DIGITAL SIGNAL PROCESSING

PROJECT

Group Members
Garimella Annapurna Gayathri (IMT2013015)
Sneha A. (IMT2013045)
Trisha Mittal (IMT2013053)

Due: November 9, 2015

Objective: ECG signal analysis using data obtained from Wipro AssureHealth device. To remove the powerline and baseline interference and then proceed to r-peak detection.

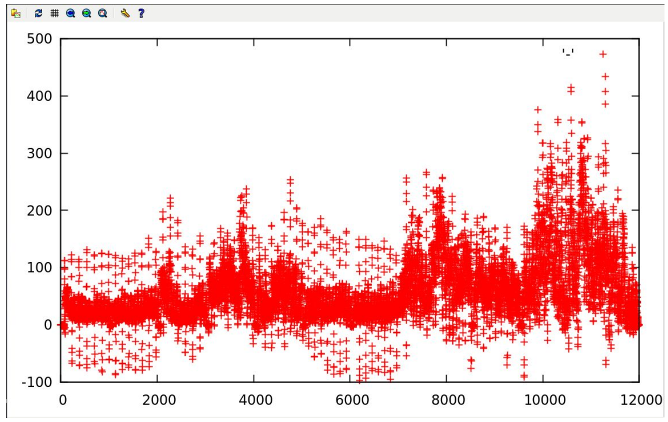
PART 1: Removal of Powerline and Baseline Interference

Electrocardiogram (ECG) reflect the electrical activity of the hearts and remain the most important tool for heart disease diagnoses .However the ECG signal are corrupted by different types of noises and artifacts such as power- line noise (50/60 Hz), baseline wander, motion artifacts, muscle contraction and other external noises so the removal or reducing the effect of these noise and mainly power-line and baseline wander without affecting the embedded parameters of ECG signal will play a vital role to get clean ECG which in turn help in diagnoses.

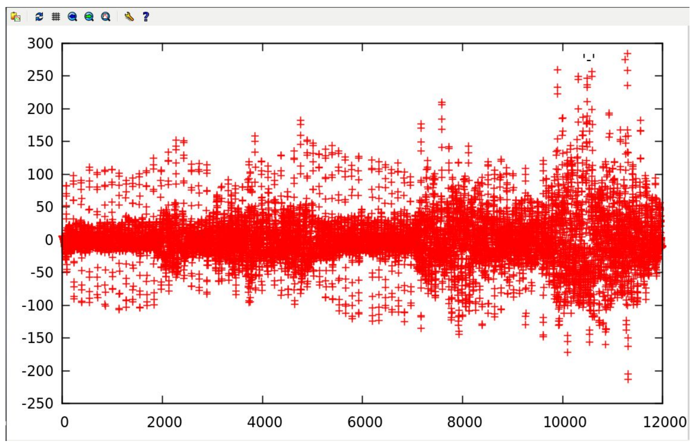
The first attempt to remove Powerline interference was performed in Matlab. To remove Powerline interference as multiple papers have suggested, we applied a notch filter at 50Hz. When the signals before and after removing the Powerline interference were plotted on the same graph, the signals were same except a small delay. Also, this delay is a result of processing delay and not the effect of any filter. Hence, it was assumed that the signals had negligible Powerline Interference and was not implemented in C.

Baseline Interference removal was also implemented in Matlab first. The aim was to plot the frequency spectrum of the signal and then remove the values of lower frequencies as they contain the noise in the ECG signal. In Matlab, the fft and the ifft commands were exploited. When implemented in C, fft and ifft functions had to be written.

Another way to remove baseline wander interference was applying a simple moving average filter in the time domain. This was implemented directly in C, and worked well. The plots of graphs of baseline interference in C have been shown.

