American Journal of Biomedical Engineering 2016, 6(6): 180-201
DOI: 10.5923/j.ajbe.20160606.04

EKG Signals – De-noising and Features Extraction

Saleh Alomari*, **Mohammed Shujauddin, Vahid Emamian**

Department of Electrical Engineering, St. Mary's University, Camino Santa Maria, San Antonio Texas, USA

Abstract The need of effective method to obtain and analyse electrocardiogram (EKG) signal has inspired This research paper to designed an efficient algorithm that can handle any (EKG), remove the most dominant noises associated with it, and extract the important futures. EKG signal is an electrical signal represents the physical human's heart activity. Nonetheless, this signal is affected by various noise including baseline wondering and power interference. These noises affect the signal to noise ratio (SNR) especially in P and T waves which have less amplitudes than R peaks. Removing these noises result in cleaner signal that can be conveniently processed to extract important features such as heart health condition. EKG features play the main role in diagnosing the heart rate, normality and abnormality of heart activities, and heart diseases. For a healthy person, one heart beat consists of P, QRS Complex, T, and in some signals U waves. In this paper, a robust and numerically sufficient algorithm is developed to de-nosing EKG signal and extract all major features. For de-nosing EKG signal, FIR Equiripple High pass filter is used. FIR Equiripple Low pass filter follows this filter to remove the power interference noises. Haar wavelet transform is used to accurately detect the R peaks. Haar wavelet is found to be better than other common methods that are used to detect R peaks. Haar wavelet shows high accuracy when it is applied on EKG signal to detect R peaks. In fact, it succeeded to detect all R peaks in hundreds of EKG signals (obtained from Physio net website). All other features are detected based on the R peaks by creating a set of windows which their lengths depend on the maximum normal wave durations and locations. These filters and algorithm have been implemented in Matlab. The algorithm has been applied on 108 EKG signals collected from physionet website and could detect all EKG signals' heart rates successfully despite the fact that some signals were extremely distorted.

Keywords EKG, ECG, Base line noise, Power interference noise, FIR Equiripple High pass filter, FIR Equiripple Low pass filter, Zero phase filter, Haar wavelet transform, QRS complex detection, P wave detection, T wave detection, Matlab

1. Introduction

The EKG is a graphical record represents the cardiac physical activities which are created by re-polarization and depolarization of atria and ventricular of the heart [1]. Every heart beat consists of P, QRS Complex, T waves. Those waves are extremely important in analysis heart condition, if they are present, they must be within certain amplitude and duration limits. Exceeding these maximum limits or failing to reach these minimum limits indicate illness. Absence of any of these waves is a sign of certain type of heart diseases. Fig (1) shows the typical EKG signal. Feature extraction through accurate waves detection is significant to measure heart rate and find any suspicion of diseases related to arrhythmias such as Heart Rate Variation, Tachycardia, Bradycardia. These diseases can be diagnosed by observing the abnormalities on the heart beats [2]. Nonetheless, detecting EKG waves is not easy due to time-varying morphology of the investigated signal and occurrence of noises. This paper implements a robust and an effective algorithm that is used to detect R peaks and based on that extracts the most important features of EKG signal.

* Corresponding author:
engsalehalomari@gmail.com (Saleh Alomari)
Published online at http://journal.sapub.org/ajbe

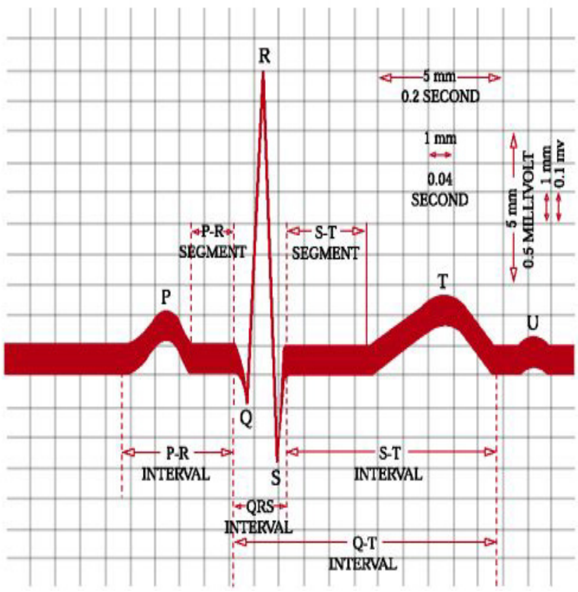
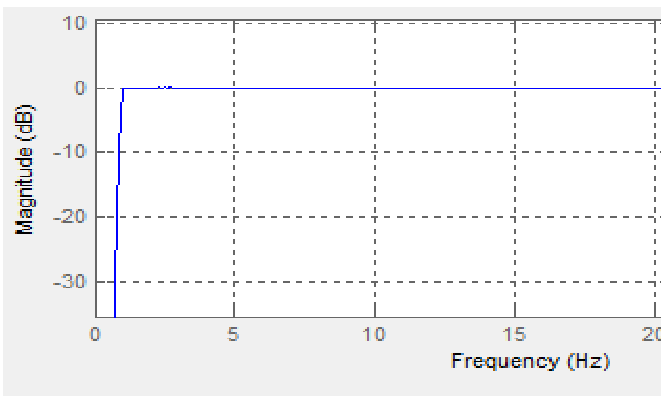


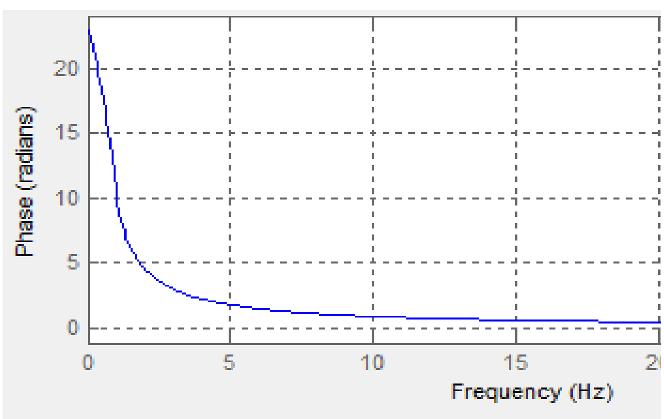
Figure 1. Normal EKG signal

2. Methodology

EKG signal holds all major features that can be extracted to diagnose heart condition. Unfortunately, the EKG signal obtained from a patient is corrupted by a lot of noises. Therefore, it must be preprocessed before extracting any feature. The wondering baseline and power interference noises are the main noises that must be removed from the signal. Wondering baseline noise presents due to low frequency produced by patient respiration. This low frequency ranges between 0.05 to 0.5 Hz [3]. Hence, High pass filter is capable to eliminate this noise from EKG signal. Nevertheless, there are enormous type of filters that can be applied. Choosing appropriate filter type is not easy task. Every filter has its own properties, advantages, and drawbacks. Choosing the right filter should be based on the signal that is processed and desired outcomes. For example, IIR filter type capable of removing the baseline noise from the signal. However, this type of filter introduces some distortion of the original signal since this type of filter has nonlinear phase response. Figure 2 shows the typical amplitude and phase response of IIR Butterworth filter. This distortion cannot be tolerated in EKG signal. Distortion means loss of some important information. Thus, FIR filter is proposed to be used.

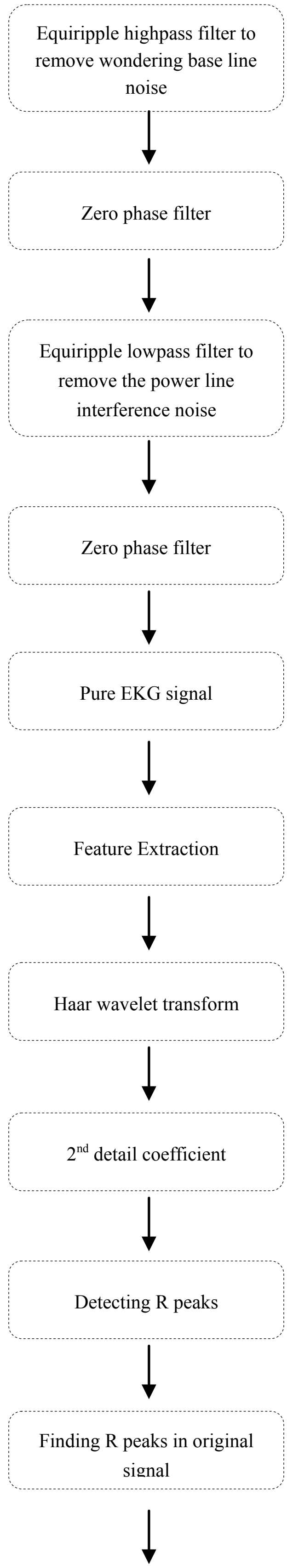


(a)



(b)

Figure 2. Amplitude (a) and phase respose (b) For IIR buterworth filter



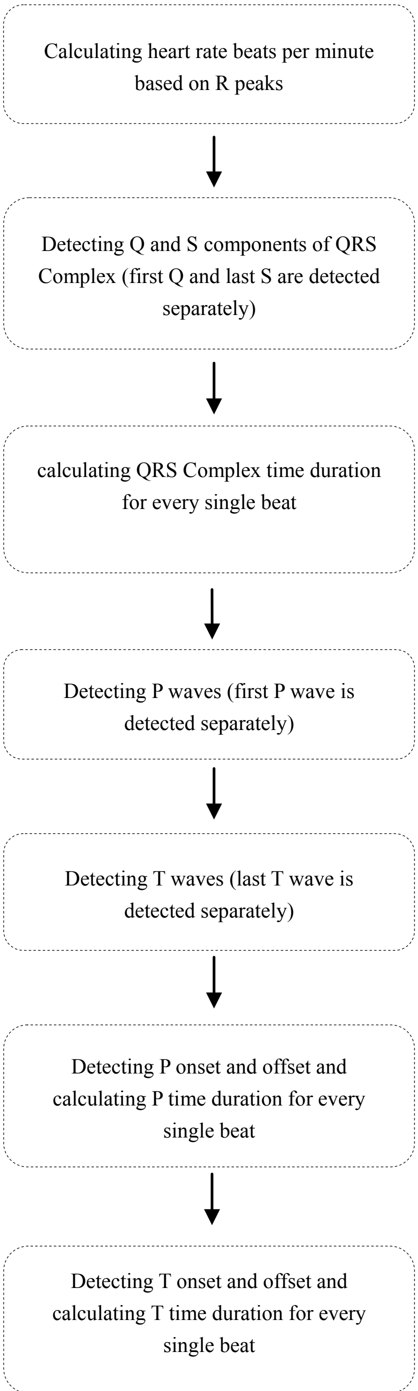


Figure 3. Flowchart for algorithm used in this paper

Hence, this paper used a very effective filter that has narrow transition width and optimum filter length that meet desired filter specification. After effectively removing all type of noises, the signal is ready to be processed for feature extraction. Haar wavelet transform is used for detecting the R peaks. Based on R peaks, all other features have been extracted using windows with different sizes for every wave. Figure 3 shows the flowchart for the algorithm that is used in this paper to de-noise EKG signal and extract the important features out of it.

