January 2015

Computationally Efficient QRS Detection Analysis In Electrocardiogram Based On Dual-Slope Method

M. Riadh Riadh Arefin

COMPUTATIONALLY EFFICIENT QRS DETECTION ANALYSIS IN ELECTROCARDIOGRAM BASED ON DUAL-SLOPE METHOD

by

M. Riadh Arefin
Bachelor of Science, University of Dhaka 2011

A Thesis

Submitted to the Graduate Faculty

of the

University of North Dakota

in partial fulfillment of the requirements

for the degree of

Master of Science

Grand Forks, North Dakota
August
2015

This thesis, submitted by M. Riadh Arefin in partial fulfillment of the requirements for the Degree of Master of Science from the University of North Dakota, has been read by the Faculty Advisory Committee under whom the work has been done and is hereby approved.

Reza Fazel-Rezai

_____

Dr. Reza Fazel-Rezai

Kouhyar Tavakolian

_____

Dr. Kouhyar Tavakolian

Sima Noghanian

_____

Dr. Sima Noghanian

This thesis is being submitted by the appointed advisory committee as having met all of the requirements of the School of Graduate Studies at the University of North Dakota and is hereby approved.

Wayne Swisher

_____

Wayne Swisher,
Dean of the Graduate School

PERMISSION

Title Computationally Efficient QRS Detection Analysis in Electrocardiogram based on Dual-Slope Method.

Department Electrical Engineering

Degree Master of Science

In presenting this thesis in partial fulfillment of the requirements for a graduate degree from the University of North Dakota, I agree that the library of this University shall make it freely available for inspection. I further agree that permission for extensive copying for scholarly purposes may be granted by the professor who supervised my thesis work or, in his absence, by the chairperson of the department or the dean of the Graduate School. It is understood that any copying or publication or other use of this thesis or part thereof for financial gain shall not be allowed without my written permission. It is also understood that due recognition shall be given to me and to the University of North Dakota in any scholarly use which may be made of any material in my thesis.

M. Riadh Arefin
June 18th , 2015

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Table Page

ACKNOWLEDGEMENTS

First, I express my gratitude to the Department of Electrical Engineering of University of North Dakota, for giving me the opportunity to research and pursue a master degree here. My solemn gratitude to Dr. Reza Fazel-Rezai who has been Philosopher and Guide to me. While being my supervisor in this academic period, he has enriched and trained me with vast knowledge and competitive skills.

I am very much grateful to my co-advisor, Dr. Kouhya Tavakolian and committee members , Dr. Sima Noghanian for their continuous support and advice.

I am also very grateful to Farzad Khosrow-khavar for reviewing the work and Andrew Blaber for helping with the data acquisition of lower body negative pressure (LBNP) database.

To my parents - My idol of strength and courage

ABSTRACT

A dramatic growth of interest for wearable technology has been fostered by recent technological advances in sensors, low-power integrated circuits and wireless communications. This interest originates from the need of monitoring a patient over extensive period of time. For cardiac patients, wearable heart monitoring sensors have already become a life-saving intervention ensuring continuous monitoring during daily life. Therefore, it is essential for an accurate monitoring and diagnosis of heart patients. Patients can be equipped with wireless, miniature and lightweight sensors. The sensors temporarily store physiological data and then periodically upload the data to a database server. These recorded data sets are then analyzed to predict any possibility of worsening patient's situation or explored to assess the effect of clinical intervention. To obtain accurate response with less computational complexity as well as long battery life time, there is a demand of developing fast and accurate algorithm and prototypes for wearable heart monitoring sensors. A computationally efficient QRS detection algorithm is indispensable for low power operation on electrocardiogram (ECG) signal.

In need of detecting QRS complex, most of the early works were proposed based on derivatives of ECG signal. They can be easily implemented with high computational speed. But owing to the inherent variability in ECG, these methods are highly affected by large derivatives of baseline noises. Algorithms based on neural network (NN) showed relatively robust performance against noise but requires exhaustive training and estimation of model parameter. On the other hand, wavelet based methods have the

choice problem of mother wavelet. Hence, none of these methods is suitable for giving a long battery performance in wearable devices with high accuracy.

Recently, Wang et al. proposed a novel dual slope QRS detection algorithm which has less computational complexity as well as high accuracy. Considering that the width of the QRS complex is relatively fixed, this algorithm is based on the fact that the largest change of slope usually happens at the peak of QRS complex. The hardware requirement is also low. However, the method has a set of time consuming slope calculations on both sides of each sample. To avoid such time consuming slope calculation, only one sample on each side can be highlighted. In addition, the multiplication of the left and right hand side slope should give us a very high value in QRS complex.

The goal of this thesis is to develop a new computationally efficient method to detect QRS complexes and compare with the other renowned QRS detection algorithms. MIT-BIH arrhythmia database based on patients of different heart diseases and database containing ECG from healthy subjects are used. To analyze the performance, false negative (FN) and false positive (FP) are evaluated. A false negative (FN) occurs when algorithm fails to detect an actual QRS complex quoted in the corresponding annotation file of the database record and a false positive (FP) means a false beat detection. Error rate (ER) , Sensitivity (Se) and Specificity (Sp) are calculated using FP and FN.

CHAPTER 1

INTRODUCTION

The heart is one of most crucial organs in the entire human body without which survival is next to impossible. Thus, the development of methods for monitoring the state of cardiac heart is of the utmost importance in medical science. Electrocardiography (ECG) is regarded as one of the most important cardiac investigations available for monitoring the functionality of the heart. It is simple and old, yet it can provide a wealth of useful information and remains an essential part of the assessment of cardiac patients.

1.1 Electrocardiogram

Electrocardiogram (ECG) is a diagnostic signal which provides the graphical representation of the electrical activity of the heart muscle. ECG signals are obtained by connecting specially designed electrodes to the surface of the body which is then be used in identification of different heart diseases [1].

The morphology of electrocardiogram always reflects the functionality of human heart. Any disorder in the morphological pattern is an indication of cardiac arrhythmia which could be detected by analyzing the recorded ECG waveform. The amplitude and duration of the P-QRS-T wave in ECG signal contains useful information about the nature of disease afflicting the heart. The ECG signal provides the following information of a human heart [2]:

- heart position and its relative chamber size
- impulse origin and propagation
- heart rhythm and conduction disturbances
- extent and location of myocardial ischemia
- changes in electrolyte concentrations
- effects of drugs on the heart.

1.2 The Anatomy of Heart

The heart has four chambers. The upper chambers are called the left and right atria, and the lower chambers are called the left and right ventricles. Several atrioventricular and sinoatrial nodes are presented in the heart. Figure 1 shows an overview of the anatomy.

The atria and ventricles are separated by fibrous which is a non-conductive tissue that keeps them electrically isolated from each other. The circulation of blood to the lungs is performed by a pumping system containing the right atrium and the right ventricle . First, oxygen-poor blood is received through large veins called the superior and inferior vena cava and flows into the right atrium. Then it contracts and forces blood into the right ventricle. The ventricle is then stretched and started to maximize its pumping (contraction) efficiency. The blood is then pumped into the lungs when the right ventricle starts contraction. The blood is oxygenated in the lungs. Similarly, the left atrium and the left ventricle together form a pump to circulate oxygen-enriched blood received from the lungs (via the pulmonary veins) to the rest of the bod [3].

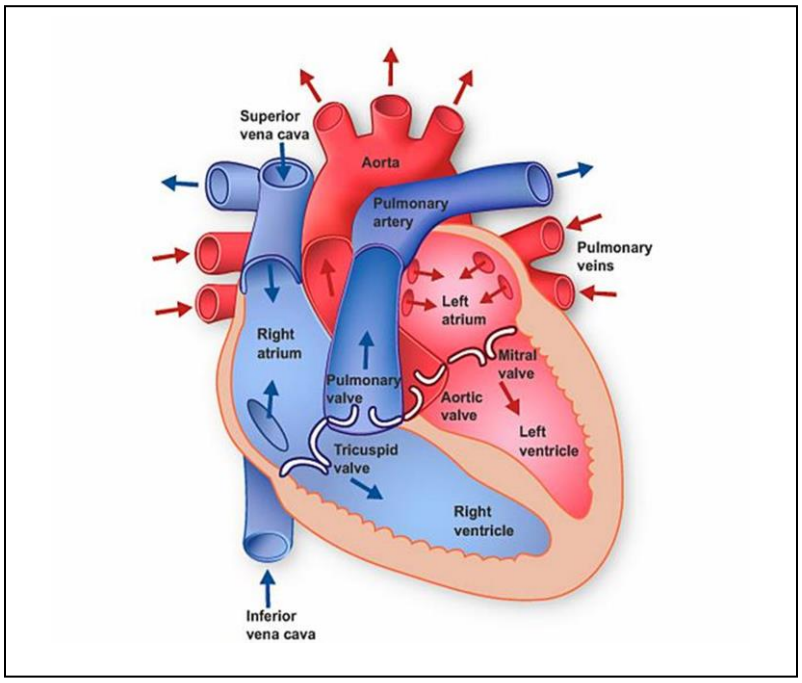

Figure 1. The Heart conduction system [3].

This whole conduction system begins with spontaneous depolarization of the sinus node situated in the high right atrium. A wave of electrical impulses then spreads through the right atrium and across the inter-atrial septum into the left atrium. This causes the muscle cell to depolarize and contract. This depolarization of the heart muscles collectively generates a strong ionic current. This current starts flowing through the resistive body tissues and causing a voltage drop. The magnitude of this voltage drop is large enough to be measured by electrodes attached to the surface of body. ECGs are thus recordings of voltage drops across the skin caused by ionic current flow generated from myocardial depolarization's [3]. Figure 2 shows the electrophysiological activity of the heart.

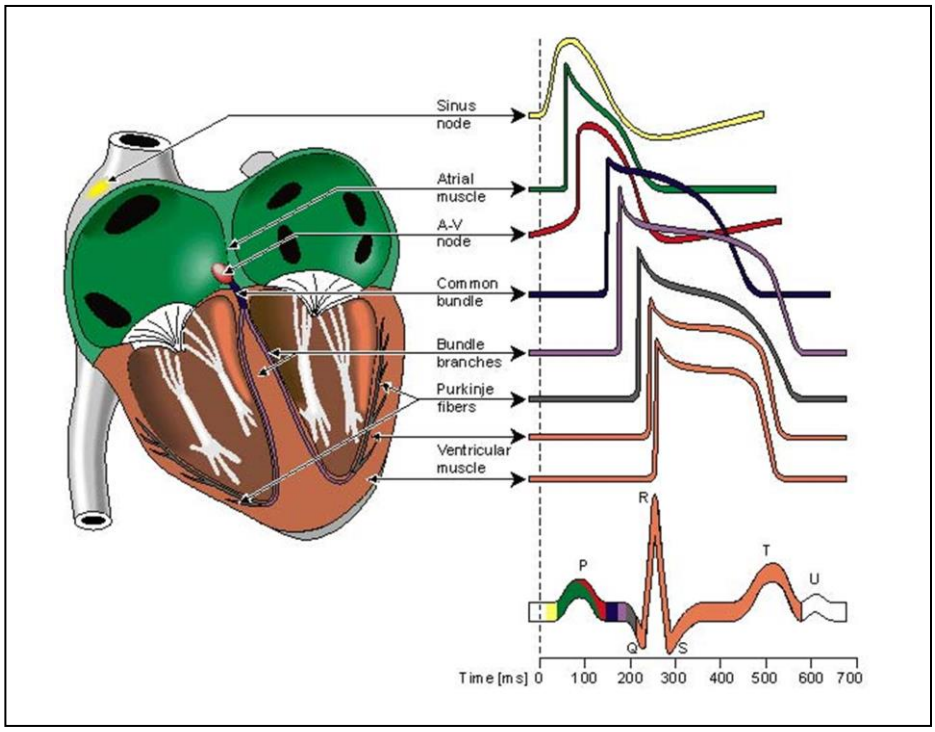

Figure 2. Basic Electrophysiology of the Heart [4].

The atrial depolarization appears as the P-wave in ECG signal. The following P-R segments depict the proceeding of electrical impulses into the ventricles. When signal leaves the atria, it enters the ventricles via AV node located at inter atrial septum. It then enters the bundle of His and spreads through the bundle brunches via Purkinji fibers along the ventricle walls. As the signal spreads through the ventricles the contractile fibers depolarize and start to contract very rapidly. This rapid depolarization of ventricles is the ECG's QRS complex. Atrial repolarization also occurs at this time. But any atrial activity is hidden on the ECG signal by the QRS complex. Finally as the signal passes out of the ventricles, the ventricular walls relax and recover. The T waves mark this ventricular repolarization. In ECG, ST segments depicts the period when ventricles are depolarized. These different waves and intervals are shown in Figure 3.

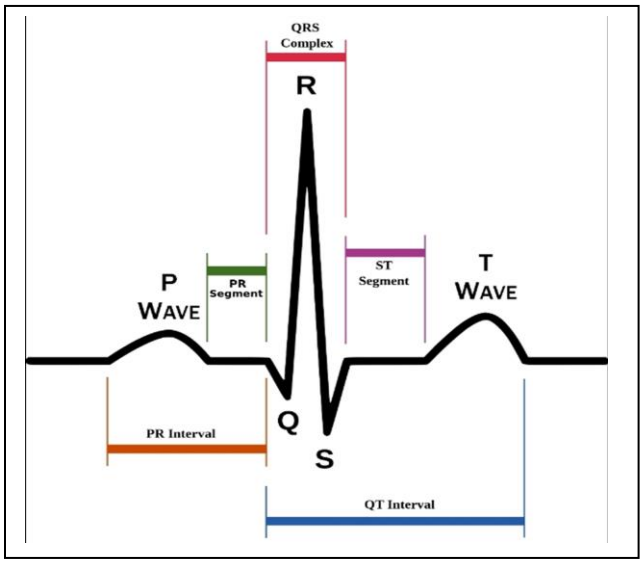

Figure 3. Different ECG Waves and Intervals [4].

1.3 ECG Measurement and Leads

ECG signals are typically measured using 12 leads system including 3 bipolar leads, 3 augmented unipolar leads and 3 chest (precordial) leads. A lead is a pair of electrodes (+ve and – ve) which is placed in some selected anatomical locations in our body and connected to a ECG recorder [2].

Bipolar leads are used to record the potential difference between two points (+ve & -ve poles) whereas the electrical potential at a particular point by means of a single exploring electrode is measured by unipolar leads.

Leads I, II and III are commonly referred to bipolar leads as they use only two electrodes to derive a view. One electrode acts as the positive electrode while the other as the negative electrode (hence bipolar) [1].

Table 1. Types of leads used in ECG monitoring

Standard Leads	**Limb Leads**	**Chest Leads**
Bipolar leads	*Unipolar leads*	*Unipolar leads*
		V1
Lead I	aVR	V2
Lead II	aVL	V3
Lead III	aVF	V4
		V5

Measuring ECG signal measurement using Einthoven leads is old and common. Einthoven's triangle is an imaginary triangle formed by the two shoulders and the pubis using three limb leads [1]. This forms an inverted equilateral triangle shape with the heart at the center that produces zero potential when the voltages are summed. Here, Lead I records potentials between the left and right arm, Lead II between the right arm and left leg and Lead III measures voltage between the left arm and left leg. Figure 4 shows the position of the leads:

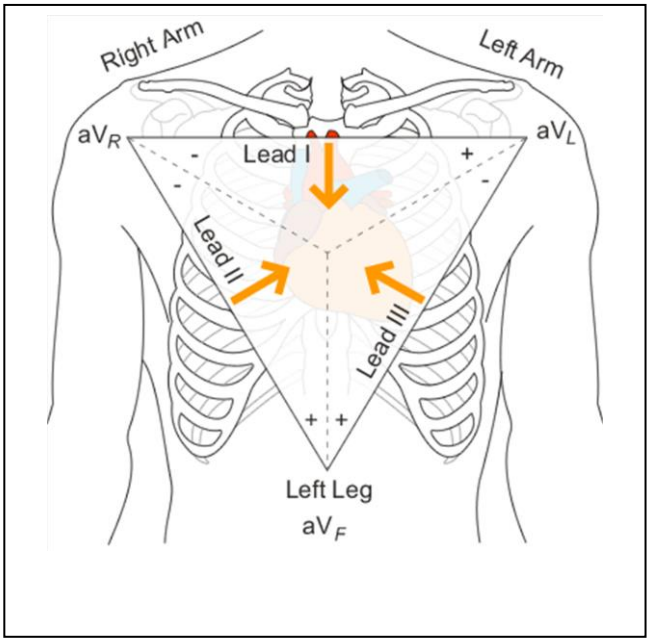

Figure 4. Einthoven's triangle - Line of site of the bipolar [1]

Goldberger leads are unipolar augmented limb leads in the frontal plane. The purpose of augmentation is to make the ECG waves more perceivable. The leads are annotated as aVR, aVL and aVF. Here, “a” stands for augmented, “V” for voltage, “R” for Right, “L” for left arm and “F” is for left foot. For measuring ECG, The lead connected to +ve terminal acts as the different electrode, while the other two limbs connected to the –ve terminal serve as the reference [3].

Wilson leads (V1–V6) are unipolar chest leads positioned on the left side of the thorax in a nearly horizontal plane. The indifferent electrode is obtained by connecting the 3 standard limb leads. When used in combination with the unipolar limb leads in the frontal plane, they provide a three dimensional view of the integral vector. This chest leads (precordial) are place as follows

V1: 4th intercostal space, right sternal edge.

V2: 4th intercostal space, left sternal edge.

V3: between the 2nd and 4th electrodes.

V4: 5th intercostal space in the midclavicular line.

V5: on 5th rib, anterior axillary line.

V6: in the midaxillary line.

To make recordings with the chest leads (different electrode), the three limb leads are connected to form an indifferent electrode with high resistances. The chest leads mainly detect potential vectors directed towards the back. These vectors are hardly detectable in the frontal plane [1]. Since the mean QRS vector is usually directed downwards and towards the left back region, the QRS vectors recorded by leads V1–V3 are usually negative, while those detected by V5 and V6 are positive [3]. In leads V1 and V2, QRS = -ve because, the chest electrode in these leads is nearer to the base of the heart, which is the direction of electronegativity during most of the ventricular depolarization process. In leads V4, V5, V6, QRS = +ve because the chest

electrode in these leads is nearer the heart apex, which is the direction of electropositivity during most of depolarization [2].

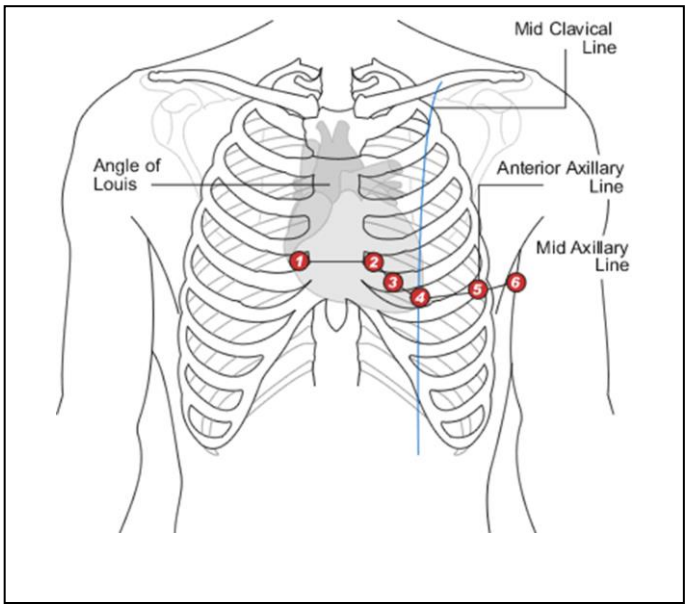

Figure 5. Precordial chest electrodes are normally placed on the left side of the chest [1].

1.4 ECG Waves and Intervals

Waves	**Representation and Significance**
P wave	As mentioned earlier, it represents the atrial depolarization. The amplitude is normally between 0.05 to 0.25 mV. A clear P wave proceeding the QRS complex represents sinus rhythm. Absence of P waves may suggest atrial fibrillation, Sinoatrial arrest, Hypokalemia or hyperkalemia. P waves are very hard to detect if the ECG signal contains a high signal-to-noise ratio.
QRS complex	The QRS complex is the most visually obvious graphical deflection part in ECG. It corresponds to the depolarization of the ventricles. The amplitude is the largest approximately 10– 20 mV but may vary in size depending on

age and gender. Cardiac diseases can be diagnosed from observing the amplitude, duration and activation time of QRS complex.

T wave — T wave corresponds to ventricular repolarization [2]. Large T waves may cause ischemia, and Hyperkalemia. Sometimes the large T waves are considered as noise when detecting QRS complex.

The Table 2 shows features of P-wave, QRS complex and T wave in maximum amplitude and its duration. According to medical definition [5], the duration of each RR-interval is about 0.4-1.2s.

Table 2. Amplitude and duration of waves, intervals and segments [5]–[7].

Features	**Amplitude (mV)**	**Duration (ms)**
P wave	0.1-0.2	60-80
PR-segment	-	50-120
PR- interval	-	120-200
QRS complex	<3	60-100
ST-segment	-	100-120
T –wave	0.1-0.3	120-160
ST-interval	-	320
RR-interval	-	(0.4-1.2)s

1.5 Artifacts in ECG

The ECG signal generally contains different types of noises and artifacts within its frequency band. Sources of noise include EMG (electromyogram) signal, power line interference, baseline wander, artifacts due to the motion of electrode and T waves containing high frequency components similar to QRS complexes. Hence, the characteristics of ECG signal

might be changed due to these noises making it difficult to extract useful information. The major noises that contaminate ECG signal are the following:

1.5.1 Power line interferences

The power line interference is caused by improper grounding. It appears as 60Hz (in U.S.)/50Hz (in Europe) spike in the frequency content of ECG signal. Additional spikes can also appear at harmonics. The amplitude can be up to 50 percent of peak-to-peak amplitude of ECG signal. By using a 60/50 Hz notch filter, this power line interference can be removed from the signal [5].

