Project 1 Report

Digital Signal Processing

Ahmed Jaheen - 900212943
Ahmed Amr - 900202813
Amna El-Saqa - 900202552

Submitted to Dr. Seif Eldawlatly
This report is prepared for CSCE3611, section 1

Computer Science and Engineering Department
The American University in Cairo
16/07/2023

Description of the approach used

Firstly, we input the full signal from the file **"data.txt"**. Next, we rectify the signal by taking the absolute value of each data point. Then, we employ a moving average filter using the convolution embedded function in Python to smoothen the signal's curve. Subsequently, we analyze the graph obtained from the moving average filter and visually identify the noise component. We calculate the standard deviation of this noise part manually, which yields a value of approximately 11.7.

Having obtained the filtered signal, we proceed to detect the first occurrence of a MUAP by identifying the first sample that surpasses the threshold. To ensure that's an MUAP, we iterate over the next $T-1$ (19 samples) and if all of them are above the threshold we register it as a MUAP. Next, we continue the iteration and if, at $T+1$, the signal remains above the threshold, we consider it as a continuation and skip another T samples (20 samples). If, at any iteration, the sample falls below the threshold, we conclude the MUAP and begin detecting another one by looping through the entire signal with a complexity of $O(nm)$, where n represents the number of samples and m is the number of T continuations. However, if for a sample which is above the threshold have any sample of the next $T-1$ below the threshold we conclude it as a noise.

Then, we successfully synchronized in a window of 25 samples. At the beginning, we tried with 20 samples and we got the correct peak in the filtered signal, and it was centralized. However, in the original signal, the peak wasn't accurate, and there was a problem because of a shifting in the peak as the first occurrence in the filtered signal doesn't have to be the first occurrence in the original signal because we are using a moving average filter. So, we increased T to be 25 until we found the peak, and then subtract $10(T/2)$ samples to determine the index of the first occurrence in the original signal.

With the preparations complete, we can now compute the templates. The first MUAP always serves as a template, which we store in a list of lists. Subsequently, we consider the second MUAP and compare it with the first template. If the difference between them falls below a threshold diff value, we treat it as the same template and calculate the average between the two MUAPs. Conversely, if the difference exceeds the threshold, we consider it a new template. We repeat this process until we have generated 5 templates.

The initial output comprised five templates; however, upon closer inspection, we identified two duplicates among them, which were already present. To rectify this, we took the

approach of averaging these redundant templates with their corresponding original signals. Subsequently, we proceeded to plot the resulting three unique signals alongside the outputs specified in the subsequent section.

Outputs of the project

1. A figure showing from sample 30000 to sample 35000 of the EMG signal with an $*$ marking the detected MUAPs colored with different colors depending on the template each MUAP belongs to.

