Filters

Prajwal M EE23BTECH11217*

Contents

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

1 Software Installation

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://raw.githubusercontent.com/
    mnepraj/DSP/master/filter/codes/sister.
    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 5.5 KHz. These represent the synthesizer key tones. Also, the key strokes are audible along with background noise. By observing spectrogram, it clearly shows that tonal frequency is under 5.5kHz. And above 5.5kHz only noise is present.

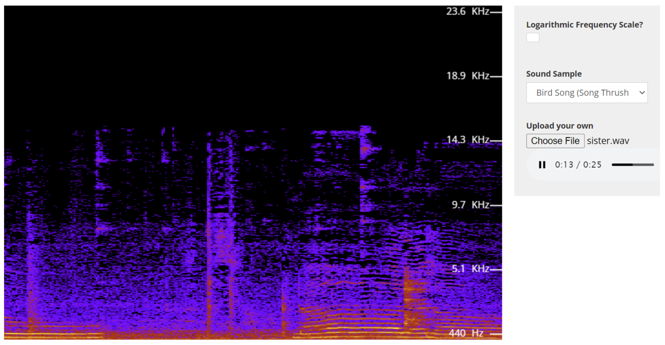

Fig. 2.2: spectrum of sister.wav

2.3 Write the python code for removal of out of band noise and execute the code.
Solution:

```
import soundfile as sf
from scipy import signal

# Parameters
input_file = "sister.wav"
output_file = "sisterlfilter.wav"
order = 4
cutoff_freq = 5500.0

# Read input file
input_signal, sampl_freq = sf.read(
    input_file)

# Calculate cutoff frequency
Wn = 2 * cutoff_freq / sampl_freq

# Design butterworth filter
b, a = signal.butter(order, Wn, "low")

# Filter the input signal
```

*The author is with the Department of Electrical Engineering, Indian Institute of Technology, Hyderabad 502285 India e-mail: gadepall@iith.ac.in. All content in the manuscript is released under GNU GPL. Free to use for anything.

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

# Write the output file
sf.write(output_file, output_signal,
    sampl_freq)

print(f"The output file '{output_file}' has
    been successfully written!")
```

2.4 The output of the python script in Problem 2.3 is the audio file sisterlfilter.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 5.5 kHz.

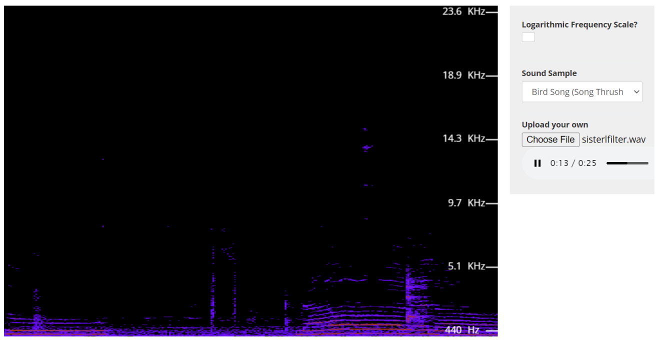

Fig. 2.4

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.

