A Software Toolkit for Acoustic Respiratory Analysis

by

Gina Ann Yi

Submitted to the Department of Electrical Engineering and Computer Science in partial fulfillment of the requirements for the degrees of

Bachelor of Science in Electrical Engineering and Computer Science and Master of Engineering in Electrical Engineering and Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

May 24, 2004

Author..
Department of Electrical Engineering and Computer Science
May 24, 2004

Certified by...
John V. Guttag
Professor, Computer Science and Engineering
Thesis Supervisor

Accepted by...
Arthur C. Smith
Chairman, Department Committee on Graduate Theses

A Software Toolkit for Acoustic Respiratory Analysis

by

Gina Ann Yi

Submitted to the
Department of Electrical Engineering and Computer Science
May 24, 2004
in partial fulfillment of the requirements for the degree of
Bachelor of Science in Electrical Engineering and Computer Science and
Master of Engineering in Electrical Engineering and Computer Science

Abstract

Millions of Americans suffer from pulmonary diseases. According to recent statistics, approximately 17 million people suffer from asthma, 16.4 million from chronic obstructive pulmonary disease, 12 million from sleep apnea, and 1.3 million from pneumonia – not to mention the prevalence of many other diseases associated with the lungs. Annually, the mortality attributed to pulmonary diseases exceeds 150,000.

Clinical signs of most pulmonary diseases include irregular breathing patterns, the presence of abnormal breath sounds such as *wheezes* and *crackles,* and the absence of breathing entirely. Throughout the history of medicine, physicians have always listened for such sounds at the chest wall (or over the trachea) during patient examinations to diagnose pulmonary diseases – a procedure also known as *auscultation.*

Recent advancements in computer technology have made it possible to record, store, and digitally process breath sounds for further analysis. Although automated techniques for lung sound analysis have not been widely employed in the medical field, there has been a growing interest among researchers to use technology to understand the subtler characteristics of lung sounds and their potential correlations with physiological conditions. Based on such correlations, algorithms and tools can be developed to serve as diagnostic aids in both the clinical and non-clinical settings.

We developed a *software toolkit,* using MATLAB, to objectively characterize lung sounds. The toolkit includes a respiration detector, respiratory rate detector, respiratory phase onset detector, respiratory phase classifier, crackle and wheeze detectors and characterizers, and a time-scale signal expander. This document provides background on lung sounds, describes and evaluates our analysis techniques, and compares our work to approaches in other diagnostic tools.

Thesis Supervisor: John V. Guttag
Title: Professor, Computer Science and Engineering

Contents

Chapter 1

Introduction

1.1 Motivation

Pulmonary (lung) diseases are among the top causes of death in the United States. In 1999, approximately 8% of a total of 2.4 million deaths were attributed to chronic lower respiratory diseases, influenza, and pneumonia [24]. Recent statistics estimate that, annually, 17 million people in the U.S. suffer from asthma [32], 16.4 million from chronic obstructive pulmonary disease [32], 12 million from sleep apnea [1], and 1.3 million from pneumonia [4]. Other prominent lung diseases include interstitial fibrosis, congestive heart failure, asbestosis, scleroderma, rheumatoid lung disease, and pneumothorax.

Clinical signs of pulmonary diseases include irregular breathing patterns, the presence of adventitious breath sounds (i.e., wheezes and crackles), and even the absence of breathing entirely. The practice of listening for such sounds is called *auscultation.* Auscultatory techniques have been employed by physicians throughout the history medicine to diagnose disease, with the earliest known practice dating as far back as 400 B.C. by the Greek physician (known as the "father of medicine") Hippocrates [36]. In 1819, the French physician Rene T. H. Laennec invented an instrument called the *stethoscope,* designed to aid physicians during auscultation [34]. Laennec's monaural stethoscope evolved into today's omnipresent binaural stethoscope.

The stethoscope was the most important tool of respiratory medicine until the invention of radiography (or the X-ray) in the early 20th century. X-ray dominated medical diagnostics ever since its discovery primarily because of its reliability in producing an accurate picture of the structural abnormalities of the lungs [11]. Because recent advancements in computer and sensor technology have made it convenient to record, store, and digitally process breath sounds, a resurgent interest has developed among medical researchers in identifying and understanding the correlation between the acoustical signal and the underlying physiologic behavior and anatomical structure of the lungs.

Rough correlations between lung acoustics, anatomy, and physiology have been identified [13] [14]. However, identifying precise correlations and developing a comprehensive model of lung acoustics remains a challenge because of the complexity of the respiratory system [30]. Despite the challenges of understanding lung acoustics, a major appeal of this area of research is the potential to use automated and computerized auscultatory techniques as a reliable, inexpensive, portable, and non-invasive alternative to other existing more expensive and invasive diagnostic methods.

Potential applications for automatic lung sound analysis include: (1) monitoring a chronic respiratory disease over extended periods of time to assess its severity and progression, and (2) diagnosing or screening for pulmonary conditions and/or pathologies.

1.2 Goal

In the past few decades, substantial research has been done on using digital signal-processing and artificial intelligence techniques to analyze lung sounds [5] [35] [29]. However, not many conclusions have been drawn about the direct correlation between specific pulmonary diseases and sounds. Research has been limited by the difficulties in processing the acoustic data. The main objective of our research was to address this technical difficulty by building a *software toolkit* that could be used to objectively characterize lung sounds. Researchers could use this toolkit to more easily develop higher-level algorithms for lung sound analysis and disease diagnostics.

Our software toolkit provides a wide variety of functions. Figure 1 summarizes the structure of our toolkit.

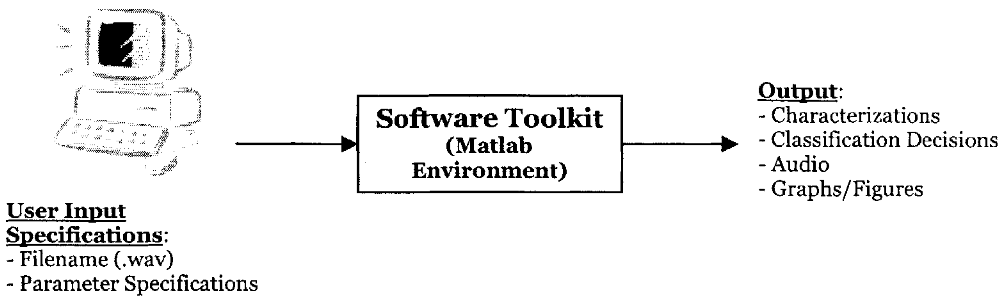


Figure 1. System block diagram of our software toolkit for analyzing lung sounds.

The tools in our toolkit are summarized below.

- **Respiration Detection, Respiratory Rate Detection, and Respiratory Phase Onset Detection** – This function detects regions in the acoustic signal over which breathing is present; over those regions, it computes the respiratory rate and the onset-times of each respiratory phase.

- **Respiratory Phase Classification** – This function classifies a respiratory phase as being either inspiration or expiration.

- **Crackles Analysis** – This function detects, counts, and identifies the locations in time of crackles. It characterizes each crackle according to its pitch and timing.

- **Wheezes Analysis** – This function detects, counts, and identifies the locations in time of wheezes. It characterizes each wheeze according to its fundamental frequency and energy as a percent of the total energy in the signal.

- **Time-Scale Waveform Expansion** – This function allows the user to stretch the signal by an integral factor and play it back such that the overall frequency content of the signal is minimally distorted.

These tools could be used in a variety of applications. A respiration detector would be useful for diagnosing and monitoring patients with *sleep apnea,* a common disorder that causes frequent and intermittent interruptions in breathing during sleep. A wheeze detector would be useful for diagnosing and assessing the severity of diseases such as *asthma*, a chronic respiratory condition characterized by the inflammation and obstruction of the upper respiratory tract, which results in restricted, labored breathing and wheezing. A crackle detector would be useful for diagnosing and assessing the severity of diseases such as *pneumonia*, an infectious disease that causes the lungs to become fluid-filled. A time-scale waveform expander would facilitate the aural detection of lung sounds that are difficult to detect at normal speeds.

1.3 Approach

A variety of techniques have been employed by researchers over recent decades to analyze and classify respiratory sounds [7] [5] [8] [9] [39]. These techniques rely extensively on the fields of digital signal-processing and machine learning.

The most commonly used signal-processing tool is the *Fourier transform*, which represents the signal in terms of its frequency components. A popular extension of the Fourier transform is the *short-time Fourier transform,* which allows the signal to be viewed as a function of both time and frequency. Time-frequency analysis is particularly useful in the application of lung sound analysis because of its ability to temporally resolve the frequency components of a signal, giving rise to a more accurate representation of the typically non-stationary lung sounds data.

Machine learning lends itself readily to medical applications such as disease diagnostics. Machine learning refers to teaching a system to assign class labels to data samples using an existing set of labeled data. Examples of machine learning methods include maximum likelihood probabilistic modeling [18], neural networks, support vector machines [3], and k-nearest neighbors. Our *respiratory phase classifier* uses support vector machines to classify inspiration and expiration.

Figure 2 shows the implementation architecture of the tools in our toolkit.

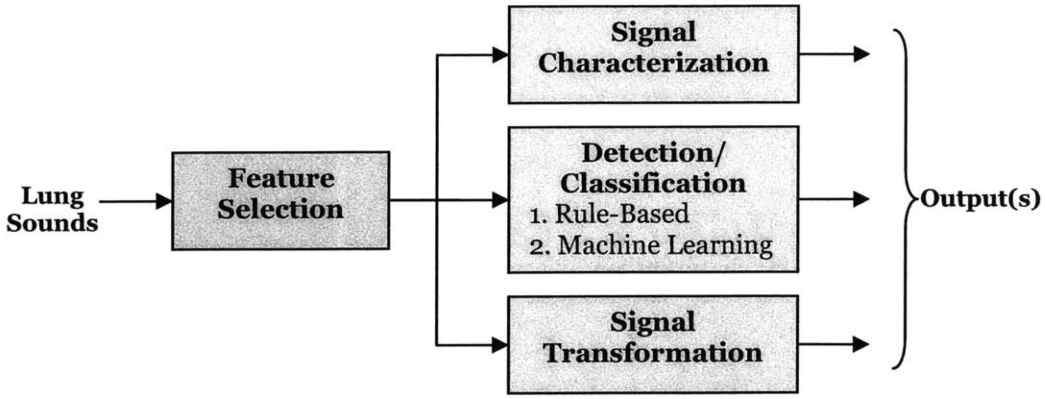


Figure 2. Implementation architecture of the tools in our software toolkit.

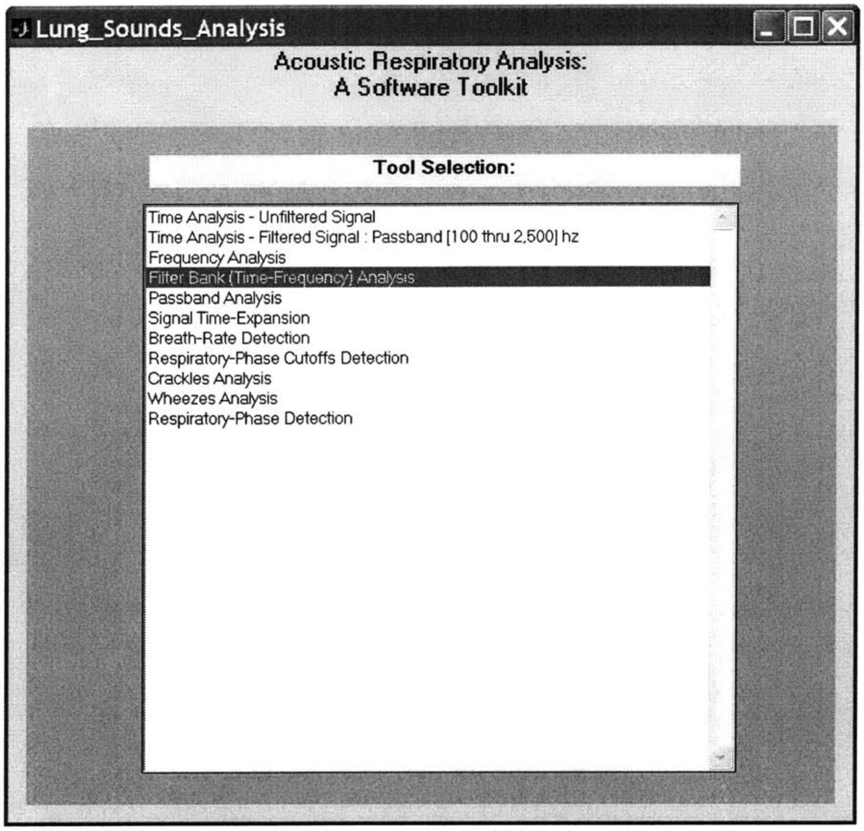
Most of the lung sound data used in this work were recorded using an electronic stethoscope (developed by Meditron and distributed by Welch-Allyn). A few data files were gathered from publicly-accessible lung sound repositories available on the World Wide Web [33] [40].

We implemented our toolkit – which includes its functions and graphical user interface (GUI)) – on the Matlab software environment.

1.4 Screen Shots

In this section, we provide examples of screen shots of the toolkit's GUI and graphical system outputs.

Users can select a tool using the *Tool Selection* GUI, as is shown in Figure 3. They can input at the command terminal the name of the (.wav) file containing the lung sounds of interest, and the requested parameter values. Figure 4 shows a screen shot of the GUI for the *Filter Bank (Time-Frequency) Analysis* tool. The system then processes the request and outputs relevant information about the data. Figure 5 is an example of a graphical output of the *Filter Bank (Time-Frequency) Analysis* tool for a particular lung sounds file that contains wheezes.



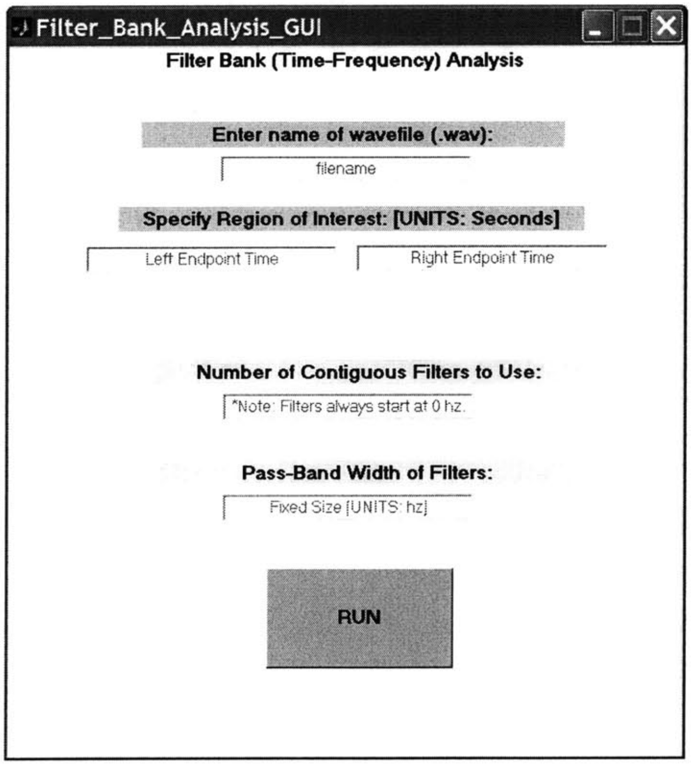
Lung_Sounds_Analysis

Acoustic Respiratory Analysis:
A Software Toolkit

Tool Selection:

- Time Analysis - Unfiltered Signal
- Time Analysis - Filtered Signal : Passband [100 thru 2,500] hz
- Frequency Analysis
- Filter Bank (Time-Frequency) Analysis
- Passband Analysis
- Signal Time-Expansion
- Breath-Rate Detection
- Respiratory-Phase Cutoffs Detection
- Crackles Analysis
- Wheezes Analysis
- Respiratory-Phase Detection

Figure 3. Screen shot of *Tool Selection* GUI.



Filter_Bank_Analysis_GUI

Filter Bank (Time-Frequency) Analysis

Enter name of wavefile (.wav):

filename

Specify Region of Interest: [UNITS: Seconds]

Left Endpoint Time

Right Endpoint Time

Number of Contiguous Filters to Use:

*Note: Filters always start at 0 hz.

Pass-Band Width of Filters:

Fixed Size [UNITS: hz]

Figure 4. Screen shot of the *Filter Bank (Time-Frequency) Analysis* tool GUI.

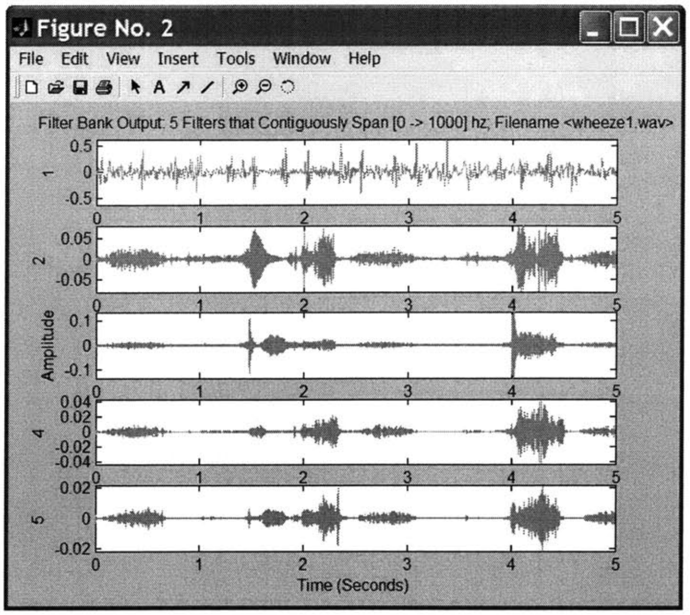


Figure No. 2
File Edit View Insert Tools Window Help
Filter Bank Output: 5 Filters that Contiguously Span [0 -> 1000] hz; Filename <wheeze1.wav>
Amplitude
1
2
4
5
0.5
0
-0.5
0.05
0
-0.05
0.1
0
-0.1
0.04
0.02
0
-0.02
-0.04
0.02
0
-0.02
0
1
2
3
4
5
Time (Seconds)

Figure 5. Screen shot of a graphical output of the *Filter Bank (Time-Frequency) Analysis* tool.

1.5 Thesis Contributions

We describe here the key contributions of this thesis.

- **Novel algorithmic perspectives in acoustic respiratory analysis.** Our approaches to classifying, detecting, and characterizing various lung sounds differ from other techniques that have been proposed in the related literature. For example, our method for respiratory phase classification differs from that proposed by [6] in the techniques uses to make classification decisions. Our approach uses *support vector machines* to classify respiratory phases, whereas [6] uses a rule-based method. Our technique uses one channel of data (tracheal breath sounds), whereas [6] uses two channels of data (tracheal breath sounds and chest sounds). Also, while others have focused their research on developing generalized methods for respiratory phase classification (which classify data from different subjects using the same classifier), we experimented with both generalized and subject-specific methods.

- **Wide range of functionalities integrated into a software toolkit.** The functions in this toolkit are able to provide the basic tools that researchers can use to analyze a variety of lung sounds when developing higher-level algorithms for diagnosing pulmonary conditions. For example, if a researcher would like to explore the possibility of classifying the pulmonary diseases *pneumonia* and *congestive heart failure (CHF)*, this toolkit could be used to identify and characterize physiologically-relevant features in the lung sounds – this would eliminate the overhead associated with extracting the basic features of the signal.

1.6 Thesis Organization

We start this thesis with a background on the human lungs and respiratory system, pulmonary pathologies and diagnostics, and lung sounds in Chapter 2. We discuss our respiration, respiratory rate, and respiratory phase onset detectors in Chapter 3. We present an overview of machine learning concepts and support vector machines theory in Chapter 4, along with a discussion of our approach to respiratory phase classification. Chapters 5 and 6 propose techniques for detecting and characterizing crackles and wheezes, respectively. We review phase vocoder theory and describe our method for time-scale expanding lung sounds in Chapter 7. We conclude this thesis in Chapter 8 with a brief overview of our work on acoustic respiratory analysis, suggestions for future work, and final remarks.

Chapter 2

Lungs Background[1]

The human lungs are a pair of organs whose primary function is for respiration, i.e., to supply the blood with oxygen (O_2) from the air and eliminate carbon dioxide (CO_2) from the blood. The lungs also protect the body against airborne irritants and infectious agents (e.g., bacteria and viruses).

2.1 Anatomy and Physiology

The lungs consist of the right and left lungs, both of which are spongy and cone-shaped; they extend from the trachea to below the heart and occupy most of the thorax. Figure 1 is a diagram of the lungs.

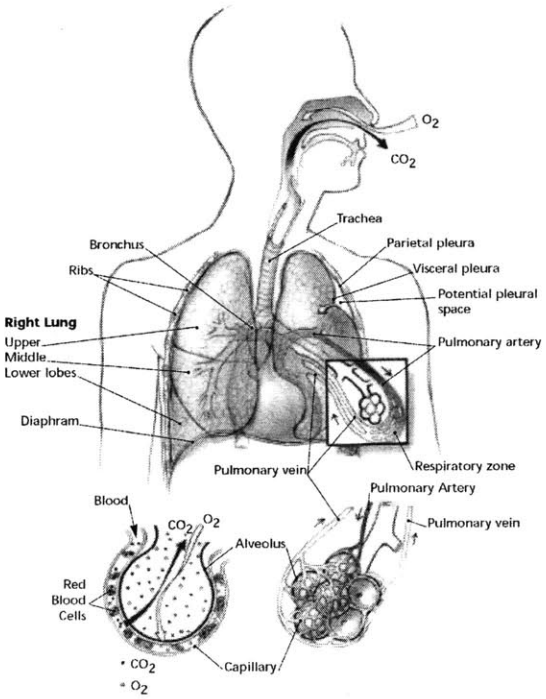


Figure 1. Diagram of the anatomy and physiology of the human lungs.

[1] We borrowed most the material for this chapter from [23] and [20].

During respiration, air enters the body from the nose or mouth, travels through airways that begin with the trachea, branch into the right and left main bronchi and further into successively smaller generations of the bronchi and bronchioles, and end at (~ 3 million) air sacs called alveoli. The exchange of O_2 and CO_2 between the inhaled air and the blood occurs in the alveoli via a surrounding network of pulmonary capillaries.

Respiration occurs in cycles, each of which consists of two phases: inspiration and expiration. During inspiration, the diaphragm (a muscle that separates the thorax and abdomen) and muscles of the rib cage contract, expanding the thoracic cavity and drawing air into the lungs. During expiration, these respiratory muscles relax, reducing the thoracic cavity back to its resting size and deflating the lungs.

2.2 Pathology

Pathologies associated with the lungs fall into two broad categories: (1) obstructive diseases, and (2) restrictive diseases. Obstructive diseases cause obstruction of the airway due to excessive mucus secretion, inflammation and/or collapse of the airway walls. Examples of obstructive diseases include asthma, chronic bronchitis, chronic laryngitis, emphysema, and neoplasms of the larynx. Restrictive diseases are characterized by interstitial fibrosis (i.e., scarring of the connective tissue of blood vessels and air sacs in the lungs) or inflammation, which cause the lungs to stiffen or to solidify from fluid secretions in the airway. Examples of restrictive diseases include sarcoidosis, pulmonary fibrosis, asbestosis, pneumonia, congestive heart failure, scleroderma, tuberculosis, and rheumatoid lung disease.

2.2.1 Diagnostic Techniques

Pulmonary pathologies are diagnosed using a variety of techniques, the most common of which include auscultation, spirometry, and X-rays.

Auscultation refers to listening for sounds from within the body in order to understand the condition of the lungs. Physicians perform auscultation with the aid of a stethoscope, an instrument designed to amplify and attenuate certain frequencies of interest.

Spirometry is a technique that measures lung function by means of a device called a spirometer. A spirometer measures the volume of air that is expelled from the lungs and the airflow rate during forced expiration. A useful parameter for assessing degree of airflow obstruction is the ratio between FEV1 (the volume of air expired during the first second of forced expiration) and FVC (the forced vital capacity of the lungs). For healthy lungs, this ratio ranges between 75 and 85 percent. Numbers that fall below this range indicate airway obstruction, with lower numbers corresponding to greater severity. Spirometry is commonly used to diagnose obstructive pulmonary diseases (e.g., asthma [32]).

An X-ray is a radiographic imaging technique that passes radiation through the chest and onto photosensitive film. This produces an image of the internal structure and state of the chest (called a radiograph). X-ray is commonly used to diagnose restrictive pulmonary diseases (e.g., pneumonia [22]).

2.3 Lung Sounds

Lung sounds (also called breath sounds) – listened for during auscultation – can be heard over the trachea and chest wall. In this section we discuss characteristics of normal, abnormal, and adventitious lung sounds, the first of which is generated by healthy lungs, and the latter two of which are generated by diseased lungs.

2.3.1 Normal Sounds

Normal lung sounds heard over the trachea are called *tracheal breath sounds*. Tracheal breath sounds have a quality similar to that of "white noise" because of their wide-band characteristic in the frequency domain, and their randomly varying amplitude in the time domain. The sound energy spans 200 to 2,000 Hz and lacks a clear pitch. Tracheal breath sounds are produced by turbulent airflow in the main and central airways (e.g., trachea and main bronchi).

Normal lung sounds that are heard over the chest wall are called *vesicular sounds.* The energy in vesicular sounds is concentrated below 200 Hz; frequencies above 200 Hz are attenuated at a rate of 10-20 dB per octave. Vesicular sounds are produced by turbulent airflow in larger airways, and possibly vortices in the more peripheral airways. Their energy is limited to lower frequency components compared to that of tracheal breath sounds because the interface between the air-filled lungs and solid chest wall acts as a low pass filter, selectively filtering out frequencies above the 200 Hz cutoff frequency.

2.3.2 Abnormal Sounds

Abnormal breath sounds refer to lung sounds heard over the chest and are also called *bronchial breathing.* They are characterized by the presence of energy in both the low and high frequencies, and have frequency characteristics similar to that of tracheal breath sounds. Bronchial breathing is caused by: (1) **consolidation** of the lungs (i.e., solidification of the lungs due to fluid-filled airways), (2) atelectasis (partial or complete collapse of the lungs), or (3) fibrosis. The presence of substantial energy in high frequencies (i.e., those above 200 Hz) is due to the change in the sound medium (in this case, the lungs) from air to fluid/solid, which reduces the low pass filter effect at the lungs/chest-wall interface. Bronchial breathing due to consolidation is commonly associated with diseases such as pneumonia and congestive heart failure.

2.3.3 Adventitious Sounds

Adventitious breath sounds consist of two types: crackles and wheezes. Crackles and wheezes can be heard both over the trachea and chest wall.

Crackles are short, explosive, non-musical sounds. Crackles that are heard over the trachea are associated with obstructive pulmonary diseases, and are produced by the passage of air bubbles through partially obstructed main airways. They typically occur early in the inspiratory phase, and in the expiratory phase as well. Crackles that are heard over the chest wall are associated with restrictive diseases, and are produced by the sudden opening of peripheral airways (especially the alveoli) from fully a deflated state. They typically occur late in the inspiratory phase.

Wheezes are musical sounds with distinct pitches. They are found most often in obstructive diseases, and are produced when air passes through slightly occluded airways, causing the walls of the airways to oscillate.

2.4 Summary

Our work focuses on the development of tools that could be useful for objectively analyzing lung function using auscultation (and, hence, lung sounds). The merits of auscultation include its convenience and non-invasiveness.

Chapter 3

Respiration Detection, Respiratory Rate Detection, and Respiratory Phase Onset Detection

In this chapter we propose an acoustically-based method for detecting: (1) respiration (i.e., whether breathing is present), (2) respiratory rate (i.e., number of respiratory cycles per minute), and (3) respiratory phase onset (i.e., start-time of inspiration or expiration). Our method uses recordings of tracheal breath sounds.

3.1 Introduction

Detecting respiration and respiratory rate is important in many clinical applications, including diagnosing sleep disorders (e.g., sleep apnea), monitoring critically ill patients in intensive care, and monitoring neonates [17] [38]. In these applications, the presence/absence of breathing and the frequency at which breathing occurs provides crucial information about the condition of the patient.

Movement of the lungs during respiration can affect heart flow and radiographic image acquisition. Compensating for these effects is useful: (1) for studying heart flow, and (2) during investigations of computer tomography, radiation therapy, or magnetic resonance imaging (MRI) [17]. Respiratory phase onset detection provides information about the precise timings (or onsets) of breath phases, and can be used to compensate for the undesirable effects of lung-movement in the aforementioned applications.

3.2 Motivation

Our goal was to design an acoustically-based system that could be used as an inexpensive and convenient tool for monitoring respiration and respiratory rate, and identifying respiratory phase onsets. Our goal for respiratory rate detection was to achieve a high sensitivity rate (i.e., percent of respiratory cycles correctly detected) and specificity rate (i.e., percent of events that are not respiratory cycles correctly classified). For respiratory phase onset detection, our goal was to identify onset-times that are in "close proximity" to the actual onset-times. More specifically, we sought to identify onset-times to within a few hundred milliseconds (~300 ms) of the actual onset-times (i.e., achieve a *timing accuracy* of less than ~300 ms), since this is the accuracy required by applications such as studying heart flow [17].

3.3 System Design

We built an algorithm that uses a single-channel input of **tracheal breath sounds** to identify regions in time over which breathing is present/absent; for those regions over which breathing is present, the algorithm computes the respiratory rate (in units of cycles per minute) and identifies onset times for each of the detected respiratory phases. Figure 1 show the general structure of our detection scheme.

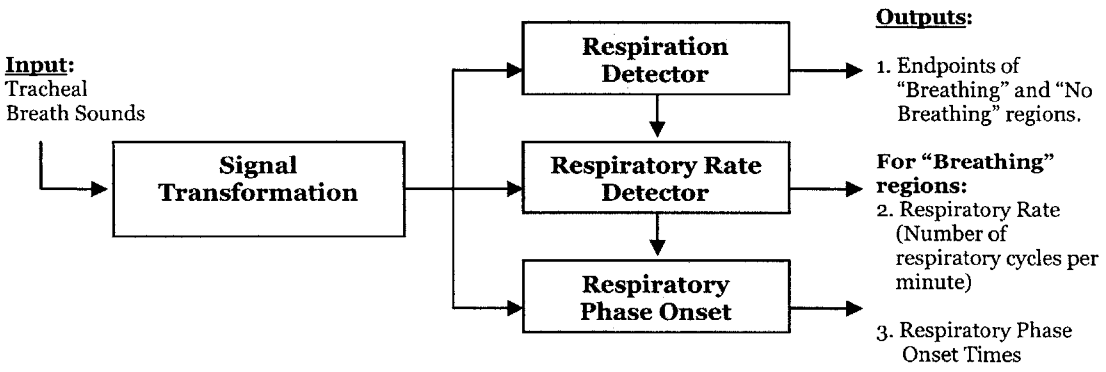

Input:
Tracheal
Breath Sounds
Signal
Transformation
Respiration
Detector
Respiratory Rate
Detector
Respiratory
Phase Onset
Outputs:
1. Endpoints of "Breathing" and "No Breathing" regions.
For "Breathing" regions:
2. Respiratory Rate (Number of respiratory cycles per minute)
3. Respiratory Phase Onset Times

Figure 1. Block diagram of system design for detecting respiration, respiratory rate, and respiratory phase onset.

The tracheal breath sound is initially processed in the *signal transformation* stage, which transforms the original time-amplitude signal into a representation that is suitable for feature extraction in the later stages. Our detection scheme:

- Uses purely time-domain information about the acoustic signal,
- Employs adaptive filtering techniques to account for the non-stationary quality of the signal, and
- Uses a series of rules to arrive upon a final decision.

The combination of the above characteristics makes our detector unique in comparison to the detectors that have been described in published literature of related work.

3.4 Background Literature/Related Work

A substantial amount of work has been published on research in respiration detection and respiratory phase onset detection. While many articles do not propose methods for respiratory rate detection explicitly, their methods for respiratory phase-onset (timing) detection implicitly propose methods for respiratory rate detection, since respiratory rate is only a function of the number of respiratory phase onsets and the duration over which the corresponding respiratory phases span. In this section, we briefly discuss related work on respiration detection and respiratory phase onset detection.

Respiration Detection

One approach for respiration detection [17] uses frequency characteristics of tracheal breath sound recordings. A "Breathing Timing Index (BTI)" is computed and compared to a threshold to detect the presence or absence of breathing in the signal. The BTI is equal to the sum of the fast Fourier transform coefficients over the [400 → 700] Hz frequency range. The threshold is defined as three times the average BTI during *apnea* (or absence of breathing). The BTI is computed over contiguous windows that span 26 milliseconds. For a particular window, if the BTI exceeds the threshold, then the detector indicates breathing; if the BTI falls below the threshold, then the detector indicates no-breathing. No performance figures are reported for this detection scheme.

Another approach for respiration detection uses nasal airflow (NAF) data instead of acoustical data, and employs a feed-forward *artificial neural network (ANN)*[2] [38]. The ANN has two hidden layers in addition to the input and output layers, uses sigmoid activation functions, and was trained using a back-propagation technique. The input to the ANN is a 50-dimensional feature vector that is the union of two 25-point signals which are derived from a normalized version of the NAF signal over a 16-seconds time-frame. The two 25-point signals are called "instantaneous respiration amplitude (IRA)" and "instantaneous respiration interval (IRI)." Regions in time over which no-breathing is detected are generally characterized by low IRA values and high IRI values – the trends (for the IRA and IRI) tend to be reversed in regions over which breathing is detected. The resulting percentage for the sensitivity and specificity of the ANN for test examples labeled as "normal breathing" were 98.4% and 94%, respectively; the percentages for test examples labeled as "apnea" were 97% and 88.7%, respectively[3].

Respiratory Phase Onset Detection

The work presented in [17], and discussed in the previous section, detects respiratory phase onsets with the "Breathing Timing Index (BTI)" signal. The respiratory phase onsets are defined to be the points in time at which the BTI classifications switch from "no-breathing" to "breathing." Their approach has an expected timing accuracy of approximately 36 ms.

[6] proposes another method for detecting respiratory phase onsets. The algorithm uses two channels of respiratory data: (1) tracheal sounds, and (2) chest sounds. The chest sounds are first used to determine the points in time at which "inspiratory peaks" occur (see the "Background Literature/Related Work" section of the "Respiratory Phase Classification" chapter for algorithm details). The tracheal signal is then used to identify potential onset-times, and is processed by computing the average fast Fourier transform power in the

[2] This article presented four different ANN approaches that differ in the feature vectors they use as inputs to their respective ANNs. We focus our discussion on the ANN referred to as "N3."

[3] For analyzing the performance of a two-class classification scheme: if we denote the classes as c1 and c2, then the sensitivities of c1 and c2 should equal the specificities of c2 and c1, respectively. In this case, the article reports two separate sets of sensitivity and specificity rates for detecting "normal breathing" and "apnea," with values that are inconsistent with the previous statement. This is because their analyses included a **third** class of data ("hypopnea," a respiratory condition characterized by partial apnea) that we do not consider in our own analyses.

[150→600] Hz frequency range within windows of duration 100 ms, overlapped by 50%. In the processed signal (which we will denote as $p[n]$), the local minima are considered to be potential onsets. Any potential onset that lies within 200 ms of the nearest inspiratory peak is discarded on the premise that the distance between the peak and onset should not be less than approximately 500 ms for a typical breath phase duration of 1 s. Furthermore, the local mean ($m[n]$) and standard deviation ($s[n]$) for each sample of $p[n]$ is computed using a window size of 1 s (again, the duration of a typical breath phase). Any potential onset with a value $p[n] > \{m[n] - 0.5s[n]\}$ is discarded. The average distance between respiratory cycles is estimated as the average distance between the detected inspiratory peaks – one-half this value is the estimated average duration for a respiratory phase. Potential onsets are "clustered" together using the estimated average respiratory phase; within each cluster, the sample at which $p[n]$ is a minimum is finally declared to correspond to a respiratory phase onset. When compared with simultaneous airflow measurements, this algorithm yielded an onset-detection accuracy rate of 100%, and an average timing accuracy of 118 ± 39.9 ms.

An approach proposed by [41] uses a parameter called *variance fractal dimension* (VFD) to detect respiratory phase onsets. VFD is a measure of complexity in the data set – in this case, the data are time-domain tracheal breath sound signals (S). VFD is defined as:

$$VDF = D_E - 1 + H,$$

where D_E is called the "embedding dimension" (D_E is equal to 1, 2, and 3 for signals that lie in curves, planes, and spaces, respectively), and

$$H = \lim_{\Delta t \to 0} \frac{\log(\mathrm{var}(\Delta S)_{\Delta t})}{2\log(\Delta t)}.$$

The VFD is computed in windows that span 12.5 ms with 50% overlap. Respiratory phase onsets are detected by applying a moving average filter of duration 0.7 seconds (the approximate duration of one-half a breath phase) to the VFD signal, and identifying the locations of resulting peaks. When compared with simultaneous airflow measurements, this method yielded an average timing accuracy of40 ± 9 ms.

Finally, we discuss the method for respiratory phase onset detection presented in [35]. This approach uses respiratory inductive plethysmography (RIP) data (which is proportional to lung volume) and two independent feedforward artificial neural networks (ANNs). In a pre-processing stage, the original RIP signal is resampled at a sampling frequency of 50 Hz. The resulting signal is analyzed over 2 s windows, with a shift-size of 20 ms (or, equivalently, 1 sample). Within a window, the data are normalized to span an amplitude-range of [0 → 1]; the normalized signal is the 100-dimensional feature vector that is used as input to the ANNs. For the training procedure, the first of the two ANNs (ANN_{insp}) is trained to detect inspiration-onsets that are located in the central 200 ms of the window; the second of the two ANNs ($ANN_{\exp}$) is trained to detect expiration-onsets that are located in the central 200 ms of the window. The parameters of the ANNs are trained via the backpropagation technique. The architecture for the ANNs is such that there is one hidden layer in addition to the input and output layers. For each ANN, optimal values for the number of hidden (sigmoid) units and training epochs are computed using cross-validation testing. The output of each ANN is compared to a threshold – if the output exceeds the threshold, the input example is tagged as '1', and if the output falls below the threshold, the input example is tagged as '0'. The scheme for detecting inspiration and expiration onsets is as follows: (1) tags for each window in the tracheal breath signal are computed using both ANN_{insp} and $ANN_{\exp}$; (2) in a post-processing stage, for each ANN output, spurious '0' tags are eliminated by switching all 'o' sequences that span less than 100 ms, and that lie between two sequences consisting of five or more consecutive '1' tags, to '1'; spurious '1' tags are eliminated by switching all isolated '1' tags and sequences of fewer than 5 consecutive '1' tags to '0'; furthermore, alternation between consecutive '1' sequences in outputs of ANN_{insp} and $ANN_{\exp}$ is imposed by switching consecutive '0' tags in ANN_{insp} that lie between two sequences of consecutive '1' tags in $ANN_{\exp}$ to '1', and by merging together two or more separate consecutive sequences of '1' tags in ANN_{insp} that lie between two consecutive sequences of '1' tags in $ANN_{\exp}$; (3) finally, inspiration and expiration onsets are defined to be the points at which the RIP signal is a minimum within each consecutive sequence of '1' tags in ANN_{insp} and $ANN_{\exp}$, respectively. This method for respiratory phase detection yields a timing accuracy of 34 ± 71 ms for inspiratory onsets and 5 ± 46 ms for expiratory onsets and an average timing accuracy of 19.5 ± 58.5 ms across both respiratory phases.

3.5 Method

We now turn to our methods. In developing and testing our respiration, respiratory rate, and respiratory phase onset detectors, we collected tracheal breath sound recordings from five healthy subjects. We used an electronic stethoscope and sampled the data at 22,050 Hz.

3.6 Approach

In this section we discuss our approach for implementing the processing-blocks shown in Figure 1.

3.6.1 Signal Transformation Stage

A detailed depiction of the *signal transformation* stage is shown in Figure 2.

Goals

The goal of the *signal transformation* stage is to transform the input time-domain signal into a representation that is useful for feature extraction in each of the *respiration detector, respiratory rate detector,* and *respiratory phase onset detector* stages. The signal-processing should accommodate for the non-stationary quality of tracheal breath sounds, since the rate of respiration (or duration of respiratory cycles) can vary with time.

In order to compute a respiratory rate, we need to be able to count the number of respiratory cycles that are present in the tracheal breath sound signal. One respiratory cycle consists of two respiratory phases – inspiration and expiration. Hence, if we could count the number of respiratory phases in the signal, then computing the number of respiratory cycles is a trivial task – simply divide the number of respiratory phases by a factor of 2.

