Fast QRS Complex Detection Algorithm Based on RMS Shifting Concept for Heart Rate Estimation Using an Electrocardiogram

Gilles Yowel Massala Mboyi[1]*, Marius Massala Mbindzoukou[2], Pierre Moukeli Mbindzoukou[2], Jung-Hoon Han[1]

[1]Electromagnetic Application Technology (EAT) Laboratory, Jeju National University, Jeju, Korea
[2]African Laboratory of Computer Science and Applied Mathematics, African Institute of Computer Science, Libreville, Gabon
Email: *gilles.massala@stu.jejunu.ac.kr

Abstract

Accurate detection of QRS complex in electrocardiogram (ECG) signals is essential for reliable estimation of the heart rate. However, traditional QRS detection algorithms often have low performance in the presence of various types of noise and signal abnormalities and require additional memory resources to track undetected peaks. In this work, we propose a novel QRS complex detection algorithm based on the root mean square (RMS) shifting concept. The concept of RMS shifting consists to remove an amount proportional to the RMS of the pre-processed signal for moving all the P and T waves of the electrocardiogram toward the negative part of the y-axis and keep only the R peaks in the positive part. Then, all the roots can be softly detected using the corollary of the intermediate value theorem known as Bolzano's theorem. The detection of R peaks is ensured by Rolle's theorem. Our proposed model has been implemented and evaluated on a diverse set of ECG datasets and its performances are comparable to that of spectral analysis based on the FFT algorithm widely used nowadays. For the construction of this model, we used a sample of an electrocardiogram signal from the MIT/BIH Arrhythmia database stored and provided by Simulink. The peaks detected by our algorithm have been verified and confirmed by the well-known Pan-Tompkins Algorithm used by MATLAB. Then, our model has been applied to a publicly shared electrocardiogram database provided by a Japanese physiological laboratory. The comparison of the heart rate estimated by the proposed method and the spectral analysis method shows a low absolute error average (0.89 bpm), a low relative error average (1.37%), a low root mean square error (1.05 bpm), and a correlation coefficient very close to 1 (0.9938). We also measured the CPU time to assess the performance of our proposed

method and we found that our algorithm is twice as fast as the conventional method. Therefore, we inferred that our model is reliable for estimating heart rate for electrocardiography applications.

Subject Areas

Biomedical Signal Processing

Keywords

Bolzano's Theorem, Electrocardiogram, Heart Rate, Intermediate Value Theorem, Pan-Tompinks Algorithm, QRS Complex, Rolle's Theorem

1. Introduction

Electrocardiogram (ECG) is a recording of the heart's electrical activity. Since the typical electro-cardiogram waveform is periodic, during one period we can obviously identify three stages: The P wave, the QRS complex, and the T wave as depicted in Figure 1. An electrocardiogram is the succession of these three phases. However, among these phases, researchers have a special interest in QRS complex detection owing to the possibility of diagnosing heart diseases. Indeed, The QRS complex shape can be used to identify various heart abnormalities. For instance, a large QRS complex (extended duration) may indicate issues with the heart's conduction system. A QRS complex with an amplitude that is too high or too low may indicate left ventricular hypertrophy or an infiltrative myocardial illness, respectively [1]. Therefore, finding and extracting every QRS complex component of an ECG wave is a crucial electrocardiography task.

On the other hand, the detection of all R-peaks allows researchers to calculate the heart rate, and then infer the heart condition. Thanks to algorithms for detecting QRS complexes, it is possible to provide physicians with digital decision-support tools for monitoring cardiac patients in real-time [2]. These recent decades, numerous QRS complex detection algorithms have been implemented. Pan and Tompkins have developed a real-time algorithm for the detection of the QRS complexes of ECG signals based on digital analysis of slope, amplitude, and width [3]. Before applying their peak detection algorithm, they applied a set of signal preprocessing subroutines. This algorithm is still widely used nowadays, for example by the well-known signal processing software MATLAB. We will show briefly the results of its pre-processing algorithms on an electrocardiogram signal in the forthcoming paragraphs. This algorithm has been used in this work to verify and confirm the QRS complexes detected by our proposed algorithm.

Friesen *et al.* performed a comparison of the noise sensitivity of nine QRS detection algorithms for a normal, single-channel lead II, synthesized ECG corrupted with five different types of synthesized noise [4]. They inferred that none of the algorithms were able to detect all QRS complexes without any false positives for all of the noise types at the highest noise level and algorithms based on
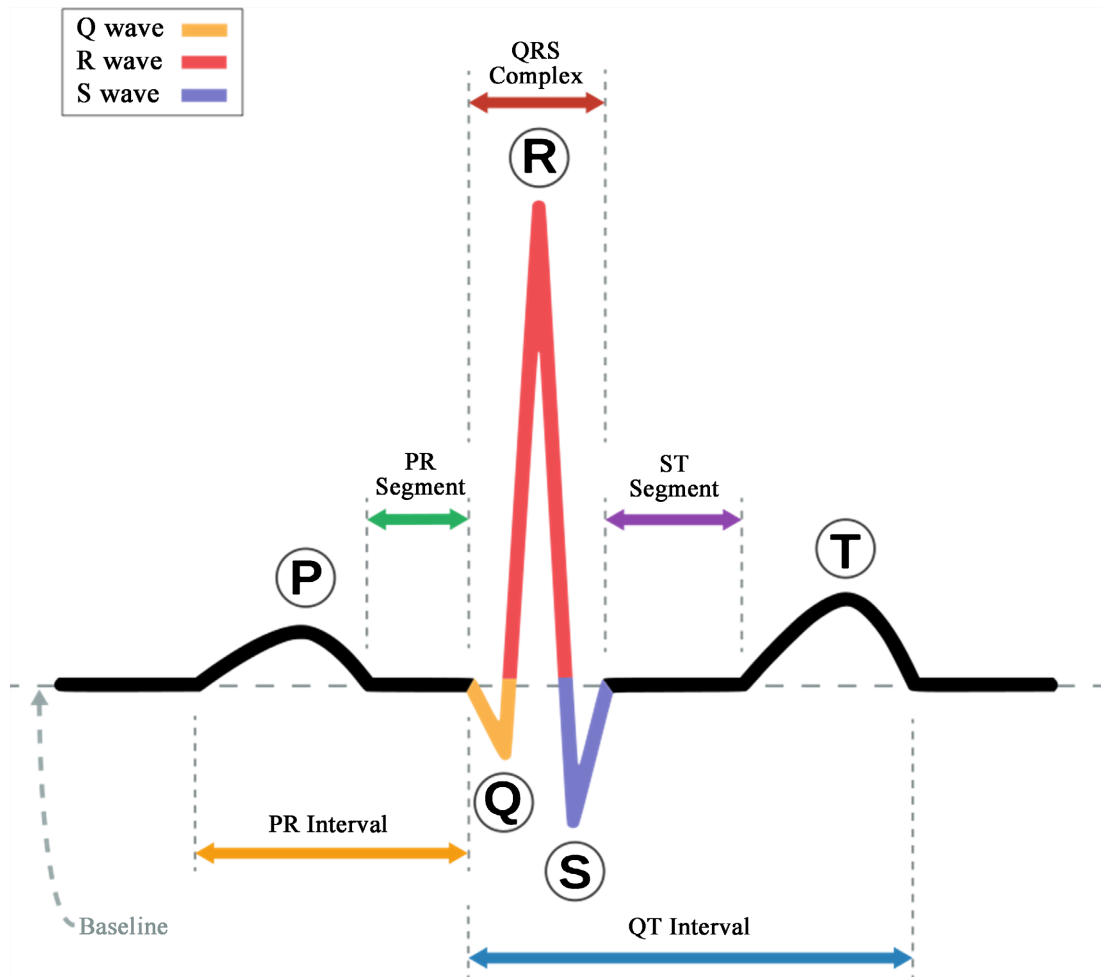
Figure 1. Representation of one period of the typical electrocardiogram waveform.

amplitude and slope had the highest performance for EMG-corrupted ECG. Furthermore, Nouira and al. made a comparative study of QRS complex detection in ECG [5]. They described several mathematical approaches for digital filtering and discrete wavelet transform (DWT) filtering, which are both used to filter ECG data. Additionally, they presented also two primary windowing-based R peak identification techniques.

The different QRS complex detection algorithms can be classified into several categories. Firstly, we have those that are based on the adaptive thresholding technique [3] [6] [7] [8] [9]. The second set of algorithms is based on the mathematical transformations: Empirical mode decomposition [10], Fourier transform (conventional method), Hilbert transform [8] [11], and Wavelet transform [9] [12]. And, nowadays, owing to the development of artificial intelligence, we have a new trend of QRS complexes detection algorithms based on artificial intelligence, machine learning, and neural network techniques [13]. Usually, most QRS complex detection algorithms fall into one or a combination of these categories.

Nowadays with the expansion of connected objects which have very limited resources, it is necessary to design non-greedy algorithms if we wish to integrate

electrocardiography techniques into the IoT. However, in the literature, the articles do not present a study of the CPU resources mobilized by their algorithms and moreover, those based on threshold techniques require additional memory resources to track undetected peaks. The challenge of this work is to design a fast and efficient algorithm using simple numerical analysis tools while proposing a performance study in terms of CPU resources used.

In this paper, we present another approach for QRS complex detection based on the concept of RMS shifting. This algorithm is preceded by two pre-processing algorithms for baseline wander removal and powerline interference attenuation, and the centralization of the pre-processed signal to remove the dc offset. Throughout this article, we are going to deal progressively with the presentation of the RMS shifting method and pre-processing algorithms in section 1. In section 2, we are going to present the different steps of the Pan-Tompinks algorithm briefly and use it to verify the R-peaks detected by our proposed algorithm on a sample of electrocardiogram signal from the MIT/BIH Arrhythmia database stored in Simulink. Finally, in the last section, we are going to use our algorithm for heart rate estimation and make a comparison with the conventional spectral method.

