A comparison of three QRS detection algorithms over a public database

Raúl Alonso Álvarez

Abstract

We have compared three of the best QRS detection algorithms, regarding their results, to check the performance and to elucidate which get better accuracy. In the literature these algorithms were published in a theoretical way, without offering their code, so it is difficult to check the real behaviour of them. This work brings the community the source code of each algorithm and their results validation over a public database. In addition, this software was developed as a framework in order to permit the inclusion of new QRS detection algorithms and also it's testing over different databases.

Keywords: ECG analysis, QRS detection, Phasor transform

1. Introduction

An electrocardiogram, also called ECG or EKG, reflects the electrical activity of the heart. Every heart contraction produces an electrical impulse that is caught by electrodes placed in the skin. The heartbeat produces a series of waves with a time-variant morphology. These waves are caused by voltage variations of the cardiac cells.

The primary function of the heart is to pump oxygen-rich blood throughout the body. The heart is composed of muscle cells, which produces the contraction; these cells are connected into a network (conduction system) that allows an electrical impulse spreads throughout the heart. A cardiac cycle is created when an impulse propagates through the conduction system. The cardiac cycle starts in the atria and goes down throw the ventricles, so the impulse is triggered in the atria and it precedes the heart contraction [1].

Each cardiac cycle is composed of two stages, contraction and relaxation, called in electrical terms depolarization and repolarization. Depolarization produces a rapid change in the cell potential (from -90 to 20 mV), and this change in voltage causes the depolarization of neighbour cells. After it's depolarization cells come back to their resting stage, produced by their repolarization.

The electrocardiogram provides important and relevant information about the heart state. Physicians all over the world are using it to diagnose cardiac diseases, which are the main cause of mortality in our society. There are a wide variety of devices that allow to analyze the heart activity, e.g. Holters, arrhythmia monitors, ECG stress test systems and so on. Nowadays the information provided by these gadgets can be easily digitized and processed by computers. Thus, using the power of computers, we can detect heart diseases or anomalies that otherwise could only be detected by experts physicians.

In ECG processing, it is very important to detect very accurately heartbeats, because it is the base for further analysis. The energy of heartbeats is located in the QRS Complex. So, an accurate QRS detector is essential for ECG analysis.

In order to analyze the ECG signal is important to know waves that forms the heartbeat, see Figure 1 on page 3. Each beat is divided in three stages, atrial depolarization (P wave), ventricular depolarization (QRS) and finally ventricular repolarization (T wave). These three stages are continuously repeated in the ECG signal, representing heartbeats over the time.

QRS detection is difficult, because the signal varies along the time and different types of noise can be present in it. From the eighties, software QRS detection has been a research topic. Whereas in the early years the performances of the algorithms were determined by its computational load and complexity, nowadays the detection performance is the major objective.

In most QRS detection algorithms there are two differentiated stages: preprocessing and decision [2]. In the pre-processing stage different techniques are applied to the signal, such as linear and nonlinear filtering or smoothing, to attenuate P and T waves as well as the noise. While in the decision stage the most important task is the determination of thresholds and in some cases the use of techniques to discriminate T waves. Some algorithms include another decision stage to reduce false positives.

To evaluate the performance of each algorithm there are available benchmark databases, where the heartbeat's positions are well annotated by two cardiologists that must be in agreement about each beat position. These databases

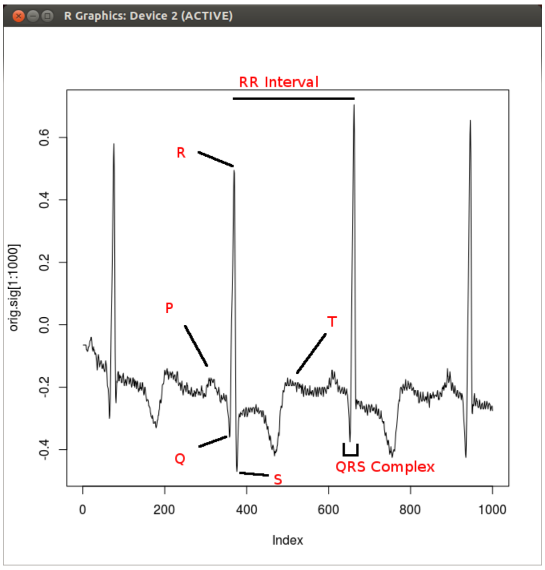


Figure 1: The QRS complex in the ECG signal

contain a large variety of ECGs, as well as signals that are rarely observed but clinically important. Some of these databases are: MIT-BIH [3], AHA (`http://www.heart.org/HEARTORG/`) or the QT Database (`http://www.physionet.org/physiobank/database/qtdb/doc/node3.html`).

Nowadays there are many algorithms for detecting heartbeats published but most of them do not offer the source code and are not validated on the same databases. Usually these algorithms are explained in a theoretical way, while others only include a few guidelines for real implementation. It becomes necessary, therefore, a tool that allows users to implement their algorithms and to compare its performance against different databases. In this research work, three of the best QRS detection algorithms, regarding their results, are compared in order to check its performance and see which get better accuracy.

This work was developed using the free software programming language R [4], typically used for statistical computing and graphics. This language is available in multiple operating systems, for this reason our software will be multiplatform. R is very powerful using matrix operations and it is simple to plot the results, which makes it very adequate for signal processing. However it is highly inefficient with loops, which makes our application to run very slow when using loops.

To solve this situation we linked R with C using a R library that allows doing computationally intensive tasks in C language, thus our software is much more faster. However this implies that it should be compiled for each different operating system and computer.

2. State of the Art

Pahlm and Sörnmo [2] noticed that most of QRS detectors are divided in two sections: pre-processing stage and decision stage. Almost all algorithms use a filter stage before the detection, in order to remove the noise and reduce the amplitude of P and T waves to facilitate the subsequent detection [5]. Some algorithms apply a bank of high-pass and low-pass filters, which is known as a band-pass filter, while others do it separately. Once the signal is pre-processed, the QRS complex is detected by thresholding the signal, where thresholds can be fixed or adaptive. Finally, most algorithms use another decision stage, where decision rules are applied in order to reduce false positives.

During the last 30 years there have been proposed a lot of algorithms for QRS detection. There are many approaches, from artificial neural networks or genetic algorithms to wavelet transforms, filter banks, heuristic methods or machine learning methods [6].