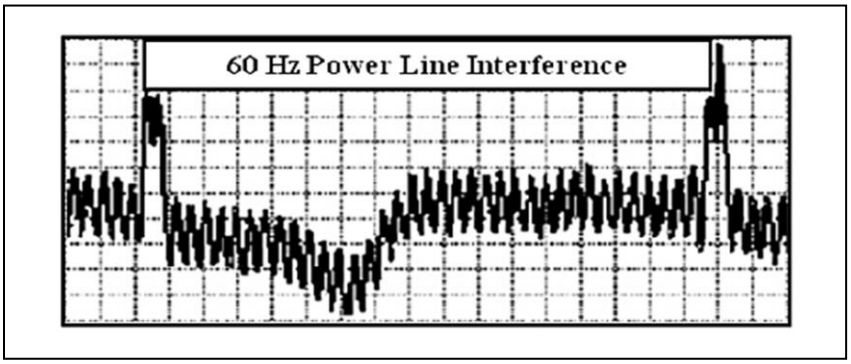

Figure 6. 60 Hz Power line interference [6].

1.5.2 Baseline drift

Baseline drift can be caused by patient's motion, deep breathing, coughing, loosely connected electrodes, metal dust on skin, dirty tips of the electrode cables, and voltage changes in the wall electricity. Variation in temperature or biased instrumentation can also be the source of this noise. The frequency range is generally bellow 0.5 Hz. A high pass filter with 0.5 cut-off frequency can be used to remove baseline drift [5].

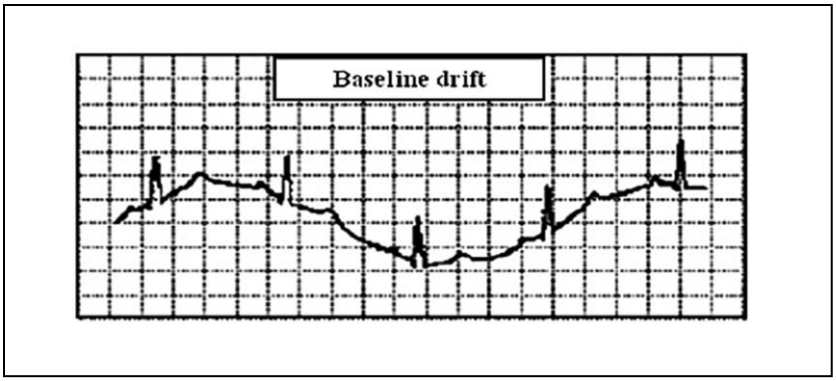

Figure 7. Baseline drifts in ECG signal [6].

1.5.3 Motion artifacts

The movement of the electrode gel under the electrode is major cause of motion artifacts. It significantly affect the transmission of signal from skin to electrode. Its peak amplitude can be 500 percent of peak-to-peak ECG amplitude. Normally, adaptive filters are designed to remove such noises [5].

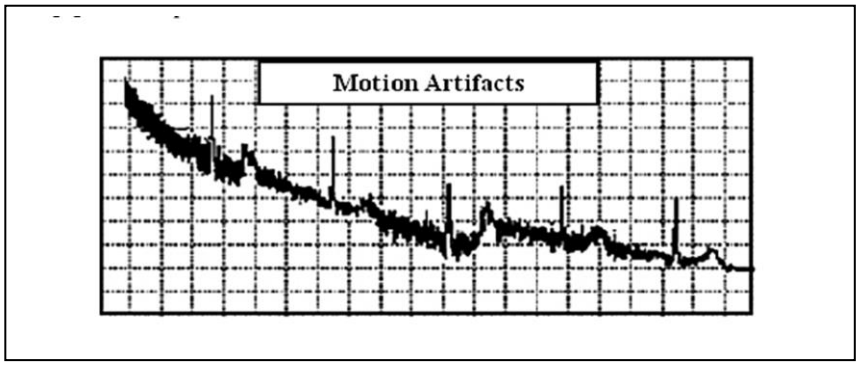

Figure 8. Motion Artifacts in ECG signal [6].

1.5.4 Muscle contraction (EMG)

Electromyogram (EMG) is the signal produced due to muscle activity. It is referred as a transient bursts of zero-mean band-limited Gaussian noise [5]. The presence of EMG signal

component causes a rapid fluctuation in ECG signal. Its frequency content is between 5 Hz to 450 Hz [3]. To remove the EMG interference, a unit square-wave structuring morphological filter can be applied.

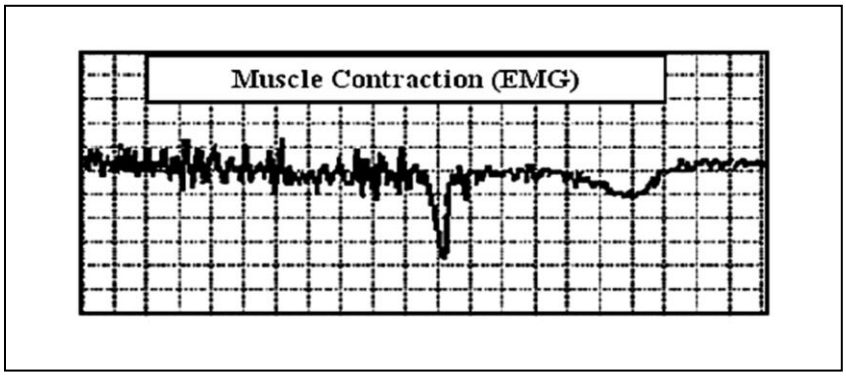

Figure 9. Muscle contraction [6].

1.6 Abnormalities in ECG Signal

In medical field, the term sinus rhythm means that the heart is beating with its normal rhythm as measured by the ECG. It indicates the fact that SA node is firing at a constant rate and pacing the heart. It has certain generic features that serve as hallmarks for comparison with normal ECGs. The normal sinus rhythm will usually fall between 60 and 100 beats per minute (bpm) [1]. The regularity of the R-R interval varies slightly with the breathing cycle.

When heart rate increase above its normal level, it is known as tachycardia. This is due to provide a higher blood circulation and is not an arrhythmia. However, when the heart is beating too fast, the ventricles don't get enough chance to be filled before contraction. The pumping efficiency can drop, adversely affecting perfusion. On the other hand, if the heart rate is too slow which is known as bradycardia, the effects can be detrimental for the vital organs.

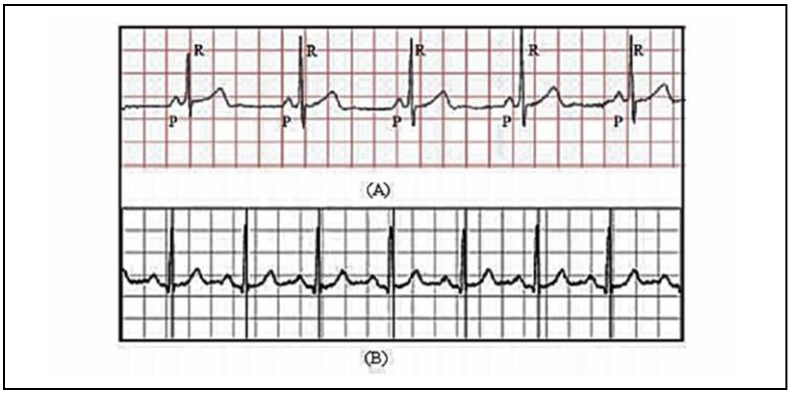

Figure 10. (A) Normal sinus rhythm, (B) Sinus tachycardia [1].

Atrial arrhythmias arise outside the S-A node. These arrhythmias types are described as follow.

1.6.1 Sinus Node Arrhythmias

This type of arrhythmia is caused by sinus node dysfunction. Symptoms may be minimal or include weakness, effort intolerance, palpitations, and syncope. These arrhythmias are the following types: Sinus arrhythmia, Sinus bradycardia, and Sinus arrest.

1.6.2 Atrial Arrhythmias

Atrial arrhythmias arise outside the S-A node. These following arrhythmias types exit:

Premature Atrial Contractions (PAC)

PACs occur when another region of the atria depolarizes before the sinoatrial node and thus triggers a premature heartbeat. It results an abnormal P wave followed by normal QRS complex and T wave. The exact cause is still unknown while several liable conditions exist. It may occur as a couplet where two PACs are generated consecutively. But three or more consecutive occurrences of PACs can lead to atrial tachycardia.

Atrial Tachycardia

Atrial tachycardia is a supraventricular tachycardia (SVT) that does not require AV junction or ventricular tissue for its initiation. The heart rate ranges from 160 to 240 beats per minute. It can occur to individual with normal heart as well as those with structurally abnormal hearts, including people with congenital heart disease (particularly after surgery for repair or correction of congenital or valvular heart disease).

Atrial Flutter

In atrial flutter, the abnormal P-waves occur so quickly that they take the shape of saw-tooth waveform which is called flutter (F) waves. The atrial rate ranges from 240 to 360 beats/minute.

Atrial Fibrillation

This arrhythmia originates from uncoordinated activation and contraction of different parts of the atria which lead to ineffective blood pumping to the ventricles. The atrial rate may exceeds 350 beats per minute in this type of arrhythmias [1].

Figure 11. Atrial arrhythmias, (A) Premature Atrial Contractions, (B) atrial tachycardia, (C) Atrial Flutter, (D) atrial fibrillation [1].

1.6.3 Junctional Arrhythmias

Junctional arrhythmias occur in the form of the impulses comprising the AV node and its Bundle. P wave abnormality such as reverse polarity to that of the normal polarity of sinus P-wave can occur since depolarisation is propagated in the opposite direction.

Premature Junctional Contractions (PJC)

In premature junctional escape contraction, a prematurely QRS complex appears without a preceding P-wave but with a normal T-wave [1]. It occurs when an ectopic pacemaker initiates a ventricular contraction in AV node.

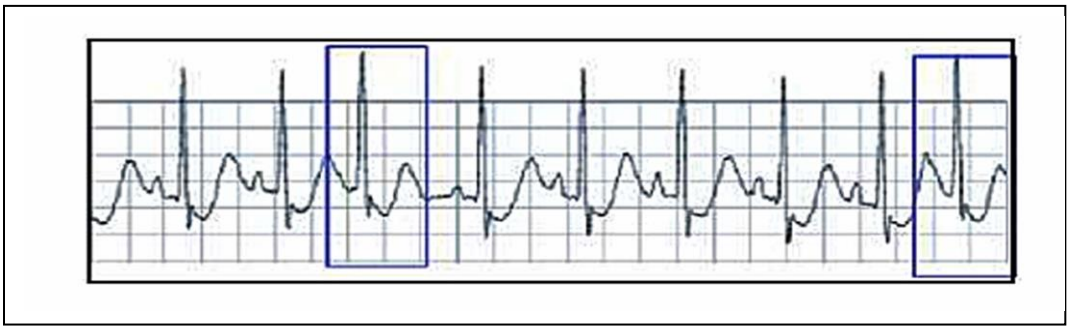

Figure 12. Junctional arrhythmias [1].

1.6.4 Ventricular Arrhythmias

In ventricular arrhythmias, wide and bizarre shape of QRS complex can be found. The spikes start off the ventricles and go through the outside to rest of the heart.

Premature Ventricular Contractions (PVC)

PVC occurs when the heartbeat is initiated not from the SA node but from Purkinje fibers in the ventricles. It may appear as a "skipped beat" or fill like palpitations in the chest. However, isolated PVC can be asymptomatic in healthy individuals and do not usually pose a danger [1].

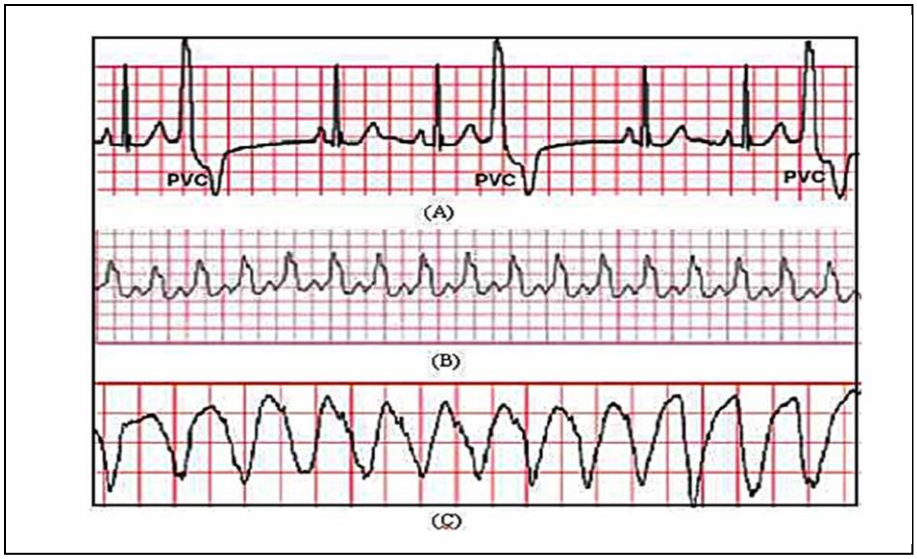

Figure 13. Junctional arrhythmias (A) Premature Ventricular Contractions, (B) Ventricular Tachycardia, (C) Ventricular Fibrillation [1].

Ventricular Tachycardia (VT)

The QRS complex is abnormally wide, out of the ordinary in shape and of a different direction from the normal QRS complex in ventricular tachycardia. The heart rate ranges from 110 to 250 beats per minute. It is life-threatening due to lack of effective ventricular blood filling which can result in a drop in cardiac output.

Ventricular Fibrillation

Similar to atrial fibrillation, non-synchronous fashion of heart beats appears as the firing of impulses begins with ectopic pacemakers not with SA node. Ventricular flutter exhibits a very rapid ventricular rate with a saw-tooth like ECG waveform.

1.6.5 Atrioventricular Blocks

In atrioventricular Blocks, the propagation of impulses in the heart is normal but it may delay or completely block the conduction to the rest of the body.

A first-degree AV block is occurred when all the P-waves are conducted to the ventricles, but the PR-interval is prolonged. Second-degree AV blocks are occurred when some of the P waves fail to conduct to the ventricles. In third-degree AV block, the rhythm of the P-waves is completely dissociated from the rhythm of the QRS-complexes. Each beat at their own rate [1].

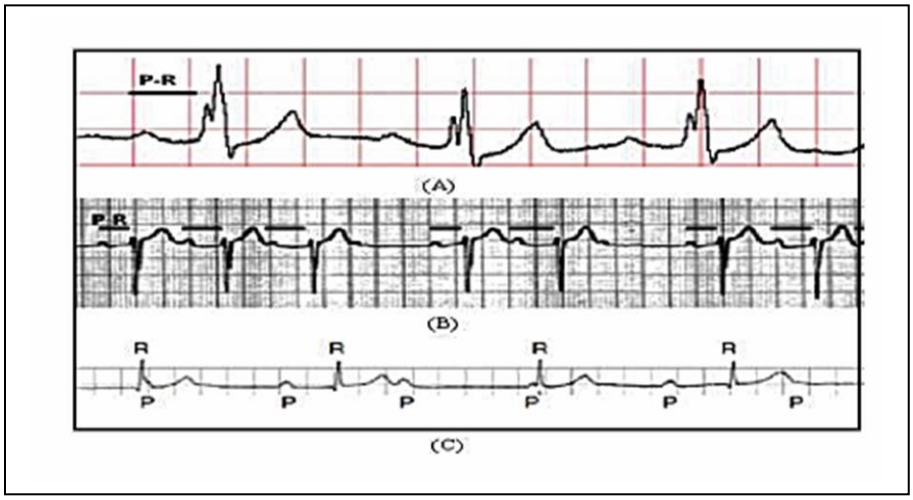

Figure 14. Atrioventricular Blocks (A) first degree AV block, (B) Second degree AV block, (C) Third degree AV blocks [1].

1.6.6 Bundle Branch blocks

Bundle branch block ceases the conduction of the impulse from the AV-node to the whole conduction system. Due to this block, myocardial infarction or cardiac surgery may occur[1].

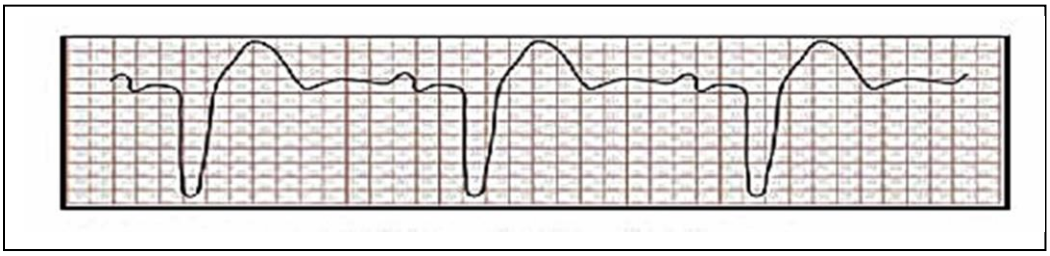

Figure 15. Bundle Branch blocks [1].

The bundle branch block are two types. These are left bundle branch block beat (LBBB) and right bundle branch block beat (RBBB). The electrical impulse from the AV node is not able propagate to the conduction network to depolarise the left ventricular when the left bundle branch is blocked. On the other hand in RBBB, prevention occurs for depolarising the right ventricular in the normal way.

1.7 Motivation

Since 1970, cardiovascular disease is the leading cause of deaths worldwide [8]. In United State, one-third of the total number of death is accounted for heart disease and stroke [9]. ECG, if properly analyzed, can provide information regarding various diseases related to heart. However, cardiac patients are needed to be monitored over an extensive period of time. Clinical observation of ECG can hence take long period of time and can be very tiresome. Moreover, the possibility of missing the crucial information by physicians is high and visual observation cannot be relied upon. Therefore, there is a growing interest in wearable technology for computer based analysis of ECG and classification of diseases. In addition, this interest originates in order to reduce the cost by providing a continuous monitoring system to cardiac patients during their daily life. Physicians can use the system to monitor several patients at a time and provide feedback on day-to-day basis, which helps to reduce the number of visits required as well. To obtain accurate response with less computational complexity as well as long battery life time in such system, developing fast and accurate QRS detection algorithms and prototypes is indispensable. Here, a method of QRS detection based on two slopes on both sides of an R peak is presented which is computationally efficient. Based on the slopes, first a variable measuring steepness is developed. Later, by introducing an adjustable R-R interval based window, and

adaptive thresholding techniques depending on the number of peaks detected in such window, R peaks are detected.

CHAPTER 2

QRS COMPLEX DETECTION BACKGROUND

Systems which are designed to perform signal processing tasks on ECG signal such as 12-lead offline ECG analysis, Holter tape analysis and real-time patient monitoring, have already proven that medical services can be achieved in an extremely efficient manner. But for all these applications at first, an accurate detection of QRS complex is crucial. The detection of QRS complex provides the fundamentals for almost all ECG analysis techniques due to the characteristic shape of the complex. Beginning about 40 years ago, many different approaches to detect QRS complex have been proposed in the literature and the research is still going on [10]. QRS detection is difficult not only because of the physiological variability of it but also because of the various types of noise mentioned earlier. However because of the rapid development of microcomputers, replacing more and more hardware based QRS detectors to software based QRS detections, it is possible to reduce the influence of such noise sources as well as improve the signal to noise ratio (SNR) without changing the morphology of ECG signal. Derivatives, digital filtering, wavelet transforms, neural networks, filter banks, hidden Markov models, genetic algorithms as well as other heuristic methods based software QRS detections are all developed to find the optimum automatic QRS complex detection algorithm [10].

2.1 Overview of Existing QRS Detection Methods

Already in the early years of developing QRS complex detection method, structure shown in Figure 16 is followed by many algorithms. This structure is divided into a preprocessing or feature extraction stage which includes linear and nonlinear filtering techniques for de-noising and pattern recognizing and a decision making stage based on peak recognition and determination logic. In order to determine the exact location of the assumed QRS complex, an extra block of processing is often introduced. In the literature, most of the approaches of detecting QRS complex can be discriminated based on their preprocessing stages. This is due to the fact that most of the time the decision making stages are rather heuristic and depends on the preprocessing results [10].

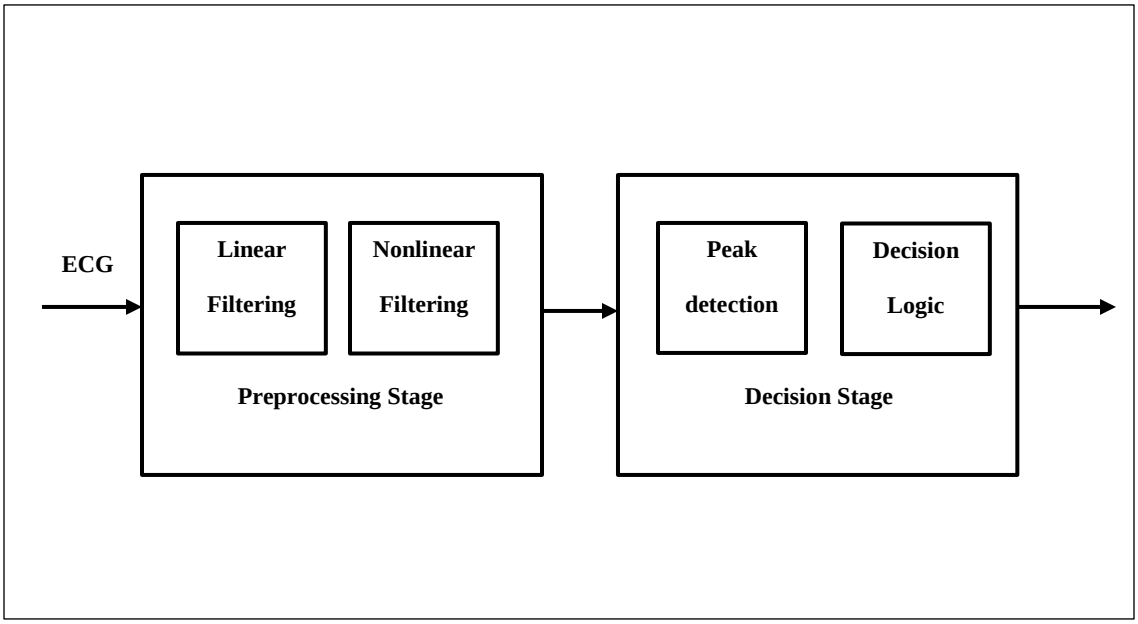

Figure 16. Common structure of QRS complex detection [10].