2. Methods

In this section, we are going to describe the mathematical formulation of the RMS shifting principle, and then apply it to a sample of an electrocardiogram signal extracted from the MIT/BIH Arrhythmia database stored in Simulink. (Figure 2)
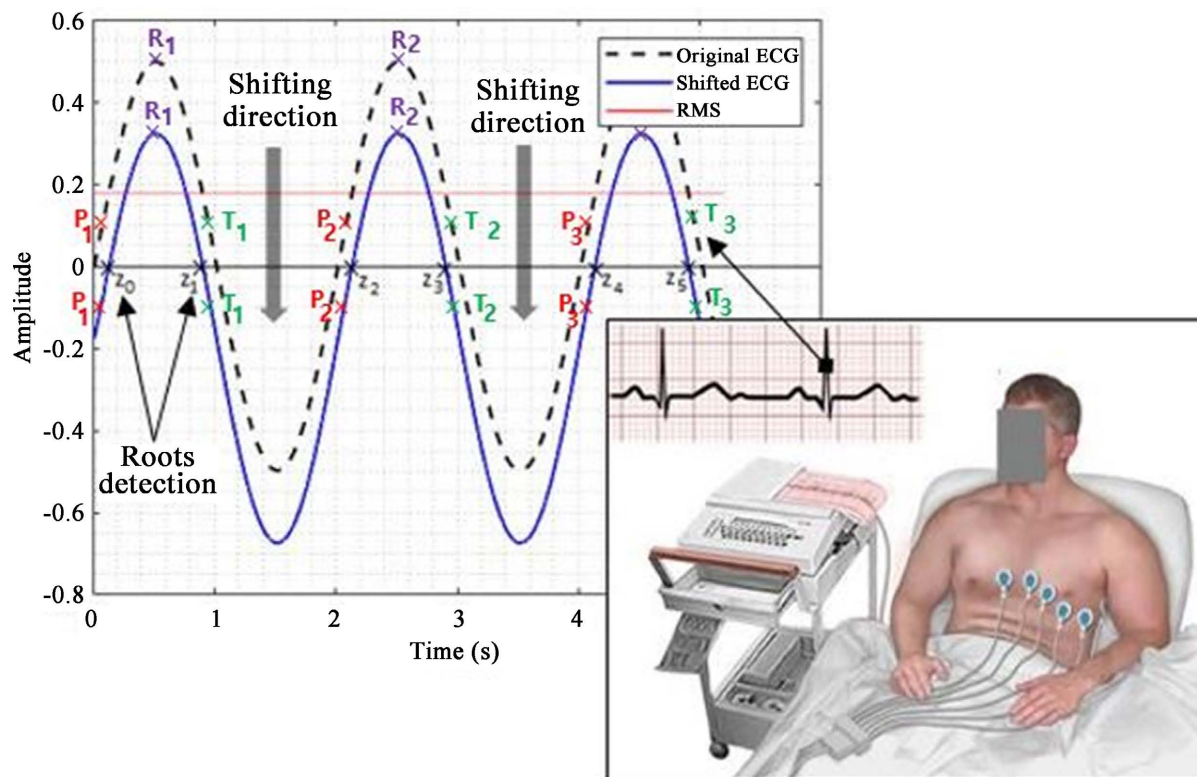

Figure 2. RMS shifting principle.

We will also deal with pre-processing algorithms for baseline wander removal and signal denoising by wavelet transform through a graphical user interface toolbox.

2.1. Mathematical Formulation

According to Abeysekera *et al.*, the QRS complex clearly resembles a 'slightly' distorted sine wave [14]. Based on this assumption, we undertook to give an estimation of the RMS of the electrocardiogram since an ECG is nothing but a periodical repetition of QRS complexes. Let $V(t) = V_{max} \sin(\omega t + \varphi)$ a sine wave representing an electrocardiogram, with $\omega = 2\pi/T$ its pulsation, T the period, and V_{max} the maximum amplitude which is nothing but the highest R-peak. The root means square of such kind of curve is given by:

$$\mathrm{RMS}^2 = \frac{1}{T}\int_{t_0}^{t_0+T} V^2(t)\,\mathrm{d}t$$

$$= \frac{1}{T}\int_{t_0}^{t_0+T} V_{max}^2 \sin^2(\omega t + \varphi)\,\mathrm{d}t$$

$$= \frac{V_{max}^2}{T}\int_{t_0}^{t_0+T} \sin^2(\omega t + \varphi)\,\mathrm{d}t$$

$$= \frac{V_{max}^2}{T}\int_{t_0}^{t_0+T} \frac{1 - \cos(2\omega t + 2\varphi)}{2}\,\mathrm{d}t$$

$$= \frac{V_{max}^2}{2T}\int_{t_0}^{t_0+T}\mathrm{d}t - \frac{V_{max}^2}{2T}\int_{t_0}^{t_0+T}\cos(2\omega t + 2\varphi)\,\mathrm{d}t$$

$$\mathrm{RMS}^2 = \frac{V_{max}^2}{2}$$

Therefore: $\mathrm{RMS} = \dfrac{V_{max}}{\sqrt{2}}$.

Throughout this article, we are going to consider this expression of the root means square. The key idea behind the RMS shifting formalism is to remove the amount of the RMS in the original signal to keep only the R-peaks in the positive part of the shifted signal. Then, the identification of R-peaks can be performed by the detection of all the zeros of the shifted signal. However, owing to the baseline wander cancellation and the denoising subroutines applied on the original electrocardiogram signal, the standard deviation of the R-peaks is naturally affected. Accordingly, instead of removing the previously calculated RMS, we set out to remove an amount proportional to this RMS value to account for the decay of the standard deviation.

Let $V_0(t)$ be the original signal and $V_s(t)$ be the shifted signal. Hence, the expression of the shifted signal with respect to the original signal is given by:

$$V_s(t) = V_0(t) - \frac{V_{max}}{\lambda\sqrt{2}}, \ \text{ with } \lambda \in \mathbb{N}\setminus\{0\}. \tag{1}$$

We can get empirically the optimal value of the coefficient in the denominator $\lambda = 4$. Using the equation above, we have shifted all the P and T waves in the negative part along the y-axis and it remains only the R-peaks in the positive

part. Thus, we can detect the zeros of the shifted signal using the intermediate value theorem. This theorem is stated as follows:

Theorem (Intermediate value theorem). *Let f be a continuous function on* [*a,* *b*] *and s be a number with: f*(*a*) < *s* < *f*(*b*)*. Then there exists some x between a and b such that f*(*x*) = *s.*

In our case, *f* stands for the shifted electrocardiogram, and *s* = 0 since we are investigating the roots of the shifted signal. Actually, this formulation of the intermediate value theorem is not suitable for our use case. To make this theorem useful for our algorithm, we have been interested in its corollary rather than its previous formulation. The used corollary is known as Bolzano's theorem:

Corollary (Bolzano's theorem). *If a continuous function has values of opposite signs inside an interval, then it has a root in that interval.*

Based on Bolzano's theorem, we inferred the following criterion for zero detection in the shifted signal expression as follow: $\forall t_0$, $V_s(t_0)$. $V_s(t_0 + \Delta t) \leq 0$ $\Rightarrow$ root detection. Here, Δt represents the time-marching step of our algorithm.

After detecting all the zeros of the shifted signal, the presence of R-peaks is ensured by Rolle's theorem. This theorem is stated below:

Lemma (Rolle's theorem). *If real-valued function f is continuous on a closed interval* [*a, b*]*, differentiable on the open interval*]*a,b*[*, and f*(*a*) = *f*(*b*)*, then there exists at least one c in the open interval*]*a,b*[*such that f*′(*c*) = 0*.*

In our use case, *c* is nothing but the position of an R-peak in the time axis. Let N_{root} be the total number of roots detected in the shifted signal. Using Rolle's theorem, we can prove that the amount of QRS complexes is given by:

$$N_{QRS} = \begin{cases} \dfrac{N_{root}}{2} & \text{if } N_{root} \text{ is an even number} \\ \dfrac{1 - N_{root}}{2} & \text{if } N_{root} \text{ is an odd number} \end{cases} \tag{2}$$

Finally, if the total duration of the raw data acquisition experiment is noted *d* in seconds, we can obviously infer the heart rate (HR) using the following formula:

$$\text{HR} = \frac{N_{QRS}}{d} \times 60 \tag{3}$$

It is really important to notice that the RSM shifting scheme should be preceded by pre-processing algorithms to ensure a good result. After recording the raw data, we need to cancel the baseline drift and denoise the signal. Then centralize the obtained signal before performing the RMS shifting scheme. The block diagram of our proposed algorithm is given in Figure 3.

2.2. Data Acquisition

A recording of the body surface potentials caused by the electrical activity of the heart is called an electrocardiogram (ECG). The ECG recording can be used by clinicians to diagnose further and assess a patient's overall health and heart condition.

Three main sources of pre-recorded ECG data including biomedical databases (such as the MIT-BIH Arrhythmia Database), ECG simulators, and real-time
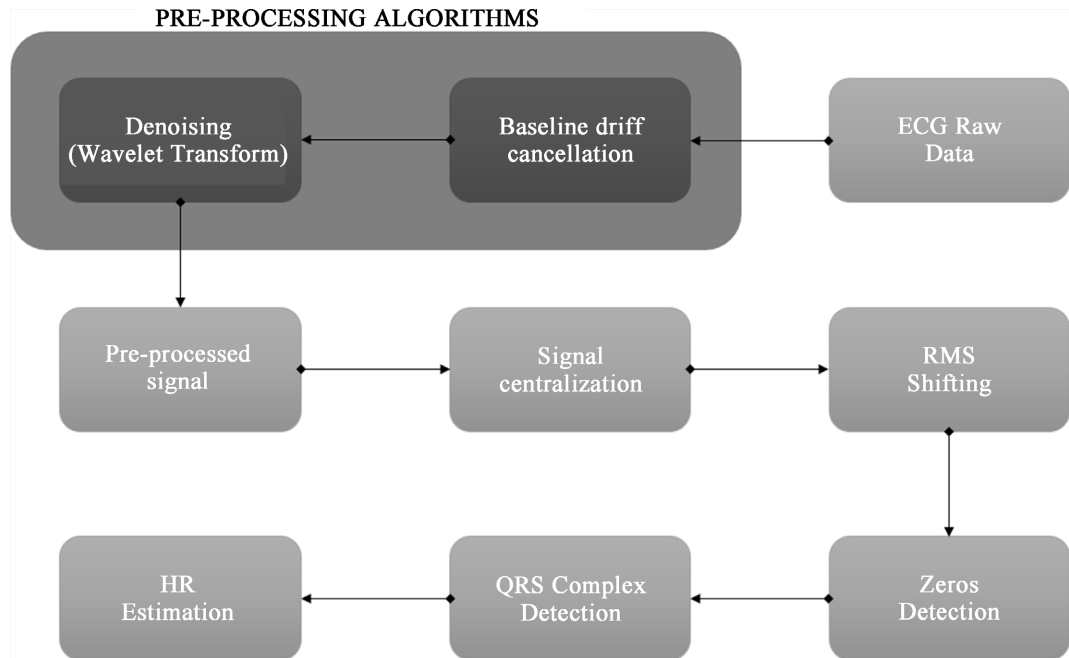
Figure 3. The block diagram of our proposed method.

