Time and frequency domain analysis of sound

Huminiuc Simona
University of "Gheorghe Asachi" Iasi
Faculty of Automatic Control and Computer Engineering

1. Project description

By using the LabWindows CVI 2020 development environment, the aim of the project is to improve knowledge of signal acquisition and processing. Data from the audio file is analysed in time and frequency domain. Time domain analysis will include calculating minimum, maximum, minimum and maximum index, dispersion, median, median, sampling frequency and the zero passing. In addition, two filters are used, the first-order element filter and the averaging filter. Analysis in frequency domain involves the application of two types of windows (Blackman and Welch) and two types of filters (EquiRipple FIR and Elliptic bandpass 900-1500H) on one second of the signal.

2. Description of project requirements

In the first stage of the project, a python script is used in order to graph the .wav file that will be used to analyse in time domain and in frequency. After finalising the graphical representation of the sound and its histogram are displayed on the graphic interface, the time domain parameters: maximum and minimum values, mean, dispersion, median, number of zero crossings, skewness and kurtosis. After this, we can perform filtering by applying the averaging filter on 16 or 32 elements or with a first order element filter. The filtered signal is displayed one second at the time on the graphic and on the non-filtered one can apply his envelope.

In the second stage of the project a second panel is introduced to represent the signal. The analysis of each second is also performed. On these new graphs are used two types of windows ((Blackman and Welch) and two types of filters (EquiRipple FIR and Elliptic bandpass 900-1500 Hz).

All these graphs described above can be saved as .jpg images. The LabWindows CVI 2020 and Python 3.11 development environment was used to make this project.

3. Time domain analysis

In the first phase, the time domain analysis is performed using the Python script. This script converts the audio file 33.wav into two files waveData.txt and waveInfo.txt. These files contain information about the sampling rate and the number of values of the audio signal. First, the signal is

displayed on a graph control on the WavePanel, then its histogram and the rest of the data extracted from the signal such as the maximum and minimum index, the median dispersion and the number of zero crossings.

The filtering functions used are the following: the 16 or 32-element averaging filter and the first order filter according to the relation: filt[i]=(1-alpha)*filt[i-1]+alpha*signal[i]. In this case, filt is a vector containing the filtered values, alpha is a coefficient in the range (0, 1), and signal contains the values of our audio signal. The filter type, and the required input data can be selected using the buttons arranged on the interface.
The filtered signal can be seen for one second only, and the Prev and Next buttons help us to skip to the previous or next second.

