2009-01-01

An Integrated Software Package For Model-Based Neuro-Fuzzy Classification Of Small Airway Dysfunction

Nazila Hafezi
University of Texas at El Paso, nhafezi@miners.utep.edu

AN INTEGRATED SOFTWARE PACKAGE FOR MODEL-BASED NEURO-FUZZY CLASSIFICATION OF SMALL AIRWAY DYSFUNCTION

NAZILA HAFEZI

Department of Electrical and Computer Engineering

APPROVED:

Homer Nazeran, Ph.D., Chair

Patricia Nava, Ph.D.

Michael D Goldman, Ph.D.

Patricia D. Witherspoon, Ph.D.
Dean of the Graduate School

To my mom, Fatemeh Soltan Ahmadi,
my father and my brother,
and all brave Iranian women rising for freedom.

AN INTEGRATED SOFTWARE PACKAGE FOR MODEL-BASED NEURO-FUZZY CLASSIFICATION OF SMALL AIRWAY DYSFUNCTION

by

NAZILA HAFEZI, BSCE

THESIS

Presented to the Faculty of the Graduate School of

The University of Texas at El Paso

in Partial Fulfillment

of the Requirements

for the Degree of

MASTER OF SCIENCE

Department of Electrical and Computer Engineering

THE UNIVERSITY OF TEXAS AT EL PASO

December 2009

Acknowledgements

People say ‘there is always a woman behind a successful man’. For me, there is always my mom behind any little move I make in life. She supports me with her positive thoughts even now that I am half a world far away from her. I especially want to thank her for all her kindness and love. I also want to thank my father who always believes in me and has taught me good moral principles to apply in life. Mom, dad, thank you for seeing the beauties in me even when I am not aware of them. Because of both of you, your advice and encouragement, today I have earned my master’s degree in the United States.

I wish to express my gratitude to Dr. Homer Nazeran. He believed in me since I came to The University of Texas at El Paso (UTEP), giving me the opportunity to work as his graduate student. His knowledge, guidance and valuable time were fundamental for the conclusion of this study. Thank you very much Dr. Nazeran, I would not have earned this degree without your mentoring and advising.

I would like to express my most sincere appreciation to Dr. Patricia Nava, giving me the opportunity to work as Teaching Assistant in Department of Electrical and Computer Engineering in UTEP during my studies. Thank you very much for your support. Without it, this thesis would not be accomplished.

I’m especially grateful to Dr. Michael Goldman for his great advices and hints during this research and his kind sympathy during last summer that I was experiencing deep sadness. Thank you very much for understanding me and always encouraging me to do this thesis.

Also, I would like to thank my special friends, Erika Meraz who always helped me in understanding this research, Maryam Veisi who helped me come to the United States, Leith Ali Sabty who kindly supported me during the last months, Shirin Kiani and her mom, Sousan Amniyeh who were like my family members here in El Paso. Thank you all for your kindness and support. I will never forget your helps.

Abstract

Respiratory disorders can be difficult to differentiate, as their symptoms are sometimes similar to one another. Concerning asthma which is the major interest in this research, early detection of small airway impairment could greatly help with timely diagnosis. In this research, an integrated software package (ISP) has been developed to help clinicians in classification of asthma severities using Impulse Oscillation data.

The ISP has been developed based on object-oriented methodology. In this package, IOS "diagnostic" parameters such as AX and frequency-dependence of resistance as well as parameter estimates of extended RIC (eRIC) and augmented RIC (aRIC) models (simplified cases of Mead's model) are used as elements of feature vectors to represent respiratory impedance. Two Co-active Neuro-Fuzzy Inference Systems (CANFIS) are implemented in this software to classify respiratory patterns. Both of the systems used train data sets containing 112 patterns belonging to four different classes of asthma severities including asthma, small airway disease (SAD), mild SAD (mSAD) and normal. The first CANFIS utilized IOS impedance measurements and parameter estimates of the aRIC model, while the second CANFIS utilized parameter estimates of the eRIC model instead.

In order to improve on the user-friendliness and running times of the previous versions of codes for these modules, they were modified and enhanced before integration into the ISP.

To encapsulate the functionalities of the different components in the developed ISP, the adapter object-oriented design pattern was used in implementing the ISP. Following the object-oriented principles led to construction of fully object-oriented software. The flexibility of the software package in enabling the developer to add new components (modules) and to upgrade current components with newly developed versions makes it a robust and reliable platform to assist researchers and clinicians in pulmonary medicine who seek more sensitive diagnostic tools to explore small airway impairments based on IOS measurements.

After performing a thorough analysis of requirements of the software, the architecture of the ISP was designed and implemented in C#, one of the supported programming languages by Microsoft Visual

Studio 2008. The ISP has been tested successfully on Windows XP and Vista and it has an easy-to-use user interface to interact with the user.

Table of Contents

List of Tables

List of Figures

Chapter 1: Introduction

1.1 PROBLEM STATEMENT

Asthma affects 3% to 5% of adults and 7% to 10% of children. Half of the people with asthma develop it before age 10 and most develop it before age 30. The rate of asthma has increased by about 60% since 1979 among all ages, races, and gender groups; however, it is slightly more common in blacks and Hispanics than in whites [1].

Asthma and Chronic Obstructive Pulmonary Disease (COPD) can greatly affect human lives. Asthma can be difficult to diagnose, as its symptoms are sometimes similar to other conditions, including allergic rhinitis (hay fever), lung infection, and even cardiac problems [2]. Early detection of small airway impairment could greatly help with timely diagnosis of asthma. Impulse oscillometry has made it possible to quantitatively evaluate the resistive and elastic components of the respiratory system and requires only resting breathing during test performance and minimal cooperation of the subjects.

Impulse oscillation system (IOS) provides valid, objective measurements of lung mechanics. Parameter estimates for such models may then possibly serve as a means to detect and diagnose respiratory diseases. Many recent reports [2], [3], [4], [5] and [6] have described impulse oscillometry as more sensitive than conventional spirometry in detecting small airway dysfunction and in providing more useful information in a quantitative fashion.

Although respiratory system impedance may be obtained directly from IOS testing, analysis and interpretation of this information has proven challenging for pulmonologists. Therefore, IOS is not currently widely used in clinical pulmonary function testing. To address these difficulties, a model-based co-active neuro-fuzzy inference system (CANFIS) has been developed and tested. In its current and enhanced implementation the software package is able to classify the severities of 4 small airway conditions based on respiratory airway impedance measurements from IOS and respiratory impedance model parameters.

This study investigates the prospects of applying object-oriented design and programming to address some biomedical engineering aspects of pulmonary medicine. The main aim of this study is to develop an integrated, interactive, and user-friendly software package to classify the severities of small

airway conditions in asthmatic and normal subjects. This goal is achievable by utilizing the object-oriented design methodologies, model-based feature extraction and an inference system that incorporates the strengths of artificial neural networks and fuzzy decision making (soft computing).

Previous research by our team has shown that model-based features and CANFIS classifiers have proven helpful for working with IOS data. In the earlier implementation of these approaches, the user had to accomplish separate tasks at different stages such as respiratory system model parameter estimation, manual selection of the best estimates, feeding the selected parameters into a program and combining them with other calculated parameters from IOS measurements (to extract all the required features), running the CANFIS program with the generated inputs and providing the classification results in an output file. Besides being a bit challenging for a medium-level computer user, human performance could be affected by many factors in this step-by-step implementation and if not done carefully this approach had the potential to lead into misclassification.

2.2 PROPOSED SOLUTION

A definition of an integrated software is a collection of computer programs designed to work seamlessly together to handle an application, either by passing data from one to another or as modules of a single system. In recent decades, many well-developed software packages have been commercialized in different areas of biomedicine such as biomedical signal processing, bioinformatics, and neuroscience. In all of these systems the primary goal is construction of an interactive computer-aided environment to create a sophisticated computing solution dealing with complex medical problems.

Typically, the applications in biomedicine involve analysis and classification of experimental data that are often composed of many different variables. The IOS measurements provide various types of measured values (features) that could make the analysis of this information for the untrained and busy practitioner difficult.

In previous implementation of the human respiratory system models (e.g. the augmented RIC and extended RIC models), the input data set for the parameter estimation models was limited to six frequencies. This rigidity caused some limitations in analysis and software development. The programs

needed to be modified, enhanced and validated to ensure more flexibility and enable the user to work with any other number of frequencies in the range of IOS measured values.

Artificial Neural Networks (ANNs) can be characterized as parallel distributed processing systems capable of solving complex problems of classification. ANNs are similar to the systems that implement mapping functions, and are characterized by their flexibility and generalization abilities. Using fuzzy logic in the decision making phase of training the ANNs, their robustness and accuracy could be further improved as the investigations carried out by our research team have shown.

The remaining task in this research was development of a robust, reliable and integrated software package with a user-friendly interface to interact with the user to get the IOS measurements and provide the classification results in an easy to understand fashion at the end.

Therefore, an object-oriented architecture and design was developed in this research. This package includes eRIC and aRIC parameter estimation models along with their associated CANFIS programs that could be considered and implemented as isolated objects with encapsulated properties and user-friendly interfaces to interact with other objects inside the package. In order to improve on the running time, the *programming language* of codes for these components was changed. In addition, the codes were enhanced and modified.

After performing a thorough analysis of requirements of the software, the architecture of the package was designed. Next step was implementing the software package in C# programming language which is a modern programming language that promotes object-oriented software development by offering syntactic constructs and semantic support for concepts that map directly to notions in object-oriented design.

This research work is presented in seven chapters. Chapter 2 reviews the anatomy of the human respiratory system, some diseases that affect this system, and pulmonary function tests developed for obtaining respiratory measurements. Chapter 3 provides details of impulse oscillation system and respiratory impedance models. Chapter 4 generally explains the basic information on artificial neural networks, their characteristics, the concept of fuzzy logic, fuzzy sets, membership functions, if-then rules as well as fuzzy inference systems and briefly introduces fuzzy models. It also includes description

of the co-active neuro-fuzzy inference system (CANFIS) and its details. In Chapter 5 the architectural design, components and the implementation of the proposed integrated software package are discussed. Chapter 6 explains the integrated software step-by-step from end-user point of view and discusses how to obtain classification results from running the software. Chapter 7 wraps up the research work by providing the conclusions drawn from analysis of the results in Chapter 6 and makes recommendations for future enhancement of the work.

Chapter 2: Physiology of the Human Respiratory System

2.1 INTRODUCTION

In order to develop efficient and useful lung testing methods to diagnose pulmonary diseases, a good knowledge about the function of the respiratory system is necessary. In this chapter, a brief review of the anatomy and physiology of the respiratory system, some diseases that affect the small airways and alveoli, and finally several pulmonary tests used in diagnosis or treatment evaluation will be discussed. Accurate diagnosis, effective treatments and advancing responsible medicine is possible through the good understanding of how the respiratory system behaves in health and in disease to help those who suffer from pulmonary disorders.

2.2 THE HUMAN RESPIRATORY SYSTEM

The primary function of the respiratory system is to allow gas exchange. It should supply the blood with oxygen in order for the blood to deliver oxygen to all cells of the body and remove the unwanted carbon dioxide generated by these cells which may be harmful to the body. The respiratory system achieves this function through breathing.

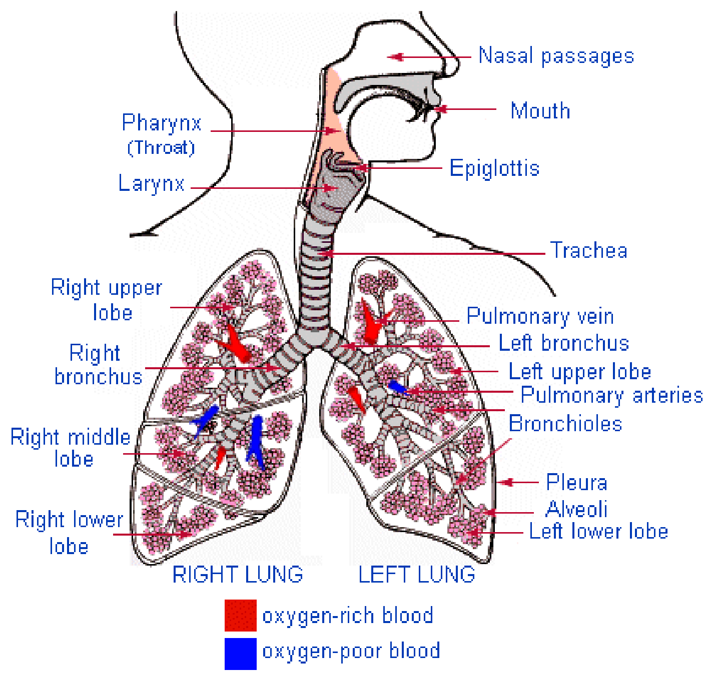


Figure 2.1: Organs of the body involved in the breathing process [7]

The organs in the human body that are involved in the process of breathing consist of the nose, pharynx, larynx, trachea, bronchi and lungs as shown in Figure 2.1 [7]. Based on anatomical classification, these organs can be subdivided into two tracts:

a. Upper respiratory tract: includes the nasal passages (nose, nasal cavity and sinuses), pharynx, larynx and trachea.
b. Lower respiratory tract: comprised of bronchi, bronchioles, lungs, and alveoli found in the lungs.

The diaphragm, the thin, dome-shaped sheet of muscle attached to the lower ribs and spine is the most important muscle of inspiration and is the other part of respiratory system which is involved in moving air in and out of the lungs by contracting and relaxing, respectively.

2.2.1 Anatomy of the Human Respiratory System

Molecules of oxygen and carbon dioxide are passively exchanged, by diffusion, between the gaseous external environment and the blood. This exchange process occurs in the alveolar region of the lungs.

Based on dividing the respiratory system into two tracts, the breathing pathway starts from nasal passages (nose) which are part of the upper respiratory tract or the conducting zone. The nasal passages open into the pharynx (throat), which serves as a common passageway for both the respiratory and digestive systems. One of the two tubes, the trachea, leads from the pharynx allowing air to be conducted to the lungs. The larynx, or the voice box, is located at the entrance of the trachea. Beyond the larynx, the trachea divides into two main branches which are part of the lower respiratory tract the right and left bronchi entering the right and left lungs, respectively. Within each lung, the bronchus continues to branch into progressively narrower, shorter and more numerous airways. The smaller branches are known as bronchioles. Clustered at the end of the terminal bronchioles are the alveoli, the tiny air sacs where gas exchange between air and blood takes place [8]. Figure 2.2 [9] shows the parts of both the upper and lower respiratory tracts.

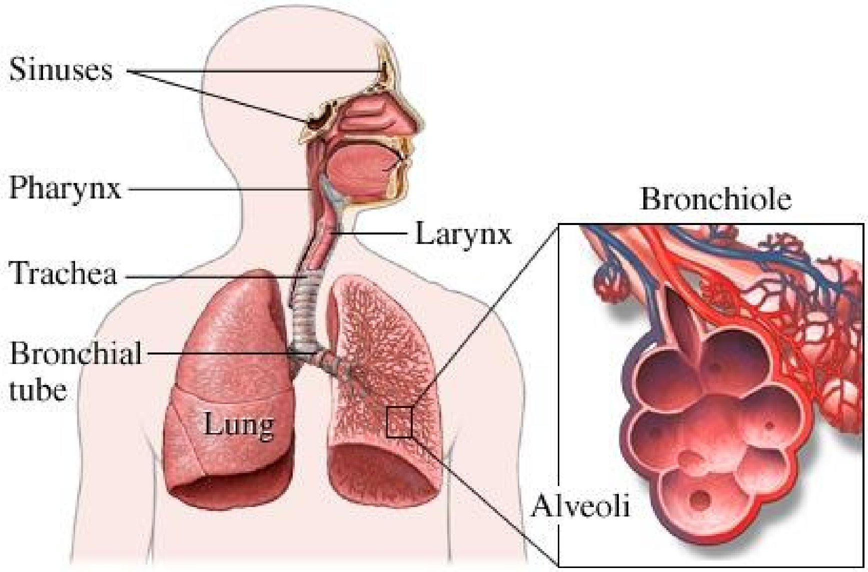


Figure 2.2: Upper and lower respiratory tracts [9]

In order to accomplish the process of gas exchange, the path of airflow, the continuum of conducting airways from the entrance through the terminal parts must remain open.

2.2.2 Gas Exchange

In physiology, the term respiration encompasses two separate but related processes: external respiration and internal respiration [8]. External respiration includes the process of exchanging oxygen and carbon dioxide between the outside environment and the tissue cells while internal respiration consists of the intracellular metabolic reactions involving the use of oxygen to drive energy from food and producing carbon dioxide as a by-product.

In this study, the external respiration is the part that is important to understand. External respiration, which is mainly the gas exchange process, takes place in three steps. First, air is alternatively moved in and out of the lungs through the airways by the mechanical act of breathing or ventilation. The second step is gas diffusion. The air passes through the upper tract of the respiratory system to the alveoli and then reaches the tiny air sacs located at the end of the airway path in the lungs where the contact with deoxygenated blood coming from the heart takes place. Permeable walls of the alveolar sacs and pulmonary capillaries are thin enough to permit gas exchange. The third step is when the

oxygenated blood transports the gases between lungs and the tissues of the body and exchange of oxygen and carbon dioxide between blood and tissues occurs.

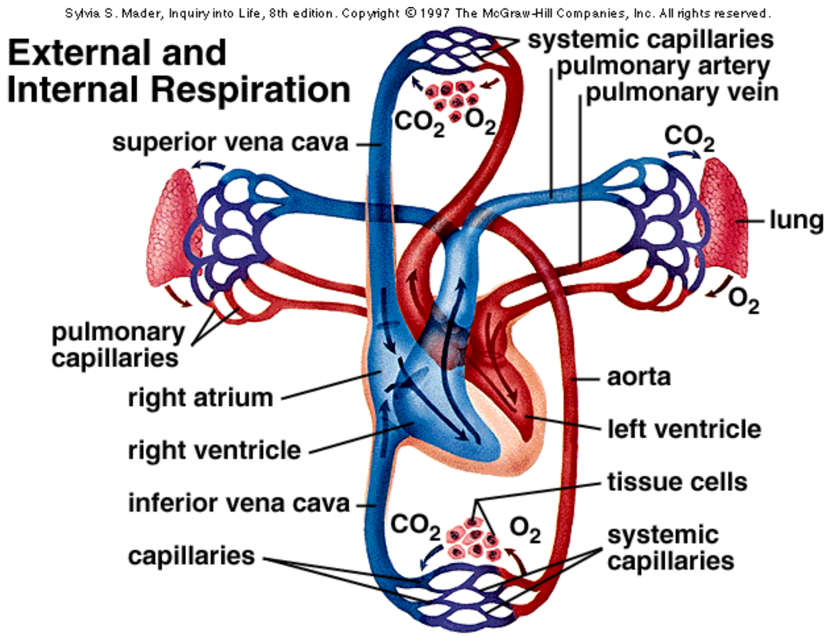


Figure 2.3: Process of gas exchange [10]

Figure 2.3 [10] shows the process of gas exchange. In the external respiration, carbon dioxide diffuses from pulmonary arteries into alveoli and oxygen diffuses from alveoli into pulmonary capillaries. Internal respiration includes oxygen diffusion from capillaries into cells and carbon dioxide diffusion from cells into capillaries.

2.2.3 Mechanics of Breathing

Breathing, or pulmonary ventilation, consists of two phases: inhalation and exhalation. Inhalation is the period when air flows into the lungs while exhalation is when gases exit the lungs.

Before describing the process of breathing, it is important to know that there are three different pressure considerations in ventilation:

Atmospheric (barometric) pressure: the pressure exerted by the air surrounding the body. At sea level, it is 760 mm Hg.

Intrapulmonary (intra-alveolar) pressure: the pressure in the alveoli. It rises and falls with phases of breathing, but it is equalized with the atmospheric pressure at the end of each phase of breathing (when flow is zero) [11].

Ventilation is a process based on the volume changes in the thoracic cavity. Pressure changes in the thoracic cavity lead to the flow of gases, following the pressure gradient, leading to volume changes.

The thoracic cavity is larger than the unstretched lungs because the thoracic wall grows more rapidly than the lungs during development [8] and there are two forces that cause lung collapse: first, because of their elasticity, lungs always tend to recoil to the smallest size and second, the surface tension of the alveolar fluid constantly acts to draw the alveoli to their smallest possible dimension.

In order to make the air flow in and out of lungs, the central nervous system sends instructions to the respiratory muscles to generate the required forces. Lungs do not have muscles and cannot expand on their own. The respiratory muscles and the diaphragm provide the necessary expansion and contraction needed to pull air in and out of the lungs. Changes in thoracic volume during inspiration and expiration take place in this order:

Inspiration: Before the beginning of inspiration, the respiratory muscles are relaxed and intra-alveolar pressure is equal to atmospheric pressure. In the period of inspiration, the respiratory muscles including the diaphragm and external intercostal muscles are stimulated to contract. The diaphragm descends and contraction of external intercostal muscles makes the rib cage elevate. These changes make the pressure in the alveoli fall, and accordingly, the volume of the thoracic cavity increase. As the lungs are stretched, intrapulmonary volume increases as air flows deep into the lungs until the intrapulmonary pressure is equal to atmospheric pressure, at the end of inspiration, when flow ceases.

Expiration: At the end of inspiration, the respiratory muscles relax. The diaphragm rises to assume its original dome-shaped position and the rib cage descends due to its inward elastic recoil. So the volume of thoracic cavity decreases and elastic lungs recoil passively which causes intrapulmonary pressure to rise leading to airflow out of the lungs until the intrapulmonary pressure equals atmospheric pressure.

Beside the elastic recoil of the lungs, other physical factors can influence pulmonary ventilation. Some of these factors include [11]:

Airway resistance: The major nonelastic source of resistance to gas flow is friction encountered in the respiratory pathways. The relationship between gas flow (F, in L/sec), pressure (P, in cmH2O) and resistance (R, in cmH2O/L/sec) is given by the following equation [11]:

$$F=\Delta P/R \quad (2)$$

Where ΔP is the ratio of the pressure difference between atmospheric and intra-alveolar pressure. Airway resistance is inversely proportional to the volume of lungs, so it decreases when the volume of the lung increases and vice versa. During inspiration, the volume of the lungs increase, so airway resistance decreases. Conversely, in the expiration phrase, the volume of the lungs decrease and airway resistance increases consequently.

The equation also indicates that gas flow decreases as the resistance due to friction increases. Gas flow (conduction) stops when the air reaches the small airways with small diameters at terminal bronchioles, and diffusion takes place as the main force driving gas movement. So the greatest resistance to gas flow occurs in the medium-sized bronchi [11].

Alveolar Surface Tension Forces: In an environment that includes gas and liquid, the molecules of liquid are more attached to each other than to gas molecules. This strong attachment between liquid molecules produces a tension at the surface of liquid, called surface tension, which binds liquid molecules closer together and reduces their contact with gas molecules. Because of existence of this tension, the liquid resists any force that tends to expand its surface area. The major component of the liquid covering the alveolar walls is water, but there is also another component called surfactant that decreases the cohesiveness of water molecules so that less energy is needed to overcome the surface tension forces and lungs are expanded during the inspiration.

Lung Compliance: Elastic recoil property can be measured and quantified by the quantity, compliance. Elastic recoil is a property of the lungs that embodies their stretchability and their tendency to snap back to their resting position when the stretching forces are removed [12]. In this research, the compliance we are looking for relates to the volume increment caused by the impulse oscillation system

(IOS). The IOS pressure increment added at the mouth causes this volume increment. After accounting for resistive pressure loss, the volume increment divided by the mouth pressure pulse is compliance to be estimated by IOS data and model analysis.

2.2.4 Respiratory Capacities

In order to determine one person's respiratory status, specific combinations of respiratory volumes need to be measured, which are called respiratory capacities. These parameters depend on the amount of air flushed in and out of the lungs during inspiration and expiration.

There are four respiratory volumes important in respiratory capacities measurements [11]:

Tidal Volume (TV): The amount of air flowing in and out of lungs during one normal breathing cycle. It is about 500 mL in an average adult.

Inspiratory Reserve Volume (IRV): The additional amount of air that can be inspired forcibly after tidal volume. It is about 2100 to 3200 mL.

Expiratory Reserve Volume (ERV): Similar to IRV, it is the additional amount of air that can be evacuated from the lungs after a tidal expiration. It is normally about 1000 to 1200 mL.

Residual Volume (RV): The amount of air remaining in lungs even after maximal expiration. It can help to keep the alveoli patent open and prevent lung collapse. It is about 1200 mL.

The respiratory capacities consist of two or more lung volumes. There are four respiratory capacities:

Inspiratory Capacity (IC): The total amount of air that can be inspired after a tidal expiration. So it is the sum of tidal volume and inspiratory reserve volume.

$$IC=TV+IRV \quad (4)$$

Functional Residual Capacity (FRC): The amount of air remaining in lungs after a tidal expiration.

$$FRC=RV+ERV \quad (5)$$

Vital Capacity (VC): The total maximum amount of exchangeable air that can be exhaled from lungs after maximum inhalation.

$$VC=TV+IRV+ERV \quad (6)$$

Total Lung Capacity (TLC): The sum of all lung volumes. It is normally around 6000 mL in males.

TLC=RV+ERV+TV+IRV=RV+VC=IC+FRC (7)

Figure 2.4 [12] indicates the different respiratory volumes and capacities. Lung volumes and capacities are smaller in women than in men because of their size difference. They are greater in athletic people than in average or unfit people.

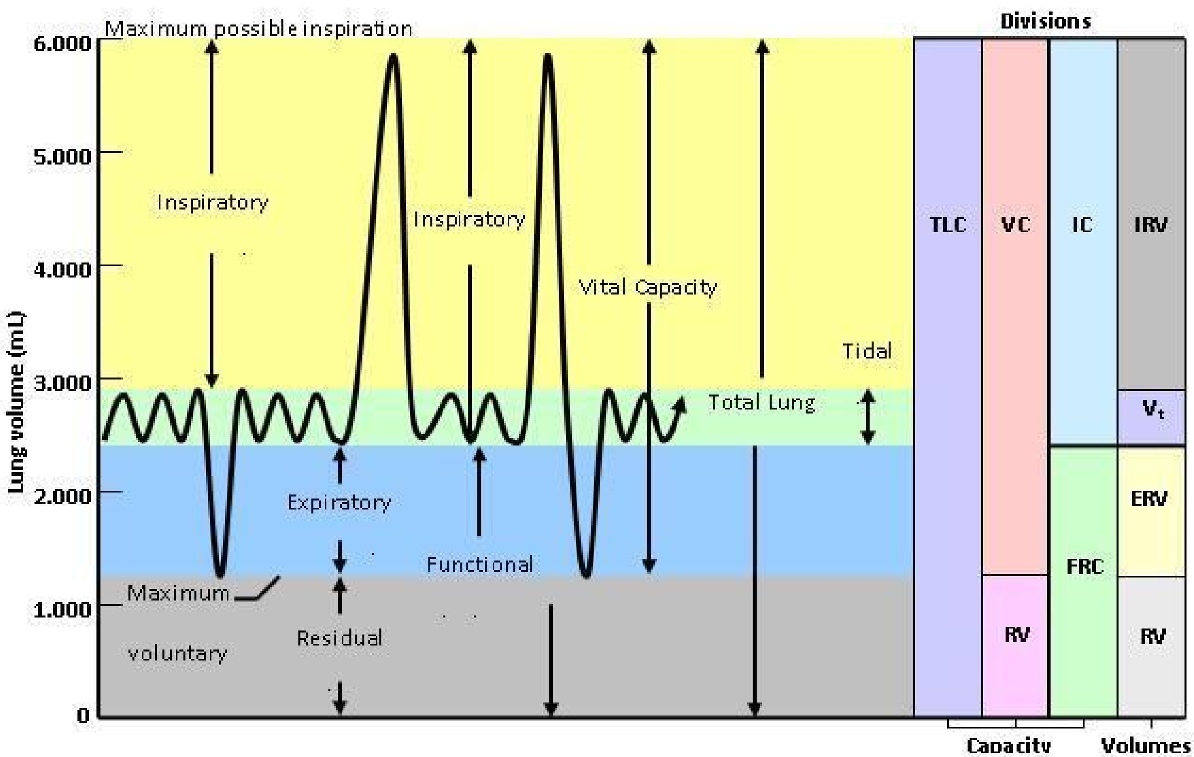


Figure 2.4: Lung volume and capacities [12]

2.2.5 Pulmonary Diseases

In normal individuals, breathing takes minimal effort from the body but the pulmonary system is sensitive to multiple agents that can trigger reactions or diseases. The problems in the respiratory system happen when there is an abnormality in flowing of air through airways. The most disabling disorders are chronic obstructive pulmonary diseases (COPD), asthma, lung cancer and restrictive lung diseases.

