Available online at sjuoz.uoz.edu.krd

journals.uoz.edu.krd

Science Journal of University of Zakho

Vol. 7, No. 4, pp. 179–183, December.-2019

p-ISSN: 2663-628X
e-ISSN: 2663-6298

DEVELOP A PATTERN ALGORITHM TO CONSTRUCT RESPIRATION SIGNAL USING ECG COMPONENTS

Dilshad Hassan Sallo [a]

[a] Dept. of Computer Science, College of Science, University of Duhok, Kurdistan Region-Iraq (dilshad.sallo@uod.ac)

Received: Jun., 2019 / Accepted: Nov., 2019 / Published: Dec.,2019 ***https://doi.org/10.25271/sjuoz.2019.7.3.612***

ABSTRACT:

The aim of this paper is designing an algorithm dubbed "pattern" to detect electrocardiogram (ECG) components accurately by searching exactly in the right places of peaks and getting exhaustive information related to the heart. Then, using the obtained results to propose a method for constructing respiration signal properly, by calculating the mean of R peaks to determine inspiration and expiration phases and calculating the amount of change for other peaks to be added during inspiration phase and subtracted during the expiration phase. The proposed method improves envelope method which only depends on the size of R to construct respiration signals. The results show that the pattern algorithm is guaranteed method and useful for detecting ECG components and exploiting them for constructing respiration signal work better than envelope method.

KEYWORDS: Breathing rate, ECG components, Respiration signal, pattern algorithm, ECG-Derived Respiration (EDR).

1. INTRODUCTION

The human body contains various biosignals that reflect the activities of parts. ECG and respiration signal are the most significant biosignals due to giving obvious information related to the mechanism of working. ECG signals give prominence information when are analysed by biomedical students properly using Matlab language. Unfortunately, Matlab does not provide built in functions to detect all peaks together simultaneously. Correspondingly, respiration signal is used to detect sleep apnea and sleep quality estimation, and fortunately it can be obtained indirectly, without extra sensor by constructing respiration from ECG signals. Various methods have been investigated to construct respiration indirectly. One of them called the envelope method, which used size of R to build breathing signal (Kim, Hong, Kim, Cha & Lee, 2006). The purpose of this paper is to develop a pattern algorithm to detect ECG signal components properly, and get holistic information related to the cardiac muscle. Furthermore, the obtained results of algorithm are used to construct a respiration signal and respiration rate. The proposed method improves the result of envelope method by calculating differences not only to R peaks, but also in PQST peaks. Then adding them to R peaks in inspiration phase or subtracting them in expiration phase.

2. RELATED WORKS

Over the last decades, various methods have been conducted to construct a respiration signal from ECG signals. In somehow, all of them have used R peak to detect respiration signal as it is a prominent peak in ECG signals. However, R peak is not the only peak that affected by breathing activity; the other peaks are also affected with little change comparing to R peak. In this section, the most relevant works to suggested method are briefly presented with specifying their notable drawbacks. Authors (Kim, Hong, Kim, Cha & Lee, 2006) proposed two methods to extract the respiration signal; one using R-R interval which calculates time difference between two successive R peaks, and other using size of R peak. The limitation of these two methods is only R peaks used for constructing breathing signal and other peaks are neglected. Authors (Ruangsuwana, Velikic & Bocko, 2010) suggested method to construct respiration signal by determining ECG mean based on R-R interval and size of R peak. The drawback of suggested method is getting a result similar to R-R interval and it is only focused on R peak in ECG signals. Authors (Sharma & Sharma, 2018) proposed method to construct respiration signal based on the principal component analysis (PCA), R-peak amplitudes (RPA), respiratory sinus arrhythmia (RSA), slopes of the QRS complex, and R-wave angle. The major issue of this method is not exploited all ECG components to get the best result. Authors (Park, Noh, Park & Yoon, 2008) introduced a method to construct respiration using QRS complex in ECG signals which represents the depolarization of the ventricles. This method neglects P and T waves that consider the second and the third highest peaks in ECG signal. Authors (Sarkar, Bhattacherjee & Pal, 2015) proposed method to generate breathing signals from both Peak Amplitude Variation (PAV) and Heart Rate Variability (HRV) together. However, it uses only R peak to construct the breathing signal and neglecting other ECG components. Authors (Singhathip, Yang, Abbod, Yeh & Shieh, 2010) used a variation of R-R maxima to calculate the variation of R-R intervals, then interpolating values to generate the breathing signal. The major issue of this method is not able to calculate the variation of the small peaks such as Q and S peaks. Therefore, it only focuses on prominent peaks. After presenting several previous works that are related to the subject of the proposed research and showing backwards features that are not covered in previous works, this research suggests algorithm and method for detecting ECG components and constructing breathing signal simultaneously, through exploiting all peaks that affected by breathing activity. This method is particularly useful in situations where the ECG signals are the only available data source.

