A STUDY OF THE PHYSIOLOGICAL EFFECTS OF SLEEP APNEA ON CEREBRAL BLOOD FLOW VELOCITY

by

GAURI BHAVE

Presented to the Faculty of the Graduate School of
The University of Texas at Arlington in Partial Fulfillment
of the Requirements
for the Degree of

MASTER OF SCIENCE IN BIOMEDICAL ENGINEERING

THE UNIVERSITY OF TEXAS AT ARLINGTON
August 2010

ACKNOWLEDGEMENTS

I would like to take this opportunity to thank Dr. Khosrow Behbehani, for providing me with the opportunity to work in his laboratory. His constant support and patience encouraged me to complete this thesis successfully. With his enthusiasm and his great effort to explain things clearly and simply, he helped make research fun for me. He has been a mentor and role model for me.

I would also like to thank Dr. Donald Watenpaugh for his valuable inputs and great ideas throughout the course of this thesis. I would also like to thank Dr. Rong Zhang for all his help and advice.

I would like to thank my research group partners for their understanding and help whenever I needed it. A special thanks to all the volunteers who participated in this study. This study would not have been possible without their time and interest. All the help and co-operation of the faculty and staff of the Department of Bioengineering is greatly appreciated.

And finally, I would like to thank my family for supporting me in every possible way. A big thank you to Viraj Gore for his support and motivation.

July 15, 2010

ABSTRACT

A STUDY OF THE PHYSIOLOGICAL EFFECTS OF

SLEEP APNEA ON CEREBRAL BLOOD

FLOW VELOCITY

Gauri Bhave, M.S.

The University of Texas at Arlington, 2010

Supervising Professor: Khosrow Behbehani

It is estimated that sleep apnea is prevalent in the US in about 18 million people. The prevalence rate is approximately 6% in the US. Sleep apnea reduces the quality of life of patients. Sleep apnea is characterized by repetitive pauses in breathing during sleep. Recent studies also show loss of gray matter in people diagnosed with sleep apnea.

To study the physiological effects of sleep apnea on cerebral blood flow velocity (CBFV), CBFV was measured in subjects simulating apnea. Sixteen volunteers (29 ± 5 yr and BMI 24.1±4.8, 9M and 7F) were recruited as control subjects. We measured CBFV with the Transcranial Doppler (TCD) and two protocols varying in the length of normal breathing between breath holds were followed for simulating apnea. The volunteers performed the maneuvers in two different positions, sitting and supine. Order of the protocols was randomized. We also studied five volunteer subjects (53.6 ± 7.4 yrs and BMI 33.66±7.27, 4M and 1F) undergoing an in-lab sleep apnea test (8 hour polysomnography). We hypothesized that there is an increase in the CBFV during sleep apnea. Various metrics were extracted from this data and comparisons were made between the measurements for the periods of normal breathing versus those during periods of apnea. The results show a significant increase in the trend of the velocity during apnea as compared to the

trend of velocity during normal breathing. In the simulated apnea study, we found that the mean slope of the trend of the maximum velocity was $0.45X10^{-3}\pm$ 0.75 $X10^{-5}$ cm/s^2 during normal breathing in the sitting position was and 0.76 $X10^{-4}$ ± $0.19X10^{-7}$ cm/s^2 in the supine position but these values increased to $0.12X10^{-2}$ ± $0.24X10^{-4}$ in the sitting position and 0.015 ± $0.112X10^{-3}cm/s^2$ in the supine position during simulated apnea. In the sleep apnea study, we found that the slopes of the trends increased from $0.25X10^{-2}$ ± $0.56X10^{-3}$ cm/s^2 during normal breathings to 0.005 ± 0.528 $X10^{-4}cm/s^2$ during hypopnea and 0.0075 ± 0.365 $X10^{-3}$ cm/s^2 during apnea. From these results, we can conclude that the CBFV increased during simulated apnea and also during apnea and hypopnea as compared to normal breathing.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

Figure Page

LIST OF TABLES

Table Page

CHAPTER 1

INTRODUCTION

1.1 Definitions and Types of Sleep Apnea

Apnea is a Greek word that means “want of breath”. Sleep Apnea is defined as the temporary loss of breathing during sleep. An apnea is a period of time during which breathing stops. Hypopnea is a decrease in respiratory flow which is not as severe as an apnea event. Apnea events usually occur during sleep and sleep is disrupted due to high levels of CO_2 and low levels of O_2 in the blood. [1] When sleep apnea is diagnosed, its severity is indicated in part using the apnea-hypopnea index (AHI). This is calculated by dividing the number of apneas and hypopneas by the number of hours of sleep.

There are three types of sleep apnea:

1) Central sleep apnea (CSA)
2) Obstructive sleep apnea (OSA)
3) Mixed or complex sleep apnea (both CSA and OSA)

CSA occurs when the brain fails to send the signal to the respiratory diaphragm to breath. In this case there is no effort made by the muscles to breath. OSA occurs when the oropharyngeal airway collapses. So, even though the respiratory muscles make an effort to breath, there is inadequate flow of air due to restriction or obstruction of the airway. Complex sleep apnea occurs when there is both CSA and OSA.

1.1.1 Prevalence

It is estimated that sleep apnea is prevalent in the US in about 18 million people. The prevalence rate is approximately 6% in the US. An estimated 10 million Americans remain undiagnosed [2]. A closer look at the statistics shows that sleep apnea is more prevalent in men than women until menopause. Prevalence increases with age. The prevalence is 24% in men and

9% in women. People more likely to have or develop sleep apnea snore loudly and are overweight or suffer from hypertension [2].

1.1.2 Physiological Effects of Apnea

Sleep apnea adversely affects the quality of life. The effects range from annoying to life threatening. Depression, hypertension, irritability and learning and memory problems have all been linked to sleep apnea. People suffering with sleep apnea are two to three times more likely to be involved in automobile crashes [3].

The National Commission on Sleep Disorders Research reports that roughly 38,000 cardiovascular deaths annually are in some way related to sleep apnea. The links include high blood pressure, heart disease and stroke. People with untreated sleep apnea face a risk of stroke that is four times higher than those not afflicted with apnea. They are also three times more likely to have heart disease. Roughly half of all hospital patients with hypertension are also afflicted with sleep apnea. Conversely around half of all sleep apnea sufferers face a diagnosis of hypertension. Sleep apnea has also been linked to congestive heart disease and abnormal heart rhythms. [4]

Recent studies show that people with sleep apnea have reduced concentrations of gray matter in multiple areas of their brain. The studies show structural changes in the cerebral cortex in people suffering from sleep apnea. Information processing takes place in the cerebral cortex. These recent findings may help explain memory, cardiovascular and other problems experienced by people with OSA [5].

1.1.3 Effects on Cerebral Blood Flow and Blood Pressure

Episodes of apnea cause blood oxygen levels to drop and carbon dioxide levels to increase. Hence during the night, the heart, brain and other vital organs are repeatedly deprived of oxygenated blood, which affects their function. [6]

Normally the brain regulates its own blood flow to meet its metabolic needs. Recent studies have found that the repeated increases and drops in blood pressure might affect the

brain's ability to regulate its own blood flow. Frequent changes in cerebral blood flow velocity have been linked to risks of stroke. [5]

High blood pressure, diabetes, congestive heart failure and high cholesterol levels are known to damage the endothelium [5]. Endothelial impairment leads to elevated blood pressures and further vessel damage. Sleep apnea is an independent factor for endothelial damage in the absence of other factors [5]. Blood pressure normally falls during sleep. Studies have shown that if the blood pressure does not decrease during the night, there is an increase in the risk of cardiovascular disease. One of the effects of sleep apnea is that blood pressure does not fall at night [5,6].

1.2 Apnea and Cerebral Blood Flow

1.2.1 Impact of Apnea on Cerebral Blood Flow

Apnea occurs when there is a cessation in breathing during sleep. Due to repetitive pauses in breathing, the oxygen saturation of blood decreases. Cerebral autoregulation tries to maintain a consistent supply of oxygen to brain tissue. To achieve this, blood flow in cerebral arteries fluctuates. There are various short term and long term effects of these fluctuations including chronic and pathophysiologically elevated cerebral blood flow. Fluctuations may weaken blood vessels. This can result in stroke. Cerebral autoregulation is only partially independent of blood pressure, such that fluctuations in blood pressure partially affect cerebral blood flow. Under normal conditions blood pressure decreases during sleep and rises again during the day. During apnea however, the blood pressure rises during sleep due in part to hypoxia and hypercapnia. Consequently, daytime blood pressure often remains higher than usual. This unusual trend of blood pressure affects the cerebral flow [7].

1.2.2 Importance of Cerebral Blood Flow for Brain Health

In adults at rest 15% of the cardiac output, which is about 750 ml per minute is supplied to the brain. The brain requires about 50 to 54 ml of blood per 100 gm of tissue per minute. This value is closely regulated. Excessive blood flow can be dangerous, as it can increase intracranial pressure and damage brain tissue by compression. Of course, brain tissue can also be damaged by low blood supply. If the blood supply falls below 8 to 10 ml per 100 gm of tissue per minute, tissue death occurs. Factors that affect cerebral blood flow are the viscosity of the blood, dilation of the vessels, and cerebral perfusion pressure. Cerebral blood flow is tightly regulated to meet metabolic needs of the brain. The measurement of cerebral blood flow in particular regions of the brain usually reflects the metabolic activity of those regions. Such studies help in functional mapping of brain activities.

1.3 Research Goal

The goal of this study is to investigate the effect of sleep apnea on cerebral blood flow velocity. We hypothesize that cerebral blood flow velocity increases during apneas. The objectives of this study are:

1) Devise quantitative features of characterizing changes in cerebral blood flow velocity that result from apnea.

2) Investigate the value of these features for distinguishing normal respiration versus apnea episodes.

3) Perform the investigation stated in controlled simulated apnea as well as in sleep apnea patients.

These objectives will characterize the physiological response of CBFV to apnea as compared to the CBFV in normal breathing. The study will consider the changes in both, the values and morphology of the CBFV response to both simulated and actual apnea.

The next chapter will discuss the methods and protocols followed to acquire the data for the controlled simulated apnea experiments as well as the methods followed for the acquisition of data from sleep apnea patients. In chapter 3, the results obtained from analysis of data are reported. In chapter 4, the physiological interpretations of the results stated in chapter 3 are discussed.

CHAPTER 2

METHODS

2.1 Cerebral Blood Flow Velocity Measurement: Using TCD

To test our hypotheses, we measured cerebral blood flow velocity using a Trans Cranial Doppler (TCD, Doppler Box, DWL, Compumedics, Singen, Germany). Specifically, we measured the velocity of the blood flow in the MCA by using a TCD instrument which works on the principle of ultrasound Doppler. This chapter discusses the basic principles of operation of the instrument and describes how the measurements of the brain blood flow velocity were made.

2.1.1 Cerebrovascular System

Cerebrovascular is defined as of or relating to the blood vessels that supply the brain.

It is this system of blood vessels that carries blood to and away form various parts of the brain. The supply of blood to the head and neck, including the brain, originates from three main vessels arising from the aortic arch: the innominate, left common carotid and the subclavian arteries.

The innominate artery is the first branch of the ascending aorta. It divides into the right common carotid artery and the right subclavian artery. The subclavian and left common carotid arteries originate independently from the aortic arch on the left side. The subclavian arteries supply blood to the upper extremities. The vertebral artery is the first branch of the subclavian artery. The extracranial vertebrals ascend upward into the foramina transversarium at the C1 level. The vertebral arteries then become intracranial as they enter the skull through the foramen magnum. The common carotid arteries arise from the base of the neck where they bifurcate into the internal carotid and the external carotid arteries at the level of C4. The following section explains the various parts of the brain and the vessels that supply blood to the various regions of the brain.

2.1.1.1 Anatomy of the Cerebral Cortex

The brain is the center of the human nervous system. The brain is divided in various anatomical structures. The externally visible parts of the brain are shown in figure 2.1.

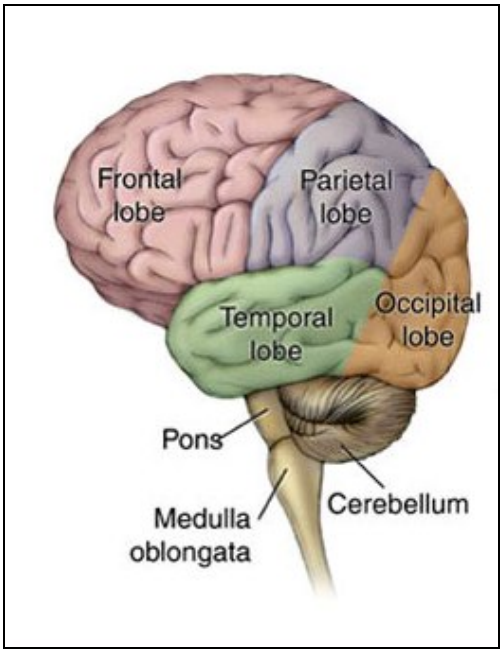

Figure 2.1: Anatomy of the Cerebral Cortex[10]

The various parts of the brain are responsible for different functions.

The frontal lobe is located at the front of each hemisphere and is positioned anterior to the parietal lobes and above and anterior to the temporal lobe. The Central Sulcus separates the frontal lobe from the parietal lobe. The lateral sulcus separates the frontal lobe from the temporal lobe. It is separated from the parietal lobe by the primary motor cortex which controls voluntary movements. The frontal lobe is associated with attention, long-term memory, planning and motivation.

The parietal lobe is positioned above the occipital lobe and behind the frontal lobe. The central sulcus separates it from the frontal lobe, the parieto-occipital sulcus separates it form the occipital lobe, the lateral sulcus separates it from the temporal lobe and the medial longitudinal

fissure divides the two hemispheres. It comprises of the somatosensory cortex and the dorsal stream of the visual sensory system. The parietal cortex maps objects, perceived visually, into body co-ordinate positions. It integrates sensory information from different modalities and determines spatial sense and navigation.

The temporal lobe is located beneath the Sylvian fissure on both hemispheres. The temporal lobe is responsible for auditory processing and contains the primary auditory cortex. It also plays a role in processing of semantics for speech and vision. The temporal lobe also contains the hippocampus which plays an important role in long term memory formation.

The occipital lobe is the smallest of the four lobes. It is the part of the cortex that lies beneath the occipital bone. It is the main center for visual processing. It contains the primary visual cortex located on the medial side of the occipital lobe. It also contains regions specialized for tasks such as visuo-spatial processing, color discrimination and motion perception.

2.1.1.2 Various Arteries Supply Blood to Different Parts of the Brain

Three main arteries supply blood to the brain.

1) Anterior Cerebral Artery (ACA)

2) Middle Cerebral Artery (MCA)

3) Posterior Cerebral Artery (PCA)

The ACA and MCA originate from the cerebral portion of the internal carotid artery. The PCA originates from the intersection of the posterior communicating artery and the basilar artery. Figure 2.2 shows the network of arteries supplying blood to the brain.

The ACA are a pair of arteries that arise from the internal carotid artery and are a part of the Circle of Willis. The areas supplied by the ACA include the medial surface of the frontal lobe, the anterior four-fifths of the corpus callosum, one inch of the lateral surface of the frontal and parietal lobe next to the medial longitudinal fissure, anterior portions of the basal ganglia and the internal capsule and the olfactory bulb and tract.

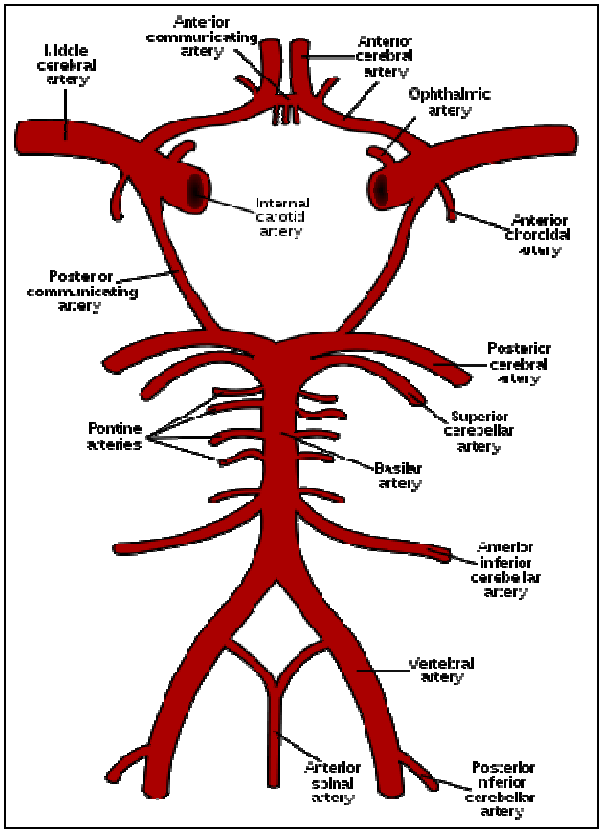

Figure 2.2 Network of Arteries Supplying Blood to the Brain[11]

The MCA arises from the internal carotid. The MCAs are not a part of the Circle of Willis. The MCA supplies blood to a bulk of the lateral surface, Broca's area, Wernicke's area, the basal ganglia and the internal capsule.

The PCA arises near the intersection of the posterior parietal artery and the basilar artery and connects with the ipsilateral MCA. It supplies blood to the occipital lobe.

Figure 2.3 shows the sections of the brain supplied by the different arteries.

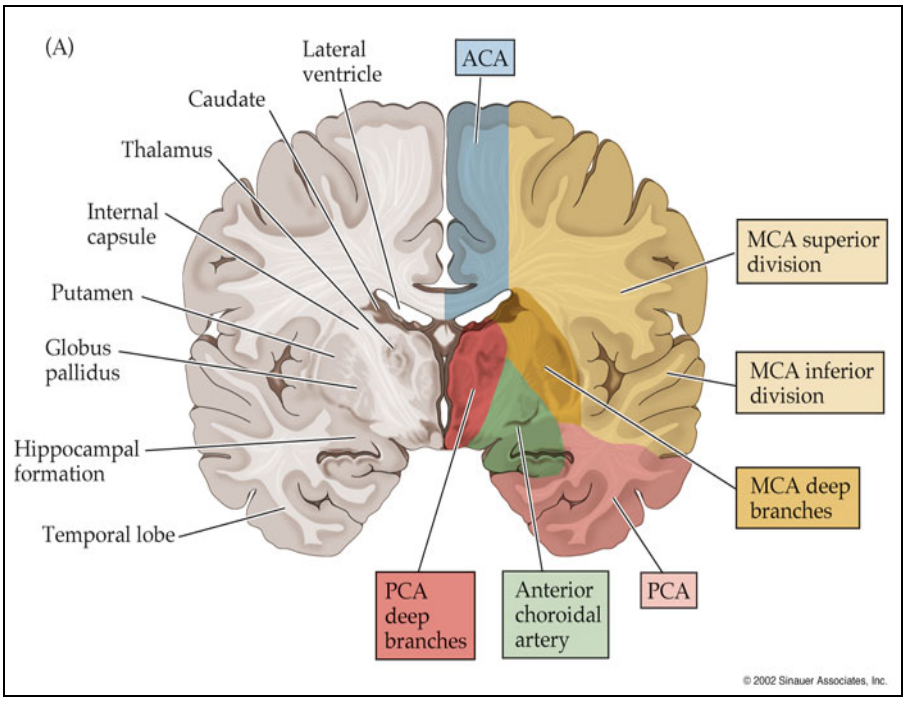

Figure 2.3 Sections of Brain Supplied by Different Arteries[10]

2.1.1.3 Circle of Willis

The Circle of Willis is a circle of arteries that supply blood to the brain. The arteries in the Circle of Willis are the Anterior Cerebral Artery, the Anterior Communicating Artery, the Internal Carotid Artery, the Posterior Cerebral Artery and the Posterior Cerebral Artery. The basilar artery and the MCA are not a part of the Circle of Willis. The physiological significance of the Circle of Willis is that it creates redundancies in the cerebrovascular system, so that even when there is a partial or complete blockage of one of the arteries or of an entire side, the other blood vessels can still maintain perfusion well enough to avoid symptoms of ischemia.

2.1.1.4 Middle Cerebral Artery

The MCA is one of the major arteries that supplies blood to the brain. It arises from the internal carotid artery and continues into the lateral sulcus. The MCA is not considered to be a part of Circle of Willis. The MCA is divided into four parts.

M1 is the sphenoidal segment. This segment perforates the brain with numerous anterolateral central arteries and irrigates the basal ganglia. The M2 segment is known as the

insular segment. The branches bifurcate and trifurcate into trunks in this segment and then the branches terminate towards the cortex. The M3 segment is sometimes grouped with the M2 segment. It extends laterally exteriorly from the insula to the cortex. The M4 segment supplies blood to the cortex. It begins on the Sylvian fissure and extends distally away on the cortex.

The MCA supplies blood to various parts of the brain. The bulk of the lateral surface of the hemisphere except for the superior frontal and parietal lobes and the base of the temporal lobe, the Broca's area which is the area responsible for language expression, the Wernicke's area which is responsible for language comprehension, the basal ganglia and the internal capsule are supplied by the MCA.

In this experiment the blood flow velocity in the MCA is measured. The mean blood flow velocity in the MCA under normal conditions (in the supine posture) is 55 cm/sec with a standard deviation of 12 cm/sec.