2.1. Different Approaches

- Derivate-based algorithms: algorithms based in filters and derivative as in [7]. They often use a high-pass filter and derivative is used to determine the maximum slope, which corresponds to QRS complex.

- Algorithms based on digital filters: other algorithms with use more sophisticated filters [8] [9]. Two different filters process the ECG, low-pass and high pass ones, with different cut-off frequencies, forming the band-pass filtered signal. Also thresholds are compared adaptively [10] [11].

- Wavelets: wavelet based approaches discompose the signal into different scale components to analyze the signal in different frequency bands [12] [13] [14]. Then fixed thresholds are applied to obtain the characteristic points.

- Neural Networks: neural networks are used to predict current signal values from the past ones, and therefore apply suitable filters to attenuate the noise [14] [15].

- Hidden Markov Models: HMMs model the data sequence according to an underlying Markov chain [16]. The algorithm infers the underlying state from the observed signal.

- Genetic algorithms: they intend to get optimal polynomial filters, for preprocessing stage, and parameters for decision stage [17].

- Phasor Transform: transform each sample of the signal into a complex value preserving the signal information [18]. It enhances ECG waves and then the detection is easier applying thresholds.

Table 1 shows the performance of each algorithm. Okada algorithm [9] was not considered because it uses only 4 ECG records obtained from patients in surgery, and a cardiologist analyzed the performance position-by-position directly. In the case of Arzeno's paper [7], he compared traditional first-derivative based algorithms, where he obtained the best results with the Hamilton & Tompkins algorithm [11].

Most algorithms are developed by research groups, they own the source code and they only share with scientific community the behaviour of their algorithms with a few guidelines and the results of its validation over a database. There are also commercial systems for ECG analysis, but these do not offer their code neither the validation data.

Algorithm	Database	Sensitivity	Pos. Predictivity
N. Arzeno *2008* [7]	MIT-BIH	99.68%	99.63%
V. Afonso *1999* [8]	MIT-BIH	99.59%	99.56%
J. Pan *1985* [10]	MIT-BIH	99.3%	-
P. Hamilton *1986* [11]	MIT-BIH	99.69%	99.77%
J. Martinez *2004* [12]	MIT-BIH, QT, ST-T, CSE	99.66%	99.56%
C. Li *1995* [13]	MIT-BIH	99.8%	-
B. Abibullaev *2011* [14]	MIT-BIH	97.2%	98.52%
Q. Xue *1992* [15]	MIT-BIH	99.5%	97.5%
D. Coast *1990* [16]	AHA	97.25%	85.67%
R. Poli *1995* [17]	MIT-BIH	99.6%	99.51%
A. Martinez *2010* [18]	MIT-BIH, QT, ST-T,TWA	99.81%	99.89%

Table 1: Performance of some QRS detection algorithms. For each algorithm we show the best result provided by their authors

3. Original contribution

In this research work we decided to implement two algorithms based on digital filters, Pan & Tompkins algorithm [10] and Hamilton & Tompkins algorithm [11] and a new algorithm based on the phasor transform [18].

The Pan & Tompkins algorithm was a major breakthrough at the time of its publication. Furthermore it is the most cited paper (more than 850 times) related with QRS detection. It made use of the latest techniques at that time and it is an algorithm that can quickly adapt to the signal changes and get a good detection even in noisy signals.

Hamilton & Tompkins algorithm was published the following year and it is very similar to Pan & Tompkins. It uses the same pre-processor with slight variations but with completely different decision rules. Theoretically it improves a bit the detection compared with the previous algorithm.

The phasor-based algorithm is a recent work, published in 2010, and it is characterized by its robustness, low computational cost and mathematical simplicity. The results are even better than those of the other two algorithms, according to their authors.

In this section the developed software is presented, highlighting those parts that are an original contribution. Besides the algorithm, we also explain other features of this software.

3.1. Programming language

As a programming language to implement the detection algorithms we have chosen R [4]. It is a programming language widely known in the scientific community, therefore will be not trouble to understand the source code. The advantages of using R are:

- It allows plotting very easily, this allows check the signal morphology or the positions marked as a beat, which facilitates the identification of errors.

- It is a language based on matrix operations, useful for searching minimum, maximum, mean, median or do sums, differences and products of matrices. This kind of operations are widely used in signal processing.

- It is open source, which allows us to share our software with the scientific community, with the benefits of that.

3.2. Package structure

We have divided the package in four components: Linux Scripts, ECG Records, R code and Results (see Figure 2). Linux scripts contain all necessary code for doing the automatic validation. In ECG Records folder there are the records to analyze, while Results folder contains the output after apply the automatic validation.

The code of the three detection algorithms developed are in the R code folder, it also includes the C functions to enhance the performance.

The three algorithms follow the same workflow, represented in Figure 3.

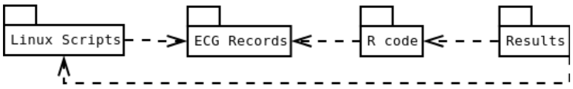


Figure 2: Package components

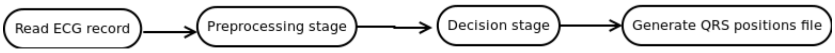


Figure 3: Algorithms workflow

3.3. Database and validation process

In this context, a database is a collection of recordings, available as a set flat of files. Some ECG databases are freely available via Physionet [19] (`http://www.physionet.org/`) to the research community. Many of these databases were developed at MIT and at Boston's Beth Israel Hospital (MIT-BIH).

From all the available ECG databases in Physiobank (`http://www.physionet.org/physiobank/database/#ecg`), we have selected the MIT-BIH Arrhythmia Database, because it contains a large variety of ECGs, including signals, which are rarely observed but clinically important.

The difference with other databases is that it is focussed in heartbeat positions with different QRS morphologies, while others are focussed in other waves, P, T or another features such as ST episodes, rhythm changes, and signal quality changes. In addition, this database has been the most used for QRS detection algorithms validation, see table 1.

Each ECG record contains three files with different suffixes that indicate their content. A record contains:

- .dat (signal) file: a binary file containing the digitized samples of the ECG signal.
- .hea (header) file: a short text file describing the signal features.
- .atr (annotations) file: it contains labels, each of which describes signal features at specific time, such as QRS complex location, and type.

The MIT-BIH Arrhythmia Database contains 48 half-hour excerpts of two-channel ambulatory ECG recordings. However in our software we only analyze one channel because, as Pan & Tompkins noticed, the electrode positions in an ECG device recording are orthogonally placed, so a high-quality signal on one channel normally implies a low-amplitude ECG with poor signal-to-noise ratio on the second channel [10].