```
wget https://github.com/mnepraj/DSP/raw/
    master/filter/codes/xnyn.py
```

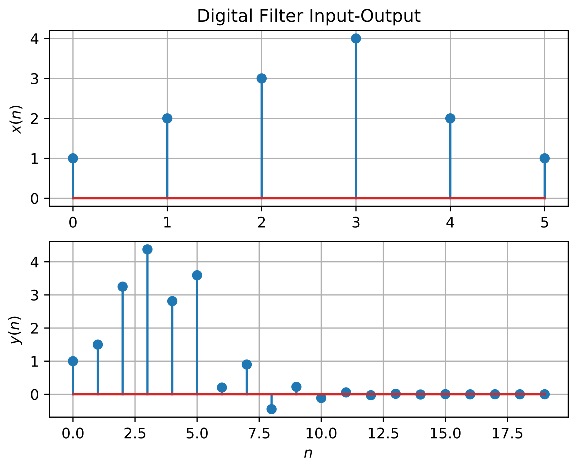

Fig. 3.2

4 Z-transform

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

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

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),

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

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

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

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

4.2 Find

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

from (3.2) assuming that the Z-transform is a linear operation.

Solution: Applying (4.6) in (3.2),

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

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

4.3 Find the Z transform of

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

and show that the Z-transform of

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

is

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

Solution: It is easy to show that

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

and from (4.11),

$$U(z) = \sum_{n=0}^{\infty} z^{-n} \tag{4.14}$$

$$= \frac{1}{1 - z^{-1}}, \quad |z| > 1 \tag{4.15}$$

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

4.4 Show that

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

Solution:

$$x(n) = a^n u(n) \tag{4.17}$$

$$x(n) \overset{\mathcal{Z}}{\rightleftharpoons} X(z) \tag{4.18}$$

using (4.1),

$$X(z) = \sum_{n=-\infty}^{\infty} (a^n u(n)) z^{-n} \tag{4.19}$$

$$= \sum_{n=0}^{\infty} (a^{-1}z)^{-n} \tag{4.20}$$

using (4.15),

$$= \frac{1}{1 - az^{-1}} \quad |z| > |a| \tag{4.21}$$

4.5 Let

$$H\left(e^{j\omega}\right) = H\left(z = e^{j\omega}\right). \tag{4.22}$$

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

Solution:

Substituting $z = e^{j\omega}$ in (4.9), we get

$$\left|H\left(e^{j\omega}\right)\right| = \left|\frac{1 + e^{-2j\omega}}{1 + \frac{1}{2}e^{-j\omega}}\right| \tag{4.23}$$

$$= \sqrt{\frac{(1 + \cos 2\omega)^2 + (\sin 2\omega)^2}{\left(1 + \frac{1}{2}\cos\omega\right)^2 + \left(\frac{1}{2}\sin\omega\right)^2}} \tag{4.24}$$

$$= \frac{4|\cos\omega|}{\sqrt{5 + 4\cos\omega}} \tag{4.25}$$

$$\left|H\left(e^{j(\omega+2\pi)}\right)\right| = \frac{4|\cos(\omega + 2\pi)|}{\sqrt{5 + 4\cos(\omega + 2\pi)}} \tag{4.26}$$

$$= \frac{4|\cos\omega|}{\sqrt{5 + 4\cos\omega}} \tag{4.27}$$

$$= \left|H\left(e^{j\omega}\right)\right| \tag{4.28}$$

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

The following code plots Fig. 4.5.

```
wget https://raw.githubusercontent.com/
    mnepraj/DSP/master/filter/codes/dtft.py
```

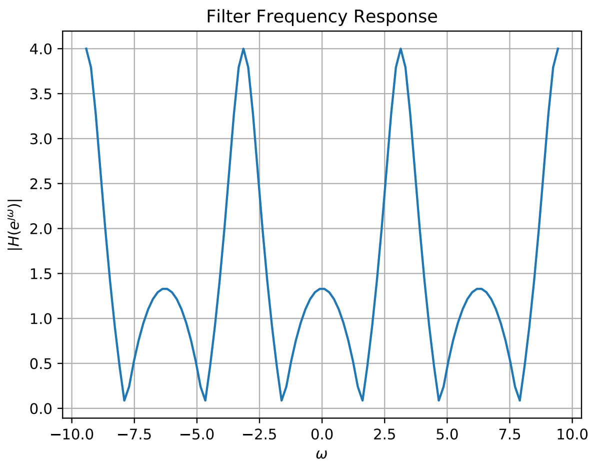

Fig. 4.5: $\left|H\left(e^{j\omega}\right)\right|$

5 Impulse Response

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

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

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),

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

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

using (4.16) and (4.6).

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