2.1.2 Definition of Ultrasound Doppler

Doppler ultrasound is a measurement technique that can detect and measure the velocity of flow of liquids. A particular application of this principle is the measurement of blood flow velocity. Transcranial Doppler(TCD) is a technique that measures the blood flow velocity in arteries in the brain. It is used for detection of emboli, stenosis, vasospasm and various other problems. It is relatively inexpensive and easily portable.

Ultrasound can be used diagnostically in two modalities: Continuous wave and pulsed wave Doppler. In the continuous wave modality, the transducer sends out continuous waves the ultrasound wavelength region (frequency above 20 kHz). A separate receiver receives the Doppler shifted ultrasound wave reflected from the flowing fluid. The frequency shift corresponds to the velocity of flow. Continuous wave ultrasound does not have a limit on the maximum frequency shift. Hence it does not have a maximum limit on the velocity that can be measured. However, the drawback of the continuous wave ultrasound Doppler is that all the vessels in the

path up to a point where the signal becomes too attenuated to be reflected, reflect the ultrasound signals. As a result, the continuous wave Doppler is unable to determine the specific location of the velocities that constitute the signal. Hence, continuous wave Doppler is used primarily in applications requiring the measurement of very high velocities. For example it is used in adult cardiac scanners to measure high velocities in the aorta.

In pulsed-wave Doppler, a single element transmits and receives ultrasound energy. The main advantage of a pulsed wave system is that provides velocity data from a small segment along the path of the beam. The operator can set this segment, known as the sample volume. One disadvantage of pulsed-wave Doppler is that it has a limited range and cannot measure velocities beyond a maximum limit. It can measure in the physiologic range. It cannot record faithfully, velocities over 1.5 to 2 m/sec. The CBFV in humans is generally less than 1.5 m/s hence the operating range is adequate for the measurement of CBFV. The maximum value that can be faithfully recorded depends on the range of location of the sample volume. These properties of the pulsed Doppler modality are discussed in the next section. We used pulsed-Doppler to measure cerebral blood flow velocity.

2.1.3 Principle of Working of Ultrasound Doppler

In the experiment for testing our hypotheses, we have used a pulsed ultrasound Doppler. The basic principle of the ultrasound Doppler is that, sound waves are reflected by moving targets and the frequency of the reflected waves is different from the frequency of the transmitted wave. The shift in frequency corresponds to the velocity of the moving target. In the ultrasound transducer, the piezoelectric crystal acts as both the transmitter and receiver. The piezoelectric crystal shape changes when a voltage is applied to it. Voltage applied to opposite faces of the crystal changes the dimensions of the crystal. When an alternating voltage is applied, the crystal produces pressure waves. These waves can be modulated to the ultrasonic frequencies. In the pulsed Doppler, short pressure pulses of desired frequency (in the ultrasound range) are

transmitted. In ultrasound Doppler, the transmitted ultrasound pulse travels through the tissue until it is reflected by a red blood cell. Since the red blood cells travel with a particular velocity, the frequency of the reflected wave is different from that of the transmitted wave. This shift in frequency is proportional to velocity of the red blood cells. The transmitted pulse travels for a particular amount of time in tissue. Since the velocity of the pulse in tissue is approximately constant, the reflected pulse travels with the same velocity but with a shifted frequency. Thus, the round trip time of the pulse can be easily measured. This time is directly related to the distance of the red blood cell that reflected the beam, from the transducer surface. The size of the sample volume can be changed in pulsed wave systems. The pulse repetition frequency is constrained by the range of the sample volume. The time between pulses must be sufficient for each pulse to complete the return trip from the transducer to the red blood cell and back. If the second pulse is sent before the first returns, there would be ambiguity in the discrimination of the origin of the pulses and so ambiguity in the determination of sample volume would ensue. This gating of pulses according to sample volume is range gating. Consequently, the range gating is dependent on a timing mechanism that only samples returning data from a particular sample volume. Hence, the operator can select the depth of the measurement.

At the particular depth of sample volume, the ultrasound pulses are transmitted and reflected by the red blood cells. There is an apparent change in the frequency of the reflected pulse. This shift in frequency can be measured and bears a direct relation with the relative velocity of the red blood cell that reflected the pulse. A pulse wave system is shown in the Figure 2.4.

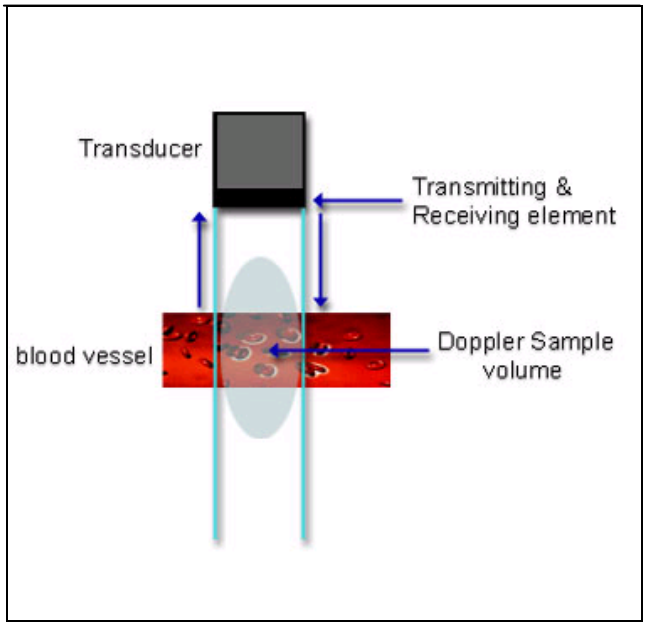

Figure 2.4 Pulsed Wave Ultrasound System [6]

The sample volume is a three dimensional tear drop shape. Its volume varies with different size and frequency of transducers. Its width depends on the width of the ultrasound beam and its length depends on the length of each pulse. [6]

The Nyquist theorem defines that aliasing will occur unless the pulse repetition frequency is at least twice the Doppler shift frequency. But, the maximum pulse repetition frequency is limited by the distance of the sample volume from the surface of the transducer. The farther the sample volume, the lower the pulse repetition frequency needs to be. In the measurement of cerebral blood flow velocity, the distance of the sample volume from the surface of the transducer is not variable. The dependence of the pulse repetition time on the distance of the sample volume from the surface of the transducer occurs because the roundtrip pulse travel time is much lower in the near field as compared to when the pulse has to travel longer distances.

The maximum recordable velocity is also related to the frequency of the transducer. A lower frequency transducer increases the ability of the pulse wave system to record high velocities. However, a lower transducer frequency lowers the signal to noise ratio. For this reason

we chose a transducer with a frequency on 2MHz which was optimum for our application. The equations used to calculate the velocity of blood flow are explained in the following section.

2.1.4 Equations Used in Calculation of Velocity

To calculate the velocity of blood flow, the area of interest (the MCA) is insonated (exposed to ultrasound waves), with the ultrasound beam with a frequency of 2MHz. These pulses are reflected by the red blood cells in the MCA. The Figure 2.5 explains the principle of the ultrasound Doppler. If a red blood cell is moving with a velocity V, with the beam to flow angle θ, the velocity can be calculated by measuring the frequency shift of the reflected wave. As the beam is reflected by the moving red blood cell, there will be a shift in the frequency of the reflected wave. The receiver can measure this frequency shift and the velocity of the red blood cell can be calculated using the following equation:

$$fd = \frac{2 \times ft \times V \times \cos\Theta}{c}$$

Where,

fd: doppler shift (Hz)

C: speed of sound in tissue (cm/s)

ft=: transmitted beam frequency (Hz)

V: velocity of blood (cm/s)

θ: angle of insonation (degrees).

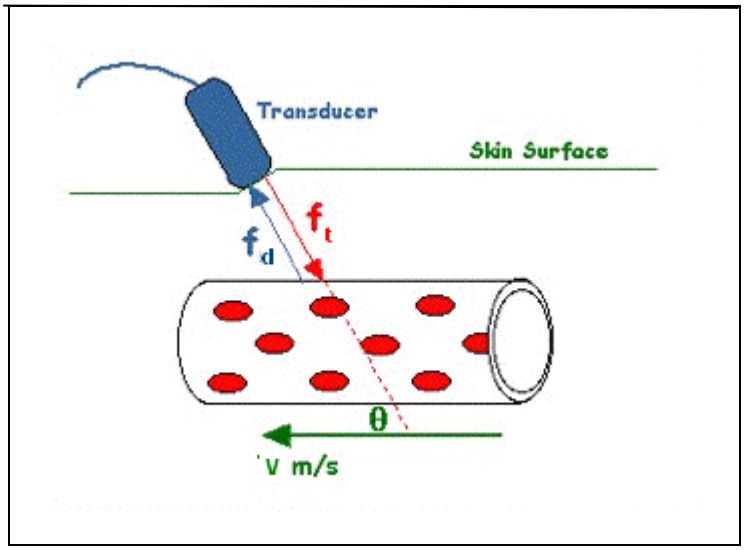

Figure 2.5 Principle of Ultrasound Doppler [6]

In this case, the transducer measures the frequency shift. The reflected waves change the dimensions of the piezoelectric transducer, which in turn produces a proportional voltage on the opposite surfaces of the crystal. This voltage is then measured and the frequency shift is calculated. This can be calculated because we know the original voltage that was applied to the transducer to transmit the ultrasonic beam and hence we know the frequency of the transmitted beam. Hence in the equation above, we know fd, the Doppler shift, ft, the frequency of the transmitted beam, C, the velocity of the beam in tissue, which is constant and theta, the angle of insonation. From the equation V, which is the velocity of blood flow is calculated from all the other parameters.

2.1.5 Importance of Angle of Insonation

Our aim is to increase the signal to noise ratio to get more accurate values of velocity. For this, it is essential to obtain a higher Doppler frequency as the voltage generated by the piezoelectric crystal is directly proportional to the Doppler frequency (fd). Sample volume is optimized with adjustment of depth. Higher Doppler frequency can be obtained if:

1) Velocity is increased

2) Higher frequency is used

3) Beam is more aligned to the direction of flow

In this case we cannot control the velocity, as velocity is the variable that we are attempting to measure. The frequency that we are using is 2 MHz this frequency is decided based on different factors which were discussed in the earlier sections. Hence, the optimum transmitter frequency for our application is 2 MHz, which cannot be adjusted. Hence the angle of the beam to the flow is the parameter that can be controlled to obtain a higher Doppler frequency. From the equation used in the calculation of Doppler frequency, it can be seen that the Doppler frequency is directly proportional to the cosine of the angle of insonation in degrees.

Given that the value of the cosine of zero degrees is one and that of ninety degrees is zero, the smaller the angle of insonation, the higher is the value of the Doppler frequency. This principle is explained in Figure 2.6 below. As shown in the figure. beam A is more aligned to the direction of flow and a higher signal is obtained. Beam B has a higher angle to the flow direction as compared to beam A, which is more aligned to the direction of blood flow . Hence, the signal produced by beam B is not as high as that produced by beam A. Beam C makes an angle of almost ninety degrees with the flow direction and hence a very low signal is obtained. The flow at D is away from the beam and hence a negative signal is obtained. The Figure 2.6 illustrates the resulting pulses for each of the four beams.

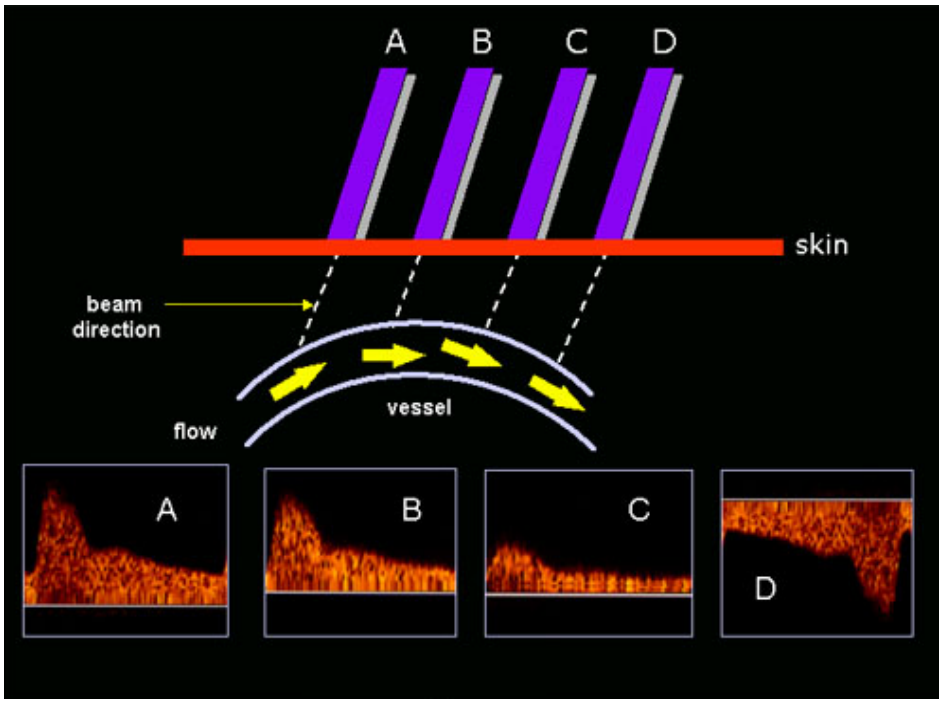

Figure 2.6 Effect of Angle of Insonation [6]

From this, it can be seen that the angle of insonation can have an effect on the absolute value of the velocity that is calculated. In an ideal case, the angle of insonation should be zero. However it is not possible to achieve this angle for every measurement. Hence a small error is expected to occur in the absolute value of velocity that is calculated. Studies have shown that the angle of insonation should be kept below sixty degrees since the cosine function has a steeper curve above this angle and errors in angle correction will be magnified. However, most of the results that are presented here consider comparisons between values obtained for each particular individual. During the measurement, the angle is adjusted once and is then kept constant more or less throughout the test. Hence the error is almost completely eliminated.

2.2 Experimental Design and Data Collection

2.2.1 Subject Demographics

For our experiment we tested two groups of volunteers. The first group was the control group. It consisted of sixteen healthy volunteers who had not been diagnosed as having sleep apnea or were not diagnosed with any other respiratory disorder. The mean age for this group was 29 years with a standard deviation of 4.9 years. The group consisted of 9 men and 7 women subjects. Table 2.1 gives the detailed demographics of this group.

Table 2.1: Subject Demographics for Control Subjects

Number of subjects	Age (years)	Gender	Height (cm)	Weight (kg)	BMI (Kg/m^2)
16	29 ± 4.9	M (9) F (7)	165.9 ± 9.3	67.2 ± 19.3	24.1 ±4.8

The second group was the patient group. The volunteers were diagnosed with sleep apnea but no other respiratory disorder. We tested 5 subjects in this group. The group consisted of 4 men and 1 woman. The average BMI was 34± 7. The table below gives details of the demographics of the second set of subjects.

Table 2.2: Subject Demographics for Sleep Lab Subjects

Number of subjects	Age (years)	Gender	Height (cm)	Weight (kg)	BMI (Kg/m^2)
5	53.6 ± 7.4	M(4) F(1)	166.1± 6.6	93 ± 23	34 ± 7

2.2.2 Instrumentation: Parameters Measured

While the subjects performed the different protocols in the experiment, various physiological parameters were simultaneously measured.

ECG Measurement

ECG electrodes were placed on the subject's chest to record ECG signals. A BIOPAC electrocardiogram system was used to obtain the ECG signals.

CO_2 Measurement

A cannula, attached to the CAPNOGARD ETCO2 monitor (Novametrix, USA), to measure the CO_2 content in the subject's breath, was placed at the entrance of the nose. Exhaled CO_2 concentration was obtained continuously throughout the test.

Blood Pressure Measurement

Noninvasive arterial blood pressure measurement was carried out continuously using a Finapres blood pressure monitoring system (Finapres model 2300, Ohmeda Inc. Englewood, Colorado, USA). The transducer was placed around the middle or adjacent fingers.

Arterial Oxygen Saturation Measurement

Oxygen saturation (SaO_2) was measured at the forehead using a pulse oximeter (Nellcor N-100, Nellcor Inc., Hayward, California, USA).

Cerebral Blood Flow Velocity Measurement

The velocity of blood flow through the MCA was measured using a TCD Doppler Box (DWL, Compumedics, Singen, Germany). The 2MHz transducer was placed on the temple of the subject and the ultrasonic beam was adjusted to insonate the root of the MCA. Continuous measurements were recorded.

2.2.3 Experimental Protocols

Control group volunteers held their breath to simulate apnea. The duration of breath hold was as long as possible for the volunteers. The time between consecutive breath holds was varied to form two different protocols. The data from these protocols was used to derive conclusions about the effects of the frequency of apnea episodes and the effect of the changes in the physiological parameters on subsequent simulated apnea episodes. The Figure 2.7 graphically shows the two different protocols. In these experiments, the period between

subsequent simulated apneas is called ‘normal breathing’. Although during the entire period, the physiological parameters are not normal, the period is called ‘normal breathing’ to identify the segment of data.

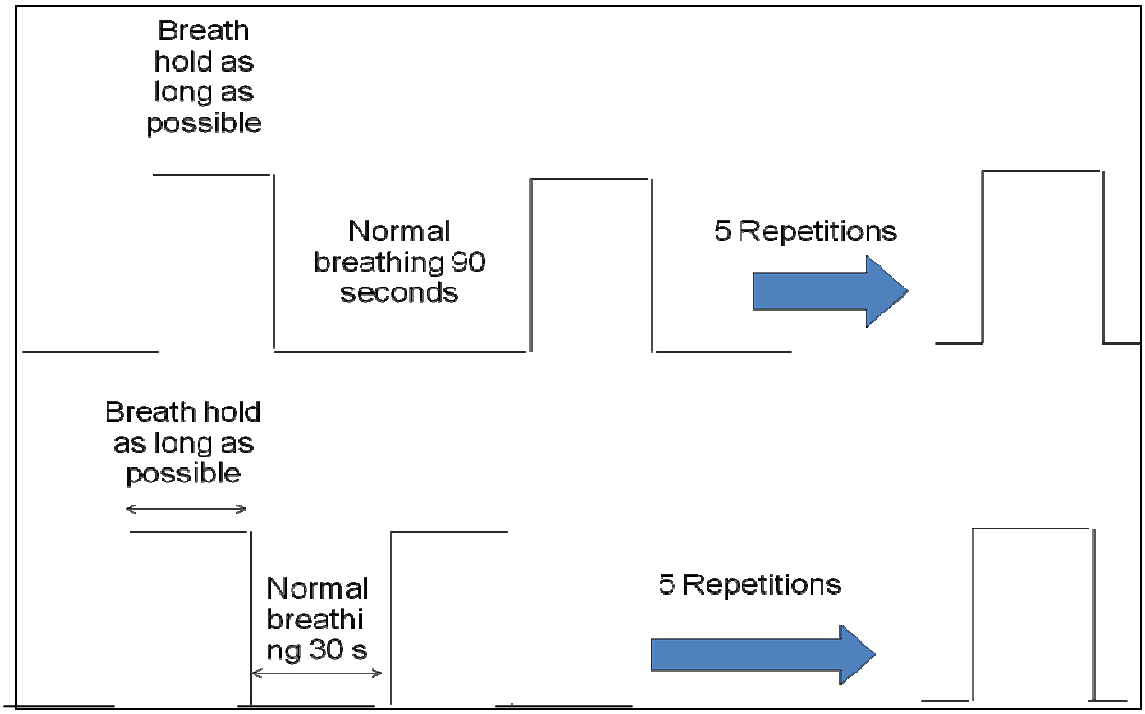

Figure 2.7 Protocols Followed during Data Acquisition (top plot shows protocol A and the bottom plot shows protocol B)

The protocol A consisted of a baseline of 60 seconds of natural breathing. Following the baseline breathing, the volunteer was asked to hold his/her breath as long as possible. After the breath hold, the volunteer was allowed to breathe normally for 90 seconds before, the next breath hold. This cycle of breath hold followed by normal breathing was repeated five times. In protocol B, the volunteer was allowed to breathe normally for 30 seconds between breath holds. The rest of the protocol were otherwise similar. The experiments in protocol A and B were each conducted with the subject in sitting and supine position. The order of the postures and the protocols was randomized for each subject.

2.2.4 Transcranial Doppler Machine: Details and Specifications

To measure the cerebral blood flow velocity, we have used a Trans Cranial Doppler machine. It is a medical ultrasound device for subcutaneously measuring the blood flow velocities in arteries and veins. The Doppler-Box^{TM} is certified by CE and meets the requirements of the 93/42/EEC – Annex II.3 Medical Device Directive. Up to now, when using the device correctly, the manufacturer knows no side effects and risks. The device can only be operated in Power Limit Mode. Maximum intensity that can be reached is 720 mW/cm^2. The depth was initially adjusted to 60mm and was then adjusted for every subject. The sample volume was adjusted to 12mm.

A 2MHz probe is used. The dimension of the Doppler-Box^{TM} is 10.5x9x27 (WxHxD in cm) and is 1.8 kgs in weight. The figure 2.9 shows the front and rear panels of the Doppler-Box^{TM}.

