Report on Assignment 1

Matej Bevec
Email: matejbevec98@gmail.com

***Abstract*—Heartbeat (QRS complex) detection is a common task in biomedical signal processing. In the following exercise, we follow a research paper to implement a real-time QRS detection algorithm, based around two signal processing steps, followed by a decison-making step. We analize the method's performance and evaluate it on the industry standard databases MIT-DB and LTST-DB, as well as explore a number of improvements.**

I. Introduction

Heartbeat detection (i.e. QRS detection) from an ECG record is a long standing problem in the signal processing field. It usually constitutes detecting peaks of the R wave phase of a QRS complex (i.e. the point of a recorded heartbeat with the largest amplitude, see Figure 1).

QRS detection is challenging due to numerous factors, such as low-frequency wander, high-frequency noise and other artifacts introduced by the data collection method, the varying heart-rate and duration of the QRS complex etc. Algorithms who are expected to perform detection in real time face the additional challenge of only "observing" a short segment of the incoming signal at a time, and so not being able to rely on global analysis of the entire signal.

In this exercise, we implement one such real-time QRS detection algorithm, proposed in 2003 [1].

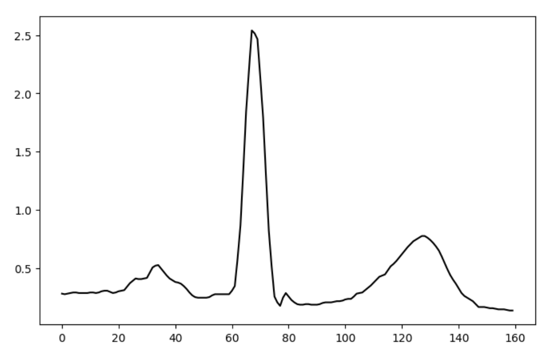

Fig. 1. A QRS complex.

II. Methods

a) Algorithm: The algorithm in question consists of two signal processing steps (i.e. digital filters) and the decision-making step (see Figure 2). The first step is a linear high-pass filter (HPF), consisting of an M_{hp}-point moving average and a $(M_{hp}+1)/2$ delay in parallel. The second step is a nonlinear low-pass filter (LPF) consisting of a sequential squaring of the signal and an M_{lp}-point summation. After the two filters, the so-called feature signal is ready for the decision-making step. This step involves removing (zeroing) all samples with amplitude below a threshold thr. The threshold is initialized to some value and then adjusted after every peak detection due to the formula $thr = \alpha * \gamma * peakHeight + (1-\alpha) * thr$. The formula essentially sets the threshold to a proportion of the average of previous peak heights. Then peaks are finally detected with an unspecified algorithm.

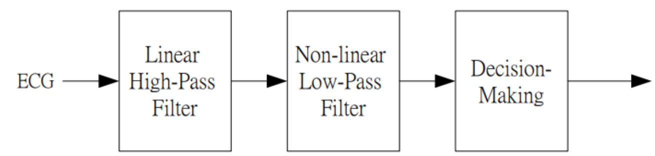

Fig. 2. The detection algorithm steps

b) Our implementation: We implement the algorithm in *python* using the *wfdb* library to read *wfdb* records and *numpy/scipy* for computation. Assuming a fixed sampling rate (fs) of 250, we set the values to $M_{hp} = 7, M_{lp} = 10$ and $\alpha = 0.2$. To make the algorithm real-time, it is implemented to accept short segments (100 samples) of the input signal. One segment is processed at a time, and the filters' memory (zi in *scipy*) is stored to be used when processing the next segment (see Figure 3). For the peak-finding step, we test two implementations. The first is *scipy*'s *findpeaks* method with a required prominence of 0.05. Note that the detection is performed only on the given short signal segment, keeping the algorithm real-time. The second is a simple *convolution* with a *[-1, 2, -1]* kernel, followed by a thresholding at > 0 to find local maxima. In either case, this step is followed by a final, spurious-detection-removal step. If any peaks are closer than a minimum distance between heartbeats (set to 50 samples) only the maximum-valued peak is kept. This is especially necessary when using the convolution approach.

To avoid adaptive tuning of the algorithm parameters given the input's sampling rate, we resample the input signal to a common sampling rate of 250.

III. Results

We evaluate the implemented detector on the first 30 min of every *MIT-DB* and every *LTST-DB* record in terms of *positive predictivity (PPV)* and *sensitivity* (see Table I).