To validate our detector we used some WFDB functions (`http://www.physionet.org/physiotools/wfdb.shtml`). WFDB is software for viewing, analyzing, and creating recordings of physiologic signals. Our software produces as output a text file with the heartbeat positions that later are converted into an annotation (.atr) file. Then this new annotation file is compared with the original ECG annotation file, and the results are saved in a new text file with the accuracy of sensitivity and positive predictivity.

The comparison is performed by *bxb* routine, a basic functionality of the WFDB library, which compares annotations beat-by-beat. The performance metric is expressed in terms of sensitivity and positive predictivity, given by:

Sensitivity $(Se) = \frac{TP}{TP+FN}$

Positive Predictivity $(P+) = \frac{TP}{TP+FP}$

where TP is the number of true detection, FN is the number of undetected beats and FP is the number of false detections.

To automate this process, we have created some Linux scripts in order to automatically check the result of all ECG records and calculate the mean sensitivity and positive predictivity. This allows users framework users to obtain in a fast and convenient way the results of their detection algorithm.

3.4. Pan & Tompkins algorithm

This algorithm was developed in assembly language, using integer arithmetic to be more computationally efficient, and it is able to do real-time QRS detection. The R wave slope is the typical feature to locate the QRS complex, however it is insufficient for proper QRS detection. For this reason this algorithm analysis is based on the slope, amplitude and width of the signal [10]. As we mentioned in section 2 this algorithm is divided in two different stages, pre-processing and decision. In the pre-processing stage the signal is prepared for later detection, removing noise, smoothing the signal and amplifying the QRS slope and width. Later, in the decision stage, thresholds are applied to the signal in order to remove noise peaks and consider only signal peaks.

3.4.1. Pre-processing stage

The first step of this stage is to pass the signal through a block of filters [20] to reduce noise and influence of the T wave. The block of filters is a cascade of low-pass and high-pass filters to achieve a 3 dB pass band of 5-12 Hz. Low-pass filter is used to remove noise while high-pass filter attenuates P and T waves and baseline drift. This digital band-pass filter improves the signal-to-noise ratio and permits the use of lower thresholds than would be possible on the unfiltered signal.

In the next step the derivative is applied to filter output, providing complex slope information. Then the signal is squared point by point, which intensifies the slope of the derivative signal and helps to reduce false positives caused by higher than usual T waves.

Finally a moving window integrator is applied, which includes information about slope and width of the signal. The length of the integration window is very important because it must always contain the QRS complex that can be of different extents and, nevertheless, not be mixed with other waves such as T wave. In our case we get the best results using a window size of 150ms, which in case of a sample rate of 360 samples/s, implies 54 samples wide.

All the equations needed for this pre-processing stage are described elsewhere, [10]. Moreover it should be noted that band-pass filters and the derivative operations cause a delay, which must be considered when the fiducial mark is placed.

3.4.2. Decision stage

Once the signal is pre-processed, it is ready for QRS detection. In this stage two sets of thresholds are applied to both, filtered ECG and the signal after moving window integration (henceforth mwi). By using thresholds in both signals, the reliability of detection is improved compared to using only one waveform.

The detection thresholds float over the noise that is sensed by the algorithm (considering as noise each peak which is not classified as signal peak), adjusting them to the signal changing conditions automatically. To accomplish this, first is needed a peak detection, considering that a peak is a local maximum, and then classifying every peak as noise or signal, according to the established threshold.

In our case we developed a local maximum detection algorithm, which detects all local maximum of the signal (see Figure 4a). The local maximum is the highest peak in an interval; in our case we get the best results with an interval of 80 samples. A smaller interval would detect too many peaks, while a larger one may not detect some signal peaks.

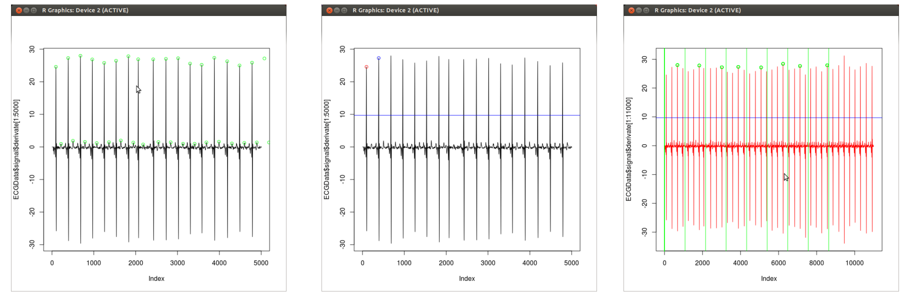
In this algorithm, in order to establish the threshold over the noise (peaks), and then automatically classify peaks as noise or signal, it becomes necessary to seek the correct initial threshold. Once we have this done, we can proceed with the execution of the algorithm and then threshold will be automatically updated and it automatically discriminates between noise and signal peaks.

To set the initial threshold, in our case we decided to find the two first R peaks (see Figure 4b), and then classifying these two peaks as signal peaks and all peaks found until that second R peak, classified as noise peaks. From this point we will have information on signal peaks and noise peaks so that it is easy to establish initial threshold over the noise. The threshold is calculated as follows:

$$SPKI = 0.125 \times PEAKI + 0.875 \times SPKI$$
$$NPKI = 0.125 \times PEAKI + 0.875 \times NPKI$$
$$THR1 = NPKI + 0.25 \times (SPKI - NPKI)$$
$$THR2 = 0.5 \times THR1$$

- PEAKI is the current signal peak in SPKI calculation or current noise peak in NPKI calculation

- SPKI is the running estimate of signal peaks



(a) Peaks, detected as local maximum

(b) Two first R peaks and initial threshold

(c) After splitting the signal in segments, and initial threshold

Figure 4: Stages in the ECG pre-processing, from Pan & Tompkins and Hamilton & Tompkins algorithms

- NPKI is the running estimate of noise peaks
- THR1 and THR2 are thresholds; the second threshold is half of the first because it is used in case a search-back is needed.

To classify the two R peaks correctly independently to the waveform morphology, we have implemented our own technique as an original contribution. This technique classifies correctly the two first peaks even in signals with higher than usual T waves.