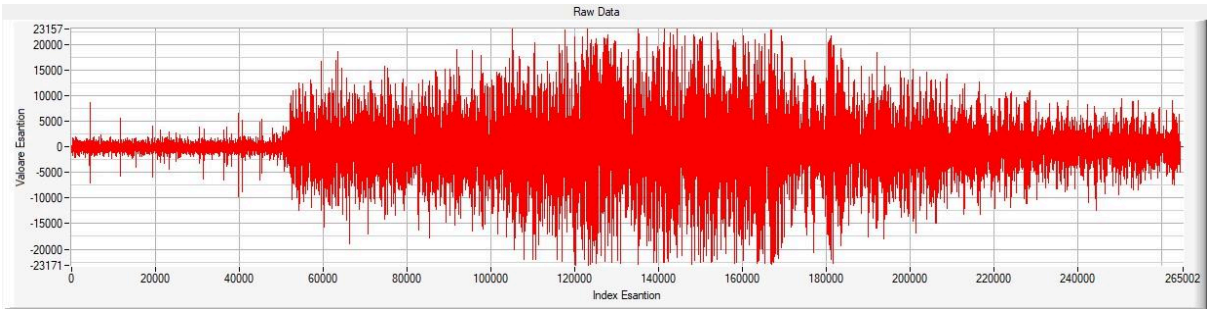

Fig 1. Original non-filtered signal

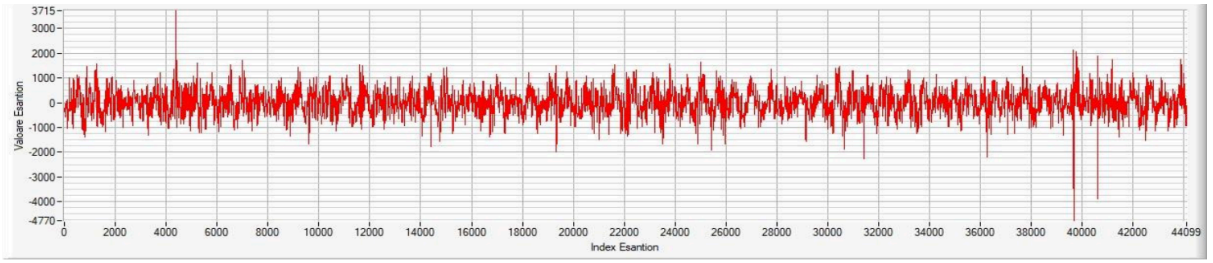

Fig 2. Averaging filter on 16 elements, seconds 0-1

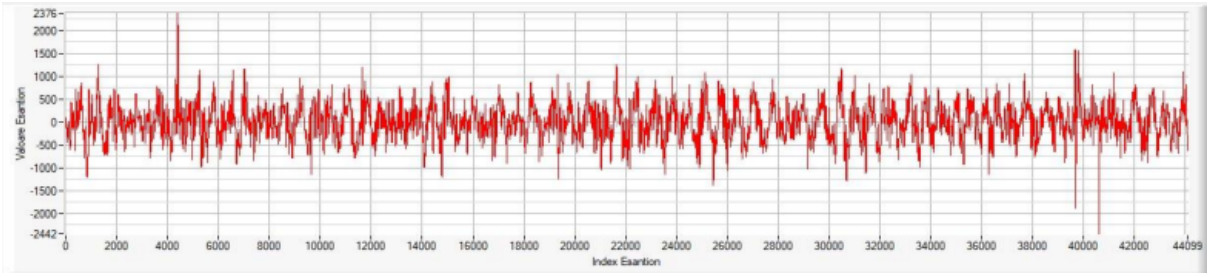

Fig 3. First order filter with alpha = 0.05, seconds 0-1

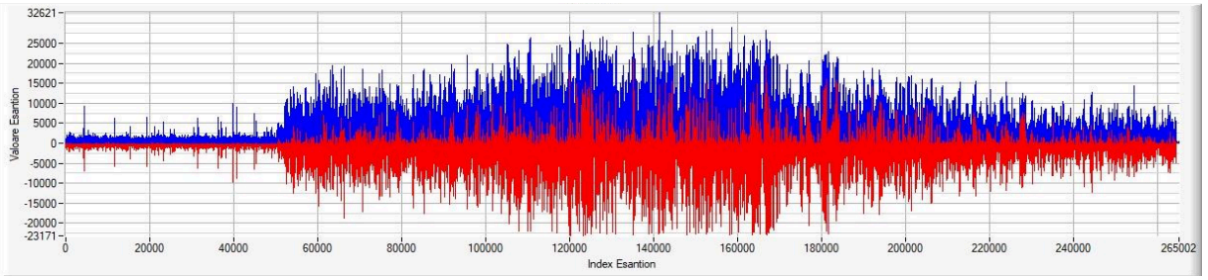

Fig 4. Original signal with envelope

4. Frequency domain analysis

In the second part of the project a new panel called FrequncyPanel is being made which will allow to display the spectrum for the whole sound, as well as for each individual second. On the original signal first, two types of separate windowing are applied which "flatten" the signal at the ends for each second of the signal, the Blackman-Harris and Welch window. Since the initial signal had a lot of points and a duration of about 6 seconds, we proceeded as follows: we represented each second of it separately for a more detailed visualisation. In this context two types of filters were also applied: Equiripple FIR and Ecliptic band-pass 900-1500 Hz.