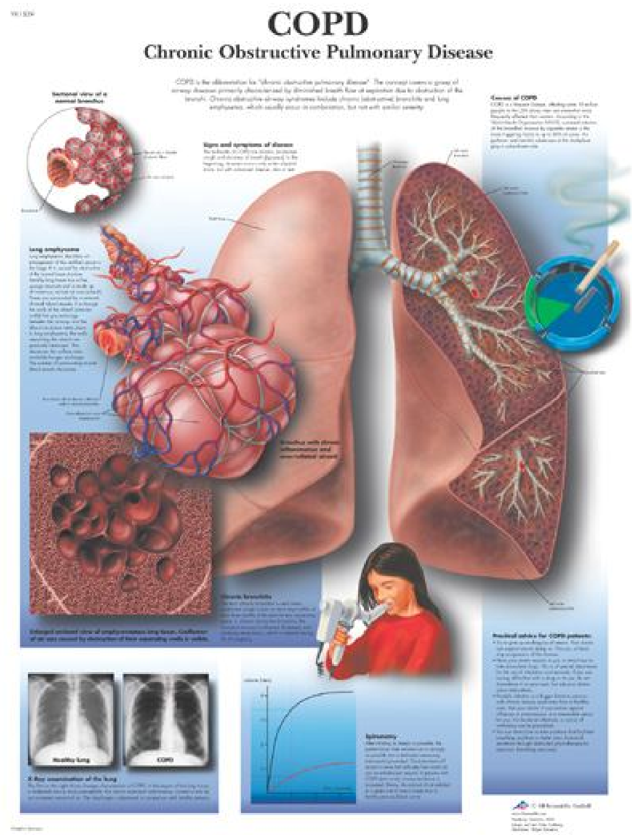


Figure 2.5: Pathogenesis of Chronic Obstructive Pulmonary Diseases (COPD) [11]

Chronic Obstructive Pulmonary Diseases (COPD): This category is the major cause of death and disability in the United States among all pulmonary diseases. Two main examples of them are obstructive emphysema and chronic bronchitis.

Obstructive emphysema starts with deterioration of the alveolar walls (it is the deterioration of alveolar walls that causes enlargement of alveoli) and the gas exchange area of the lungs get reduced. The lungs lose their elasticity and the airways collapse during the expiration and obstruct the outflow of air.

Chronic bronchitis is a long-term inflammatory condition in lower respiratory airways. The mucosa of the lower respiratory pathways produce more mucus than normal which is the result of cigarette smoke, air pollutants, or allergens. The air pathways inflame and these responses obstruct the airways and impair lung ventilation and gas exchange. Due to the higher amount of produced mucus, bacteria thrive in the slugged pools of mucus, so lung infections happen frequently.

Figure 2.5 [11] shows the pathogenesis of COPD.

Asthma: Asthma is a respiratory disorder characterized by single or combination episodes of coughing, dyspnea, wheezing and chest tightness. Although it may be classified as COPD because it is an obstructive disorder, asthma is marked by acute exacerbations followed by symptom-free periods [11]. The cause of asthma has been hard to diagnose. Researchers have found out that the first step in an asthma attack is the airway inflammation which is an immune response. Along with this problem, extra mucus is produced and the air flow reduces dramatically. During the asthma attack, smaller airways get obstructed because of spasms of the smooth muscle in the walls of these airways. Figure 2.6 [13] shows the difference between a normal airway condition and the airway condition in an asthma attack.

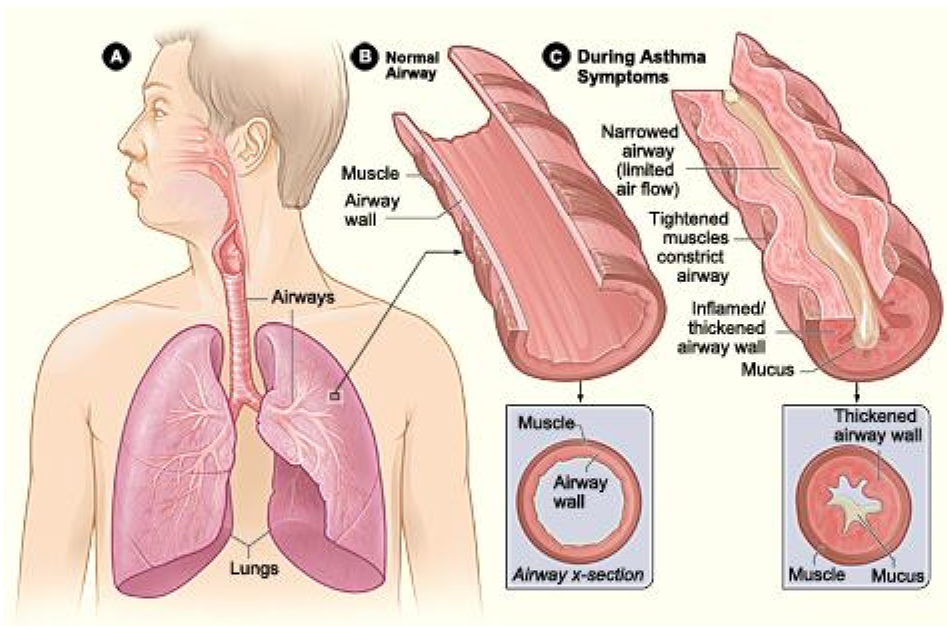


Figure 2.6: Differences between a normal airway and the airway in asthma attack [13]

The bronchiolar diameter reduces more during expiration rather than during inspiration in an asthma episode. So during the asthma attack, the functional residual capacity and residual volume of the lungs increase.

The symptoms of asthma vary in children and adults. Sometimes the symptoms of asthma are similar to other lung conditions and make it hard to diagnose, but early detection and proper treatments would be very helpful in lives of people with this disorder.

Lung Cancer: The incidence of lung cancer is associated with smoking since 90% of patients with this type of cancer are smokers. In a normal person, nasal hairs, sticky mucus and cilia play a good role in protecting the lungs from chemical and biological irritants, but in smokers these parts are overwhelmed and stop functioning properly. It will cause the production of more mucus and increases the risk of infections including pneumonia and COPD. Later on, like other cancers, the disorders lead to uncontrolled creation of cells. There are several types of lung cancers, but the most common ones are those involving the epithelial cells lining the bronchi and bronchioles [12].

Restrictive lung diseases: Another category of respiratory disease is restrictive lung diseases characterized by a loss of lung compliance causing incomplete lung expansion and increased lung stiffness. They may occur either because of an alteration in lung parenchyma or because of a disease of the pleura, chest wall, or neuromuscular apparatus [14]. In physiological terms, restrictive lung diseases are characterized by reduced total lung capacity (TLC), vital capacity, or resting lung volume. Accompanying characteristics are preserved airflow and normal airway resistance, and reduced functional residual capacity (FRC). If caused by parenchymal lung disease, restrictive lung disorders are accompanied by reduced gas transfer, which may be marked clinically by oxygen desaturation during and after exercise.

The many disorders that cause reduction or restriction of lung volumes may be divided into 2 groups based on anatomical structures.

The first is intrinsic lung diseases or diseases of the lung parenchyma. The diseases cause inflammation or scarring of the lung tissue (interstitial lung disease) or result in filling of the air spaces with exudate and debris (pneumonitis). These diseases can be characterized according to etiological factors. They include idiopathic fibrotic diseases, connective-tissue diseases, drug-induced lung disease, and primary diseases of the lungs (including sarcoidosis).

The second is extrinsic disorders or extraparenchymal diseases. The chest wall, pleura, and respiratory muscles are the components of the respiratory pump, and they need to function normally for effective ventilation. Diseases of these structures result in lung restriction, impaired ventilatory function, and respiratory failure (e.g., neuromuscular disorders).

2.3 PULMONARY FUNCTION TESTING TECHNIQUES

Lung function tests, or pulmonary function tests, evaluate how well the lungs work. They are a group of procedures that measure how much air the lungs can hold, what is the speed of inspiration and expiration, and how well the lungs exchange oxygen and carbon dioxide in blood. Pulmonary function tests can help doctors diagnose a range of respiratory diseases which might not be obvious to the doctor or the patient. The tests are also used to measure how a lung disease is progressing, and how serious the lung disease has become. Pulmonary function tests also can be used to assess how a patient is responding to different treatments. Generally, the following types of data could be acquired by using lung function tests [12]:

How much air volume can be moved in and out of the lungs

How fast the air in the lungs can be moved in and out

How stiff the lungs and chest wall are

How efficiently oxygen is transferred into the blood

The diffusion characteristics of the membrane through which the gas moves (determined by special tests)

How the lungs respond to physical therapy procedures of the chest area

There are two common techniques of pulmonary function tests, the *standard spirometry* and the *Impulse Oscillometry System (IOS)*. In the following subsection, both of these testing techniques will be explained. In this research, the IOS method has been used.

2.3.1 Spirometry

Spirometry is the most common pulmonary function test to measure the changes in the volume of lungs during breathing. A spirometer is the instrument that is used for a spirometry test. It is a volume recorder that can consist simply of an air-filled drum floating in a water-filled chamber, (figure 2.7) or currently more commonly, a flow-measuring device incorporating an integrating function. As the person breathes air in and out of the drum through a connecting tube, the resulting rise and fall of the drum are recorded as a spirogram, which is calibrated to the magnitude of the volume change. During exhalation, the light weight floating drum goes up and the pen goes down, marking a moving chart. First, normal

breathing can be seen and some of the test measurements can be obtained by normal, quiet breathing. Next, the person takes a maximal inspiration and follows this by a maximal expiration. The expired volume is the vital capacity (VC).

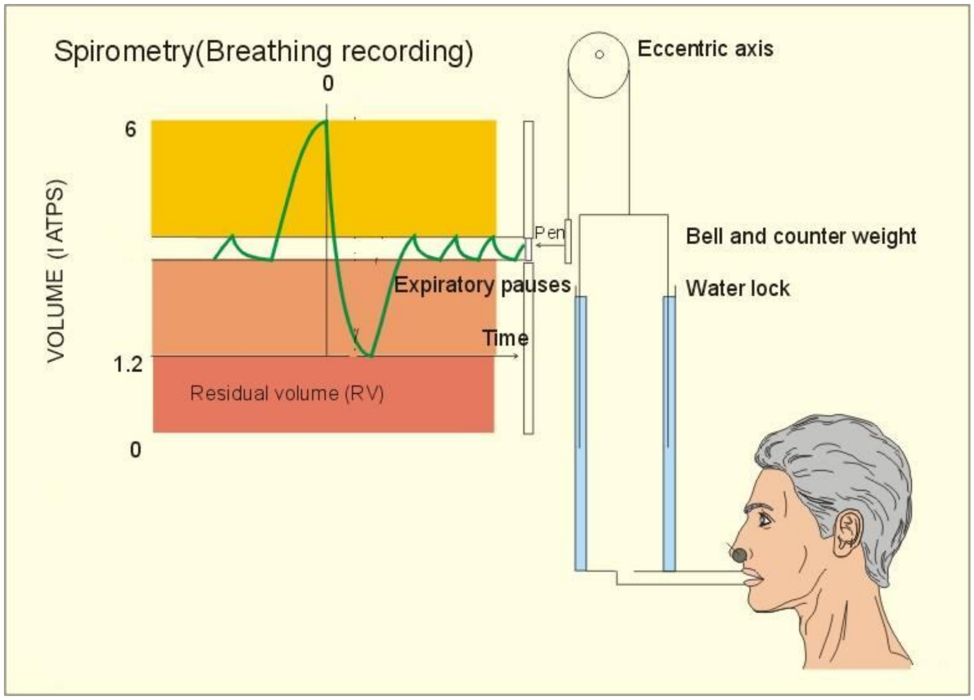


Figure 2.7: Sample of a Spirometry test [15]

As shown in Figure 2.7 [15], during the test, soft nose clips are used to prevent air escaping through the nose. Not shown is the usual filter mouthpieces used to prevent the spread of microorganisms.

Modern spirometers consist of pneumotachometers, which produce electronic signal outputs which can measure flow (volume per unit of time) as well [12].

They record, measure and assess the flow-volume loop and associated parameters. The evaluated data can be immediately printed out on a printer or saved to the internal memory. Modern spirometers are portable and handy devices that allow assessing the inspiratory and expiratory lung volumes including pre and post measurements with date and time. An interpretation program automatically can assess the measurement.

The general respiratory parameters that can be measured by spirometry are:

Forced vital capacity (FVC): The maximum volume of air, measured in liters during a rapid and forced expiration.

Forced expiratory volume in one second (FEV1): The volume of air the person exhales in the first second of a forced expiration.

FEV1 divided by FVC (FEV1/FVC): The proportion of air in the lungs, which can be exhaled in one second.

Other parameters that can be calculated from the spirogram are vital capacity (VC), tidal volume (Vt), breathing rate and ventilation rate (tidal volume x breathing rate). Spirometry measures vary in different cases based on age, height, sex and body size of subjects.

The information obtained from a spirometer is presented as a percent of that predicted for age, gender and height and are used in diagnosis of certain lung problems. To diagnose COPD, the FEV1 parameter is useful. When it is less than 70% predicted, it is a mild COPD, while less than 35% classifies it as severe COPD. When the FEV1/FVC parameter is less than 80% of a normal value, this may be indicative of expiratory flow obstruction. This problem happens in obstructive conditions such as asthma, chronic bronchitis and emphysema. In restrictive diseases such as pulmonary edema and fibrosis, FEV1/FVC is higher than normal.

In order to test the reversibility of an airway obstruction, clinicians may perform spirometry before and after inhalation of short-acting medication known as bronchodilator. The bronchodilator dilates the airways and reduces airflow obstruction allowing air to pass through freely. This is commonly referred to as a reversibility test, or a post bronchodilator test (Post BD), and is an important part in diagnosing asthma versus COPD. A post bronchodilator test is a good way of evaluating the amount of bronchoconstriction that was present in the airway, and how responsive a patient is to a bronchodilator medication [12].

With a simple spirometer, neither the FRC nor the residual volume can be measured. There is another way to measure them which is called the helium dilution method. In this method, the person is connected to a spirometer containing a known concentration of helium, which is almost insoluble in blood. After some breaths, the helium concentration in the spirometer and lung become the same. Since

no helium has been lost, the amount of helium present before equilibration equals the amount after it. In this way, the FRC volume is measurable.

There is another way to measure FRC by using a body plethysmograph device. It resembles a telephone booth, a large airtight box that the person sits in. After a normal expiration, a shutter closes the mouthpiece and the person is asked to take a deep inspiration. In the inhalation step, the person takes the air inside the box into his lungs so the lung volume increases and the box pressure rises since its volume is decreased. By using Boyle's law and knowing the pressure in the box before and after inspiration and the volume of the box before inspiration, the change in the volume of lungs can be calculated. With this information, FRC, which is the volume of lungs before inspiration, can be derived [16].

Although the results from spirometry are useful in studying lung function, a successful spirometric measurement requires maximal coordinated inspiratory and expiratory efforts from the patient. It is even harder for young children, aged people, and patients with severe problems in breathing. Since spirometry results are dependent on patient cooperation, FEV_1 and FVC can only be underestimated, never overestimated [17]. These issues can make disease diagnosis more difficult and less accurate when the clinician uses the Spirometry lung function test.

2.3.2 Impulse Oscillometry System (IOS)

The other technique, which requires minimum cooperation from the patient, is forced oscillation technique (FOT). It is less popular than Spirometry mainly due to the sophistication of its required mathematical and technical infrastructure. It was developed by Dubois in 1956. It evaluates the mechanical properties of the respiratory system and more specific information about resistive, inertive and elastic properties of the respiratory system can be obtained than is possible by a spirometric test [18]. It uses a superimposed pressure burst at the mouth to gain information about airflow obstruction by measuring the respiratory impedance (Z) and its two components, resistance (R) and reactance (X), over a wide range of frequencies. A user-friendly version of Dubios' FOT has been developed by Jaeger in Germany since 1980. This new technique is called the Impulse Oscillometry System (IOS) and measures different mechanical parameters of lungs. The IOS provides respiratory impedance with very precise

results. Even in the early stages of airflow obstruction, IOS is more sensitive to baseline abnormality than spirometry. Sensitivity of FOT to peripheral airway obstruction makes it a better way to diagnose peripheral airway diseases since spirometry mainly measures overall change of volume over time and does not provide a clear indication of *peripheral airway* obstruction. The IOS also provides useful guidance in clinical patient management [19], and it can be used to test older children and adults the same as younger children and aged people.

Figure 2.8: Position of the subject during the IOS test [20]

IOS does not use the respiratory muscles as the source of force. It uses rectangular pulse-flow signals as input for measurement of input impedance. When the test is being administered as shown in Figure 2.8 [20], the subject usually sits in an upright position with the nose clipped and lips firmly closed around the mouthpiece. The cheeks may or may not be supported with the hands.

IOS measurements yield frequency-dependent impedance values which may be correlated with respiratory system models consisting of electrical components that are analogous to the mechanical resistance, inertance and compliance of the central and peripheral airways. Parameter estimations of

these models can serve as reference values for the detection, diagnosis and treatment of various diseases/pathologies [21].

Since IOS is the main method used in this research, the next chapter will be the detailed description of this method.

Chapter 3: Impulse Oscillometry System (IOS)

3.1 INTRODUCTION

As discussed in the previous chapter, most of the methods of lung function testing provide lung measurements obtained during specific breathing action of the patient and usually demand maximum cooperation of the patient. In contrast, as a tool for the investigation of respiratory mechanics in clinical practice, forced oscillation technique (FOT) has the advantage of being a noninvasive, versatile and theoretically a well-supported method. FOT is a reliable method to obtain differentiated tidal breathing analysis. With FOT, more specific information of the resistive and elastic properties of the respiratory system can be obtained than is possible by spirometric test [18].

Spirometry is not a suitable method to recognize peripheral airway obstruction. The most important characteristic of FOT compared to spirometry is that it has greater sensitivity to peripheral airway disease.

This chapter reviews how forced oscillation techniques have been developed to measure the respiratory input impedance for use in clinical applications.

3.2 METHODOLOGY OF IMPULSE OSCILLATION TECHNIQUE

The forced oscillation technique (FOT) is a noninvasive, patient-friendly method with which measurements of respiratory mechanics are possible. FOT generates and superimposes small external pressure signals on the normal breathing of the subject and by measuring the resulting flow, it determines the mechanical properties of the respiratory system.

The first FOT was developed by DuBois in 1956 [22]. It used a superimposed pressure burst at the mouth of the subject to obtain information about airflow obstruction by measuring the respiratory impedance (Z) and its two components, resistance (R) and reactance (X). Since then, numerous variants of the FOT have been developed in terms of measurement configuration, oscillation frequencies and evaluation principles. Research and study continued till recently a commercial unit with improved technical capabilities emerged.

Impulse oscillation technique is one of the oscillometry methods that apply an impulse-shaped external pressure signal on the respiratory system of the subject. Other methods may use continuous, mono or multifrequency external signals [19].

The first and main benefit of applying impulse oscillation technique rather than other FOT pulses is that impulse oscillation allows the user to measure up to 10 impedance spectra per second. It gives a useful analysis of breath in different impedance. However, because of working with high impulse rates, recording longer respiratory time constants is not possible which could be more informative about respiratory abnormalities [19]. For this reason, recording of 5 impedance spectra per second is the common application of impulse oscillation technique. Another benefit of using impulse oscillation technique is its smaller and simpler electronic and mechanical structures that work with minimum power loss to generate the forced signals [19].

By applying the entire energy of impulse oscillation pressure to the respiratory system within a very short time, a higher impact to the respiratory system compared to sinusoidal or pseudo-random noise (PRN) applied pressures is obtained and it causes less inconvenience for the patients. This characteristic can be considered as the other benefit of using impulse oscillation technique [19].

3.2.1 Aperiodic waveforms of impulse oscillation method

The impulse generator produces the pressure pulses for 30 to 40 ms duration and with limited magnitude of 0.3 kPa. This produces an aperiodic pressure/flow signal (time course). The frequency domain representation of this aperiodic time course is a continuous spectrum after transforming it using a Fourier integral rather than a Fourier series. This transformation is then achieved by using the fast Fourier transform (FFT) algorithm in the IOS. Cyclic pressure impulse shaped waveforms with short duration (200 msec/cycle – 5 impulses/sec) cause cyclic flow signals to flow into the pulmonary system. For calculation of pulmonary system impedance, the average of a certain number of pressure and flow impulses is determined.

Figure 3.1 [19] represents data for spectra of respiratory resistance, reactance and coherence between 3 and 35 Hz for a normal adult during forced oscillation using generated impulse-shaped signals by the impulse oscillometry system (IOS). So the important advantage of continuous spectra is

that it has infinite frequency resolution and the abnormalities in the respiratory system can be recognized with the regional nonhomogeneities where resistance, reactance and coherence spectra show (Figure 3.1) deviations in the smooth and uniform continuous spectra. Figure 3.2 [19] shows the power spectra of flow and pressure generated by IOS in the frequency range of 0.1 to 35 Hz.

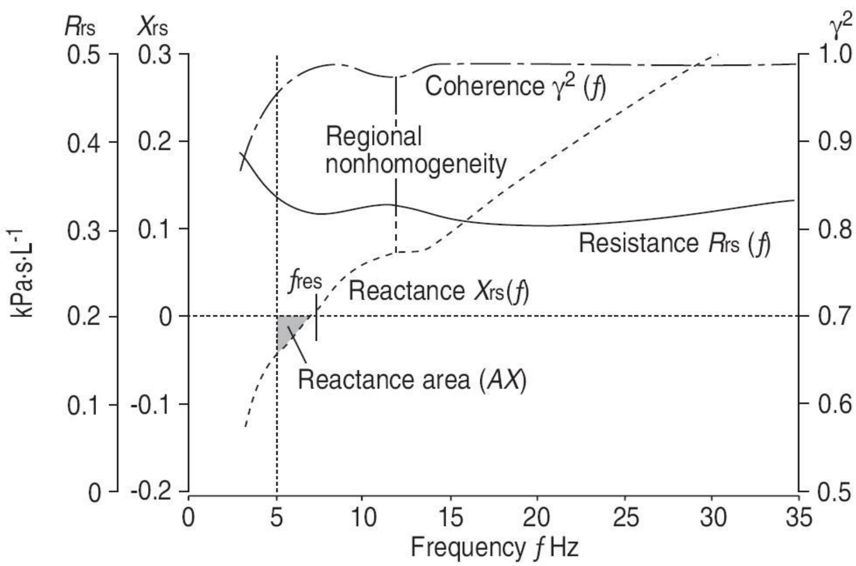


Figure 3.1: Representative data for spectra of respiratory resistance, reactance and coherence [19]

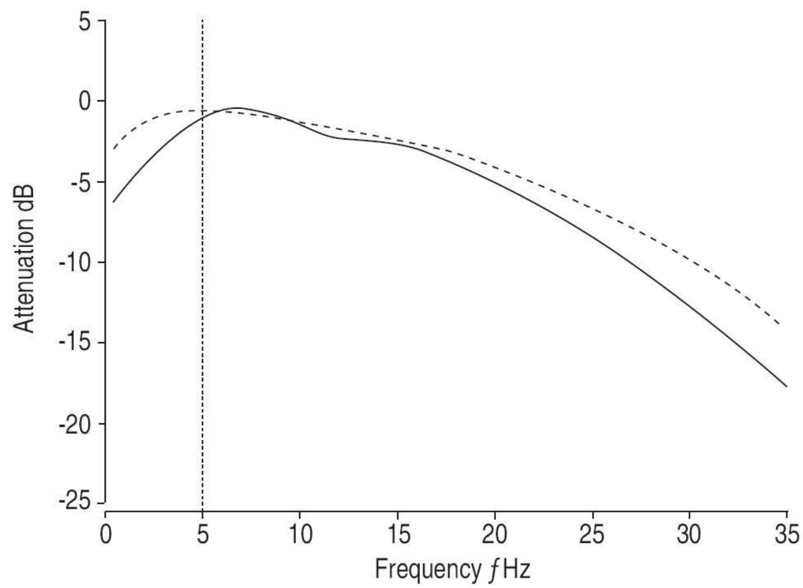


Figure 3.2: Power spectra of flow and pressure for impulse-shaped forcing signal generated by IOS [19]

3.2.2 Impulse oscillometry system

Schematic diagram of the impulse oscillometry system (IOS) is shown in Figure 3.3 [19]. The measuring-head of the IOS is functionally similar to Psuedo Random Noise (PRN) FOT systems. The IOS generates the recurrent aperiodic impulse-shaped forcing signals and this is its specific characteristics [19].

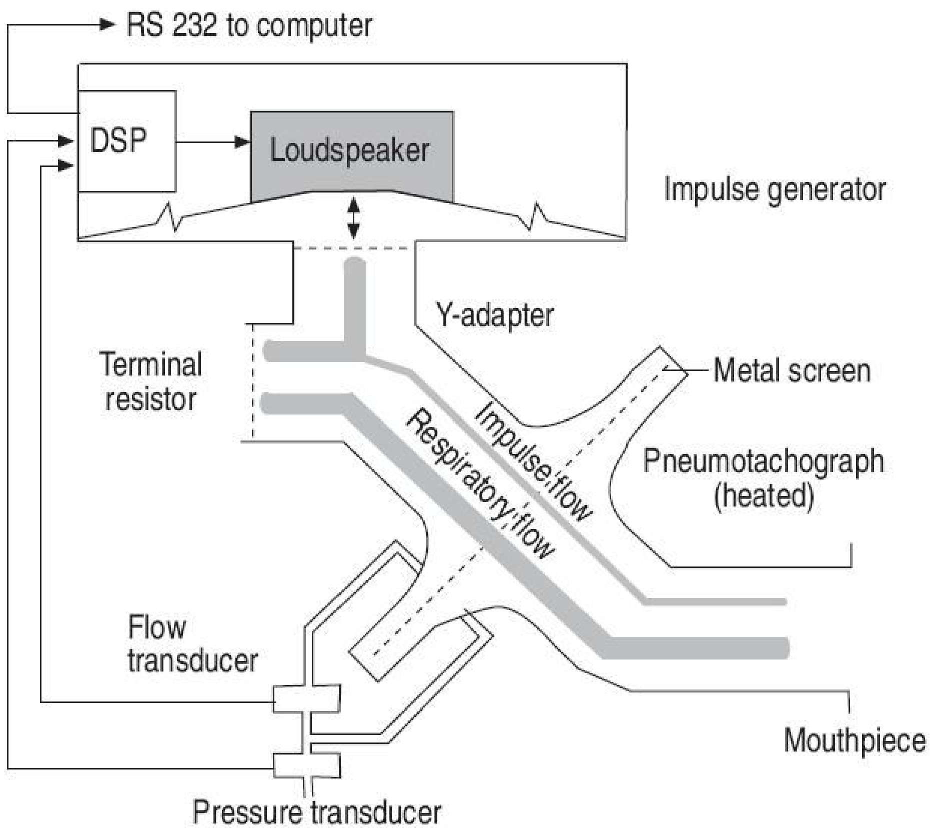


Figure 3.3: Schematic diagram of the IOS [19]

As described above in the IOS device, recurrent pressure impulses generate an oscillating flow, which is conducted to the airways. The device picks up the pressure and flow signals recorded by the pneumotachograph and isolates the signals produced by the oscillating flow. These signals are then used to calculate the respiratory impedance as described below [19].