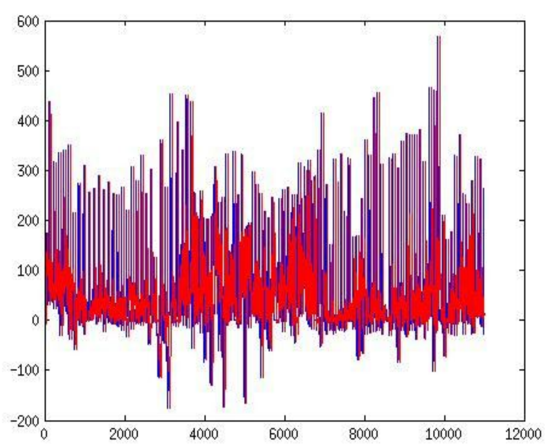


a. Before Baseline Removal (Plotted in Gnuplot)



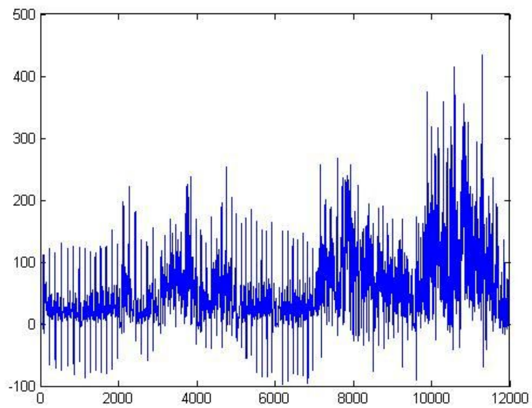
b. After Baseline Removal (Plotted in Gnuplot)

The plots of the Matlab code are as shown below.

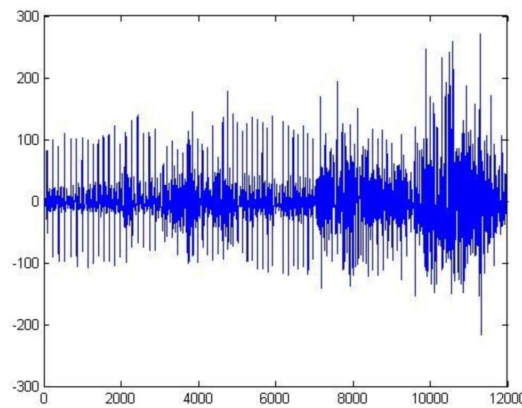


a. Powerline interference plotted in Matlab

Signal before and after removing powerline interference. The blue signal is the input and the red signal is after powerline removal.



a. Baseline Interference



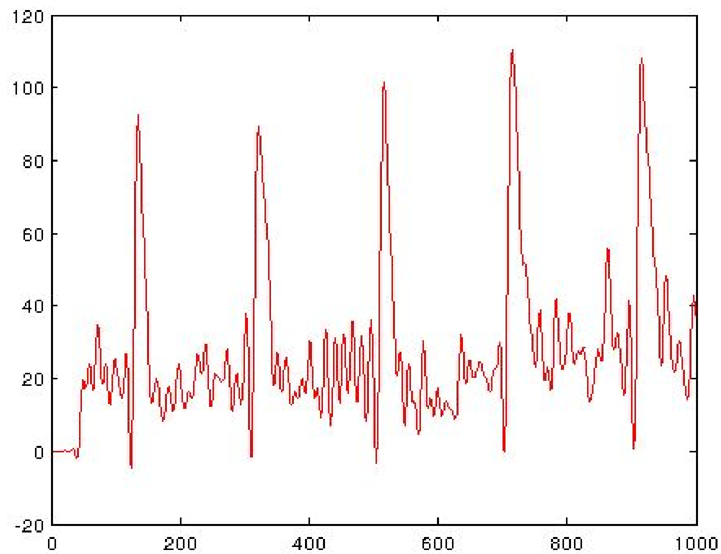
b. After Removing Interference

PART 2: R peak Detection

The next part of the project was to locate the samples associated with R-peak in the ECG signal. This section describes the strategy and the specific algorithm used for the task followed by the results.
Algorithm:
Input: a discrete time ecg signal with a given sampling frequency
Step 1: Double differentiate the signal
Step 2: Square the double differentiated signal
Step 3: Decide on a window size
Step 4: Window through the double differentiated signal and mark the highest valued sample in each window.
Output: sample instants corresponding to the r-peaks(determined in step 4)
This algorithm was adopted based on both reference to some papers(mentioned later) and some trial and error.

