

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
saml_freq=fs

#order of the filter
order=4

#cutoff frequency
cutoff_freq=10000.0

#digital frequency
Wn=2*cutoff_freq/saml_freq

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

#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>.

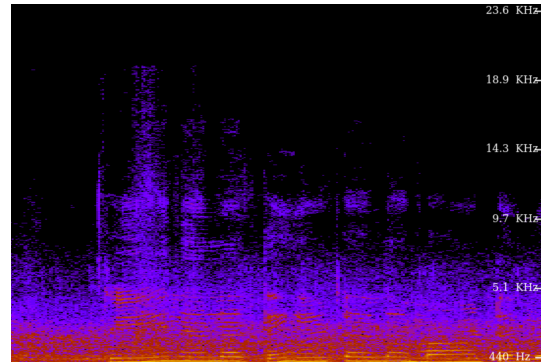


Fig. I.3. Spectrogram of the audio file before Filtering

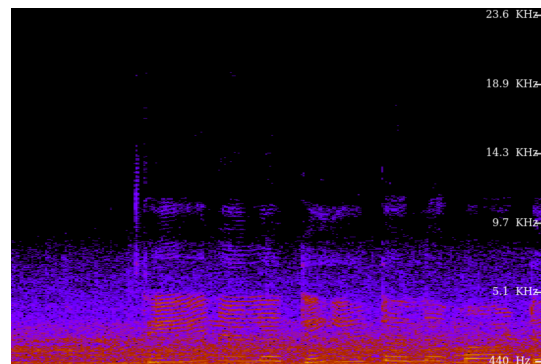


Fig. I.3. Spectrogram of the audio file after Filtering

II. DIFFERENCE EQUATION

- II1. Let

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

Sketch $x(n)$.

- II2. Let

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

$$y(n) = 0, n < 0 \quad (2)$$

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

$$H(e^{j\omega}) = H(z = e^{j\omega}). \quad (22)$$

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

Solution: Substituting $z = e^{j\omega}$ in (12),

$$H(e^{j\omega}) = \frac{1 + e^{-2j\omega}}{1 + \frac{1}{2}e^{-j\omega}} \quad (23)$$

$$|H(e^{j\omega})| = \left| \frac{1 + \cos 2\omega - j \sin 2\omega}{1 + \frac{1}{2}(\cos \omega - j \sin \omega)} \right| \quad (24)$$

$$= \sqrt{\frac{(1 + \cos 2\omega)^2 + (\sin 2\omega)^2}{\left(1 + \frac{1}{2}\cos \omega\right)^2 + \left(\frac{1}{2}\sin \omega\right)^2}} \quad (25)$$

$$= \frac{4|\cos \omega|}{\sqrt{5 + 4\cos \omega}} \quad (26)$$

$$|H(e^{j(\omega+2\pi)})| = \left| \frac{1 + e^{-2j(\omega+2\pi)}}{1 + \frac{1}{2}e^{-j(\omega+2\pi)}} \right| \quad (27)$$

$$= \frac{4|\cos \omega|}{\sqrt{5 + 4\cos \omega}} \quad (28)$$

$$= |H(e^{j\omega})| \quad (29)$$

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

\Rightarrow DTFT of a signal is always periodic.

The following code plots (III.5):

https://github.com/BATCHUIISHITHA/EE-1205/blob/main/audio_filtering/codes/3.5.py

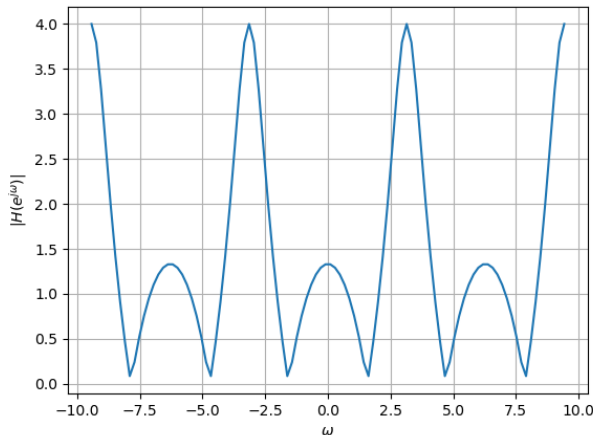


Fig. III.5. $|H(e^{j\omega})|$

IV. IMPULSE RESPONSE

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

$$h(n) \xleftrightarrow{Z} H(z) \quad (30)$$

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 (12),

$$H(z) = \frac{1}{1 + \frac{1}{2}z^{-1}} + \frac{z^{-2}}{1 + \frac{1}{2}z^{-1}} \quad (31)$$

$$\Rightarrow h(n) = \left(-\frac{1}{2}\right)^n u(n) + \left(-\frac{1}{2}\right)^{n-2} u(n-2) \quad (32)$$

from (21) and (9).