ECG data gathering are where the majority of the ECG signals utilized in the creation and testing of biomedical signal processing algorithms come from.

In this paper, to verify our model, we used a sample of the MIT-BIH Arrhythmia Database stored in Matlab. One collection of genuine ECG data was recorded and came from a healthy volunteer with an average heart rate of 82 beats per minute (bpm). Before being fed to the 12-bit ADC, this ECG data was pre-filtered and boosted by the analog front end. This signal has the following features: a sampling frequency of 360 Hz, an additive uniformly distributed measurement noise of 0.005 mV, and a standard deviation of heart rate of 1 bpm [15]. The scheme of the data acquisition simulation implemented in Simulink is depicted in **Figure 4**.

The electrocardiogram signal processing block details are given in **Figure 5**. This describes the different stages of the well-known Pan-Tompinks algorithm. We will give further explanation about this model in the next section.

The raw data displayed by the scope are depicted in **Figure 6**. It's a sample of the electrocardiogram signal recorded for 10 seconds. On the same graph, we can see below the waveform of the filtered ECG. Especially, the P and T waves have been strongly attenuated. Consequently, the filtering makes it possible to highlight the R-peaks of the electrocardiogram signal.

Furthermore, we can also see the evolution of the heart rate versus time in **Figure 7**, and the variation of the adaptive threshold implemented in the Pan-Tompinks algorithm.

2.3. Pre-Processing Algorithms

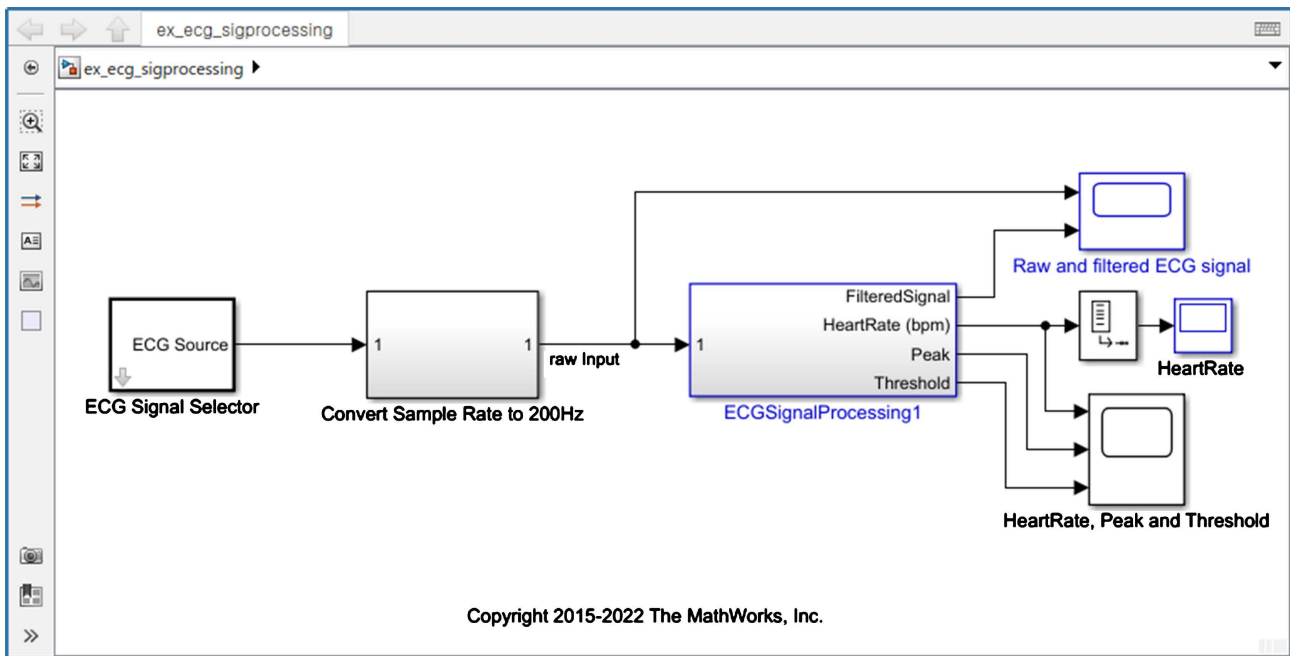While recording the heart's electrical activity, the electrocardiogram signal is

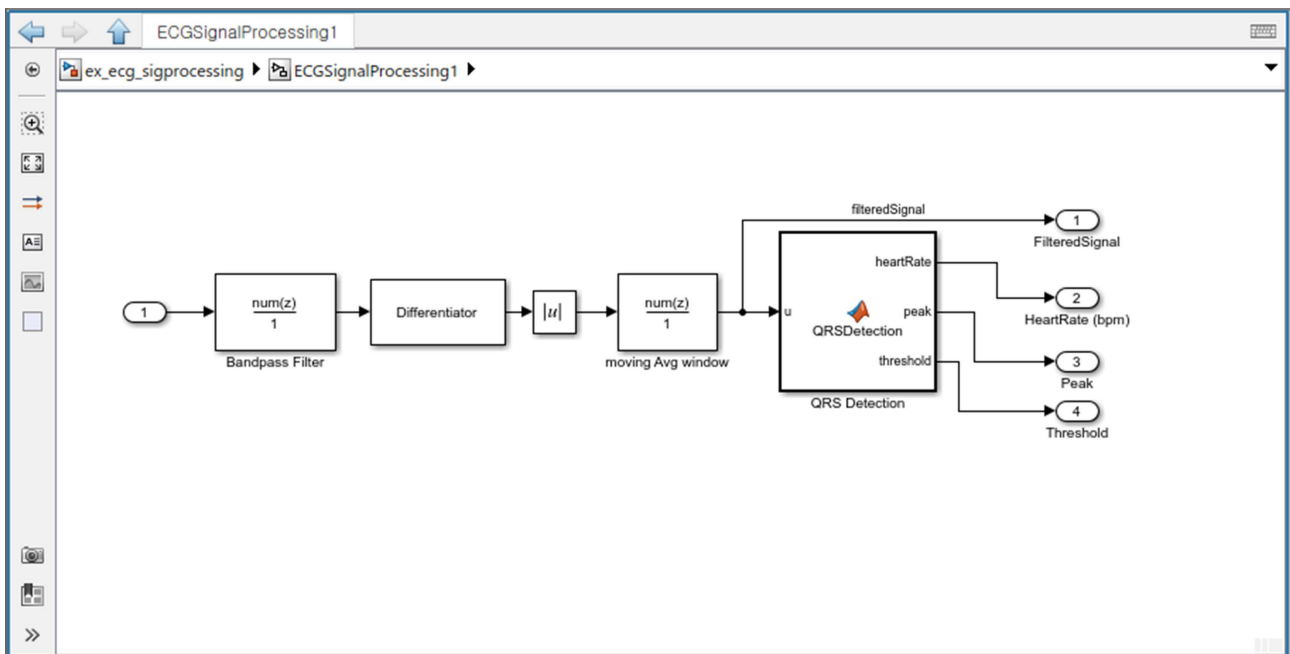**Figure 4.** Data acquisition scheme implemented in Simulink.

Figure 5. Detailed ECG signal processing block.

