Veer Gokhale 3160362G
Arda Callialp 2795083C

Assignment 2 Digital Signal Processing FIR Filters

1) Time and Frequency Analysis of the Recorded ECG

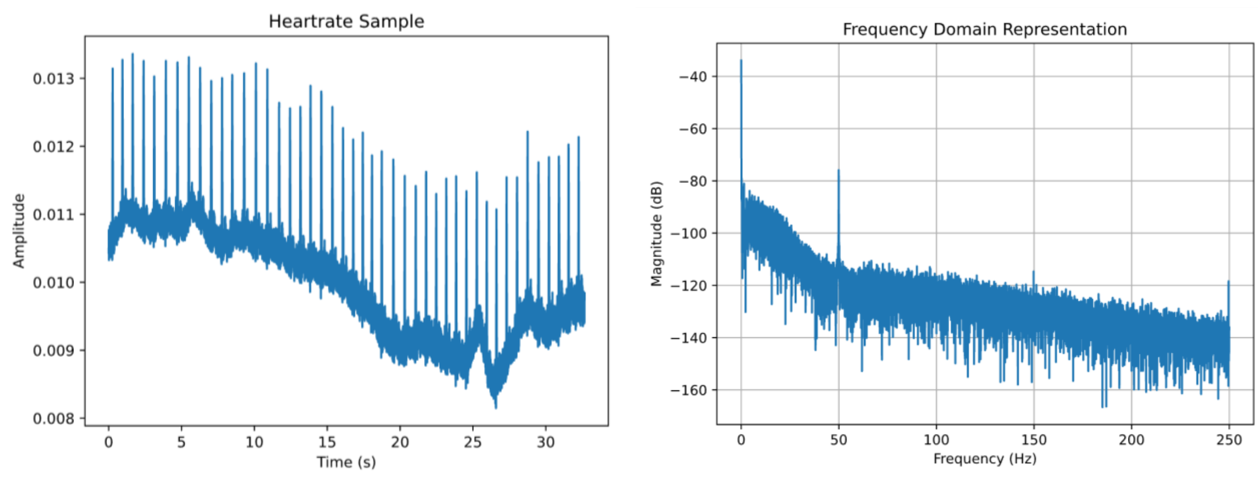

Figure 1: Time Domain ECG Signal and the Frequency Domain ECG Signal

We recorded ECG using the Attys setup with a 500Hz sampling rate. For the ECG recording we have put the electrodes on ankles and wrists while standing upright. The ECG is in column 7 of the TSV file. We have plotted the ECG trace both in the time domain and the frequency domain. When the frequency domain ECG signal in figure 1 is taken into consideration there is a high 50Hz noise, DC signal and a baseline wander. A heartbeat can be in the range of 45 beats per minute up to 180 beats per minute. So, the heartbeat itself is in the range of 0.7 Hz and 3Hz. Therefore, we considered removing baseline wander from 0Hz to 0.7Hz to not interfere with the heartbeat signal. The noise signals which will be removed are the DC signal, 0 to 0.7Hz to eliminate baseline wander, and 50Hz noise.

2) FIR Filter Design

For our FIR, we decided to use 5001 taps. This is because we have a sampling rate of 500 Hz, and we wanted a frequency resolution of 0.1 Hz, so that we could precisely implement a stopband of 0-0.7 Hz. The result of this decision was that we had a ~7 second delay in our filter before we got meaningful results.

We also applied a hamming windowing function to reduce ripples and sidelobes resulting from our filter.

3) Real Time FIR Filtering

For our FIR filter, we used stopbands of 0-0.7 Hz to remove low frequency noise, and 40 Hz - fs/2, to remove high frequency noise, including the 50 Hz noise. In the implementation of dofilter, we used a ring buffer, to maximize the speed of our realtime filter. Shown below is a plot of our filter's output, and a zoomed-in plot showing that PQRST integrity was maintained.