3. Filter Design

Removing the wandering baseline noise needs a highly efficient filter that has short transition band. This noise corresponds to frequency ranges between 0.15 to 0.5 Hz. This sharp edges mathematically forms discontituaty that can not be implimented in practice. Designning effective filter depends mainly on finding filter with low order whose frequency response efficinetly approximate the desireded specification. Among all type of filters, the Equiripple FIR filter is superior of optimizing the transition width and ripple hight in both stop and pass bands. Equiriplle filter algorithm has been developed by parks and mcclellan. They derived a new algorithm from the general remez exchange algorithm. The parks and mcclellan filter works to minimizing chebychev error [4]. The parks mcclellan algoritnm considers wighted approximation error between designed and intended frequency response which is distributed evenly across passband and stopband minimizing the maximum error. The difference equation for FIR filter design in general is given as:

$$y(n) = b_0 x(n) + b_1 x(n-1) + ... + b_{M-1} x(n-M+1) \qquad (1)$$

$$y(n) = \sum_{k=0}^{M-1} b_k x(n-k) \qquad (2)$$

b_k is the filter coefficients, the output can be presented also as a form of input signal convolve with unit response as following:

$$y(n) = \sum_{k=0}^{M-1} h(k) x(n-k) \qquad (3)$$

The FIR filter can be described by the system function as following equation:

$$H(z) = \sum_{k=0}^{M-1} h(k) z^{-1} \qquad (4)$$

3.1. FIR Filter Designing

To precisely describe Equiripple filter designing, the ripple magnitude that occurs in passband and stopband must be bounded by the following limits [5]:

$$1 - \delta_1 \le H_r(\omega) \le 1 + \delta_1 \qquad |\omega| \le \omega_p \qquad (5)$$

$$-\delta_2 \le H_r(\omega) \le \delta_2 \qquad |\omega| > \omega_s \qquad (6)$$

Where δ_1 and δ_2 are the ripples in passband and stopband respectively. There are 4 cases that result in a linear FIR filter. These cases can be handled by equiriple filter. Two of these cases are the symmetric unit sample response and the other two are the antisymmetric. In both cases, the filter order can be either even or odd. The following table summarize all cases:

Table 1. Frequency response functions for linear phase FIR filters

Filter type	$Q(\omega)$	$P(\omega)$
$h(n) = h(M-1-n)$ M odd Case (1)	1	$\sum_{k=0}^{(M-1)/2} a(k)\cos(\omega k)$
$h(n) = h(M-1-n)$ M even Case (2)	$\cos\frac{\omega}{2}$	$\sum_{k=0}^{(M/2)-1} b^{\sim}(k)\cos(\omega k)$
$h(n) = -h(M-1-n)$ M odd Case (3)	$\sin\omega$	$\sum_{k=0}^{(M-3)/2} c^{\sim}(k)\cos(\omega k)$
$h(n) = -h(M-1-n)$ M even Case (4)	$\sin\frac{\omega}{2}$	$\sum_{k=0}^{(M/2)-1} d^{\sim}(k)\cos(\omega k)$

The frequency response $H(\omega)$ can be expressed as:

$$H(\omega) = Q(\omega)\,P(\omega) \tag{7}$$

Where

$$Q(\omega) = \begin{cases} 1 \\ \cos\frac{\omega}{2} \\ \sin\omega \\ \sin\frac{\omega}{2} \end{cases}$$

$$P(\omega) = \alpha(k)\cos(\omega k)$$

This is a common form of equiriple filter where the length of filter (L) and coefficient $\alpha(k)$ changes based on which linear phase case is presented. Thus, Parks and McClellan algorithm can be implemented by finding symmetric or antisymmetric which minimizes the maximum weighted Chebyshev error as follow:

$$|E(\omega)| = \min[\max_{\omega\in B}(W(\omega)(H(\omega) - D(w))] \tag{8}$$

Where $H(\omega)$ is the actual frequency response, $D(\omega)$ is the desired frequency response, $W(\omega)$ is the weighted Chebyshev error, and B ranges between $[0, \pi]$. Plugging in the equivalent form of $H(\omega)$ and manipulating the previous equation results in the final form of the weighted error function:

$$|E(\omega)| = \min[\max_{\omega\in \overline{B}}(\overline{W}(\omega)(P(\omega) - \overline{D}(w))] \tag{9}$$

Where

$$\overline{W}(\omega) = W(\omega)Q(\omega) \qquad \overline{D}(\omega) = \frac{D(\omega)}{Q(\omega)}$$

$$P(\omega) = \sum_{k=0}^{M} \alpha(k)\cos(\omega k)$$

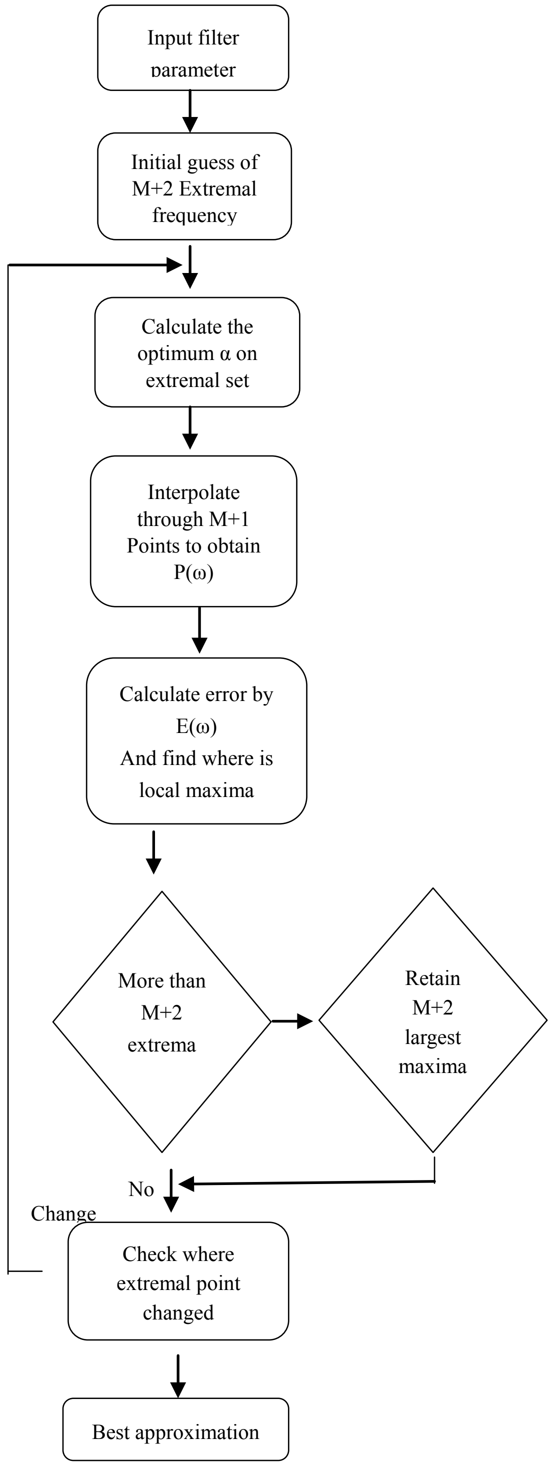


Figure 4. Flowchart for algorithm used by Parks McClellan

Alteration theorem:

$\overline{B}$ is the subset of interval $[0, \pi]$ which consists of frequencies of desired filter in both passband and stopband. $\overline{B}$ can be described by the following equation:

$$\overline{B} = B - \left(\text{endpoint where } Q(\omega) = 0\right) \tag{10}$$

Thus, there are in $\overline{B}$ at least M+2 extremal points $\omega 1$, ……, $\omega L+2$, such that:

$$E(\omega_i) = c(-1)^i[E(\omega)] \tag{11}$$

where i = 1, 2, ……..., M+2.

$|E(\omega)|$ reaches its maximum point at minimum of M+2 points. The error function changes its sign between two successive extremal frequencies from which this theorem takes its name “alteration theorem”. As a result, the weighted error function shows an equiripple manner. Figure 4 describes the Parks McClellan algorithm to design equiripple filter [5].

3.2. Equiripple Highpass Filter Used to Remove Baseline Noise

Baseline noise typically corrupts EKG signal due to patient’s respiration, motion of patient’s body, and electrodes. This noise could mask some important features. Therefore, it is extremely important to remove this noise. Equiripple highpass filter is capable of removing this noise completely without affecting the other important features of the signal. Equiripple highpass filter allows the main components of EKG signal to pass on such as P, QRS complex, and T waves as well as PR segment, PR interval, ST segment and QT interval. All mentioned intervals and segments correspond to certain frequencies. Hence, maintaining important frequencies are crucial. Based on American health association, the smallest component frequency is 0.05 Hz. Nonetheless, in practical, the baseline noise has frequency extend to the value of 1Hz. This means there is some feature that is distorted due to using highpass filter [6]. Nevertheless, the ST segment is not the area of interest in this paper. In fact, among the advantages that Equiripple filter has is the narrow band width that can be built and this feature can be maintained. However, building narrow transition band width that would maintain the ST segment using Equiripple requires filter with high order exceeds 5000. This order prevents us from using the zero phase filter built in as a function in Matlab to eliminate the time delay introduced by high pass filter which is crucial to preserve important feature. As a matter of fact, Matlab does not recognize zero or negative indices value that might be introduced due to time delay. This built-in function (filtfilt) requires the signal length to be more than three times of filter order. The Equiripple highpass filter used has a filter order of 2746, cutoff prequency at 1 Hz, stop frequency at 2 Hz, and stop attenuation of 80 dB. Figure 5,6,7 shows the original EKG signal, FFT of the signal, and EKG after removing baseline nose from the signal. It is so obvious that the baseline noise is completely removed while all features are preserved. After this stage, the DC offset due to baseline noise is successfully and completely removed.