corrupted by numerous phenomena such as muscle activity noise or artifacts due to electrode motion. Therefore, considerable attention should be paid to applying algorithms for the purpose of removing baseline wander and powerline interference present in the ECG raw data. To reduce variations in beat morphology that do not have a cardiac cause, baseline wander must be eliminated. We can see below the electrocardiogram raw data extracted from Simulink in **Figure 8(a)**, and the waveform after applying the baseline wander removal
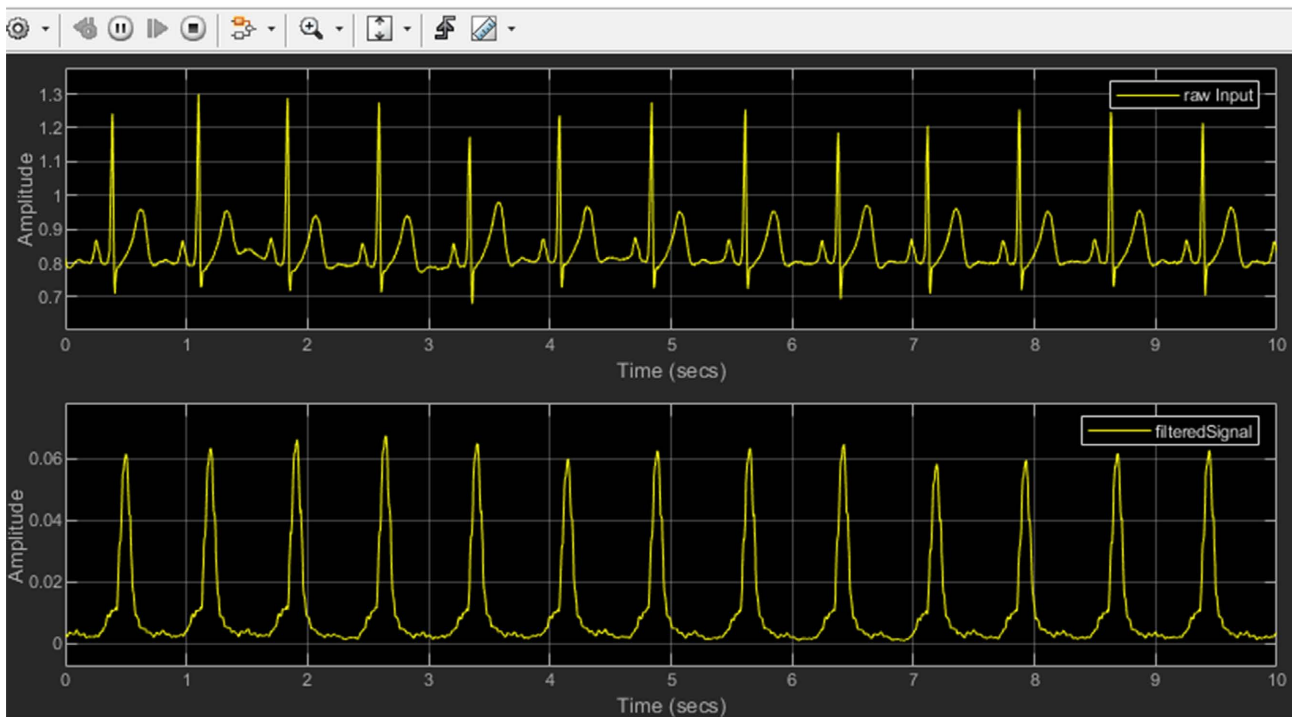
Figure 6. Raw and filtered ECG signal sample extracted from Simulink.
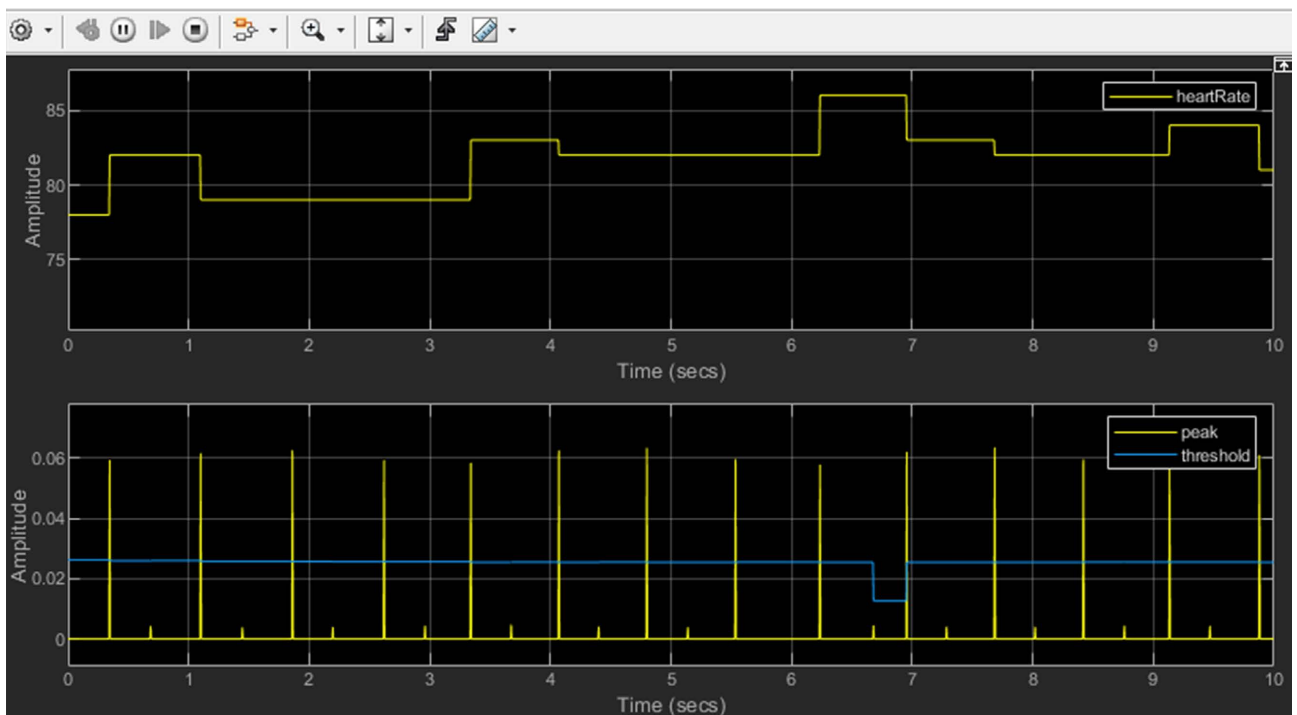

Figure 7. Heart rate, peak, and threshold of the extracted sample.

subroutine in **Figure 8(b)**.

Electromagnetic fields caused by power supply represent also a common noise source that introduces spurious waveforms in the electrocardiogram raw data. This type of noise is known as powerline interference. To reduce the effect of

such kind of noise we undertook to denoise the ECG raw data using the wavelet transform. The signal processing software Matlab provides a graphical user interface (GUI) to denoise a one, two, or three-dimensional signal through the wavelet transform. This GUI is called wavelet Analyzer toolbox [16]. In our use case, the electrocardiogram was a one-dimensional time-domain signal. In Table 1, we can find the parameters used in our case and a screenshot of the wavelet Analyzer Toolbox in Figure 9.

Finally, we applied the RMS shifting scheme onto the resulting denoised signal to detect the R-peaks. RMS shifting algorithm has been performed after the centralization of the denoised signal. All the peaks detected are pinpointed in the graph shown in Figure 10.

Furthermore, our algorithm indicates that 54 R-peaks have been detected. Since the total duration of the recording experiment is $d = 39.72$ s , thus we can readily infer the value of the heartbeat rate of this sample:

$$\text{HR} = \frac{N_{QRS}}{d} \times 60 = \frac{54}{39.72} \times 60 = 81.57 \text{ bpm}.$$

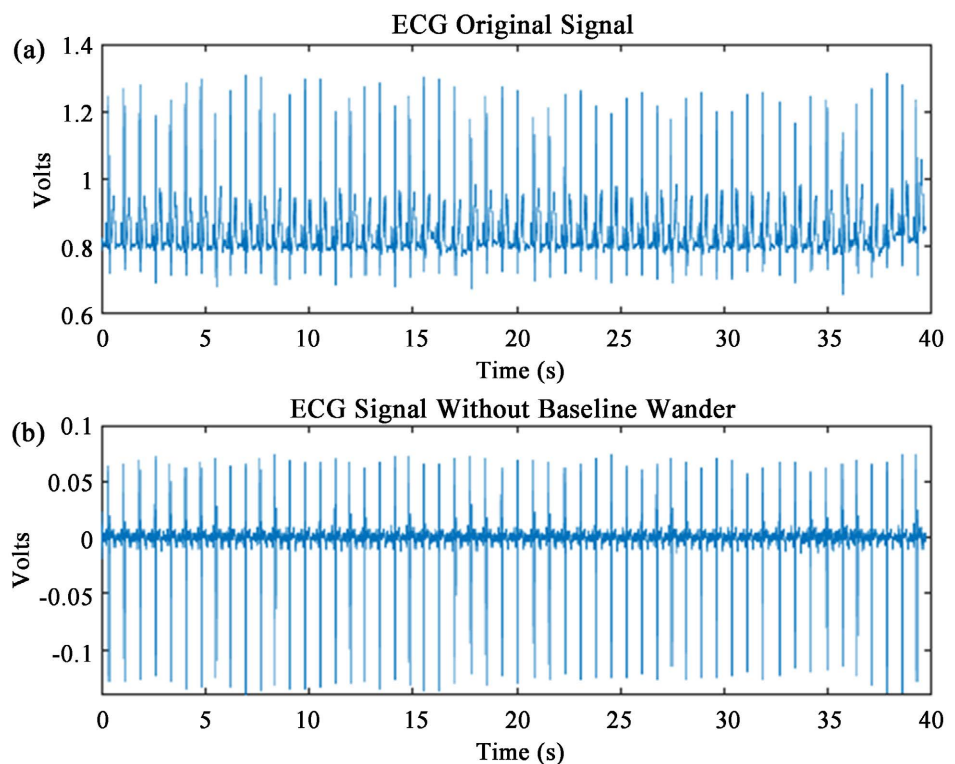This result is in agreement with the features of this ECG signal described in

Figure 8. Baseline wander removal.

Table 1. Wavelet analyzer toolbox settings used in this work.

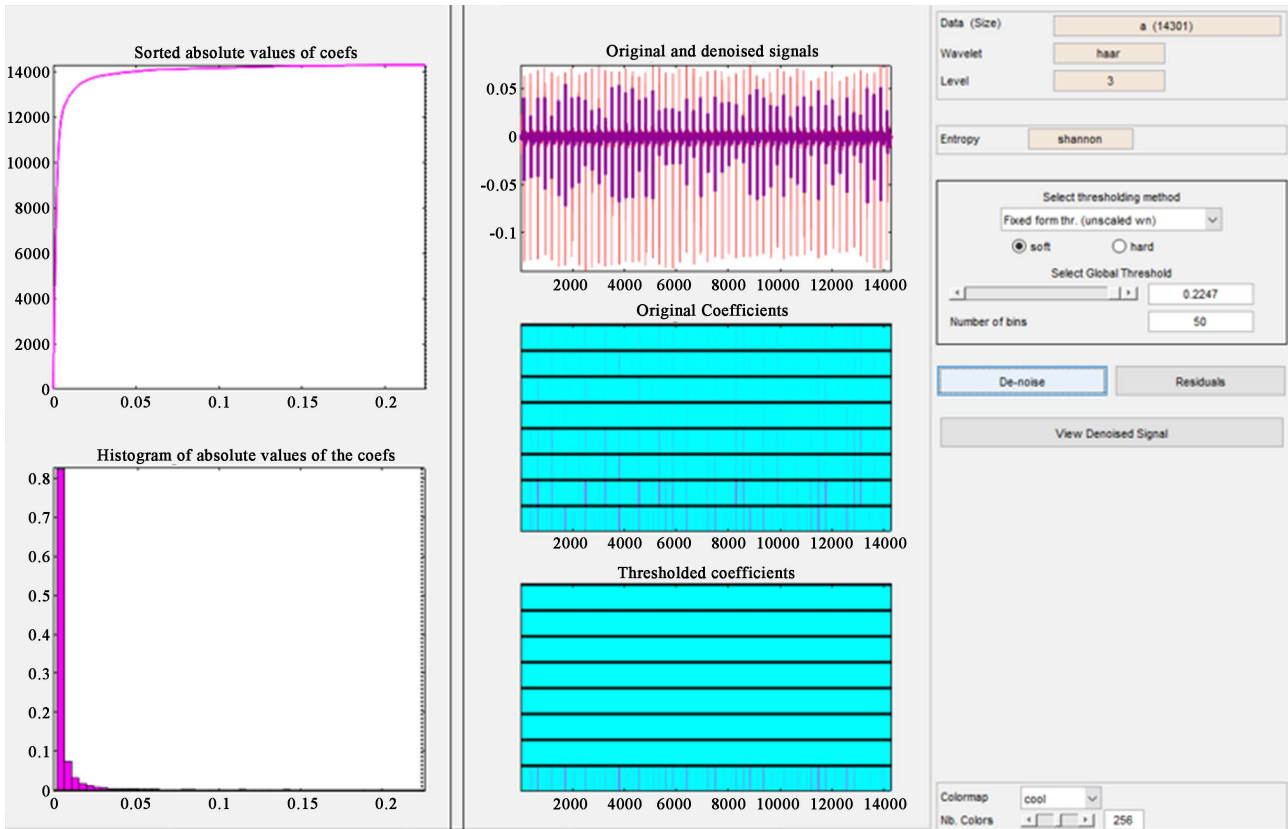| Parameter	Type	Level	Entropy	Thresholding method	Global threshold	Number of bins
Value	*haar*	*3*	*Shannon*	*Soft fixed thr.*	*0.2247*	*50*