3. THE BREATHING INFLUENCE OF THE CARDIAC ELECTRICAL VECTOR

ECG signals recorded from the surface of the human body are influenced by the activity of breathing. The expansion and contraction of the chest accompany respiration action cause to change the mean electrical axis of the heart, which is the summation of all the vectors occurring in a cardiac cycle. During inspiration, filling lungs causes shift down of the diaphragm and stretch the apex of the heart towards the abdomen. During expiration the elevation of the diaphragm helps the emptying of the lungs changes the apex of the heart toward the breast (Moody, Mark, Zoccola & Mantero, 1985). Therefore, the amplitude of ECG components is oscillated according to the breathing activity (Espiritu Santo & Carbajal, 2010) as shown in figure (1).

Figure 1. ECG signals affected by breathing activity (Mason, 2002).

4. ECG COMPONENTS DETECTION USING A PATTERN ALGORITHM

The algorithm is developed to detect all peaks in ECG signals and dubbed a "pattern" because the heart works in the pattern mechanism. It commences with the P wave representing the depolarization of the atria; then followed by the QRS complex, which represents the depolarization of the ventricles; finally, the T wave acting the repolarization of the ventricles and continues in this manner as shown in figure (2) (Mason, 2002).

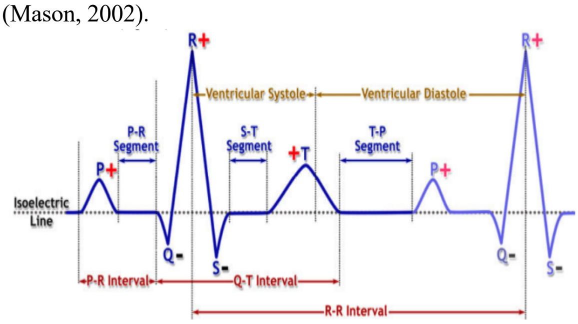

Figure 2. ECG components including Intervals and Segments (Pflanzer and McMullen, 2012).

The algorithm can takes significant advantage from continuing pattern by expecting where and when ECG components take place. Basically, the threshold is determined as default setting by the user and is often close to 0, depending on signals acquisition, due to threshold represents the baseline of periods when the ECG electrodes did not detect electrical activity. During acquiring ECG signals, threshold is often zero or close to zero. This simplifies detecting peaks by determine which peak is a negative under threshold and which is a positive above threshold. Step 1, the algorithm is detecting R peaks by searching above the threshold and finding highest peak based on sample rate. When one peak is being found, it will be saved and move to find other highest peak till detecting all R peaks. Step 2, it calculates time differences between R – R peaks. This will specify how many samples are constituted for each ECG signal. Because the number of samples that constitute one ECG signal varies depend on human activities. Normally, it is 75 beats (ECG signals) per 60 seconds, but after exercise activity may exceed 100 (beat per minute). After calculating Δ time for all R - R peaks, the equation (1) is used to determine samples that represent each single ECG signal (Pflanzer and McMullen, 2012).

.

$$ECGSamples = \Delta RPeakTime * SampleRate \quad (1)$$

Where: *ECGSamples* is the number of samples for each ECG signal

$\Delta RPeakTime$ is the time between two successive R peaks

SampleRate is the number of samples per seconds

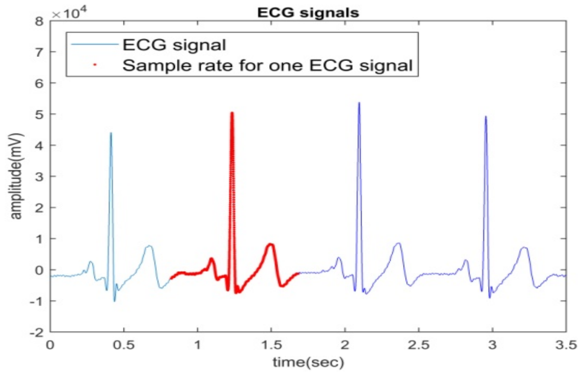

Figure 3. Showing samples for one ECG signal

Step 3, after calculating samples for each ECG signal, the algorithm determines all three segments of each ECG, as the segment is a time measurement that does not include peaks as shown in figure (2). Determining segments have two advantages; first this period doesn't contain peaks and it is not necessary to search in. Second, speed up the algorithm by skipping searching in samples that belong to this period. All three segments samples are calculated using equations (2, 3, and 4) below and based on the standard value for them (Pflanzer and McMullen, 2012).

