January 2016

A Morphological Approach To Identify Respiratory Phases Of Seismocardiogram

Nasim Taghizadeh Alamdari

A MORPHOLOGICAL APPROACH TO IDENTIFY RESPIRATORY PHASES OF

SEISMOCARDIOGRAM

by

Nasim Taghizadeh Alamdari
Bachelor of Science, Islamic Azad University, 2013

A Thesis

Submitted to the Graduate Faculty

of the

University of North Dakota

in partial fulfillment of the requirements

for the degree of

Master of Science

Grand Forks, North Dakota

August
2016

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

_____
Dr. Kouhyar Tavakolian

_____
Dr. Reza Fazel-Rezai

_____
Dr. Sima Noghanian

_____
Dr. Alireza Akhbardeh

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

_____

Dr. Grant McGimpsey,

Dean of the Graduate School

_____

Date

PERMISSION

Title   A Morphological Approach To Identify Respiratory Phases Of Seismocardiogram

Department   Electrical Engineering

Degree   Master of Science

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

Nasim Taghizadeh Alamdari
July 5th , 2016

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

GLOSSARY

AC	Aortic valve closure
AO	Aortic valve opening
BCG	Ballistocardiography
ECG	Electrocardiography
HR	Heart rate
HRV	Heart rate variability
RR	Respiration rate
PPG	Phonocardiography
SCG	Seismocardiography
VCG	Vibrocardiography

ACKNOWLEDGEMENTS

ABSTRACT

Respiration affects the cardiovascular system significantly and the morphology of signals relevant to the heart changes with respiration. Such changes have been used to extract respiration signal from electrocardiogram (ECG). It is also shown that accelerometers placed on the body can be used to extract respiration signals. It has been demonstrated that the signal morphology for seismocardiogram, the lower frequency band of chest accelerations, is different between inhale and exhale. For instance, systolic time intervals (STI), which provide a quantitative estimation of left ventricular performance, vary between inhale and exhale phases. In other words, heart beats happening in exhale phase are different compared to those in inhale phase. Thus, our main goal in this thesis is investigating feasibility of finding an automatic morphological based method to identify respiratory phases of heart cycles.

In this thesis, forty signal recordings from twenty subjects were used. In each recording, the reference respiratory belt signal, three dimensional (3D) chest acceleration signals, and electrocardiogram signals were recorded. The first stage was is choosing a proper estimated respiratory signal. The second stage, was the automatic respiratory phase detection of heart cycles using the selected estimated respiratory signal. The result shows that among estimated respiratory signals, accelerometer-derived respiration (ADR), in z-direction, has a potential m to identify respiratory phase of heart cycles with total accuracy of about 77%.

CHAPTER 1

INTRODUCTION

The global problem of disease has been flowing gradually from transmissible diseases in children to non-transmissible disease in adults [1]. Specifically, cardiovascular diseases have effects on an outsized range of individuals worldwide. For instance, globally 7,249,000 deaths due to cardiovascular diseases (12.7% of total deaths) were recorded in 2008 [2]. Cardiovascular diseases are the leading causes of death globally and in the United States. According to the American Heart Association 2013, the leading reason of 80000 deaths annually are related to cardiovascular diseases in the US [3]. Therefore, in USA alone, nearly $95 billion annually is spent on diagnosis and treatment of cardiovascular diseases [4]. Thus, cardiovascular diseases cause a significant financial and health burden [5]. In addition, the average life range has been increased by quality enhancement of life and advancement of health-care. Consequently, many developed nations considered to have aging population. Accordingly, the concern regarding elderly's quality of health care in residential and nursing homes, and home healthcare has been raised [6].

Today, the cardiovascular system and heart function are standard observation targets in humans. In the USA, a useful compelling cardiovascular observation framework has been proposed across the nation [3]. With respect to the performance of cardiovascular disease monitoring systems, the advancement of low-cost measurement tools is an important subject. Today the function of the heart and cardiovascular system can be observed by a variety of

techniques. Visual observation, palpation, and auscultation are some old and simple methods. In summary, in order to discover cardiac arrhythmias and other heart dysfunctions, by combining visual, auditory and sensory symptoms, an expert physiologist can diagnose a great number of cardiovascular abnormalities [5].

Generally, non-intrusiveness and non-invasiveness are the most important features of home healthcare. Thus, telemedicine techniques or remote monitoring of physiological or daily living signs should have these features: (a) they should not cause any pain for the patients and (b) they should not have unnecessary alteration to life style. [7]. In the cardiovascular monitoring area, non-obstructive techniques are preferred, mostly because of their cost efficiency and lower related risks [5].

The observation and estimation of respiration is vital in many circumstances. Respiration is the most significant modulator of heart rate, and it is responsible for the short term heart rate variability (HRV). Understanding autonomic regulation of heart rate can be achieved by analyzing heart rate variability.

Regarding cardiovascular monitoring, it has been demonstrated that the signal morphology for seismocardiogram, the lower frequency portion of acceleration signal recorded from the chest, is different between inhale and exhale. For instance, systolic time intervals (STI) ,which provide a quantitative estimation of left ventricular performance, vary between inhale and exhale phases. In other words, heart beats happening in exhale phase are different compared to those happening during inhale phase. In order to perform signal averaging and increasing the accuracy of diagnosis, the heart cycles related to inspiration should be discriminated from those related to the expiration, and this has been the major objective of this project.

Continuous respiratory monitoring methods can be classified to three different categories: (i) motion measurement devices, tissue or volume based monitoring such as trans-thoracic impedance techniques, inductance plethysmography, (ii) airflow-based measurements such as thermistors for measurement of oro-nasal airflow, and (iii) blood gas changes-based monitoring such as pulse oximetry, or end-tidal O_2 measurement [8]. The respiratory signal is usually recorded with techniques like spirometry, pneumography, or plethysmography. These techniques require the use of cumbersome devices that may interfere with natural breathing, and are unmanageable in certain applications such as ambulatory monitoring, stress testing, and sleep studies. Nonetheless, the joint study of the respiratory and cardiac systems is of great interest in certain applications. As such, the use of methods for indirect extraction of respiratory information is particularly attractive to pursue [9].

Therefore, first stage to identify respiratory phase of heart cycles of seismocardiogram is estimating respiratory signal from electromechanical signals such as ECG and SCG signals. The next stage, is the automatic respiratory phase detection of heart cycles using the selected estimated respiratory signal.

Therefore, the first stage to identify respiratory phase of heart cycles of seismocardiogram is estimating respiratory signal from SCG and ECG signals. The next stage, is the automatic respiratory phase detection of heart cycles using the extracted respiration signal.

This thesis is organized as follows. In the second chapter, the background regarding cardiorespiratory physiology and non-invasive techniques for monitoring cardiac and respiratory functions are reviewed. In chapter three, methods and materials, which are used in this thesis are discussed. In chapter four, the acquired results and measurements from the volunteer subjects are

presented. Finally, the importance of the proposed technique and future studies are represented in chapter five.

1.1 Study Objectives

The objectives are this thesis are the followings:

- Derive respiratory signals from electromechanical signals (e.g. electrocardiogram (ECG), 3D seismocardiogram (SCG) signals).

- Find the most proper signal among the estimated respiratory signals. Define respiratory phases of heart cycles by using the selected estimated respiratory signal. In other words, defining which heart cycle occurs during inhale and which ones occur during exhale.

CHAPTER 2

BACKGROUND

In this chapter, first physiology of cardiovascular system including cardiac cycles phases, and respiration physiology will be described. As we mentioned in the previous chapter, our goal is discrimination of heart beats to inhale and exhale phase to improve estimating systolic time intervals using SCG signal. For this reason, an indirect derived respiratory signal from SCG is used to detect the beginning of respiratory phases. Section two of this chapter defined different non-invasive contact based respiratory monitoring measurement techniques to emphasize beneficial aspects of using an indirect derived-respiratory method instead of utilizing direct respiratory system measurement. Different non-invasive heart function monitoring methods were described in the third section including seismocardiography and electrocardiography. Finally, the last section of this chapter review previous studies regarding computing electrocardiogram-derived-respiratory (EDR) signal and accelerometer-derived-respiratory (ADR) signal using different approaches.

2.1 Cardiovascular Physiology

2.1.1 Cardiac cycle phases

The cardiac cycle states to a complete pulse from its generation to the start of the next beat, thus contains the diastole, the systole, and the intervening pause (Figure 1). The heart rate is the frequency of the cardiac cycle, which is typically defined as beats per minute. Each heart cycle includes five main stages.

The initial two stages, regularly viewed together as the "ventricular filling" stage, include the blood movement from the atria into the ventricles. The following three stages include the blood movement from the ventricles to the aorta (in the left ventricle) and the pulmonary artery (in the right ventricle) [10].

Diastole, the initial stage, happens when the semilunar valves (the aortic valve, and the pulmonary valve) close, the atrioventricular (AV) valves (the tricuspid valve and the mitral valve) open, and the entire heart is relaxing. In the next stage, the atrium contracts, and blood flows from atrium to the ventricle which is called "atrial systole". When the ventricles start to contract, the AV and semilunar valves close, and there is no change in volume. This is called "isovolumic contraction". In the "ventricular ejection", fourth stage, the ventricles are contracting and draining, and the semilunar valves are open. During "isovolumic relaxation time", the final stage, pressure reduces, blood don't come into the ventricles, contraction of ventricles stops and relaxation of ventricles starts, and the semilunar valves close because of the blood pressure in the aorta [10]. Decreasing and increasing blood pressure happen in the entire cardiac cycle. A series of electrical impulses that are generated by particular pacemaker cells originated in the sinoatrial node and the atrioventricular node produce a heartbeat. In normal condition, each heart cycle lasts for 0.8 seconds [10].

2.1.2 Respiration physiology

During the breathing, the oxygen (air) from the environment pulled toward the lungs and the carbon dioxide pulled away from the lungs and release into the atmosphere by the contraction of lungs. These two acts are named inspiration and expiration phases, respectively.
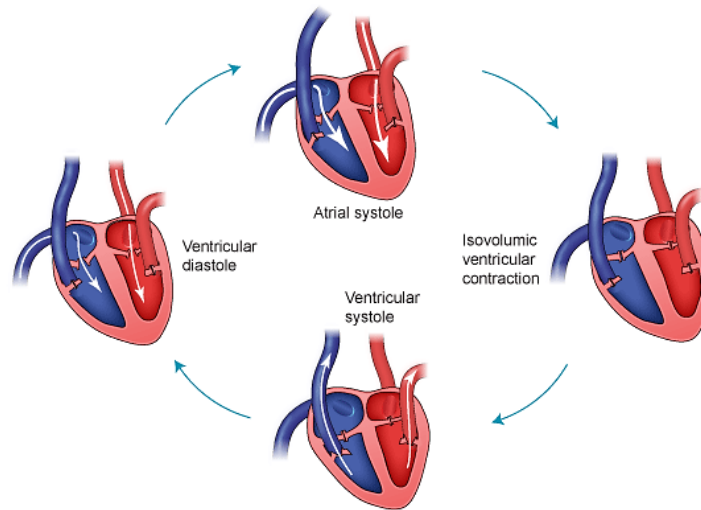
Figure 1 Cardiac cycle phases [11].

As shown in Figure 2, the participation of the different organs in the respiratory system cause the inhale and exhale phases [12]. These organs divided into two section: upper and lower respiratory tract. Nostrils, Nasal Cavities, Pharynx, Epiglottis, and the Larynx are considered as upper respiratory tract and Trachea, Bronchi, Bronchioles, and the Lungs considered as lower respiratory tract.

Change in volume level of the thoracic cavity causes pulled towards or away from the lungs. Diaphragm with its surrounded muscles (e.g. intercostal muscles) effect the lungs contraction to inhale the air into the lungs. In contrast, muscles contraction causes the passive action of exhalation [12].
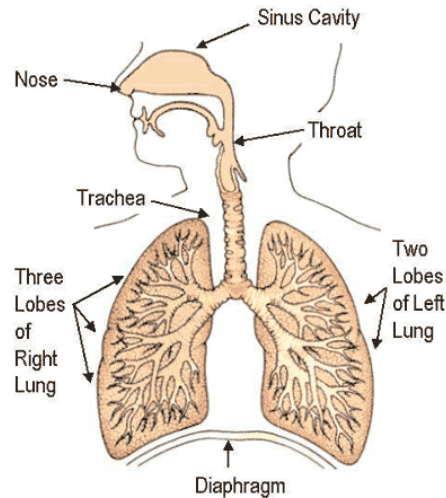
Figure 2 The anatomy of human respiratory system [13].

2.2 Non-invasive contact based respiratory monitoring

One of the important tasks in vital sign monitoring is respiration rate observation and measurement. Techniques that compute respiration rate can be categorized by variety of ways depending on their application and their operation.

Generally, contact respiration rate monitoring methods are based on measuring respiratory airflow, respiratory sounds, respiratory related chest or abdominal movements, oximetry probe SpO_2, or respiratory CO_2 emission [14]. The following techniques are the contact-based respiration monitoring systems.

2.2.1 Acoustic based methods

One common location for measuring breathing sounds is on the neck close to the respiratory airways or over the throat, like Figure 3 shows. In this case, placed microphone in the aforementioned places can detect the variation of sound [14]. These sounds are relatively large in amplitude and have a wider range of frequencies than sounds recorded at the chest wall, and also

have a close relation to the tracheal airflow [15]. Therefore, a frequency analysis and estimation of the loudness of the sound can be performed to estimate respiratory rate [14].

Recent studies proposed a miniaturized wearable respiration system that contains a microphone placed on the neck to record the largest breathing acoustic [16]. In addition, feasibility of detecting sleep apnea in infants using sound-based system has been proposed [17].

Figure 3 Example of acoustic based respiratory method [18].

2.2.2 Airflow-based methods

In airflow-based respiratory systems, change in temperature, humidity and CO_2 level can differentiate between inhale and exhale phase. In exhale phase, the air is warmer, humidity and level of CO_2 are higher. There variation can be utilized to indicate the respiratory rate (RR). The most common place for this method is attaching sensor to the airway [19]. The sensor can be nasal thermistor which measure temperature variation during inhaled and exhaled air. Pressure sensor, more specifically, a nasal pressure transducer also can be used to detect RR and its results is more

accurate than other sensors [14]. The disadvantages of placing a sensor to the airflow is the discomfort felt by patients.

2.2.3 Chest and abdominal measurements

The common way to measure the respiratory rate based on movements of chest and the abdominal wall is either using impedance methods or mercury strain gauges. The principle of the strain gauge sensor is based on increase area of the conductor, and therefore, increase in the conductor resistance during the inspiration and expiration. As illustrated in Figure 4, in respiratory inductance plethysmography two bands compute the RR. The abdominal band which is placed over the abdomen at the level of the umbilicus and the other band is the thoracic band which is placed around the rib cage [20].

Figure 4 Example of chest and abdominal respiratory detection [21].

The inspiratory thoracic and abdominal expansion usually should be synchronous. However, in some conditions such as upper airway partial obstruction, the movements of the thorax and abdomen can become asynchronous [22]. This condition can be seen in infants that their chest wall compliance is greater [23].

2.2.4 *Transcutaneous CO$_2$ monitoring*

Transcutaneous CO$_2$ monitoring is based on overall CO$_2$ variation rather than measuring changes in CO$_2$ level minute by minute readings. This method contains a heated electrode (about 428 °C) which is attached to the skin (usually an arm). In this sensor, electrode is bounded by a solution to provide conductivity. To avoid burning sensitive and neonatal skin, the procedure should be performed carefully [24]. The advantages of this method is its ability to measure consequences of abnormal ventilation instead of measuring only the RR [14].

2.2.5 *Oximetry probe (SpO$_2$)*

Oximetry probe-based method or pulse oximetry is a non-invasive method which measures blood oxygen saturation (SpO$_2$). SpO$_2$ is the percentage of blood saturation. When inhaled air come into the lungs, its oxygen binds to the hemoglobin in red blood cells. Then, the oxygen is moved throughout the body in arterial blood. A pulse oximeter passes two spectrum bands, the red and infrared frequencies, through the body part to a photodetector to estimate the percentage of hemoglobin in the blood that is loaded with oxygen. This percentage is called blood saturation [14].

Figure 5 Pulse oximetry [25].

Respiration rate can also be derived from the electromechanical signals such as ECG and SCG. Following sections will introduce these heart function monitoring techniques and how the respiration event can be extracted from them.

2.3 Non-invasive Heart Function Monitoring

In addition to well-established methods such as cardiovascular system palpation and auscultation are both well-established techniques [26][27], technological developments have brought upon new windows to understanding heart functions. A variety of techniques have been proposed to comprehend cardiac functions throughout the 20th century (see Figure 6). Using more modern and advanced non-invasive cardiovascular function assessment techniques are required in addition to cardiac auscultation and palpation which are still routinely used in clinical practice [5].
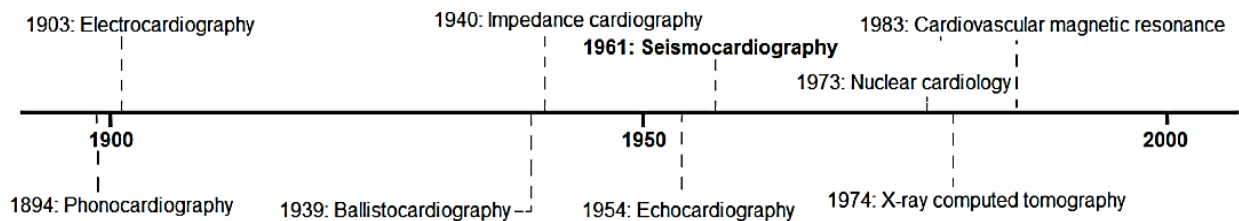
Figure 6 Modern and advanced non-invasive cardiovascular assessment techniques [5].

2.3.1 Echocardiography (ECHO)

Echocardiography or cardiac echo (ECHO), is an imaging method based on ultrasound. ECHO is a sonogram of the heart. This method can utilized two-dimensional, three-dimensional or Doppler-based ultrasound to create heart images. The primary demonstrative endeavors utilizing ultrasound were made in a part of the late 1940s by Karl Theo Dussik [28]. At the present time, the ECHO are routinely utilized in diagnosis cardiovascular diseases and estimation of heart function such as cardiac output, ejection fraction and how well the heart is relaxing.

2.3.2 Cardiovascular magnetic resonance imaging (cMRI)

Cardiovascular magnetic resonance imaging (cMRI) or cardiac MRI is a medical imaging method for non-invasive assessment of the structure and function of the cardiovascular system

based on principles of nuclear magnetic resonance. Magnetic resonance imaging (MRI) was first introduced in the 1930s and 1940s. Clinical potential of cMRI was first broadly addressed in 1983. Today, the main application of cMRI is the investigation of cardiovascular disease such as congenital heart defects which is the most common type of major birth defect among children [29].

2.3.3 X-ray computed tomography (X-CT)

X-ray computed tomography (X-CT) or CT scan, is an imaging modality where X-rays are used to create volume of data or slice images of a subject. CT scan are typically used for preventive medicine or screening for disease such as cancer or high risk heart disease. More specifically, it is used for coronary angiography and, the detection of coronary atherosclerosis [29].

2.3.4 Nuclear cardiology

Nuclear cardiology is non-invasive method to measure the myocardial blood flow, assess the pumping capacity of the heart and in addition visualize the size and area of the heart attack by injecting radioisotopes into the vascular system and the emitted radiation. Myocardial perfusion imaging is the most widely used technique among nuclear cardiology techniques [30].

2.3.5 Electrocardiography (ECG)

In electrocardiography (ECG or EKG), the electrical activity of the heart over period of time is studied using electrodes located on the skin. Einthoven's electrocardiogram [31] is broadly well-thought-of to be the origin of ECG. In this method, electrodes detect every tiny electrical activity changes to show pattern of depolarization during each heartbeat. Today, ECG is commonly used as cardiology test and remote monitoring [32].

2.3.6 Phonocardiography (PCG)

Phonocardiography or PCG is a techniques to record sound and murmurs of the heart and blood flow during a cardiac cycle [ref]. Two main sound that can be recorded by PPG are sounds resulted from vibration created by closure of atrioventricular valve close at the start of systolic and the other when the aortic valve and pulmonary valve is closing at the end of systolic [10]. In 1894, Einthoven introduce the first applied PCG method [33]. Likewise auscultation, PCG is one of the most commonly measured non-invasively from the chest, but with a microphone. The advantages of using PCG is that it is adding inaudible frequencies to the observations and also, less dependent on the person performing during a trial [33].

2.3.7 Impedance cardiography (ICG)

Impedance cardiography (ICG) uses the phenomenon that fluid content variation of the chest make variation in thoracic impedance. In other words, ICG is computing total electrical conductivity of the thorax and its variation to process important cardiac parameters. These parameters are heart rate (HR), stroke volume (SV), cardiac output (CO), pre-ejection period, ventricular ejection time (VET) [34][35].

2.3.8 Ballistocardiography (BCG) and seismocardiography (SCG)

Seismocardiography (SCG) and ballistocardiography (BCG) are non-invasive techniques to study mechanical vibrations of the body which are created by cardiovascular movements, as SCG defined by Zanetti and Salerno [36]. Put it differently, SCG shows the local vibrations of the chest wall corresponding to the heartbeat, while BCG measures the cardiac reaction forces created by blood circulation affecting the entire body [37] [38]. SCG originally goes backed to the nineteenth century, when observing a heartbeat during standing on a scale was reported by Gordon [39].

In addition to Gordon's measurements, Isaac Starr's work starts new era in non-invasive measurement of cardiac vibration [40]. Starr worked on the reaction forces of cardiovascular activities on the whole body, BCG. Then Zanetti and Salerno [36] reintroduced the SCG and BCG respectively. Zanetti introduced SCG technique to clinical applications. More up-to-date utilizations of BCG incorporate implanting the measurement of cardiac vibration into everyday matters such as chairs [41], scales [42], and beds [43].

Gradually, the physical sizes of accelerometers and sensors to measure SCG have become small. Furthermore, at the time of Zanetti's study, an acceleration sensor weight's was about 0.8 kilograms, which now is much lighter. Today's technology offers smaller and less obtrusive transducers. For this reason, many research groups are now working on SCG. SCG has received a lot of attention, as shown in Figure 7?, and it is useful in observation and measurement of cardiac time intervals [38].

Currently, a tri-axial accelerometer components that are present in all three axes can be placed on the chest wall to show specific patterns [12][14]. However, majority of studies focus on SCG in z direction. In Figure 8, direction of SCG in x, y, and z has been shown [37]. Every time that heart is contacted, cardiac vibration can be measured by a BCG and SCG waveforms. Characterizing each waveform happen by several peaks and valleys which are representing precise heart activity. Figure 9 illustrates a general ECG, head-to-foot BCG, and 3D SCG signals [37].

Modern measurement techniques, ECG, PCG, ICG, SCG, and BCG, provide comprehensive information about the cardiovascular system. The type of information that they offer are the electrical phenomena, sounds, local vibrations, impedance variations, and recoil forces produced by the functions of the heart [5]. However, there are number of open issues that still remain to be addressed regarding using SCG and BCG, according to Inan *et al.* [37]:

1) Effects of respiration, posture, right ventricle, and sensor adherence on the signal waveform/quality.