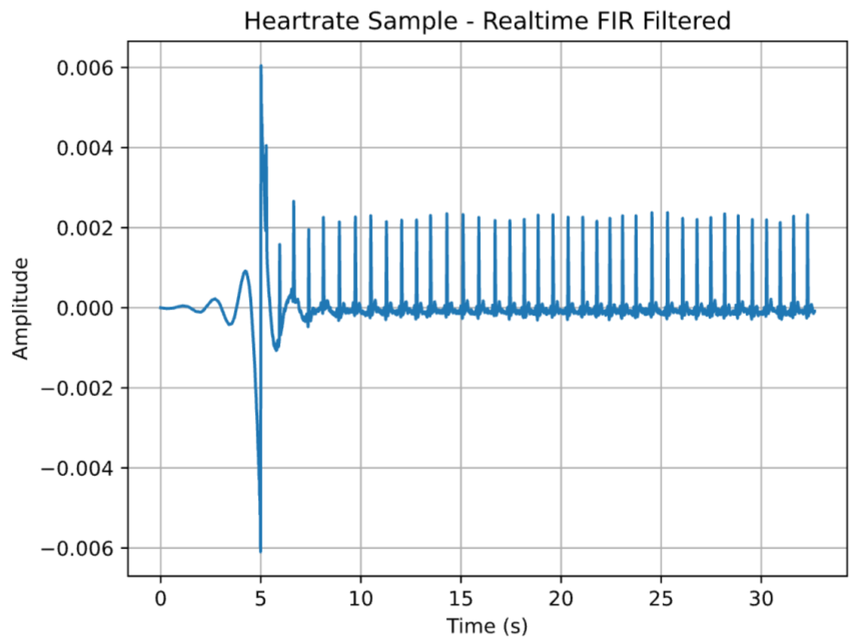

Figure 2: Heartrate sample filtered using realtime FIR filter

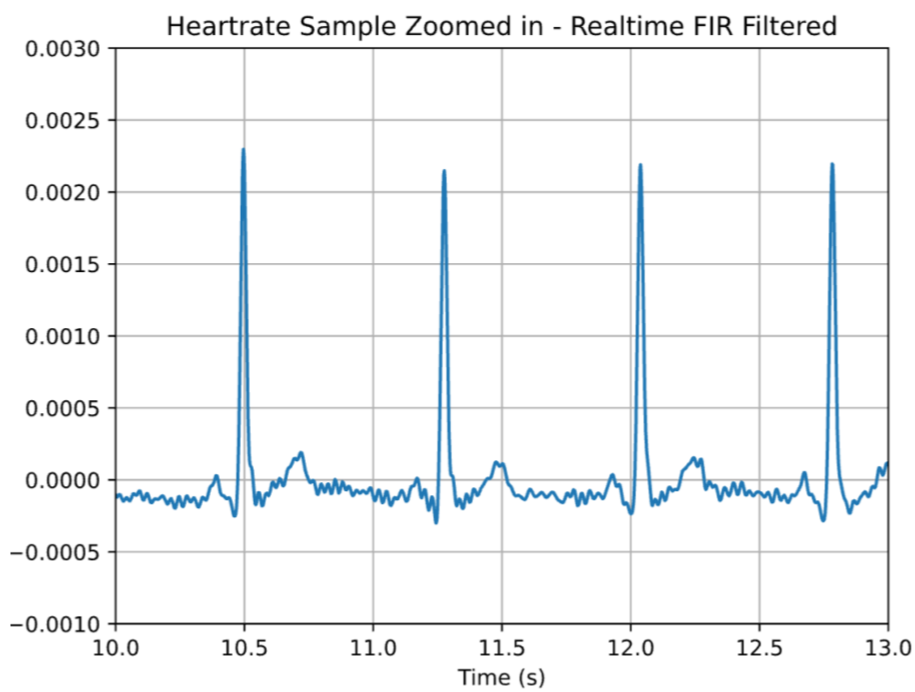

Figure 3: Zoomed-in plot of FIR filtered ECG signal showing PQRST

4) Adaptive Noise Cancellation Using LMS Filtering

For our adaptive LMS filter, we used a simple FIR filter which used a regular buffer as opposed to a ring buffer. The method for this simple filter is doFilterSimple(). For our noise source, we used column 8 from the tsv file, which corresponds to an unused channel from the ECG. We chose this noise source because it is the noise produced by the ECG hardware itself. We added this with a 50 Hz sine wave for better cancellation of the 50 Hz noise. The resulting filtered signal is shown below:

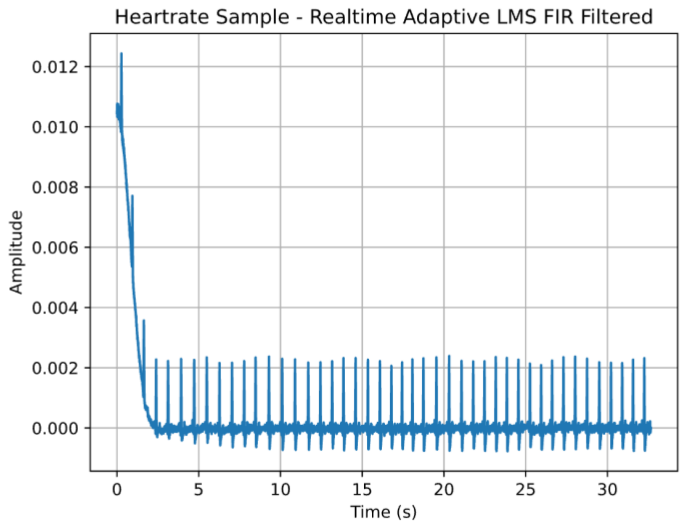

Figure 4: Heartrate sample filtered using realtime adaptive LMS FIR filter

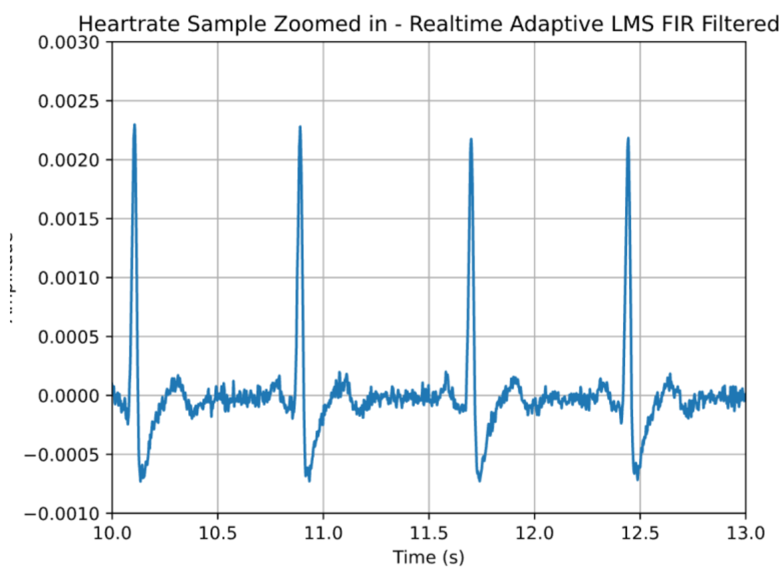

Figure 5: Zoomed-in plot of FIR filtered ECG signal showing PQRST

As can be seen by a comparison of figure 4 and figure 6, the adaptive LMS filter produces a noisier signal. This is because this filter is not a true bandstop filter, and can only learn from the noise samples we give it. The noise sample from ECG channel 8 doesn't account for muscle noise, since this is only present in channel 7, so our filter was only able to reliably move the 50 Hz noise and the baseline wander caused by the ECG hardware.

5) R-Peak Detection Using a Wavelet-Based Matched Filter and Heart Rate Estimation

For our template, we used a morlet wavelet, generated in our code using the function wavelet(). Shown below is a side-by-side comparison of the wavelet with a single heartbeat.