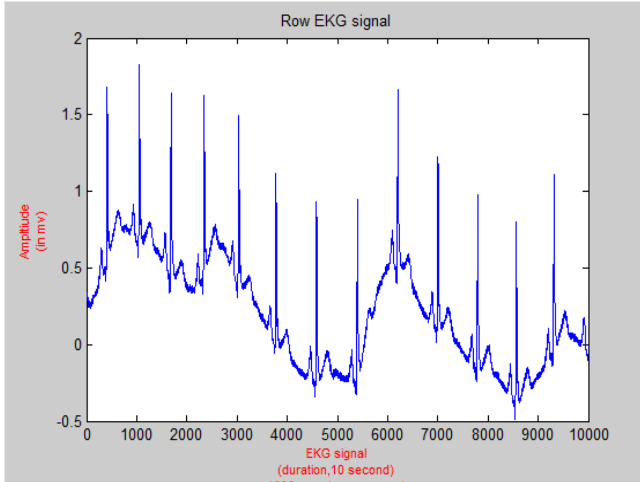


Figure 5. Row EKG Signal

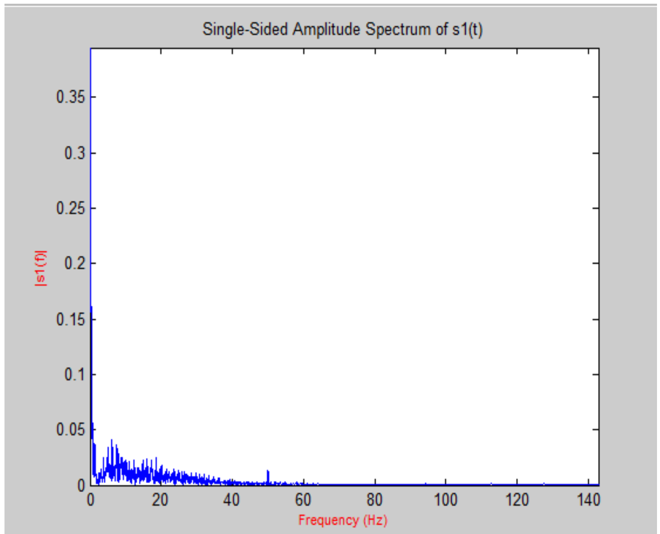


Figure 6. FFT for the EKG Signal

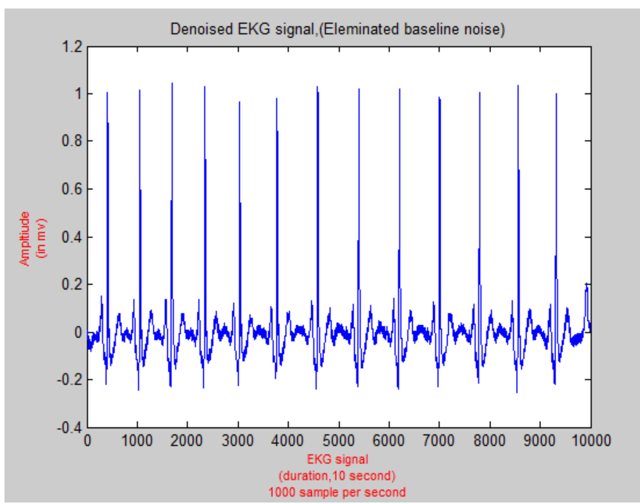


Figure 7. EKG after completely removing baseline noise

3.3. Equiripple Lowpass Filter Used to Remove Power Interference Noise

Due to improper grounding, power line noise interferes with EKG signal. This interference adds up 50 or 60 Hz (depends on power frequency standard that is used). The power interference noise appears as spike in frequency components analysis (FFT) Fig.6 at 50 Hz. This frequency component can be removed by using notch filter. However, all other frequency components which exceeds this value (50 Hz) are not important and does not contribute to the important features that we are looking for. Therefore, lowpass filter is adequate for this purpose. FIR equiripple lowpass filter is used with filter order of 506. The cutoff frequency is at 40 Hz. This filter is followed also by another filter with zero phase for avoiding time delay using same Matlab function filtfilt. Fig.8 shows the magnitude (in dB) and phase response of the designed filter.

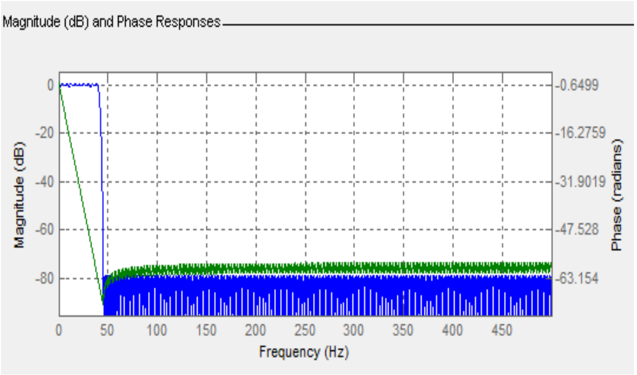


Figure 8. Magnitude and phase response of the designed equiripple lowpass filter

Figure 9 shows the resultant signal after removing the power interference noise using FIR Equiripple Lowpass filter.

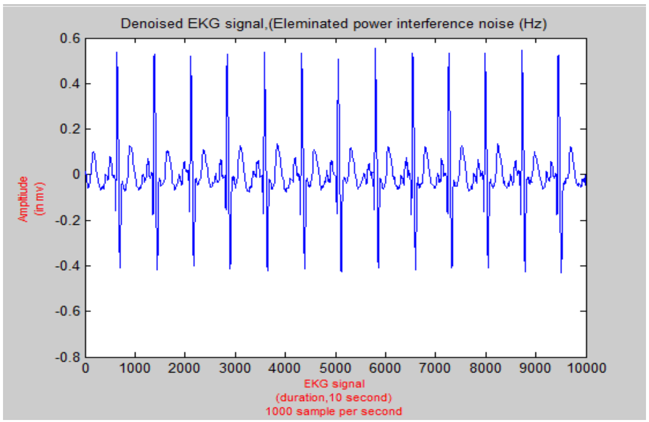


Figure 9. EKG signal after removing power interference noise

4. Features Extraction

Now, the EKG signal is ready to be processed for features extraction. In this stage, there are many methods and algorithms can be used to extract the EKG features. Some of these algorithms are easy to be implemented while others are complicated. Nevertheless, Haar wavelet transform is selected to be the method that is used to extract the EKG features. This decision was not arbitrary, in fact, based on many research papers, Haar wavelet is outstanding and promising. It provided high accuracy when it is applied to signals to detect important features.

4.1. Haar Wavelet Transform

Haar wavelet first introduced by Alfred Haar in 1910. Then, many definition and generalization follow it [7]. Haar wavelet is widely used in image coding, edge extraction, and feature extraction. The advantage of wavelet transform over the Fourier transform is that the wavelet transform can keep track of both time and frequency while in Fourier transform the high frequency in a short time is hard to be detected. Haar wavelet decomposes the processed signal into two sub-signals of half of its original length. In fact, there are two main functions that form wavelet analysis. The scaling function Φ and the wavelet Ψ. Figure 10 shows the basic Haar scaling function.

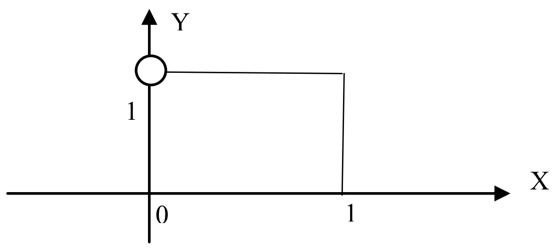


Figure 10. Haar Scaling Function

multiresolution analysis. The Haar scaling function as follows [8]:

$$\Phi(x)=\begin{cases}1, & \text{if } 0 \le x < 1\\ 0, & \text{elsewhere}\end{cases} \tag{12}$$

The same signal can be shifted over any finite set of integers. Let V_0 is the function that is shifted and scaled as follows (figure 11 shows elements in $V_{0)}$:

$$V_0=\sum_{k\in z} a_k\varphi(x-k) \qquad a_k \in R \tag{13}$$

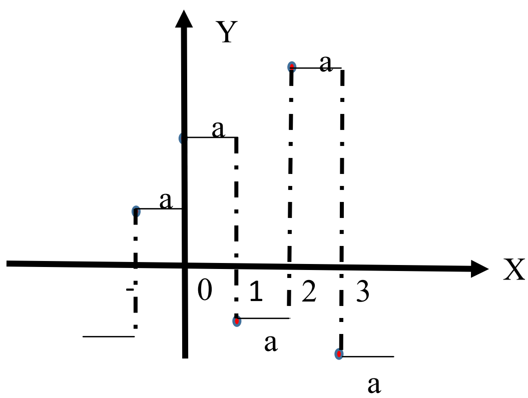


Figure 11. V_0 Components

4.2. General Form of Haar Wavelet

Assume i is a positive integer, V_i is the space spanned by the following set:

$$\{.........,\varphi(2^i x+1),\varphi(2^i x),\varphi(2^i x-1),.........\} \quad (14)$$

A function in V_0 is contained in V_1 and so forth.

$$V_0 \subset V_1 \subset V_{i-1} \subset V_i \subset V_{i+1} \quad (15)$$

V_i has all information up to scale 2^{-i}. when i become larger, the resolution become finer. The fact that $V_i \subset V_{i+1}$ indicates that there is no information is missed when the resolution become finer. The representation of $\Phi\ (2^i x)$ is spike of width $(1/2^i)$. Thus, when i is large, the $\Phi\ (2^i x)$ is just spike that it might be filtered out if we desired to. For filtering noises, the wavelet Ψ must be part of Haar wavelet to isolate those spikes that is mentioned previously. The main concept is to decompose the V_i as an orthogonal sum of V_i and its complement. To clarify this, let i = 1. Thus, we have V_1 and V_0. V_0 is found by Φ and its shifted form. Therefore, V_1 must be orthogonal and generated by the mother function Ψ. Ψ function must satisfy certain condition to be valid as a complement for V_0. Ψ must be contained in V_1. Hence, it can be expressed as:

$$\psi(x)=\sum_K a_K \varphi(2x-K) \quad (16)$$

$$\int \psi(x)\varphi(x-k)dx=0 \quad (17)$$

The simplest $\Psi(x)$ that is satisfy both conditions is the following function:

$$\psi(x)=\varphi(2x)-\varphi(2x-1) \quad (18)$$

It is obvious that this function satisfies the first condition (Ψ is contained in V_1) as well as the second condition (Ψ is orthogonal to V_0) as it is proofed in the following equation:

$$\int_{-\infty}^{+\infty} \psi(x)\varphi(x)=\int_0^{1/2} 1dx-\int_{1/2}^{1} 1dx=0 \quad (19)$$

Thus, $\Psi(x)$ is called Haar wavelet. Figure 12 represents haar wavelet that has the amplitude of 1 ($a_1 = 1$) at $\varphi\ (2x)$ and amplitude of -1 (a_2=-1) at $\varphi\ (2x\text{-}1)$.