```
wget https://raw.githubusercontent.com/
    mnepraj/DSP/master/filter/codes/hn.py
```

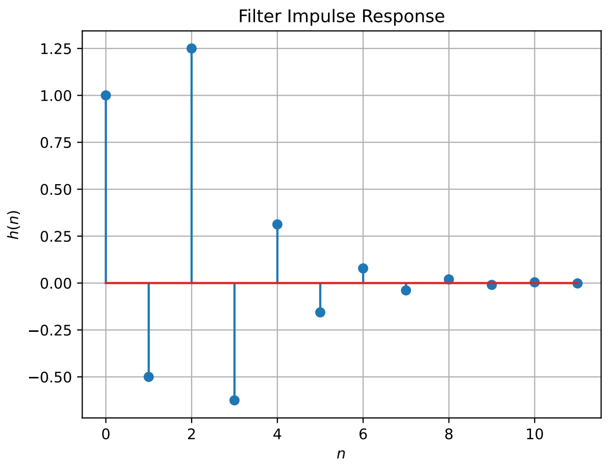

Fig. 5.2: $h(n)$ as the inverse of $H(z)$

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:

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

$$= \sum_{n=0}^{\infty} \left(-\frac{1}{2}\right)^{n} + \sum_{n=2}^{\infty} \left(-\frac{1}{2}\right)^{n-2} \tag{5.6}$$

$$= \frac{2}{3} + \frac{2}{3} = \frac{4}{3} < \infty \tag{5.7}$$

Hence, the system is stable.

5.4 Compute and sketch $h(n)$ using

$$h(n) + \frac{1}{2}h(n-1) = \delta(n) + \delta(n-2), \tag{5.8}$$

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://raw.githubusercontent.com/
    mnepraj/DSP/master/filter/codes/hndef.py
```

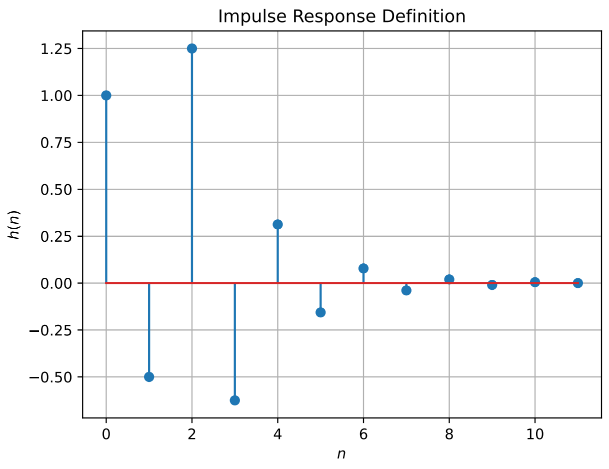

Fig. 5.4: $h(n)$ from the definition

5.5 Compute

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

Comment. The operation in (5.9) 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://raw.githubusercontent.com/
    mnepraj/DSP/master/filter/codes/ynconv.
    py
```

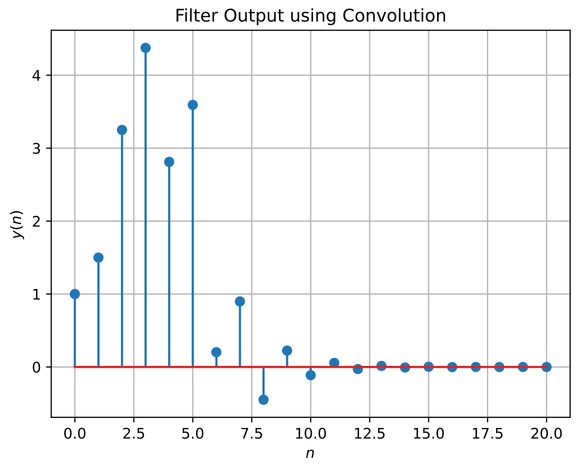

Fig. 5.5: $y(n)$ from the definition of convolution

5.6 Show that

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

Solution: In (5.9), we substitute $k = n - k$ to get

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

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

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

6 DFT and FFT

6.1 Compute

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

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

6.2 Compute

$$Y(k) = X(k)H(k) \tag{6.2}$$

6.3 Compute

$$y(n) = \frac{1}{N} \sum_{k=0}^{N-1} Y(k) \cdot e^{j2\pi kn/N}, \quad n = 0, 1, \ldots, N-1 \tag{6.3}$$

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

```
wget https://raw.githubusercontent.com/
    mnepraj/DSP/master/filter/codes/DFT.py
```

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

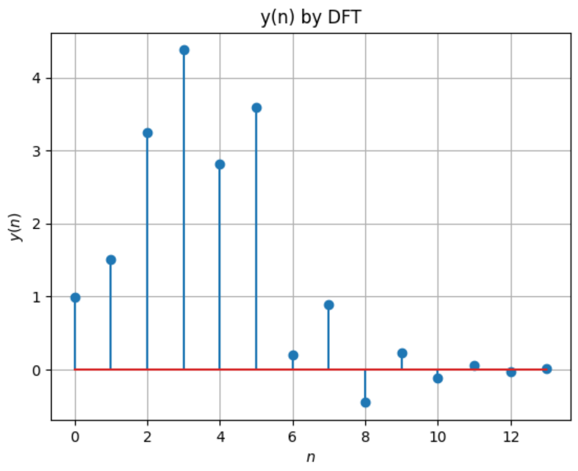

Fig. 6.3: $y(n)$ obtained from IDFT

6.4 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.