There are two different channels to measure pressure and flow. In order to avoid phase differences in both of the channels, matched transducers of the same type have been used. Recording of

pressure and flow signals occurs at the 200 Hz frequency and then the signals are converted by a 12-bit analog-to-digital converter [19]. The IOS impulse generator can be tuned by both volume displacement of the loudspeaker membrane and magnitude of the terminating resistor.

The IOS performs recording the respiratory measurements as follows: the subject breathes spontaneously through the tubing between mouthpiece and terminating resistor. The loudspeaker generator transmits brief pressure impulses via the Y-adapter, pneumotachograph and mouthpiece into the respiratory tract. Then the signals containing both breathing activities and the forcing impulse signal are registered by pneumotachograph and pressure transducer [19].

3.3 INTERPRETATION OF OSCILLATION MECHANICS

The process of respiration can be described as two time-dependent variables, the rate of air flow (V) into the lungs and the air pressure (P) at the entrance of the respiratory system. The linear part of the response of the air flow rate to oscillatory perturbations of the external pressure defines a complex "impedance" (Z) as the relation between their complex amplitudes at a given angular frequency ω as:

$$Z(\omega) = Z_r(\omega) + j\, Z_x(\omega) \qquad (1)$$

FOT measures the impedance to 'forced' pressure oscillations produced by a loudspeaker. Both respiratory system flow and pressure, and the imposed forced oscillation signals are achievable from the flow and pressure channels.

By definition, respiratory impedance is the transfer function or ratio of pressure and flow, derived from the imposed forced oscillations, after being discriminated from underlying respiratory pressure and flow and their harmonics [19].

In other words, since Z embodies both the in-phase and out-of-phase relationships between pressure and airflow, it is a generalization of resistance. The in-phase component is called the real part of Z (or resistance: R), which describes the dissipative mechanical properties of the respiratory system. The out-of-phase relationship is expressed by the imaginary part (or reactance: X) which is related to the energy storage capacity and determined by the elastic properties dominant at low oscillation frequencies. The inertive properties become progressively more important with increasing the frequency ($\omega = 2\pi f$). Both R and X appear as functions of the frequency of oscillation [23].

In monofrequency oscillations, respiratory impedance includes airway resistance, as well as elasticity of lungs and chest wall at one oscillation frequency. In contrast, multifrequency oscillation methods provide measures of respiratory mechanical properties as a function of frequency and allow the recognition of characteristic respiratory responses at different oscillation frequencies. Most commonly, multifrequency oscillation methods use the oscillation frequency scale in the range of 5 to 30 Hz.

3.3.1 Respiratory resistance

The resistance component of respiratory impedance (R) includes central and peripheral airways, lung tissue and chest wall resistance. The lung tissue and chest wall resistance are usually negligible [19].

R is almost independent of oscillation frequency in healthy subjects but at higher frequencies it may increase a bit because of upper airways shunt effect [19]. When central or peripheral airway obstructions occur, R increases above normal values. The central airways obstruction increases R independent of oscillation frequency. In peripheral airways obstruction, R is highest at low oscillation frequencies and falls when the frequency increases [19]. This is called the frequency dependence of resistance.

From the mechanical point of view, respiratory resistance is the change in air pressure over flow. It can be expressed as voltage over current based on Ohm's law in electric world.

$$R = \Delta P / Q \qquad (2)$$

3.3.2 Respiratory reactance

The reactance component of the respiratory impedance is composed of the inertance (I), mass-inertive forces of the moving air in the conducting airways, and the capacitance (C), the elastic properties of lung periphery. With knowing that ω equals to $\omega=2.\pi.f$, the relation between the magnitudes of these parameters can be expressed as shown in the following formula.

$$X(f) = \omega.I - (1 / \omega.C) \qquad (3)$$

The inertance is related to the mass of air (pressure over change in flow with respect to time) in mechanical terms and it is equal to voltage over the changes of current from an electrical point of view.

$$I = \Delta P / (dQ/dt) \qquad (4)$$

$$L = V / (di/dt) \quad (5)$$

The compliance (in mechanical terms) is the volume increment divided by the pressure increment and, in electrical terms, it is the accumulated electrical charge divided by changes in voltage.

$$C = Volume / \Delta P \quad (6)$$

$$C = Q / \Delta V \quad (7)$$

The range of oscillation frequency influences X. Low frequency capacitive X expresses that the respiratory tract has the ability to store energy in the lung periphery [19].

Low oscillation frequencies reflect both small airways as well as large airways, while high oscillation frequencies reflect only larger airways. The magnitude of low-frequency reactance is increased in COPD due to functional peripheral airway obstruction, with resultant contraction of surface area of the lung periphery exposed to low-frequency oscillations. Indeed, low-frequency reactance is more sensitive to peripheral airway obstruction in COPD/emphysema than resistance. In the presence of peripheral airway obstruction in patients with COPD, relatively small increments to airways resistance may occur because of the large cumulative cross-sectional diameter of all airways in the lower generations of airways [19].

Capacitive and inertive elements of respiratory impedance can be analyzed with respiratory system equivalent electrical circuit models. Some different models of varying complexity and fidelity have been developed to define an approximation of a normal X spectrum with a zero-crossing exactly at resonant frequency and positive slope throughout. Parameter estimates for these models can be used as reference values for detection and diagnosis of different respiratory pathologies.

3.3.3 Other components of IOS

There are other parameters driven from IOS including resonant frequency (Fres), reactance area (AX), and frequency dependence of resistance. Resonant frequency is defined as the point at which the magnitudes of capacitive and inertive reactances are equal or the point at which the reactance is zero during the test. Reactance area refers to the integrated low frequency respiratory reactance magnitude between 5 Hz and resonant frequency. This parameter provides a single quantity that reflects changes in the degree of peripheral airway obstruction and correlates closely with frequency dependence of

resistance. Frequency dependence of resistance, R3-R20 and R5-R20, are the characteristics for peripheral airway dysfunction [24].

3.4 RESPIRATORY IMPEDANCE MODELLING

IOS yields frequency-dependent impedance values in shape of curves that are visually analyzed to distinguish between a healthy respiratory system and a diseased one. These values can be deployed to develop mechanical and equivalent electrical circuit models to evaluate and quantify lung mechanics. In these equivalent models, various electric circuit models with lumped parameter components representing the respiratory system's resistances, inertances, and compliances have been studied over the years. These parameters have proven useful in respiratory system disease detection and diagnosis. The simplest are the well-known series Resistance-Capacitance (RC) and Resistance-Inductance-Capacitance (RIC) models, while some work has also been done on the DuBois model [25].

In our research group, six models have been examined using two different parameter estimation procedures applied to IOS data for adults diagnosed with various respiratory ailments. This effort to date, has established that the fitting error performance of the extended RIC (eRIC) model and the Augmented RIC (aRIC) model, ranked in the middle of a series of conventional models developed over the past decades. However, they also provide parameter estimates that are physiologically more realistic and in line with expected values [26]. In these models, R is usually measured in cmH2O/L/s or kPa/L/s, I is measured in cmH2O/L/s2 or kPa/L/s2, and C is measured in L/cmH2O or L/kPa.

3.4.1 RC model

The resistance of the airways and the compliance of the alveoli are modeled as a simple RC circuit as shown in Figure 3.4. The impedance of the circuit can be calculated with formula 8.

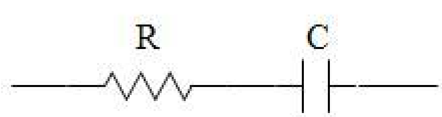


Figure 3.4. RC model

$Z = R - j/\omega.C$ (8)

where $j = \sqrt{-1}$.

3.4.2 RIC model

The three element circuit with lung inertance I included in the RC model as shown in Figure 3.5 and the formula 9 computes the impedance.

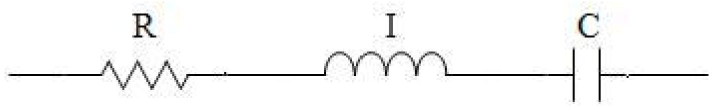


Figure 3.5. RIC model

$Z = R + j(\omega I - 1/\omega C)$ (9)

3.4.3 DuBois Model

In this model, the resistance is divided into the airway resistance (R_{aw}) and tissue resistance (R_t) and the inertance is divided into the airway inertance (I_{aw}) and the tissue inertance (I_t). Compliance is also divided into the tissue compliance (C_t) and the alveolar compliance (C_g) as shown in Figure 3.6.

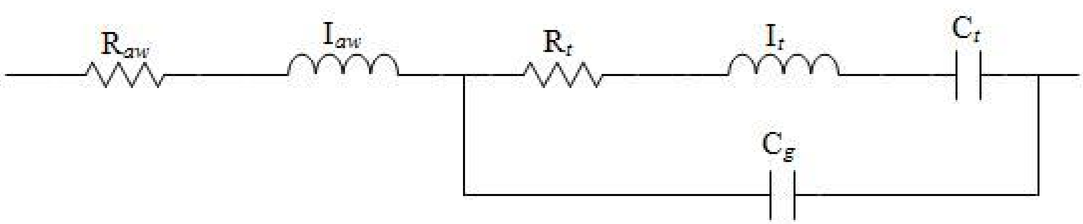


Figure 3.6. DuBois model.

The impedance of the circuit is:

$$Z = R_{aw} + j\omega I_{aw} + \frac{R_t C_t^2 \omega^2}{(C_g C_t I_t)^2\omega^6 + [C_g C_t(C_g C_t R_t^2 - 2I_t(C_g + C_t))]\omega^4 + (C_g + C_t)^2\omega^2} - \frac{j\omega[C_g C_t^2 I_t^2 \omega^4 - C_t(2C_g I_t + C_t I_t - C_g C_t R_t^2)\omega^2 + (C_g + C_t)]}{(C_g C_t I_t)^2\omega^6 + [C_g C_t(C_g C_t R_t^2 - 2I_t(C_g + C_t))]\omega^4 + (C_g + C_t)^2\omega^2} \quad (10)$$

3.4.4 Mead Model

In the Mead's model, the modeled respiratory parameters are the lung inertance (I), central resistance (R_c), peripheral resistance (R_p), lung compliance (C_l), chest wall compliance (C_w), bronchial tube compliance (C_b) and extrathoracic compliance (C_e). The circuit is shown in Figure 3.7.

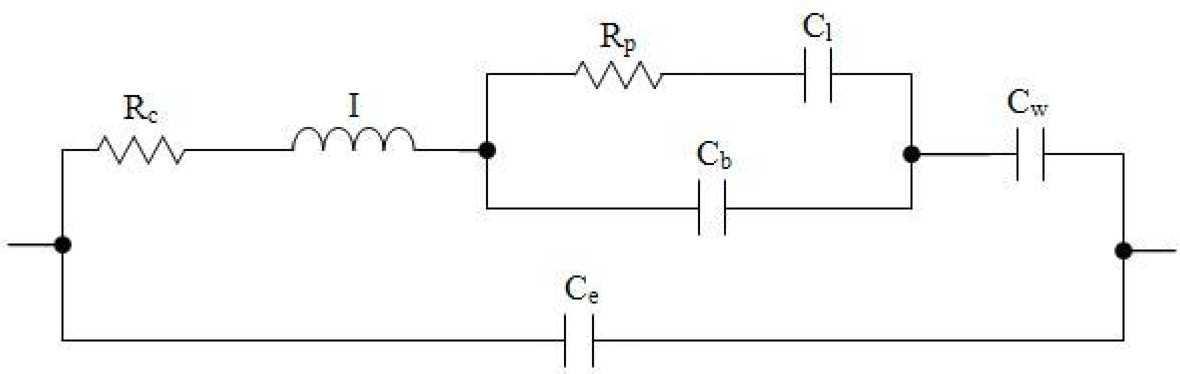


Figure 3.7. Mead model.

The impedance of the circuit is:

$$Z = - j/\omega C_e \parallel Z_m \qquad (11)$$

where

$$Z_m = \frac{R_p C_l^2}{\omega^2 R_p^2 C_b^2 C_l^2 + (C_b + C_l)^2} + R_c$$

$$+ j\left(- \frac{\omega^2 R_p^2 C_b^2 C_l^2 + C_b + C_l}{\omega[\omega^2 R_p^2 C_b^2 C_l^2 + (C_b + C_l)^2]} + \omega I - \frac{1}{\omega Cw}\right) \qquad (12)$$

$$= R_m + jX_m \qquad (13)$$

so

$$Z_R = Re(Z) = \frac{R_m}{1 - 2\omega C_e X_m + \omega^2 C_e^2 (R_m^2 + X_m^2)} \qquad (14)$$

$$Z_X = Im(Z) = \frac{X_m - \omega C_e (R_m^2 + X_m^2)}{1 - 2\omega C_e X_m + \omega^2 C_e^2 (R_m^2 + X_m^2)} \qquad (15)$$

3.4.5 Extended RIC Model

It was developed as an improvement of the RIC model. The additional peripheral resistance (R_p), connected in parallel with the compliance of peripheral airways, models the resistance presented by the respiratory system's small airways. This model can be considered to be a simplification of either the DuBois model (with It equal to zero and C_t equal to infinity) or the Mead model (with C_l and C_w equal to infinity and C_e equal to zero) [27]. The circuit of the extended RIC model is shown in Figure 3.8.

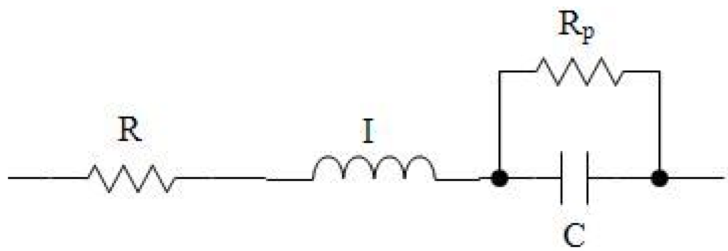


Figure 3.8. Extended RIC model.

The impedance of the circuit is:

$$Z = R + \frac{R_p}{1 + (\omega R_p C)^2} + j\left(\omega I - \frac{\omega R_p^2 C}{1 + (\omega R_p C)^2}\right) \qquad (16)$$

3.4.6 Augmented RIC Model

This model was proposed as an improvement to the extended RIC model and can be considered as a simplification of the Mead's model when lung and chest wall compliances are very, very large. The additional parameter (C_e) represents extrathoracic compliance, which is thought to increase the real part of the respiratory system's impedance at the higher frequencies due to upper airways shunt effects as observed in IOS data. The circuit of the augmented RIC model is shown in Figure 3.9 and its impedance is given by formula 17.

$$Z = \frac{A(RA+R_p)}{[A(1 - \omega^2 IC_e) + (\omega^2 R_p^2 CC_e)]^2 + [\omega C_e(RA + R_p)]^2} +$$

$$j \frac{\omega(IA - R_p^2 C)[A - \omega^2 C_e(IA - R_p^2 C)] - \omega C_e(RA+R_p)^2}{[A(1 - \omega^2 IC_e) + (\omega^2 R_p^2 CC_e)]^2 + [\omega C_e(RA + R_p)]^2} \qquad (17)$$

where $A = 1 + (\omega R_p C)^2$

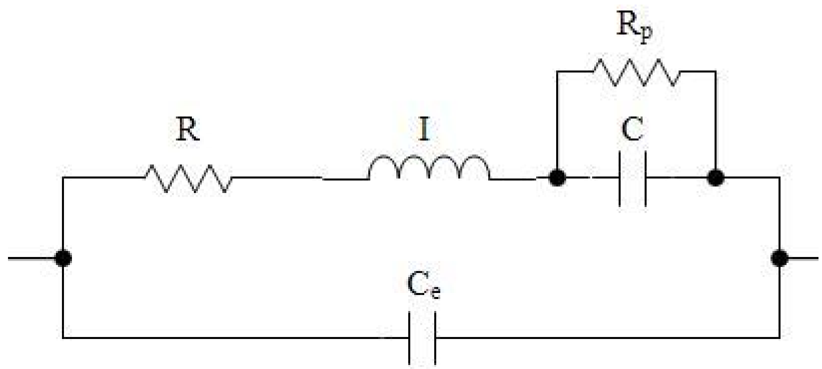


Figure 3.9. Augmented RIC model.

3.5 PARAMETER ESTIMATION TECHNIQUE

A brief characterization of a subject's lung properties can be provided by determination of the parameters of any model that best fit the behavior of it. This information can be then used in comparison with reference characteristics either to determine the presence of underdeveloped features, or the existence of pathological conditions [25].

As mentioned before, the two better models used in this research are eRIC and aRIC models. The average values of the IOS parameters (Resistance and Reactance at frequencies 5, 10, 15, 20, 25 and 35 Hz) were used to estimate the model parameters for these models [28].

The parameters for the aRIC model, R (resistance of the large airways), R_p (peripheral resistance), I (inertance of the air column), C_p (compliance of peripheral airways) and C_e (extrathoracic compliance), were estimated for all gathered data from tested subjects. The same procedure repeated to estimate the parameters of eRIC model (R, R_p, I and C). As an example, in table 3.1, the average values of resistance and reactance of lungs of a 13 years old girl with 60.5 inches height and 103.6 pounds weight before and after inhaling the bronchodilator in 5 to 35 Hz frequency range are shown.

Table 3.1: An example of respiratory system resistance and reactance in different frequencies recorded with IOS in pre- bronchodilation and post- bronchodilation

Test	**R_5**	**R_{10}**	**R_{15}**	**R_{20}**	**R_{25}**	**R_{35}**	**X_5**	**X_{10}**	**X_{15}**	**X_{20}**	**X_{25}**	**X_{35}**
PRE	0.59	0.54	0.49	0.43	0.42	0.52	-0.18	-0.07	-0.07	0	0.09	0.24

average												
POST average	0.44	0.41	0.38	0.36	0.37	0.44	-0.1	-0.01	0.01	0.07	0.14	0.27

In table 3.2, the parameter estimations for this IOS data set based on aRIC model and eRIC model are presented in tables 3.2 and table 3.3 respectively.

Table 3.2: Results of running augmented RIC parameter estimation model on the data sets given in table 3.1

Test	**R**	**R_p**	**I**	**C**	**C_e**
PRE average	0.34224	0.37215	0.001834	0.041998	0.002725
POST average	0.31821	0.23541	0.001531	0.10074	0.002406

Table 3.3: Results of running extended RIC parameter estimation model on the data sets given in table 3.1

Test	**R**	**R_p**	**I**	**C**
PRE average	0.4456	0.44093	0.001087	0.08616
POST average	0.38143	0.38788	0.001172	0.1735

Parameter estimation of a model is similar in concept to curve-fitting. Therefore, it is necessary to first select a suitable error criterion (E) that is to be minimized. The least squares (LS) criterion is by far the most commonly used for curve-fitting and parameter estimation. In this study, the least squares criterion has been used to minimize the sum of the squared errors between the measured IOS Z_R and Z_X data samples at 5, 10, 15, 20, 25 and 35 Hz, and the estimated resistive ($Z_{R,est}$) and reactive ($Z_{X,est}$) impedance values at the corresponding frequencies. The LS criterion was chosen for this work due to its relation with other system identification algorithms and its availability in different software packages. Equation 18 presents the LS equation.

$$E = \sum \{ [Z_R(f) - Z_{R,est}(f)]^2 + [Z_X(f) - Z_{X,est}(f)]^2 \qquad (18)$$

where f is 5, 10, 15, 20, 25 and 35.

The nonlinear LS was used for parameter estimation in these models because aRIC and eRIC impedance functions are nonlinearly dependent on these parameters. The optimization tool box in MATLAB software has this algorithm as a function, named “lsqnonlin”. For the RIC model, the linear LS algorithm (lsqlin function in MATLAB) can be used. The nonlinear LS algorithm is based on the interior-reflective form of Newton’s method. The converged upon solution of parameter estimates might only be locally minimal instead of being globally minimal. To address this issue, a procedure was employed in which for every estimation run, an initial parameter estimate vector was produced by a random number generator with values ranging uniformly from 0 to 5, to 0.5 and to 0.05 for the resistances, capacitances and inductance, respectively. This was then repeated at least forty more times per model for each set of test data to find the parameter estimates minimizing the error function (18), with the program stopping each time when the error value (E) changed by less than a factor of 10^{-9} from one iteration to the next. Such a procedure increases the likelihood that the algorithm converges to parameter estimates corresponding to a global error minimum. This LS error value at the end of running the algorithm provides an overall measure of the goodness of fit to the given test data for each model.

All IOS data used in this research have been reviewed and quality assured by our expert clinician in order to reject artifacts caused by air leaks, swallowing, breath holding or vocalization.

The estimated values for R, R_p, I, C and C_e parameters after running the estimating algorithm, and the four parameters, AX, Fres, R5 and R5-R15, that are direct data from IOS device, will be fed later to a neuro-fuzzy system to help the diagnosis of the problem in the tested subjects.

Chapter 4: CANFIS, a Neuro-Fuzzy Classification System

4.1 INTRODUCTION

It is not easy to classify pulmonary diseases by only observing the resistance and reactance curves measured by the IOS. In these types of problems, Artificial Neural Networks (ANNs) are excellent classifiers. They are particularly useful in cases where the problem to be solved is intractable and development of an algorithmic solution is difficult. This attribute of ANNs makes them particularly suitable to classify impulse oscillometric patterns in health and disease.

The implemented neuro-fuzzy classification system, which will be described in this chapter, is a computer-aided system that uses a combination of IOS measures (resistance and reactance) of lung function, estimated parameters of respiratory impedance models, and a set of fuzzy rules applied to a back propagation artificial neural network to classify respiratory patterns. It classifies the IOS test patterns acquired from a subject into one of 4 categories: asthma, mild small airway disease (mSAD), small airway disease (SAD) or normal. This classification system could create an efficient, robust and automatic means of airway disease recognition and diagnosis.

This chapter presents an overview of artificial neural networks and their applications, fuzzy logic and its implementations, and then a detailed description of the implemented neuro-fuzzy system.

4.2 ARTIFICIAL NEURAL NETWORKS

An artificial neural network (ANN) is a mathematical model or computational model to simulate the structure and/or functional aspects of biological neural networks. It consists of an interconnected group of artificial neurons and processes information using a connectionist approach to computation. In most cases an ANN is an adaptive system that changes its structure based on external or internal information that flows through the network during the learning phase. Neural networks are non-linear statistical data modeling tools. They can be used to model complex relationships between inputs and outputs or to find patterns in data.

ANNs are parallel distributed processing systems capable of solving complex classification problems by their remarkable ability to derive meaning from complicated or previously unseen data.

Today there is a tremendous amount of interest in ANNs also known as connectionist models, parallel distributed processing models and neuromorphic systems. Initially inspired by biological nervous systems, the development of ANNs has been motivated by their applicability to certain types of problems and their potential for parallel implementation. ANNs have been studied and developed for many years in the hope to achieve brain-like performance in signal processing and pattern recognition.

ANNs have been developed as generalizations of mathematical models of human cognition or neural biology based on the assumptions that [29]:

1- Information processing occurs in many simple elements called neurons.
2- Signals are passed between neurons over connection links.
3- Each connection link has an associated weight, which, in a typical neural net, provides a weighting factor for the transmitted signal.
4- Each neuron applies an activation function (usually nonlinear) to its net input (sum of weighted input signals) to determine its output signal.

4.2.1 Biological Neural Networks

The brain is made up of a huge network of nerve cells or neurons, which have the ability to gather and transmit information between one another through an electrochemical process. A biological neuron is made of three components: dendrites, soma, and axon. The dendrites receive signals from other neurons. The signals are in the form of electric impulses that are transmitted across a synaptic gap by means of a chemical process. The action of the chemical transmitter modifies the incoming signal in a manner similar to the action of the weights in an artificial neural network.

A typical biological neuron is shown in Figure 4.1 [12], together with axons from two other neurons from which the illustrated neuron could receive signals and dendrites for two other neurons to which it could send signals.

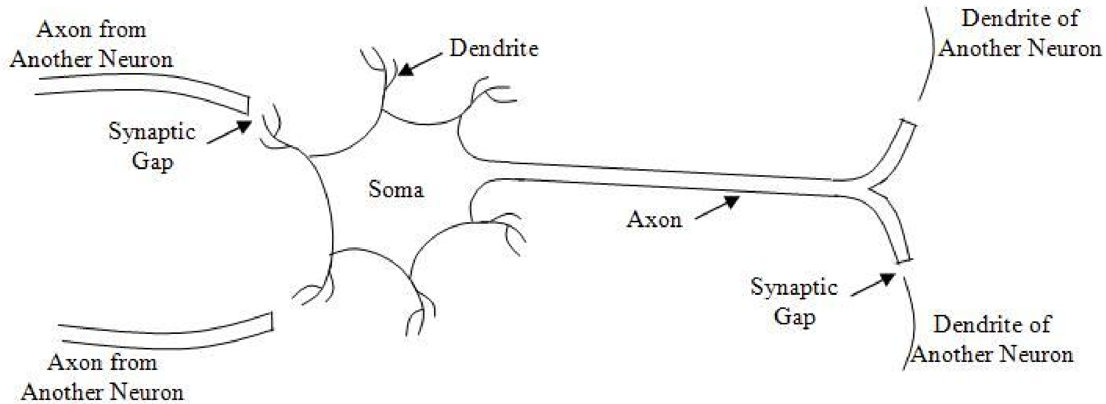


Figure 4.1: Biological Neuron [12]

The cell body of a neuron sums the incoming signals at a time and when a sufficient input is received, the cell transmits a signal over its axon to other cells connected to it. An action potential, which is the result of differential concentrations of ions on either side of the neuron's axon sheath, transmits the signal from a particular neuron. The involved ions are mostly potassium, sodium, and chloride.

The ANNs have been built based on several main features of biological neural networks including [12]:

1- The neuron receives many signals.
2- Signals may be modified by a weight at the receiving synapse.
3- The processing element of neuron sums the weighed inputs.
4- Based on receiving a sufficient input, the neuron transmits a single output.
5- The output from a particular neuron may go to many other neurons (the axon branches).
6- Information processing is local.
7- Memory is distributed: Long-term memory resides in the neurons' synapses or weights and short-term memory corresponds to the signals sent by the neurons.
8- A synapse's strength may be modified by experience.
9- Neurotransmitters for synapses may be excitatory or inhibitory.