Figure 2.9 Front and Rear End Panels of the Doppler-Box^{TM}

The rear panel consists of an analog input, a connector for the power transformer, a network connection through which the box is connected to the PC from which the software, QL software 2.4 is run, and an analog output. The QL software controls the Doppler box and can be run on any PC with a Windows operating system. It is a one-channel software that allows normal measurements using probes of various frequencies. The raw data was acquired through the analog output connection. The analog signal was passed on to a data acquisition card and was digitized. This signal was sampled at 1KHz using a custom designed software in LabVIEW environment. The TCD analog output was sampled simultaneously along with all the other measured signal that were described earlier such as ECG, blood pressure and SaO_2.

2.2.5 Measurements

In our experiments, we measured the CBFV in the MCA of the volunteer. To focus the ultrasonic beam on the MCA, we adopted a transtemporal approach. With the volunteer in a comfortable sitting or supine position (depending on the protocol being followed) ultrasonic gel was applied to the transducer and the temporal region of the volunteer's cranium, just above the eye. The sample volume was set to 10mm and insonation depth was set as 60mm. Power and scale settings were kept low to reduce the volunteer's exposure to ultrasound. The probe was moved over the area in front of the volunteer's ear to locate the temporal window. After the MCA signal was found, it was optimized by making slight changes in transducer position and angulation. The depth and sample volume were also adjusted. To be sure that the signal obtained is from the MCA, specific rules were followed. We checked that the flow was towards the probe and that the strongest signal was found at the depth of about 60mm. Normal flow velocities in the MCA range between 50 and 75 cm/s with an average of 64 cm/s.

During the acquisition of data, the Doppler-Box^{TM} hardware is controlled using the QL software. During our experiments, we kept the Doppler-Box in the monitoring mode. Thermal Cranial Index (TIC) is a thermal index, derived by estimating the maximum heating of tissues while accounting for signal attenuation due to cranial bone, and by taking into consideration transducer frequency operating frequency, power output, and crystal size. The formula used to calculate TIC is given below:

TIC = Acoustic Power (mW) / (40 x probe diameter in cm)

For the probe that we used, the maximum TIC is 1.99.

2.3 Computer Algorithms for Processing Data

Various computer programs embodying mathematical algorithms were written to process the data in MATLAB environment. The data was acquired in LabView using a data acquisition card. During the testing of control subjects, the breath hold and normal breathing events were controlled. The volunteer was told when to hold his breath at the end of a normal breathing event and the breath hold and normal breathing events were marked using an electrical signal. The date files were visually reviewed and the segments containing the data of interest were clipped to get separate (smaller) files for the breath hold and normal breathing events. These files were then processed separately and various comparisons were made depending on the hypotheses.

For the data acquired from the sleep lab, the data was scored and the sleep lab personnel marked apnea events. This data was also clipped into apnea and normal breathing files. These files were then processed separately and various comparisons were made, similar to the control subject data.

2.3.1 Graphical User Interface for Viewing and Clipping Data

The data was acquired in LabVIEW as .lvm files. A custom-designed graphical user interface (GUI) was developed and programmed in the MATLAB environment to visualize the data and clip it as required. We acquired data simultaneously from ten channels. The sampling frequency was 1000 Hz for each channel. This data was then visualized using the GUI and was clipped. Although the GUI was created for handling this particular type of data, it has been made flexible to handle different types of files, with different number of channels and various sampling frequencies. It has the capability of visualizing large amounts of data at a time and saving various parts of the data as separate binary files. The Figure 2.9 shows a screen shot of the GUI. The functionality provided in this software is explained below:

Choosing the file to be viewed: The first step to using the GUI is choosing the file which is to be viewed and clipped. An edit box is provided in which the name of the file (entire path) must be entered. A browse button is also provided. When this button is clicked, a pop-up menu will appear

and the user can choose the file to be viewed. The GUI will accept files with '.lvm' or '.mat' extensions only. This is done because the data in our experiments is stored in '.lvm' or '.mat' files.

Axes: Four sets of axes have been provided so up to four different signals can be plotted at a time. In our data, we have recorded various physiological parameters like ECG, SaO_2, CBFV and BP in synchronization. A total of ten different signals have been recorded in synchronization and are stored as separate columns in a single file for each subject. Four out of these ten parameters can be viewed simultaneously in this GUI. Hence, various combinations of four signals can be visualized at any time. The X-axis for all four signals displayed on the GUI remains the same, since the time period of the signal to be displayed is chosen by the user. The Y-axis for each signal displayed is auto scaled according to the amplitudes of the signals.

Choosing Channels: A text box is provided to choose the channels to be displayed on the axes of the GUI panels. The user can enter the channel numbers of the channels to be viewed.

Choosing Start Time: A text box is provided on the GUI where the user can enter the initial time point from which the signal is to be viewed. Since our data was sampled at a frequency of 1000 Hz, the program calculates the time with the assumption that the data is sampled at 1000 Hz. Within the displayed window, the user can enter the point from where the clip of the signal to be displayed starts. The program assumes that the time point entered is in seconds.

Choosing End Time: Two options are provided to the user in the case of choosing an end point in time for the signal to be displayed. The user can either enter the exact time point up to which the signals are to be displayed or the user can enter the number of seconds which will determine the length of the signal to be displayed. Two separate text boxes are provided to enter these values. Either of the two values can be entered. If the user enters both values, the exact time point that has been entered is given higher priority. The display on the axes starts at the time point chosen by the user and ends at the point in time defined by the user or goes on for the number of seconds defined by the user.

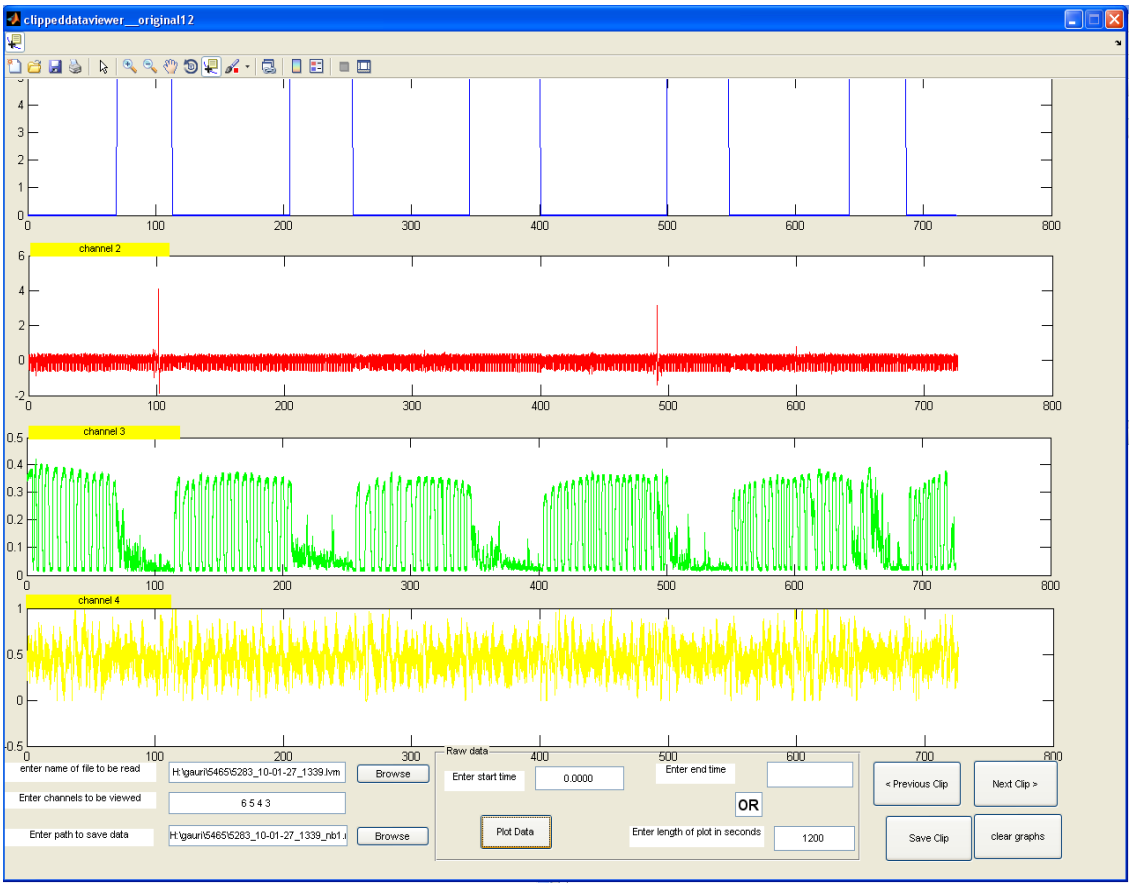

Figure 2.9 Screen Shot of the GUI for Clipping Data. The panels from top to bottom show 1) channel 1: on-off signal 2) channel 2: SaO2 signal 3) channel 3: CO2 signal 4) channel 4: end-tidal CO2 signal

Next Clip: A radio button is provided on the GUI to allow browsing through data by a window length which displays the next segment of data that has been read from the long data file on the axes. This clip of data is of the same length as the previous clip and begins at the point at which the previous clip ends. A 'previous clip' button is also provided to display the previous clip of the same length. The user can browse forward and backward using these buttons.

Clipping Data: We clipped the data collected in our experiments and stored different sections of the data in separate files. In the data collected from subjects who participated in the simulated apnea tests, where the simulated apnea and normal breathing segments were controlled, we clipped the data so that these segments were saved in separate files. For clipping the data, the

user can select the data cursor on the toolbar at the top of the GUI. After placing the data cursor on the point from which the clip is to be made, the data cursor should be clicked twice which will select the starting point of the clip. Similarly the end point of the clip can be chosen using the same data cursor and clicking the point twice. When two points are chosen as the start and end points, the smaller time point is assumed to be the start point of the clip.

Saving Clipped Data: Once the data is clipped, it can be saved in a separate file. The user can enter the path and file name of the file in which the data has to be stored. The file will not be stored if a path is not entered. A text box is provided for this purpose. Clicking on the 'Save Clip' push button will save the clip in the specified file. We saved each of the 'normal breathing' and each of the 'simulated apnea' clips in a separate file for each subject. For each subject, a convention was followed for naming the files. The file names consisted of the subject id followed by the date and time at which the data was collected, followed by the term 'nb' (for normal breathing) or 'bh' (for breath hold) followed by the number representing the count for the apnea or 'normal breathing' clip.

2.3.2 Algorithm for Detection of Peaks

Once the signals have been clipped and saved, processing is done on the clipped data. One of first steps is to detect the peaks and valleys in each clip of data. This allows one to extract time-based features from the data such as peak-to-peak interval or rise in the max CBFV. For detecting these points the following algorithms were followed. The raw data was stored in MATLAB binary files. The Figure 2.10 shows a sample of the TCD waveform that is collected.

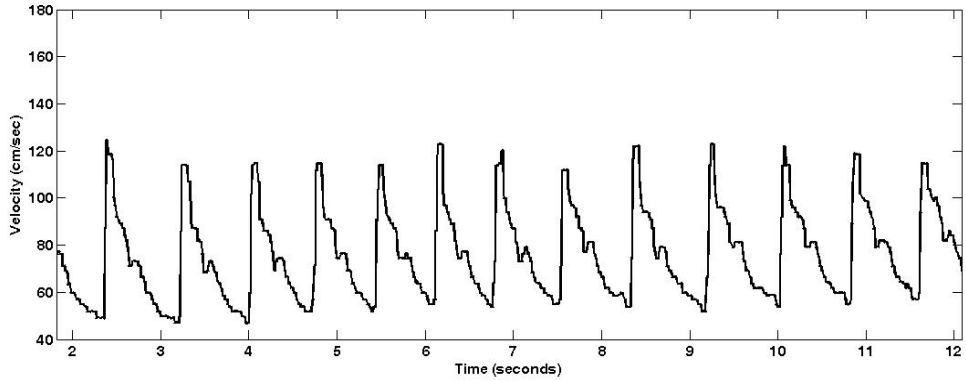

Figure 2.10 Sample of Waveform Collected

The peaks and valleys were detected in the entire clip. The peak and valley detection algorithm is given in Appendix A.

2.3.3 Algorithm for Calculating Various Metrics

Various metrics within the data were identified which were thought to potentially change during simulated apnea as compared to normal breathing. Various algorithms were used to calculate these metrics. The metrics considered are shown in the Figure 2.11. These metrics are calculated as follows.

Area under the TCD waveform: The area under the TCD waveform is calculated by adding the amplitudes of each point in each pulse in the wave (rectangular integration). The valleys are detected using the peak detection algorithm and each point between two valleys is added.

If the valleys are at time points (n) and (n+m), and x is the amplitude of the signal,

Area under the TCD waveform = $\sum_{m=1}^{m} x(n+m)$

The area under each pulse is calculated using this equation and stored in an array.

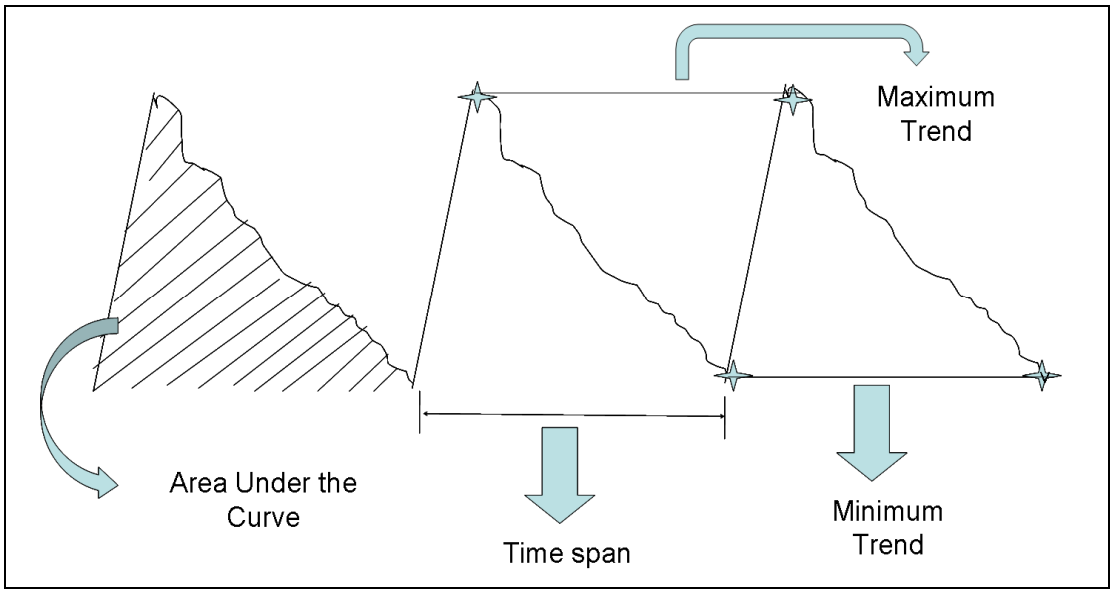

Figure 2.11 Metrics Considered

Time span: The time span was calculated by subtracting the time point of one valley from the time point of the next one. This corresponds to the R-R interval. If the time points of the valleys are (n) and (n+m), then the time span is calculated as follows

Time span= (n+m) – (n)

The time span is calculated using the above formula for each pulse and stored in a separate array.

Slope of Anacrotic Limb: The slope of the anacrotic limb of each CBFV pulse was calculated by subtracting the amplitude of each valley from each subsequent peak and dividing the difference by the time interval between these two points. If the valley appears at a point in time 'n' and has an amplitude x(n) at that point and if the subsequent peak appears at the $(n+m)^{th}$ instant of time with an amplitude y(n+m) then,

Slope of anacrotic limb = [y(n+m)-x(n)]/[(n+m)-(n)]

This is done for the anacrotic limb of each pulse and stored in a separate array.

Similarly, slope for the catacrotic limb of each pulse is calculated by subtracting the amplitude of each peak from the amplitude of each subsequent valley and dividing the difference

by the difference between the instants of time at which these points occur. If a peak occurs at the $(n)^{th}$ instant of time with an amplitude y(n) and if the subsequent valley appears at the $(n+p)^{th}$ instant of time with an amplitude x(n+p), then the slope of the falling edge is calculated as follows.

Slope of catacrotic limb = [x(n+p)-y(n)]/[(n+p)-(n)]

This is done for the catacrotic limb of each pulse and the values for the entire clip are stored in an array.

Trends: For each of the clips, trends were calculated for the peaks, valleys and means. For calculating the trends, the following procedure was followed. The peaks were interpolated using a cubic spline which was resampled. The slope between each two points is calculated and the slopes are stored in separate arrays for each clip. A similar procedure is followed for the valleys and mean values. Hence, if the amplitudes of the fitted curve at samples (n) and (n+1) are x(n) and x(n+1) respectively, the trends are calculated as given below.

Trend = [x(n+1)-x(n)]/[(n+1)-(n)]

A similar procedure is followed to calculate the trends of the valleys and means. The slopes of the trends are then stored in separate arrays for each clip, which can then be compared with the trends for other clips.

2.4. Various Metrics Considered

A number of metrics were identified from within the waveform and calculated. It was checked whether there was any change in the metrics during simulated apnea as compared to normal breathing. The metrics that were considered are shown in Figure 2.11.

2.4.1 Area under the TCD Waveform

The area under the waveform for the velocity signal represents the volume of blood flowing through the insonated artery. This metric was considered to determine whether there is any change in the total volume of blood flowing to the brain during simulated apnea as compared to normal breathing for each pulse. The area under each pulse was calculated and the values in

the clip for simulated apnea were compared with the values in the clip for normal breathing for each subject.

2.4.2 Time span

The time span of each pulse represents the length of each pulse which can be related to the heart rate. The time span of each pulse is calculated for the simulated apnea clips as well as for the normal breathing. These were then compared. A difference in the time spans would signify a change in the length of the pulses during simulated apnea as compared to the length of the pulses during normal breathing.

2.4.3 Slopes of Trends

The peaks of the velocity waveform were interpolated and the slope of this fitted waveform was considered as a metric. Similarly slopes of valleys and mean value trends were also calculated. The slopes signify the change in the velocity of the blood flow in the insonated artery. The slopes were calculated for the simulated apnea clips as well as the normal breathing clips. These were then compared. A difference in the slopes would signify that the velocity of the blood flow changes during simulated apnea as compared to the velocity of blood flow during normal breathing.

2.5. Statistics

Comparisons were made between the simulated apnea clips and the 'normal breathing' clips and the baselines in the control subjects and between the apnea and normal breathing clips in the sleep lab subjects while the subjects were asleep. The various metrics which were compared are explained in the section above. Statistics Toolbox in MATLAB and SAS were used to apply proper statistical testing as outlined below.

ANOVA: In some cases ANOVA was used. In this test, observed variances are partitioned into different groups based on different sources of variation including inter-subject variation. One way

ANOVA is used to test differences among two or more independent groups. We used one-way ANOVA to test differences among normal breathing, apnea and hypopnea in sleep lab data. Repeated measures ANOVA, under certain restrictive assumptions, does account for the within-subject dependence in the data. However, linear regression and ANOVA provide estimates of the population average of the rate of change or the means. No measure of between subject variability is given. Measurements on the same subject are much more similar than measurements on different subjects.

Tukey-Kramer Test: is a single-step multiple comparison procedure which applies simultaneously to the set of all pairwise comparisons The confidence coefficient for the set, when all sample sizes are equal, is exactly $1 - \alpha$. For unequal sample sizes, the confidence coefficient is greater than $1 - \alpha$. In other words, the Tukey method is conservative when there are unequal sample sizes. To work with unequal sample sizes, estimated standard deviation for each pairwise comparison is calculated as formalized by Clyde Kramer in 1956. So this procedure is called the Tukey-Kramer method. This test is a pairwise comparison test. This test is used for multiple comparisons of means and confidence levels. Multiple comparison tests enable us to infer differences between means and also to construct simultaneous confidence intervals for these differences.

Mixed Linear Model: A mixed model is a statistical model containing both fixed effects and random effects, that is mixed effects. The mixed linear model is an extension of regression and ANOVA that allows one to model the within-subject dependence and get a picture of the subject-level pattern of change, not just the population average pattern of change. Additionally, mixed linear models allow each subject to have their own time points of observations with any pattern of missing data. This model is used in settings where repeated measurements are made on the same statistical units, or where measurements are made on clusters of related statistical units.

The general linear model can be described by the equation:

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}$$

Where,

y denotes the vector of observed yi's

X is the known matrix of xij's

β is the unknown fixed-effects parameter vector,

$\boldsymbol{\epsilon}$ is the unobserved vector of independent and identically distributed Gaussian random errors.

However, many times the distributional assumption about $\boldsymbol{\epsilon}$ is too restrictive. The mixed model extends the general linear model by allowing a more flexible specification of the covariance matrix of $\boldsymbol{\epsilon}$. In other words, it allows for both correlation and heterogeneous variances, although normality is still assumed.

The mixed model is written as:

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \mathbf{Z}\boldsymbol{\gamma} + \boldsymbol{\epsilon}$$

Where,

Z is the design matrix

γ is the vector of unknown random effects parameter

The matrix Z can contain either continuous or dummy variables, just like X. The name mixed model comes from the fact that the model contains both fixed-effects parameters, β , and random-effects parameters, **γ.**

Hence, various statistical methods were used to analyze the data collected. Most of the tests were done to see if group means of selected metrics changed significantly during simulated apnea or apnea as compared to normal breathing. Tests were also carried out in simulated sleep apnea subjects to check whether there was a significant difference in the metrics due to change in the

posture of the subject (sitting or supine) or due to change in the length of the normal breathing periods between consecutive simulated apnea periods (two protocols followed). The results of these tests are presented in the following chapter.

