Assignment-1C

BANDI SAI LAXMAN - EE18BTECH11049

1 Software Installation

Run the following commands

```
sudo apt−get update
sudo apt−get install libffi−dev libsndfile1 python3
    −scipy python3−numpy python3−matplotlib
sudo pip install cffi pysoundfile
```

Make sure gcc is Installed , compile and execute using below commands

```
gcc file.c −o file −lm
./file
```

2 Digital Filter

2.1 Download the sound file from

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

2.2 Generate .dat file From the above sound file

```
codes/generate_dat.py
```

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

```
import soundfile as sf
from scipy import signal

#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
    )
print("Coeffients are as follows")
print(b)
print(a)

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

3 Difference equation

3.1 Write the difference equation of the above Digital filter obtained in problem **??**.
Solution:

$$\sum_{m=0}^{M} a(m)\, y(n-m) = \sum_{k=0}^{N} b(k)\, x(n-k) \tag{3.0.1}$$

$$\begin{aligned} y(n) - 2.52y(n-1) + 2.56y(n-2) - 1.206y(n-3) \\ +0.22013y(n-4) = 0.00345x(n) + 0.0138x(n-1) \\ +0.020725x(n-2) + 0.0138x(n-3) + 0.00345x(n-4) \end{aligned} \tag{3.0.2}$$

4 Z-transform

4.1 Find fourier Transform of $x(n)$ and $h(n)$ obtained using difference equation in problem **??** using C .
Solution:
$x(n)$ = audio signal input
$h(n)$ = filter.

Convolution : $y(n) = x(n) * h(n)$,
Taking Fourier Transform, we get:
$Y(K) = H(K)X(K)$

$x(n)$ will be generated from Sound_Noise.wav. $h(n)$ is generated using the equation in problem **??** .
To obtain the Fourier transform in $O(nlogn)$, we break the N point DFT into 2 - $N/2$ point DFT recursively.
The code to find the Fourier Transform of $x(n)$ and $h(n)$ can be found here. Generates dat files for X , H .

codes/fft_x_h.c

generating FFT plots of x(n) and h(n)

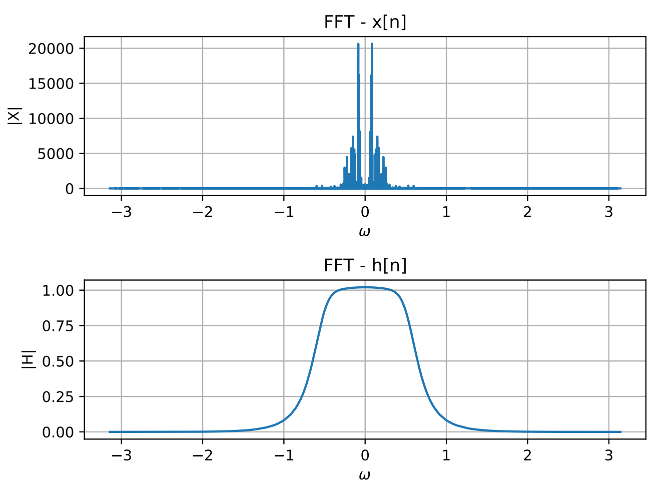

Fig. 4.1: FFT of x(n) and h(n)

Verification:
we obtain the Inverse Fourier Transform of Y(K) to obtain y(n) and the filtered sound.

codes/ifft_Y.c

It looks similar to the plots generated from the in built function of IFFT.

The code to obtain these plots.

codes/plotX_H.py codes/plot_y.py

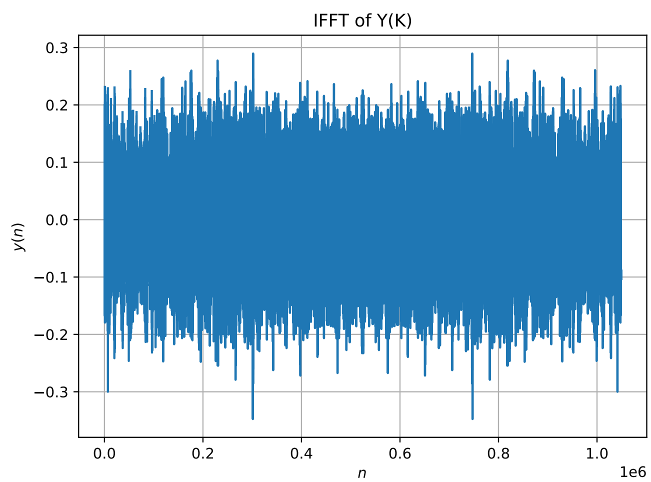

Fig. 4.1: IFFT - Y(K)