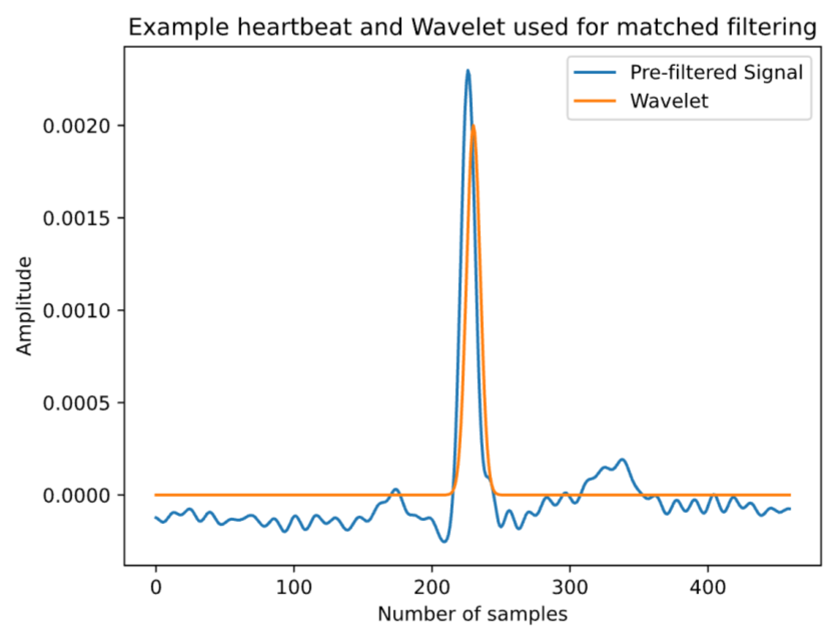

Figure 6: Wavelet and example (prefiltered) heartbeat side-by-side

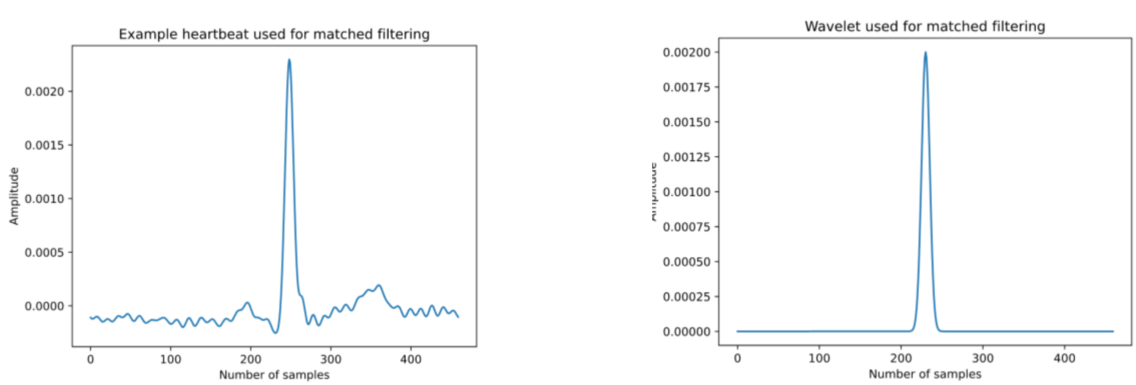

Figure 7: Separate plots of prefiltered heartbeat and wavelet

For our prefiltering, we used our FIR filter from part 3 of the assignment. After squaring the matched filter output, we then applied heuristics to detect heartbeats. The conditions we applied were as follows:

1. The threshold was 0.1e-8
2. Heartbeats were only counted if they occurred at least 6 seconds after the beginning of the sample, to account for ringing from the FIR filter setup
3. A Heartbeat was only valid if it had taken place at least 0.3 seconds after the last detected heartbeat. With this resolution, we are able to detect heartbeats up to 200 BPM, which is significantly higher than the normal range for a heartbeat.

Shown below are relevant plots showing the momentary heart rate and our detections, where 1 signals a detection and 0 signals no detection.

