Audio Filter

EE23BTECH11032 - Kaustubh Khachane *

I. Digital Filter

I.1 Download the sound file from

https://github.com/Kaustubh−32/EE1205−Assignments/blob/main/Audio_Processing/audio/Kk3.wav

I.2 You will find a spectrogram at https://academo.org/demos/spectrum-analyzer. Upload the sound file that you downloaded in Problem I.1 in the spectrogram and play. Observe the spectrogram. What do you find?

Solution:

There are few yellow and red lines between 440 Hz to approximately 1 KHz. These are the high intensity instrumental strokes. We also observe frequencies upto approximately 9KHz and some notes which go upto 18KHz also. Majority of these are represented by the darker colors which implies they have low intensity.

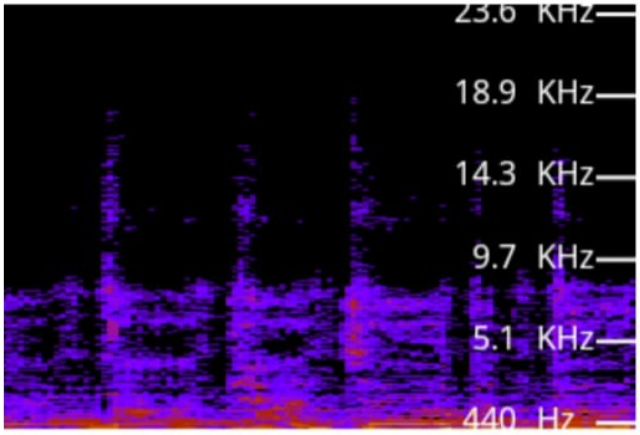
I.3 Write the python code for removal of out of band noise and execute the code.

Solution:

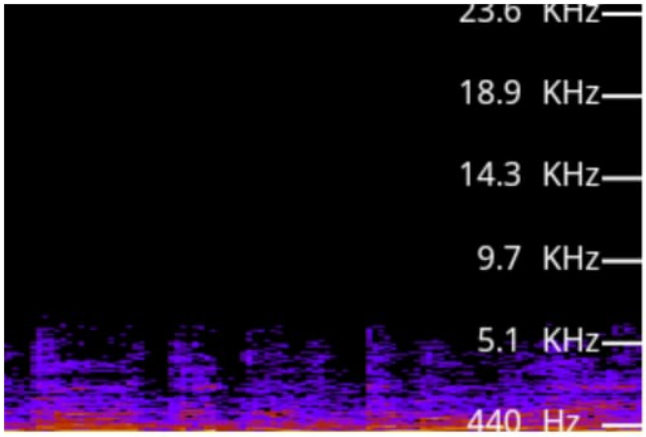
```
import soundfile as sf
from scipy import signal
input_signal,fs = sf.read("Kk3.wav")
sample_freq = fs
order = 4
cutoff_freq = 5000.0
Wn= 2*cutoff_freq/sample_freq
b,a = signal.butter(order,Wn,'low')
print(a)
print(b)
output_signal = signal.filtfilt(b,a,input_signal
    )
sf.write('Sound_with_reduced_noise.wav',
    output_signal,fs)
```

I.4 The output of the python script in Problem I.3 is the audio file Sound_With_ReducedNoise.wav. Play the file in the spectrogram in Problem I.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 5.1 kHz.



Spectogram observed for input audio



Spectogram observed for filtered audio

II. Difference Equation

II.1 Let

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

c

II.2 Let

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

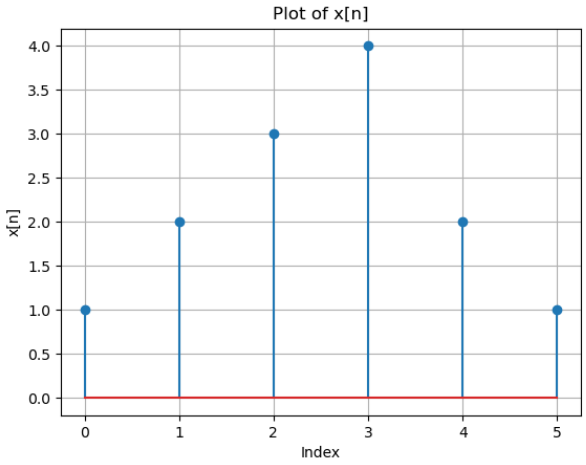
Sketch $y(n)$.