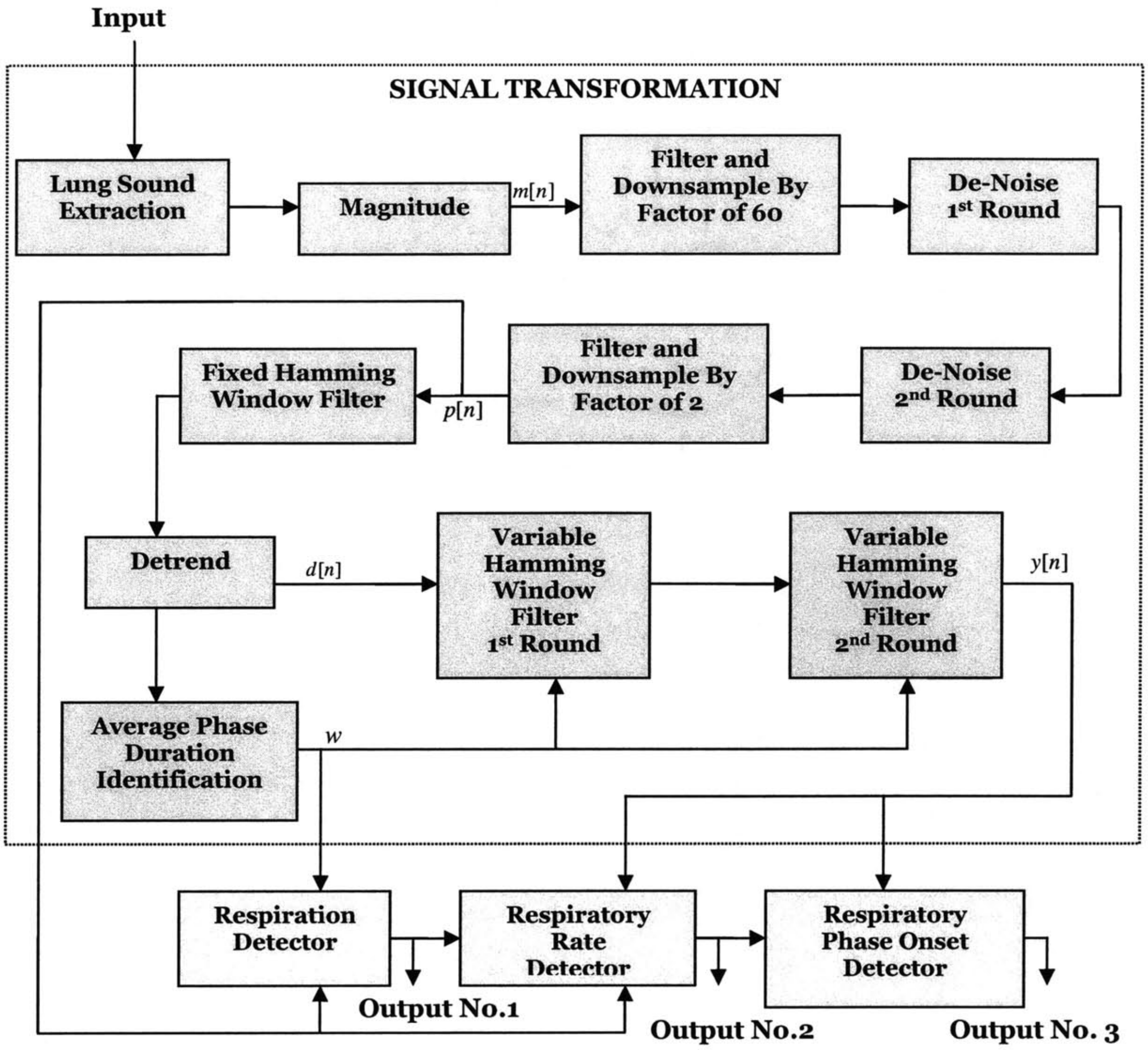
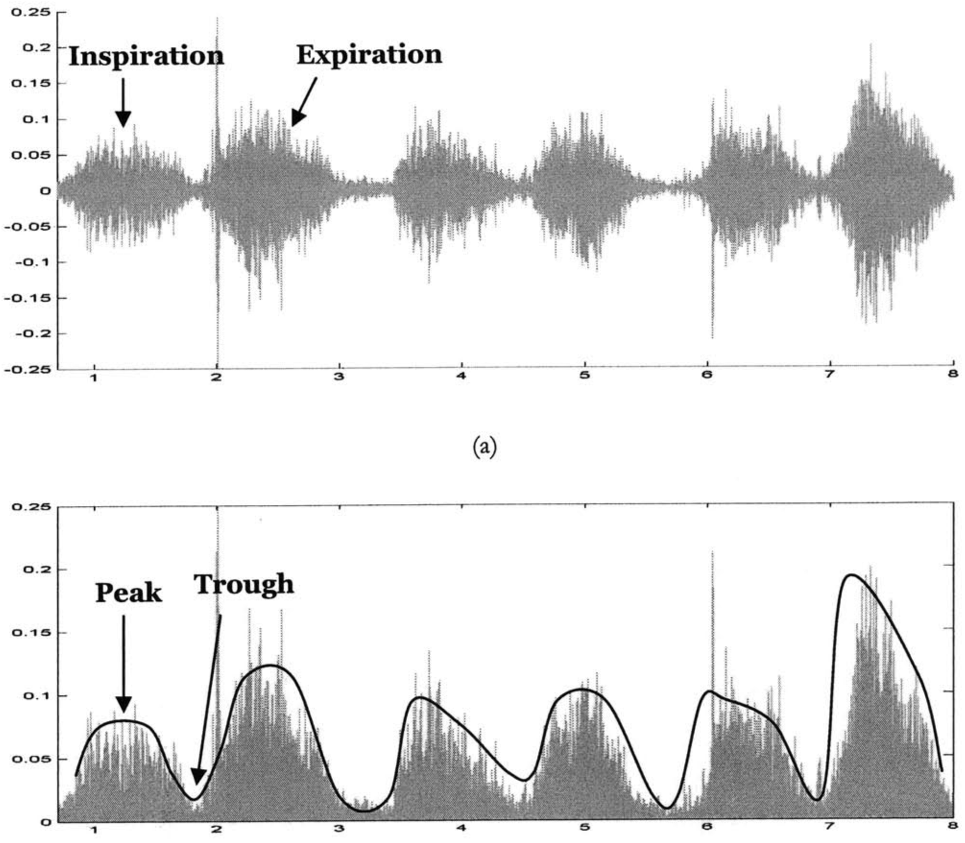

Input
SIGNAL TRANSFORMATION
Lung Sound Extraction
Magnitude
m[n]
Filter and Downsample By Factor of 60
De-Noise 1st Round
De-Noise 2nd Round
Filter and Downsample By Factor of 2
p[n]
Fixed Hamming Window Filter
Detrend
d[n]
Variable Hamming Window Filter 1st Round
Variable Hamming Window Filter 2nd Round
y[n]
Average Phase Duration Identification
w
Respiration Detector
Respiratory Rate Detector
Respiratory Phase Onset Detector
Output No.1
Output No.2
Output No. 3

Figure 2. Block diagram of *signal transformation* stage.

Therefore, the first goal that we tried to achieve in implementing the rate detector was to find a method for counting (and thus detecting) respiratory phases. We approached this problem by noticing that tracheal breath sound signals typically have a distinct shape – the intensity of each phase starts with a gradual increase from a baseline intensity (i.e., the intensity in regions over which no breathing occurs), reaches a peak close to midway, and gradually decreases back to the baseline value. An outline of the magnitude (i.e., the absolute value) of the signal looks like a slowly-varying wave, with definite peaks and troughs corresponding to the phase midpoints and endpoints, respectively. (We will refer to this outline as the **envelope** of the signal.) This point is illustrated in Figure 3. Figure 3a shows an example of the time-amplitude plot of the breath signal. Figure 3b plots the magnitude of the signal, and overlays an approximate trace of its envelope.



(a)

(b)

Figure 3. (a) Time in seconds (x-axis) vs. Amplitude (y-axis) plot of a tracheal breath sound signal; (b) Time vs. Magnitude plot of the same signal in (a), with an approximate curve manually fitted to the its envelope.

Another goal of the *signal transformation* stage was to remove noise from the signal that might adversely affect the performance of the detectors. The primary type of noise that we address is in the form of random spikes in the input waveform. These are typically generated by the movement of the stethoscope's sensor against a subject's skin during the recording process.

Implementation Details

This section describes the signal-processing techniques and parameters used to achieve the two goals for implementation mentioned in the previous section. It describes the successive stages of processing[4] and illustrates their outputs for a sample input signal.

We begin with the input to our system, a tracheal breath sound signal. Figure 4 plots the original waveform. It is not obvious from this waveform how many respiratory cycles are in the signal and where, exactly, they occur. This file actually contains 4 respiratory cycles (and hence, 8 respiratory phases) that continuously span the first ten seconds of the signal.

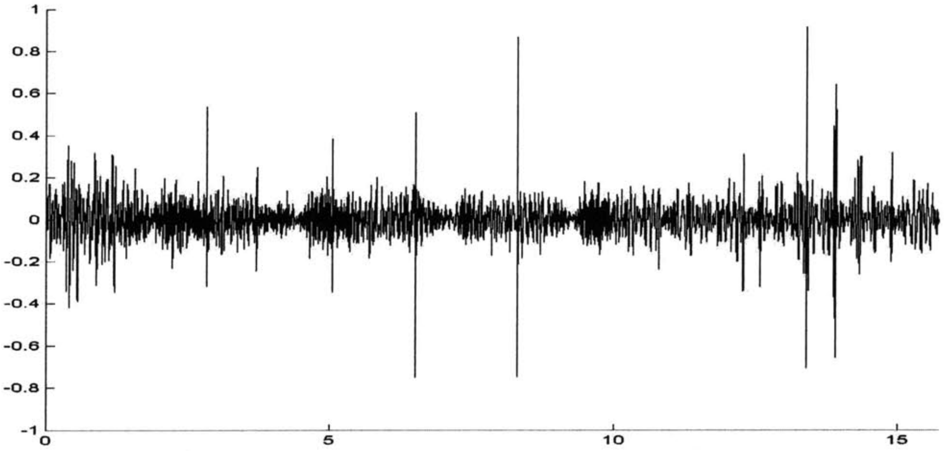


Figure 4. Time in seconds (x-axis) vs. Amplitude (y-axis) plot of the original input waveform.

The *lung sound extraction* stage removes signal components outside the [100 → 2,500] Hz frequency range, since tracheal breath sounds generally do not fall outside this range. To do this, we use a finite impulse response (FIR) bandpass filter with low-frequency and high-frequency cutoffs at 100 Hz and 2,500 Hz, respectively. The filter was designed using the *window method* and a Blackman window of size one-twelfth the sampling frequency of the input signal[5]. This filtering is especially important for removing the typically high-energy heart sounds that are inevitably picked up along with the breath sounds during the recording process. Figure 5 shows the output of this stage. The respiratory phases are readily visible, since the low frequency noise from the heart-sounds has been removed.

The *magnitude* stage takes the absolute value of the signal. Figure 6a shows the output $m[n]$. The *filter and downsample by factor of 60* stage applies a non-causal, 60-point moving average

[4] Refer to the block diagram in Figure 2.

[5] Refer to [27] for a detailed discussion on FIR filters.

filter to $m[n]$ and keeps only every 60th sample. The purpose of downsampling the signal is to reduce computational burden. Recall that our main interest is in finding the overall shape, or envelope, of the signal. Because the envelope is slow-moving relative to the higher-frequency waveform that is densely compacted into the envelope, downsampling does not adversely alter the shape of the envelope – the only restriction is that the new sampling frequency must be at least twice the frequency of the envelope signal of a typical tracheal breath signal. If we assume that the duration of a typical respiratory phase is 1 second, then the envelope signal is expected to be 1 Hz – this means that the new sampling frequency should be at least 2 Hz. In this case, we start with a sampling frequency (fs) of 22,050 Hz, and downsample by a factor (r) of 60, which yields a new sampling frequency of 367 Hz ($= fs / r = 22{,}050 / 60$). Since 367 Hz is far greater than 2 Hz, we expect that the envelope signal will be preserved. Figure 6b plots the output of this stage.

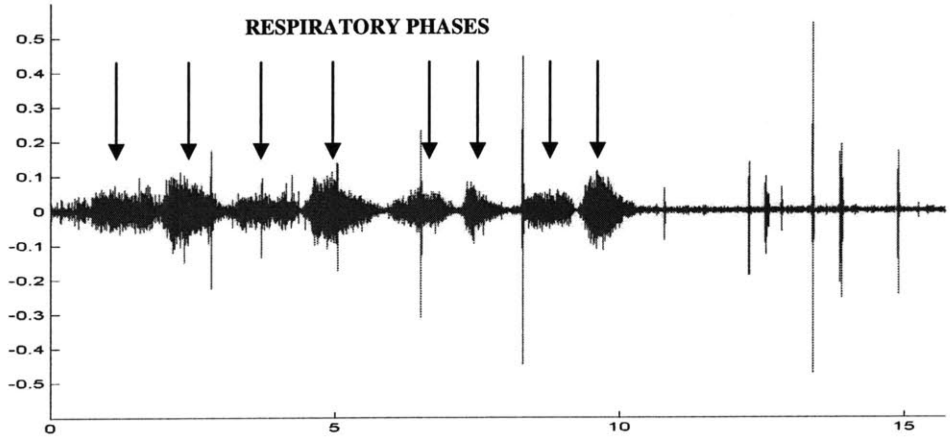


Figure 5. Time in seconds (x-axis) vs. Amplitude (y-axis) plot of the output waveform of the *lung sound extraction* stage. The arrows point out each of the (eight) respiratory phases in the signal.

A comparison of the magnitude signal (Figure 6a) to its downsampled version (Figure 6b) confirms that a negligent amount of relevant information is lost about the signal envelope as a result of the downsampling.

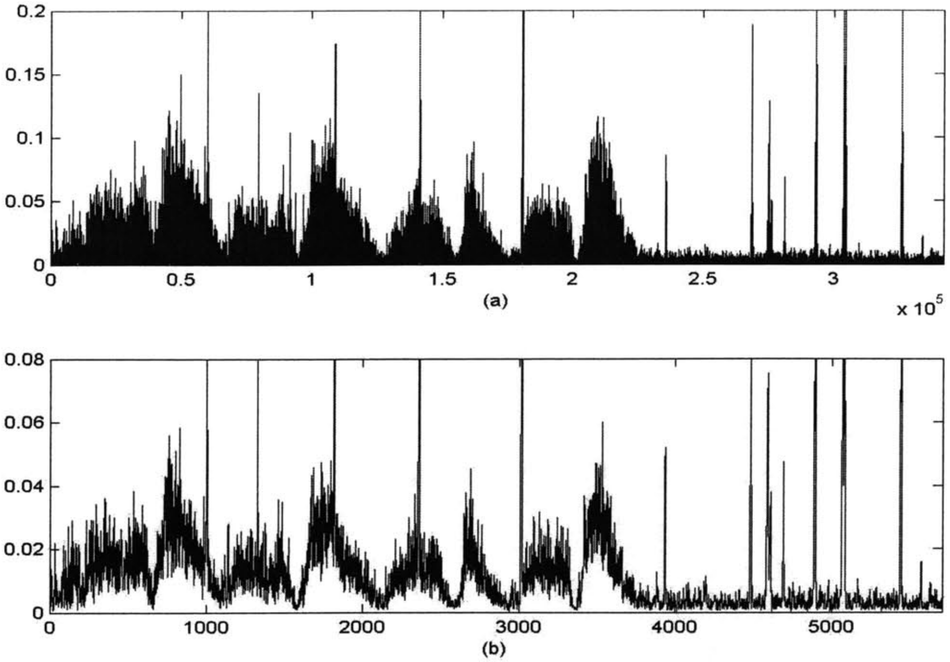


Figure 6. Index in samples (x-axis) vs. Magnitude (y-axis) plots: (a) Output of *magnitude* stage $m[n]$; (b) Output of *filter and downsample by factor of 60* stage.

The next two stages – *de-noise 1st round* and *de-noise 2nd round*– eliminate the random noise spikes (i.e., high-amplitude, nearly instantaneous impulses) from the signal.

We start with the *de-noise 1st round* stage, which uses a median filter to reduce the magnitude of the spikes. The filtering operation first partitions the signal into contiguous 0.5 second windows. For each window, the median value *(m)* is computed – the amplitude at each sample is compared to a threshold of *5m*. If the threshold is exceeded, then that sample is interpreted as noise, and its value is set to *zero*; otherwise, its value is unchanged. The same process is repeated in the *de-noise 2nd round* stage to further reduce the amplitude of the spikes. The final output is shown in Figure 7. Notice the effectiveness of the noise reduction scheme in removing the high-magnitude spikes, while still maintaining the integrity of the signal envelope.

The *filter and downsample by factor of 2* stage applies a non-causal, 2-point moving average filter to the signal, and then takes every other point. The purpose of the downsampling is to further reduce the computational load. The new sampling frequency is 183 Hz (=367 Hz/2). Let the output signal be $p[n]$.

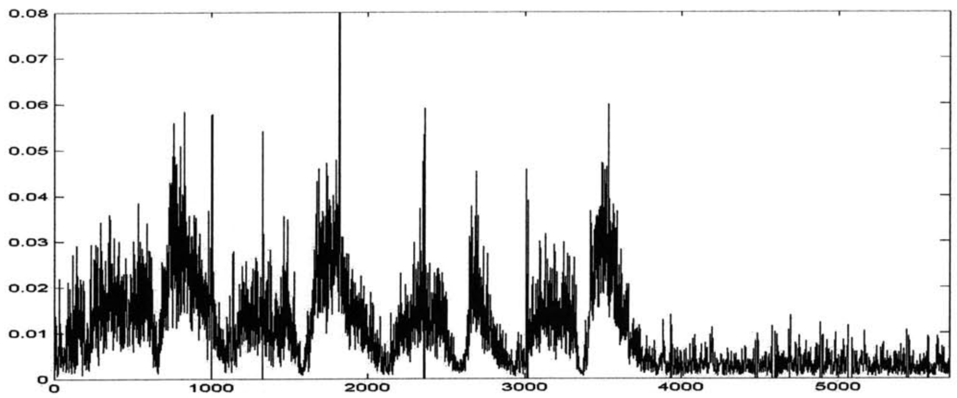


Figure 7. Index in samples (x-axis) vs. Magnitude (y-axis) plot of output of *de-noise 2nd round* stage.

Next, the *fixed Hamming window filter* stage applies a low-pass filter to $p[n]$. The impulse response of the filter is a Hamming window that spans 0.05 seconds. The purpose of this step is to reduce the high-variance, high-frequency noise that rides on top of the signal's envelope, similar to that shown in Figure 7. The result $d[n]$ is graphed in Figure 8a. It is apparent that with each successive filtering step, the signal transforms into a smoother representation of its envelope. Note, however, that $d[n]$ is still not smooth enough to represent each phase midpoint as a single peak and each phase endpoint as a single trough (see Figure 3b for an idea of the type of envelope signal we would ideally want to have). Thus, more low-pass filtering is called for.

Applying a low-pass filter that has an impulse response of arbitrary and fixed length to the signal in Figure 8a could yield undesirable results due to the non-stationary nature of breath sounds. Because breath phase durations can span a wide range of values, a single filter that yields desirable results in one case may not necessarily yield desirable results in another case.

We address this issue by making the length of the filter's impulse response (i.e., filter-length) depend on the characteristics of the input signal. The filter-length should approximate the average duration of a respiratory phase in the input signal. To determine the filter-length, we start with the *detrend* stage, which subtracts the global mean of the input signal from each sample value such that the mean (or DC-value) of the output (shown in Figure 8b) is zero.

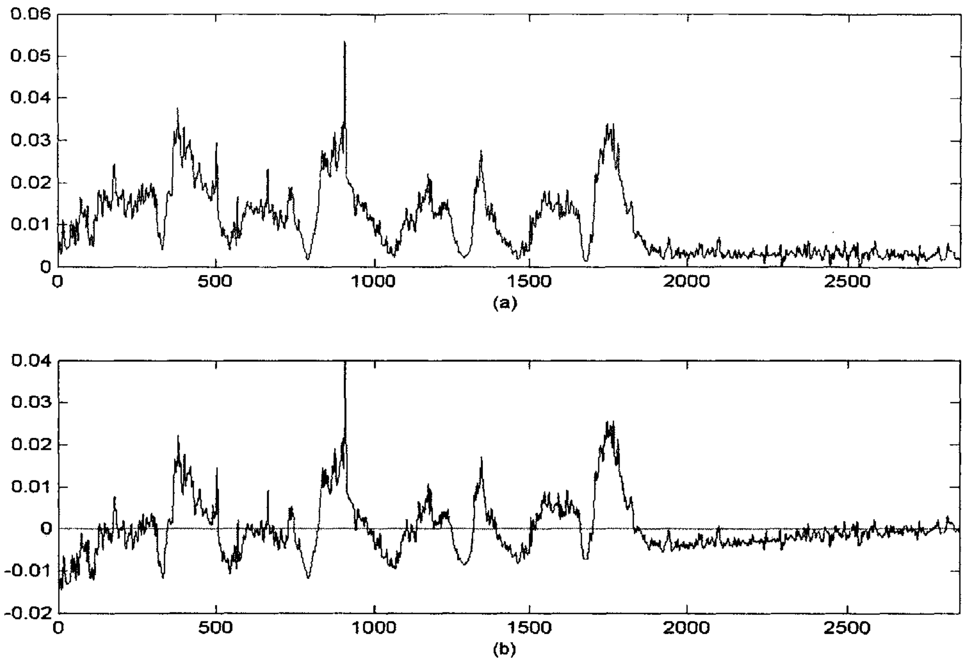


Figure 8. Index in samples (x-axis) vs. Amplitude (y-axis) plots: (a) Output of the *fixed Hamming window filter* stage; (b) Output of the *detrend* stage.

Notice that a rough approximation of the input signal's average phase duration is proportional to the average width of the peaks' lobes. The *average phase duration identification* stage approximates this average width by finding the *widest* peak lobe that intersects the zero-amplitude line in Figure 8b. Suppose that the lobe's cross-section with the zero-amplitude line has a length of L. Then the window size w that we use to approximate the average phase duration is a function of L, with parameters set to empirically optimized values. Specifically, $w = f(L) = ceil(w/2) + 2$.

The low-pass filter's impulse response is a Hamming window of length w. This filter is applied to $d[n]$ at the *variable Hamming window 1st round* stage. Figure 9a plots the output signal, which is significantly smoother than $d[n]$ and takes the shape of the desired envelope of the original signal. The same the filter is applied once more to this signal in the *variable Hamming window 2nd round* stage to further smoothen it. Figure 9b plots the output $y[n]$.

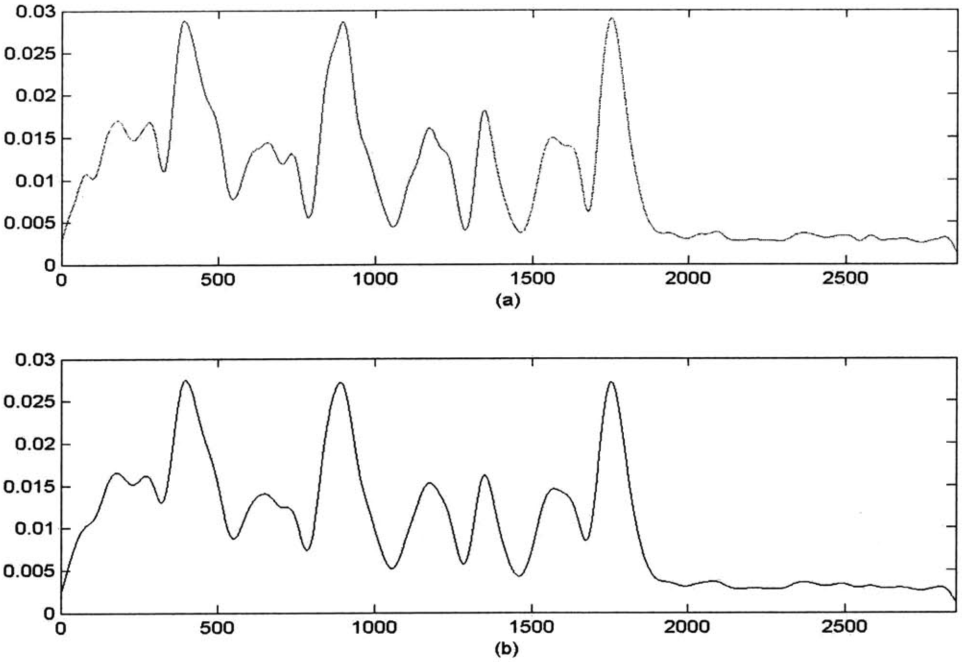


Figure 9. Index in samples (x-axis) vs. Magnitude (y-axis) plots: (a) Output of the *variable Hamming window 1st round* stage; (b) Output of the *variable Hamming window 2nd round* stage.

This completes the *signal transformation* stage of our system. On a final note, to illustrate that $y[n]$ closely resembles the envelope of the magnitude of the original signal ($m[n]$), Figure 10 overlays a scaled version of $y[n]$ on top of $m[n]$.

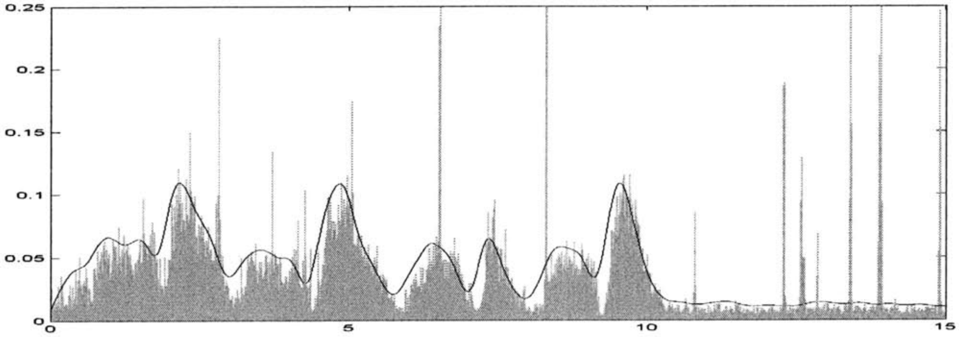


Figure 10. Plot of a scaled version of $y[n]$ (the envelope signal) overlaid onto $m[n]$.

For this particular example, the envelope signal $y[n]$ does not represent each respiratory phase with a single peak. Recall that unique correspondences between the respiratory phases in the input signal and peaks in the envelope signal are desired. We address this issue in the *respiratory rate detector* stage by employing a rule-based method for selecting only relevant peaks.

3.6.2 Respiration Detector Stage

Our respiration detector uses time-domain information to parse the tracheal breath sound signal into regions classified as either "breathing" or "no breathing." The detection scheme is shown in Figure 11.

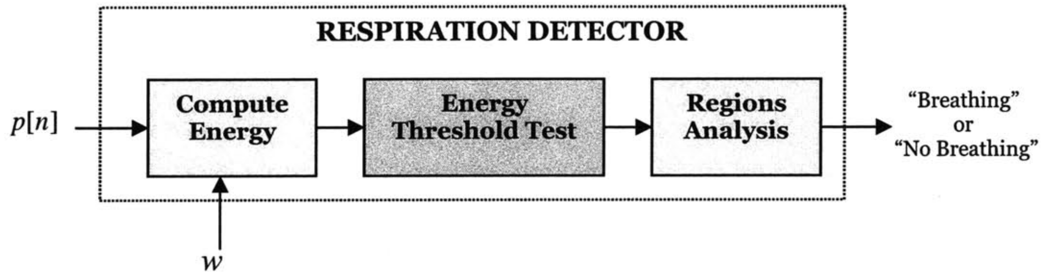


Figure 11. A block diagram of the *respiration detector.*

The respiration detector uses $p[n]$ (the output of the *filter and downsample by factor of 2* stage) and w (the window size that approximates the average respiratory phase duration), both of which are outputs of intermediate processing-blocks in the *signal transformation* stage of Figure 1. First, $p[n]$ is split into contiguous segments of length $v = 3w$. For each segment:

1. The *compute energy* stage computes the total energy E by summing all the values.
2. The *energy threshold test* stage compares E to a threshold. The threshold is equal to the **baseline energy** ($E_{baseline}$) of $p[n]$ over the length of the segment (v). If we define the **baseline amplitude** ($A_{baseline}$) of the signal to be the average amplitude of the signal in areas where no breathing occurs, then $E_{baseline} = vA_{baseline}$. We used $A_{baseline} = 0.0035$, a value that was determined empirically. If E exceeds the threshold, then breathing is assumed to be present in the segment; if E falls short of

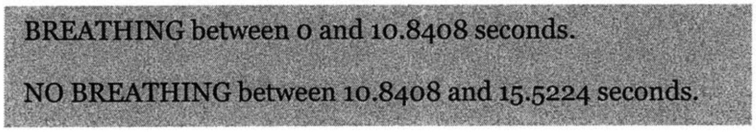
the threshold, then no breathing is assumed to be present in the segment. A value of '1' or '0' is assigned to the segment if breathing is detected or is not detected, respectively.

This procedure produces a string of binary values that correspond to the classifications made for each segment in the *energy threshold test* stage. For the example considered here, the algorithm outputs the following bit-string:

```
1  1  1  1  1  1  1  1  0  0  0  0
```

The *regions analysis* stage groups continuous streams of either bit together. For example, if the string is '111111110000' (as shown above), then the first eight bits of value '1' are grouped together, and the last four bits of value '0' are grouped together.

Finally, each group is given a label of either "breathing" or "no breathing" if it consists of the bit-value '1' or '0,' respectively. The output for the example is:



```
BREATHING between 0 and 10.8408 seconds.
NO BREATHING between 10.8408 and 15.5224 seconds.
```

3.6.3 Respiratory Rate Detector Stage

We mentioned earlier that determining the respiratory rate amounts to counting the number of respiratory phases in the signal. The respiratory rate detector uses $y[n]$ and $p[n]$ (outputs of the *signal-transformation* stage) to count the number of respiratory phases over the regions that are classified as "breathing" by the *respiration detector.* It outputs a respiratory rate in units of cycles per minute for each region. Figure 12 summarizes the respiratory rate detection scheme.

For each "breathing" region, the *peaks & troughs detection* stage identifies potential respiratory phases by first identifying the **peaks** (i.e., local maxima) and **troughs** (i.e., local minima) of $y[n]$. Each peak is coupled with the trough that immediately follows it in time; the peak-trough pairs are then passed through a series of (three) tests. The purpose of these tests is to attempt to eliminate any peak that does not correspond to a respiratory phase.

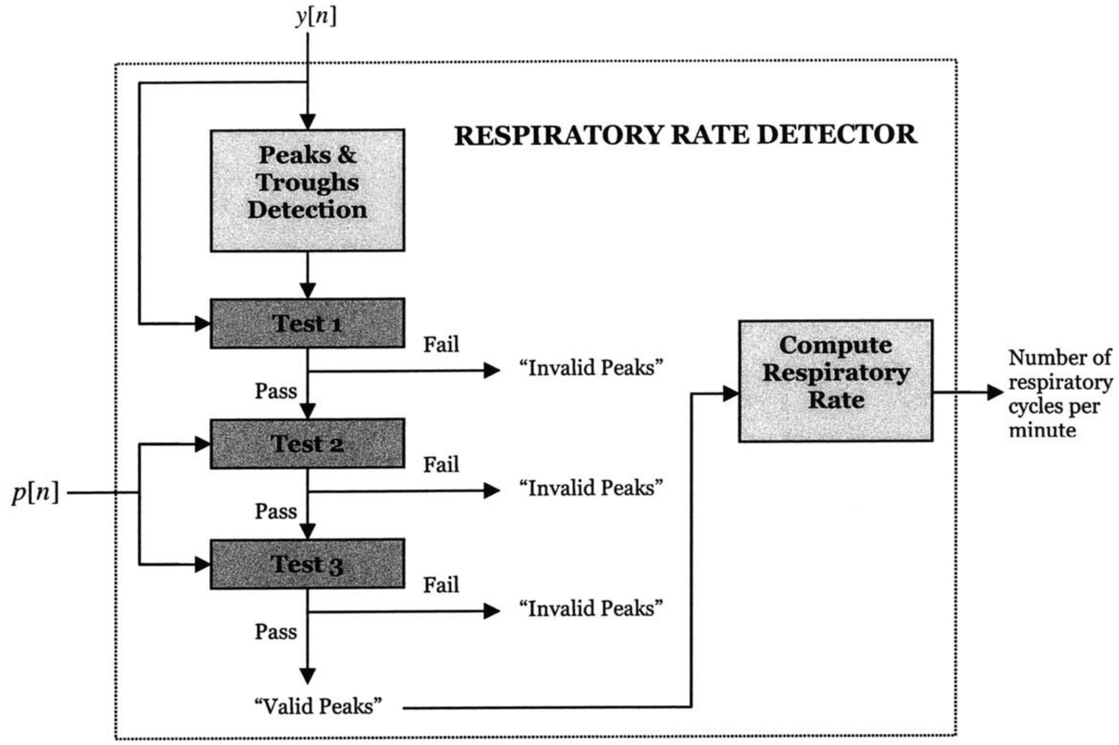


Figure 12. Block diagram of the *respiratory rate detector*.

The algorithm runs the following tests on each peak-trough pair:

- ***Test 1*** – The amplitude in $y[n]$ of the trough must fall below 90% of the amplitude in $y[n]$ of the peak.
- ***Test 2*** – The energy of $p[n]$ in a 0.05 second window (of sample-length w_{test}) centered at the trough location must fall below a threshold level ($E_{threshold}^{trough}$). The threshold is defined as a function of the global mean of $p[n]$ (denoted by $\overline{p[n]}$):

$$E_{threshold}^{trough} = w_{test}\left(\overline{p[n]}\right).$$

- ***Test 3*** – The energy of $p[n]$ in a 0.05 second window (of sample-length w_{test}) centered at the peak location must exceed a threshold level ($E_{threshold}^{peak}$). The threshold is defined as a function of the global mean of $p[n]$ (again, denoted by $\overline{p[n]}$): $E_{threshold}^{peak} = \frac{1}{2} w_{test}\left(\overline{p[n]}\right)$.

We refer to the peaks that pass all three tests as **valid peaks** – i.e., peaks that correspond to the respiratory phases in the signal. Figure 13 (top) graphs the peaks detected in the *peaks & troughs detection* stage; Figure 13 (bottom) graphs the valid peaks. Notice that the tests in this processing-stage were able to eliminate the peaks (in $y[n]$) that do not uniquely correspond to the respiratory phases.

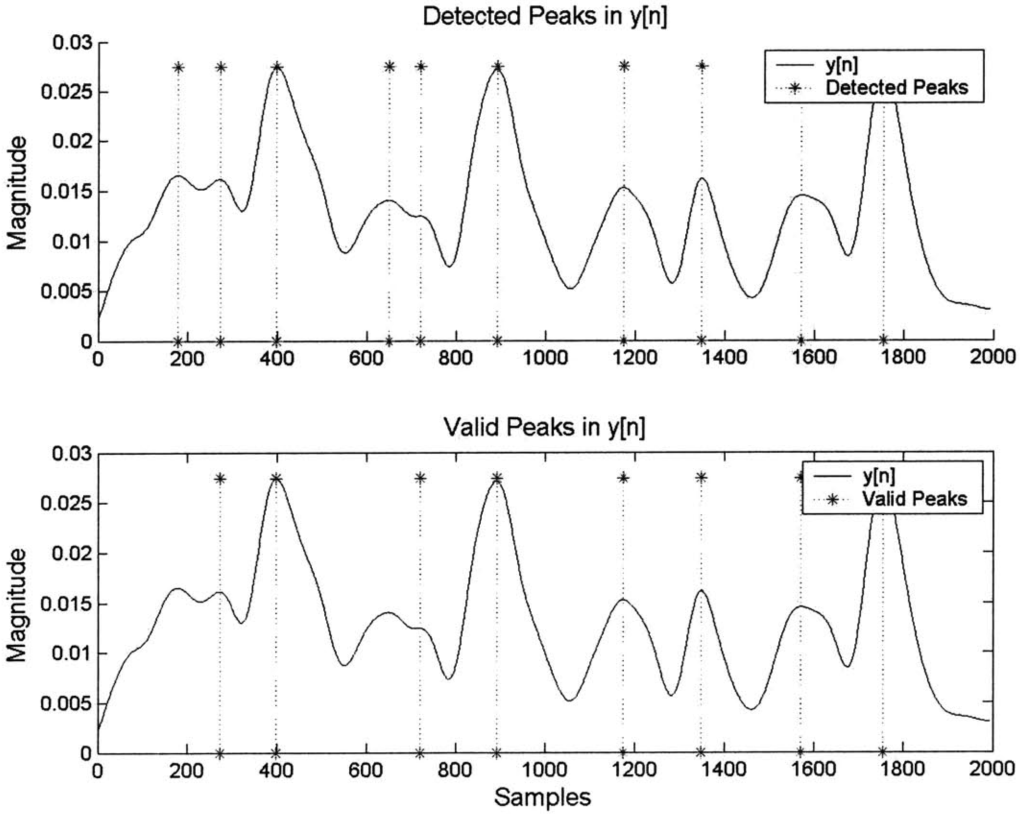


Figure 13. Identifying the respiratory phases: (top) plot of $y[n]$ and the output of the *peaks & troughs detection* stage; (bottom) plot of $y[n]$ and the "valid peaks" output of *Test 3*.

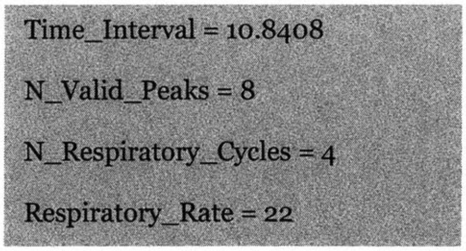
Finally, the respiratory rate ($R_{respiration}$) is function of the number of valid peaks (N_{valid_peaks}) and the time interval of interest (T):

$$R_{respiration} = \frac{N_{respiratory\_cycles}}{T},$$

where

$$N_{respiratory\_cycles} = \frac{N_{valid\_peaks}}{2}.$$

The units for the respiratory rate are in **respiratory cycles per minute**. The algorithm outputs the following parameter-values for the [0 → 10.8408] second region over which "breathing" was detected in the sample signal:



```
Time_Interval = 10.8408

N_Valid_Peaks = 8

N_Respiratory_Cycles = 4

Respiratory_Rate = 22
```

3.6.4 Respiratory Phase Onset Detector Stage

The respiratory phase onset detector uses $y[n]$ (an output of the *signal transformation* stage) and the locations of the "valid peaks" (an output of the *respiratory rate detector* stage) to identify the locations in time at which respiratory phase onsets occur for regions of the signal labeled as "breathing" (by the *respiration detector* stage). Figure 14 summarizes the respiratory phase onset detection scheme in a block diagram.

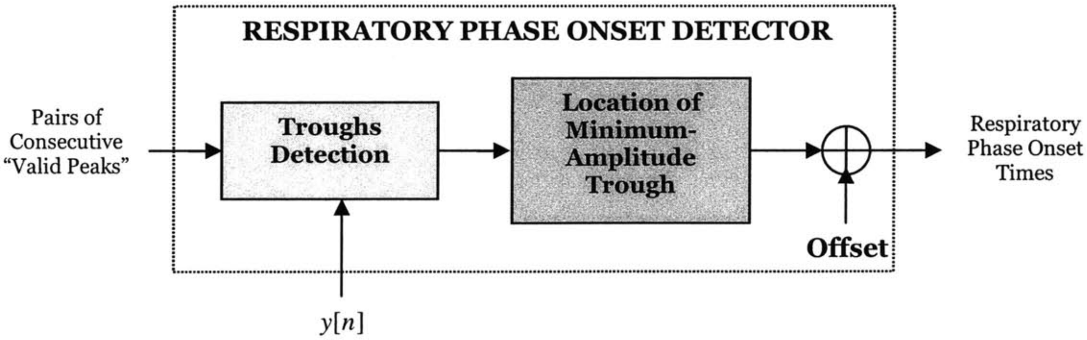

RESPIRATORY PHASE ONSET DETECTOR
Pairs of Consecutive "Valid Peaks"
Troughs Detection
Location of Minimum-Amplitude Trough
Offset
Respiratory Phase Onset Times
y[n]

Figure 14. Block diagram of *respiratory phase onset detector.*

For each "breathing" region, each pair of consecutive valid peaks is analyzed as follows: (1) the *troughs detection* stage identifies the locations in $y[n]$ of all troughs that lie between the two valid peaks in time; (2) the *location of minimum-amplitude trough* identifies the location in time of the trough with the smallest amplitude in $y[n]$; (3) the sum of the minimum-amplitude trough's location and an empirically determined offset of 100 ms yields the respiratory phase onset time. Figure 15 plots the sample tracheal breath signal, (solid) demarcation lines to separate the "breathing" and "no breathing" regions, and (dotted) demarcation lines to indicate locations of the detected respiratory phase onsets.

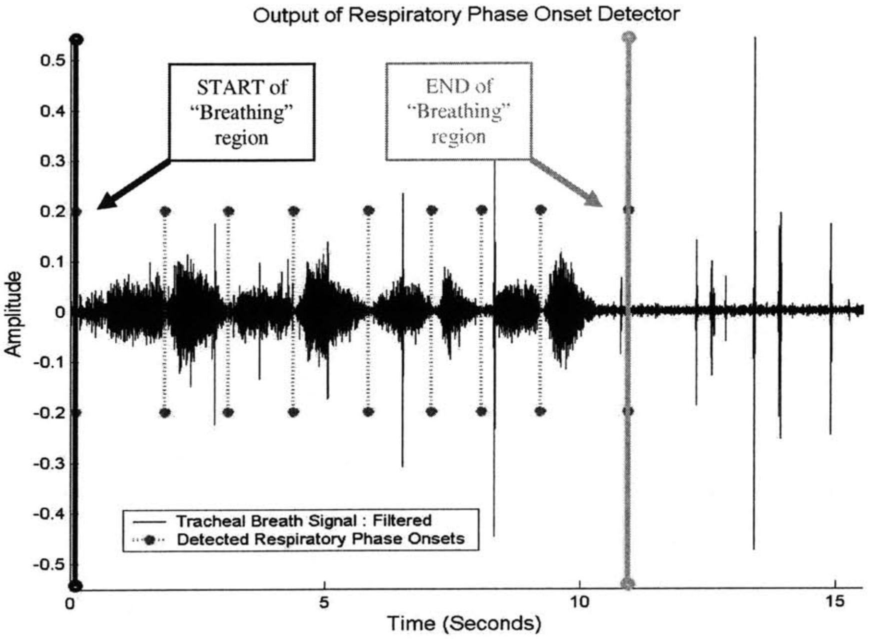


Figure 15. Plot of tracheal breath signal (solid black line), demarcation lines for detected respiratory phase onsets (dotted magenta lines), and demarcation lines indicating the start and end of the "breathing" region (solid blue and cyan lines, respectively).

3.7 Performance

In this section, we present the results of performance tests on the respiratory rate detector. We do not, however, report figures for the performance of the respiration and respiratory phase onset detectors for the following reasons: (1) we focused primarily on achieving high accuracy-rates for the respiratory rate detector; (2) a gold standard (e.g., simultaneous airflow measurements or the markings of the data by a physician) to which the outputs of the respiratory phase onset detector needs to be compared in order to compute the timing accuracy was unavailable.

Respiratory Rate Detector

To test the performance of the respiratory rate detector, we used 60 examples of respiratory phases for each of the five subjects from whom tracheal breath data were collected – this yielded a total of 300 (=5*60) respiratory phase examples for our analysis. We use the following *performance metric* for the detection scheme: percent of respiratory phases correctly represented by a single, unique peak in the set of "valid peaks" outputted by the detector. We will refer to this performance metric as the **accuracy rate** ($R^{accuracy}$) of the respiratory phase detector. Figure 16 plots the accuracy rate computed for each subject.

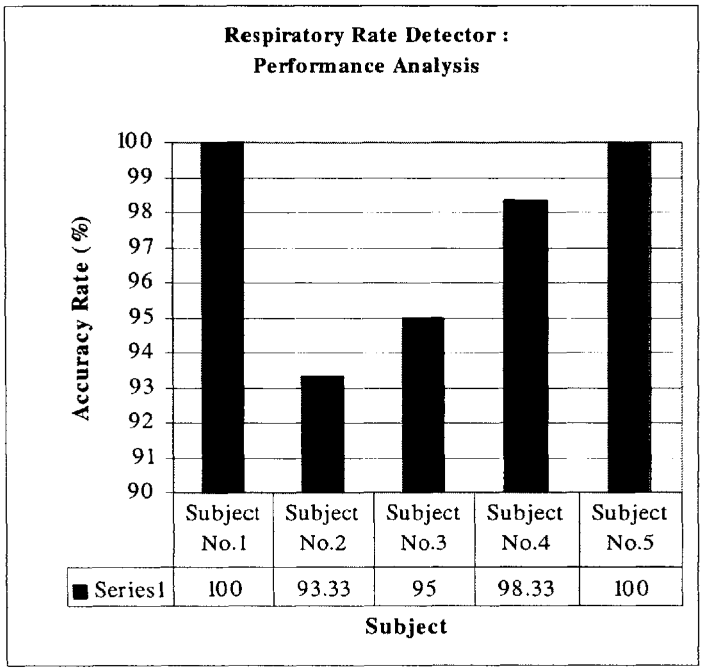


Figure 16. Bar graph of accuracy rates (for each of five subjects) computed to test the performance of the *respiratory rate detector.*

The average ($R^{accuracy}_{average}$) and standard deviation ($R^{accuracy}_{\sigma}$) for the accuracy rate across all of the subjects is $97 \pm 3\,\%$ ($R^{accuracy}_{average} \pm R^{accuracy}_{\sigma}\,\%$).