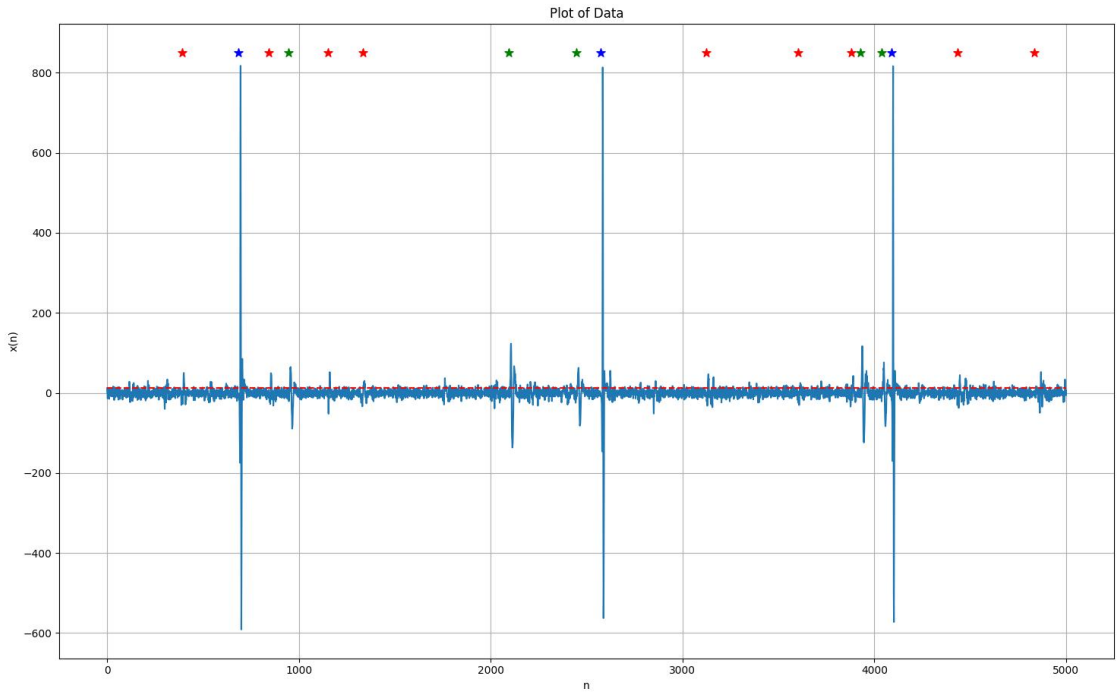

Figure1: "Detected MUAP"

2. A figure showing the waveform of the first template of the detected templates.

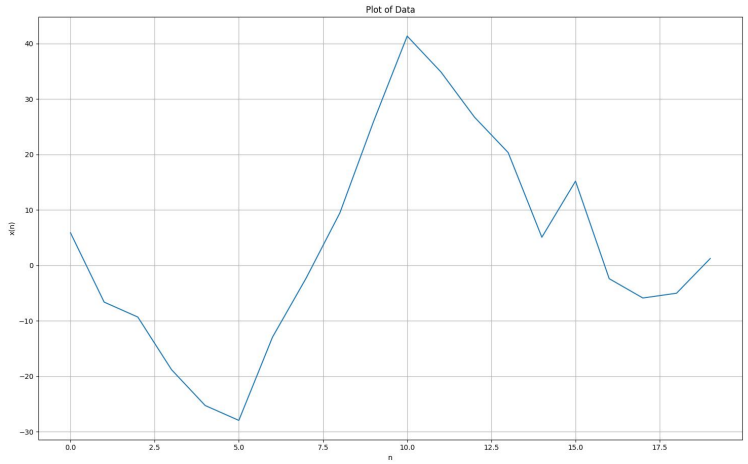

Figure2: "Template 1"

3. A figure showing the waveform of the second template of the detected templates.

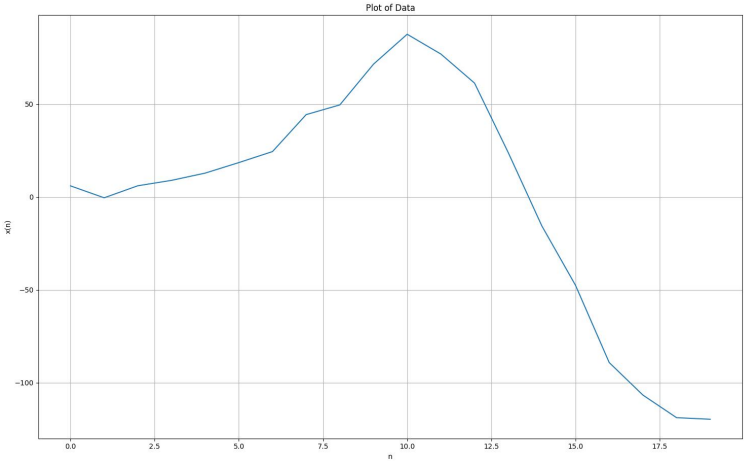

Figure3: "Template 2"

4. A figure showing the waveform of the third template of the detected templates.

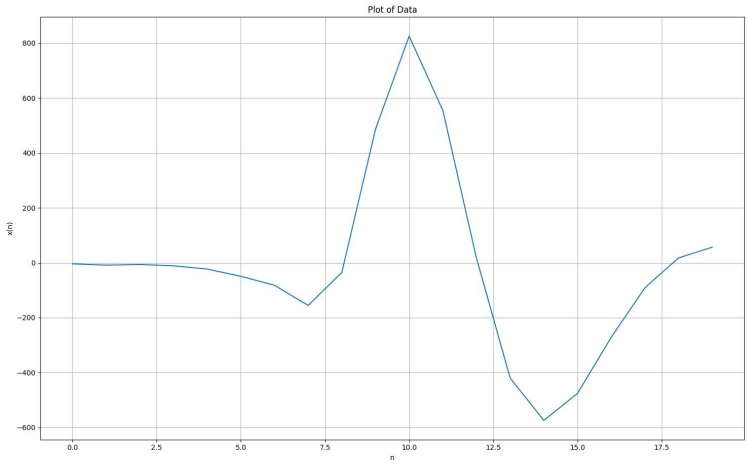

Figure4: "Template 3"