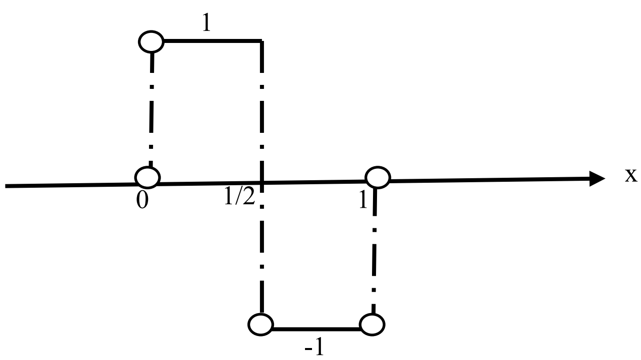


Figure 12. Haar Wavelet $\Psi(x)$

The previous function is a form of Haar wavelet that consists of the wavelet $\Psi(x)$ and the scaling function φ. Those two functions used to decompose a given signal and reconstruct it. The scaling function can be controlled to give wider or thiner scale that is called multiresolution analysis of a signal which helps to deeply diagnose the signal and filter out an undesirable component or noise [8].

4.3. Haar Wavelet Decomposition Algorithm

For a given function f, V_i is the nested spaces such that:

$$\ldots V_{-2} \subset V_{-1} \subset V_0 \subset V_1 \subset V_{2\ldots} \quad (20)$$

Assume W_i is the orthogonal of V_i with respect to subsequent space V_{i+1}. Therefore,

$$V_{i+1} = V_i \oplus W_i \quad (21)$$

That is, W_i has all missing details from V_i to obtain V_{i+1}. Therefore, by repetition, any space V_i can be obtain by following formula [9]:

$$V_i = W_i \otimes W_{i-1} \otimes W_{i-2} \otimes W_{i-3} \ldots \quad (22)$$

Assuming f is a function in V_i, then, f can be decomposed as following equation:

$$f = w_{r-1} + w_{r-2} + \cdots + w_0 + f_0 \quad (23)$$

w_r denote spikes of (f) that has the width $1/(2^{j+1})$.

For r large enough, these spikes are narrow enough to represent noise, assume spike of width 0.001 denotes noise; then, $2^{-10} < 0.001 < 2^{-9}$.

Therefore, any w_r with $r \geq 9$ denotes noise, to remove these noises, these components (w_r) set to zero value. The remaining components represents the signal that is free of noise [8].

5. Detecting Important Features through Applying Haar Wavelet to EKG

In many research papers that compared many methods used for features extraction, wavelet transform in general and Haar wavelet in specific has introduced the highest accuracy. Haar wavelet produce multiresolution analysis for a signal. Using this method, R Peaks detection is easily obtained. Haar wavelet transform generate two coefficients called approximation and detail coefficients. In the second detail coefficient, R peaks are dominant since the QRS complex has a higher frequency in a shorter time.

Figure 13 shows the decomposed EKG signal second detail coefficient using Matlab function wavdec and detcoef. Hence, R peaks are easy to be detected using Haar wavelet transform. In this stage, the amplitude threshold is applied. Signal that has at least 60% of the maximum value is maintained and other samples are set to zero (see figure 14).

The result was a bunch of values that are adjacent and repeated every period. Those values are representing the R peaks. Nonetheless, R peak should be represented by a unique value not by a bunch of values. Therefore, the highest amplitude value of each bunch of values is selected to be unique R peak. However, R peaks locations might be different in Haar wavelet from locations in the original signal. Therefore, a window of 100 samples of width is used to being searched through in the original signal after each R peaks location is multiplied by 4. This multiplication because in the second detail coefficient, the signal length is reduced by 4; this is the Haar wavelet property. Figure 15 shows the detected R peaks in the original EKG signal.

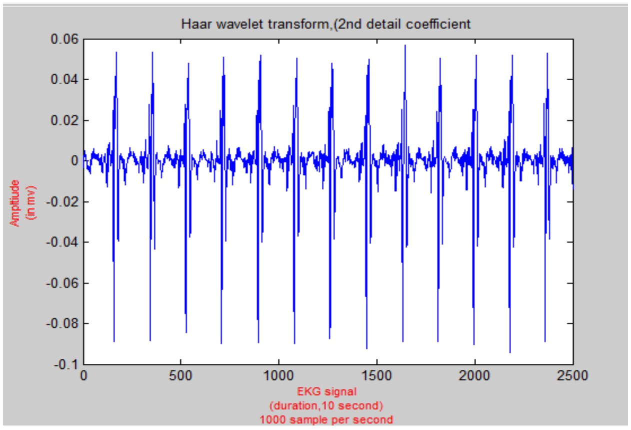


Figure 13. Haar Wavelet Transform Second Detail Coefficient

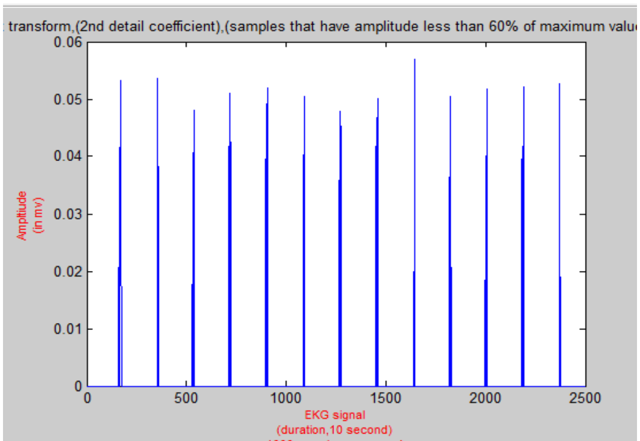


Figure 14. Detected R Peaks from the Second Detail Coefficient

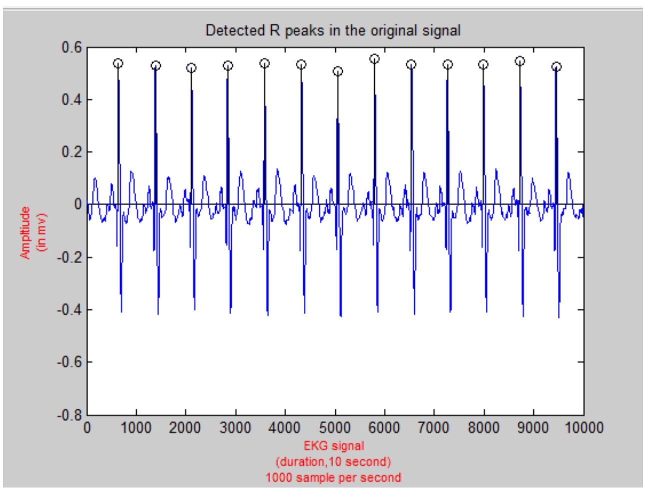


Figure 15. Detected R peaks in Original EKG Signal

Based on R peaks which are successfully detected, The P, Q, S, and T waves are being searched for with reference to R peaks locations. The number of beats per minute is calculated using the following formula:

Number of beats = R peaks * length of signal (number of samples) / (Fs *60 seconds)

Figure 16 shows the successfully detected R peaks with calculated heart rate per minute.

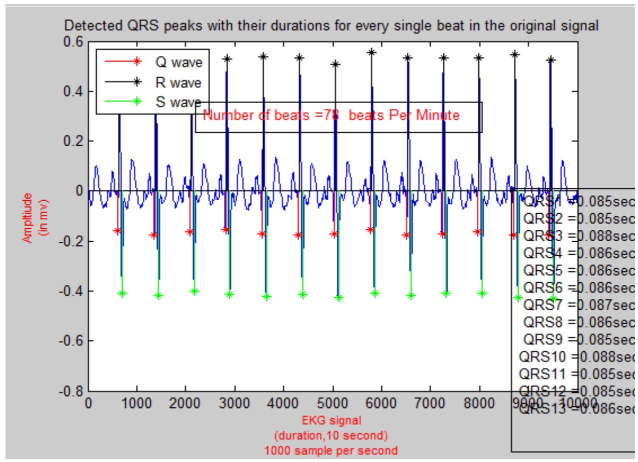


Figure 16. Calculated Heart Based on Detected R Peaks

Creating windows and searching within these windows is the method that is used to detect other waves. To detect P peaks location, window of 160 samples is created. This window extends from 200 samples to 60 samples to the left of each R peaks. Within these windows, P peaks are located at the samples that have the maximum amplitude value. The same manner, Q peaks are detected with reference to R peaks locations. Windows of 90 samples extend on ranges start 100 samples on the left side of R peaks location and end 10 samples away from R peaks locations. In these windows, the minimum amplitude values are the Q peaks. S peaks are detected the same way; yet, instead of searching on the right side of R peaks, the left side is searched through windows of 95 samples. These windows start 5 samples on the right of R peaks and end at 100 samples away from R peaks. In these windows, minimum amplitude values are the S peaks. Figure 17 shows detected QRS complex and the time of these complexes for every single beat in EKG signal. T waves are the farthest waves from R peaks. They are detected using windows of 300 samples of width. These windows start at 100 samples on the right of R peaks and end at 400 samples away from R peaks. In these windows the maximum amplitude value are the T waves.

Thus, all peaks are successfully detected. Figure 18 shows detected T waves.

However, there was a problem that urged through processing some EKG signals. Location of Windows that are created to detect waves peaks, in some signal, have some of negative values. Those negative values are used as arrays indices. However, Matlab does not allow for zero or negative values as indices. Therefore, some errors appeared with some EKG signals. To overcome this obstacle, the Matlab code that is developed, search the first peak of P wave and Q wave which are on the right side of R peak separately outside the main for loop function that is used for other peaks. If statement is set to measure the length of the first window. Then the minimum location is found. If the minimum location has zero or negative value, the window is narrowed down. Then, it is tested again. If it still has zero or negative value, it is narrowed down again and so forth.