$$PRSegment = SampleRate * 0.05(standard\ value) \quad (2)$$

$$STSegment = SampleRate * 0.1\ (standard\ value) \quad (3)$$

$$TPSegment = SampleRate * 0.2\ (standard\ value) \quad (4)$$

It is also possible to change the range of duration when necessary to obtain optimal results. As heart activity acts in continuing pattern, begins with depolarization of the atria (P wave), then depolarization of the ventricles (QRS waves) and repolarization of the ventricles (T wave); the algorithm designed carefully to follow this pattern and detect them accurately as shown in figure (**4**). It commences to seek above threshold to find highest peak (1 P). Then, after skipping PR segment, it searches beneath of threshold to find lowest peak (2 Q). Next, it turns to search above threshold again to find highest peak (3 R). After that, the algorithm seeks below of threshold to capture lowest peak (4 S). Finally, T wave (5 T) is detected by skipping ST segment and searching above of threshold to find highest peak. After skipping TP segment, it begins again to detect next signal using the same pattern.

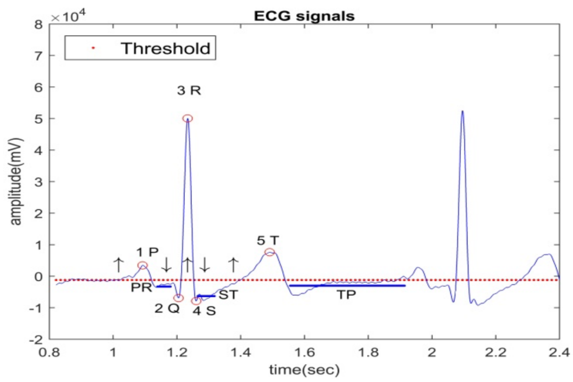

Figure 4. How pattern algorithm detects ECG components.

The algorithm continues in this manner until detect all peaks. As a result, all ECG components are detected as shown in figure (5) and save them separately to be used for various purposes. The algorithm has been applied on 35 Subjects' real signals aged between 22 and 37, 12 Subjects (34.28%) were female and the remaining 23 Subjects (65.71%) were male. The Subjects' signals downloaded from website PhysioNet (Goldberger et al., 2000), which offers free web access to large collections of recorded physiologic signals. Each subject's signals contain ECG signals and respiration signal that have been measured using sensor.

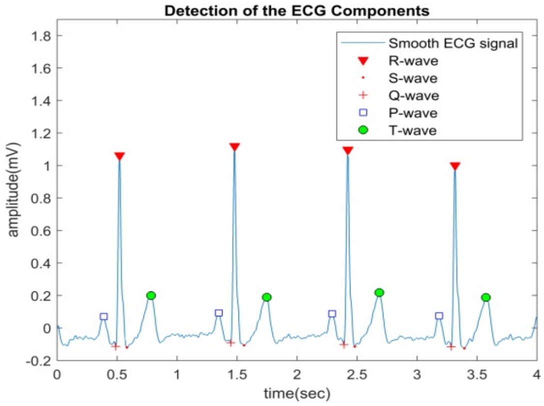

Figure 5. Detection of ECG components

5. METHOD OF CONSTRUCTING RESPIRATION SIGNAL

Various methods are used to construct respiration signal from ECG signals, and one of them is envelope method, which used size of R wave to generate breathing signal. Size of R wave is increased during inspiration. On the contrary, it decreases with expiration (Ruangsuwana, Velikic & Bocko, 2010). The proposed method improves the envelope method by detecting differences of size for all ECG components with the help of the pattern algorithm. The envelope method has been chosen because of dependence on amplitude size to construct respiration. When the mean electrical axis of the heart is changed during inspiration and expiration, influence not only R size but also PQST size as shown in figure (6) (Moody, Mark, Zoccola & Mantero, 1985). However, change over PQST waves is small compared to R but very useful and can be exploited to enhance the envelope method.

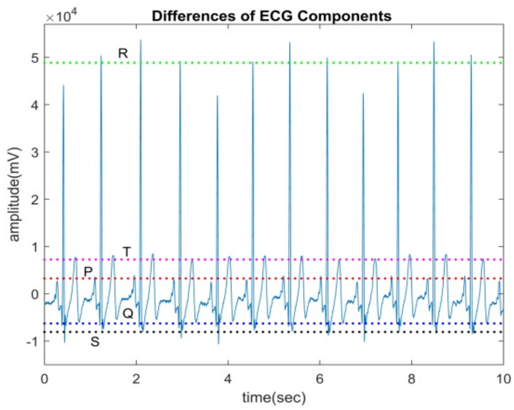

Figure 6. PQRST peaks affected by breathing activity.