3.8 Discussion

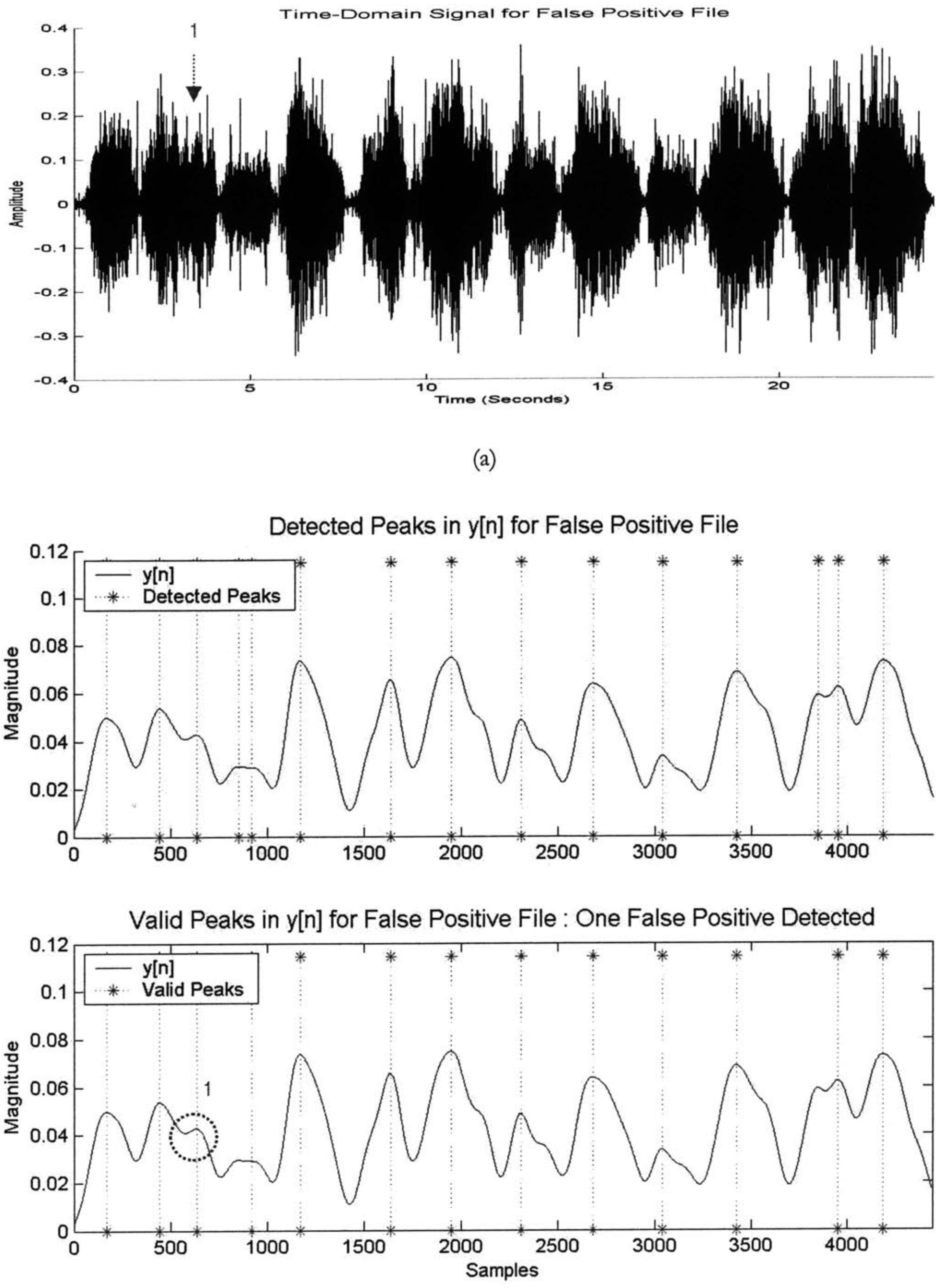
We discuss here the results of the performance analysis for our respiratory rate detector (presented in the previous section). We also compare the salient attributes of our approach for detecting respiration, respiratory rate, and respiratory phase onsets with those of other approaches described in the "Background Literature/Related Work" section of this chapter.

Respiratory Rate Detector

Our respiratory rate detector did not exhibit perfect accuracy, as the average accuracy rate across all subjects was 97%. Errors arose when, for a particular respiratory phase: (1) two "valid peaks" were detected instead of one, due to variations in the breath signal intensity throughout the duration of that respiratory phase (yielding a false positive); or (2) no "valid peak" was detected because the energy in that respiratory phase's peak fell short of the threshold in *Test 3* of the *respiratory rate detector* stage (yielding a false negative).

Figure 17 shows an example of a tracheal breath sound file in which a false positive was detected (in the 2nd respiratory phase in time). Figure 17a plots the filtered time-domain signal from the output of the *lung sound extraction* stage; Figure 17b (top) plots $y[n]$ and the locations of the peaks detected from the *peaks & troughs detection* stage; and Figure 17b (bottom) plots $y[n]$ and the locations of the "valid peaks." The falsely detected peak is indicated by a dotted arrow in Figure 17a and a dotted circle in Figure 17b (bottom).

Our respiratory rate detector does not distinguish between inspiration and expiration when detecting respiratory phase peaks. It cannot distinguish, for example, between a pair of inspiratory-expiratory peaks and two consecutive peaks of inspiration (or expiration) that are caused by inspiring (or expiring) shallowly, holding one's breath, and then inspiring (or expiring) further. While the two peaks from the latter scenario correspond to a single respiratory phase, our detector would incorrectly treat them as corresponding to two separate respiratory phases.



(a)

(b)

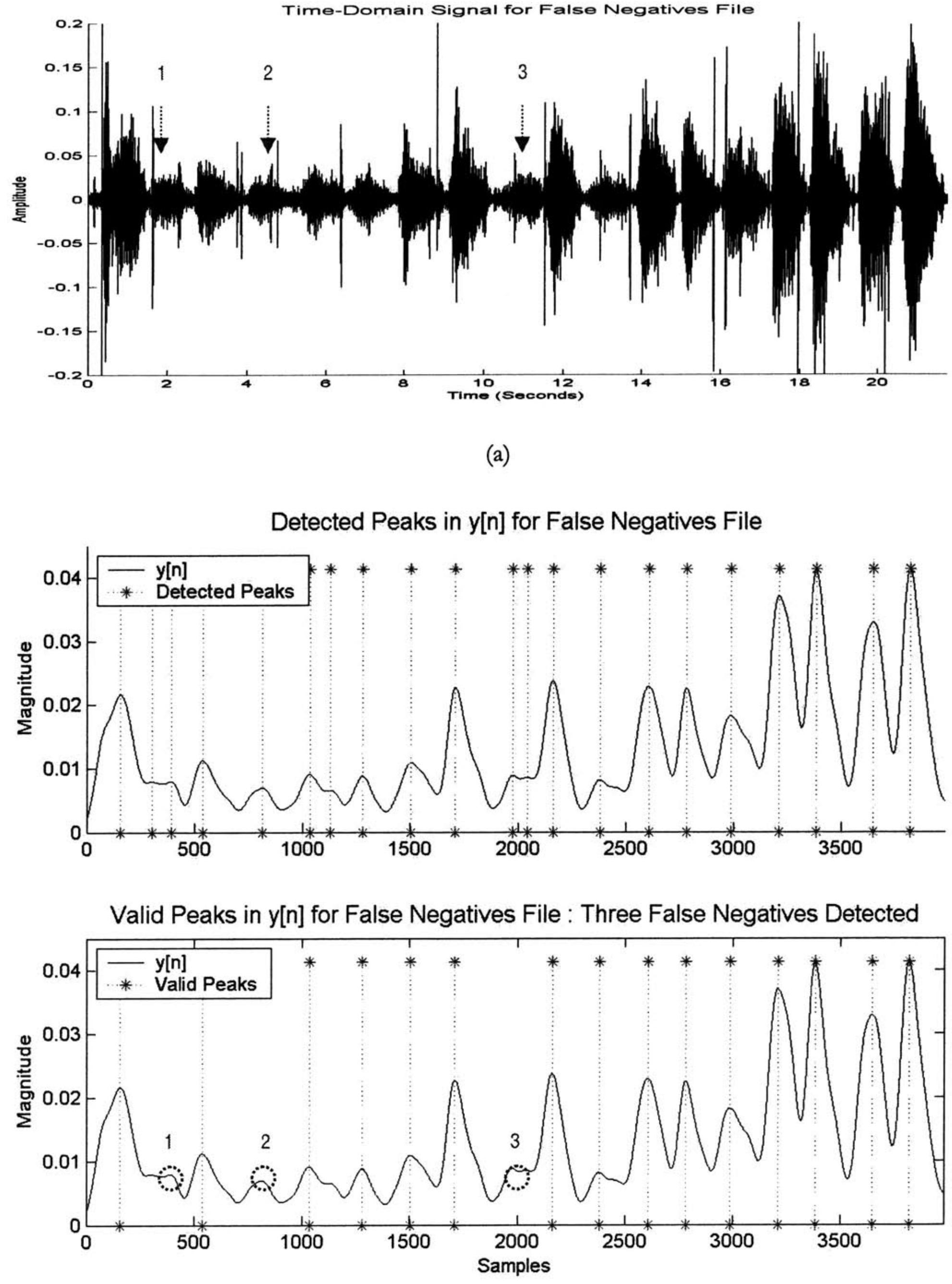
Figure 17. False positive example: (a) filtered, time-domain tracheal breath signal; (b) (top) plot of $y[n]$ and the output of the *peaks & troughs detection* stage, and (bottom) plot of $y[n]$ and the "valid peaks" output of *Test 3* in the *respiratory rate detector* stage. One false positive was detected – it is indicated by a dotted arrow in (a), and a dotted circle in (b) (bottom).

Figure 18 shows an example of a tracheal breath sound file with three false negatives (i.e., three respiratory phases that were not detected). Figure 18a is the filtered, time-domain signal from the output of the *lung sound extraction* stage; Figure 18b (top) plots $y[n]$ and the locations of the peaks detected from the *peaks & troughs detection* stage; and Figure 18b (bottom) plots $y[n]$ and the locations of the "valid peaks." The false negatives are indicated by dotted arrows in Figure 18a and dotted circles in Figure 18b (bottom).

All three false negatives had at least one peak in $y[n]$ at the output of the *peaks & troughs detection* stage (as shown in Figure 18a). However, those peaks all failed *Test 3* in the *respiratory rate detector* stage. Recall that *Test 3* compares the energy of a potential peak in the 0.05 second window, centered at the peak's location, with a threshold. Notice from Figure 18 that the three respiratory phases corresponding to false negatives have relatively low energy compared to the other respiratory phases in the signal (esp. compared to those respiratory phases occurring after 14 seconds). The intensities of the respiratory phases increase considerably from the start to the end of the signal. Because the threshold in *Test 3* is a function of the global mean, the presence of higher intensity respiratory phases later in the signal causes the threshold to be too high to accurately detect the lower intensity respiratory phases in the earlier part of the signal.

The false negatives in Figure 18 might have been avoided if the respiratory rate was detected in smaller time-frames, e.g., if the detector computed the respiratory rate separately for the first and latter halves of the signal. Our current implementation computes one respiratory rate for each detected region of breathing in a particular file. A better method might be to, for each detected region of breathing, compute respiratory rates over fixed-length sliding windows (which span the duration of the signal), where the length of the window is determined empirically to minimize the probability of having a false negative.

We now compare the "accuracy rate" of our respiratory rate detector (or, equivalently, respiratory phase detector) with that of the only other approach (by [6]) for which the same performance metric was used. [6] reports an accuracy rate of 100% for their respiratory phase detection scheme, when tested on 17 sets of tracheal and chest sound signals collected from 11 healthy subjects. A comparison between their method for respiratory phase detection and ours, strictly based on accuracy rate, favors their method, since our average accuracy rate (at 97%) is slightly lower.



(a)

(b)

Figure 18. False negatives example: (a) filtered, time-domain tracheal breath signal; (b) (top) plot of $y[n]$ and the output of the *peaks & troughs detection* stage, and (bottom) plot of $y[n]$ and the "valid peaks" output of *Test 3* in the *respiratory rate detector* stage. The three false negatives are indicated by dotted arrows in (a), and dotted circles in (b) (bottom).

However, recall that [6] uses two channels of acoustic data to detect respiratory phases – one from the trachea, and the other from the chest-wall. Our method uses only one channel of acoustic data (from the trachea). In situations that favor fewer devices (either for convenience or lower equipment costs) or smaller data storage requirements, our method is preferable.

There is a caveat associated with comparing the different approaches for respiratory phase detection. A larger *signal-to-noise ratio (SNR)* in the test data set is expected to yield better performance results (i.e., accuracy rates) for any particular detector. Hence, a truly fair comparison between different approaches for respiratory phase detection cannot be made unless the test data sets used are either identical or, at the very least, characterized by similar SNRs. Because the data set used to test our detector was different from that used to test the detector proposed by [6], we cannot guarantee that one approach is more accurate than the other.

Respiratory Phase Onset Detector

An objective comparison between our respiratory phase onset detector and others' respiratory phase onset detectors is not possible, since the precise timing accuracy of our detector is unknown. However, the respiratory phase onset detectors proposed by [17], [6], [41], and [35], can be compared by ranking their expected performances according to their reported timing accuracies. The timing accuracies are ranked, from smallest to largest, as follows: [35]<[17]<[41]<[6]. Under the assumption that a smaller timing accuracy is better, the approach proposed by [35] is expected to achieve the best performance. Note, however, that [35]'s method is the only one that uses non-acoustical data; furthermore, theirs is the only method that uses machine learning (more specificially, artificial neural networks) to solve the detection problem rather than a rule-based approach. [17] and [6] both use frequency-domain information, whereas [41] uses time-domain information. [17] and [41] use only tracheal signals, whereas [6] uses both tracheal and chest signals.

Our approach for respiratory phase onset detection uses tracheal breath signals and time-domain information. The detection scheme looks for the location in time of the minimum amplitude (in the $y[n]$ signal) between the locations of two consecutive "valid peaks." In retrospect, a better approach for detecting onset-times might have been to use the unfiltered magnitude of the tracheal signal (i.e., the output $m[n]$ of the *magnitude* stage in Figure 2) instead. This is because, if a respiratory phase onset is assumed to be the point at which the energy of the tracheal signal is a minimum between two respiratory phases, filtering may cause the location of that minimum to shift and consequently yield poorer timing accuracy.

Respiration Detector

Sensitivity and specificity rates for [17]'s and our approaches for respiration detection are unavailable, and therefore cannot be compared to the rates reported for [38]'s approach. [38] uses artificial neural networks for the decision scheme, while [17] and we use a rule-based method. [38] uses non-acoustical data; [38] and we use tracheal breath sounds. The primary difference between [17]'s approach and our approach is that they use frequency-domain information for features, whereas we use time-domain information.

The main problem in [17]'s and our approaches for respiration detection is that it only works when the "baseline amplitude" (i.e., the amplitude in the tracheal signal where no breathing occurs) stays constant. In both detection schemes, the energy of the signal over a window in time is compared to a fixed threshold, which is a function of the current baseline amplitude. However, if the baseline amplitude drifts (or varies) with time, then the threshold must be adjusted accordingly in order for the detector to perform consistently.

3.9 Future Work

An important next step for future work in respiration, respiratory rate, and respiratory phase onset detection is to test the effect of variations in airflow rate on the signal-to-noise ratio in tracheal breath sounds, and furthermore, the effect of variations in signal-to-noise ratio on detection performance. While our respiratory rate detector seems robust (based on initial performance analyses), the respiration and respiratory phase onset detectors could be improved further using suggestions proposed in the previous section.

An ultimate goal for this area of research should be to integrate the detectors into a highly-accurate, convenient hardware and software interface that can be used as a tool for continuously monitoring respiration in the wide variety of applications in home and clinical environments.

Chapter 4

Respiratory Phase Classification

This chapter will discuss a method that we have developed for classifying respiratory phases (inspiration and expiration) using a machine learning classifier called a *support vector machine (SVM).*

4.1 Introduction

Respiratory phase detection is important in studying swallowing disorders. Airflow measurements from pneumotachographs are typically used to detect inspiration and expiration. However, there are cases when gathering data using pneumotachographs may be difficult or infeasible – an example is when dealing with children with severe neurological impairments [6]. In such situations an alternative method for detecting respiratory phases is needed. An acoustical approach to respiratory phase detection is an attractive alternative because it is minimally-invasive, relatively inexpensive, and convenient to use.

For an acoustical approach to be viable for detecting respiratory phases, features must exist that are different in the inspiratory and expiratory phase signals. Studies have shown that inspiration and expiration have statistically significant differences in acoustic data (e.g., tracheal and chest sounds) and non-acoustic respiratory data (e.g., airflow and respiratory volume).

Suppose we are given that:

(A) The mean amplitude and mean frequency of tracheal breath sounds (which we denote by $\overline{a}_{trachea}$ and $\overline{f}_{trachea}$, respectively) are functions of respiratory airflow rate [5];

(B) Inspiration and expiration have distinctive airflow vs. lung-volume curves [31] (see Figure 1); and

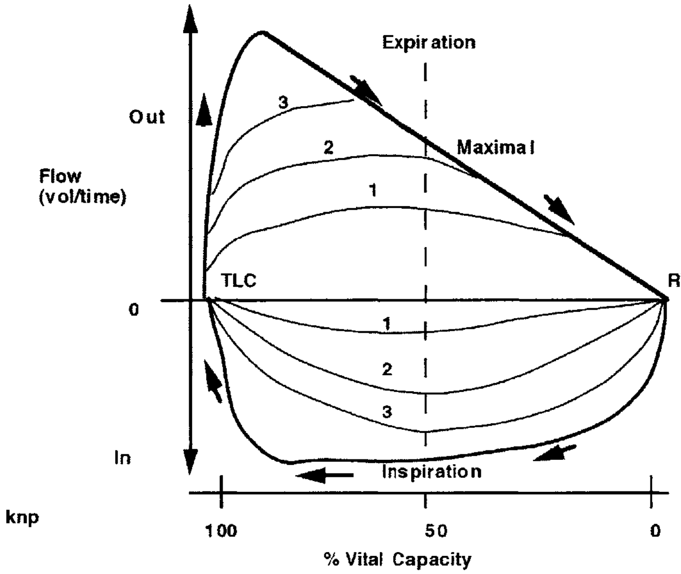
(C) The lung-volume vs. time curve for each respiratory phase is nearly linear except near the start and end of the phase [31] (see Figure 2).

From (B) and (C) we deduce that:

(D) Inspiration and expiration have distinctive airflow vs. time curves.

From (A) and (D) we deduce that inspiration and expiration have distinctive $\overline{a}_{trachea}$ vs. time and $\overline{f}_{trachea}$ vs. time curves. This reasoning suggests that inspiration and expiration can be distinguished using tracheal breath sounds.

Flow Patterns During Expirations and Inspirations of the Same Volume but with Different Rates of Gas Movement



The outer curve (darkest line) represents the most forceful expiration and inspiration. The other graphs represent the flow volume relationships with progressively less forceful expiration or inspiration (3-1).
Note the difference in the shape and values of the flow volume curves for inspiration and expiration -- they are not mirror images of each other.

Figure 1. Airflow vs. lung-volume (as percent of vital capacity) curve (from [31]).

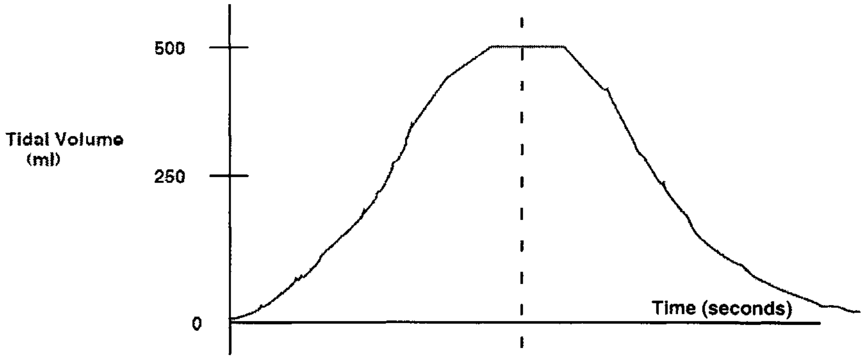


Figure 2. Lung-volume vs. time curve (from [31]).

4.2 Background Literature/Related Work

In a study carried out by [17], simultaneous recordings of *tracheal breath sounds* and airflow were collected from ten healthy subjects at rest – the airflow signal was used as a reference for locating the inspiratory and expiratory phases in time. The mean and standard deviation of several time-domain and frequency-domain parameters were computed for inspiratory and expiratory tracheal breath sounds during the early, middle, and late thirds of the phases. A few of the parameters are described below.

- In the time domain:
  - Ratio between the root mean square (S_{rms}) of the breath signal and no-breath (or apnea) signal → $S_{rms}(breathing)/S_{rms}(apnea)$, where the root mean square is defined as $S_{rms}=\sqrt{\frac{1}{n}\sum_{t}x(t)^2}$ for the time signal $x(t)$ of length n.
  - Ratio between the maximum amplitude ($S_{\max}$) of the breath signal and apnea signal → $S_{\max}(breathing)/S_{\max}(apnea)$, where the maximum amplitude is defined as $S_{\max}=\max(x(t))$.

- In the frequency domain:
  - Median frequency (Hz) → $f_{50}=\min_{f_i}(|\sum_{k=1}^{i}f_k-\sum_{k=i+1}^{n}f_k|)$, where f_i is the ith coefficient of a length n fast Fourier Transform (FFT).

Figure 3 graphs the mean value for each parameter as a function of timing in the respiratory phase (where "start" corresponds to the early third, "mid" corresponds to the middle third, and "stop" corresponds to the late third).

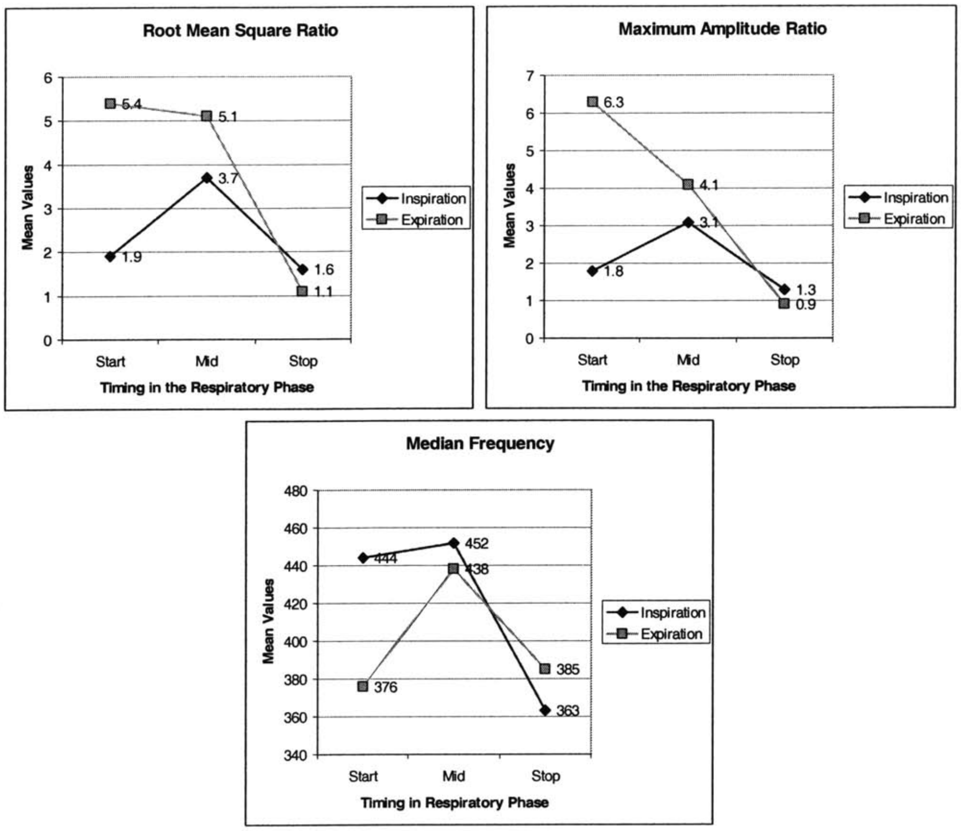


Figure 3. Graphs of mean values for the root mean square ratio, maximum amplitude ratio, and median frequency parameters computed for tracheal breath sounds in [17].

It is evident from these results that the trends in the parameters, as a function of timing in the respiratory phase, are quite distinct for inspiration and expiration. This suggests that it may be possible to develop a respiratory phase classifier that uses only tracheal breath signals and has a high accuracy rate.

[6] proposes a method for respiratory phase classification that uses acoustic recordings from the *chest wall.* They studied seventeen sets of tracheal and chest sounds that were recorded from eleven healthy subjects. This detection approach is based on the fact that respiratory signal intensity at the chest wall is greater in inspiration than expiration. They use the short-time Fourier transform to compute the average power in the chest signal in the [150 → 300]

Hz bandwidth as a function of time. The detect peaks in this average-power signal. Peaks that exceed a threshold value are classified as inspiratory peaks, and peaks that fall below the threshold are classified as expiratory peaks. Additionally, they use tracheal sounds to detect the respiratory phase onset-times (refer to the "Background Literature/Related Work" section of the "Respiration, Respiratory Rate, and Respiratory Phase Onset Detection" Chapter for implementation details). Their detection scheme yielded an average classification accuracy rate of 93% with a standard deviation (across the seventeen sets of tracheal sounds) of 7%.

Another approach by [35] uses non-acoustical *respiratory inductive plethysmography (RIP)* data (which is proportional to respiratory volume) and two artificial neural networks (ANNs) to classify respiratory phases. They detect the **onsets** of inspiration and expiration using the ANNs denoted by ANN_{insp} and $ANN_{\exp}$, respectively. The details of their approach are described in the "Background Literature/Related Work" section of the "Respiration, Respiratory Rate, and Respiratory Phase Onset Detection" Chapter. They collected (from six subjects) a total of 4,000 examples for the training data set, and 2,080 examples for the validation data set; and they collected (from three patients) approximately 29,820 examples for each of inspiration and expiration for the test data set[6]. They report sensitivity and specificity rates of 98% and 95%, respectively, for inspiration, and 98.7% and 95.6%, respectively, for expiration.

4.3 System Design

The approach that we use for classifying respiratory phases has the following properties:

- Uses tracheal breath sounds,
- Incorporates time-frequency information about the signal into a feature vector, and
- Employs a support vector machine (SVM) classifier.

Figure 4 illustrates the system architecture of our classifier.

[6] We are referring to the test data set called 'TS1' in this article.

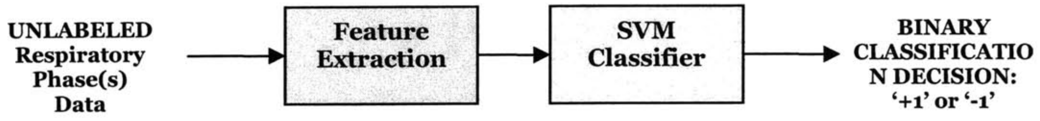

UNLABELED Respiratory Phase(s) Data
Feature Extraction
SVM Classifier
BINARY CLASSIFICATIO N DECISION: '+1' or '-1'

Figure 4. Block diagram of the *respiratory phase classification* system.

The input to the system is the tracheal breath sound signal corresponding to unlabeled respiratory phase(s). Respiratory phases can automatically be extracted from the tracheal signal using the "Respiratory Phase Onset Detector" function in our toolkit. The *feature extraction* stage converts the time signal into either a 280-length or 560-length feature vector representation. The feature vector is then used by the *SVM classifier* stage to assign a class label of either '+1' or '-1' to the unknown respiratory phase example.

Before discussing the details of our classification scheme, we present a technical overview about key concepts in machine learning and support vector machines.

4.4 Machine Learning Concepts

Machine learning refers to the ability of an autonomous system (or "learning machine") to classify examples by learning from past examples and their associated classifications.

Learning machines can be a powerful tool in a wide variety of applications that could benefit from automating decision-making processes or having a system that is capable of making more accurate decisions than can be made from human reasoning alone.

A variety of machine learning techniques are available to classify outputs that are either continuous or discrete in nature. An example of a technique that predicts continuous values is *linear regression*, which maps input data to a number on the real line; an example of a technique that predicts discrete values is the *support vector machine*, which maps input data to one of only two labels $\{\pm 1\}$, where each label is associated with its own **class**. Our discussion of machine learning concepts will focus primarily on machines that predict discrete classes for a given input example – these machines are called **classifiers**.

A classifier is able to "learn" about the existing correlations between specific input examples and their associated class labels by analyzing a **training data set,** which is an available set of examples for which class labels are explicitly known[7]. The precise technique that is used by a classifier for learning new information depends on its **model** type. The classification scheme that results from the learning process depends on the numerical values of the **model parameters** associated with that model. "Training" a classifier essentially refers to the process of determining the numerical values for the model parameters, given the model type.

An essential property of a good classifier is that it *generalizes* well, i.e., that it achieves a high classification accuracy rate when presented with a set of new examples, called a **test data set.** A high classification accuracy rate on the training data set (which corresponds to low a *training error*) does not necessarily guarantee good generalization. For example, it is possible for a classifier to achieve perfect accuracy on the training data set, but yield poor accuracy rates for the test data set – this phenomenon is referred to as *overfitting.*

Key factors that affect the performance of a classifier include: (1) feature selection, and (2) model selection. Feature selection is the process of identifying a set of parameters (called **features**) from the given data that is most relevant for distinguishing between prediction classes. The collection of features used to represent the data (called the **feature vector**) maps the raw data into a new space (or **feature space**). A well-selected feature vector can separate examples into distinct regions (called **clusters**) within the feature space. Larger separations between clusters are expected to yield better generalization. Model selection is the process of choosing a classification model that is expected to yield the best generalization given the assumptions made about the classification problem.

A technique that can be used to estimate the *generalization error* (defined to be the expected rate at which test examples are misclassified) for a particular classifier, and depends only on training examples, is called **cross-validation.** *N-fold cross-validation* estimates the

[7] While it is possible to learn from unlabelled or partially labeled training data, our focus will be on learning from completely labeled training data, since this is the approach we use for respiratory phase classification.

generalization error by: (1) splitting the training examples into N bins of equal capacity, (2) training the classifier on examples in every bin except for the ith bin, (3) testing the classifier on the examples in the ith bin and computing the error, (4) repeating steps (2)-(3) until all possible values of $i \in \{1,...,N\}$ have been accounted for, and (5) averaging the computed errors from all N tests.

4.5 Support Vector Machines

A support vector machine (SVM) is a non-probabilistic, binary classifier that relies on solving a quadratic programming problem to determine the optimal boundary (or "separating hyperplane") in a feature space for separating two classes of data.

SVMs have the ability to generalize well for arbitrarily large feature vectors – a quality that generally does not hold for other classification systems. This makes it a particularly attractive classification tool in applications that may require more extensive representations of the data (via larger feature vectors) in order to achieve optimized performance[8]. [3] attributes this phenomenon to the *maximum margin* property of SVMs which we discuss, along with the basic theory behind how SVMs work, in the next section.

4.5.1 Theory[9]

Suppose that we have a set of N labeled training data points. Let us denote the feature vectors for the data points by $\mathbf{x}_i \in \Re^d$ (where d is the dimension (or length) of the feature vectors), and their associated labels by $y_i \in \{+1,-1\}$, for $i \in \{1,...,N\}$. Let a linear hyperplane be the set of $\mathbf{x}_i$ that satisfy $\mathbf{x}_i \cdot \mathbf{w} + w_0 = 0$, where $\mathbf{w}$ is orthogonal to the hyperplane and $| w_0 | / \| \mathbf{w} \|$ is the closest distance between the hyperplane and the origin. If the distances between the hyperplane and a nearest positive example (i.e., label (y_i) = +1)

[8] This was the primary reason why we chose to use SVMs to solve the classification problem presented in this chapter.

[9] Much of material and notation in this section comes from [3].

and a nearest negative example (i.e., label (y_i) = -1) are denoted by d_+ and d_-, respectively, then the **margin** associated with that hyperplane is defined to be $(d_+ + d_-)$.

Under the assumption that the training data is **separable** (i.e., there exists at least one linear hyperplane that can perfectly separate the training data), and satisfies the constraints

$$\mathbf{x_i} \cdot \mathbf{w} + w_0 \geq +1 \quad \text{for } y_i = +1 \qquad \text{(Eq.1)}$$

$$\mathbf{x_i} \cdot \mathbf{w} + w_0 \leq -1 \quad \text{for } y_i = -1, \qquad \text{(Eq.2)}$$

the SVM finds an "optimal" separating hyperplane by choosing that which yields the *maximum margin*. This guarantees that the separation between the positive and negative examples is maximized, which is expected to yield the best generalization. To illustrate this point, Figure 5 shows a 2-dimensional (i.e., $d = 2$) feature space where the feature vector is $\mathbf{x} = [x_1 \quad x_2]'$. The x-axis corresponds to the value of the first feature x_1, and the y-axis corresponds to the value of the second feature x_2. Notice that the data points are separable in the feature space. In this case, the SVM would choose the hyperplane in the right plot over that in the left plot because it maximizes the margin. On either side of the hyperplane, the data points that have the minimum distance to the hyperplane are called *support vectors*.

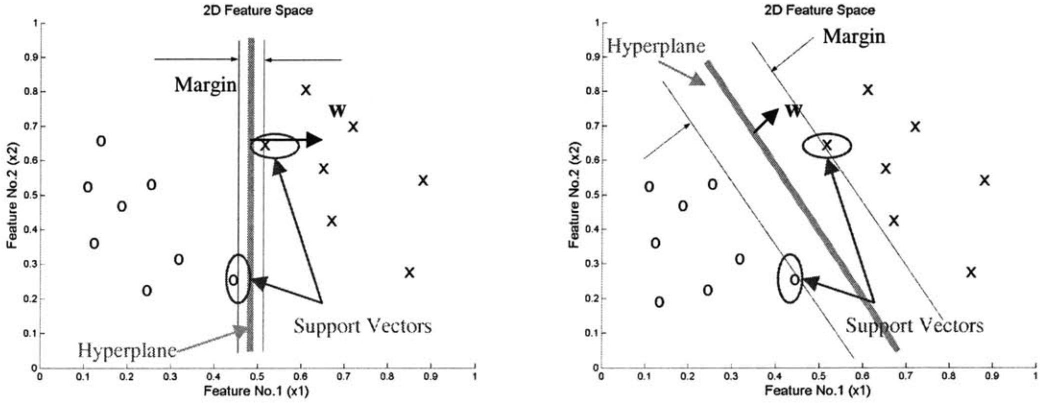


Figure 5. 2D feature space: suboptimal margin (left) vs. optimal margin (right). Data points legend: x → class ($y = +1$) , 0 → class ($y = -1$) .

An alternative notation for the constraints (Eq.1) and (Eq.2) is given by

$$y_i(\mathbf{x}_i \cdot \mathbf{w} + w_0) - 1 \geq 0 \text{ for } i \in \{1,...,N\}. \qquad \text{(Eq.3)}$$

Maximizing the margin is identical to minimizing $\|\mathbf{w}\|^2$. Hence, the problem of maximizing the margin subject to the constraint (Eq.3) can be solved using the Lagrangian method.

Suppose the Lagrange multipliers α_i, for $i \in \{1,...N\}$, are introduced. We can set up a "primal" objective function

$$J_P(\mathbf{w}, w_0, \alpha) = \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{i=1}^{N} \alpha_i (y_i(\mathbf{x}_i \cdot \mathbf{w} + w_0) - 1). \qquad \text{(Eq.4)}$$

An optimal solution can be found by *minimizing* J_P with respect to $\mathbf{w}$ and w_0 under the additional constraint $\alpha_i \geq 0$, for $i \in \{1,...,N\}$.

An alternate method for finding the optimal solution (which yields the same results as the problem formulation using the primary objective function) *maximizes* a "dual" objective function

$$J_D(\alpha) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2}\sum_{i,j=1}^{N} \alpha_i \alpha_j y_i y_j \mathbf{x}_i \cdot \mathbf{x}_j \qquad \text{(Eq.5)}$$

with respect to α_i subject to the constraints $\alpha_i \geq 0$, and $\sum_{i=1}^{N} \alpha_i y_i = 0$. The resulting solution for $\mathbf{w}$ is then

$$\mathbf{w} = \sum_{i=1}^{N} \alpha_i y_i \mathbf{x}_i = \sum_{i=1}^{Ns} \alpha_i y_i \mathbf{s}_i, \qquad \text{(Eq.6)}$$

where $Ns \in \{2,3,...\}$ is the number of support vectors, and $\mathbf{s}_i$ for $i \in \{1,...,Ns\}$ is the set of support vectors. The second equality holds because α_i is a non-zero value only for support vectors. The solution for w_0 can be computed by choosing a data point i and solving the equation

$$\alpha_i(y_i(\mathbf{w} \cdot \mathbf{x}_i + w_0) - 1) = 0, \qquad \text{(Eq.7)}$$

a necessary condition in the optimization problem. Because (Eq.7) can only be solved when α_i is non-zero, $\mathbf{x}_i$ must be a support vector (i.e., $\mathbf{x}_i \in \{\mathbf{s}_j : j = 1,...,Ns\}$).

In the case that the training data is **non-separable**, using the problem formulation presented for the separable case will yield an invalid solution. To solve this problem, *slack variables* ξ_i are introduced in the constraints (Eq.3). Those would allow the SVM to misclassify data points. The new constraints can be expressed as

$$y_i(\mathbf{x}_i \cdot \mathbf{w} + w_0) - (1 - \xi_i) \geq 0,$$

where $\xi_i \geq 0$ for $i \in \{1,...,N\}$.

Figure 6 presents a graphical interpretation of slack variables. In this case, a data point (in the class represented by an 'x' in the feature space) is misclassified because it lies on the side of the boundary that corresponds to the other class (represented by an 'o' in the feature space). Notice that the orthogonal distance between the misclassified data point and the hyperplane of the margin corresponding to its own class is a function of ξ_i. Larger distances correspond to a larger value of ξ_i. In general, if a data point lies on the correct side of its corresponding margin-hyperplane in the feature space, then $\xi_i = 0$; if the point lies on the incorrect side of that margin-hyperplane, then $\xi_i \geq 0$. A point is misclassified (i.e., an *error* occurs) if $\xi_i > 1$.

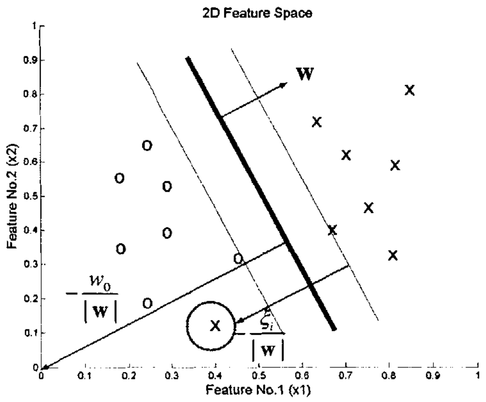


Figure 6. 2D feature space: Graphical interpretation of slack variables. A data point (circled) is misclassified.

The problem formulation for this non-separable case penalizes errors by adding a cost function F_C to the primal objective for the separable case (Eq.4):

$$F_c = C(\sum_{i=1}^{N} \xi_i),$$

where C is a **cost parameter** that is specified by the user[10]. A larger value of C corresponds to increasing the error penalty. The primal objective function is now given by

$$J_P(\mathbf{w}, w_0, \alpha, \xi, \mu) = \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{N}\xi_i - \sum_{i=1}^{N}\alpha_i(y_i(\mathbf{x}_i\cdot\mathbf{w}+w_0)-1+\xi_i) - \sum_{i=1}^{N}\mu_i\xi_i\,,$$

where μ_i is the Lagrangian multiplier which enforces the constraint $\xi_i \geq 0$.

The dual objective function turns out to be equivalent to that in the separable case (Eq.5), except for the constraint on α_i which is instead $0 \leq \alpha_i \leq C$. The solution for $\mathbf{w}$ is given by (Eq.6); the solution for w_0 is computed by solving (Eq.7) for any $0 < \alpha_i < C$.

In both the separable and non-separable cases, the SVM classification decision scheme becomes:

$$f(\mathbf{x}) = \mathbf{w}\cdot\mathbf{x} + w_0 = \left(\sum_{i=1}^{Ns}\alpha_i y_i \mathbf{s}_i\right)\cdot\mathbf{x} + w_0 = \sum_{i=1}^{Ns}\alpha_i y_i \mathbf{s}_i\cdot\mathbf{x} + w_0 \begin{array}{c}\text{"}y_i\text{=+1"}\\ > \\ = \\ < \\ \text{"}y_i\text{=}-\text{1"}\end{array} 0$$

Note that the problem formulation for non-separable training data can also be applied to separable data – in other words, it is possible to allow data points to be misclassified even when they are perfectly separable. So, if separable data (by definition) can be perfectly classified, why would it be advantageous to misclassify any?

Consider the example illustrated in Figure 7. Here we have separable data distributed over a 2D feature space. In the first scenario (shown by the left graph), a hyperplane that perfectly classifies the data points has been identified; in the second scenario (shown by the right graph), a hyperplane that misclassifies one data point has been identified. Notice that despite the larger error in the second scenario, the margin it achieves is substantially larger than that of the left graph, which is a desirable property. It is also apparent that the misclassified point in the second scenario lies at a relatively large distance away from the rest of the points in its class. One might argue that this point is simply an outlier (or a spurious

[10] It is possible to assign different values of C to the two classes ('x' and 'o' in this case). This might be beneficial in situations where a larger penalty is associated with misclassifying one class over the other. In classifying respiratory phases, we use the same C for both classes because it is assumed that the same loss is incurred from misclassifying either inspiration or expiration.

data point whose features have been corrupted by noise). Ignoring (or misclassifying) any outlier results in a boundary solution that more accurately reflects the true separation between the two data classes, and hence, achieves better generalization. For this example, under the assumption that the misclassified point is an outlier, opting for the hyperplane solution in the second scenario is optimal, since it is expected to yield better generalization.

