Assignment 1

KRATI ARELA - EE18BTECH11050

Download all python codes from

```
https://github.com/Krati012/EE3025/tree/main/Assignment1/codes
```

and latex-tikz codes from

```
https://github.com/Krati012/EE3025/tree/main/Assignment1
```

1 Digital Filter

1.1 Download the sound file from

```
wget https://raw.githubusercontent.com/gadepall/EE1310/master/filter/codes/Sound_Noise.wav
```

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

```
import soundfile as sf
from scipy import signal

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

#sampling frequency of input signal
sampl_freq = fs

#order of the filter
order = 4

#cutoff frequency 4kHz
cutoff_freq = 4000.0

#digital frequency
Wn = 2*cutoff_freq/sampl_freq

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

#filter the input signal with butterowrth filtler
output_signal = signal.filtfilt(b, a, input_signal)
#output_signal = signal.lfilter(b, a, input_signal)

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

2 Difference equation

2.1 Write the difference equation of the above Digital filter obtained in problem 1.2.
Solution:

$$\sum_{m=0}^{M} a(m)\, y(n-m) = \sum_{k=0}^{N} b(k)\, x(n-k) \tag{2.0.1}$$

$$\begin{aligned} y(n) - 2.52y(n-1) + 2.56y(n-2) - 1.206y(n-3) \\ +0.22013y(n-4) = 0.00345x(n) + 0.0138x(n-1) \\ +0.020725x(n-2) + 0.0138x(n-3) + 0.00345x(n-4) \end{aligned} \tag{2.0.2}$$

2.2 Sketch x(n) and y(n).
Solution: The following code yields Fig. 2.2

```
codes/x_y.py
```

The filtered sound signal obtained through difference equation is found in

```
codes/Sound_diffEq.wav
```

3 Z-transform

3.1

$$X(z) = \mathcal{Z}\{x(n)\} = \sum_{n=-\infty}^{\infty} x(n) z^{-n} \tag{3.0.1}$$

Show that

$$\mathcal{Z}\{x(n-1)\} = z^{-1} X(z) \tag{3.0.2}$$

and find

$$\mathcal{Z}\{x(n-k)\} \tag{3.0.3}$$

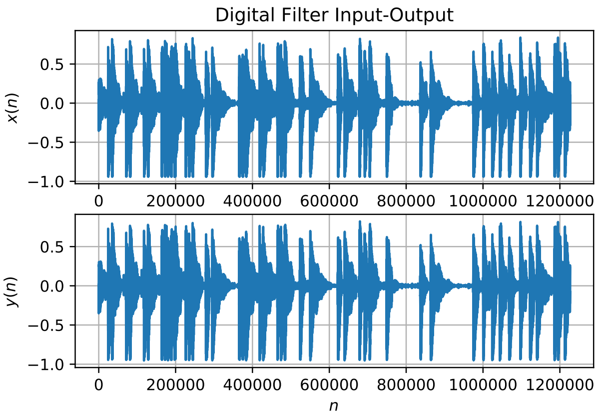

Fig. 2.2

Solution: From (3.0.1),

$$\mathcal{Z}\{x(n-k)\} = \sum_{n=-\infty}^{\infty} x(n-1)z^{-n} \tag{3.0.4}$$

$$= \sum_{n=-\infty}^{\infty} x(n)z^{-n-1} = z^{-1}\sum_{n=-\infty}^{\infty} x(n)z^{-n} \tag{3.0.5}$$

resulting in (3.0.2). Similarly, it can be shown that

$$\mathcal{Z}\{x(n-k)\} = z^{-k}X(z) \tag{3.0.6}$$

3.2 Find

$$H(z) = \frac{Y(z)}{X(z)} \tag{3.0.7}$$

from (2.0.2) assuming that the Z-transform is a linear operation.
Solution: Applying (3.0.6) in (2.0.2) we get,

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

$$= \frac{b[0] + b[1]z^{-1} + b[2]z^{-2} + b[3]z^{-3} + b[4]z^{-4}}{a[0] + a[1]z^{-1} + a[2]z^{-2} + a[3]z^{-3} + a[4]z^{-4}} \tag{3.0.8}$$

3.3 Let

$$H\left(e^{\jmath w}\right) = H\left(z = e^{\jmath w}\right). \tag{3.0.9}$$

Plot $|H\left(e^{\jmath w}\right)|$.
Solution: The following code plots Fig. 3.3.