2.1.1 Derivatives and Digital Filter Based Approaches

The range of the frequency component of QRS complex is typically from 10 Hz to 25 Hz [10]. Therefore, a combination of high and low pass filters in other words a bandpass filter can be used to remove noises such as P-wave, T-wave, baseline drift and incoupling noises. Whereas a high pass filtering accomplishes the removing of P-wave, T-wave and baseline drift, a low pass filter is required to remove incoupling noise before the real detection begins. This filtered signal can then be used for generation of some features. QRS complex is detected by comparing these features with fixed or adaptive thresholds. This logic of detection is then frequently completed after applying some additional decision rules in order to reduce false positive detections.

Some algorithms, such as [11]–[16], use only high-pass filter part. This high-pass filter part is often, particularly in the older algorithms, realized as a differentiator which indicates the usage of the characteristic steep slope of the QRS complex for its detection. Some algorithm also computed second derivatives [11] [17] to extract the features.

Algorithms based on more sophisticated digital filters were published in [18]–[30]. In [30], two different low-pass filters with different cut-off frequencies are used in parallel along with a nonlinear operation on the signal. This causes a slight smoothing of the peaks. The threshold is computed adaptively.

In [22] and [26], a bandpass filter and afterwards differentiator are used as preprocessor. After squaring and averaging the output of the differentiator, features are computed. The bandpass and differentiator use filter coefficients that are particularly suited for an implementation on fixed-point processors with a short word length. For peak detection, a

variable is introduced along with an adaptive decision system. Figure 17 shows the fundamental of this kind of peak detection technique.

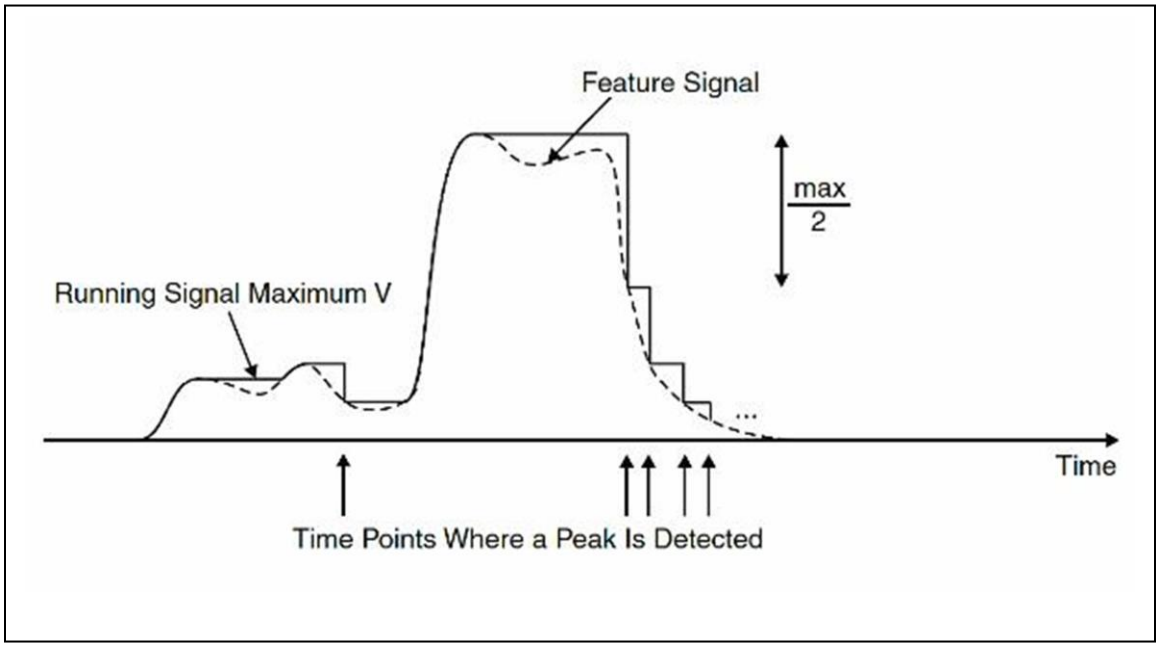

Figure 17. R peak detection proposed in [10].

One of the most popular QRS detection algorithms, introduced by Pan and Tompkins in [26], is based on this principle which is virtually included in all biomedical signal processing textbooks. Among all the techniques based on derivative and digital filtering, this method reliably recognizes QRS complexes. An overview of this algorithm is given below.

Pan and Tomkins Algorithm

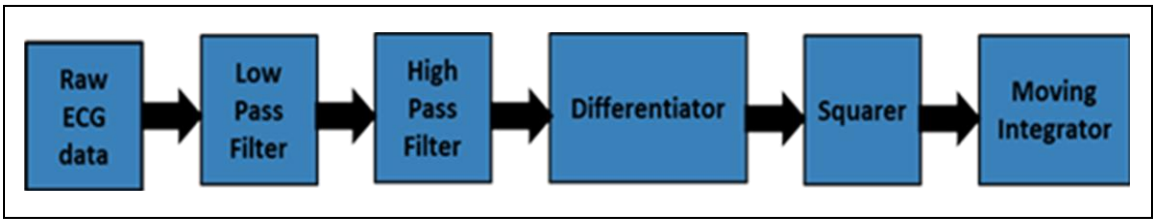

Figure 18. Graphical representation of QRS detection algorithm by Pan and Tomkins [26].

As shown in Figure 18, this algorithm uses basically the same preprocessor technique. At first, the ECG signal is passed through a low pass and a high pass filter. The low pass filter is given by the following equation:

$$y(n) = 2y(n-1) - y(n-2) + x(n) - 2x(n-6) + x(n-12) \qquad (1)$$

The high pass filter is described by

$$y(n) = y(n-1) - \frac{1}{32}x(n) + x(n-16) - x(n-17) + \frac{1}{32}x(n-32) \qquad (2)$$

After filtering, the signal is differentiated in order to get slope information by using the following equation

$$y(n) = \frac{1}{8}[2x(n) + x(n-1) - x(n-3) - 2x(n-4)] \qquad (3)$$

To make all the data points positive for emphasizing the higher frequencies, the signal is then squared point by point.

$$y(n) = x^2(n) \qquad (4)$$

After squaring, the algorithm performs moving window integration to obtain waveform feature information.

$$y(n) = \frac{1}{N}\left[x\left(n-(N-1)\right) + x\left(n\ -(N-2)\right) + \cdots + x(n)\right] \qquad (5)$$

where N is the size of the moving window and depends on the sampling rate. Figure 19 shows signals at various stages in processing.

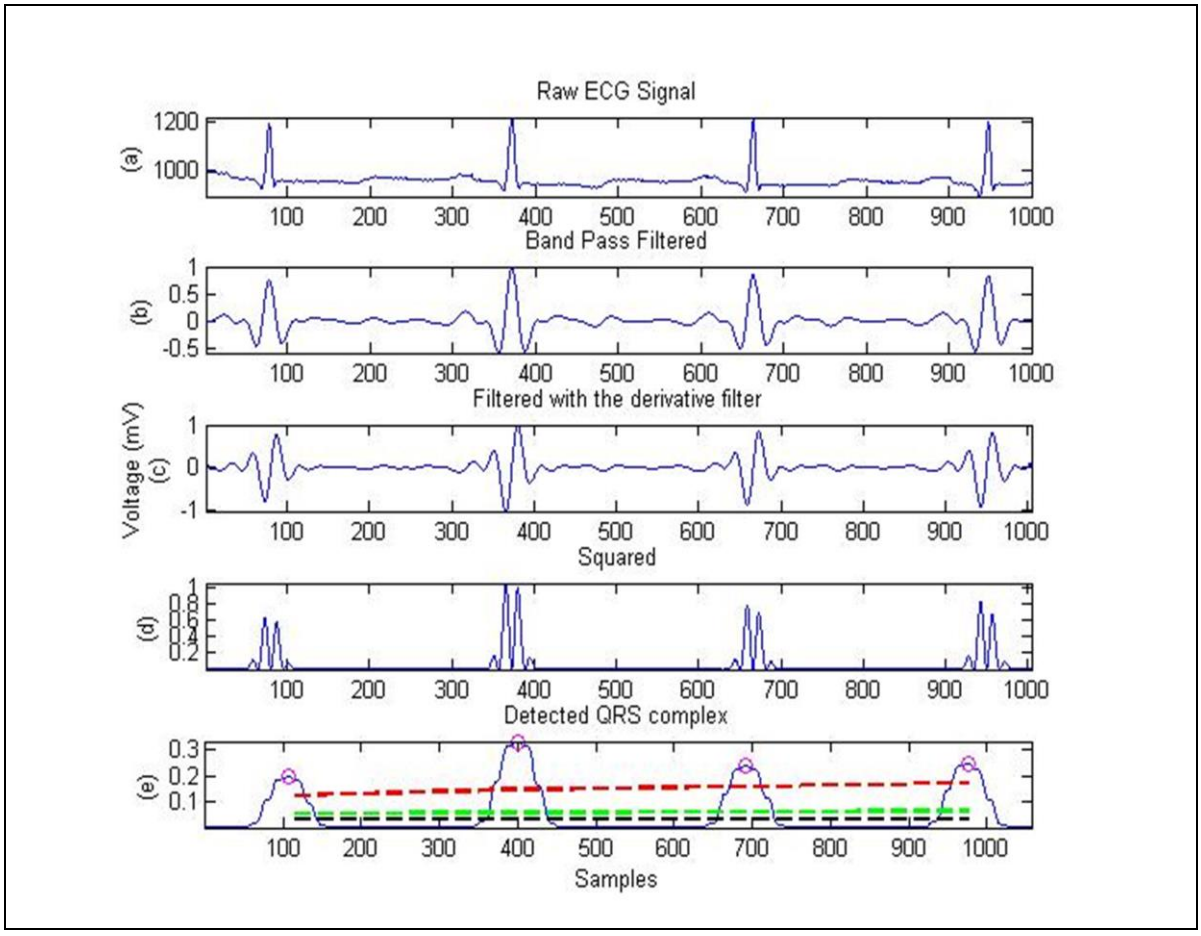

Figure 19. Different processing steps of Pan-Tompkins method. (a) Raw ECG signal (b) Output of bandpass filter (c) Output of differentiator (d) Output of Squaring process (e) Detected QRS (Averaged with 30 samples length, Black is noise level, Green is Adaptive Threshold, RED is Sig Level and the Red circles are QRS adaptive threshold).

A temporal location of the QRS is then marked from the rising edge of the integrated waveform. For peak detection, two thresholds are adjusted. Comparing with the higher of the two thresholds, R peaks are evaluated from the feature signal. If no peak is detected in a certain

time interval then the lower threshold is used for a search back technique where the algorithm has to search back in time for a lost peak. A peak is a local maxima determined by observing when the signal changes direction within a predefined time interval. When a new peak is identified, this peak is classified as a signal peak if it exceeds the high threshold or the low threshold if we search back in time. If it does not exceeds then it is classified as a noise peak. Based on these signal and noise peaks, the thresholds are automatically adjusted to float over the noise. The integration waveform and the filters signals are further investigated to get different values of thresholds at different point of the processing making it robust in terms of detection. To be identified as a QRS complex, a peak must be recognized as a QRS in both integration and filtered waveform.

In [31] using a similar way to [22] and [26] but different filters, the feature signal is achieved. This feature signal is then divided into segments of 15 points where the maximum of each segment is compared to an adaptive noise level. An adaptive peak level estimate classifies depending on the distance to each of the estimates.

Based on the assumption that a QRS complex is characterized by simultaneously occurring frequency components within the passbands of the two bandpass filters, [32] is proposed. Here by using a feature extractor based on two different bandpass digital filters and a multiplier performing AND operation, QRS detection is performed.

The algorithms described in [28] and [29] are based on the MOBD (multiplication of backward difference) technique. It is essentially an AND-combination of adjacent magnitude.

In [33], an algorithm is proposed based on the use of recursive and nonrecursive filters. Feature signal is acquired by using a combination of two median filters and one smoothing filter. The additional signal processing steps are similar to [22] and [26] .

Generalized digital filters which have a linear phase response are proposed in [18] and [27] in order to detect QRS complex with a transfer function. These are computationally highly efficient.

2.1.2 Wavelet Based Approaches

Like time-frequency representation of short-time Fourier transform (STFT), wavelet transform (WT) of a function gives a time-scale representation. In contrast to STFT, WT uses a set of analyzing functions derived from a mother wavelet to get details of time and frequency resolution for different frequency bands. This mother wavelet is a short oscillation with zero mean as depicted in Figure 20, as an example. Using the set of analyzing functions deduced from different mother wavelet function, different approaches are made to detect QRS complex from sampled ECG signal.

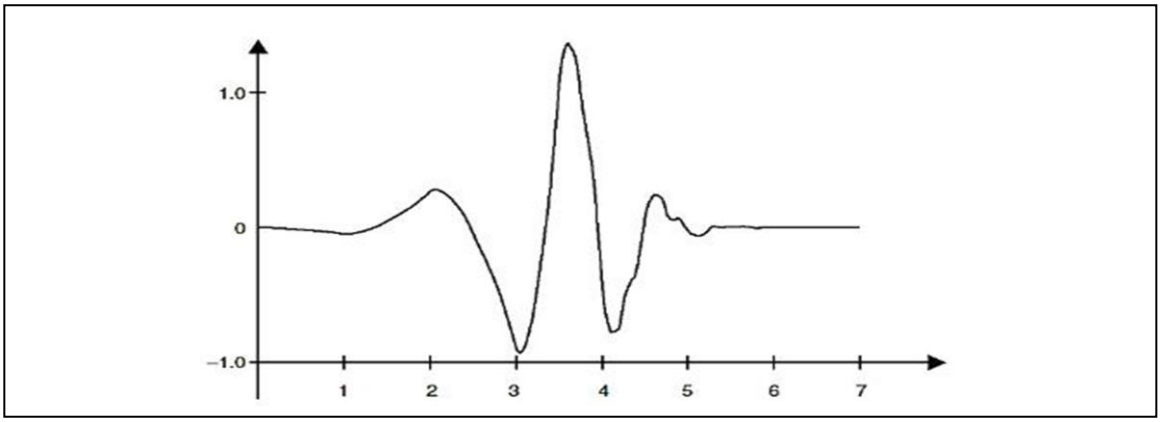

Figure 20. Example of a wavelet function (Daubechies-4 wavelet) [10].

Detection methods mentioned in [34]–[38] are based on Mallat's and Hwang's approach for singularity detection and classification using local maxima of the wavelet coefficient signals [39]. That means the analysis involves with the correlation between a function's local maxima in

its wavelet transform and its singularity. Figure 21 clarifies this relation. The peak classification is acquired by computing the singularity degree also known as the peakiness of the wave.

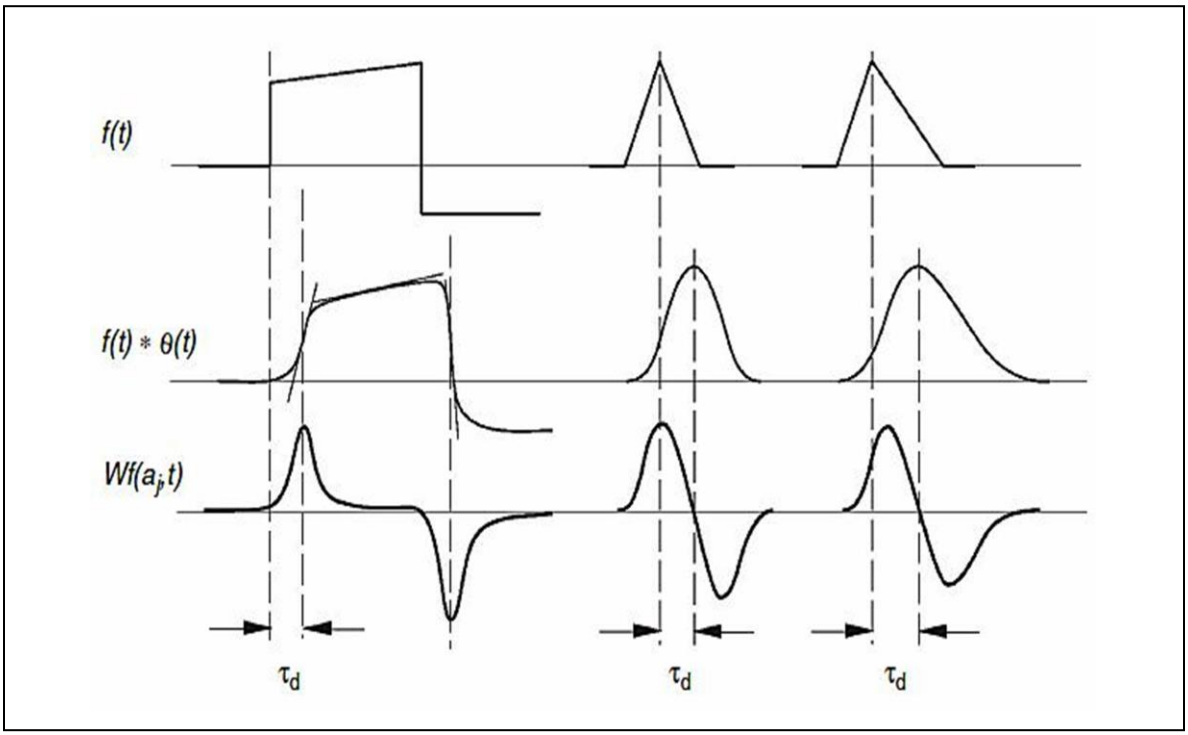

Figure 21. Example of the correlation between a function's local maxima in its wavelet transform *Wf (a, t)* and its singularity. The mother wavelet is the derivative of a smoothing function *θ(t)* [10].

In [37], by searching for modulus maxima in the relative scales of the WT, R-peaks are detected. Lipschitz regularity is checked for a valid detection. Beside this condition, further heuristic decision rules are applied such as within the different scales, putting condition on peak occurrence sign and timing. [34] is directly derived from [37]. Both algorithms are much more comprehensible than the original approach.

QRS detection algorithms based on local maxima and the coefficients of discrete WT are presented in [35] [36] and [38]. In [35], characteristic points are acquired by comparing the coefficients of the discrete WT on selected scales against fixed thresholds. [38] divides the ECG into segments of a fixed length. R-peaks are detected when the locations of modulus maxima of adjacent scales exceed a threshold that is calculated for every segment. In [40], the wavelet-based zero crossing representation from [41] is used for pattern recognition. The wavelet transform has also been used for classification in [42]–[44] with several other noise reduction techniques.

In [45], Afonso et al. reported a low computational QRS detection algorithm based on filter bank technique. Filter banks are closely related to wavelets and can be used for low computation compared to the wavelet based algorithms. The theory behind this algorithm is given below.

Filter Banks Technique by Afonso

The algorithm consists of two stages: a feature extraction stage and a sophisticated detection logic stage as shown in figure. 22.

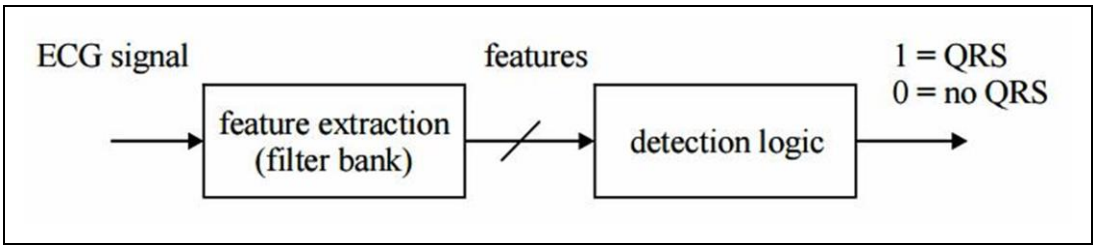

Figure 22. Signal flow diagram of the algorithm [45].

In the feature extraction stage, a 32 channel filter bank is used for extracting three features which is shown in figure 23. The outputs $w_i(n)$ are band pass filtered and downsampled versions of the input signal $x(n)$.

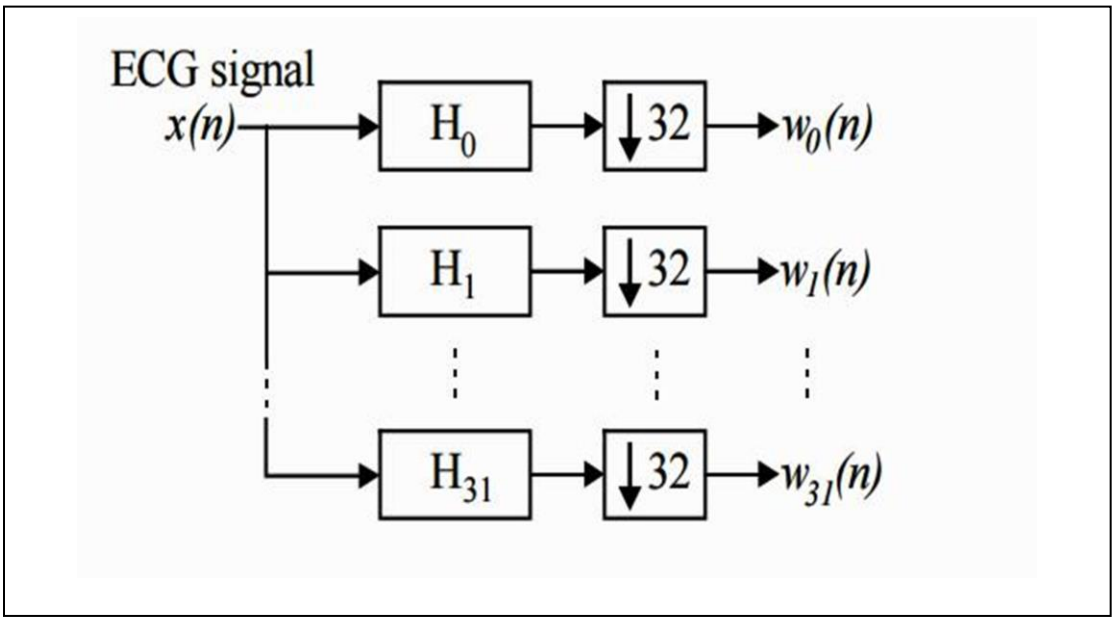

Figure 23. Filter bank for the generation of the feature signals [45].

The bandpass filters are designed according to a method proposed in [46]. For a lossy compression of the ECG, the filter bank is used to allow a perfect or near perfect reconstruction. Because such filter banks gives us a much higher stop band attenuation. In this algorithm, the order of the filters is N=320. This allows for a signal to noise ratio of about 60 dB after the reconstruction which is highly sufficient for ECG signals. If reconstruction is not desired then other filters that extract the same frequency information can be used with less constraints. Hence, they would need fewer filter coefficients.