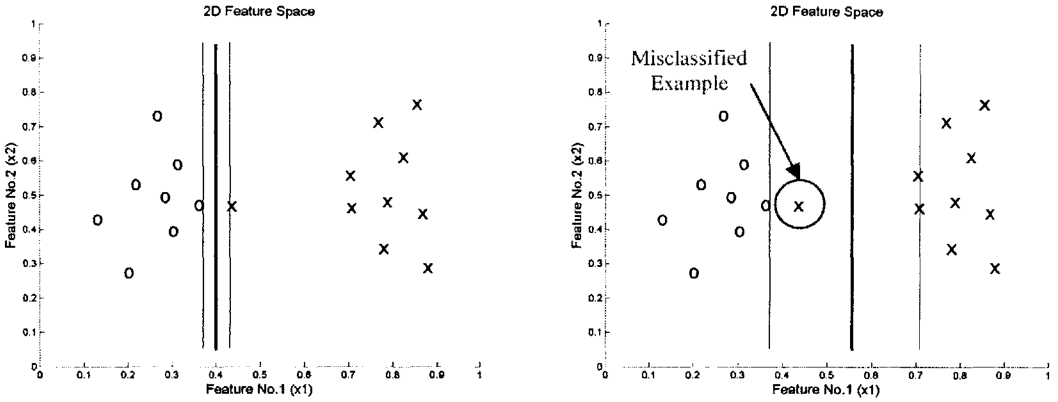


Figure 7. Illustration of the effect of choosing a different cost parameter (C) value on the resulting hyperplane solution.

Achieving either of the hyperplane solutions in Figure 7 entails choosing an appropriate value for the cost parameter C. Let C_1 and C_2 denote the cost parameters for the first and second scenarios, respectively. Given that a larger cost parameter corresponds to a larger error penalty, since the error in the second scenario is larger compared to the first scenario, we can deduce that the error penalty must be smaller in the second scenario, i.e., $C_1 > C_2$.

Our discussion so far has been limited to finding optimal **linear** boundaries between two classes of data in a feature space. In some cases, **non-linear** boundaries might be more suitable. A non-linear boundary can be achieved by *transforming* the original data points $\mathbf{x}$ into a different set of features Φ in the following way:

$$\mathbf{x}_i \in \Re^d \to \Phi(\mathbf{x}_i) \in \Re^p,$$

where $\mathbf{x}_i$ lies in a d-dimensional space (which we will refer to as the "original space") and $\Phi(\mathbf{x}_i)$ lies in a p-dimensional space (which we will refer to as the "feature space").

Suppose we are given data points $\mathbf{x}_i$ that are not separable by a linear hyperplane. If the transformation function is chosen accordingly, the *linearly non-separable* data points $\mathbf{x}_i$ in the original space could be mapped to *linearly separable* points $\Phi(\mathbf{x}_i)$ in the feature space.

The new problem formulation requires a simple substitution of $\mathbf{x}_i$ with $\Phi(\mathbf{x}_i)$ for $i \in \{1,\ldots,N\}$. The dual objective function in (Eq.5) becomes

$$J_D(\alpha) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2}\sum_{i,j=1}^{N} \alpha_i \alpha_j y_i y_j \Phi(\mathbf{x}_i)\cdot\Phi(\mathbf{x}_j), \qquad \text{(Eq.8)}$$

which yields the following classification decision scheme

$$f(\mathbf{x}) = \sum_{i=1}^{Ns} \alpha_i y_i \Phi(\mathbf{s}_i)\cdot\Phi(\mathbf{x}) + w_0 \;\; \overset{"y_i=+1"}{\underset{"y_i=-1"}{\overset{>}{\underset{<}{=}}}} \;\; 0. \qquad \text{(Eq.9)}$$

Notice that both the objective function (Eq.8) and the classification decision scheme (Eq.9) are functions of the Φ's only via their dot-products. A useful method for transforming the original feature space to a new feature space without having to explicitly compute the new feature vectors $\Phi(\mathbf{x})$ is to replace the dot-products of Φ with a **kernel function**. A kernel function can be expressed as

$$K(\mathbf{x}_i,\mathbf{x}_j) = \Phi(\mathbf{x}_i)\cdot\Phi(\mathbf{x}_j).$$

We list a few examples of kernel functions below:

- Linear → $K(\mathbf{x}_i,\mathbf{x}_j) = \mathbf{x}_i \cdot \mathbf{x}_j$
- Polynomial (order = p) → $K(\mathbf{x}_i,\mathbf{x}_j) = (1+\mathbf{x}_i \cdot \mathbf{x}_j)^p$
- Radial Basis Function (RBF) (sigma = σ) → $K(\mathbf{x}_i,\mathbf{x}_j) = e^{-\|\mathbf{x}_i-\mathbf{x}_j\|^2/2\sigma^2}$

Note that using the radial basis function (RBF) corresponds to having infinite-dimensional feature vectors, making the feature vectors impossible to compute explicitly. However, since kernel functions eliminate the necessity to compute the feature vectors explicitly, the RBF kernel makes it feasible to map the original space to an infinite-dimensional feature space.

The polynomial and RBF kernels are functions of the parameters p and σ, respectively. These parameters are specified by the user. Larger values of p and smaller values of σ correspond to higher dimensional feature spaces, which in turn allows for more complex boundaries to be resolved in the original feature space. While higher-dimensional feature spaces can achieve better classification accuracy for the training data, it could lead to poorer generalization.

On a final note, achieving the optimal balance between classification accuracy for the training and test data sets using SVMs amounts to identifying an optimal combination of the cost parameter C, kernel function K, and any associated kernel parameters.

We present the details of our work on respiratory phase classification next.

4.6 Method

We collected tracheal breath sounds from five normal subjects using an electronic stethoscope. One subject was a female (Subject No.1), and the rest were males (Subject No.'s 2 thru 5). The subjects were instructed to breathe continuously for an arbitrary period of time; the recordings were made when the subjects were at rest. The respiratory rates (in cycles per minute) ranged from 6 to 61 for Subject No.1, 15 to 16 for Subject No.2, 12 to 17 for Subject No.3, 24 to 29 for Subject No.4, and 11 to 15 for Subject No.5. Across all of the subjects, the respiratory rates had an average of 20 and a standard deviation of 10. Each recording contained multiple respiratory cycles, sampled at 22,050 Hz. We located all respiratory phase (inspiration and expiration) onsets automatically using the "Respiratory Phase Onset Detector" in our toolkit, and visually inspected the results, manually making any necessary corrections. We then used the onset information to parse the files into individual phase examples. We used a total of 806 examples (403 examples for each of inspiration and expiration) across all subjects.

We carried out the SVM cross-validation, training, and testing procedures, using a publicly available software package made for the MATLAB environment [37].

We tested the performance of two different feature vector representations on subject-specific data sets. We tested the performance of one feature vector representation on generalized (i.e., non-subject-specific) data sets.

4.7 Feature Selection

We identified two transformation procedures that map the original (time-domain) respiratory phase data to a feature vector. Our goal was to design feature vectors that illuminate the distinguishing characteristics between inspiration and expiration such that high classification accuracy could be achieved. For both transformation procedures, the features contain time-dependent frequency and energy information, consistent with the parameters that were studied by [17] and presented in the introductory section of this chapter. Our feature extraction (i.e., transformation) procedures are illustrated in Figure 8.

We begin with a description of *Feature Extraction Method No.1*. At the input to the system is the time-domain signal of a respiratory phase $x[n]$.

The first stage computes the *short-time Fourier transform (STFT)* of the input signal, which is a time-frequency representation. This representation is useful for analyzing signals that have varying properties (e.g., amplitude, frequency and phase) over time [27]. To compute the STFT, we use a window size of 0.0227 seconds, an overlap of 80% of the window size, and a fast Fourier transform (FFT) length equal to the sample-length of the window. Because the sampling frequency for our data is 22,050 Hz, the number of samples in the window size, overlap size, and FFT are 500, 400, and 500, respectively. Each sample of the FFT corresponds to a frequency bandwidth of 44 Hz (=22,050 Hz/(FFT length)=22,050/500).

The next stage extracts only the frequency samples that fall within the *[220→2,640] Hz frequency range*, since tracheal breath sounds are known to span primarily [200→2,000] Hz [11]. Then the *magnitude response* of the STFT is computed.

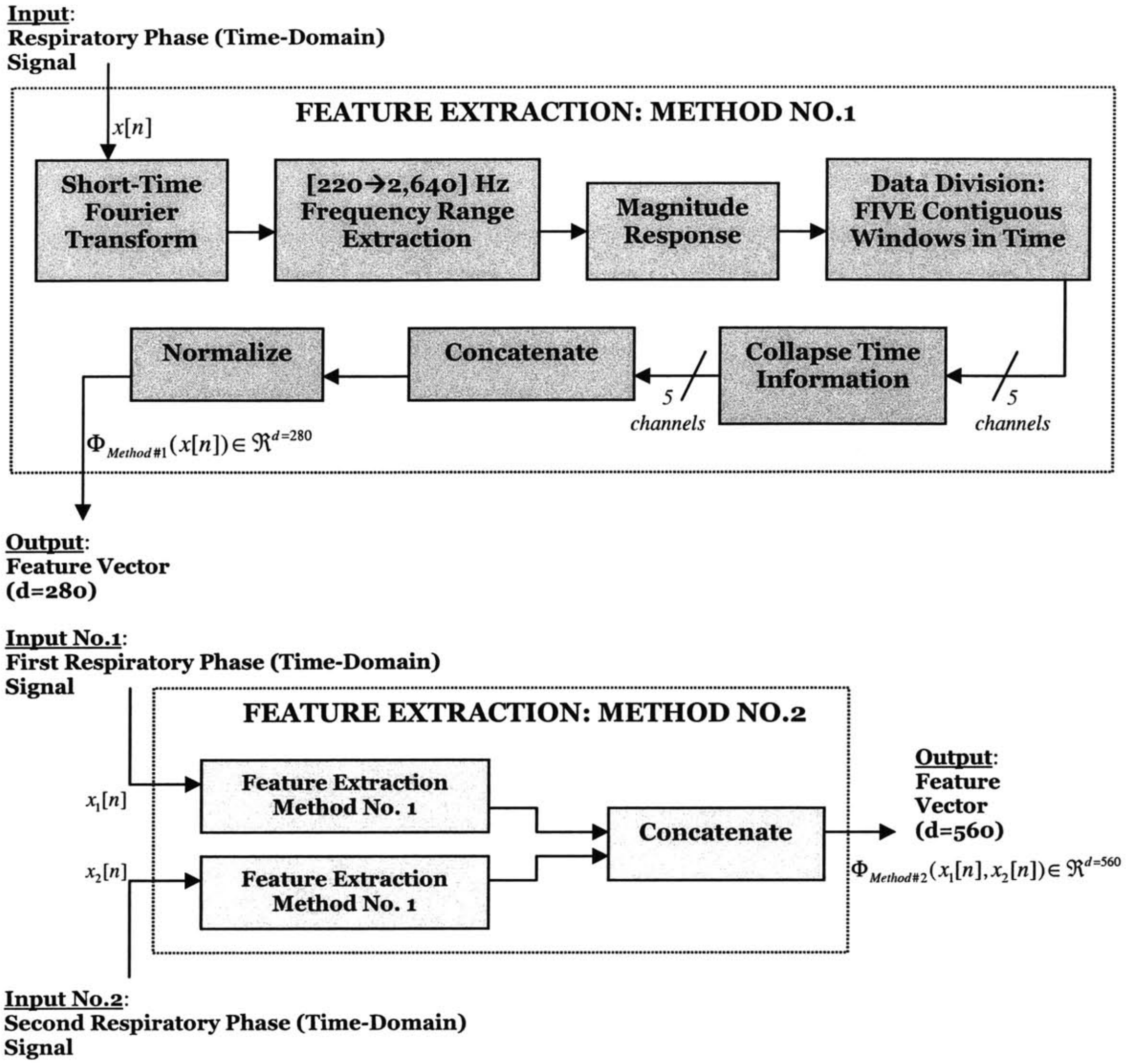

Input:
Respiratory Phase (Time-Domain) Signal
$x[n]$
FEATURE EXTRACTION: METHOD NO.1
Short-Time Fourier Transform
[220→2,640] Hz Frequency Range Extraction
Magnitude Response
Data Division: FIVE Contiguous Windows in Time
Collapse Time Information
5 channels
Concatenate
5 channels
Normalize
$\Phi_{Method\#1}(x[n]) \in \Re^{d=280}$
Output:
Feature Vector (d=280)
Input No.1:
First Respiratory Phase (Time-Domain) Signal
FEATURE EXTRACTION: METHOD NO.2
$x_1[n]$
Feature Extraction Method No. 1
$x_2[n]$
Feature Extraction Method No. 1
Concatenate
Output:
Feature Vector (d=560)
$\Phi_{Method\#2}(x_1[n], x_2[n]) \in \Re^{d=560}$
Input No.2:
Second Respiratory Phase (Time-Domain) Signal

Figure 8. Block diagrams for feature extraction: (top) Method No.1, and (bottom) Method No.2.

In the *data division: five contiguous windows in time* stage, the STFT is split into five non-overlapping sections. Each section corresponds to a time-frame of one-fifth the duration of the respiratory phase. For each section, the *collapse time information* stage adds the magnitudes of each frequency sample across the entire time-frame – this yields a vector whose indices correspond to frequency samples (each of which spans 44 Hz), and whose elements give the energy in the signal in a particular 44 Hz frequency band at that section in time.

The five vectors, corresponding to the five time-sections, are *concatenated* in an order that preserves the chronology of the sections; the output is a single vector, which we denote as $\Phi_{UNnormalized}$. Finally, $\Phi_{UNnormalized}$ is *normalized* by the sum of its elements, which yields the

280-dimensional feature vector output of Feature Extraction Method No.1: $\Phi_{Method\#1}(x[n]) \in \Re^{d=280}$. Figure 9 illustrates this procedure graphically.

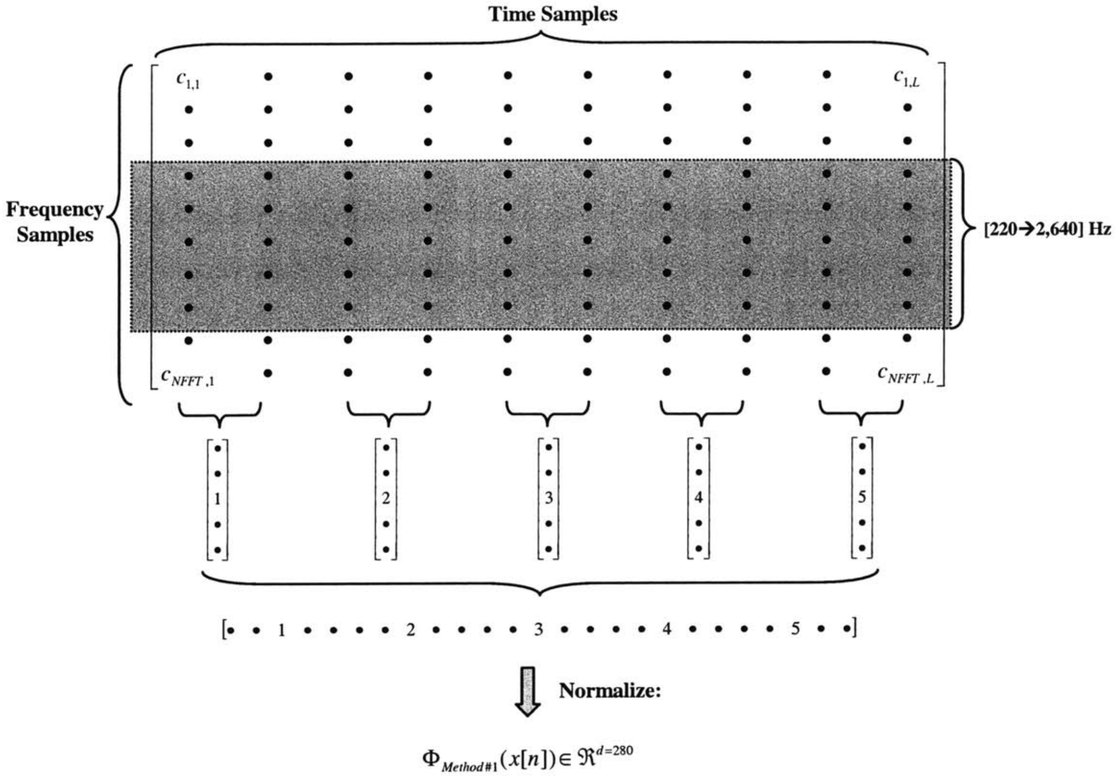


Figure 9. Illustration of Feature extraction Method No.1. The matrix contains the STFT magnitude coefficients.

Feature Extraction Method No.2 has two respiratory phase inputs, rather than one, to represent an example. More specifically, an example consists of an arbitrary respiratory phase (Input No.1) along with the respiratory phase which immediately follows it in time (Input No.2).

For example, if we denote inspiration and expiration of the ith respiratory cycle by $Inspiration_i$ and $Expiration_i$, respectively, and the inspiration and expiration of the immediately succeeding (i+1)th respiratory cycle by $Inspiration_{i+1}$ and $Expiration_{i+1}$, respectively, then valid {Input No.1, Input No.2} pairs are: $\{Inspiration_i, Expiration_i\}$ and $\{Expiration_i, Inspiration_{i+1}\}$.

Let $x_1[n]$ and $x_2[n]$ be the time-domain signals for Input No.1 and Input No.2, respectively. The feature extraction procedure computes feature vectors for both inputs using Method No.1. The feature vectors are then concatenated into a single vector, in chronological order, which yields the 560-dimensional feature vector output of Feature Extraction Method No.2: $\Phi_{Method\#2}(x_1[n], x_2[n]) \in \Re^{d=560}$.

Examples of feature vectors of respiratory phase data collected from a subject (Subject No.1) are shown in Figure 10. Figure 11 shows the *average* of the feature vectors of Figure 10.

4.8 Class Labels

For the feature vectors from Method No.1, we assign "Inspiration" examples a label of 'y=+1' and "Expiration" examples a label of 'y=-1.'

For the feature vectors from Method No.2, we assign {Inspiration, Expiration} pairs a label of 'y=+1' and {Expiration, Inspiration} pairs a label of 'y=-1.'

4.9 Cross-Validation Test Method

We used **10-fold** SVM cross-validation (CV) tests to compute optimal kernel parameters and cost parameters for the following kernel functions: linear, polynomial, and radial basis function (RBF).

For each kernel function, we defined a **[lower bound → upper bound] range** and **step-size** for the kernel parameter and cost parameter values over which CV tests were iterated using the training data set.

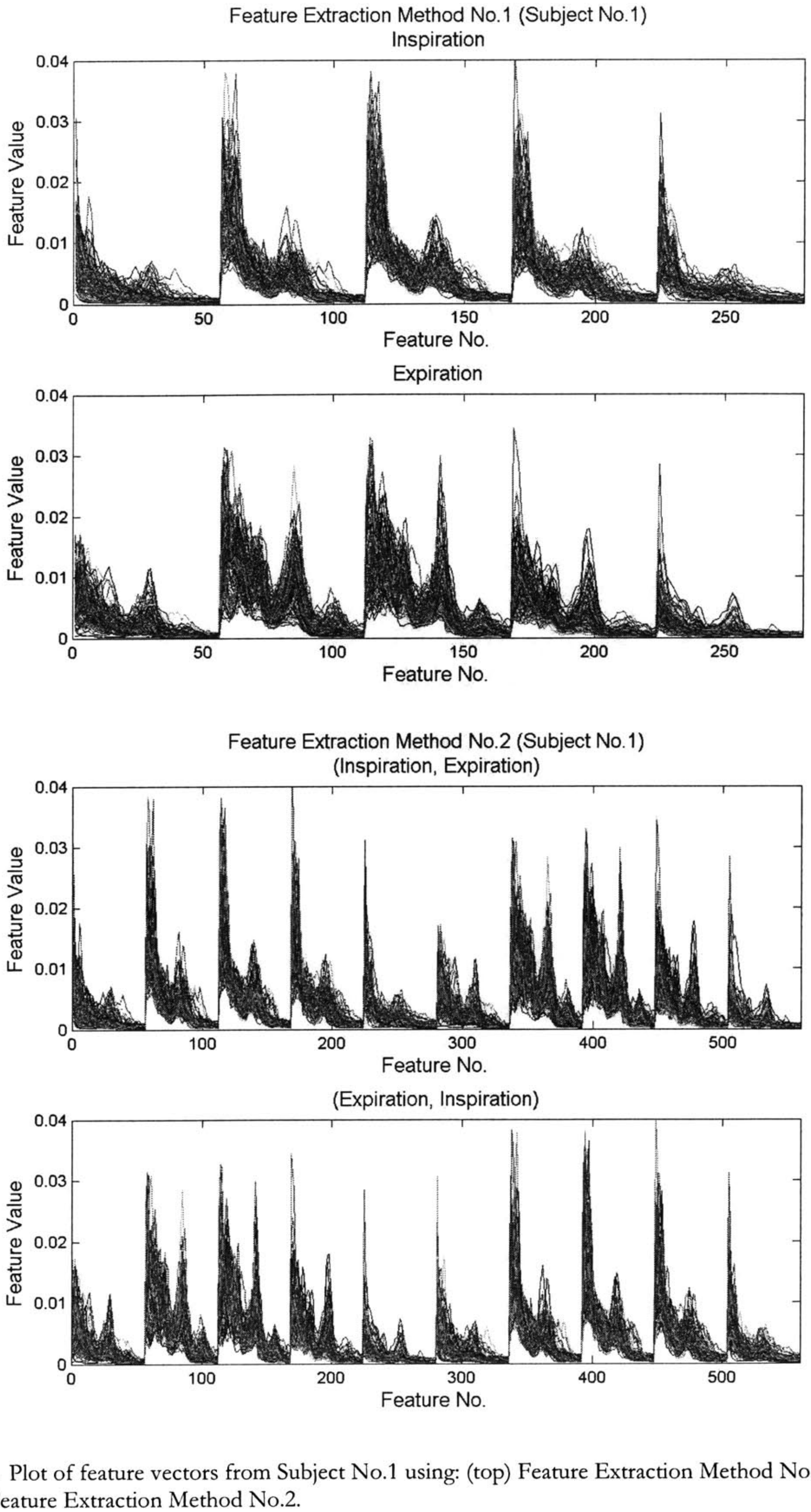


Figure 10. Plot of feature vectors from Subject No.1 using: (top) Feature Extraction Method No.1, and (bottom) Feature Extraction Method No.2.

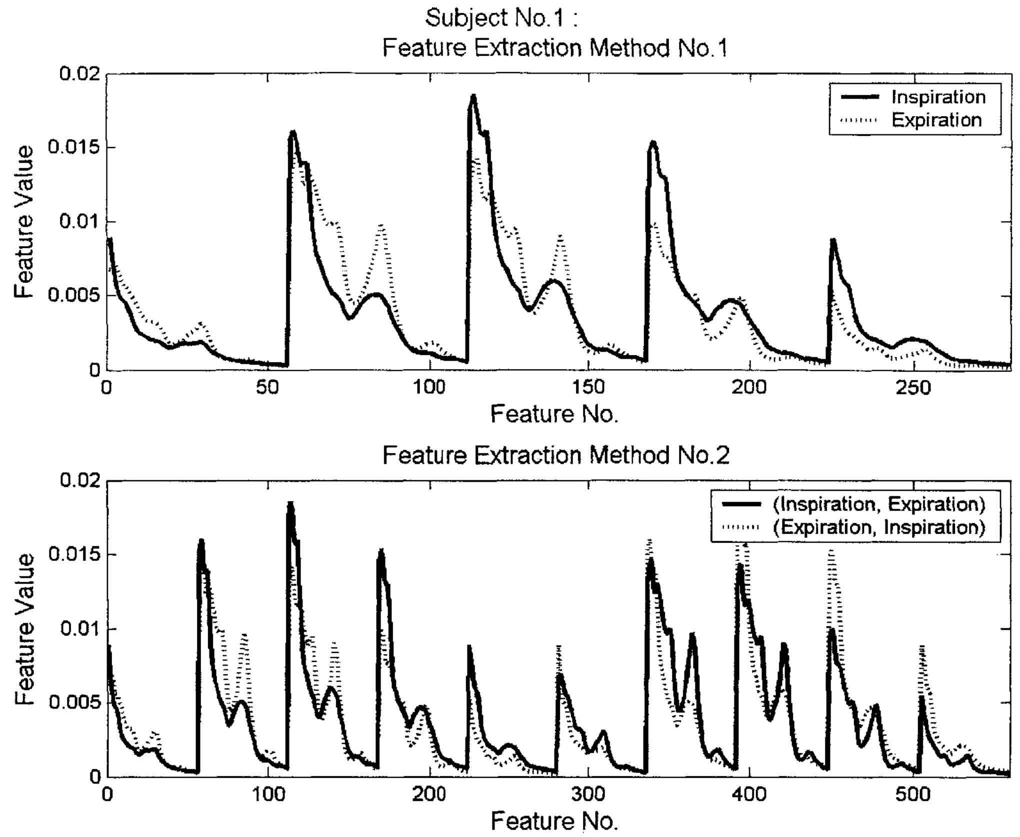


Figure 11. Plot of average feature vectors for Subject No.1 using: (top) Feature Extraction Method No.1, and (bottom) Feature Extraction Method No.2.

The cost parameter range and step-size (s) were [1 → 2000] and 100, respectively, for the linear and RBF kernels; the range and step-size were [1 → 400] and 20, respectively, for the polynomial kernel. If c(i-1) was the cost parameter value at the (i-1)th iteration, then the cost parameter at the ith iteration was c(i)=c(i-1)+s. The kernel parameter range and step-size depended on the kernel function. This information is summarized in Table 1.

	Lower Bound	Upper Bound	Step-Size (s)	ith Parameter ($k(i)$)
Linear	NA	NA	NA	NA
Polynomial	1	5	1	$k(i) = k(i-1) + s$
RBF	0.001	1	$\sqrt{2}$	$k(i) = sk(i-1)$

Table 1. Kernel parameter values used in the cross-validation tests.

For each kernel function, the **optimal** kernel parameter (k) and cost parameter (c) were identified by choosing the {k, c} pair which yielded the **lowest CV error**, where CV error is the percent of misclassified data points.

4.10 Training Procedure

After the optimal {k, c} parameter pair was identified from the cross-validation tests, we "trained" the SVM on the entire training data set.

4.11 Testing Procedure

We tested the performance of the "trained" SVM by computing the percent of the test data set that it classified correctly.

4.12 Subject-Specific Tests

In the *subject-specific* approach, we trained and tested the SVM separately for each of the five subjects from whom data were collected. The number of examples that were used for each subject varied. The minimum, maximum, mean and standard deviation of the number of examples we used from the subjects were 116, 240, 161, and 53, respectively. For all subjects, half of the data set consisted of "inspiration" examples, and the other half consisted of "expiration" examples. For each **subject**:

- We tested the feature vector representations from both Feature Extraction Method No.1 and Feature Extraction Method No.2. For each **feature vector representation**:
    - We tested the linear, polynomial, and RBF kernel functions. For each **kernel function**:
        - We ran the cross-validation tests, training procedure, and test procedure over FIVE rounds. For each **round**:

- The training data set and test data set were constructed by: (1) randomizing the order of all the examples, and (2) assigning the examples to either of two equally-sized sets, of which one was the training data set and the other was the test data set.

4.13 Generalized Tests

In the *generalized* approach, we trained the SVM on data from four of the subjects, and then tested the SVM on data from the fifth subject (i.e., the subject whose data were not included in the training procedure). We used 116 examples from each subject – it was important to use the same number of examples from each subject in this generalized approach because we did not want the results of the training procedure to be *biased* toward examples from any particular subject. For all subjects, half of the data set consisted of "inspiration" examples, and the other half consisted of "expiration" examples. For each **subject**:

- We tested the feature vector representation from only Feature Extraction Method No.2, since this method yielded better results than Method No.1 for the subject-specific tests.

- Its examples were set aside as the test data, whereas the examples from the other four subjects were set aside as the training data.

- We tested the linear, polynomial, and RBF kernel functions.

- We ran the cross-validation tests, training procedure, and test procedure once.

4.14 Results

Figures 12-15 show the results of the subject-specific tests, and Figure 16-17 shows the result of the generalized tests. For each subject and kernel function, we present the average percent (across the five rounds) of examples from the test data set that the SVM correctly classified; we also present the optimal {k, c} pairs from the test round that yielded the lowest classification error.

The mean (m) and standard deviation (σ) of the classification accuracy rates across all of the subjects and kernel functions are presented in Table 2 for each of the three classification models presented above.

Classification Model		$m \pm \sigma \%$
Subject-Specific	Feature Extraction Method No.1	$95 \pm 3\%$
	Feature Extraction Method No.2	$98 \pm 2\%$
Generalized, Feature Extraction Method No.2		$85 \pm 10\%$

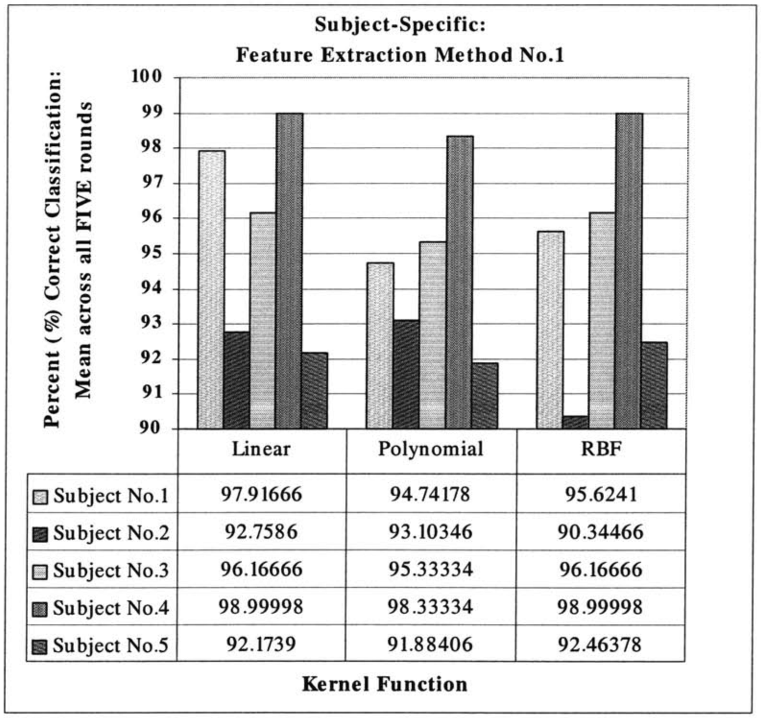
Table 2. Classification accuracy rates for each classification model: mean (m) and standard deviation (σ) across all subjects and kernel functions.

We discuss these results and compare them to those reported for other methods of respiratory phase classification in the next section.

4.15 Discussion

It is apparent from the classification accuracy rates summarized in Figures 12, 14, 16, and Table 2 that the Feature Extraction Method No.2 (FEM2) is more effective than Feature Extraction Method No.1 (FEM1), and that the subject-specific approach is more effective than the generalized approach.

The accuracy rate is higher for FEM2 than it is for FEM1 probably because FEM2 uses information about *two* consecutive respiratory phases, whereas FEM1 uses information about only a *single* respiratory phase. The important point is that there is a better chance of classifying a respiratory phase correctly if its contextual information (e.g., the acoustical characteristics of a neighboring respiratory phase) is known as well.



	Linear	Polynomial	RBF
▨ Subject No.1	97.91666	94.74178	95.6241
▨ Subject No.2	92.7586	93.10346	90.34466
▨ Subject No.3	96.16666	95.33334	96.16666
▨ Subject No.4	98.99998	98.33334	98.99998
▨ Subject No.5	92.1739	91.88406	92.46378

Figure 12. Subject-specific SVM test results for Feature Extraction Method No.1.

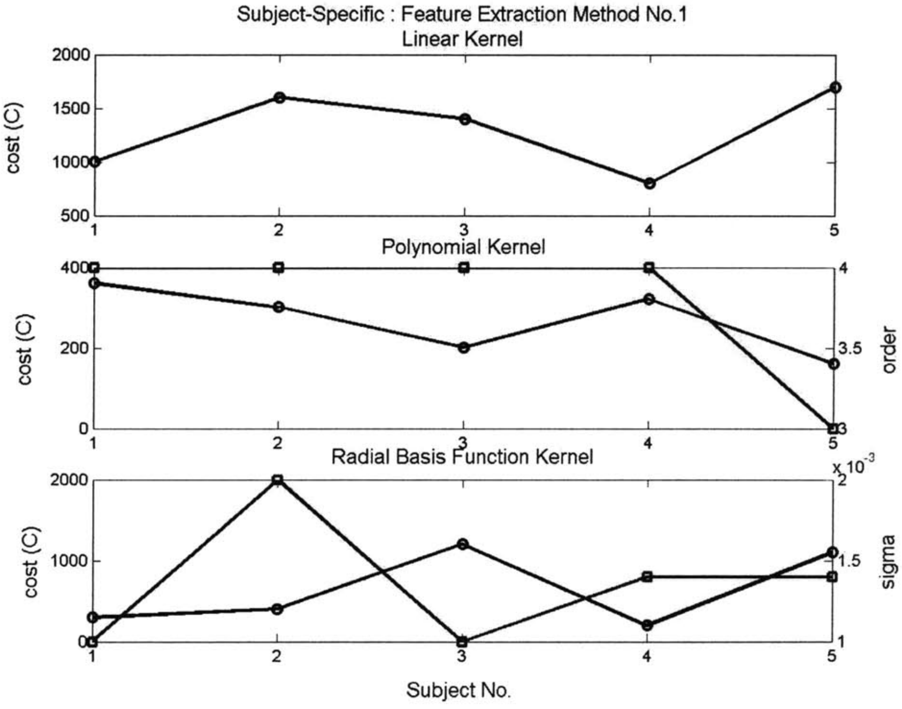
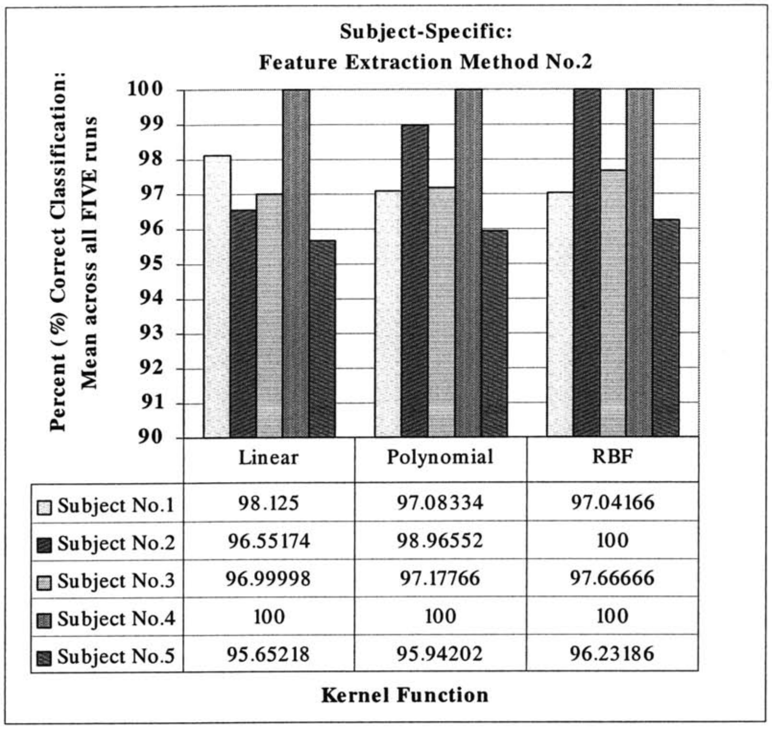


Figure 13. Subject-specific SVM using Feature Extraction Method No.1: Optimal parameters computed from cross-validation (CV) tests for each subject and kernel function; parameters were selected from the test round that yielded the lowest classification error. Left y-axis and plotted blue squares correspond to cost parameter values; right y-axis and plotted black circles correspond to kernel parameter values.



	Linear	Polynomial	RBF
□ Subject No.1	98.125	97.08334	97.04166
■ Subject No.2	96.55174	98.96552	100
▨ Subject No.3	96.99998	97.17766	97.66666
▦ Subject No.4	100	100	100
▩ Subject No.5	95.65218	95.94202	96.23186

Figure 14. Subject-specific SVM test results for Feature Extraction Method No.2.

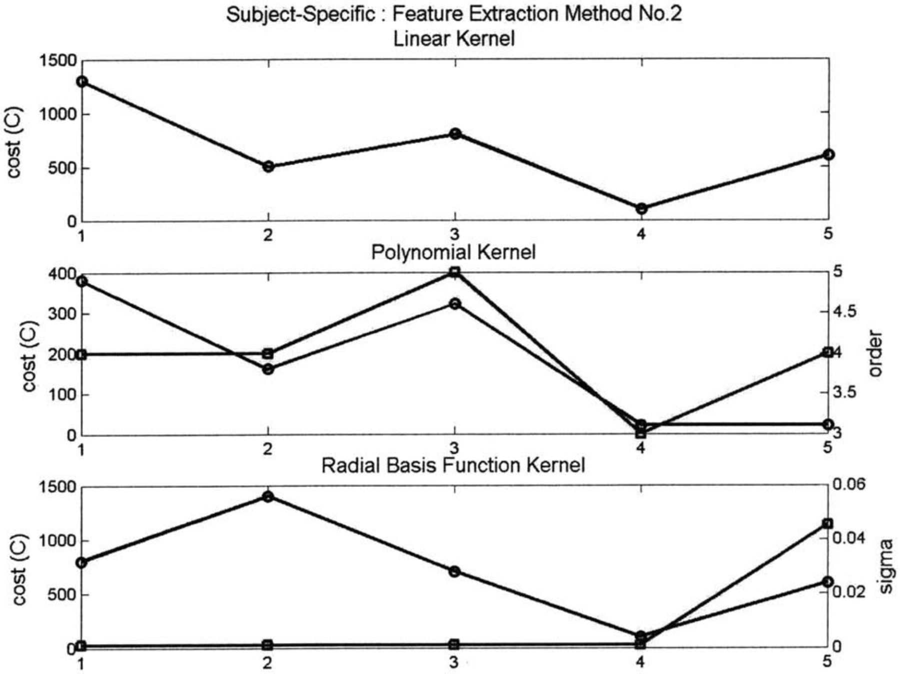
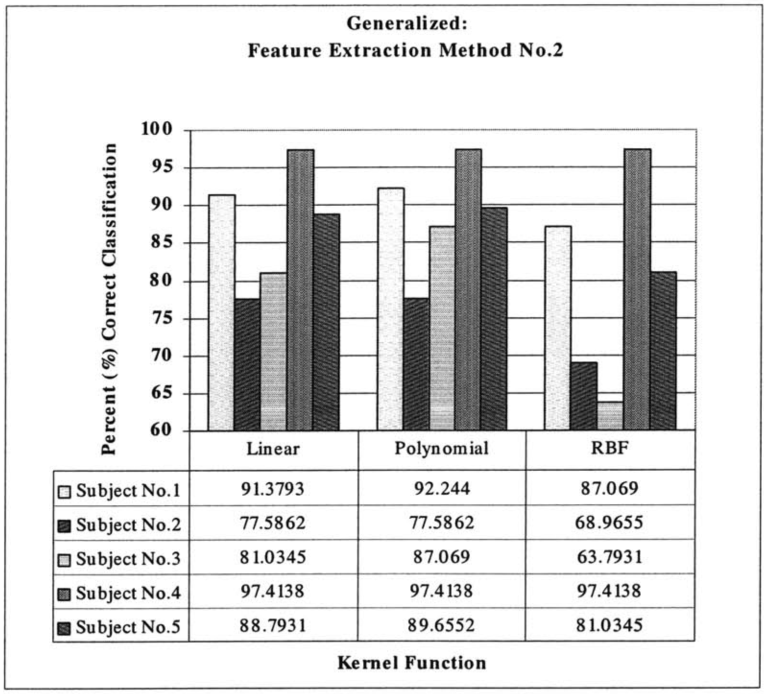


Figure 15. Subject-specific SVM using Feature Extraction Method No.2: Optimal parameters computed from cross-validation (CV) tests for each subject and kernel function; parameters were selected from the test round that yielded the lowest classification error. Left y-axis and plotted blue squares correspond to cost parameter values; right y-axis and plotted black circles correspond to kernel parameter values.



	Linear	Polynomial	RBF
□ Subject No.1	91.3793	92.244	87.069
■ Subject No.2	77.5862	77.5862	68.9655
□ Subject No.3	81.0345	87.069	63.7931
■ Subject No.4	97.4138	97.4138	97.4138
▨ Subject No.5	88.7931	89.6552	81.0345

Figure 16. Generalized SVM test results for Feature Extraction Method No.2.

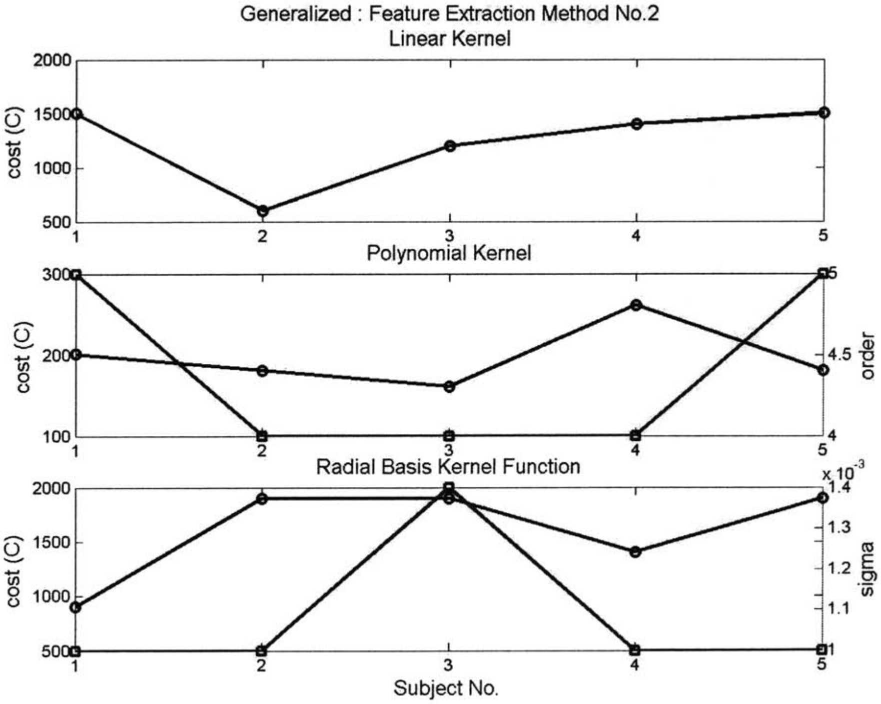


Figure 17. Generalized SVM using Feature Extraction Method No.2: Optimal parameters computed from cross-validation (CV) tests for each subject and kernel function; parameters were selected from the test round that yielded the lowest classification error. Left y-axis and plotted blue squares correspond to cost parameter values; right y-axis and plotted black circles correspond to kernel parameter values.