The proposed method finds the mean of R peaks to determine inspiration and expiration phases as the effecting over them is much bigger than other waves as shown in figure (7).

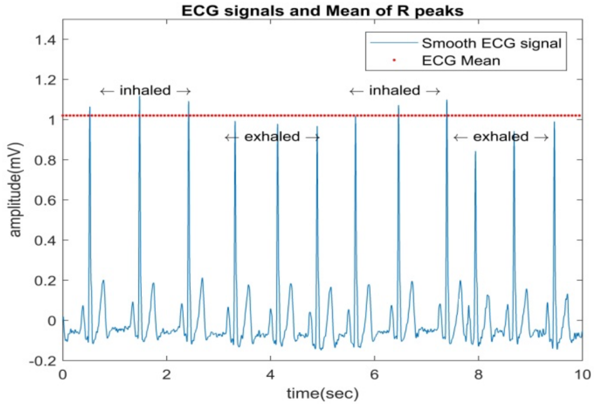

Figure 7. Determining the mean of R peaks and inspiration and expiration phases.

As PQST peaks have small changes compared to R peaks, the proposed method needs to find out how much each peak is affected separately. This is done by using equation (5), which calculates the changes of each peak from its mean. Then, each change will be added to R size in inspiration as shown in figure (8) or it will be subtracted from R size in expiration as shown in figure (9). After getting modified R size, it is resambled based on the original sample rate that was used in measuring.

$$\Delta PeakAmplitude = (x - \bar{x}) \qquad (5)$$

Where: Δ *PeakAmplitude* is the amount of changed peak
x is the one of (PQST) peaks
$\bar{x}$ is the mean of each (PQST) peak

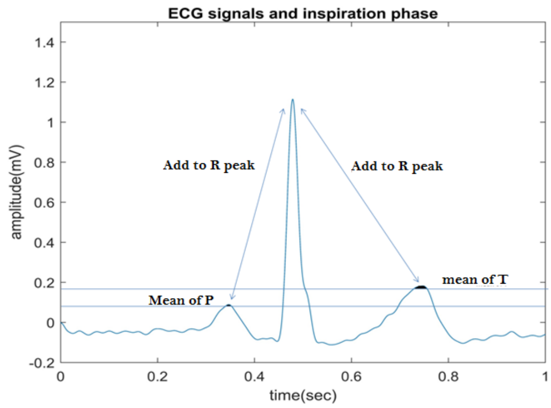

Figure 8. How differences of only PT peaks add to R in inspiration phase.

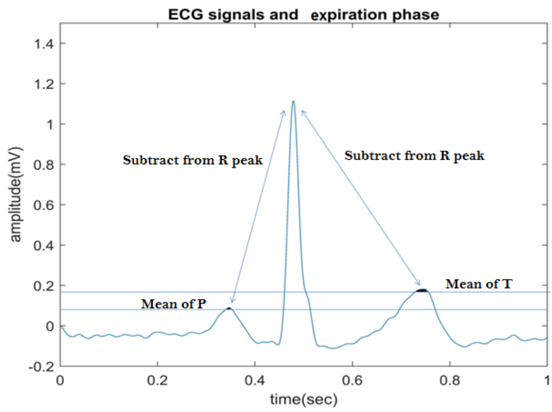

Figure 9. How differences of only PT peaks subtract from R in expiration phase.

As the result, the proposed method captures all differences of all peaks size accurately and use them to generate respiration signal correctly as shown in figure (10).

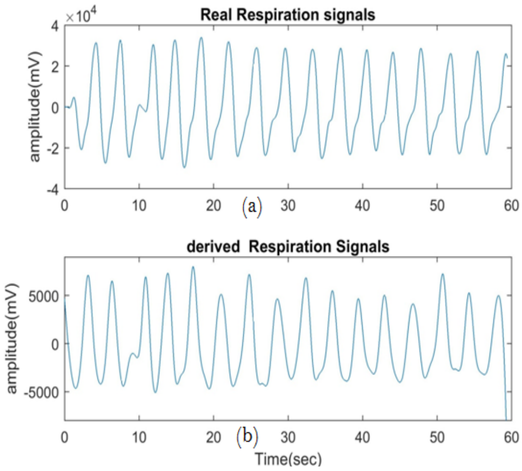

Figure 10. (a) Real Respiration from Sensor. (b) Derived Respiration signal from all peaks of ECG signals.