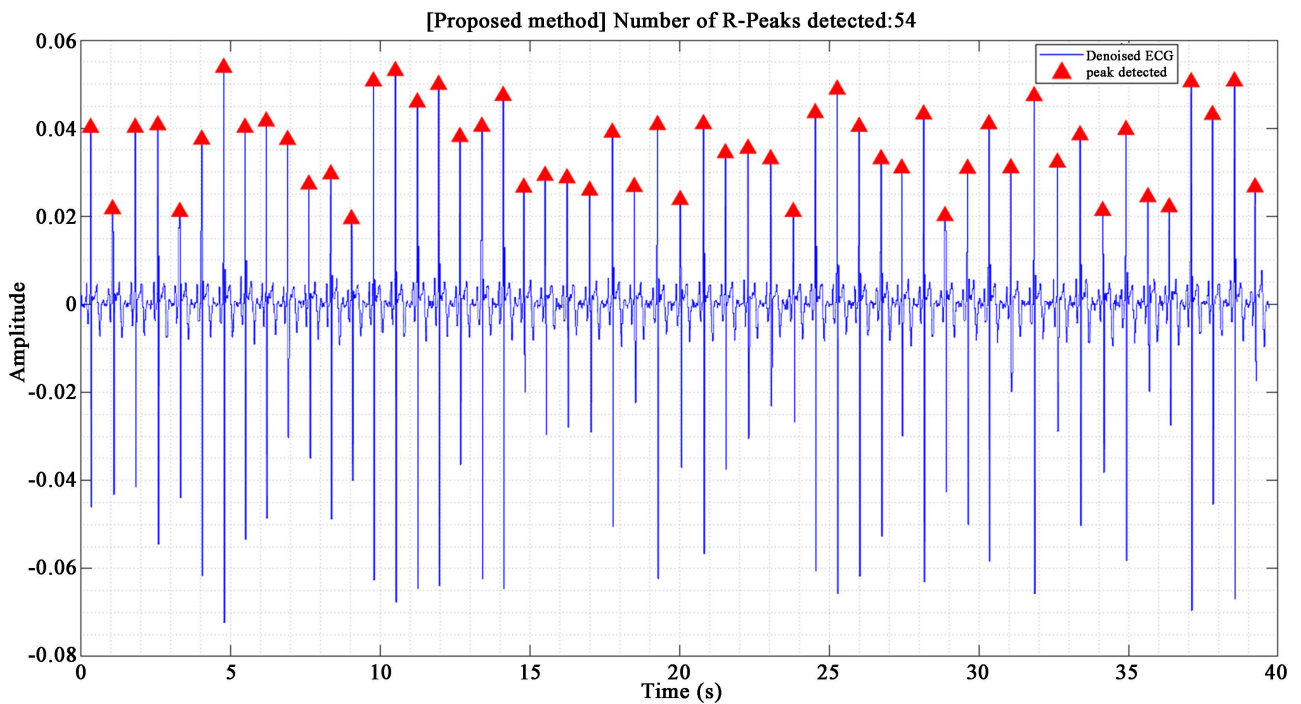**Figure 9.** ECG signal denoising through Wavelet Analyzer toolbox.

Figure 10. R-peaks detected by our proposed method.

the previous section. Indeed, according to Matlab documentation, the sample of the ECG signal provided has an average heart rate of 82 bpm with a standard deviation of 1 bpm. Therefore, our algorithm accurately estimated the heart rate value for this sample. Additionally, to verify and confirm again the R-peaks detected by our algorithm, we undertook to use another QRS complex detection algorithm. We used the well-known Pan-Tompinks algorithm implemented in Matlab to accomplish this task. In the next section, we will outline this algorithm and show its effects on the ECG signal.

3. Outcomes Verification with Another QRS Complex Detection Algorithm

3.1. Pre-Processing Algorithms and ECG Signal Filtering

Simulink has been used to create a real-time QRS detection algorithm on the presumption that the input ECG signal's sampling frequency is always 200 Hz (or 200 samples/s) [15].

The recorded real ECG data, on the other hand, can have various sampling frequencies ranging from 200 Hz to 1000 Hz, for example, 360 Hz in our use case. A sample rate converter block is used to convert the sample rate to 200 Hz in order to bridge the disparate sampling frequencies. To make sure the duration of the incoming ECG signal is greater than the sample-rate converter block's computed decimation factor, a buffer block is inserted as depicted in Figure 4. To provide a windowed estimate of the energy in the QRS frequency region, the ECG signal is filtered. These steps comprise the filtering process: FIR bandpass filter with cut-off frequencies of 5 to 26 Hz, taking the derivative of the bandpass filtered signal, taking the absolute value of the signal, and averaging the absolute value over an 80 ms window [15]. All of these stages are presented in the ECG signal processing block details (Figure 5). The effects of each pre-processing algorithm on the signal are depicted in Figure 11.

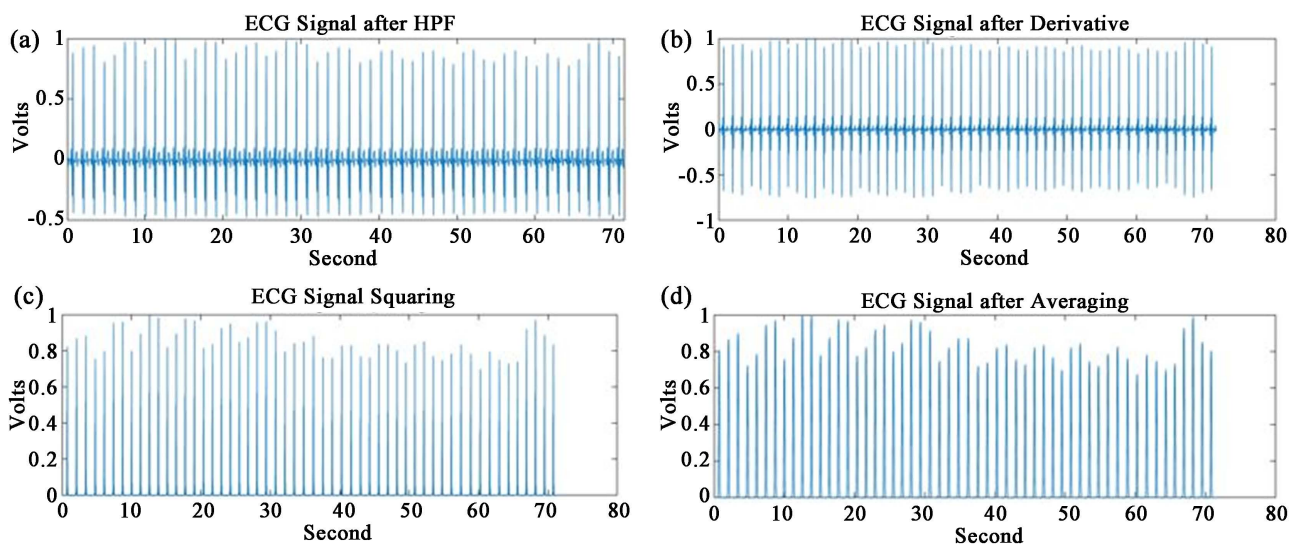**Figure 11.** Pan-Tompinks algorithm steps applied to the extracted signal.

3.2. QRS Complex Detection Using Pan-Tompinks Algorithm

Simulink has been used to create a real-time QRS detection algorithm on the presumption that the input ECG signal's sampling frequency is always 200 Hz (or 200 samples/s) [15].

Real-time peak identification of the filtered ECG signal is performed by the QRS detection block (Figure 5). This block implements the well-known Pan-Tompinks algorithm. Based on the mean estimation of the average noise peak and the average QRS peak, the detection threshold is automatically modified. Depending on whether the observed peak is above the threshold, it is either categorized as a QRS complex or as noise. The PIC-based QRS detector used in that block is referenced by the following QRS detection rules: Ignore all peaks that precede or follow larger peaks by less than 196 ms (306 bpm); if a peak occurs, check to see whether the raw signal contains both positive and negative slopes if true report a peak is found, otherwise, the peak represents a baseline shift; if the peak is larger than the detection threshold, classify it as a QRS complex, otherwise, classify it as noise.

If no QRS has been detected within 1.5 R-to-R intervals, but there is a peak that was larger than half the detection threshold, and that peak followed the preceding detection by at least 360 ms, classify that peak as a QRS complex [3] [15]. Further details on Pan-Tompinks algorithm and its behavior in noisy signals are presented in [17]. After following these rules, we got the results shown in Figure 12.

As well as our proposed method, the Pan-Tompinks algorithm also detected 54 R-peaks. The peaks detected overlap perfectly with those detected by our

Figure 12. QRS complexes detected through Pan-Tompinks algorithm.

proposed method. This result proves that the peaks detected using the RMS shifting scheme are reliable.