```
codes/dtft.py
```

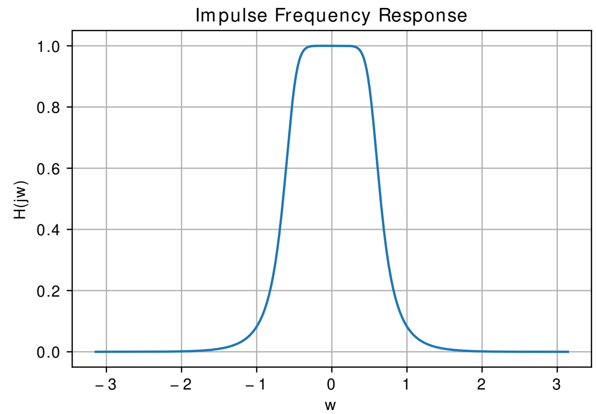

Fig. 3.3: $|H\left(e^{\jmath w}\right)|$

4 Impulse Response

4.1 From the difference equation eq. 2.0.2. Sketch h(n).
Solution: We know that when $x(n) = \delta(n)$ (input is impulse), we get the Impulse response h(n) of the system as output.
From eq.2.0.1,
Substitute $x(n-k) = \delta(n-k)$,
$y(n-k)$ becomes $h(n-k)$ for all k=0,1,2,3,4.
Now, the following code plots Fig. 4.1

```
codes/hn.py
```

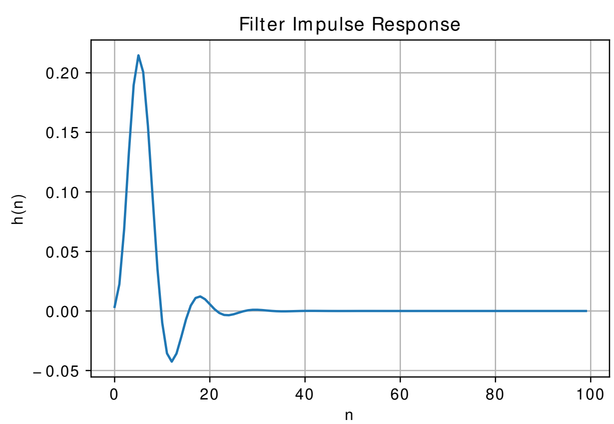

Fig. 4.1: $h(n)$

4.2 Is the system defined in eq. 2.0.2 for impulse response obtained above stable?
Solution: The system is defined by the eq. 2.0.2 For a system to be stable, output should be bounded for every bounded input. This is known as BIBO stability.

Since the audio input x(n) is bounded, let B_x be some finite value, we have

$$|x(n)| < B_x < \infty \tag{4.0.1}$$

From convolution property,

$$|y(n)| = \left|\sum_{-\infty}^{\infty} h(k)x(n-k)\right| \tag{4.0.2}$$

$$|y(n)| \le \sum_{-\infty}^{\infty} |h(k)|\,|x(n-k)| \tag{4.0.3}$$

Let B_x be the maximum value x(n-k) can take, then

$$|y(n)| \le B_x \sum_{-\infty}^{\infty} |h(k)| \tag{4.0.4}$$

If

$$\sum_{-\infty}^{\infty} |h(k)| < \infty \tag{4.0.5}$$

Then

$$|y(n)| \le B_y < \infty \tag{4.0.6}$$

Therefore we can say that y(n) is bounded if x(n) and h(n) are bounded.
Since the audio input is bounded, the system is said to be stable if h(n) is also bounded.

$$\sum_{n=-\infty}^{\infty} |h(n)| < \infty \tag{4.0.7}$$

The above equation can be re-written as,

$$\sum_{n=-\infty}^{\infty} |h(n)| \left|z^{-n}\right|_{|z|=1} < \infty \tag{4.0.8}$$

$$\sum_{n=-\infty}^{\infty} \left|h(n)z^{-n}\right|_{|z|=1} < \infty \tag{4.0.9}$$

From Triangle inequality,

$$\sum_{n=-\infty}^{\infty} \left|h(n)z^{-n}\right|_{|z|=1} < \left|\sum_{n=-\infty}^{\infty} h(n)z^{-n}\right|_{|z|=1} \tag{4.0.10}$$

$$\implies |H(n)|_{|z|=1} < \infty \tag{4.0.11}$$

Therefore, the Region of Convergence (ROC) should include the unit circle for the system to be stable.
Since, h(n) is right sided the ROC is outside the outer most pole. From the equation (3.0.8)
Poles of the given transfer equation is:

$$z(approx) = 0.694 \pm 0.41i, \quad 0.566 \pm 0.134i \tag{4.0.12}$$

From the above poles, we can see that that the ROC of the system is $|z| > \sqrt{0.694^2 + 0.41^2}$.
$\implies |z| > 0.806$

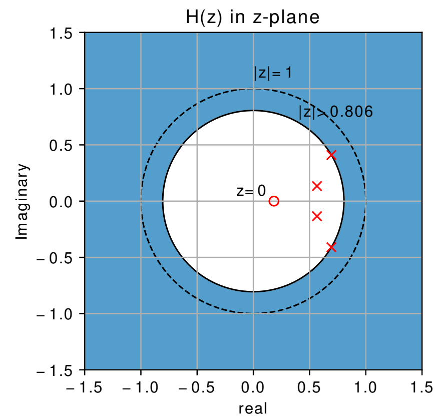

Fig. 4.2: $H(z)$ in z-plane

The code for plotting H(z) in z-plane is:

```
codes/roc.py
```

From the figure we can observe that ROC of the system includes unit circle $|z| = 1$. Which implies that the given IIR filter is stable, because h(n) is absolutely summable.
Verification:
Given input audio signal x(n) which is bounded, and system difference equation 2.0.2
From python code we can get that the maximum value of x(n) is 0.839 and minimum value is -0.9417.
Similarly we can also get that the maximum value of y(n) is 0.82225 and minimum value is -0.95376 and it tends to zero as n tends to infinity.
We can say that the bounded input x(n) gives bounded output y(n). Therefore we can say that the system is BIBO stable.

4.3 Using h(n) obtained in 4.1 compute filtered output using the below equation of convolution

$$y(n) = x(n) * h(n) = \sum_{n=-\infty}^{\infty} x(k)h(n-k) \tag{4.0.13}$$

Solution: The following code plots Fig. 4.3

codes/ynconv.py

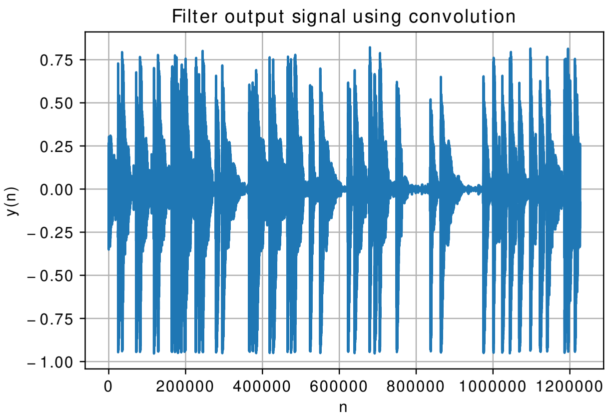

Fig. 4.3: $y(n)$ from the definition of convolution

The filtered sound signal through convolution from this method is found in

codes/Sound_conv.wav

We can observe that the output obtained is same as y(n) obtained in Fig. 2.2

5 DFT and FFT