Solution: The following code yields Fig. II.2.

https://github.com/Kaustubh-32/EE1205-Assignments/blob/main/Audio_Processing/codes/2_1_1.py

The values used were generaed by the code

https://github.com/Kaustubh-32/EE1205-Assignments/blob/main/Audio_Processing/codes/2_1_1.c



Plot for x(n)

The following code yields Fig. II.2.

https://github.com/Kaustubh-32/EE1205-Assignments/blob/main/Audio_Processing/codes/2_1_2.py

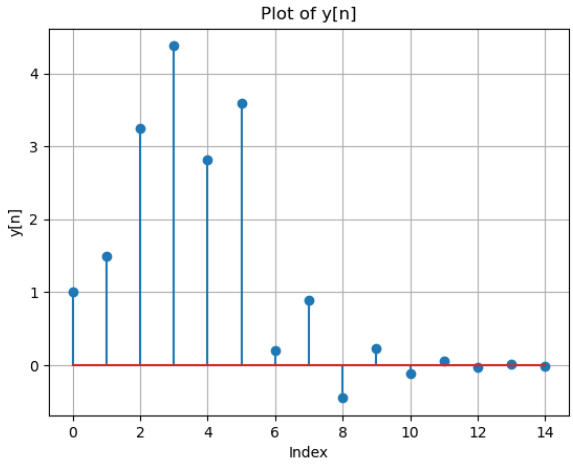
The values used were generaed by the code

https://github.com/Kaustubh-32/EE1205-Assignments/blob/main/Audio_Processing/codes/2_1_2.c

III. Z-transform

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

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



Plot for x(n)

Show that

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

and find

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

Solution: From (3),

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

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

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

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

III.2 Find

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

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

Solution: Applying (8) in (2),

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

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

III.3 Find the Z transform of

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

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

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

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

Solution: The following code plots the magnitude of transfer function.

https://github.com/Kaustubh-32/EE1205-Assignments/blob/main/Audio_Processing/codes/3_5.py

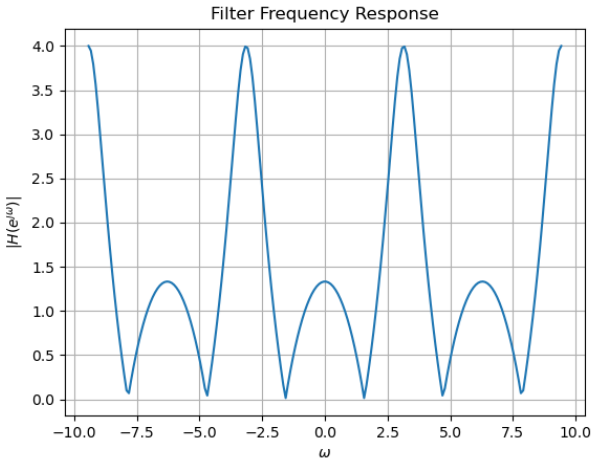


Fig. 1. $\left|H\left(e^{j\omega}\right)\right|$

IV. Impulse Response

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

$$h(n) \overset{\mathcal{Z}}{\longleftrightarrow} H(z) \tag{22}$$

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),

$$H(z) = \frac{1}{1 + \frac{1}{2}z^{-1}} + \frac{z^{-2}}{1 + \frac{1}{2}z^{-1}} \tag{23}$$

$$\implies h(n) = \left(-\frac{1}{2}\right)^n u(n) + \left(-\frac{1}{2}\right)^{n-2} u(n-2) \tag{24}$$

using (18) and (8).