In addition to these features, ANNs get their fault tolerance characteristics form biological neural networks. Similar to biological networks, ANNs are able to identify input signals and they can be retrained in cases of significant damage (like loss of data and some connections).

4.2.2 Structure of an Artificial Neural Network

Similar to biological neural networks, ANNs consist of a large number of neurons that are the simple processing elements in a neural network. They have also been called units, cells or nodes. Each neuron performs a simple operation to compute its output (or activation) from its inputs and is connected to other neurons by its weighted links. Each neuron has an internal state called its activation function or activity level, which is a function of its inputs. Typically, a neuron sends its activation as a signal (or input) to other neurons. It is important to note that a neuron can send only one signal at a time, although the signal is broadcast to several other neurons [29]. The information from one neuron to another is transmitted by the links between them. Each link has a weight. These weights are adjusted based on information from a set of data called the training data that include various input conditions and corresponding desired outputs. During the training phase, for a particular input, these weights are adjusted to minimize the difference between the ANN's output and the desired output specified in the training data for the same input. After training, the entire set of weights in an ANN corresponds to a mapping function in the input space that can be used to obtain the proper output for the future, potentially previously unseen, input data sets [30].

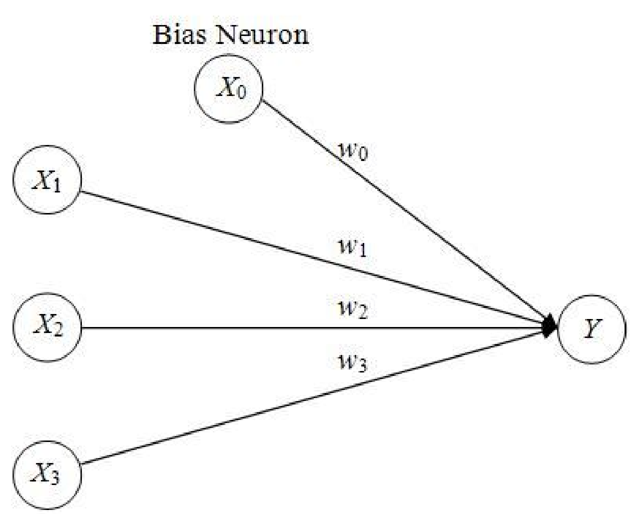


Figure 4.2: A simple artificial neuron [12]

As shown in Figure 4.2 [12], which is a simple artificial neuron, the neuron Y receives inputs from the bias neuron X_0 and the neurons X_1, X_2, and X_3. The activations (output signals) of these neurons are x_0, x_1, x_2, and x_3, respectively and the weights on the connections from these neurons to neuron Y are w_0, w_1, w_2, and w_3, respectively. The input of neuron Y (y_in) in this example is the sum of the weighted signals from neurons X_0, X_1, X_2, and X_3.

$$y\_in = w_0x_0 + w_1x_1 + w_2x_2 + w_3x_3 \qquad (1)$$

The bias neuron (x_0) always equals 1 and allows its weight to act as an adaptive bias. In Figure 4.3 [12], the neuron Y is connected to neurons Z_1 and Z_2, with weights v_1 and v_2. Neuron Y sends its output signal y to each of these neurons. The values received by neurons Z_1 and Z_2 would be different because each output signal is scaled by the appropriate weight, v_1 or v_2. In a typical net, the activations z_1 and z_2 of neurons Z_1 and Z_2 would depend on inputs from several or even many neurons, not just one, as shown in this simple example [12].

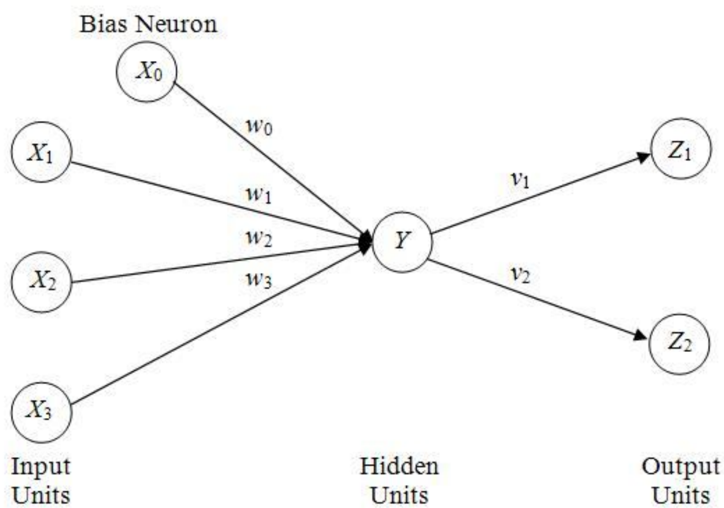


Figure 4.3: A very simple neural network [12]

Although the neural network shown in Figure 4.3 [12] is very simple, but it shows the main difference between an ANN and a normal input-output system. ANNs can solve many more problems

that can't be solved by a net with only input and output units, which can only solve problems with linear solutions.

4.2.3 Characteristics of Artificial Neural Networks

A neural network is characterized by its architecture, which is the pattern of connection between the neurons that make up the network, its training or learning algorithm, that sets the values of the weights of the links, and its activation function.

4.2.3.1 Architecture

The architecture of an ANN is the way in which the neurons connect together to make the layers of the network and defines the connection patterns between different layers. The simple ANN shown in Figure 4.3 [12] consists of three layers: input layer, output layer, and hidden layer. Usually the main difference between input-output nets and ANNs is the hidden layer where the internal neurons have been built and linked to each other in a specific way. Architectures of ANNs are usually grouped under two categories; the feed-forward architecture and the feedback architecture.

Feed-forward ANNs, as shown in Figure 4.4 [12], allow signals to travel one way only; from input to output. Therefore the output of each layer does not affect that same layer. In a fully interconnected net, each neuron in one layer is connected to every neuron on the next layer and there is no connection between the neurons of the same layer. Feed-forward ANNs tend to be straight forward networks that associate inputs with outputs. Feed-forward ANNs are the most widely used ANNs. As described before, this ANN organizes its neurons into three layers: the input layer, the hidden layer and the output layer. Each neuron in the input layer receives external input signal and transmits it to all neurons in the hidden layer through links. Similarly, neurons in the hidden layer receive input signals from the input layer and send the processed signal to each neuron in the output layer. Obviously, neurons in the output layer generate the ANN's output by combining signals transmitted from the hidden layer [31].

Feed-forward networks are commonly used for classification by varying the number of nodes in the hidden layer, the number of layers, and the number of input and output nodes. They are extensively used in pattern recognition. This type of setting is also referred to as bottom-up or top-down.

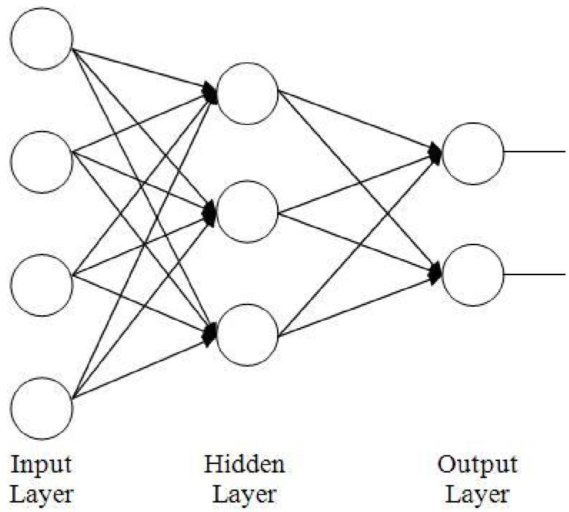


Figure 4.4: Feed-Forward Network [12]

Feedback (or recurrent) ANNs can have signals travelling in multiple directions by introducing loops in the network so the neurons are interconnected in different manners. Feedback networks are very powerful and can get extremely complicated. One example of a fully interconnected recurrant network of a single layer is the Hopfield network, illustrated in Figure 4.5. The feedback feature in this single-layer network effectively makes this type of network function as an ANN memory. Feedback networks are dynamic, which means that their state is changing continuously until they reach an equilibrium point. They remain at the equilibrium point until the input changes and a new equilibrium needs to be found.

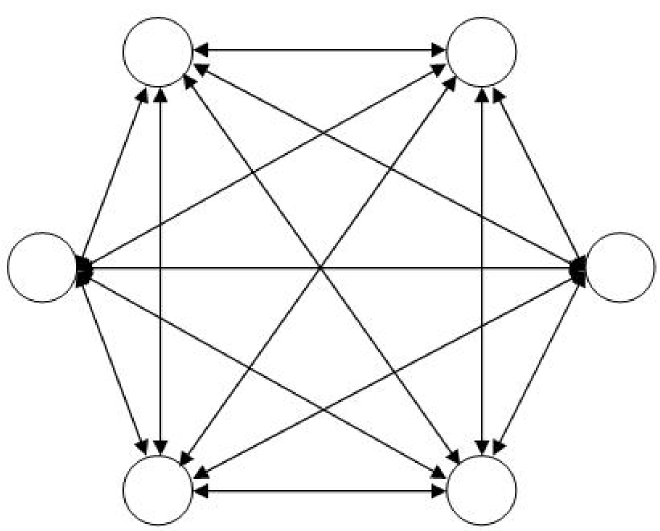


Figure 4.5: Recurrent Network [31]

Feedback architectures are also referred to as interactive or recurrent, although the latter term is often used to denote feedback connections in single-layer settings. Figure 4.5 [31] shows a simple example of a feedback (Hopfield) ANN.

4.2.3.2 Activation Function

After adding the weighted input signals in a neuron of an ANN, the neuron should apply an activation function to produce its output. In fact, the behavior of a neural network depends on these basic operations in any neuron of an ANN. The activation function controls the amplitude of the output of the neuron. Typically, a nonlinear activation function is used to achieve the advantage of multi-layer nets and the activation function used for all neurons in a particular layer is usually the same [29].

There are many easy-to-define activation functions used in an ANN. According to [29] some of the common activation functions are:

Identity Function: The identity function is usually used on the neurons in the input layer of multi-layer networks. So the output activity is proportional to the total weighted output, and it is expressed as:

$$F(x) = x \qquad \text{for all } x. \qquad (2)$$

Threshold (binary step) Function: In this function, the output is set at one of two levels, depending on whether the total input is greater than or less than some threshold value (θ). Single-layer networks often use this activation function to convert the net input to an output unit that is binary or bipolar signal based on the threshold. It is expressed as:

$$F(x) = \begin{cases} 1 & \text{if } x \geq \theta \\ 0 & \text{if } x < \theta \end{cases} \qquad (3)$$

Binary Sigmoid function: It is a common type of sigmoid functions (S-shaped curves). The output of this function has a range of 0 to 1, and is often used as the activation function for neural nets in which the desired output values happen in binary form or in the interval between 0 and 1. It is also called the logistic function. In the neural networks with backpropagation training algorithm, binary

sigmoid function is used as the activation function since the relationship between the value of the function at a point and the value of the derivative at that point reduces the computational burden during training [12]. This function is expressed as:

$F(x) = 1 / [1 + \exp(-\sigma x)]$ (4)

where σ is the steepness parameter.

Bipolar Sigmoid Function: Similar to binary sigmoid function, it is a logistic function that can have the output values in any range appropriate for the given problem. Common range of values is from -1 to 1 which is called the bipolar sigmoid.

4.2.3.3 Training algorithms

The training or learning algorithm is the strategy in the training phase of an ANN. In the training phase, if the network does not generate the desired output, the parameters and weights in the network should be changed in order to obtain the proper output. If the proper output is not achievable, the process should be continued until the network converges. In other words, the training algorithm is the ability to improve the network's performance for a given problem.

The type of training algorithm depends on the type of problem to be solved. The three major types of training algorithms are: supervised, unsupervised, and reinforcement learning [29].

Supervised learning algorithms use the training data to train the ANN. An algorithm in this category generates outputs based on the input data and if the resulting output is not close enough to the desired output, the parameters are changed. This process continues until the error between the ANN's output and the training data's target output reduces to a specific small value.

Unsupervised learning algorithms are similar to clustering techniques [31]. This type of learning algorithm groups similar input vectors together without the use of training data, and no target vectors are specified. The net modifies its weights so that the most similar input vectors are assigned to the same output (or cluster) unit. The neural net will produce a vector for each cluster formed [29].

Reinforcement learning algorithm uses a trial and error approach to get to desired outputs. A set of training data is provided to the net and if the calculated output is not close enough to the desired output, the feedback will be fed into the net and some random values will be chosen as weights to the

links and the process will continue. The feed-back in this kind of algorithm is not as informative as the feedback in supervised learning.

The developed artificial neural network to classify the pulmonary system diseases uses a specific type of supervised learning algorithm named backpropagation (BP) for training the ANN of this problem. This algorithm has been implemented on a six layer feed-forward network whose activation function is a binary sigmoid function and would be presented in next chapter.

In this algorithm, the total squared error of the output computed by the network in two continuous steps should be minimized. So, the errors propagate backwards from the output neurons to the hidden neurons to calculate the desired output. Figure 4.6 [12] shows a feed-forward neural network with one hidden layer that utilizes the BP training algorithm. This figure just shows the direction of information flow for the forward pass of operation. During the backward pass, signals are sent in the reverse direction. The output neurons (the Z neurons) and the hidden neurons may have bias neurons. The binary sigmoid function was chosen as the activation function because it is continuous, differentiable, and monotonically non-decreasing. The binary sigmoid function has all of these attributes.

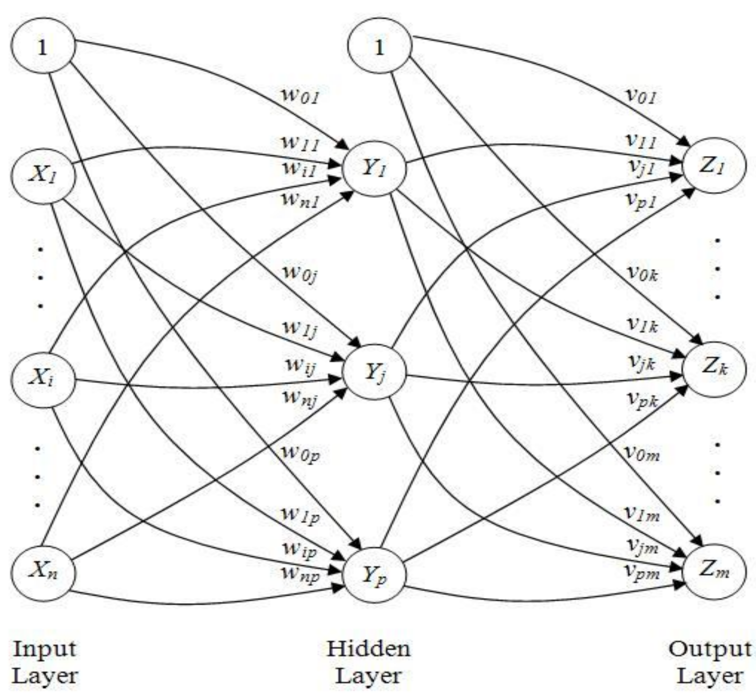


Figure 4.6: Neural network with one hidden layer [12]

In this network, each neuron in the hidden and output layers performs a simple two-step operation. According to [31], Figure 4.7 [12] shows the computation of a neuron in a neural network that utilizes BP training. In the first step, neuron Y_j calculates a weighted sum of all signals on its input. In the second step, the calculated sum is fed to the activation function to produce the output signal of the neuron. The final output signal of a neuron is expressed as:

$$y_j = F(s_j) = F(w_{0j}x_0 + \Sigma w_{ij} \times x_i) \quad i = 1, 2, \ldots, n \qquad (5)$$

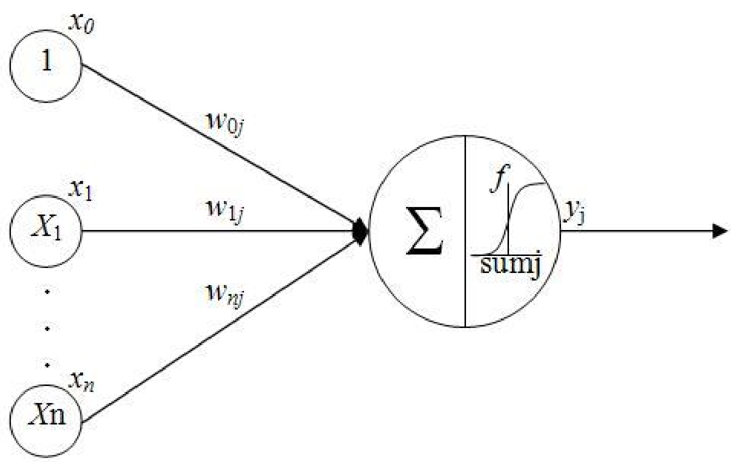


Figure 4.7: The Computation of a Neuron [12]

The detailed BP training algorithm used in this research will be described in the next chapter.

4.3 FUZZY LOGIC AND FUZZY INFERENCE SYSTEMS

Fuzzy logic is a form of multi-valued logic derived from fuzzy set theory to deal with reasoning that is approximate rather than precise. In contrast with binary logic, the fuzzy logic variables may have a membership value of not only 0 or 1, but also a value between 0 and 1.

Fuzzy inference is the process of formulating the mapping from a given input to an output using fuzzy logic. The mapping then provides a basis from which decisions can be made, or patterns discerned. In order to understand fuzzy inference systems, a brief summary about fuzzy sets, membership functions, logical operations, and If-Then rules is necessary.

4.3.1 Fuzzy Sets

Fuzzy sets are sets whose elements have degrees of membership. In classical set theory, an element either belongs or does not belong to a set. By contrast, an element can be a member of one or more sets with a degree of belonging or membership. According to this definition, a fuzzy set needs a function (membership function) that maps objects in a domain of concern to their membership value in the set [31].

The operations on fuzzy sets are similar to the basic operations on classical sets. The union of two fuzzy sets A and B is a fuzzy set C, which contains both A and B. It is expressed as C = A U B and the membership function of C is:

$$\mu_C(x) = \max(\mu_A(x), \mu_B(x)) = \mu_A(x) \vee \mu_B(x) \qquad (6)$$

The intersection of two fuzzy sets A and B is a fuzzy set C, which is the largest fuzzy set contained in both A and B. This is expressed as C = A) B and the membership function of C is expressed as:

$$\mu_C(x) = \min(\mu_A(x), \mu_B(x)) = \mu_A(x) \wedge \mu_B(x) \qquad (7)$$

The complement of a fuzzy set A is a fuzzy set $\bar{A}$, which is expressed as $\bar{A} = \neg A$ or $\bar{A}$ = NOT A and the membership function of $\bar{A}$ is:

$$\mu_{\bar{A}}(x) = 1 - \mu_A(x) \qquad (8)$$

4.3.2 Membership Functions

Membership functions are the graphical representations of the degree of membership of an element to a fuzzy set. A membership function maps an element of a fuzzy set to the interval [0, 1]. The degree of membership of an element x to a fuzzy set A is expressed as $\mu A(x)$, where $0 \leq \mu A(x) \leq 1$. Membership function associates a weighting with each of the inputs that are processed, defines functional overlap between elements, and ultimately determines an output response.

There are different types of membership functions defined on fuzzy sets. Some of them were used in this research, in order to determine the optimum membership function that best classifies the data patterns. Gaussian, triangular and trapezoidal membership functions turned out to have given the best accuracy percentage among all the applied membership functions in this research.

4.3.2.1 Gaussian Membership Function

A Gaussian membership function is defined by:

Gaussian(x; c, σ) = exp – $(x - c)^2 / 2\sigma^2$ (9)

where the parameters c and σ control the center and width of the membership function. A plot of the Gaussian membership function is presented if Figure 4.8 [12]. Adjusting σ would change the width of the membership function while adjusting c changes the center of the membership function.

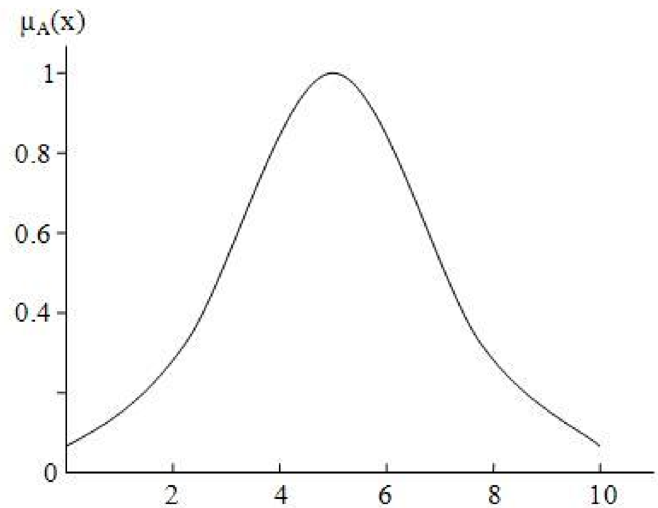


Figure 4.8: Gaussian(x; 5, 2.5) [12]

4.3.2.2 Triangular Membership Function

The Triangular membership function with straight lines can formally be defined by three parameters as follows:

$$\text{Triangle}(x; a, b, c) = \begin{cases} 0, & x \leq a \\ (x - a) / (b - a), & a \leq x \leq b \\ (c - x) / (c - b), & b \leq x \leq c \\ 0, & c \leq x \end{cases} \qquad (10)$$

where a, b and c with a < b < c represent the x coordinates of the three corners of the triangular membership function. One typical plot of the triangular membership function is given in Figure 4.9 [12]. Adjusting a and/or b changes the width of the membership function while increasing or decreasing parameter c shifts the membership function to the right or the left respectively.

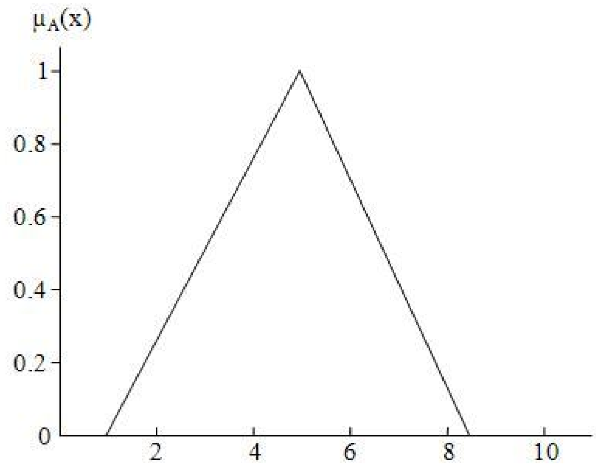


Figure 4.9: Triangle(x; 1, 5, 8.5) [12]

4.3.2.3 Trapezoidal Membership Function

A trapezoidal membership function is specified by four parameters {a, b, c, d}, and is expressed as:

$$\text{Trapezoid}(x; a, b, c, d) = \begin{cases} 0, & x \leq a \\ (x-a)/(b-a), & a \leq x \leq b \\ 1, & b \leq x \leq c \\ (d-x)/(d-c), & c \leq x \leq d \\ 0, & d \leq x \end{cases} \qquad (11)$$

where a, b , c and d with $a < b \leq c < d$ represent the x coordinates of the four corners of the trapezoidal membership function. An example of a trapezoidal function has furnished in Figure 4.10 [12]. When parameter b is equal to parameter c, the trapezoidal membership function reduces to a triangular membership function. Adjusting a and/or d is similar to adjusting a and/or b in the triangular membership function.

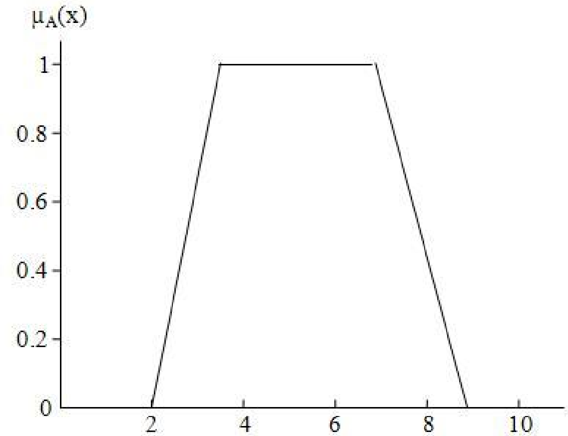


Figure 4.10: Trapezoid(x; 2, 3.5, 7, 9) [12]

4.3.3 Fuzzy Operations

Fuzzy intersection: The intersection on two fuzzy sets A and B is defined in general by a function T with the following definition [12]:

$\mu_{A\cap B}(x) = T(\mu_A(x), \mu_B(x)) = \mu_A(x) * \mu_B(x)$ (12)

where * is a binary operator for the function T. This class of fuzzy intersection operators is usually referred to as T-norm (triangular norm) operators and they meet the following basic requirements.

Boundary Requirement:	$T(0, 0) = 0, T(a, 1) = T(1, a) = a$
Monotonicity Requirement:	$T(a, b) \leq T(c, d)$ if $a \leq c$ and $b \leq d$
Commutativity Requirement:	$T(a, b) = T(b, a)$
Associativity Requirement:	$T(a, T(b, c)) = T(T(a, b), c)$

Fuzzy union: The union operator on two fuzzy sets A and B is defined in general by a function S with the following definition [12]:

$\mu_{AUB}(x) = S(\mu_A(x), \mu_B(x)) = \mu_A(x) + \mu_B(x)$ (13)

where + is a binary operator for the functions S. This class of fuzzy union operators is usually referred to as T-conorm (or S-norm) operators and they satisfy the following basic requirements.

Boundary Requirement:	$S(1, 1) = 1, S(0, a) = S(a, 0) = a$
Monotonicity Requirement:	$S(a, b) \leq S(c, d)$ if $a \leq c$ and $b \leq d$
Commutativity Requirement:	$S(a, b) = S(b, a)$

Associativity Requirement: $S(a, S(b, c)) = S(S(a, b), c)$

4.3.4 If-Then Rules

Processing the input data in a fuzzy system is accomplished through the fuzzy (If-Then fuzzy) rules. Fuzzy rules are in the form of "If A then B", where A is the antecedent and B is the consequent. The antecedent is a condition that if true, implies the consequent will be drawn as the conclusion or result. Fuzzy rules, therefore, associate a condition, described using linguistic variables and fuzzy sets, to a conclusion. The antecedent of a fuzzy rule may be a combination of some multiple simple conditions that make a complex condition. It can use three logic connectives between the simple conditions: AND (conjunction), OR (disjunction), and NOT (negation). As an example of a fuzzy rule, one of the Impulse Oscillometry system's condition and behavior can be expressed in the following fuzzy rule:

IF the airway resistance of a patient is Fair AND
(the peripheral resistance of the patient is High OR
the patient has a Moderate small airway inertance) AND
the alveoli compliance is NOT High
THEN the patient is asthmatic

where Fair, High and Moderate are fuzzy sets defined by any of the membership functions discussed in the previous subsections [12].

The consequent of a fuzzy rule can be classified into crisp, fuzzy, or functional consequent categories [31]. In general, the fuzzy rules with a crisp consequent are used more often because their processing procedure is more efficient. Averaging and weighting the resulting outputs from all the individual rules into one single output decision which decides what to do, is the process of defuzzification that produces the crisp value as output result.

4.3.5 Fuzzy Inference System

Fuzzy Inference Systems use fuzzy logic including all membership functions, If-Then rules and fuzzy operations to build a computing framework to solve different engineering problems in a wide variety of fields.

