FACULTÉ DES SCIENCES APPLIQUÉES
Ecole d'ingénieurs et d'informaticiens

ELEN0071
Applied digital signal processing
P. Sacré - P. Embrahimbababaie

Homework 2 : Noise elimination

Beaulieu Guillaume
Custinne François
Escalona Saul

3^{rd} BAC Civil Engineering
Faculty of Applied Sciences

Academic years 2018-2019

During this laboratory session, we talked about filtering signal noises. By this way, we tried to filter a given noise-corrupted electrocardiogram signal (This signal will be called *ecg* in the following sections) and recover the original signal one.
In the following section, we'll talk about the different filtering process steps and we'll answer the several questions asked.

1 Noise-corrupted signal plot

Firstly, we had to plot the Noise-corrupted *ecg* signal.
We knew that the sampling frequency is $F_s = 250Hz$ and then, easily, we deduced the sampling period like $T_s = \frac{1}{F_s} = 0.004s$. After that, we used the function `length` from MATLAB to determine N, the length of the signal, and then, computed the ending time T_{max} like :

$$T_{max} = (N-1) \cdot T_s = 29.996 \ sec$$

This *ecg* signal is plotted as a function of time t. Thus, because the vector t should have the same length of the given signal *ecg*, it begins from 0 and ends to T_{max} with a step of T_s. It is represented in Figure 1 below.

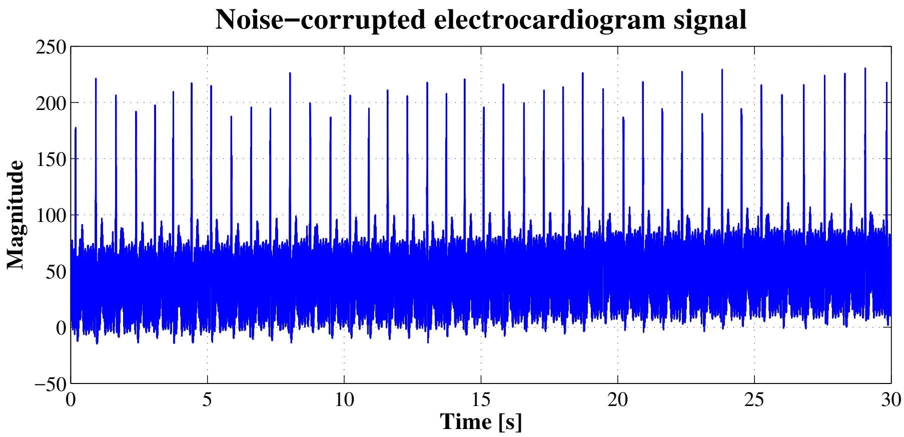

FIGURE 1 – Given noise-Corrupted electrocardiogram signal.

2 Zoom in the noise-corrupted signal plot

Secondly, we had to plot the noise-corrupted signal between 2 and 5 seconds. We used the function `xlim` predefined in MATLAB to plot the only 3 seconds wanted. This plot is shown in Figure 2 below.

3 Single-sided magnitude spectrum of the signal

This step aims to determine the noise frequencies of our noise-corrupted signal. In order to do that, we had to work in the frequency domain. By this way, we computed the Fast Fourier Transform (FFT) of our noise-corrupted signal. Then, we scaled this FFT, took its absolute value and kept only the first half of the result to plot the single-sided magnitude spectrum asked. In the same way, to fully offset this last step, we multiplied all the values of the signal except the DC component by 2. This Single-sided magnitude spectrum is shown in Figure 3.

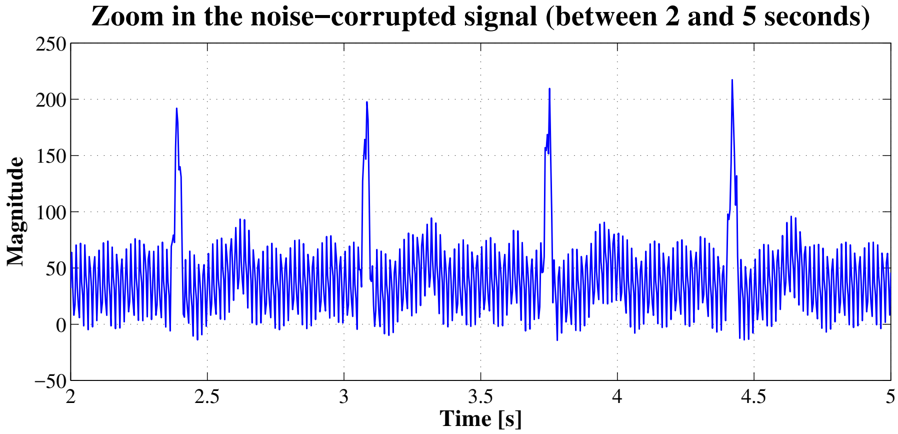

FIGURE 2 – 3 seconds of the given signal (from 2 to 5 seconds).