From the output signals of the filter bank, three features are computed as follows

$$P_1(n) = \sum_{l=1}^{3} |w_l(n)| \qquad (6)$$

$$P_2(n) = \sum_{l=1}^{4} |w_l(n)| \qquad (7)$$

$$P_3(n) = \sum_{l=2}^{4} |w_l(n)| \qquad (8)$$

Afterwards the features P_i are smoothened by a moving window integrator. The downsampling and the usage of only four out of 32 channels for the computation of the features results in a significant reduction of samples to process and hence a significant reduction of the computation complexity. These features are then compared with a signal adaptive thresholds. A sophisticated detection logic processes the detected events, determines the probability for an event to be a QRS complex and eventually performs the classification. The exact timing information about the QRS complex is obtained by a maximum search in the ECG signal within a certain range around the detected event. For the determination of the timing information the electrical position of the heart must be taken into consideration.

2.1.3 Neural Network Based Approaches

Artificial neural networks have been widely applied in nonlinear signal processing, classification, and optimization. In many applications, their performance was shown to be superior to classical linear approaches. In ECG signal processing, mostly the multilayer perceptron (MLP), radial basis function (RBF) networks, and learning vector quantization (LVQ) networks are used [10].

As shown in Figure 24, multilayer perceptron (MLP) has several layers of interconnected neurons. Here, a neuron always represent a processing function.

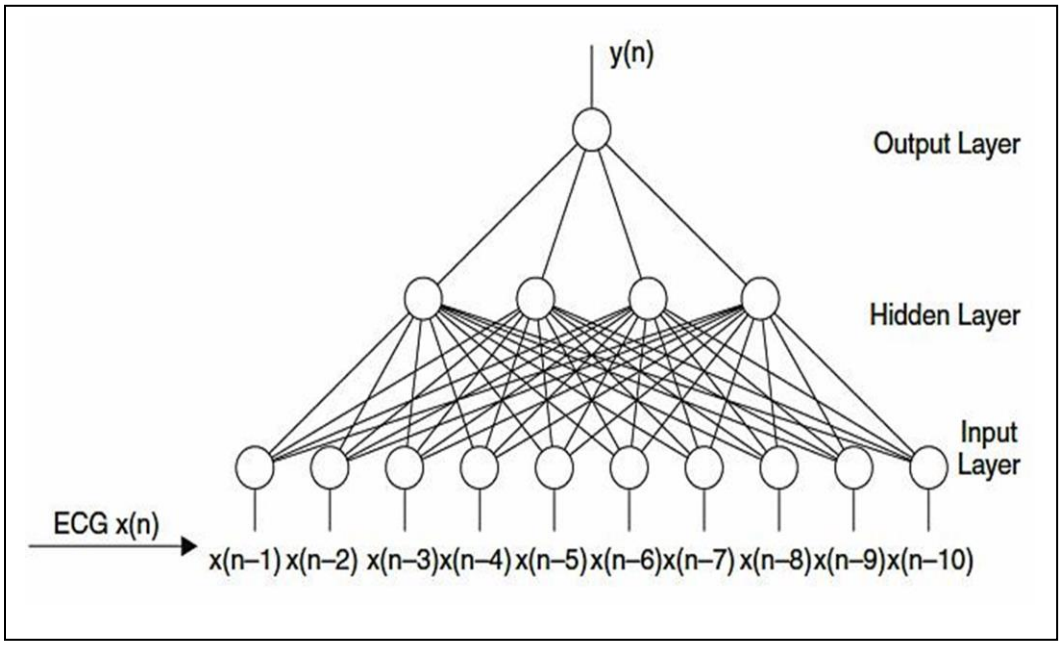

Figure 24. Multilayer perceptron [10].

Radial basis function (RBF) networks are closely related to fuzzy logic methods [47]. The possibility to interpret the parameters is a big advantage of radial basis function networks over multilayer perceptron networks. This gives us more reliable results.

The LVQ network consists of an input layer, a competitive layer, and a linear layer. The structure of the LVQ network is shown in Figure 25.

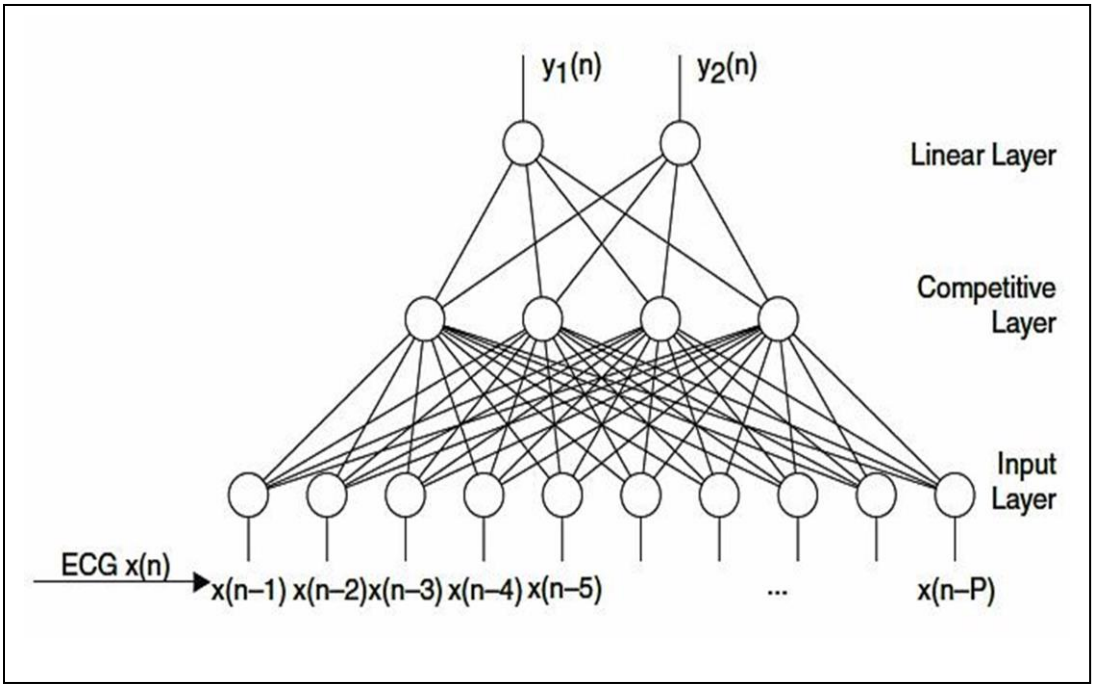

Figure 25. Example of a wavelet function (Daubechies-4 wavelet) [10].

In order to accomplish the application-dependent task (e.g. approximation or classification), the parameters of the network need to be trained. Whereas the MLP and RBF networks are trained by supervised learning algorithms, the LVQ network is adjusted in an unsupervised manner. Appropriate training algorithms are described in the literature; for example, in [48] and [49]. The application of neural networks in the field of ECG waveform classification is reported in [32], [50]–[60].

2.1.4 Additional Approaches

To detect QRS complex different other techniques have also been used. The application of adaptive prediction filters to QRS detection has been investigated in [22] and [61]. In [62] the application of hidden Markov models (HMM) to QRS and ECG waveform detection is investigated. The use of mathematical morphology operators for QRS detection was described in [63]. Besides the neural-network-based matched filtering approach in [64], there are linear matched filtering approaches as reported in [65]–[68]. In [69], genetic algorithm has been applied to a combined design of optimal polynomial filters for the preprocessing of the ECG and the parameters of a decision stage. In [70], the use of the Hilbert transform for QRS detection is proposed. In [71], the application of length and energy transforms to QRS detection is investigated. The transforms are defined for multichannel ECG signals but may also be used for single-channel ECG analysis. QRS detection based on zero crossing counts is proposed in [72]. Syntactic algorithms for ECG processing have been proposed in [73]–[76].

In [77], Wang et al. proposed a novel dual-slope QRS detection algorithm with low computational complexity. This method calculates the slopes on both sides of a peak in ECG signal. Based on these slopes, three criteria are developed for simultaneously checking steepness, shape and height of the signal in order to locate the QRS complex. Our proposed method is based on this discovery. To understand the concept of dual slope technique in detection of QRS complex, a short description of their algorithm is given in the next section.

Dual-Slope QRS Detection Algorithm by Wang *et al*.

Considering the fact that the half of the width of QRS complex is in the range of 0.3-0.5 sec, all slopes between the current sample and the samples 0.3-0.5 sec away on both sides are

calculated. The maximum and minimum slopes from each side are then evaluated with signs. Subtracting the maximum slope of one side and minimum slope of other side, a variable $S_{diff,max}$ is defined by taking the maximum value of slope difference. As the wearable devices are for heart patients, the range is extended to 0.027-0.063 sec to increase the sensitivity for abnormal heart beats.

The equation of calculating the maximum and minimum slopes on both sides and the variable measuring steepness $S_{diff,max}$ are given below:

$$S_{L,max} = \max_{a \le k \le b} \left(\frac{z^{-b} - z^{-(b-k)}}{k} \right), \quad (9)$$

$$S_{L,min} = \min_{a \le k \le b} \left(\frac{z^{-b} - z^{-(b-k)}}{k} \right), \quad (10)$$

$$S_{R,max} = \max_{a \le k \le b} \left(\frac{z^{-b} - z^{-(b+k)}}{-k} \right), \quad (11)$$

$$S_{R,min} = \min_{a \le k \le b} \left(\frac{z^{-b} - z^{-(b+k)}}{-k} \right), \quad (12)$$

$$S_{diff,max} = max \begin{pmatrix} (S_{R,max} - S_{L,min}) - \\ (S_{L,max} - S_{R,min}) \end{pmatrix}, \quad (13)$$

where f_s is the sampling frequency, a and b are the nearest integers of $0.027f_s$ and $0.063f_s$, respectively and z^n is the n^{th} sample in ECG signal.

An adaptive preset threshold is defined as Θ_{diff} which must be updated according to average value of $S_{diff,max}$ of previously detected 8 peaks. $S_{diff,max}$ must be larger than Θ_{diff} to satisfy the first criteria. The rules for updating are given below:

$$\Theta_{diff} = \frac{7680}{f_s} \quad \text{if } S_{ave} > \frac{20480}{f_s}$$

$$\Theta_{diff} = \frac{7680}{f_s} \quad \text{if } \frac{12800}{f_s} < S_{ave} < \frac{20480}{f_s} \quad (14)$$

$$\Theta_{diff} = \frac{7680}{f_s} \qquad \text{if } S_{ave} < \frac{12800}{f_s}$$

To avoid false high values of $S_{diff,max}$ causing from flat slope on one side and steep slope from other side, another criterion is introduced to check the shape. QRS complex should have a ramp like shape on both sides at an R peak. So the sign of the slope on both sides should be opposite and the value should be grater then a minimum value. The conditions are:

$S_{min} = min(|S_{L,max}|, |S_{R,min}|)$ and

$$sgn(S_{L,max}) = -sgn(S_{R,min})$$

$(if\ S_{L,max} - S_{R,min} > S_{R,max} - S_{L,min})$, (15)

$S_{min} = min(|S_{R,max}|, |S_{L,min}|) > \Theta_{min}$ and

$$sgn(S_{R,max}) = -sgn(S_{L,min})$$

$(if\ S_{R,max} - S_{L,min} > S_{L,max} - S_{R,min})$, (16)

where the value of Θ_{min} is 1536/ f_s.

The third criterion is based on the height of the slope. For noise and other sharp waves, we have high values of slope but the heights of the slopes are not as high as in QRS complex. So checking the height will eliminate the possibility of detecting such unwanted peaks. Therefore, the third condition is:

$$H_{cur} > H_{ave} \times 0.4 \qquad (17)$$

where H_{cur} is the height of current slope and H_{ave} is the average slope height of previous 8 detected peaks. If all the criteria are satisfied then we look forward to find the local extremes in that signal section followed by adjustment to avoid multiple detections within one section. The one with large value of $S_{diff,max}$ is considered as the R peak.

2.2 Available Benchmark Databases

There are several well-annotated standard ECG databases available in order to evaluate QRS detection algorithms. Evaluation of theses algorithms on such databases produce reproducible and comparable results. Furthermore, these databases contain large variations with rarely observed but clinically important signals. Available standard databases include:

2.2.1 MIT-BIH Database

The MIT/BIH arrhythmia database [78] is used in almost all study for performance evaluation of algorithms. It is a standard database of 48 ECG recordings. These recordings have 11 bit resolution over 10 mV and are sampled at 360Hz. Each tape includes two channel taken from modified limb lead II and one of the modified leads V1, V2, V4 or V5. The duration is 30 minutes for each recording selected from 24-hr recordings of 47 individuals. The subjects were taken from, 25 men aged 32 to 89 years, and 22 women aged 23 to 89 years and the records 201 and 202 came from the same male subject. Tape 100 to 124 (containing 23 recordings) are provided as a representative sample of routine clinical. On the other hands, Tape 200 to 234 (containing 25 recordings) are the signals containing different arrhythmias. The database contains annotation for both timing information and beat class information verified by independent experts [79].

2.2.2 LBNP Database

When a subject goes through a graded lower body negative pressure (LBNP), their heart rate is increased [80]. LBNP Database was created from such signals. Eighteen of the recordings were taken from 18 subjects (age: 27.6±3.6 years, weight: 71.22±11.9 kg and height: 174.3± 7 cm, 15 male and 3 female). Signals were recorded using an NI 9205 analog input module

(National Instruments, Austin, TX). The ECG's R wave was manually annotated for a total of 37000 cardiac cycles. Signals were recorded at Aerospace Physiology Laboratory at Simon Fraser University Under an Ethics Approval from the same institute. Written consent was given by all participants.

2.2.3 AHA Database

The AHA (American Heart Association) Database is a database containing 155 recordings for analyzing ventricular arrhythmia detectors [81]. The signals have a resolution of 12 bits over 20 mV and sampling frequency of 250 Hz. Each tape has 2.5 hours of unannotated signal followed by 30 minutes of annotated ECG. In each recording, eight groups are arranged to represent different levels of ectopic excitation. Tape 1001 to 1020 of the first group show no extra systoles, whereas records 8001 to 8010 consist of ventricular fibrillation.

2.2.4 AAMI Database

To emphasize on classifying ventricular ectopic beats from the non- ventricular ectopic beats, MIT-BIH heartbeat types are combined according to Association for the Advancement of Medical Instrumentation (AAMI) recommendation [82]. Each recording contains one ECG signal sampled at 720 Hz with 12-bit resolution. AAMI also recommends that each ECG beat can be classified into either Normal (N), supraventricular ectopic (SVEBs), ventricular ectopic beats (VEBs), fusion (F) or unclassified (Q) beats.

2.2.5 Other standard Database

More libraries available for evaluation of detection and classification algorithms are the Common Standards for Electrocardiography (CSE) Database [83], the IMPROVE Data Library

[84], European ST-T Database [85], the ECG Reference Data Set of the Physikalisch-Technische Bundesanstalt (PTB) [86], the QT Database [87] and the MGH Database [88].

2.3 Evaluation and Comparison of Detection Algorithms

To analyze the performance of detection algorithms, true positive (TP), false negative (FN) and false positive (FP) are normally evaluated. When an algorithm detects R peaks correctly a true positive happens. A false negative (FN) occurs when an algorithm fails to detect an actual QRS complex quoted in the corresponding annotation file of the record. And false positive (FP) means a false beat detection. In order to compare algorithms, error rate, sensitivity and positive predictivity should be used as benchmark parameters.

Using FP and FN, error rate (ER) is calculated based on the following equation:

$$ER(\%) = \frac{FP + FN}{Total\ QRS} \tag{18}$$

where total QRS is total number of QRS complex in the ECG data. The other two parameters, sensitivity and positive predictivity, are computed by

$$Se = \frac{TP}{TP + FN} \tag{19}$$

$$+P = \frac{TP}{TP + FP} \tag{20}$$

The percentage of true beats that were detected correctly is reported by sensitivity *Se*. The positive predictivity *+P* reports the percentage of beat detection which are true beats according to annotation.

CHAPTER 3

PROPOSED METHOD

Typically, the Q, R and S peaks are three deflections, which depict specific events and occur in a periodic manner in ECG signal. Starting from Q wave, a downward deflection, R wave follows with steepest upward deflection and S wave is any downward deflection after the R wave. The time taken by this event is relatively fixed in normal conditions, in the range of 0.06-0.1 second. If we calculate the slope of straight line between two samples, which is about the half of the width of QRS complex away from each other, the largest values of slope can be found in QRS complex. Moreover, QRS complexes are periodic, normally in the range of 0.6-1 seconds [77]. Therefore, by introducing a window of time which is automatically adjustable to ECG changes such as QRS morphology and heart rate, a faster QRS detection process is possible. Within this window, adaptive thersholding techniques can be used in order to consider a signal section as QRS complex. If thersholding criteria are satisfied then local extreme can be searched within QRS complex to locate R peak. Based on this idea, this new QRS detection algorithm is proposed.

3.1.1 Calculation of slopes

As half of QRS complex width is in the range of 0.03-0.05 second, the processing of sample begins by calculating one slope between the current sample and the sample 0.3 second away on both sides. From the range of 0.3-0.5 second, this distance between samples is carefully chosen

to 0.3 second over others in order to eliminate the possibility of low slope values caused by the different shapes of QRS complexes. The equations to calculate the slopes are:

$$S_{Left} = \left(\frac{ECG(i) - ECG(i-k)}{k}\right) \quad (21)$$

$$S_{Right} = \left(\frac{ECG(i) - ECG(i+k)}{-k}\right) \quad (22)$$

where $ECG(i)$ is the current sample. $ECG(i-k)$ and $ECG(i+k)$ are the samples which are $0.03f_s$ away from both sides of the current sample. f_s is the sampling frequency and k is equivalent to the nearest integer value of $0.03f_s$. By multiplying S_{Left} with S_{Right} for each sample, a steepness measuring variable S_{Mult} is evaluated as follows:

$$S_{Mult} = S_{Left} \times S_{Right} \quad (23)$$

As the sign of slope on both sides should be opposite at R peaks, the following condition is verified:

$$sgn(S_{Left}) = -sgn(S_{Right}) \quad (24)$$

If the sign of slopes on both sides is opposite, this steepness measuring variable S_{Mult} indicates very high values of this variable around QRS complex as shown in the Figure 26.

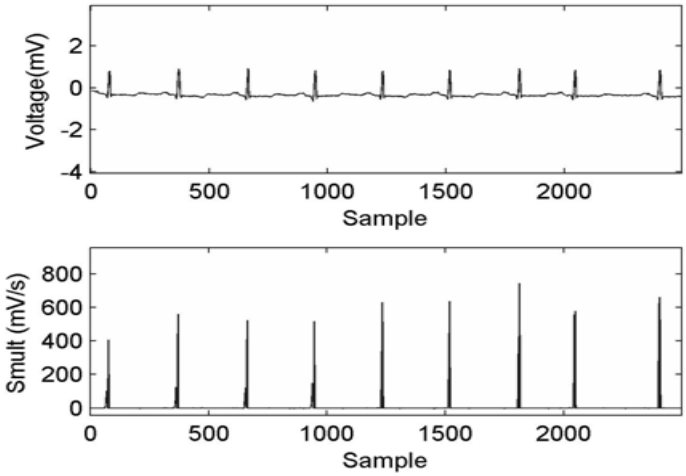

Figure 26. Tape 100 of MIT-BIH database showing values of variable S_{Mult}.

To eliminate false beat detection, the S_{Mult} must be larger than a threshold value. In order to define a universal adaptive thresholding technique, a learning stage based on first 3 second of data is introduced where at least 2 QRS complexes are required to occur to initialize parameters based on RR-interval. Therefore, to initialize detection threshold and RR-interval based parameters, at first S_{Mult} has been calculated for the first 3 seconds. Then, the threshold is considered as:

$$T = \frac{Max(S_{mult})}{3} \tag{25}$$

This means a signal is considered as R peaks if the value of S_{Mult} is greater than one-third of the maximum value of S_{Mult} for first 3 second. This value of T is selected by trial and error to get optimal result. In order to adapt with the ECG changes, for subsequent signal sections this threshold has been updated after every 8 detected peaks based on the average value of S_{Mult} of these peaks. The updating equation is:

$$T = \frac{avg(S_{Mult_R})}{3} \tag{26}$$

where S_{Mult_R} is the average value of S_{Mult} variable of last eight detected peaks.

3.1.2 R-R interval window

An R-R interval is the interval between two R peaks which is in the range of 0.6-1 second for normal resting adult human heart. But R peaks can occur faster than the average to respond to many conditions such as external stress. However, once an R peak is detected, there is a 200 millisecond refractory period before the next R peak since physiologically it cannot happen more closely than this refractory period [26]. For faster detection of QRS complex, a "moving window" of time can be used based on most recent RR interval values. To achieve reliable performance, this moving window of time is calculated as follows:

$$RR_{int} \pm \Delta R \quad (27)$$

where Where RR_{int} is the average interval between last 8 detected peaks and ΔR is considered as $0.48f_s$. This ΔR is carefully chosen based on the refractory time and average RR-interval of human heart to eliminate the chances of creating false negatives.

In this period of time, for normal resting heart only one peak should be detected. But as mentioned earlier, R peaks can occur more quickly than the average. At the same time, T wave, P wave or noise sections can produce large slopes as well. However, the height of this slopes are usually much smaller than those of QRS complex.

To avoid wrong detection of R peaks, if two peaks are detected within single period of window, then both of them are compared with the average height of previously detected eight peaks H_{ave}. If they are larger than a certain factor, then both of them are considered as R peaks. Otherwise, the peak with high value of S_{Mult} is considered as R peaks only. Following is the condition:

$$H_{cur} > H_{ave} \times 0.7 \quad (28)$$

Within one period of window more than two peaks cannot be possible as based on recent RR_{int} the moving window is adapting with time. So, if the detection of the peaks is more than two then the peaks with lower values of S_{Mult} among them are ignored.

3.1.3 Search back technique

Sometimes the amplitude of the ECG as well as the amplitude of R peaks gets significantly shorter when switching from one patient's ECG to another. So it might be possible that peaks cannot be detected in that region because of low value of the calculated slopes while the value of threshold based on previously detected peaks is high. Such a case is shown in Figure 27.