2) Identifying possible basic elements of SCG and BCG.

3) Detecting more parameters resultant from the examination of the BCG and 3D SCG signals.

4) Ways to encourage the utilization of these signs in clinical practice.

5) Reference information for both healthy and unhealthy subjects with different conditions (e.g. ages, body size).
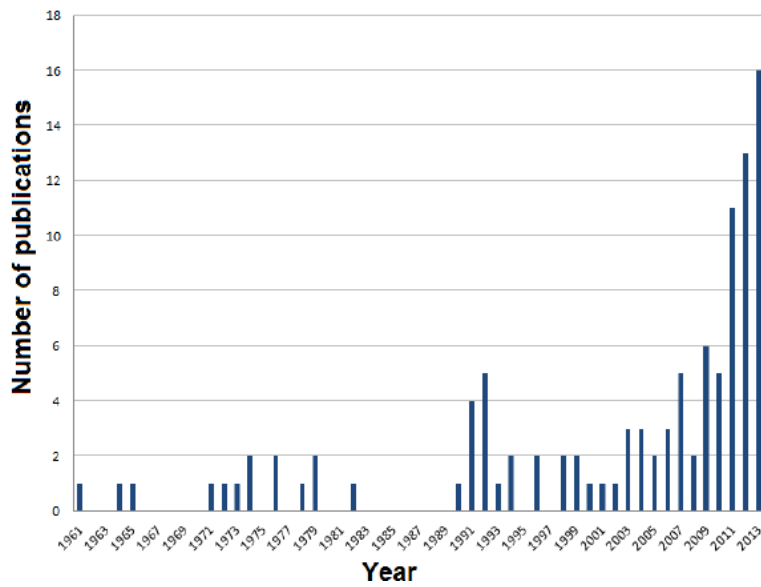
Figure 7 Number of publications on SCG between 1961 and 2013. The databases used from PubMed and Scopus [5]. The keyword "Seismocardiogram" was used in the search.
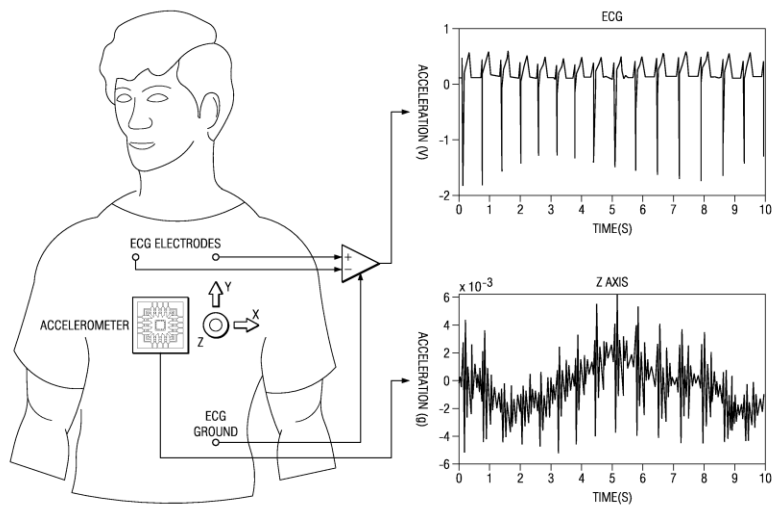
Figure 8 Seismocardiogram in three direction [44].
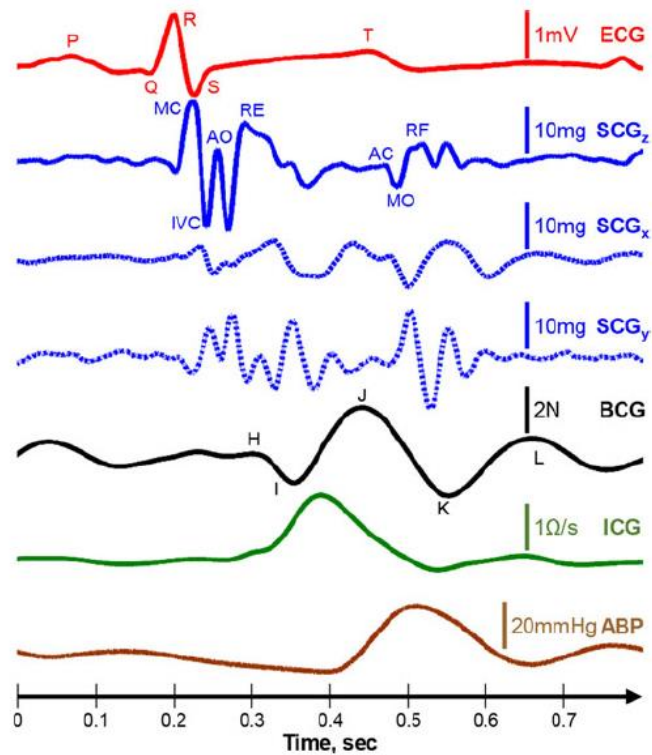
Figure 9 Simultaneously acquired Lead II of ECG, three dimension SCG signal in *z*, *x*, and *y* direction respectively, BCG signal, impedance cardiogram (ICG), and finally arterial blood pressure (ABP) measured at the finger from one subject [37].

2.3.9 Cardiac waveforms

Opening and closing the heart valves cause the cardiac sounds. Two main cardiac sounds, the first heart sound (S1) and the second heart sound (S2), can be extracted using phonocardiogram (PCG) [12].

Sudden closure of the AV valves causes cardiac vibrations and acoustic waves, which can be identified as the first sound of the heart (S1). As the Wiggers diagram, illustrated in Figure 10, immediately after QRS complex in the ECG, S1 happens. In addition, at the start of the isovolumetric relaxation phase, the aortic and pulmonic valves are closing. During this time, the S2 happens, and also partial of the T wave in the ECG [12].
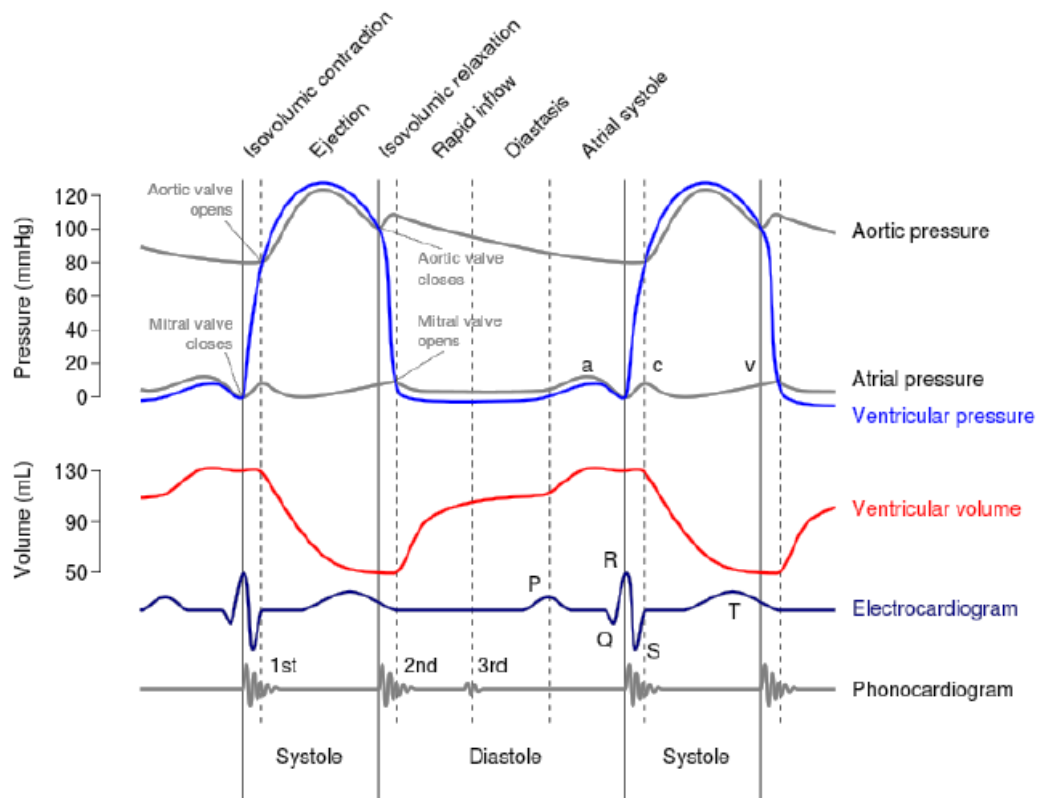
Figure 10 the Wiggers diagram of the cardiac cycles through the left ventricular contractions [37].

According to the Figure 10, the cardiac cycle events over the PCG and ECG are found very near to each other. Moreover, an upgraded adaptation of Wigger diagram which is proposed by Tavakolain *et al.* [45], suggested that cardiac events across t he SCG and ECG are recognized greatly near one another.Derived-Respiratory Techniques using Electromechanical Signals.

2.4   Derived-Respiratory Signals Using ECG and SCG Signals

2.4.1   *Electrocardiogram-derived respiratory (EDR) signal*

Numerous studies have developed techniques to derive respiratory data from the recorded ECG, which is called ECG-derived respiratory (EDR) information. Some methods are based on respiration-induced variations in beat-to-beat morphology, and some others try to derive respiratory information from the Heart rate [46].

Respiratory-affected changes in the ECG emerge because of a few components. To begin with, the electrical impedance of the thorax varies due to variation in lung volume [47]. Second, the heart vector varies because of variation in the movement and location of the heart related to the ECG electrodes [48]. Third, heart rate varies because of respiratory-induced differs to the autonomic nervous system [49]. These variables offer the morphological changes in ECG features with respect to the breathing phases [50].

The first study on the impact of breath-induced heart movement was by Einthoven *et al.* [51] and measured in more details in [52][53]. In addition, heart rate is modulated by the respiration in the same sort of increasing during inhale phase and decreasing during exhale phase [54].

Moreover, in 1974, a method based on morphologic variations was proposed by Wang *et al.* [55] This method is based on the heart mechanics related to breath volume of the lungs. Rotation of the fundamental components of the vector cardiogram (VCG) also has been used to get the

respiratory information [56]. Moody *et al.* [57] suggested a technique based on changes in the direction of the electrical axis. In addition, principal component analysis (PCA) was applied to ECG signal, and the first principal component carried the respiratory information [46].

There are three methods to extract respiratory signals from ECG in association with window; independent lead method with fixed QRS data window; independent lead method with variable QRS data window; finally, dependent lead method with variable data window [58]. Mazzanti *et al.* [59] suggested a method to select the best lead(s) from multiple ECG lead and compute the EDR signal from variation in the QRS-area. However, some studies investigate measuring the EDR signal from a single-lead EDR algorithms [60][61][62].

In addition to previous studies, respiratory signal also can be derived based on HR information [63]. Bialon *et al.* developed a robust method based on the rotation angles of the heart's electrical axis to drive respiratory information during stress testing, when ECGs contain highly nonstationary noise and exhibit changes in QRS morphology [46].

Brien *el al.* [8] provide a numerical comparison some methods to estimate respiratory signal from ECG, and the published method Behbehani *et al.* [64] used. Their results shows that estimating respiratory signal using single lead are more robust than methods based on the mean electrical axis.

In 2012, Langley *et al.* proposed an algorithm to analyze ECG morphology variation using principal component analysis (PCA). Their conclusion was that correlation and coherence of PCA method were significantly larger in comparison to the RR algorithms [65]. Figure 11 illustrates different respiratory surrogate derive from ECG signal [ref]. The third PCAs provide a better estimation of respiratory signal.

In addition to previous studies to find a proper techniques to drive respiratory signal from ECGs, Widjaja *et al.* [66] estimated respiratory signal using kernel PCA (kPCA). Orphanidou *et al.* [67] proposed a novel method for estimating respiratory rate from the ECG which fuses frequency information from the two methods, respiratory sinus arrhythmia (RSA) or the R-peak amplitude (RPA) modulation of the ECG. Labate *et al.* [68] proposed two techniques of decomposition of the ECG signal into suitable bases of functions, such as the empirical mode decomposition (EMD) and the wavelet analysis. Finally, recent study [69] utilized an adapted independent component analysis (AICA) algorithm to obtain EDR signal, and extend the normal linear PCA technique based on the best principal component (PC) selection (APCA, adapted PCA) to improve its performance further. The results yield a statistically significant improvement between reference respiration and AICA, APCA and KPCA, respectively [69].

2.4.2 *Accelerometer-derived respiratory (ADR) signal*

In addition to EDR, current sensor technology allows the progress of new kinds of portable and suitable ways to extract respiratory information, such as smart shirts, which can be an efficient process in hospitals. Accelerometer sensor is one of these sensors. Many studies took advantages of this sensor to extract cardio-respiratory information which is a great deal of information in healthcare monitoring.

Accelerometer placed on the chest changes due to the periodical movement of the thorax caused by breathing. In the rest position, the accelerometer sensor computes the acceleration of gravity and reveals these periodical variation as the respiration [71].

Castiglioni *et al.* [72] established that sternal SCG signals can be divided into different frequencies band, low frequency components (<20 Hz) which is related to the cardiac output, and higher-frequency components (>20 Hz) that is originated from the heart sounds [73].
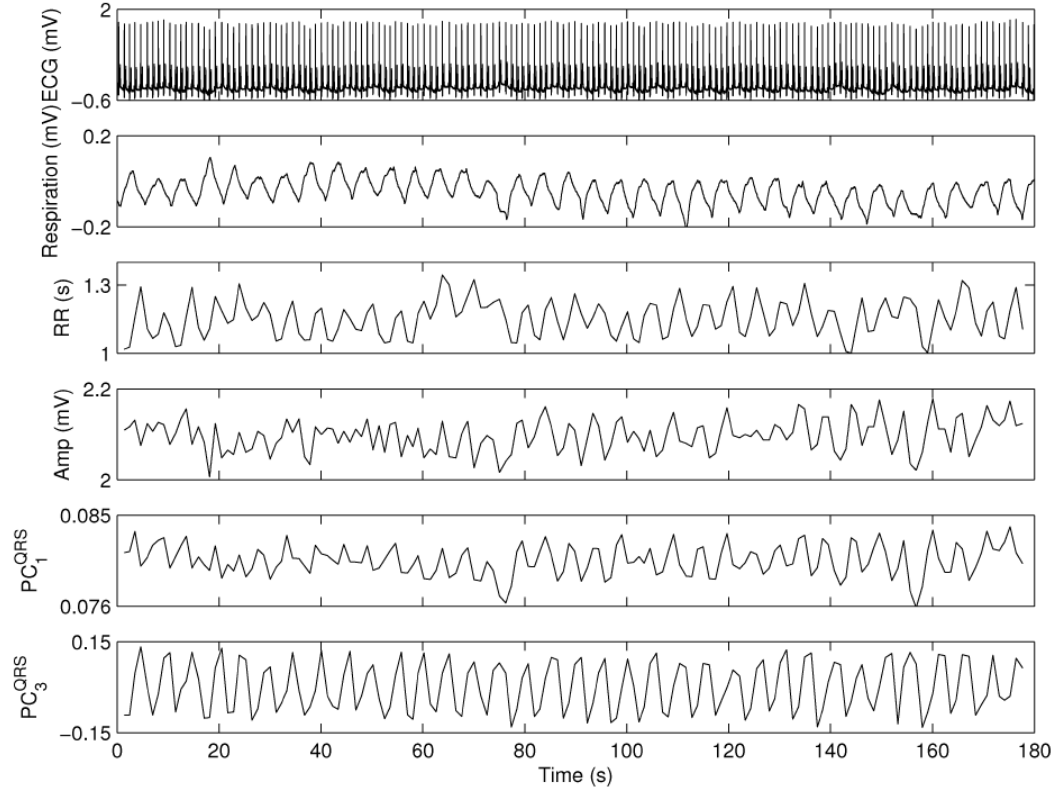
Figure 11 Respiratory surrogate derived from ECG signals [70].

The ultra-low frequency (<1 Hz) variations of the SCG signal caused by movement of the chest wall is related to the respiration information [74][75]. In 2008, Tavakolian *et al.* [76] investigated the morphological variation in the SCG signal produced by respiration in the context of improving the procedure for averaging SCG heartbeats. Reinvou *et al.* [75] investigated feasibility and reliability of a sensor belt with a high-resolution accelerometer (capacitive Microelectromechanical systems (MEMS)) and an EMFit (Electroactive Ferroelectret) pressure sensor to measure the respiratory rate. Results showed that the reliability of the MEMS-based sensors was 90%. Moreover, the results shows the importance of the position of the sensor module on the chest [75].

Hung *et al.* [71] designed several experiments in normal, apnea, deep breathing status and in vertical (sitting, standing), or horizontal (lying down) positions to find a method using SCG signal to identify some respiratory malfunction, for example during the obstructive apnea automaticall.

Their results illustrated that there was a constant phase shift between accelerometer-derived respiratory signal and the reference respiratory belt signal, as the Figure 12 shows. This phase shift is small during deep breathing while it is larger in case of normal breathing.
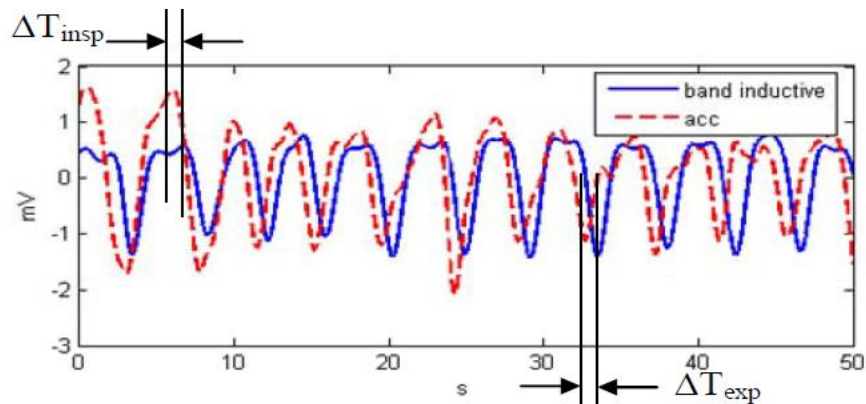
Figure 12 Phase shift between reference respiratory belt signal and accelerometer derived-respiratory signal [71].

In addition to single axis accelerometer sensor, the tri-axial accelerometer is used during the respiratory movements. Jin *et al.* [77] used tri-axial accelerometer to extract respiratory information. The results demonstrated that Hybrid-PCA, which is a combination of derived-respiratory PCA-based method and "selecting the best axis-based" techniques, performs comparable to PCA and is able to determine the respiratory rate accurately [77]. They fuse axes in order to obtain the final derived-respiratory signal, as shown in Figure 13.

There are several other studies which took advantages of accelerometer signal to extract respiratory information and derived-respiratory signal [73][78].
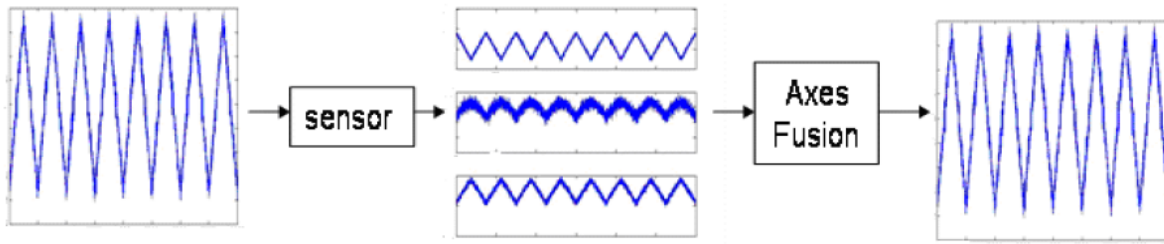
Figure 13 The simulated respiratory effort (left), its projection to the three axes of the accelerometer (middle), and reconstructed respiratory effort information by axes fusion (right) [77].

Pandia *et al.* [79] indicated that in addition to the time-domain techniques to analyze and classify respiratory variations in the SCG signal, the frequency-domain methods could also make corresponding information related to the respiratory modulation of the SCG [79]. In another study, in order to achieve more high-quality respiratory signals and to compensate the weakness of using a single accelerometer, Yoon *et al.* [80] fused results of an accelerometer with gyro sensor by using a Kalman filter. Finally, in the latest study, Vahid *et al.* used a machine learning based techniques to identify respiratory phases of heart beats using SCG. The SVM-based method shows that it could correctly identify 88% respiratory phases of heart beats in the testing data [81].

CHAPTER 3

METHODOLOGY

In this chapter, we are going to discuss human subject dataset that is used in this thesis. Then, in the signal processing section, first pre-processing methods will be discussed, and then six different methods will be introduced, which were used to derive thirty different respiratory signals from lead II of ECG signal and 3D SCG signals. In order to identify respiratory phases of heart beats, first, we need to define the beginning of inhale and exhale phase from reference respiratory belt signal and derived-respiratory signals. For this reason, two approaches were used. First, manual annotation of respiratory phases to evaluate the feasibility of detecting respiratory phases of heart beats. The second approach, which is vital for practical applications, is implementing automatic annotation of respiratory phases. The techniques utilized in this part will be discussed at the end of this chapter.

3.1   Human Subject Protocol

Twenty healthy male subjects with no known pulmonary or cardiovascular disease were recruited for this study (age 24.9±3.01, height 180.7±5.55 cm, and weight 78.9±8.82 Kg). Data from one recording from one of the subjects was not used for the analysis due to poor reference respiration belt signal. The details characteristics of each subject are shown in Table 1. The trial was performed twice for each subject, and each trial took 10 minutes with 1 KHz sampling frequency (total of 40 trials). The accelerometer sensor (Model: SCA610-C21H1A, *Murata*

Electronic) was mounted on the sternum using double-sided adhesive tape along *x*, *y*, and *z* axes. The accelerometer system was positioned about one centimeter above the xiphoid process. The respiration signal was measured using a respiration belt sensor (BN-RESP-XDCR, BIOPAC Systems Inc., US). The belt sensor was tightened around the body so that it did not resist breathing, but still it was too tight for of breathing recording.

All measurements of healthy individuals were performed at Aalto University School of Electrical Engineering, Espoo, Finland in the spirit of Helsinki declaration. The study did not contain any intervention in the physical integrity of the test subjects, or any other features, needing an ethical review as considered by the National Advisory Board on Research Ethics in Finland. A written consent was received from each subject. Overall forty recording from twenty subjects were used for this study. In each recording, the reference respiratory belt signal, 3D accelerometer, and electrocardiogram signals were recorded for ten minutes.

TABLE 1 Human subject details.

Subject number	Age	Height	Weight
1	27	170	78.5
2	20	178	82.5
3	26	183	79
4	27	186	88
5	25	177	81
6	23	178	92
7	25	182	91
8	29	187	75
9	22	181	73
10	27	182.5	82.5
11	26	176	77
12	23	187	100
13	28	186	74.5
14	23	175	68.5

15	22	181	68
16	23	170	72
17	19	177	65
18	26	181	77
19	26	188	72
20	31	188.5	82
Average	24.9	180.7	78.93
STD	3.01	5.55	8.82

TABLE 2 Respiration rate (RR) and beats per minutes (BPM) of every subject.