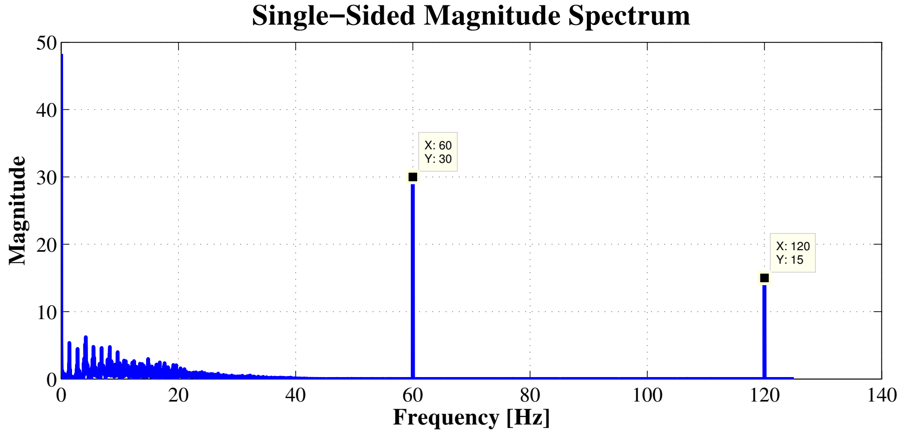

FIGURE 3 – Single-sided magnitude spectrum of the given signal

4 Identification of noise frequencies

Thanks to the single-sided magnitude spectrum plotted in the previous section, we could determine the noise frequencies. As it can be seen in Figure 3, there are two peaks in this signal. The first one occurs at 60 Hz and the second one occurs at 120 Hz. These two peaks correspond to the two main noise frequencies which we had to consider to filtered the noise-corrupted signal and recovered the original one.

5 Recovering of the original signal

By using filters on the two noise frequencies found above, we could recover the original electrocardiogram signal "without" any noise. This signal is shown in Figure 4 below.

In order to compare this final signal to the given one, we plotted both these two signals in Figure 5. Moreover, this Figure 5 represents a 3 seconds zoom (from 2 to 5 seconds) of both signal.
We can see in this figure that the noise has been removed thanks to the filter design procedure (this procedure is described in the next section at section 6).

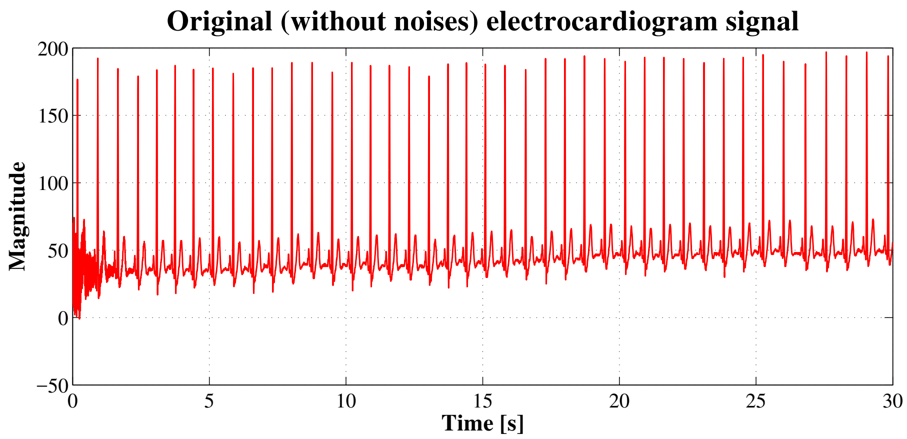

FIGURE 4 – Filtered electrocardiogram signal

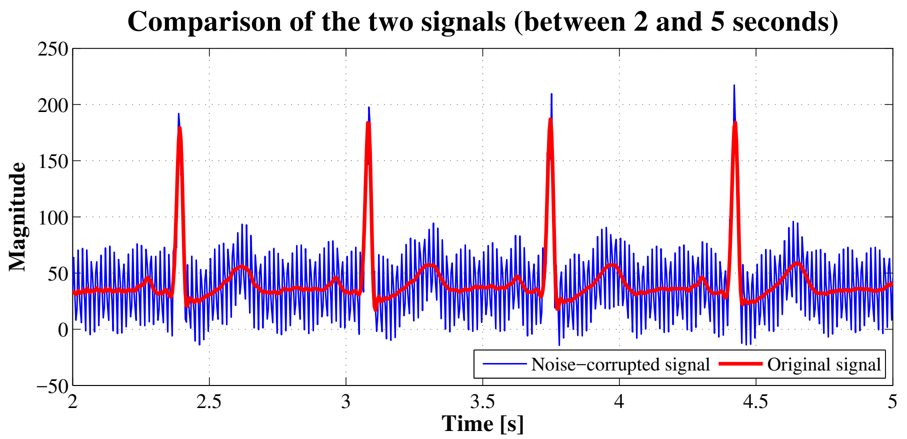

FIGURE 5 – Original and noise-corrupted electrocardiogram signals

6 Filter design procedure

In order to recover the original electrocardiogram signal based on the noise-corrupted given one, we had to filter this last one. In this way, we firstly found the two noise frequencies at 60 and 120 Hz. Secondly, we designed two filters in `Matlab - Apps - Filter design` window. These two filters are characterized by a notching response type with a single notch

$$\text{with} \quad \begin{array}{lcll} F_{notch} & = & 60 \text{ Hz} & \text{for the first filter} \\ F_{notch} & = & 120 \text{ Hz} & \text{for the second filter} \end{array}$$

Thus, we filtered our noise-corrupted signal by using these two filters in cascade. This procedure scheme is given in Figure 6 below. So, by passing the given signal through the two filters in serial we recovered the original one.

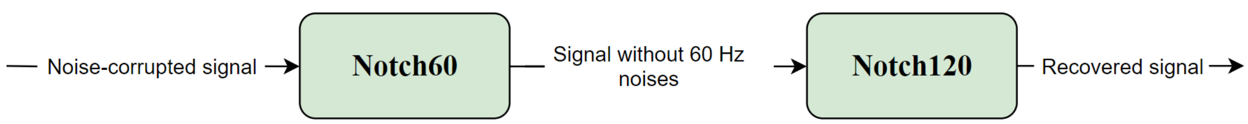

FIGURE 6 – Filtering procedure scheme