```
wget https://raw.githubusercontent.com/
    mnepraj/DSP/master/filter/codes/FFT.py
```

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

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

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

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

$$\mathbf{X} = \mathbf{W}\mathbf{x} \tag{6.5}$$

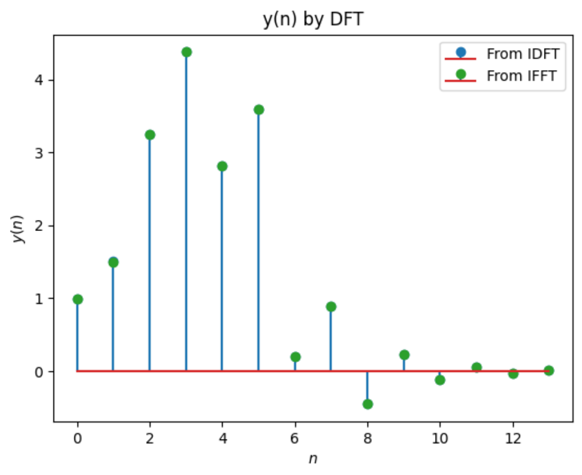

Fig. 6.4: $y(n)$ obtained from IDFT and IFFT is plotted and verified

where

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

$$\mathbf{X} = \begin{pmatrix} X(0) \\ X(1) \\ \vdots \\ X(n-1) \end{pmatrix} \tag{6.7}$$

Thus we can rewrite (6.2) as:

$$\mathbf{Y} = \mathbf{X} \odot \mathbf{H} = (\mathbf{W}\mathbf{x}) \odot (\mathbf{W}\mathbf{h}) \tag{6.8}$$

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.

```
wget https://raw.githubusercontent.com/
    mnepraj/DSP/master/filter/codes/
    DFTmatrix.py
```

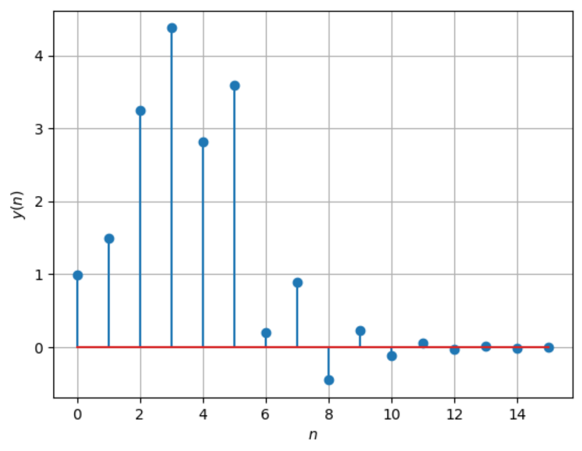

Fig. 6.5: $y(n)$ obtained from DFT Matrix

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

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

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: The following code replaces signal.filtfilt with my routine 'mneprajfilter'.

```
wget https://raw.githubusercontent.com/
    mnepraj/DSP/master/filter/codes/
    filtfiltreplaced.py
```

```
import numpy as np
import soundfile as sf
from scipy import signal
from mneprajfilter import mneprajfilter #
    Importing the mneprajfilter function

# Read .wav file
input_signal, fs = sf.read('sister.wav')

# Extract number of channels
num_channels = input_signal.shape[1] if len
    (input_signal.shape) > 1 else 1

# Sampling frequency of Input signal
sampl_freq = fs

# Order of the filter
order = 4
```

```
# Cutoff frequency 6kHz
cutoff_freq = 5500.0

# Digital frequency
Wn = 2 * cutoff_freq / sampl_freq

# Get butterworth filter coefficients
b, a = signal.butter(order, Wn, 'low')

# Write coefficients to a text file
with open('filter_coefficients.txt', 'w') as file
    :
    file.write("b␣coefficients:\n")
    np.savetxt(file, b, delimiter=',') # Write b
        coefficients
    file.write("\n\na␣coefficients:\n")
    np.savetxt(file, a, delimiter=',') # Write a
        coefficients

