Audio Filter

EE23BTECH11010 - Venkatesh Bandawar*

I. Digital Filter

I.1 Download the sound file from

```
https://github.com/venkatesh11010/Audio-filtering-11010/blob/main/audio%20filter/Audio-files/Venkatesh-singing.wav
```

I.2 Below is the Python Code to perform the Audio Filtering:

```
import soundfile as sf
from scipy import signal

# Read .wav file
input_signal, fs = sf.read('Venkatesh-singing.wav')

# Order of the filter
order = 3

# Cutoff frequency 4kHz
cutoff_freq = 3000.0

# Digital frequency
Wn = 2 * cutoff_freq / fs

# b and a are numerator and denominator polynomials, respectively
b, a = signal.butter(order, Wn, 'low')

output_signal = signal.lfilter(b,a,input_signal)
# Write the output signal into a .wav file
sf.write('filteredsong18.wav', output_signal, fs)
```

I.3 The audio file is analyzed using spectrogram using the online platform https://academo.org/demos/spectrum-analyzer.
The orange and yellow areas represent frequencies that have high intensities in the sound. Also, the signal is blank for frequencies above 5.1 kHz.

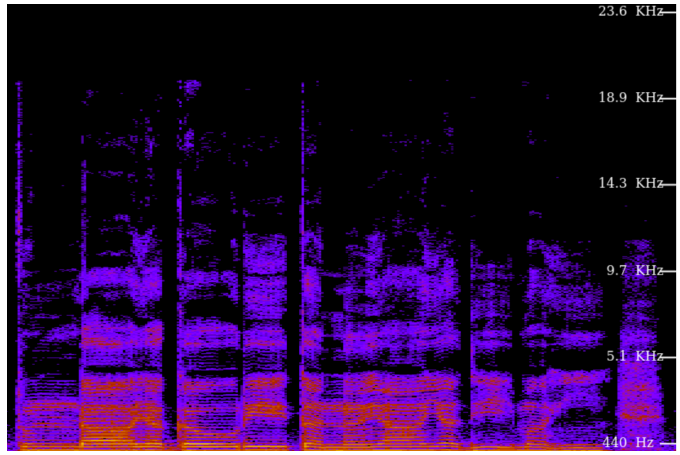

Fig. 1: Spectrogram of Input Audio Signal

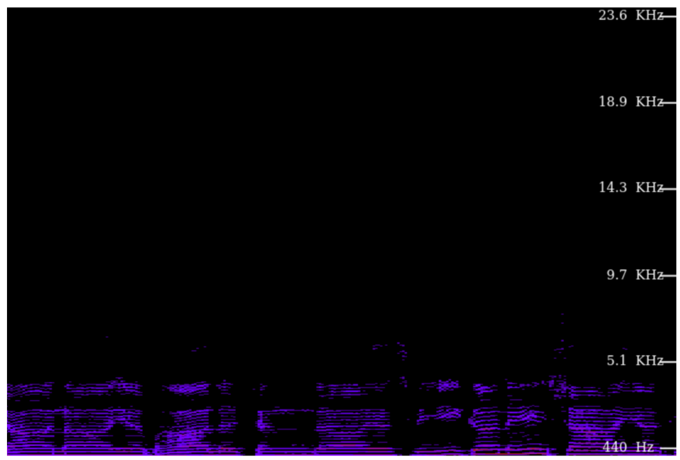

Fig. 2: Spectrogram of Filtered Output Audio Signal

II. Difference Equation

II.1 Let

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

Sketch $x(n)$.

II.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 C code calculates $y(n)$ and Python plots the graph.

https://github.com/venkatesh11010/Audio−filtering−11010/blob/main/audio%20filter/codes/x_n−y_n.c

Below are the plots of the $x(n)$ and $y(n)$:

https://github.com/venkatesh11010/Audio−filtering−11010/blob/main/audio%20filter/codes/x_n−y_n.py

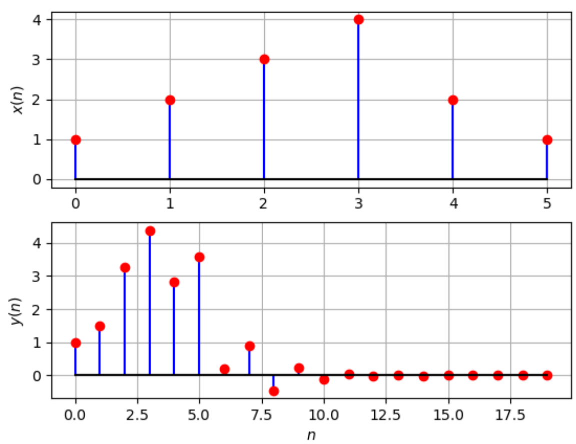

Fig. 3: Plot of $x(n)$ and $y(n)$

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: 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

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

III.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),

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

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

III.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.