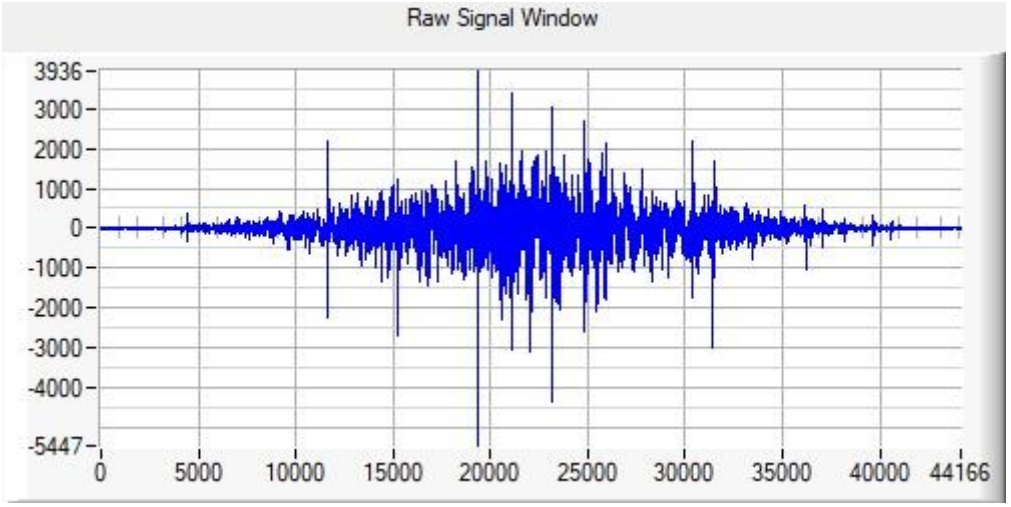

Fig 5. Original signal with Blackman-Harris windowing

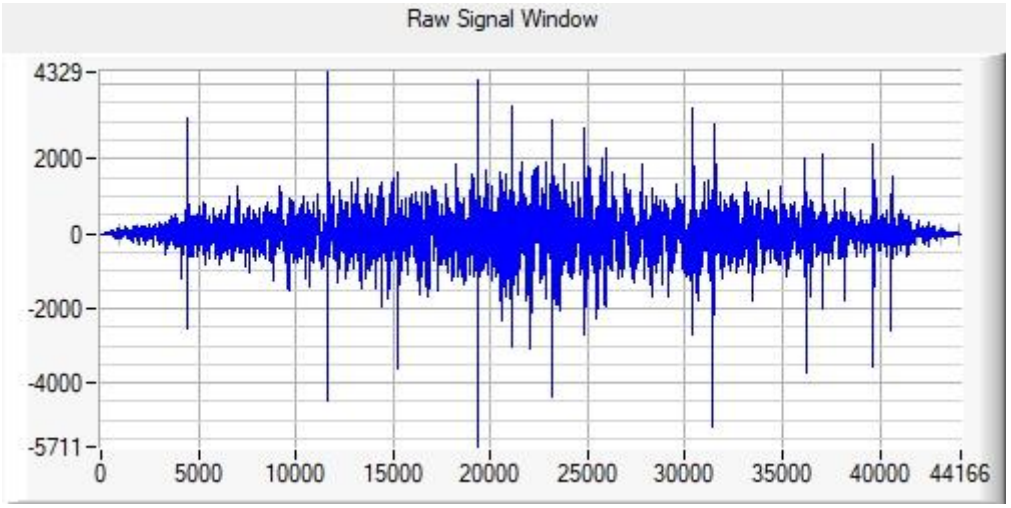

Fig 6. Original signal with Welch windowing

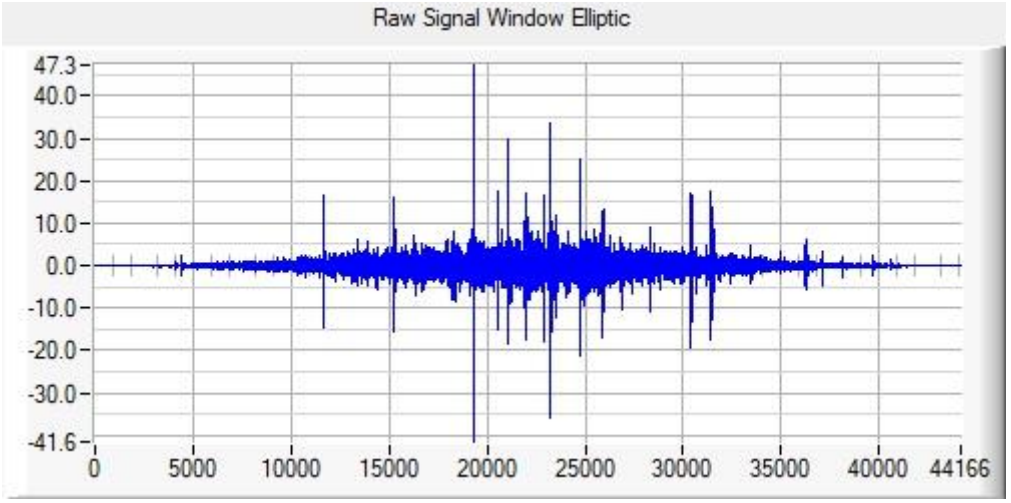

Fig 7. Elliptic filter and Blackman-Harris windowing

