Assignment 1

V.L.Narasimha Reddy - EE18BTECH11046

Download all python codes from

```
https://github.com/narasimha−123/EE3250/
    Assignment1/codes
```

and latex-tikz codes from

```
https://github.com/narasimha−123/EE3250/
    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 frquency 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 butterworth filter
output_signal = signal.filtfilt(b, a,
    input_signal)
#output_signal = signal.lfilter(b, a,
    input_signal)

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

2 Difference equation

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

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

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

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

```
codes/xnyn.py
```

The filtered sound signal obtained through difference equation is found in

```
codes/Sound_de.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

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

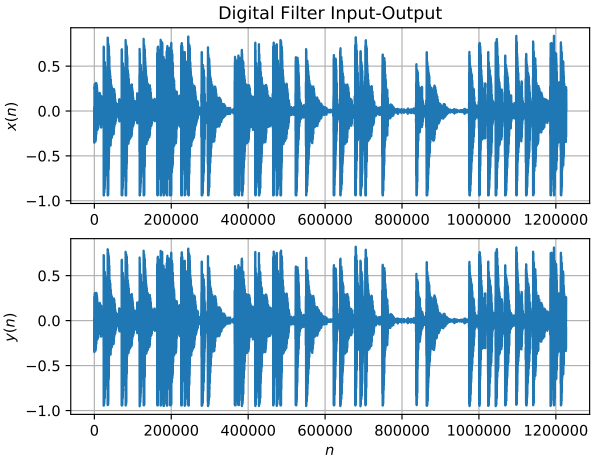

Fig. 2.2

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,

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

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

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
```

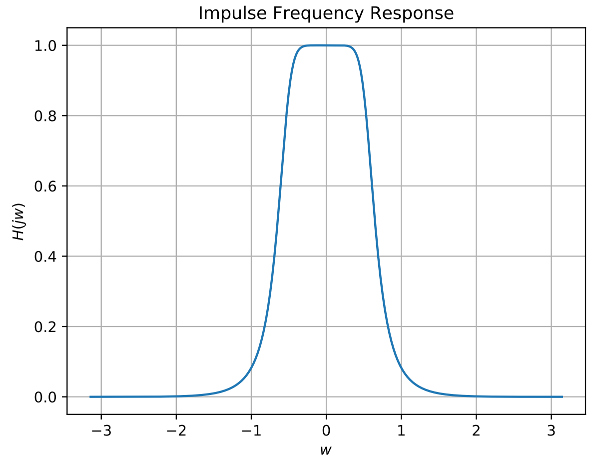

Fig. 3.3: $|H\left(e^{\jmath w}\right)|$

4 Impulse Response

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

```
codes/hndef.py
```

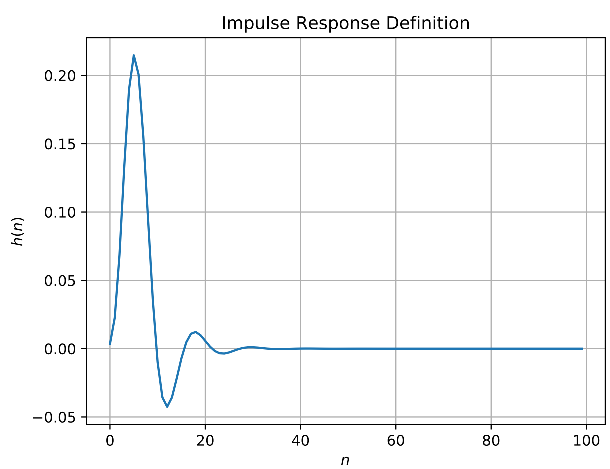

Fig. 4.1: $h(n)$

4.2 Check whether h(n) obtained is stable.
Solution: The system is defined by the equation 2.0.2 For a system to be stable, output

should be bounded when the input is bounded. This is known as BIBO stability.
Since x(n) is bounded, let B_x be some finite value

$$|y(n)| \leq B_x \sum_{-\infty}^{\infty} |x(n-k)| \tag{4.0.1}$$

From convolution formula,

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

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

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

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

If

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

Then

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

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

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

The above euation can be re written as,

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

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

From Triangle inequality,

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

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

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:

$$\begin{aligned} z(approx) = 0.69382 \pm 0.41i, \\ 0.56617835 \pm 0.134423 \end{aligned} \tag{4.0.12}$$

From the above poles, we can see that that the ROC of the system is $|z| > \sqrt{0.69382^2 + 0.41^2}$. From the figure we can observe that ROC of

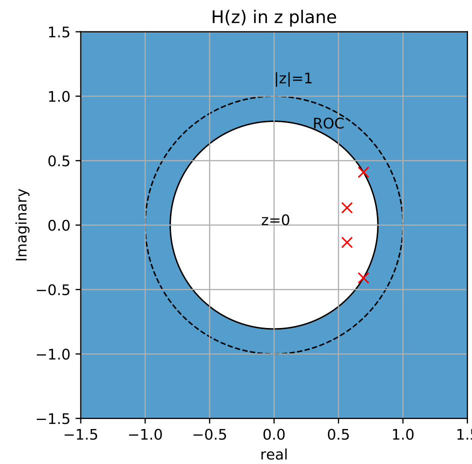

Fig. 4.2: $X(k) and H(k)$

the system includes unit circle $|z| = 1$.
The code for plotting the figure is:

```
codes/ROC.py
```

Which implies that the given IIR filter is stable, beacuse h(n) is absolutely summable.
Verification:-
Given bounded input x(n) (audio sample) and system difference equation 2.0.2
From 2.2 we can see that the maximum value of x(n) is 0.839 and minimum value is greater than -0.94171.
Simillarly 2.2 we can also see that the maximum value of y(n) is 0.822256 and minimum value is -0.953761 and it tends to zero after the length of signal.
We can see that the bounded input x(n) gives bounded output y(n). Therefore we can say that the system is BIBO stable.

4.3 Compute Filtered output using convolution formula using h(n) obtained in 4.1

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

Solution: The following code plots Fig. 4.3

```
/codes/ynconv.py
```

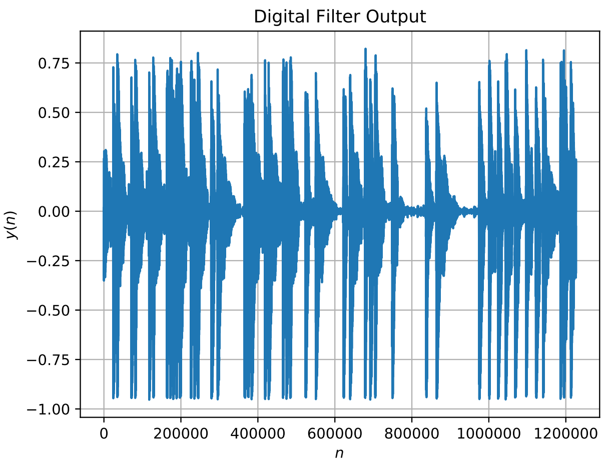

Fig. 4.3: $y(n)$ from the definition of convolution

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 FFT and IFFT

5.1 Compute

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

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

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

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

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

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

```
codes/xnhnfft.py
```

Magnitude and Phase plots obtained through above code is

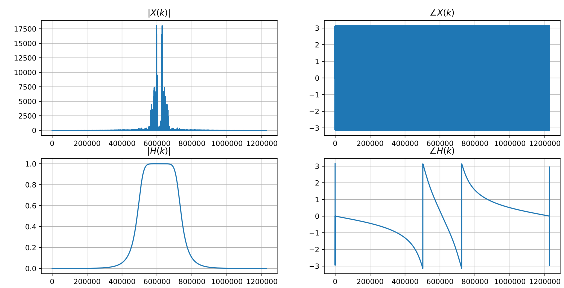

Fig. 5.1: $X(k) and H(k)$

5.2 From

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

Compute

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

Solution: The following code plots Fig.5.2

```
codes/ynfft.py
```

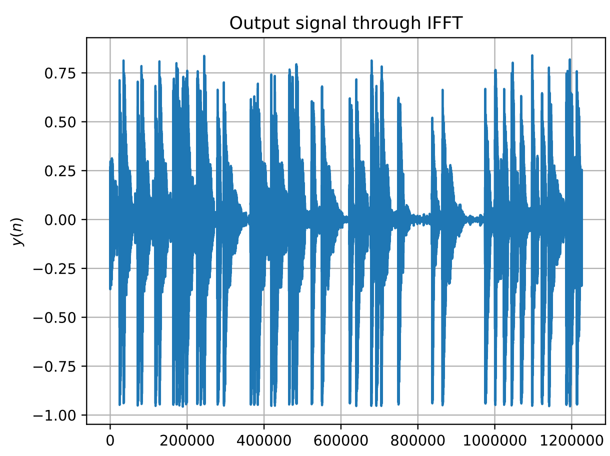

Fig. 5.2: $y(n)$

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