IV.2 Sketch $h(n)$. Is it bounded? Convergent?

Solution: The following code plots Fig. 2.

https://github.com/Kaustubh-32/EE1205-Assignments/blob/main/Audio_Processing/codes/4_2.py

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

$$\sum_{n=-\infty}^{\infty} h(n) < \infty \tag{25}$$

Is the system defined by (2) stable for the impulse response in (22)?

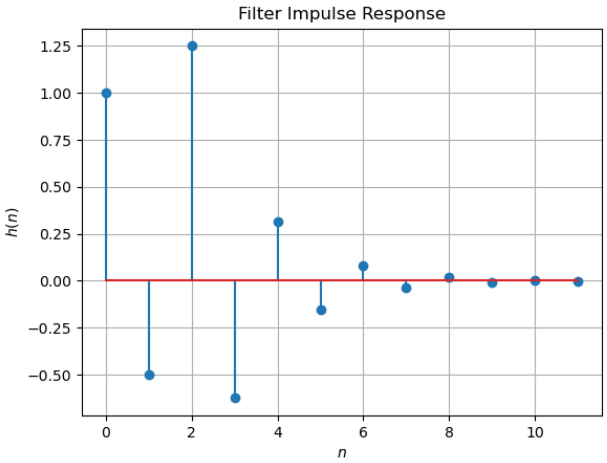


Fig. 2. $h(n)$ as the inverse of $H(z)$

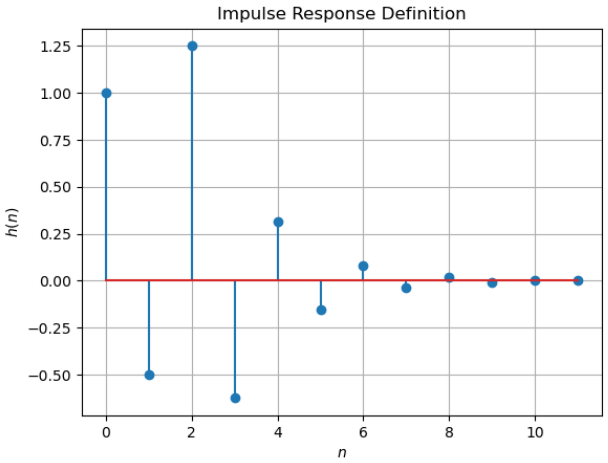


Fig. 3. $h(n)$ from the definition

Solution:
By equation (24),

$$h(n) = \left(-\frac{1}{2}\right)^n u(n) + \left(-\frac{1}{2}\right)^{n-2} u(n-2) \tag{26}$$

$$\implies \lim_{n\to\infty} h(n) = 0 \tag{27}$$

We got to equation (27) as u(n) and u(n-2) are 1 for the limit and as $|-\frac{1}{2}| < 1$ due to which it raised to very large n will be 0.
Thus, as $lim_{n\to\infty} h(n) = 0$, we can say that h(n) will converge.
Hence,

$$\sum_{n=-\infty}^{\infty} h(n) < \infty \tag{28}$$

As it converges, it is stable.

IV.4 Compute and sketch $h(n)$ using

$$h(n) + \frac{1}{2}h(n-1) = \delta(n) + \delta(n-2), \tag{29}$$

Solution:
This is the definition of $h(n)$.
Solution: The following code plots Fig. 3. Note that this is the same as Fig. 2.

https://github.com/Kaustubh-32/EE1205-Assignments/blob/main/Audio_Processing/codes/4_4.py

IV.5 Compute

$$y(n) = x(n) * h(n) = \sum_{n=-\infty}^{\infty} x(k)h(n-k) \tag{30}$$

Comment. The operation in (30) is known as *convolution*.
Solution: The following code plots Fig. 4. Note that this is the same as $y(n)$ in Fig. II.2.

