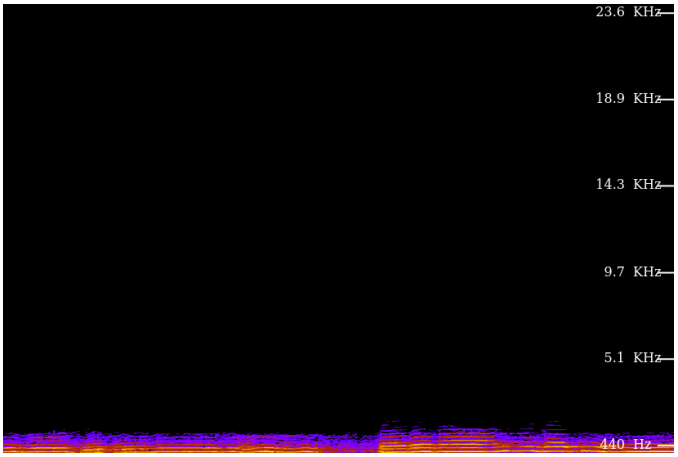

Fig. 2. Filtered audio

he above figure depicts the spectogram of the filtered audio.

III. Difference Equation

1) Let

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

Sketch $x(n)$.

2) Let

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

Sketch $y(n)$.

Solve

Solution: The values of $y(n)$ are generated using C code from

https://github.com/hemanth23ee/ASSIGNMENTS/blob/main/Audio%20filter%20assignment/codes/2_2.c

The following python codes are used to plot (1) and (2)

https://github.com/hemanth23ee/ASSIGNMENTS/blob/main/Audio%20filter%20assignment/codes/2_2.py \\
https://github.com/hemanth23ee/ASSIGNMENTS/blob/main/Audio%20filter%20assignment/codes/2_2.1.py

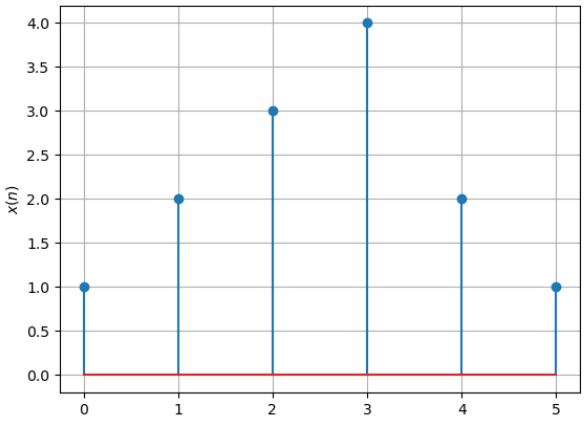

Fig. 3. Plot of $x(n)$ and $y(n)$

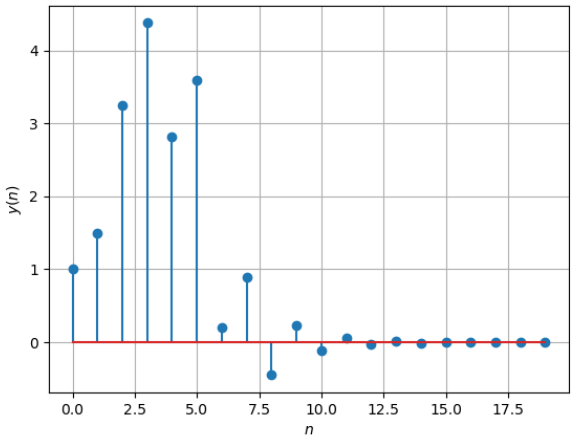

Fig. 4. Plot of $x(n)$ and $y(n)$

IV. Z-Transform

IV.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: From (3),

$$\mathcal{Z}\{x(n-k)\} = \sum_{n=-\infty}^{\infty} x(n-1)z^{-n} \tag{6}$$

$$= \sum_{n=-\infty}^{\infty} x(n)z^{-n-1} = z^{-1}\sum_{n=-\infty}^{\infty} x(n)z^{-n} \tag{7}$$

resulting in (4). Similarly, it can be shown that

$$\mathcal{Z}\{x(n-k)\} = z^{-k}X(z) \tag{8}$$

IV.2 Find

$$H(z) = \frac{Y(z)}{X(z)} \tag{9}$$

from (2) assuming that the Z-transform is a linear operation.

