Audio Filter

EE23BTECH11008 - Annapureddy Siva Meenakshi*

.1 The sound file used for this code is obtained from the below link

```
Put later!!
```

.2 A Python Code is written to achieve Audio Filtering

```
import soundfile as sf
from scipy import signal

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

# Sampling frequency of Input signal
sampl_freq = fs

# Order of the filter
order = 4

# Cutoff frequency
cutoff_freq = 1000.0

# Digital frequency
Wn = 2 * cutoff_freq / sampl_freq

# Butterworth filter coefficients
b, a = signal.butter(order, Wn, 'low')

# Filter the input signal with Butterworth
    filter using filtfilt for zero−phase filtering
output_signal = signal.filtfilt(b, a,
    input_signal)

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

The audio file is analyzed using spectrogram using the online platform https://academo.org/demos/spectrum-analyzer.

The darker areas are those where the frequencies have very low intensities, and the orange and yellow areas represent frequencies that have high intensities in the sound.

Fig. 1. Spectrogram of the audio file before Filtering

Fig. 2. Spectrogram of the audio file after Filtering

I. Difference Equation

I.1 Let

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

Sketch $x(n)$.

I.2 Let

$$y(n) + \frac{1}{2}y(n-1) = x(n) + x(n-2), \quad y(n) = 0, n < 0 \tag{2}$$

Sketch $y(n)$.
Solve

Solution: The C code calculates $y(n)$ and generates values in a text file.

```
Put later!!
```

The following code plots (**??**) and (**??**)

```
Put later!!
```