CHAPTER 3

RESULTS

In this chapter, comparisons have been made between various data sets. The results of these comparisons are presented in this chapter. In the first section, results for the comparisons of the various datasets collected for control subjects have been presented. In the second section results for similar comparisons for sleep lab subjects are presented.

3.1 Comparisons for Simulated Apnea

As explained in chapter 2, sixteen volunteers were recruited to simulate apnea. The maneuvers were performed by these volunteers in two different postures, sitting and supine. Two protocols were followed in each of the postures. The protocols differed from each other in the length of the normal breathing periods between the simulated apnea episodes. In this section we will examine the results obtained on comparison between various datasets obtained from these experiments.

3.1.1 Separate Comparisons for Each Protocol

In this section we will present results in which comparisons are made between the simulated apnea and 'normal breathing' clips. Each protocol in each posture is analyzed separately. In protocol A, the 'normal breathing' datasets are clipped between the 55^{th} and 85^{th} second out of the 90 second inter apneic period. In protocol B, the clip considered is from the 5^{th} to the 25^{th} second of the 30 second inter-apneic period.

The following graphs show the average values obtained for various metrics. The average values and standard deviations are shown for all metrics for each state i.e. baseline, BH1, BH2, BH3, BH4, BH5, NB2, NB3, NB4 and NB5.

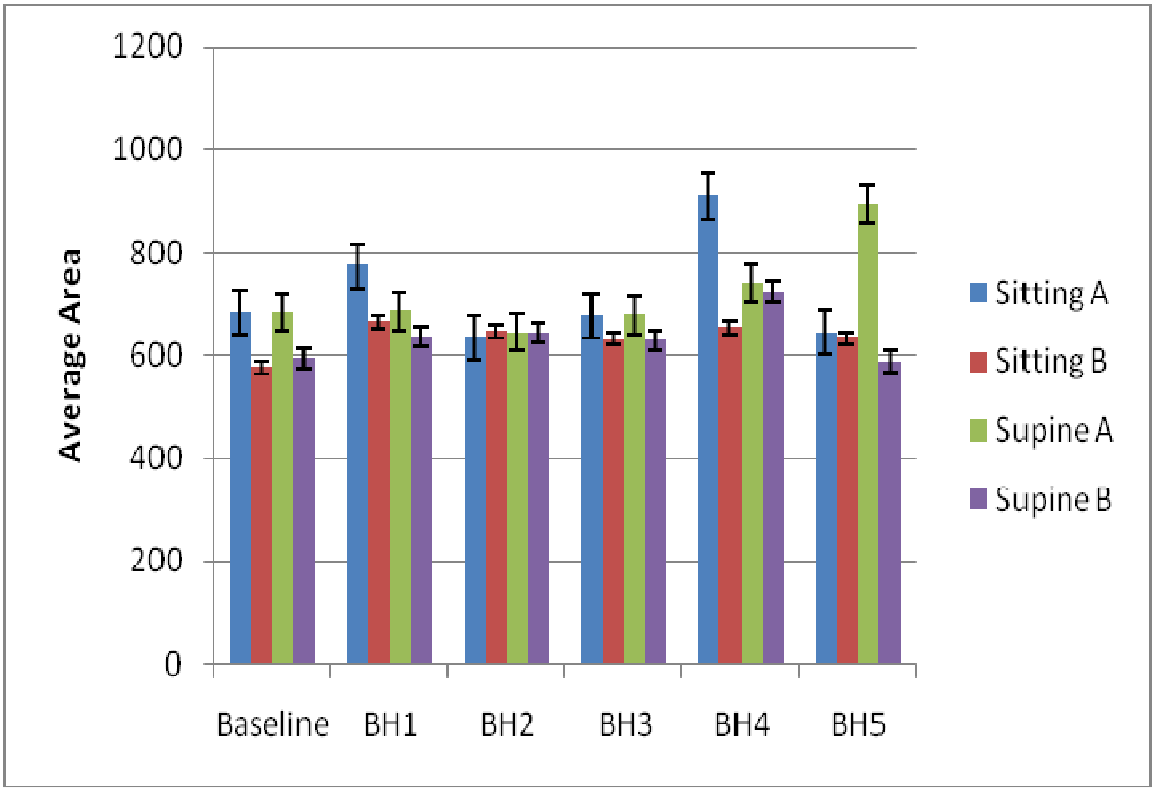

Figure 3.1 Average values of Area Under the Curve (in cm^2) for baseline and breath holds

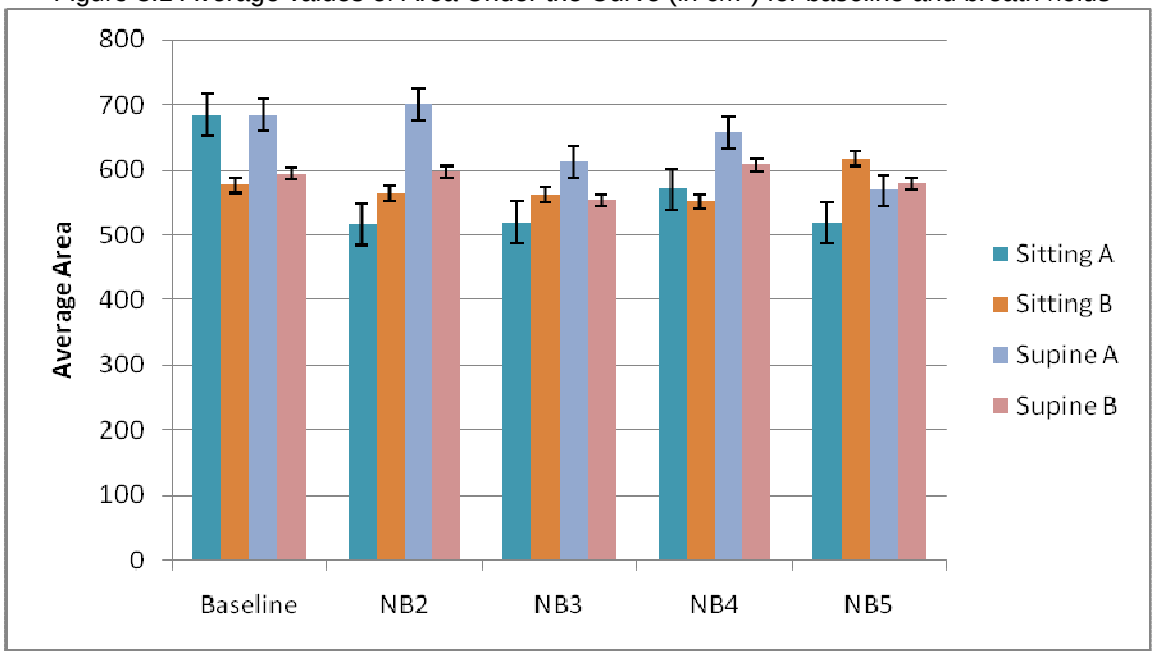

Figure 3.2 Average values of Area Under the Curve (in cm^2) for baseline and 'normal breathing'

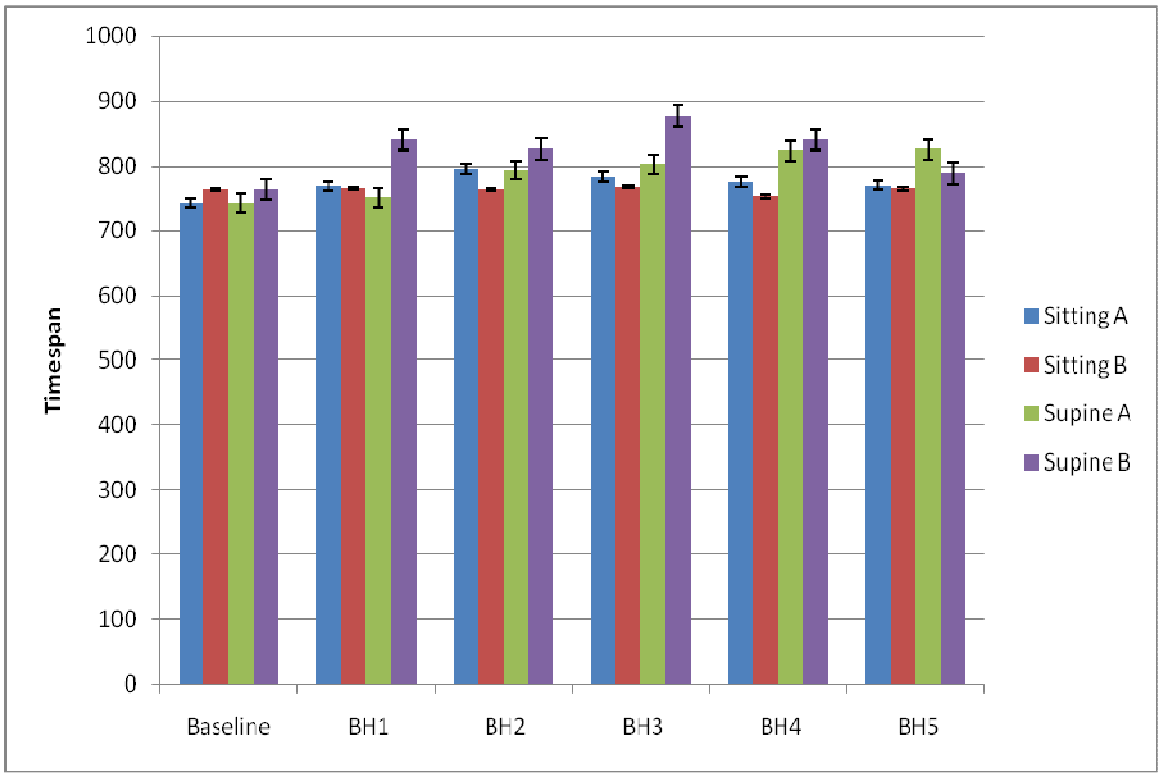

Figure 3.3 Average values of Timespan (in cm/s) for baseline and breath holds

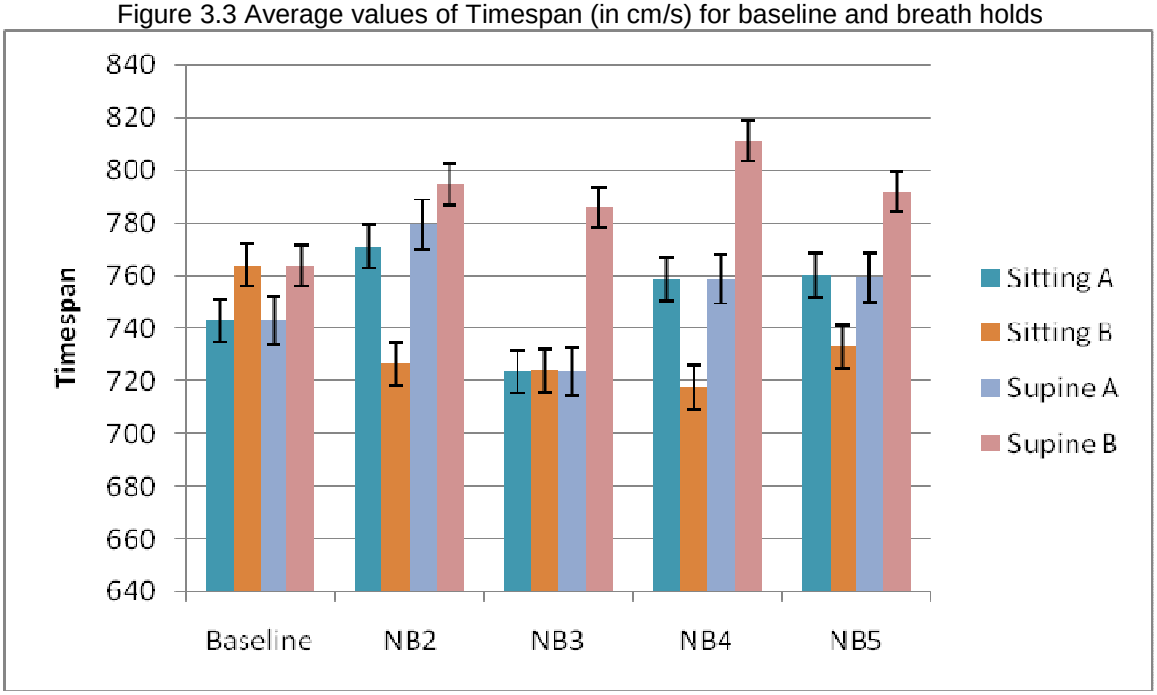

Figure 3.4 Average values of Timespan (in cm/s) for baseline and normal breathing

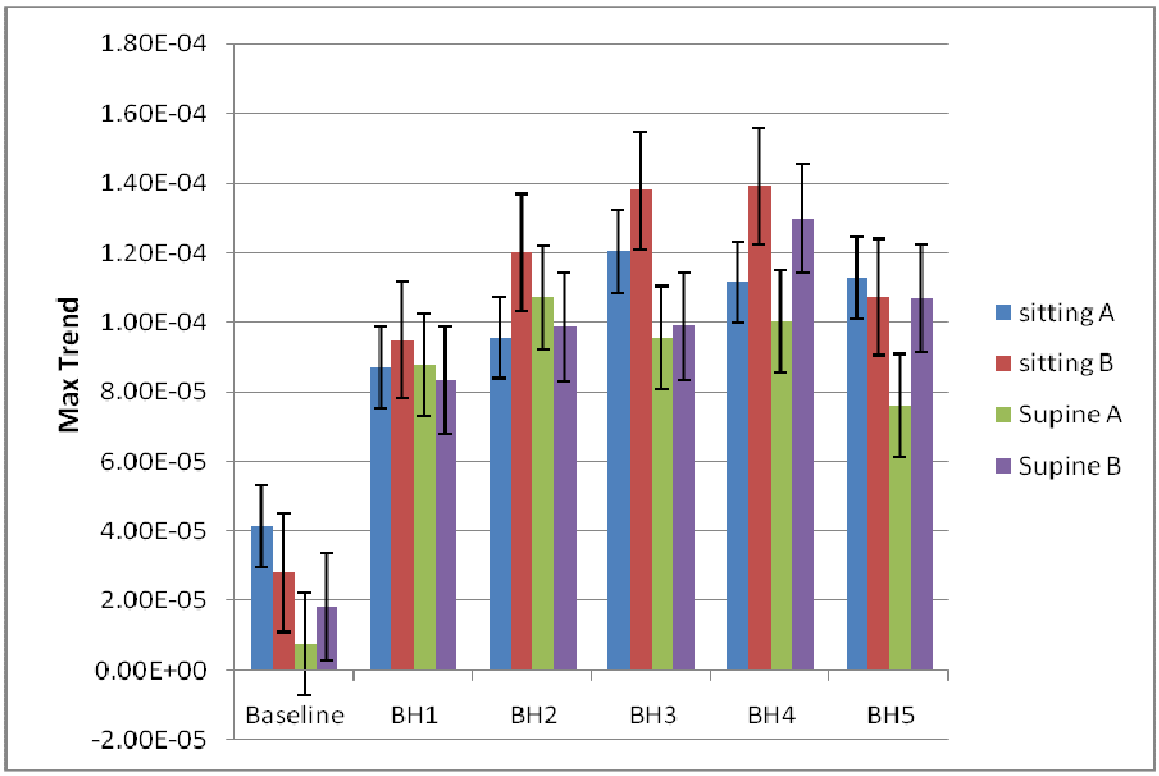

Figure 3.5 Average values of slopes of Maximum trend (in cm/s^2) for baseline and breath hold

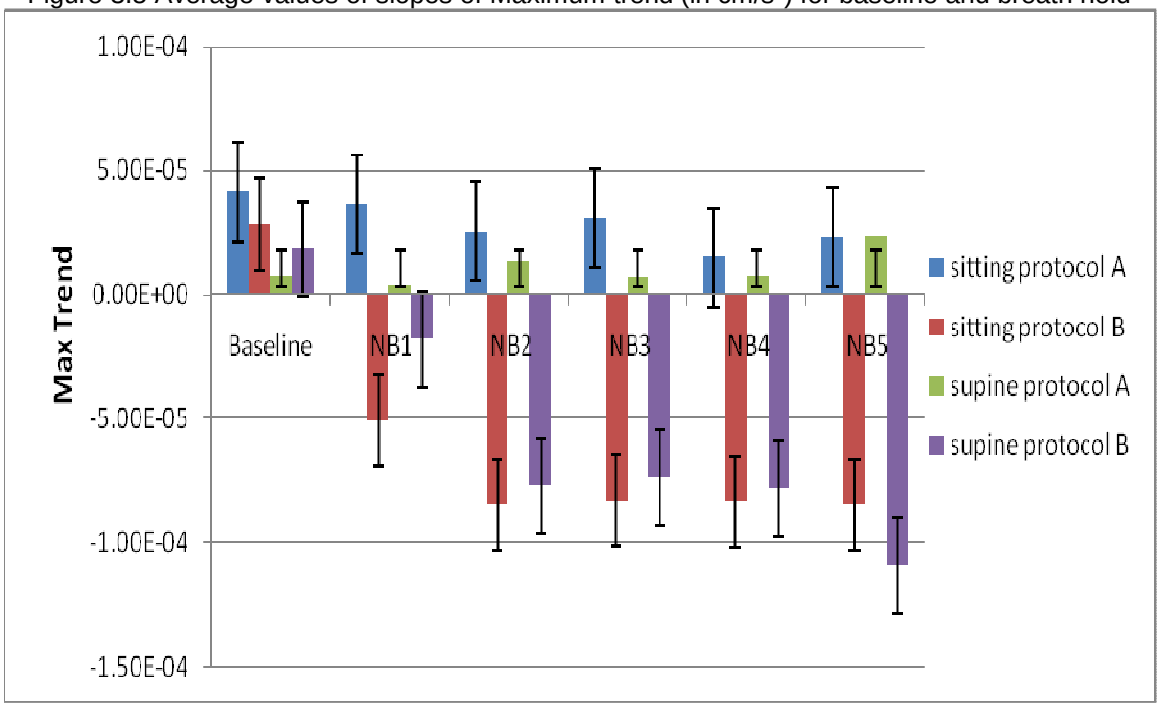

Figure 3.6 Average values of slopes of Maximum trend (in cm/s^2) for baseline and normal breathing

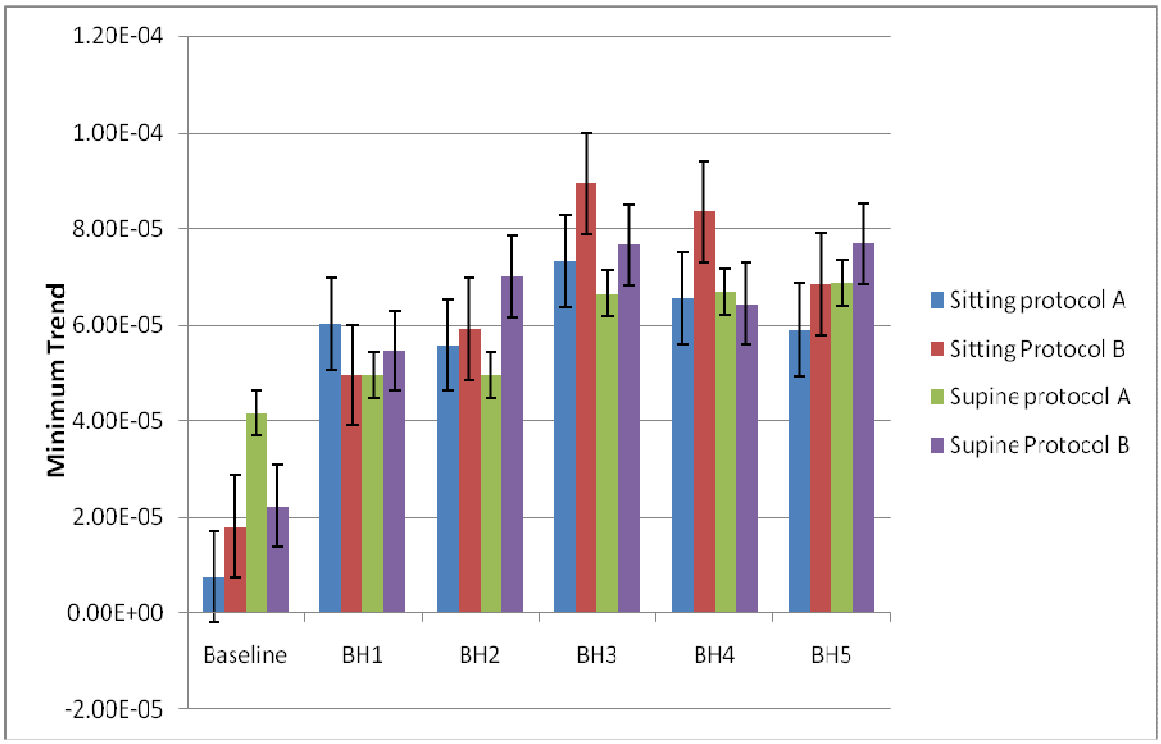

Figure 3.7 Average values of slopes of Minimum trend (in cm/s^2) for baseline and breath hold

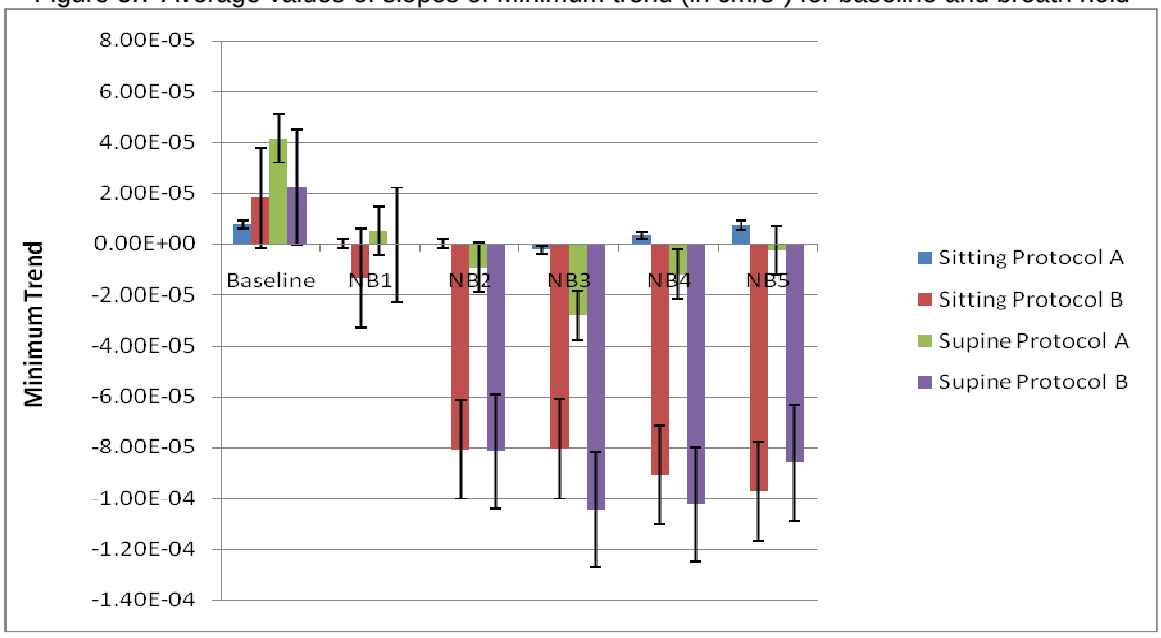

Figure 3.8 Average values of slopes of Minimum trend (in cm/s^2) for baseline and normal breathing

In the following tables, the p-values obtained upon analysis of the data using the mixed linear model are given. In each table, we compare the breath holds and the 'normal breathings' against the baselines separately for each protocol. A separate table is given for each metric. A p-value that is less than 0.05 represents a significant difference in the data sets being compared. The significantly different values are written in bold. The data for all subjects was pooled for these comparisons.