6. ANALYSIS AND RESULTS

The algorithm is developed in Matlab language (version 2017) and the result of pattern algorithm is also analysed by the same language. After the pattern algorithm has been performed efficiently on 35 Subjects' signals, breathing signals were derived properly using PQRST peaks in ECG signals. The verification of the results has been done using cross correlation, which measures the similarity between breathing signal that constructing from ECG and breathing signal that measure from the sensor as shown in figure (11).

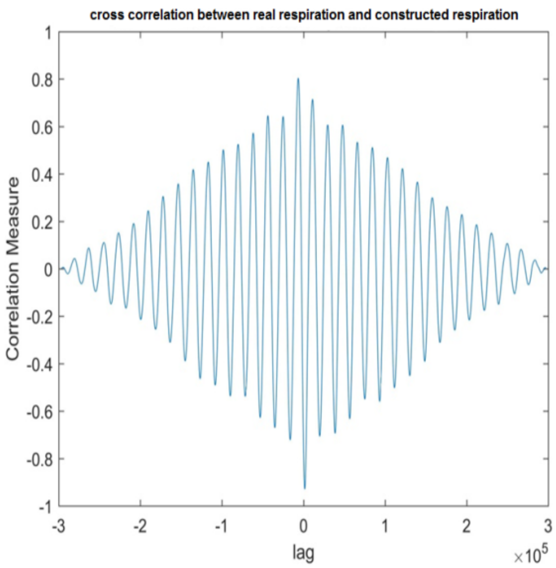

Figure 11. Cross correlation between real respiration and derived respiration.

Once, the verification has been applied between real respiration and derived respiration using R peaks (envelope method), and another time applied between real respiration and derived respiration using PQRST peaks (proposed method), to show enhancement that has been offered by the proposed method. As the result, the proposed method works better than the envelope method as shown in table 1.

Table 1. Results of Cross-Correlation Coefficients of Breathing Signal and both ECG Derived Signal using R peak and using PQRST peaks for 35 Subjects

Subjects' signals for 10 minutes		
Subjects data number	**Envelope method**	**Proposed method**
1	0.8259	0.8611
2	0.9005	0.9324
3	0.9010	0.9120
4	0.9161	0.9269
5	0.9173	0.9246
6	0.8954	0.9014
7	0.8124	0.8538
8	0.9101	0.9357
9	0.7369	0.7798
10	0.8366	0.8599
11	0.7874	0.7996
12	0.7840	0.7941
13	0.8325	0.8547
14	0.8326	0.8399
15	0.7974	0.8147
16	0.8656	0.8698
17	0.6550	0.6897
18	0.7481	0.7758
19	0.7112	0.7414
20	0.7598	0.7654
21	0.8655	0.8711
22	0.7412	0.7445
23	0.8366	0.8571
24	0.9125	0.9327
25	0.7712	0.7810
26	0.8363	0.8411
27	0.7415	0.7541
28	0.8445	0.8528
29	0.9127	0.9245
30	0.8740	0.8865
31	0.7992	0.8010
32	0.8255	0.8645
33	0.9179	0.9237
34	0.8976	0.8996
35	0.7883	0.7978

Respiration rate is one of the important factors to detect sleep apnoea and sleep quality estimation. It can be measured for both derived respiration and real respiration by detecting highest peaks and applying the equation (6) (Ruangsuwana, Velikic & Bocko, 2010), to show similarities between them as shown in table 2.

$$BR = (60/\Delta\, PeakTime)[breaths/minute] \quad (6)$$

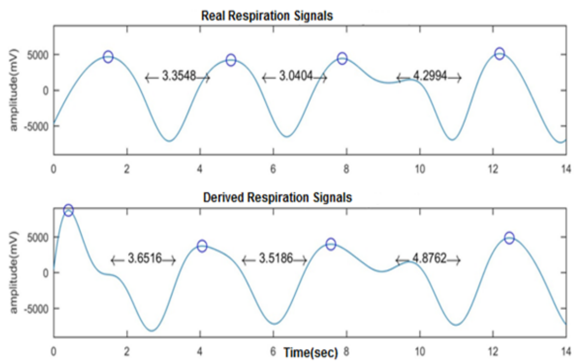

Figure 12. Respiration cycles for real breathing signal and derived signal

Table 2. Respiration rate for both real breathing signal and derived signal for 35 Subjects

Subjects' signal for 1 minute		
Subjects data number	**Breathing rate using sensor**	**Respiration rate derived by using ECG signals**
1	16.8	16.4
2	14.2	14.1
3	20.4	20.1
4	17.3	17.5
5	19.5	19.1
6	21.4	21.6
7	15.5	15.2
8	16.3	16.1
9	15.6	15.8
10	16.8	17.1
11	19.3	19.1
12	20.9	21
13	17.6	17.9
14	18.2	18.4
15	17.9	17.6
16	22.5	22.6
17	18.3	18.2
18	19.3	19.5
19	20.7	21
20	21.5	21.4
21	17.8	18
22	16.1	16.4
23	17.5	17.4
24	18.6	18.1
25	19	18.8
26	23.5	23.1
27	20	19.8
28	17	17.1
29	19.5	19.6
30	17.6	17.2
31	18.9	19
32	15.8	15.7
33	16.3	16.4
34	19.4	19.3
35	20.3	20.4

Generally, the result of respiration depends on how fast the breathing is performed as well as condition of the body (seated, stand and after exercise).