Solution: Applying (8) in (2),

$$Y(z) + \frac{1}{2}z^{-1}Y(z) = X(z) + z^{-2}X(z) \tag{10}$$

$$\implies \frac{Y(z)}{X(z)} = \frac{1+z^{-2}}{1+\frac{1}{2}z^{-1}} \tag{11}$$

IV.3 Find the Z transform of

$$\delta(n) = \begin{cases} 1 & n = 0 \\ 0 & \text{otherwise} \end{cases} \tag{12}$$

and show that the Z-transform of

$$u(n) = \begin{cases} 1 & n \geq 0 \\ 0 & \text{otherwise} \end{cases} \tag{13}$$

is

$$U(z) = \frac{1}{1-z^{-1}}, \quad |z| > 1 \tag{14}$$

Solution: It is easy to show that

$$\delta(n) \overset{\mathcal{Z}}{\longleftrightarrow} 1 \tag{15}$$

and from (13),

$$U(z) = \sum_{n=0}^{\infty} z^{-n} \tag{16}$$

$$= \frac{1}{1-z^{-1}}, \quad |z| > 1 \tag{17}$$

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

IV.4 Show that

$$a^n u(n) \overset{\mathcal{Z}}{\longleftrightarrow} \frac{1}{1-az^{-1}} \quad |z| > |a| \tag{18}$$

Solution:

$$a^n u(n) \overset{\mathcal{Z}}{\longleftrightarrow} \sum_{n=0}^{\infty} \left(az^{-1}\right)^n \tag{19}$$

$$= \frac{1}{1-az^{-1}} \quad |z| > |a| \tag{20}$$

IV.5 Let

$$H\left(e^{\mathrm{J}\omega}\right) = H\left(z = e^{\mathrm{J}\omega}\right). \tag{21}$$

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 python code is used to plot the magnitude of transfer function.

https://github.com/hemanth23ee/ASSIGNMENTS/blob/main/Audio%20filter%20assignment/codes/3.5.py

Substituting $z = e^{j\omega}$ in (11), we get

$$\left|H\left(e^{j\omega}\right)\right| = \left|\frac{1+e^{-2j\omega}}{1+\frac{1}{2}e^{-j\omega}}\right| \tag{22}$$

$$= \sqrt{\frac{(1+\cos 2\omega)^2 + (\sin 2\omega)^2}{\left(1+\frac{1}{2}\cos\omega\right)^2 + \left(\frac{1}{2}\sin\omega\right)^2}} \tag{23}$$

$$= \frac{4|\cos\omega|}{\sqrt{5+4\cos\omega}} \tag{24}$$

$$\left|H\left(e^{j(\omega+2\pi)}\right)\right| = \frac{4|\cos(\omega+2\pi)|}{\sqrt{5+4\cos(\omega+2\pi)}} \tag{25}$$

$$= \frac{4|\cos\omega|}{\sqrt{5+4\cos\omega}} \tag{26}$$

$$= \left|H\left(e^{j\omega}\right)\right| \tag{27}$$

Therefore its fundamental period is 2π , which verifies that DTFT of a signal is always periodic.