https://github.com/Kaustubh-32/EE1205-Assignments/blob/main/Audio_Processing/codes/4_5.py

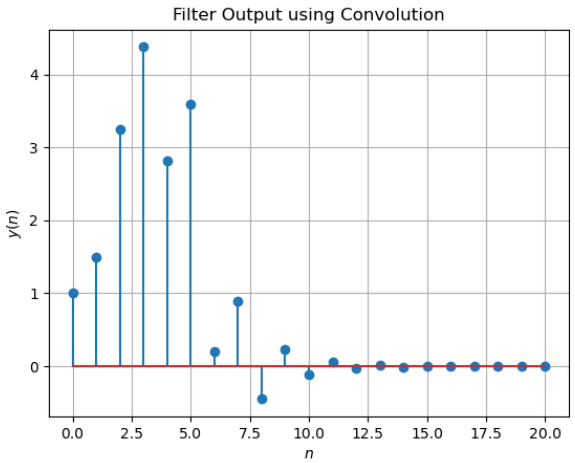


Fig. 4. $y(n)$ from the definition of convolution

IV.6 Show that

$$y(n) = \sum_{k=-\infty}^{\infty} x(n-k)h(k) \tag{31}$$

Solution: In (30), we substitute $r = n - k$ to

get

$$y(n) = \sum_{r=-\infty}^{\infty} x(r)\,h(n-r) \tag{32}$$

$$= \sum_{n-r=-\infty}^{\infty} x(n-r)\,h(r) \tag{33}$$

$$= \sum_{r=-\infty}^{\infty} x(n-r)\,h(r) \tag{34}$$

V. DFT and FFT

V.1 Compute

$$X(k) \triangleq \sum_{n=0}^{N-1} x(n)e^{-\mathrm{j}2\pi kn/N}, \quad k = 0, 1, \ldots, N-1 \tag{35}$$

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

V.2 Compute

$$Y(k) = X(k)H(k) \tag{36}$$

V.3 Compute

$$y(n) = \frac{1}{N}\sum_{k=0}^{N-1} Y(k)\cdot e^{\mathrm{j}2\pi kn/N}, \quad n = 0, 1, \ldots, N-1 \tag{37}$$

Solution: The following code plots Fig. 4. Note that this is the same as $y(n)$ in Fig. II.2.

https://github.com/Kaustubh-32/EE1205-Assignments/blob/main/Audio_Processing/codes/5_123.py

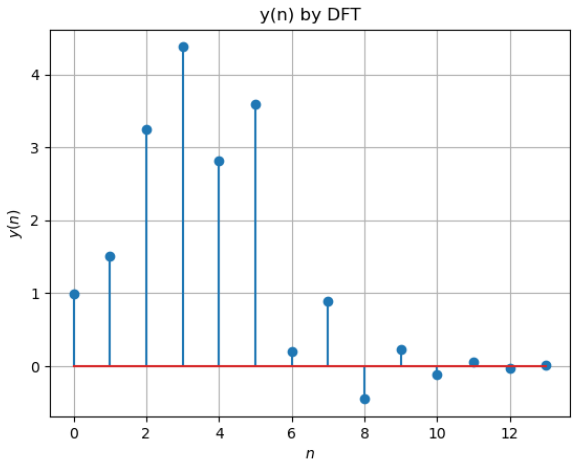


Fig. 5. $y(n)$ from the DFT

V.4 Repeat the previous exercise by computing $X(k), H(k)$ and $y(n)$ through FFT and IFFT.