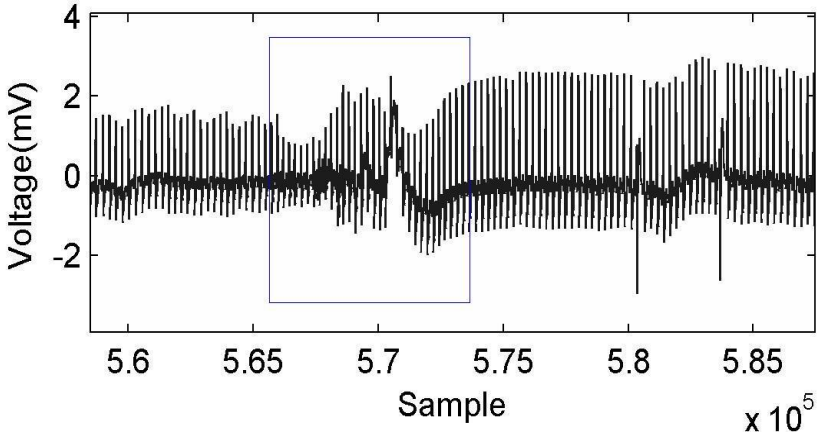

Figure 27. Tape 109 of MIT-BIH database showing variation of amplitude in ECG signal.

The algorithm needs to be rapidly adaptable without requiring any special learning phases in a situation like this. Thus for threshold re-adjustment, if in a period of time corresponding to average QRS complex interval elapses without a beat detection, the threshold is lowered by half of the previous threshold value. By searching back through the same time interval, the sensitivity of detection can be increased to avoid missing valid R peaks on that region.

3.1.4 Adjustment within QRS complex

If all the above criteria are met then local maximum and minimum are searched in current signal section to determine the location of R peaks. To avoid multiple detections within one QRS complex, the one with the largest value of S_{Mult} is retained as an R peak, while the other is discarded.

A flow chart of the proposed method is given bellow:

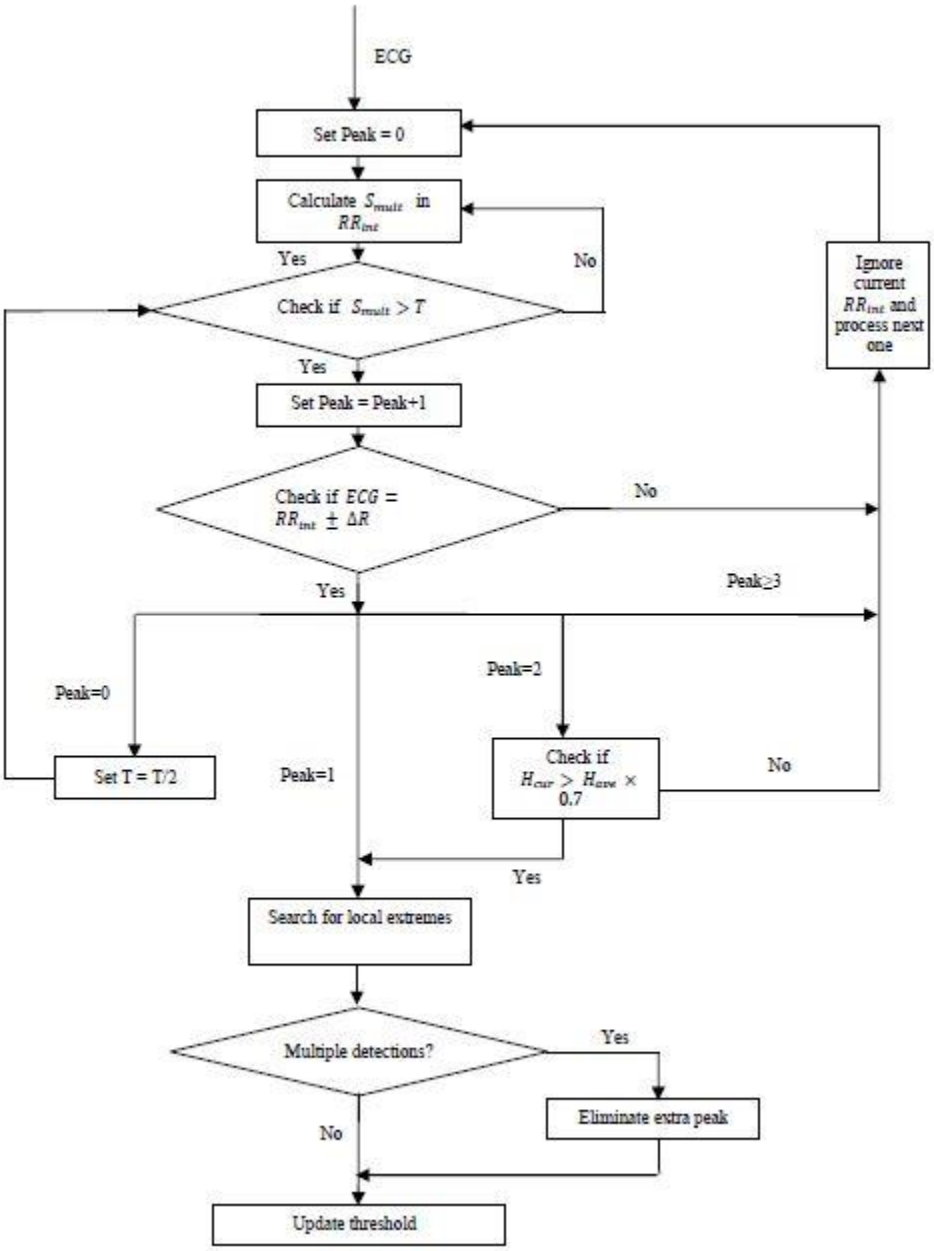

Figure 28. Block diagram of the proposed algorithm

CHAPTER 4

RESULTS AND DISCUSSION

Several QRS detection algorithms have been proposed during the last four decades. However, this large variety of QRS detection algorithms and continuous attempt for their improvement indicates that no single algorithm is clearly superior over all others. This is due to large diversity of QRS waveform and wide variation of noise and artifact in ECG signal [10]. Among all these algorithms, Pan-Tompkins's algorithm [26] is probably the most widely used QRS detection algorithm. On the other hand, the algorithm developed by Afonso *et al.* [45] has been reported to have lower computational load as well as good accuracy [8]. Due to the high acceptability of Pan-Tompkins method and low computational complexity of Afonso method, these two algorithms were used to be compared with the proposed approach.

For evaluation, two types of database were selected. As both of the algorithms, Pan-Tompkins and Afonso were evaluated on MIT-BIH database by the respective authors, MIT-BIH database was selected to analyze the performance based on different arrhythmias. Testing was performed on channel 1. The LBNP database was used to check the performance of the algorithms for healthy subjects under different stress.

4.1 Performance on MIT-BIH database

Since Pan-Tompkins and Afonso methods were not applied to the same data on the same computer, in order to get an impartial comparison of the accuracy and complexity, all three algorithms were evaluated in the same length of data using MATLAB environment on the same computer. The MATLAB code for Pan-Tompkins approach is provided by [89]. For Afonso technique, codes from [90] was used. Table 3 shows the accuracy of these codes according to their error rates mentioned by the authors based on MIT-BIH database. The accuracy of these m files (MATLAB files) is close enough to consider them for evaluation of these techniques.

Table 3. Accuracy of the M files

Method	**Total QRS**	**Paper**			**M file**		
		FP	***FN***	***Error rate (%)***	***FP***	***FN***	***Error rate (%)***
Pan and Tompkins [26]	109,508	507	277	0.72	441	757	1.09
Afonso [45]	90,782	406	374	0.86	1,189	613	1.98

Table 4 shows the summary of QRS detection for all algorithms based on FN and FP. T is the runtime of the corresponding program using the same MATLAB environment on the same computer for all methods. As mentioned in [10], Tape 108 is very noisy and has very low SNR. This proposed algorithm in this paper has 41 FPs and 41 FNs whereas the other algorithms have more than 100 false detections.

As the variable S_{Mult} has been calculated based on the width of QRS complex, this algorithm is sensitive to the width. In this database, some of the annotated R peaks because of premature

ventricular contraction (PVC) have more width than the usual range. Tape 208 has more than 992 beats with wider width. Because of this reason the sensitivity of the algorithm (detection of FN's) is greatly affected by PVC. Those PVCs have been eliminated before running the algorithm as those are not our main concern. The total number of deleted signal contains 1,041 R peaks and is 0.067% of the total data. Tape 208 was not considered because of its high PVC contains.

Table 4. Performance of the algorithms using MIT-BIH database.

Tape	**Total**	**Pan Tompkins** [26]			**Afonso** [45]			**New Dual-slope based algorithm**		
		FN	***FP***	***T (sec)***	***FN***	***FP***	***T(sec)***	***FN***	***FP***	***T(sec)***
100	2273	0	0	5.23	2	0	2.10	2	0	2.45
101	1865	2	3	3.72	1	2	1.67	1	2	2.13
102	2187	0	0	3.93	1	0	1.69	4	0	2.26
103	2084	0	0	3.41	2	0	1.68	2	0	2.33
104	2229	104	101	3.79	44	81	1.72	16	49	2.36
105	2572	17	43	4.22	11	40	1.74	5	23	2.66
106	1507	0	0	3.81	0	0	1.75	0	0	1.98
107	2137	18	2	3.54	20	58	1.73	16	1	2.43
108	1774	125	282	4.54	19	184	1.68	41	41	2.05
109	2532	5	0	2.81	97	3	1.72	15	0	2.71
111	2124	9	0	3.88	3	2	1.68	1	0	2.38
112	2539	0	0	3.52	2	1	1.72	1	0	2.66
113	1795	0	0	3.68	3	0	1.67	1	0	2.07
114	1879	2	7	4.30	2	4	1.69	23	6	2.03
115	1953	0	0	3.91	3	0	1.68	1	0	2.20
116	2412	24	4	3.41	24	2	1.72	27	0	2.46
117	1535	0	0	3.88	2	0	1.70	1	0	1.80
118	2278	0	1	3.82	2	4	1.69	1	0	2.50
119	1543	0	1	3.86	0	0	1.70	0	0	1.07
121	1863	2	0	3.91	2	3	1.68	3	0	2.19
122	2476	0	0	3.44	4	0	1.68	1	0	2.65
123	1518	3	0	3.92	2	0	1.63	4	0	1.80
124	1572	0	0	3.37	0	1	1.64	0	0	1.90
200	2601	4	1	4.05	8	14	1.74	9	7	2.23
201	1625	0	0	3.95	0	0	1.69	0	0	1.91
202	2136	8	1	3.63	10	0	1.69	46	0	2.12

Table 4. cont.

203	2980	4	19	3.98	60	51	1.80	212	34	2.41
205	2656	8	0	3.78	10	0	1.75	49	0	2.68
207	1477	0	4	3.77	3	16	1.70	8	5	2.87
209	3004	0	1	4.19	2	2	1.78	2	0	2.86
210	2423	0	5	3.74	0	9	1.75	0	5	2.52
212	2748	0	1	3.92	3	2	1.75	1	0	2.82
213	2641	0	0	4.17	0	0	1.90	0	0	3.03
214	2265	3	0	3.72	10	2	1.71	19	0	2.32
215	2533	0	0	5.12	0	0	1.82	0	0	2.87
217	2209	8	1	3.42	4	2	1.71	10	1	2.42
219	1553	0	0	3.48	0	0	1.73	0	0	2.32
220	2048	0	0	3.70	3	0	1.65	2	0	2.28
221	2031	0	0	3.85	0	0	1.75	0	0	2.18
222	2483	18	19	4.38	3	2	1.75	31	1	2.27
223	2605	10	0	3.75	9	0	1.74	59	0	2.60
228	2053	16	7	4.32	7	61	1.70	18	17	2.17
230	2256	0	0	4.30	3	2	1.68	2	0	2.45
231	1571	0	0	3.75	2	0	1.77	2	0	2.05
232	1780	1	2	4.66	1	9	1.76	2	1	2.03
233	3079	2	0	3.85	22	1	1.82	18	0	2.53
234	2753	1	0	4.00	6	0	1.75	5	0	2.84
Total	105112	394	505	3.90	412	558	1.73	661	193	2.34

Furthermore, Table 5 shows the overall error rate, sensitivity, positive predictivity, and processing time based on TP, FP, and FN. As shown in the table, because of the higher positive predictivity the overall error rate is higher. Due to wider width of PVCs the sensitivity of this method is lower than the others. On the other hand, the complexity has been analyzed based on time taken to run the code. The proposed method is faster than Pan Tompkins by 1.56 second and slower than Afonso's method by 0.61 second which can be considered small enough for wearable devices.

Table 5. Comparison of the algorithms based on MIT-BIH database.

Method	Total TP	FN	FP	Accuracy	Sensitivity	Positive predictivity	Processing Time
Afonso et al. [45]	101745	412	558	99.0505	0.9960	0.9945	1.73
Pan and Tompkins [26]	101763	394	505	99.1200	0.9961	0.9951	3.90
New Algorithm	101496	661	193	99.1640	0.9935	0.9981	2.34

4.2 Performance on LBNP database

The LBNP database contains noisy data as the subjects were under different stresses. Sometimes because of the movement of the subject or displacement of ECG electrodes, noises similar to what is shown in Figure 29 can appear. An algorithm needs to be robust enough to detect such cases. Other than this, different types of other noises can also appear when an individual is running or moving rapidly. To implement a QRS detection algorithm on wearable devices, signals from such cases are needed to be analysed. This database has been chosen to evaluate the algorithms on such ECG signals.

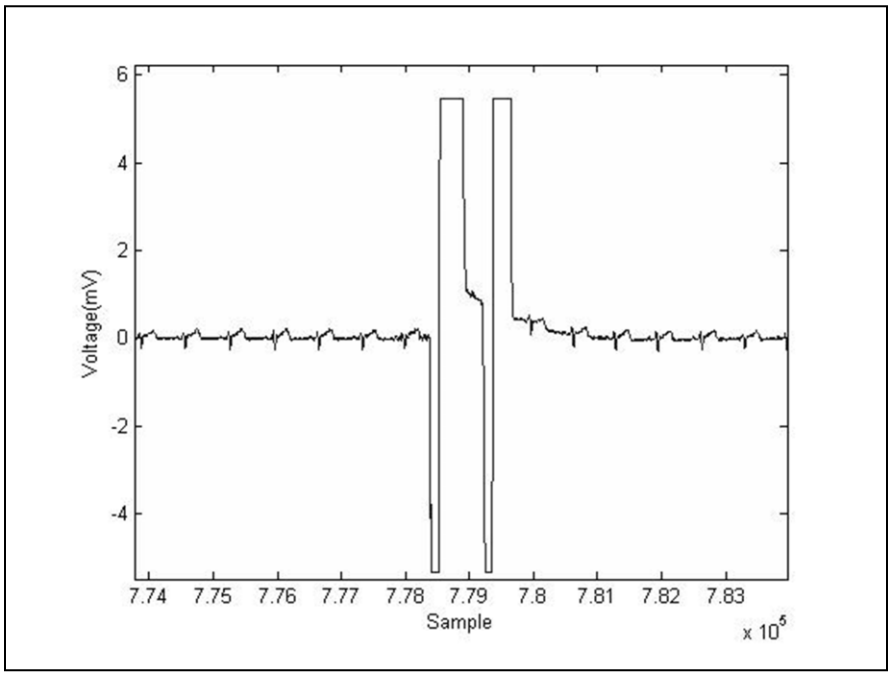

Figure 29. Tape 17 of LBNP database showing rapid movement noises.

Tables 6 and 7 show the performance of all algorithms on LBNP database in the same way. If noises appear, similar to what is shown in Figure 29, Pan-Tompkins method is very sensitive to them. For tape 05, 14 and 17, Pan-Tompkins method was not able to run properly because of getting very high derivative values and missing out several R peaks. This new dual slope based algorithm is also sensitive for such tapes as S_{mult} gets very high values because of large slopes on such signal sections. But by eliminating such value of S_{mult} from calculating the average S_{Mult_R}, this new algorithm can run smoothly for such cases. If the height of S_{mult} of current section is 3 times greater than the last average then to calculate the next average using the value of previous S_{Mult_R} instead of this new high S_{mult} this problem can be solved. On the other hand, having a frequency-based technique Afonso method does not need such condition. Results on Table 4 are shown after eliminating these signal sections as Pan-Tompkins method was not detecting R peaks for several tapes.

Table 6. Performance of the algorithms on LBNP database.

Tape	Total	**Pan Tompkins** [26]			**Afonso** [45]			**New Dual-slope based algorithm**		
		FP	***FN***	***T (sec)***	***FP***	***FN***	***T (sec)***	***FP***	***FN***	***T (sec)***
01	1822	2	0	7.97	9	0	7.34	1	0	5.46
02	1829	2	0	6.65	15	0	9.28	1	9	6.15
03	1665	3	2	6.13	40	5	9.27	13	0	5.98
04	2184	20	3	6.92	2	0	9.28	2	3	6.14
05	2303	0	0	9.38	6	1	14.31	3	25	6.40
06	2002	0	0	6.31	0	2	9.86	0	0	5.74
07	1071	0	1	6.99	2	4	10.15	0	0	6.55
08	2114	4	0	6.46	9	2	9.47	0	1	6.12
09	1772	0	0	5.83	0	1	8.38	1	2	5.46
10	2292	14	2	6.25	41	4	8.39	12	2	6.11
11	2813	0	79	6.57	0	0	9.69	0	0	6.39
12	1143	7	3	7.01	16	4	9.05	6	1	5.80
13	2390	0	0	6.45	0	2	9.33	0	0	6.14
14	2417	0	0	5.44	0	0	9.28	3	8	6.40
15	2067	0	0	6.32	1	3	9.74	0	4	5.74
16	2948	0	1	6.67	3	3	9.22	2	5	6.55
17	2370	0	0	5.91	9	1	11.05	13	0	8.67
18	2252	2	4	8.32	5	0	6.35	2	1	5.61
Total	37454	54	95	6.75	158	32	9.41	59	61	6.19

From Table 7, this new approach is showing highest accuracy among all. The sensitivity and positive predictivity are also very similar. But at a sampling rate of 1000, frequency based Afonso's algorithm does not have comparable processing time. More time is needed to process the data.

Table 7. Comparison of the algorithms based on LBNP database.

Method	*Total TP*	*Total FN*	*Total FP*	Accuracy	Sensitivity	Positive predictivity	*Processing Time*
Afonso et al. [26]	37422	32	158	99.4927	99.9146	99.5796	9.41
Pan and Tompkins [45]	37359	95	54	99.6022	99.7464	99.8557	6.75
New Algorithm	37393	61	59	99.6796	99.8371	99.8425	6.19

Figs. 30 and 31 show the robustness of this new algorithm against baseline drift and signals with large T waves respectively by demonstrating the detected R peaks marked as circle "o". It is clear that regardless of baseline drift or signals with large T waves the QRS complex can be accurately detected.

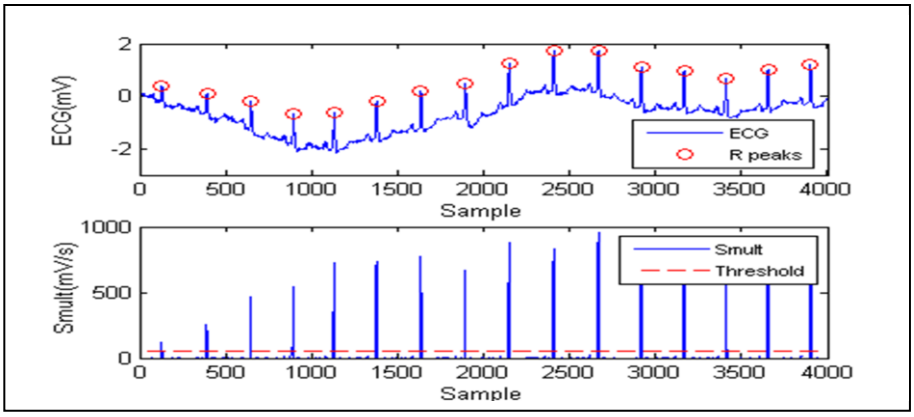

Figure 30. QRS detection over tape 105 of MIT-BIH database with baseline drifts.

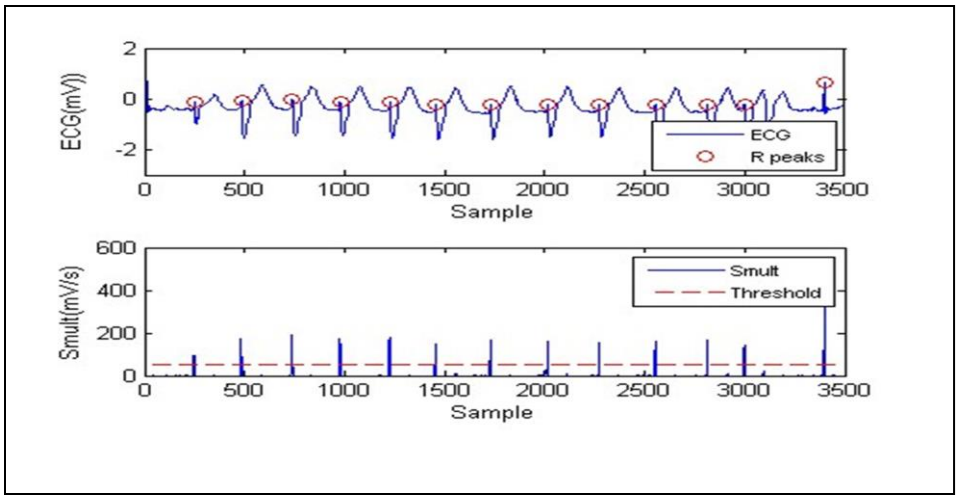

Figure 31. QRS detection over tape 205 of MIT-BIH database with large T waves.

4.3 Comparison with Wang et al method [77]:

As this new approach is based on the approach introduced by Wang et al., comparing this algorithm with their approach was essential and also our first to-do task. At the beginning of developing this algorithm, these two methods were compared with each other using MIT-BIH algorithm. The Wang et al. approach was implemented in MATLAB based on the technique mentioned in [77]. This comparison is shown in table 8.

Table 8. Comparison with Wang et al method [77].