5. This figure represents the spectrum for template 1.

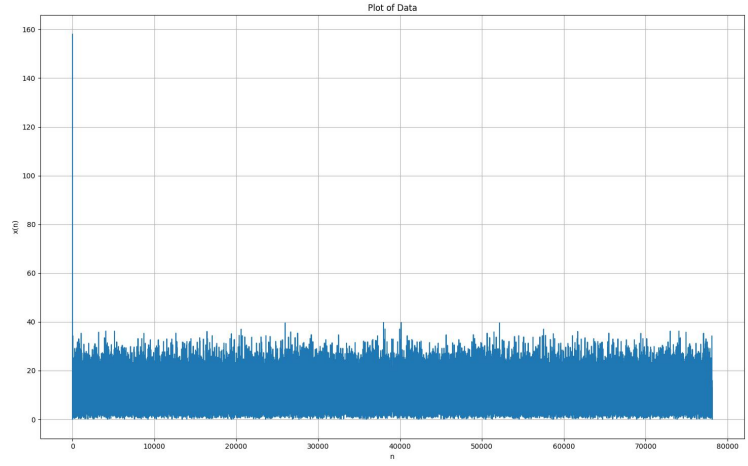

Figure5: Spectrum 1

6. This figure represents the spectrum for template 2.

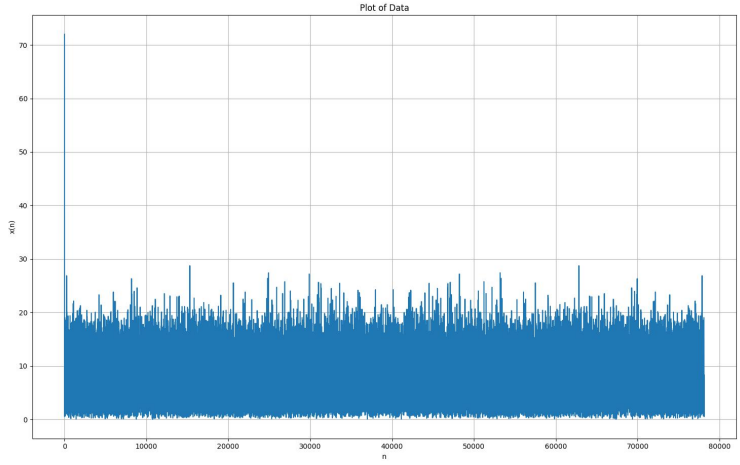

Figure6: Spectrum 2

7. This figure represents the spectrum for template 3.

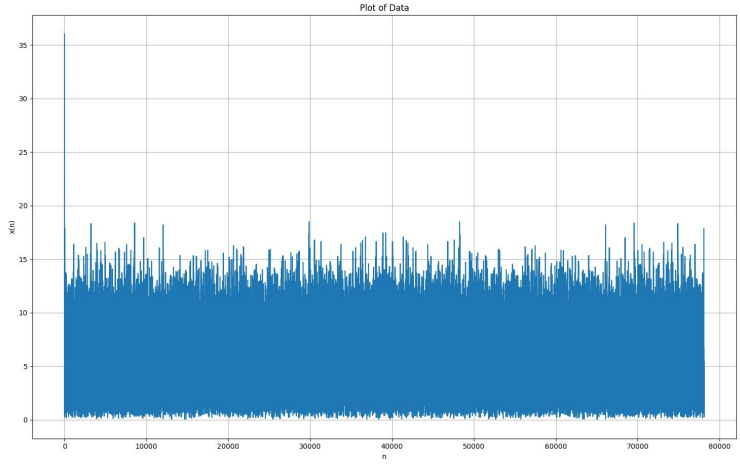

Figure7: Spectrum 3