5.1 Compute

$$X(k) \triangleq \sum_{n=0}^{N-1} x(n) e^{-j2\pi kn/N}, \quad k = 0, 1, \dots, N-1 \tag{5.0.1}$$

and $H(k)$ using h(n).
Solution: For this given IIR system with audio sample as x(n) and h(n) as impulse response h(n) obtained in 4.1
DFT of a Input Signal $x(n)$ is

$$X(k) \triangleq \sum_{n=0}^{N-1} x(n) e^{-j2\pi kn/N}, \quad k = 0, 1, \dots, N-1 \tag{5.0.2}$$

DFT of a Impulse Response $h(n)$ is

$$H(k) \triangleq \sum_{n=0}^{N-1} h(n) e^{-j2\pi kn/N}, \quad k = 0, 1, \dots, N-1 \tag{5.0.3}$$

The following code plots FFT of $x(n)$ and $h(n)$ in Fig. 5.1.

codes/xhfft.py

Magnitude and Phase plots obtained through above code is

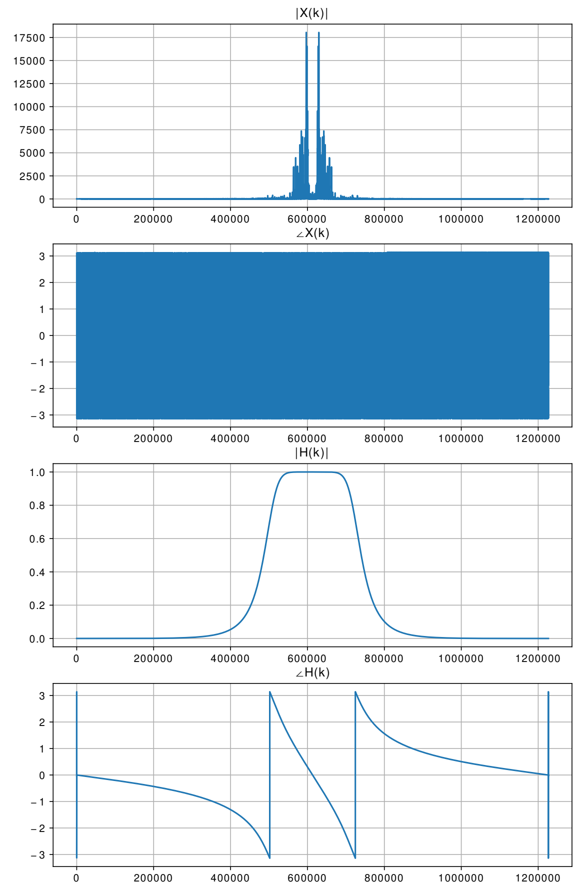

Fig. 5.1: $X(k)$ and $H(k)$

5.2 Compute

$$Y(k) = X(k)H(k) \tag{5.0.4}$$

and using this find

$$y(n) \triangleq \sum_{k=0}^{N-1} Y(k) e^{j2\pi kn/N}, \quad n = 0, 1, \dots, N-1 \tag{5.0.5}$$

Solution: The following code plots Fig.5.2

codes/yfft.py

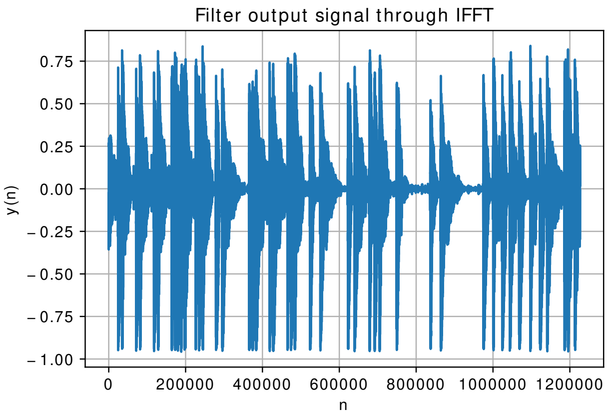

Fig. 5.2: $y(n)$ from IFFT

The filtered sound signal from this method is found in

codes/Sound_fft.wav

We can observe from the above plot that it is same as the y(n) observed in Fig.2.2