	Respiration Rate (RR)	**Beats per minutes (BPM)**	**Respiration Rate (RR)**	**Beats per minutes (BPM)**
Subject Number	**ref1**	**ref1**	**ref2**	**ref2**
1	7.7	56.1	7.55	61.2
2	14.8	58.4	13.55	57.3
3	-----	-----	15.9	56.4
4	12.95	57.6	11.1	55.7
5	13.6	64.1	11.7	66.6
6	10.45	60.6	14.3	64.9
7	13.65	72.8	10.45	54.5
8	14.15	45	16.95	50.1
9	11.2	74.8	12.05	77.3
10	8.25	52.9	10.55	53.7
11	6.55	44.2	6.7	46.2
12	11.9	75.7	11.25	58.7
13	7.5	52.6	6.8	56.1
14	7.5	61.6	8.15	59.2
15	7.35	56.3	8.65	62
16	17.25	75	13.4	58.8
17	13.65	69.1	11.65	74.7
18	11.85	59.5	14.4	57.5
19	14.35	53.5	13.6	56.5
20	14.4	50.5	16.9	56.7
Avg.	11.65	59.6		
STD	±3.09	±8.37		

3.2 Signal Pre-processing

In the preprocessing stage, raw signals were normalized to zero mean and standard deviation of one. Moreover, to subtract offset or trend of the signal, proper de-trending method was applied to the signals. The phase detection of reference respiratory signal was accomplished manually by two specialists, and as a result of this inspection, one trial of one of the subjects was discarded.

3.3 Electromechanical-Derived Respiratory Signals

The Signal Processing Toolbox of MATLAB™ was used for the signal processing purposes.

3.3.1 Moody's method

As discussed in the chapter 2, Moody's method was used in this study as one the common EDR signals (Appendix E). Moody's method computes EDR signal from a given single-lead ECG signal, based on the signed area under the QRS complex and computed derived-respiratory signal using is titled as EDR_Moody.

3.3.2 Principal component analysis (PCA)

PCA is known as one of the popular dimensionality reduction or manifold learning methods. PCA has been used in analyzing SCG [82], as well as estimating respiration signal from electrocardiogram signal in many studies using linear or non-linear kernel [83]. This method aims to illustrate as much as of the variance in the data as possible by using only a few principal components. In this study, first principal component was used. To derive respiratory signal from electrocardiogram using the principal component algorithm, first, heart beats were segmented utilizing the R peaks (Appendix D). In the next step, heart beats were collected in the form of $m \times n$

matrix, where *m* is the number of beats and *n* is the number of samples per beat. In the final step, principal component analysis was applied to the matrix to produce m principal component [83].

In order to evaluate how many principal components is needed to derive respiratory signals, all the principal components is plotted which is shown in the figure 14. According to the figure 14, only principal component is enough to derive respiratory from ECG and SCG signals, since first component illustrates general shape of one heart cycle. Total 12 different EDR and ADR signals were computed using PCA method with linear kernel and nonlinear kernel such as polynomial and Gaussian kernels. For instance, deriving respiratory signal from SCG_z using PCA with Gaussian kernel is called *ADR_z_Gauss_PCA* signal.
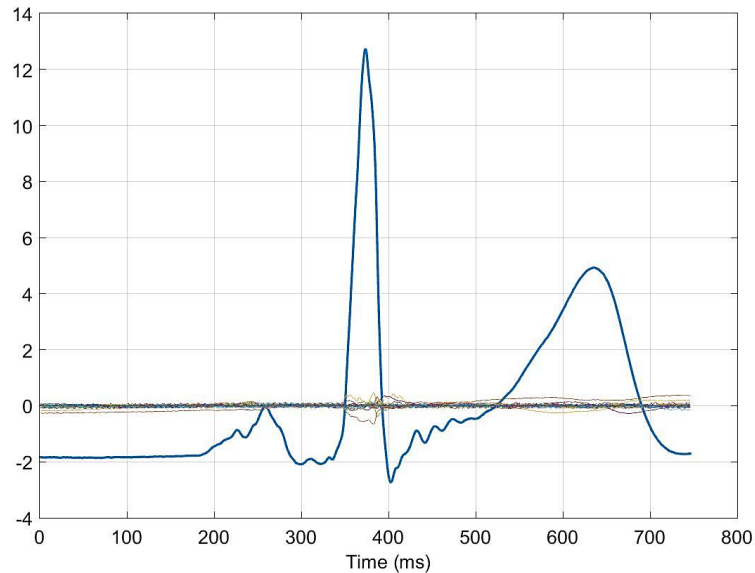
Figure 14 principal components.

3.3.3 Envelope detection

Upper and lower envelopes electrocardiogram and seismocardiogram data were calculated based on moving time window secant method. The secant method is a numerical technique that can be used to find the maximum slope within each time window [84]. In this method, time window

size was the control parameter and was set to a value which covered dynamic range for respiration range of adult human. Two second windowing was used for envelope detection.

An envelope of the signal mostly corresponded to its respiration component. ADR was obtained by applying envelope detection method to SCG. Figure 15(a) shows the reference respiration signal. Figure 15(b) illustrates the SCG signals and its corresponding upper and lower envelope. By applying PCA or envelope detection method to ECG and SCG signals, ADR and EDR were obtained. Figure 16 shows respiration, ECG, SCG and estimated signals. Total 12 different derived-respiratory signals were calculated using lead II of ECG and 3D SCG signals. For example, estimating respiratory signal from upper envelope of SCG_x is called *ADR_x_upper_Env* signal. In addition to single axis ADR, also ADR in z and x direction were fused using equation (2); moreover, ADR in x, y, and z direction were fused using equation (3) to gain *ADR_x*z and *ADR_xyz* signals. Appendix C contains codes to compute lower and upper envelope of a given signal.

$$ADR_{xz} = \sqrt{ADR_x^2 + ADR_z^2} \tag{2}$$

$$ADR_{xyz} = \sqrt{ADR_x^2 + ADR_y^2 + ADR_z^2} \tag{3}$$

(a)
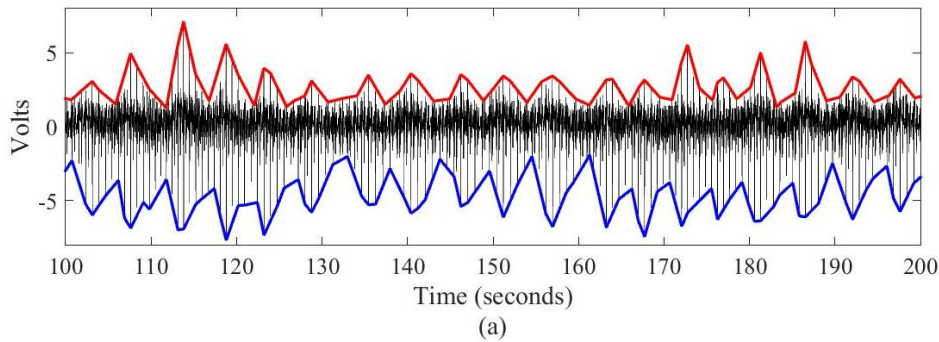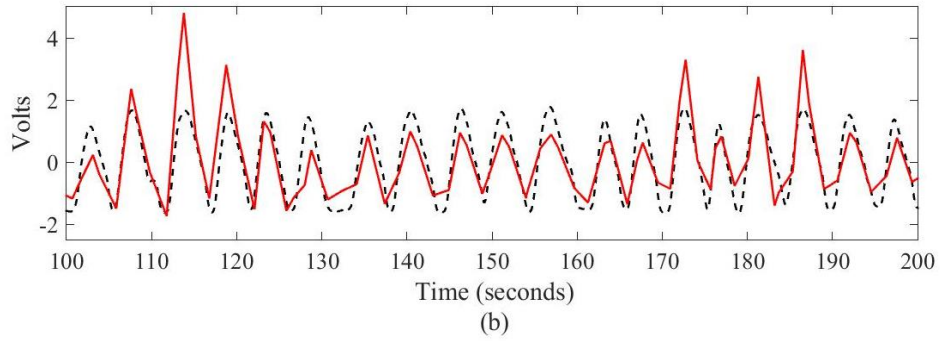
Figure 15 (a) SCG signal with its upper and lower envelope, (b) dashed line is reference respiratory signal and solid line is upper envelope of SCG signal.
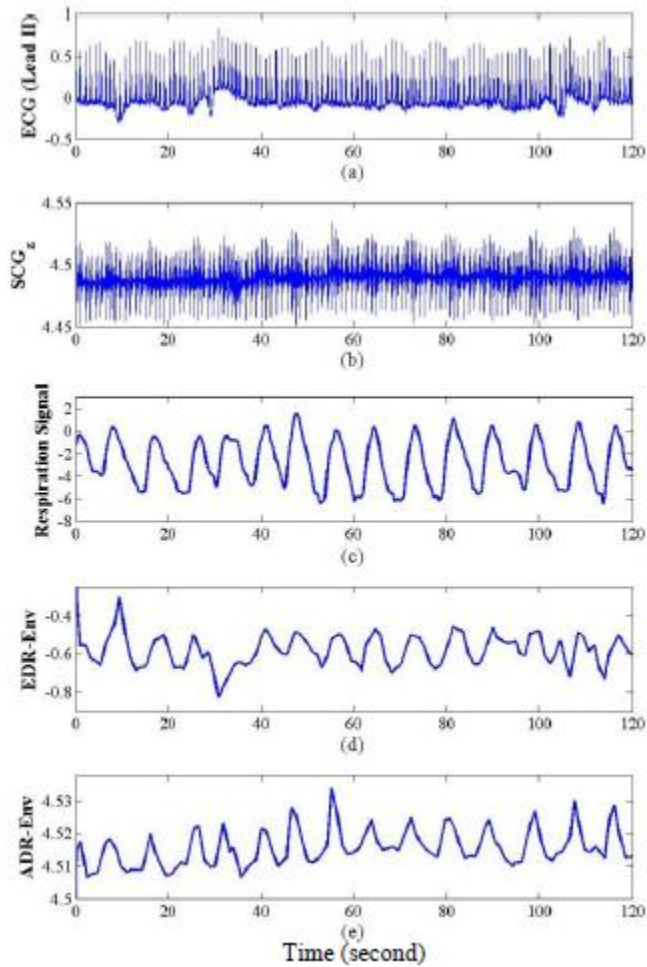
Figure 16 (a) Representation of ECG (Lead II), (b) SCG in z direction, (c) respiration signal, (d) ECG signal after applying envelope detection method (EDR), and (e) ADR signal.

3.3.4 Derived-respiratory signals through R-peak amplitude interpolation

Another method to derived-respiratory signals from ECG signal is through interpolation of the amplitude values of the R-peaks, since the ECG is modulated as an effect of respiration. This process creates two discrete-indexed vectors: one is containing R-peak occurrence times and the other is the R-peaks amplitude. The obtained derived-respiratory signal is called EDR_r_amp.

3.3.5 Derived-respiratory signals through Ao-peak amplitude interpolation

Since the SCG is modulated as an effect of respiration, another method to derive respiratory signals from SCG signal is through interpolation of the amplitude values of the Ao peaks. This technique is very similar to the previous method, the difference is that instead of using amplitude of R-peaks in ECG signal, amplitude of Ao-peaks from SCG signals in in three different direction of x, y, and z were used. To compute ADR_Ao_x, ADR_Ao_y and ADR_Ao_z, first a fixed window size of 300 millisecond was defined around each R-peak of ECG signal (150 ms before and after each R-peak). Then, all the Ao-peaks were detected inside the defined windows. Final stage was, performing linear interpolation through Ao-peak amplitudes. Table 3 listed all the twenty-nine estimated respiratory signals.

3.4 Manual Annotation: Respiratory phase detection

In order to assess the possibility of detecting the timings of respiration phases, the reference respiration signal and the estimated signals were manually annotated. The challenge of this process is that maxima and minima of the respiratory signal were not necessarily corresponding to the beginning of the exhale and inhale phases, respectively. Appendix A contains all the MATLAB coeds related to the derived-respiratory signals using ECG and SCG signals.

TABLE 3 List of derived respiratory signals from SCG signals in *x*, *y*, and *z* directions, and ECG signal.

index	Signal	index	Signal
1	EDR_Moody	16	ADR_y_Poly_PCA
2	EDR_r_amp	17	ADR_z_Poly_PCA
3	ADR_Ao_x	18	EDR_lower_Env
4	ADR_Ao_y	19	EDR_upper_Env
5	ADR_Ao_z	20	ADR_x_lower_Env
6	EDR_lin_PCA	21	ADR_x_upper_Env
7	EDR_Gauss_PCA	22	ADR_y_lower_Env
8	EDR_Poly_PCA	23	ADR_y_upper_Env
9	ADR_x_lin_PCA	24	ADR_z_lower_Env
10	ADR_y_lin_PCA	25	ADR_z_upper_Env
11	ADR_z_lin_PCA	26	ADR_xz_lower_Env
12	ADR_x_Gauss_PCA	27	ADR_xz_upper_Env
13	ADR_y_Gauss_PCA	28	ADR_xyz_lower_Env
14	ADR_z_Gauss_PCA	29	ADR_xyz_upper_Env
15	ADR_x_Poly_PCA		

Another challenging feature of this process is the existence of multiple local minima and maxima. Therefore, a simple algorithm is not adequate. For this reason, the annotations were done using "find peaks" MATLAB function to detect beginning of the inhale and exhale phases. All the detected points were then visually inspected. Those that were not detected by the "findpeak" function, manually were added manually. The annotation of estimated respiratory signals (ADR and EDR) was done in absence of original respiration signal (reference signal) to avoid any biased

detection. Approximately, a total of 2,391 respiration phases were manually annotated from the reference signal.

Figure 17, illustrates the respiratory phases of detection from reference respiratory signal, and ADR$_z$ signal using the envelope method. Based on visual observation, envelope detection method is selected as a proper technique to derive the respiratory signal from SCG signals.
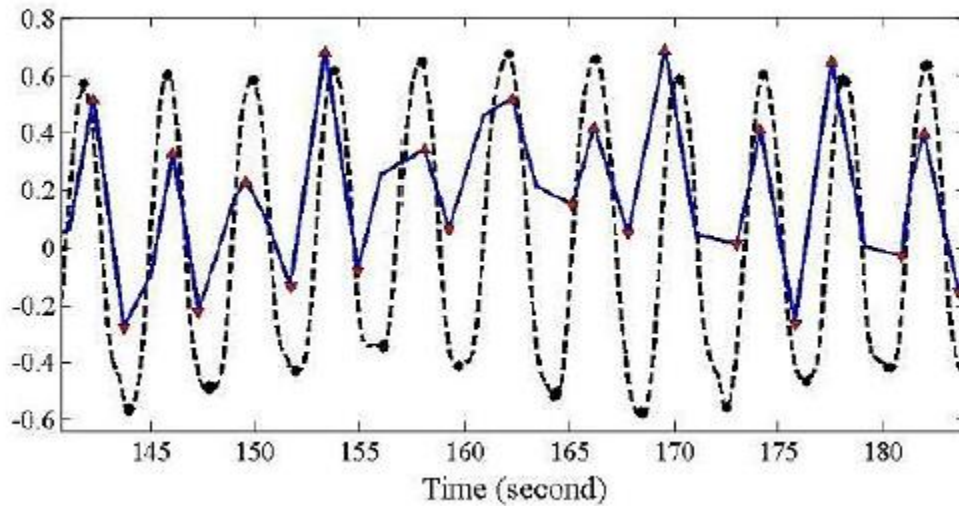
Figure 17 Respiratory phases detection. Dashed and solid signals illustrate the reference respiration signal and the ADR$_z$ using the envelope detection method respectively.

3.5   Automatic annotation: Quality Estimation of Signals

3.5.1   Total harmonic distortion (THD)

To find respiratory phases of the heart cycle, either the upper or lower envelopes of the SCG signal can be used. However, the quality of lower envelope and upper envelope are not the same. For instance, lower envelope of the SCG signal in a subject can demonstrate a decent correlation with reference respiratory signal; whereas, upper envelope may have poor correlation.

34

The correlation between the respiratory signal and SCG upper/lower envelopes may vary in different subjects. To select the most correlated envelope (upper/lower) to the reference respiration signal, the total harmonic distortion (THD) factor has been used as a quality metric to choose between upper or lower envelope of SCG signal.

In this study, many methods and factors have been used, such as norm, signal to noise ratio (SNR), dynamic range, dynamic time wrapping (DTW), respiratory rate, cross correlation, and total harmonic distortion (THD). Amongst all, THD factor has shown acceptable results. Therefore, THD has been used as a quality metric to evaluate quality of lower and upper envelopes of SCG signals, which are denoted as respiratory signals ADR_{lower} and ADR_{Upper}, respectively.

THD of a signal is defined as the ratio of the sum of all harmonic components to the power of fundamental frequency [85]. The THD concept was first used to characterize the quality of audio signals. For a given signal y the THD is defined by the following equation (1):

$$THD = \frac{\sum_{2}^{\infty} harmonics}{Fundamental} = 20 \log(\sqrt{\frac{\sum_{n=2}^{\infty}|c_n|^2}{|c_1|^2}})\qquad(1)$$

where c_n represents the harmonic content in an alternating signal, and c_1 represents the fundamental frequency of the signal. First five harmonics utilizing a modified periodogram of the same length as the input signal was used. In bio-signals the lower THD means that the processed signal is a more accurate reproduction of the recorded signal [86]. Therefore, the curve with lower THD is more desirable.

3.5.2 *Machine Learning Approach*

Another method that has been used as a quality metric to select proper estimated respiratory signal is a method based on machine learning techniques. In this morphological-based application, the quality of estimated respiratory signals can be deteriorated due to body movements and other

artifacts; therefore, the derived respiration signals from 3D accelerometer and ECG sensors can vary from subject to subject. It was shown in the previous section that there is no solid feature as a quality metric that always can detect the best-estimated respiration signal in each segment of the recorded data. Therefore, we are investigating the possibility of using a properly trained classification method to classify the quality of estimated respiratory signals in each segment based on extraction of some specific features.

3.5.2.1 Data segment length

The first step is determining length of signal that need to estimated its quality. For this reason, the performance of respiratory heart identification of heart cycles was computed for various segment lengths of the each derived-respiratory signal. Segment sizes of thirty seconds, one minute, two minutes, 5 minutes and the 10 minutes (the complete data length) were been examined to determine the proper segment length. Thirty nine recordings from twenty subjects were used for this study. This means the total number of signals that need to determine their quality is 39 (recording) $\times$ 5 (two-minute segment length data) $\times$ 29 (estimated respiratory signal) = 5655. The reason that two minute data segments were used, will be described in the result section.

3.5.2.2 Labeling

Accuracy of respiratory heart cycle detection was used to label all the 5655 signals to either zero or one. In other words, this is a binary classification. If the accuracy of respiratory phase identification of heart cycles in a signal is greater than 70%, that signal was defined as label 1; otherwise its label is zero. By utilizing this procedure, six electromechanical-derived respiratory signals were selected among twenty-nine signals for further processing. In other words, six signals were recognized as robust signals which their accuracy were higher than other signals. Now to estimate quality of these six selected derived-respiratory signal in each thirty-nine recording, the

36

proposed machine learning approach was used. Different steps of this techniques is described in the figure 18.
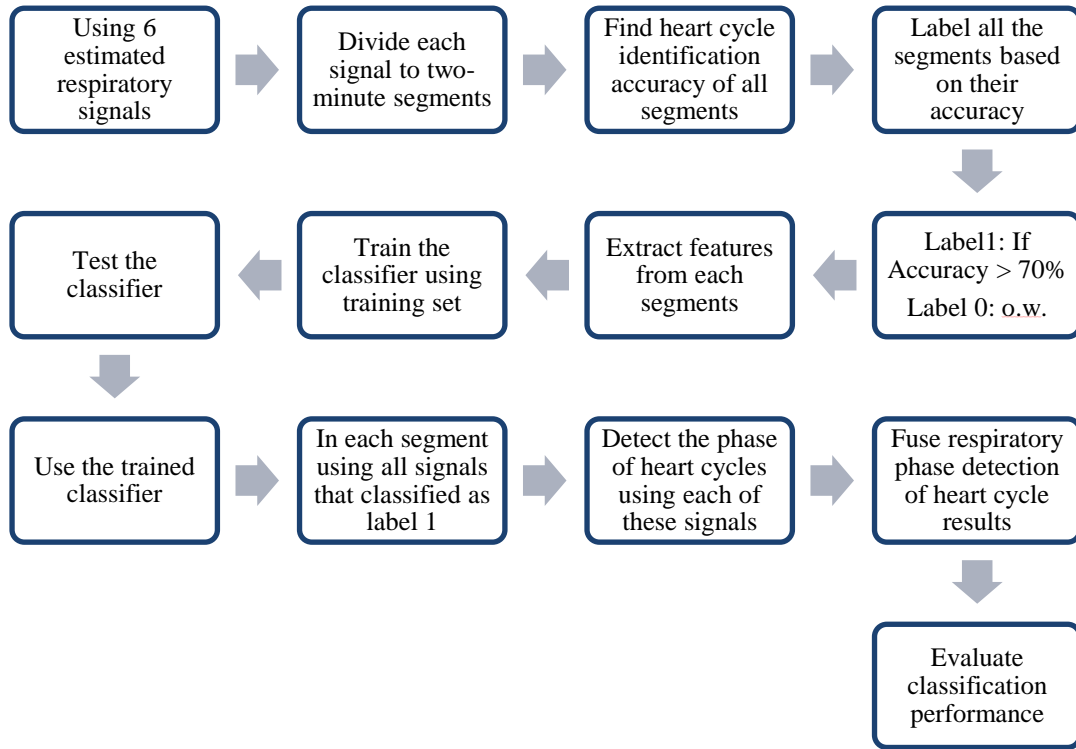
Figure 18 Block diagram of proposed method using machine learning as a quality factor finder of estimated respiratory signals.

3.5.2.3 Feature extraction and classification

Several physiological-related features in time and frequency domains, including respiratory rate (RR), dynamic range, total harmonic distortion, and power spectral density, were extracted from each signal. Then support vector machine (SVM) was used as a multilevel signal quality classification algorithm to distinguish these signals.

3.6 Automatic Annotation: Respiratory Phase Detection

In Section 3.4, a manual annotation method was used to detect start of the inhale and exhale phases in reference respiratory, EDR, and ADR signals. However, in practice, where the

respiration signal is recorded for the minimum of 2 minutes, an automated phase detector is needed. Therefore, in this study, beginning of inhale and exhale phases of derived-respiratory signals were estimated automatically.

Before phase detection, time lag of four signals, subject 17, 18, 19 in the first set of recordings, and subject 11 in the second set of recordings were computed with respect to the reference respiratory signal. Subsequently, the computed signal was shifted to be synchronized by the reference respiration. This was done because these signals had a constant delay in comparison to the reference respiratory signal. This delay might have happened due to not accurate synchronization process.