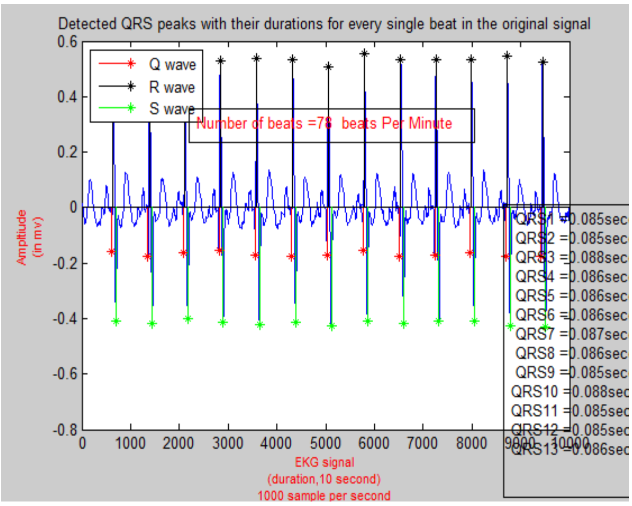


Figure 17. Detected QRS Complex with Its Time Duration for Every Single Beat

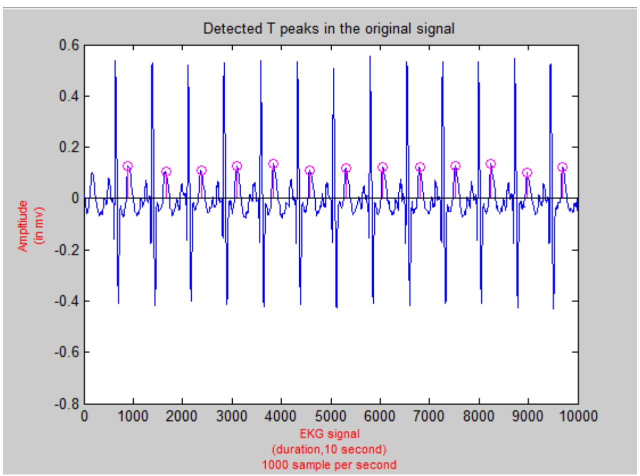


Figure 18. Detected T Waves in EKG Signal

This procedure is done iteratively till certain limit. If it still has zero or negative value, the window width is set to zero and consider the first peaks is not existing. As a matter of fact, this is true because in practical it is not guaranteed that the EKG signal which being recorded has the first P or Q peaks. in fact, it might be when the sensor is connected to the patient, the time of recording is exactly at the appearance of R peaks. In this case, the P or Q is not existing. The same method is used with S and T waves which are located on the right-hand side of R peaks. Nonetheless, instead of processing the first peak individually, the last peaks are processed separately and instead of checking for zero or negative indices, the indices that exceed the EKG signal length (10000 sample) is checked using find the maximum value command instead of minimum. This strict algorithm, searching windows that are always within the range. The algorithm and matlab code being smart and able to handle any EKG signal regardless of how and when the signals are recorded. After all, waves and peaks are found. The code measures the QRS complex duration, P wave duration and T wave duration. To have these waves duration measured, onset and offset for every single wave should be found. Using if condition nested inside for loop for every type of wave (P, QRS, T), the onset and offset are found. If condition checks for sample that proceed the sample which is bigger than zero and follow sample which is less than zero. The search for every peak duration starts from the peaks itself. For peaks onset, windows are created to the right of peaks. Within each window, the value that is satisfy the if statement condition is consider the wave onset. However, in some waves, sample that satisfy if condition is not unique. Hence, the first sample that satisfy if condition is considered and ignore other samples with help of break command. In fact, the closest sample to the wave peak is considered. That is, the first zero crossing sample is considered to be onset. The same way that offset for every single wave is measured; yet, instead of searching to the left side of the peak, the right side of the peak is the point of interest. Having all peaks of waves onset and offset measured, subtracting the offset from onset for every single wave result in the duration (in many samples) for every single wave. To convert the duration to be in time (seconds), the following formula is used:

Duration in samples = offset - onset

Duration in time = (Duration in samples * 1)/Fs

Fs is the sampling frequency. The same method is used for both P and T wave durations. Figure 19 and 20 show the P and T waves onset and offset detection and duration respectively.

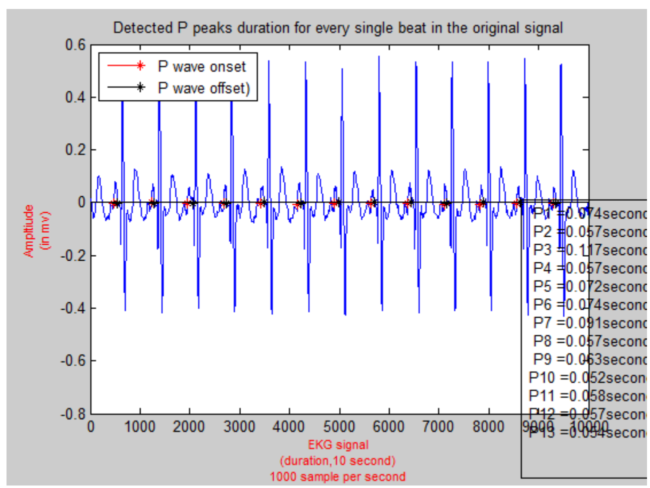


Figure 19. Detected P Waves and Their Duration

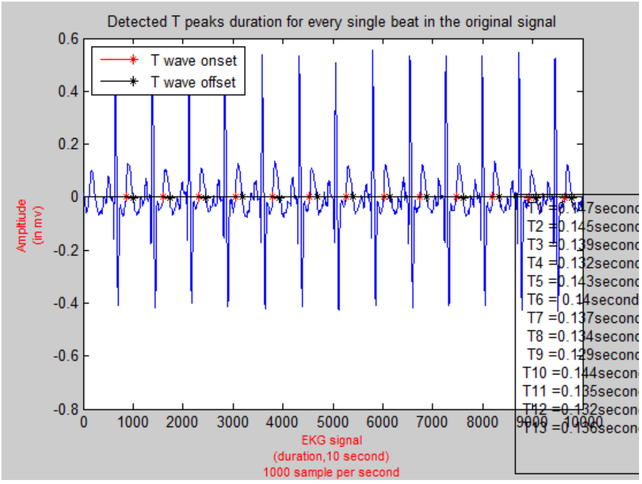


Figure 20. Detected T Waves and Their Duration

6. Conclusions

This algorithm has been tested on 108 EKG signals obtained from physionet website. Those signals are real signals collected from patients for research aim. The algorithm succeeded to detects all R peaks in each signal and all other important waves. In addition, the heart rate, the duration of P, QRS Complex, and T waves have been successfully and accurately calculated. The only wave that the algorithm fails to detect R peaks in it is the EKG obtained from the patient 146. This signal is extremely distorted. Table 2 shows the heart rate calculated for different patients.

Table 2. Measured Heart rate for Some Patient's EKG Signals'

No	Patient's EKG Signal	Detected Heart Rate
1	patient001_s0010_re	78
2	patient002_s0015_Ire	78
3	patient003_s0017_Ire	78
4	patient006_s0022_Ire	84
5	patient010_s0036_Ire	96
6	patient012_s0043_Ire	54
7	patient015_s0057_Ire	78
8	patient016_s0076_Ire	54
9	patient018_s0082_Ire	72
10	patient021_s0065_Ire	72
11	patient023_s0081_Ire	66
12	patient024_s0094_Ire	78
13	patient026_s0088_Ire	78
14	patient029_s0122_Ire	84
15	patient033_s0113_Ire	60
16	patient034_s0109_Ire	120
17	patient037_s0120_Ire	72
18	patient038_s0162_Ire	60
19	patient040_s0131_Ire	72
20	patient041_s0136_Ire	84
21	patient043_s0278_Ire	90

No	Patient's EKG Signal	Detected Heart Rate
22	patient045_s0147_Ire	108
23	patient046_s0156_Ire	102
24	patient048_s0180_Ire	72
25	patient049_s0173_Ire	78
26	patient051_s0213_Ire	66
27	patient053_s0191_Ire	78
28	patient055_s0194_Ire	108
29	patient056_s0196_Ire	66
30	patient059_s0208_Ire	90
31	patient060_s0209_Ire	84
32	patient062_s0212_Ire	72
33	patient063_s0214_Ire	72
34	patient065_s0226_Ire	66
35	patient066_s0225_Ire	78
36	patient068_s0228_Ire	84
37	patient070_s0235_Ire	78
38	patient071_s0236_Ire	66
39	patient073_s0238_Ire	66
40	patient074_s0406_Ire	72
41	patient075_s0242_Ire	84
41	patient077_s0254_Ire	78
42	patient078_s0259_Ire	72
43	patient079_s0269_Ire	66
44	patient080_s0260_Ire	102
45	patient084_s0281_Ire	78
46	patient084_s0281_Ire	66
47	patient085_s0345_Ire	66
48	patient088_s0339_Ire	78
49	patient090_s0356_Ire	54
50	patient091_s0353_Ire	84
51	patient097_s0380_Ire	84
52	patient099_s0388_Ire	84
53	patient100_s0399_Ire	114
54	patient101_s0400_Ire	108
55	patient105_s0303_Ire	72
56	patient106_s0030_re	78
57	patient108_s0013_re	78
58	patient111_s0203_re	84
59	patient115_s0023_re	60
60	patient118_s0183_re	84
61	patient121_s0311_Ire	84
62	patient125_s0006_re	84
63	patient127_s0342_Ire	72
64	patient130_s0166_re	54
65	patient131_s0273Ire	102
66	patient135_s0334Ire	72
67	patient138_s0005_re	90
68	patient141_s0307Ire	78
69	patient146_s0007_re	36 (not accurate)

No	Patient's EKG Signal	Detected Heart Rate
70	patient148_s0335lre	84
71	patient151_s0206_re	60
72	patient156_s0299lre	72
73	patient159_s0390lre	84
74	patient165_s0322lre	60
75	patient171_s0364lre	78
76	patient176_s0188_re	84
77	patient180_s0374lre	66
78	patient187_s0207_re	78
79	patient190_s0040_re	72
80	patient195_s0337lre	84
81	patient199_s0404lre	72
82	patient200_s0405lre	54
83	patient201_s0424_re	54
84	patient208_s0429_re	66
85	patient210_s0432_re	72
86	patient211_s0433_re	108
87	patient214_s0436_re	96
88	patient217_s0439_re	66
89	patient219_s0441_re	174
90	patient221_s0443_re	60
91	patient223_s0446_re	66
92	patient227_s0450_re	84
93	patient229_s0453_re	54
94	patient230_s0454_re	60
95	patient233_s0483_re	60
96	patient243_s0472_re	90
97	patient245_s0480_re	60
98	patient249_s0484_re	60
99	patient250_s0485_re	54
100	patient255_s0491_re	66
101	patient258_s0494_re	60
102	patient260_s0496_re	60
103	patient266_s0502_re	72
104	patient270_s0507_re	72
105	patient273_s0511_re	42
106	patient280_s0535_re	60
107	patient291_s0554_re	84
108	patient291_s0554_re	72

6.1. Enhancement Work

This algorithm search for peaks in a designed window and assume that the waves (P, QRS complex, and T) are always existing. It does not take into consideration that the disappearance of some waves is possible due to any kind of disease or even dysfunctionality of hardware devises. This algorithm can be smarter by applying another threshold for peaks amplitude so that even if the peak is found, it does not consider as a peak till satisfy the threshold condition. In addition, the PQ segment and ST segment is not calculated which can be easily detected using same window method.