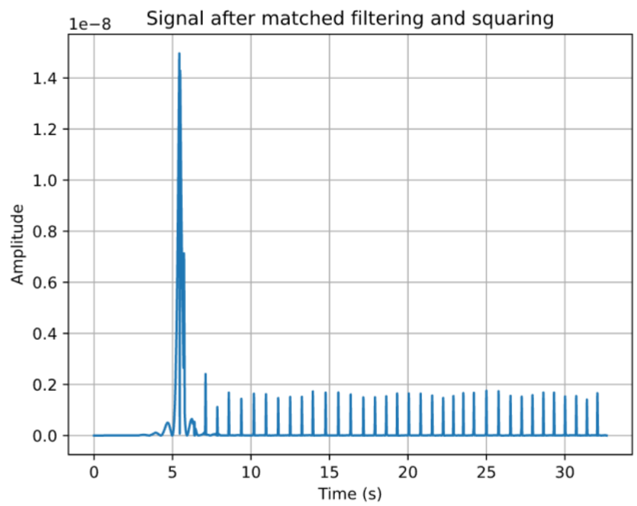

Figure 8: Signal after matched filtering and squaring

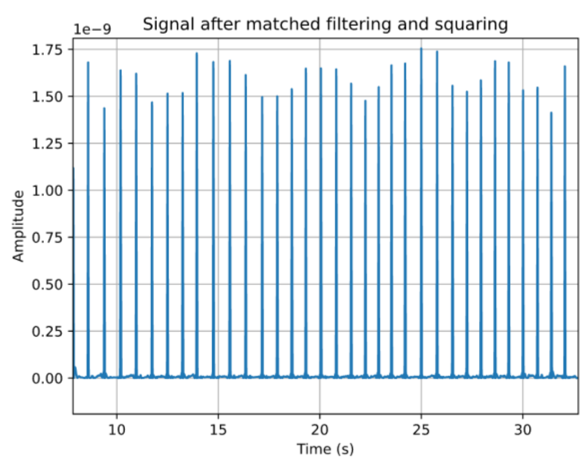

Figure 9: Zoomed in plot of signal after matched filtering and squaring

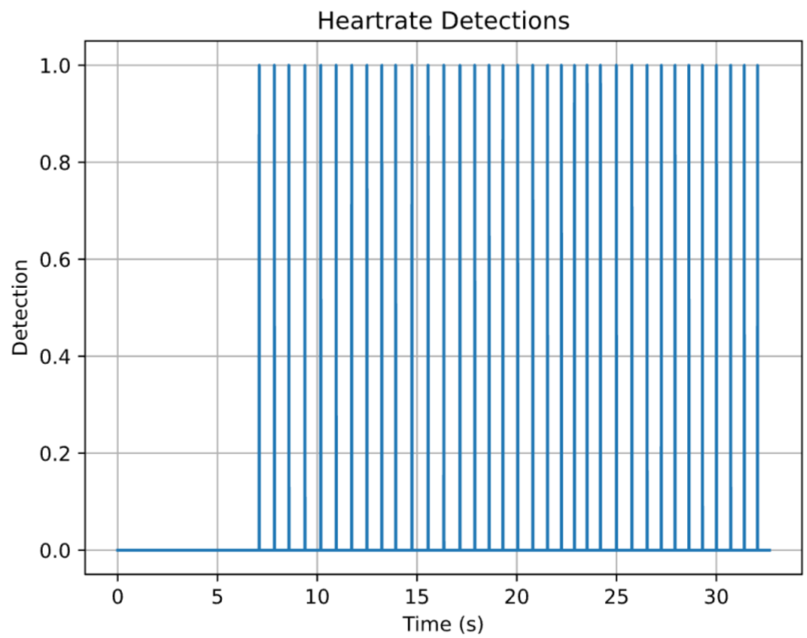

Figure 10: Heartrate detections

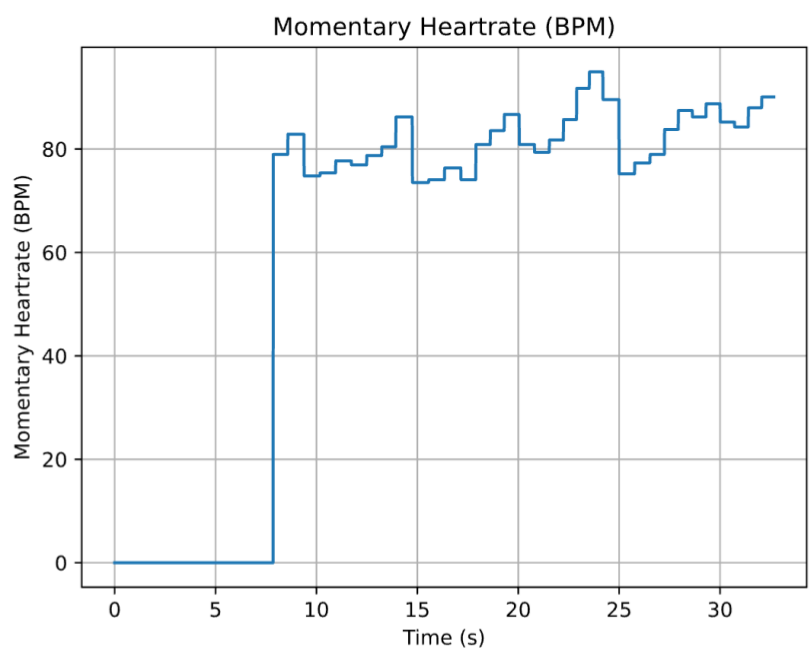

Figure 11: Momentary Heartrate