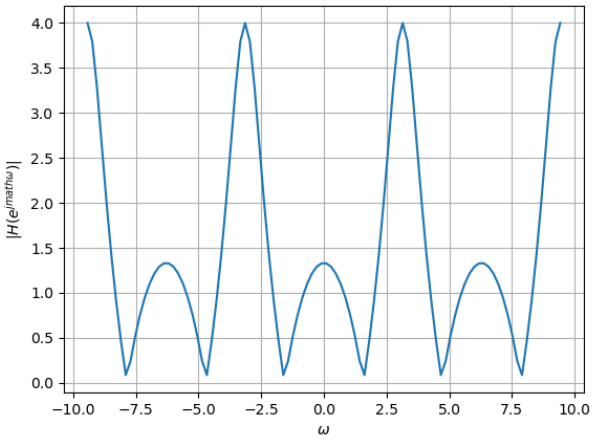

Fig. 5. $\left|H\left(e^{j\omega}\right)\right|$

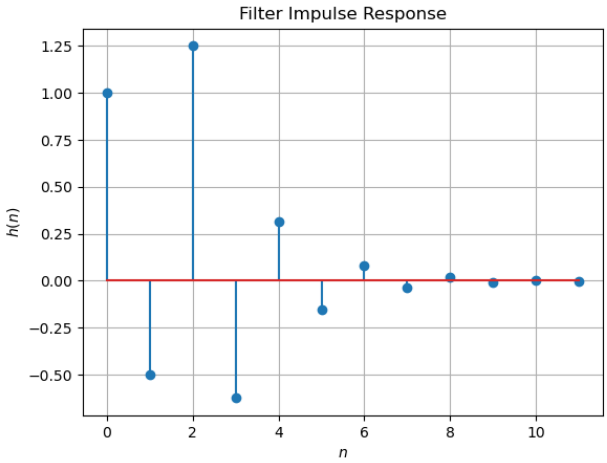

Fig. 6. $h(n)$ as the inverse of $H(z)$

V. Impulse Response

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

$$h(n) \xleftrightarrow{\mathcal{Z}} H(z) \tag{28}$$

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 (11),

$$H(z) = \frac{1}{1+\frac{1}{2}z^{-1}} + \frac{z^{-2}}{1+\frac{1}{2}z^{-1}} \tag{29}$$

$$\implies h(n) = \left(-\frac{1}{2}\right)^n u(n) + \left(-\frac{1}{2}\right)^{n-2} u(n-2) \tag{30}$$

using (18) and (8).
2) Sketch $h(n)$. Is it bounded? Convergent?
Solution: The python code used to plot $h(n)$ can be obtained from

https://github.com/hemanth23ee/ASSIGNMENTS/blob/main/Audio%20filter%20assignment/codes/4.2.py

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

$$\sum_{n=-\infty}^{\infty} h(n) < \infty \tag{31}$$

Is the system defined by (2) stable for the impulse response in (28)?
Solution: For stable system (31) should converge.
By using ratio test

$$\lim_{n\to\infty} \left|\frac{h(n+1)}{h(n)}\right| < 1 \tag{32}$$

(33)

For large n

$$u(n) = u(n-2) = 1 \tag{34}$$

$$\lim_{n\to\infty} \left(\frac{h(n+1)}{h(n)}\right) = 1/2 < 1 \tag{35}$$

Therefore it converges. Hence it is stable.
4) Compute and sketch $h(n)$ using

$$h(n) + \frac{1}{2}h(n-1) = \delta(n) + \delta(n-2), \tag{36}$$

This is the definition of $h(n)$.
Solution:
Definition of $h(n)$: The output of the system when $\delta(n)$ is given as input.

The following code plots Fig. 7. Note that this is the same as Fig. 6.