Table 3.1 Area Under Curve - Simulated Apnea Separate Comparisons for Each Protocol

	sitting A	**sitting B**	**supine A**	**supine B**
baseline Vs bh1	0.4103	0.9249	0.0868	**0.0155**
baseline Vs bh2	0.9317	0.1839	0.0893	**0.021**
baseline Vs bh3	0.5971	0.3182	0.2177	0.0672
baseline Vs bh4	0.2146	0.4632	0.1748	**0.0008**
baseline Vs bh5	0.9851	0.5459	0.2185	**0.0003**
baseline Vs nb2	**0.0033**	**0.0475**	0.0846	0.5788
baseline Vs nb3	**0.0001**	0.6529	**0.019**	0.1227
baseline Vs nb4	**0.0001**	0.4534	0.0718	0.855
baseline Vs nb5	**0.0027**	0.8375	0.1653	0.874

Table 3.2 Timespan - Simulated Apnea Separate Comparisons for Each Protocol

	sitting A	**sitting B**	**supine A**	**supine B**
baseline Vs bh1	**0.044**	0.6941	0.6918	0.1203
baseline Vs bh2	**0.0272**	0.806	0.7562	**0.0126**
baseline Vs bh3	**0.003**	0.4148	0.0541	0.0965
baseline Vs bh4	0.2417	0.9958	**0.0001**	0.1893
baseline Vs bh5	**0.0305**	0.4829	**0.0001**	0.5378
baseline Vs nb2	**0.0357**	0.055	0.2283	0.3876
baseline Vs nb3	0.5369	0.4979	0.7348	0.5839
baseline Vs nb4	0.5059	**0.008**	0.1266	0.3166
baseline Vs nb5	0.3133	**0.0072**	0.7929	0.4597

Table 3.3 Slope of Maximum Trend - Simulated Apnea Separate Comparisons for Each Protocol

	sitting A	sitting B	supine A	supine B
baseline Vs bh1	**0.0255**	0.2896	**<0.0001**	**0.0028**
baseline Vs bh2	**0.0093**	**<0.0001**	**<0.0001**	**0.0074**
baseline Vs bh3	**0.0107**	**<0.0001**	**<0.0001**	**0.015**
baseline Vs bh4	**0.0214**	**<0.0001**	**<0.0001**	**0.0007**
baseline Vs bh5	**0.0090**	**0.0004**	**<0.0001**	**<0.0001**
baseline Vs nb2	0. 1567	**<0.0001**	0. 1847	**<0.0001**
baseline Vs nb3	0.0717	**0.0022**	0.4648	**<0.0001**
baseline Vs nb4	0.1191	**<0.0001**	0. 4356	**<0.0001**
baseline Vs nb5	0.3698	**0.0008**	0. 1765	**<0.0001**

Table 3.4 Slope of Minimum Trend - Simulated Apnea Separate Comparisons for Each Protocol

	sitting A	**sitting B**	**supine A**	**supine B**
baseline Vs bh1	0.1872	0.4398	**<0.0001**	**0.0066**
baseline Vs bh2	0.4513	**<0.0001**	**<0.0001**	**0.003**
baseline Vs bh3	0.7815	**<0.0001**	**<0.0001**	**0.0002**
baseline Vs bh4	0.8935	**<0.0001**	**<0.0001**	**<0.0001**
baseline Vs bh5	0.8905	**0.0004**	**<0.0001**	**<0.0001**
baseline Vs nb2	**<0.0001**	**<0.0001**	0.2324	**<0.0001**
baseline Vs nb3	**<0.0001**	**<0.0001**	0.9599	**<0.0001**
baseline Vs nb4	**<0.0004**	**<0.0001**	0.2932	**<0.0001**
baseline Vs nb5	**<0.0001**	**<0.0001**	0.2152	**<0.0001**

Table 3.5 Slope of Anacrotic Limb - Simulated Apnea Separate Comparisons for Each Protocol

	sitting A	sitting B	supine A	supine B
baseline Vs bh1	0.3981	0.866	0.5385	0.5949
baseline Vs bh2	0.4596	0.0939	0.6906	0.6564
baseline Vs bh3	0.373	0.1023	0.1021	0.5598
baseline Vs bh4	0.4293	0.0626	**0.026**	0.6283
baseline Vs bh5	0.5454	0.2076	**0.0472**	0.7841
baseline Vs nb2	0.0838	**0.0024**	0.4524	0.5796
baseline Vs nb3	0.8929	0.8608	0.4673	0.4335
baseline Vs nb4	0.6153	**0.0004**	0.8212	0.6916
baseline Vs nb5	0.7782	**0.0001**	0.4367	0.5627

Table 3.6 Slope of Catacrotic Limb - Simulated Apnea Separate Comparisons for Each Protocol

	sitting A	sitting B	supine A	supine B
baseline Vs bh1	**0.0047**	**0**	**0.0023**	**0.0064**
baseline Vs bh2	**0.0458**	0.1671	**0.0025**	**0.0024**
baseline Vs bh3	<0.0001	0.1708	**0.0001**	**0.0068**
baseline Vs bh4	**0.0475**	0.0962	**0.0346**	**0.0063**
baseline Vs bh5	**0.0167**	0.3481	**0.001**	0.2036
baseline Vs nb2	0.7407	0.1029	0.7407	0.3759
baseline Vs nb3	0.8798	0.361	0.8798	0.3531
baseline Vs nb4	0.6579	**0.0074**	0.6579	0.8716
baseline Vs nb5	0.678	**0.0349**	0.678	0.3294

3.1.2 Comparisons Across Positions and Protocols

In this section, we present results calculated on comparisons between the various simulated apnea, baseline and 'normal breathing' clips across the two postures and protocols in which the subject performed the maneuvers.

In the following table, we present the p-values obtained upon analysis of the data using the mixed linear statistical model. We compare the baseline data in the Sitting position in Protocol A against the baseline data in sitting position in Protocol B and so on.

A separate table of p-values is given for each metric. A p-value which is less than 0.05 signifies a difference in the means of the data sets being compared. The data for all subjects was pooled for these comparisons.

Table 3.7 Area Under Curve - Simulated Apnea Comparisons Across Positions and Protocols

	Sit A vs. Sit B	**Sup A vs Sup B**	**Sit A vs Sup A**	**Sit B vs Sup B**
Baseline	0.6377	0.42	0.6736	0.4262
NB2	0.5638	0.8795	0.8024	0.3087
NB3	0.4568	0.7499	0.7913	0.5277
NB4	0.5423	0.8278	0.9918	0.4107
NB5	0.7664	0.8979	0.5679	0.4367
BH1	0.6151	0.6808	0.4342	0.9204
BH2	0.5078	0.6325	0.3932	0.7813
BH3	0.766	0.6563	0.6273	0.8083
BH4	0.5855	0.7528	0.5855	0.9114
BH5	0.8922	0.8077	0.7659	0.8008

Table 3.8 Timespan - Simulated Apnea Comparisons Across Positions and Protocols

	Sit A vs. Sit B	Sup A vs Sup B	Sit A vs Sup A	Sit B vs Sup B
Baseline	0.7309	0.8252	0.6311	0.7628
NB2	0.2232	0.6301	0.8529	0.0926
NB3	0.5708	0.5235	0.6726	0.0765
NB4	0.2589	0.4147	0.5151	**0.0155**
NB5	0.3897	0.4349	0.8647	0.0968
BH1	0.7724	0.2315	0.5855	0.2215
BH2	0.4547	0.1205	0.4135	0.082
BH3	0.7622	0.5336	0.7914	0.2116
BH4	0.7419	0.6406	0.7419	0.1412
BH5	0.9359	0.6067	0.3228	0.6597

Table 3.9 Slope of Maximum Trend - Simulated Apnea Comparisons Across Positions and Protocols

	Sit A vs. Sit B	Sup A vs Sup B	Sit A vs Sup A	Sit B vs Sup B
Baseline	0.053	**0.0001**	**0.0001**	0.9162
NB2	**0.0052**	**0.0036**	0.9269	0.6497
NB3	**0.0001**	**0.026**	0.3131	0.3155
NB4	**0.0001**	**0.0002**	0.5823	0.198
NB5	**0.0001**	**0.0001**	0.1647	0.4897
BH1	**0.0161**	0.8698	**0.0364**	0.3847
BH2	0.2327	0.1071	0.063	0.2819
BH3	0.9799	0.7234	0.2368	0.2058
BH4	0.9973	0.4522	0.9973	0.6533
BH5	0.3011	0.2106	0.0705	0.5805

Table 3.10 Slope of Minimum Trend - Simulated Apnea Comparisons Across Positions and Protocols

	Sit A vs. Sit B	Sup A vs Sup B	Sit A vs Sup A	Sit B vs Sup B
Baseline	**0.005**	0.1045	**0.0001**	0.7388
NB2	**0.0011**	**0.0004**	0.5785	0.4069
NB3	**0.0001**	**0.0017**	0.3666	0.8942
NB4	**0.0001**	**0.0001**	0.2734	0.1679
NB5	**0.0001**	**0.0001**	0.5455	0.7389
BH1	0.1686	0.7489	0.1552	0.9779
BH2	0.3676	0.8671	0.7558	0.8422
BH3	0.4058	0.8245	0.3689	0.9468
BH4	0.173	0.6128	0.173	0.582
BH5	0.7235	0.7658	0.1574	0.4019

Table 3.11 Slope of Anacrotic Limb - Simulated Apnea Comparisons Across Positions and Protocols

	Sit A vs. Sit B	Sup A vs Sup B	Sit A vs Sup A	Sit B vs Sup B
Baseline	0.9622	0.5737	0.9287	0.5611
NB2	**0.0344**	0.8542	0.7811	**0.0465**
NB3	0.063	0.5395	0.6467	**0.0483**
NB4	0.0637	0.7523	0.9151	**0.0419**
NB5	0.1671	0.3849	0.4454	0.0962
BH1	0.98	0.3021	0.7375	0.3407
BH2	0.2662	0.3879	0.5396	0.1593
BH3	0.5253	0.829	0.7488	0.2361
BH4	0.9877	0.877	0.9877	0.1783
BH5	0.597	0.8528	0.4771	0.3666

Table 3.12 Slope of Catacrotic Limb - Simulated Apnea Comparisons Across Positions and Protocols

	Sit A vs. Sit B	Sup A vs Sup B	Sit A vs Sup A	Sit B vs Sup B
Baseline	**0.0007**	0.7136	**0.0067**	0.056
NB2	0.176	0.8047	0.9357	0.1395
NB3	0.1711	0.8129	0.9695	0.105
NB4	0.2558	0.8536	0.8593	0.1245
NB5	0.2993	0.4159	0.9853	0.0583
BH1	0.6282	0.689	0.7954	0.3198
BH2	0.3923	0.8487	0.7507	0.1792
BH3	0.4162	0.8596	0.6233	0.2404
BH4	0.7857	0.8989	0.7857	0.2368
BH5	0.6038	0.5703	0.3525	0.3532

3.2 Processing of Sleep Lab Data

For analyzing the data collected during nocturnal polysomnography performed on five subjects, the data was clipped into segments of normal breathing, hypopnea apnea. The data for all subjects was then pooled and comparisons were performed using one-way-anova and the Tukey-Kramer test. The following graphs show the average values of the various metrics during the normal breathing, hypopnea and apnea episodes.

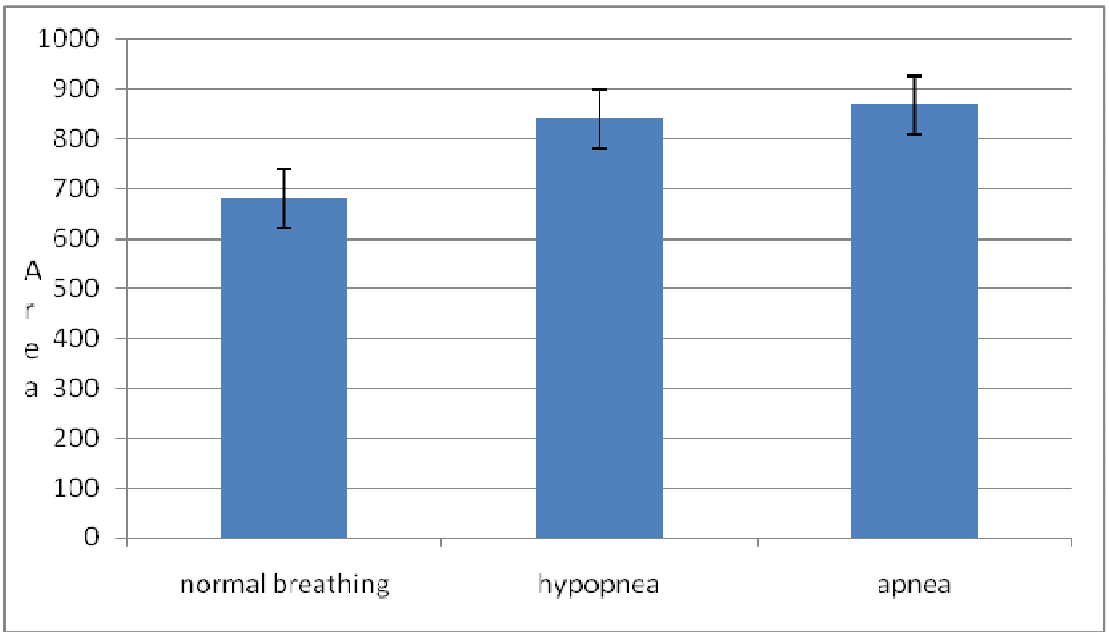

Figure 3.9 Average values of Area Under Curve (in cm^2) for sleep apnea study

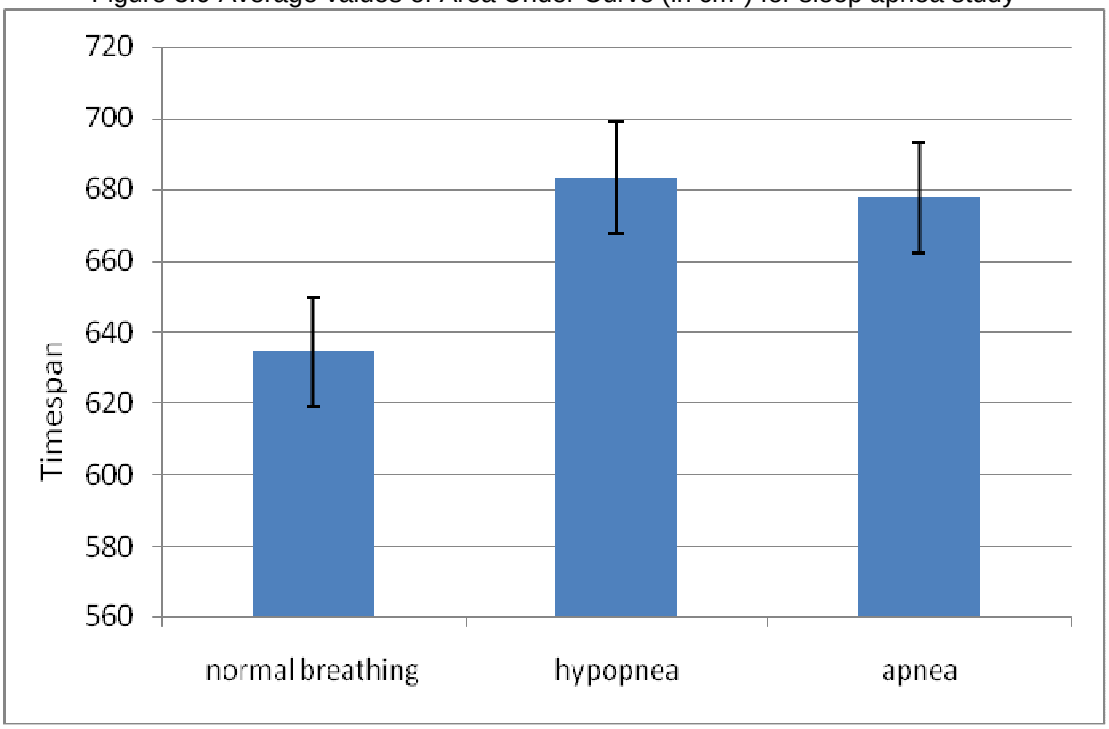

Figure 3.10 Average values of Timespan (in cm/s) for sleep apnea study

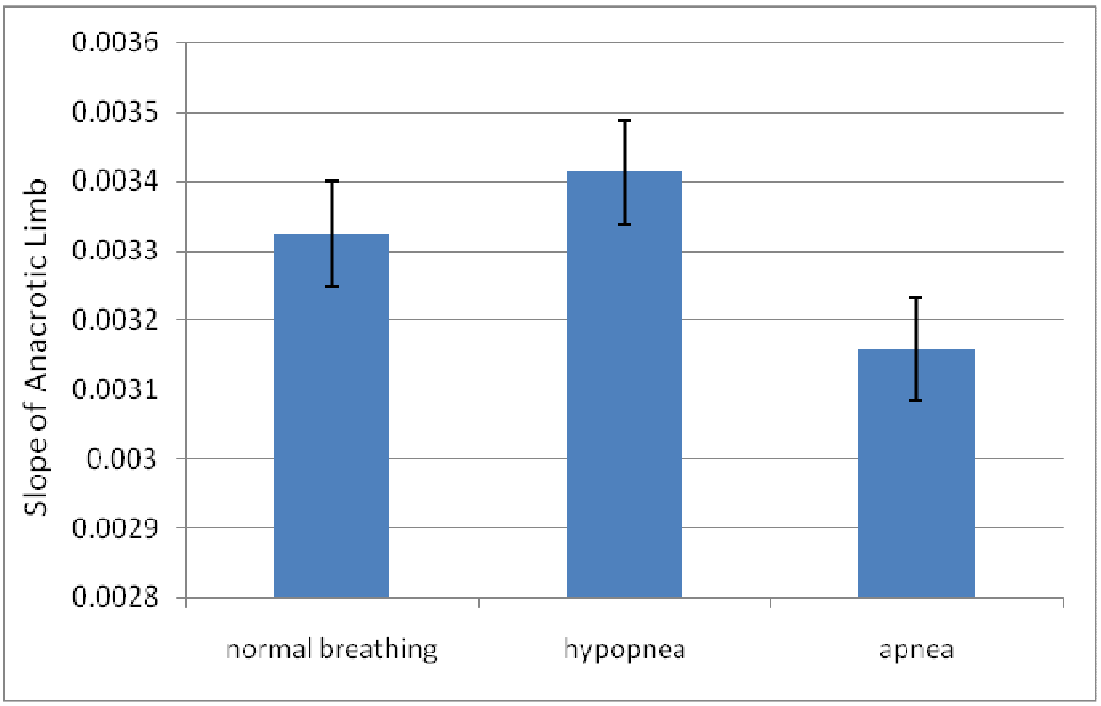

Figure 3.11 Average values of Slope of Anacrotic Limb (in cm/s^2)for sleep apnea study