Appendix A

Matlab code that is used to de-noise EKG signal and extract the various features.

```
[filename, pathname] = uigetfile({'*.*','All Files' },'ECG Signal',...
        'C:\Users\engsa\Desktop\Data Aquisition Analysis and Display Spring 2016\The Research Papers\DataInTextFormat\all signals')
if isequal(filename,0)
   disp('User selected Cancel')
else
   disp(['User selected ', fullfile(pathname, filename)])
end
xx= strcat(pathname,filename);
x = load(xx);

Fs=1000;
N=10000;
length(x);
size(x);
n = 0:1:N-1;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
s1 = x(:,2);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
figure(1);
plot(n,s1)
title('Row EKG signal')
xlabel({'EKG signal','(duration,10 second)','1000 sample per second'},...
    'FontSize',8,'Color','r')
ylabel({'Ampltiude','(in mv)'},'FontSize',8,'Color',...
    'r')
NFFT = 2^nextpow2(length(s1));
Y = fft(s1,NFFT)/length(s1);
f = Fs/2*linspace(0,1,NFFT/2+1);

% Plot single-sided amplitude spectrum.
figure(2)
plot(f,2*abs(Y(1:NFFT/2+1)))
title('Single-Sided Amplitude Spectrum of s1(t)')
xlabel('Frequency (Hz)','FontSize',8,'Color','r')
ylabel('|s1(f)|','FontSize',8,'Color','r')
%*****************************************************
d= fdesign.highpass('N,Fst,Fp,Ast',2746,1,2,80,1000);
```

```
Hd =design(d,'equiripple');

s = coeffs(Hd);
s.Numerator;
sf = filtfilt(s.Numerator,1,s1);
figure(3)
plot(sf)

title('Denoised EKG signal,(Eleminated baseline noise)')
xlabel({'EKG signal','(duration,10 second)','1000 sample
per second'},...
    'FontSize',8,'Color','r')
ylabel({'Ampltiude','(in mv)'},'FontSize',8,'Color',...
    'r')
dL= fdesign.lowpass('N,Fp,Fst,Ap',506,40,45,1,1000);
Ld =design(dL,'equiripple');
%fvtool(Ld);
%
sLd = coeffs(Ld);
sLd.Numerator;
sL = filtfilt(sLd.Numerator,1,sf);
figure(4)
plot(sL)

title('Denoised EKG signal,(Eleminated power interference
noise (Hz)')
xlabel({'EKG signal','(duration,10 second)','1000 sample
per second'},...
    'FontSize',8,'Color','r')
ylabel({'Ampltiude','(in mv)'},'FontSize',8,'Color',...
    'r')

signal = sL;
[c l] = wavedec(signal,9,'haar');
c;
l,

ca1=appcoef(c,l,'haar',1);
figure(5)
plot(ca1)

title('Haar wavelet transform,(recounstructed signal from
approaximation coefficient)')
xlabel({'EKG signal','(duration,10 second)','1000 sample
per second'},...
    'FontSize',8,'Color','r')
ylabel({'Ampltiude','(in mv)'},'FontSize',8,'Color',...
    'r')

[cd1,cd2,cd3,cd4,cd5,cd6,cd7,cd8,cd9] = detcoef(c,l,[1 2 3
4 5 6 7 8 9]);
figure(6)
plot(cd4)

title('Haar wavelet transform,(4th detail coefficient')
xlabel({'EKG signal','(duration,10 second)','1000 sample
per second'},...
    'FontSize',8,'Color','r')
ylabel({'Ampltiude','(in mv)'},'FontSize',8,'Color',...
    'r')


figure(7)
stem(cd2,'ro')

title('Haar wavelet transform,(2nd detail coefficient')
xlabel({'EKG signal','(duration,10 second)','1000 sample
per second'},...
    'FontSize',8,'Color','r')
ylabel({'Ampltiude','(in mv)'},'FontSize',8,'Color',...
    'r')

figure(8)
plot(cd2)

title('Haar wavelet transform,(2nd detail coefficient')
xlabel({'EKG signal','(duration,10 second)','1000 sample
per second'},...
    'FontSize',8,'Color','r')
ylabel({'Ampltiude','(in mv)'},'FontSize',8,'Color',...
    'r')

rrec = waverec(c,l,'haar');
mx = max(cd2);
thresholdmx = 0.60*mx;
cd2(cd2<thresholdmx)= 0;
figure(9)
cd2;
plot(cd2)

title('Haar wavelet transform,(2nd detail
coefficient),(samples that have amplitude less than 60% of
maximum value are eleminated ')
xlabel({'EKG signal','(duration,10 second)','1000 sample
per second'},...
    'FontSize',8,'Color','r')
ylabel({'Ampltiude','(in mv)'},'FontSize',8,'Color',...
    'r')

rec = upcoef('d',cd2,'haar');
% T = waverec(c,l,'haar')
figure(10)
plot(rec)
```

```
title('Reconstructed EKG Signal from the haar wavelet snd
Detail coefficient' )
xlabel({'EKG signal','(duration,10 second)','1000 sample
per second'},...
    'FontSize',8,'Color','r')
ylabel({'Ampltiude','(in mv)'},'FontSize',8,'Color',...
    'r')

mx = max(cd2);
thresholdmx = 0.60*mx;
P = find(cd2 >= thresholdmx);
P1= P;
P2=[];
last=P1(1);
P2=[P2 last];
for(i=2:1:length(P1))
    if(P1(i)>(last+50))
        % In this step we find R peaks which are atleast
50 samples apart
        last=P1(i);
        P2=[P2 last];
        RP = length(P2)
    end
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%
l = length (cd2)*4;
sduration = l/1000;
mduration = sduration/60;
beats_in_minutes = round((RP/mduration))


P3=P2*4;
Rloc=[];
%%^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
^^^^^^^^^^^^^^^^^^^^^^^^^^
%%(R)(R)(R)(R)()()()()()()()()()()()()()()()()()()()()()()()()(
)()()()
%%
rangefirst = [P3(1)-50:P3(1)+50];
rngmn = min(rangefirst);
if(rngmn<=0)
rangefirst = [P3(1)-30:P3(1)+50];
    if(rngmn<=0)
        rangefirst = [P3(1)-10:P3(1)+50];
        if(rngmn<=0)
        rangefirst =[P3(1)-0:P3(1)+50];
        end
    end
end

    m1=max(sL(rangefirst));
    l1=find(sL(rangefirst)==m1);
    pos1=rangefirst(l1);
    Rloc1= pos1;

%!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!
rangelast = [P3(end)-50:P3(end)+50];
rngmx = max(rangelast);
if(rngmx>=10000)
rangelast = [P3(1)-50:P3(1)+30];
    if(rngmx>=10000)
        rangelast = [P3(1)-50:P3(1)+10];
        if(rngmn>=10000)
        range1ast=[P3(1)-50:P3(1)+0];
        end
    end
end

    mlast = max(sL(rangelast));
    llast=find(sL(rangelast)==mlast);
    poslast=rangelast(llast);
    Rlocend= poslast;

%%^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
^^^^^^^^^^^^^^^^^^^^^^^^^^
%%(R)(R)(R)(R)()()()()()()()()()()()()()()()()()()()()()()()()(
)()()()

for( i=2:1:length(P3)-1)
    range = [P3(i)-50:P3(i)+50];

    % Search within a window of +-10 samples in the
original signal with
    % reference to up scaled R locations detected in
downsampled signal.
    %%^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
^^^^^^^^^^^^^^^^^^^^^^^^^^

    m=max(sL(range));
    l=find(sL(range)==m);
    pos=range(l);
    Rloc=[Rloc pos];
end
Rloc = [Rloc1 Rloc Rlocend] ;
Ramp=sL(Rloc);
figure(11)
plot(sL)
hold on
stem (Rloc,Ramp,'k')

title('Detected R peaks in the original signal')
```

```
xlabel({'EKG signal','(duration,10 second)','1000 sample
per second'},...
       'FontSize',8,'Color','r')
ylabel({'Ampltiude','(in mv)'},'FontSize',8,'Color',...
       'r')
%*********************************
% QRS onset - offset
%*********************************
%%%window

% Firstly, If you observe the waveform, it will be very clear
that from
% R location if you select a window of Rloc? 200 to
% Rloc?60 and find the maximum, than that maxima is P
peak.

X = Rloc;
y = sL;
%%&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
ap1=Rloc(1)-200:Rloc(1)-60;
       n0= min(ap1);
       if (n0<0)
              ap1=Rloc(1)-150:Rloc(1)-60;
              n0= min(ap1);
              if (n0<0)
                     ap1=Rloc(1)-100:Rloc(1)-60;
                     n0=min(ap1);
                     if(n0<0)
                            ap1=Rloc(1)-80:Rloc(1)-60;
                            n0 =min(ap1);
                            if(n0<0)
                                   ap1=0;
                            end
                     end
              end
       end
       if (ap1 ~=0)
              m1 = max(y(ap1));
              b1 = find(y(ap1)==m1);
              b1=b1(1);
              b1=ap1(b1);
              Ploc(1)= b1;
              Pamp(1)= m1;
       end

%%*********************************************
*************************
for(j=2:1:length(Rloc))
       ap = Rloc(j)-200:Rloc(j)-60;
       mp = max(y(ap));
       bp = find(y(ap)== mp);
       bp=bp(1);
       bp=ap(bp);
       Ploc(j)= bp;
       Pamp(j)= mp;
       figure(12);
       plot(sL)
       hold on
       stem(Ploc,Pamp,'ro')

title('Detected P peaks in the original signal')

xlabel({'EKG signal','(duration,10 second)','1000 sample
per second'},...
       'FontSize',8,'Color','r')
ylabel({'Ampltiude','(in mv)'},'FontSize',8,'Color',...
       'r')

       hold off
end
%%-----------------------------------------------------------------
-----%
% P duration
% y is my signal
%%++++++++++++++++++++++++++++++++++++++
++++++++++++++++++++++++++++++++
       pon1 = Ploc(1):-1:Ploc(1)-150;
       n0= min(pon1);
       if (n0<0)
              pon1= Ploc(1):-1:Ploc(1)-100;
              n0= min(pon1);
              if (n0<0)
                     pon1=Ploc(1):-1:Ploc(1)-70;
                     n0=min(pon1);
                     if(n0<0)
                            pon1=Ploc(1):-1:Ploc(1)-30;
                            n0 =min(pon1);
                            if(n0<0)
                            pon1=0;
                            end
                     end
              end
       end

if (pon1 ~=0)
       for k = 1:1:length(pon1)-1
              f1 = 0
              if (y(pon1(k))>0 & y(pon1(k+1))<=0 )
                     g1 = pon1(k+1);
                     found1=g1;
                     pdon(1)= found1;
                     pdon_amp(1)= y(pdon(1));
                     f1 = 1;
```

```
            if (f1==1)
                break
            end
            if (f1~=1)
                g1 = min(pon1);
                found1=g1;
                pdon(1)= found1;
                pdon_amp(1)= y(pdon(1));
                f1 = 1;
            end

        end
    end
end



%%++++++++++++++++++++++++++++++++++++++++
++++++++++++++++++++++++++++++++++
for i = 2:1:length(Ploc);
    pon = Ploc(i):-1:Ploc(i)-150;
    f = 0
    for k = 1:1:length(pon)-1
        if (y(pon(k))>0 & y(pon(k+1))<=0 )
            g = pon(k+1);
            found =    g;
            found = found(1);
            pon(i) = found;
            f = 1;

            if (f==1)
                break
            end
            if (f~=1)
                g = min(pon);
                found =    g;
                found = found(1);
                pon(i) = found;
                f = 1;
            end

        end
    end

    pdon(i) = pon(i);
    pdon_amp(i) = y(pdon(i));


end
figure(13)
plot(y)
hold on
stem(pdon,pdon_amp,'r*')

title('Detected P peaks onset in the original signal')

xlabel({'EKG signal','(duration,10 second)','1000 sample
per second'},...
    'FontSize',8,'Color','r')
ylabel({'Ampltiude','(in mv)'},'FontSize',8,'Color',...
    'r')

hold off
%%##############################################
###########################
%% P offset
%% to make sure that first P peak is there (to avoiding error
due to that
%% may some signal the first P peak is not existing

if (pon1 ~=0)
    poff1 = Ploc(1):1:Ploc(1)+150;

    for k = 2:1:length(poff1)
        foff1 = 0;
        if (y(poff1(k))<0 & y(poff1(k-1))>=0 )
            goff1 = poff1(k);
            foundoff1=goff1;
            pdoff1(1)= foundoff1;
            pdoff(1) = pdoff1(1);
            pdoff_amp(1) = y(pdoff(1));
            foff1 = 1;
            if (foff1==1)
                break
            end
            if (foff1~=1)
                goff1 = max(poff1);
                foundoff1=goff1;
                pdoff1(1)= foundoff1;
                pdoff(1) = pdoff1(1);
                pdoff_amp(1) = y(pdoff(1));
                foff1 = 1;
            end
        end
    end
end

%%##############################################
###########################

for i = 2:1:length(Ploc);
    poff = Ploc(i):1:Ploc(i)+150;
    for k = 2:1:length(pon)
        foff = 0;
```

```
            if (y(poff(k))<0 & y(poff(k-1))>=0 )
                g = poff(k);
                found =    g;
                found = found(1);
                poff(i) = found;
                foff = 1;

                if (foff==1)
                    break
                end
                if (foff~=1)
                    g = max(poff);
                    found =    g;
                    found = found(1);
                    poff(i) = found;
                    foff = 1;
                end

            end
        end

        pdoff(i) = poff(i);
        pdoff_amp(i) = y(pdoff(i));


end
figure(14)
plot(y)
hold on
stem(pdoff,pdoff_amp,'r*')

title('Detected P peaks offset in the original signal')

xlabel({'EKG signal','(duration,10 second)','1000 sample
per second'},...
    'FontSize',8,'Color','r')
ylabel({'Ampltiude','(in mv)'},'FontSize',8,'Color',...
    'r')

hold off

%%!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!
% P duration
pdurationsamples = pdoff - pdon;
pduration = (pdurationsamples * 1)/Fs;
pduration = pduration';

%%!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!
figure(15)
plot(y)
hold on

%%PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
ponset = stem(pdon,pdon_amp,'r*');
% text (pdon(1),pdon_amp(1),'P Onset','FontSize',8)
poffset = stem(pdoff,pdoff_amp,'k*');
% text (pdoff(2),pdoff_amp(2),'P Offset','FontSize',8)




legend([ponset,poffset],'P wave onset','P wave
offset)','location',...
    'northwest')
%% Annotation Preperation
pppduration = num2str(pduration);
DDp = length(pppduration);