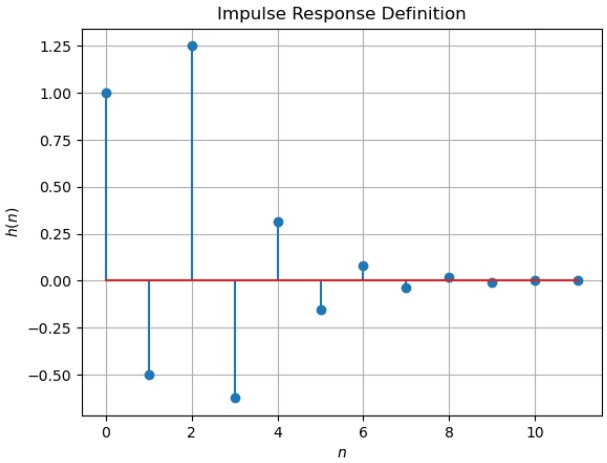

Fig. 7. $h(n)$ from the definition is same as Fig. **??**

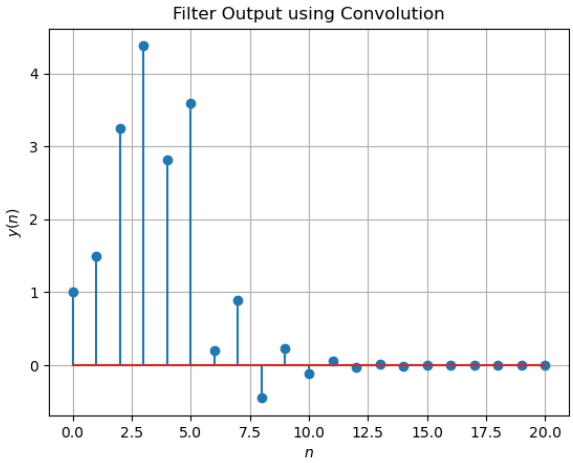

Fig. 8. $y(n)$ from the definition of convolution

https://github.com/hemanth23ee/ASSIGNMENTS/blob/main/Audio%20filter%20assignment/codes/4.2.py

5) Compute

$$y(n) = x(n) * h(n) = \sum_{n=-\infty}^{\infty} x(k)h(n-k) \quad (37)$$

Comment. The operation in (37) is known as *convolution*.

Solution: The python code used to plot Fig. 8 can be obtained from link below. Note that this is the same as $y(n)$ in Fig. 4.

https://github.com/hemanth23ee/ASSIGNMENTS/blob/main/Audio%20filter%20assignment/codes/4.5.py

6) Show that

$$y(n) = \sum_{n=-\infty}^{\infty} x(n-k)h(k) \quad (38)$$

Solution: In (37), we substitute $k = n - k$ to get

$$y(n) = \sum_{k=-\infty}^{\infty} x(k)\,h(n-k) \quad (39)$$

$$= \sum_{n-k=-\infty}^{\infty} x(n-k)\,h(k) \quad (40)$$

$$= \sum_{k=-\infty}^{\infty} x(n-k)\,h(k) \quad (41)$$

VI. DFT and FFT

1) Compute

$$X(k) \stackrel{\triangle}{=} \sum_{n=0}^{N-1} x(n)e^{-\mathrm{j}2\pi kn/N}, \quad k = 0, 1, \ldots, N-1 \tag{42}$$

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

2) Compute

$$Y(k) = X(k)H(k) \tag{43}$$

3) Compute

$$y\,(n) = \frac{1}{N}\sum_{k=0}^{N-1} Y\,(k)\cdot e^{\mathrm{j}2\pi kn/N}, \quad n = 0, 1, \ldots, N-1 \tag{44}$$

Solution: The above three questions are solved using the code from

https://github.com/hemanth23ee/ASSIGNMENTS/blob/main/Audio%20filter%20assignment/codes/5_sol.py

4) Repeat the previous exercise by computing $X(k), H(k)$ and $y(n)$ through FFT and IFFT.
Solution: The solution of this question can be found in the code from

https://github.com/hemanth23ee/ASSIGNMENTS/blob/main/Audio%20filter%20assignment/codes/5.4.py

This code verifies the result by plotting the obtained result with the result obtained by IDFT.