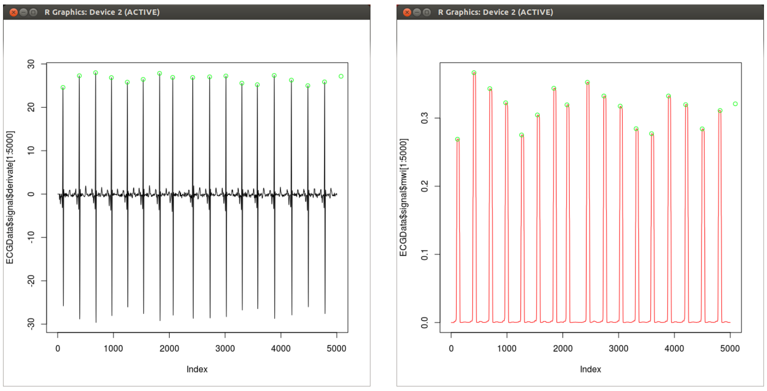
This technique is to find the median of the first N maximum peaks found and use a percentage of the median as a threshold. Then the first two points above this threshold are considered R peaks. To search the N first maximum, what we did was split the signal into segments, see Figure 4c, in which at least one heartbeat is present, regardless heart rate from the ECG.

Considering that physiologically minimum heart rate is 25 beats/min, this means that every 2.4 seconds we have at least one heartbeat. In our case, segments are marked every 3 seconds and we look for the maximum peak in every segment and when we have all the maximum peaks the median is calculated. Next we set the threshold at 35% the median, this percentage was determined experimentally after a lot of testing in database's records.

When we have located the first two peaks, the threshold is applied to the signal and peaks are classified. Thus the majority of R peaks that represent QRS complex are obtained. However, there are lost peaks in some irregular signals or waves with abrupt changes in amplitude or frequency.

To solve this problem the algorithm uses prior information about RR intervals. So if the current RR interval is a 166% higher than the average of the last RR intervals, a search-back is applied, R peak is considered as the higher peak that is in the current interval between the THR1 and THR2 thresholds. Thus detecting heartbeats that otherwise would not be detected.

Moreover, this algorithm includes another technique consisting in false positives removal, which consists in removing peaks considered as T waves. If after a R peak another peak is detected within 200 ms is considered to be a T-wave, and if it is caught between 200 ms and 360 ms a judgement is made to decide



(a) R peaks detected in the derivative signal

(b) R peaks detected in the mwi signal

Figure 5: Placing the fiducial marks in the R peaks in the derivative and mwi signals

whether it is a T-wave or an ectopic beat. Is considered to be a T-wave if the slope in the derivative signal is less half of the previous peak slope, otherwise it is considered a QRS peak.

To place the fiducial mark, Pan & Tompkins mentioned that can be placed at the point of maximum slope of derivative signal or at the R peak. We tried to put it in both positions and we obtained better results by placing the fiducial mark at the R peak (see Figure 5). Furthermore, as we had located the R peaks and classified as signal peaks or noise peaks, fiducial mark placed at the R peak did not mean to add more computational calculation to the algorithm.

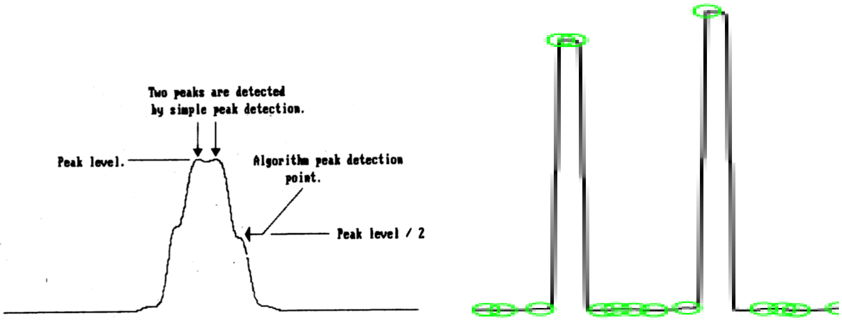
As we said before, this algorithm applies thresholds over two different signals, derivative signal (Figure 5a) and the signal after moving window integrator (Figure 5b). So we detect the fiducial mark in both signals and then we only consider peaks coincide in both signals as QRS complexes.

To check if a peak is the same in both signals, it is important to take into account the delay between these signals, in this case the delay is half the width of the integration window. Since it is very unlikely that the R peak coincides in the same sample in both signals even when taken into account the delay, it is also taken into account a possible deviation of the peak, which after testing we established in 50 samples. Therefore, when the difference between both R peaks is less than 50 samples the peak is considered as a true detection

To check if the R peaks match, we always start from the signal with less peaks detected and then look in the other signal if each peak match another. In our case we always marked the final peaks in the mwi signal, and finally place those peaks in the original signal considering the delay.

3.5. Hamilton & Tompkins algorithm

Hamilton & Tompkins algorithm [11] consists of a pre-processing stage, which provides a number of events classified as QRS complex and noise, and then a decision stage at which they added new rules to discriminate among QRS and noise events.



(a) Ripples in the mwi. Image taken from [11]

(b) Multiple peaks detected due to the ripples in the mwi signal in our implementation

Figure 6: The problem of detect various peaks in the same R peak due to the ripples in the wave

The most important decision rule is where to place the detection threshold. Other rules are blanking, ignore events that follow a QRS in a set time, back search and using the slope to distinguish T-waves and ectopic beats.

To develop this algorithm, we have taken advantage of Pan & Tompkins preprocessing, including the peak detector. The only condition needed in this algorithm is to have event vectors, of maximum local peaks, which will be classified as noise peaks or signal peaks. Therefore, in this algorithm the pre-processing stage will not be explained since it was explained previously (see 3.4.1).

We are only going to highlight how we solved one of the problems detected in this algorithm: the local maxima peak detector could falsely detect multiple peaks due to ripples. Figure 6 shows the problem of multiple ripples and how Hamilton & Tompkins algorithm solve it (6a) and an example of using a narrow interval in the peak detector in our development (6b)

Hamilton & Tompkins developed an algorithm that avoid multiple detection, they store the first peak position and height and a new peak is defined only after a height which is less than half of the previous peak. In our implementation we avoid the multiple detection, considering a peak when its value is higher than its neighbourg in both sides of an interval. This way it does not need to use another algorithm and it correctly detect only one peak, as maxima.

3.5.1. Decision stage

As in the previous algorithm, this stage is also based on differences between noise peaks and signal peaks. Then, it is set a level for noise peaks and another for signal peaks, and detection threshold is set between them.