In the first step, a MATLAB built-in function for automated annotation was used to find the maximum and minimum of the estimated respiratory signals. The peaks in a respiratory signal represents the start of the exhale phases and valleys represents start of inhale phases. In some cases, the local peaks/valley were also detected, which affected the accuracy of the analysis. Therefore, to improve the accuracy of the correlation, two criteria were used to remove the unwanted peak/valley points as follows:

- Ignoring the peaks/valley points that were very close to each other. The acceptable peak-to-peak separations were restricted to the values that are greater than N seconds (this value is equal to the window used to detect the envelope of SCG signal). In other words, the highest peaks in the signal that were separated by at least N seconds were chosen and all the peaks with less than N seconds separation were eliminated.

- Removing signals that do not drop by M% of their maximum. Signals must decrease to at least M% of the peak on each side before the signal starts to increase again, Otherwise, the signal was removed.

In this thesis, based on the data set that is used, values of 2.0 and 40 were chosen for N and M parameter respectively. By using these two criteria, many local minimum and maximum points are eliminated.

In the second step, the flat peaks are corrected. In some cases, peaks are flat, where the peak value is repeated a few times. In this case, the starting point of the flat part of the signal is selected as a peak. To accomplish this process, the amplitude of all detected peaks was compared to their adjacent points. If the amplitude difference between two consecutive points is less than 0.1% (1 mV in this work), then first detected point will be considered as the peak. This procedure will continue until the difference between amplitudes of all adjacent peaks is higher than 1 mV.

To define the respiratory phases of heart cycles, two points are needed: (1) the beginning of inhale phase (or end of exhale phase) and, (2) end of inhale phase (or beginning of exhale phase). Sometimes, more than one valley is detected between peak-peak interval due to the morphology of estimated respiratory signal. Therefore, the third step is needed for valley point's correction. To take the corrective action, one technique is to estimate the ratio of involuntary exhale duration versus involuntary inhale duration. In a rhythmic breathing pattern, this ratio is less than 2 and greater than 1; also in athletes this ratio could increase to 2 [87].

All the peak-peak intervals were divided to three equal subinterval. In this procedure, based on the breathing physiology, the first priority is to find the points that are located in the middle segments, where the breathing ratio is more than 1 and less than 2. The second priority is to find valley in the last interval, where breathing ratio is more than 2. Finally, if we couldn't find any valleys in those intervals, valleys of the first interval (breathing ratio of less than 1) will be considered. If in each interval, more than one valley were detected, mean of those valleys were

considered as the valley point. If no valleys were found between a specific peak-to-peak intervals, then the second peak was removed. Figure 19 illustrated this procedure.
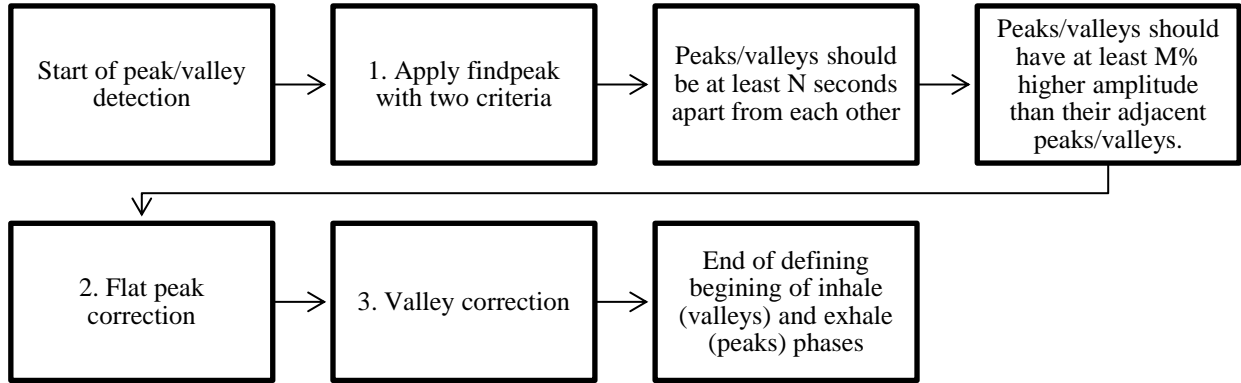
Figure 19 Block diagram of automatic respiratory phase detection.

3.7   Respiratory Phase Identification of Heart Cycles

After finding the beginning of the inhale and exhale phases manually or automatically, the heart cycles that are happening during the inhale and exhale phases must be identified. To achieve this, first R peaks of all heart cycles were detected using the simultaneously recorded lead II of ECG signal. In this procedure, the R peaks are selected as the surrogate of the complete heart cycles. The reason for this selection is that if a heart cycle lies between the inhale and exhale phases, the decision on the respiratory phase identification should be made based on the location of R peak.

By continuing this procedure for all the other peaks and valleys, all the heart cycles will be labeled to either inhale or exhale phase. The peaks and valleys of reference respiratory signal were used to mark heart cycles as true labels. Finally, the label of heart cycles using derived-respiratory signal was compared to true labels, and accuracy of respiratory phase detection of heart beats was computed. Appendix B include codes related to this section.

40

CHAPTER 4

RESULTS AND DISCUSSION

In this chapter, the results were divided to results of manual and automatic annotations of respiratory phases, respectively. In the manual annotation section, results of detecting beginning of inhale and exhale phases from each derived-respiratory signal, the effect of window size around each phase of reference respiratory signal and the average accuracy of discrimination of heart cycles to inhale and exhale phases were described. In the last section of the manual annotation, results of fusing ADR and EDR signals, in decision level, were defined. Fusion were used in order to improve the performance of detecting the beginning of respiratory phases. These results consider ideal case, since fusion is performed in the presence of a reference respiratory signal; while, in the real application, all the procedure should be in the absence of reference signal. Therefore, for a practical reasons, we need an accurate automatic technique. Results of the proposed automatic method are described after results of manual annotation.

4.1 Manual Annotation

4.1.1 Respiratory phase detection

Respiratory phases (inhale and exhale) were identified in ADR and EDR signals by use of respiratory signal recorded using a chest band strain gauge. We defined a window around peaks and valleys of the reference signal, to find their corresponding on estimated signals. Depending on the width of this window, different detection accuracies were obtained. The "true positive" was defined as the number of peaks and valleys that were detected from estimated signal over total number of inhale and exhale phases in the reference respiration signal. Likewise, the "false negative" was defined as the ratio of number of start points that estimated signal did not recognized them over total number of inhale and exhale phases in the reference respiration signal.

In this study we computed "sensitivity" for each subject for different window sizes to find a proper window size. Window sizes of 1.0, 1.5, 2.0, 2.5, 3.0, 3.5 seconds were examined. Average sensitivity for first recording of nineteen subjects (subject 1-20 except subject 3), is shown in the figure 20.
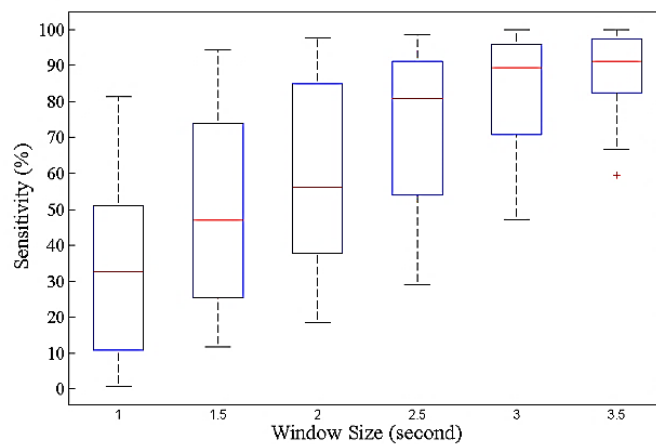
Figure 20 Effect of window size on the respiratory phase detection from ADR signal in *z* direction using envelope detection.

42

4.1.2 Timing analysis

Another factor to evaluate the performance of ADR and EDR is the error in timing of ADR driven breath cycles (inhale and exhale) in comparison to the breath cycles extracted using the respiration signal (respiratory signal recorded using a chest band strain gauge). In other words, how much of a typical breath cycle was wrongly associated to the adjacent breath cycle.

Average heart rate, and breathing rate for all nineteen subjects were 60 and 12 per minute, respectively. Therefore on average there were five heartbeats per respiration cycle for the nineteen subjects. A typical breath cycle, corresponding cardiac cycles and time difference between reference peaks and detected peaks, from ADR signal, is shown in the figure 21.
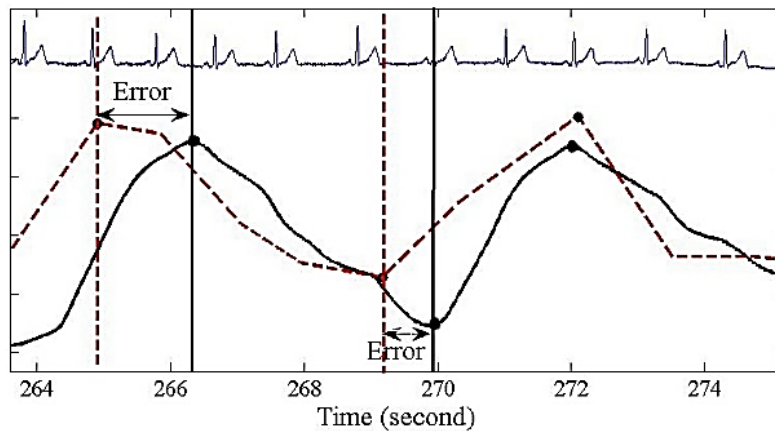
Figure 21 Cardiac cycles in each respiration phase. The solid line is the reference respiration signal and dashed line is the ADR signal.

Timing analysis for the 3.0 second window size (1.5 second before and after the peak or valley point) was performed to establish truly identified (reliable) portion of breath cycles in the ADR, in z-direction. The reliable percentage of respiration phases were about 87.8% for the exhale phases, and 86.1% for the inhale phases.

We also computed the accuracy of labeling heart cycles of the inhale or the exhale classes using estimated respiratory signals (ADR, EDR using envelope method). As table 4 shows both

43

up and down envelope methods were applied to the signals. Moreover, to compare our approach to previous studies [82], PCA method was performed for EDR, once with negative polarity and once with positive polarity. EDR signal with negative polarity is the inverted form of regular EDR signal using PCA method.

Best result between up and down envelope was chosen for each subject, and then the average of these results was reported in the last column of table 4. Likewise, for EDR using PCA method, last column is showing the best result of positive and negative polarity.

The reference respiration signal and the derived signals were hand annotated for respiration phase detection. Based on our initial observation, envelope detection method is a proper method to derive respiratory signal in addition to PCA method. Moreover; ADR can be an alternative way to detect and estimate breath cycles.

TABLE 4 Average accuracy of discrimination of heart cycles to inhale and exhale phases in respiratory signal.

Estimated Signal	Up Envelope	Down Envelope	Best Result
ADR_z	61.1%	69.9%	76.0%
EDR	66.7%	46.1%	67.2%
	Positive Polarity	Negative Polarity	Best Result
EDR_PCA	56.9%	59.1%	61.3%

The initial results suggest that ADR in z direction, using envelope method with a window size of 3.0 seconds has the sensitivity of about 85% for detection of breath cycles. About 87% of breath cycles extracted from ADR had acceptable time lag compared to the ground truth (respiratory signal recorded using a chest band strain gauge). This means that if on average 6% of initial and 6% of final part of each detected respiration phase segment were removed the remaining is correctly assigned to the right respiration phase.

In addition, according to the table 4, ADR in *z* direction using envelope method achieved 76% accuracy to discriminate heart cycles to the inhale or exhale phases. Among data from 19 subjects, three of them showed poor performance in this classification. By excluding these three subjects an accuracy of about 80% can be achieved. In these three subjects we saw irregular time interval patterns and dynamic range was low. This could be due to the location of sensors or anatomy of subjects, since EDR signal using envelope method had better performance for these three subjects. Thus, fusion of ADR and EDR signals can be a solution for low amplitude signals, such as those from these subjects. We also can conclude that if the dynamic range of estimated signal is lower than a specific value (threshold), the performance will be unsatisfactory.

The next step was to fuse ADR and EDR signals in decision level to improve performance of detecting beginning of inhale or exhale phases.

4.1.3   Fusion of ADR and EDR Signals

In this techniques, to detect respiratory phases and find their corresponding peaks of estimated signals, we defined a window around each peak and valley of the reference signal. Depending on the width of this window, different detection accuracies were obtained. To fuse outputs (detected phases) of estimated signals, and to improve the overall robustness of the results, we defined four labels for four different conditions of phase estimation:

- Label 3: when both ADR and EDR detected a respiratory phase inside the defined window;

- Label 2: when only EDR detected the phase;

- Label 1: when phase was found only from ADR;

- Label 0: when neither the ADR nor EDR signals could estimate the respiration phase.

Table 5, shows the mean accuracy of phase detection with 3.0 second window size (1.5 second before and after the start of a phase) for four different conditions.

Mean voting fusion, one of the most commonly-used methods for fusing different physiological signals [88], was used when the phase detection condition was labeled 3. In other words, if both ADR and EDR have a peak with distance of less than 1.5 second (3.00 window size) with a reference peak, which this condition is defined as label 3, we performed mean voting fusion to the estimated peaks from ADR and EDR. If only one of the ADR or EDR signals had the desired peak (a peak with distance of less than 1.5 second with a reference peak), which is defined as label 2 or label 1, that single peak was considered as the detected peak.

TABLE 5 Mean performance accuracy for four different phase detection.

Respiratory Phase	Label 3	Label 2	Label 1	Label 0
Exhale (%)	73.05	12.64	12.67	1.63
Inhale (%)	66.38	17.27	12.81	3.54

Consequently, sensitivity of the phase detection was computed for each subject. Table 6, shows the mean sensitivity of phase detection using EDR, ADR, and fusing output of ADR and EDR signals.

TABLE 6 Sensitivity of respiratory phases detection in individual estimated respiratory signals and fusion of them.

Respiratory Phase	ADR	EDR	Fusion of ADR and EDR
Exhale (%)	85.94	85.93	98.37
Inhale (%)	82.36	83.75	96.46

As shown in the table 5, the mean sensitivity of respiratory phase detection by both ADR and EDR is 73.05% for exhale phase and 66.38% for the inhale phases. The sensitivity when only one of the estimated respiratory signals could detect the phase, was about 12.6% for the exhale phases.

This indicates that capability of ADR and EDR to estimate respiratory signal and detect inhale and exhale phases is similar. Therefore, to perform weighted averaging fusing to phases which were detected from ADR or EDR, weights for both should be 0.5. As table 6 illustrates, results of fusion of estimated signals outperformed the individual ADR or EDR signal.

Since the reference respiratory belt signal was used in detection procedure, capability of fusion of ADR and EDR signals evaluated in ideal case. The main goal was to find a proper surrogate to detect the respiratory phases (inhale and exhale phases). Therefore, our next step is evaluation of fusing estimated signals in absent of reference respiratory signal by using proper quality estimator of estimated respiratory signals.

4.2   Automatic Annotation

4.2.1   Total harmonic distortion

As explained in the Chapter 3, total harmonic distortion was utilized as a quality metric of envelope detection. This parameter is used in two different ways: global THD, and piecewise THD. In global THD, THD of whole ADR signal was computed, and in piecewise THD, in each 1-minute segment of ADR, THD was calculated. Then the average of all values was used as the quality metric. The accuracy of piecewise THD in the selection of correct envelope of SCG signal was 84.6% while the accuracy of global THD was 71.8%. Therefore, piecewise THD was used to further processing.

In five subjects, subject 10, 11, and 13 in the first trial and subject 7, and 17 in the second trial, ADR signals were not well correlated to the reference respiratory. In other words, their phase identification results were lower than chance level (50%). This caused to have an overall lower accuracy for respiratory phase identification. As shown in figure 22, to remove these subjects

before further processing, the values of piecewise THD in all subjects were observed. Resutls shows THD values of these signals was lower or equal to -5.0 in either ADR_{lower} or ADR_{Upper} signal. We used this threshold to remove undesired signals.
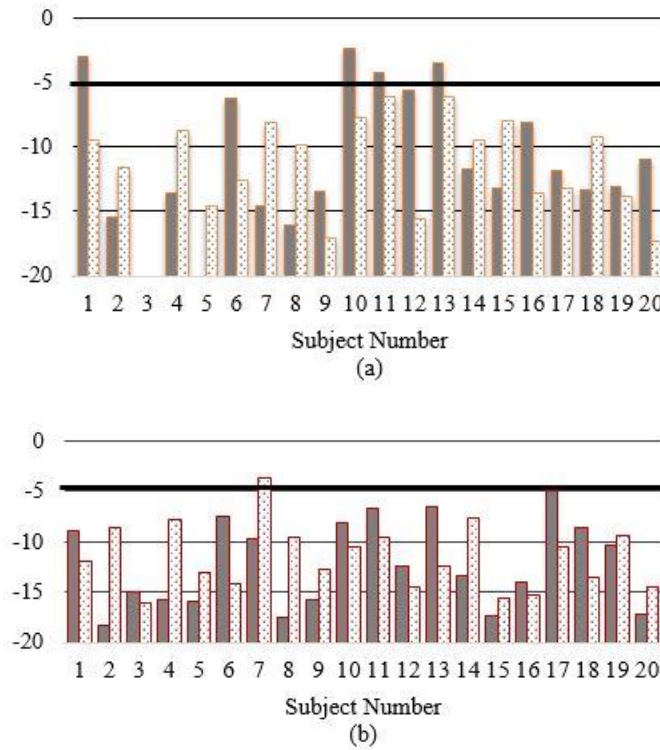
Figure 22 Piecewise total harmonic distortion. The pattern dotted color is ADR_{lower} and the solid color is ADR_{Upper}. (a) THD values in all subjects in the first recording, (b) THD values in the second recording.

4.2.2 *Machine learning approach*

Another approach to estimate quality of derived-respiratory signal in each recording of each subject, is to use a proper classification method. To reduce the number of estimated respiratory signal from 29 to 6 signals in each segment, performance of respiratory phase identification of heart beats of all estimated signals were evaluated in each segment and the maximum accuracy in each segment was calculated. The histogram of all the maximum accuracies and label of maximum

accuracies that shows their corresponding estimated respiratory signals, are illustrated for different segment length of the data in the Figures 23-26.

<div align="center">(a)</div>

<div align="center">(b)</div>
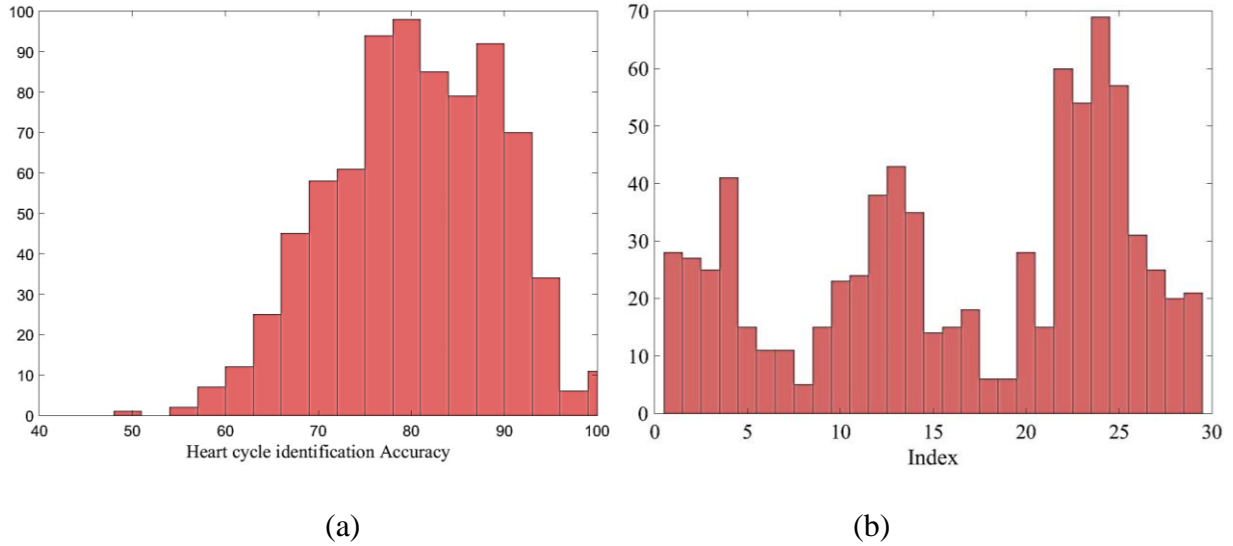
Figure 23 (a) Maximum accuracy of respiratory phase identification of heart cycles in 30 seconds data segments, (b) index of signals that has maximum accuracy in part (a).

<div align="center">(a)</div>

<div align="center">(b)</div>
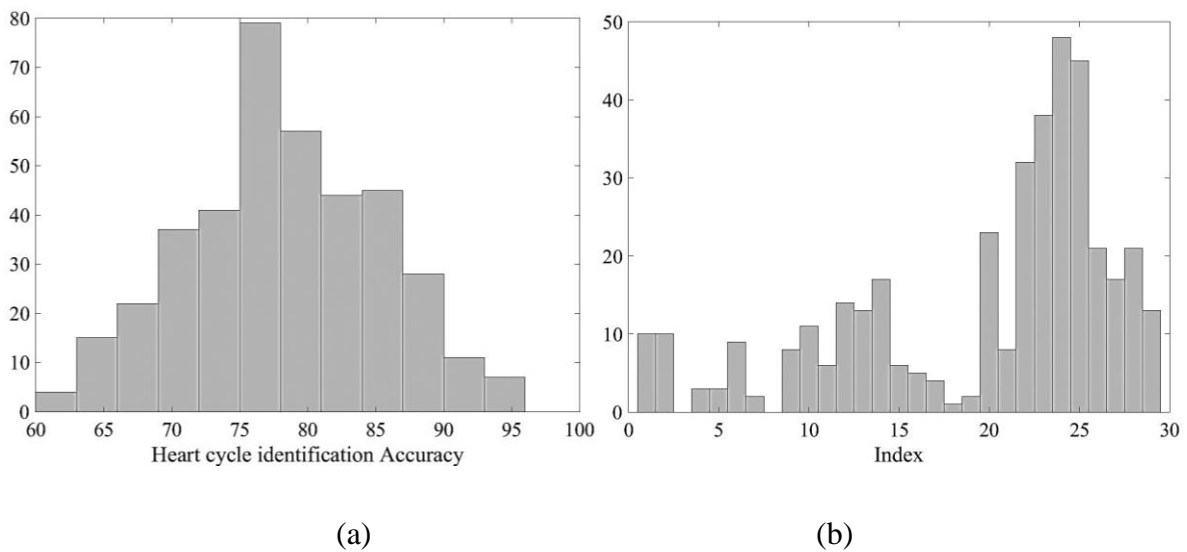
Figure 24 (a) Maximum accuracy of respiratory phase identification of heart cycles in one minute data segments, (b) index of signals that has maximum accuracy in part (a).