The subject-specific approach yielded accuracy rates that exceeded the accuracy rates of the generalized approach by ~ 10% (on average). The substantially higher accuracy rate of the subject-specific approach compared to that of the generalized approach could be attributable to the differences in breathing patterns between subjects. Figures 11 and 18-21 show the inter-subject variability of the (averaged) feature vectors for the five subjects that we studied. Even though some similarities may exist, the precise time-frequency characteristics of tracheal breath sounds (e.g., the locations of the spectral peaks, or the progression of the signal energy from the start to end of the respiratory phases) seem to differ for each subject. An example of a similarity in the averaged feature vectors across all of the subjects is that expiration has more energy in at least one high-frequency spectral peak than inspiration.

Different SVM kernel functions, kernel parameters, and cost parameters were optimal for different subjects. The test results from the subject-specific FEM1 classification model show that the linear kernel function with a cost parameter value of 1001 is optimal for Subject No.1; and the polynomial kernel function with a kernel parameter value (or order) of 4 and a cost parameter value of 301 is optimal for Subject No.2. It is also apparent from Figures 13, 15, and 17 that for any particular kernel function, the kernel parameter and cost parameter values vary between subjects.

The optimal kernel parameter values (graphed in Figures 13, 15, and 17) for all possible combinations of the classification model, kernel function, and subject have a minimum, maximum, and median of: (1) 3, 5, and 4, respectively, for the polynomial kernel function, and (2) 0.001, 0.0453, and 0.001, respectively, for the radial basis kernel function (RBF). Recall that we tested the range [1 → 5] for the **order** of the polynomial kernel function, and [0.001 → 1] for the **sigma** of the RBF. This means that all of the optimal kernel parameter values for the polynomial kernel function lie in the *upper* half of the corresponding parameter-range; and the optimal kernel parameters for the RBF lie in the *lower* half of its corresponding parameter-range. We mentioned earlier in this chapter that a larger polynomial order or smaller RBF sigma yields a classification decision boundary in the original data space that has a higher degree of complexity. Thus, we can conclude that for the respiratory phase classification problem presented here, complex decision boundaries are favorable since they yield higher accuracy rates.

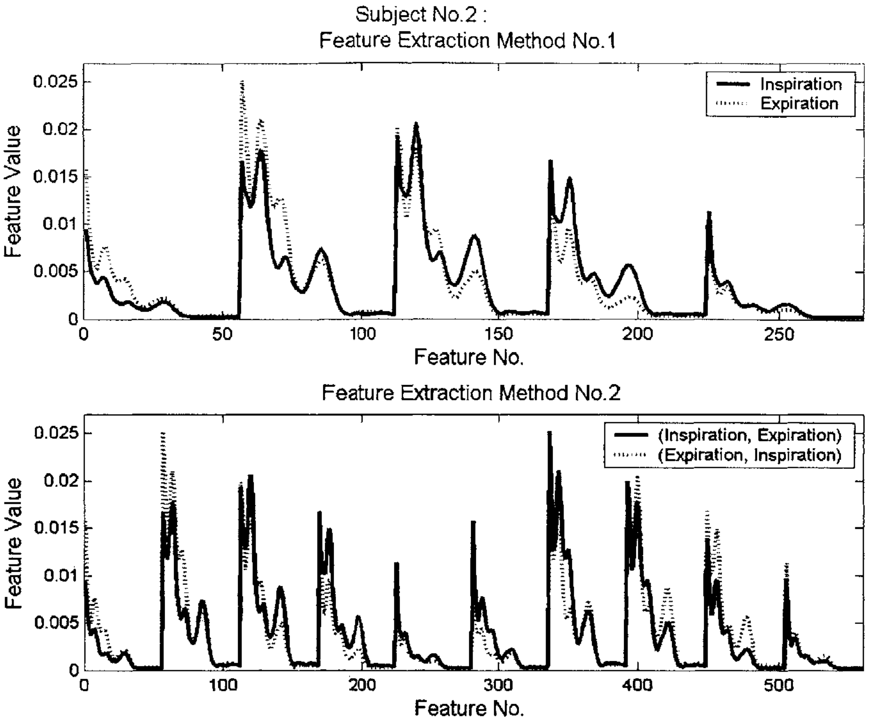


Figure 18. Plot of average feature vectors for Subject No.2 using: (top) Feature Extraction Method No.1, and (bottom) Feature Extraction Method No.2.

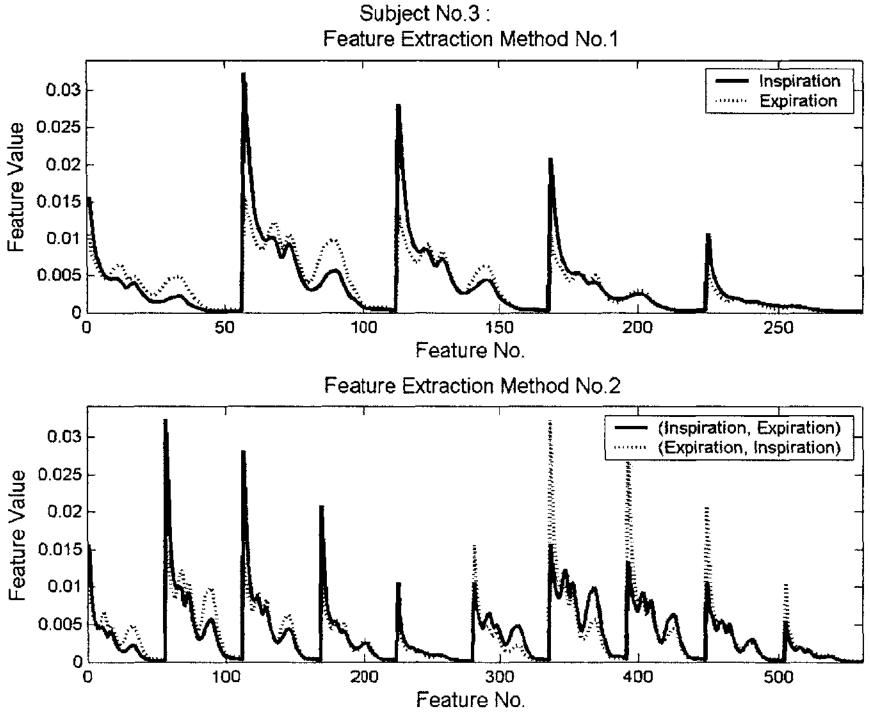


Figure 19. Plot of average feature vectors for Subject No.3 using: (top) Feature Extraction Method No.1, and (bottom) Feature Extraction Method No.2.

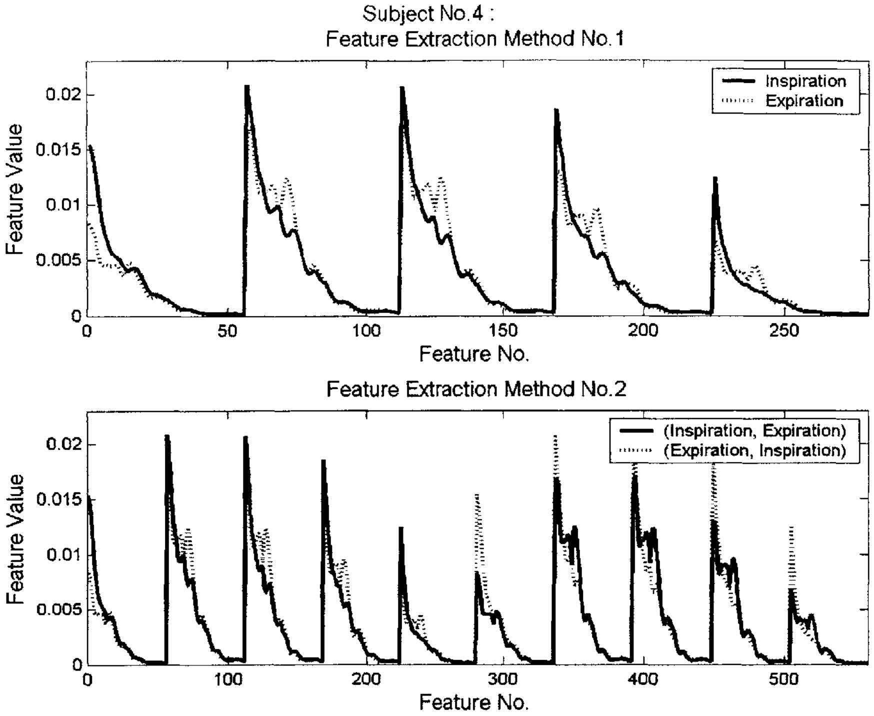


Figure 20. Plot of average feature vectors for Subject No.4 using: (top) Feature Extraction Method No.1, and (bottom) Feature Extraction Method No.2.

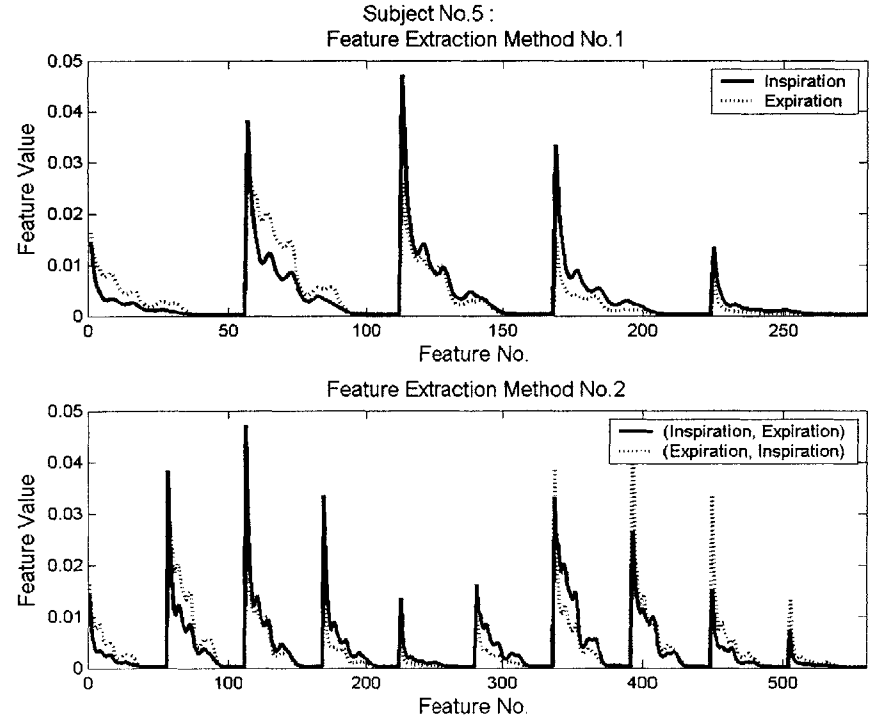


Figure 21. Plot of average feature vectors for Subject No.5 using: (top) Feature Extraction Method No.1, and (bottom) Feature Extraction Method No.2.

In general, the subject-specific classification model yielded high accuracy rates, with the accuracy rates (for each kernel function and subject) exceeding 90% and 95% for FEM1 and FEM2, respectively. This suggests that: (1) tracheal breath sounds for both respiratory phases are similar across different respiratory cycles for any particular subject (i.e., the intra-subject variability of tracheal breath sounds is low), and (2) inspiration can be distinguishable from expiration using tracheal breath sounds.

Average values for the parameters used by [17] to study tracheal breath sounds have not been computed for our data set, so we cannot verify whether the characteristics of our data are consistent with their findings.

The respiratory phase classifier proposed by [6] uses breath sounds from the chest and a generalized detection scheme. Their method has a reported accuracy rate (93%) that is considerably higher than the accuracy rate of our generalized detection method (85%). [6]'s algorithm performs well because it is based on the claimed 6 dB difference in the power of chest signals between inspiration and expiration. The accuracy rates of our subject-specific approaches (95% for FEM1 and 98% FEM2) are higher than the accuracy rate of their generalized approach.

The sensitivity and specificity rates reported by [35] for their respiratory phase classifier are between 95-98%. Their method uses RIP signals and a generalized detection scheme. Again, their approach is more accurate than our generalized detector, but has approximately the same accuracy as our subject-specific approaches. The main difference between their detector and ours is that theirs does not use acoustical data, whereas ours does.

Improving our generalized detector requires identifying better features to include in the feature vector. We used FEM2 to construct feature vectors; however, the feature vectors might contain information that is too subject-specific, making the detector unable to generalize well. On the other hand, both FEM1 and FEM2 seem to work well for the subject-specific classifiers.

The main advantage of using SVM classifiers is that they are able to accommodate high-dimensional feature vectors. SVMs can "learn" about the subtle, distinguishing characteristics between two classes given the right information (or features). The disadvantage of using SVM classifiers is that they are time-consuming to train – this is especially true for the subject-specific approach, since a different classifier needs to be trained for each subject. Subject-specific SVM classifiers also require access to a substantial amount of data examples to use for the training process.

4.16 Future Work

Further improvements need to be made to our respiratory phase classification scheme in order to make it robust to noise. High signal-to-noise (SNR) ratio is important in analyzing respiratory signal; however, shallow breathing or large mass over a subject's neck make acquiring useful (high SNR data) difficult. Tests need to be performed to analyze the effect of SNR on classification accuracy rates.

Additionally, if an application for respiratory phase classification requires data acquisition in a more spontaneous setting (e.g., during sleep or physical exertion), data collected from such a setting needs to be tested on. Our tests were performed on data that was collected in a relatively controlled setting, in which subjects were instructed to breathe continuously while at rest.

Chapter 5

Crackles Analysis

This chapter presents a method for automatically detecting and characterizing crackles in lung sounds. For each of the respiratory phases (inspiration and expiration) in a respiratory cycle, our system detects crackles and counts the number of crackles detected. For each detected crackle, the system outputs its: location in time, timing with respect to the respiratory phase in which it occurs, and pitch (via parameters called *initial deflection width (IDW)* and *two-cycle duration (2CD)).* The system also outputs graphical aids for analyzing the detected crackles.

5.1 Introduction[11]

Crackles are adventitious lung sounds that have a short, explosive, and non-musical quality. The duration of a single crackle is defined to be on the order of 20 ms [7]. Crackles can be heard at either the chest or mouth, and are indicative of pulmonary disease[12]. Crackles have distinctive characteristics that depend on the specific mechanisms by which they are generated, and hence, the underlying pulmonary disease. The parameters by which crackles are characterized are: (1) quantity (e.g., scanty or profuse), (2) pitch (e.g., fine or coarse), and (3) timing (e.g., early/mid/late in the respective respiratory phases in which they occur).

One technique that is used to identify the pitch of a crackle is called **time-expanded waveform analysis (TEWA)**. This technique refers to visualizing the waveform of a crackle on a resolution of $3{,}000\,mm \cdot \sec^{-1}$ [5], which can be achieved by zooming in on the digitized waveform on a computer screen. Crackles have a distinct morphology when observed in TEWA. An initial, low deflection in the amplitude of the waveform is followed

[11] Most of the discussion about crackles in this section is borrowed from [11] and [20].

[12] Crackles can be generated from healthy lungs as well. In these cases, however, because the crackles arise from forced maneuvers, they are rare occurrences.

by deflections with higher amplitudes. Figure 1 shows a sketch of a typical crackle waveform.

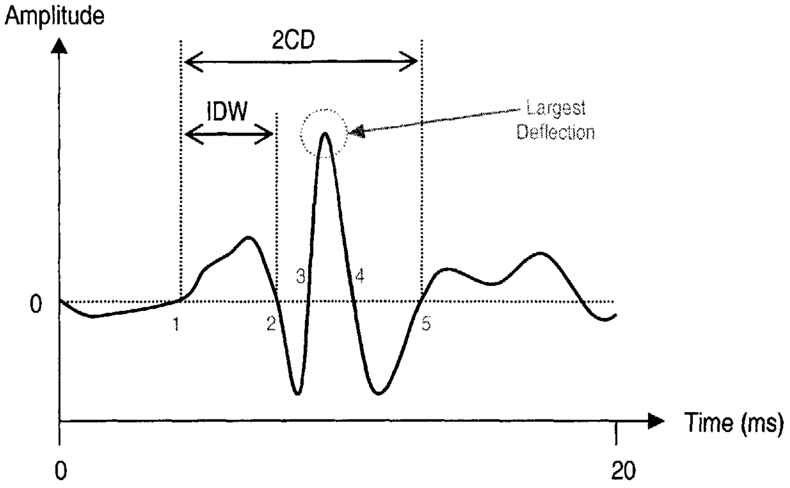


Figure 1. Sketch of a time-expanded crackle waveform.

A high pitch-quality is associated with fine crackles, and a low pitch-quality is associated with coarse crackles. *Initial deflection width (IDW)* and *two-cycle duration (2CD)* are the two most commonly used features for classifying fine and coarse crackles. The IDW corresponds to the duration of the initial deflection of the waveform; the 2CD corresponds to the duration of two cycles of the waveform, starting from the initial deflection. These parameters are depicted in Figure 1.

Several sources suggest **mean values** of IDW and 2CD durations for fine and coarse crackles:

- American Thoracic Society (ATS) [5]
  - Fine Crackles → [IDW ~ 0.7 ms] and [2CD ~ 5 ms]
  - Coarse Crackles → [IDW ~ 1.5 ms] and [2CD ~ 10 ms]
- Computerized Respiratory Sound Analysis (CORSA) [5]
  - Fine Crackles → [2CD < 10 ms]
  - Coarse Crackles → [2CD > 10 ms]
- Holford [15]
  - Fine Crackles → [IDW ~ 0.917 ms] and [2CD ~ 6.02 ms]
  - Coarse Crackles → [IDW ~ 1.25 ms] and [2CD ~ 9.32 ms]

There are two types of mechanisms by which crackles can be generated. The first type of mechanism occurs when a bolus of gas passes through an obstruction (e.g., excessive sputum) in one of the main airways (i.e., trachea or main bronchi). Crackles that are produced by this mechanism typically are heard at the mouth, are coarse in quality, and occur in early inspiration and in expiration. These crackles are associated with *obstructive* pulmonary diseases, which include chronic bronchitis, asthma, and emphysema.

The second type of mechanism that produces crackles occurs when the alveoli (i.e., air sacs in the lungs that are located at the end of the peripheral airways) pop open (from a completely deflated state) during inspiration. Crackles that are produced by this mechanism typically are heard at the chest, are fine in quality, and occur in late inspiration. These crackles are associated with *restrictive* pulmonary diseases, which include interstitial fibrosis, asbestosis, pneumonia, congestive heart failure, pulmonary sarcoidosis, scleroderma, and rheumatoid lung disease.

For certain pulmonary diseases (such as interstitial fibrosis) the quantity of crackles in a respiratory cycle can indicate the severity the disease [5].

5.2 Motivation

A computerized system that automatically detects and characterizes crackles in lung sounds could serve as a valuable diagnostic aid. It provides a convenient, minimally-invasive, and objective means for analyzing crackles. Descriptive parameters outputted from the system (e.g., quantity, timing, IDW, and 2CD) can be used to construct feature vectors for classifying diseases. Well-chosen feature vectors (i.e., those that **cluster**[13] the classes effectively) can lead to the development of automatic disease classifiers. Since the quantity of crackles is associated with the severity of certain pulmonary diseases, an automatic crackle detector can also be used to monitor the progression of those diseases.

[13] A cluster refers to the grouping of examples of a class of data in a specified feature space. If multiple data classes form *distinctive* clusters within the feature space, then the features can be used to classify unlabeled examples using one of a wide variety of classification schemes.

5.3 System Design

Figure 2 is a block diagram of our *crackles analysis* system.

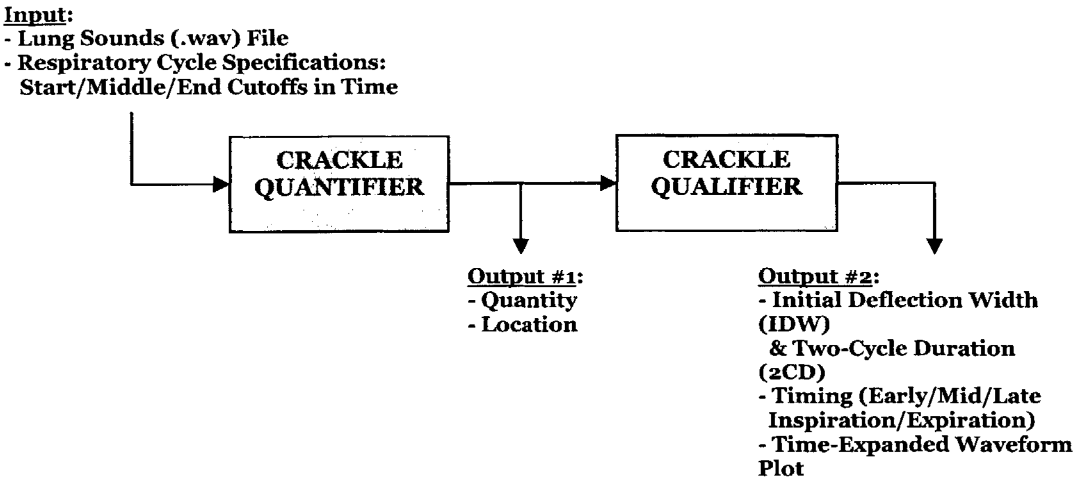

Input:
- Lung Sounds (.wav) File
- Respiratory Cycle Specifications:
Start/Middle/End Cutoffs in Time
CRACKLE QUANTIFIER
CRACKLE QUALIFIER
Output #1:
- Quantity
- Location
Output #2:
- Initial Deflection Width (IDW)
& Two-Cycle Duration (2CD)
- Timing (Early/Mid/Late Inspiration/Expiration)
- Time-Expanded Waveform Plot

Figure 2. Block diagram of *crackles analysis* system.

The system takes as input a file containing lung sounds and time-specifications that indicate the start of inspiration, transition between inspiration and expiration, and end of expiration (in time) for the respiratory cycle of interest. The *crackle quantifier* stage detects crackles and outputs their quantity and locations. The key aspects of our approach to crackle detection are that it:

- Analyzes the signal in purely the time-domain, and
- Employs adaptive threshold testing.

The *crackle qualifier* stage uses the output of the *crackle quantifier* stage to further characterize the signal. For each detected crackle, this stage outputs its initial deflection width (IDW) and two-cycle duration (2CD), and plots its time-expanded waveform so that users of the system can easily visualize its morphology.

Other methods for detecting crackles have been developed according to related literature, some of which are briefly described in the next section.

5.4 Background Literature/Related Work

[25] proposes a method for separating fine crackles from vesicular (i.e., normal and noise-like) lung sounds[14] using a non-linear filtering scheme (called the "stationary-non-stationary separating, ST-NST, filter"). They studied lung sound recordings (collected at the chest wall) from six patients with pulmonary fibrosis. They high-pass filtered all of the data using a cutoff frequency of 100 Hz prior to further processing. Their method for separating crackles from vesicular sounds is based on the premise that vesicular signals are stationary and, hence, can be modeled using the autoregressive method; whereas crackles are non-stationary signals that appear as transient bursts in (time-domain) amplitude. For a given input signal, the system has *two* outputs – the stationary and non-stationary signals – that correspond to estimates of the vesicular and crackles components of the input signal, respectively. Their algorithm for computing the two outputs is shown in Figure 3.

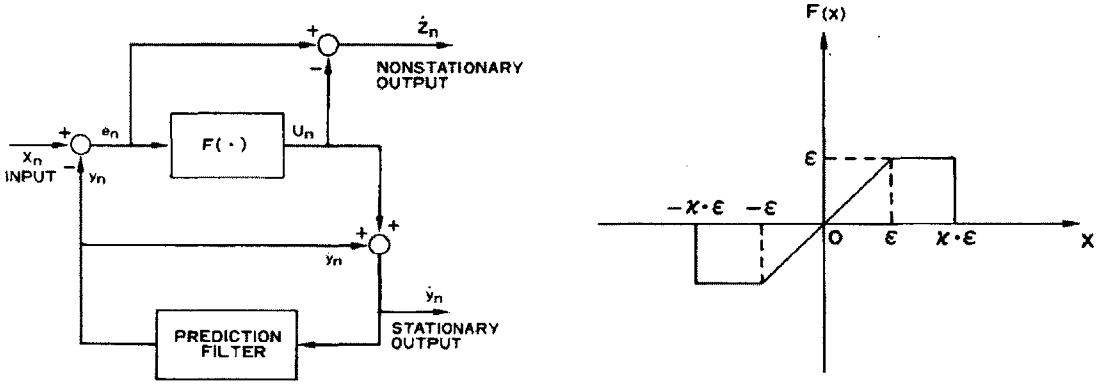


Figure 3. ST-NST filter proposed by [25]: (left) block diagram of system; (right) non-linear function corresponding to the $F(\cdot)$ block in the block diagram of the system.

The "prediction filter" block in the block diagram corresponds to the autoregressive (or linear prediction) model for the vesicular sounds. Its nth output ($\tilde{y}_n$) is a linear function of the M previous input values ($\hat{y}_{n-k}, \forall k = 1,\ldots,M$), i.e.,

[14] We discuss vesicular sounds in more depth in the "System Implementation - Crackles Quantifier Stage" section of this chapter.

$$\tilde{y}_n = \sum_{k=1}^{M} a_k \hat{y}_{n-k} ,$$

where $a_k, \forall k = 1,...,M$, are the prediction coefficients of the autoregressive model. The value of ε for $F(x)$ (as depicted in the right graph in Figure 3) can be expressed as a function of the probability of detecting a crackle (γ):

$$\int_{-\varepsilon}^{+\varepsilon} p(x)dx = 1 - \gamma ,$$

where $p(x)$ is the probability density function of the prediction error of the original signal (approximated as a Gaussian distribution). They empirically chose to use values of 15, 0.04, and 2 for M, γ, and χ, respectively. [12] reports performance figures for this (ST-NST) crackle detection method using their own data set. The sensitivity rates ($average \pm \sigma$%) for detecting fine and coarse crackles are 70.26 ± 10.78% and 64.38 ± 18.94%, respectively.

[12] proposes another method (called the "wavelet transform-based stationary-nonstationary, WTST-NST, filter") for separating crackles (both fine and coarse) from vesicular sounds. Their method relies on multi-resolution *wavelet analyses* of the input lung sounds signal in a decomposition-reconstruction iterative procedure. They studied six and five recordings of fine and coarse crackles, respectively. The crackles were labeled by a physician. The premise of their algorithm for separating crackles from vesicular sounds is that the wavelet transform coefficients (in terms of absolute value) of explosive peaks in the time-domain signal (corresponding to crackles) are large over many *wavelet scales* (i.e., frequency bands), whereas those of vesicular sounds attenuate rapidly with increasing scales. The algorithm begins by initializing $f(\lambda)$ to equal the N -length, normalized input signal at the first iteration (i.e., $k = 1$), and loops through the following procedure for each iteration:

(1) Compute the wavelet transform (WT) coefficients of $f(\lambda)$ for each of the M ($= \log_2 N$) wavelet scales (denoted by $WT^k(\lambda)$).

(2) Let $WT^{kC}(\lambda)$ and $WT^{kR}(\lambda)$ equal $WT^k(\lambda)$.

(3) For each of the M wavelet scales (i.e., $j = 1,...,M$):

   a. Compute the standard deviation of the WT coefficients (σ_j^k).

   b. Let the threshold (THR_j^k) be three times σ_j^k, i.e., $THR_j^k = F_{adj} \cdot \sigma_j^k$.

c. For each of the coefficients (i.e., $WT_j^k(i)$, $\forall i = 1,\ldots,\left(\frac{N}{2^j}\right)$):

i. Compare $WT_j^k(i)$ with THR_j^k. If $\left|WT_j^k(i)\right| < THR_j^k$, then set $WT_j^{kC}(i) = 0$; otherwise, set $WT_j^{kR}(i) = 0$.

(4) Construct *two* new signals – $C_k(\lambda)$ and $R_k(\lambda)$ – by applying the inverse wavelet transform to $WT^{kC}(\lambda)$ and $WT^{kR}(\lambda)$, respectively.

(5) Check whether the "stopping criterion" is satisfied. Let $STC = \left|E\{R_{k-1}^2(\lambda)\} - E\{R_k^2(\lambda)\}\right|$, where $E\{\cdot\}$ is the expected value function. If $STC \geq \varepsilon$, then set $f(\lambda) = R_k(\lambda)$, $k = k+1$, and start over at step (1); otherwise, let $L = k$ and leave the loop.

The crackles signal component is $DAS(\lambda) = \sum_{k=1}^{L} C_k(\lambda)$, and the vesicular signal component is $PVS(\lambda) = R_L(\lambda)$. They chose to use values of 3 and 10^{-5} for F_{adj} and ε, respectively. The reported sensitivity rates ($average \pm \sigma$ %) for detecting fine and coarse crackles using this (WTST-NST) scheme are 100 ± 0% and 97.5 ± 5.59%, respectively.

5.5 Method

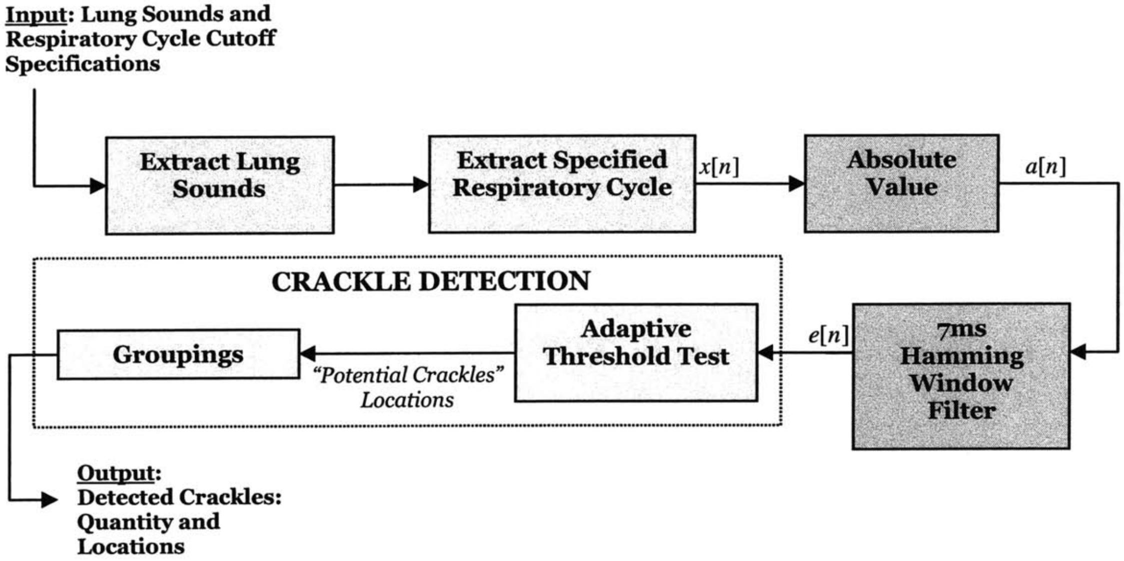
We collected and analyzed 4 files that contained crackles from public data repositories on the World-Wide Web [33]. Some files were described qualitatively (e.g., according to the pitch and timing of the crackles), while others were described as having been collected from patients diagnosed with certain pulmonary diseases (e.g., pneumonia and congestive heart failure).

5.6 Approach

In this section we discuss the implementation details for each stage of our *crackles analysis* system.

5.6.1 Crackle Quantifier Stage

The main objective of the *crackle quantifier* stage is to detect, quantify, and locate the crackles in the input signal. Figure 4 is a block diagram of the *crackle quantifier* stage.



Input: Lung Sounds and Respiratory Cycle Cutoff Specifications
Extract Lung Sounds
Extract Specified Respiratory Cycle
x[n]
Absolute Value
a[n]
CRACKLE DETECTION
Groupings
"Potential Crackles" Locations
Adaptive Threshold Test
e[n]
7ms Hamming Window Filter
Output: Detected Crackles: Quantity and Locations

Figure 4. Block diagram of the *crackle quantifier* stage.

Assume the input to our system is the crackles-containing lung sound signal depicted in Figure 5. This particular file contains four and one-half respiratory cycles, the third of which is pointed out on the plot. Suppose we want to analyze this third respiratory cycle. We need to input to the system the following time-cutoff specifications (in seconds): 8.5, 11, and 13, corresponding to the start of inspiration, transition between inspiration and expiration, and end of expiration, respectively.

We begin by removing signal components that lie outside of the frequency range for lung sounds, i.e., outside of [100 → 2,500] Hz, in the *extract lung sounds* stage. To do this, we use a finite impulse response (FIR) bandpass filter with low-frequency and high-frequency cutoffs at 100 Hz and 2,500 Hz, respectively. The filter was designed using the *window method* and a Blackman window of size one-twelfth the sampling frequency of the input signal[15].

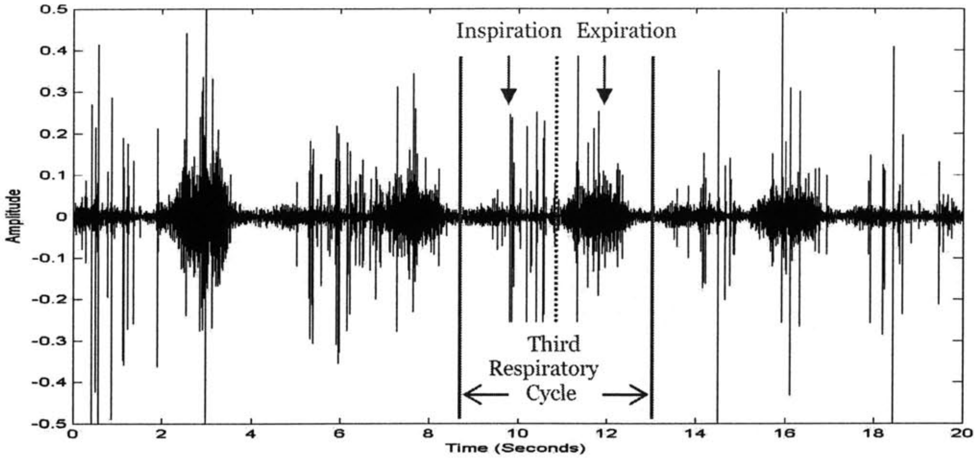


Figure 5. Time-domain plot of the input signal containing crackles.

The *extract specified respiratory cycle* stage extracts the part of the signal corresponding to the third respiratory cycle, as specified by the input time-cutoffs. Figure 6 graphs this signal (denoted by $x[n]$).

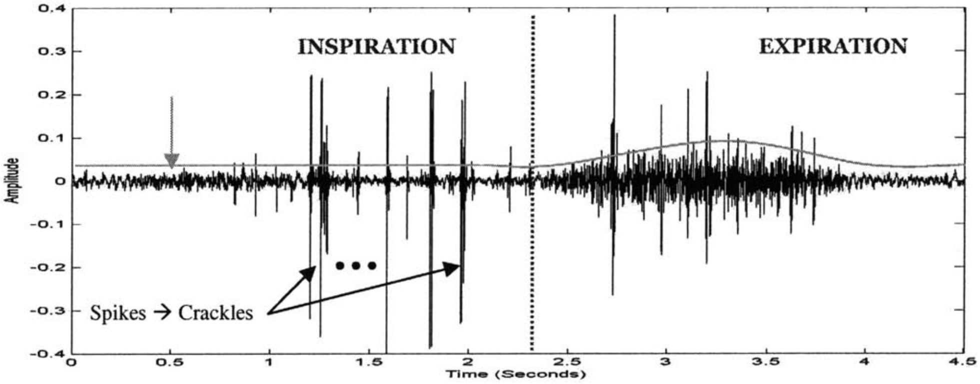


Figure 6. Time-domain plot of output ($x[n]$) of *extract specified respiratory cycle* stage.

Recall that crackles are defined as short, explosive, non-musical sounds. While some crackles are harder to detect because they are buried in vesicular (i.e., normal and noise-like) breath sounds, most are easily distinguishable both audibly and visually due to their high-energy, transient deviations from the vesicular breath signal. Notice in Figure 6 that the

[15] Refer to [27] for a detailed discussion on FIR filters.

vesicular breath sounds have a slowly varying, low-amplitude envelope. The crackles are high-amplitude "spikes" that are superimposed on the vesicular breath sounds. Later in the system, the crackle detector uses an adaptive threshold test to identify these spikes in the signal.

The next stage outputs the *absolute value* of $x[n]$ (denoted by $a[n]$). The *7 millisecond (ms) Hamming window filter* stage applies a normalized, causal, FIR filter (whose impulse response is a Hamming window that spans 7 ms) to $a[n]$ in order to extract the characteristics of its envelope – we denote the output as $e[n]$. We use a 7 ms window size is because it is smaller than the typical crackle duration (~ 20 ms). Since the performance of the crackle detector depends on how well the magnitude of crackles are preserved, we need to use a filter size that is small enough to prevent the diminishment of the spikes, while keeping it large enough to get a relatively smooth signal (which approximates the envelope of $a[n]$) at the output. A window size of 7 ms yields a desirable tradeoff. Figure 7 plots $a[n]$ and $e[n]$.

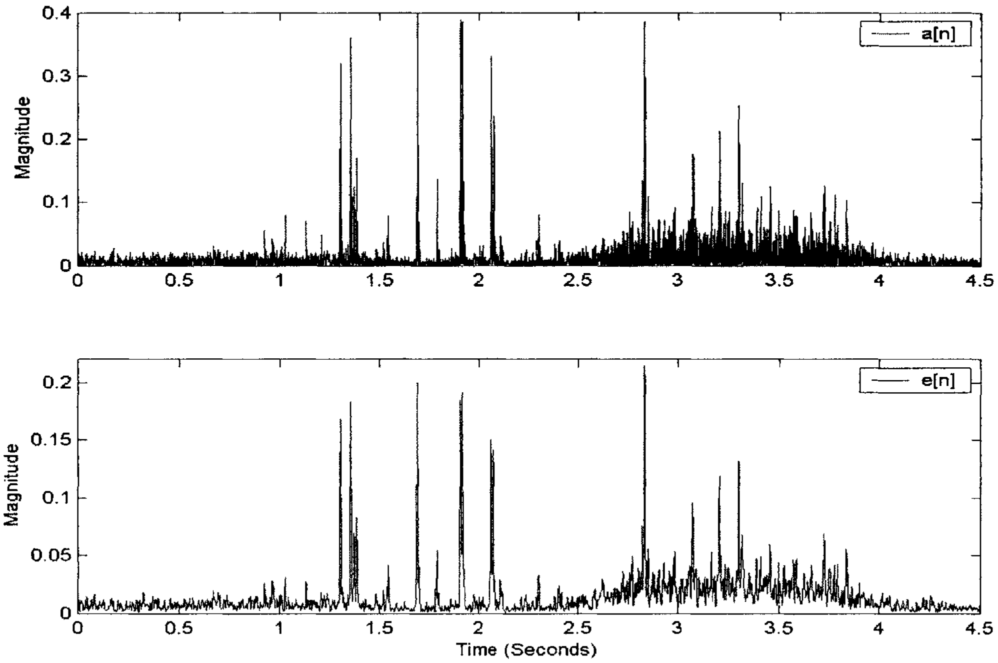


Figure 7. Plots: (top) output of *absolute value* stage ($a[n]$); (bottom) output of *7 ms Hamming window filter* stage ($e[n]$).

Crackle Detection Stage

The *crackle detection* stage detects crackles by searching $e[n]$ for high-amplitude spikes. The detection algorithm begins with the *adaptive threshold test* stage. For each sample of $e[n]$, the algorithm:

- Computes a threshold (T), which is defined as 2.8 times the mean value (m) of the samples of $e[n]$ that lie in the **40 ms** window centered around the current sample, i.e., $T = 2.8m$.

- Compares the value of the current sample to T. If the value exceeds the threshold, then it is considered to be part of a crackle; if it falls below the threshold, then it is considered to be part of the vesicular breath signal. It outputs a vector ($s[n]$) using the method described below.

```
If e[n] > T,
        s[n]=e[n]
Else
        s[n]=0
End
```

Figure 8 plots $s[n]$ overlaid on $e[n]$. Notice that the non-zero values of $s[n]$ trace the tips of the prominent spikes in the signal.

We chose to use a window duration of 40 ms for this procedure because:

1. It is small enough to allow for an **adaptive** analysis of the signal. What is meant by "adaptive" is that the threshold used for testing each data point is contingent upon the *local* characteristics (more specifically the local mean) of the signal. For example, notice that the vesicular signal has more energy (i.e., larger magnitudes) in the expiratory phase than the inspiratory phase. If we used a non-adaptive approach, each data point would be tested against a constant threshold (defined by the global mean of $e[n]$), which might lead to more missed spikes in the generally lower-magnitude inspiratory phase, and more false alarms in the higher-magnitude expiratory phase, thereby decreasing the accuracy of the detector. However, with an

adaptive approach, variations in the background signal are accounted for by defining the threshold to be a function of the local mean. This constrains our window size to be *less than the duration of a typical breath phase, ~ 1 second.*

2. It is large enough to actually detect spikes in the local region of analysis. Given that the test compares values to a threshold that is 2.8 times the local mean, if the window size is too small (e.g., on the order of a typical crackle duration, ~ 20 ms), then not even a very large spike in the signal would be detected. This constrains our window size to be *larger than the duration of a typical crackle, ~ 20 milliseconds = 0.02 second.*

A window size of 40 ms (= 0.04 second) met the above constraints (i.e., 0.02 second < 0.04 second < 1 second), and therefore was a reasonable choice. Figure 8 plots the variable-threshold sequence for this particular example. Notice the signal-dependence of the thresholds at each sample.