# Filter the input signal with butterworth
    filter
output_signal = mneprajfilter(a, b,
    input_signal, sampl_freq)

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

7.2 Repeat all the exercises in the previous sections for the above a and b.
Solution: Let sister.wav be the x(n) sampled at 44.1kHz. The code plots x(n).

```
wget https://raw.githubusercontent.com/
    mnepraj/DSP/master/filter/codes/
    xnexercise.py
```

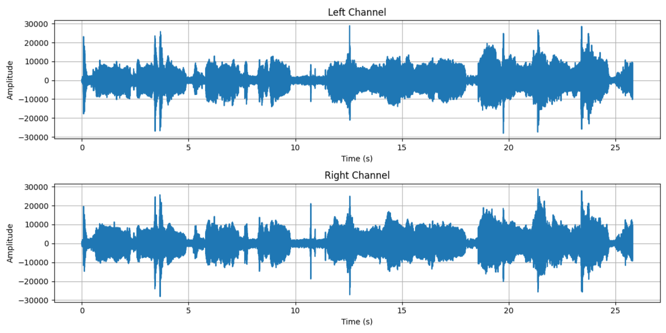

Fig. 7.2: $x(n)$ (.wav file)

taking Z-transform of (7.1)

$$Y(z)\sum_{m=0}^{M} a(m)(z^{-1})^m = X(z)\sum_{k=0}^{N} b(k)(z^{-1})^k \quad (7.2)$$

$$H(z) = \frac{\sum_{k=0}^{N} b(k)(z^{-1})^k}{\sum_{m=0}^{M} a(m)(z^{-1})^m} \quad (7.3)$$

The code computes the frequency response of the filter and plots it.

```
wget https://raw.githubusercontent.com/
    mnepraj/DSP/master/filter/codes/
    hnexercise.py
```

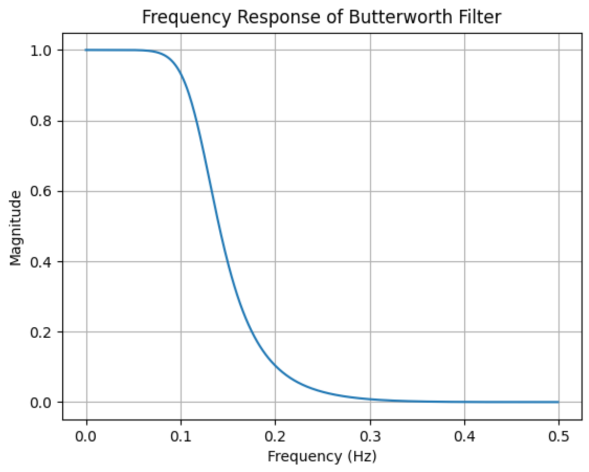

Fig. 7.2: frequency response of filter

$$Y(z) = X(z)H(z) \quad (7.4)$$

$$y(n) \overset{\mathcal{Z}}{\rightleftharpoons} Y(z) \quad (7.5)$$

The code computes y(n) and DFT of y(n) and plots it.

```
wget https://raw.githubusercontent.com/
    mnepraj/DSP/master/filter/codes/
    ynexercise.py
```

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 modifying the code with different input parameters and to get the best possible output.
Solution: The given butterworth filter is low pass with order=4 and cutoff-frequency=5.5kHz.

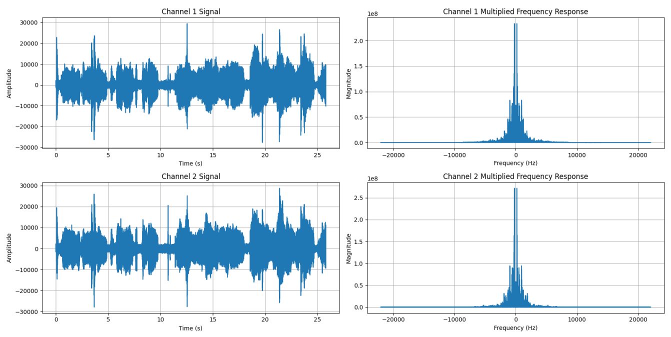

Fig. 7.2: $y(n)$ and its DFT