(a)   (b)
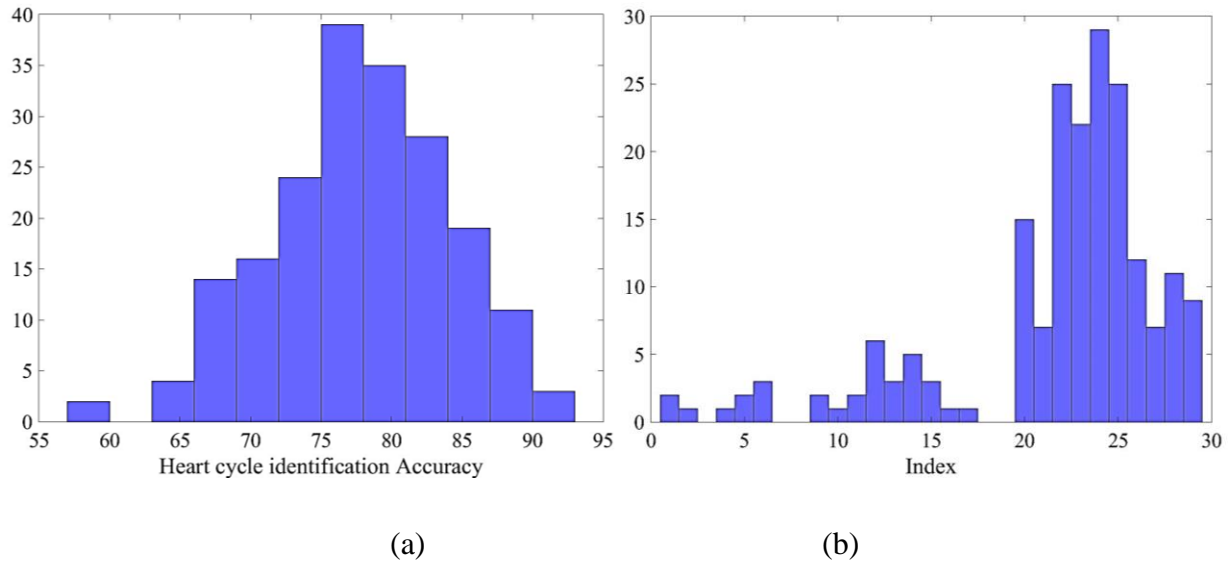
Figure 25 (a) Maximum accuracy of respiratory phase identification of heart cycles in two minutes data

segments, (b) index of signals that has maximum accuracy in part (a).

(a)   (b)
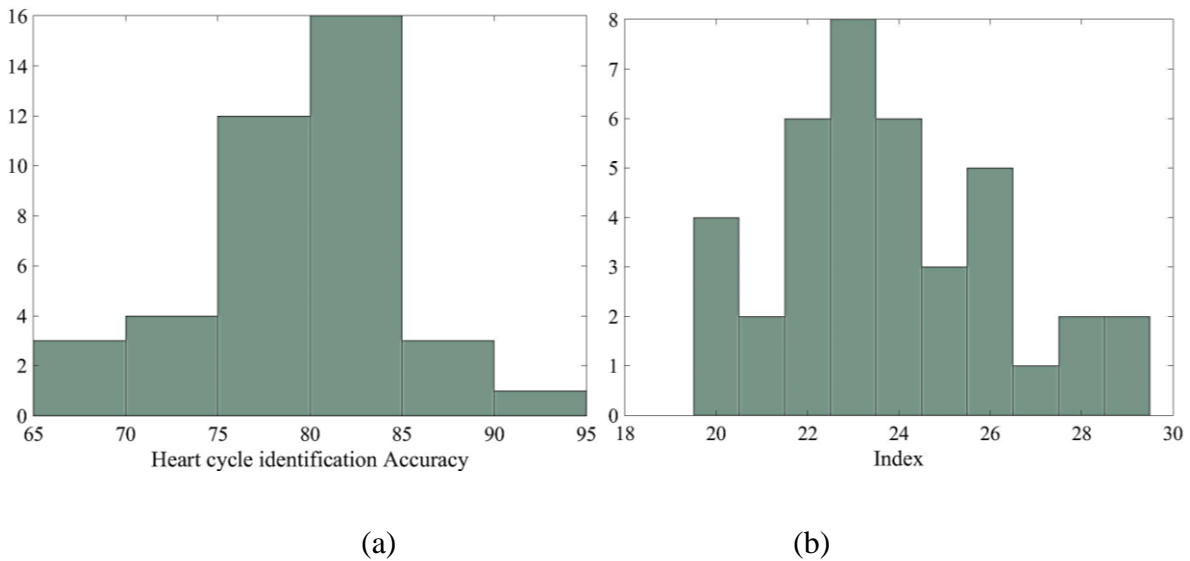
Figure 26 (a) Maximum accuracy of respiratory phase identification of heart cycles in ten minutes data (the

whole data), (b) index of signals that has maximum accuracy in part (a).

As can be seen form the figures 23-27, the minimum segment length required to find six robust derived-respiratory signal, is two minutes and the six robust derived-respiratory signals were shown in the Table 7.

TABLE 7 Selected derived-respiratory signals for further processing.

Selected derived-respiratory signal	Index
ADR_y_lower_Env	22
ADR_y_upper_Env	23
ADR_z_lower_Env	24
ADR_z_upper_Env	25
ADR_x_lower_Env	20
ADR_xz_lower_Env	26

In an ideal case, results of classification of the selected six signals to either zero or one is 100%. Results show that in a perfect case, by using signals with predicted label 1 in each segment data, the overall performance of respiratory phase detection of heart cycles will be greater than 90%. Therefore, the potential of using machine learning as a quality estimation of respiratory signals to detect respiratory phases of heartbeats is more than 90%.

Several feature extraction methods were used, and many classification methods were considered, such as linear SVM, Kernel SVM, decision tree, and K-nearest neighbor (KNN). However, the best accuracy of classification was 72.6% using linear SVM, which indicates that current classification method is not a promising way to improve the morphological-based phase identification of heart cycles.

4.2.3 Respiratory Phase Identification of Heart Cycles

The final step in our analysis is the automatic detection of respiratory phases of the selected ADR signal. By marking the beginning and the end of inhalation and exhalation phases, we defined which heart cycles happened in the exhale phase and which happened in the inhale phase. Table 8 shows the result of defining respiratory phases of heart cycles.

First column of the Table 8, which is defined as *"Best Results"*, is the results when proper envelope of SCG signal was selected manually. Second column is results when signal was selected using piecewise THD. As mentioned before, the piecewise THD technique was introduced as a quality factor to specify the quality of the lower and upper envelopes of the ADR signal, derived from SCG, and to select the superior ones. The last two columns are the results when we used either the lower or the upper envelope of SCG signal (ADR_{lower} and ADR_{Upper}).

Results confirmed that the automated respiratory phase detection results in over 77% of accuracy in overall identification of heart cycles respiratory. The results show that there is a small difference between results of phase identification using piecewise THD, 77%, and "Best Results" which was 79%.

Table 8 Respiratory phase identification of heart cycles using ADR signal.

	Best Results	Using Both Upper and Lower Envelopes	Using Lower Envelope	Using Upper Envelope
Mean	79.19%	77.22%	63.52%	73.65%
STD	10.25	10.88	18.39	16.62

CHAPTER 5

CONCLUSION AND FUTURE WORK

In this thesis, feasibility of defining respiratory phases of heart cycles using electromechanical signals (ECG) and 3D SCG signals, were investigated. Total twenty-nine different derived-respiratory signal were estimated using six different methods. Among them, the envelope detection method was selected to estimate surrogate of respiratory signal from the accelerometer. Manual and automatic techniques were used to detect respiratory phases of derived-respiratory signals. Fusion of ADR and EDR signal in presence of reference respiratory signal, the ideal case, shows significant improvement, about 98% total accuracy in detecting beginning of respiratory phases.

Recording different signals simultaneously with many electrodes and sensors, often suggests bad-contacts especially when using gel. These signals are so-called 'outliers'. In our study, among 39 recording, five trials had low-quality ADR signal. In these trials, irregular time interval patterns and low dynamic range were observed. These could be due to the location of sensors or anatomy of subjects. THD thresholding was used to remove undesired signals from further processing. In addition, THD piecewise method was used as a quality metric to select between upper and lower envelope in each recording. By using aforementioned methods, THD thresholding and piecewise THD, the ability of accelerometer-derived-respiratory (ADR) signal to detect respiratory phases

of heart cycles is increased by about 4% compared to using only upper envelope (ADR$_{Upper}$) and 14% compared to using only lower envelope (ADR$_{lower}$).

To apply this morphological approach in practical application, first size of the buffer which shows recording time, needs to be defined. For example if the size of buffer is two minutes, analyzing acceleration signal to identify respiratory phases of cardiac cycles will be done every two-minutes, offline. In other words, there will be always short delay to show the results of recorded signal. This delay depend on the length of the recorded signal. During this delay, first ADR signal will be computed using envelope detection. THD thresholding will be used to estimate the quality of the signal, then piecewise THD will be used to select between ADR$_{lower}$ and ADR$_{Upper}$. In the next step, beginning of respiratory phases of ADR will be defined automatically. Finally by having location and timing of R-peaks from ECG signal, we are able to discriminate heart-beats of inhale from exhale phase.

The limitations and drawbacks of this study and suggestions as future work to resolve them, are as followings:

- Using data of only young male subjects. To assure that proposed method is working for different subjects in different conditions, testing the algorithm on different male/female subjects with different health conditions is necessary.

- Defining window size for the envelope detection, covering dynamic range of respiration in each subject. To solve this limitation, first thirty seconds or first one minute of each recording from a subject can be used to annotate respiratory phases of ADR manually, then computed respiratory rate can assigned as the window size in envelope detection method.

- Delay between ADR and the reference respiratory belt signal. This delay can be due to the location of sensors or anatomy of subjects. Regarding improving the quality of ADR signal,

we suggest placing single axis accelerometer sensor in z direction on the chest instead of three-dimension accelerometer sensor.

- Due to the sensitivity of accelerometer sensor to movements, the proposed method can be used only in supine resting position.

- Another drawback is that accuracy of the proposed automatic respiratory phase detection is not 100%. In other words, some peaks or valley points corresponding to the beginning of exhale and inhale phases respectively, can be missed and this yields lower accuracy in defining respiratory phases of heart cycles.

- The THD assumes that most of the respiratory power is concentrated at fundamental frequency. In pathological conditions, like sleep disorders, this is not the case. Therefore, one future trend for this study is to find a quality metric and a technique to fuse ADR and EDR signals to improve detection of heart cycles related to inhale or exhale phases. Our collaborate at University of British Colombia (UBC), works on precordial acceleration signals analysis for identifying the respiratory phase of cardiac cycles using machine learning approach. In his approach, each heart cycle divided to some bins in both time and frequency domains and the task performed in three different scenarios: (i) Atlas based, (ii) patient-specific, and (iii) combination of them. Then, the results of classification compared with ground truth which computed using manual annotation described in the chapter 3. Although his results were higher than morphological approach, it needs to train the classifier with a big dataset including different subjects with different respiratory rate in order to have a sufficient performance.

5.1 My Contribution

My contribution to this research was to develop an algorithms to estimate respiratory signal using SCG , automatic detection of respiratory phases of heart beats and determination of feasibility of identifying respiratory phases of seismocardiogram signal. As a result of research conducted in this thesis, the following scientific papers were published/prepared:

1. Vahid Zakeri, Alireza Akhbardeh, **Nasim Alamdari**, Reza Fazel-Rezai, Kouhyar Tavakolian. "Precordial Acceleration Signals Analysis for Identifying the Respiratory Phase of Cardiac Cycles." Biomedical Engineering, IEEE Transactions on, 2016. (submitted)

2. **Nasim Alamdari**, Kouhyar Tavakolian, Reza Fazel-Rezai, Mikko Paukkunen, Raimo Sepponen, and Alireza Akhbardeh. "Using electromechanical signals recorded from the body for respiratory phase detection and respiratory time estimation: a comparative study." *Computing in Cardiology* 2015.

3. **Nasim Alamdari**, Alireza Akhbardeh, Kouhyar Tavakolian, Vahid Zakeri, and Reza Fazel-Rezai. "Fusion of electrocardiogram and accelerocardiogram derived respiration methods for estimation of respiratory phases," *Journal of Medical Devices*, vol. 10, no. 2, p.020928, 2016.

4. **Nasim Alamdari**, Alireza Akhbardeh, Kouhyar Tavakolian, Vahid Zakeri, and Reza Fazel-Rezai. "A Morphological Approach to Detect Respiratory Phases of Seismocardiogram" 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC). IEEE, 2016. (accepted for presentation)

APPENDICES

APPENDIX A

Appendix A contains the MATLAB® of all methods to drive respiratory signals from ECG and