III.4 Show that

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

Solution:

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

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

III.5 Let

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

Plot $\left|H\left(e^{j\omega}\right)\right|$. Comment. $H(e^{j\omega})$ is known as the *Discrete Time Fourier Transform* (DTFT) of $x(n)$.
Solution: Below is the code which plots the magnitude of Transfer Function:

```
https://github.com/venkatesh11010/Audio−filtering−11010/blob/main/audio%20filter/codes/H(z).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, the fundamental period is 2π , which implies that DTFT of a signal is always periodic.

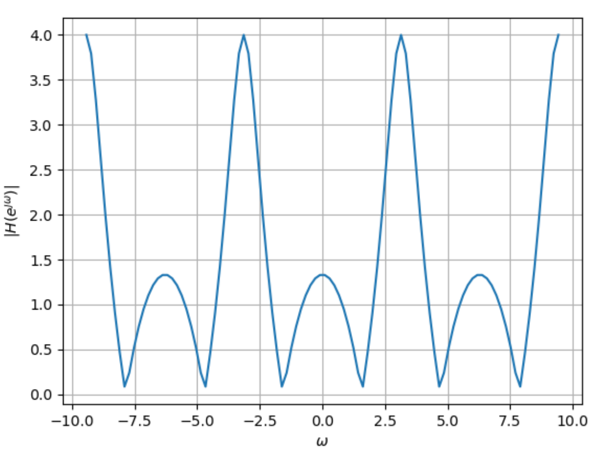

Fig. 4: $\left|H\left(e^{j\omega}\right)\right|$ vs ω

IV. Impulse Response

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

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

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).

IV.2 Sketch $h(n)$. Is it bounded? Convergent?
Solution: The following code plots $h(n)$

```
https://github.com/venkatesh11010/Audio−filtering−11010/blob/main/audio%20filter/codes/h(n).py
```

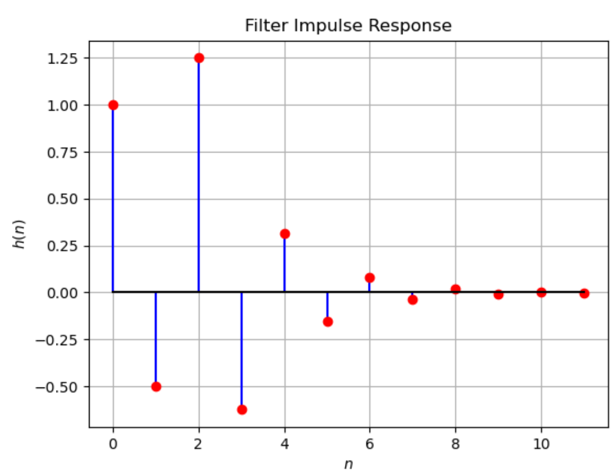

Fig. 5: $h(n)$ vs n

IV.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 for convergence:

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

$$\tag{33}$$

For large n

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

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

Hence it is stable.

IV.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. 6. Note that this is the same as Fig. 5.

```
https://github.com/venkatesh11010/Audio-filtering-11010/blob/main/audio%20filter/codes/h(n)def.py
```

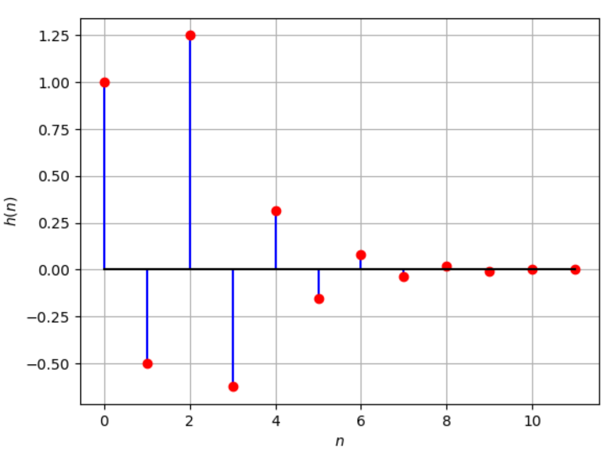

Fig. 6: $h(n)$ vs n using definition

IV.5 Compute

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

Comment. The operation in (37) is known as *convolution*.
Solution: Below code plots Fig. 7. Note that this is the same as $y(n)$ in Fig. 3.