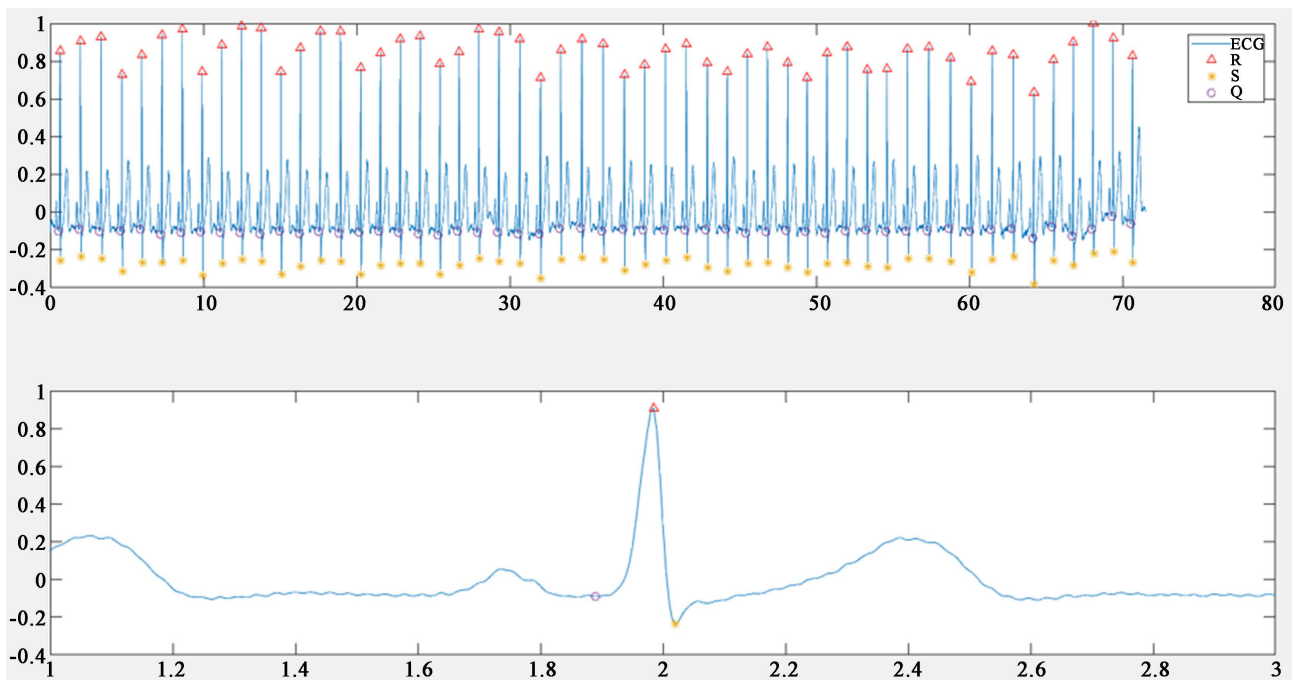As well as our proposed method, the Pan-Tompinks algorithm also detected 54 R-peaks. The peaks detected overlap perfectly with those detected by our proposed method. This result proves that the peaks detected using the RMS shifting scheme are reliable.

However, let us mention that the time duration in the Pan-Tompinks graph is different from that used in our scheme. This is due to the difference in the sampling frequencies used in the two schemes. In our case, we kept the original sampling frequency of 360 Hz whereas in the Pan-Tompinks case, a sample rate converter block has been used to reduce the sampling frequency to 200 Hz. The modification of the sampling frequency affects the time vector since its size remains the same in the two cases. Indeed, the time step value of the time axis is nothing but the inverse of the sampling frequency.

In the last section, we will use the RMS shifting algorithm to estimate the heart rate of 9 healthy volunteers. The results obtained will be compared with those obtained by the conventional method of calculating the heart rate based on the spectral analysis of the ECG signal.

4. Experimental Configuration and Data Acquisition

A physiological laboratory in Japan has publicly shared a well-characterized medical radar dataset that can be reused by biomedical researchers to study signal processing algorithms for noncontact vital signs measurement [18]. They provided noncontact respiratory and cardiac signal datasets obtained through the use of a medical radar, they also simultaneously collected reference signals using an electrocardiogram (ECG) and respiratory belt transducer. The files associated with this dataset are licensed under a Creative Commons Attribution 4.0 International license [CC BY 4.0] [19].

Doppler radars operating at 24.25 GHz (New Japan Radio, NJR4262) and 10.525 GHz (New Japan Radio, NJR4178J) were used to capture the medical radar signals. Using a touch instrument, the ECG and respiratory belt signals were concurrently recorded (BIOPAC, BN-REPEC). The ADC (USB-6003, National Instruments) at a sampling rate of 1000 Hz with 16 bits resolution was used to convert the analog signals to digital signals, and LabVIEW data acquisition software was used to record the results.

The experiment was carried out on nine healthy volunteers with an average age of 24, 5 males, and 4 females. Measurements were made on each volunteer for 10 minutes. The volunteers were told to remain in a supine position on a bed, resting. The V5 instructions were followed for attaching the ECG, and the respiratory belt was placed on the volunteers' abdomen for breath signal measurements (Figure 13). A sample of the waveforms recorded is depicted in Figure 14.

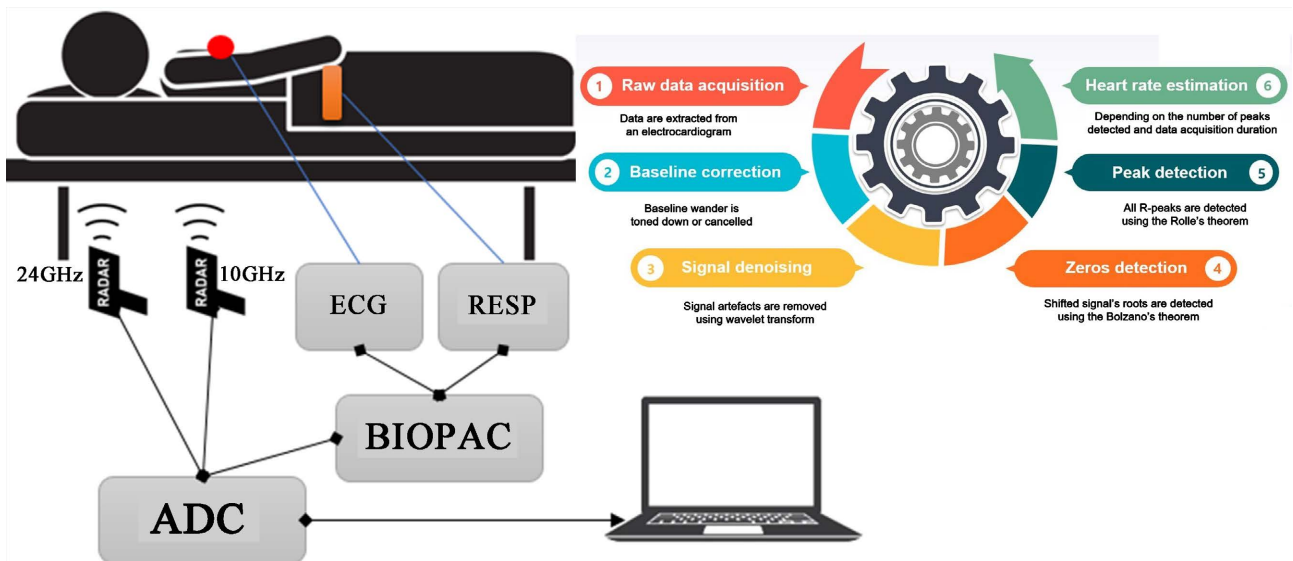Furthermore, to pre-process the signals and determine the respiratory and
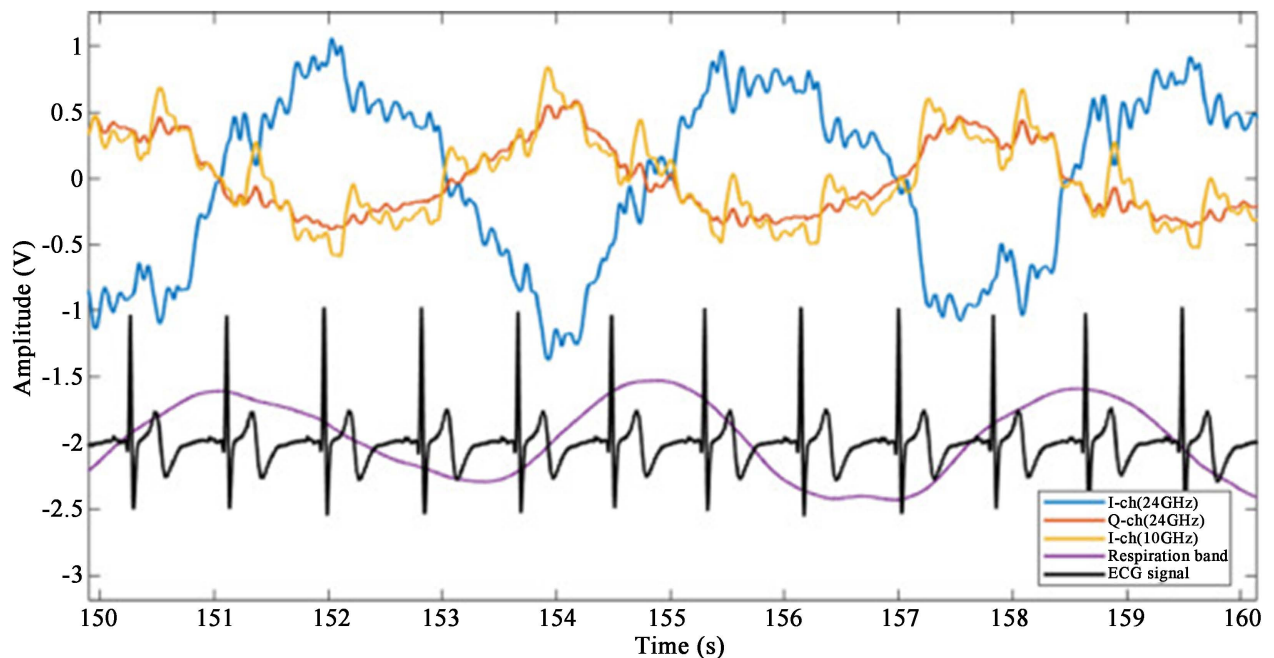
Figure 13. Experiment setup.

Figure 14. A sample of waveforms recorded [18].

cardiac rates, they also implemented Matlab code. The dataset composed by the clinical application LabVIEW data files (.lvm) and CSV files (.csv) and Matlab source code can be found in [19]. A screenshot of the outcomes of this algorithm is displayed in **Figure 15**.

5. Results and Discussion

5.1. Accuracy and Reliability Assessment

In this work, we particularly focused on the electrocardiogram data. For each volunteer, we collected a sample with an acquisition time of one minute ranging

from 100 s to 160 s. In order to assess the accuracy of our proposed method, we first set out to detect R peaks using both our algorithm and the Pan-Tompinks algorithm implemented in Matlab. The results of the two algorithms are shown in Table 2. The results clearly show that for all cases, both algorithms found the same number of peaks. This implies that the two algorithms give the same heart rate values for all cases since the heart rate only depends on the number of peaks detected and the acquisition time.