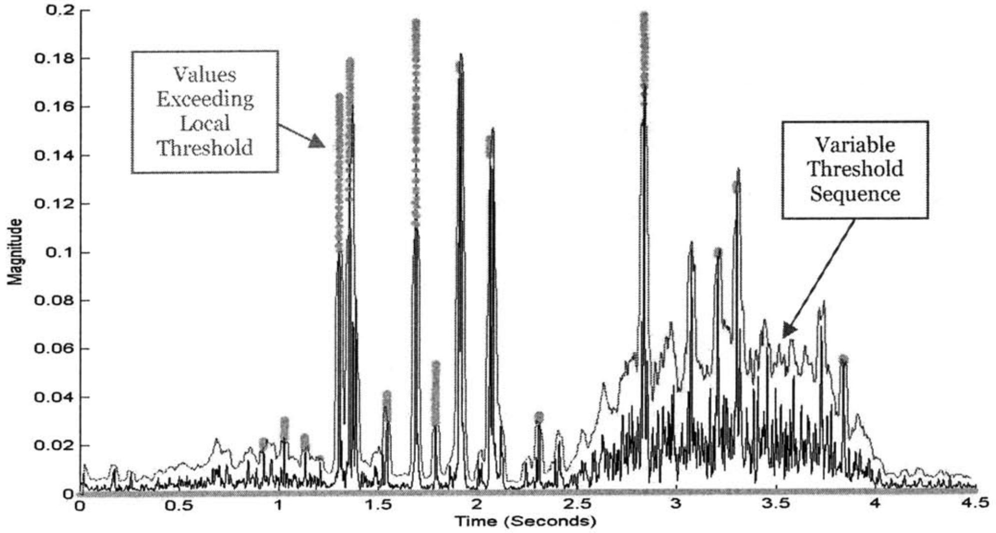


Figure 8. Plots of $e[n]$ (solid black line), $s[n]$ (dotted cyan line), and the variable-threshold sequence (solid red line).

Any time-segment in $s[n]$ that consists of only non-zero values is considered to be a "potential crackle." Each potential crackle is assigned a location that is defined to be the midpoint of the time-segment over which it spans. These potential crackle locations are then passed to the *groupings* stage of the crackle detector, which groups potential crackles that are too close together (in time) to be considered separate crackles. We define the **distance threshold** be 10 ms, which is half the approximate duration of a crackle (20 ms). Neighboring potential crackles that are located *less* than this distance threshold away from each other are grouped together. Potential crackles that are located more than the distance threshold from its neighbors form their own (single-member) group.

Each resulting group is considered as comprising a single **detected crackle**, and is assigned a location that is the average of the locations of the potential crackles that form its associated group. Finally, the *crackle quantifier* outputs the quantity and locations of the detected crackles. For this example, the detector outputs a crackle-quantity of 12 and 4 for inspiration and expiration, respectively. Figure 9 illustrates the results by plotting vertical lines over $x[n]$ at the locations of the detected crackles.

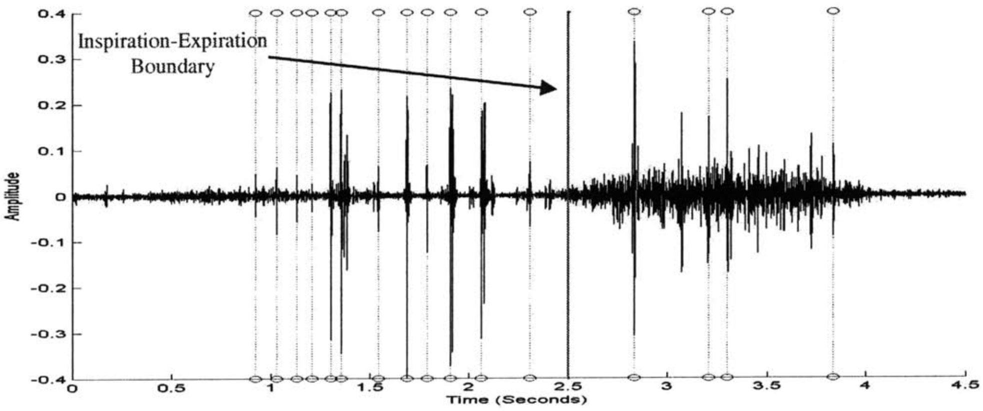


Figure 9. Plots of: (1) $x[n]$ (solid black line), (2) vertical lines that indicate locations of detected crackles (dotted blue lines), and (3) the inspiration-expiration boundary (solid red line).

5.6.2 Crackle Qualifier Stage

The *crackle qualifier* stage further characterizes each of the detected crackles. For each crackle, our system outputs its initial deflection width (IDW) and two-cycle duration (2CD). Our system also outputs a graphical aid for time-expanded waveform analysis (TEWA). Finally, for each of inspiration and expiration, our system graphs the distribution of crackles over the early, middle, and late thirds of the respiratory phase.

Initial Deflection Width and Two-Cycle Duration

In this section, we discuss the method we employ to compute the IDW and 2CD for each detected crackle.

Notice in Figure 1 that over the 2CD of the crackle waveform, there are total of five zero-crossings (labeled with the numbers 1 thru 5). Thus, the IDW is the distance between zero-crossings No. 1 and No. 2; similarly, the 2CD is the distance between zero-crossings No. 1 and No. 5.

For each crackle, the maximum magnitude in $x[n]$ of the waveform is identified within a 20 ms time-frame (equivalent to the approximate duration of a crackle) centered about its location. We assume that the maximum magnitude detected corresponds to the "largest deflection" pointed out in Figure 1. We then search for the three zero-crossings that lie to the left of it (No.'s 1, 2, 3) and the two zer0-crossings that lie to the right of it (No.'s 4, 5), and store their locations. The absolute difference between the locations of zero-crossings No. 1 and No.2 is the IDW; the absolute difference between the locations of zero-crossings No. 1 and No. 5 is the 2CD. The system outputs these IDW and 2CD values, and plots the inspiratory crackles (as red stars) and expiratory crackles (as blue circles) in the 2CD vs. IDW feature space, as shown in Figure 10. The 8th inspiratory crackle (in time) has an IDW and 2CD of 2.3 and 10 ms, respectively, and is circled in the plot.

Time-Expanded Waveform Analysis

The system outputs a graphical aid for time-expanded waveform analysis (TEWA). It plots the waveforms of all the detected crackles over their respective 2CD regions (as depicted in Figure 11). The user can zoom into any crackle of interest in order to view the details of its morphology. Figure 11 points out the 8th inspiratory crackle (in time); and Figure 12 shows that crackle over its 2CD region, zoomed in.

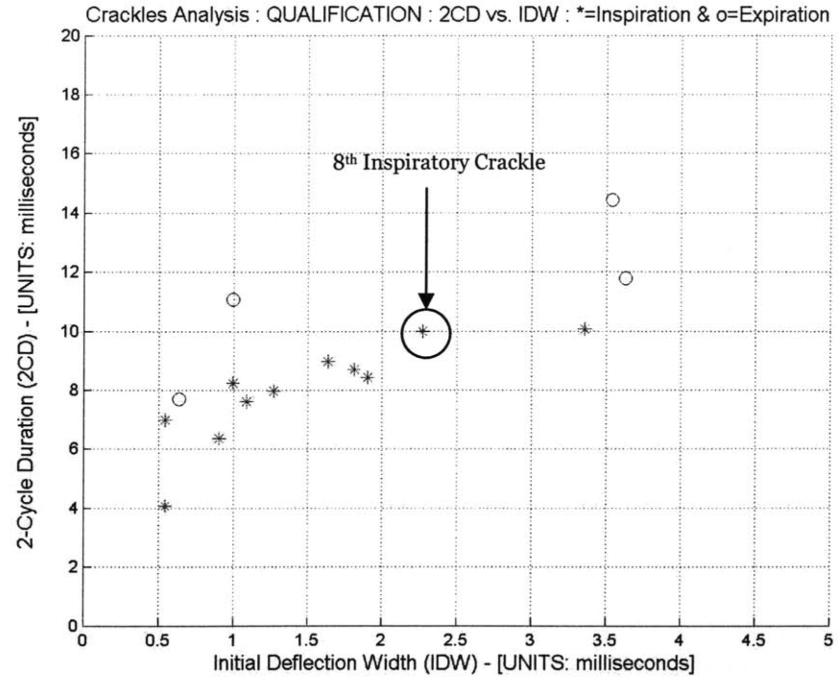


Figure 10. Plot of detected inspiratory crackles (red stars) and expiratory crackles (blue circles) in the two-dimensional feature space: 2CD vs. IDW.

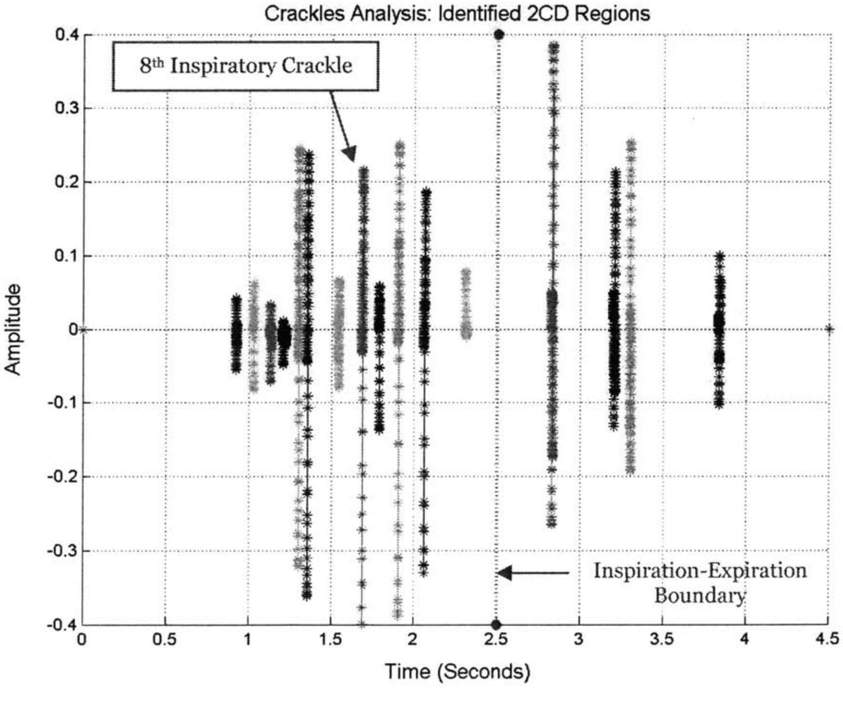


Figure 11. Graphical aid for *time-expanded waveform analysis* (TEWA): Plot of all detected crackles' waveforms over their respective 2CD regions. The 8th inspiratory crackle (in time) is pointed out.

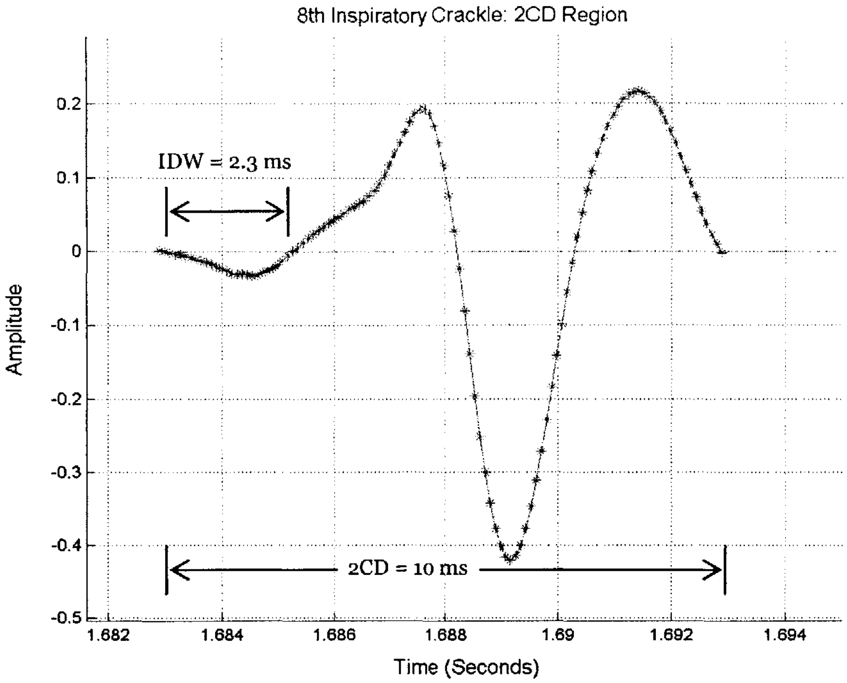


Figure 12. Graphical aid for *time-expanded waveform analysis* (TEWA): Waveform of the 8th detected inspiratory crackle (in time), zoomed in.

Timing Analysis

Recall that a useful characterization of a crackle is its timing with respect to the respiratory phase in which it occurs. Common nomenclature for describing the timing of crackles is "early/mid/late inspiration/expiration."

Our system classifies each detected crackle according to its timing in the respiratory phase using the classification scheme outlined in Table 1.

Class Label (Timing)	Region in respiratory cycle over which a crackle occurs:
Early-Inspiratory	First third of inspiration
Mid-Inspiratory	Middle third of inspiration
Late-Inspiratory	Last third of inspiration
Early-Expiratory	First third of expiration
Mid-Expiratory	Middle third of expiration
Late-Expiratory	Last third of expiration

Table 1. Classification scheme for *timing analysis* of detected crackles: Class labels are associated with regions in the respiratory cycle over which crackles can occur.

The system computes the number of crackles for each class label and displays the results, as shown in Figure 13. The results indicate that 0, 7, and 5 crackles were detected in early, mid, and late inspiration, respectively; and 1, 2, and 1 crackles were detected in early, mid, and late expiration, respectively.

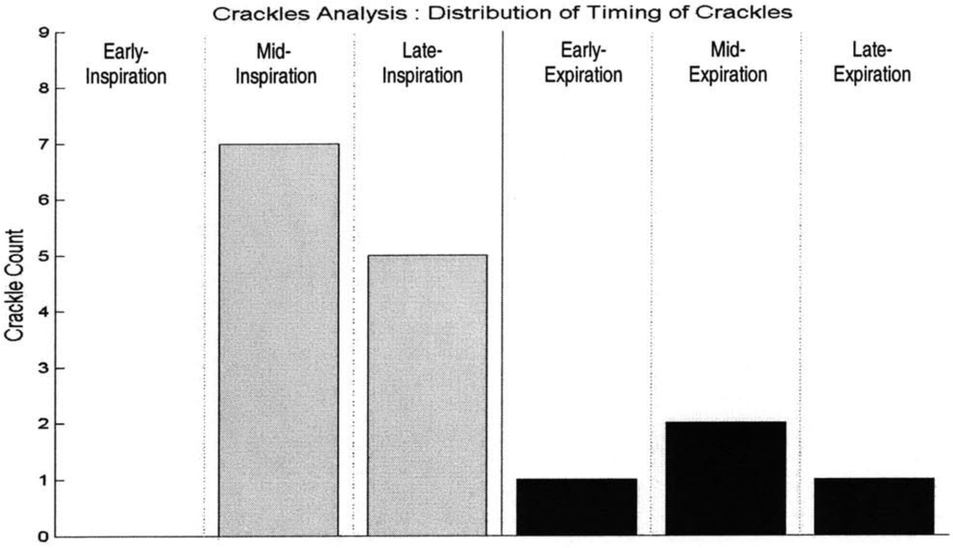


Figure 13. Timing of crackles: Number of detected crackles per class label.

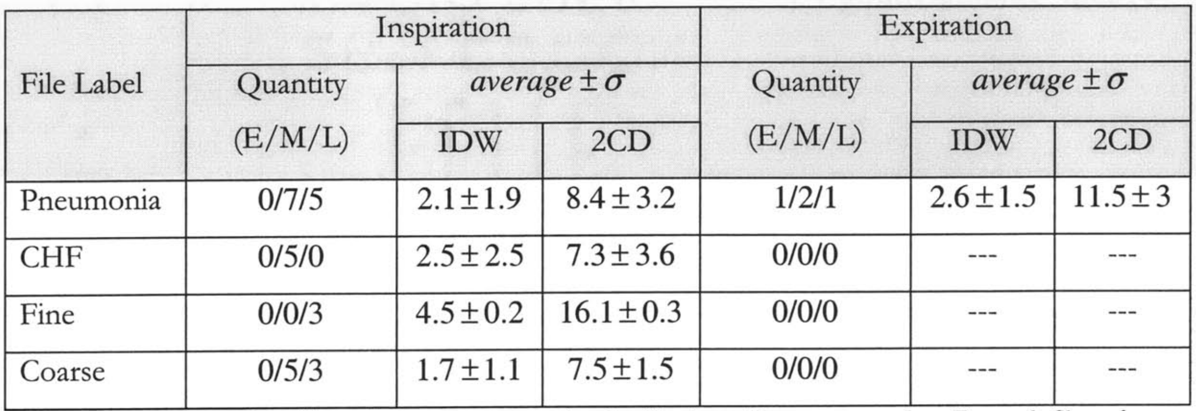
In the next section, we discuss the performance of our *crackles analysis* system.

5.7 Performance

To get an idea of its overall performance, we ran our *crackles analysis* system on 4 crackles, 5 tracheal breath (each from a different subject), and 6 wheeze sound files. We analyzed one respiratory cycle in each of the crackles files, and the first four seconds of each of the wheeze and tracheal breath files.

The repositories from which we obtained the crackles sounds for our analyses did not explicitly label the individual crackles; hence, we do not provide performance statistics (e.g., sensitivity and specificity rates) for our *crackle detector*. However, the crackles files were labeled qualitatively as "pneumonia,"[16] "congestive heart failure (CHF)," "fine," and "coarse," respectively.

Crackles were detected in all of the crackles files. Table 2 summarizes the output of the system. For each of the files, and for each of inspiration and expiration, the table indicates the number of crackles detected over early, mid, and late respiratory phase, and the average and standard deviation of the IDW and 2CD values (in milliseconds).



File Label	Inspiration			Expiration		
	Quantity (E/M/L)	*average* ± σ		Quantity (E/M/L)	*average* ± σ	
		IDW	2CD		IDW	2CD
Pneumonia	0/7/5	2.1±1.9	8.4±3.2	1/2/1	2.6±1.5	11.5±3
CHF	0/5/0	2.5±2.5	7.3±3.6	0/0/0	---	---
Fine	0/0/3	4.5±0.2	16.1±0.3	0/0/0	---	---
Coarse	0/5/3	1.7±1.1	7.5±1.5	0/0/0	---	---

Table 2. Summary of *crackles analysis* system output for the (four) crackles test examples. For each file and respiratory phase, the quantity of crackles (in early/mid/late respiratory phase), and average and standard deviation of the IDW and 2CD values are reported.

Crackles were falsely detected in 2 out of 5 of the tracheal breath files. 6 false positives were detected in one of these files; 1 false positive was detected in the other file

Crackles were falsely detected in 3 out of 6 of the wheeze files. Two of these three files contained 1 and 2 false positives, respectively, that are probably attributable to noise; the remaining file seemed to contain crackles, 5 of which the detector picked up.

5.8 Discussion

In this section, we discuss the performance results for our *crackles analysis* system. We also compare our work on analyzing crackles with those of others (discussed in the "Background Literature/Related Work" section of this chapter).

[16] The crackles test example from this "pneumonia" file is the same as the example used earlier in the "System Implementation" section of this chapter.

5.8.1 Sensitivity Analysis

We begin by discussing the results in Table 2. Assume that we the criterion for classifying fine and coarse crackles suggested by CORSA (i.e., the 2CD of fine and coarse crackles are less than and greater than 10 ms, respectively). Because the average 2CD value (for inspiration) for the "pneumonia" and "CHF" files are 8.4 and 7.3, respectively, these crackles (on average) can be classified as fine crackles. For both files, the inspiratory crackles all occur either in the middle or late third of their respective inspiratory phases. Hence, the crackles can be classified as late-inspiratory. Recall that pneumonia and CHF are both restrictive pulmonary diseases, which are associated with late-inspiratory and fine crackles. Therefore, the timing and average IDW/2CD values of the detected crackles in the "pneumonia" and "CHF" files seem to be consistent with the expected timing and IDW/2CD values of crackles for their associated diseases[17].

On the other hand, the results (in Table 2) for the "fine" and "coarse" files are not consistent with the expected average IDW and 2CD values. According to any of the suggested criteria for classifying fine and coarse crackles, and given the average IDW and 2CD values for the two files, the "fine" file should be classified as containing coarse crackles, and the "coarse" file should be classified as containing fine crackles. Possible explanations for this apparent discrepancy include: (1) our method for computing the IDW and 2CD is inappropriate, or (2) some of the detected crackles might not have at least two complete cycles in their waveforms.

IDW is the time between zero-crossings of the initial deflection of the crackles waveform, and 2CD is the time between the start of the initial deflection and the point at which two cycles of the waveform is completed. There are a few cases in which identifying the "true" start and end of the crackle's waveform (and hence, the "true" IDW and 2CD) is more complex than simply identifying zero-crossings [15]. One case is when the crackles are superimposed on a "wandering baseline" due to the background vesicular signal; in this case, [15] suggests using baseline-crossings instead of zero-crossings to compute the IDW and 2CD values. Another case is when there is a "rounded knee" at the beginning of the

[17] Note that the average IDW and 2CD values for the "pneumonia" and "CHF" files are not consistent with "fine crackles" if either the ATA or [15] criterion for classifying fine and coarse crackles is used instead.

waveform, which makes identifying the start of the initial deflection difficult. In this case, [15] suggests identifying the start by locating the intersection between the baseline and the line tangent to the steepest part of the first upstroke or downstroke of the initial deflection. Figure 14 illustrates the "rounded knee" phenomenon and the suggested method for identifying the start of the initial deflection.

In both the "coarse" and "fine" files, no wandering baseline seemed to be present at the locations over which crackles were detected. Rounded knees seemed to be present in 1 out of 3 of the detected crackles in the "fine" file, and 1 out of the 8 detected crackles for the "coarse" file. Accounting for the rounded knees would *decrease* the IDW and 2CD values for those crackles in which rounded knees were present, thereby decreasing the average IDW and 2CD values for their respective files. However, these modifications do not change the average IDW and 2CD values significantly; and the pitch-quality classifications do not change. Therefore, the first explanation probably does not account for the discrepancy between the file labels and pitch-quality classifications.

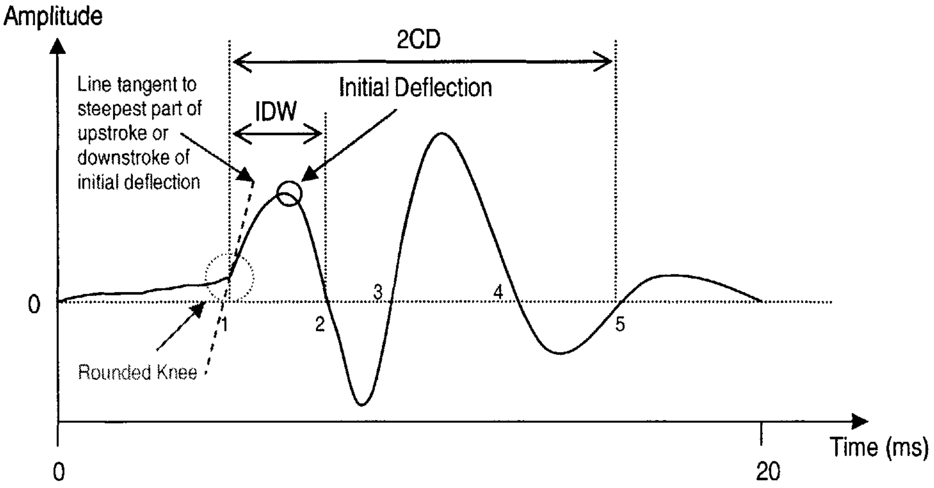

Amplitude
2CD
Line tangent to steepest part of upstroke or downstroke of initial deflection
IDW
Initial Deflection
0
1
2
3
4
5
Rounded Knee
0
20
Time (ms)

Figure 14. Illustration of "rounded knee" phenomenon in the crackle waveform and a method for identifying the starting point of the initial deflection in such a case.

The second explanation is possible because in most (but not all) cases, crackles have at least two complete cycles in their respective waveforms [15]. In the case that a crackle does ***not*** have at least two complete cycles, the 2CD parameter is not applicable. This explanation likely accounts for the unusually high average IDW and 2CD values for the "fine" file.

Figure 15 shows the region in the $x[n]$ signal over which the three crackles for the "fine" file were detected and points out the individual crackle occurrences. Figure 16 shows the plotted 2CD regions of the detected crackles. Notice that for each crackle, the waveform over the 2CD region has a rapid, high-intensity deflection at its center that spans no more than one complete cycle; the rest of the waveform (at its periphery) looks as if it is part of the background vesicular signal (observable in Figure 15 as a low-intensity, slow-moving waveform over which the three detected crackles are superimposed). The inclusion of vesicular sounds when computing the IDW and 2CD seems to attribute to the high average IDW and 2CD values for the "fine" file.

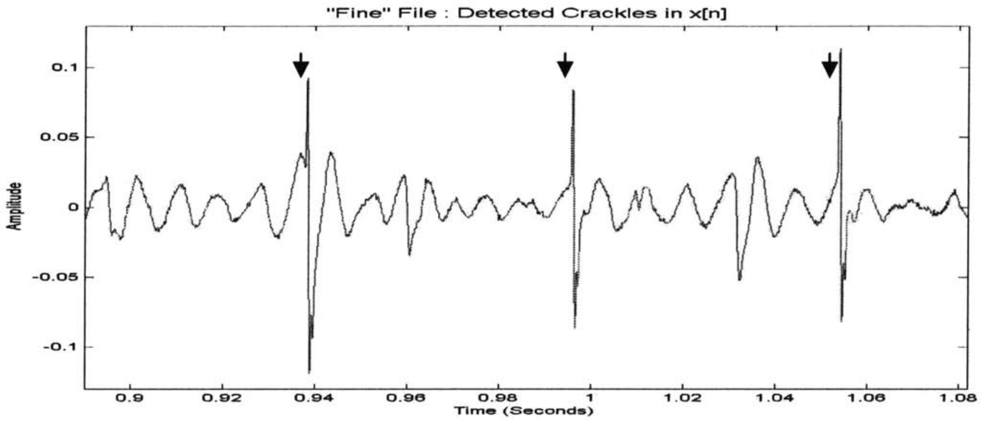


Figure 15. Plot of the region in $x[n]$ of the "fine" file over which crackles were detected: the (three) crackles are pointed out with arrows.

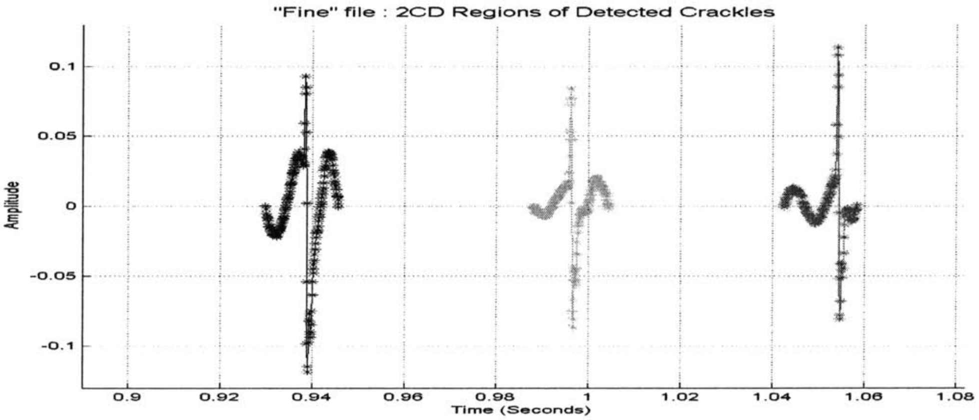
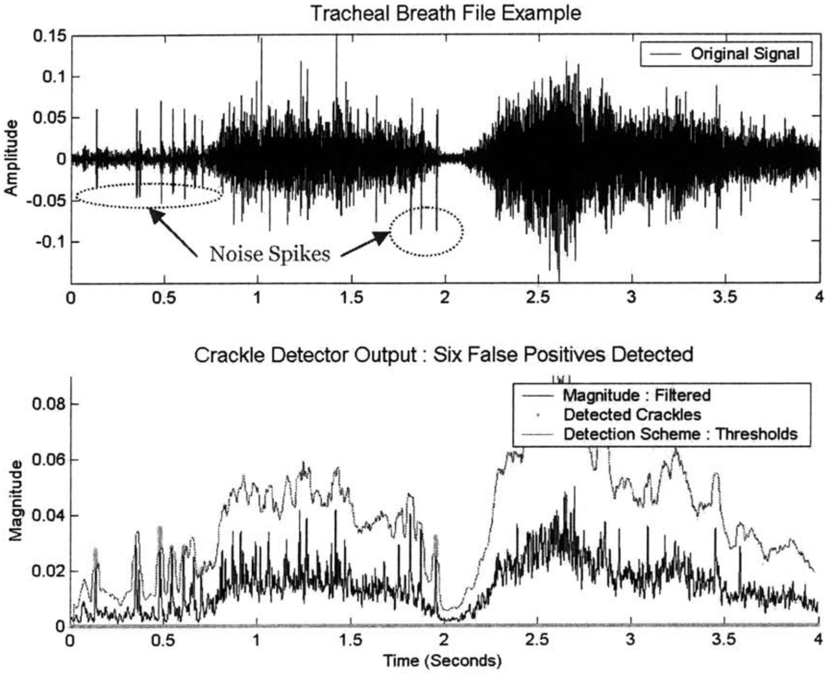


Figure 16. Plot of the 2CD regions of the (three) detected crackles in the "fine" file.

5.8.2 Specificity Analysis

The tests on the tracheal breath and wheeze sound files indicate that our crackle detector cannot distinguish between the high-intensity spikes from crackles and those from noise. Consider the tracheal breath example and corresponding crackle detector output in Figure 18. Notice the noise spikes in the signal (caused by movements of the stethoscope over the subject's skin during the recording process), six of which are falsely detected as crackles.


Tracheal Breath File Example

Crackle Detector Output : Six False Positives Detected

Figure 18. Tracheal breath file example: (top) time-domain plot of the tracheal breath signal with noise spikes pointed out; (bottom) six false positives (dotted cyan line) detected in $e[n]$ (black solid line) using the variable threshold (red solid line) crackle detection scheme.

Figure 19 shows the waveform morphology (over the 2CD region) of the 4th detected false positive (in time). This particular waveform has an IDW of 3.1 ms and a 2CD of 12.3 ms, both of which fall within the respective IDW and 2CD ranges for typical crackles. Notice, however, that the waveform of the 4th false positive has high-frequency, low-intensity noise riding on top of its waveform, whereas a typical crackle (see Figure 12) has a smooth waveform. This suggests that distinguishing between crackles and random noise spikes may be possible.

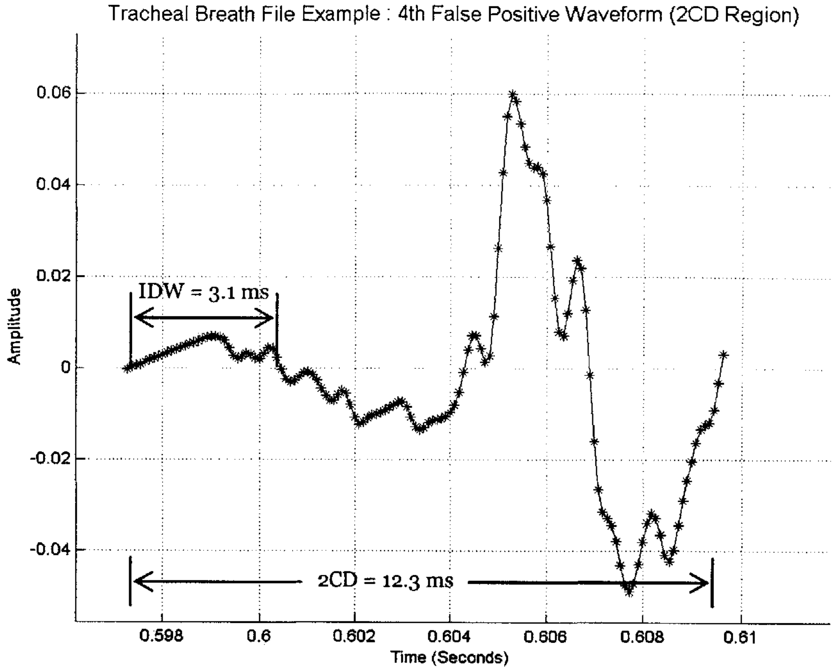


Figure 19. Tracheal breath file example: Waveform morphology of 4th false positive (in time) over the 2CD region.

The adaptive threshold technique that we use for crackle detection is effective in detecting crackles of all sizes (e.g., small and large in magnitude) as long as they are isolated (i.e., more than ~20 ms away) from each other. It is not very effective in detecting crackles that lie in close proximity to (i.e., within ~20 ms of) each other. Recall that the detection algorithm compares the magnitude of each crackle to a threshold, which is a function of the local mean. The local mean for each crackle is computed using the signal energy within a 40 ms window centered at its location. If neighboring crackles fall within ~20 ms of any particular crackle, the local mean for that crackle can rise considerably, such that the threshold exceeds its waveform peak, and the crackle is not detected.

For the threshold test in our crackle detector, changing the window size over which the local means are computed and/or the constant by which the local mean is multiplied, in order to compute the threshold, will change the performance of the detector. Selecting the optimal combination of window size and multiplicative constant requires a trade-off between improving the sensitivity and specificity of the detector or, equivalently, a trade-off between decreasing the number of misses and false positives.

We cannot compare performance figures for our crackle detection algorithm to those proposed by [25] and [12] because we do not report a sensitivity or specificity rate. In terms of sensitivity, [12] is better at detecting both fine and coarse crackles than [25]. Specificity figures were not reported for any of these detection methods.

The drawback of the ST-NST algorithm (proposed by [25]) is that it requires updating the parameters of the filter (e.g., $F(\cdot)$ and the coefficients of the autoregressive model) for each input signal according to its characteristics (e.g., whether the crackles are fine or coarse). On the other hand, the WTST-NST algorithm (proposed by [12]) and our crackle detector do not require users to input information about signal characteristics prior to processing, and therefore provides greater convenience.

The WTST-NST algorithm "peels" the lung sounds into layers by using the multi-scale wavelet decomposition, and an iterative decomposition-reconstruction scheme to separate the crackles from the vesicular sounds. Their method incorporates both time and frequency information, whereas the ST-NST and our method uses only time domain information.

Neither [25] nor [12] propose methods for automatically characterizing crackles according to their pitch (e.g., via the IDW and 2CD parameters) or timing (e.g., early/mid/late respiratory phase).

5.9 Future Work

Our *crackles analysis* system needs to be tested on lung sounds data for which the location of individual crackles, and their respective pitches and timings are labeled (e.g., by a physician). This would allow objective assessments to be made about the accuracy of the system.

Issues that need to be addressed in order to improve the accuracy of the system include distinguishing between crackles and noise spikes, and increasing sensitivity to crackles that occur in close proximity to each other. Once the desired accuracy is achieved, further improvements can be made, including increasing computational speed and developing a user interface that is appropriate for the intended application.

Chapter 6

Wheezes Analysis

In this chapter we discuss a scheme for automatically detecting, quantifying, and qualifying wheezes. Given an input lung sound signal the system outputs the number of wheezes detected in a specified time region; for each wheeze, it identifies its location in time, duration, fundamental frequency, and energy (as a fraction of the total energy in the signal); and, finally, it computes the percentage of the signal duration occupied by wheezes.

6.1 Introduction[18]

Wheezes form a class of adventitious sounds that are characterized by their sustained durations and well-defined pitches. They can be heard over the mouth and chest wall, and can be distinguished from typically noisy normal breath sounds by their continuous, "musical quality." This musical quality translates to an acoustic waveform that is pseudo-periodic (and most often sinusoidal) in nature, with a distinct fundamental frequency and a few (if any) harmonics [5]. The pitch of a wheeze is defined by the fundamental frequency, a value which generally lies between 100 and 1600 Hz. Figure 1 compares a normal lung sound signal to a wheeze over a 50 milliseconds time-frame. Notice the randomness of the normal lung sound signal in comparison to the wheeze signal, which is clearly a sinusoid with a distinct (fundamental) frequency.

The American Thoracic Society defines the minimum duration of a wheeze to be 250 milliseconds [5].

A wheeze is produced by a mechanism similar to that of a toy trumpet. The frequency (or pitch) produced by a toy trumpet is determined by the mass and elasticity of the reed. For wheezes, the fundamental frequency is produced when a narrow constriction in the airway

[18] Most of the material on wheezes in this section is from [11].

causes opposite walls of the airway to almost come into contact with each other and oscillate as air passes.

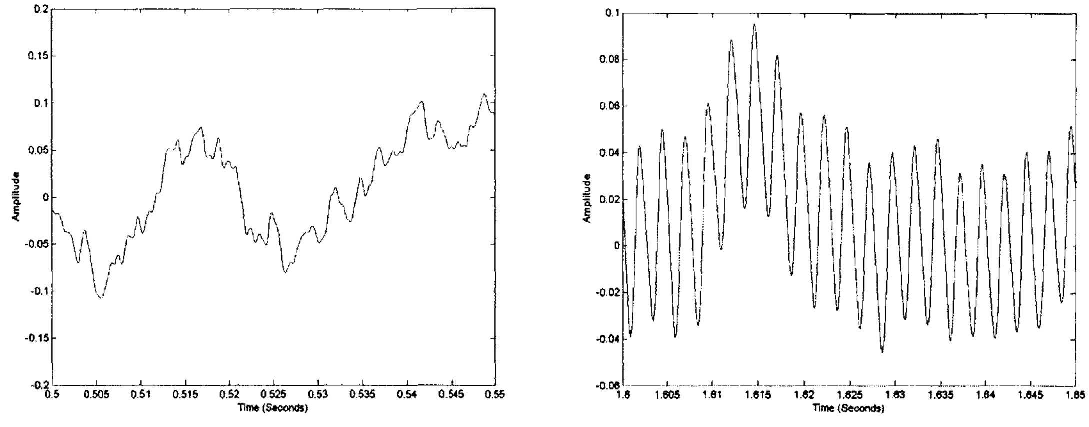


Figure 1. Comparison of a normal breath signal (left) and a wheeze signal (right) over a 50 ms time-frame.

A wheeze can be classified as being either fixed monophonic, random monophonic, sequential inspiratory, or expiratory polyphonic. Fixed monophonic wheezes are composed of a single, constant pitch, and are usually indicative of an incomplete occlusion of the bronchus (e.g., by a tumor). A random monophonic wheeze typically occurs when there is widespread airflow obstruction (e.g., with asthma). They occur randomly across respiratory cycles, can be found in both inspiration and expiration, and are usually a result of bronchial spasm or swelling of the mucous membrane. A sequential inspiratory wheeze is typically generated when peripheral airways open and oscillate, late in inspiration. They are characterized as sequences of short, monophonic wheezes, each with a different pitch and sound intensity. Pulmonary diseases associated with sequential inspiratory wheezes include fibrosing alveolitis, asbestosis, and other diffuse interstitial pulmonary diseases. Expiratory polyphonic wheezes are produced by the passage of air through many obstructed bronchial airways simultaneously, creating "several harmonically unrelated musical sounds," and are associated with the chronic obstructive bronchitis.

6.2 Motivation

A computerized system that automatically detects wheezes using only acoustic data may be useful for conveniently and inexpensively monitoring the progression of chronic pulmonary diseases such as asthma in both a clinical and non-clinical environment. Monitoring the progression of diseases can lead to better treatment of the disease. Consider, for example, *chronic asthma*, a disease that affects 14 to 17 million people in the United States and often goes undiagnosed [32]. Asthma is classified into four categories that depend on the frequency (of occurrence) and duration of symptoms: (1) mild intermittent asthma, (2) mild persistent asthma, (3) persistent asthma, and (4) severe persistent asthma. Different treatments should be prescribed depending on the asthma class. Furthermore, medication needs to be adjusted in response to changes in the classification in order to eradicate the disease and restore normal lung function.

Asthma can be diagnosed using a technique called spirometry. The severity of asthma can be determined by the FEV1 (Forced Expiratory Volume in the 1st second) value, where lower FEV1 values indicate greater severity. While FEV1 has already been established as an accurate method for assessing lung function, continuous lung function monitoring may be impossible using this technique due to the discomfort associated with forced breathing.

FEV1 has been correlated with spectral characteristics of lung sounds [29][19], which suggests that monitoring lung function may be possible using lung sounds (e.g., wheezes). In applications requiring continuous lung function monitoring, an acoustical approach offers the convenience of allowing patients to breathe naturally.

6.3 System Design

Figure 2 is a block diagram of our *wheezes analysis* system. The user inputs to the system the filename containing the lung sounds, along with the specifications for the left endpoint and right endpoint cutoff times that indicate to the system the region over which the signal should be analyzed. The *wheeze quantifier* and *wheeze qualifier* stages output measurements that characterize the wheezes detected by the *wheeze detector* stage.

[19] More on their work is discussed in the "Background Literature/Related Work" section of this chapter.

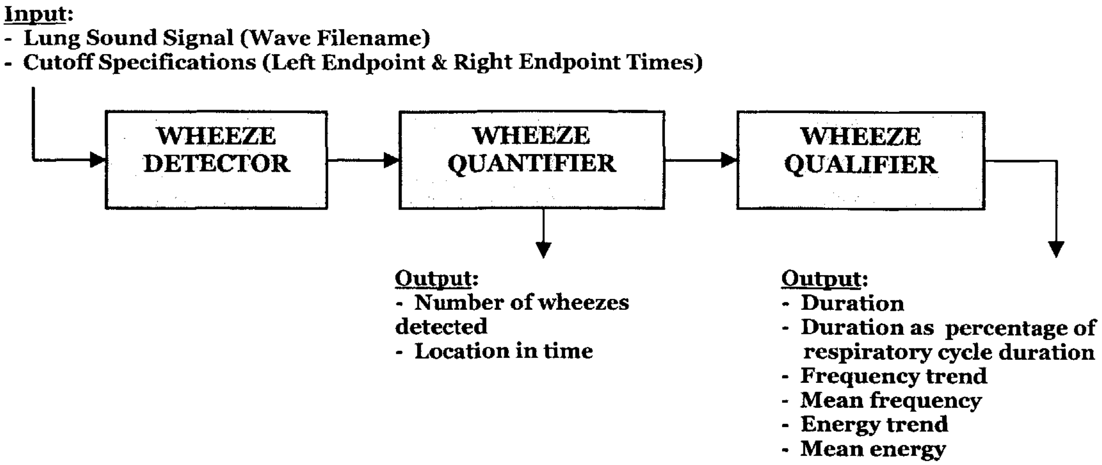

Input:
- Lung Sound Signal (Wave Filename)
- Cutoff Specifications (Left Endpoint & Right Endpoint Times)
WHEEZE DETECTOR
WHEEZE QUANTIFIER
WHEEZE QUALIFIER
Output:
- Number of wheezes detected
- Location in time
Output:
- Duration
- Duration as percentage of respiratory cycle duration
- Frequency trend
- Mean frequency
- Energy trend
- Mean energy

Figure 2. Block diagram of the *wheezes analysis* system.

We employ a time-dependent frequency analysis of the signal to detect wheezes. This method looks for a **prominent** and **isolated** peak in the estimated power spectral density[20] of the signal by applying a series of threshold tests to all detected peaks in the power spectrum that lie within a specific frequency range.