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

$$\mathbf{W} = \begin{pmatrix} \omega^0 & \omega^0 & \ldots & \omega^0 \\ \omega^0 & \omega^1 & \ldots & \omega^{N-1} \\ \vdots & \vdots & \ddots & \vdots \\ \omega^0 & \omega^{N-1} & \ldots & \omega^{(N-1)(N-1)} \end{pmatrix} \tag{45}$$

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

$$\mathbf{X} = \mathbf{W}\mathbf{x} \tag{46}$$

where

$$\mathbf{x} = \begin{pmatrix} x(0) \\ x(1) \\ \vdots \\ x(n-1) \end{pmatrix} \tag{47}$$

$$\mathbf{X} = \begin{pmatrix} X(0) \\ X(1) \\ \vdots \\ X(n-1) \end{pmatrix} \tag{48}$$

Thus we can rewrite (43) as:

$$\mathbf{Y} = \mathbf{X} \odot \mathbf{H} = (\mathbf{W}\mathbf{x}) \odot (\mathbf{W}\mathbf{h}) \tag{49}$$

where the $\odot$ represents the Hadamard product which performs element-wise multiplication.

The below code computes $y\,(n)$ by DFT Matrix and then plots it.

https://github.com/hemanth23ee/ASSIGNMENTS/blob/main/Audio%20filter%20assignment/codes/5.5.py

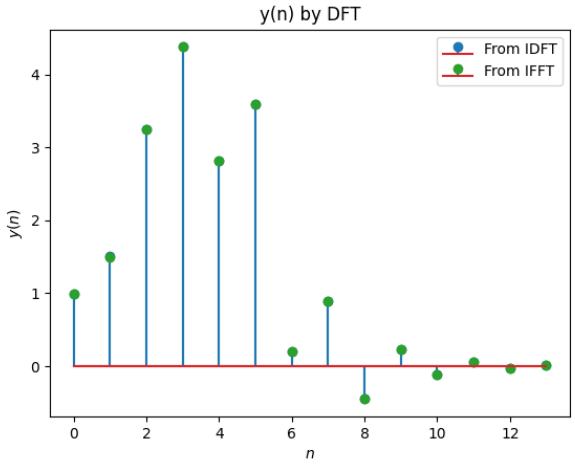

Fig. 9. $y(n)$ obtained from IDFT and IFFT is plotted and verified

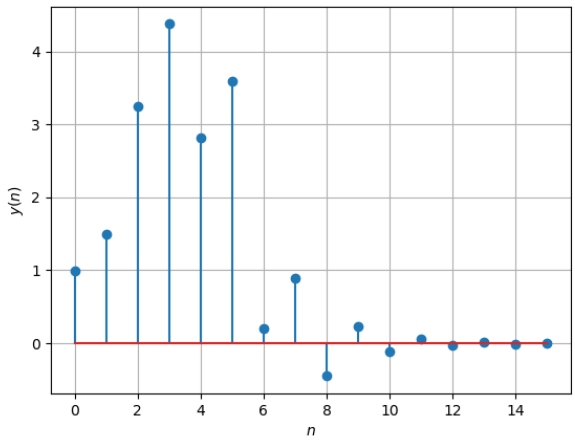

Fig. 10. $y(n)$ obtained from DFT Matrix

VII. EXERCISES

Answer the following questions by looking at the python code in Problem 3.

1) The command

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

in Problem 3 is executed through the following difference equation

$$\sum_{m=0}^{M} a(m)\, y(n-m) = \sum_{k=0}^{N} b(k)\, x(n-k) \quad (50)$$

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 code gives the output of an Audio Filter without using the built in function signal.lfilter.

https://github.com/hemanth23ee/ASSIGNMENTS/blob/main/Audio%20filter%20assignment/codes/6.1.py

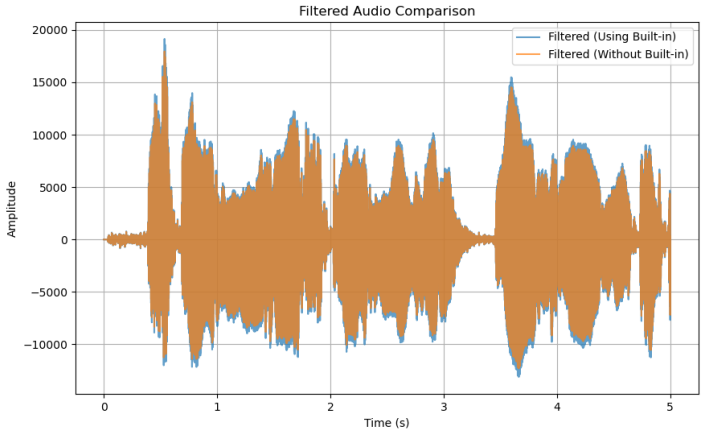

Fig. 11. Both the outputs using and without using function overlap

2) Repeat all the exercises in the previous sections for the above a and b.
Solution: The code in 3 generates the values of a and b which can be used to generate a difference equation.
And,