Solution:

The above exercise has been performed in the code below

https://github.com/Kaustubh-32/EE1205-Assignments/blob/main/Audio_Processing/codes/5_4.py

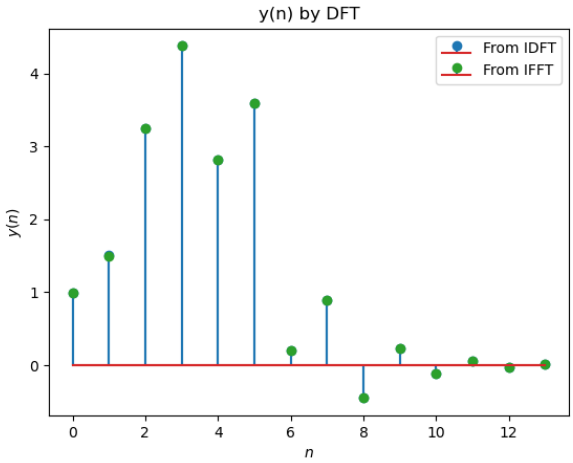


Fig. 6. $y(n)$ from the FFT abd IFFT

V.5 Wherever possible, express all the above equations as matrix equations.

Solution: Any DFT equation can be written as

$$\mathbf{X} = \mathbf{W}\mathbf{x} \tag{38}$$

Where matrix W is given by

$$\mathbf{W} = \begin{pmatrix} \omega^0 & \omega^0 & \ldots & \omega^0 \\ \omega^0 & \omega^1 & \ldots & \omega^{N-1} \\ \vdots & \vdots & \ddots & \vdots \\ \omega^0 & \omega^{N-1} & \ldots & \omega^{(N-1)(N-1)} \end{pmatrix} \tag{39}$$

$\omega = e^{-\frac{j2\pi}{N}}$

$$\mathbf{X} = \begin{pmatrix} X(0) \\ X(1) \\ \vdots \\ X(n-1) \end{pmatrix} \tag{40}$$

Thus we can rewrite (36) as:

$$\mathbf{Y} = \mathbf{X} \odot \mathbf{H} = (\mathbf{W}\mathbf{x}) \odot (\mathbf{W}\mathbf{h}) \tag{41}$$

$\odot$ is the element-wise multiplication of the matrices (Hadamard product).

VI. Exercises

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

VI.1 The command

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

in Problem I.3 is executed through the following difference equation

$$\sum_{m=0}^{M} a(m)\, y(n-m) = \sum_{k=0}^{N} b(k)\, x(n-k) \quad (42)$$

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 defines a custom signal.filtfilt function :

https://github.com/Kaustubh-32/EE1205-Assignments/blob/main/Audio_Processing/codes/6_1.py

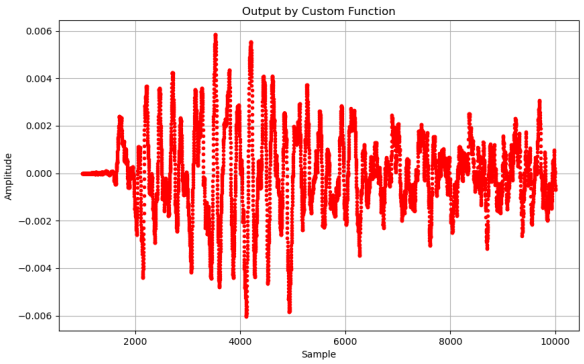


Fig. 7. Plot using my routine for signal.filtfilt

VI.2 Repeat all the exercises in the previous sections for the above a and b.

Solution:

$$M = 5 \quad (43)$$

$$N = 5 \quad (44)$$

a = 1 , -2.15448443 , 1.98942448 , -0.86509739 , 0.14814218

b = 0.00737405 , 0.02949621 , 0.04424432 , 0.02949621, 0.00737405

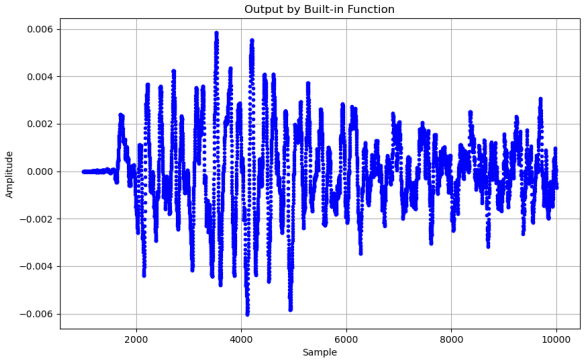


Fig. 8. Plot using builtin function

$$a(0)\, y(n) + a(1)\, y(n-1) + a(2)\, y(n-2) + a(3) \quad (45)$$
$$y(n-3) + a(4)\, y(n-4) = b(0)\, x(n) + b(1)\, x(n-1) + b(2)\, x(n-2) + b(3)\, x(n-3) + b(4)\, x(n-4)$$

Difference Equation is given by :

$$\begin{aligned} &y(n) - (-2.15448443)\, y(n-1) + (1.98942448)\, y(n-2) \\ &- (-0.86509739)\, y(n-3) + (0.14814218)\, y(n-4) \\ &= (0.00737405)\, x(n) + (0.02949621)\, x(n-1) \\ &+ (0.04424432)\, x(n-2) + (0.02949621)\, x(n-3) \\ &+ (0.00737405)\, x(n-4) \end{aligned} \quad (46)$$

From (42)

$$H(z) = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2} + \ldots + b_M z^{-N}}{a_0 + a_1 z^{-1} + a_2 z^{-2} + \ldots + a_N z^{-M}} \quad (47)$$

$$H(z) = \frac{\sum_{k=0}^{N} b(k) z^{-k}}{\sum_{k=0}^{M} a(k) z^{-k}} \quad (48)$$

Partial fraction on (48) can be generalised as:

$$H(z) = \sum_i \frac{r(i)}{1 - p(i) z^{-1}} + \sum_j k(j) z^{-j} \quad (49)$$

Now,

$$a^n u(n) \xleftrightarrow{\mathcal{Z}} \frac{1}{1 - a z^{-1}} \quad (50)$$

$$\delta(n-k) \xleftrightarrow{\mathcal{Z}} z^{-k} \quad (51)$$

Taking inverse z transform of (49) by using (50) and (51)

$$h(n) = \sum_i r(i)[p(i)]^n u(n) + \sum_j k(j)\delta(n-j) \quad (52)$$

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

https://github.com/Kaustubh−32/EE1205−Assignments/blob/main/Audio_Processing/codes/6_2_2_hn.py

Stability of h(n):

$$H(z) = \sum_{n=0}^{\infty} h(n) z^{-n} \tag{53}$$

$$H(1) = \sum_{n=0}^{\infty} h(n) = \frac{\sum_{k=0}^{N} b(k)}{\sum_{k=0}^{M} a(k)} < \infty \tag{54}$$

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

https://github.com/Kaustubh−32/EE1205−Assignments/blob/main/Audio_Processing/codes/6_2_3.py

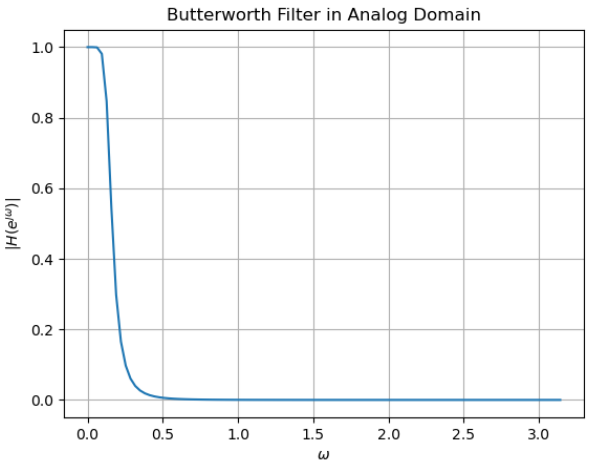


Fig. 9. Frequency Response of Audio Filter

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

$$z = \frac{1 + sT/2}{1 - sT/2} \tag{55}$$

https://github.com/Kaustubh−32/EE1205−Assignments/blob/main/Audio_Processing/codes/6_2_4.py