Each fuzzy inference system consists of three main subsystems including a rule base, a database, and a reasoning mechanism. The rule base contains all the rules that are needed to cover all different input conditions in order to reach the proper output of the system. The membership functions are defined in database and the reasoning mechanism is the approach to generating the output from the given input conditions. In Figure 4.11 [12] a block diagram of a basic fuzzy inference system is shown. In this block diagram, the input of inference system can be either a crisp or a fuzzy value and the output is a crisp value after the defuzzification process.

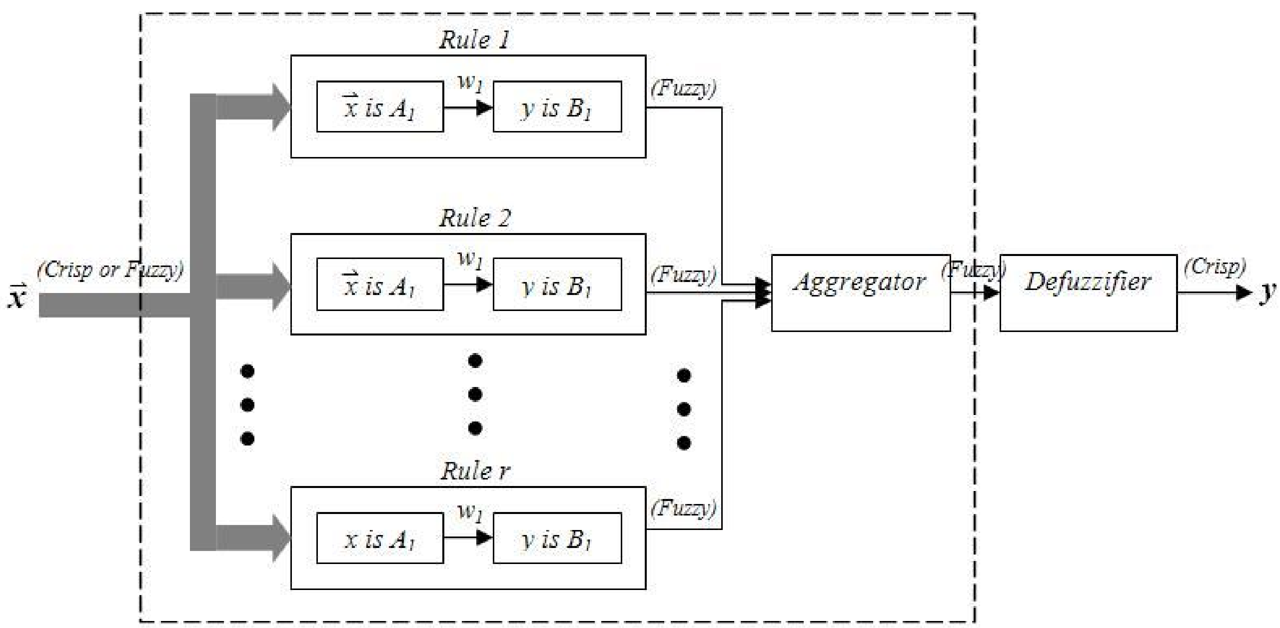


Figure 4.11: Block diagram for a fuzzy inference system [12]

According to [32], the steps of fuzzy reasoning performed by fuzzy inference systems are:

1- Compare the input variables with the membership functions on the antecedent part to obtain the membership values. This step is often called "fuzzification".

2- Combine the membership values through a specific operator on the antecedent part to get firing strength or weight of each rule.

3- Generate the qualified fuzzy or crisp consequent of each rule depending on the weight.

4- Aggregate the qualified consequents to produce a crisp output which is called "defuzzification".

4.3.6 Fuzzy Models

There are three types of fuzzy models which are more common in use. These models are Mamdani model, Tsukamoto model, and Sugeno model.

In Mamdani model, both input and output values are crisp. A simple one-input one-output fuzzy system with using Mamdani model can be expressed as follows:

$$\begin{cases} \text{IF x is High THEN y is High} \\ \text{IF x is Medium THEN y is Medium} \\ \text{IF x is Low THEN y is Low} \end{cases}$$

where X is a crisp input and Y is a crisp output.

In Tsukamoto model, the inferred output of each rule is defined as a crisp value induced by the rule's firing strength. For example:

$$\begin{cases} \text{IF x is High THEN y is } C_1 \\ \text{IF x is Medium THEN y is } C_2 \\ \text{IF x is Low THEN y is } C_3 \end{cases}$$

where High, Medium, Low, C_1, C_2, and C_3 are membership functions.

In Sugeno Fuzzy Model, a typical fuzzy rule is expressed as:

IF x is A AND y is B THEN z = f(x, y)

where A and B are fuzzy sets in the antecedent, while z = f(x, y) is a crisp function in the consequent. Usually, f(x, y) is a polynomial in the input variables x and y, or any function that can properly describe the output of the model within the fuzzy region specified by the antecedent of the rule. Since the only fuzzy part of the Sugeno model is in its antecedent, it is easy to demonstrate the distinction between a set of fuzzy rules and nonfuzzy ones. When f(x, y) is a first-order polynomial, the resulting fuzzy inference system is called a first-order Sugeno fuzzy model. The developed neuro-fuzzy classification system in this research is based on the first-order Sugeno fuzzy model because of its transparency and efficiency [12].

4.4 CANFIS: CO-ACTIVE NEURO-FUZZY INFERENCE SYSTEM

A nero-fuzzy system is a hybrid system generated by merging neural network and fuzzy logic technologies to incorporate the strengths of both. ANNs come with a learning algorithm and therefore, can learn from data and feedback, while fuzzy logic uses linguistic variables and if-then rules to develop fuzzy rule-based models that are easy to comprehend. Since neural networks can learn, their learning concepts are incorporated into fuzzy inference systems to produce neuro-fuzzy systems.

4.4.1 Neural Networks and Fuzzy Logic

A neuro-fuzzy network is a fuzzy inference system (FIS) in the body of an artificial neural network. Depending on the FIS type, there are several layers that simulate the processes involved in a fuzzy inference like fuzzification, inference, aggregation and defuzzification. Embedding an FIS in a general structure of an ANN has the benefit of using available ANN training methods to find the parameters of a fuzzy system.

4.4.2 Neuro-Fuzzy Architecture

Similar to ANNs, neuro-fuzzy systems consist of nodes and links but their architectures have some differences. The nodes and links in a neuro-fuzzy net correspond to a specific component such as the antecedent membership functions of the input variables in the first layer of the network. The other difference is that the nodes of a layer in a neuro-fuzzy net are not necessarily fully connected to all nodes of the adjacent layer. A neuro-fuzzy system usually consists of more layers than a typical ANN. A neuro-fuzzy architecture is shown in Figure 4.12 [12].

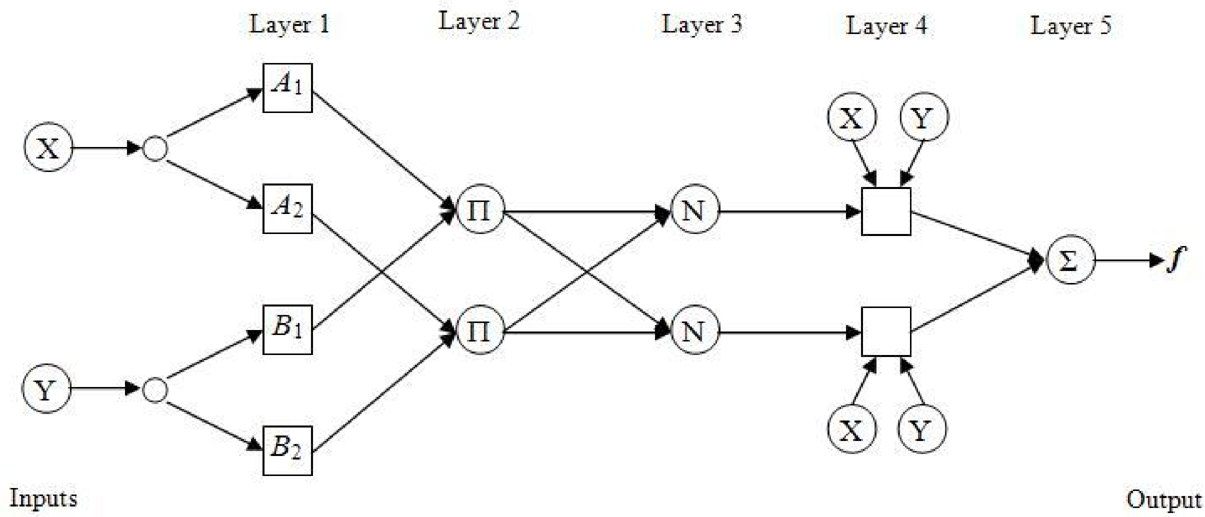


Figure 4.12: A neuro-fuzzy architecture [12]

According to [31], the functionalities associated with different layers of a neuro-fuzzy system which constitute fuzzy rule-based inference include the following:

- Compute the degree of membership to a fuzzy condition involving one input variable.
- Compute the degree of membership of a fuzzy condition involving multiple input variables.
- Compute the normalized degree of membership.
- Compute the conclusion inferred by a fuzzy rule.
- Combine the conclusion of all fuzzy rules in a model.

4.4.3 ANFIS: Adaptive Network-based Fuzzy Inference System

ANFIS, a multi-layer architecture, is a fuzzy inference system which provides a base to build a set of fuzzy if-then rules with appropriate membership functions to generate a specific output based on a specific input in a fuzzy system. As the CANFIS is an extended ANFIS model, in this section ANFIS model will be described.

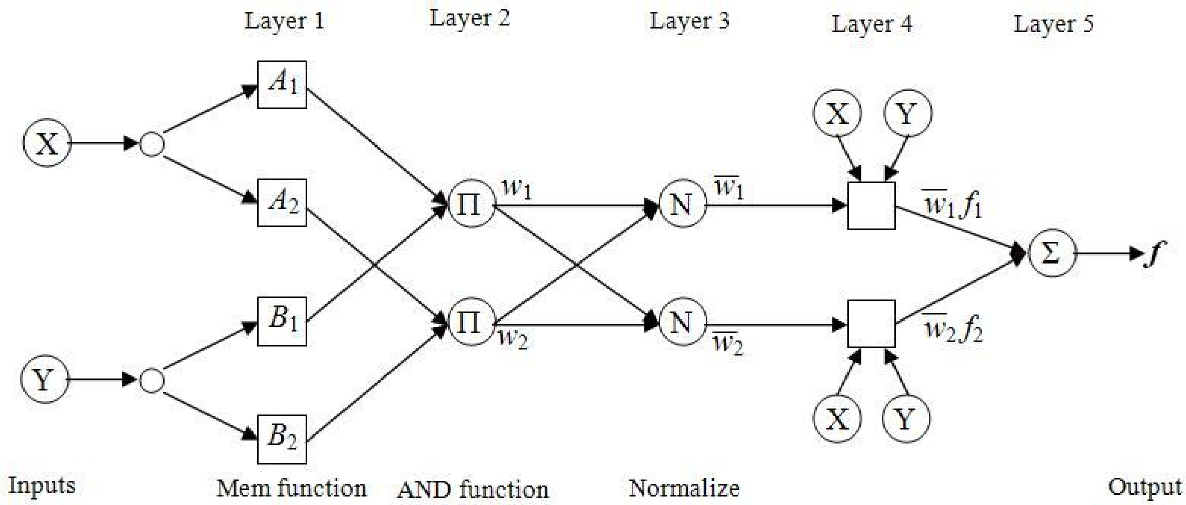


Figure 4.13: Two-input ANFIS architecture with two rules [12]

Figure 4.13 [12] shows ANFIS architecture with five layers. Assume that the fuzzy inference system has two inputs x and y and one output z. For a first-order Sugeno fuzzy model, a common rule set with two fuzzy if-then rules is the following [12]:

Rule 1: If x is A_1 and y is B_1, then $f_1 = p_1x + q_1y + r_1$

Rule 2: If x is A_2 and y is B_2, then $f_2 = p_2x + q_2y + r_2$

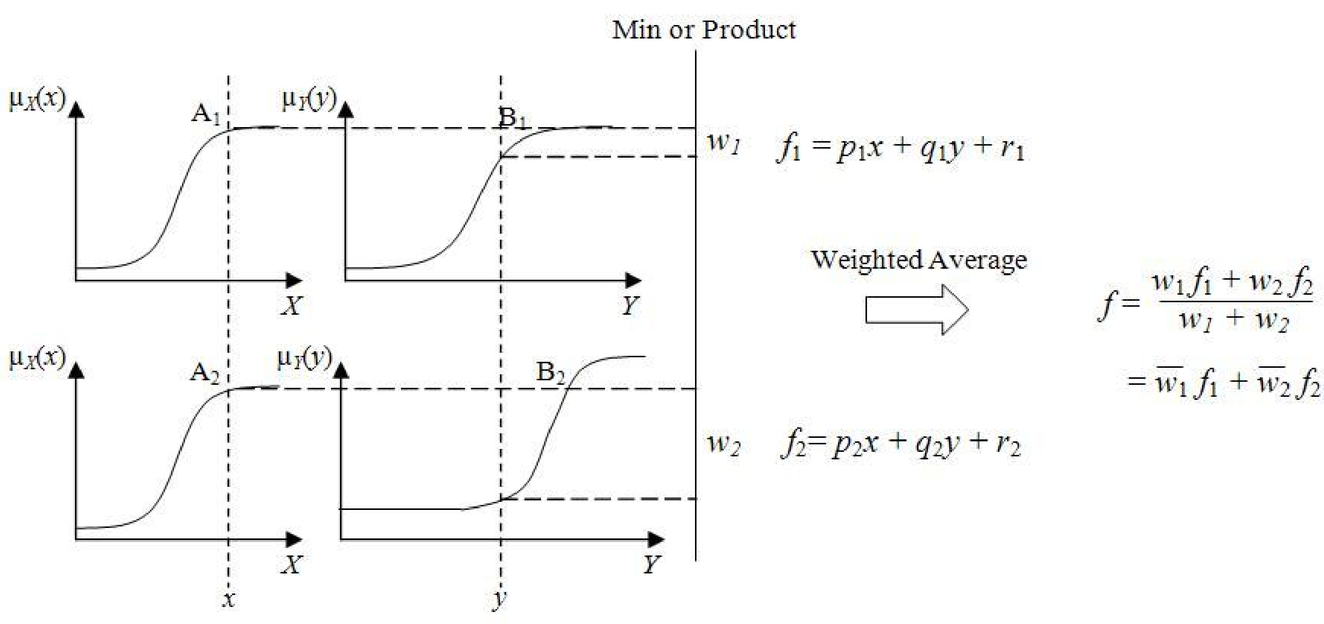


Figure 4.14: Two-input first-order Sugeno fuzzy model with two rules [12]

In Figure 4.14 [12] the reasoning mechanism for this Sugeno fuzzy model is shown. The functionalities of the nodes in the different layers of the ANFIS model shown in 4.13 are as follows (notice that the output of the ith node in layer l is denoted as $O_{l,i}$):

Layer 1: Every node i in this layer is an adaptive node with a node function:

$O_{1,i} = \mu A_i(x)$ for i = 1, 2 or (14)

$O_{1,i} = \mu B_i(y)$ for i = 3, 4 (15)

The output of the nodes in this layer is:

$O_{1,1} = \mu A_1(x)$

$O_{1,2} = \mu A_2(x)$

$O_{1,3} = \mu B_1(y)$

$O_{1,4} = \mu B_2(y)$

where x or y is the input to the nodes and A_1, A_2, B_1 and B_2 are linguistic variables such as “small” or “large” associated with each node. In other words $O_{1,i}$ is the degree of membership of a fuzzy set A (= A_1, A_2) or B = (B_1 or B_2) and it specifies the degree to which the given input x or y satisfies the quantifier A. Here, the membership function for A can be any appropriate parameterized membership function described in section 4.3, such as the generalized bell function:

$\mu A(x) = 1 / (1 + |(x - c_i)/ a_i|^{2b})$ (16)

where $\{a_i, b_i, c_i\}$ is the parameter set.

Layer 2: Every node in this layer is a fixed node labeled Π, whose output is the combination of all the incoming signals. The output of each node represents the firing strength of a rule. The output of the nodes in layer 2 of Figure 4.13 [12] is:

$O_{2,1} = w_1 = \mu A_1(x)\ \mu B_1(y)$

$O_{2,2} = w_2 = \mu A_2(x)\ \mu B_2(y)$

Layer 3: Every node i in this layer is a fixed node labeled N. The ith node calculates the ratio of the ith rule’s firing strength to the sum of all rules’ firing strengths as shown below.

$O_{3,1} = \overline{w}_1 = w_1 / (w_1 + w_2)$

$O_{3,2} = \overline{w}_2 = w_2 / (w_1 + w_2)$

The outputs of this layer are called normalizing firing strengths.

Layer 4: Every node i in this layer is an adaptive node with a node function:

$$O_{4,i} = \overline{w_i} f_i = \overline{w_i}(p_i x + q_i y + r_i) \quad (17)$$

The output of the nodes in layer 4 of Figure 4.13 [12] is:

$$O_{4,1} = \overline{w_1} f_1 = \overline{w_1}(p_1 x + q_1 y + r_1)$$

$$O_{4,2} = \overline{w_2} f_2 = \overline{w_2}(p_2 x + q_2 y + r_2)$$

where $\overline{w_i}$, is a normalized firing strength from layer 3 and $\{p_i, q_i, r_i\}$ is the parameter set of this node. Parameters in this layer are referred to as consequent parameters.

Layer 5: The single node in this layer is a fixed node labeled Σ, which computes the overall output as the summation of all incoming signals. The output of the node in this layer is:

$$O_{5,1} = \Sigma \overline{w_i} f_i = \Sigma_i w_i f_i / \Sigma_i w_i$$

$$O_{5,1} = \overline{w_1} f_1 + \overline{w_2} f_2$$

$$O_{5,1} = (w_1 f_1 + w_2 f_2) / w_1 + w_2$$

4.4.4 The Structure of CANFIS

CANFIS is an extended ANFIS model. The extension takes many of the advantages of the artificial neural networks and the linguistic interpretability of the fuzzy inference system. In the following section, this neuro-fuzzy inference system will be described.

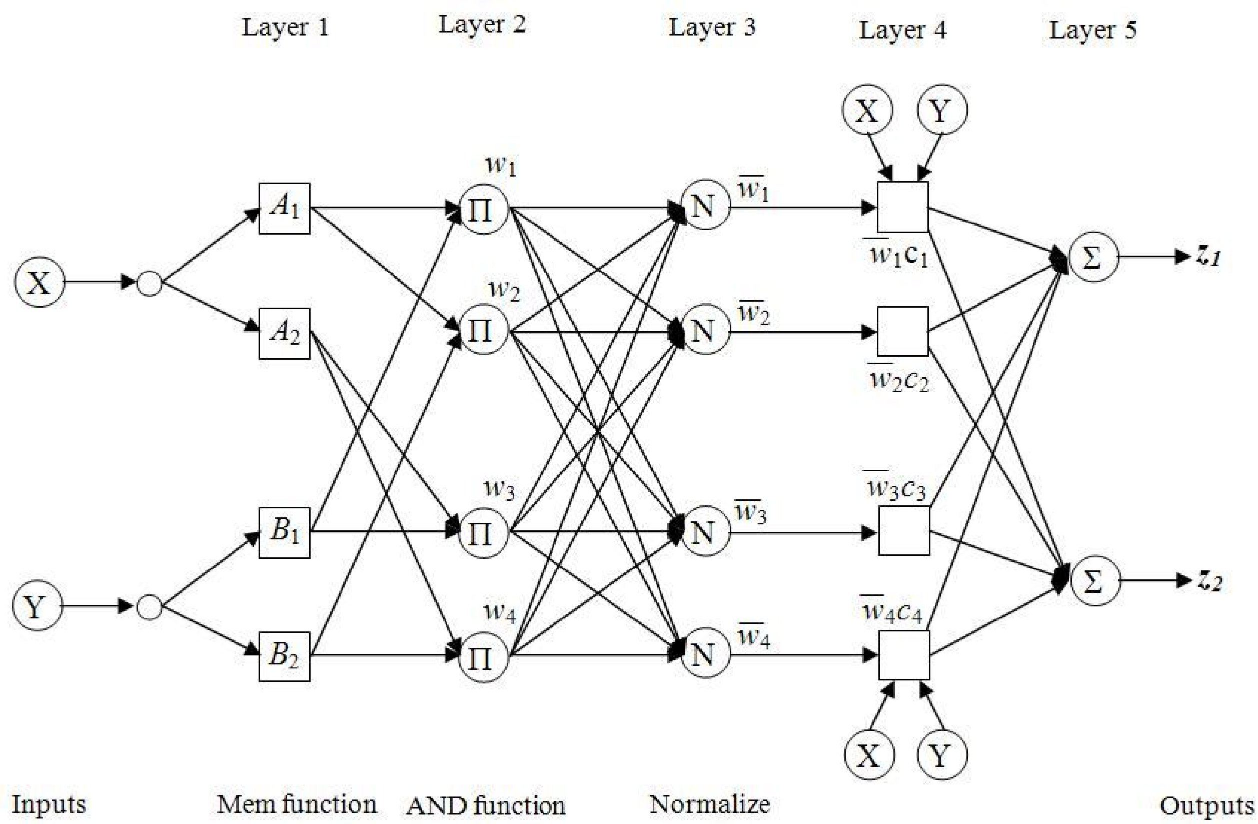


Figure 4.15: Two-input, two-output CANFIS architecture with four rules [12]

To build an n-output ANFIS model, one approach is by placing n number of ANFIS models side by side. The other approach of generating multiple-output ANFIS models is by maintaining the same antecedents of fuzzy rules among multiple ANFIS models; this is known as the CANFIS model. Figure 4.15 [12] according to [33] shows a CANFIS architecture for a two-input (x and y), two-output (z1 and z2) and Sugeno fuzzy model with the four rules given below.

Rule 1: If x is A_1 and y is B_1, then $C_1 = p_1x + q_1y + r_1$
Rule 2: If x is A_1 and y is B_2, then $C_2 = p_2x + q_2y + r_2$
Rule 3: If x is A_2 and y is B_1, then $C_3 = p_3x + q_3y + r_3$
Rule 4: If x is A_2 and y is B_2, then $C_4 = p_4x + q_4y + r_4$

The hidden and output layers of a neuro-fuzzy network can be shown as the consequent layer and fuzzy association layer as shown in Figure 4.16 [12].

In consequent layer, every node i is an adaptive node with a node function $C_i = f(p_i x + q_i y + r_i)$ where $\{p_i, q_i, r_i\}$ is the parameter set of each node referred to as consequent parameters. The output of the nodes in the consequent layer of Figure 4.16 [12] is:

$C_1 = f(p_1 x + q_1 y + r_1)$
$C_2 = f(p_2 x + q_2 y + r_2)$
$C_3 = f(p_3 x + q_3 y + r_3)$
$C_4 = f(p_4 x + q_4 y + r_4)$

The fixed nodes in fuzzy association layer compute the overall output as the summation of all incoming signals using the normalized firing strength from layer 3:

$$z_1 = \overline{w}_1 C_1 + \overline{w}_2 C_2 + \overline{w}_3 C_3 + \overline{w}_4 C_4$$
$$= (w_1 C_1 + w_2 C_2 + w_3 C_3 + w_4 C_4) / w_1 + w_2 + w_3 + w_4$$

$$z_2 = \overline{w}_1 C_1 + \overline{w}_2 C_2 + \overline{w}_3 C_3 + \overline{w}_4 C_4$$
$$= (w_1 C_1 + w_2 C_2 + w_3 C_3 + w_4 C_4) / w_1 + w_2 + w_3 + w_4$$

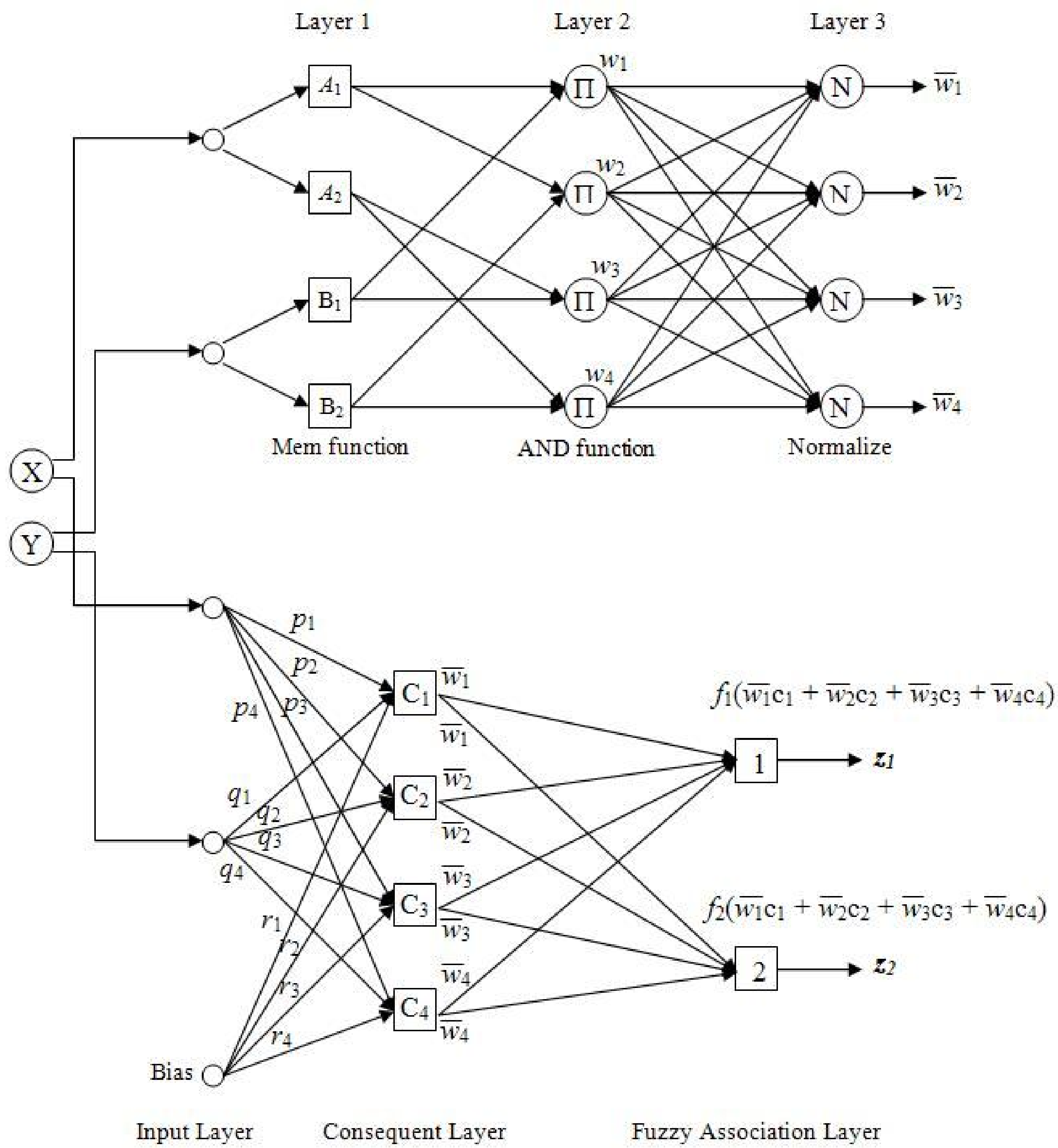


Figure 4.16: Two-input, two-output CANFIS architecture with four rules [12]

The neuro-fuzzy CANFIS system implemented in this research has been modified as a module in the software package and its implementation details will be given in the next chapter.