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

https://github.com/BATCHUIISHITHA/EE-1205/blob/main/audio_filtering/codes/4.2.py

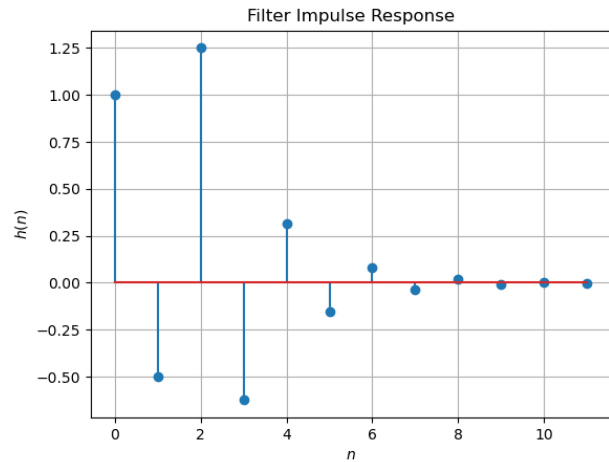


Fig. IV.2. $h(n)$ as the inverse of $H(z)$

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

$$\sum_{n=-\infty}^{\infty} h(n) < \infty \quad (33)$$

Is the system defined by (2) stable for impulse response in (30)?

Solution: For stable system (33) must be converging.

For $n \rightarrow \infty$,

$$u(n) = u(n-2) = 1 \quad (34)$$

$$\Rightarrow h(n) = \left(-\frac{1}{2}\right)^n + \left(-\frac{1}{2}\right)^{n-2} \quad (35)$$

Since, both terms of $h(n)$ tends to 0 as $n \rightarrow \infty$, $h(n) \rightarrow 0$.

\Rightarrow output remains bounded for bounded inputs, ie: $h(n)$ is stable.

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

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

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/BATCHUIISHITHA/EE-1205/blob/main/audio_filtering/codes/4.4.py

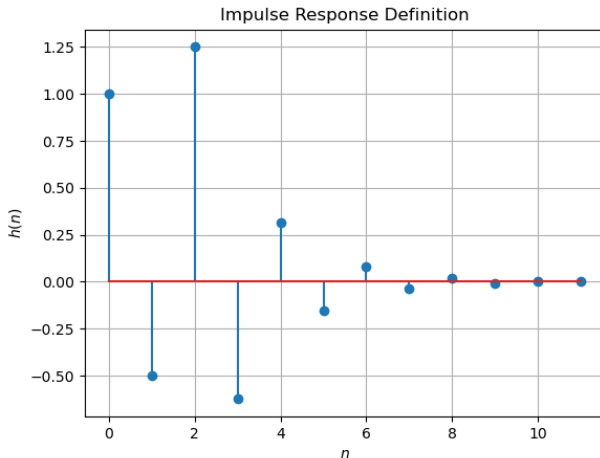


Fig. IV.4. $h(n)$ from the definition

IV.5 Compute

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

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

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

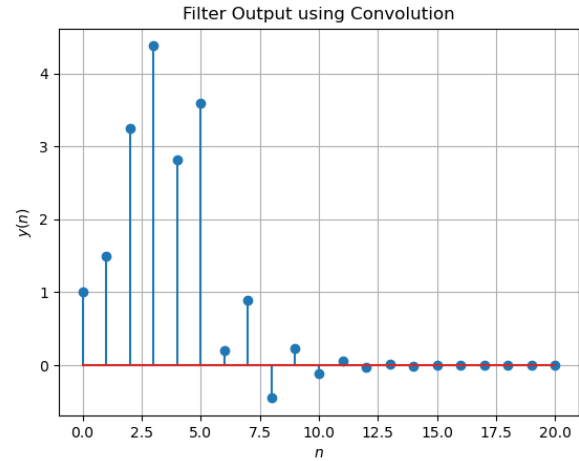


Fig. IV.5. $y(n)$ from the definition of convolution

https://github.com/BATCHUIISHITHA/EE-1205/blob/main/audio_filtering/codes/4.5.py

IV.6 Show that

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

Solution: In (37), replacing k by $n-k$

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

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

V. DFT AND FFT

V.1 Compute

$$X(k) \triangleq \sum_{n=0}^{N-1} x(n)e^{-\frac{2\pi jkn}{N}}, \quad k = 0, 1, \dots, N-1 \quad (41)$$

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

V.2 Compute

$$Y(k) = X(k)H(k) \quad (42)$$

V.3 Compute

$$y(n) = \frac{1}{N} \sum_{k=0}^{N-1} Y(k) \cdot e^{-\frac{2\pi jkn}{N}}, \quad n = 0, 1, \dots, N-1 \quad (43)$$