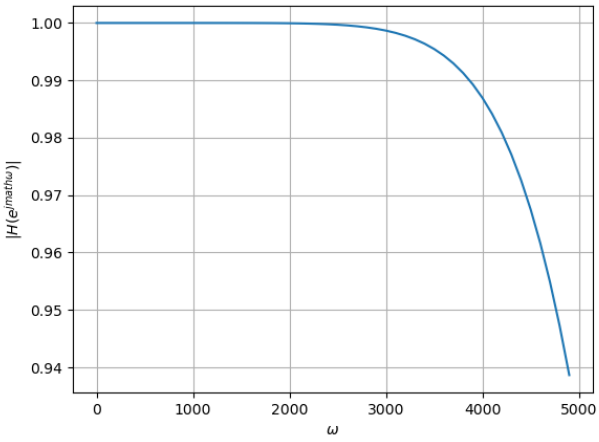


Fig. 10. Butterworth Filter Frequency response in analog domain

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

https://github.com/Kaustubh−32/EE1205−Assignments/blob/main/Audio_Processing/codes/6_2_2.py

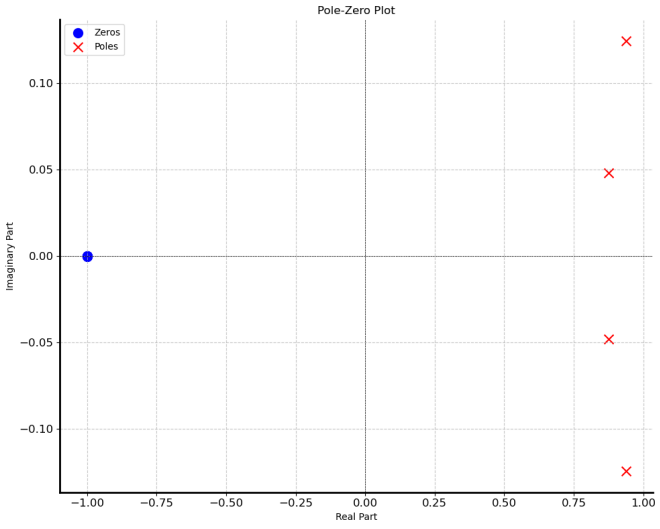


Fig. 11. There are complex poles. So $h(n)$ should be damped sinusoid.

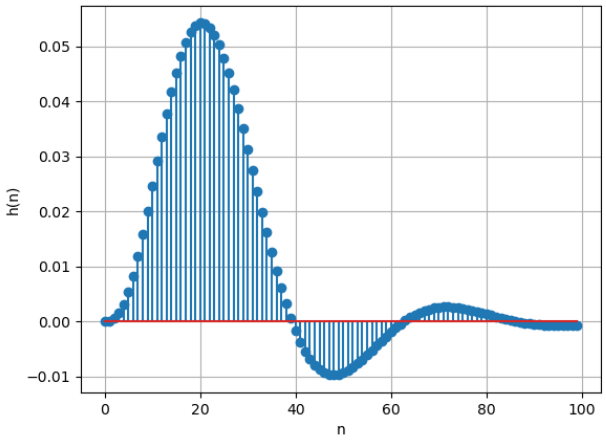


Fig. 12. h(n) of Audio Filter.It is a damped sinusoid.

VI.3 What is the sampling frequency of the input signal?

Solution:

Sampling frequency(fs)=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=2 and cutoff-frequency=4kHz.

VI.5 Modifying the code with different input parameters and to get the best possible output.

Solution:

The output can be improved by increasing the order of the filter. Also, the cutoff frequency can be changed to ensure undesired frequencies are filtered out.