The main difference with the above algorithm is the manner in which it establishes noise level and signal level. While in previous algorithm an equation is used to update levels when each peak is detected, in this algorithm different approaches were used to obtain these levels.

They focused on optimizing the decision rules, and as we mentioned above the most important of these is where to set the detection threshold. They tested the performance of three estimators to place the adaptive threshold: the mean, median, and an iterative peak level. They concentrate primarily on explaining the advantages and disadvantages of each of these three estimators and their results obtained (detection accuracy, false positive rate and false negative rate).

For our development we use the median as estimator, which is the one they reported as the better, as much for detection accuracy as for fewer false positives. For calculating the adaptive threshold we used this equation:

$DT = NPL + TC \times (QRSPL - NPL)$

Where DT is the detection threshold, NPL is the noise peak level, TC is the threshold coefficient and QRSPL is the QRS peak level. The authors affirm they got the best results with a threshold coefficient of 0.133 and using an even number of previous peaks to calculate the median, greater than 6. We attempt with 8, 10 and 12 data peaks and we achieve similar results. To calculate the first two peaks we use the same technique as in the Pan & Tompkins algorithm (see Figure 4).

In this algorithm, in addition to peak level estimation, another decision rules are used, as search back and refractory blanking. The search back in this algorithm is done in a simpler way than the previous. The information of the previous 8 RR intervals is stored, and when a QRS complex is not found in an interval of length the 150 percent of the median of these RR intervals, the search back is applied.

We always keep the highest peak that not exceeds the DT threshold but exceeds the search back threshold, because this peak happens to be considered a QRS complex if there is no R peak exceeding the DT threshold in the search back interval. Search back threshold was 30% of the detection threshold as authors established in their paper [11]

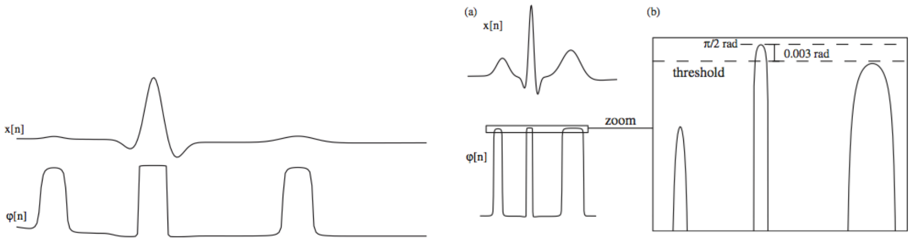
Finally, the last decision rule we used was the refractory blanking, which is to ignore any peak detected in 200ms since the last peak considered a QRS complex. Initially we tried to classify as T wave or QRS those peaks that are between 200 and 360ms from the previous peak, as in the previous algorithm, but in this case we lowered the sensitivity of the detection, so we finally discard this classification.

According to Hamilton & Tompkins, the fiducial mark should be placed in the largest peak in the band-passed signal, searching this peak in a range from 225 to 125 ms preceding each detected peak in the mwi signal. We placed the fiducial mark in three different signals: low-pass signal, high-pass signal and derivative signal, by moving back according to the delay that exists in each signal. The best results were obtained by placing the fiducial mark in the derivative signal, and then moving the fiducial mark to the original signal according to the delay.

3.6. Phasor algorithm

The detection algorithm based on the Phasor transform (henceforth PT) is recent, year *2010*, and it is characterized by its robustness, low computational cost and mathematical simplicity. In this paper, besides the QRS complex detection, they have made the delineation of all ECG waves, i.e. mark its beginning and end, which brings more utility to ECG analysis for both physicians and biomedical engineers. Nevertheless we are only interested in the part of this article that focuses on QRS detection.

The PT converts a real function in the complex domain preserving its information, regarding root mean square and phase values. It converts each instan-



(a) (b)

Figure 7: (a) Example of an ECG beat, x[n], where waves are notably low amplitude, and the resulting PT signal, φ[n]. (b) Establishing a threshold of 0.003 rad. below the maximum phase variation ($\pi/2$). Images extracted from [18].

taneous ECG sample into a complex number, called phasor. A constant value R_v is considered the real part, whereas the imaginary component is the original value of the ECG sample:

$$y[n] = R_v + jx[n]$$

Where y[n] is the phasor. The magnitude M[n] and the phase φ[n] are computed as:

$$M[n] = \sqrt{R_v^2 + x[n]^2} \text{ and } \varphi[n] = tan^{-1}(\frac{x[n]}{R_v})$$

Converting each instantaneous ECG sample into a phasor enhances the ECG waves. Thus, considering instantaneous phase variation in consecutive samples of the phasor transformed ECG, the slight variations provoked by P and T waves in the original signal are maximized (Figure 7a). The equations are explained in depth elsewhere [18].

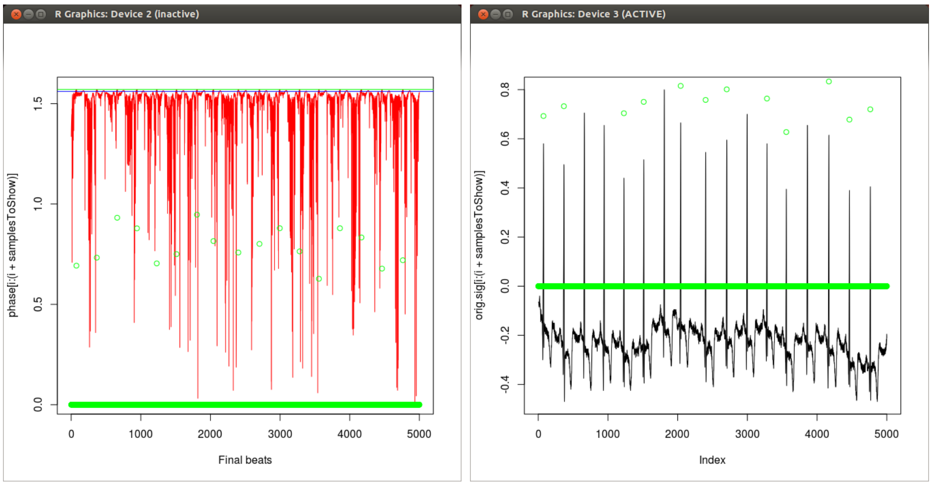
3.6.1. QRS detection

QRS complexes were detected applying the PT with a value of $R_v = 0.001$ to the absolute value of the original ECG, previously removing its baseline wander with a forward/backward high-pass filter of 0.5 Hz cut-off frequency. Thus, by establishing a threshold of 0.003 rad. below the maximum phase variation ($\pi/2$), the QRS can be located at those segments exceeding the threshold, Figure 7b. The R peak is set in the maxima magnitude M[n] point of each segment.