SCG signals.

```matlab
function [All_Signals_dTrend, R_peak] = EstimatedSignals_All (ECG, SCG_x,
SCG_y, SCG_z, resp,fs, envelopeWindowSize)


%% G.Moody method to find the EDR
% Beat Detection (R wave peaks)
% Inputs:  S       ecg signal data
%          fs      sample rate
% OUTPUT: QRS     fiducial points of qrs complexes
signal                                  = ECG;
R_peak                                  = nqrsdetect(signal,fs);
R_ams                                   = ECG(R_peak);
RR_intervals                            = diff(R_peak);
%*********************

% check if signal is upside-down
if mean(ECG(R_peak))<mean(ECG)
    ECG                                 = -ECG;
end


%*********************
r_peaks                                 = R_peak/1000;
data_type                               = 0; % 0 --> if Matlab file
EDR_M                                   =
edr_GMoody(data_type,signal,r_peaks,fs,[],[],[],[],[]);
% EDR_Moody                             = EDR_M (:,2);
EDR_Moody                               =
resample(EDR_M(:,2),length(ECG),length(EDR_M(:,2)));
EDR_Moody                               = EDR_Moody(:);
EDR_Moody                               = (EDR_Moody-
mean(EDR_Moody))/std(EDR_Moody);
All_Signals(:,1)                        = -EDR_Moody;


%% Beat Collection From ECG Signal
numberOfBeats                           = size(R_peak,1);

for no_dims= 2:numberOfBeats
    RR_interval(no_dims-1)              = R_peak(no_dims) -
R_peak(no_dims-1);
end

minRR_interval                          = min (RR_interval);
if rem(minRR_interval,2) ~= 0
    minRR_interval                      = minRR_interval+1;
end
windowSize_ecg                          = minRR_interval;
```

```matlab
% Creating m*n Matrix which m corresponding to the number of beats and n
% corresponding to the number of samples per beat.
Matrix_pca_ecg                              = zeros (numberOfBeats,
windowSize_ecg);
for no_dims= 1:numberOfBeats
    Matrix_pca_ecg (no_dims,:)            = ECG(R_peak(no_dims)-
(windowSize_ecg/2):(R_peak(no_dims)+(windowSize_ecg/2)-1))';
end


%% %% R-amp based resp estimation
gain = 10;
timeVec                                 = 1:length(resp);
polarity                                = -1;
%estimated_resp_R_amp                    =
interp1(timeVec(R_peak),R_ams,timeVec);
EDR_r_amp                               =
resample(R_ams,length(ECG),length(R_ams));
EDR_r_amp                               = polarity*gain*(EDR_r_amp-
nanmean(EDR_r_amp));
EDR_r_amp                               = EDR_r_amp(:);
All_Signals(:,2)                        = EDR_r_amp;
%% Beat Detection (Ao wave peaks in SCG signal)

dir                                     = 1;
for scgData                             = [SCG_x, SCG_y, SCG_z];
    Ao_amp                              = [];
    Ao_Location                         = [];
    for no_dims                         = 1:numberOfBeats
        %window_scg (i)                 = scg_z ((R_peak(i)-150) :
(R_peak(i)+(150)-1));
        [timePoint, value]             = max (scgData ((R_peak(no_dims))
: (R_peak(no_dims)+(200)-1))); %window size is 300 ms, 150 ms before and
after R peak
        timePoint                      = timePoint + R_peak(no_dims);
        Ao_amp                         = [Ao_amp, value];
        Ao_Location                    = [Ao_Location, timePoint];
    end
    %%%%% Beat Collection from SCG Signal
    Ao_amps{dir}                       = Ao_amp;
    Ao_Loc{dir}                        = Ao_Location;


    numberOfBeats                      = size(Ao_Loc{dir},2);
    for no_dims                        = 2:numberOfBeats
        Ao_interval(no_dims-1)         = Ao_Loc{dir}(no_dims) -
Ao_Loc{dir}(no_dims-1);
    end

    minAoAo_interval                   = min (Ao_interval);
    if rem(minAoAo_interval,2) ~= 0
        minAoAo_interval               = minAoAo_interval+1;
    end
    windowSize_scg                     = round(minAoAo_interval);
```

```matlab
    % Creating m*n Matrix which m corresponding to the number of beats and n
    % corresponding to the number of samples per beat.
    Mat_pca_scg                                = zeros (numberOfBeats,
windowSize_scg);
    for no_dims                                = 1:numberOfBeats
        Mat_pca_scg (no_dims,:)                = scgData( (Ao_Loc{dir}(no_dims)-
(windowSize_scg/2)):(Ao_Loc{dir}(no_dims)+(windowSize_scg/2)-1))';
    end
    Matrix_pca_scgZ{dir}                       = Mat_pca_scg;
    dir                                        = dir+1;
end

%% %%%%% Ao-amp based resp estimation
Ao_ADR_x                                   =
resample(Ao_amps{1},length(SCG_x),length(Ao_amps{1}));
Ao_ADR_x                                   = Ao_ADR_x(:);
All_Signals(:,3)                           = Ao_ADR_x;

Ao_ADR_y                                   =
resample(Ao_amps{2},length(SCG_y),length(Ao_amps{2}));
Ao_ADR_y                                   = Ao_ADR_y(:);
All_Signals(:,4)                           = Ao_ADR_y;

Ao_ADR_z                                   =
resample(Ao_amps{3},length(SCG_z),length(Ao_amps{3}));
Ao_ADR_z                                   = Ao_ADR_z(:);
All_Signals(:,5)                           = Ao_ADR_z;

%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Principal Component Analysis
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

no_dims                                    = 1; % use PC 1 for resp signal
estimation; PCs 2, 3, etc are variant of resp but deformed.
polarity                                   = -1; % resp signal provided in
this example has opposite polarity compared to signal S: LeadII
[mapped_ecg, mapping_ecg]                  =
compute_mapping(Matrix_pca_ecg', 'PCA', no_dims);
gain_pca                                   = 100;
EDR_lin_PCA                                =
resample(mapping_ecg.M(:,no_dims),length(ECG),size(mapping_ecg.M,1));
EDR_lin_PCA                                = polarity*gain_pca*(EDR_lin_PCA-
nanmean(EDR_lin_PCA));
All_Signals(:,6)                           = EDR_lin_PCA(:);


% ECG drived respiration using Kernel(Gaussian) PCA:
no_dims                                    = 1; % use PC 1 for resp signal
estimation; PCs 2, 3, etc are variant of resp but deformed.
polarity                                   = -1; % resp signal provided in
this example has opposite polarity compared to signal S: LeadII
[mapped_Kecg, mapping_Kecg]                =
compute_mapping(Matrix_pca_ecg,'KernelPCA',no_dims,'gauss');
gain_Kpca                                  = 100;
```

```
EDR_Gauss_PCA                                =
resample(mapping_Kecg.V(:,no_dims),length(ECG),size(mapping_Kecg.V,1));
EDR_Gauss_PCA                                =
polarity*gain_Kpca*(EDR_Gauss_PCA-nanmean(EDR_Gauss_PCA));
All_Signals(:,7)                             = EDR_Gauss_PCA(:);

% ECG drived respiration using Kernel('Poly') PCA:
no_dims                                      = 1; % use PC 1 for resp signal
estimation; PCs 2, 3, etc are variant of resp but deformed.
polarity                                     = -1; % resp signal provided in
this example has opposite polarity compared to signal S: LeadII
[mapped_Kecg, mapping_Kecg]                  = compute_mapping(Matrix_pca_ecg,
'KernelPCA',no_dims,'poly');
gain_Kpca                                    = 100;
EDR_Poly_PCA                                 =
resample(mapping_Kecg.V(:,no_dims),length(ECG),size(mapping_Kecg.V,1));
EDR_Poly_PCA                                 =
polarity*gain_Kpca*(EDR_Poly_PCA-nanmean(EDR_Poly_PCA));
All_Signals(:,8)                             = EDR_Poly_PCA(:);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%********************************
*********************************************************
% Accelerometer (SCG_x) drived respiration (ADR_x_lin_PCA)
no_dims                                      = 1; % use PC 1 for resp signal
estimation; PCs 2, 3, etc are variant of resp but deformed.
polarity                                     = 1; % resp signal provided in
this example has opposite polarity compared to signal S: LeadII
[mapped_scg, mapping_scg]                    =
compute_mapping(Matrix_pca_scgZ{1}','PCA', no_dims);
gain_pca                                     = 100;
ADR_x_lin_PCA                                =
resample(mapping_scg.M(:,no_dims),length(SCG_x),size(mapping_scg.M,1));
ADR_x_lin_PCA                                =
polarity*gain_pca*(ADR_x_lin_PCA-nanmean(ADR_x_lin_PCA));
All_Signals(:,9)                             = ADR_x_lin_PCA(:);

% Accelerometer (SCG_y) drived respiration (ADR_y_lin_PCA)
no_dims                                      = 1; % use PC 1 for resp signal
estimation; PCs 2, 3, etc are variant of resp but deformed.
polarity                                     = 1; % resp signal provided in
this example has opposite polarity compared to signal S: LeadII
[mapped_scg, mapping_scg]                    =
compute_mapping(Matrix_pca_scgZ{2}','PCA', no_dims);
gain_pca                                     = 100;
ADR_y_lin_PCA                                =
resample(mapping_scg.M(:,no_dims),length(SCG_y),size(mapping_scg.M,1));
ADR_y_lin_PCA                                =
polarity*gain_pca*(ADR_y_lin_PCA-nanmean(ADR_y_lin_PCA));
All_Signals(:,10)                            = ADR_y_lin_PCA(:);

% Accelerometer (SCG_z) drived respiration (ADR_z_lin_PCA)
no_dims                                      = 1; % use PC 1 for resp signal
estimation; PCs 2, 3, etc are variant of resp but deformed.
polarity                                     = 1; % resp signal provided in
this example has opposite polarity compared to signal S: LeadII
```

```matlab
[mapped_scg, mapping_scg]                       =
compute_mapping(Matrix_pca_scgZ{3}','PCA', no_dims);
gain_pca                                        = 100;
ADR_z_lin_PCA                                   =
resample(mapping_scg.M(:,no_dims),length(SCG_z),size(mapping_scg.M,1));
ADR_z_lin_PCA                                   =
polarity*gain_pca*(ADR_z_lin_PCA-nanmean(ADR_z_lin_PCA));
All_Signals(:,11)                               = ADR_z_lin_PCA(:);


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%*******************************
************************************************************
%  ADR_x_GaussianKernel_PCA
no_dims                                         = 1; % use PC 1 for resp signal
estimation; PCs 2, 3, etc are variant of resp but deformed.
polarity                                        = 1; % resp signal provided in
this example has opposite polarity compared to signal S: LeadII
[mapped_scg, mapping_scg]                       =
compute_mapping(Matrix_pca_scgZ{1},'KernelPCA',no_dims,'gauss');
gain_pca                                        = 100;
ADR_x_Gauss_PCA                                 =
resample(mapping_scg.V(:,no_dims),length(SCG_x),length(mapping_scg.V));
ADR_x_Gauss_PCA                                 =
polarity*gain_pca*(ADR_x_Gauss_PCA-nanmean(ADR_x_Gauss_PCA));
All_Signals(:,12)                               = ADR_x_Gauss_PCA(:);


% ADR_y_kernel_PCA
no_dims                                         = 1; % use PC 1 for resp signal
estimation; PCs 2, 3, etc are variant of resp but deformed.
polarity                                        = 1; % resp signal provided in
this example has opposite polarity compared to signal S: LeadII
[mapped_scg, mapping_scg]                       =
compute_mapping(Matrix_pca_scgZ{2},'KernelPCA',no_dims,'gauss');
gain_pca                                        = 100;
ADR_y_Gauss_PCA                                 =
resample(mapping_scg.V(:,no_dims),length(SCG_y),length(mapping_scg.V));
ADR_y_Gauss_PCA                                 =
polarity*gain_pca*(ADR_y_Gauss_PCA-nanmean(ADR_y_Gauss_PCA));
All_Signals(:,13)                               = ADR_y_Gauss_PCA(:);


% ADR_z_kernel_PCA
no_dims                                         = 1; % use PC 1 for resp signal
estimation; PCs 2, 3, etc are variant of resp but deformed.
polarity                                        = 1; % resp signal provided in
this example has opposite polarity compared to signal S: LeadII
[mapped_scg, mapping_scg]                       =
compute_mapping(Matrix_pca_scgZ{3},'KernelPCA',no_dims,'gauss');
gain_pca                                        = 100;
ADR_z_Gauss_PCA                                 =
resample(mapping_scg.V(:,no_dims),length(SCG_z),length(mapping_scg.V));
ADR_z_Gauss_PCA                                 =
polarity*gain_pca*(ADR_z_Gauss_PCA-nanmean(ADR_z_Gauss_PCA));
All_Signals(:,14)                               = ADR_z_Gauss_PCA(:);


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%*******************************
************************************************************
%  ADR_x_PolynomialKernel_PCA
```

```matlab
no_dims                                      = 1; % use PC 1 for resp signal
estimation; PCs 2, 3, etc are variant of resp but deformed.
polarity                                     = 1; % resp signal provided in
this example has opposite polarity compared to signal S: LeadII
[mapped_scg, mapping_scg]              =
compute_mapping(Matrix_pca_scgZ{1},'KernelPCA',no_dims,'poly');
gain_pca                               = 100;
ADR_x_Poly_PCA                         =
resample(mapping_scg.V(:,no_dims),length(SCG_x),size(mapping_scg.V,1));
ADR_x_Poly_PCA                         =
polarity*gain_pca*(ADR_x_Poly_PCA-nanmean(ADR_x_Poly_PCA));
All_Signals(:,15)                      = ADR_x_Poly_PCA(:);


% ADR_y_kernel_PCA
no_dims                                      = 1; % use PC 1 for resp signal
estimation; PCs 2, 3, etc are variant of resp but deformed.
polarity                                     = 1; % resp signal provided in
this example has opposite polarity compared to signal S: LeadII
[mapped_scg, mapping_scg]              =
compute_mapping(Matrix_pca_scgZ{2},'KernelPCA',no_dims,'poly');
gain_pca                               = 100;
ADR_y_Poly_PCA                         =
resample(mapping_scg.V(:,no_dims),length(SCG_y),size(mapping_scg.V,1));
ADR_y_Poly_PCA                         =
polarity*gain_pca*(ADR_y_Poly_PCA-nanmean(ADR_y_Poly_PCA));
All_Signals(:,16)                      = ADR_y_Poly_PCA(:);


% ADR_z_kernel_PCA
no_dims                                      = 1; % use PC 1 for resp signal
estimation; PCs 2, 3, etc are variant of resp but deformed.
polarity                                     = 1; % resp signal provided in
this example has opposite polarity compared to signal S: LeadII
[mapped_scg, mapping_scg]              =
compute_mapping(Matrix_pca_scgZ{3},'KernelPCA',no_dims,'poly');
gain_pca                               = 100;
ADR_z_Poly_PCA                         =
resample(mapping_scg.V(:,no_dims),length(SCG_z),size(mapping_scg.V,1));
ADR_z_Poly_PCA                         =
polarity*gain_pca*(ADR_z_Poly_PCA-nanmean(ADR_z_Poly_PCA));
All_Signals(:,17)                      = ADR_z_Poly_PCA(:);


%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Envelope Detection
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% ECG drived respiration (EDR_env.)
%no_dims                           = 2;  % use 2 second windowing for
envelope detection. it might change subject by subject
signal_pol                  = +1;
EDR_lower_Env               =
detect_envelope(signal_pol*ECG,envelopeWindowSize*fs,'down');
gain_ecg                    = 100;
polarity_ecg                = +1;
EDR_lower_Env               = polarity_ecg*gain_ecg*(EDR_lower_Env -
mean(EDR_lower_Env));  % make it zero mean and amplify it
All_Signals(:,18)           = -EDR_lower_Env(:);
```

```matlab
% EDR_upper envelope
%no_dims                          = 2;  % use 2 second windowing for
envelope detection. it might change subject by subject
signal_pol                        = +1;
EDR_upper_Env                     =
detect_envelope(signal_pol*ECG,envelopeWindowSize*fs,'up');
gain_ecg                          = 100;
polarity_ecg                      = +1;
EDR_upper_Env                     = polarity_ecg*gain_ecg*(EDR_upper_Env -
mean(EDR_upper_Env));  % make it zero mean and amplify it
All_Signals(:,19)                 = EDR_upper_Env(:);


% SCG_x driven resp estimation (ADR_x)
% Down Envelope
%no_dims                          = 2;  % use 2 second windowing for
envelope detection. it might change subject by subject
ADR_x_lower_Env                   =
detect_envelope(SCG_x,envelopeWindowSize*fs,'down');
gain_scg                          = 100;
polarity_scg                      = 1;
ADR_x_lower_Env                   = polarity_scg*gain_scg*(ADR_x_lower_Env -
mean(ADR_x_lower_Env));  % make it zero mean and amplify it
All_Signals(:,20)                 = -ADR_x_lower_Env(:);


%  ADR_x with upper envelope
%no_dims                          = 2;  % use 2 second windowing for
envelope detection. it might change subject by subject
ADR_x_upper_Env                   =
detect_envelope(SCG_x,envelopeWindowSize*fs,'up');
gain_scg                          = 100;
polarity_scg                      = 1;
ADR_x_upper_Env                   = polarity_scg*gain_scg*(ADR_x_upper_Env -
mean(ADR_x_upper_Env));  % make it zero mean and amplify it
All_Signals(:,21)                 = ADR_x_upper_Env(:);
% end

% SCG_y driven resp estimation (ADR_y)
% Down Envelope
%no_dims                          = 2;  % use 2 second windowing for
envelope detection. it might change subject by subject
ADR_y_lower_Env                   =
detect_envelope(SCG_y,envelopeWindowSize*fs,'down');
gain_scg                          = 100;
polarity_scg                      = 1;
ADR_y_lower_Env                   = polarity_scg*gain_scg*(ADR_y_lower_Env -
mean(ADR_y_lower_Env));  % make it zero mean and amplify it
All_Signals(:,22)                 = -ADR_y_lower_Env(:);

%  ADR_y with upper envelope
%no_dims                          = 2;  % use 2 second windowing for
envelope detection. it might change subject by subject
ADR_y_upper_Env                   =
detect_envelope(SCG_y,envelopeWindowSize*fs,'up');
gain_scg                          = 100;
```

```matlab
polarity_scg                            = 1;
ADR_y_upper_Env                         = polarity_scg*gain_scg*(ADR_y_upper_Env -
mean(ADR_y_upper_Env));   % make it zero mean and amplify it
All_Signals(:,23)                       = ADR_y_upper_Env(:);


% SCG_z driven resp estimation (ADR_z)
% Down Envelope
%no_dims                                = 2;  % use 2 second windowing for
envelope detection. it might change subject by subject
ADR_z_lower_Env                         =
detect_envelope(SCG_z,envelopeWindowSize*fs,'down');
gain_scg                                = 100;
polarity_scg                            = 1;
ADR_z_lower_Env                         = polarity_scg*gain_scg*(ADR_z_lower_Env -
mean(ADR_z_lower_Env));   % make it zero mean and amplify it
All_Signals(:,24)                       = -ADR_z_lower_Env(:);


%  ADR_z with upper envelope
%no_dims                                = 2;  % use 2 second windowing for
envelope detection. it might change subject by subject
ADR_z_upper_Env                         =
detect_envelope(SCG_z,envelopeWindowSize*fs,'up');
gain_scg                                = 100;
polarity_scg                            = 1;
ADR_z_upper_Env                         = polarity_scg*gain_scg*(ADR_z_upper_Env -
mean(ADR_z_upper_Env));   % make it zero mean and amplify it
All_Signals(:,25)                       = ADR_z_upper_Env(:);


% SCG_xz driven resp estimation (ADR_xz)
%no_dims                                = 2;  % use 2 second windowing for
envelope detection. it might change subject by subject
scg_xz                                  = sqrt ((SCG_x.^2)+(SCG_z.^2));
ADR_xz_lower_Env                        =
detect_envelope(scg_xz,envelopeWindowSize*fs,'down');
gain_scg                                = 100;
polarity_scg                            = 1;
ADR_xz_lower_Env                        = polarity_scg*gain_scg*(ADR_xz_lower_Env -
mean(ADR_xz_lower_Env));   % make it zero mean and amplify it
All_Signals(:,26)                       = -ADR_xz_lower_Env(:);


% SCG_xz driven resp estimation (ADR_xz)
%no_dims                                = 2;  % use 2 second windowing for
envelope detection. it might change subject by subject
scg_xz                                  = sqrt ((SCG_x.^2)+(SCG_z.^2));
ADR_xz_upper_Env                        =
detect_envelope(scg_xz,envelopeWindowSize*fs,'up');
gain_scg                                = 100;
polarity_scg                            = 1;
ADR_xz_upper_Env                        = polarity_scg*gain_scg*(ADR_xz_upper_Env -
mean(ADR_xz_upper_Env));   % make it zero mean and amplify it
All_Signals(:,27)                       = ADR_xz_upper_Env(:);
```

```matlab
% SCG_xyz driven resp estimation (ADR_xyz)
%no_dims                          = 2;  % use 2 second windowing for
envelope detection. it might change subject by subject
scg_xyz                          = sqrt ((SCG_x.^2)+(SCG_y.^2)+(SCG_z.^2));
ADR_xyz_lower_Env                =
detect_envelope(scg_xyz,envelopeWindowSize*fs,'down');
gain_scg                         = 100;
polarity_scg                     = 1;
ADR_xyz_lower_Env                = polarity_scg*gain_scg*(ADR_xyz_lower_Env
- mean(ADR_xyz_lower_Env));  % make it zero mean and amplify it
All_Signals(:,28)                = -ADR_xyz_lower_Env(:);


% SCG_xyz driven resp estimation (ADR_xyz)
%no_dims                          = 2;  % use 2 second windowing for
envelope detection. it might change subject by subject
scg_xyz                          = sqrt ((SCG_x.^2)+(SCG_y.^2)+(SCG_z.^2));
ADR_xyz_upper_Env                =
detect_envelope(scg_xyz,envelopeWindowSize*fs,'up');
gain_scg                         = 100;
polarity_scg                     = 1;
ADR_xyz_upper_Env                = polarity_scg*gain_scg*(ADR_xyz_upper_Env
- mean(ADR_xyz_upper_Env));  % make it zero mean and amplify it
All_Signals(:,29)                = ADR_xyz_upper_Env(:);



RR_interval_interp               = -
interp1(R_peak(2:end),RR_intervals,R_peak(2):R_peak(end));
RR_interval_interp               = [repmat(0,R_peak(2), 1);
RR_interval_interp(:); repmat(0, length(resp)-R_peak(end), 1)];
All_Signals(:,30)                = RR_interval_interp(:);



%% Normalization and Dtrending all the signals
for i = 1:size(All_Signals,2)

    signal_withTrend             = All_Signals(:,i);
    [p,s,mu]                     =
polyfit((1:numel(signal_withTrend))',signal_withTrend,6);
    f_y                          =
polyval(p,(1:numel(signal_withTrend))',[],mu);
    All_Signals(:,i)             = signal_withTrend - f_y;       % Detrend
data
    All_Signals_dTrend(:,i)      = (All_Signals(:,i)-
mean(All_Signals(:,i))/std(All_Signals(:,i)));
end

resp                             = (resp-mean(resp))/std(resp);
```

APPENDIX B

Appendix B contains MATLAB code of automatic respiratory phase detection of derived-

respiratory signals and then identifying respiratory phase of heart cycles and finally compute the

performance accuracy.

```matlab
clc;clear;close all
ref                                             = 2;
segment_len                                     = 600 *1000 % 2 minute
for envelopeWindowSize                          = 2.0;
    for SubjectNumber                           = [1:20]
        SubjectNumber
        data_ref                                =
strcat('subject',num2str(SubjectNumber),'_ref',num2str(ref),'_SV','.mat');
        % Load the data
        load(data_ref);
        format long
        %% Raw signals

        if SubjectNumber == 3 | SubjectNumber == 4 | SubjectNumber == 11
            ECG                                 = data (:,8);
            SCG_x                               = data (:,5);
            SCG_y                               = data (:,6);
            SCG_z                               = data (:,7);
        else
            ECG                                 = data (:,9);
            SCG_x                               = data (:,6);
            SCG_y                               = data (:,7);
            SCG_z                               = data (:,8);
        end

        % LeadI                                 = data (:,1);
        % LeadIII                               = LeadII - LeadI;
        resp                                    = data (:,2);
        fs                                      = 1000;
        [sixSignals_dTrend, R_peak]             =
EstimatedSignals_selectedSix (ECG, SCG_x, SCG_y, SCG_z, resp,fs,
envelopeWindowSize);
        resp                                    = (resp-
mean(resp))/std(resp);

        %[All_Signals_dTrend, R_peak]           = EstimatedSignals_All
(ECG, SCG_x, SCG_y, SCG_z, resp,fs, envelopeWindowSize); % Produce all the
signals
        %saveName1                              =
strcat('AllEstimatedSignals_Subject',num2str(SubjectNumber),'_ref',num2str(re
f),'.mat');
        %save (saveName1,'All_Signals_dTrend', 'R_peak' )
        %saveName1                              =
strcat('AllEstimatedSignals_Subject',num2str(SubjectNumber),'_ref',num2str(re
f),'.mat');
        %load (saveName1 )
```

```matlab
        for indexSignal                            = 1:6;
            indexSignal
            signal                                 =
sixSignals_dTrend(:,indexSignal); % chose the signal
            %% plot in estimated signals to compare them with reference
signal

            %             time = (1:length(resp))/fs;
            %             figure;
            %             plot (time, resp, 'k', 'MarkerSize', 6,
'LineWidth', 1), hold on
            %             plot(time, signal, 'b', 'MarkerSize', 6,
'LineWidth', 1); title('ADR-down v.s. Reference signal')
            %
            %% %% %% %% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Segmenting the
data into 30 seconds segments
            %         segment_len                      = 600 *1000
            numSegment                             =
floor(length(resp)/segment_len); % #Segments should be = 20
            start_point                            = 1;

            inhale_Heart_beats_sensitivity         = cell(numSegment,1);
            exhale_Heart_beats_specificity         = cell(numSegment,1);
            inhale_Heart_beats_PPV                 = cell(numSegment,1);
            exhale_Heart_beats_NPV                 = cell(numSegment,1);
            total_heartCyc_accuracy                = cell(numSegment,1);

            for nn                                 = 1:numSegment
                close all
                clear signal_portion resp_portion max_locs max_val min_val
min_locs mins_idx min_t index min_value alter_proper_min_idx
alter2_proper_min_idx
                clear max_loc ia peak_resp valley_resp max_iidx min_iidx
resp_max_loc resp_min_loc resp_cycle find_valley_idx find_valley
                clear  R_peak_idx_inhale R_peak_idx_exhale
R_peaks_exhale_S_all R_peaks_inhale_S_all R_peaks_exhale_ref_all
R_peaks_inhale_ref_all index
                end_point                          = start_point +
segment_len -1;
                signal_portion                     = signal([start_point
: end_point],:);
                resp_portion                       = resp(start_point :
end_point,:);

                % Normalization and Dtrending all the signals
                signal_withTrend                   = signal_portion;
                [p,s,mu]                           =
polyfit((1:numel(signal_withTrend))',signal_withTrend,6);
                f_y                                =
polyval(p,(1:numel(signal_withTrend))',[],mu);
                signal_portion                     = signal_withTrend -
f_y;       % Detrend data
                %signal_portion                     = (signal_portion-
mean(signal_portion(:))/std(signal_portion(:)));
```

```matlab
            signal_portion                          =
zscore(signal_portion);

            %% finding peaks and valleys of the ADR signal

            fs                                      = 1000; %sampling
freq. = 1000 Hz
            % finding maximums
            if indexSignal                          == 6
                [max_val,max_locs]                  =
findpeaks(signal_portion, fs, 'MinPeakDistance', 3.0);
            else
                [max_val,max_locs]                  =
findpeaks(signal_portion, fs, 'MinPeakDistance', 2.0, 'MinPeakProminence',
0.4);
            end

            % finding minimums
            if indexSignal                          == 6
                [min_val,min_locs]                  = findpeaks(-
signal_portion, fs, 'MinPeakDistance', 3.0);
            else
                [min_val,min_locs]                  = findpeaks(-
signal_portion, fs, 'MinPeakDistance', 2.0,  'MinPeakProminence', 0.4);
            end
            %% The flat peaks Correction.
            % Require a minimum amplitude difference of  10^{-4} between
a peak and its neighbors.
            figure;
            plot(signal_portion); hold on; title('signal by applying
envelope detection with W = 2.0 sec.')
            plot( max_locs*1000,max_val,'ko');
            plot( min_locs*1000,-min_val,'ro');
            plot (resp_portion, 'k', 'MarkerSize', 6, 'LineWidth', 1);

            for k                                   = 1 :
length(max_locs)-1
                num                                 = 1;
                flatPeak                            = 1;
                while flatPeak                      == 1; % suspect
having a flat peak
                    sample                          = 1000*max_locs(k)-
num;
                    sample_val                      =
signal_portion(round(sample));
                    diff_samples                    = abs(max_val(k) -
sample_val );
                    if diff_samples                 > 1e-3
                        flatPeak                    = 0; % we don't have
a flat peak
                        % max_locs(k)               = (1000*
                        % max_locs(k)-num+1)/1000 ; % make the begining
of flat part as peak
                        max_locs(k)                 = (   (1000*
max_locs(k)-num+1)+(1000*max_locs(k))  )/2000 ;
                        if num          > 1
```