7. CONCLUSIONS

This paper shows that how the pattern algorithm is developed properly as well as it able to detect ECG components accurately. This algorithm is promised to be used widely for various purposes such as analysing ECG signals. The paper also demonstrates how the results of the algorithm have been exploited to introduce new method for constructs respiration signal and offering improvement over the envelope method by getting all differences of peaks.

8. REFERENCES

Espiritu Santo, A., & Carbajal, C. (2010). Respiration rate extraction from ECG signal via discrete wavelet transform. 2010 2Nd Circuits And Systems For Medical And Environmental Applications Workshop (CASME). doi: 10.1109/casme.2010.5706679.

Goldberger, A., Amaral, L., Glass, L., Hausdorff, J., Ivanov, P., & Mark, R. et al. (2000). PhysioBank, PhysioToolkit, and PhysioNet. Circulation, 101(23). doi: 10.1161/01.cir.101.23.e215.

Kim, J., Hong, J., Kim, N., Cha, E., & Lee, T. (2006). Two Algorithms for Detecting Respiratory Rate from ECG Signal. World Congress On Medical Physics And Biomedical Engineering 2006, 4069-4071. doi: 10.1007/978-3-540-36841-0_1030.

Mason, L. (2002). Signal Processing Mehods for Non-Invasive Respiration Monitoring (Ph.D). University of Oxford.

Moody, G., Mark, R., Zoccola, A., & Mantero, S. (1985). Derivation of Respiratory Signals from Multi-lead ECGs. Computers In Cardiology. Los Alamitos, CA: IEEE Computer Society Press, vol.12, pp. 113–116.

Park, S., Noh, Y., Park, S., & Yoon, H. (2008). An improved algorithm for respiration signal extraction from electrocardiogram measured by conductive textile electrodes using instantaneous frequency estimation. Medical & Biological Engineering & Computing, 46(2), 147-158. doi: 10.1007/s11517-007-0302-y.

Pflanzer, R. and McMullen, W. (2012). Electrocardiography (ECG) I Introduction. Biopac Student Lab Manual. BIOPAC Systems, Inc., pp.pp. I1-I4.

Ruangsuwana, R., Velikic, G., & Bocko, M. (2010). Methods to extract respiration information from ECG signals. IEEE International Conference On Acoustics, Speech And Signal Processing, pp. 570-573.

Sarkar, S., Bhattacherjee, S., & Pal, S. (2015). Extraction of respiration signal from ECG for respiratory rate estimation. Michael Faraday IET International Summit 2015. doi: 10.1049/cp.2015.1654.

Sharma, H., & Sharma, K. (2018). ECG-derived respiration using Hermite expansion. Biomedical Signal Processing And Control, 39, 312-326. doi: 10.1016/j.bspc.2017.08.016.

Singhathip, R., Yang, S., Abbod, M., Yeh, R., & Shieh, J. (2010). EXTRACTING RESPIRATION RATE FROM RAW ECG SIGNALS. Biomedical Engineering: Applications, Basis And Communications, 22(04), 307-314. doi: 10.4015/s1016237210002079.