Method	*Total FN*	*Total FP*	*Error rate*	*Processing Time*
Proposed Dual-Slope	206	409	0.62	1.66
Dual-Slope [77]	192	291	0.56	20.71

CHAPTER 5

CONCLUSION AND FUTUREWORK

5.1 Conclusion

In this paper, a new QRS detection algorithm was presented and its performance was compared with the performances of Pan-Tompkins's and Afonso's algorithms when applied to MIT-BIH and LBNP database in MATLAB. The number of annotated beats, used in evaluation of the algorithms, can be considered sufficient enough as these databases contain a large variation of the possible morphologies found in ECG signals. When applying to MIT-BIH database, the sensitivity of this new approach is 0.9935 but overall detection rate is 99.1640 which is almost same as the detection rate of Pan-Tomkins (99.1200) and Afonso's (99.0505) method because of its high positive preditivity (0.9981). On LBNP database, it also showed similar results. The detection rate is 99.6796 with the sensitivity as well as positive predictivity of 0.9984 where the detection rate of Pan-Tompkins is 99.6022 and Afonso is 99.4927. Overall, this new algorithm showed over 99% accuracy on both databases. This makes this new approach very useful in applications where the environment is rapidly changing and the distortions in ECG are different from the one in laboratory or ambulatory measured ECG.

Furthermore, the performance of Afonso and Pan Tompkins approach were not consistent on the databases during the evaluation. The Pan-Tomkins method was not detecting when the signal contains a transient rise of signal amplitude due to noise. Whereas Afonso's processing time was not consistent due to variation in sampling frequency of the signal. With less processing time and better stability, this new algorithm showed almost similar results. Hence, the method is highly suitable for battery-driven ECG signal devices.

5.2 Future work

The future work will aim at extensive testing of this new dual slope based algorithm. Different other databases and algorithms will be taken account for verifying its performance. EC13 and EMI databases can be used next because of their large variety of signal. And also new approaches will be introduced to analyze and detect premature ventricular contraction (PVC) beats in future. In this algorithm, as the calculations are based on only one slope on both side of each sample, this detection process is giving low value of S_{mult} when it comes to premature ventricular contraction (PVC) beats if they have large width. We are hoping that by taking another slope far away from the sample, we will be able to modify the S_{mult} variable to detect PVC beats. Also different thersholding values are taken based on trial and error method. These thresholding approaches can be investigated in order to check the possibility of reducing the error rate furthermore.

5.3 My Contribution

My contribution to this research was to develop an algorithm to detect QRS complexes in ECG signal as accurate as possible with low computational complexity. So that it can be implemented for low power operation on ECG signal. As the result of research conducted in this thesis, the following journal and conference papers were published:

1. **R. Arefin** and R. Fazel-Rezai, "Computationally Efficient QRS Detection Analysis based on Dual-Slope Method," in Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBS), Chicago, USA, 2014.

2. **R. Arefin,** K. Tavakolian and R. Fazel-Rezai, " QRS Complex Detection in ECG Signal for Wearable Devices," in Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBS), Milan, Italy, 2015. (Submitted)

3. M. R. Ravanfar, **R. Arefin**, K. Tavakolian, and R. Fazel-Rezai, "Electrocardiogram Artifact Cancelation based on Empirical Mode Decomposition and Peak Detection using Dual-Slope Algorithm," in Computing in Cardiology, Boston, USA, 2014.

4. **R. Arefin,** K. Tavakolian and R. Fazel-Rezai, " Computationally Efficient QRS Complex Detection in Electrocardiogram," in Physiological Measurement, Institute of Physics (IOP) Journal. (Submitted)

APPENDICES

APPENDIX A

Appendix A contains the MATLAB® of the proposed algorithm

```
function [mslope,RR,I,t] = new21(val,fs)


RR = 0; I = 0; p=0;

tic;

a = round(.027*fs); b=round(.063*fs);
k=a;

for (i=1+b:3*fs-b)

    slopeleft(i) = (val(i)-val(i-k))/k;
    sloperight(i) = (val(i)-val(i+k))/(-(k));

    if(sign(slopeleft(i))~= sign(sloperight(i)))
        mslope(i+1) = (-1*slopeleft(i)*sloperight(i));
    else
        mslope(i+1) = 0;
    end
    i=i+1;

end

maximum = max(mslope);
i=1;
Index_prev =0;j=1;l=1;

while (i<3*fs-b)
    if (mslope(i)> maximum/2)
        [Index]=extreme(val,i,a,b); % This index will give me perfect the
index in my data (here "val")
        if(Index > (Index_prev+round(0.2*fs)) && i>=Index) %Can't be the same
index of previous and the index needs to be i==index otherwise lot of points
around Index will be considered
            I(j)= Index;
            peakvalue(j)=val(I(j));          % If We need to check height of R
peaks
            peakmslope(j)= mslope(Index);    % For updating adaptive threshold
            if (j>=2)                        % Calculation of RR interval #for
RR we need at least 2 peaks
                RR(l) = (I(j) - I(j-1));
                l = l+1;
            end
            j=j+1;
            Index_prev=Index;
            mslope(i:Index+b) = 0;
            i=Index+b;
        end
```

```
    end
    i=i+1;
end

%End of 2 seconds period. Start to check whole data. Variables update 1st.

peakvalue_avg =(sum(peakvalue)/length(peakvalue));
peakmslope_avg = round(sum(peakmslope)/length(peakmslope));

RR_avg = round(sum(RR)/length(RR));
delta_RR = round(0.5*fs);

i = I(j-1); lm = I(j-1); n=1;m=1;

start = lm+RR_avg(end)-delta_RR;
finish = lm+RR_avg(end)+delta_RR;

%run the rest of the data

while (i<length(val)-b)

    if(finish+k>=length(val))
        break;
    end

    q=0;
    for(i = start:1:finish)

        slopeleft(i) = (val(i)-val(i-k))/k;
        sloperight(i) = (val(i)-val(i+k))/(-(k));
        if(sign(slopeleft(i))== 1 && sign(sloperight(i))== -1)
            mslope(i+1) = (-1*slopeleft(i)*sloperight(i));
        else
            mslope(i+1) = 0;
        end

        if (mslope(i)> maximum/3)
            [Index]=extreme(val,i,a,b); % This index will give me perfect the
index in my data (here "val")
            if(Index > (Index_prev+round(0.2*fs)) && i>=Index) %Can't be the
same index of previous and the index needs to be i==index otherwise lot of
points around Index will be considered
                q=q+1;
                p=1;

                if(q<2)
                    I(j)= Index;
                    peakvalue(j)=val(I(j));          % If We need to check
height of R peaks
                    peakmslope(j)= mslope(Index);    % For updating adaptive
threshold
                    if (j>=2)                        % Calculation of RR
interval #for RR we need at least 2 peaks
                        RR(l) = (I(j) - I(j-1));
```

```
                    l = l+1;
                end
                j=j+1;
                Index_prev=Index;
                mslope(i:Index+b) = 0;
                i=Index+b;

            elseif(q==2)

                if (length(RR)==8*m+4)                      % avg
                    peakvalue_avg(m)  = sum(peakvalue(j-8:j-1))/8;
                    peakmslope_avg(m)  = sum(peakmslope(j-8:j-1))/8;
                    maximum = peakmslope_avg(m);
                    RR_avg(m) = round(sum(RR(l-9:l-2))/8);
                    m=m+1;
                end

                peakvalue1=val(Index);
                if(peakvalue1>0.7*peakvalue_avg(end))
                    I(j)= Index;
                    peakvalue(j)=val(I(j));
                    peakmslope(j)= mslope(Index);   % For updating
adaptive threshold
                    if (j>=2)                       % Calculation of RR
interval #for RR we need at least 2 peaks
                        RR(l) = (I(j) - I(j-1));
                        l = l+1;
                    end
                    j=j+1;
                    Index_prev=Index;
                    mslope(i:Index+b) = 0;
                    i=Index+b;
                end

            else
                break;
            end

        end
    end
end

%If nothing is detected go back and change the threshold maximum/6

if(length(RR_avg)>1)
    if(p==0)
        [maximum1,Index1] = max(mslope(start:finish));
        Index = start+Index1;
        if (maximum1> maximum/6)
            if(Index > (Index_prev+round(0.2*fs))) %Can't be the same
index of previous and the index needs to be i==index otherwise lot of points
around Index will be considered
                I(j)= Index;
                p=1;
```

```
                    peakvalue(j)=val(I(j));          % If We need to check
height of R peaks
                    peakmslope(j)= mslope(Index);    % For updating adaptive
threshold
                    if (j>=2)                        % Calculation of RR
interval #for RR we need at least 2 peaks
                        RR(l) = (I(j) - I(j-1));
                        l = l+1;
                    end
                    j=j+1;
                    Index_prev=Index;
                    mslope(i:Index+b) = 0;
                    i=Index+b;
                end
            end
        end
    end

    % Leave that window if nothing is detected after maximum/6

    if(p~=1)
        lm = lm+RR_avg(end);
        start = lm+RR_avg(end)-delta_RR;
        finish = lm+RR_avg(end)+delta_RR;
    else
        lm = I(j-1);
        i = I(j-1);
        start = lm+RR_avg(end)-delta_RR;
        finish = lm+RR_avg(end)+delta_RR;
    end

    %Update RR_avg and peakmslope_avg

    if (length(RR)==8*m+4)                       % avg
        peakvalue_avg(m)  = sum(peakvalue(j-8:j-1))/8;
        peakmslope_avg(m)  = sum(peakmslope(j-8:j-1))/8;
        maximum = peakmslope_avg(m);
        RR_avg(m) = round(sum(RR(l-9:l-2))/8);
        m=m+1;
    end

    p=0;

end

t=toc;

% mslope1=(mslope(1:end)/max(mslope(1:end)));
% figure;plot(mslope1);
%
% figure;plot(mslope1/max(mslope1),'r');
% hold on
% x = 1:1:length(val);
% h2 = plot(x,2+val/max(val),x(I),2+val(I)/max(val),'bo');
```

```
% % saveas(h2,'our_result.fig')
%
% I3=diff(I);
% h1 = figure;plot(I3);
% % saveas(h1,'our_RR peaks.fig')


end
```

APPENDIX B

Appendix B contains the MATLAB® of Data extracting coding for MIT-BIH database.

```
function [I_new,Rpeaks_new] = detection(I,Rpeaks)

r=0;
l=1;
for p=1:length(I);
    for j=1:length(Rpeaks);
if  I(p)>=Rpeaks(j)-50 && I(p)<=Rpeaks(j)+50;
   I_new(l)=I(p);
   Rpeaks_new(l)=Rpeaks(j);
   l=l+1;
else
    r=r+1;
end
    end
end

end


function [Rpeaks] = convert(textdata)

j=1;

for i=2:length(textdata)
c = textdata{i,3};
ch =
['N';'L';'R';'B';'A';'a';'J';'S';'V';'r';'F';'e';'j';'n';'E';'/';'f';'Q';'?';
];
for k=1:length(ch)
if c == ch(k)
a = textdata{i,2};
b = str2num(a);
Rpeaks(j)= b;
j=j+1;
end
end
end
```

APPENDIX C

Appendix B contains the MATLAB® of the Pan-Tompins Algorithm

```
function [qrs_i_raw,t1]=pan_tompkin(ecg,fs,gr)

%% function [qrs_amp_raw,qrs_i_raw,delay]=pan_tompkin(ecg,fs)
% Complete implementation of Pan-Tompkins algorithm

%% Inputs
% ecg : raw ecg vector signal 1d signal
% fs : sampling frequency e.g. 200Hz, 400Hz and etc
% gr : flag to plot or not plot (set it 1 to have a plot or set it zero not
% to see any plots
%% Outputs
% qrs_amp_raw : amplitude of R waves amplitudes
% qrs_i_raw : index of R waves
% delay : number of samples which the signal is delayed due to the
% filtering
%% Method :

%% PreProcessing
% 1) Signal is preprocessed , if the sampling frequency is higher then it is
downsampled
% and if it is lower upsampled to make the sampling frequency 200 Hz
% with the same filtering setups introduced in Pan
% tompkins paper (a combination of low pass and high pass filter 5-15 Hz)
% to get rid of the baseline wander and muscle noise.

% 2) The filtered signal
% is derivated using a derivating filter to high light the QRS complex.

% 3) Signal is squared.4)Signal is averaged with a moving window to get rid
% of noise (0.150 seconds length).

% 5) depending on the sampling frequency of your signal the filtering
% options are changed to best match the characteristics of your ecg signal

% 6) Unlike the other implementations in this implementation the desicion
% rule of the Pan tompkins is implemented completely.

%% Decision Rule
% At this point in the algorithm, the preceding stages have produced a
roughly pulse-shaped
% waveform at the output of the MWI . The determination as to whether this
pulse
% corresponds to a QRS complex (as opposed to a high-sloped T-wave or a noise
artefact) is
% performed with an adaptive thresholding operation and other decision
% rules outlined below;

% a) FIDUCIAL MARK - The waveform is first processed to produce a set of
weighted unit
```

```
% samples at the location of the MWI maxima. This is done in order to
localize the QRS
% complex to a single instant of time. The w[k] weighting is the maxima
value.

% b) THRESHOLDING - When analyzing the amplitude of the MWI output, the
algorithm uses
% two threshold values (THR_SIG and THR_NOISE, appropriately initialized
during a brief
% 2 second training phase) that continuously adapt to changing ECG signal
quality. The
% first pass through y[n] uses these thresholds to classify the each non-zero
sample
% (CURRENTPEAK) as either signal or noise:
% If CURRENTPEAK > THR_SIG, that location is identified as a “QRS complex
% candidate” and the signal level (SIG_LEV) is updated:
% SIG _ LEV = 0.125 ×CURRENTPEAK + 0.875× SIG _ LEV

% If THR_NOISE < CURRENTPEAK < THR_SIG, then that location is identified as a
% “noise peak” and the noise level (NOISE_LEV) is updated:
% NOISE _ LEV = 0.125×CURRENTPEAK + 0.875× NOISE _ LEV
% Based on new estimates of the signal and noise levels (SIG_LEV and
NOISE_LEV,
% respectively) at that point in the ECG, the thresholds are adjusted as
follows:
% THR _ SIG = NOISE _ LEV + 0.25 × (SIG _ LEV ? NOISE _ LEV )
% THR _ NOISE = 0.5× (THR _ SIG)
% These adjustments lower the threshold gradually in signal segments that are
deemed to
% be of poorer quality.


% c) SEARCHBACK FOR MISSED QRS COMPLEXES - In the thresholding step above, if
% CURRENTPEAK < THR_SIG, the peak is deemed not to have resulted from a QRS
% complex. If however, an unreasonably long period has expired without an
abovethreshold
% peak, the algorithm will assume a QRS has been missed and perform a
% searchback. This limits the number of false negatives. The minimum time
used to trigger
% a searchback is 1.66 times the current R peak to R peak time period (called
the RR
% interval). This value has a physiological origin - the time value between
adjacent
% heartbeats cannot change more quickly than this. The missed QRS complex is
assumed
% to occur at the location of the highest peak in the interval that lies
between THR_SIG and
% THR_NOISE. In this algorithm, two average RR intervals are stored,the first
RR interval is
% calculated as an average of the last eight QRS locations in order to adapt
to changing heart
% rate and the second RR interval mean is the mean
% of the most regular RR intervals . The threshold is lowered if the heart
rate is not regular
% to improve detection.
```

```
% d) ELIMINATION OF MULTIPLE DETECTIONS WITHIN REFRACTORY PERIOD - It is
% impossible for a legitimate QRS complex to occur if it lies within 200ms
after a previously
% detected one. This constraint is a physiological one - due to the
refractory period during
% which ventricular depolarization cannot occur despite a stimulus[1]. As QRS
complex
% candidates are generated, the algorithm eliminates such physically
impossible events,
% thereby reducing false positives.

% e) T WAVE DISCRIMINATION - Finally, if a QRS candidate occurs after the
200ms
% refractory period but within 360ms of the previous QRS, the algorithm
determines
% whether this is a genuine QRS complex of the next heartbeat or an
abnormally prominent
% T wave. This decision is based on the mean slope of the waveform at that
position. A slope of
% less than one half that of the previous QRS complex is consistent with the
slower
% changing behaviour of a T wave - otherwise, it becomes a QRS detection.
% Extra concept : beside the points mentioned in the paper, this code also
% checks if the occured peak which is less than 360 msec latency has also a
% latency less than 0,5*mean_RR if yes this is counted as noise

% f) In the final stage , the output of R waves detected in smoothed signal
is analyzed and double
% checked with the help of the output of the bandpass signal to improve
% detection and find the original index of the real R waves on the raw ecg
% signal

%% References :

%[1]PAN.J, TOMPKINS. W.J,"A Real-Time QRS Detection Algorithm" IEEE
%TRANSACTIONS ON BIOMEDICAL ENGINEERING, VOL. BME-32, NO. 3, MARCH 1985.

%% Author : Hooman Sedghamiz
% Linkoping university
% email : hoose792@student.liu.se
% hooman.sedghamiz@medel.com

% Any direct or indirect use of this code should be referenced
% Copyright march 2014
%%
tic;

if ~isvector(ecg)
  error('ecg must be a row or column vector');
end


if nargin < 3
    gr = 1;   % on default the function always plots
end
```

```
ecg = ecg(:); % vectorize

%% Initialize
qrs_c =[]; %amplitude of R
qrs_i =[]; %index
SIG_LEV = 0;
nois_c =[];
nois_i =[];
delay = 0;
skip = 0; % becomes one when a T wave is detected
not_nois = 0; % it is not noise when not_nois = 1
selected_RR =[]; % Selected RR intervals
m_selected_RR = 0;
mean_RR = 0;
qrs_i_raw =[];
qrs_amp_raw=[];
ser_back = 0;
test_m = 0;
SIGL_buf = [];
NOISL_buf = [];
THRS_buf = [];
SIGL_buf1 = [];
NOISL_buf1 = [];
THRS_buf1 = [];

%% Upsample or downsample to 200 Hz default sampling rate of Pan-tompkins
algorithm
out =~rem(fs,200)*fs/200 ;
if out ~= 0
ecg = downsample(ecg,out);
fs = 200; %new fs
end

% ecg = resample(ecg,200,fs);
% fs = 200;

%% Plot differently based on filtering settings
if gr
 if fs == 200
  figure,  ax(1)=subplot(321);plot(ecg);axis tight;title('Raw ECG Signal');
 else
  figure,  ax(1)=subplot(3,2,[1 2]);plot(ecg);axis tight;title('Raw ECG
Signal');
 end
end
%% Noise cancelation(Filtering) % Filters (Filter in between 5-15 Hz)
if fs == 200
%% Low Pass Filter  H(z) = ((1 - z^(-6))^2)/(1 - z^(-1))^2
b = [1 0 0 0 0 0 -2 0 0 0 0 0 1];
a = [1 -2 1];
h_l = filter(b,a,[1 zeros(1,12)]);
ecg_l = conv (ecg ,h_l);
ecg_l = ecg_l/ max( abs(ecg_l));
delay = 6; %based on the paper
if gr
ax(2)=subplot(322);plot(ecg_l);axis tight;title('Low pass filtered');
```

```
end
%% High Pass filter H(z) = (-1+32z^(-16)+z^(-32))/(1+z^(-1))
b = [-1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 32 -32 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1];
a = [1 -1];
h_h = filter(b,a,[1 zeros(1,32)]);
ecg_h = conv (ecg_l ,h_h);
ecg_h = ecg_h/ max( abs(ecg_h));
delay = delay + 16; % 16 samples for highpass filtering
if gr
ax(3)=subplot(323);plot(ecg_h);axis tight;title('High Pass Filtered');
end
else
%% bandpass filter for Noise cancelation of other sampling
frequencies(Filtering)
f1=5; %cuttoff low frequency to get rid of baseline wander
f2=15; %cuttoff frequency to discard high frequency noise
Wn=[f1 f2]*2/fs; % cutt off based on fs
N = 3; % order of 3 less processing
[a,b] = butter(N,Wn); %bandpass filtering
ecg_h = filtfilt(a,b,ecg);
ecg_h = ecg_h/ max( abs(ecg_h));
if gr
ax(3)=subplot(323);plot(ecg_h);axis tight;title('Band Pass Filtered');
end
end
%% derivative filter H(z) = (1/8T)(-z^(-2) - 2z^(-1) + 2z + z^(2))
h_d = [-1 -2 0 2 1]*(1/8);%1/8*fs
ecg_d = conv (ecg_h ,h_d);
ecg_d = ecg_d/max(ecg_d);
delay = delay + 2; % delay of derivative filter 2 samples
if gr
ax(4)=subplot(324);plot(ecg_d);axis tight;title('Filtered with the derivative
filter');
end
%% Squaring nonlinearly enhance the dominant peaks
ecg_s = ecg_d.^2;
if gr
ax(5)=subplot(325);plot(ecg_s);axis tight;title('Squared');
end



%% Moving average Y(nt) = (1/N)[x(nT-(N - 1)T)+ x(nT - (N - 2)T)+...+x(nT)]
ecg_m = conv(ecg_s ,ones(1 ,round(0.150*fs))/round(0.150*fs));
delay = delay + 15;

if gr
ax(6)=subplot(326);plot(ecg_m);axis tight;title('Averaged with 30 samples
length,Black noise,Green Adaptive Threshold,RED Sig Level,Red circles QRS
adaptive threshold');
axis tight;
end

%% Fiducial Mark
% Note : a minimum distance of 40 samples is considered between each R wave
% since in physiological point of view no RR wave can occur in less than
```

```
% 200 msec distance
[pks,locs] = findpeaks(ecg_m,'MINPEAKDISTANCE',0.2*fs);



%% initialize the training phase (2 seconds of the signal) to determine the
THR_SIG and THR_NOISE
THR_SIG = max(ecg_m(1:2*fs))*1/3; % 0.25 of the max amplitude
THR_NOISE = mean(ecg_m(1:2*fs))*1/2; % 0.5 of the mean signal is considered
to be noise
SIG_LEV= THR_SIG;
NOISE_LEV = THR_NOISE;


%% Initialize bandpath filter threshold(2 seconds of the bandpass signal)
THR_SIG1 = max(ecg_h(1:2*fs))*1/3; % 0.25 of the max amplitude
THR_NOISE1 = mean(ecg_h(1:2*fs))*1/2; %
SIG_LEV1 = THR_SIG1; % Signal level in Bandpassed filter
NOISE_LEV1 = THR_NOISE1; % Noise level in Bandpassed filter
%% Thresholding and online desicion rule

for i = 1 : length(pks)

   %% locate the corresponding peak in the filtered signal
    if locs(i)-round(0.150*fs)>= 1 && locs(i)<= length(ecg_h)
          [y_i x_i] = max(ecg_h(locs(i)-round(0.150*fs):locs(i)));
       else
          if i == 1
            [y_i x_i] = max(ecg_h(1:locs(i)));
            ser_back = 1;
          elseif locs(i)>= length(ecg_h)
            [y_i x_i] = max(ecg_h(locs(i)-round(0.150*fs):end));
          end

     end


  %% update the heart_rate (Two heart rate means one the moste recent and the
other selected)
    if length(qrs_c) >= 9

        diffRR = diff(qrs_i(end-8:end)); %calculate RR interval
        mean_RR = mean(diffRR); % calculate the mean of 8 previous R waves
interval
        comp =qrs_i(end)-qrs_i(end-1); %latest RR
        if comp <= 0.92*mean_RR || comp >= 1.16*mean_RR
            % lower down thresholds to detect better in MVI
                THR_SIG = 0.5*(THR_SIG);
                %THR_NOISE = 0.5*(THR_SIG);
               % lower down thresholds to detect better in Bandpass filtered
                THR_SIG1 = 0.5*(THR_SIG1);
                %THR_NOISE1 = 0.5*(THR_SIG1);
```

```
        else
            m_selected_RR = mean_RR; %the latest regular beats mean
        end

    end

      %% calculate the mean of the last 8 R waves to make sure that QRS is
not
       % missing(If no R detected , trigger a search back) 1.66*mean

       if m_selected_RR
           test_m = m_selected_RR; %if the regular RR availabe use it
       elseif mean_RR && m_selected_RR == 0
           test_m = mean_RR;
       else
           test_m = 0;
       end

    if test_m
          if (locs(i) - qrs_i(end)) >= round(1.66*test_m)% it shows a QRS is
missed
              [pks_temp,locs_temp] = max(ecg_m(qrs_i(end)+
round(0.200*fs):locs(i)-round(0.200*fs))); % search back and locate the max
in this interval
              locs_temp = qrs_i(end)+ round(0.200*fs) + locs_temp -1;
%location

              if pks_temp > THR_NOISE
               qrs_c = [qrs_c pks_temp];
               qrs_i = [qrs_i locs_temp];

               % find the location in filtered sig
               if locs_temp <= length(ecg_h)
                [y_i_t x_i_t] = max(ecg_h(locs_temp-
round(0.150*fs):locs_temp));
               else
                [y_i_t x_i_t] = max(ecg_h(locs_temp-round(0.150*fs):end));
               end
               % take care of bandpass signal threshold
               if y_i_t > THR_NOISE1

                      qrs_i_raw = [qrs_i_raw locs_temp-round(0.150*fs)+
(x_i_t - 1)];% save index of bandpass
                      qrs_amp_raw =[qrs_amp_raw y_i_t]; %save amplitude of
bandpass
                      SIG_LEV1 = 0.25*y_i_t + 0.75*SIG_LEV1; %when found with
the second thres
               end

               not_nois = 1;
               SIG_LEV = 0.25*pks_temp + 0.75*SIG_LEV ;  %when found with the
second threshold
            end

          else
```

```
            not_nois = 0;

        end
    end



    %%  find noise and QRS peaks
    if pks(i) >= THR_SIG

                % if a QRS candidate occurs within 360ms of the previous QRS
                % ,the algorithm determines if its T wave or QRS
                if length(qrs_c) >= 3
                     if (locs(i)-qrs_i(end)) <= round(0.3600*fs)
                       Slope1 = mean(diff(ecg_m(locs(i)-
round(0.075*fs):locs(i)))); %mean slope of the waveform at that position
                       Slope2 = mean(diff(ecg_m(qrs_i(end)-
round(0.075*fs):qrs_i(end)))); %mean slope of previous R wave
                           if abs(Slope1) <= abs(0.5*(Slope2)) || (locs(i)-
qrs_i(end)) <= round(0.4*test_m)  % slope less then 0.5 of previous R
                               nois_c = [nois_c pks(i)];
                               nois_i = [nois_i locs(i)];
                               skip = 1; % T wave identification
                               % adjust noise level in both filtered and
                               % MVI
                               NOISE_LEV1 = 0.125*y_i + 0.875*NOISE_LEV1;
                               NOISE_LEV = 0.125*pks(i) + 0.875*NOISE_LEV;
                           else
                               skip = 0;
                           end

                     end
                end

        if skip == 0  % skip is 1 when a T wave is detected
        qrs_c = [qrs_c pks(i)];
        qrs_i = [qrs_i locs(i)];

        % bandpass filter check threshold
         if y_i >= THR_SIG1
                       if ser_back
                          qrs_i_raw = [qrs_i_raw x_i];  % save index of
bandpass
                       else
                          qrs_i_raw = [qrs_i_raw locs(i)-round(0.150*fs)+
(x_i - 1)];% save index of bandpass
                       end
                          qrs_amp_raw =[qrs_amp_raw y_i];% save amplitude of
bandpass
          SIG_LEV1 = 0.125*y_i + 0.875*SIG_LEV1;% adjust threshold for
bandpass filtered sig
         end

        % adjust Signal level
```

```
        SIG_LEV = 0.125*pks(i) + 0.875*SIG_LEV ;
        end


    elseif THR_NOISE <= pks(i) && pks(i)<THR_SIG

         %adjust Noise level in filtered sig
         NOISE_LEV1 = 0.125*y_i + 0.875*NOISE_LEV1;
         %adjust Noise level in MVI
         NOISE_LEV = 0.125*pks(i) + 0.875*NOISE_LEV;



    elseif pks(i) < THR_NOISE
        nois_c = [nois_c pks(i)];
        nois_i = [nois_i locs(i)];

        % noise level in filtered signal
        NOISE_LEV1 = 0.125*y_i + 0.875*NOISE_LEV1;
        %end

         %adjust Noise level in MVI
        NOISE_LEV = 0.125*pks(i) + 0.875*NOISE_LEV;


    end






    %% adjust the threshold with SNR
    if NOISE_LEV ~= 0 || SIG_LEV ~= 0
        THR_SIG = NOISE_LEV + 0.25*(abs(SIG_LEV - NOISE_LEV));
        THR_NOISE = 0.5*(THR_SIG);
    end

    % adjust the threshold with SNR for bandpassed signal
    if NOISE_LEV1 ~= 0 || SIG_LEV1 ~= 0
        THR_SIG1 = NOISE_LEV1 + 0.25*(abs(SIG_LEV1 - NOISE_LEV1));
        THR_NOISE1 = 0.5*(THR_SIG1);
    end


% take a track of thresholds of smoothed signal
SIGL_buf = [SIGL_buf SIG_LEV];
NOISL_buf = [NOISL_buf NOISE_LEV];
THRS_buf = [THRS_buf THR_SIG];

% take a track of thresholds of filtered signal
SIGL_buf1 = [SIGL_buf1 SIG_LEV1];
NOISL_buf1 = [NOISL_buf1 NOISE_LEV1];
THRS_buf1 = [THRS_buf1 THR_SIG1];
```

```
 skip = 0; %reset parameters
 not_nois = 0; %reset parameters
 ser_back = 0;  %reset bandpass param
end

if gr
hold on,scatter(qrs_i,qrs_c,'m');
hold on,plot(locs,NOISL_buf,'--k','LineWidth',2);
hold on,plot(locs,SIGL_buf,'--r','LineWidth',2);
hold on,plot(locs,THRS_buf,'--g','LineWidth',2);
if ax(:)
linkaxes(ax,'x');
zoom on;
end
end




%% overlay on the signals
if gr
figure,az(1)=subplot(311);plot(ecg_h);title('QRS on Filtered Signal');axis
tight;
hold on,scatter(qrs_i_raw,qrs_amp_raw,'m');
hold on,plot(locs,NOISL_buf1,'LineWidth',2,'Linestyle','--','color','k');
hold on,plot(locs,SIGL_buf1,'LineWidth',2,'Linestyle','-.','color','r');
hold on,plot(locs,THRS_buf1,'LineWidth',2,'Linestyle','-.','color','g');
az(2)=subplot(312);plot(ecg_m);title('QRS on MVI signal and Noise
level(black),Signal Level (red) and Adaptive Threshold(green)');axis tight;
hold on,scatter(qrs_i,qrs_c,'m');
hold on,plot(locs,NOISL_buf,'LineWidth',2,'Linestyle','--','color','k');
hold on,plot(locs,SIGL_buf,'LineWidth',2,'Linestyle','-.','color','r');
hold on,plot(locs,THRS_buf,'LineWidth',2,'Linestyle','-.','color','g');
az(3)=subplot(313);plot(ecg-mean(ecg));title('Pulse train of the found QRS on
ECG signal');axis tight;
line(repmat(qrs_i_raw,[2 1]),repmat([min(ecg-mean(ecg))/2; max(ecg-
mean(ecg))/2],size(qrs_i_raw)),'LineWidth',2.5,'LineStyle','-.','Color','r');
linkaxes(az,'x');
zoom on;
end

t1=toc;




end
```