69

```matlab
                    plot( max_locs (k)*1000 ,max_val (k),'rx');
                end
            end
            num                                    = num + 1;
        end

    end

    %% % correcting minimums
    figure;
    plot(signal_portion); hold on; title('signal by applying
envelope detection with W= 2.0 sec.')

    if length(max_locs) >= 3
        for i                                      =
2:length(max_locs);
            mins_idx                               = find(min_locs
>= max_locs(i-1) & min_locs <= max_locs(i));
                if isempty(mins_idx)               == 1 %if we
couldn't find the min in that interval, then what?
                    %disp('check the maximums and decide what you
should do!')
                    max_locs(i)                    = max_locs(i-1)
;
                    max_val(i)                     = max_val (i-1);
                elseif length(mins_idx)            == 1;
                    min_t (i-1)                    =
min_locs(mins_idx);
                    index                          =  mins_idx ;
                    min_value (i-1)                = -
min_val(index);
                    plot( min_t (i-1)*1000,min_value (i-1),'ro',
'MarkerSize', 6, 'LineWidth', 1);
                elseif length(mins_idx) > 1 % if more than one valley
was found
                    detected_mins                  =
min_locs(mins_idx);
                    interval                       = max_locs(i) -
max_locs(i-1);
                    proper_min_idx                 =
find(detected_mins >= (max_locs(i-1)+ 0.5*interval) & detected_mins <=
(max_locs(i-1)+ (2/3)*interval));
                        if isempty (proper_min_idx)       ~= 1
                            min_t (i-1)                    =
detected_mins(proper_min_idx);
                            index                          =
mins_idx(proper_min_idx);
                            min_value (i-1)                = -
min_val(index);
                            plot( min_t (i-1)*1000,min_value (i-1),'ro',
'MarkerSize', 6, 'LineWidth', 1);
                        else %if we couldn't find the min in that
interval,then what
                            alter_proper_min_idx           =
find(detected_mins >= (max_locs(i-1)+ (2/3)*interval) & detected_mins <
max_locs(i));
```

```matlab
                                if isempty (alter_proper_min_idx) ~= 1;
                                    if length(alter_proper_min_idx) >1
                                        alter_proper_min_idx =
max(alter_proper_min_idx);
                                    end
                                    min_t (i-1)               =
detected_mins(alter_proper_min_idx);
                                    index                     =
mins_idx(alter_proper_min_idx);
                                    min_value (i-1)           = -
min_val(index);
                                    plot( min_t (i-1)*1000,min_value (i-
1),'ro',  'MarkerSize', 6, 'LineWidth', 1);
                                else %if we couldn't find the min in thoese
two intervals, then what
                                    alter2_proper_min_idx     =
find(detected_mins > max_locs(i-1) & detected_mins < (max_locs(i-1)+
0.5*interval));
                                    if length(alter2_proper_min_idx) >1
                                        alter2_proper_min_idx =
max(alter2_proper_min_idx);
                                    end
                                    min_t (i-1)               =
detected_mins(alter2_proper_min_idx);
                                    index                     =
mins_idx(alter2_proper_min_idx);
                                    min_value (i-1)           = -
min_val(index);
                                    plot( min_t (i-1)*1000,min_value (i-
1),'ro',  'MarkerSize', 6, 'LineWidth', 1);
                                end
                            end
                        end
                    end
                    plot (resp_portion, 'k', 'MarkerSize', 6, 'LineWidth',
1);

                    [max_loc, ia, ic]                         =
unique(max_locs);
                    max_val                                   = max_val(ia);
                    plot( max_loc*1000,max_val,'co',  'MarkerSize', 6,
'LineWidth', 1);
                    meanCycle_maxPoint                        =
mean(diff(max_loc));
                    meanCycle_minPoint                        = mean(diff(min_t
));
                    % ADR_RR                         = mean([diff(ADR_max_loc);
diff(ADR_min_t')])/2
                    % ADR_RR                         =
median([diff(ADR_max_locs); diff(ADR_min_locs)])/2
                    % ADR_RR                         = mode([diff(ADR_max_locs);
diff(ADR_min_locs)])/2

                    %%  peaks and valleys of the signal and Reference
respiratory signals
```

```matlab
                peak_resp                              =
load(strcat('resp','_Subject',num2str(SubjectNumber),'_ref',num2str(ref)
,'_peaks','.txt'));
                valley_resp                            =
load(strcat('resp','_Subject',num2str(SubjectNumber),'_ref',num2str(ref)
,'_valleys','.txt'));


                max_iidx                               =
find(peak_resp(:,1)>= start_point & peak_resp(:,1) <= end_point);
                resp_max_loc                           =
peak_resp(max_iidx,1)- ((nn-1)*segment_len);
                resp_max_val                           =
peak_resp(max_iidx,2);


                min_iidx                               =
find(valley_resp(:,1)>= start_point & valley_resp(:,1) <= end_point);
                resp_min_loc                           =
valley_resp(min_iidx,1) - ((nn-1)*segment_len);
                resp_min_val                           =
valley_resp(min_iidx,2);


                %                       figure;
                %                       plot(signal_portion); hold on;
title('Signal by applying envelope detection once');
                %                       plot( max_loc*1000,max_val,'ko',
'MarkerSize', 6, 'LineWidth', 1);
                %                       plot( min_t*1000,min_value,'ro',
'MarkerSize', 6, 'LineWidth', 1);
                %                       plot (resp_portion, 'k');
                %                       plot (resp_min_loc, resp_min_val,
'bo',  'MarkerSize', 6, 'LineWidth', 1);
                %                       plot (resp_max_loc, resp_max_val,
'bo',  'MarkerSize', 6, 'LineWidth', 1);



                %% Comparing automatic peaks detection with manual peak
annotation

                % peak_ADR_manual          =
strcat(num2str(signal_1),'_Subject',num2str(SubjectNumber),'_ref',num2str(ref
) ,'_peaks_manual',num2str(choice_ADR),'.txt');
                % valley_ADR_manual        =
strcat(num2str(signal_1),'_Subject',num2str(SubjectNumber),'_ref',num2str(ref
) ,'_valleys_manual',num2str(choice_ADR),'.txt');
                %
                % inhale_ADR_manual        = load (valley_ADR_manual);
                % exhale_ADR_manul         = load (peak_ADR_manual);
                %
                % inhale_ADR_manual(:,2)    =
ADR(inhale_ADR_manual(:,1));
                % exhale_ADR_manul(:,2)     = ADR(exhale_ADR_manul(:,1));
                %
                % figure;
                % plot(ADR); hold on; title('Comparing automatic peaks
detection with manual peak annotation')
                % plot(exhale_ADR_manul(:,1), exhale_ADR_manul(:,2),'bo')
```

```matlab
                        % plot(inhale_ADR_manual(:,1),
inhale_ADR_manual(:,2),'go');
                        % plot( ADR_max_loc*1000,ADR_max_val,'ko');
                        % plot( ADR_min_t*1000,ADR_min_value,'ro');
                        % legend ('ADR signal','manual peak annotation', 'manual
valley annotation','automatic peaks detection','automatic valley detection')

                        %% Phase detection

                        R_peaks_exhale_from_S             = [];
                        R_peaks_inhale_from_S             = [];

                        max_loc                           = 1000* max_loc;
                        min_t                             = 1000* min_t;

                        for i                             = 2:length(max_loc)
                            resp_cycle(i)                 = max_loc (i) -
max_loc (i-1); % resp cycle = exhale + inhale
                            find_valley_idx               = find(min_t >
max_loc (i-1) & min_t < max_loc (i));
                            if isempty(find_valley_idx) == 1 |
length(find_valley_idx) > 1
                                    disp('Error1!: Couldnt detect start of inhale
phase');
                                    max_loc (i);
                            end
                            find_valley(i-1)              =
min_t(find_valley_idx);

                            %Exhale Phase
                            R_peak_idx_exhale             = find(R_peak >=
max_loc (i-1) & R_peak < find_valley(i-1));
                            if isempty(R_peak_idx_exhale)     == 1
                                disp('Error2!:  Couldnt detect R-peak in exhale
phase')
                                    R_peak(i);
                            end

                            % Inhale Phase
                            R_peak_idx_inhale             = find(R_peak >=
find_valley(i-1) & R_peak < max_loc (i));
                            if isempty(R_peak_idx_inhale)     == 1
                                    disp('Error3!: Couldnt detect R-peak in inhale
phase')
                                    R_peak(i);
                            end

                            R_peaks_exhale_from_S {i-1}       =
{R_peak(R_peak_idx_exhale)};
                            R_peaks_inhale_from_S {i-1}       =
{R_peak(R_peak_idx_inhale)};

                        end
```

```matlab
%*************************************************************************
*******************
                    clear resp_cycle find_valley_idx find_valley
R_peak_idx_exhale R_peak_idx_inhale
                    R_peaks_exhale_from_ref                 = [];
                    R_peaks_inhale_from_ref                 = [];



                for i                                       =
2:length(resp_max_loc)
                        resp_cycle(i)                       = resp_max_loc (i)
- resp_max_loc (i-1); % resp cycle = exhale + inhale
                        find_valley_idx                     = find(resp_min_loc
> resp_max_loc (i-1) & resp_min_loc < resp_max_loc (i));
                        if isempty(find_valley_idx)         == 1 |
length(find_valley_idx) > 1
                                disp('Error4!: Couldnt detect start of inhale
phase')
                                resp_max_loc (i);
                        end
                        find_valley(i-1)                    =
resp_min_loc(find_valley_idx);

                        %Exhale Phase
                        R_peak_idx_exhale                   = find(R_peak >=
resp_max_loc (i-1) & R_peak < find_valley(i-1));
                        if isempty(R_peak_idx_exhale)       == 1
                                disp('Error5!:  Couldnt detect R-peak in exhale
phase')
                                R_peak(i);
                        end

                        % Inhale Phase
                        R_peak_idx_inhale                   = find(R_peak >=
find_valley(i-1) & R_peak < resp_max_loc (i));
                        if isempty(R_peak_idx_inhale)       == 1
                                disp('Error6!: Couldnt detect R-peak in inhale
phase')
                                R_peak(i);
                        end

                        R_peaks_exhale_from_ref {i-1}    =
{R_peak(R_peak_idx_exhale)};
                        R_peaks_inhale_from_ref {i-1}    =
{R_peak(R_peak_idx_inhale)};

                end

                %% gather all the heart beat detection
                k1                                          = 1;
                for i                                       =
1:length(R_peaks_exhale_from_S)
                        R_pks_len                           =
length(R_peaks_exhale_from_S{i}{1,1});
```

```matlab
                k2                              = k1 - 1 +
R_pks_len;
                R_peaks_exhale_S_all(k1:k2)     =
R_peaks_exhale_from_S{i}{1,1};
                k1                              = k2 + 1;
            end

            k1                                  = 1;
            for i                               =
1:length(R_peaks_inhale_from_S)
                R_pks_len                       =
length(R_peaks_inhale_from_S{i}{1,1});
                k2                              = k1 - 1 +
R_pks_len;
                R_peaks_inhale_S_all(k1:k2)     =
R_peaks_inhale_from_S{i}{1,1};
                k1                              = k2 + 1;
            end

            k1                                  = 1;
            for i                               =
1:length(R_peaks_exhale_from_ref)
                R_pks_len                       =
length(R_peaks_exhale_from_ref{i}{1,1});
                k2                              = k1 - 1 +
R_pks_len;
                R_peaks_exhale_ref_all(k1:k2)   =
R_peaks_exhale_from_ref{i}{1,1};
                k1                              = k2 + 1;
            end

            k1                                  = 1;
            for i                               =
1:length(R_peaks_inhale_from_ref)
                R_pks_len                       =
length(R_peaks_inhale_from_ref{i}{1,1});
                k2                              = k1 - 1 +
R_pks_len;
                R_peaks_inhale_ref_all(k1:k2)   =
R_peaks_inhale_from_ref{i}{1,1};
                k1                              = k2 + 1;
            end



            %% Compute the accuracy (inhale considered as the
positive class and Exhale as the Negative class)

            TP                                  = 0;
            for m                               =
1:length(R_peaks_inhale_ref_all)
                index                           =
find(R_peaks_inhale_S_all == R_peaks_inhale_ref_all(m));
                if isempty(index)               ~= 1
                    TP                          = TP+1;
```

```matlab
                        end
                    end

                FN                                        =
length(R_peaks_inhale_ref_all) - TP;


                TN                                        = 0;
                for j                                     =
1:length(R_peaks_exhale_ref_all)
                    index                                 =
find(R_peaks_exhale_S_all == R_peaks_exhale_ref_all(j));
                    if isempty(index)                     ~= 1
                        TN                                = TN+1;
                    end
                end

                FP                                        =
length(R_peaks_exhale_ref_all)-TN;
                inhale_Heart_beats_sensitivity{nn}        = (TP/(TP+FN)) *
100;
                exhale_Heart_beats_specificity{nn}        = (TN/(TN+FP)) *
100;

                inhale_Heart_beats_PPV{nn}                =
(TP/(TP+FP))*100;
                exhale_Heart_beats_NPV{nn}                =
(TN/(TN+FN))*100;
                total_heartCyc_accuracy{nn}               =
(((inhale_Heart_beats_sensitivity{nn}*m)+(exhale_Heart_beats_specificity{nn}*
j))/(j+m));

                start_point                               = end_point + 1;
            else
                inhale_Heart_beats_sensitivity{nn}        = 0;
                exhale_Heart_beats_specificity{nn}        = 0;

                inhale_Heart_beats_PPV{nn}                = 0;
                exhale_Heart_beats_NPV{nn}                = 0;
                total_heartCyc_accuracy{nn}               = 0;

                start_point                               = end_point + 1;
            end
        end
        close all
        inhale_all_sensitivity(indexSignal,:)             =
cell2mat(inhale_Heart_beats_sensitivity);
        exhale_all_specificity(indexSignal,:)             =
cell2mat(exhale_Heart_beats_specificity);
        heartCyc_all_accuracy(indexSignal,:)              =
cell2mat(total_heartCyc_accuracy);
    end
            saveName                                             =
strcat('SensSpes_',num2str(segment_len/60000),'minute_',num2str(envelopeWindo
wSize),'W_Subject',num2str(SubjectNumber),'_ref',num2str(ref),'.mat');
```

```
                save (saveName, 'inhale_all_sensitivity',
'exhale_all_specificity', 'heartCyc_all_accuracy')
    end
end
```

APPENDIX C

Appendix C contains the MATLAB® of detecting lower or upper envelope of a signal.

```matlab
function [env x_new data_new] = detect_envelope(data, window, direction)


direction = strcmpi( {'up','down'}, direction ) * [ 1 ; -1 ];
x_data=1:length(data);
data = data(:);
L = length(data );

x_new = [];
data_new = [];
i = 1;
while i < L
    Ind = i+1:min( i + window, L );
    [ m, idx ] = max( ( data(Ind) - data(i) ) ./ (Ind-i)' .* direction );
    i = i + idx;
    x_new = [ x_new x_data(i) ];
    data_new = [ data_new data(i) ];
end;

% env = interp1( x_new, data_new, x_data, 'spline' );
env = interp1( x_new, data_new, x_data, 'linear', 'extrap' );
```

APPENDIX D

Appendix B contains the MATLAB® of the detecting R-peaks of ECG signal.

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


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


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

```matlab
Wn0=Bwn;      %bandwidth of the first filter
Wn1=[Bwn 2*Bwn];     %bandwidth of the second filter
Wn2=[2*Bwn 3*Bwn];
Wn3=[3*Bwn 4*Bwn];
Wn4=[4*Bwn 5*Bwn];


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


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


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


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


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

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

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

```matlab
%-------------------------------------------------------------------
%-------------------------------------------------------------------
%-------------------------------------------------------------------
%subfunctions

function y=MWI(S)

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

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

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

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

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

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

```matlab
        sumnoise=sumnoise+mwi(P);
        NL=sumnoise/(length(Noise)+1);   %Updating the Noise Level
    end
end
%------------------------------------------------------------
function [pnew]=conversion(S,FL2,pold,M,N,fs)

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

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

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

cutbeginning=find(Signal<N);      %Cutting out the first points because of
initial transient of the filter in polyphase_imp
fpointsb=Signal(cutbeginning);
cutend=find(Signal>length(S)-N); %Cutting out the last points
fpointse=Signal(cutend);
pnew=setxor(Signal,[fpointsb;fpointse]);
%-----------------------------------------
function yn=history(ynm1,xn)

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

lambda=0.8; %forgetting factor
yn=(1-lambda)*xn+lambda*ynm1;
```

APPENDIX E

```matlab
function y= edr_GMoody(varargin)
%
% y = edr(data_type,signal,r_peaks,fs,pqoff, jpoff, gain_ecg, channel, show)
%
% ECG-derived Respiratory (EDR) signal computation from given
% single-lead ECG signal based on the signed area under the QRS complex.
%
% Required Parameters:
%
% data_type
%       A 1x1 integer specifying the file data_type
%       0 --> if Matlab file
%       1 --> if record in MIT format
%
% signal
%       A Nx1 integer array containing the ECG signal in mV (if data_type=0)
%       OR a char string containing record name (if data_type=1)
%
% fs
%       A 1x1 integer specifying the sampling frequency in hz (for Matlab
variables only)
%
% r_peaks
%
%       A Mx1 integer array containing locations of r peaks on signal in s
%       OR a char string containing the extension of the annotation file
%       with r peaks in samples (e.g. "qrs") (if data_type=1)
%
% optional parameters:
%
% gain_ecg
%       A 1x1 integer specifying dig_max/phy_max (default=1)
%
% channel
%       A 1x1 integer>1 (default=1) indicating ECG channel (if data_type=1)
%
% pqoff
%       A 1x1 integer>0 specifying average distance between PQ junction and
%       R peak, in samples
%
% jpoff
%       A 1x1 integer>0 specifying average distance between R peak and
%       J point, in samples
%
% show
%       A 1x1 boolean if true, generates a plot of the estimated
%       respiration signal (default = 0).
%
%
% output:
%
% y
%       A Mx2 integer matrix containing time in seconds and edr
%
```

```matlab
% This code was written by Sara Mariani at the Wyss Institute at Harvard
% based on the open-source PhysioNet code edr.c
% (http://www.physionet.org/physiotools/edr/)
% by George Moody
%
% Author: Sara Mariani, 2014
% Last Modified: November 17, 2014
%
% please report bugs/questions at sara.mariani@wyss.harvard.edu
%
% Example - Extract EDR signal from ECG in PhysioNet's Remote server:
% signal='fantasia/f1o02';
% r_peaks='ecg';
% data_type=1;
% channel=2;
% show=1;
% y=edr(data_type,signal,r_peaks,[],[],[],channel,show);
% wfdb2mat('f1o02')
% [~,signal,Fs,~]=rdmat('f1o02m');
% resp=signal(:,1);
% resp=resp-mean(resp);
% resp=resp*200;
% sec=length(resp)/Fs;
% xax=[.25:.25:sec];
% r=interp1(y(:,1), y(:,2), xax, 'spline');
% figure
% plot(xax,r)
% hold on
% plot([1:length(resp)]/Fs,resp,'r')
% legend('edr','respiratory signal')
% xlabel('time (s)')
%
% see also: ecgpuwave, gqrs


%endOfHelp



%Set default pararameter values
inputs={'data_type','signal','r_peaks','fs','pqoff','jpoff', 'gain_ecg',
'channel' ,'show'};
show=0;
Ninputs=length(inputs);
if nargin>Ninputs
    error('Too many input arguments')
end
if nargin<3
    error('Not enough input arguments')
end

for n=1:nargin
    eval([inputs{n} '=varargin{n};'])
end
for n=nargin+1:Ninputs
    eval([inputs{n} '=[];'])
end
```

88

```matlab
% check format and obtain all the features I need
if data_type==0 %matlab

    if  isempty(gain_ecg)
        gain_ecg=1;
    end
    ECGm=signal*gain_ecg;
    if isempty(r_peaks)
        error('R peaks locations not provided')
    else
        tqrs=round(r_peaks*fs); %samples where I have the R peak
    end

elseif data_type==1 %wfdb record
    if isempty(channel)
        channel=1;
    end
    % read the signal
    wfdb2mat(signal);
    pp=strfind(signal,'/');
    if ~isempty(pp)
        signal2=signal(pp(end)+1:length(signal));
    else signal2=signal;
    end
    [~,sig,fs]=rdmat([signal2 'm']);
    ECGm=sig(:,channel);
    if numel(fs)>1
        fs=fs(channel);
    end
    % read the header
    signal
    siginfo=wfdbdesc(signal);
    siginfo=siginfo(:,channel);
    gainstring=siginfo.Gain;
    sp=strfind(gainstring,' ');
    try
        gain_ecg=str2num(gainstring(1:sp-1));
    catch
        gain_ecg=1;
    end
    if strfind(gainstring(end-1),'m')
        gain_ecg=gain_ecg*1000;
    end

    ECGm=ECGm*gain_ecg;
    % read r_peaks if annotation file
    if ischar(r_peaks)
        [ann,ty]=rdann(signal,r_peaks);
        tqrs=ann(ty=='N');
        r_peaks=tqrs/fs;
    else
        tqrs=round(r_peaks*fs); %samples where I have the R peak
    end

else error('format data_type must be 0 or 1')
```

```matlab
end

% check if signal is upside-down
if mean(ECGm(tqrs))<mean(ECGm)
    ECGm=-ECGm;
end

% EDR COMPUTATION
% 1) filter the signal with a moving window of lpflen=25 ms
lpflen=0.025;
lp=round(lpflen*fs);
w=ones(lp+1,1)./(lp+1);
sample=filter(w,1,ECGm);
% correct for the delay of lp/2
sample(1:round(lp/2))=[];
% correct for the initialization
for i=1:round(lp/2)
    sample(i)=mean(ECGm(1:i+round(lp/2)));
end
% 2) find the baseline: moving window again of bflen=1 s
bflen=1;
b=round(bflen*fs);
w2=ones(b+1,1)./(b+1);
baseline=filter(w2,1,sample);
% correct for the delay of b/2
baseline(1:round(b/2))=[];
% correct for the initialization
for i=1:round(b/2)
    baseline(i)=mean(sample(1:i+round(b/2)));
end

% 3) find average boundaries of QRS interval
if isempty(jpoff)||isempty(pqoff)
    [pqoff, jpoff]=boundaries(sample, baseline, tqrs, fs);
end

% now estimate signed area under QRS complex
sb=sample(1:length(baseline))-baseline;
snar=zeros(size(tqrs));

for i=2:length(tqrs)-1
    win=sb(tqrs(i)-pqoff:tqrs(i)+jpoff);
    snar(i)=sum(win);
end
if tqrs(end)+jpoff>length(sb)
    win=sb(tqrs(end)-pqoff:end);
else
    win=sb(tqrs(end)-pqoff:tqrs(end)+jpoff);
end
snar(end)=sum(win);

% now start from signed area and estimate edr
xm=0;
xd=0;
xdmax=0;
xc=0;
```

```matlab
x=snar;
r=zeros(size(x));
for i=25:length(x)
    d=x(i)-xm;
    if xc<500
        xc=xc+1;
        dn=d/xc;
    else
        dn=d/xc;
        if dn>xdmax
            dn=xdmax;
        elseif dn<-xdmax
            dn=-xdmax;
        end
    end
    xm=xm+dn;
    xd=xd+abs(dn)-xd/(xc);
    if xd<1
        xd=1;
    end
    xdmax=3*xd/(xc);
    r(i)=d/xd;
end
y=r*50;
while (max(y)>127 || min(y)<-128)
    y(y<-128)=y(y<-128)+255;
    y(y>127)=y(y>127)-255;
end

if(show)
    scrsz = get(0,'ScreenSize');
    figure('Position',...
        [0.05*scrsz(3) 0.05*scrsz(4) 0.8*scrsz(3) 0.89*scrsz(4)],...
        'Color',[1 1 1]);
    ax(1)=subplot(211);
    plot([1:length(sample)]/fs,sample)
    hold on
    plot([1:length(baseline)]/fs,baseline,'g')
    plot((tqrs-pqoff)/fs,mean(ECGm)*ones(size(tqrs)),'*m')
    plot((tqrs+jpoff)/fs,mean(ECGm)*ones(size(tqrs)),'*c')
    legend('filtered ecg','baseline','window start','window end')
    set(gca,'fontsize',18)
    xlabel('time (s)','fontsize',18)
    ylim([mean(ECGm)-5*std(ECGm) mean(ECGm)+5*std(ECGm)])
    ax(2)=subplot(212);
    plot(r_peaks,y,'r')
    title('edr','fontsize',18)
    set(gca,'fontsize',18)
    xlabel('time (s)','fontsize',18)
    linkaxes(ax,'x')
end
 y=[r_peaks y];
end


%%%% Helper function %%%%%
```

```matlab
function[pqoff, jpoff]=boundaries(sample, baseline, tqrs, fs)
% estimate the noise level
sb=sample(1:length(baseline))-baseline;
nlest=mean(abs(sb));
display(['The estimated noise level is ' num2str(nlest) ' microvolts']);
dlthresh=2*nlest;
dlthmax=1200;
dlthmin=140;
if dlthresh>dlthmax, dlthresh=dlthmax;
elseif dlthresh<dlthmin, dlthresh=dlthmin;
end

% determine if samples are baseline
vwindow=100;
twin1=0.033;
twin2=0.067;
% time of the 51st QRS
last=tqrs(51);
sample2=sample(1:last);
bline=zeros(size(sample2));
% a sample is baseline if I have twin1 or twin2 consecutive samples
% that vary in amplitude by no more than dlthresh
for i=1:length(sample2)-twin1*fs
    vmax=sample(i);
    vmin=sample(i);
    if abs(baseline(i)-vmax)<vwindow, twindow=twin1;
    else twindow=twin2;
    end
    ww=sample(i:i+round(twindow*fs));
    if max(ww)-min(ww)<dlthresh
        bline(i)=1;
    end
end
% for first 50 beats, look for PQ junction and J point
tlim2=0.060;
tlim3=0.100;
PQ=zeros(50,1);
J=zeros(50,1);

for j=1:50
    % search to the left
    try
        w=bline(round(tqrs(j)-tlim2*fs):tqrs(j)-1);
    catch
        display(j)
        w=bline(1:tqrs(j)-1);
    end
    f=find(w);
    if numel(f)>0
        PQ(j)=length(w)-max(f)+1;
    else
        PQ(j)=length(w);
    end
    % search to the right
```

92

```matlab
        w=bline(tqrs(j)+1:round(tqrs(j)+tlim3*fs));
        f=find(w);
        if numel(f)>0
            J(j)=min(f);
        else
            J(j)=length(w);
        end
    end
end

% incremental average
pqoff=PQ(1);
for i=1:length(PQ)
    if PQ(i)<pqoff
        pqoff=pqoff-1;
    elseif PQ(i)>pqoff
        pqoff=pqoff+1;
    end
end
jpoff=J(1);
for i=1:length(J)
    if J(i)<jpoff
        jpoff=jpoff-1;
    elseif J(i)>jpoff
        jpoff=jpoff+1;
    end
end
end
```

