Task 3 – Equalizer

Introduction: Signal equalizer is a basic tool in the music and speech industry. It also serves in several biomedical applications like hearing aid abnormalities detection.

Description: Develop a web application that can open a signal and allow the user to change the magnitude of some frequency components via sliders and reconstruct back the signal.

- The application should be able to work in different modes:
  1. **Uniform Range Mode:** where the total frequency range of the input signal is divided uniformly into 10 equal ranges of frequencies, each is controlled by one slider in the UI.
  2. **Vowels Mode:** where each slider can control the magnitude of specific vowel. You will need to do your own research for how the speech vowels are composed of different frequency components/ranges.
  3. **Musical Instruments Mode:** where each slider can control the magnitude of a specific musical instrument in the input music signal.
  4. **Biological Signal Abnormalities:** where each slider can control the magnitude of a specific abnormality (e.g. ECG arrhythmia) in the input biological signal.
- The user can switch between the modes easily (probably through an option menu or combobox). The UI is not expected to change dramatically among the different modes. The main expected change in UI when changing from one mode to the other is the change in the sliders captions and maybe the number of sliders.
- Beside the sliders, the UI should contain:
  - Two signal viewers, one for the input and one for the output signals. The two viewers should allow the signals to run in time in a synchronous way (i.e. the two viewers have to be exactly linked. They should always show the same exact time-part of the signal if the user scroll or zoom on any one of them).
    - There should be a play/stop/pause/speed-control panel to control the playing of the signals.
    - Note that your signal viewer should take care of the boundary conditions. Intuitively, no scroll should be allowed before your signal starts or after it ends or above its maximum values or below its minimum values.
  - Two spectrograms, one for the input and one for the output signals. Upon any change in any of the equalizer sliders, the output spectrogram should reflect the performed changes.
    - There should be an option to toggle show/hide of the spectrograms.
- To validate your work, each group should prepare a synthetic signal file. The signal is an artificial signal that you should prepare, and it is composed of a summation of several pure single frequencies. This should help in tracking what happens to each frequency when an equalizer action is taken. The decision of what are the frequencies to include and how many of them is left to the judgment and brainstorming of each group. Always remember that the main aim of this file is to validate if your equalizer is acting correctly or not.

Code practice:

1. Same practices from Task 1 & 2 (i.e. proper variable and functions names) will continue with task 3.
2. Avoid code repetition! If you notice similar lines in your code with only few numbers or variables different, then these lines are very good candidates for a “function”! Code repetition is a very bad practice and will be penalized (each repetition will be penalized with -5%) from now on in the course tasks evaluation.