When a peak is not detected in an interval longer than the 150% of the last RR distance, they apply a new search-back with lowered thresholds until successful detection. In the other hand, when two RR points are localized in an interval lower than 40% of the last RR distance, they remove the one with lower magnitude to prevent double R detection within a beat.

3.6.2. Our implementation

At first we tried to read the digitized signal as we did in the previous algorithms. In ECG records the information of the signal is digitized by default, so


R Graphics: Device 2 (inactive)
phase[i:(i + samplesToShow)]
0.0
0.5
1.0
1.5
0
1000
2000
3000
4000
5000
Final beats
R Graphics: Device 3 (ACTIVE)
orig.sig[i:(i + samplesToShow)]
-0.4
-0.2
0.0
0.2
0.4
0.6
0.8
0
1000
2000
3000
4000
5000
Index

(a) PT signal, with the beats posions

(b) Original signal with the beats positions

Figure 8: PT and original signals in our software with the same beats in both signals. The peaks are placed according to the magnitude M[n], for this reason the higher beats does not appear in the original signal plot.

its value depends on the analog-to-digital converter gain. In the previous algorithms we get the independence of the ADC normalizing the signal, however in this case we did not achieve the desired results, so we decided to use the ECG's physical units that are represented in millivolts.

Once the signal is read, it is normalized and then a forward/backward filter is applied. We first did the normalization as in the previous algorithm, between 0 and 1 for the whole signal. Regarding the filter, we used the filtfilt function provided in the signal package (`http://cran.r-project.org/web/packages/signal/index.html`), which allows filtering in both directions with the desired cut off frequency without producing a delay. Finally we apply the PT to the absolute value of the filtered signal.

Next we apply the threshold to the PT and seek the maximum magnitude M[n] values in those segments exceeding the threshold (Figure 8a). Although due to the irregular morphology of the signal, in those sites where the amplitude of the QRS complex is small, the PT does not reach the threshold. Therefore our results were far from the article results.

As we were unable to achieve the results obtained for their authors, we contacted one of the them, A. Martinez, who told us that there was a pre-processing stage that consists of removing the baseline, the DC component, and normalizing the signal. And, although the article does not indicate it, it is appropriate windowing a stretch of the ECG signal and also using adaptive thresholds.

Thus we tried a windowed normalization, both between 0 and 1, and -1 and 1. Finally we discarded this option because, although the detection improved slightly, it added too computational cost seeking the minimum and maximum value each time the window is moved along the signal.

However, using adaptive thresholds significantly improved our detection, because it adapts to the signal changes. Due to the experience we had from the previous algorithms, we decided to calculate the adaptive threshold based in the median of the latest detected peaks, placing the threshold 0.001 below the median of the last R peaks, (see Figure 8a).

We also used search back and remove false positive techniques, but instead of using only the latest RR interval, we decided to improve it and use the median of the past 8 RR as in the Hamilton & Tompkins algorithm.

After applying these techniques we reached a little better detection rates, but the results of our phasor based algorithm are still far from the other two algorithms. Due to the lack of time we could not do more improvements to this algorithm.

4. Results and discussion

In this section we present a comparison of results obtained with the algorithms implemented. The values presented are the results we got after doing multiple tests with each algorithm, and testing different detection parameters.

We can get better results in sensitivity or predictivity changing the parameters, but improving a term causes worsen the other. For example, in Hamilton & Tompkins or Pan & Tompkins algorithm, if we decide that the interval for searching local peaks is smaller, the algorithm would detect more peaks, which would increase the sensitivity, in this case the predictivity would be worse. The same occurs in Pan & Tompkins algorithm to check if a peak is the same in the derivative signal and mwi signal, if we reduce the interval we get better predictivity otherwise we get better sensitivity.

Hence, the results presented in the following tables and graphics are the best we get for each algorithm in terms of sensitivity and predictivity.

In table 2 we present the average results we obtained after validated each algorithm with the entire database. Comparing the results with table 1 we can see that we improve the result of the first algorithm, almost reached the second and we were not able to get the third. To perceive the significant differences between these results, table 5 shows the p values obtained comparing algorithms.

Tables 3 and 4 summarized all the information of the sensitivity and positive predictivity results, to examine how the distributions of the results behave.

In addition, in Figures 9 and 10 can be seen the results presented by each algorithm in each of the 48 analyzed records, from the MIT-BIH Arrhythmia Database. In both figures a zoom is shown to allow differentiating values near 100% and finally to highlight the difference between the two best algorithms 2 boxplots are provided, (Figure 11).

Algorithm	Sensitivity	+ Predictivity	RMS RR error
1- Pan & T.	99.79% ± 0.35	99.84% ± 0.39	99.40ms
2- Hamilton & T.	99.71% ± 0.46	98.65% ± 3.37	103.99ms
3- Phasor	87.12% ± 24.44	93.06% ± 10.96	31387.46ms

Table 2: Mean results for each algorithm over all database records, in terms of sensitivity, positive predictivity and the RR interval error

Algorithm	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
Pan & T.	98.06	99.79	99.94	99.80	100.00	100.00
Hamilton & T.	97.68	99.73	99.92	99.72	100.00	100.00
Phasor	0.12	88.02	99.29	87.12	99.86	100.00