BIBLIOGRAPHY

[1]   S. S. Lim, T. Vos, A. D. Flaxman, G. Danaei, K. Shibuya, H. Adair-Rohani, M. A. AlMazroa, M. Amann, H. R. Anderson, K. G. Andrews, M. Aryee, C. Atkinson, L. J. Bacchus, A. N. Bahalim, K. Balakrishnan, J. Balmes, S. Barker-Collo, A. Baxter, M. L. Bell, J. D. Blore, F. Blyth, C. Bonner, G. Borges, R. Bourne, M. Boussinesq, M. Brauer, P. Brooks, N. G. Bruce, B. Brunekreef, C. Bryan-Hancock, C. Bucello, R. Buchbinder, F. Bull, R. T. Burnett, T. E. Byers, B. Calabria, J. Carapetis, E. Carnahan, Z. Chafe, F. Charlson, H. Chen, J. S. Chen, A. T.-A. Cheng, J. C. Child, A. Cohen, K. E. Colson, B. C. Cowie, S. Darby, S. Darling, A. Davis, L. Degenhardt, F. Dentener, D. C. Des Jarlais, K. Devries, M. Dherani, E. L. Ding, E. R. Dorsey, T. Driscoll, K. Edmond, S. E. Ali, R. E. Engell, P. J. Erwin, S. Fahimi, G. Falder, F. Farzadfar, A. Ferrari, M. M. Finucane, S. Flaxman, F. G. R. Fowkes, G. Freedman, M. K. Freeman, E. Gakidou, S. Ghosh, E. Giovannucci, G. Gmel, K. Graham, R. Grainger, B. Grant, D. Gunnell, H. R. Gutierrez, W. Hall, H. W. Hoek, A. Hogan, H. D. Hosgood, D. Hoy, H. Hu, B. J. Hubbell, S. J. Hutchings, S. E. Ibeanusi, G. L. Jacklyn, R. Jasrasaria, J. B. Jonas, H. Kan, J. A. Kanis, N. Kassebaum, N. Kawakami, Y.-H. Khang, S. Khatibzadeh, J.-P. Khoo, C. Kok, F. Laden, R. Lalloo, Q. Lan, T. Lathlean, J. L. Leasher, J. Leigh, Y. Li, J. K. Lin, S. E. Lipshultz, S. London, R. Lozano, Y. Lu, J. Mak, R. Malekzadeh, L. Mallinger, W. Marcenes, L. March, R. Marks, R. Martin, P. McGale, J. McGrath, S. Mehta, Z. A. Memish, G. A. Mensah, T. R. Merriman, R. Micha, C. Michaud, V. Mishra, K. M. Hanafiah, A. A. Mokdad, L. Morawska, D. Mozaffarian, T. Murphy, M. Naghavi, B. Neal, P. K. Nelson, J. M. Nolla, R. Norman, C. Olives, S. B. Omer, J. Orchard, R. Osborne, B. Ostro, A. Page, K. D. Pandey, C. D. Parry, E. Passmore, J. Patra, N. Pearce, P. M. Pelizzari, M. Petzold, M. R. Phillips, D. Pope, C. A. Pope, J. Powles, M. Rao, H. Razavi, E. A. Rehfuess, J. T. Rehm, B. Ritz, F. P. Rivara, T. Roberts, C. Robinson, J. A. Rodriguez-Portales, I. Romieu, R. Room, L. C. Rosenfeld, A. Roy, L. Rushton, J. A. Salomon, U. Sampson, L. Sanchez-Riera, E. Sanman, A. Sapkota, S. Seedat, P. Shi, K. Shield, R. Shivakoti, G. M. Singh, D. A. Sleet, E. Smith, K. R. Smith, N. J. Stapelberg, K. Steenland, H. Stöckl, L. J. Stovner, K. Straif, L. Straney, G. D. Thurston, J. H. Tran, R. Van Dingenen, A. van Donkelaar, J. L. Veerman, L. Vijayakumar, R. Weintraub, M. M. Weissman, R. A. White, H. Whiteford, S. T. Wiersma, J. D. Wilkinson, H. C. Williams, W. Williams, N. Wilson, A. D. Woolf, P. Yip, J. M. Zielinski, A. D. Lopez, C. J. Murray, and M. Ezzati, "A comparative risk assessment of burden of disease and injury attributable to 67 risk factors and risk factor clusters in 21 regions, 1990–2010: a systematic analysis for the Global Burden of Disease Study 2010," *Lancet*, vol. 380, no. 9859, pp. 2224–2260, 2012.

[2]   J. Finegold, P. Asaria, and D. Francis, "Mortality from ischaemic heart disease by country, region, and age: statistics from World Health Organisation and United Nations," *Int. J. Cardiol.*, 2013.

[3]   S. Sidney, W. Rosamond, V. Howard, and R. Luepker, "The 'heart disease and stroke statistics—2013 update' and the need for a national cardiovascular surveillance system," *Circulation*, 2013.

[4]   V. L. Roger, A. S. Go, D. M. Lloyd-Jones, E. J. Benjamin, J. D. Berry, W. B. Borden, D. M. Bravata, S. Dai, E. S. Ford, C. S. Fox, H. J. Fullerton, C. Gillespie, S. M. Hailpern, J. A. Heit, V. J. Howard, B. M. Kissela, S. J. Kittner, D. T. Lackland, J. H. Lichtman, L. D.

Lisabeth, D. M. Makuc, G. M. Marcus, A. Marelli, D. B. Matchar, C. S. Moy, D. Mozaffarian, M. E. Mussolino, G. Nichol, N. P. Paynter, E. Z. Soliman, P. D. Sorlie, N. Sotoodehnia, T. N. Turan, S. S. Virani, N. D. Wong, D. Woo, M. B. Turner, and American Heart Association Statistics Committee and Stroke Statistics Subcommittee, "Heart disease and stroke statistics--2012 update: a report from the American Heart Association.," *Circulation*, vol. 125, no. 1, pp. e2–e220, Jan. 2012.

[5]   M. Paukkunen, "Seismocardiography: Practical implementation and feasibility," 2014.

[6]   T. Fahey, A. A. Montgomery, J. Barnes, J. Protheroe, T. Fahey, and L. Fahey@, "Quality Of Care For Elderly Residents In Nursing Homes And Elderly People Living At DD2 4AD Bristol BS10 6SP Quality of care for elderly residents in nursing homes and elderly people living at home: controlled observational study," *Source BMJ Br. Med. J. BMJ*, vol. 326326, no. 7389, 2003.

[7]   N. Hill and C. Smithers, "Options for wireless technology in telemedicine and telecare applications," *J. Telemed. Telecare*, 1999.

[8]   C. O'Brien and C. Heneghan, "A comparison of algorithms for estimation of a respiratory signal from the surface electrocardiogram," *Comput. Biol. Med.*, 2007.

[9]   R. Bailón, L. Sörnmo, and P. Laguna, "A robust method for ECG-based estimation of the respiratory frequency during stress testing," *IEEE Trans. Biomed. Eng.*, vol. 53, no. 7, pp. 1273–1285, 2006.

[10]   J. E. Hall, A. C. Guyton Professor, J. F. Kennedy Blvd, M. Schenk Marketing Manager, and M. Lieberman, "Guyton and Hall Textbook of Medical Physiology," 2006.

[11]   "The Student Physiologist." [Online]. Available: https://thephysiologist.org/.

[12]   M. Tadi, "An Accelerometer-Based Method for Simultaneous Extraction of Respiratory and Cardiac Gating Signals for Nuclear Medicine Imaging," 2014.

[13]   "Drug Development Technology." [Online]. Available: http://www.drugdevelopment-technology.com/.

[14]   F. AL- Khalidi, R. Saatchi, and D. Burke, "Respiration rate monitoring methods: A review," *Pediatric*, 2011.

[15]   P. Harper, S. Kraman, and H. Pasterkamp, "An acoustic model of the respiratory tract," *IEEE Trans.*, 2001.

[16]   J. Werthammer, J. Krasner, J. DiBenedetto, and A. Stark, "Apnea monitoring by acoustic detection of airflow," *Pediatrics*, 1983.

[17]   P. Corbishley and E. Rodriguez-Villegas, "Breathing Detection: Towards a Miniaturized, Wearable, Battery-Operated Monitoring System," *IEEE Trans. Biomed. Eng.*, vol. 55, no. 1, pp. 196–204, Jan. 2008.

[18]   "Acoustic Respiration Rate." [Online]. Available: http://www.masimo.fr/rra/.

[19]   M. Folke, L. Cernerud, M. Ekström, and B. Hök, "Critical review of non-invasive respiratory monitoring in medical care," *Med. Biol.*, 2003.

[20]   J. Blom, *Monitoring of respiration and circulation*. 2003.

[21]   "RESPIRATION   SENSOR."   [Online].   Available:

http://www.thoughttechnology.com/sciencedivision/pages/products/respiration.html.

[22] G. Prisk, J. Hammer, and C. Newth, "Techniques for measurement of thoracoabdominal asynchrony," *Pediatr. Pulmonol.*, 2002.

[23] O. Mayer, R. Clayton, and A. Jawad, "Respiratory inductance plethysmography in healthy 3-to 5-year-old children," *Chest*, 2003.

[24] "Frontmatter," in *Sleep: A Comprehensive Handbook*, Hoboken, NJ, USA: John Wiley & Sons, Inc., 2005, pp. i–xxxiii.

[25] G. A. Millikan, "The Oximeter, an Instrument for Measuring Continuously the Oxygen Saturation of Arterial Blood in Man," *Rev. Sci. Instrum.*, vol. 13, no. 10, p. 434, 1942.

[26] M. Tavel, "Cardiac Auscultation A Glorious Past—But Does It Have a Future?," *Circulation*, 1996.

[27] V. Clark and J. Kruse, "Clinical methods: the history, physical, and laboratory examinations," *JAMA*, 1990.

[28] I. Edler and K. Lindström, "The history of echocardiography.," *Ultrasound Med. Biol.*, vol. 30, no. 12, pp. 1565–644, Dec. 2004.

[29] E. E. van der Wall, "Crown years for non-invasive cardiovascular imaging (Part IV): 30 years of cardiac computed tomography," *Netherlands Hear. J.*, vol. 21, no. 7–8, pp. 315–318, Jul. 2013.

[30] M. Jafari Tadi, T. Koivisto, M. Pänkäälä, and A. Paasio, "Accelerometer-Based Method for Extracting Respiratory and Cardiac Gating Information for Dual Gating during Nuclear Medicine Imaging.," *Int. J. Biomed. Imaging*, vol. 2014, p. 690124, Jan. 2014.

[31] W. Einthoven, "The string galvanometer and the human electrocardiogram," *KNAW Proc.*, 1903.

[32] A. Martínez, E. Everss, and J. Rojo-Álvarez, "A systematic review of the literature on home monitoring for patients with heart failure," *Telemed. …*, 2006.

[33] S. Lukkarinen, "Phonocardiography: Development of a clinical system and its application to screening for paediatric heart murmurs," 2012.

[34] L. Jensen, J. Yakimets, and K. Teo, "A review of impedance cardiography," *Hear. Lung J. Acute*, 1995.

[35] H. Keren, D. Burkhoff, and P. Squara, "Evaluation of a noninvasive continuous cardiac output monitoring system based on thoracic bioreactance," *Am. J.*, 2007.

[36] J. Zanetti, "Seismocardiography: A new technique for recording cardiac vibrations. concept, method, and initial observations," *J. Cardiovasc. Technol.*, 1990.

[37] O. T. Inan, P.-F. Migeotte, K.-S. Park, M. Etemadi, K. Tavakolian, R. Casanella, J. Zanetti, J. Tank, I. Funtova, G. K. Prisk, and M. Di Rienzo, "Ballistocardiography and seismocardiography: a review of recent advances.," *IEEE J. Biomed. Heal. informatics*, vol. 19, no. 4, pp. 1414–27, Jul. 2015.

[38] K. Tavakolian, B. Ngai, and A. Blaber, "Infrasonic cardiac signals: Complementary windows to cardiovascular dynamics," *Med. …*, 2011.

[39] J. Gordon, "Certain molar movements of the human body produced by the circulation of the

blood," *J. Anat. Physiol.*, 1877.

[40]   I. Starr, A. Rawson, and H. Schroeder, "Studies on the estimation of cardiac ouptut in man, and of abnormalities in cardiac function, from the heart's recoil and the blood's impacts; the," *Am. J.*, 1939.

[41]   S. Junnila, A. Akhbardeh, and A. Värri, "An EMFi-film sensor based ballistocardiographic chair: performance and cycle extraction method," *Syst. Des. ...*, 2005.

[42]   O. Inan, M. Etemadi, and R. Wiard, "Robust ballistocardiogram acquisition for home monitoring," *Physiological*, 2009.

[43]   A. Vehkaoja, S. Rajala, and P. Kumpulainen, "Correlation approach for the detection of the heartbeat intervals using force sensors placed under the bed posts," *J. Med.*, 2013.

[44]   K. Pandia, S. Ravindran, and E. Cole, "Estimation of blood flow and hemodynamic parameters from a single chest-worn sensor, and other circuits, devices and processes," *US Pat. App. 12/861,888*, 2010.

[45]   K. Tavakolian and F. Khosrow-Khavar, "Seismocardiographic adjustment of diastolic timed vibrations," *Med. ...*, 2012.

[46]   R. Bailón, L. Sörnmo, and P. Laguna, "A robust method for ECG-based estimation of the respiratory frequency during stress testing," *Biomed. Eng. IEEE*, 2006.

[47]   G. Hahn, I. Sipinková, F. Baisch, and G. Hellige, "Changes in the thoracic impedance distribution under different ventilatory conditions.," *Physiol. Meas.*, vol. 16, no. 3 Suppl A, pp. A161–73, Aug. 1995.

[48]   R. Pallas-Areny, J. Colominas-Balague, and F. J. Rosell, "The effect of respiration-induced heart movements on the ECG," *IEEE Trans. Biomed. Eng.*, vol. 36, no. 6, pp. 585–590, Jun. 1989.

[49]   D. L. Eckberg, "The human respiratory gate," *J. Physiol.*, vol. 548, no. 2, pp. 339–352, Apr. 2003.

[50]   P. Langley, E. Bowers, and A. Murray, "Principal component analysis as a tool for analysing beat-to-beat changes in electrocardiogram features: application to electrocardiogram derived respiration," *IEEE Trans. Bio. ...*, 2010.

[51]   W. EINTHOVEN, G. FAHR, and A. DE WAART, "On the direction and manifest size of the variations of potential in the human heart and on the influence of the position of the heart on the form of the electrocardiogram.," *Am. Heart J.*, vol. 40, no. 2, pp. 163–211, Aug. 1950.

[52]   J. T. Flaherty, S. D. Blumenschein, A. W. Alexander, R. D. Gentzler, T. M. Gallie, J. P. Boineau, M. S. Spach, M. S. Spach, et al., H. V. Pipberger, F. W. Stallman, K. Yano, H. W. Draper, C. A. Caceres, et al., B. Taccardi, D. A. Brody, E. R. Powsner, L. H. Nahum, and A. Mauro, "Influence of respiration on recording cardiac potentials," *Am. J. Cardiol.*, vol. 20, no. 1, pp. 21–28, Jul. 1967.

[53]   H. Riekkinen and P. Rautaharju, "Body position, electrode level, and respiration effects on the Frank lead electrocardiogram.," *Circulation*, vol. 53, no. 1, pp. 40–5, Jan. 1976.

[54]   P. Grossman and K. Wientjes, "Respiratory Sinus Arrhythmia and Parasympathetic Cardiac Control: Some Basic Issues Concerning Quantification, Applications and Implications," in

Cardiorespiratory and Cardiosomatic Psychophysiology, Boston, MA: Springer US, 1986, pp. 117–138.

[55] R. C. Wang and T. W. Calvert, "A model to estimate respiration from vectorcardiogram measurements," *Ann. Biomed. Eng.*, vol. 2, no. 1, pp. 47–57, Mar. 1974.

[56] T. Y. Young and W. H. Huggins, "The Intrinsic Component Theory of Electrocardiography," *Ire Trans. Biomed. Electron.*, vol. 9, no. 4, pp. 214–221, 1962.

[57] G. Moody, R. Mark, and M. Bump, "Clinical validation of the ECG-derived respiration (EDR) technique," *Group*, 1986.

[58] D. Caggiano and S. Reisman, "Respiration derived from the electrocardiogram: a quantitative comparison of three different methods," in *Proceedings of the IEEE 22nd Annual Northeast Bioengineering Conference*, 1996, pp. 103–104.

[59] C. Lamberti and J. de Bie, "Validation of an ECG-derived respiration monitoring method," *Comput. Cardiol. 2003*, 2003.

[60] D. Dobrev and I. Daskalov, "Two-electrode telemetric instrument for infant heart rate and apnea monitoring," *Med. Eng. Phys.*, vol. 20, no. 10, pp. 729–734, 1999.

[61] C. L. Mason and L. Tarassenko, "Quantitative assessment of respiratory derivation algorithms," in *2001 Conference Proceedings of the 23rd Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, 2001, vol. 2, pp. 1998–2001.

[62] B. Raymond, R. M. Cayton, R. A. Bates, and M. Chappell, "Screening for obstructive sleep apnoea based on the electrocardiogram-the computers in cardiology challenge," in *Computers in Cardiology 2000. Vol.27 (Cat. 00CH37163)*, 2000, pp. 267–270.

[63] M. Varanini, M. Emdin, F. Allegri, M. Raciti, F. Conforti, A. Macerata, A. Taddei, R. Francesconi, G. Kraft, A. L. Abbate, and C. Marchesi, "Adaptive filtering of ECG signal for deriving respiratory activity," in *[1990] Proceedings Computers in Cardiology*, 1990, pp. 621–624.

[64] K. Behbehani, S. Vijendra, and J. Burk, "An investigation of the mean electrical axis angle and respiration during sleep," *Med. ...*, 2002.

[65] P. Langley, J. Bowers, and a Murray, "Principal Component Analysis as a Tool for Analysing Beat-to-Beat Changes in Electrocardiogram Features: Application to Electrocardiogram Derived Respiration.," *IEEE Trans. Biomed. Eng.*, vol. PP, no. 99, p. 1, 2009.

[66] D. Widjaja, J. Taelman, S. Vandeput, M. A. Braeken, R. A. Otte, B. R. Van Den Bergh, and S. Van Huffel, "ECG-Derived Respiration: Comparison and New Measures for Respiratory Variability," vol. 37. pp. 149–152, 2010.

[67] C. Orphanidou, S. Fleming, S. A. Shah, and L. Tarassenko, "Data fusion for estimating respiratory rate from a single-lead ECG," *Biomed. Signal Process. Control*, vol. 8, no. 1, pp. 98–105, 2013.

[68] D. Labate, F. La Foresta, G. Occhiuto, F. C. Morabito, A. Lay-Ekuakille, and P. Vergallo, "Empirical Mode Decomposition vs. Wavelet Decomposition for the Extraction of Respiratory Signal From Single-Channel ECG: A Comparison," *IEEE Sens. J.*, vol. 13, no. 7, pp. 2666–2674, Jul. 2013.

[69] S. Tiinanen, K. Noponen, M. Tulppo, A. Kiviniemi, and T. Seppänen, "ECG-derived respiration methods: adapted ICA and PCA.," *Med. Eng. Phys.*, vol. 37, no. 5, pp. 512–7, May 2015.

[70] P. Langley, E. Bowers, and A. Murray, "Principal component analysis as a tool for analyzing beat-to-beat changes in ECG features: application to ECG-derived respiration," *... , IEEE Trans.*, 2010.

[71] P. Hung and S. Bonnet, "Estimation of respiratory waveform using an accelerometer," *... Imaging From Nano ...*, 2008.

[72] P. Castiglioni, A. Faini, G. Parati, and M. Di Rienzo, "Wearable seismocardiography.," *Conf. Proc. ... Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. IEEE Eng. Med. Biol. Soc. Annu. Conf.*, vol. 2007, pp. 3954–7, 2007.

[73] K. Pandia, O. T. Inan, G. T. A. Kovacs, and L. Giovangrandi, "Extracting respiratory information from seismocardiogram signals acquired on the chest using a miniature accelerometer," *Physiol. Meas.*, vol. 33, no. 10, pp. 1643–1660, Oct. 2012.

[74] D. Sánchez Morillo, J. L. R. Ojeda, L. F. C. Foix, and A. L. Jiménez, "An accelerometer-based device for sleep apnea screening," *IEEE Trans. Inf. Technol. Biomed.*, vol. 14, no. 2, pp. 491–499, 2010.

[75] T. Reinvuo and M. Hannula, "Measurement of respiratory rate with high-resolution accelerometer and EMFit pressure sensor," *... . Proc. ...*, 2006.

[76] K. Tavakolian, A. Vaseghi, and B. Kaminska, "Improvement of ballistocardiogram processing by inclusion of respiration information.," *Physiol. Meas.*, vol. 29, no. 7, pp. 771–781, Jul. 2008.

[77] A. Jin, B. Yin, and G. Morren, "Performance evaluation of a tri-axial accelerometry-based respiration monitoring for ambient assisted living," *Eng. Med. ...*, 2009.

[78] A. M. Chan, N. Ferdosi, and R. Narasimhan, "Ambulatory respiratory rate detection using ECG and a triaxial accelerometer.," *Conf. Proc. ... Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. IEEE Eng. Med. Biol. Soc. Annu. Conf.*, vol. 2013, pp. 4058–61, 2013.

[79] K. Pandia, O. T. Inan, and G. T. A. Kovacs, "A frequency domain analysis of respiratory variations in the seismocardiogram signal.," *Conf. Proc. ... Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. IEEE Eng. Med. Biol. Soc. Annu. Conf.*, vol. 2013, pp. 6881–4, 2013.

[80] H. R. Yoon, J. W., Noh, Y. S., Kwon, Y. S., Kim, W. K., & Yoon, "Improvement of Dynamic Respiration Monitoring Through Sensor Fusion of Accelerometer and Gyro-sensor." Journal of Electrical Engineering & Technology, 2014.

[81] V. Zakeri and K. Tavakolian, "Identification of respiratory phases using seismocardiogram: A machine learning approach," in *2015 Computing in Cardiology Conference (CinC)*, 2015, pp. 305–308.

[82] V. Zakeri and K. Tavakolian, "Preliminary results on quantification of seismocardiogram morphological changes, using principal component analysis," *... Med. ...*, 2014.

[83] E. Bowers, A. Murray, and P. Langley, "Respiratory rate derived from principal component analysis of single lead electrocardiogram," *Comput. Cardiol. ...*, 2008.

[84] A. Kaw, E. Kalu, and D. Nguyen, "Numerical Methods with Applications," *Textbooks

Collection. 2009.

[85]   I. V. Blagouchine and E. Moreau, "Analytic Method for the Computation of the Total Harmonic Distortion by the Cauchy Method of Residues," *IEEE Trans. Commun.*, vol. 59, no. 9, pp. 2478–2491, Sep. 2011.

[86]   R. Cabot, "Fundamentals of modern audio measurement," *Audio Eng. Soc. Conf. UK 12th …*, 1997.

[87]   R. Eklund and G. Tenenbaum, *Encyclopedia of sport and exercise psychology*. 2014.

[88]   M. A. F. Pimentel, G. D. Clifford, and D. A. Clifton, "Bayesian fusion of algorithms for the robust estimation of respiratory rate from the photoplethysmogram," in *2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, 2015, pp. 6138–6141.