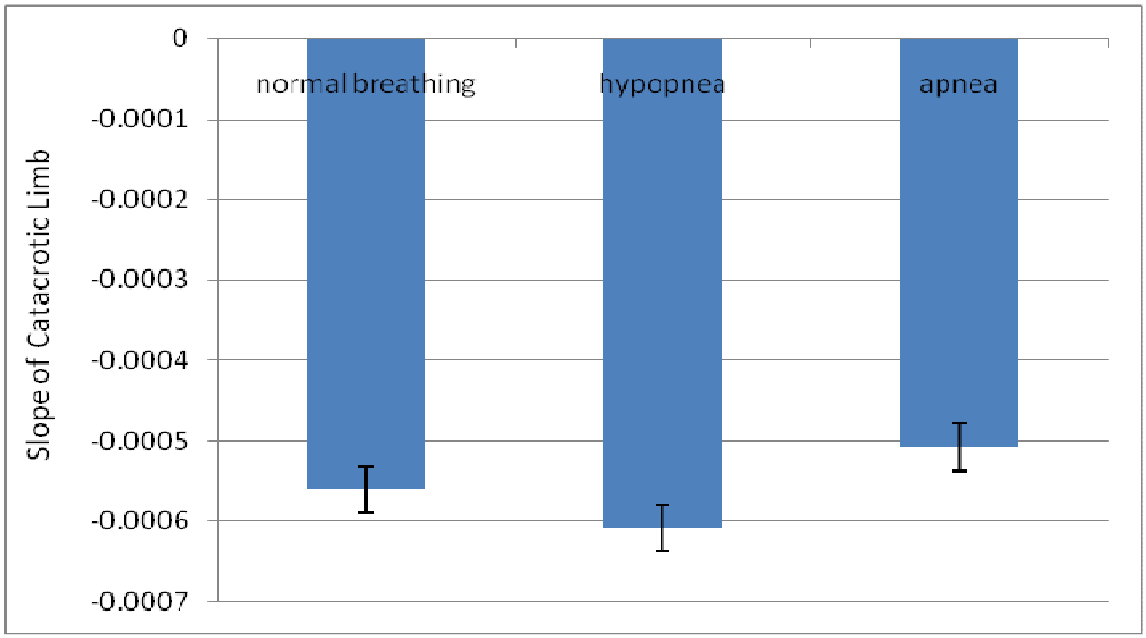

Figure 3.12 Average values of Slope of Catacrotic Limb (in cm/s^2) for sleep apnea study

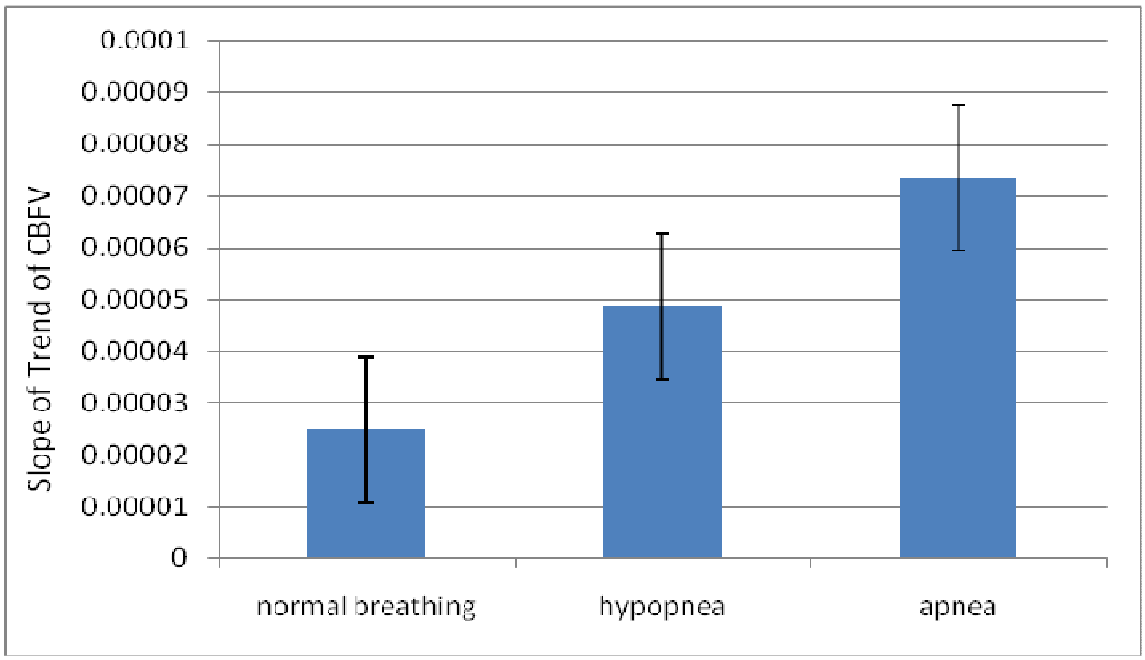

Figure 3.13 Average values of Slope of Trend of Maximums (in cm/s^2) for sleep apnea study

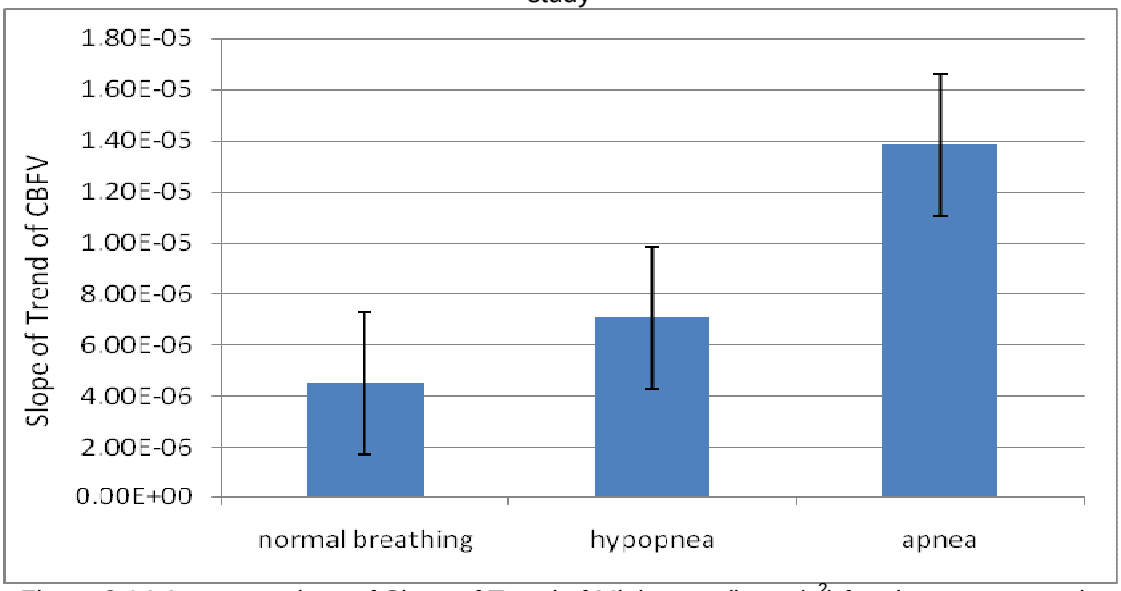

Figure 3.14 Average values of Slope of Trend of Minimums (in cm/s^2) for sleep apnea study

The following tables show the p-values obtained by analyzing the data sets using the Tukey-Kramer method. Comparisons are made between normal breathing and apnea, normal breathing and hypopnea and between hypopnea and apnea.

Table 3.13 P-values for comparisons of sleep lab data

	Confidence Interval		P value
Area Under Curve			
Normal Breathing Vs. Hypopnea	-209.538	-166.456	<0.0001
Normal Breathing Vs. Apnea	-160.186	-182.122	<0.0001
Hypopnea Vs. Apnea	-27.812	-50.634	0.0043
Timespan			
Normal Breathing Vs. Hypopnea	-43.207	-64.026	<0.0001
Normal Breathing Vs. Apnea	-48.608	-69.809	<0.0001
Hypopnea Vs. Apnea	5.401	-16.565	0.566
Slope of Anacrotic Limb			
Normal Breathing Vs. Hypopnea	-0.00026549	0.00008637	0.0296
Normal Breathing Vs. Apnea	-0.00001287	0.00034544	0.0013
Hypopnea Vs. Apnea	-0.00044224	-0.00006946	0.2328
Slope of Catacrotic Limb			
Normal Breathing Vs. Hypopnea	0.00001136	0.00008325	0.002
Normal Breathing Vs. Apnea	-0.00008958	-0.00001638	0.0007
Hypopnea Vs. Apnea	0.0000622	0.00013837	<0.0001
Slope of Trend of Maximums			
Normal Breathing Vs. Hypopnea	0.0000404	0.00005699	<0.0001
Normal Breathing Vs. Apnea	0.00001647	0.00003304	<0.0001
Hypopnea Vs. Apnea	0.00001563	0.00003225	<0.0001
Slope of Trend of Minimums			
Normal Breathing Vs. Hypopnea	-0.000013	-5.66E-06	<0.0001
Normal Breathing Vs. Apnea	-0.0000105	-3.10E-06	<0.0001
Hypopnea Vs. Apnea	-6.25E-06	1.12E-06	0.1032

CHAPTER 4

DISCUSSION

In this chapter, we discuss the results presented in the previous chapter. Interpretations are made about the physiological effects of the various results. Initially, we discuss various metrics considered and their physiological interpretations. Next, we discuss the physiological interpretation of the results. In the last section, we discuss the limitations and future scope.

4.1 Choice of Metrics

As explained in chapter 2, sixteen volunteers were recruited to simulate apnea. The maneuvers were performed by these volunteers in two different postures, sitting and supine. Two protocols, A and B, were followed in each of the postures. The protocols differed from each other in the length of the normal breathing periods between the simulated apnea episodes, thus the frequency of simulated apneas were higher in protocol B. For the data collected in the sleep lab, the data was scored by the sleep lab personnel and then the clips were compared with normal breathing clips. The results obtained by these comparisons are given in chapter 3. In this section, physiological interpretation of the metrics will be presented. It will then be followed by discussion of the results obtained from comparison of the metrics for the protocols A and B.

The velocity profile of the cerebral blood flow is pulsatile in nature. Like the blood pressure waveform, the CBFV waveform increases up to a maximum point and falls back to a minimum point during each pulse. The CBFV is controlled by the basic mechanism of auto regulation. Auto regulation is defined in terms of vascular resistance changes or simply arteriolar caliber changes as blood pressure or perfusion pressure varies.

4.1.1 Area Under the Velocity Curve

The first metric that was considered for comparison was the area under the velocity curve. For each heart beat, the CBFV went from a minimum value to a maximum value and back to a

minimum value. The area was calculated separately for each pulse and compared. The number of area points for each subject for each simulated apnea depended on the length of the breath hold and the heart rate for that subject. Hence, the number of area points obtained varied in the case of each simulated apnea. The area under the velocity curve is the integration of velocity over time. Hence, if one can make an assumption that diameter and mechanical property of the MCA remain the same throughout a given pulse, the area under the curve will be proportional to the volume of blood flowing through the MCA during each pulse.

4.1.2 Time Span

The time span is the time between two consecutive valleys in the series of CBFV pulses. The time span represents the heart rate as each pulse represents each heart beat. When the time span is compared in conjunction with the area under the curve and the velocity trend, we would know if the change in CBFV is due to the change in heart rate or volume or both.

4.1.3 Slopes of Anacrotic and Catacrotic Limbs

The slopes of the rising and falling edges of each pulse were calculated. The slope of the rising edge of the CBFV pulse represents the acceleration of the blood through the MCA. This will indicate the stiffness of the MCA over time. Hence we can conclude whether the stiffness of the MCA has changed over time due to apnea. In this case, to draw conclusions about the stiffness of the MCA, we assume that the blood pressure has not changed. The slope of the falling edge represents the deceleration of the blood through the MCA at the end of the pulse.

4.1.4 Slopes of Trends

The slopes of the trends of the velocity curves represent the rate of change of velocity at the time at which they are calculated. Hence the slopes of the trends are interpreted as the rate of change in the maximum, minimum or mean velocity of the cerebral blood flow.

4.2 Physiological Interpretation of Results

In this section, we discuss the results presented in chapter 3 and give the physiological interpretation of those results.

4.2.1 Comparisons for Simulated Apnea

For simulated apnea, the comparisons carried out were: 1) separate comparisons for each protocol; 2) comparisons across protocols and; 3) comparisons across positions. In this section, the results obtained for these comparisons are discussed.

4.2.1.1 Separate Comparisons for Each Protocol

The tables in 3.1 to 3.6 give the p-values obtained on comparisons between the simulated apnea clips and the baseline at the beginning of the tests for each protocol in each position. P-values are also obtained for comparisons of the baseline with the 'normal breathing clips for each metric for each protocol.

From the tables and the graphs of the average values, we can see that that the area under the curve, which is the volume of blood flowing through the MCA changes significantly during simulated apnea as compared to the baseline. Also, the trend of the velocity changes significantly. These results match the results obtained from a parallel study of the blood pressure waveform by Raichel Alex. The blood pressure data was acquired synchronously with the CBFV data. However, there is no significant change in the time span of each pulse, or in the slope of the rising or falling edge of each pulse. Findings, from a parallel study by Aditya Bashaboyina focusing on ECG data acquired synchronously with the CBFV data, state that the heart rate shows a significant change during simulated apnea as compared to normal breathing. However we find that the CBFV does not show this change. To explain this phenomenon, we can speculate that this might be due to the auto regulation of the CBFV. In this case we assume that the diameter of the CBFV is constant.

When we consider the comparisons of the baseline with the normal breathing clips, we see that the slopes of trends of velocity are not significantly different for the Protocol A in the two

positions but we get significant differences in the protocol B in both positions. This can be explained when we consider the clips used for this comparison. In this comparison, for protocol A, the clips that we used for normal breathing began at the 55th second after the preceding breath hold. In this case, 55 seconds were available after the breath hold for all the physiological responses to simulated apnea to return to the normal levels. In protocol B however, the clip that we considered, began 5 seconds after the preceding breath hold. Hence, this difference may be attributed to this difference that for protocol B, the physiological parameters may not have returned back to the normal levels in the normal breathing clips.

4.2.1.2 Comparisons Across Positions and Protocols

The tables 3.7 to 3.12 give the p-values obtained by comparisons of the data across protocols and positions. From the tables we see that there is no significant difference in the metrics during the comparison across positions. We do not get significant differences when we compared data from Sitting Protocol A against Supine Protocol A and so on. This shows that the posture does not significantly change the effects of apnea on CBFV.

From the tables, we can see that, when we make comparisons across protocols, i.e. Sitting Protocol A against Sitting Protocol B, we do not get significant differences on comparisons of simulated apneas. This shows that the frequency of the apnea episodes did not affect the parameters of the CBFV waveform that we considered. It also shows that the effect of the apnea episodes did not change in relation with the status of the physiological parameters prior to the apnea episode. The change in the CBFV was similar inspite of the fact that the physiological parameters were not at the same levels before the beginning of the apnea episodes in both the protocols. Hence, for these metrics it is not important whether the physiological responses had returned to the baseline value prior to the apnea event. In these tables we see significant differences in the metrics upon comparisons of the normal breathing clips across protocols. From this we can conclude that the normal breathing clips for protocol A are not the same as those in protocol B for the same posture. This may be due to the fact that in protocol A, the normal

breathing clip begins at the 55th second after the breath hold as against beginning at the 5th second after the breath hold in the case of protocol B.

From the analysis above, we can conclude that the volume of blood flowing through the MCA and the trends of the CBFV changes significantly during simulated apnea as compared to normal breathing (assuming the diameter of the MCA is constant). The cerebral auto regulation also has an effect on the CBFV due to which the rate of the pulses does not change significantly during simulated apnea. Also, we can see that the posture does not significantly change the effect of the simulated apnea on the CBFV. The frequency of the simulated apneas does not significantly change the effect of the simulated apneas on the CBFV and it also does not matter whether the physiological parameters are at normal levels before the beginning of an apnea.

4.2.2 Comparisons for Sleep Lab Data

The results for the sleep lab data are shown in figures 3.9 to 3.14 and in table 3.13. the average values for various metrics during the normal breathing, apnea and hypopnea are shown in the figures. The table gives the p-values obtained upon the comparisons of Normal breathing Vs. Apnea, Normal breathing Vs. Hypopnea and Apnea Vs. Hypopnea.

From the table we can see that there is a significant difference in the slopes of the trends during apnea and hypopnea as compared to normal breathing. However the trends are not very significant when apnea is compared with hypopnea. These results are similar to the results obtained from the simulated apnea study.

4.3 Limitations of the Study

Some inherent shortcomings in the measurement of CBFV using TCD limit the scope of measurements and the analysis of the data in certain ways. The first limitation is that it may not be possible to find a window in every volunteer and hence this method might not work for some volunteers. Also, the exact value of the velocity depends on many factors like the density of the skull bone. Hence comparing the actual values of velocity across subjects might not be

meaningful. Also, since the value of the velocity is a function of the cosine of the angle of insonation, care has to be taken not to move the transducer during the test. Since even a small change in the angle can make a difference in the values of the velocity measured, comparison of the absolute values of data acquired at distant points in not valid. For values that are acquired within a short time, one can assume that the angle has not changed.

4.4 Conclusion

In conclusion, our data demonstrates that there is a significant increase in the CBFV during simulated apnea as compared to the CBFV during normal breathing. From the results, we found that there was a significant increase in the volume of blood flowing through the MCA during the simulated apnea. Also there was a significant increase in the rate of change of the CBFV. Further, our results show that the posture or the time between consecutive apneas did not make a difference in the effects of simulated apnea on the CBFV. From the data for apnea studies we can conclude that, there was a significant rise in the CBFV during apnea and hypopnea as compared to normal breathing. These results correspond to the findings of the simulated apnea study.