```
https://github.com/venkatesh11010/Audio-filtering-11010/blob/main/audio%20filter/codes/y(n)byconv.py
```

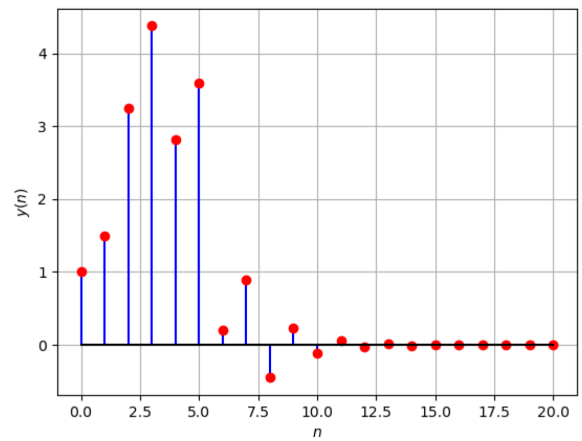

Fig. 7: $y(n)$ from the definition of convolution

IV.6 Show that

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

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

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

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

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

V. DFT and FFT

V.1 Compute

$$X(k)\triangleq\sum_{n=0}^{N-1} x(n)e^{-\jmath 2\pi kn/N}, \quad k=0,1,\dots,N-1 \tag{42}$$

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

V.2 Compute

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

V.3 Compute

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

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

```
https://github.com/venkatesh11010/Audio-filtering-11010/blob/main/audio%20filter/codes/5sol.py
```

V.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 below.

```
https://github.com/venkatesh11010/Audio-filtering-11010/blob/main/audio%20filter/codes/y(n)_verify.py
```

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

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

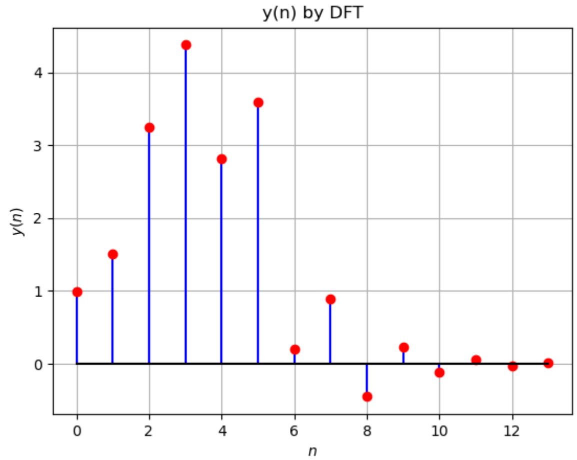

Fig. 8: $y(n)$ obtained from DFT

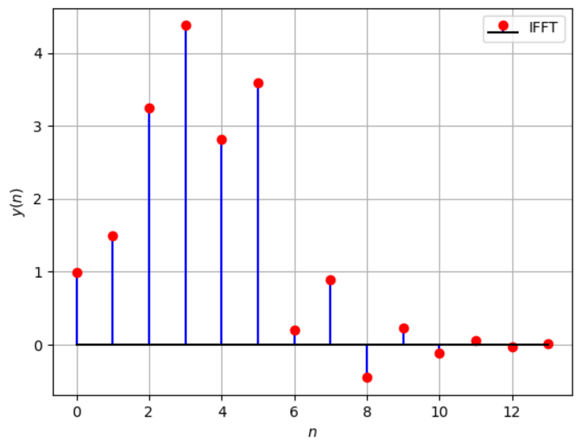

Fig. 9: $y(n)$ obtained from IFFT

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:

$$\mathbf{Y} = \mathbf{X} \cdot \mathbf{H} = (\mathbf{Wx}) \cdot (\mathbf{Wh}) \tag{49}$$

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

```
https://github.com/venkatesh11010/Audio−filtering
    −11010/blob/main/audio%20filter/codes/matrix
    .py
```

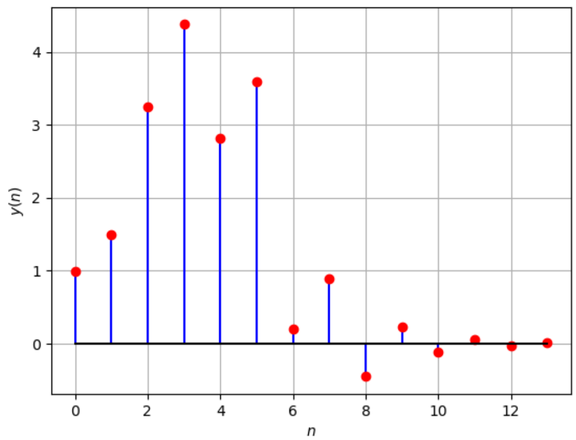

Fig. 10: $y(n)$ from DFT Matrix

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

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

where the input signal is $x(n)$ and the output signal is $y(n)$ with initial values all 0. Replace **signal. lfilter** 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/venkatesh11010/Audio−
    filtering−11010/blob/main/audio%20filter
    /codes/lfilter.py
```