P = cellstr(num2str((1:DDp)','P%d'))
outputp = strcat(P,' = ',' ',pppduration,'seconds')
dim = [.8 .5 .1 .1];
% ann =
annotation('textbox',dim,'String',output,'FitBoxToText','on');

annpp =
annotation('textbox',dim,'String',outputp,'FitBoxToText','on'
);


% ylim=get(gca,'ylim');
% xlim=get(gca,'xlim');
% text(xlim(end)-100,ylim(1),str)

%% Annotation Preperation


title('Detected P peaks duration for every single beat in the
original signal')

xlabel({'EKG signal','(duration,10 second)','1000 sample
per second'},...
    'FontSize',8,'Color','r')
ylabel({'Ampltiude','(in mv)'},'FontSize',8,'Color',...
    'r')


hold off

%%PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
hold off
```

```
%% Q Detection
%%*********************************************
***************************
aq1=Rloc(1)-100:Rloc(1)-10;
    n0= min(aq1);
    if (n0<0)
        aq1=Rloc(1)-80:Rloc(1)-10;
        n0= min(aq1);
        if (n0<0)
            aq1=Rloc(1)-60:Rloc(1)-10;
            n0=min(aq1);
            if(n0<0)
                aq1=Rloc(1)-40:Rloc(1)-10;
                n0 = min(aq1);
                if(n0<0)
                    aq1=0;
                end
            end
        end
    end
    if(aq1~=0)
        mq1 = min(y(aq1));
        bq1 = find(y(aq1)==mq1);
        bq1 =aq1(bq1);
        Qloc(1)= bq1;
        Qamp(1)= mq1 ;
    end

%%*********************************************
***************************
for(j=2:1:length(X))
    aq = Rloc(j)-100:Rloc(j)-10;
    mq = min(y(aq));
    bq = find(y(aq)== mq);
    bq = bq(1);
    bq =aq(bq);
    Qloc(j)= bq;
    Qamp(j)= mq ;
    figure(16);
    plot(sL)
    hold on
    stem(Qloc,Qamp,'go')

title('Detected Q peaks in the original signal')

xlabel({'EKG signal','(duration,10 second)','1000 sample
per second'},...
    'FontSize',8,'Color','r')
ylabel({'Ampltiude','(in mv)'},'FontSize',8,'Color',...
    'r')

    hold off
end

%*********************************************
***************************%
%% S Detection
%%*********************************************
***************************
%%SSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSS
SSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSS

aslast=Rloc(length(Rloc))+5:Rloc(length(Rloc))+100;
    n0= max(aslast);
    if (n0>10000)

aslast=Rloc(length(Rloc))+5:Rloc(length(Rloc))+80;
        n0= max(aslast);
        if (n0>10000)

aslast=Rloc(length(Rloc))+5:Rloc(length(Rloc))+60;
            n0=max(aslast);
            if(n0>10000)

aslast=Rloc(length(Rloc))+5:Rloc(length(Rloc))+40;
                n0 = max(aslast);
                if(n0>10000)
                    aslast=0;
                end
            end
        end
    end
    if(aslast~=0)
        mslast = min(y(aslast));
        bslast = find(y(aslast)==mslast);
        bslast =aslast(bslast);
        Sloc(length(Rloc))= bslast;
        Samp(length(Rloc))= mslast;
    end
%%SSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSS
SSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSS
for(j=1:1:length(Rloc)-1)
    sss= 0;
    as = Rloc(j)+5:Rloc(j)+100;
    mins = min(as);
    maxs = max(as);
    if (mins < 0) || (maxs>10000)
        as = Rloc(j)+0:Rloc(j)+80;
        if (mins< 0) || (maxs>10000)
            as = Rloc(j)+0:Rloc(j)+50;
            if (mins< 0) || (maxs>10000)
                sss = 1;
            end
        end
```

```
        end
        if (sss~=1)
            mnn = min(y(as));
            bs =find(y(as)== mnn);
            bs = bs(1);
            bs = as(bs);
            Sloc(j)= bs;
        end
        Samp(j)= mnn;
        figure(17);
        plot(sL)
        hold on
        stem(Sloc,Samp,'ko')

        title('Detected S peaks in the original signal')

xlabel({'EKG signal','(duration,10 second)','1000 sample per second'},...
        'FontSize',8,'Color','r')
ylabel({'Ampltiude','(in mv)'},'FontSize',8,'Color',...
        'r')

        hold off
end

%% QRS COMPLEX QRS COMPLEX QRS COMPLEX QRS COMPLEX QRS COMPLEX QRS COMPLEX
%%
%%                          QRS Duration
if (length(Qloc) > length(Sloc))
        Qloc(1)=[];
        Qamp(1) = [];
elseif(length(Sloc) > length(Qloc))
        Sloc(end)=[];
        Samp(end) = [];
end
qrsdurationsamples = Sloc - Qloc;

qrsduration = (qrsdurationsamples * 1)/Fs;
qrsduration = qrsduration';
%!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
if(aq1~=0)
        Qloc = [bq1 Qloc];
        Qamp = [mq1 Qamp];
end
if(aslast~=0)
        Sloc = [Sloc bslast];
        Samp = [Samp mslast];
end
figure(18)
plot(y)
hold on
Qwave = stem(Qloc,Qamp,'r*');

Rwave = stem(Rloc,Ramp,'k*');

Swave = stem(Sloc,Samp,'g*');

legend([Qwave,Rwave,Swave],'Q wave ','R wave ','S wave','location',...
        'northwest');
%% Annotation Preperation
duration = num2str(qrsduration);
DDqrs = length(duration);
QRS = cellstr(num2str((1:DDqrs)','QRS%d'));
qrsoutput = strcat(QRS,' = ',' ',duration,'seconds')
dim = [.8 .5 .1 .1];
qrsann =
annotation('textbox',dim,'String',qrsoutput,'FitBoxToText','on');
%%-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-
beats = beats_in_minutes % is it calculated before
Number_of_beats = num2str(beats);
beatsoutput = strcat('Number of beats',' = ',Number_of_beats ,'    beats'...
        ,' Per Minute');
dim2 = [.3 .64 .25 .15];
qrsann3 = annotation('textbox',dim2,'string',beatsoutput,...
        'FitBoxToText',...
        'on','Color','red');

title('Detected QRS peaks with their durations for every single beat in the original signal')

xlabel({'EKG signal','(duration,10 second)','1000 sample per second'},...
        'FontSize',8,'Color','r')
ylabel({'Ampltiude','(in mv)'},'FontSize',8,'Color',...
        'r')


hold off



%% QRS COMPLEX QRS COMPLEX QRS COMPLEX QRS COMPLEX QRS COMPLEX QRS COMPLEX


%@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
%% T peaks
```

```
%%&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
&&&&&&&&&&&&&&
atlast=Rloc(length(Rloc))+100:Rloc(length(Rloc))+400;
    n0= max(atlast);
    if (n0>10000)

atlast=Rloc(length(Rloc))+100:Rloc(length(Rloc))+350;
        n0= max(atlast);
        if (n0>10000)

atlast=Rloc(length(Rloc))+100:Rloc(length(Rloc))+300;
            n0=max(atlast);
            if(n0>10000)

atlast=Rloc(length(Rloc))+100:Rloc(length(Rloc))+250;
                n0 =max(atlast);
                if(n0>10000)

atlast=Rloc(length(Rloc))+100:Rloc(length(Rloc))+200;
                    n0 =max(atlast);
                    if(n0>10000)
                        atlast = 0;
                    end

                end
            end
        end
    end
    if (atlast ~=0)
        mxtlast = max(y(atlast));
        btlast = find(y(atlast)==mxtlast);
        btlast = atlast(btlast);
        Tloc(length(Rloc))=btlast;
        Tamp(length(Rloc))=mxtlast;

    end

%%&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
&&&&&&&&&&&&&&
%% T Peak
for(j=1:1:length(Rloc)-1)
    at = Rloc(j)+100:Rloc(j)+400;
    mxt = max(y(at));
    bt = find(y(at)== mxt);
    bt = bt(1);
    bt = at(bt);
    Tloc(j)=bt;
    Tamp(j)=mxt;
    figure(19);
    plot(sL)
    hold on
    stem(Tloc,Tamp,'mo')

    title('Detected T peaks in the original signal')

xlabel({'EKG signal','(duration,10 second)','1000 sample
per second'},...
    'FontSize',8,'Color','r')
ylabel({'Ampltiude','(in mv)'},'FontSize',8,'Color',...
    'r')

hold off
end

%%
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~~~~~~~~~
%% T duration
%%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
ftlast= 0;
ft = 0;
%%+++++++++++++++++++++++++++++++++++++++
++++++++++++++++++++++++++++++++++
%% T offset last peaks
%% to make sure that last T peak is there (to avoiding error
due to that
%% may some signal the last T peak is not existing
%%+++++++++++++++++++++++++++++++++++++++
++++++++++++++++++++++++++++++++++
tofflast    = 0
if (atlast ~=0)
    tofflast =
Tloc(length(Tloc)):1:Tloc(length(Tloc))+150;
    n0= max(tofflast);
    if (n0>10000)
        tofflast =
Tloc(length(Tloc)):1:Tloc(length(Tloc))+120;
        n0= max(tofflast);
        if (n0>10000)
            tofflast =
Tloc(length(Tloc)):1:Tloc(length(Tloc))+100;
            n0=max(tofflast);
            if(n0>10000)
                tofflast =
Tloc(length(Tloc)):1:Tloc(length(Tloc))+80;
                n0 =max(tofflast);
                if(n0>10000)
                    tofflast =
Tloc(length(Tloc)):1:Tloc(length(Tloc))+60;
                    if(n0>10000)
                        tofflast=1;
```

```
                    end

                end
            end
        end
    end
end

if (tofflast ~=1)
    for k = 1:1:length(tofflast)-1
        if (y(tofflast(k))>0 & y(tofflast(k+1))<=0 )
            gtofflast = tofflast(k+1);
            foundofflast = gtofflast;
            tdoff(length(Tloc))= foundofflast;
            tdoff_amp(length(Tloc))=
y(tdoff(length(Tloc)));
            ftlast = 1;
            if (ftlast==1)
                break
            end
        end
    end
end

%%-----------------------------------------------------------------
------
for i = 1:1:length(Tloc)-1;
    ft = 0;
    toff = Tloc(i):1:Tloc(i)+300;
    for k = 1:1:length(toff)-1
        if (y(toff(k))>0 & y(toff(k+1))<=0 )
            gtoff = toff(k+1);
            foundtoff =    gtoff;
            foundtoff = foundtoff(1);
            toff(i)= foundtoff;
            tdoff(i) = toff(i);
            ft = 1;

            if (ft==1)
                break
            end
        end

    end
end

tdoff(i) = toff(i)
tdoff_amp(i) = y(tdoff(i));
%%-----------------------------------------------------------------
------

figure(20)
plot(y)
hold on
stem(tdoff,tdoff_amp,'k*')

title('Detected T peaks offset for every single beat in the
original signal')

xlabel({'EKG signal','(duration,10 second)','1000 sample
per second'},...
    'FontSize',8,'Color','r')
ylabel({'Ampltiude','(in mv)'},'FontSize',8,'Color',...
    'r')

hold off
%%###############################################
############################
%% T onset
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%

%%###############################################
############################
%%FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF

if (tofflast ~=0)
    tonlast =
Tloc(length(Tloc)):-1:Tloc(length(Tloc))-150;

    for k = 1:1:length(tonlast)-1
        ftonlast = 0;
        if (y(tonlast(k))>0 & y(tonlast(k+1))<=0 )
            gtonlast = tonlast(k+1);
            foundonlast=gtonlast;
            tdon(length(Tloc))= foundonlast;
            tdon_amp(length(Tloc)) =
y(tdon(length(Tloc)));
            ftonlast = 1;
            if (ftonlast==1)
                break
            end
        end
    end
end

%%FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF

for i = 1:1:length(Tloc)-1;
    fton = 0;
    toni = Tloc(i):-1:Tloc(i)-150;
```

```
    for k = 1:1:length(toni)-1
        if (y(toni(k))>0 & y(toni(k+1))<=0 )
            gton = toni(k+1);
            foundton =    gton;
            foundton = foundton(1);
            ton(i)= foundton;
            tdon(i) = ton(i);
            fton = 1;

            if (fton==1)
                break
            end
        elseif (fton~=1)
            gton = min(toni);
            foundton =    gton;
            foundton = foundton(1);
            ton(i)= foundton;
            tdon(i) = ton(i);
            fton = 1;
        end
    end
end

    tdon(i) = ton(i);
    tdon_amp(i) = y(tdon(i));

figure(21)
plot(y)
hold on
stem(tdon,tdon_amp,'r*')

title('Detected T peaks onset for every single beat in the
original signal')

xlabel({'EKG signal','(duration,10 second)','1000 sample
per second'},...
    'FontSize',8,'Color','r')
ylabel({'Ampltiude','(in mv)'},'FontSize',8,'Color',...
    'r')

hold off

%%!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!
% T duration


%%!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!

%% T DURATION T DURATION T DURATION T
DURATION T DURATION T DURATION
%% Subtracting T offset from T onset should give how
many samples in T
%% duration. Then, by manipolation numbers, time
duration should be a
%% result. ex. if # of samples 150, the T wave duration is
Tduration =
%% (150*1)/1000 = 0.15 seconds.
%% T DURATION T DURATION T DURATION T
DURATION T DURATION T DURATION

%!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!

if (length(tdon) > length(tdoff))
    tdon(end)=[];
    tdon_amp(end) = [];
end
%!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!
tdurationsamples = tdoff - tdon;
tduration = (tdurationsamples * 1)/Fs;
tduration = tduration';

figure(22)
plot(y)
hold on
tonset = stem(tdon,tdon_amp,'r*');

toffset = stem(tdoff,tdoff_amp,'k*');

legend([tonset,toffset],'T wave onset','T wave
offset','location',...
    'northwest')
%% Annotation Preperation
duration = num2str(tduration);
DD = length(duration);
T = cellstr(num2str((1:DD)','T%d'))
output = strcat(T,' = ',' ',duration,'seconds')
dim = [.8 .5 .1 .1];

ann =
annotation('textbox',dim,'String',output,'FitBoxToText','on');
```