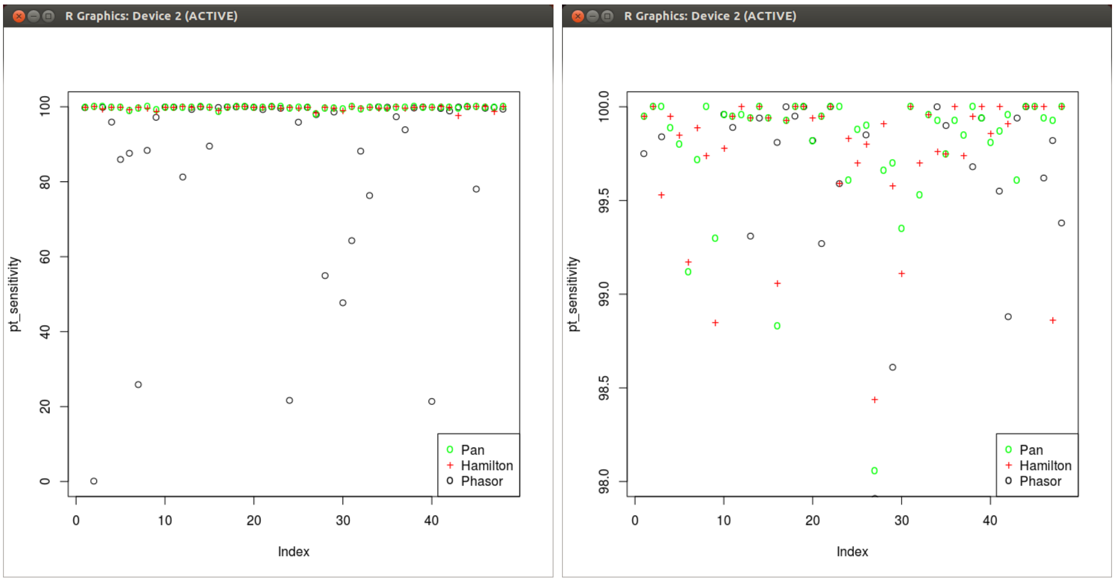
Table 3: Summary of *positive sensitivity* results for each algorithm

Algorithm	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
Pan & T.	98.13	99.89	100.00	99.89	100.00	100.00
Hamilton & T.	80.44	99.26	99.85	98.65	100.00	100.00
Phasor	52.59	89.97	99.09	93.07	99.95	100.00

Table 4: Summary of *predictivity* results for each algorithm

Compared algorithms	Sensitivity p value	Predictivity p value
Pan & T. vs Hamilton & T.	0.3679	0.0192
Pan & T. vs Phasor	0.0007	9.293e-05
Hamilton & T. vs Phasor	0.0008	0.0013

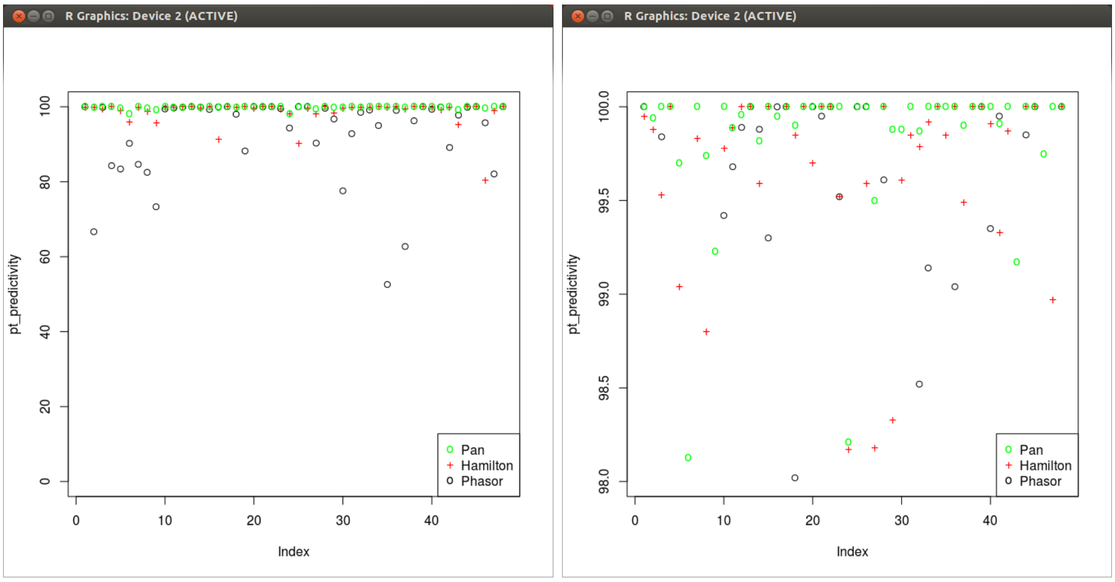
Table 5: Significant differences in the comparison among algorithms



(a) Comparison of all the results

(b) Zoom of the comparison

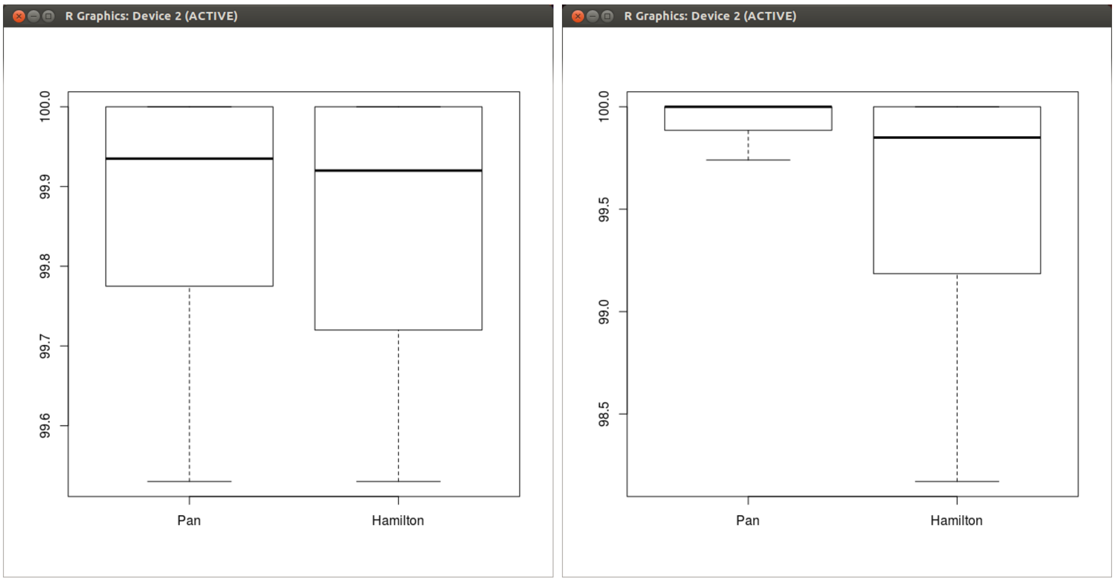
Figure 9: *Sensitivity* of the three algorithms for the 48 records



(a) Comparison of all the results

(b) Zoom of the comparison

Figure 10: *Positive predictivity* of the three algorithms for the 48 records



(a) Sensitivity results

(b) Predictivity results

Figure 11: Comparison of the sensitivity and predictivity boxplots in the detection algorithm with best results

As can be see in the results, Pan & Tompkins algorithm got the best results. With Hamilton & Tompkins we achieved good results, very close to those in the papers, but as can be seen in Figure 11b, the predictivity is significantly lower than Pan & Tompkins algorithm. Finally with the Phasor algorithm we got the worst results.