Therefore, for further assessment of our algorithm, we undertook to compare the heart rate that had been calculated using our proposed method and the conventional method. The conventional method consists of the spectral analysis of the signal. Let $s(t)$ be a signal, its spectrum is given by the Fourier transform: In this work, we particularly focused on the electrocardiogram data. For each volunteer, we collected a sample with an acquisition time of one minute ranging from 100 s to 160 s. In order to assess the accuracy of our proposed method, we first set out to detect R peaks using both our algorithm and the Pan-Tompinks
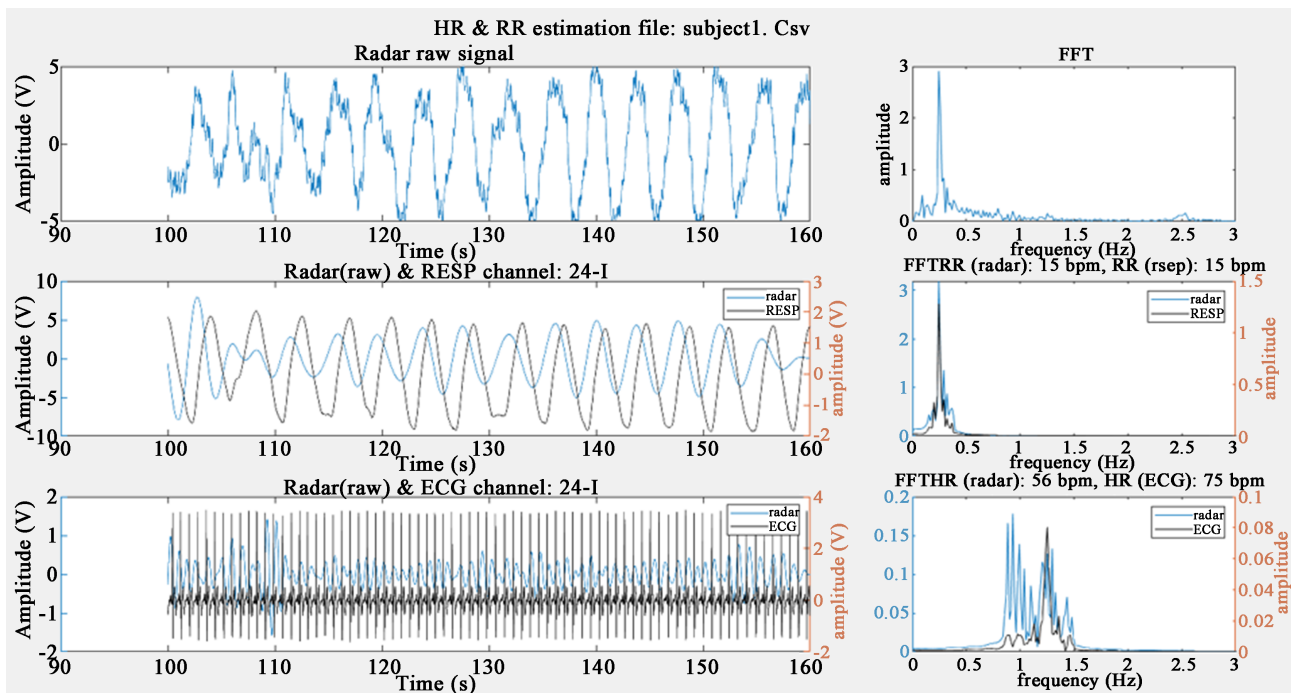

Figure 15. Breath rate and Heart rate estimation using data recorded through the radar, the respiratory belt, and the electrocardiogram.

Table 2. Number of peaks detected by both algorithms.

Experiment	Volunteer 1	Volunteer 2	Volunteer 3	Volunteer 4	Volunteer 5	Volunteer 6	Volunteer 7	Volunteer 8	Volunteer 9
Proposed Method	75	77	71	71	63	84	66	54	57
Pan-Tompinks Algorithm	75	77	71	71	63	84	66	54	57

algorithm implemented in Matlab. The results of the two algorithms are shown in Table 2. The results clearly show that for all cases, both algorithms found the same number of peaks. This implies that the two algorithms give the same heart rate values for all cases since the heart rate only depends on the number of peaks detected and the acquisition time.

Therefore, for further assessment of our algorithm, we undertook to compare the heart rate that had been calculated using our proposed method and the conventional method. The conventional method consists of the spectral analysis of the signal. Let $s(t)$ a signal, its spectrum is given by the Fourier transform:

$$S(f) = FT\{s(t)\} = \int_{-\infty}^{\infty} s(t) e^{-j2\pi ft} dt \tag{4}$$

To be able to utilize the Fourier transform while working with a digital signal, the digital Fourier Transform (DFT) should be used. The fast Fourier transform (FFT) is an efficient and fast way of evaluating the DFT of a signal. After performing the FFT algorithm, the highest peak in the frequency domain is nothing but the heart rate.

We applied the conventional method and the RMS shifting scheme to the dataset and the comparison of the results of the two algorithms is shown in Figure 16.

The results show that for the two first cases, the values found by the proposed method are exactly the same as those found through the conventional method. However, there is a difference of 1 bpm from the third volunteer to the eighth one, whereas we have a difference of 2 bpm in the last case. This difference can be justified by the fact that the conventional method doesn't account for the
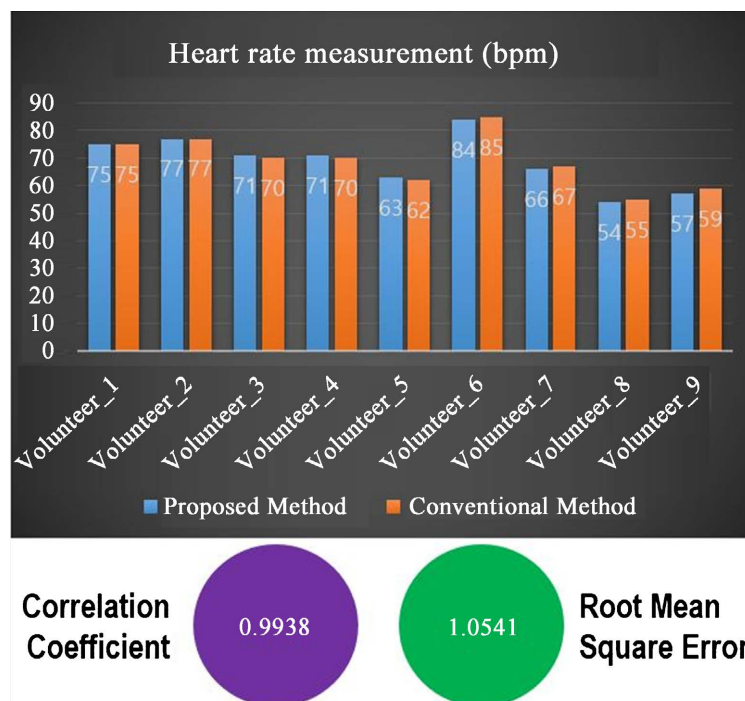

Figure 16. Comparative study of heart rates measured by our proposed method.

powerline interface, the baseline wander, electrode motion noise, or other artifacts that can make the signal noisy.

On the other hand, we also performed the calculation of the correlation coefficient (r) and the root mean square error (RMSE) between these two sequences of data to assess their statistical relationship:

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_i \|x_i - y_i\|^2} \tag{5}$$

$$r = \frac{\sum(x_i - \overline{x})(y_i - \overline{y})}{\sqrt{\sum(x_i - \overline{x})^2 \sum(y_i - \overline{y})^2}} \tag{6}$$

where x_i and y_i are respectively the heartbeat rate of the *ith* volunteer calculated through our proposed method and the conventional method, and $\overline{x}$ and $\overline{y}$ are their respective average. After performing the calculation of the correlation coefficient, we found $r = 0.9938$ (Figure 16). Thus, the results of the two methods have a strong linear relationship.

The Root Mean Square Error (RMSE) is a statistical coefficient that calculates the amount of error between two data sets. In other words, it contrasts a value that was anticipated with one that was observed or known. The anticipated and observed values are closer together when the RMSE is small. As shown in Figure 16, we found the value of RMSE = 1.0541 bpm. Therefore, this means that there is only 1 bpm difference between the heart rate values estimated by the two methods.

Finally, we have also calculated the averages of the absolute and relative errors of our proposed methods with respect to the conventional method. The outcomes are displayed in the graph in Figure 17. In both cases, the error values are very
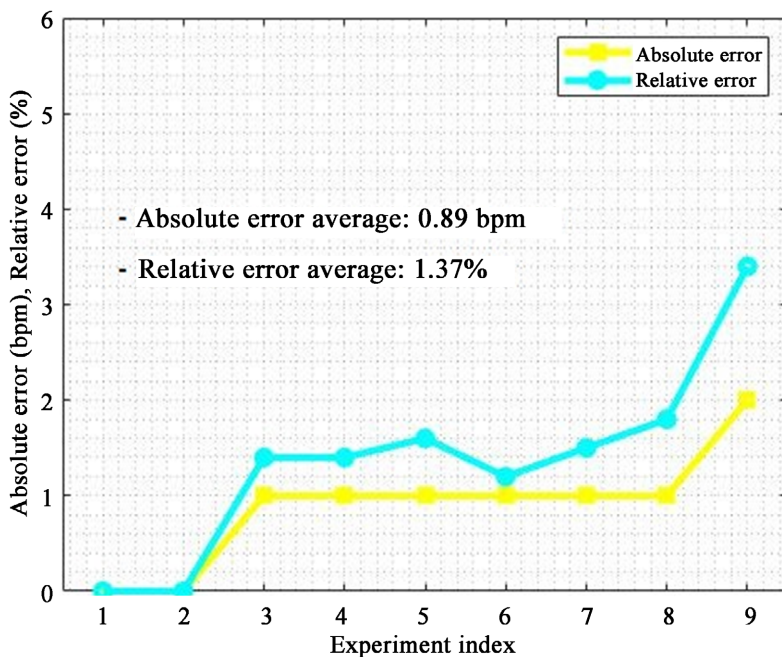

Figure 17. Absolute and relative errors of our proposed method with respect to reference device measurements.

low. We found an absolute error average of 0.89 bpm and a relative error average of 1.3%. Consequently, all these indicators confirm the accuracy and reliability of our algorithm for heart rate estimation for electrocardiography applications.