APPENDIX A

MATLAB PROGRAM FOR CALCULATION OF METRICS AND GRAPHICAL USER INTERFACE FOR CLIPPING DATA

```
% PROGRAM FOR CALCULATION OF METRICS
% clear all;
% close all;
% clc;
% A=load('C:\Documents and
Settings\Administrator\Desktop\essam_clipped_data\withphysiocal_bh2.mat');
% % time=A(:,1); x=A(:,2); sync=A(:,5);
% time=A.savedata(:,1);
% x=A.savedata(:,2);
% sync=A.savedata(:,5);
%x=y;
% Savedata is the input matrix, bring in the data from the 7th column
% (channel)
y=savedata(:,7);

% t is the no. of rows in the column
t=length(y);

% set windows size
winsize=10;
% create an array of t length
time=[1:1:t];

% transpose the array: time
time=time';

% save time array into sync array
sync=time;
% x=detrend(y, 'constant');
% x=abs(x);
x=y;
% remove noise from sample
x=filter(ones(1,winsize)/winsize,1,x);


figure
plot(time,x)
hold on
plot(time,sync)
hold off
d1=x;
d2=time;

% d1=decimate(x,2);
% d2=decimate(time,2);
% d1=abs(d1);
% d2=abs(d2);
figure
plot(d1)

% create max.min arrays and specify a delta
max = [];
```

```
min = [];
delta=0.15;

% array indices
mn = 0; mx = 0;
mnpos = 0; mxpos = 0;

lookformax = 1;
% PEAK DETECTION
for i=1:length(d1)
a1 = d1(i);
if a1 > mx, mx = a1; mxpos = i;
end
if a1 < mn, mn = a1; mnpos = i;
end

if lookformax
if a1 < mx-(delta)
max = [max ; mxpos mx];
mn = a1; mnpos = i;
lookformax = 0;
end
else
if a1 > mn+(delta)
min = [min ; mnpos mn];
mx = a1; mxpos = i;
lookformax = 1;
end
end
end


v1=max(:,2);
v2=min(:,2);
index=max(:,1);
index2=min(:,1);
% d2=abs(d2);
% index=abs(index);
% index2=abs(index2);
t1=d2(index);
t2=d2(index2);
%r=decimate(index2,10);
n=length(t2);
l=0;
k=1;
movingavg=v1;
valleys=v2;
% windowsize = 5;
% movingavg=filter(ones(1,windowsize)/windowsize,1,v1);
% sp=1:index;
%  xx=0:0.1:10;
% movingavg=spline(xx,movingavg,index);
```

```
v1len=length(v1);
v2len=length(v2);
if v1len>v2len
   for i=1:1:v2len-1
      mean(i)=(movingavg(i)+valleys(i))/2;
      meanlen(i)=index2(i);
   end
   meanlen=meanlen';
   mean=mean';
   %mean=detrend(mean,'constant');
   figure
   plot(index,movingavg);
   hold on;
   plot(index2,valleys, 'g');
   hold on;
   plot(meanlen,mean, 'r');
%    figure
else
   for i=1:1:v1len-1
      mean(i)=(movingavg(i)+valleys(i))/2;
      meanlen(i)=index(i);
   end
   meanlen=meanlen';
   mean=mean';
   %mean=detrend(mean,'constant');
    figure
   plot(index,movingavg);
   hold on;
   plot(index2,valleys, 'g');
   hold on;
   plot(meanlen,mean, 'r');
   hold on;

end


for i=1:1:n-1
   timespan(i)=(((index2(i+1))-index2(i)));
   for k=(l+1):1:(timespan(i)+l-1);
      wave(k-l)=x((k+index2(i))-l-1);

   end
   sloperising(i)=(v1(i+1)-v2(i))/(index(i+1)-index2(i));
   slopefalling(i)=(v1(i)-v2(i+1))/(index(i)-index2(i+1));
   wave=wave';
   l=l+timespan(i);
   area(i)=sum(wave);
end
sloperising=sloperising';
slopefalling=slopefalling';
area=area';
```

```
%movingavg=detrend(movingavg, 'constant');
%[pmovingavg,w]=pwelch(movingavg, 16, [], 8912,100);
%plot(w, pmovingavg);
figure
plot(d2,d1);
hold on
plot(t1,d1(index),'r*');
hold on
plot(t2,d1(index2),'+r');
hold off
p=detrend(x, 'constant');
%[psd, freq]=pwelch(x, 16384, [], 262144, 1000);
%figure
%plot(freq, psd);
timespan=timespan';



%PROGRAM FOR DATA CLIPPING
function varargout = clippeddataviewer__original12(varargin)
% CLIPPEDDATAVIEWER__ORIGINAL12 M-file for
% clippeddataviewer__original12.fig
%      CLIPPEDDATAVIEWER__ORIGINAL12, by itself, creates a new
CLIPPEDDATAVIEWER__ORIGINAL12 or raises the existing
%      singleton*.
%
%      H = CLIPPEDDATAVIEWER__ORIGINAL12 returns the handle to a new
CLIPPEDDATAVIEWER__ORIGINAL12 or the handle to
%      the existing singleton*.
%
%      CLIPPEDDATAVIEWER__ORIGINAL12('CALLBACK',hObject,eventData,handles,...) calls
the local
%      function named CALLBACK in CLIPPEDDATAVIEWER__ORIGINAL12.M with the
%      given input arguments.
%
%      CLIPPEDDATAVIEWER__ORIGINAL12('Property','Value',...) creates a new
CLIPPEDDATAVIEWER__ORIGINAL12 or raises the
%      existing singleton*.  Starting from the left, property value pairs are
%      applied to the GUI before clippeddataviewer__original12_OpeningFcn gets called.  An
%      unrecognized property name or invalid value makes property application
%      stop.  All inputs are passed to clippeddataviewer__original12_OpeningFcn via varargin.
%
%      *See GUI Options on GUIDE's Tools menu.  Choose "GUI allows only one
%      instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help clippeddataviewer__original12

% Last Modified by GUIDE v2.5 04-Apr-2010 18:55:25

% Begin initialization code - DO NOT EDIT
```

```
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @clippeddataviewer__original12_OpeningFcn, ...
                   'gui_OutputFcn',  @clippeddataviewer__original12_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT


% --- Executes just before clippeddataviewer__original12 is made visible.
function clippeddataviewer__original12_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to clippeddataviewer__original12 (see VARARGIN)
%A = importdata('bh_ga_21-04-2009_1.exp'); %importing the data file (.exp)%
timesclicked =0;
global g
g=1;
global Flag
Flag=1;
global XY1
XY1=[0, 0];
global XY2
XY2=[0, 0];
global A
global c
global status
% Choose default command line output for clippeddataviewer__original12
handles.output = hObject;

% Update handles structure
set(hObject, 'toolbar', 'figure');
datacursormode on;
guidata(hObject, handles);

% UIWAIT makes clippeddataviewer__original12 wait for user response (see UIRESUME)
% uiwait(handles.figure1);
```

```
% --- Outputs from this function are returned to the command line.
function varargout = clippeddataviewer__original12_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;



% --- If Enable == 'on', executes on mouse press in 5 pixel border.
% --- Otherwise, executes on mouse press in 5 pixel border or over text1.
function text1_ButtonDownFcn(hObject, eventdata, handles)
% hObject    handle to text1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)


% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global A
global c
global status
filename = get(handles.input_filename, 'String');
fid=0;

%seg1 = A;
eventendtime1=get(handles.input_edit2, 'String');
eventstarttime1=get(handles.input_edit1, 'String');
lengthofplot1=get(handles.input_edit6, 'String');
lengthofplot=str2double(lengthofplot1);
timesclicked = 0;
if lengthofplot == 0
     l=e-n;

else
    l=lengthofplot*1000;
end
clear fid;
 compstr = strrep(filename, 'txt', 'mat');
  c = strcmp(filename, compstr)
  if c==1
      A1=load(filename, '-mat');
     A=(A1.savedata);
  else
%     A= load(filename);
    fid=fopen(filename);
```

```
[vg]=textscan(fid, '%f %f %f %f %f %f %f %f %f %f %f', l);
A=cell2mat(vg);
%     A=cell2mat(A1);
  end

%status=fseek(fid, 1, 'bof');
global d
global Flag
global XY1
global XY2
channel = get(handles.channels, 'String');
channels = str2num(channel);
channel1=channels(1)
channel2=channels(2)
channel3=channels(3)
channel4=channels(4)
lengthofplot=0

%x=[0:10];
%y=[0:10];
time1 = A(:, 1);
if channel1 ==1
    dataseg1 = A(:, 1);
elseif channel1 ==2
    dataseg1 = A(:, 2);
elseif channel1 ==3
    dataseg1 = A(:, 3);
elseif channel1 ==4
    dataseg1 = A(:, 4);
elseif channel1 ==5
    dataseg1 = A(:, 5);
elseif channel1 ==6
    dataseg1 = A(:, 6);
elseif channel1 ==7
    dataseg1 = A(:, 7);
elseif channel1 ==8
    dataseg1 = A(:, 8);
elseif channel1 ==9
    dataseg1 = A(:, 9);
elseif channel1 ==10
    dataseg1 = A(:, 10);
elseif channel1 == 11
    dataseg1 = A(:, 11);
elseif channel1 ==12
    dataseg1 = A(:, 12);
elseif channel1 ==13
    dataseg1 = A(:, 13);
elseif channel1 ==14
    dataseg1 = A(:, 14);
elseif channel1 ==15
    dataseg1 = A(:, 15);
elseif channel1 ==16
```

```
    dataseg1 = A(:, 16);
elseif channel1 ==17
    dataseg1 = A(:, 17);
elseif channel1 ==18
    dataseg1 = A(:, 18);
elseif channel1 ==19
    dataseg1 = A(:, 19);
elseif channel1 ==20
    dataseg1 = A(:, 20);
elseif channel1 ==21
    dataseg1 = A(:, 21);
end

 if channel2 ==1
     dataseg2 = A(:, 1);
 elseif channel2 ==2
     dataseg2 = A(:, 2);
 elseif channel2 ==3
     dataseg2 = A(:, 3);
 elseif channel2 ==4
     dataseg2 = A(:, 4);
 elseif channel2 ==5
     dataseg2 = A(:, 5);
 elseif channel2 ==6
     dataseg2 = A(:, 6);
 elseif channel2 ==7
     dataseg2 = A(:, 7);
 elseif channel2 ==8
     dataseg2 = A(:, 8);
 elseif channel2 ==9
     dataseg2 = A(:, 9);
 elseif channel2 ==10
     dataseg2 = A(:, 10);
 elseif channel2 == 11
     dataseg2 = A(:, 11);
 elseif channel2 ==12
     dataseg2 = A(:, 12);
 elseif channel2 ==13
     dataseg2 = A(:, 13);
 elseif channel2 ==14
     dataseg2 = A(:, 14);
 elseif channel2 ==15
     dataseg2 = A(:, 15);
 elseif channel2 ==16
     dataseg2 = A(:, 16);
 elseif channel2 ==17
     dataseg2 = A(:, 17);
 elseif channel2 ==18
     dataseg2 = A(:, 18);
 elseif channel2 ==19
     dataseg2 = A(:, 19);
 elseif channel2 ==20
```

```
    dataseg2 = A(:, 20);
 elseif channel2 ==21
    dataseg2 = A(:, 21);
 end

if channel3 ==1
   dataseg3 = A(:, 1);
elseif channel3 ==2
   dataseg3 = A(:, 2);
elseif channel3 ==3
   dataseg3 = A(:, 3);
elseif channel3 ==4
   dataseg3 = A(:, 4);
elseif channel3 ==5
   dataseg3 = A(:, 5);
elseif channel3 ==6
   dataseg3 = A(:, 6);
elseif channel3 ==7
   dataseg3 = A(:, 7);
elseif channel3 ==8
   dataseg3 = A(:, 8);
elseif channel3 ==9
   dataseg3 = A(:, 9);
elseif channel3 ==10
   dataseg3 = A(:, 10);
elseif channel3 == 11
   dataseg3 = A(:, 11);
elseif channel3 ==12
   dataseg3 = A(:, 12);
elseif channel3 ==13
   dataseg3 = A(:, 13);
elseif channel3 ==14
   dataseg3 = A(:, 14);
elseif channel3 ==15
   dataseg3 = A(:, 15);
elseif channel3 ==16
   dataseg3 = A(:, 16);
elseif channel3 ==17
   dataseg3 = A(:, 17);
elseif channel3 ==18
   dataseg3 = A(:, 18);
elseif channel3 ==19
   dataseg3 = A(:, 19);
elseif channel3 ==20
   dataseg3 = A(:, 20);
elseif channel3 ==21
   dataseg3 = A(:, 21);
end

if channel4 ==1
   dataseg4 = A(:, 1);
elseif channel4 ==2
```

```
    dataseg4 = A(:, 2);
elseif channel4 ==3
    dataseg4 = A(:, 3);
elseif channel4 ==4
    dataseg4 = A(:, 4);
elseif channel4 ==5
    dataseg4 = A(:, 5);
elseif channel4 ==6
    dataseg4 = A(:, 6);
elseif channel4 ==7
    dataseg4 = A(:, 7);
elseif channel4 ==8
    dataseg4 = A(:, 8);
elseif channel4 ==9
    dataseg4 = A(:, 9);
elseif channel4 ==10
    dataseg4 = A(:, 10);
elseif channel4 == 11
    dataseg4 = A(:, 11);
elseif channel4 ==12
    dataseg4 = A(:, 12);
elseif channel4 ==13
    dataseg4 = A(:, 13);
elseif channel4 ==14
    dataseg4 = A(:, 14);
elseif channel4 ==15
    dataseg4 = A(:, 15);
elseif channel4 ==16
    dataseg4 = A(:, 16);
elseif channel4 ==17
    dataseg4 = A(:, 17);
elseif channel4 ==18
    dataseg4 = A(:, 18);
elseif channel4 ==19
    dataseg4 = A(:, 19);
elseif channel4 ==20
    dataseg4 = A(:, 20);
elseif channel4 ==21
    dataseg4 = A(:, 21);
end
eventstarttime=str2double(eventstarttime1);
eventendtime=str2double(eventendtime1);
%dataseg3 = A(:, 3);

%dataseg4 = A(:, 4);
%time1(1,:)=[];  %deleting first few rows containig patient name and details
%time1(1,:)=[];
%time1(1,:)=[];
%time1(1,:)=[];
%time1(1,:)=[];
%time1(1,:)=[];
%[m,n]=size(time1);
```

```
[m,n]=size(A);
%for i=1:1:m+1
 %   dataseg1(i)=num2str(dataseg1d(i));
%end
% for i=1:1:m
%     if eventstarttime== time1(i)
%         TF(i) =1;
%     else
%         TF(i) = 0;
%     end
%     %TF=strcmp(eventstarttime,dataseg1);
%     if eventendtime== time1(i)
%         TF1(i) =1;
%     else
%         TF1(i) = 0;
%     end
% end
%TF1=strcmp(eventendtime,dataseg1);

% for i=1:1:m
%     if TF(i) ==1
%          break;
%     end
%   end
% n=i;
% for i=1:1:m
%     if TF1(i) == 1
%         break;
%     end
% end
% e=i;

% for i=1:1:l
%     time(i)=time1(i+n-1);
%     dataseg11(i)=dataseg1(i+n-1);
% %     dataseg12(i)=dataseg2(i+n-1);
% %    dataseg13(i)=dataseg3(i+n-1);
% %    dataseg14(i)=dataseg4(i+n-1);
% end
%[dataseg13,w]= pwelch(dataseg12,[],[],4096,1000);
%  d=dataseg11;
% t=[0:l-1];
 axes(handles.axes1);
 %time=downsample(time,100000);
 %dataseg11=downsample(dataseg11,100000);
 plot(time1, dataseg1, 'b');
 hold on;
 axes(handles.axes2);
 plot(time1, dataseg2, 'r');
 hold on;
 axes(handles.axes3);
 plot(time1, dataseg3, 'g');
```

```
 hold on;
 axes(handles.axes4);
 plot(time1, dataseg4, 'y');
 hold on;
% l1=l;
guidata(hObject, handles);




function input_edit1_Callback(hObject, eventdata, handles)
% hObject    handle to input_edit1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of input_edit1 as text
%        str2double(get(hObject,'String')) returns contents of input_edit1 as a double
guidata(hObject, handles);


% --- Executes during object creation, after setting all properties.
function input_edit1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to input_edit1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function input_edit2_Callback(hObject, eventdata, handles)
% hObject    handle to input_edit2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
    %eventstarttime= get(hObject, 'String');
% Hints: get(hObject,'String') returns contents of input_edit2 as text
%        str2double(get(hObject,'String')) returns contents of input_edit2 as a double
guidata(hObject, handles);

% --- Executes during object creation, after setting all properties.
function input_edit2_CreateFcn(hObject, eventdata, handles)
% hObject    handle to input_edit2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
```

```
end
function importfile(fileToRead1)
%IMPORTFILE(FILETOREAD1)
%  Imports data from the specified file
%  FILETOREAD1:  file to read

%  Auto-generated by MATLAB on 10-Jun-2009 17:17:37

% Import the file
newData1 = importdata(fileToRead1);

% Create new variables in the base workspace from those fields.
vars = fieldnames(newData1);
for i = 1:length(vars)
   assignin('base', vars{i}, newData1.(vars{i}));
end


% --- Executes on button press in pushbutton2.
function pushbutton2_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
cla(handles.axes1, 'reset');
cla(handles.axes2, 'reset');
cla(handles.axes3, 'reset');
cla(handles.axes4, 'reset');
guidata(hObject, handles);



function edit5_Callback(hObject, eventdata, handles)
% hObject    handle to edit5 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit5 as text
%        str2double(get(hObject,'String')) returns contents of edit5 as a double


% --- Executes during object creation, after setting all properties.
function edit5_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit5 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
   set(hObject,'BackgroundColor','white');
end
```

```
function input_edit6_Callback(hObject, eventdata, handles)
% hObject    handle to input_edit6 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
input=str2double(get(hObject, 'String'));
if (isempty(input))
   set(hObject, 'String', '0')
end
% Hints: get(hObject,'String') returns contents of input_edit6 as text
%        str2double(get(hObject,'String')) returns contents of input_edit6 as a double
guidata(hObject, handles);

% --- Executes during object creation, after setting all properties.
function input_edit6_CreateFcn(hObject, eventdata, handles)
% hObject    handle to input_edit6 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
   set(hObject,'BackgroundColor','white');
end


% --- Executes on button press in pushbutton3.
function [z]=pushbutton3_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton3 (see GCBO)
global d
global A
global c
dataseg=d;
channel = get(handles.channels, 'String');
channels = str2num(channel);
% channel1=channels(1)
% channel2=channels(2)
% channel3=channels(3)
% channel4=channels(4)

saveclip = get(handles.savedata, 'String');
% compstr = strrep(filename, 'txt', 'mat');
% c = strcmp(filename, compstr)
% if c==1
%     A1=load(filename, '-mat');
%      A=(A1.Data);
% else
%     A = load(filename);
%
%     %A=cell2mat(A1)
% end
```

```
%A = load(filename); %importing the data file (.exp)%
%A = A;
% if channel1 ==1
%     dataseg1 = A(:, 1);
% elseif channel1 ==2
%     dataseg1 = A(:, 2);
% elseif channel1 ==3
%     dataseg1 = A(:, 3);
% elseif channel1 ==4
%     dataseg1 = A(:, 4);
% elseif channel1 ==5
%     dataseg1 = A(:, 5);
% elseif channel1 ==6
%     dataseg1 = A(:, 6);
% elseif channel1 ==7
%     dataseg1 = A(:, 7);
% elseif channel1 ==8
%     dataseg1 = A(:, 8);
% elseif channel1 ==9
%     dataseg1 = A(:, 9);
% elseif channel1 ==10
%     dataseg1 = A(:, 10);
% elseif channel1 == 11
%     dataseg1 = A(:, 11);
% elseif channel1 ==12
%     dataseg1 = A(:, 12);
% elseif channel1 ==13
%     dataseg1 = A(:, 13);
% elseif channel1 ==14
%     dataseg1 = A(:, 14);
% elseif channel1 ==15
%     dataseg1 = A(:, 15);
% elseif channel1 ==16
%     dataseg1 = A(:, 16);
% elseif channel1 ==17
%     dataseg1 = A(:, 17);
% elseif channel1 ==18
%     dataseg1 = A(:, 18);
% elseif channel1 ==19
%     dataseg1 = A(:, 19);
% elseif channel1 ==20
%     dataseg1 = A(:, 20);
% elseif channel1 ==21
%     dataseg1 = A(:, 21);
% end
%
% if channel2 ==1
%     dataseg2 = A(:, 1);
% elseif channel2 ==2
%     dataseg2 = A(:, 2);
% elseif channel2 ==3
%     dataseg2 = A(:, 3);
```

```
% elseif channel2 ==4
%     dataseg2 = A(:, 4);
% elseif channel2 ==5
%     dataseg2 = A(:, 5);
% elseif channel2 ==6
%     dataseg2 = A(:, 6);
% elseif channel2 ==7
%     dataseg2 = A(:, 7);
% elseif channel2 ==8
%     dataseg2 = A(:, 8);
% elseif channel2 ==9
%     dataseg2 = A(:, 9);
% elseif channel2 ==10
%     dataseg2 = A(:, 10);
% elseif channel2 == 11