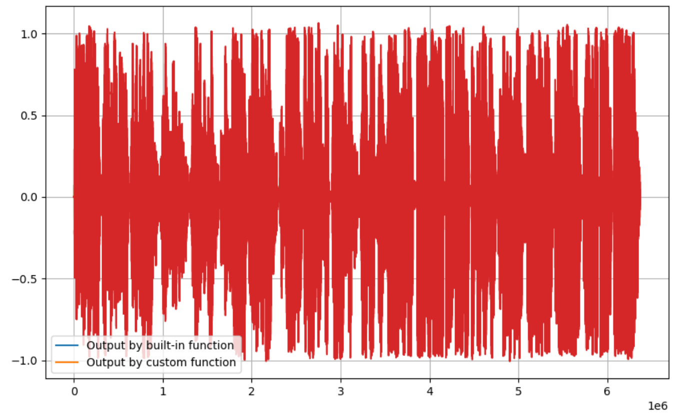

Fig. 11: Both the outputs using and without using function overlap

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

$$a = \begin{bmatrix} 1 & -2.21916862 & 1.71511783 & -0.45354593 \end{bmatrix}$$
$$b = \begin{bmatrix} 0.00530041 & 0.01590123 & 0.01590123 & 0.0053004 \end{bmatrix}$$

$$M = 3 \tag{51}$$
$$N = 3 \tag{52}$$

From 50

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

$$y(n) - 2.219y(n-1) + 1.715y(n-2) - 0.453y(n-3) = 0.005x(n) + 0.016x(n-1) + 0.016x(n-2) + 0.005x(n-3)$$

From (50)

$$H(z) = \frac{b(0) + b(1)z^{-1} + b(2)z^{-2} + \ldots + b(N)z^{-N}}{a(0) + a(1)z^{-1} + a(2)z^{-2} + \ldots + a(M)z^{-M}} \tag{53}$$

$$H(z) = \frac{\sum_{k=0}^{N} b(k)z^{-k}}{\sum_{k=0}^{M} a(k)z^{-k}} \tag{54}$$

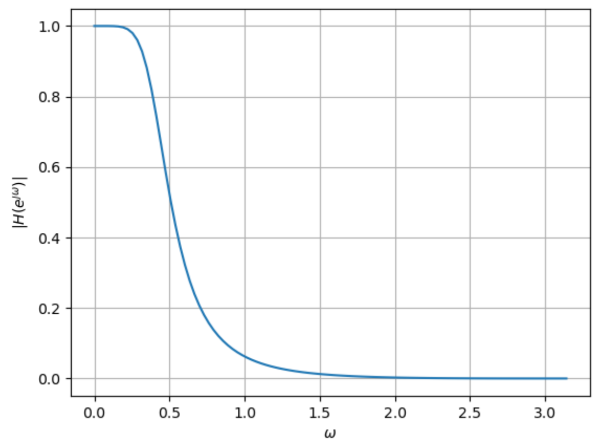

Fig. 12: $|H(e^{j\omega})|$

$$h(n) = 0.005\delta(n) + 0.028\delta(n-1)+ \\ 0.068\delta(n-2) + 0.112\delta(n-3)\cdots\cdots \tag{55}$$

Stability of h(n):

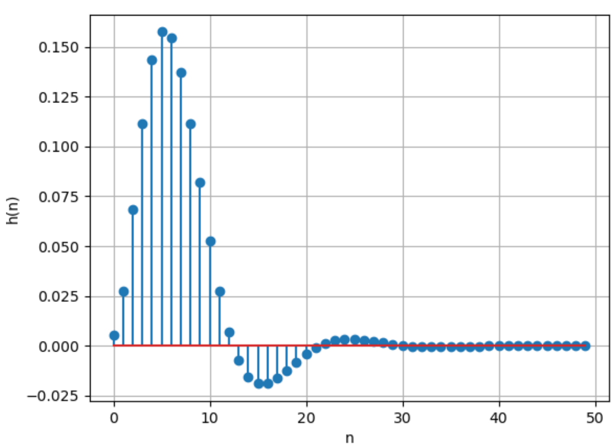

Fig. 13: $h(n)$

According to (31)

$$H(z) = \sum_{n=0}^{\infty} h(n) z^{-n} \tag{56}$$

$$H(1) = \sum_{n=0}^{\infty} h(n) = \frac{\sum_{k=0}^{N} b(k)}{\sum_{k=0}^{M} a(k)} < \infty \tag{57}$$

As both $a(k)$ and $b(k)$ are finite length sequences they converge.

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

VI.4 What is type, order and cutoff-frequency of the above butterworth filter
Solution: The given butterworth filter is low-pass with order = 3 and cutoff-frequency = $3kHz$.

VI.5 Modify the code with different input parameters and get the best possible output. **Solution:** A better filtering was found when order of the filter is 4.