6.4 Background Literature/Related Work

According to [29], studies have shown that statistically significant correlations exist between the percentage of wheeze observed in sound recordings and degree of airway obstruction. In their work, they were able to build a classifier that assigns lung-function labels to tracheal breath sounds. The labels consisted of a discrete set of FEV1 values. They studied 10 asthmatic patients, and simultaneously measured FEV1 values and tracheal breath sounds at various stages during the progression of an asthma attack. They used 5 to 15 examples of respiratory cycles per FEV1 value from each of 16 sessions. The number of FEV1 measurements made per session ranged from 9 to 21. Their classification scheme uses the *k nearest neighbors* method with k=1. Feature vectors for each data sample are constructed by the following method: (1) compute the power spectrum of the tracheal signal using frequency bin-widths of 59 Hz, (2) extract only the [100 → 1300] frequency range from the power spectrum, (3) normalize the spectrum such that it sums to 1, and (4) apply a "Box-

[20] Refer to [26].

cox" transformation, where the ith spectral component (f_i) is mapped to a new value (x_i) using

$$x_i \leftarrow \frac{f_i^{0.1} - 1}{0.1}.$$

When testing their classifier, the label outputted by the classifier for a test example was considered to be "correct" when the corresponding FEV1 value for that label was within 4% of the true FEV1 value. They reported classification a accuracy rate of $77 \pm 8\,\%$ across all 16 sessions.

An approach for wheeze detection using tracheal breath sounds is proposed by [16]. Their study used tracheal signals that were recorded during forced expiration, and for airflow rates of 0 to 1.2 liters/second (l/s); a total of 60 recordings were collected from 15 normal subjects and 16 asthmatic patients. Their wheeze detection approach (referred to as "Local Adaptive Wheezes Detection (LAWD)") splits the signal into windows of size 52.1 ms, overlapped by 50%. They compute a normalized power spectrum of the signal and detect peaks in the power spectrum with amplitudes that fall between empirically-determined lower and upper boundaries. Each peak is considered to correspond to an actual wheeze if it passes a *scoring test*. The scoring test is rule-based and checks whether the spectral characteristics surrounding each peak satisfy empirically-determined constraints for wheezes. A "grouping algorithm" is used to group the detected 51.2 ms windows of wheezes together according to time, frequency, and amplitude constraints – each resulting group is considered to be a single wheeze. They also compute the following wheeze parameters to characterize the detected wheezes: number of wheezes, mean frequency of wheeze with the highest power peak, mean frequency of wheeze with highest mean power, averaged mean frequency, and percentile of signal occupied by wheezes. The sensitivity rates of their wheeze detection algorithm for signals corresponding to airflow rates of 1.2-0.4 l/s, 0.4-0.2 l/s, and 0.2-0 l/s were 100%, 87.8%, and 71%, respectively.

6.5 Method

To develop our system, we analyzed a total of 6 wheeze recordings, collected from public lung sound repositories on the World Wide Web [40]. The wheezes in each file were not labeled and characterized by the websites from which they were downloaded – hence, we identified the wheezes according to our own best judgments. In all of the files, the wheezes could be heard as distinct musical sounds, and visualized as horizontal striations in the spectrogram.

6.6 Implementation

In this section we describe the implementation details for each processing block in Figure 2.

6.6.1 Wheeze Detector Stage

Unlike crackles, which are characterized by large and rapid amplitude deviations in the *time-domain* signal, wheezes are characterized by large and rapid amplitude deviations in the *frequency-domain.* Because the energy of a wheeze signal is concentrated in the fundamental frequency component, we would expect to find a **prominent** and **isolated** peak in the frequency spectrum corresponding to this fundamental frequency. Our primary goal in developing the *wheeze detector* was to create a scheme for identifying the spectral peaks that correspond to wheeze occurrences. The structure of the detection algorithm is depicted in Figure 3.

Approach

To illustrate the wheeze detection process, assume that the input to the system is the lung sound signal containing expiratory wheezes, as shown in Figure 4. This signal contains approximately four and a half respiratory cycles.

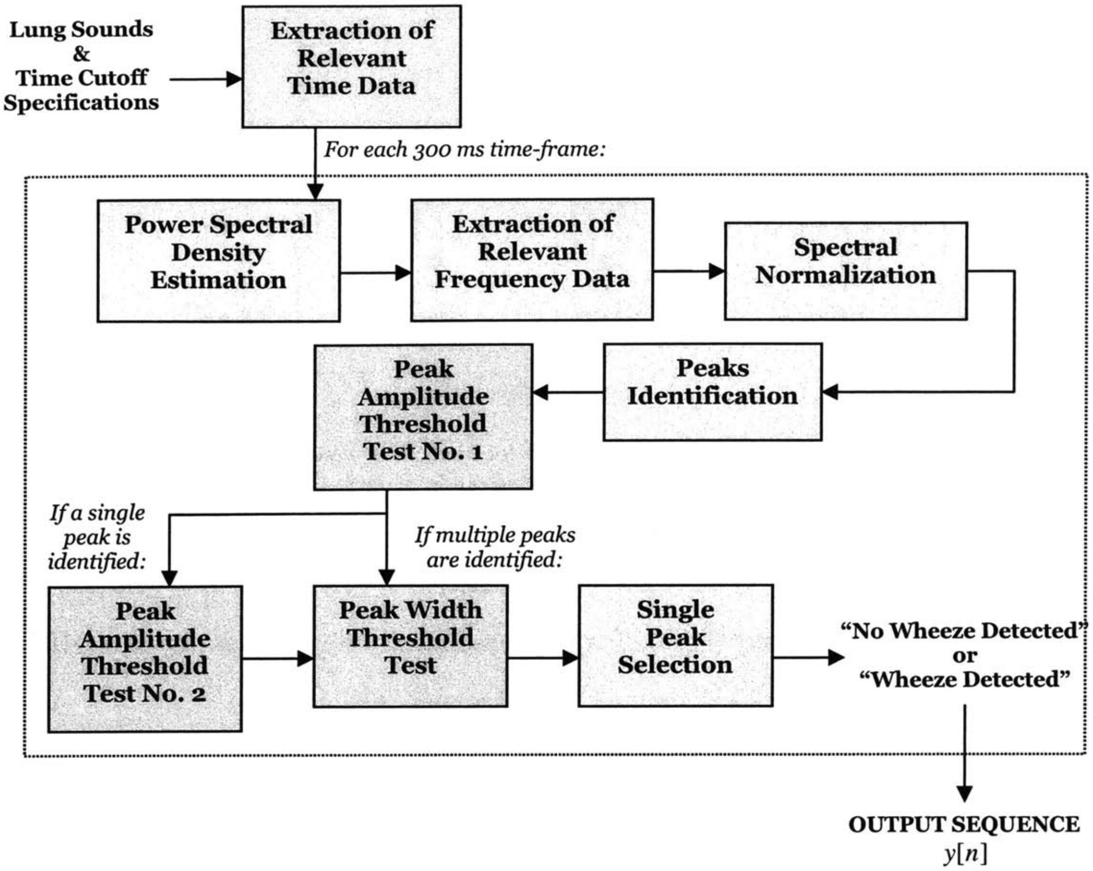

Lung Sounds & Time Cutoff Specifications
Extraction of Relevant Time Data
For each 300 ms time-frame:
Power Spectral Density Estimation
Extraction of Relevant Frequency Data
Spectral Normalization
Peaks Identification
Peak Amplitude Threshold Test No. 1
If a single peak is identified:
If multiple peaks are identified:
Peak Amplitude Threshold Test No. 2
Peak Width Threshold Test
Single Peak Selection
"No Wheeze Detected" or "Wheeze Detected"
OUTPUT SEQUENCE
y[n]

Figure 3. Block diagram of the *wheeze detector* stage.

In the *extraction of relevant time data* stage, the region of the signal specified by the input start and end cutoff times is extracted. Suppose the user inputs start and end times of 0 and 10 seconds, respectively. The system will analyze the corresponding ten-second duration of the signal (same as in Figure 4) in 300 milliseconds (ms) time-frames, separated by 150 ms shift-sizes. We chose to use a 300 ms time-frame because it approximates the minimum wheeze duration of 250 ms (mentioned earlier), and yielded more accurate detection results for our test data set than a 250 ms time-frame. Each time-frame is classified as either "No Wheeze Detected" or "Wheeze Detected," depending on whether wheezing was detected in the signal over that time-frame.

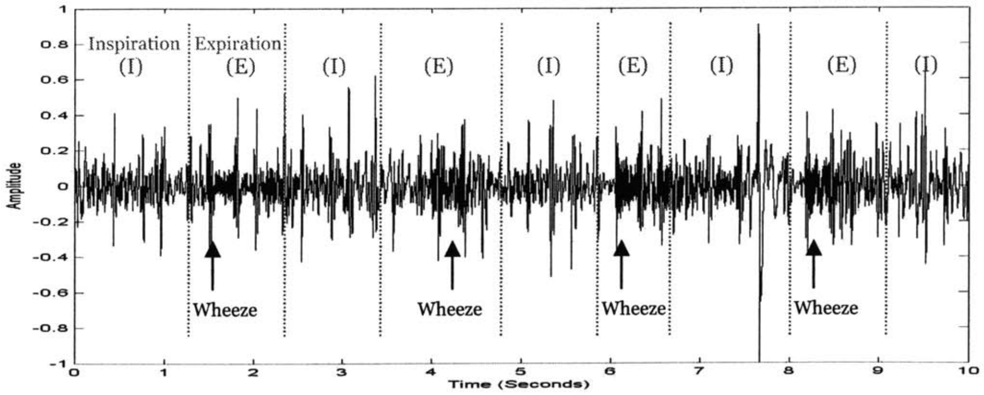


Figure 4. Time-domain plot of input signal. Approximate locations of respiratory phase transitions are indicated by vertical (dotted) lines; wheeze epochs are pointed out by arrows.

Let the signal in the ith time-frame be denoted by $x_i[n]$. In the *power spectral density estimation* stage, the power spectral density (PSD) of $x_i[n]$ is estimated using the *auto-regressive (AR) time series* model of the signal[21],

$$x_i[n] = -\sum_{k=1}^{p} a_k x_i[n-k] + \varepsilon[n],$$

where p is the model order, a_k is the estimated kth AR parameter, and $\varepsilon[n]$ is a white-noise random process with zero mean and variance σ^2. The resulting PSD of $x_i[n]$ is

$$P_{x_i}(w) = \frac{\sigma^2}{\left|1 + \sum_{k=1}^{p} a_k e^{-jwk}\right|^2}.$$

The amplitude of the PSD at frequency w (i.e., $P_{x_i}(w)$) corresponds to the power in the signal at that frequency. The power spectral density can also be expressed as

$$P_{x_i}(w) = \lim_{M\to\infty} E\left[\frac{1}{2M+1}\left|\sum_{n=-M}^{M} x[n]e^{-jwn}\right|^2\right],$$

which can be estimated by the squared absolute value of the discrete Fourier transform of $x_i[n]$, scaled by a constant.

For the wheeze detector, we found that an auto-regressive PSD model of **order 100** was able to trace the peaks in the frequency spectrum accurately. We compute the parametric spectral

[21] A detailed discussion of PSD estimation can be found in Chapter 2 of [26].

estimate instead of, for example, the discrete Fourier transform because the spectral estimate outlines the general shape of the Fourier transform (FT), yielding a *smoother* spectral representation. This is important later in the detection process, when peaks in the spectrum are assumed to correspond directly to wheeze amplitude deviations. The difference between the Fourier transform and estimated power spectral density is illustrated in Figure 5, where the two spectral plots of the signal in Figure 4 are graphed. It is clear from this example that the PSD curve is significantly smoother than the FT curve, while still resolving the prominent peak in energy, centered at a little above 400 Hz.

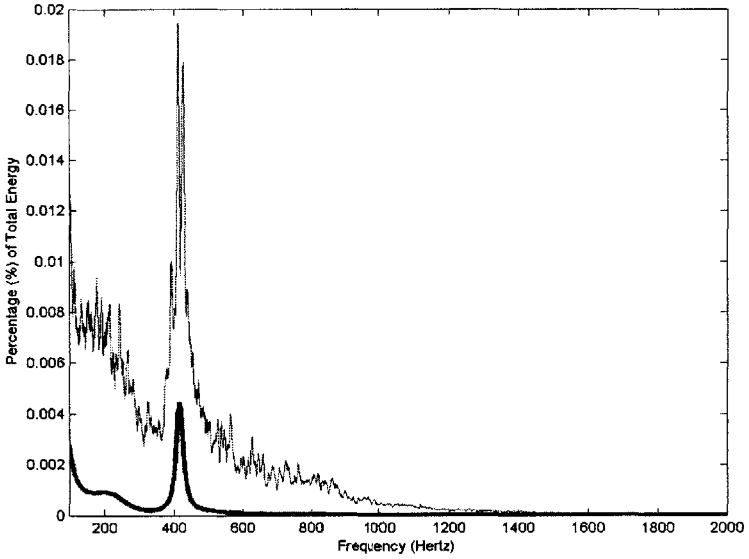


Figure 5. Plots of two spectral representations of a signal: (1) Fourier transform (thin red line), and (2) power spectral density using the autoregressive model with an order of 100 (bold blue line).

The y-axis scale in Figure 5 is in normalized units, given by the percent of total energy. Notice that at each frequency, the amplitude of the Fourier transform is relatively larger than that of the power spectral density. This relative difference in amplitude is irrelevant for our wheeze detection procedure, since we are only interested in the relative shape of the spectrum.

Let us now analyze the difference between the PSD of a time-segment containing a wheeze and a time-segment without a wheeze. Figure 6 shows the power spectral estimates of 300 ms non-wheeze and wheeze time-segments from the signal in Figure 4 starting at 0.9 and 1.65 seconds, respectively.

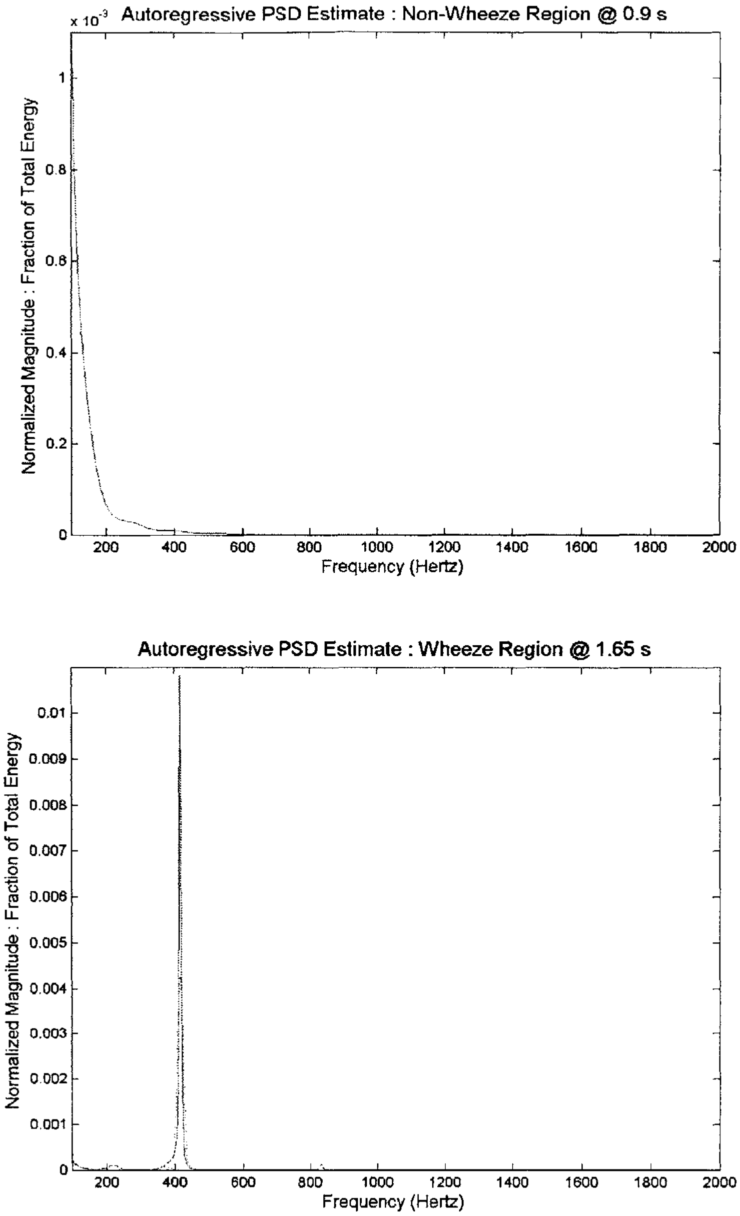


Figure 6. Power spectral density estimates of: (top) non-wheeze signal, and (bottom) wheeze signal.

We will focus on only the [100 → 2,000] Hz frequency range in this analysis, since that is the range over which wheeze frequencies are defined. Starting from 100 Hz and moving to higher frequencies, it is apparent that the PSD of the non-wheeze signal rapidly falls off in amplitude and exhibits no distinct peaks. On the other hand, the PSD of the wheeze signal has a prominent (i.e., high-amplitude and narrowband), isolated peak at a little above 400 Hz.

These differences are used by the wheeze detector to distinguish between the "Wheeze detected" and "No wheeze detected" signal classes – more specifically, a wheeze is detected when there exists a peak in the PSD whose amplitude exceeds an amplitude-threshold and whose width falls below a width-threshold.

Since we are interested only in the [100 → 2,000] Hz range, the next stage in the wheeze detector, *extraction of relevant frequency data,* extracts only the spectral components that lie within that range. The *spectral normalization* stage normalizes the spectral amplitudes by the total energy of the signal in the new frequency range. In the *peaks identification* stage, all "peaks" (defined as local maxima) in the spectral signal are identified.

Next, the detected peaks each get passed to a sequence of tests that are summarized below. A peak must pass all relevant tests in order to be considered a wheeze-peak.

- *Peak Amplitude Threshold Test No. 1* – The amplitude of the spectral peak must exceed 0.04, where the amplitude is in units of fraction of the total energy in the [100 → 2000] Hz range.

- *Peak Amplitude Threshold Test No. 2* – If it is the only detected peak, the amplitude of the spectral peak must exceed 0.05.

- *Peak Width Threshold Test* – The sum of the lower and upper "widths" of the peak must fall below 160 Hz. The lower and upper widths are defined to be the frequency spans bounded by $[f_{LOW} \to f_{PEAK\_LOCATION}]$ and $[f_{PEAK\_LOCATION} \to f_{HIGH}]$, respectively. The frequency parameters are defined in Table 1.

At the *single peak selection* stage, if no peaks remain, then the ith time-frame is classified as "No Wheeze Detected," and the corresponding output $y[n=i]$ is set to '-1.' If at least one peak remains, then the ith time-frame is classified as "Wheeze Detected," and the corresponding output $y[n=i]$ is set to '+1.' The peak with the largest amplitude is selected to correspond to the fundamental frequency of the wheeze.

Parameter	Definition
$f_{PEAK\_LOCATION}$	The frequency at which the peak (i.e., local maximum) occurs.
f_{LOW}	The frequency at which the peak falls off to 5 % of its maximum height at the lower frequency end.
f_{HIGH}	The frequency at which the peak falls off to 5 % of its maximum height at the upper frequency end.

Table 1. Definition of frequency parameters used in *width threshold test.*

The output sequence $y[n]$ is overlaid on the input signal in Figure 7.

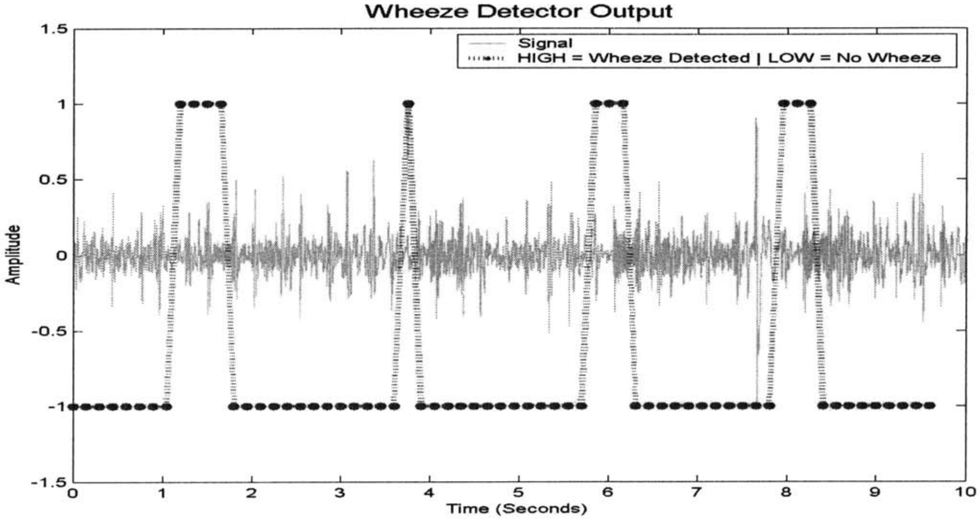


Figure 7. Wheeze detector output $y[n]$ (dotted red line) overlaid on input signal (cyan line).

6.6.2 Wheeze Quantifier Stage

The *wheeze quantifier* stage outputs the total number of wheezes detected in the signal and their onset times.

A single wheeze epoch is defined as a continuous sequence of '+1' values in $y[n]$. Therefore, the quantifier outputs a wheeze count of 4 for this example. For a given wheeze, if its corresponding sequence of '+1' values begins at the ith index of $y[n]$, then the onset time is given by: (i-1)*(shift size)=(i-1)*(150 ms). Hence, for this example, the computed onset time for the first wheeze (for which i=9) is: (9-1)*(150 ms)=1,200 ms=1.2 seconds.

6.6.3 Wheeze Qualifier Stage

The *wheeze qualifier* stage outputs the duration, mean frequency, frequency trend, energy, and energy trend of each wheeze. It also outputs the duration of the total wheezing present in the signal over the entire region of interest as a percentage the duration of that region.

If a wheeze corresponds to a sequence of '+1' values that starts and ends at ith and jth indices of $y[n]$, then its duration is given by: (j-i)*(shift size)+(window size)=(j-i)*(150 ms)+(300 ms). Therefore, for this example, the computed duration for the first wheeze (for which i=9 and j=12) is: (12-9)*(150 ms)+(300 ms)=750 ms=0.75 seconds.

The frequency trend for a wheeze corresponds to the progression of the fundamental frequencies detected over the 300 ms time-frames that span its duration. The mean frequency of a wheeze is defined as the average frequency in the frequency trend. For this example, the computed frequency trend and mean frequency for the first wheeze are [413, 393, 393, 417] and 404, respectively.

We define the energy of a wheeze within a single 300 ms time-frame to be the energy within the spectral peak (corresponding to the fundamental frequency) over the $[f_{LOW} \to f_{HIGH}]$ frequency range, expressed as a percentage of the total energy in the signal over the $[0 \to f_s / 2]$ Hz frequency range, where f_s is the sampling frequency. The energy trend of a wheeze is the progression of the energies computed for each 300 ms time-frame that span the entire duration of the wheeze. The mean energy is the average energy in the energy trend. For this example, the computed energy trend and mean energy for the first wheeze are [3.5, 7.2, 9.7, 2.4] and 5.7, respectively.

The frequency trends, mean frequency, energy trends, and mean energy for this example are graphed in Figure 8, along with the plot of vector $y[n]$ overlaid on the original time signal, and the spectrogram of the signal.

The duration of wheezing as a percentage of the signal duration is given by: ((sum of durations of all the wheezes)/(duration of the signal))*(100 %). The computed value for this example is 22.5% – this means that wheezes are present over roughly a quarter of the signal duration.

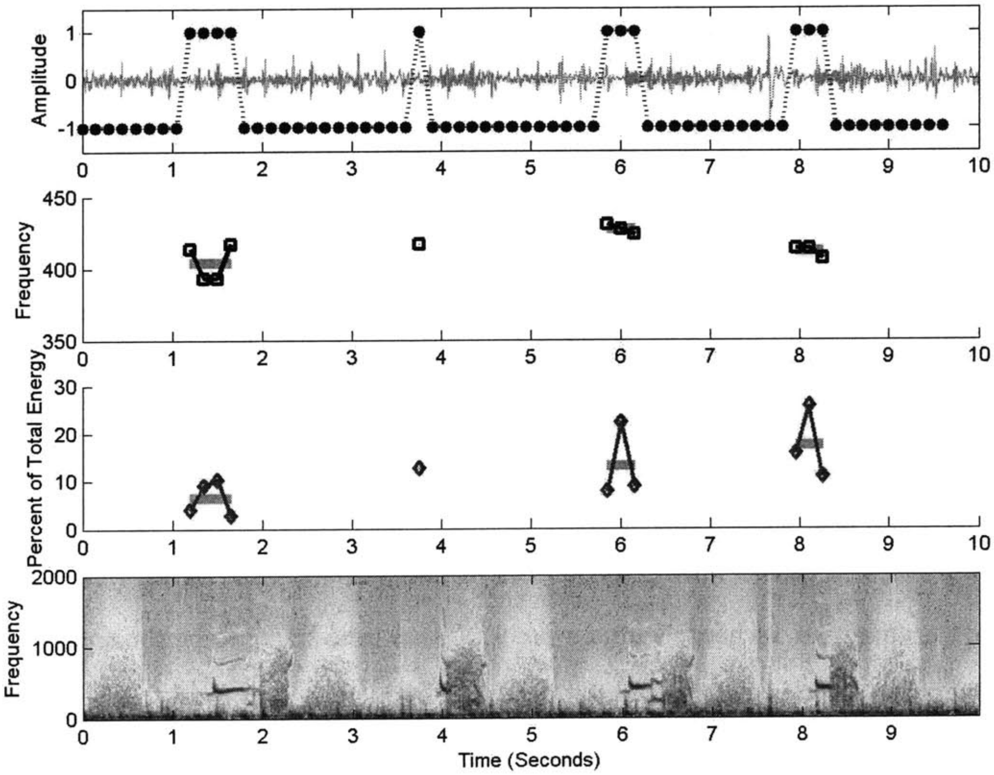


Figure 8. From top to bottom: (1) vector $y[n]$ (dotted blue line) overlaid on original time-signal (green line); (2) frequency trends of detected wheezes (black squares) overlaid on mean frequencies (solid green lines); (3) energy trends of detected wheezes (red diamonds) overlaid on mean energies (solid green lines); and (4) spectrogram of original signal.

6.7 Performance

To validate the performance of our wheeze detector, we tested whether wheezes are detected in 6 wheeze files, 13 crackle files, and 5 normal tracheal breath files (obtained from 5 subjects). We used the first 4 seconds of each file in our analyses.

For the wheeze files, we expected wheezes to be detected wherever distinct musical sounds were heard and horizontal striations in the spectrogram were visible. Because a "gold-standard" was unavailable to which our wheeze detector outputs could be compared, we cannot report objective accuracy rates. However, because we do know that each wheeze file contains at least one wheeze, we will report that wheezes were found to be present in 5 out of the 6 wheeze files. For the files in which wheezes were detected, wheezes were generally detected over regions in time where we could distinctly hear musical sounds and observe horizontal striations in the spectrogram. The wheeze file in which wheezes were not detected contained wheeze-like sounds that were barely audible.

For the crackle and tracheal breath files, we expected *no* wheezes to be detected by our wheeze detector. No wheeze was detected in 11 out of the 13 crackle files (yielding an 84.6% specificity rate), and 5 out of the 5 tracheal breath files (yielding a 100% specificity rate). For the two crackle files in which wheezes were detected, the falsely classified "wheeze" regions contained narrow-band noise. Examples of the wheeze detector output for a tracheal breath file and a crackle file are shown in Figures 9 and 10, respectively. For each figure, the top graph contains plots of the original (time-domain) signal and the detector output $y[n]$; and the bottom graph contains the spectrogram of the signal. Notice for both examples that samples of $y[n]$ are low (i.e., have the value '-1' instead of '+1') throughout the duration of the signal, which means that no wheezes were detected.

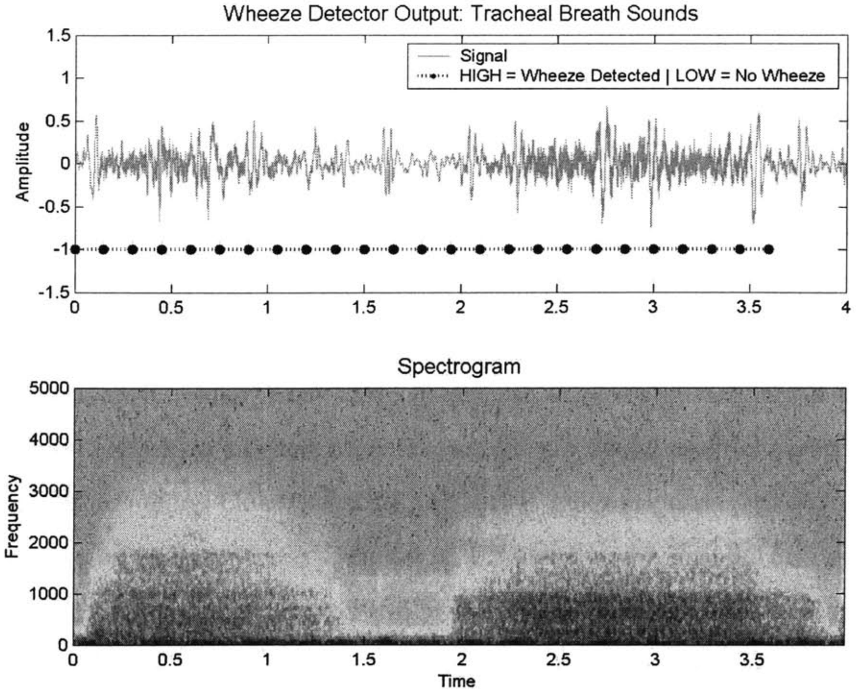


Figure 9. Wheeze detector output for a tracheal breath file: (top) wheeze detector output $y[n]$ (dotted blue) overlaid on original time-signal (solid green); (bottom) spectrogram of signal.

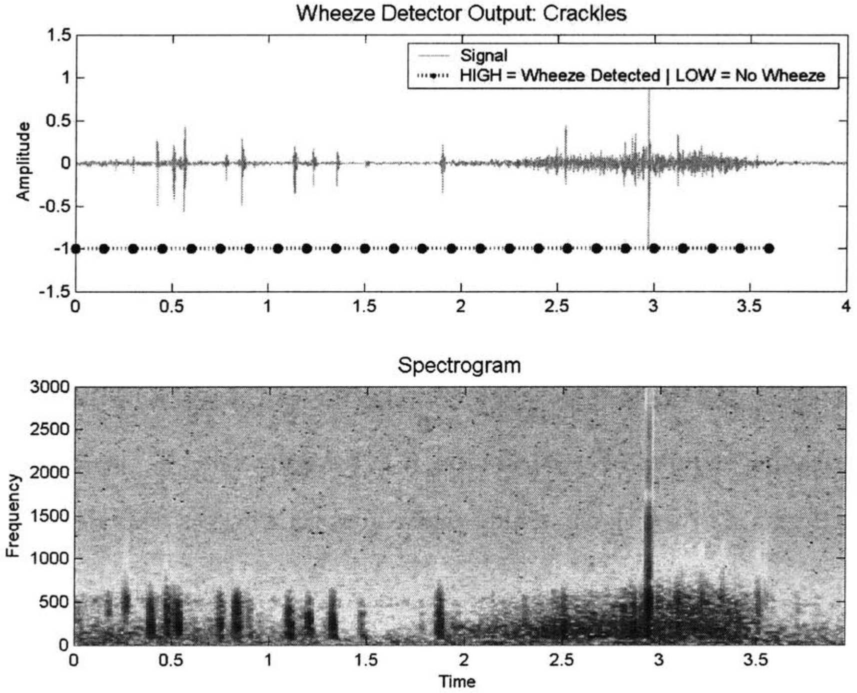


Figure 10. Wheeze detector output for a crackle file: (top) wheeze detector output $y[n]$ (dotted blue) overlaid on original time-signal (solid green); (bottom) spectrogram of signal.

6.8 Discussion

The results for our wheeze detector predict true-negative (or specificity) rates on crackles and tracheal breath sounds of 84.6% and 100%, respectively. Given that the false-positive classifications in the crackles files were due to narrow frequency-band noise in the signal, these results suggest that it is possible to distinguish wheezes from other respiratory events (e.g., crackles and tracheal breath sounds) with high accuracy.

The wheeze detector proposed by [16] is very similar to our wheeze detector. Both detectors use tracheal breath sounds as input, detect wheezes by identifying isolated and prominent peaks in the frequency domain, and employ a rule-based decision scheme. The difference between their approach and ours is in the details of the overall approach. Their approach uses a smaller window size over which wheezes are detected (51.2 ms) than ours (300 ms), does not exclude from consideration frequencies outside the [100 → 2000] Hz range (as ours does), and uses different rules to identify peaks in the power spectrum that correspond to actual wheezes. Another (more important) difference is that their detection method allows multiple wheeze peaks (e.g., those corresponding to the several harmonically unrelated pitches of a polyphonic wheeze) to be detected for a particular instant in time, whereas ours only allows for a single peak (e.g., that corresponding to the fundamental frequency of a monophonic wheeze) to be detected. Also, they employ a "grouping algorithm" which has a function similar to our method of treating a sequence of '+1' values in the $y[n]$ signal as a single wheeze. Their grouping algorithm is a bit more complex because it takes into account not only time information (e.g., whether the locations in time of the windows in which wheezes were detected are close enough), but also frequency information (e.g., whether the frequencies of the wheezes between windows are close enough).

The work proposed by [29] on classifying tracheal breath sounds according to discrete FEV1 values had a classification accuracy rate of $77 \pm 8\,\%$. One possible problem with their classification scheme is that its uses the coefficients of the tracheal signal's power spectrum as its feature vector. Consider the case where two wheezes have different fundamental frequencies, but correspond to the same FEV1 value – the power spectrum will look very different for the two wheezes on a feature-by-feature basis, since (due to the different fundamental frequencies) their wheeze peaks would not be aligned with each other. Understanding that these two (very different) feature vector examples should be assigned the same class label (or FEV1 value) would probably be too difficult for a single classifier. If the reported classification accuracy rate was higher, their work might obviate the need for a wheeze detector, quantifier, and qualifier, since the primary purpose of analyzing wheezes (via tracheal breath sounds data) is to understand lung function, and their method would directly map tracheal breath sounds data to lung function. However, given that the accuracy rate is relatively low, a better approach to understanding lung function might be to instead: (1) correlate wheeze parameters to lung function; (2) monitor wheeze parameters using a wheeze detector, quantifier, and qualifier; and (3) based on the progression of the wheeze parameters, make decisions about the presence of airway obstruction and its degree of severity.

6.9 Future Work

The next step for our work in wheeze detection, quantification, and qualification is to run extensive tests on the detector to determine its accuracy (or sensitivity) rate. This requires using test data for which the wheezes have been clearly and correctly labeled (e.g., by an experienced physician).

The current implementation of our wheeze detector allows only monophonic wheezes to be detected. The detection algorithm needs to be expanded to accommodate the detection of polyphonic wheezes as well. Furthermore, because it is useful to characterize wheezes by their timings within the respiratory phases (inspiration and expiration), adding to the system a function for automatically identifying wheeze timings is desirable.

Once the wheeze detector and quantifier are refined to achieve high accuracy rates, and the wheeze qualifier is expanded to output all of the necessary/relevant wheeze parameters, additional features could be introduced to the system. For example, the system could be programmed (or trained) to classify segments of an acoustic respiratory data stream as one of the following: fixed monophonic, random monophonic, sequential inspiratory, or expiratory polyphonic. The system could also be trained to classify data segments according to the degree of airway obstruction (or lung function).

Finally, if the application of the wheeze detector, quantifier, and qualifier requires continuous monitoring, the system must be robust against a wide variety of noises. This includes noise from: abrasions between the stethoscope's sensor and the patient's skin, speech, cough, snore, and the environment. In other words, the wheeze detector must have a high specificity rate.

With high sensitivity and specificity rates, a wheeze detector could provide invaluable information about respiratory function and the progression of pulmonary pathologies such as asthma.

Chapter 7

Time-Scale Expansion of Lung Sounds

In many instances, listening to lung sounds at slower speeds can be useful for making more accurate qualitative and quantitative assessments about the signal. Consider, for example, characterizing the presence of crackles. The typically quick succession of crackles over the duration of a respiratory cycle can make them difficult to aurally detect. In this chapter, we propose a *time-scale expander* that can stretch a lung sound signal in time by an integral number, while minimally distorting its frequency content.

7.1 Introduction

While **time** domain features are important for characterizing lung sounds, they can provide only a partial view of the physiological and anatomical state of the lungs. Equally important in characterizing lung sounds are its **frequency** domain features. The *phase vocoder* is a popular method for time-scale expanding speech and other acoustic signals that allows the frequency content of the signals to be preserved [10] [19] [21]. We use the phase vocoder to implement our tool for time-scale expanding lung sounds.

7.2 Phase Vocoder Theory

This section provides a brief overview of the basic theories underlying the phase vocoder for time-scale expansion of an acoustic signal[22].

Consider an acoustic signal $x(t)$. According to the theory of *Fourier transform*, $x(t)$ can be represented as a sum of weighted and shifted sinusoids[23]. Such a representation of $x(t)$ is given by the *Fourier synthesis* equation

[22] Much of this material is based on [10].

$$x(t) = \frac{1}{2\pi}\int_{-\infty}^{\infty} X(jw)e^{jwt}dw. \qquad \text{(Eq.1)}$$

The term $X(jw)$ is generally complex for real-valued signals, and contains information about the exact values of the weights and shifts for each sinusoidal component of frequency w. It is called the *Fourier transform* of $x(t)$, and is given by the *Fourier analysis* equation

$$X(jw) = \int_{-\infty}^{\infty} x(t)e^{-jwt}dt. \qquad \text{(Eq.2)}$$

Suppose $x(t)$ is passed through a system consisting of *N* band-pass filters $f_1(t),...,f_N(t)$, each of which has a relatively flat magnitude response and linear phase characteristics in the passband region, and that contiguously span the entire frequency spectrum. This procedure, known as subband analysis [2] or filter-bank analysis [27], produces at the output of the nth filter $x_n(t)$ such that the input signal can approximated as

$$x(t) \cong \sum_{n=1}^{N} x_n(t). \qquad \text{(Eq.3)}$$

If the impulse response of the nth filter is

$$f_n(t) = h(t)\cos w_n t, \qquad \text{(Eq.4)}$$

where w_n is the center frequency of the passband and $h(t)$ is the impulse response of a low-pass filter, then the output of the nth filter is the convolution of its impulse response with the input signal,

$$x_n(t) = \int_{-\infty}^{t} x(\gamma)h(t-\gamma)\cos[w_n(t-\gamma)]d\gamma = \Re\{e^{jw_n t}X(w_n,t)\}, \qquad \text{(Eq.5)}$$

where

$$X(w_n,t) = \int_{-\infty}^{t} x(\gamma)h(t-\gamma)e^{-jw_n t}d\gamma \qquad \text{(Eq.6)}$$

is the *short-time Fourier transform (STFT)* of a windowed segment of $x(t)$ (scaled accordingly by the coefficients of $h(t)$), evaluated at w_n. In general, the STFT is a function of two parameters, w and t, that correspond to frequency and time indices, respectively. In this case, because the frequency parameter w is fixed to w_n, the STFT shown in (Eq.6) can

[23] Refer to [28].

essentially be viewed as function of time. Since the STFT is generally complex-valued, it can be decomposed into its magnitude component $|X(w_n,t)|$ and phase component $\angle X(w_n,t)$,

$$X(w_n,t) = |X(w_n,t)|\, e^{j\angle X(w_n,t)}. \qquad \text{(Eq.7)}$$

Thus, the output of the nth filter can alternatively be expressed as

$$\begin{aligned} x_n(t) &= \Re\{e^{jw_n t}\,|X(w_n,t)|\,e^{j\angle X(w_n,t)}\} = |X(w_n,t)|\,\Re\{e^{j(w_n t+\angle X(w_n,t))}\} \\ &= |X(w_n,t)|\cos[w_n t + \angle X(w_n,t)], \end{aligned} \qquad \text{(Eq.8)}$$

which can be interpreted as a sinusoidal carrier signal of frequency w_n, modulated by the short-time Fourier magnitude and phase (evaluated at w_n) for each time instant t.

Let $\phi(w_n,t) = \angle X(w_n,t)$. If the derivative of $\phi(w_n,t)$ with respect to t is $\phi'(w_n,t)$, then an equivalent expression for the phase is

$$\phi(w_n,t) = \int_{-\infty}^{t} \phi'(w_n,\gamma)\,d\gamma + \phi_0, \qquad \text{(Eq.9)}$$

where ϕ_0 is its initial value.

Now, suppose we would like to expand the input signal by a factor of *r*. To do this, we need to:

(1) Multiply the frequency/phase terms, essentially $[w_n t + \angle X(w_n,t)]$ in (Eq.8), by a factor of *r* for each filter output;

(2) Synthesize new signals using the scaled frequency/phase term and the original magnitude response for each filter output as in (Eq.7);

(3) Sum all of the modified filter output signals as in (Eq.3); and

(4) Playback the signal at a speed that is *r* times as slow as the original speed. This is equivalent to digitally **interpolating** the signal by a factor of *r* and playing the resulting signal back at the original speed.

These steps will produce a signal that preserves the frequency components of the original signal, while expanding the duration of the signal by the desired factor *r*.

To illustrate how this expansion procedure works, take, for example, the discrete-time signal[24] $g[n] = v[n]w[n]$, where

$$v[n] = \cos(0.1\pi n),$$

and

$$w[n] = 0.54 - 0.46\cos(2\pi n / 100)$$ [25].

Suppose that $g[n]$ is an output of one of the filters of a filter-bank, with center frequency $w_n = 0.1\pi$. This means that $v[n]$ is the "carrier" signal of the filter output, and $w[n]$ is the modulating "envelope." Figures 1 plots the functions $g[n]$, $v[n]$, and $w[n]$ over the sample range $1 \leq n \leq 100$.

Say we would like to time-expand the waveform $g[n]$ by a factor of *r=2.* Starting with step (1), multiplying the frequency/phase term, which in this case is $w_n n = 0.1\pi n$, by a factor of two yields $2w_n n = 0.2\pi n$. This step doubles the frequency of the carrier signal; the new carrier signal (shown in Figure 2) is given by

$$\tilde{v}[n] = \cos(0.2\pi n).$$

In step (2), the original magnitude response, given in this case by $w[n]$, is multiplied by the new phase/frequency term given by $\tilde{v}[n]$. This yields the signal

$$\tilde{g}[n] = \tilde{v}[n]w[n],$$

also shown in Figure 2.