After developing the 3 algorithms we can conclude that Pan & Tompkins and Hamilton & Tompkins, can be implemented and reproduce results similar to those obtained by their authors, while for the case of the Phasor the results were not reproducible.

In the phasor work, they do not fully explained what are the necessary pre-processing operations to prepare the signal before applying the transform. Because of this ambiguity or our ignorance, we were not able to achieve better results.

5. Conclusions and future works

In this research work we have compared the performance of three of the best QRS detection algorithms, getting the results presented in the previous section.

As a result, besides the implementation of the three algorithms, we have begun the development of a ECG analysis software tool. This tool will permit to read ECGs records in different formats, ECG analysis, and visualization by different plots and automated validation of the obtained results. At this moment, this tool has already incorporated: three detection algorithms, different plotting functions and scripts for automatic validation using the WFDB library.

We have created our software as an open source tool to share with the scientific community. This way anyone can use it as a basis for their work, and expand it or improve it if they wish. In addition, having a detection algorithm with a good accuracy that greatly facilitates the detection or subsequent delineation of all ECG waves.

Our software is also suitable for those who want to start in electrocardiography, having the opportunity to test some functionality that until now was not available, and have access to source code and documentation explaining how work each part of the developed algorithms.

A future improvement will be: complete the package, reading different ECG formats, and validate the software over different databases.

6. References

[1] L. Sörnmo and P. Laguna, *Bioelectrical Signal Processing in Cardiac and Neurological Applications.* Elsevier Academic Press, June 2005.

[2] O. Pahlm and L. Sörnmo, "Software qrs detection in ambulatory monitoring — a review," *Medical and Biological Engineering and Computing*, vol. 22, no. 4, pp. 289–297, 1984.

[3] G. Moody and R. Mark, "The impact of the mit-bih arrhythmia database," *Engineering in Medicine and Biology Magazine, IEEE*, vol. 20, no. 3, pp. 45–50, 2001.

[4] R Core Team, *R: A Language and Environment for Statistical Computing.* R Foundation for Statistical Computing, Vienna, Austria, 2012. ISBN 3-900051-07-0.

[5] G. Friesen, T. Jannett, M. Jadallah, S. Yates, S. Quint, and H. Nagle, "A comparison of the noise sensitivity of nine qrs detection algorithms," *Biomedical Engineering, IEEE Transactions on*, vol. 37, no. 1, pp. 85–98, 1990.

[6] B.-U. Kohler, C. Hennig, and R. Orglmeister, "The principles of software qrs detection," *Engineering in Medicine and Biology Magazine, IEEE*, vol. 21, no. 1, pp. 42–57, 2002.

[7] N. Arzeno, Z.-D. Deng, and C.-S. Poon, "Analysis of first-derivative based qrs detection algorithms," *Biomedical Engineering, IEEE Transactions on*, vol. 55, no. 2, pp. 478–484, 2008.

[8] V. Afonso, W. J. Tompkins, T. Nguyen, and S. Luo, "Ecg beat detection using filter banks," *Biomedical Engineering, IEEE Transactions on*, vol. 46, no. 2, pp. 192–202, 1999.

[9] M. Okada, "A digital filter for the qrs complex detection," *Biomedical Engineering, IEEE Transactions on*, vol. BME-26, no. 12, pp. 700–703, 1979.

[10] J. Pan and W. J. Tompkins, "A real-time qrs detection algorithm," *Biomedical Engineering, IEEE Transactions on*, vol. BME-32, 1985.

[11] P. S. Hamilton and W. J. Tompkins, "Quantitative investigation of qrs detection rules using the mit/bih arrhythmia database," *Biomedical Engineering, IEEE Transactions on*, vol. BME-33, no. 12, pp. 1157–1165, 1986.

[12] J. Martinez, R. Almeida, S. Olmos, A. Rocha, and P. Laguna, "A wavelet-based ecg delineator: evaluation on standard databases," *Biomedical Engineering, IEEE Transactions on*, vol. 51, no. 4, pp. 570–581, 2004.

[13] C. Li, C. Zheng, and C. Tai, "Detection of ecg characteristic points using wavelet transforms," *Biomedical Engineering, IEEE Transactions on*, vol. 42, no. 1, pp. 21–28, 1995.

[14] B. Abibullaev and H. Seo, "A new qrs detection method using wavelets and artificial neural networks," *Journal of Medical Systems*, vol. 35, no. 4, pp. 683–691, 2011.

[15] Q. Xue, Y. Hu, and W. J. Tompkins, "Neural-network-based adaptive matched filtering for qrs detection," *Biomedical Engineering, IEEE Transactions on*, vol. 39, no. 4, pp. 317–329, 1992.

[16] D. Coast, R. Stern, G. Cano, and S. Briller, "An approach to cardiac arrhythmia analysis using hidden markov models," *Biomedical Engineering, IEEE Transactions on*, vol. 37, no. 9, pp. 826–836, 1990.

[17] R. Poli, S. Cagnoni, and G. Valli, "Genetic design of optimum linear and nonlinear qrs detectors," *Biomedical Engineering, IEEE Transactions on*, vol. 42, no. 11, pp. 1137–1141, 1995.

[18] A. Martínez, R. Alcaraz, and J. J. Rieta, "Application of the phasor transform for automatic delineation of single-lead ecg fiducial points," *Physiological Measurement*, vol. 31, no. 11, p. 1467, 2010.

[19] A. L. Goldberger, L. A. N. Amaral, L. Glass, J. M. Hausdorff, P. C. Ivanov, R. G. Mark, J. E. Mietus, G. B. Moody, C.-K. Peng, and H. E. Stanley, "PhysioBank, PhysioToolkit, and PhysioNet: Components of a new research resource for complex physiologic signals," *Circulation*, 2000.

[20] P. Lynn, "Online digital filters for biological signals: some fast designs for a small computer," *Medical and Biological Engineering and Computing*, vol. 15, no. 5, pp. 534–540, 1977.