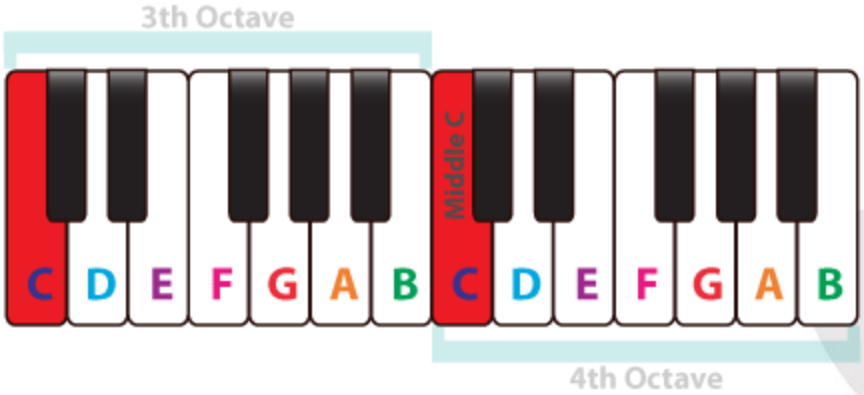
Faculty of Information Engineering and Technology
Department of Communication Engineering

Prof. Dr. Ahmed El-Mahdy

Signals and Systems Theory (COMM401)

Lab Project (PIANO)-Milestone 2
Noise Cancelation

SPYDER



GUC
German University in Cairo
الجامعة الألمانية بالقاهرة

Background

- Noise generation:

Assume that additional noise is generated by a child who kept pressing the same 2 random piano keys during your song playing interval such that the noise is represented as follows,

$$n(t) = \sin(2f_{n_1}\pi t) + \sin(2f_{n_2}\pi t)$$

Where, f_{n_1} and f_{n_2} are two frequencies selected randomly from the 3rd and 4th octaves frequencies list respectively. Your final song after the noise becomes,

$$x_n(t) = x(t) + n(t)$$

Goal: Noise Cancelation: using frequency domain conversion

Milestone 2: Frequency Conversion

- Procedure:

0) You will need to add this line at the beginning of your code:

```
from scipy.fftpack import fft
```

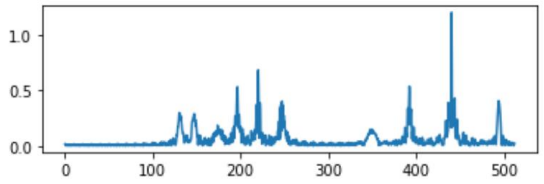
1) Set the number of samples N to $your\_song\_duration$ x 1024. Then set the frequency axis range to,

$$N = 3\text{x}1024$$
$$f = \text{np.linspace}(0\ , 512\ , \text{int}(N/2))$$

2) Convert the time signal $x(t)$ to the frequency signal $X(f)$ from,

$$x\_f = \text{fft(x)}$$
$$x\_f = 2/\text{N} * \text{np.abs}(x\_f[0{:}\text{np.int(N/2)}])$$

Your final output will look as following,



GUC German University in Cairo
الجامعة الألمانية بالقاهرة

Milestone 2: Noise Generation

- Procedure:

3) Generate the noise signal as described in slide 3 where the two random frequencies are selected as follows,

$$f_{n_1} \,\&\, f_{n_2} = \text{np.random.randint}(0, 512, 2)$$

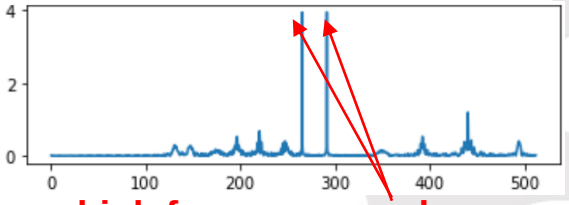
Hint: Each time you run the code you will get different noise.

4) Add the generated noise $n(t)$ to the signal $x(t)$,

$$x_n(t) = x(t) + n(t)$$

Hint: Play the new signal to notice the noise effect on the sound.

5) Convert the noise contaminated signal $x_n(t)$ to the signal $X_n(f)$ in frequency domain, it looks as follows,



Hint: You will notice that the signal contains **two very high frequency peaks**

Milestone 2: Noise Cancelation

- Procedure:

6) Find the two random noise frequencies: by finding the frequency indices that corresponds to the peaks of the signal $X_n(f)$ which is higher than the maximum peak of the original signal $X(f)$ without noise rounded to the next integer (to avoid including the maximum value due to double precision errors).

7) Round the frequency values at that indices in f to get $\hat{f}_{n_1}$ and $\hat{f}_{n_2}$ (since we generate random integer values of frequencies)

8) Filter the noise by subtracting two tones with the two found frequencies

$$x_{filtered}(t) = x_n(t) - [\sin(2f_{n_1}\pi t) + \sin(2f_{n_2}\pi t)]$$

9) Play the filtered signal to make sure that you get the same original sound without noise

$\text{sd.play}(x_{filtered}, 3 * 1024)$

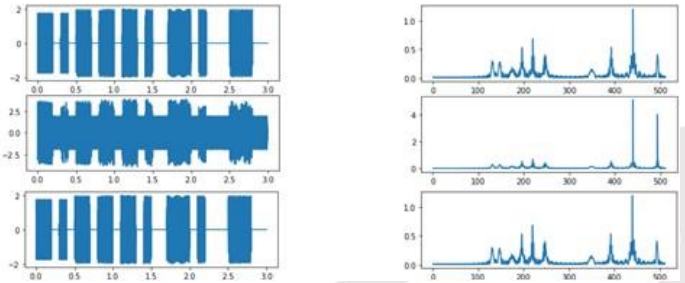
Project Submission

- Copied reports or code means **zero** for both versions !

- At the end of this milestone you should submit a zip folder that contains:

Code(s): Your Python script/s of both of milestone 1 and 2 together

Report (Word Typed): Update your report with a brief description of the carried out steps followed by the output figure/s obtained (Hence: You should have six plots divided into two figures, the first figure contains the time domain signal without and with noise, and after noise cancellation. And the second figure contains the frequency domain signal without and with noise and after noise cancellation.)



- Submissions will be e-mailed to your TA through:

Eng. Omnia Mahmoud: signalslab2022.omnia@gmail.com
Eng. Maggie Shammaa: signalslab2022.maggie@gmail.com
Eng. Randa El-Khosht: signalslab2022.randa@gmail.com
Eng. Sarah Azzam: signalslab2022.sarah@gmail.com
Eng. Maha El-Feshawy: signalslab2022.maha@gmail.com

GUC
German University in Cairo
الجامعة الألمانية بالقاهرة

GUC
German University in Cairo
الجامعة الألمانية بالقاهرة