APPENDIX D

Appendix D contains the MATLAB® of the Afonso algorithm

```
function [QRS,t2]=afonso(S,fs);
% nqrsdetect - detection of QRS-complexes
%
%   QRS=nqrsdetect(S,fs);
%
% INPUT
%   S       ecg signal data
%   fs      sample rate
%
% OUTPUT
%   QRS     fiducial points of qrs complexes
%
%
% see also: QRSDETECT
%
% Reference(s):
% [1]: V. Afonso, W. Tompkins, T. Nguyen, and S. Luo, "ECG beat detection
using filter banks,"
%   IEEE Trans. Biomed. Eng., vol. 46, no. 2, pp. 192--202, Feb. 1999
%
% [2]: A.V. Oppenheim, R.W. Schafer, and J.R. Buck,  Discrete-Time Signal
%   Processing, second edition, Prentice Hall, 1999, chapter 4.7.3

% Copyright (C) 2006 by Rupert Ortner
%
%% This program is free software; you can redistribute it and/or modify
%% it under the terms of the GNU General Public License as published by
%% the Free Software Foundation; either version 2 of the License, or
%% (at your option) any later version.
%%
%% This program is distributed in the hope that it will be useful, ...
%% but WITHOUT ANY WARRANTY; without even the implied warranty of
%% MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
%% GNU General Public License for more details.
%%
%% You should have received a copy of the GNU General Public License
%% along with this program; if not, write to the Free Software
%% Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA  02111-1307
%% USA
tic;

S=S(:);
S=full(S);
N=round(fs);   %Filter order
%---------------------------------------
%Replaces filter bank in [1]
Bw=5.6;     %filter bandwidth
Bwn=1/(fs/2)*Bw;
M=round((fs/2)/Bw); %downsampling rate

Wn0=Bwn;    %bandwidth of the first filter
```

```
Wn1=[Bwn 2*Bwn];    %bandwidth of the second filter
Wn2=[2*Bwn 3*Bwn];
Wn3=[3*Bwn 4*Bwn];
Wn4=[4*Bwn 5*Bwn];

h0=fir1(N,Wn0); %impulse response of the first filter
h1=fir1(N,Wn1,'bandpass');
h2=fir1(N,Wn2,'bandpass');
h3=fir1(N,Wn3,'bandpass');
h4=fir1(N,Wn4,'bandpass');

%Polyphase implementation of the filters
y=cell(1,5);
y{1}=polyphase_imp(S,h0,M); %W0 (see [1]) filtered and downsampled signal
y{2}=polyphase_imp(S,h1,M); %W1
y{3}=polyphase_imp(S,h2,M); %W2
y{4}=polyphase_imp(S,h3,M); %W3
y{5}=polyphase_imp(S,h4,M); %W4
%-----------------------------------------------

cut=ceil(N/M);  %Cutting off of initial transient because of the filtering
y1=[zeros(cut,1);y{1}(cut:length(y{1}))];
y2=[zeros(cut,1);y{2}(cut:length(y{2}))];
y3=[zeros(cut,1);y{3}(cut:length(y{3}))];
y4=[zeros(cut,1);y{4}(cut:length(y{4}))];
y5=[zeros(cut,1);y{5}(cut:length(y{5}))];
%----------------------------------------

P1=sum([abs(y2) abs(y3) abs(y4)],2); %see [1] equation (13)
P2=sum([abs(y2) abs(y3) abs(y4) abs(y5)],2);
P4=sum([abs(y3) abs(y4) abs(y5)],2);

FL1=MWI(P1); %Feature 1 according to Level 1 in [1]
FL2=MWI(P2); %Feature 2 according to Level 2
FL4=MWI(P4); %Feature 4 according to Level 4
%-------------------------------------
%Level 1 [1]
d=sign(diff(FL1));
d1=[0;d];
d2=[d;0];
f1=find(d1==1);
f2=find(d2==-1);
EventsL1=intersect(f1,f2); %Detected events
%-----------------------------------------------------------
%Level 2 [1]
meanL1=sum(FL2(EventsL1),1)/length(EventsL1);
NL=meanL1-meanL1*0.1;   %Start Noise Level
SL=meanL1+meanL1*0.1;   %Start Signal Level
threshold1=0.08;    %Threshold detection block 1
threshold2=0.7;     %Threshold detection block 2
[SignalL21,Noise1,DS1,Class1]=detectionblock(FL2,EventsL1,NL,SL,threshold1);
[SignalL22,Noise2,DS2,Class2]=detectionblock(FL2,EventsL1,NL,SL,threshold2);
%-----------------------------------------------------
%Level 3 [1]
ClassL3=[];
for i=1:length(EventsL1)
```

```
    C1=Class1(i);
    C2=Class2(i);
    if C1==1
        if C2==1
            ClassL3=[ClassL3 1];   %Classification as Signal
        else
            delta1=(DS1(i)-threshold1)/(1-threshold1);
            delta2=(threshold2-DS2(i))/threshold2;
            if delta1>delta2
                ClassL3=[ClassL3 1]; %Classification as Signal
            else
                ClassL3=[ClassL3 0];  %Classification as Noise
            end
        end
    else
        if C2==1;
            ClassL3=[ClassL3 1]; %Classification as Signal
        else
            ClassL3=[ClassL3 0];  %Classification as Noise
        end
    end
end
SignalL3=EventsL1(find(ClassL3));   %Signal Level 3
NoiseL3=EventsL1(find(ClassL3==0)); %Noise Level 3
%-------------------------------------------
%Level 4 [1]
threshold=0.3;
VSL=(sum(FL4(SignalL3),1))/length(SignalL3);
VNL=(sum(FL4(NoiseL3),1))/length(NoiseL3);
SL=(sum(FL4(SignalL3),1))/length(SignalL3);   %Initial Signal Level
NL=(sum(FL4(NoiseL3),1))/length(NoiseL3);      %Initial Noise Level
SignalL4=[];
NoiseL4=[];
DsL4=[];   %Detection strength Level 4
for i=1:length(EventsL1)
    Pkt=EventsL1(i);
    if ClassL3(i)==1;   %Classification after Level 3 as Signal
       SignalL4=[SignalL4,EventsL1(i)];
       SL=history(SL,FL4(Pkt));
       Ds=(FL4(Pkt)-NL)/(SL-NL);       %Detection strength
       if Ds<0
           Ds=0;
       elseif Ds>1
           Ds=1;
       end
       DsL4=[DsL4 Ds];
    else        %Classification after Level 3 as Noise
       Ds=(FL4(Pkt)-NL)/(SL-NL);
       if Ds<0
           Ds=0;
       elseif Ds>1
           Ds=1;
       end
       DsL4=[DsL4 Ds];
       if Ds>threshold          %new classification as Signal
           SignalL4=[SignalL4,EventsL1(i)];
           SL=history(SL,FL4(Pkt));
```

```
        else                        %new classification as Noise
            NoiseL4=[NoiseL4,EventsL1(i)];
            NL=history(NL,FL4(Pkt));
        end
    end
end
%-------------------------------------------------
%Level 5
%if the time between two RR complexes is too long => go back and check the
%events again with lower threshold
SignalL5=SignalL4;
NoiseL5=NoiseL4;
periods=diff(SignalL4);
M1=100;
a=1;
b=1/(M1)*ones(M1,1);
meanperiod=filter(b,a,periods); %mean of the RR intervals
SL=sum(FL4(SignalL4))/length(SignalL4);
NL=sum(FL4(NoiseL4))/length(NoiseL4);
threshold=0.2;
for i=1:length(periods)
    if periods(i)>meanperiod*1.5     %if RR-interval is to long
        intervall=SignalL4(i):SignalL4(i+1);
        critical=intersect(intervall,NoiseL4);
        for j=1:length(critical)
            Ds=(FL4(critical(j))-NL)/(SL-NL);
            if Ds>threshold         %Classification as Signal
                SignalL5=union(SignalL5,critical(j));
                NoiseL5=setxor(NoiseL5,critical(j));
            end
        end
    end
end
%-----------------------------------------------------
%Umrechnung auf Originalsignal (nicht downgesamplet)
Signaln=conversion(S,FL2,SignalL5,M,N,fs);
%-----------------------------------------------------
%Level 6 If interval of two RR-complexes <0.24 => go back and delete one of
them
height=FL2(SignalL5);
Signal=Signaln;
temp=round(0.1*fs);
difference=diff(Signaln);  %Difference between two signal points
k=find(difference<temp);
for i=1:length(k)
    pkt1=SignalL5(k(i));
    pkt2=SignalL5(k(i)+1);
    verg=[height(k(i)),height(k(i)+1)];
    [x,j]=max(verg);
    if j==1
        Signal=setxor(Signal,Signaln(k(i)+1)); %Deleting first Event
    else
        Signal=setxor(Signal,Signaln(k(i))); %Deleting second Event
    end
end

QRS=Signal;
```

```
t2=toc;



%--------------------------------------------------------------------
%--------------------------------------------------------------------
%--------------------------------------------------------------------
%subfunctions

function y=MWI(S)

% MWI - Moving window integrator, computes the mean of two samples
%   y=MWI(S)
%
% INPUT
%   S       Signal
%
% OUTPUT
%   y       output signal
a=[0;S];
b=[S;0];
c=[a,b];
y=sum(c,2)/2;
y=y(1:length(y)-1);
%-------------------------------------------------
function y=polyphase_imp(S,h,M)

% polyphase_imp - polyphase implementation of decimation filters [2]
%   y=polyphase_imp(S,h,M)
%
% INPUT
%   S       ecg signal data
%   h       filter coefficients
%   M       downsampling rate
%
% OUTPUT
%   y       filtered signal
%

%Determining polyphase components ek
e=cell(M,1);
l=1;
m=mod(length(h),M);
while m>0
    for n=1:ceil(length(h)/M)
        el(n)=h(M*(n-1)+l);
    end
    e{l}=el;
    l=l+1;
    m=m-1;
end
clear el;
for i=l:M
    for n=1:floor(length(h)/M)
```

```
        el(n)=h(M*(n-1)+i);
    end
    e{i}=el;
end
%Filtering
max=ceil((length(S)+M)/M);
Sdelay=S;
for i=1:M
    Sd=downsample(Sdelay,M);
    a=filter(e{i},1,Sd);
    if length(a)<max
        a=[a;zeros(max-length(a),1)];
    end
    w(:,i)=a;
    Sdelay=[zeros(i,1);S];
end
y=sum(w,2);
%-------------------------------------------------------------
function [Signal,Noise,VDs,Class]=detectionblock(mwi,Events,NL,SL,threshold)

% detectionblock - computation of one detection block
%
%   [Signal,Noise,VDs,Class]=detectionblock(mwi,Events,NL,SL,threshold)
%
% INPUT
%   mwi         Output of the MWI
%   Events      Events of Level 1 (see [1])
%   NL          Initial Noise Level
%   SL          Initial Signal Level
%   threshold   Detection threshold (between [0,1])
%
% OUTPUT
%   Signal      Events which are computed as Signal
%   Noise       Events which are computed as Noise
%   VDs         Detection strength of the Events
%   Class       Classification: 0=noise, 1=signal

Signal=[];
Noise=[];
VDs=[];
Class=[];
sumsignal=SL;
sumnoise=NL;
for i=1:length(Events)
    P=Events(i);
    Ds=(mwi(P)-NL)/(SL-NL); %Detection strength
    if Ds<0
        Ds=0;
    elseif Ds>1
        Ds=1;
    end
    VDs=[VDs Ds];
    if Ds>threshold     %Classification as Signal
        Signal=[Signal P];
        Class=[Class;1];
        sumsignal=sumsignal+mwi(P);
```

```
        SL=sumsignal/(length(Signal)+1);    %Updating the Signal Level
    else        %Classification as Noise
        Noise=[Noise P];
        Class=[Class;0];
        sumnoise=sumnoise+mwi(P);
        NL=sumnoise/(length(Noise)+1);  %Updating the Noise Level
    end
end
%-------------------------------------------------------------
function [pnew]=conversion(S,FL2,pold,M,N,fs)

% conversion - sets the fiducial points of the downsampled Signal on the
% samplepoints of the original Signal
%
%   [pnew]=conversion(S,FL2,pold,M,N,fs)
%
% INPUT
%   S           Original ECG Signal
%   FL2         Feature of Level 2 [1]
%   pold        old fiducial points
%   M           M downsampling rate
%   N           filter order
%   fs          sample rate
%
% OUTPUT
%   pnew        new fiducial points
%

Signaln=pold;
P=M;
Q=1;
FL2res=resample(FL2,P,Q);       %Resampling
nans1=isnan(S);
nans=find(nans1==1);
S(nans)=mean(S);    %Replaces NaNs in Signal
for i=1:length(Signaln)
    Signaln1(i)=Signaln(i)+(M-1)*(Signaln(i)-1);
end
%------------------- Sets the fiducial points on the maximum of FL2
Signaln2=Signaln1;
Signaln2=Signaln2';
int=2*M;    %Window length for the new fiducial point
range=1:length(FL2res);
for i=1:length(Signaln2)
    start=Signaln2(i)-int/2;
    if start<1
        start=1;
    end
    stop=Signaln2(i)+int/2;
    if stop>length(FL2res)
        stop=length(FL2res);
    end
    intervall=start:stop;       %interval
    FL2int=FL2res(intervall);
    pkt=find(FL2int==max(FL2int));  %Setting point on maximum of FL2
    if length(pkt)==0   % if pkt=[];
```

```
        pkt=Signaln2(i)-start;
    else
        pkt=pkt(1);
    end
    delay=N/2+M;
    Signaln3(i)=pkt+Signaln2(i)-int/2-delay;    %fiducial points according to
FL2
end
%Sets the fiducial points on the maximum or minimum
%of the signal
Bw=5.6;
Bwn=1/(fs/2)*Bw;
Wn=[Bwn 5*Bwn];
N1=32;
b=fir1(N1,Wn,'bandpass');
Sf=filtfilt(b,1,S);     %Filtered Signal with bandwidth 5.6-28 Hz
beg=round(1.5*M);
fin=1*M;
for i=1:length(Signaln3)
    start=Signaln3(i)-beg;
    if start<1
        start=1;
    end
    stop=Signaln3(i)+fin;
    if stop>length(Sf)
        stop=length(Sf);
    end
    intervall=start:stop;   %Window for the new fiducial point
    Sfint=abs(detrend(Sf(intervall),0));
    pkt=find(Sfint==max(Sfint));    %Setting point on maximum of Sfint
    if length(pkt)==0   %if pkt=[];
        pkt=Signaln3(i)-start;
    else
        pkt=pkt(1);
    end
    pkt=pkt(1);
    Signaln4(i)=pkt+Signaln3(i)-beg-1;
end
Signal=Signaln4';   %New fiducial points according to the original signal

cutbeginning=find(Signal<N);    %Cutting out the first points because of
initial transient of the filter in polyphase_imp
fpointsb=Signal(cutbeginning);
cutend=find(Signal>length(S)-N); %Cutting out the last points
fpointse=Signal(cutend);
pnew=setxor(Signal,[fpointsb;fpointse]);

%-----------------------------------------
function yn=history(ynm1,xn)

% history - computes y[n]=(1-lambda)*x[n]+lambda*y[n-1]
%
%   yn=history(ynm1,xn)

lambda=0.8; %forgetting factor
```

BIBLIOGRAPHY

[1] U. Rajendra Acharya, J. S. Suri, J. A. E. Spaan, and S. M. Krishnan, *Advances in cardiac signal processing*. 2007.