REFERENCES

[1] S. Z. Mahmoodabadi, A. Ahmadian, and M. D. Abolhasani, "ECG Feature Extraction using Daubechies Wavelets," in *Proceedings of the fifth IASTED International conference on Visualization, Imaging and Image Processing*, 2005.

[2] R. J. Huzar, "Basic Dysrhythmias, Interpretation and Management," 1988.

[3] G. K. Singh, A. Sharma, S. Velusami, "A Research Review on Analysis and Interpretation of Arrhythmias using ECG Signals," *'ISSN: 0974-5343 IJMST,* pp. pp::37-55, 2009.

[4] Vijay K. Madisetti , Douglas B. Williams, Digital Signal Processing Handbook, Boca Raton: CRC Press LLC, 1999.

[5] John G. Proakis, Dimitris G. Manolakis, "Digital Signal Processing"," Principles, Algorithms, and Applications", Third Edition ed., PRENTICE-HALL INTERNATIONAL, INC., 1996.

[6] Yurong Luo, Rosalyn H. Hargraves, Ou Bai, Kevin R. Ward, "A Hierarchical Method for Removal of Baseline Drift from Biomedical Signals Application in ECG Analysis," *The Scientific World Journal,* vol. Volume 2013, p. 1:2, 2013.

[7] Radomir S. Stankovic, Bogdan J. Falkowski, "The Haar wavelet transform its status and achievements," p. 25:44, 2003.

[8] Albert Boggess, Francis J. Narcowich, A first course in wavelets with fourier analysis, 6th Edition ed., U. S. River, Ed., Prentice Hall, 2001, p. 155:221.

[9] Michael F. Cohen, Tony D. DeRose, Alain Fournier, Michael Lounsbery, Leena-Maija Reissell, Peter Schroder, Wim Sweldens, Wavelets and their Applications in Computer Graphics.