Chapter 5: Integrated Software Package

5.1 INTRODUCTION

In order to obtain the final classification results before development of the integrated software package (ISP), several different steps were involved. The researcher had to run separate programs (modules): the model parameter estimation program, sort its results to achieve the best estimated values, and then run the CANFIS program based on the selected respiratory impedance model (augmented RIC or extended RIC) to gain the final result. Considering the fact that human performance can be influenced by many factors, any mistake in this step-by-step process potentially can cause a misclassification and lead into wrong diagnosis.

The ISP described in this chapter is a user-friendly software package that accepts the IOS measurements as inputs, and after processing all the related programs (modules), shows the diagnosis results to the user.

Taking design and implementation principles of software development into consideration, in this package the IOS-based model parameter estimation codes and the developed CANFIS codes were converted to the components of an object-oriented design to seamlessly work together and build a robust and reliable software package to classify the severity of asthma.

Software development is the set of activities that results in software products. It may include research, new development, modification, reuse, re-engineering, maintenance, or any other activities that result in software products. There are several models for such processes including variety of tasks or activities that take place during the process.

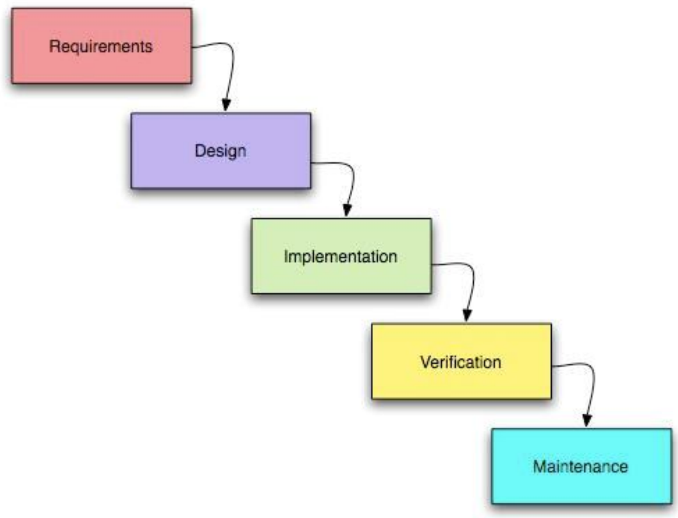


Figure 5.1: Major activities in software development process

The major steps in any software development methodology include problem analysis, design, implementation, verification, and maintenance. Figure 5.1 shows the diagram of the major activities in the software development process. In the following sections, the deployed methodology and its implementation will be described.

5.2 ANALYSIS AND SYSTEM OVERVIEW

The important task in creating a software product is extracting the requirements or requirements analysis. This phase involves understanding what the software is expected to process and produce and what the different functionalities of the software are.

At the beginning of a software development process, drawing a flowchart diagram that shows processes of the system and their order of connections is very useful. In Figure 5.2, the basic flowchart of the developed ISP is shown.

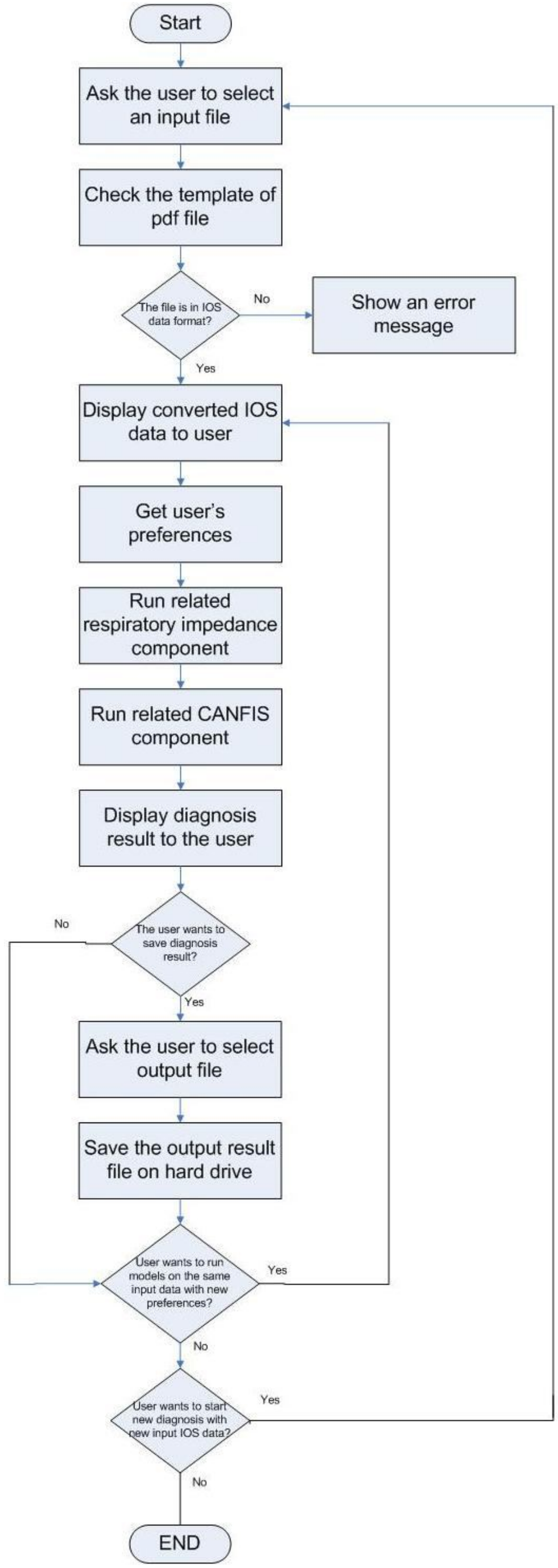


Figure 5.2: The basic flowchart diagram for the developed ISP

As discussed in chapter 3, IOS provides the respiratory resistance and reactance in both inspiration and expiration and also calculated the values of both inspiration and expiration phases. The IOS software provides the resistance and reactance values in the frequency range of 3, 5, 10, 15, 20, 25 and 35 Hz. It also produces some other parameters such as resonant frequency (F_{res}) and reactance area (AX). All these data and the personal information of the subject are saved in a file with .pdf format, readable with adobe acrobat reader software. This file is considered as the input for the developed ISP to extract the subject's information.

Inside the ISP, the next step is to run one of the two current parameter estimation model programs (modules) based on the user's preferences through the ISP interactive user interface. In this step, the user is able to overview the IOS data and decide on which test data and in what frequency range the model should be run.

Then the ISP will call the proper component and run. The result of this step is a set of numbers (model parameter estimates) including central airway resistance (R), peripheral (small) airway resistance (R_p), airway inertance (I), peripheral (small) airway compliance (C_p), and extrathoracic compliance (C_e specifically for aRIC model). After this step, the ISP will call the relevant CANFIS module, which uses selected IOS values as well as respiratory system's model-based parameters (feature vector) as inputs and automatically produces the classification results as outputs. The output appears in one of four different categories including Asthma, small airway disease (SAD), mild SAD (mSAD) or normal. In the next step, the classification results will be shown to the user and she/he can save the results of all steps in a file for later retrieval purposes. Figure 5.3 shows the relations and parameter passing steps of the classification process.

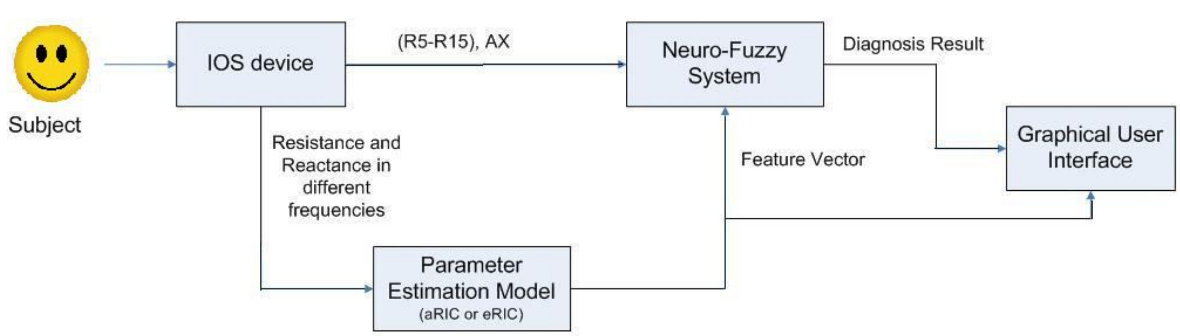


Figure 5.3: The major relations between main parts of the diagnosis process

5.3 SOFTWARE DESIGN

Software design is a process of problem-solving for a software solution that involves converting the informational and functional requirements of the system into unified design specifications. It would later be used to script programs during the development phase. Design phase includes low-level component and algorithm implementation issues as well as the architectural view.

Program designs are constructed in various ways. Using a top-down approach, designers first identify and link major program components and interfaces, then expand design layouts as they identify and link smaller subsystems and connections. Using a bottom-up approach, designers first identify and link minor program components and interfaces, then expand design layouts as they identify and link larger systems and connections. As the modules of this software were already developed (but needed to be modified and integrated), a top-down approach was used in this research.

There are two main design methodologies which are function-oriented design and object-oriented design. A function-oriented design strategy relies on decomposing the system into a set of interacting functions with a centralized system state shared by these functions. Functions may also maintain local state information but only for the duration of their execution [34]. In this methodology, data flows, graphical structure charts, and conditional statements and looping constructs are the main tools to design a software system. In contrast, object-oriented design strategy focuses on data rather than processes. It consists of classes and objects that each of them contains all the information needed to manipulate its own data structure. An object contains encapsulated data and procedures grouped together to represent its assistance. An object-oriented program can be considered as the interaction between the objects of the system.

The main reason to define and develop object-oriented design methodology was the tendency to consider data and behavior separately and encapsulate data and procedures of each object. In the object-oriented world, each object is capable of receiving messages, processing data, and sending messages to other objects and can be viewed as an independent 'machine' with a distinct role or responsibility.

As described before, even though different modules (programs) included in this software package depended on data from each other, but they were developed and used as stand-alone modules

and worked independently from each other. To classify one subject's IOS data, the user had to run the software modules more than one time to check the different results. The evidence presented above clearly translated into the suitability that an object-oriented design is more suitable for developing the ISP for this research.

5.3.1 What is a Design Pattern?

A design pattern is a reusable, generic solution to common problems found while designing software. It's not an algorithm to solve a specific instance of a problem, nor a piece of code or a template. Design patterns are abstract paradigms that need to be applied in their own specific form to each instance of the problem [35]. In most cases, they don't specify how the solution has to be implemented; but typically they show relationships and interactions between objects.

Design patterns can speed up the development process by providing tested, proven development paradigms. Effective software design requires considering issues that may not become visible until later in the implementation.

5.3.2 Object-Oriented Design Patterns

The key purpose of design patterns is reuse of experience. According to [36], there are three categories of object-oriented design patterns: creational, behavioral, and structural. Creational design patterns concern the process of object creation. Behavioral design patterns characterize the ways in which classes or objects interact and distribute responsibility. Patterns belonging to the structural category deal with the composition of classes or objects.

It is useful to mention that based on object-oriented definitions, the terms 'class' and 'object' refer to different concepts. A class is an abstract definition for a set of entities that are similar in defining their attributes. For example, 'car' is a class that generally describes an entity that has specific color, doors, an engine, etc. and it can perform specific functionalities such as turning on or off or moving. An 'object' is an instance of a class. In this example, a 2007 blue four door Camry Toyota that belongs to John Smith is one object derived from the general class of car. In this software, one of main and useful structural patterns, adapter pattern, has been used.

5.3.3 Adapter Design Pattern

This design pattern is useful when some components cannot work together because of their incompatible interfaces. It occurs when a component offers compelling functionality that you would like to reuse, but its structure is not compatible with the architecture of the system currently being developed.

Reuse has always been painful and elusive. One reason has been the tribulation of designing something new, while reusing something old. There is always something not quite right between the old and the new. It may be physical dimensions or misalignment. It may be timing or synchronization. It may be unfortunate assumptions or competing standards.

The adapter pattern is adapting between different components. Like any adapter in the real world, it is used to be an interface, a bridge between two objects. In real world we have adapters for power supplies, adapters for camera memory cards, and so on. Probably everyone has seen some adapters for memory cards. If you cannot plug in the camera memory card in your laptop, you can use an adapter.

In a software program, when the developer has a component expecting a specific type of input, and also has another component offering the same features, but exposing a different interface, he will not implement one of them again. With an adapter, the interfacing layer between two components can be created.

As an example, in Figure 5.4 the adapter object (an instance of adapter class) inherits the target interface that the end-user expects to see, while it holds an instance the adaptee. When the user calls the request() method on its target object (the adapter), the request is translated into the corresponding specific request on the adaptee.

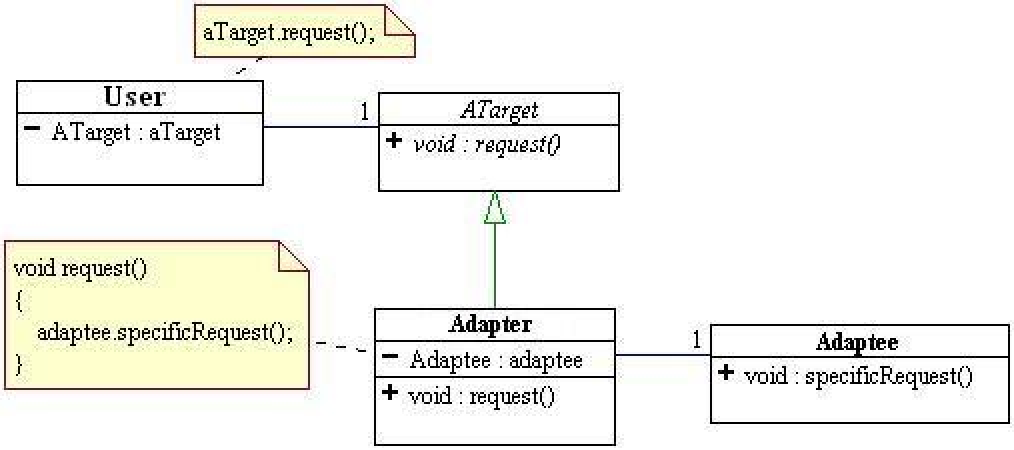


Figure 5.4: An example of an adapter design pattern with other components of a system

5.4 SOFTWARE COMPONENTS

The major component (module) in the IPS is the 'LugNet' component that is the implementation of the adapter design pattern and manages all the relations between classes and components inside the software. Other major components in this package are the parameter estimation components for respiratory impedance and the neuro-fuzzy classifier components (CANFIS). These components have been implemented in different programming languages. The codes of parameter estimation models (for both eRIC and aRIC models) are in MATLAB, while the codes of CANFIS models are in C++. Other main components used in the developed ISP is the Krypton toolkit which is a set of free user interface controls that allow the developer to quickly create professional looking applications.

5.4.1 Parameter Estimation Component

IOS measures airways resistance and reactance of the subjects at six different frequencies (chapter 3). A nonlinear least squares (LS) algorithm, which is a gradient-based method, can be applied to the augmented RIC and extended RIC models to estimate their parameters because of the nonlinear dependence of their impedance functions on the parameters. For simpler models such as RIC model, a linear LS algorithm can be used.

Unlike the linear LS algorithm, the nonlinear LS algorithm may produce parameter estimates corresponding to a local error minimum rather than a global minimum. In order to address this issue, a procedure was used whereby each estimation run began with an initial guess, i.e., a parameter estimate

vector produced by a random number generator that's appropriately weighted. This was then repeated at least 10 more times per model for each test data to find the parameters yielding the (globally) minimum error.

The earlier version of the codes for eRIC and aRIC parameter estimation components used the MATLAB optimization toolbox in order to implement the nonlinear least squares algorithm. MATLAB optimization toolbox extends the MATLAB technical computing environment with tools and widely used algorithms for standard and large-scale optimization. These algorithms solve constrained and unconstrained continuous and discrete problems. The toolbox includes functions for linear programming, quadratic programming, nonlinear optimization, nonlinear least squares, solution of systems of nonlinear equations, multi-objective optimization, and binary integer programming.

MATLAB is well-known for its programming environment that provides fast and easy-to-use tools to solve the problems with very high computing loads. For the current version of the ISP, it was decided to improve upon the earlier MATLAB codes for eRIC and aRIC models and use them in the package. MATLAB compiler allows us to build an executable or a shared library from a MATLAB application or program. Executables and libraries created with the MATLAB compiler use a runtime engine called the MATLAB compiler runtime (MCR) provided with MATLAB compiler for distribution with applicants and can be deployed for free. In the developed ISP, the MATLAB codes were compiled with MCR and used as dynamic link libraries (in .dll format) inside the software. It is important to mention that any time later in this research, if there is a need to add or change the MATLAB codes for model parameter estimation algorithms, it is easily possible with changing the MATLAB code, recompiling it, and then replacing the new .dll file with the older version in the folder that the software has been installed in.

The earlier version of the MATLAB code for eRIC and aRIC models could read the input data for a particular subject from an excel file (with .csv format) and fill a matrix. Since the IOS output is in .pdf format, the task of reading data from .pdf file and inserting its data into .csv file was the user's responsibility. In the developed integrated software package, a .pdf reader component has been modified

in order to read the data directly from IOS file and pass it to MATLAB code. After running the eRIC or aRIC component, the outputs are ready to be passed to CANFIS classifier component.

5.4.2 CANFIS Component

The coactive neuro-fuzzy inference system (CANFIS) is used as the reasoning mechanism of the classification system, and the IOS as well as aRIC and eRIC model parameters were explored as feature vectors with the expectation of obtaining an excellent performance with respect to the percentage of correct classification during memorization and generalization [12].

Among the impedance values that IOS measures, only the difference between resistances R_5 and R_{15} ($R_5 - R_{15}$) also known as frequency-dependence of resistance and the reactance area (AX) were selected as elements of the feature vector for the CANFIS classifier. The remaining elements of the feature vector were the estimated parameters of the respiratory impedance model: central airway resistance (R), airway inertance (I), peripheral (small) airway compliance (C_p), peripheral airway resistance (R_p) and the ratio of the peripheral (small) airway resistance to compliance (R_p/C) for the eRIC model and an additional extrathoracic compliance (C_e specifically for the aRIC model). The degrees of membership of these inputs form the data base of the classification system. The output of the CANFIS classifies the severity of asthma in subjects as reflected in the mechanical attributes of the peripheral airways into four groups as shown in table 5.1 [12].

Table 5.1: Different groups of asthma severity

Group	Inference
Group 1	Asthma
Group 2	Mild Small Airway Disease (mild SAD)
Group 3	Small Airway Disease (SAD)
Group 4	Normal (non-Asthmatic)

The fuzzy rules which make up the rule base of the classification system are selected based on the input variables of the CANFIS.

Two different CANFIS models were implemented using the C++ language to classify or differentiate between the four groups of asthma severity. As explained above, the first CANFIS utilized eight parameters of the IOS and aRIC respiratory model as elements of the input (feature) vectors, while the other CANFIS used seven parameters of the IOS and eRIC respiratory model. The details of the CANFIS programs were described in chapter 4 of this thesis.

In order to keep the end-user's focus on the main purpose of using the developed software, the software does not include the training phase of CANFIS and it only works with the trained network. After running the CANFIS outside of the software and obtaining the converged consequence weights, converged fuzzy weights, and momentum of the system, these data are added to a specific folder of the software. An executable file of the implemented CANFIS, which has been set to classify (not train) the input data, was added to the same folder of the software. This ability gives the flexibility to improve the CANFIS program apart from the rest of the software any time in the future and just add the needed files inside the software. In other words, the software works independently from the training algorithm of the neuro-fuzzy component.

5.4.3 LugNet Component

The LugNet component is the main component in this software because it manages all the interactions between the other components of the software. It can be considered as the implementation of the adapter design pattern, which is defined as a component between other components to make them communicate with each other regardless to their different interfaces.

This component includes the main classes of the software and calls the main methods to run when the user interacts with the software via its interface.

This component was built with visual C# (pronounced C sharp) programming language supported inside Microsoft Visual Studio 2008 which is an integrated development environment (IDE) from Microsoft Company. C# is a modern programming language that promotes object-oriented software development by offering syntactic constructs and semantic support for concepts that map

directly to notions in object-oriented design. This is in contrast to C++, which supports procedural as well as object-oriented and generic programming.

There are three major classes defined and used inside LugNet component, which is described in the next section.

5.5 MAIN CLASSES OF ISP

There are three main classes in the developed ISP that have the format of .cs (format of C# files) including PDFParser, Form, and DiagnosisProcess. Based on the class hierarchy, the Form class is the parent of two other classes which are mainForm and resultForm. When the program starts to run, an object of mainForm class is being generated to show all the control buttons and other controls on the form to the user to interact with the program. After completing the settings by user, when he clicks on one of the respiratory impedance models, a new form which is an object of ResultForm class will be appear on the monitor with the diagnosis result and other detailed information about it. PDFParser is a class with all attributes and methods necessary to convert a .pdf file to text. Every time that a new .pdf file is selected to be processed, one object of this class is being created to build a specific data structure with the information inside the .pdf file. DiagnosisProcess is a class that contains methods and functions for calling the different components to start the diagnosis process step-by-step.

A data flow diagram of the developed ISP is shown in Figure 5.5.

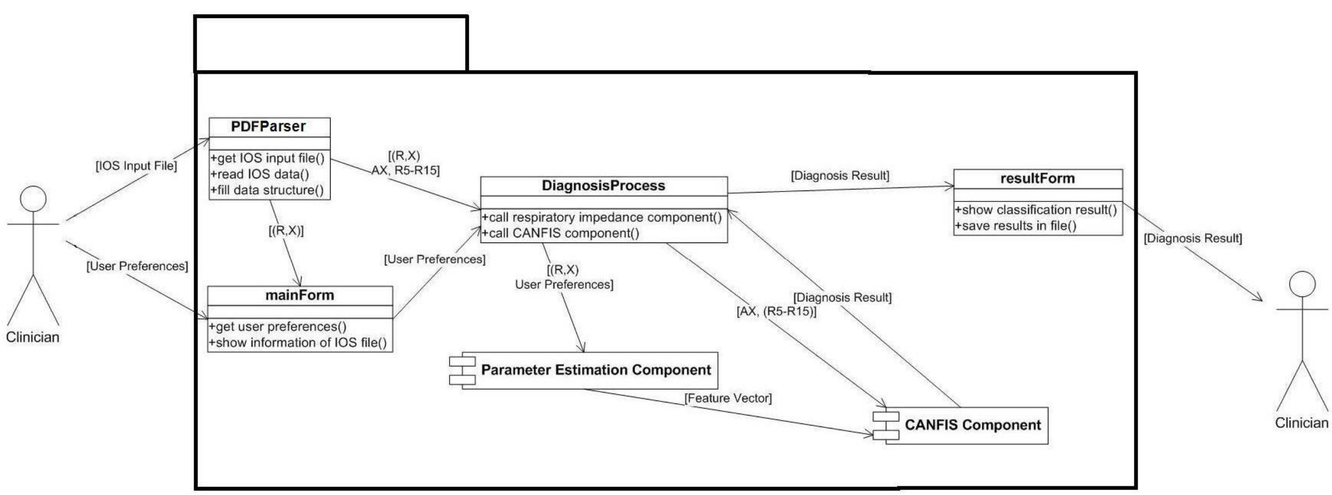


Figure 5.5: Data flow diagram inside the integrated software package

In Appendix A, a brief summary of source code, assemblies of ISP, designed classes, and software's pre-requirements are presented. In additional, appendix B and C show the relationship between assemblies of software and relationship between classes of software respectively.

Chapter 6: Results and Discussion

6.1 INTRODUCTION

This chapter details the user interface interactions between the end user of the software package and the system. The designed ISP is capable of distinguishing between the different severities of the asthma. It is developed with flexibility in mind to enable us make future changes in each of its main components with ease and allow us to add new diagnostic features derived from other respiratory system models and IOS parameters to expand respiratory disease classes and optimize performance of the package.

The programming language to develop this software is visual C#. Visual C# is Microsoft's implementation of the C# language. It targets the .NET Framework, along with the language services that allows the Visual Studio IDE support C# projects. While the language services are a part of Visual Studio, the compiler is available separately as a part of the .NET Framework. The Visual C# 2008 compiler supports version 3.0 of the C# language specifications. Visual C# supports the Visual Studio Class designer, Forms designer, and Data designer among others.

The Microsoft .NET Framework is a software framework that can be installed on computers running Microsoft Windows operating systems. It includes a large library of coded solutions to common programming problems and a virtual machine that manages the execution of programs written specifically for the framework. The .NET Framework is a Microsoft offering and is intended to be used by most new applications created for the Windows platform. Because interaction between new and older applications is commonly required, the .NET Framework provides means to access functionality that is implemented in programs that execute outside the .NET environment.

In this chapter, the procedure of using the ISP will be explained from the beginning of installation to demonstrating the final results.

6.2 INSTALLATION

In Visual Studio 2008 that this software has been developed with, there is a feature to build an installation file after debugging and compiling the program. It can be done by setting the solution

configuration to 'release' instead of 'debug'. As shown with an arrow in Figure 6.1, this setting should occur during the implementation phase.

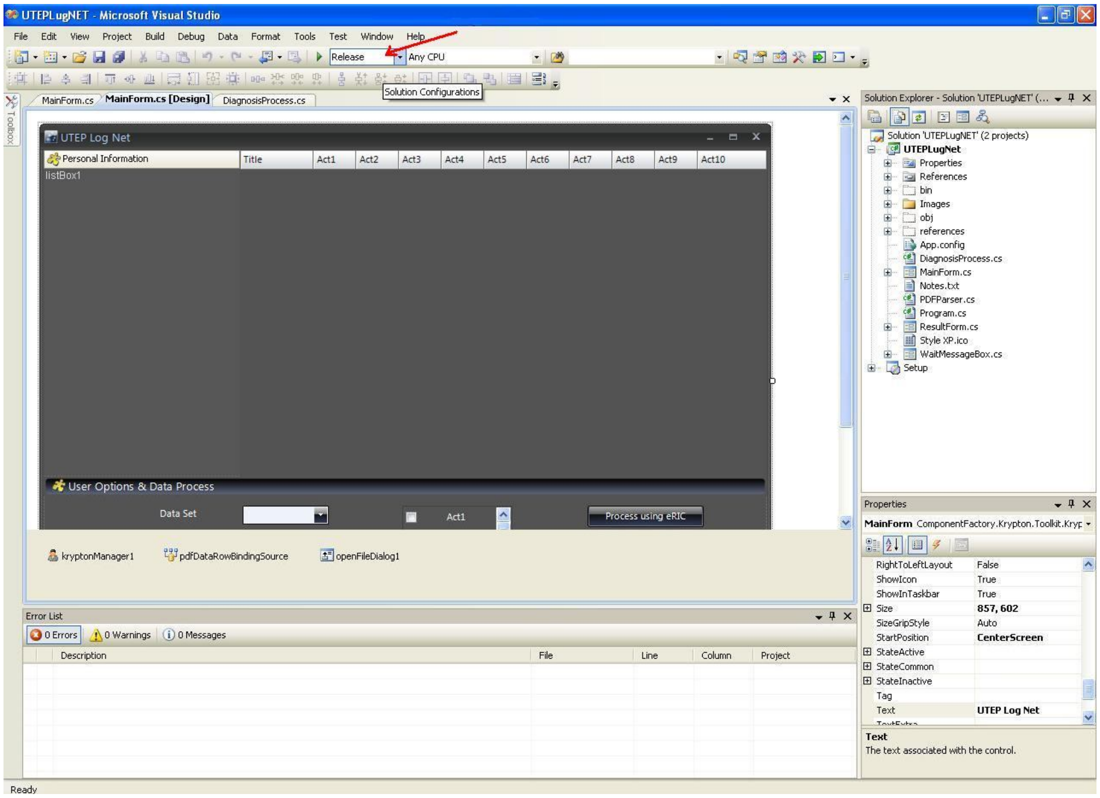


Figure 6.1: The Visual Studio Environment for Programming in C# language

When debugging the program is finished and no errors are found in the code, a folder named 'Setup' is created in the main folder of the project. This folder contains all the essential files and shared libraries and components that the program needs in order to run on another computer that may or may not have visual studio program and .NET framework. During the implementation phase, the developer can select the side components supported by visual studio 2008 as the prerequisites of installing the program such as .NET framework or windows installer.