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

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

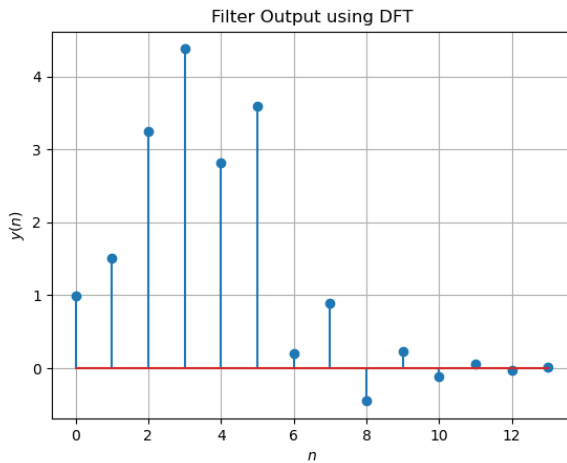


Fig. V.3. $y(n)$ from the DFT

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/BATCHUIISHITHA/EE-1205/blob/main/audio_filtering/codes/5.4.py

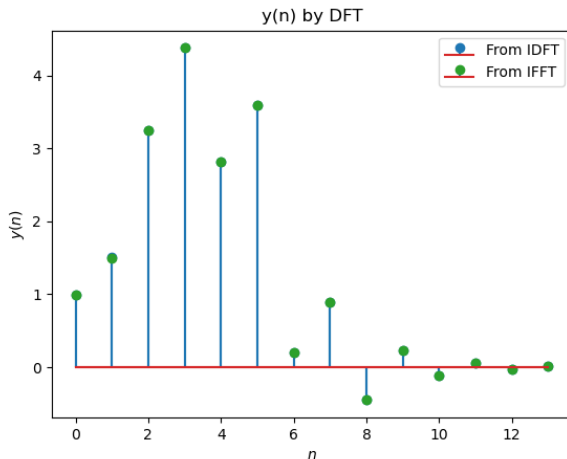


Fig. V.4. $y(n)$ from the DFT, IDFT and from the FFT, IFFT are plotted and verified

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

Solution: The DFT matrix is defined as :

$$\mathbf{W} = \begin{pmatrix} \omega^0 & \omega^0 & \dots & \omega^0 \\ \omega^0 & \omega^1 & \dots & \omega^{N-1} \\ \vdots & \vdots & \ddots & \vdots \\ \omega^0 & \omega^{N-1} & \dots & \omega^{(N-1)(N-1)} \end{pmatrix} \quad (44)$$

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

$$\mathbf{X} = \mathbf{W}\mathbf{x} \quad (45)$$

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

$$\mathbf{x} = \begin{pmatrix} x(0) \\ x(1) \\ \vdots \\ x(n-1) \end{pmatrix} \quad (46)$$

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

Thus we can rewrite (42) as:

$$\mathbf{Y} = \mathbf{X} \odot \mathbf{H} = (\mathbf{W}\mathbf{x}) \odot (\mathbf{W}\mathbf{h}) \quad (48)$$

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

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

$$\sum_{m=0}^M a(m) y(n-m) = \sum_{k=0}^N b(k) x(n-k) \quad (49)$$

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/BATCHUIISHITHA/EE-1205/blob/main/audio_filtering/codes/6.1.py

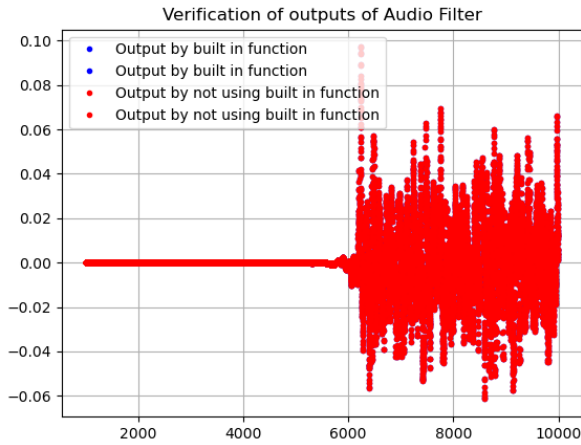


Fig. VI.1. output of an audio signal with and without inbuilt function signal.lfilter are plotted and verified

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,

$$M = 5 \quad (50)$$

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

From 49

$$a(0)y(n) + a(1)y(n-1) + a(2)y(n-2) + a(3)y(n-3) + a(4)y(n-4) = b(0)x(n) + b(1)x(n-1) + b(2)x(n-2) + b(3)x(n-3) + b(4)x(n-4) \quad (52)$$

$$y(n) - (3.66)y(n-1) + (5.05)y(n-2) - (3.099)y(n-3) + (0.715)y(n-4) = (1.45 \times 10^{-5})x(n) + (5.74 \times 10^{-5})x(n-1) + (8.62 \times 10^{-5})x(n-2) + (5.74 \times 10^{-5})x(n-3) + (1.43 \times 10^{-5})x(n-4) \quad (53)$$