[2] D. E. Ward, "Noninvasive electrocardiology: Clinical aspects of holter monitoring: edited by arthur j. moss and shiomo stern w.b. saunders, philadelphia (1996) 542 pages, illustrated, $59.00 isbn: 9-7020-1925-9," *Clin. Cardiol.*, vol. 20, no. 3, pp. 312–312, Mar. 1997.

[3] "ABC of Clinical Electrocardiography: Francis Morris, William J. Brady, John Camm: 9781405170642: Amazon.com: Books." [Online]. Available: http://www.amazon.com/ABC-Clinical-Electrocardiography-Francis-Morris/dp/1405170646. [Accessed: 26-Apr-2015].

[4] "ECG Waveform Detail, Available at:https://anboswell.files.wordpress.com/2014/11/ecg-waveform-detail-from-medical-addicts.jpg." [Online]. Available: https://anboswell.files.wordpress.com/2014/11/ecg-waveform-detail-from-medical-addicts.jpg. [Accessed: 26-Apr-2015].

[5] Y.-C. Yeh and W.-J. Wang, "QRS complexes detection for ECG signal: The Difference Operation Method," *Comput. Methods Programs Biomed.*, vol. 91, no. 3, pp. 245–254, Sep. 2008.

[6] "https://anboswell.files.wordpress.com/2014/11/ecg-waveform-detail-from-medical-addicts.jpg." [Online]. Available: http://en.wikipedia.org/wiki/Electrocardiography. [Accessed: 26-Apr-2015].

[7] "Biomedical Signal Analysis: A Case-Study Approach: 9780471208112: Medicine & Health Science Books @ Amazon.com." [Online]. Available: http://www.amazon.com/Biomedical-Signal-Analysis-Case-Study-Approach/dp/0471208116. [Accessed: 26-Apr-2015].

[8] D. Mozaffarian, E. J. Benjamin, a. S. Go, D. K. Arnett, M. J. Blaha, M. Cushman, S. de Ferranti, J.-P. Despres, H. J. Fullerton, V. J. Howard, M. D. Huffman, S. E. Judd, B. M. Kissela, D. T. Lackland, J. H. Lichtman, L. D. Lisabeth, S. Liu, R. H. Mackey, D. B. Matchar, D. K. McGuire, E. R. Mohler, C. S. Moy, P. Muntner, M. E. Mussolino, K. Nasir, R. W. Neumar, G. Nichol, L. Palaniappan, D. K. Pandey, M. J. Reeves, C. J. Rodriguez, P. D. Sorlie, J. Stein, a. Towfighi, T. N. Turan, S. S. Virani, J. Z. Willey, D. Woo, R. W. Yeh, and M. B. Turner, "Heart Disease and Stroke Statistics--2015 Update: A Report From the American Heart Association," *Circulation*, vol. 131, no. 4, pp. e29–e322, 2014.

[9] "CDC - Chronic Disease - Heart Disease and Stroke Prevention - At A Glance," 2011.

[10] B. U. Köhler, C. Hennig, and R. Orglmeister, “The principles of software QRS detection,” *IEEE Engineering in Medicine and Biology Magazine*, vol. 21, no. 1. pp. 42–57, 2002.

[11] M. L. Ahlstrom and W. J. Tompkins, “Automated high-speed analysis of Holter tapes with microcomputers.,” *IEEE Trans. Biomed. Eng.*, vol. 30, no. 10, pp. 651–657, 1983.

[12] M. L. Ahlstrom and W. J. Tompkins, “Automated High-Speed Analysis of Holter Tapes with Microcomputers,” *IEEE Trans. Biomed. Eng.*, vol. BME-30, no. 10, pp. 651–657, Oct. 1983.

[13] J. Fraden and M. R. Neumann, “QRS Wave Detection,” *Med. Biol. Eng. Comput*, vol. 18, pp. 125–132, 1980.

[14] D. E. Gustafson, A. S. Willsky, J. Y. Wang, M. C. Lancaster, and J. H. Triebwasser, “ECG/VCG rhythm diagnosis using statistical signal analysis--I. Identification of persistent rhythms.,” *IEEE Trans. Biomed. Eng.*, vol. 25, no. 4, pp. 344–353, 1978.

[15] W. P. Holsinger, K. M. Kempner, and M. H. Miller, “A QRS Preprocessor Based on Digital Differentiation,” *IEEE Trans. Biomed. Eng.*, vol. BME-18, no. 3, 1971.

[16] P. Morizet-Mahoudeaux, C. Moreau, D. Moreau, and J. J. Quarante, “Simple microprocessor-based system for on-line e.c.g. arrhythmia analysis.,” *Med. Biol. Eng. Comput.*, vol. 19, no. 4, pp. 497–500, Jul. 1981.

[17] R. A. Balda, G. Diller, E. Deardorff, J. Doue, and P. Hsieh, “Trends in Computer-Processed Electrocardiograms,” in *The HP ECG analysis program*, W. J. van Bemmel JH, Ed. North Holland, Amsterdam, The Netherlands., 1977, pp. 197–205.

[18] P. O. Börjesson, O. Pahlm, L. Sörnmo, and M. E. Nygårds, “Adaptive QRS detection based on maximum a posteriori estimation.,” *IEEE Trans. Biomed. Eng.*, vol. 29, no. 5, pp. 341–351, 1982.

[19] Z. Dokur, T. Ölmez, E. Yazgan, and O. K. Ersoy, “Detection of ECG waveforms by neural networks,” *Med. Eng. Phys.*, vol. 19, no. 8, pp. 738–741, 1997.

[20] W. A. H. Engelse and C. Zeelenberg, “A single scan algorithm for QRS-detection and feature extraction,” *Comput. Cardiol.*, vol. 6, no. 9, pp. 37–42, 1979.

[21] T. Fancott and D. H. Wong, “A minicomputer system for direct high speed analysis of cardiac arrhythmia in 24 h ambulatory ECG tape recordings.,” *IEEE Trans. Biomed. Eng.*, vol. 27, no. 12, pp. 685–693, 1980.

[22] P. S. Hamilton and W. J. Tompkins, “Quantitative investigation of QRS detection rules using the MIT/BIH arrhythmia database.,” *IEEE Trans. Biomed. Eng.*, vol. 33, no. 12, pp. 1157–1165, 1986.

[23] L. Keselbrener, M. Keselbrener, and S. Akselrod, “Nonlinear high pass filter for R-wave detection in ECG signal,” *Med. Eng. Phys.*, vol. 19, no. 5, pp. 481–484, 1997.

[24] J. Leski and E. Tkacz, “A new parallel concept for QRS complex detector,” *1992 14th Annu. Int. Conf. IEEE Eng. Med. Biol. Soc.*, vol. 2, pp. 555–556, 1992.

[25] M. E. Nygards and J. Hulting, “An automated system for ECG monitoring,” *Comput. Biomed. Res.*, vol. 12, no. 2, pp. 181–202, 1979.

[26] J. Pan and W. J. Tompkins, “A real-time QRS detection algorithm.,” *IEEE Trans. Biomed. Eng.*, vol. 32, no. 3, pp. 230–236, 1985.

[27] L. Sornmo, O. Paklm, and M.-E. Nygards, “Adaptive QRS Detection: A Study of Performance,” *IEEE Transactions on Biomedical Engineering*, vol. BME-32, no. 6. pp. 392–401, 1985.

[28] Y. Sun, S. Suppappola, and T. A. Wrublewski, “Microcontroller-based real-time QRS detection,” *Biomed. Instrum. Technol.*, vol. 26, no. 6, pp. 477–484, 1992.

[29] S. Suppappola and Y. Sun, “Nonlinear transforms of ECG signals for digital QRS detection: A quantitative analysis,” *IEEE Trans. Biomed. Eng.*, vol. 41, no. 4, pp. 397–400, 1994.

[30] M. Okada, “A digital filter for the QRS complex detection.,” *IEEE Trans. Biomed. Eng.*, vol. 26, no. 12, pp. 700–703, 1979.

[31] A. Ligtenberg and M. Kunt, “A robust-digital QRS-detection algorithm for arrhythmia monitoring.,” *Comput. Biomed. Res.*, vol. 16, no. 3, pp. 273–286, 1983.

[32] Z. Dokur, T. Olmez, M. Korurek, and E. Yazgan, “Detection of ECG waveforms by using artificial neural networks,” *Proc. 18th Annu. Int. Conf. IEEE Eng. Med. Biol. Soc.*, vol. 3, 1996.

[33] B. C. YU, C. S. LIU, M. LEE, C. Y. CHEN, and B. N. CHIANG, “A nonlinear digital filter for cardiac QRS complex detection,” *J. Clin. Eng.*, vol. 10, no. 3, pp. 193–201, 1985.

[34] M. Bahoura, M. Hassani, and M. Hubin, “DSP implementation of wavelet transform for real time ECG wave forms detection and heart rate analysis,” *Comput. Methods Programs Biomed.*, vol. 52, no. 1, pp. 35–44, 1997.

[35] V. Di Virgilio, C. Francaiancia, S. Lino, and S. Cerutti, “ECG fiducial points detection through wavelet transform,” *Proc. 17th Int. Conf. Eng. Med. Biol. Soc.*, vol. 2, 1995.

[36] S. Kadambe, R. Murray, and G. F. Boudreaux-Bartels, “The dyadic wavelet transform based QRS detector [ECG analysis],” *[1992] Conf. Rec. Twenty-Sixth Asilomar Conf. Signals, Syst. Comput.*, 1992.

[37] C. Li, C. Zheng, and C. Tai, “Detection of ECG characteristic points using wavelet transforms,” *IEEE Trans Biomed Eng*, vol. 42, pp. 21–28, 1995.

[38] K. D. Rao, “DWT Based Detection of R-peaks and Data Compression of ECG Signals,” Mar. 2015.

[39] S. Mallat and W. L. Hwang, “Singularity detection and processing with wavelets,” *IEEE Trans. Inf. Theory*, vol. 38, no. 2 pt II, pp. 617–643, 1992.

[40] T. A. Gyaw and S. R. Ray, “The wavelet transform as a tool for recognition of biosignals,” in *Biomedical Sciences Instrumentation*, 1994, vol. 30, pp. 63–68.

[41] S. Mallat, “Zero-crossings of a wavelet transform,” *IEEE Trans. Inf. Theory*, vol. 37, no. 4, pp. 1019–1033, 1991.

[42] J. A. Crowe, N. M. Gibson, M. S. Woolfson, and M. G. Somekh, “Wavelet transform as a potential tool for ECG analysis and compression.,” *J. Biomed. Eng.*, vol. 14, no. 3, pp. 268–272, 1992.

[43] L. Khadra, A. S. al-Fahoum, and H. al-Nashash, “Detection of life-threatening cardiac arrhythmias using the wavelet transformation.,” *Med. Biol. Eng. Comput.*, vol. 35, no. 6, pp. 626–632, 1997.

[44] L. Senhadji, G. Carrault, J. J. Bellanger, and G. Passariello, “Comparing wavelet transforms for recognizing cardiac patterns,” *IEEE Eng. Med. Biol. Mag.*, vol. 14, no. 2, pp. 167–173, 1995.

[45] V. X. Afonso, W. J. Tompkins, T. Q. Nguyen, and S. Luo, “ECG beat detection using filter banks.,” *IEEE Trans. Biomed. Eng.*, vol. 46, no. 2, pp. 192–202, 1999.

[46] N. Fliege and N. Fliege, “Multirate digital signal processing: multirate systems, filter banks, wavelets,” 1994.

[47] H.-H. Bothe, *Neuro-Fuzzy-Methoden: Einführung in Theorie und Anwendungen*, vol. 18. Springer Berlin Heidelberg, 1997.

[48] C. M. Bishop, *Neural Networks for Pattern Recognition*, vol. 92. 1995.

[49] S. Haykin and N. Network, “A comprehensive foundation,” *Neural Networks*, 2004.

[50] S. Barro, M. Fernández-Delgado, J. A. Vila-Sobrino, C. V. Regueiro, and E. Sánchez, “Classifying multichannel ECG patterns with an adaptive neural network,” *IEEE Engineering in Medicine and Biology Magazine*, vol. 17, no. 1. pp. 45–55, 1998.

[51] M. Fernández-Delgado and S. Barro Ameneiro, “MART: A multichannel ART-based neural network,” *IEEE Trans. Neural Networks*, vol. 9, no. 1, pp. 139–150, 1998.

[52] F. M. Ham and S. Han, “Classification of cardiac arrhythmias using fuzzy ARTMAP.,” *IEEE Trans. Biomed. Eng.*, vol. 43, no. 4, pp. 425–430, 1996.

[53] Y. H. Hu, W. J. Tompkins, J. L. Urrusti, and V. X. Afonso, “Applications of artificial neural networks for ECG signal detection and classification.,” *J. Electrocardiol.*, vol. 26 Suppl, pp. 66–73, 1993.

[54] M. Lagerholm and G. Peterson, “Clustering ECG complexes using hermite functions and self-organizing maps,” *IEEE Trans. Biomed. Eng.*, vol. 47, no. 7, pp. 838–848, 2000.

[55] P. Laguna, R. Jane, S. Olmos, N. V Thakor, H. Rix, and P. Caminal, “Adaptive estimation of QRS complex wave features of ECG signal by the Hermite model.,” *Med. Biol. Eng. Comput.*, vol. 34, no. 1, pp. 58–68, 1996.

[56] N. Maglaveras, T. Stamkopoulos, K. Diamantaras, C. Pappas, and M. Strintzis, “ECG pattern recognition and classification using non-linear transformations and neural networks: A review,” in *International Journal of Medical Informatics*, 1998, vol. 52, no. 1–3, pp. 191–208.

[57] N. Maglaveras, T. Stamkopoulos, C. Pappas, and M. Strintzis, “ECG processing techniques based on neural networks and bidirectional associative memories,” *J Med Eng Technol*, vol. 22, no. 3, pp. 106–111, 1998.

[58] N. Mahalingam and D. Kumar, “Neural networks for signal processing applications: ECG classification.,” *Australas. Phys. Eng. Sci. Med.*, vol. 20, no. 3, pp. 147–151, 1997.

[59] Y. Suzuki, “Self-organizing QRS-wave recognition in ECG using neural networks,” *IEEE Trans. Neural Networks*, vol. 6, no. 6, pp. 1469–1477, 1995.

[60] P. Trahanias and E. Skordalakis, “Syntactic pattern recognition of the ECG,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 12, no. 7, pp. 648–657, 1990.

[61] A. Kyrkos, E. Giakoumakis, and G. Carayannis, “QRS detection through time recursive prediction techniques,” *Signal Processing*, 1988.

[62] D. A. Coast, R. M. Stern, G. G. Cano, and S. A. Briller, “An approach to cardiac arrhythmia analysis using hidden Markov models,” *IEEE Trans. Biomed. Eng.*, vol. 37, no. 9, pp. 826–836, 1990.

[63] P. E. Trahanias, “An approach to QRS complex detection using mathematical morphology,” *IEEE Trans. Biomed. Eng.*, vol. 40, no. 2, pp. 201–205, 1993.

[64] Q. Xue, Yu Hen Hu, and W. J. Tompkins, “Neural-network-based adaptive matched filtering for QRS detection,” *IEEE Trans. Biomed. Eng.*, vol. 39, no. 4, pp. 317–329, 1992.

[65] A. Ruha, S. Sallinen, and S. Nissilä, “A real-time microprocessor QRS detector system with a 1-ms timing accuracy for the measurement of ambulatory HRV,” *IEEE Trans. Biomed. Eng.*, vol. 44, no. 3, pp. 159–167, 1997.

[66] D. Ebenezer and V. Krishnamurthy, “Wave digital matched filter for electrocardiogram preprocessing.,” *J. Biomed. Eng.*, vol. 15, no. 2, pp. 132–134, 1993.

[67] K. G. Lindecrantz and H. Lilja, “New software QRS detector algorithm suitable for real time applications with low signal-to-noise ratios.,” *J. Biomed. Eng.*, vol. 10, no. 3, pp. 280–284, 1988.

[68] S. E. DOBBS, N. M. SCHMITT, and H. S. OZEMEK, “QRS Detection By Template Matching Using Real-Time Correlation On A Microcomputer,” *J. Clin. Eng.*, vol. 9, no. 3, 1984.

[69] R. Poli, S. Cagnoni, and G. Valli, “Genetic design of optimum linear and nonlinear QRS detectors,” *IEEE Trans. Biomed. Eng.*, vol. 42, no. 11, pp. 1137–1141, 1995.

[70] Z. S.-K. Z. Song-Kai, W. J.-T. W. Jian-Tao, and X. J.-R. X. Jun-Rong, “The real-time detection of QRS-complex using the envelope of ECG,” *Proc. Annu. Int. Conf. IEEE Eng. Med. Biol. Soc.*, 1988.

[71] F. Gritzali, “Towards a generalized scheme for QRS detection in ECG waveforms,” *Signal Processing*, 1988.

[72] B. Kohler, “QRS detection using zero crossing counts,” *Appl. ...*, 2003.

[73] G. Papakonstantinou and F. Gritzali, “Syntactic filtering of ECG waveforms,” *Comput. Biomed. Res.*, 1981.

[74] E. Skordalakis, “Recognition of noisy peaks in ECG waveforms,” *Comput. Biomed. Res.*, 1984.

[75] P. Trahanias and E. Skordalakis, “Syntactic pattern recognition of the ECG,” *Pattern Anal. Mach. ...*, 1990.

[76] E. Ciaccio, S. Dunn, and M. Akay, “Biosignal pattern recognition and interpretation systems. 4. Review of applications,” *... Med. Biol. Mag. ...*, 1994.

[77] Y. Wang, C. J. Deepu, and Y. Lian, “A computationally efficient QRS detection algorithm for wearable ECG sensors,” in *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBS*, 2011, pp. 5641–5644.

[78] A. L. Goldberger, L. A. 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*, vol. 101, no. 23, pp. E215–E220, 2000.

[79] T. Ince, S. Kiranyaz, and M. Gabbouj, "A generic and robust system for automated patient-specific classification of ECG signals," *Biomed. Eng. IEEE ...*, 2009.

[80] K. Tavakolian, G. a Dumont, G. Houlton, and A. P. Blaber, "Precordial vibrations provide noninvasive detection of early-stage hemorrhage.," *Shock*, vol. 41, no. 2, pp. 91–6, 2014.

[81] "American Heart Association, AHA Database, ECRI, 5200 Butler Pike, Plymouth Meeting, PA 19462 USA." .

[82] "American National Standard for Ambulatory Electrocardiographs, publication ANSI/AAMI EC38-1994, Association for the Advancement of Medical Instrumentation, 1994." .

[83] "J.H. van Bemmel and J.L. Williams, 'Standardisation and validation of medical decision support systems: The CSE project,' Methods Inform. Med., vol. 29, pp. 261-262, 1990." .

[84] "VTT Technical Research Center of Finland. IMPROVE Data Library. Available: http://www.vtt.fi/tte/samba/projects/improve/." .

[85] "National Research Council (CNR). European ST-T Database. Institute of Clinical Physiology, Dept. of Bioengieering and Medical Informatics, Pisa, Italy. Available: http://www.ifc.pi.cnr.it/." .

[86] "Physikalisch-Technische Bundesanstalt. ECG Reference Data Set. Available: h t t p : / / w w w . b e r l i n . p t b . d e / 8 / 8 3 / 8 3 1 /dbaccess/ecgrefdataset.html." .

[87] "P. Laguna, R.G. Mark, A. Goldberger, and G.B. Moody, 'A database for evaluation of algorithms for measurement of QT and other waveform intervals in the ECG,' Comput. Cardiology, vol. 24, pp. 673-676, 1997." .

[88] "Massachusetts General Hospital, Massachusetts General Hospital/Marquette Foundation Waveform Database, Dr. J. Cooper, MGH, Anesthesia Bioengineering Unit, Fruit Street, Boston, MA." .

[89] H. Sedghamiz, "Complete Pan Tompkins Implementation ECG QRS detector," 2014. [Online]. Available: http://www.mathworks.com/matlabcentral/fileexchange/45840-complete-pan-tompkins-implementation-ecg-qrs-detector. [Accessed: 27-Apr-2015].

[90] A. Delorme and S. Makeig, "EEGLAB - Open Source Matlab Toolbox for Electrophysiological Research," 2004. [Online]. Available: http://sccn.ucsd.edu/eeglab/. [Accessed: 27-Apr-2015].