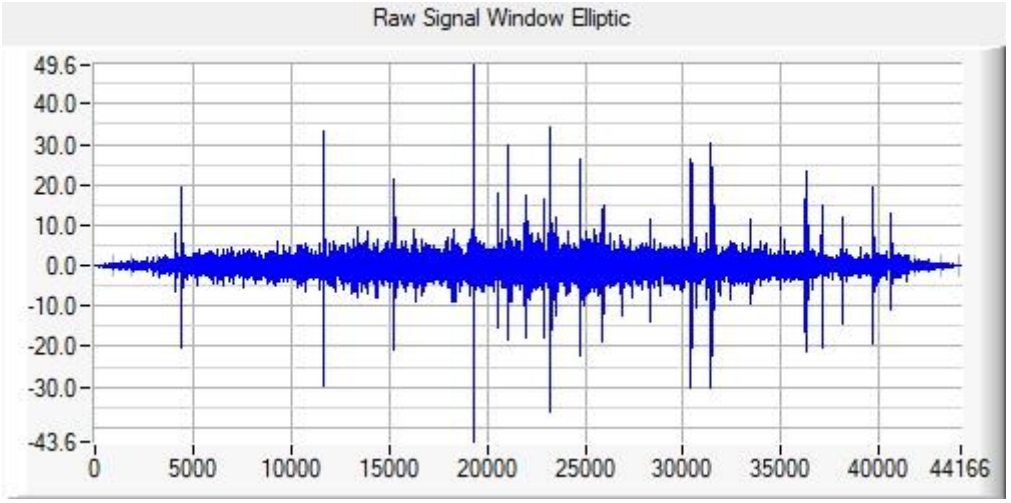

Fig 8. Elliptic filter and Welch windowing

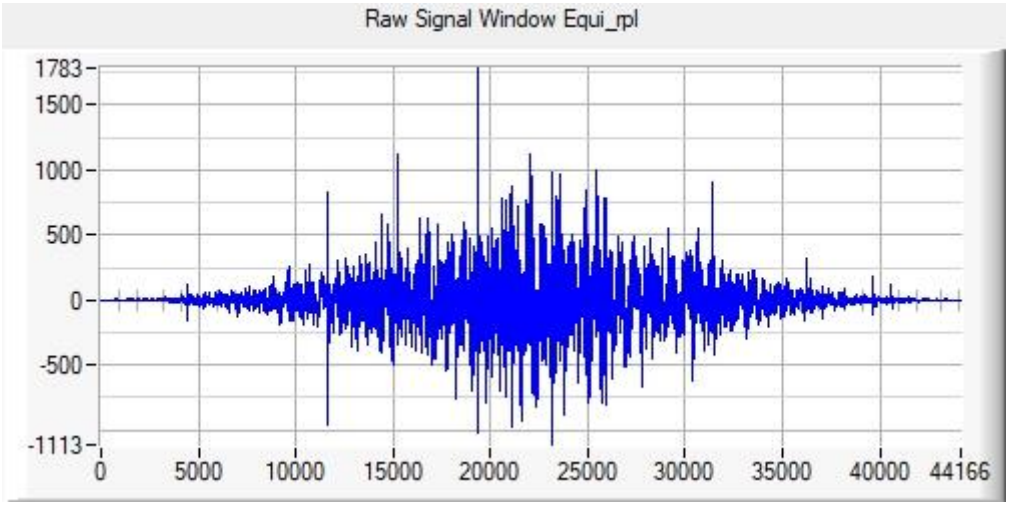

Fig 9. EquiRipple filter with Blackman-Harris windowing

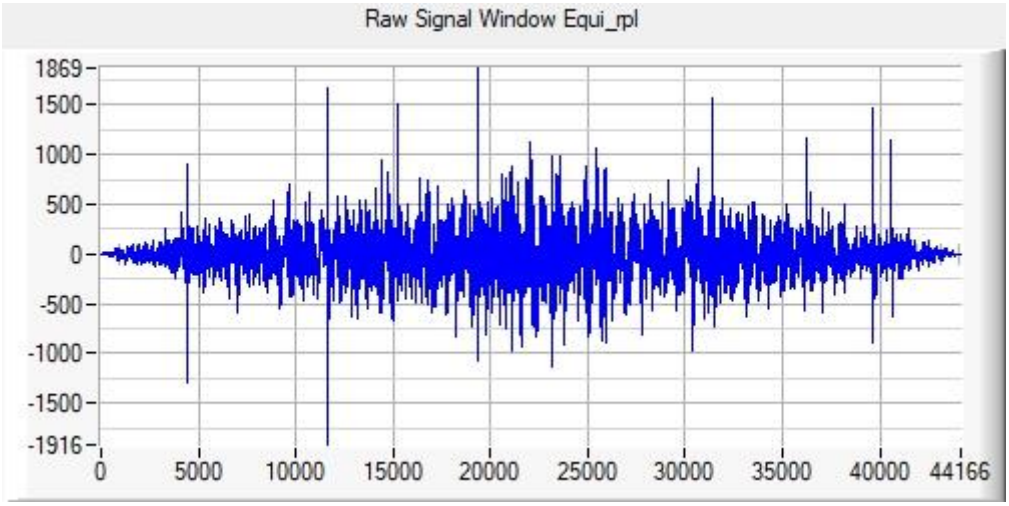

Fig 10. EquiRipple filter with Welch windowing