5.2. Performance Assessment

In order to assess the performance of our algorithm, we set out to measure the CPU time of our proposed method, the Pan-Tompinks algorithm, and the conventional method. Then, we compared them. CPU time, also known as process time, is the length of time a central processing unit (CPU) was employed to carry out an operating system's or computer program's instructions [20]. The overall empirical effectiveness of two functionally equivalent algorithms is measured using the CPU time. The algorithm with the lowest CPU time is the most efficient in terms of execution speed.

We compared the three algorithms with different data sizes. The first experiment was done using data collected over a period of 1 minute. Since the sampling frequency is 1000 Hz, hence, the size of the input data is 60,000 samples. We carried out a total of 10 experiments. When we went from an experience of rank n to an experience of rank n + 1 we added 60,000 additional values. The objective was to test the speed of the algorithms with increasingly large data sizes.

The algorithms were executed on an INTEL Core i7-9700 type processor, with a clock frequency of 3 GHz (8 CPUs) and a 64-bit architecture. The RAM memory size was 8 GB. The results are shown in Figure 18. It can be clearly seen that our proposed method has a shorter execution cycle than the other two methods. We also note that for our algorithm, the CPU time remains almost constant
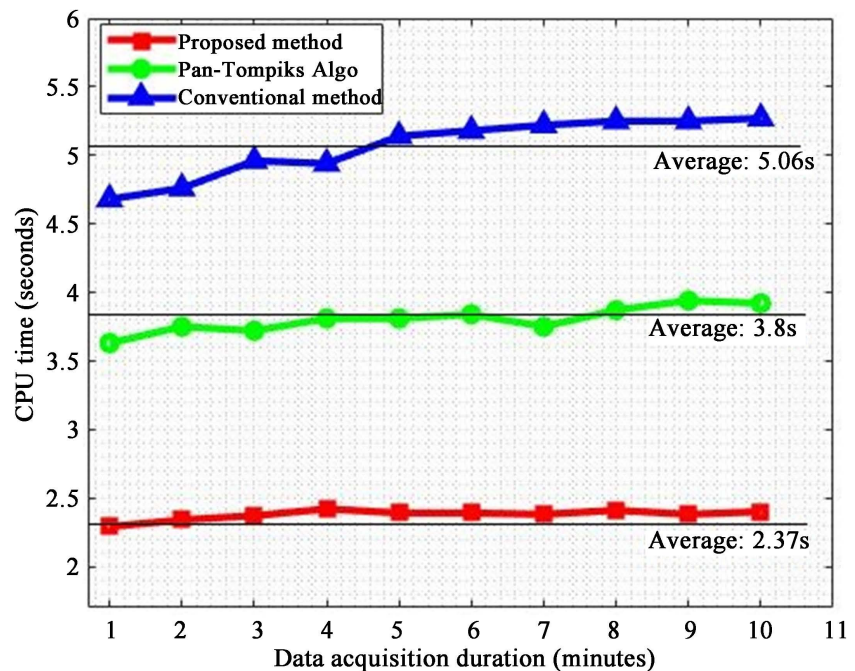

Figure 18. CPU time measured on our proposed method, the Pan-Tompinks algorithm, and the Conventional method.

while the size of the data increases, whereas the CUP time of the other algorithms increases when the size of the data increases. Also, by analyzing the CPU time averages, we realize that our algorithm is twice as fast as the conventional method.

The parameters that we have studied show that our algorithm presents quite reliable performances, an execution speed twice superior to the conventional method based on spectral analysis and its quasi-insensitivity to the mass of input data. The last two properties make our algorithm a good alternative to be implemented in systems with low resources like connected objects. However, the speed of this algorithm can still be improved by integrating parallelism. The main idea is to split the main program into independent subroutines and to execute these subroutines in parallel on different processor cores.

6. Conclusion

We presented throughout this article a time-domain approach for heart rate estimation based on RMS shifting principle. Before applying RMS shifting scheme, we performed firstly pre-processing algorithms for baseline wander cancellation and powerline interference attenuation by denoising the signal using the wavelet transform. Then, we also performed the signal centralization procedure to remove the dc offset. We applied our proposed procedure to a sample signal from MIT/BIH Arrhythmia database stored and provided by Simulink, and we accurately estimated the heart rate which complied with the features given in Matlab documentation. Using the well-known Pan-Tompinks algorithm implemented in Matlab, we verified and confirmed the peaks detected by the proposed method. Finally, we set out to apply this method to a publicly shared medical dataset provided by a Japanese physiological laboratory. The experiment was carried out on nine healthy volunteers. The comparison with the results of the conventional method based on spectral analysis shows that our algorithm is reliable for electrocardiography applications. By measuring the CPU time using data of increasing sizes, we observed that our algorithm was very efficient and was on average twice as fast as the conventional method. Further research is expected to increase the speed of our algorithm by exploiting the potential of parallel computing.

Authors' Contributions

This article is part of a Ph.D. work supervised by Pr. Han Jung-Hoon. All authors have read and agreed to this version of the manuscript.

Funding

Availability of Data and Materials

The data presented in this study and the materials used are available on request

from the corresponding author.

Conflicts of Interest

The authors declare no conflicts of interest.

References

[1] Klingspor, M. (2015) Hilbert Transform: Mathematical Theory and Applications to Signal Processing. LiTH-MAT-EX.

[2] Sutton, R.T., Pincock, D., Baumgart, D.C., *et al.* (2020) An Overview of Clinical Decision Support Systems: Benefits, Risks, and Strategies for Success. *NPJ Digital Medicine*, **3**, Article No. 17. https://doi.org/10.1038/s41746-020-0221-y

[3] Pan, J. and Tompkins, W.J. (1985) A Real-Time QRS Detection Algorithm. *IEEE Transactions on Biomedical Engineering*, **BME-32**, 230-236. https://doi.org/10.1109/TBME.1985.325532

[4] Friesen, G.M., Jannett, T.C., Jadallah, M.A., Yates, S.L., Quint, S.R. and Nagle, H.T. (1990) A Comparison of the Noise Sensitivity of Nine QRS Detection Algorithms. *IEEE Transactions on Biomedical Engineering*, **37**, 85-98. https://doi.org/10.1109/10.43620

[5] Nouira, I., Abdallah, A.B., Kouaja, I. and Bedoui, M.H. (2012) Comparative Study of QRS Complex Detection in ECG. *International Journal of Medical, Health, Bio-engineering*, **6**, 5.

[6] Rahman, S., Kim, Y. and Kim, S. (2018) A Simple and Robustness Algorithm for ECG R-Peak Detection. *Journal of Electrical Engineering and Technology*, **13**, 2080-2085.

[7] Kaur, A., Agarwal, A., Agarwal, R. and Kumar, S. (2019) A Novel Thresholding Technique for R-Peak Detection in ECG. 2019 *Second International Conference on Advanced Computational and Communication Paradigms* (*ICACCP*), Gangtok, 25-28 February 2019, 1-4. https://doi.org/10.1109/ICACCP.2019.8882913

[8] Sahoo, S., Biswal, P., Das, T. and Sabut, S. (2016) De-Noising of ECG Signal and QRS Detection Using Hilbert Transform and Adaptive Thresholding. *Procedia Technology*, **25**, 68-75. https://doi.org/10.1016/j.protcy.2016.08.082

[9] Behbahani, S. and Dabanloo, N.J. (2011) Detection of QRS Complexes in the ECG Signal Using Multiresolution Wavelet and Thresholding Method. 2011 *Computing in Cardiology*, Hangzhou, 18-21 September 2011, 805-808.

[10] Hadj Slimane, Z.E. and Naït-Ali, A. (2010) QRS Complex Detection Using Empirical Mode Decomposition. *Digital Signal Processing*, **20**, 1221-1228. https://doi.org/10.1016/j.dsp.2009.10.017

[11] Amhia, H. and Wadhwani, A.K. (2021) Designing an Optimum and Reduced Order Filter for Efficient ECG QRS Peak Detection and Classification of Arrhythmia Data. *Journal of Healthcare Engineering*, **2021**, Article ID: 6542290. https://doi.org/10.1155/2021/6542290

[12] Narayana, K.V.L. and Rao, A.B. (2011) Wavelet Based QRS Detection in ECG Using MATLAB. *Innovative Systems Design and Engineering*, **2**, 60-69.

[13] Peterkova, A. and Stremy, M. (2015) The Raw ECG Signal Processing and the Detection of QRS Complex. 2015 *IEEE European Modelling Symposium*, Madrid, 6-8 October 2015, 80-85.

[14] Abeysekera, R.M.S.S., Bolton, R.J., Westphal, L.C. and Boashash, B. (1986) Patterns

in Hilbert Transforms and Wigner-Wille Distributions of Electrocardiogram Data. *IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP* 1986, Tokyo, 7-11 April 1986, 1973-1978.

[15] The Official Website of Matlab (for Matlab Documentation) (2022) Real-Time ECG QRS Detection.
https://ch.mathworks.com/help/dsp/ug/real-time-ecg-qrs-detection.html?s_tid=src htitle_Real-Time%20ECG%20QRS%20Detection_1

[16] Denoise a Signal with the Wavelet Signal Denoiser. The official website of Matlab (for Matlab documentation).
https://ch.mathworks.com/help/wavelet/ug/denoise-a-signal-with-the-wavelet-signal-denoiser.html

[17] Fariha, M.A.Z., Ikeura, R., Hayakawa, S. and Tsutsumi, S. (2020) Analysis of Pan-Tompkins Algorithm Performance with Noisy ECG Signals. *Journal of Physics: Conference Series*, **1532**, Article ID: 012022.
https://doi.org/10.1088/1742-6596/1532/1/012022

[18] Edanami, K. and Sun, G. (2022) Medical Radar Signal Dataset for Non-Contact Respiration and Heart Rate Measurement. *Data in Brief*, **40**, Article ID: 107724.
https://doi.org/10.1016/j.dib.2021.107724

[19] Edanami, K. (2021) Medical Radar Signal Dataset.
https://data.mendeley.com/datasets/6rp6wrd2pr/2

[20] Wikipedia (2023) CPU Time. https://en.wikipedia.org/wiki/CPU_time

Abbreviations

BPM: Beats Per Minute
CPU: Central Processing Unit
DFT: Discrete Fourier Transform
DWT: Discrete Wavelet Transform
ECG: Electrocardiography
EMG: Electromyography
FFT: Fast Fourier Transform
GUI: Graphical User Interface
HR: Heart Rate
MIT/BIH: Massachusetts Institute of Technology/Beth Israel Hospital
RAM: Random Access Memory
RMS: Root Mean Square
RMSE: Root Mean Square Error