Then developer just needs to copy and paste the 'Setup' folder on a target computer and run the setup.exe file in this folder. The installer will search for the necessary files and components on the target PC and if it doesn't find them, it will install them before installing the software. After confirming the

path and directory of the software with the user, the software package will be installed. After installation, ‘UTEP Lug Net’ program can be found in the startup window, under programs directory.

6.3 THE FORMAT OF SYSTEM’S INPUT

With opening the ‘UTEP Lug Net’ program, a Microsoft window similar to Figure 6.2 will appear on the screen. This window is a created object of mainForm class described in the previous chapter. In each running time, one object of this class is created to interact with the end-user.

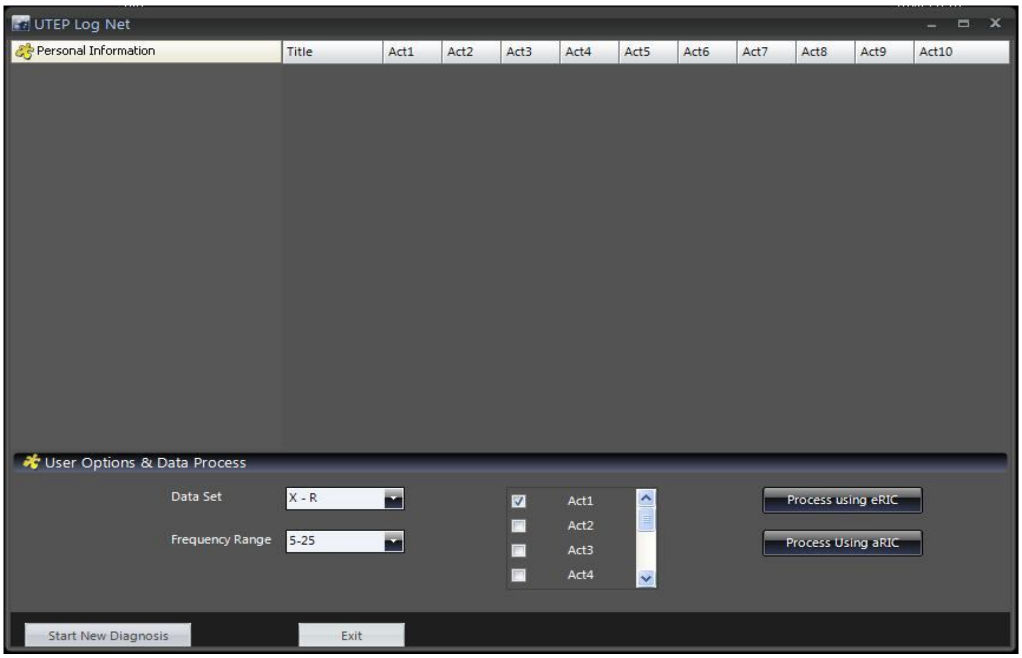


Figure 6.2: The first appearance of integrated software package after installation

In order to start diagnosis, the user should click on the ‘Start New Diagnosis’ button on this window. This button calls method of this event that opens a dialogue window to ask the user to select an input file. Figure 6.3 presents how this dialogue window will be shown to the user.

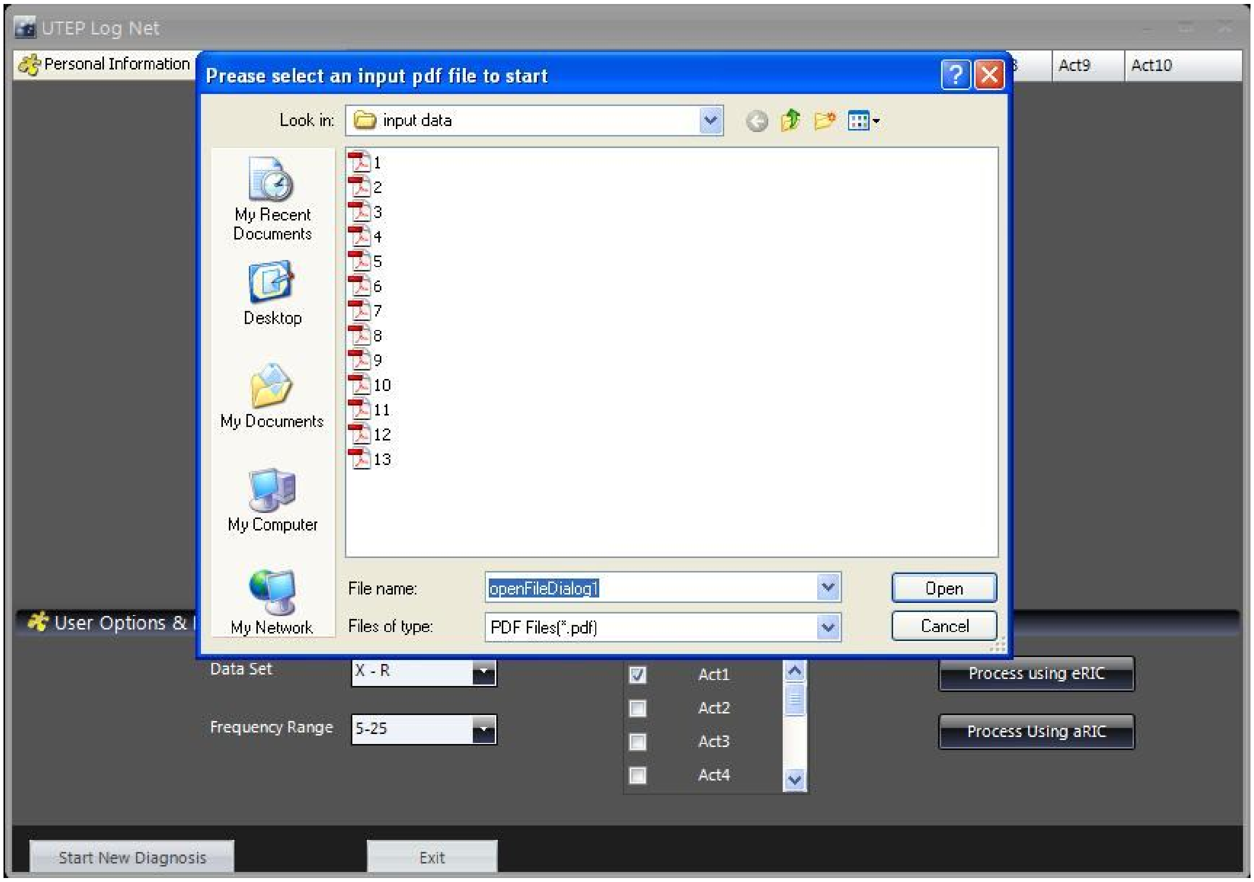


Figure 6.3: The dialogue window to ask the user to insert an input file

The input file as discussed before should be one of the IOS output files in .pdf format. Each IOS .pfd file contains the information and measured data for a subject. In appendix D, a sample IOS file is given.

If the user selects an improper file that does not contain IOS data, the program will show an error message. After a proper file is selected, the subject's personal information and the measured IOS data for every time that a test has been taken will appear in the window.

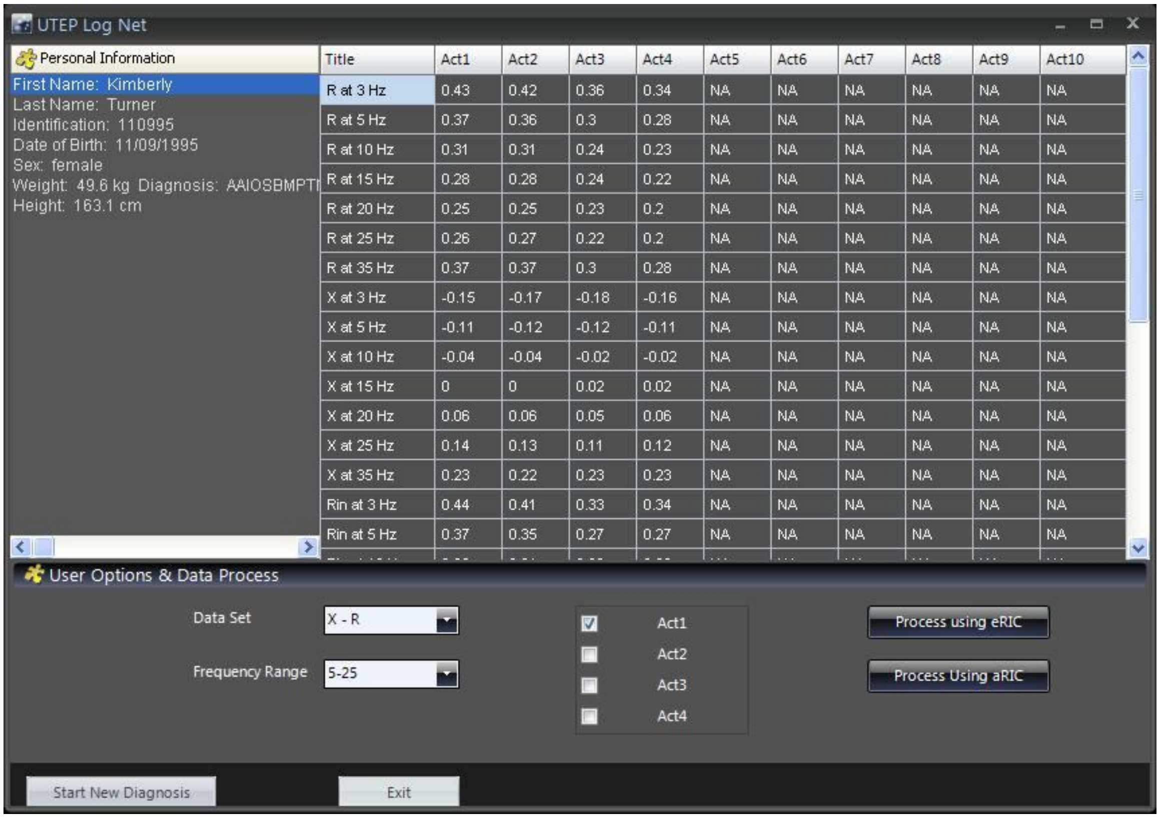


Figure 6.4: Personal information and IOS data of the subject whose IOS file has been used as system's input

As an example shown in Figure 6.4, there are four sets of IOS test data for the subject named Kimberly Turner, with the given personal information presented in left column of the window. With the scroll bar next to the right column, the user can scroll up and down to observe the values of resistance (R) and reactance (X) of the subject's respiratory system stored in different frequencies.

6.4 SETTING USER PREFERENCES

The buttons under the information columns in the main window shown in Figure 6.4 give full control to the user to select his preferred model and test data for this subject. The 'Data Set' option provides the ability of choosing the inspiration or the average data values to run a test. As shown in Figure 6.5, 'Frequency Range' selects the specific range of frequency in which the user wants to test the selected data set.

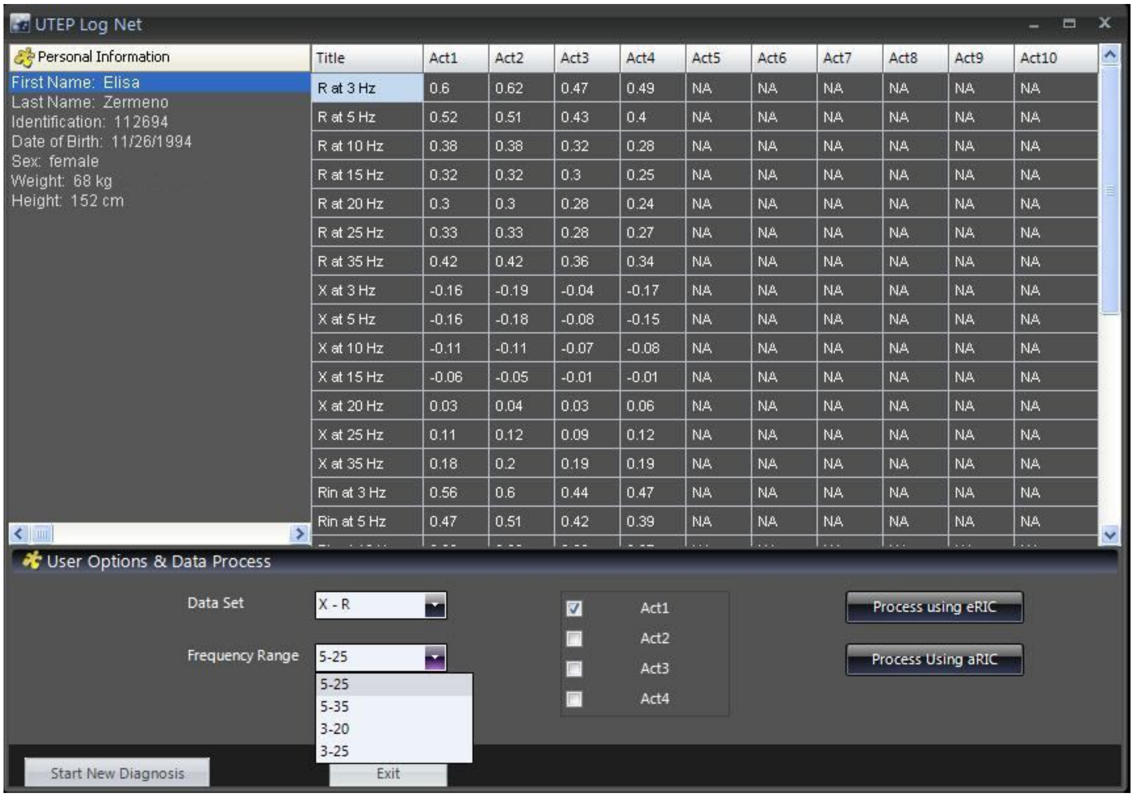


Figure 6.5: The range of frequencies that user can choose to run a test

The next option lists all different IOS data sets that the user can choose to feed to the respiratory models. The user can also select several data sets. The program computes the average values of the selected data sets and runs the models with the average values. After all these settings have been done, the last setting in this window is choosing one of the models by clicking on one of ‘Process using eRIC’ or ‘Process using aRIC’ buttons.

6.5 DIAGNOSIS PROCESS

With choosing the processing method of IOS data for model (eRIC or aRIC) parameter estimation, the software will direct the user to the next window, which is an object of the resultForm class.

Note that with the same input data, the user can select completely different preferences and run the software separately and independently from the previous preferences. It means that the software allows the user to observe different results of the diagnostic process by selecting different respiratory impedance models and different frequency ranges at the same time. As the design and implementation

are completely based on object-oriented principles, running the system with the same input file but different user preferences do not have any conflict with each other and they work independently. The power of object-oriented methodology can be realized in this aspect of the implemented ISP.

6.6 THE OUTPUT RESULT OF THE ISP

Figure 6.6 shows an example of running the software for a subject who has been diagnosed 'normal'. In the left side of this window, the user can browse all the information extracted from the IOS data file, the details of his selected references for processing the data and the results of each step of processing phase inside the program.

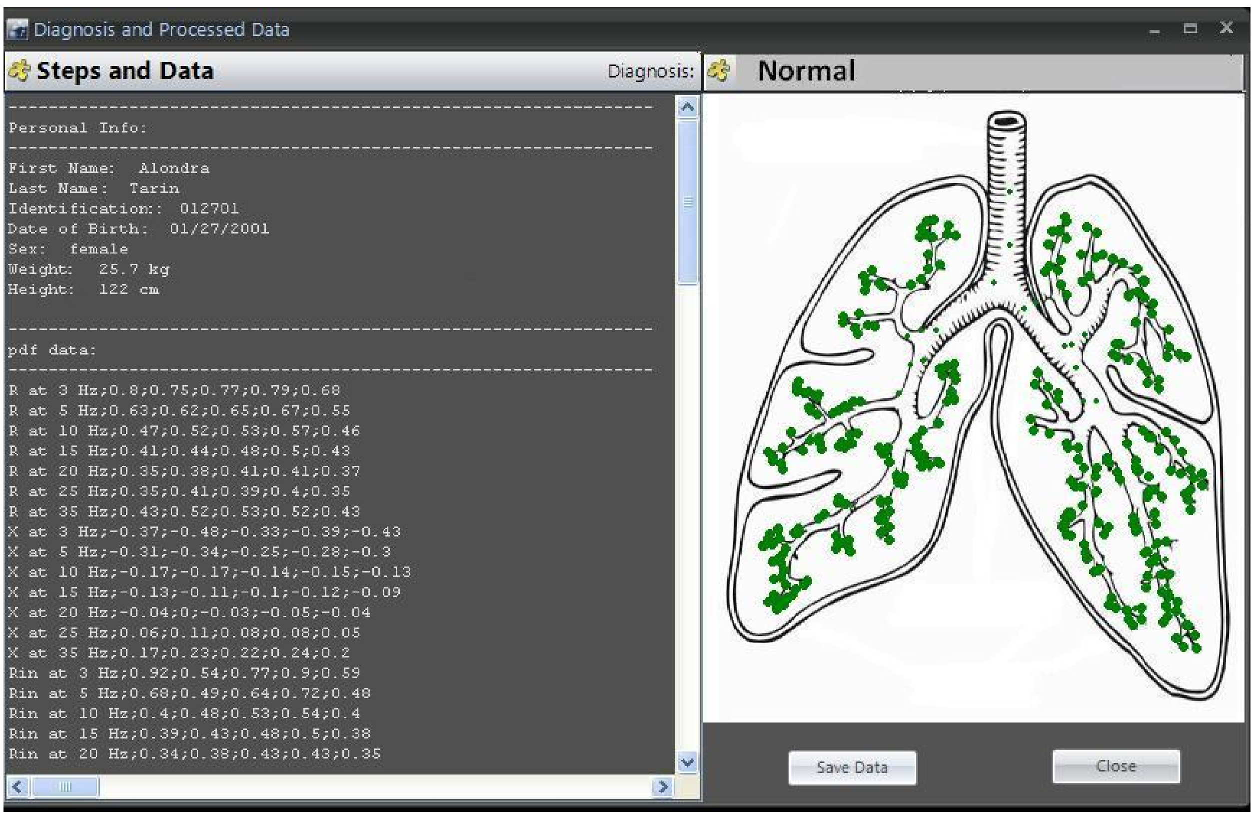


Figure 6.6: An example of the output result of the integrated software package

The different parts of the 'diagnosis and processed data' are:

Personal Information: includes the first name, last name, identification number (during the IOS test), date of birth, sex, weight and height of the subject.

PDF data: includes all measured IOS values for resistance (R) and reactance (X) at different frequencies, and the value of AX.

User Preferences: shows the user's selected set of IOS data fed to the respiratory system model. It also shows user's selected frequency range, the inspiration phase or the average amounts of measured values, and the selected respiratory system model (i.e. eRIC or aRIC).

MATLAB Input: shows the values that have been passed to MATLAB component after reading the IOS data into a proper data structure inside the software.

Diagnosis Input File (Feature Vector): shows the computed values of R_5-R_{15}, AX, R, R_p, I, C_p, C_e (just for aRIC model), and R_p/ C_p.

Diagnosis Result: shows the diagnosis result after running the proper CANFIS program on the input data set. It will be asthma, mild SAD, SAD or normal.

On the right column of the result window, a proper picture is shown to the user that shows the classification result in a graphical way. When the subject is normal (as shown in Figure 6.6), the picture is clear with green spots on specific areas of lungs. The green color becomes yellow, orange and red if the patient is diagnosed by mSAD, SAD or asthma respectively.

Under this picture, there is a control button named 'Save Data' that gives the user the ability to save all the detailed data and results of each processing step in an Excel file for future use.

The user can leave this window open and go back to main window of the program to perform a new classification on the same subject's input file. With changing the settings in this window and then clicking on one of the models, the software will start a new thread with the same input file and run the program again. The result of this run will be shown in a new resultForm window and the user is able to compare the results of different preferences on the same data at this point and also save the results in another Excel file.

6.7 VERIFICATION OF THE RESULTS

As discussed in chapter 5, the software development process is not fully accomplished with implementation alone. Testing and verification phase is an essential phase in any software development process.

For this purpose, the developed software has been tested on different data sets in different phases of its performance as listed below:

Test 1: The first verification step was comparing the IOS data in the input pdf file with the extracted data after the instance of PDFReader class ran its specific method to convert the data. In this part of the test process, there were no errors and both data sets were completely the same.

Test 2: The next step was running both eRIC and aRIC parameter estimation models over different frequency ranges (with five and six input frequencies) with both inspiration data and average data. In this step, 20 different testing combinations were fed into the software to check the accuracy of software's results. The data patterns tested with six frequency values could be verified against previous results obtained by the research team. This test provided further verification for correct input and output data sets after running the parameter estimation models in MATLAB.

In this test, the eRIC and aRIC parameter estimation models were also fed with respiratory impedance data at five frequencies (5, 10, 15, 20, 25 Hz and 3, 5, 10, 15, 20 Hz). Our expert clinician then verified that the parameter estimates calculated based on IOS values at these frequencies produced values in the expected range of the diagnosed category.

Test 3: In this step the accuracy of results after running CANFIS programs inside the software package were evaluated. Since the training data set and the CNAFIS programs were not modified from the previous work reported by our research team [12], the result of each IOS test pattern was verified by comparing it with the classification result of CANFIS program outside the new developed software.

The classification performance investigation was launched with 50 different processing data patterns in this step. The diagnostic (classification) accuracy of CANFIS was 100% compared to the gold standard (expert diagnosis).

Three test steps outlined above demonstrate that the developed integrated software package is highly reliable and robust tool and could be used in our research lab and others to support and facilitate the process of IOS data analysis and classification of small airway dysfunction in asthmatic children. This package has the potential to be expanded to include adult IOS data analysis and could therefore be deployed to address the classification of different pulmonary conditions.

6.8 DISCUSSION

There were two major issues in the process of the integrated software package development that the developer considered building the system based on them. The first and main issue can be referred to as encapsulation. The system had to be developed in a way that each component could function completely independent of the rest of the components and allow seamless communication (compatibility) between the components through passing parameters between them.

Other consideration about encapsulation of components was the accuracy of their implemented code. They were assumed to work correctly and without any errors outside of the software and then combined into the integrated software later. As this research is currently evolving and there would be further improvements and additions to the modules and the codes of each component in the future, the software has been designed in a modular fashion, independent and separate from all kinds of changes that may happen to the components. Any changes in the respiratory impedance models or CANFIS programs should be tested and verified outside of the software. The developer in the maintenance step of the software development process should build a shared library or executable file of the changed component and then replace it with the current version inside the software package in the specific directory of the package.

The other main issue in development of the integrated software package was considering adding new components (respiratory models and a more comprehensive and sophisticated neuro-fuzzy classifier) into the system in the future. For this purpose, two main functions have been implemented in the software that can be easily updated to call the related components for the new classification method and add the related control buttons in the interface class (mainForm). For example, if the research team decides to add optimized versions of the eRIC and aRIC or the Mead model (discussed in chapter 3) to the diagnosis package in the future, their corresponding parameter estimation MATLAB programs should be implemented and tested outside of the package first. Then a library file (with .dll format) should be created from it inside the MATLAB software, and that file should be added to the package with calling the function ‘DllImport’. In a similar way, the C++ CANFIS code for this method should be implemented and tested outside of the package and its executable version with all required files for it should be added to a folder specific for this method.

Chapter 7: Conclusions and future Enhancements

7.1 SUMMARY OF RESULTS

The aim of this research was to develop and test an integrated software package based on the object-oriented methodology to classify the severities of small airway impairment such as asthma by using previously developed computer programs in the biomedical engineering laboratory of the University of Texas at El Paso.

IOS is a model-based pulmonary function test. Therefore, IOS data were used to estimate the capacitive, resistive and inertive elements of the central and peripheral airways in the human respiratory impedance by using equivalent electrical circuit models (extended RIC and augmented RIC models) and well-established parameter estimation algorithms. Based on insightful advice from the expert clinician/pulmonologist in the research team, the IOS and model parameters were classified in four categories including asthma (A), small airway disease (SAD), mild small airway disease (mSAD) and normal (N). The two CANFIS modules were trained by using IOS data and parameter estimates of the eRIC and aRIC models.

The described components constitute the essential parts of an integrated software package, which was developed in C# programming language to provide a user-friendly, reliable, efficient and flexible research tool to support future projects in pulmonary physiology and medicine in the UTEP biomedical engineering research laboratory.

The testing phase of software development process in this project included examination and assessment of the functionalities of each subsystem and thorough evaluation of the functionality and results of the whole system.

7.2 CONCLUSIONS

The object-oriented structure of the developed ISP provides a firm and easy-to-use platform for both clinicians and software developers who are interested in investigating the utility of human respiratory system models and automatic pattern classification of IOS data. The following list is some of the important conclusions that could be drawn from the results of the developed ISP:

1- The human respiratory system models and their parameter estimations could be improved and replaced with newer versions at any time in the future.
2- The current CANFIS models could be replaced with improved versions inside the ISP. It is important to mention that our research team is generating new IOS training data sets related to other pulmonary disorders. Therefore, our expanded CANFIS classifiers would classify more classes of pulmonary disorders with acceptable clinical accuracy, sensitivity and specificity than now.
3- The ISP of this research is able to merge any new component (respiratory impedance model) prepared by any common programming language such as MATLAB or C++. An executable or shared library file is needed to be created from the code of the new model and it can be added inside the package easily. This ability originates from the firm encapsulation aspects of the developed software package.
4- Running MATLAB components inside C# program takes more time especially with six frequencies. As the software has been developed for evaluating one subject's data after training phase, the longer runtime is a minor problem.
5- The software package has been tested on several computers and laptops with both Windows XP and Windows Vista operating systems and it works perfectly, without any errors and bugs on these operating systems. Running the software on other operating systems such as Linux needs major changes in the code and has not been experienced yet.
6- For developing the graphical user interface of the software package, the Krypton toolkit which is a set of free user interface controls has been used instead of the .NET features because of more professional look of its interface designs.

7.3 FUTURE ENHANCEMENTS

The designed, developed and well-tested ISP serves as a valuable and user-friendly research tool to support the investigations or our research team in biomedical engineering, pulmonary physiology and medicine. In addition to the very important goal of improving the current components and modules

inside the existing software, there are some major works that our research team desires to accomplish in the future.