Difference Equation is given by :

$$\begin{aligned} & y(n) - (3.66)y(n-1) + (5.05)y(n-2) \\ & - (3.099)y(n-3) + (0.715)y(n-4) \\ & = (1.45 \times 10^{-5})x(n) + (5.74 \times 10^{-5})x(n-1) \\ & + (8.62 \times 10^{-5})x(n-2) + (5.74 \times 10^{-5})x(n-3) \\ & + (1.43 \times 10^{-5})x(n-4) \end{aligned} \quad (53)$$

From (49)

$$H(z) = \frac{b_0 + b_1z^{-1} + b_2z^{-2} + \dots + b_Mz^{-M}}{a_0 + a_1z^{-1} + a_2z^{-2} + \dots + a_Nz^{-N}} \quad (54)$$

$$H(z) = \frac{\sum_{k=0}^N b(k)z^{-k}}{\sum_{k=0}^M a(k)z^{-k}} \quad (55)$$

Partial fraction on (55) can be generalised as:

$$H(z) = \sum_i \frac{r(i)}{1 - p(i)z^{-1}} + \sum_j k(j)z^{-j} \quad (56)$$

Now,

$$a^n u(n) \xleftrightarrow{\Gamma^r} Z \frac{1}{1 - az^{-1}} \quad (57)$$

$$\delta(n - k) \xleftrightarrow{\Gamma^r} Z z^{-k} \quad (58)$$

Taking inverse z transform of (56) by using (57) and (58)

$$h(n) = \sum_i r(i)[p(i)]^n u(n) + \sum_j k(j)\delta(n - j) \quad (59)$$

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

https://github.com/BATCHUIISHITHA/EE-1205/blob/main/audio_filtering/codes/6.2.py

$r(i)$	$p(i)$	$k(i)$
$0.58129538 - 2.51766121j$	$0.13676769 + 0.19533079j$	2.02482078
$0.58129538 + 2.51766121j$	$0.13676769 - 0.19533079j$	$-$
$-0.40482158 + 0.3262658j$	$0.188968140 + 0.6515556j$	$-$
$-0.40462158 - 0.3262658j$	$0.188968140 - 0.6515556j$	$-$

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

Stability of $h(n)$:

According to (33)

$$H(z) = \sum_{n=0}^{\infty} h(n)z^{-n} \quad (60)$$

$$H(1) = \sum_{n=0}^{\infty} h(n) = \frac{\sum_{k=0}^N b(k)}{\sum_{k=0}^M a(k)} < \infty \quad (61)$$

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

The below code plots Filter frequency response

Solution: A better filtering was found on setting the order of the filter to be ...

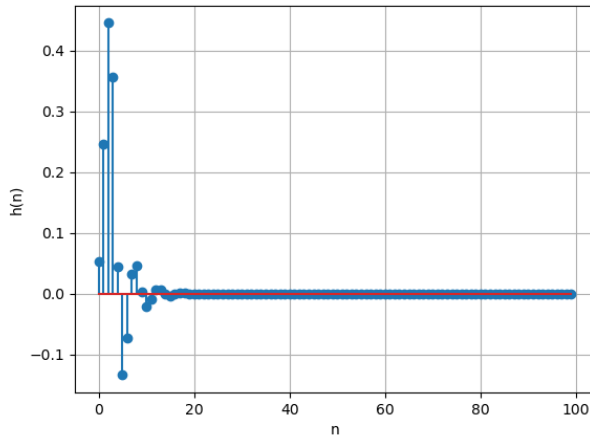


Fig. VI.2. $h(n)$ of Audio Filter

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

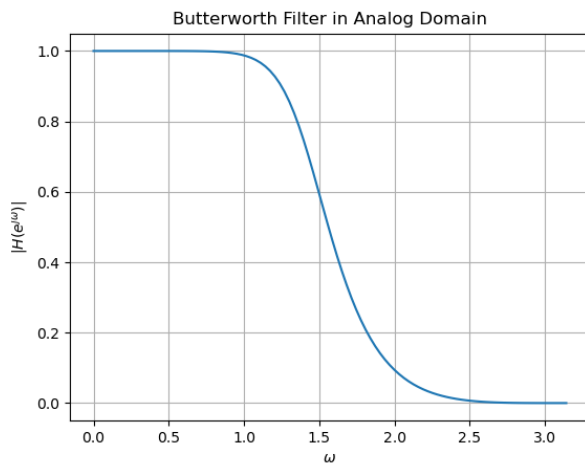


Fig. VI.2. Frequency Response of Audio Filter

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

Solution: The sampling frequency of the input signal is 4.80kHz.

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

Solution: The given butterworth filter is lowpass with order=4 and cutoff-frequency=10kHz.

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