$$M = 5 \quad (51)$$

$$N = 5 \quad (52)$$

From 50

$$a(0)\, y(n) + a(1)\, y(n-1) + a(2)\, y(n-2) + a(3) \quad (53)$$

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

Difference Equation is given by :

$$y(n) - (3.63)\, y(n-1) + (4.95)\, y(n-2) - (3.012)\, y(n-3) + (0.689)\, y(n-4) = \left(2.15 \times 10^{-5}\right) x(n) + \left(8.61 \times 10^{-5}\right) x(n-1) + \left(1.291 \times 10^{-5}\right) x(n-2) + \left(8.608 \times 10^{-5}\right) x(n-3) + \left(2.152 \times 10^{-5}\right) x(n-4) \quad (54)$$

From (50)

$$H(z) = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2} + \ldots + b_M z^{-N}}{a_0 + a_1 z^{-1} + a_2 z^{-2} + \ldots + a_N z^{-M}} \quad (55)$$

$$H(z) = \frac{\sum_{k=0}^{N} b(k) z^{-k}}{\sum_{k=0}^{M} a(k) z^{-k}} \quad (56)$$

Partial fraction on (56) can be generalised as:

$$H(z) = \sum_i \frac{r(i)}{1 - p(i) z^{-1}} + \sum_j k(j) z^{-j} \quad (57)$$

Now,

$$a^n u(n) \xleftrightarrow{z} \frac{1}{1 - a z^{-1}} \quad (58)$$

$$\delta(n-k) \xleftrightarrow{z} z^{-k} \quad (59)$$

Taking inverse z transform of (57) by using (58) and (59)

$$h(n) = \sum_i r(i) [p(i)]^n u(n) + \sum_j k(j) \delta(n-j) \quad (60)$$

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

https://github.com/hemanth23ee/ASSIGNMENTS/blob/main/Audio%20filter%20assignment/codes/6.2.py

$r(i)$	$p(i)$	$k(i)$
$0.06558697 - 0.15997359j$	0.87507075+0.0480371j	3.124×10^{-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 $r(i), p(i), k(i)$

Stability of h(n):
According to (31)

$$H(z) = \sum_{n=0}^{\infty} h(n) z^{-n} \tag{61}$$

$$H(1) = \sum_{n=0}^{\infty} h(n) = \frac{\sum_{k=0}^{N} b(k)}{\sum_{k=0}^{M} a(k)} < \infty \tag{62}$$

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

https://github.com/hemanth23ee/ASSIGNMENTS/blob/main/Audio%20filter%20assignment/codes/6_filter_response.py

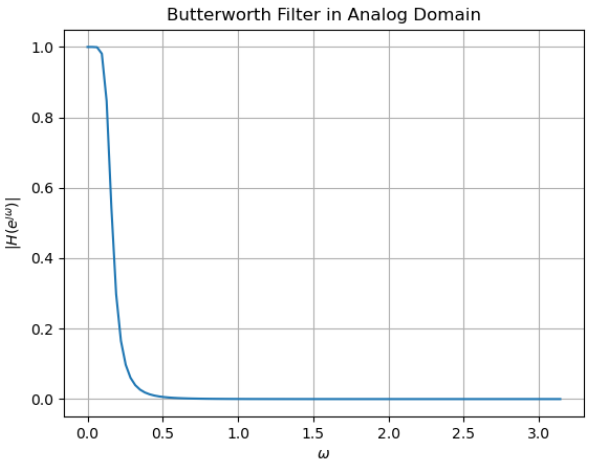

Fig. 12. Frequency Response of Audio Filter

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

$$z = \frac{1 + sT/2}{1 - sT/2} \tag{63}$$

https://github.com/hemanth23ee/ASSIGNMENTS/blob/main/Audio%20filter%20assignment/codes/analog_filt.py

