Assignment 1

KRATI ARELA - EE18BTECH11050

Download all C and python codes from

```
https://github.com/Krati012/EE3025/tree/main/
    Assignment1_C/codes
```

and latex-tikz codes from

```
https://github.com/Krati012/EE3025/tree/main/
    Assignment1_C
```

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 generates x(n) as x.dat file from .wav file

```
codes/generateX.py
```

The following code computes $x(n)$ and $y(n)$ from the difference equation.

```
codes/x_y.c
```

The following code plots $x(n)$ and $y(n)$ in Fig. 2.2

```
codes/plot_xy.py
```

3 DFT and FFT

3.1 Compute

$$X(k) \triangleq \sum_{n=0}^{N-1} x(n) e^{-j2\pi kn/N}, \quad k = 0, 1, \ldots, N-1 \tag{3.0.1}$$

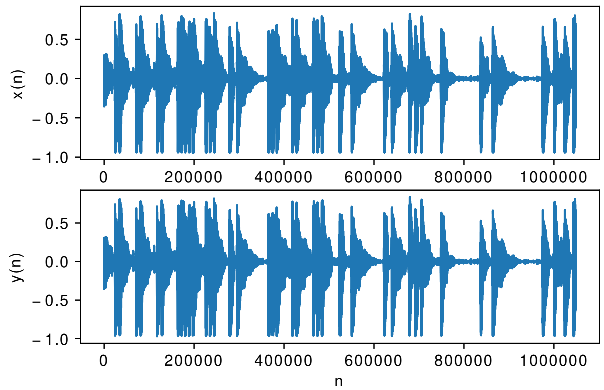

Fig. 2.2: Digital Filter Input-Output

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

$$X(k) \triangleq \sum_{n=0}^{N-1} x(n)e^{-j2\pi kn/N}, \quad k = 0, 1, \ldots, N-1 \tag{3.0.2}$$

DFT of Impulse Response $h(n)$ is

$$H(k) \triangleq \sum_{n=0}^{N-1} h(n)e^{-j2\pi kn/N}, \quad k = 0, 1, \ldots, N-1 \tag{3.0.3}$$

The following C code computes FFT of $x(n)$ and $h(n)$ using divide and conquer approach using recursive calls and saves the results in .dat file

```
codes/fft.c
```

The following code plots FFT of $x(n)$ and $h(n)$ using .dat files obtained from the above code

```
codes/plotFFT.py
```

Magnitude plots of $|X(k)|$ and $|H(k)|$ obtained through above code is in Fig. 3.1. We can observe that these above plots look similar to the ones generated using the in-built FFT function.

3.2 From

$$Y(k) = X(k)H(k) \tag{3.0.4}$$

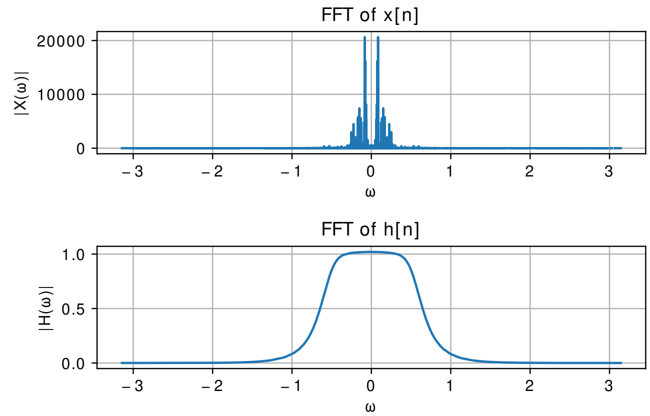

Fig. 3.1: $X(k)$ and $H(k)$

compute

$$y(n) \triangleq \sum_{k=0}^{N-1} Y(k)e^{j2\pi kn/N}, \quad n = 0, 1, \ldots, N-1 \tag{3.0.5}$$

Solution: The following C code computes $Y(k)$ by multiplying $X(k)$ and $H(k)$ and then computes IFFT to obtain $y(n)$

```
codes/ifft.c
```

The following code plots $y(n)$ in Fig.3.2 from .dat obtained from above code

```
codes/plotIFFT.py
```

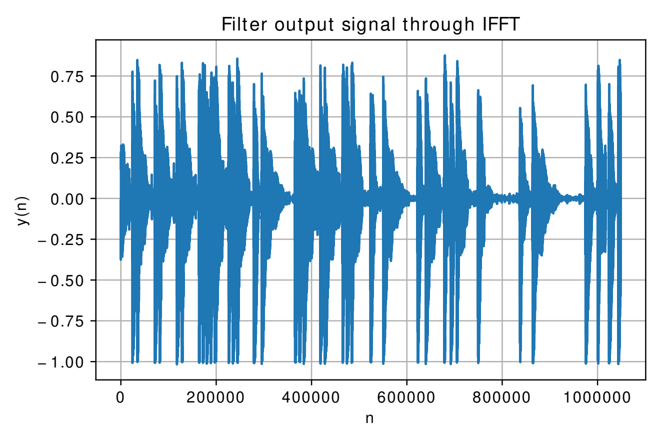

Fig. 3.2: $y(n)$ from IFFT

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