The weights of the links in a neural network are more accurate when it is trained with more data patterns in the training phase. Based on this theory, one of the next primary missions is testing the ISP with more training data patterns in order to improve its classification performance. The current training data patterns are recorded from 112 children with different asthma severity problems between the ages of 5 and 17 in El Paso Texas [12]. A new set of IOS data acquired from Swedish normal children and adolescents will be used in the near future to retrain the neuro-fuzzy classifier in the ISP and improve its classification performance.

The research team plans expand the functionality of the developed ISP with more data patterns acquired from different age groups. It is also of interest to test the system with data patterns acquired from subjects with different ethnicities living in different geographical regions.

Other major future work for our research team is model-based, neuro-fuzzy classification of other pulmonary system diseases to assist with their more sensitive, specific and accurate diagnosis. For example, impulse oscillometeric evaluation of respiratory function in fire fighters, first responders, military personnel, factory workers, school children, smokers and non-smokers would be of great interest to our research team. As mentioned before, it is also intended to investigate and implement more respiratory impedance models inside the existing package.

The ISP can also be an excellent tool for medical researchers to investigate the causes of the airway obstructions in different respiratory ailments.

The graphical user interface is another part of this software development process that the developers want to improve to display the anatomical locations of the classification results in a three dimensional way.

Some of the desired future directions would be development of a long-term investigation program into quantitative screening, diagnosis, monitoring and classification of respiratory diseases in the US-Mexican border region. Generation, documentation and annotation of a large database on

different respiratory diseases to improve the quality of respiratory health care in the region and promote community-based health awareness programs are at the core of our research team research mission.

References

[1] Allergic and Environmental Asthma, eMedicine Journal. November 26, 2001.

[2] Barua M., Nazeran H., Nava P., Diong B., Goldman M., "Classification of Impulse Oscillometric Patterns of Lung Function in Asthmatic Children Using Artificial Neural Networks," Proceedings of the IEEE-Engineering in Medicine and Biology Society (EMBS), 27th Annual International Conference, 2005.

[3] Barua M., "Classification of Pulmonary Diseases Using Artificial Neural Networks," M. S. Thesis, The University of Texas at El Paso, 2004.

[4] Dencker M., Malmberg L. P., Valind S., Thorsson O., Karlsson M. K., Pelkonen A., Pohjanpalo A., Haahtela T., Turpeinen M., Wollmer P., "Reference values for respiratory system impedance by using impulse oscillometry in children aged 2–11 years," Clinical Physiological Function Imaging, pp 247 – 250, 2006.

[5] Smith H. J., Reinhold P., Goldman M. D., "Forced Oscillation Technique and Impulse Oscillometry," Eur Respir Mon, pp 72 – 105, 2005.

[6] Woo T., Diong B., Mansfield L., Nazeran H., Goldman M., Nava P., "A Comparison of Various Respiratory System Models Based on Parameter Estimates From Impulse Oscillometry Data," Proceedings of the 26th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, pp. 3828-3831, 2004.

[7] Google Pages, "Human Physiology and Anatomy," http://humananatomyandphysiology.googlepages.com, 2008.

[8] Sherwood L., "Human Physiology: From Cells to Systems," Thomson Learning, Inc., California, 2004.

[9] Health Library (Centra Health), "Respiratory System," http://myhealth.centrahealth.com/library/healthguide/en-us/support/topic.asp?hwid=ug2841, 2008.

[10] The Dynamic Human, "Respiratory visual," http://www.mhhe.com/biosci/ap/dynamichuman2/content/index2.mhtml, 2008.

[11] Marieb E., "Human Anatomy & Physiology," Pearson Benjamin Cummings, California, 2004.

[12] Maduko E., "A Co-Active Neuro-Fuzzy Inference System for Classification of Asthma," M. S. Thesis, The University of Texas at El Paso, 2007.

[13] National Heart Lung and Blood Institute, "Asthma," http://www.nhlbi.nih.gov/health/dci/Diseases/Asthma/Asthma_WhatIs.html, 2008.

[14] eMedicin from Web MD, "Restrictive lung diseases," http://emedicine.medscape.com/article/301760-overview, 2009.

[15] Online text book in Physiology, "Mechanics Of Breathing And Lung Disorders," http://www.mfi.ku.dk/ppaulev/chapter13/chap_13.htm, 2000.

[16] West J., "Pulmonary Physiology and Pathophysiology," Lippincott Williams & Wilkins, Maryland, 2007.

[17] Wikipedia-The Free Encyclopedia, "Respiratory Physiology", http://en.wikipedia.org/wiki/Respiratory_physiology, 2007.

[18] Dencker M., Malmberg L. P., Valind S., Thorsson O., Karlsson M. K., Pelkonen A., Pohjanpalo A., Haahtela T., Turpeinen M., Wollmer P., "Reference values for respiratory system impedance by using impulse oscillometry in children aged 2–11 years," Clinical Physiological Function Imaging, pp 247 – 250, 2006.

[19] Smith H. J., Reinhold P., Goldman M. D., "Forced Oscillation Technique and Impulse Oscillometry," Eur Respir Mon, pp 72 – 105, 2005.

[20] Viasys® Health Care, “Pulmonary Function – Vmax® IOS Brochure,” http://www.viasyshealthcare.com/prod_serv/downloads/178_Vmax_IOS_Brochure.pdf, 2007.
[21] Woo T., Diong B., Mansfield L., Nazeran H., Goldman M., Nava P., “A Comparison of Various Respiratory System Models Based on Parameter Estimates From Impulse Oscillometry Data,” Proceedings of the 26th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, pp. 3828-3831, 2004.
[22] Dubois A., Brody A., Lewis D., Burgess B., “Oscillation mechanics of lungs and chest in man,” Journal of Applications for Physiology, pp 8: 587–94, 1956.
[23] Oostveen E., MacLeod D., Lorino H., Farre R., Hantos Z., Desager K., Marchal F., “The forced oscillation technique in clinical practice: methodology, recommendations and future developments,” European Respiratory Journal, pp 1026–1041, 2003.
[24] Goldman M., Saadeh C., Ross D., “Clinical application to assess peripheral airway function,” Respiratory Physiology and Neurobiology, vol. 148, pp. 179-194, 2005.
[25] Lorandi L., Diong B., Nava P., Solis F., Menendez R., Ortiz G., Nazeran H., “Parametric Sensitivity Analysis of Human Respiratory Impedance,” Annual International Conference of the IEEE EMBS, 2003.
[26] Diong B., Rajagiri A., Goldman M., Nazeran H., “The augmented RIC model of the human respiratory system,” Medical and Biological Engineering and Computing journal, vol. 47:395–404, 2009.
[27] Meraz E., Nazeran H., Diong B., Menendez R., Ortiz G., Goldman M., “Modeling Human Respiratory Impedance in Hispanic Asthmatic Children,” Annual International Conference of the IEEE EMBS, 2007.
[28] Doing B., Nazeran H., Nava P., Goldman M., “Modeling Human Respiratory Impedance,” IEEE Engineering in Medicine and Biology Magazine, pp 48 – 55, 2007.
[29] Fausset L., “Fundamentals of Neural Networks: Architectures, Algorithms, and Applications,” Prentice Hall, New Jersey, 1994.
[30] Yaeger L., Lyon R., Webb B., “Effective Training of a Neural Network Classifier for word Recognition,” Advances in Neural Information Processing Systems 9, Cambridge, MA, MIT Press, 1997.
[31] Yen J., Langari R., “Fuzzy Logic: Intelligence, Control and Information,” Prentice Hall, New Jersey, 1999.
[32] Jang J-S. R., “ANFIS: Adaptive-Network-Based Fuzzy Inference System,” IEEE Transactions on Systems, Man and Cybernetics, Volume 23, No. 3, pp 665 – 685, 1993.
[33] Jang J. –S. R., Sun C. –T., Mizutani E., “Neuro-Fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence,” Prentice Hall, 1997.
[34] Computer Science at the University of St Andrews, “Function-oriented Design,” http://www.cs.st-andrews.ac.uk/~ifs/Resources/Notes/Design/FunctionDesign.pdf, 1994.
[35] AC Associated Content, “Software Development: What Are Design Patterns?,” http://www.associatedcontent.com/article/2303773/software_development_what_are_design.html?cat=15, 2009.
[36] Gamma E., Helm R., Johnson R., Vlissides J., “Design Patterns: Elements of Reusable Object-Oriented Software,” 1995.

Appendixes

APPENDIX A. A brief documentation of the designed Integrated Software Package

1. SOURCE CODE

The source code is a Visual Studio solution containing two projects:

- UTEPLugNET: A winform program. While compiled, it produces LugNet.exe as the output.
- Setup: It is a setup project that produces and installer for the software.

Classes in the winform program are contained in two folders (‘UI’ and ‘Processing’) according to their namespace. Images are in a folder with the same name and all external assemblies and references are put in the folder named ‘References’ inside the root directory of this program. To update any of these assemblies and files, you must copy the new versions in this folder. You may need to copy some of these files (images, data files and C++ executalbles) into the ‘bin/Release’ subdirectory, if you want to test the program without reinstallation.

The final output of the program is the diagnosis made by C++ program. This data is interpreted using certain words or phrases in the output text file. These representative phrases are configurable through the configuration file. Using the configuration variables provided, you can distinguish up to 5 categories using corresponding pattern in the output file. Output0 is for not classified data, output 1 for Asthma, output 2 for Small Airway Disease (SAD), output 3 for mild SAD, and output4 for Normal. An appropriate message will be shown to user accordingly (Message0 – Message4).

2. ASSEMBLIES

These executables and assemblies work together in this software:

- UTEPLugNet: the main executable of the software which consists of the GUI, I/O operations and data processing of the pdf input and output of the MATLAB program; and doing so, integrates the separate parts of the program.
- aRICNET.dll/eRICNET.dll: these assemblies are the output of .NET compilation of the MATLAB program. They work as an object-oriented wrapper (façade) for interation between .NET and MATLAB, through a class named aRICNETclass/eRICNETclass. To call the

MATLAB procedures, an instance of these classes is created and its method aRIC_main/eRIC_main is called in the C# code. The actual code in the program is:

```
if (mode == ProcessMode.eRIC)
{

    eRICNet.eRICNetclass eric = new eRICNet.eRICNetclass();
    eric.eRIC_main();
}
else
{
    aRICNET.aRICNETclass aric = new aRICNET.aRICNETclass();
    aric.aRIC_main();
}
```

- NF1_aRIC.exe/NF1_eRIC.exe: These executables are the C++ codes of the aRIC/eRIC method and are compiled independently. They are executed directly using Process.Run command in C#. Minor modifications in the original codes of these programs are made before compilations that prevent them to get any input from users. These modifications are commented in the source files NF1_aRIC.cpp.NF1_eRIC.cpp.
- MWArray.dll: It is a prerequisite class library that is necessary for calling MATLAB functions in a .NET program.
- ComponentFactory.Krypton.Toolkit.dll: User interface of the program is designed using Krypton library, included in this assembly.
- iTexSharp.dll and SharpZipLib.dll: These assemblies are referenced during extraction of text data out of the pdf input.

3. Classes

The most important assembly introduced in the previous section is LugNET.exe, which will be undermined in this section. Classes contained in this assembly are categorized in two namespaces UTEP.LugNET.UI and UTEP.LugNET.Processing, each containing user interface and business-logic layers of the software respectively, as their name suggest. All fields, properties and methods of these classes are documented in the source code.

UTEP.LugNET.UI: There are three user interface classes in this namespace:

- MainForm: this is the form shown after running the program. It contains the necessary GUI controls to load the input file, show the contents and get the user preferences.
- ResultForm: after clicking the process buttons in the previous form, the result of diagnosis and full preview of data generated in the steps of program, is shown in this form. User can decide to save these data.
- WaitMessageBox: When running relatively time-consuming processes, it is recommended to show a message box to let the user know that he/she must wait until the process is done. This form acts as such a message box, by getting a thread and showing a wait message during its progress and closing after it is done.

UTEP.LugNET.Processing: It contains processing methods and classes.

- DiagnosisProcess: This class is the engine of business-logic layer. It provides various methods to perform different task during diagnosis procedure. All user-interface classes are dependent these classes to process data. Some tasks, such as MATLAB and C++ programs, are not actually performed using this class and they are handed over other classes to be actually performed.
- ProcessMode: This is a simple enumeration which determines the mode of diagnosis procedure: eRIC and aRIC.
- ProcessStatus: Another enumeration which tells the step to which the diagnosis process is preceded. This enumeration defines different values for pdf data reading, preparing input files, reading output files and running MATLAB and C++ programs.
- PDFParser: Input data for this program is fed as a pdf file. To extract the actual data out of this file, this class reads the file out and converts the whole document into a single character string. Then numeric data is fetched from this string by ProcessStatus class.
- PDFDataRow: Data in the pdf file is tabular and every instance of this class represents a single row of it.
- PdfRowType: This enumeration shows the different kind of rows in pdf data table, according to the header column, consisting: A, X, Ain, Xin, Rin and Ain.

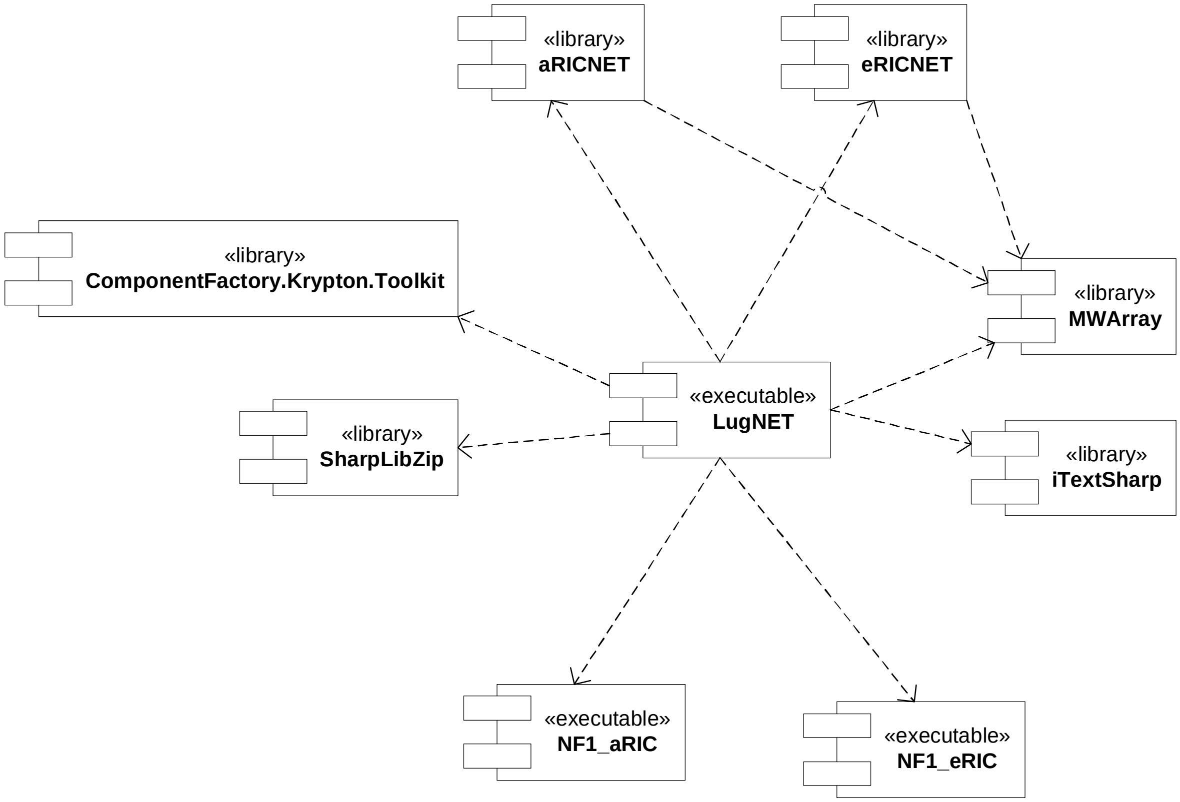
- MatlabOutRow: Another tabular data in the program is the output of the MATLAB program. To handle this kind of data, MatlabOutRow is defined.

4. RUNTIME ENVIRONMENT

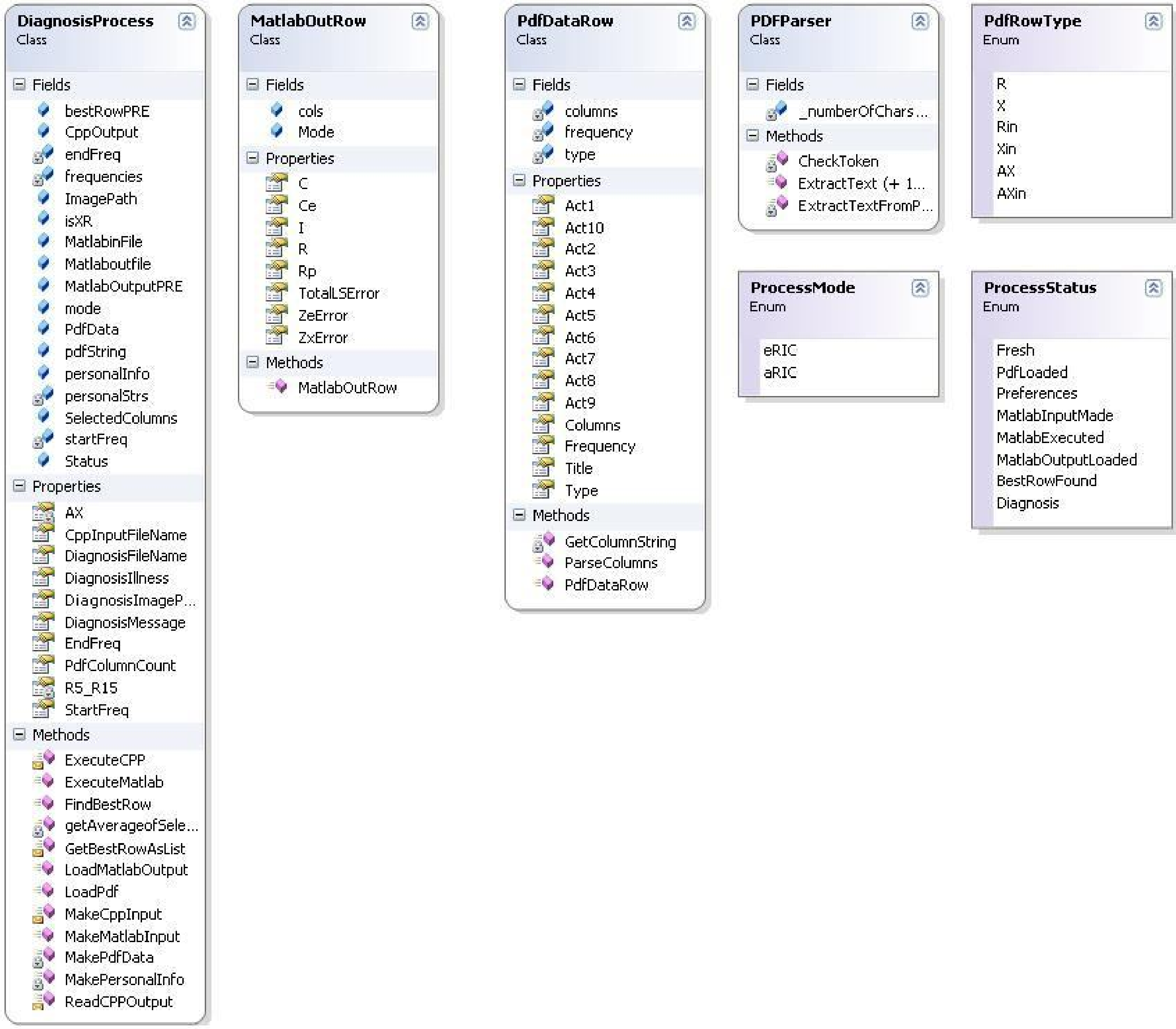
To run this program you need these prerequisites:

- Microsoft .NET Framework 2.0 or higher
- MATLAB Component Runtime 7.6

APPENDIX B. Independence relationship between assemblies of ISP



APPENDIX C. Attributes and methods of main classes in ISP



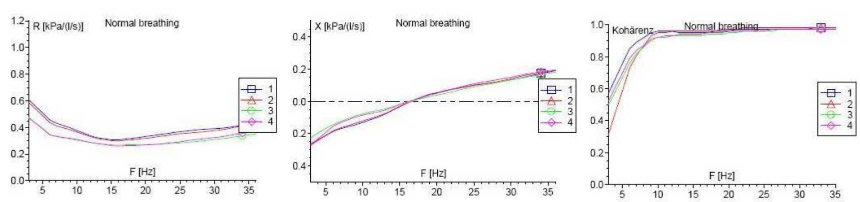
APPENDIX D. Sample IOS file

Southwest Pulmonary Research, Inc
Los Angeles, CA 90034
TEL 310-838-8874 FAX 310-838-8092

Last Name:	Talamantes	First Name:	Arely
Identification:	4497	Date of Birth:	04/04/1997
Sex:	female	Height:	150 cm
Weight:	55 kg	Diagnosis:	

		Act1	Act2	Act3	Act4	Act5	Act6	Act7	Act8	Act9	Act10
Date		06/12	06/12	06/12	06/12						
Time		02:06	02:07	02:42	02:43						
R at 3 Hz	**[kPa/L/s]**	0.61	0.58	0.47	0.47						
R at 5 Hz	**[kPa/L/s]**	0.51	0.48	0.39	0.39						
R at 10 Hz	**[kPa/L/s]**	0.38	0.36	0.30	0.31						
R at 15 Hz	**[kPa/L/s]**	0.31	0.30	0.27	0.26						
R at 20 Hz	**[kPa/L/s]**	0.33	0.32	0.27	0.27						
R at 25 Hz	**[kPa/L/s]**	0.37	0.35	0.28	0.29						
R at 35 Hz	**[kPa/L/s]**	0.42	0.42	0.34	0.36						
X at 3 Hz	**[kPa/L/s]**	-0.26	-0.27	-0.22	-0.27						
X at 5 Hz	**[kPa/L/s]**	-0.21	-0.21	-0.17	-0.19						
X at 10 Hz	**[kPa/L/s]**	-0.13	-0.12	-0.08	-0.09						
X at 15 Hz	**[kPa/L/s]**	-0.03	-0.03	-0.02	-0.02						
X at 20 Hz	**[kPa/L/s]**	0.06	0.05	0.04	0.05						
X at 25 Hz	**[kPa/L/s]**	0.10	0.10	0.09	0.11						
X at 35 Hz	**[kPa/L/s]**	0.18	0.19	0.17	0.19						
AX	**[kPa/L]**	1.28	1.23	0.85	0.93						
Resonant frequency	**[1/s]**	16.47	16.63	16.46	16.22						
Rin at 3 Hz	**[kPa/L/s]**	0.62	0.63	0.47	0.42						
Rin at 5 Hz	**[kPa/L/s]**	0.51	0.51	0.38	0.36						
Rin at 10 Hz	**[kPa/L/s]**	0.41	0.38	0.30	0.32						
Rin at 15 Hz	**[kPa/L/s]**	0.35	0.34	0.28	0.28						
Rin at 20 Hz	**[kPa/L/s]**	0.33	0.32	0.26	0.27						
Rin at 25 Hz	**[kPa/L/s]**	0.35	0.34	0.27	0.28						
Rin at 35 Hz	**[kPa/L/s]**	0.43	0.43	0.33	0.36						
Xin at 3 Hz	**[kPa/L/s]**	-0.34	-0.35	-0.30	-0.38						
Xin at 5 Hz	**[kPa/L/s]**	-0.25	-0.26	-0.21	-0.24						
Xin at 10 Hz	**[kPa/L/s]**	-0.13	-0.13	-0.08	-0.09						
Xin at 15 Hz	**[kPa/L/s]**	-0.07	-0.06	-0.03	-0.03						
Xin at 20 Hz	**[kPa/L/s]**	0.02	0.02	0.02	0.04						
Xin at 25 Hz	**[kPa/L/s]**	0.09	0.10	0.08	0.11						
Xin at 35 Hz	**[kPa/L/s]**	0.21	0.20	0.18	0.21						
AX in	**[kPa/L]**	1.50	1.49	1.02	1.05						
Rex at 3 Hz	**[kPa/L/s]**	0.58	0.54	0.46	0.46						
Rex at 5 Hz	**[kPa/L/s]**	0.50	0.46	0.39	0.40						
Rex at 10 Hz	**[kPa/L/s]**	0.35	0.35	0.30	0.29						
Rex at 15 Hz	**[kPa/L/s]**	0.27	0.27	0.26	0.24						
Rex at 25 Hz	**[kPa/L/s]**	0.38	0.36	0.30	0.30						
Xex at 3 Hz	**[kPa/L/s]**	-0.20	-0.22	-0.16	-0.12						
Xex at 5 Hz	**[kPa/L/s]**	-0.18	-0.17	-0.13	-0.12						
Xex at 10 Hz	**[kPa/L/s]**	-0.13	-0.12	-0.07	-0.08						
Xex at 15 Hz	**[kPa/L/s]**	-0.01	-0.01	-0.01	-0.01						
Xex at 25 Hz	**[kPa/L/s]**	0.11	0.10	0.10	0.11						
AX ex	**[kPa/L]**	1.17	1.08	0.70	0.76						
COin at 3 Hz		0.5	0.2	0.4	0.5						
COin at 5 Hz		0.7	0.5	0.6	0.7						
COin at 10 Hz		1.0	1.0	0.9	0.9						
COin at 15 Hz		1.0	1.0	0.9	0.9						
COin at 20 Hz		1.0	1.0	1.0	1.0						
COex at 3 Hz		0.6	0.4	0.6	0.6						
COex at 5 Hz		0.8	0.6	0.8	0.7						
COex at 10 Hz		1.0	1.0	0.9	0.9						
COex at 15 Hz		1.0	1.0	0.9	0.9						

AAIOSBMPTNCMPLET 09/04/2008 19:43



Act 1,2 = pre-BD; Act 3,4 = post-BD

IOS DX = SAI wtih 30% and 20% small and large airway BD responses respectively

Curriculum Vita

Nazila Hafezi was born on May 2nd, 1978 in Oroumiyeh, West Azerbaijan, Iran. She is the third child of Manoochehr Hafezi and Fatemeh Soltan Ahmadi. Nazila is the first member in her immediate family to earn a Master degree. She graduated from Sharif University of Technology with a Bachelors degree in Software Engineering in the winter of 2003. After graduation, she worked in Speech Processing Laboratory of Sharif University of Technology under supervision of one of her college professors. The project that she worked on with her teammates was patented in 2007 by Iran Informatics Association. She published several papers in national and international conferences on this work. After working for about five years as a software engineer and a research programmer, she started her Master degree at the University of Texas at El Paso (UTEP) in the Department of Electrical and Computer Engineering.

During her studies at UTEP, she worked as a teaching assistant in the Digital Design Laboratory and a research assistant in Biomedical Engineering Research Laboratory. She presented a paper at the 25th Southern Biomedical Engineering Conference in Miami, Florida and contributed to a paper that was presented at the European Respiratory Society during her studies as a Master student. She was admitted to the PhD program at the Department of Electrical and Computer Engineering at UTEP in Fall 2009 and will start her studies in Spring 2010.

Permanent address: 4111 Westcity Ct, Apt H162,
El Paso, Texas, 79902

This thesis was typed by Nazila Hafezi.