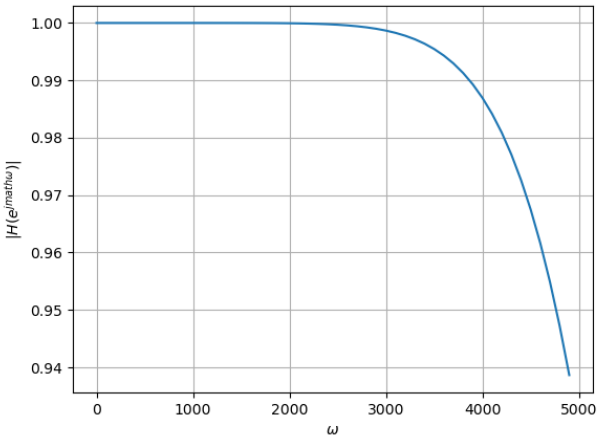

Fig. 13. Butterworth Filter Frequency response in analog domain

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

https://github.com/hemanth23ee/ASSIGNMENTS/blob/main/Audio%20filter%20assignment/codes/6.2_pole-zero.py

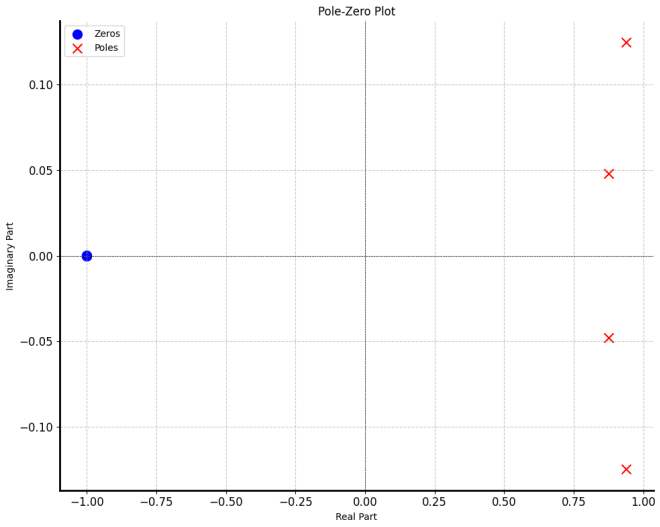

Fig. 14. There are complex poles. So $h(n)$ should be damped sinusoid.

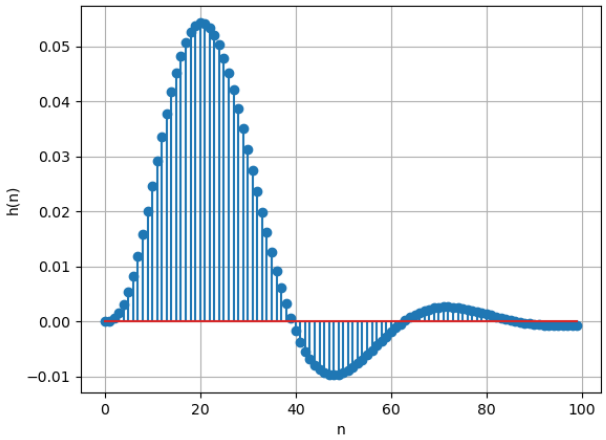

Fig. 15. h(n) of Audio Filter.It is a damped sinusoid.

3) What is the sampling frequency of the input signal?
   Solution: The Sampling Frequency is 44.1KHz
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=1kHz.
5) Modify the code with different input parameters and get the best possible output.
   Solution: A better filtering was found on setting the order of the filter to be 5.