9. APPENDIX

```
clc;
close all;
clear all;
load(' '); %Load data(database) for workplace
ecgSignals=val( ); %Get ECG signals from database
respirationSignals = val( ); %Get real respiration signals from database
fs=5000; %Sample rate
t=0:1/fs:length(ecgSignals)/fs-1/fs; %Time change minutes to seconds

ecgSignalsThreshold = detrend(ecgSignals); %Adjusting threshold to 0
respirationSignals = detrend(respirationSignals); %Adjusting threshold to 0
meanECG = mean(ecgSignalsThreshold); %Calculating  the Mean to find R peaks

controlCase = 0;%Changing cases
RPeaks = [ ; ]; %Array to put R peaks &  their locations
RPeaksDiff = []; %Different between R-R peaks
tempArray = [;]; %Temprary array for find max value in array
rowCol = 0; %Index for Temp Array
flag = true; %Change Status of switch case
index = 1; %Index for RPeaks Array
i = 1; %Counter
% Detecting R peaks
while i <= length(ecgSignalsThreshold)

    switch controlCase
   case 0 %Find highest R Peak
         if ecgSignalsThreshold(i) >= meanECG
            rowCol = rowCol + 1;
            tempArray(rowCol,1) = ecgSignalsThreshold(i); %Getting R peaks
            tempArray(rowCol,2) = t(i); %Getting R time
            flag = false;
         elseif flag == false
            [maxValue,  indexMaxValue]  =  max(tempArray(:,1));  %Get  highest  value  and
index
            RPeaks(index,2) = maxValue; %Give R value to matrix
            RPeaks(index,1) = tempArray(indexMaxValue,2); %Give index of R to matrix
            flag = true;
            rowCol = 0; %Reset
            tempArray = [];
            controlCase = 1; %Change Case
         end
        case 1 %Move to next row of R Peaks
            controlCase = 0 ;
            index = index + 1;
    end
  i = i + 1;
end


%Detecting PQRST
 RPeaksDiffTime = diff(RPeaks(:,1)); %Find different time between each interval R-R
 meanRTime = mean(RPeaksDiffTime); %Find mean time
 sampleRpeak = fs * meanRTime; %Generally find how many samples in each signal
 meanRPeaks = mean(RPeaks(:,2)); %Find mean of R peak to determine which is inhale and
exhale

%total samples rate for each singal
sampleTotal = RPeaksDiffTime * fs;
wave = 0; %Changing Switch case
PRSegment = 0;
STSegment = 0;

%Matrix include all peaks values and their location
waveMatrix = [ ; ]; %Saving PQRST peaks
arrFindMax = [ ; ]; %Array to save temp data for find max value
rowCol = 0; %Index for arrFindMax
flag = true; %Change case
index = 1; %Change row for array of PQRST
```

```
i = 1; %Counter
indexPQRST = 1;%Find next sample rate which consist from different R peaks to calculate
Segments
threshold = 0 ; %Threshold define by user

while i < length(ecgSignalsThreshold)
      switch wave
        case 0
            %Find  P peak
         if ecgSignalsThreshold(i) > threshold
            rowCol = rowCol + 1; %Increament for arrFindMax
            arrFindMax(rowCol,1) = ecgSignalsThreshold(i); %Getting P peak
            arrFindMax(rowCol,2) = t(i); %Getting P location
            flag = false;
         elseif flag == false
             [maxValue, indexMaxValue] = max(arrFindMax(:,1)); %Get p value and index
             waveMatrix(index,1) = maxValue; %Give p value to matrix
             waveMatrix(index,2) =  arrFindMax(indexMaxValue,2); %Give p index to matrix
             flag = true;
             indexMaxValue = 0; %Reset
             maxValue = 0; %Reset
             rowCol = 0; %Reset
             arrFindMax = []; %Reset

             % Set Sequment
             PRSegment = round(0.05 * sampleRpeak(indexPQRST));
             STSegment = round(0.1  * sampleTotal(indexPQRST));
             TPSegment = round(0.2  * sampleRpeak(indexPQRST));
             i = i +  PRSegment;
              wave = 1;%Go to Q wave
         end
          case 1
             %Find Q peak
         if ecgSignalsThreshold(i)  < threshold
            rowCol = rowCol + 1; %Increament for array of arrFindMax
            arrFindMax(rowCol,1) = ecgSignalsThreshold(i); %Getting Q peak
            arrFindMax(rowCol,2) = t(i); %Getting Q location
            flag = false;
         elseif flag == false
            [minValue, indexMinValue] = min(arrFindMax(:,1)); %Get Q value and index
            waveMatrix(index,3) = minValue; %Give Q value to matrix
            waveMatrix(index,4) =  arrFindMax(indexMinValue,2); %Give Q index to matrix
            flag = true;
            indexMinValue = 0; %Reset
            minValue = 0; %Reset
            rowCol = 0; %Reset
            arrFindMax = []; %Reset