IV. Discussion

A qualitative examination of the (visualized) detection (see Figures 3, 4, 5) reveals that the filtering steps (HPF and LPF) effectively remove artifacts and unenecessary information from the input signal. The highpass filter (HPF) smooths

TABLE I

AVERAGE PERFORMANCE OF THE EVALUATED DETECTOR ACROSS *MIT-DB* AND *LTST-DB* RECORDS. FINDPEAKS AND CONVOLUTION VARIANTS CORRESPOND TO THE TWO APPROACHES DESCRIBED ABOVE. IN THE FORMER CASE, WE SET $\gamma = 0.05$ AND $= 0.08$. IN THE LATTER CASE, WE SET $\gamma = 0.1$. OTHER PARAMETERS ARE DECRIBED IN SECTION II.

Dataset	Variant	PPV	Sensitivity
mit-db	convolution	0.9588	0.9973
	findpeaks	0.9740	0.9933
ltst-db	convolution	0.9346	0.9965
	findpeaks	0.9594	0.9632

high-frequency noise, while the lowpass filter (LPF) attenuates low-frequency patterns, such as wander and the Q and S waves. The resulting feature signal essentially only consists of amplified, positive-signed R waves, which greatly simplifies and robustifies the decision step.

However, it can be observed that there is significant variance in filtered R-wave amplitude between records. The adaptive threshold is designed to adjust for this, but if the initial threshold is too high, no peaks are detected and the threshold never adapts. We combat this by setting a low initial threshold (0.01). Additionally, we robustify the algorithm by keeping the threshold low throught ($\gamma = 0.1$) and removing any false positives after the detection.

Quantitavely, the detector achieves high accuracy and is robust. It exhibits, in both variants, a lean toward high sensitivity and lower positive predictivity, meaning it leaves almost true peaks undetected, but sometimes detects false positives. This is especially appearent in the convolution variant, likely because it relies on over-detecting peaks and removing false positives.

REFERENCES

[1] H.C. Chen and S.W. Chen. 2003. A moving average based filtering system with its application to real-time qrs detection. In *Computers in Cardiology, 2003*, 585–588. DOI: 10.1109/CIC.2003.1291223.

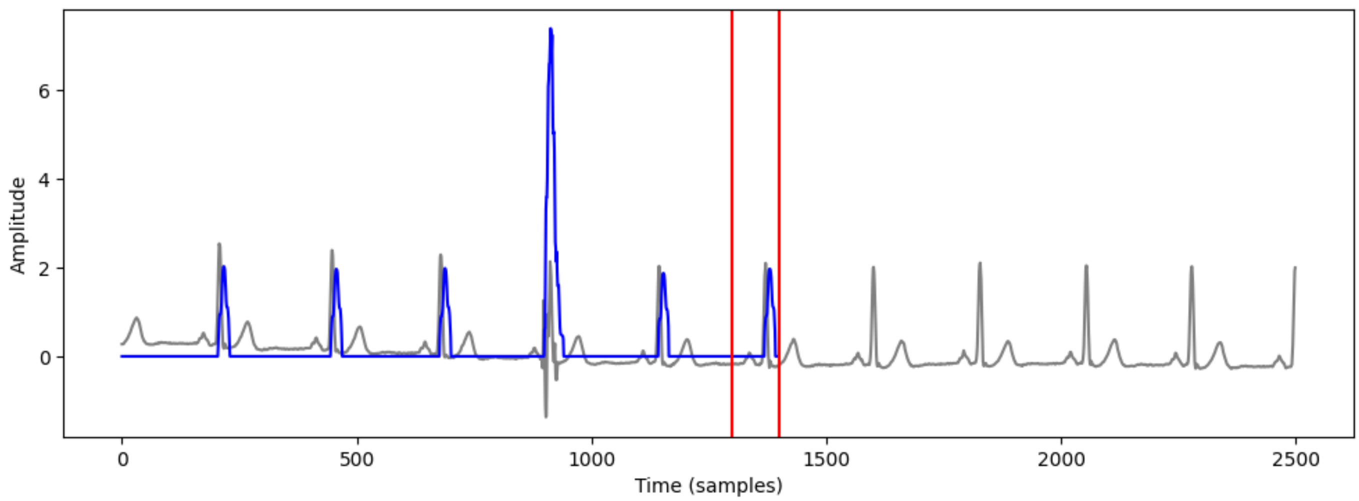

Fig. 3. Processing the signal segment from sample 1400 to sample 1500. The gray plot line represent the original signal. The blue plot represent the signal processed so-far. The red boundaries denote the currently processed segment.