[24] Discrete-time (DT) signals are continuous-time (CT) signals sampled at some *sampling frequency*, $f_s = \frac{1}{T_s}$, where T_s is the sampling interval or *period.* Therefore, if $x(t)$ is a CT signal, then its DT counterpart is $x[n] = x(nT_s)$.

[25] This is the equation for a *Hamming window* of length M=100.

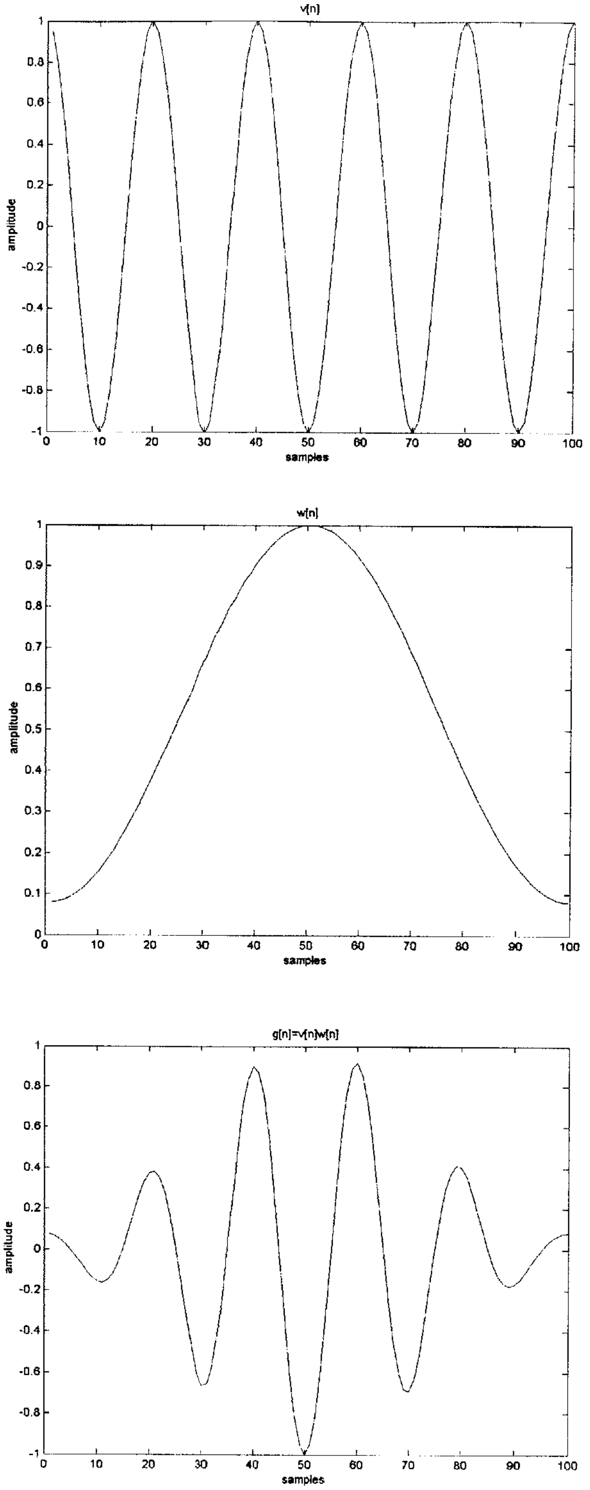


Figure 1. Plots of: (top) $v[n]$, (middle) $w[n]$, and (bottom) $g[n]$.

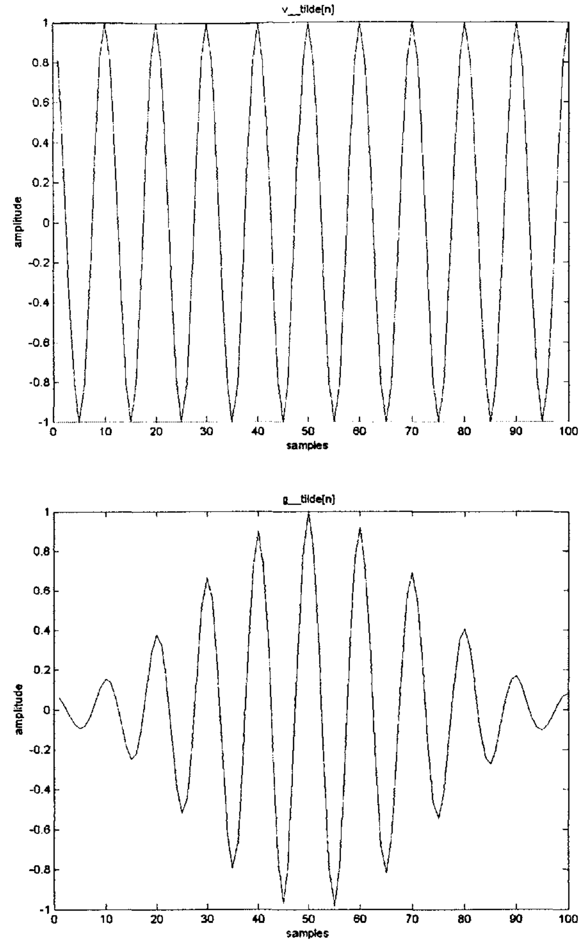


Figure 2. Plots of: (top) $\tilde{v}[n]$, and (bottom) $\tilde{g}[n]$.

Assume for now that $g[n]$ is the only output of a single-filter filter-bank; then the result of step (3) would be $\tilde{g}[n]$. Finally, in step (4), $\tilde{g}[n]$ is slowed down by a factor of two by playing the signal at half the original speed – this is equivalent to interpolating $\tilde{g}[n]$ by a factor of 2 and playing the signal back at the original speed. Figure 3 shows the interpolated signal $s[n]$, which spans $1 \leq n \leq 200$, and therefore has twice the length of the original signal $g[n]$. Notice that the new carrier signal for $s[n]$ oscillates at the same frequency as that of the original carrier signal $v[n]$. Thus, $s[n]$ is the time-scale expanded waveform of $g[n]$.

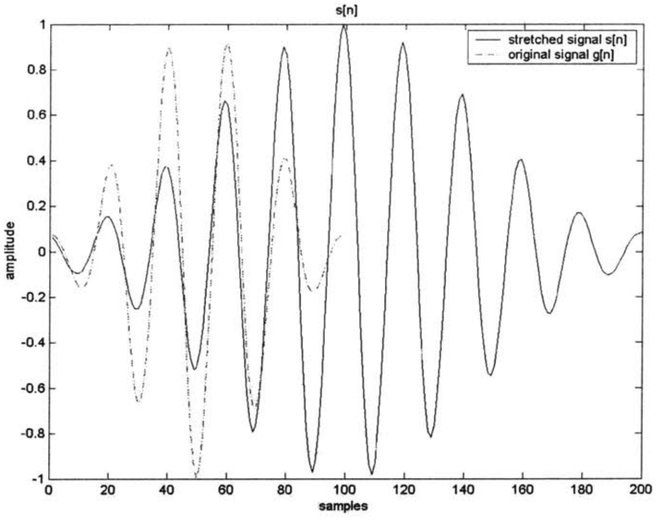


Figure 3. Plots of: time-scale expanded waveform $s[n]$ (dashed red line), and original waveform $g[n]$ (solid black line).

We discuss our approach to time-scale expanding lung sound signals in the next section.

7.3 Approach

Figure 4 is a block diagram of our *time-scale expansion of lung sounds* system design.

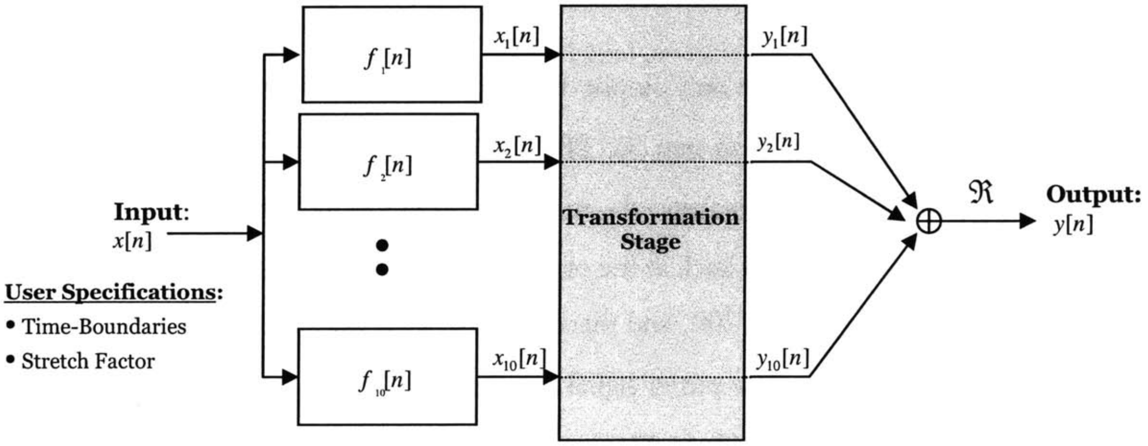


Figure 4. Block diagram of the *time-scale expansion of lung sounds* system.

This tool requires that the user specify at the input: (1) the filename containing the lung sounds data; (2) time-boundaries indicating the start and end times of the region of interest in the file; and (3) a stretch-factor $r \in Z^+$ by which the signal will be time-scale expanded. The system passes the signal to a filter-bank. The output of each filter in the filter-bank is then "transformed" in the *transformation stage.* Finally, the transformed signals are added together, yielding the time-scale expanded signal output.

7.4 Implementation

In this section we describe the implementation details of our *time-scale expansion of lung sounds* tool.

In the first step, the signal is passed through the filter-bank. This yields a complex-valued time signal at the output of each filter. The computationally efficient overlap-add method is used to compute these outputs [27]. We describe the design of this filter-bank next.

7.4.1 Filter-Bank Design

Since lung sounds typically span the [0 → 2,000] Hz frequency range, we use a filter-bank consisting of ten band-pass filters, $f_1[n],\ldots,f_{10}[n]$, which contiguously span [0 → 2,000] Hz. Each filter has a passband width of 200 Hz. The center frequency of the ith filter is

$$F_i = 100(2i-1) \text{ Hz}, \qquad \text{for } \ i = 1,2,\ldots,10.$$

Recall from (Eq.4) that the impulse response of the ith filter in continuous-time is $h(t)\cos w_i t$. In this discrete-time case, the impulse response of the ith filter is $h[n]\cos w_i n$. The relationship between w_i and F_i is given by

$$w_i = \frac{2\pi F_i}{f_s},$$

where f_s is the sampling frequency. Notice that $h[n]\cos w_i n$ is the real-part of the complex exponential $h[n]e^{jw_i n}$, since $h[n]$ is real. We use the complex-valued filter instead of its real-part such that the ith filter has an impulse response given by

$$f_i[n] = h[n]e^{jw_i n}. \qquad \text{(Eq.10)}$$

We use the *window method* [27] to define the low-pass filter $h[n]$, which has a finite impulse response (FIR) and linear-phase characteristics. We chose to use an FIR filter over an IIR (infinite impulse response) filter because of the linear-phase property that can be achieved using FIR filters if we impose the constraint that the impulse response must be symmetric. Linear phase is a desirable property for any application in digital signal-processing that requires perfect reconstruction of the original signal from its filtered components because it allows for perfect alignment of the phase information within the component signals. Misalignment of the phases can distort the original waveform upon reconstruction, and hence is an undesirable property for such applications as ours, where the user will expect to listen to an *undistorted*, time-scale expanded version of the original signal.

We use the *Blackman window* particularly because of its spectral behavior – the magnitude of its side lobes are smaller than those of other commonly used windows (e.g., rectangular, Bartlett, Hanning, and Hamming) such that passband ripples are kept to a minimum. Since we would like to have a flat passband response for our filters, minimizing ripples is a desirable property.

The impulse-responses of the filters, $f_1[n],...,f_{10}[n]$, have been designed to span a fixed duration of 0.1 seconds. We chose 0.1 seconds because it seemed to be a duration that is large enough to yield sufficiently high frequency-domain resolution in the STFT, while being small enough to be able to resolve the non-stationary time-domain events in lung sounds[26]. Figure 7 graphs the magnitude of the frequency response of the filter-bank.

7.4.2 Transformation Stage

In the *transformation stage*, the filtered signals at the output of the filter-bank, denoted by $x_1[n],...,x_{10}[n]$, are passed through a transformation procedure. Figure 5 illustrates how the ith filter output gets processed by the *transformation stage.*

[26] A trade-off exists between frequency resolution and time resolution in computing the STFT which depends directly on the window size used [27]. The trade-off (and, hence, window size) must be chosen to suit the particular application for which the STFT is being computed.

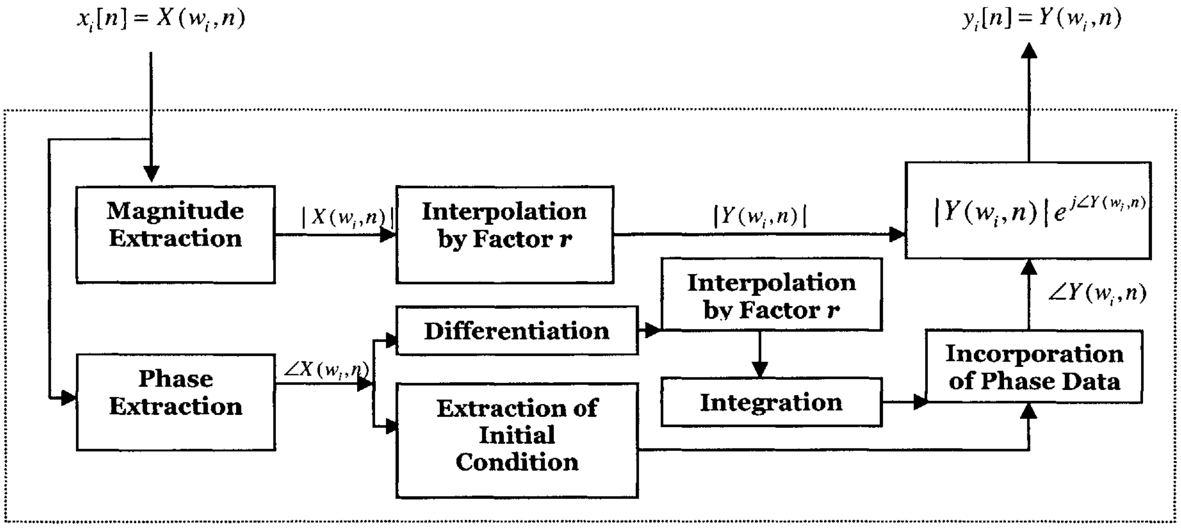


Figure 5. Block diagram of the *transformation stage.*

The first step in processing the filter output requires decomposing the signal into its magnitude and phase components. The magnitude part gets interpolated by *r* and undergoes no further processing. Interpolating the magnitude corresponds to stretching the envelope of the signal, as was demonstrated in the example in the "Phase Vocoder Theory" section of this chapter.

Next, the phase component is transformed. Note that before the phase part is processed, it must be **unwrapped**. When Matlab computes the phase signal, it takes the "unwrapped" phase signal and maps phase values that lie outside the $[-\pi \rightarrow \pi]$ range to their equivalent counterparts in the $[-\pi \rightarrow \pi]$ range, yielding the "wrapped" phase signal. If we denote the unwrapped phase signal by $\phi_{unwrapped}$ and the wrapped phase signal by $\phi_{wrapped}$, then the following must hold:

$$e^{j\cdot\phi_{wrapped}} = e^{j\cdot\phi_{unwrapped}},$$

where $\phi_{unwrapped} \in \Re$ and $\phi_{wrapped} \in [-\pi,\pi]$.

Now, suppose that the unwrapped phase signal is a continuous function whose values span a range larger than 2π. Then the wrapped version of this phase signal will be a discontinuous function with at least one point of discontinuity. Differentiating the wrapped phase signal will yield undefined slopes at every point of discontinuity. Therefore, to avoid having undefined slopes in the phase signal, the wrapped phase signal must be unwrapped prior to differentiation.

Next, the phase signal is differentiated. The initial value of the original phase part is stored, since the differentiation process loses this information. The differentiated phase signal is then interpolated by a factor of *r* and integrated. The resulting signal is incorporated with the stored initial condition to yield a new phase signal that is *r* times the length of the original phase signal. This {differentiation → interpolation → integration} procedure yields a frequency/phase term at the output of the transformation stage that the four-step procedure (described in the "Phase Vocoder Theory" section of this chapter) would also have yielded. We chose to use the former procedure over the latter one because it was simpler to implement.

After both the magnitude and phase parts have been transformed in the process describe above, they are combined into a complex-exponential form. This gives $y_i[n]$, which is the final output of the transformation stage.

Finally, the transformed signals, $y_1[n],\ldots,y_{10}[n]$, are added together to form a single waveform. The real part of this newly-synthesized, complex-valued signal is the final output of our system, and hence, the time-scale expanded version of the input lung sounds.

7.5 Performance and Discussion

To demonstrate the performance of this tool, we analyze a lung sound signal (that contains crackles) over one respiratory cycle. Figure 6 plots the original signal in the time-domain.

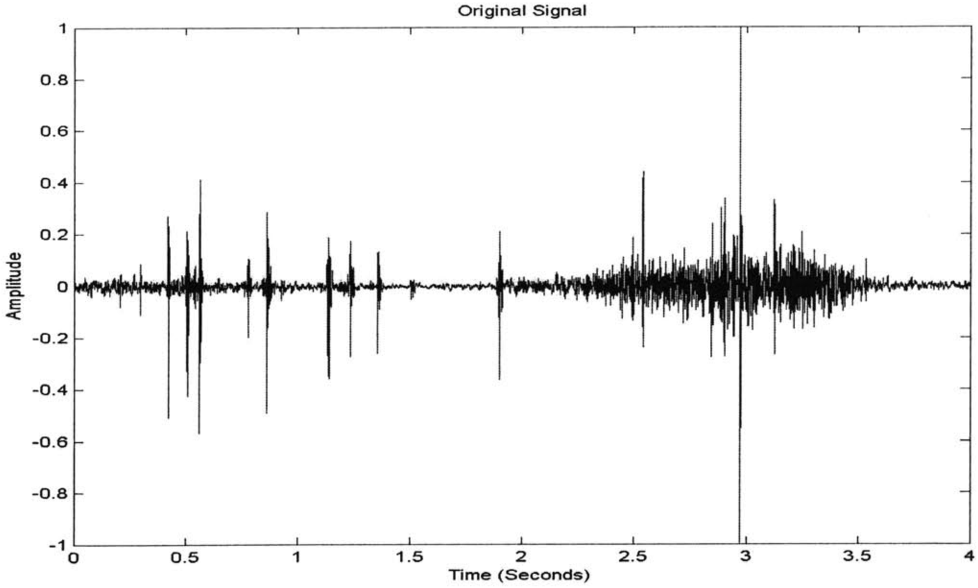


Figure 6. Time-domain plot of original signal.

Figure 7 plots the magnitude response of the filter bank overlaid on top of the magnitude response of the original signal.

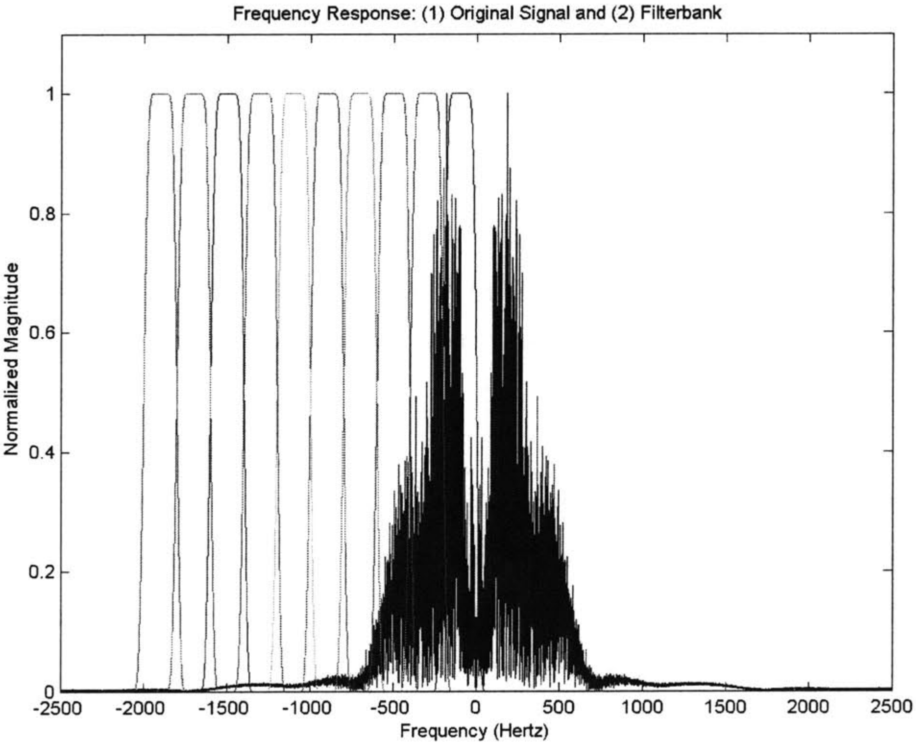


Figure 7. Plot of the magnitude responses of the original signal and the filter-bank.

Notice that the response of the filter bank is not even-symmetric – this is because the impulse-responses of the filters are complex-valued. Even-symmetry in the magnitude response is only a property of filters with real-valued impulse-responses.

Figure 8 shows the time-scale expanded version of the original signal using a stretch-factor of 2. The duration of the time-scale expanded signal is twice that of the original signal.

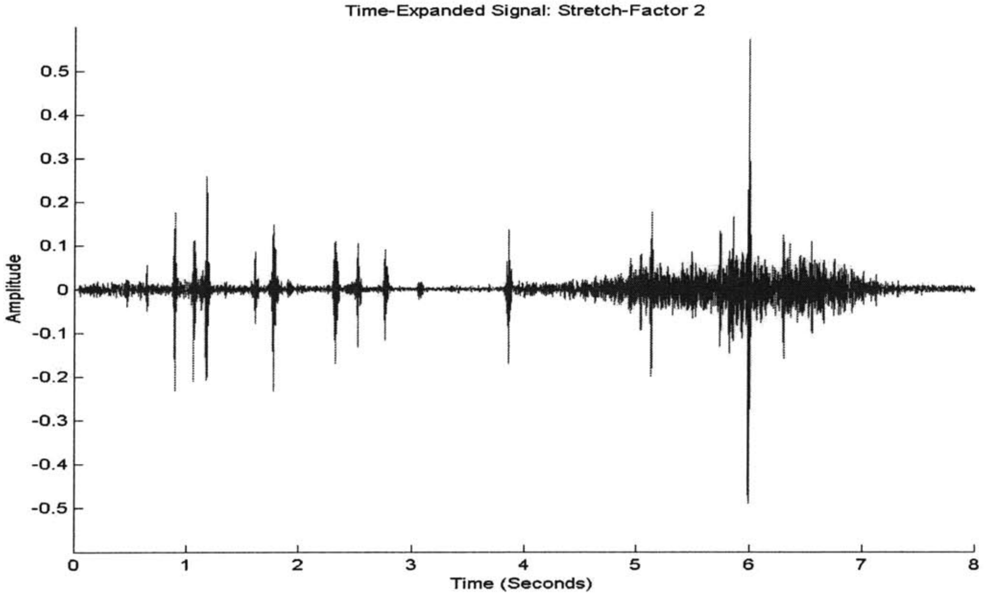


Figure 8. Time-domain plot of time-scale expanded signal using a stretch-factor of 2.

Figure 9 shows the spectrograms[27] of the original signal and the time-scale expanded signal. It is apparent that there are some distortions in the spectrogram of the time-scale expanded signal. These distortions include:

- A sharp discontinuity in the intensity at 2,000 Hz caused by the removal of all frequency components above this cutoff frequency by the filter bank.
- Horizontal striations in the [0 → 2,000] Hz frequency range caused by the fact the filters in our filter bank do not have ideal characteristics (i.e., perfectly flat passband

[27] The spectrogram is a time-frequency plot of the STFT – the intensity of the plot at each point in the time-frequency space corresponds to the magnitude of the STFT.

and stopband regions, and infinitely rapid transitions between the passband and stopband regions).

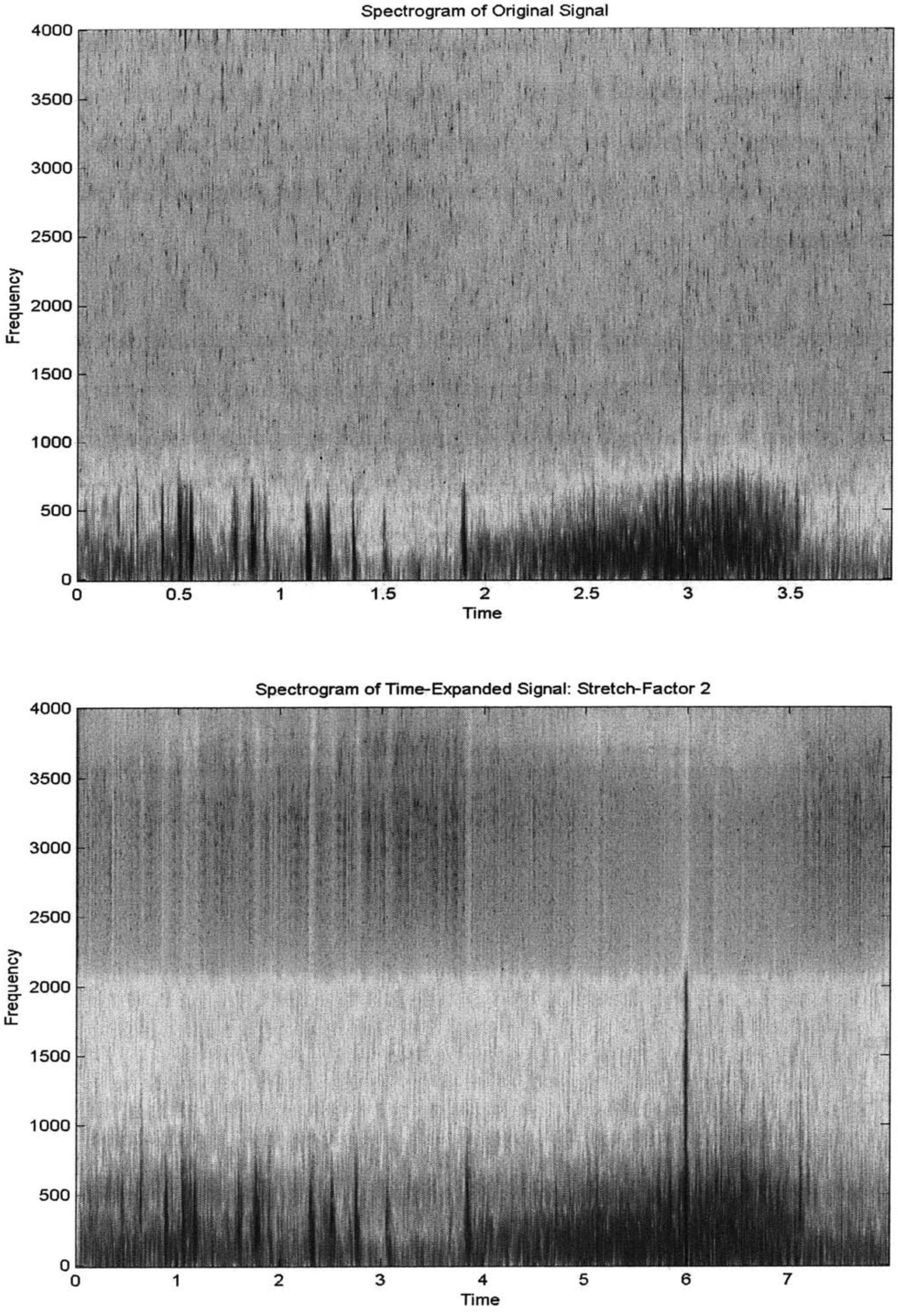


Figure 9. Plots of: (top) spectrogram of the original signal; and (bottom) spectrogram of the time-scale expanded signal using a stretch-factor of 2.

Aside from the distortions mentioned above, the "essential" frequency characteristics of the original signal seem to be well-preserved. For example, consider the first 2 seconds of the original signal over which dark, vertical striations are present in the spectrogram – these vertical striations correspond to crackles, and are also visible in the corresponding first 4 seconds of the time-scale expanded signal. The frequencies over which these vertical striations span are nearly identical for the original signal and the time-scale expanded signal, i.e., they span approximately 0 to 800 Hz for both signals. This indicates that the pitch of the crackles is preserved.

Now compare the spectrograms of the original and time-scale expanded signals with the spectrogram of the slowed-down (i.e., interpolated) original signal using a factor of 2, shown in Figure 10. Notice that slowing down the original signal significantly distorts its frequency content by compressing the frequency spectrum by a factor of 2. The vertical striations in the spectrogram of the slowed-down signal (in the first 4 seconds) range only from 0 to 400 Hz, causing the pitch of the crackles to be substantially reduced from its original range of 0 to 800 Hz.

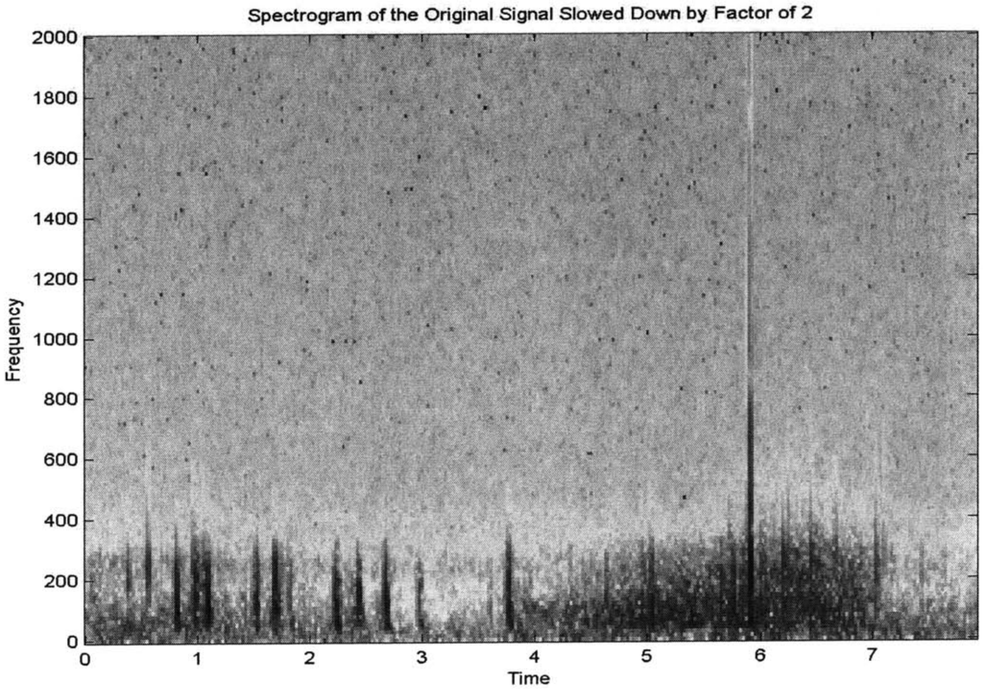


Figure 10. Spectrogram of slowed-down (i.e., interpolated) original signal using a factor of 2.

When listening to these sounds, the frequency content of the original signal seemed to be well-preserved in the time-scale expanded signal. On the other hand, the frequency content of the original signal to be seemed notably distorted in the slowed-down signal.

Furthermore, individual crackles were more easily distinguishable by ear in the time-scale expanded signal than in the original signal.

Chapter 8

Conclusion

In this chapter, we review the goals and contributions of our work on acoustic respiratory analysis, make suggestions for future work, and end the thesis with concluding remarks.

8.1 Goals and Contributions

The goal of our work was to build tools that could be used to help to understand better the characteristics of different lungs sounds, ranging from normal sounds (e.g., normal tracheal breath sounds) to pathological sounds (e.g., crackles and wheezes). This led us to develop algorithms for detecting, classifying and characterizing a variety of physiologically relevant respiratory events.

We approached this task using digital signal processing and machine learning techniques, and developed a *software toolkit* for analyzing lung sounds. These tools include a respiratory rate detector and respiratory phase classifier, crackle and wheeze detectors and characterizers, and a time-scale expander.

Our respiratory rate detector uses tracheal breath sounds in the time-domain to compute the respiratory rate (in respiratory cycles per minute) for regions in time over which breathing is detected. A complementary tool outputs the locations in time of detected respiratory phase onsets.

Our respiratory phase classifier uses a time-frequency representation of tracheal breath sounds and support vector machines to classify inspiration and expiration. We compared the classification accuracy rate of one proposed feature vector (which uses data from a single respiratory phase) with that of another feature vector (which uses data from two consecutive respiratory phases). We also compared the classification accuracy rate of a subject-specific approach with that of a generalized approach. We concluded from these comparisons that higher classification accuracy rates can be achieved using the subject-specific approach, and the feature vector which uses data from two consecutive respiratory phases.

Our crackle detector uses large deviations from the background vesicular sound intensities in the time-domain to detect crackles. Our crackle characterizer computes IDW and 2CD values, both of which are useful for classifying the crackles as being "fine" or "coarse" in pitch-quality. It also provides a graphical display of the 2CD regions for each crackle (to aid time-expanded waveform analysis), and the timing of the crackles within their respective respiratory phases.

Our wheeze detector identifies prominent, isolated peaks in the power spectrum of the lung sounds signal to detect wheezes. For each wheeze, our wheeze characterizer outputs its fundamental frequency and its energy as a percentage of the total energy in the signal. It also outputs the percentage of the duration of the entire signal that is occupied by wheezes.

Our time-scale expander stretches lung sounds in time while minimally distorting the frequency content of the signal. Slowing down lung sounds in this manner can aid in the aural detection of significant respiratory events.

For each tool, we described the system design and implementation details, provided a performance assessment, analyzed the results, and proposed suggestions for further improvement. We also compared our method with those proposed in related literature and weighed the pros and cons of each method.

8.2 Future Work

The detection/classification algorithms in our toolkit need to be tested on more data, and on data that are labeled according to gold standards, in order to gauge their true accuracy rates.

Once the algorithms achieve desired accuracy rates for any given application, a suitable graphical user interface needs to be developed.

The software might then be integrated with hardware devices such that the system can be deployed in clinical and/or home settings.

To researchers, our toolkit could be useful for finding interesting correlations between the values of parameters outputted from our system and different pulmonary conditions.

8.3 Concluding Remarks

Pulmonary diseases affect millions of people in the U.S. alone. An effective, convenient, minimally invasive, and less expensive alternative to diagnosing, screening for, and monitoring pulmonary diseases would be of great benefit to those afflicted with these diseases. With further research, an automated lung sounds analyzer could be the solution.

Bibliography

[1] American Sleep Apnea Association, *Information about Sleep Apnea.* http://www.sleepapnea.org/geninfo.html (August 2003)

[2] Baylon D.M. and Lim, J.S., Transform/Subband Analysis and Synthesis of Signals. *MIT Research Laboratory of Electronics Technical Report,* 559, June 1990.

[3] Burges, C.J.C., A Tutorial on Support Vector Machines for Pattern Recognition. *Data Mining and Knowledge Discovery,* 2, p121-167, 1998.

[4] Centers for Disease Control and Prevention and National Center for Health Statistics. *Pneumonia.* http://www.cdc.gov/nchs/fastats/pneumonia.htm

[5] Charbonneau, G., Ademovic, E., Cheetham, B.M.G., Malmberg, L.P., Vanderschoot, J. and Sovijarvi, A.R.A., Basic Techniques for Respiratory Sound Analysis. *European Respiratory Review,* 10: 77, p625-635, 2000.

[6] Chuah, J.S. and Moussavi, Z.K., Automated Respiratory Phase Detection by Acoustical Means. *Systems, Cybernetics & Informatics (SCI) Conference,* p228-231, July 2000.

[7] Cohen, A., Signal Processing Methods for Upper Airway and Pulmonary Dysfunction Diagnosis. *IEEE in Medicine and Biology Magazine,* p72-75, March 1990.

[8] Earis, J.E. and Cheetham, B.M.G., Current Methods used for Computerized Respiratory Sound Analysis. *European Respiratory Review,* 10: 77, p586-590, 2000.

[9] Earis, J.E. and Cheetham, B.M.G., Future Perspectives for Respiratory Sound Research. *European Respiratory Review,* 10: 77, p641-646, 2000.

[10] Flanagan, J.L., and Golden, R.M., Phase Vocoder. *Bell System Technical Journal,* 45, p1493-1509, November 1966.

[11] Forgacs, P., *Lung Sounds.* Tindall, London, 1978.

[12] Hadjileontiadis, L. and Panas, S., Separation of Discontinuous Adventitious Sounds from Vesicular Sounds Using a Wavelet-Based Filter. *IEEE Transactions on Biomedical Engineering,* Vol. 44, No. 12, p1269-1281, December 1997.

[13] Harper P., Kraman, S.S., Pastekamp, H. and Wodicka, G.R., An Acoustical Model of the Respiratory Tract. *IEEE Transactions on Biomedical Engineering,* Vol. 48, No. 5, p543-550, May 2001.

[14] Harper, P.V., Pasterkamp, H., Kiyokawa, H. and Wodicka, G.R., Modeling and Measurement of Flow Effects on Tracheal Sounds. *IEEE Transactions on Biomedical Engineering,* Vol. 50, No. 1, January 2003.

[15] Holford, S.K., *Discontinuous Adventitious Lung Sounds: Measurement, Classification, and Modeling.* ScD. Thesis, M.I.T., 1981.

[16] Homs-Corbera, A., Fiz, J.A. and Jane, R., Time-Frequency Detection and Analysis of Wheezes During Forced Exhalation. *IEEE Transactions on Biomedical Engineering,* Vol. 51, No. 1, p182-186, January 2004.

[17] Hult, P., Wranne, B. and Ask, P., A Bioacoustic Method for Timing of the Different Phases of the Breathing Cycle and Monitoring of Breathing Frequency. *Medical Engineering & Physics,* 22, p425-433, 2000.

[18] Jordan, M. I. and Bishop, C.M., *An Introduction to Graphical Models.* November 2000. (forthcoming)

[19] Laroche, J., and Dolson, M., Improved Phase Vocoder Time-Scale Modification of Audio. *IEEE Transactions on Speech and Audio Processing,* Vol. 7, No. 3, p323-332, May 1999.

[20] Lehrer, S., *Understanding Lung Sounds.* Saunders, Philadelphia, 1993.

[21] Malah, D., Time-Domain Algorithms for Harmonic Bandwidth Reduction and Time Scaling of Speech Signals. *IEEE Transactions on Acoustics, Speech, and Signal Processing,* Vol. ASSP-27, No. 2, April 1979.

[22] MayoClinic.com, *Pneumonia.* http://www.mayoclinic.com/invoke.cfm?id=DS00135 (October 2002)

[23] National Institutes of Health and National Heart, Lung, and Blood Institute, *The Lungs in Health and Disease.* NIH Publication No. 97-3279, August 1997.

[24] National Vital Statistics Report, *Deaths, Percent of Total Deaths, and Death Rates for the 10 Leading Causes of Death: United States 1999 (Table 1),* Vol. 49, No. 11, October 2001.

[25] Ono, R., Arakawa, K., Mori, M., Tsuneaki, S. and Harashima, H., Separation of Fine Crackles from Vesicular Sounds by a Nonlinear Digital Filter. *IEEE Transactions on Biomedical Engineering,* Vol. 36, No. 2, p286-291, February 1989.

[26] Oppenheim, A.V. and Lim, J.S., *Advanced Topics in Signal Processing.* Prentice-Hall, Englewood Cliffs, New Jersey, 1988.

[27] Oppenheim, A.V. and Schafer, R.W., *Discrete-Time Signal Processing (2nd Edition).* Prentice-Hall, Upper Saddle River, New Jersey, 1999.

[28] Oppenheim A.V. and Willsky, A.S., *Signals and Systems.* Prentice-Hall, Upper Saddle River, New Jersey, 1997.

[29] Oud, M., Dooijes, E.H. and van der Zee, J.S., Asthmatic Airways Obstruction Assessment Based on Detailed Analysis of Respiratory Sounds Spectra. *IEEE Transactions on Biomedical Engineering,* Vol. 47, No. 11, p1450-1455, November 2000.

[30] Pasterkamp, H., Kraman, S. and Wodicka, G., Respiratory Sounds: Advances Beyond the Stethoscope. *American Journal of Respiratory and Critical Care Medicine,* Vo. 156, No. 3, p974-987, September 1997.

[31] Prestwich, K.N., *Respiratory Volumes and Respiratory Dynamics.* Holy Cross College, Worcester, MA, 2003. http://www.holycross.edu/departments/biology/kprestwi/physiology/phys_pdf/Phys_Lect4_Respiration_pdf/Phys03_21_RespDynamics.pdf

[32] Pulmonology Channel. http://www.pulmonologychannel.com (July 2002)

[33] Pulmonary Rehabilitation Associates, *Lung Sounds.* http://www.pulmonaryrehab.com/lung_sounds.htm

[34] R.A.L.E. Lung Sounds 3.01, Compact Disc. http://www.rale.ca/LungSounds.htm

[35] Sa, R.C. and Verbandt, Y., Automated Breath Detection on Long-Duration Signals Using Feedforward Backpropagation Artificial Neural Networks. *IEEE Transactions on Biomedical Engineering,* Vol. 49, No. 10, October 2002.

[36] Soiferman, E., A Brief History of the Physical Diagnosis of the Chest, with an Emphasis on the Practice of Percussion. http://www.antiquemed.com/percus.html (2004)

[37] SVM Toolbox for Matlab. http://www.cis.tugraz.at/igi/aschwaig/software.html

[38] Varady, P., Micsik, T., Benedek, S. and Benyo, Z., A Novel Method for the Detection of Apnea and Hypopnea Events in Respiration Signals. *IEEE Transactions on Biomedical Engineering,* Vol. 49, No. 9, p936-942, September 2002.

[39] Waitman L. R., Clarkson K. P., Barwise, J., and King, P. H., Representation and Classification of Breath Sounds Recorded in an Intensive Care Setting Using Neural Networks, *Journal of Clinical Monitoring and Computing,* 16 (2) p95-105, 2000.

[40] Wheeze Sound Repositories:
http://www.tracheostomy.com/resources/multimedia/
http://www.rale.ca/wheezing_b.htm
http://nurspeak.com/tools/lung_sounds.htm
http://education.vetmed.vt.edu/Curriculum/VM8754/respir/sdf/sounds/sounds.htm
http://faemse.org/downloads.shtml
http://www.ymec.com/hp/signal2/lung2.htm

[41] Yap, Y.L. and Moussavi, Z., Respiratory Onset Detection Using Variance Fractal Dimension. *IEEE Engineering in Medicine and Biology,* p118, October 2001.