            wave = 2; %Go to R peak
         end
       case 2
           %Find  R peak
        if ecgSignalsThreshold(i) > threshold
            rowCol = rowCol + 1; %Increament for array of arrFindMax
            arrFindMax(rowCol,1) = ecgSignalsThreshold(i); %Getting R peak
            arrFindMax(rowCol,2) = t(i); %Getting R time
            flag = false;
         elseif flag == false
            [maxValue, indexMaxValue] = max(arrFindMax(:,1)); %Get R value and index
            waveMatrix(index,5) = maxValue; %Give R value to matrix
            waveMatrix(index,6) =  arrFindMax(indexMaxValue,2); %Give R index to matrix
            arrFindMax = []; %Reset
            flag = true;
            indexMaxValue = 0;%Reset
            maxValue = 0;%Reset
            rowCol = 0;%Reset
            wave = 3; %Go to S peak
        end
          case 3
            %Find  S peak
         if ecgSignalsThreshold(i)  < threshold
```

```
            rowCol = rowCol + 1; %Increament for array of arrFindMax
            arrFindMax(rowCol,1) = ecgSignalsThreshold(i); %Getting S peak
            arrFindMax(rowCol,2) = t(i); %Getting S time
            flag = false;
         elseif flag == false
            [minValue, indexMinValue] = min(arrFindMax(:,1)); %Get S value and index
            waveMatrix(index,7) = minValue; %Give S value to matrix
            waveMatrix(index,8) =  arrFindMax(indexMinValue,2); %Give S index to matrix
            flag = true;
            indexMinValue = 0; %Reset
            minValue = 0; %Reset
            rowCol = 0; %Reset
            arrFindMax = []; %Reset
            i = i + STSegment;
            wave = 4; %Go to T peak
         end
        case 4
            %Find T peak
        if ecgSignalsThreshold(i) > threshold
            rowCol = rowCol + 1; %Increament for array of arrFindMax
            arrFindMax(rowCol,1) = ecgSignalsThreshold(i); %Getting T peak
            arrFindMax(rowCol,2) = t(i); %Getting T time
            flag = false;
         elseif flag == false
            [maxValue, indexMaxValue] = max(arrFindMax(:,1)); %Get T value and index
            waveMatrix(index,9) = maxValue; %Give T value to matrix
            waveMatrix(index,10) =  arrFindMax(indexMaxValue,2); %Give T index to matrix
            flag = true;
            indexMaxValue = 0; %Reset
            maxValue = 0; %Reset
            rowCol = 0; %Reset
            arrFindMax = []; %Reset
            %i = i + TPSegment;
             wave = 0 ; %Go to P peak again
            index = index + 1; %Next row and peaks
        end
     end
   i = i + 1; %Next sample
    if(indexPQRST <length(RPeaksDiffTime))
   indexPQRST = indexPQRST +1; %Next sample rate
    end
end

 meanP = mean(waveMatrix(:,1));
 meanT = mean(waveMatrix(:,9));
 meanS = mean(waveMatrix(:,7));
 meanQ = mean(waveMatrix(:,3));
 pqrst = [ ; ];%For all Result

 j=1;
 while (j<= length(waveMatrix))
       if (waveMatrix(j,5)) > meanRPeaks %inhale: add changes
           pqrst(j,2) = waveMatrix(j,6);
           pqrst(j,1)   =   waveMatrix(j,5)   +   abs(waveMatrix(j   :,9)   -   meanT)   +
abs(waveMatrix( j :,1) - meanP) + abs(waveMatrix(j :,7) - meanS) + abs(waveMatrix(j :,3)
- meanQ) ;
       elseif (waveMatrix(j,5)) < meanRPeaks %exhale sub changes
           pqrst(j,2) = waveMatrix(j,6);
           pqrst(j,1)   =   waveMatrix(j,5)   -   abs(waveMatrix(j   :,9)   -   meanT)   -
abs(waveMatrix( j :,1) - meanP) - abs(waveMatrix(j :,7) - meanS) - abs(waveMatrix(j :,3)
- meanQ) ;
       else
           pqrst(j,2) = waveMatrix(j,6);
           pqrst(j,1) = waveMatrix(j,5);
       end
       j = j+1;
 end

%Constract Breathing after getting accurate changes
constractBreathingRPeaks = spline(pqrst(:,2),pqrst(:,1),t);
```

```
constractBreathingRPeaks = detrend(constractBreathingRPeaks); %Adjusting threshold to 0
[xCorR,xCorRSignal]    =    xcorr(respirationSignals,constractBreathingRPeaks,'coeff');
%Applying cross correlation between real breathing signal and constructed signal
[~,I] = max(abs(xCorR));
lagDiff = xCorRSignal(I);
timeDiff = lagDiff/fs;

%Print
figure('Name','ECG and PQRST peaks')
plot(t,ecgSignalsThreshold);
title('ECG');
hold on;
plot(waveMatrix(1:end,2),waveMatrix(1:end,1),'bs',waveMatrix(1:end,4),waveMatrix(1:end
,3),'r+',waveMatrix(1:end,6),waveMatrix(1:end,5),'rv',waveMatrix(1:end,8),waveMatrix(1
:end,7),'r.',waveMatrix(1:end,10),waveMatrix(1:end,9),'k*');

%Cross
figure('Name','Cross correlation between Real and constructed signal')
plot(xCorRSignal,xCorR);
```