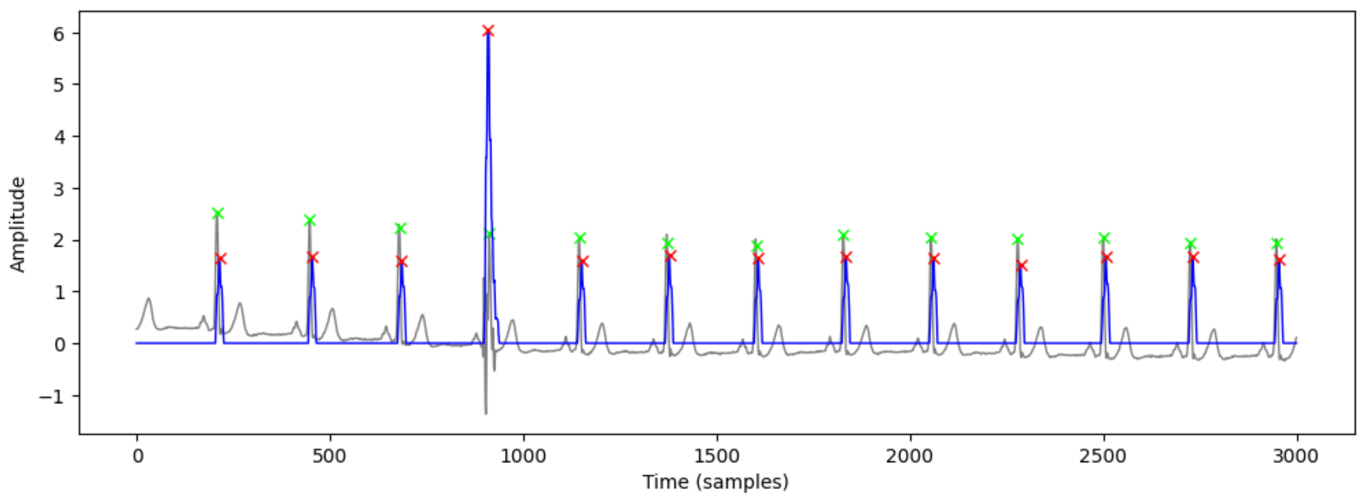

Fig. 4. Peak detections for the first 12 seconds of the s20011 LTST record. The gray plot line and green dots represent the original signal and the annotated true peaks respectively. The blue plot line and red dots represent the filtered signal (before the decision step) and the detected peaks respectively.

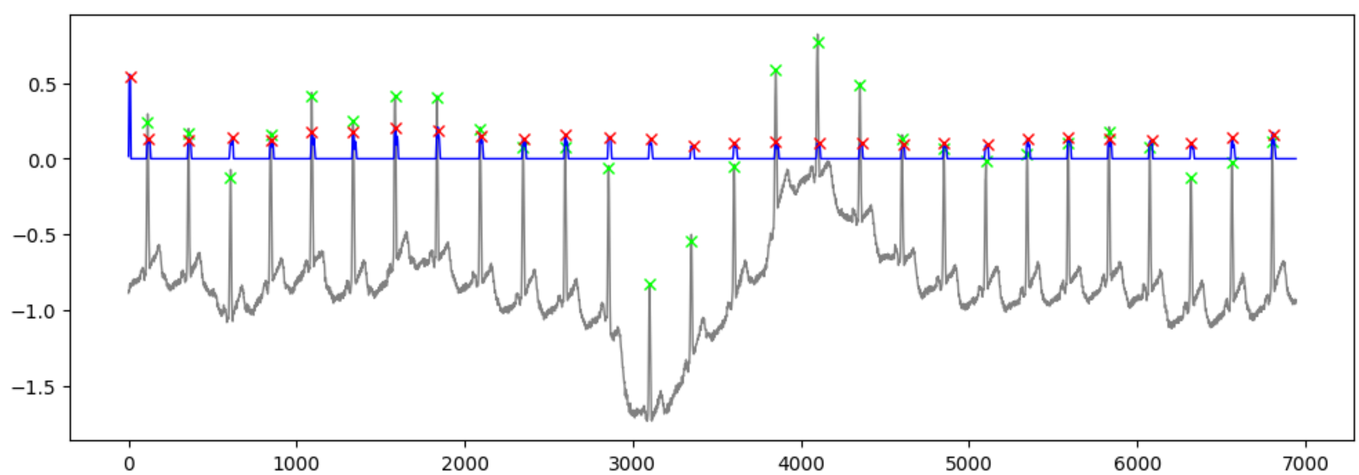

Fig. 5. Peak detections for the beginning of the 121 MIT record. The fitering step remove low-frequency wander and other artifacts observed in the original signal (gray).