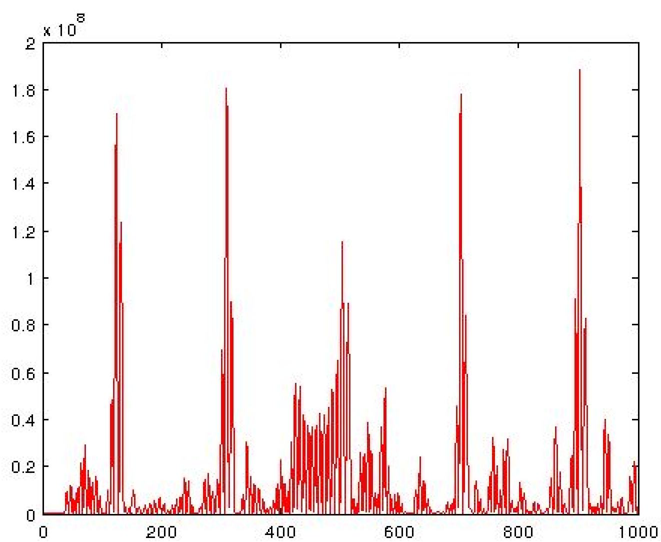
Rationale behind the algorithm:
The R-peak and the region around it(QRS complex) have high frequency content. Therefore the derivative of the ECG signal has high amplitude(due to the relatively sharper peak near the r-peak).
We take the second derivative and square to further enhance the amplitude(non-linearly due to the squaring) allowing better identification of the peak.
The next step was to identify the peaks. One alternative was to pick to maximum value from the ECG signal. Then pick all samples that fall within a threshold value from the maximum value. However, this step was not effective because some r-peaks were almost

half the size of others. And if the threshold was stretched to accommodate these r-peaks, other peaks also began to get identified.
Therefore we decided to estimate the average interval between r-peaks. We took 600 contiguous samples from various ECG recordings and made an estimate on the number of r-peaks found. On an average the distance between 2 r-peaks was found to be 150 samples for the given sampling frequency. We then processed the signal by looking at 150 samples at a time identifying the sample with highest value as the r-peak.



I

a. The figure is a plot of 1000 samples of ECG signal.



b. This is the plot of the 1000 samples after the double differentiation and squared.

This segment of the signal is critical to validate that some R-peaks have lesser amplitude than other peaks in the signal and justifies the window method.
The code was tested with all the sample data provided. We previously determined that a window size of 150 samples contained 1 heart beat. Based on the given sampling frequency of 250Hz, we calculate the number of heart per minute will be beats in 1 second as 250/150(=1.66). Subsequently, the number of heartbeats per minute will be around 100.

The following codes have been attached as a part of the report.

1. filter.c - This code deals with the processing of the raw_data to remove the baseline interference.
2. RpeakDetect.c - This code is for the detection of all the locations of all the R-peaks.
3. RpeakAlgo.m - Matlab code for both baseline removal and Rpeak detection.
4. build.sh - Script to compile both the C codes.
5. run.sh - Script to run the C codes.

References

1. Sadhukhan, Deboleena, and Madhuchhanda Mitra. "R-Peak detection algorithm for ECG using double difference and RR interval processing." *Procedia Technology* 4 (2012): 873-877.
2. Köhler, Bert-Uwe, Carsten Hennig, and Reinhold Orglmeister. "The principles of software QRS detection." *Engineering in Medicine and Biology Magazine, IEEE* 21, no. 1 (2002): 42-57.