%     dataseg2 = A(:, 11);
% elseif channel2 ==12
%     dataseg2 = A(:, 12);
% elseif channel2 ==13
%     dataseg2 = A(:, 13);
% elseif channel2 ==14
%     dataseg2 = A(:, 14);
% elseif channel2 ==15
%     dataseg2 = A(:, 15);
% elseif channel2 ==16
%     dataseg2 = A(:, 16);
% elseif channel2 ==17
%     dataseg2 = A(:, 17);
% elseif channel2 ==18
%     dataseg2 = A(:, 18);
% elseif channel2 ==19
%     dataseg2 = A(:, 19);
% elseif channel2 ==20
%     dataseg2 = A(:, 20);
% elseif channel2 ==21
%     dataseg2 = A(:, 21);
% end
%
% if channel3 ==1
%      dataseg3 = A(:, 1);
%  elseif channel3 ==2
%      dataseg3 = A(:, 2);
%  elseif channel3 ==3
%      dataseg3 = A(:, 3);
%  elseif channel3 ==4
%      dataseg3 = A(:, 4);
%  elseif channel3 ==5
%      dataseg3 = A(:, 5);
%  elseif channel3 ==6
%      dataseg3 = A(:, 6);
%  elseif channel3 ==7
%      dataseg3 = A(:, 7);
```

```
%  elseif channel3 ==8
%      dataseg3 = A(:, 8);
%  elseif channel3 ==9
%     dataseg3 = A(:, 9);
%  elseif channel3 ==10
%      dataseg3 = A(:, 10);
%  elseif channel3 == 11
%      dataseg3 = A(:, 11);
%  elseif channel3 ==12
%      dataseg3 = A(:, 12);
%  elseif channel3 ==13
%      dataseg3 = A(:, 13);
%  elseif channel3 ==14
%      dataseg3 = A(:, 14);
%  elseif channel3 ==15
%      dataseg3 = A(:, 15);
%  elseif channel3 ==16
%      dataseg3 = A(:, 16);
%  elseif channel3 ==17
%      dataseg3 = A(:, 17);
%  elseif channel3 ==18
%      dataseg3 = A(:, 18);
%  elseif channel3 ==19
%      dataseg3 = A(:, 19);
% elseif channel3 ==20
%      dataseg3 = A(:, 20);
%  elseif channel3 ==21
%      dataseg3 = A(:, 21);
%  end
% %
%  if channel4 ==1
%      dataseg4 = A(:, 1);
%  elseif channel4 ==2
%      dataseg4 = A(:, 2);
%  elseif channel4 ==3
%     dataseg4 = A(:, 3);
% elseif channel4 ==4
%     dataseg4 = A(:, 4);
% elseif channel4 ==5
%     dataseg4 = A(:, 5);
% elseif channel4 ==6
%     dataseg4 = A(:, 6);
% elseif channel4 ==7
%     dataseg4 = A(:, 7);
% elseif channel4 ==8
%     dataseg4 = A(:, 8);
% elseif channel4 ==9
%     dataseg4 = A(:, 9);
% elseif channel4 ==10
%     dataseg4 = A(:, 10);
% elseif channel4 == 11
%     dataseg4 = A(:, 11);
```

```
% elseif channel4 ==12
%     dataseg4 = A(:, 12);
% elseif channel4 ==13
%     dataseg4 = A(:, 13);
% elseif channel4 ==14
%     dataseg4 = A(:, 14);
% elseif channel4 ==15
%     dataseg4 = A(:, 15);
% elseif channel4 ==16
%     dataseg4 = A(:, 16);
% elseif channel4 ==17
%     dataseg4 = A(:, 17);
% elseif channel4 ==18
%     dataseg4 = A(:, 18);
% elseif channel4 ==19
%     dataseg4 = A(:, 19);
% elseif channel4 ==20
%     dataseg4 = A(:, 20);
% elseif channel4 ==21
%     dataseg4 = A(:, 21);
% end
% %dataseg1 = A(:, 1);
% %dataseg2 = A(:, 2);
% %dataseg3 = A(:, 3);
% %dataseg2 = A(:, 2);
% time1 = A(:,1);
% dataseg1 = A(:,1);
% dataseg2 = A(:,2);
% dataseg3 = A(:,3);
% dataseg4 = A(:,4);
% dataseg5 = A(:,5);
% dataseg6 = A(:,6);
% dataseg7 = A(:,7);
% dataseg8 = A(:,8);
% dataseg9 = A(:,9);
% dataseg10 = A(:,10);
% dataseg11 = A(:,11);
% dataseg12 = A(:,12);
% dataseg13 = A(:,13);
% dataseg14 = A(:,14);
% dataseg15 = A(:,15);
% dataseg16 = A(:,16);
% dataseg17 = A(:,17);
% dataseg18 = A(:,18);
% dataseg19 = A(:,19);
% dataseg20 = A(:,20);
% dataseg21 = A(:,21);
[m,n]=size(A);

global XY1 XY2 Flag
x1=XY1(1);
x2=XY2(1);
```

```
y1=XY1(2);
y2=XY2(2);
if x1>x2
   t1=x1-x2;
else
   t1=x2-x1;
end
l=t1*0.0001;

l1=t1*1000;
filename = get(handles.input_filename, 'String');
fid=fopen(filename);
x1=single(x1);
lg=x1*1000;
lg1=floor(lg);
[vg]=textscan(fid, '%f %f %f %f %f %f %f %f %f %f %f', l1, 'HeaderLines', lg1);
A1=cell2mat(vg);
time1=A1(:,1);
% if x1>x2
%
%
%    for i=1:1:m
%       if x1== time1(i)
%          TF(i) =1;
%       else
%          TF(i) = 0;
%       end
%    %TF=strcmp(eventstarttime,dataseg1);
%       if x2== time1(i)
%          TF1(i) =1;
%       else
%          TF1(i) = 0;
%       end
%    end
% else
%
%       for i=1:1:m
%          if x2== time1(i)
%             TF(i) =1;
%          else
%             TF(i) = 0;
%          end
%    %TF=strcmp(eventstarttime,dataseg1);
%          if x1== time1(i)
%             TF1(i) =1;
%          else
%             TF1(i) = 0;
%          end
%       end
% end
% %TF1=strcmp(eventendtime,dataseg1);
%
```

```
% for i=1:1:m
%     if TF(i) ==1
%         break;
%     end
% end
% n=i;
% for i=1:1:m
%     if TF1(i) == 1
%         break;
%     end
% end
% e=i;
% ne=e-n;
% if lengthofplot == 0
%      l=e-n;
%
% else


% for i=1:1:ne
%     time(i)= time1(i+n-1);
%    datasegs1(i)=dataseg1(i+n-1);
%    datasegs2(i)=dataseg2(i+n-1);
%    datasegs3(i)=dataseg3(i+n-1);
%    datasegs4(i)=dataseg4(i+n-1);
%    datasegs5(i)=dataseg5(i+n-1);
%    datasegs6(i)=dataseg6(i+n-1);
%    datasegs7(i)=dataseg7(i+n-1);
%    datasegs8(i)=dataseg8(i+n-1);
%    datasegs9(i)=dataseg9(i+n-1);
%    datasegs10(i)=dataseg10(i+n-1);
%    datasegs11(i)=dataseg11(i+n-1);
%    datasegs12(i)=dataseg12(i+n-1);
%    datasegs13(i)=dataseg13(i+n-1);
%    datasegs14(i)=dataseg14(i+n-1);
%    datasegs15(i)=dataseg15(i+n-1);
%    datasegs16(i)=dataseg16(i+n-1);
%    datasegs17(i)=dataseg17(i+n-1);
%    dataseg118(i)=dataseg18(i+n-1);
%    dataseg119(i)=dataseg19(i+n-1)
%    dataseg120(i)=dataseg20(i+n-1)
%    dataseg121(i)=dataseg21(i+n-1);

% end
% savedata=vertcat(datasegs1, datasegs2, datasegs3, datasegs4, datasegs5);%, datasegs6,
datasegs7, datasegs8, datasegs9, datasegs10, datasegs11, ...
   %datasegs12, datasegs13, datasegs14, datasegs15, datasegs16, datasegs7);
savedata=A1;
% saveclip = get(handles.savedata, 'String');
% savedata1 = savedata';
save((get(handles.savedata, 'String')), 'savedata');
% , 'dataseg17', 'dataseg18', 'dataseg19', 'dataseg110');
```

```
%xlswrite(saveclip, dataseg11);
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
guidata(hObject, handles);


function input_filename_Callback(hObject, eventdata, handles)
% hObject    handle to input_filename (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% global filename
% global c
% filename = get(handles.input_filename, 'String');
% compstr = strrep(filename, 'txt', 'mat');
% c = strcmp(filename, compstr)
% if c==1
%     A1=load(filename, '-mat');
%       A=(A1.Data);
% else
%     %A = load(filename);
%
%     %A=cell2mat(A1)
% end
% dataseg1 = A(:,1);
% dataseg2 = A(:,2);
% plot(dataseg1,dataseg2);
% Hints: get(hObject,'String') returns contents of input_filename as text
%        str2double(get(hObject,'String')) returns contents of input_filename as a double
guidata(hObject, handles);

% --- Executes during object creation, after setting all properties.
function input_filename_CreateFcn(hObject, eventdata, handles)
% hObject    handle to input_filename (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
   set(hObject,'BackgroundColor','white');
end


% --- Executes on button press in nextclip.
function nextclip_Callback(hObject, eventdata, handles)
% hObject    handle to nextclip (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
channel = get(handles.channels, 'String');
channels = str2num(channel);
channel1=channels(1)
channel2=channels(2)
```

```
channel3=channels(3)
channel4=channels(4)
filename = get(handles.input_filename, 'String');
% compstr = strrep(filename, 'txt', 'mat');
% c = strcmp(filename, compstr)
% if c==1
%     A=load(filename, '-mat');
%      A=(A1.Data);
% else
%     A = load(filename);
%
%     %A=cell2mat(A1);
% end
global A
global g
global d
global Flag
global XY1
global XY2
%A = load(filename); %importing the data file (.exp)%
%A = A;   %the data is imported into 1 variable A with 2 sub variables data and textdata which
are seperated in this step%
%time1= A.textdata;
eventstarttime1=get(handles.input_edit1, 'String');
eventendtime1=get(handles.input_edit2, 'String');
lengthofplot1=get(handles.input_edit6, 'String');
lengthofplot=str2double(lengthofplot1);
if lengthofplot == 0
     l=e-n;

else
    l=lengthofplot*1000;
end

l1=(l*(g+1));
l2=l*(g);
lg=l1-l2;
 compstr = strrep(filename, 'txt', 'mat');
 c = strcmp(filename, compstr)
 if c==1
      A=load(filename, '-mat');
      A1=(A.savedata);
 else
%     A = load(filename);
     fid=fopen(filename);
    [vg]=textscan(fid, '%f %f %f %f %f %f %f %f %f %f %f',lg, 'HeaderLines', l2);
    A1=cell2mat(vg);
%     %A=cell2mat(A1);
 end

%position=ftell(fid)
time1=A1(:, 1);
```

```
if channel1 ==1
   dataseg1 = A1(:, 1);
elseif channel1 ==2
   dataseg1 = A1(:, 2);
elseif channel1 ==3
   dataseg1 = A1(:, 3);
elseif channel1 ==4
   dataseg1 = A1(:, 4);
elseif channel1 ==5
   dataseg1 = A1(:, 5);
elseif channel1 ==6
   dataseg1 = A1(:, 6);
elseif channel1 ==7
   dataseg1 = A1(:, 7);
elseif channel1 ==8
   dataseg1 = A1(:, 8);
elseif channel1 ==9
   dataseg1 = A1(:, 9);
elseif channel1 ==10
   dataseg1 = A1(:, 10);
elseif channel1 == 11
   dataseg1 = A1(:, 11);
elseif channel1 ==12
   dataseg1 = A1(:, 12);
elseif channel1 ==13
   dataseg1 = A1(:, 13);
elseif channel1 ==14
   dataseg1 = A1(:, 14);
elseif channel1 ==15
   dataseg1 = A1(:, 15);
elseif channel1 ==16
   dataseg1 = A1(:, 16);
elseif channel1 ==17
   dataseg1 = A1(:, 17);
elseif channel1 ==18
   dataseg1 = A1(:, 18);
elseif channel1 ==19
   dataseg1 = A1(:, 19);
elseif channel1 ==20
   dataseg1 = A1(:, 20);
elseif channel1 ==21
   dataseg1 = A1(:, 21);
end

if channel2 ==1
   dataseg2 = A1(:, 1);
elseif channel2 ==2
   dataseg2 = A1(:, 2);
elseif channel2 ==3
   dataseg2 = A1(:, 3);
elseif channel2 ==4
   dataseg2 = A1(:, 4);
```

```
elseif channel2 ==5
   dataseg2 = A1(:, 5);
elseif channel2 ==6
   dataseg2 = A1(:, 6);
elseif channel2 ==7
   dataseg2 = A1(:, 7);
elseif channel2 ==8
   dataseg2 = A1(:, 8);
elseif channel2 ==9
   dataseg2 = A1(:, 9);
elseif channel2 ==10
   dataseg2 = A1(:, 10);
elseif channel2 == 11
   dataseg2 = A1(:, 11);
elseif channel2 ==12
   dataseg2 = A1(:, 12);
elseif channel2 ==13
   dataseg2 = A1(:, 13);
elseif channel2 ==14
   dataseg2 = A1(:, 14);
elseif channel2 ==15
   dataseg2 = A1(:, 15);
elseif channel2 ==16
   dataseg2 = A1(:, 16);
elseif channel2 ==17
   dataseg2 = A1(:, 17);
elseif channel2 ==18
   dataseg2 = A1(:, 18);
elseif channel2 ==19
   dataseg2 = A1(:, 19);
elseif channel2 ==20
   dataseg2 = A1(:, 20);
elseif channel2 ==21
   dataseg2 = A1(:, 21);
end

if channel3 ==1
   dataseg3 = A1(:, 1);
elseif channel3 ==2
   dataseg3 = A1(:, 2);
elseif channel3 ==3
   dataseg3 = A1(:, 3);
elseif channel3 ==4
   dataseg3 = A1(:, 4);
elseif channel3 ==5
   dataseg3 = A1(:, 5);
elseif channel3 ==6
   dataseg3 = A1(:, 6);
elseif channel3 ==7
   dataseg3 = A1(:, 7);
elseif channel3 ==8
   dataseg3 = A1(:, 8);
```

```
elseif channel3 ==9
    dataseg3 = A1(:, 9);
elseif channel3 ==10
    dataseg3 = A1(:, 10);
elseif channel3 == 11
    dataseg3 = A1(:, 11);
elseif channel3 ==12
    dataseg3 = A1(:, 12);
elseif channel3 ==13
    dataseg3 = A1(:, 13);
elseif channel3 ==14
    dataseg3 = A1(:, 14);
elseif channel3 ==15
    dataseg3 = A1(:, 15);
elseif channel3 ==16
    dataseg3 = A1(:, 16);
elseif channel3 ==17
    dataseg3 = A1(:, 17);
elseif channel3 ==18
    dataseg3 = A1(:, 18);
elseif channel3 ==19
    dataseg3 = A1(:, 19);
elseif channel3 ==20
    dataseg3 = A1(:, 20);
elseif channel3 ==21
    dataseg3 = A1(:, 21);
end

if channel4 ==1
    dataseg4 = A1(:, 1);
elseif channel4 ==2
    dataseg4 = A1(:, 2);
elseif channel4 ==3
    dataseg4 = A1(:, 3);
elseif channel4 ==4
    dataseg4 = A1(:, 4);
elseif channel4 ==5
    dataseg4 = A1(:, 5);
elseif channel4 ==6
    dataseg4 = A1(:, 6);
elseif channel4 ==7
    dataseg4 = A1(:, 7);
elseif channel4 ==8
    dataseg4 = A1(:, 8);
elseif channel4 ==9
    dataseg4 = A1(:, 9);
elseif channel4 ==10
    dataseg4 = A1(:, 10);
elseif channel4 == 11
    dataseg4 = A1(:, 11);
elseif channel4 ==12
    dataseg4 = A1(:, 12);
```

```
elseif channel4 ==13
   dataseg4 = A1(:, 13);
elseif channel4 ==14
   dataseg4 = A1(:, 14);
elseif channel4 ==15
   dataseg4 = A1(:, 15);
elseif channel4 ==16
   dataseg4 = A1(:, 16);
elseif channel4 ==17
   dataseg4 = A1(:, 17);
elseif channel4 ==18
   dataseg4 = A1(:, 18);
elseif channel4 ==19
   dataseg4 = A1(:, 19);
elseif channel4 ==20
   dataseg4 = A1(:, 20);
elseif channel4 ==21
   dataseg4 = A1(:, 21);
end
%dataseg1 = A(:, 1);
%dataseg2 = A(:, 2);
eventstarttime=str2double(eventstarttime1);
eventendtime=str2double(eventendtime1);
%dataseg1= num2str(dataseg1d);%the column of data containing blood flow velocity in cm/s is
the 2nd column. Seperated from the rest of data
%dataseg3 = A(:, 3);
%dataseg4 = A(:, 4);
%time1(1,:)=[];  %deleting first few rows containig patient name and details
%time1(1,:)=[];
%time1(1,:)=[];
%time1(1,:)=[];
%time1(1,:)=[];
%time1(1,:)=[];
[m,n]=size(dataseg1);
% for i=1:1:m
%     if eventstarttime== time1(i)
%         TF(i) =1;
%     else
%         TF(i) = 0;
%     end
% %TF=strcmp(eventstarttime,dataseg1);
%     if eventendtime== time1(i)
%         TF1(i) =1;
%     else
%         TF1(i) = 0;
%     end
% end
% %TF=strcmp(eventstarttime1,dataseg1);
% %TF1=strcmp(eventendtime1,dataseg1);
% for i=1:1:m
%     if TF(i) ==1
%         break;
```

```
%     end
%   end
% n=i;
% for i=1:1:m
%     if TF1(i) == 1
%         break;
%     end
% end
% e=i;
% for i=1:1:l
%     time(i)=time1(i+l1-l2+n+l1-1);
%     dataseg11n(i)=dataseg1(i+l1-l2+n+l1-1);
%     dataseg12n(i)=dataseg2(i+l1-l2+n+l1-1);
%     dataseg13n(i)=dataseg3(i+l1-l2+n+l1-1);
%     dataseg14n(i)=dataseg4(i+l1-l2+n+l1-1);
% end
g=g+1;

%[dataseg13n,w]= pwelch(dataseg12,1024,[],4096,1000);
% d=dataseg11n;
% t=[1:1:l];
cla(handles.axes1, 'reset');
axes(handles.axes1);
plot(time1, dataseg1, 'b');
hold on;
cla(handles.axes2, 'reset');
axes(handles.axes2);
plot(time1, dataseg2, 'r');
hold on;
cla(handles.axes3, 'reset');
axes(handles.axes3);
plot(time1, dataseg3, 'g');
hold on;
cla(handles.axes4, 'reset');
axes(handles.axes4);
plot(time1, dataseg4, 'y');
hold on;

guidata(hObject, handles);


% --- Executes on button press in previousclip1.
function previousclip1_Callback(hObject, eventdata, handles)
% hObject    handle to previousclip1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
channel = get(handles.channels, 'String');
channels = str2num(channel);
channel1=channels(1)
channel2=channels(2)
channel3=channels(3)
channel4=channels(4)
 filename = get(handles.input_filename, 'String');
```

```
% compstr = strrep(filename, 'txt', 'mat');
% c = strcmp(filename, compstr)
% if c==1
%     A1=load(filename, '-mat');
%     A=(A1.Data);
% else
%     A = load(filename);
%
%     %A=cell2mat(A1);
% end
global A
global g
global d
global Flag
global XY1
global XY2
%A = load(filename); %importing the data file (.exp)%
%A = A;   %the data is imported into 1 variable A with 2 sub variables data and textdata which
are seperated in this step%
%time1= A.textdata;
eventstarttime1=get(handles.input_edit1, 'String');
eventendtime1=get(handles.input_edit2, 'String');
lengthofplot1=get(handles.input_edit6, 'String');
lengthofplot=str2double(lengthofplot1);
if lengthofplot == 0
     l=e-n;

else
    l=lengthofplot*1000;
end
l1=(l*(g+1));
l2=l*(g);
lg=l1-l;
 compstr = strrep(filename, 'txt', 'mat');
 c = strcmp(filename, compstr)
 if c==1
     A1=load(filename, '-mat');
     A=(A1.savedata);
 else
%     A = load(filename);
    fid=fopen(filename);
    [vg]=textscan(fid, '%f %f %f %f %f %f %f %f %f %f %f %f', l, 'HeaderLines', lg);
    A1=cell2mat(vg);

%     %A=cell2mat(A1);
 end
time1=A1(:, 1);
if channel1 ==1
    dataseg1 = A1(:, 1);
elseif channel1 ==2
    dataseg1 = A1(:, 2);
elseif channel1 ==3
```

```
    dataseg1 = A1(:, 3);
elseif channel1 ==4
    dataseg1 = A1(:, 4);
elseif channel1 ==5
    dataseg1 = A1(:, 5);
elseif channel1 ==6
    dataseg1 = A1(:, 6);
elseif channel1 ==7
    dataseg1 = A1(:, 7);
elseif channel1 ==8
    dataseg1 = A1(:, 8);
elseif channel1 ==9
    dataseg1 = A1(:, 9);
elseif channel1 ==10
    dataseg1 = A1(:, 10);
elseif channel1 == 11
    dataseg1 = A1(:, 11);
elseif channel1 ==12
    dataseg1 = A1(:, 12);
elseif channel1 ==13
    dataseg1 = A1(:, 13);
elseif channel1 ==14
    dataseg1 = A1(:, 14);
elseif channel1 ==15
    dataseg1 = A1(:, 15);
elseif channel1 ==16
    dataseg1 = A1(:, 16);
elseif channel1 ==17
    dataseg1 = A1(:, 17);
elseif channel1 ==18
    dataseg1 = A1(:, 18);
elseif channel1 ==19
    dataseg1 = A1(:, 19);
elseif channel1 ==20
    dataseg1 = A1(:, 20);
elseif channel1 ==21
    dataseg1 = A1(:, 21);
end

if channel2 ==1
    dataseg2 = A1(:, 1);
elseif channel2 ==2
    dataseg2 = A1(:, 2);
elseif channel2 ==3
    dataseg2 = A1(:, 3);
elseif channel2 ==4
    dataseg2 = A1(:, 4);
elseif channel2 ==5
    dataseg2 = A1(:, 5);
elseif channel2 ==6
    dataseg2 = A1(:, 6);
elseif channel2 ==7
```

```
    dataseg2 = A1(:, 7);
elseif channel2 ==8
    dataseg2 = A1(:, 8);
elseif channel2 ==9
    dataseg2 = A1(:, 9);
elseif channel2 ==10
    dataseg2 = A1(:, 10);
elseif channel2 == 11
    dataseg2 = A1(:, 11);
elseif channel2 ==12
    dataseg2 = A1(:, 12);
elseif channel2 ==13
    dataseg2 = A1(:, 13);
elseif channel2 ==14
    dataseg2 = A1(:, 14);
elseif channel2 ==15
    dataseg2 = A1(:, 15);
elseif channel2 ==16
    dataseg2 = A1(:, 16);
elseif channel2 ==17
    dataseg2 = A1(:, 17);
elseif channel2 ==18
    dataseg2 = A1(:, 18);
elseif channel2 ==19
    dataseg2 = A1(:, 19);
elseif channel2 ==20
    dataseg2 = A1(:, 20);
elseif channel2 ==21
    dataseg2 = A1(:, 21);
end

if channel3 ==1
    dataseg3 = A1(:, 1);
elseif channel3 ==2
    dataseg3 = A1(:, 2);
elseif channel3 ==3
    dataseg3 = A1(:, 3);
elseif channel3 ==4
    dataseg3 = A1(:, 4);
elseif channel3 ==5
    dataseg3 = A1(:, 5);
elseif channel3 ==6
    dataseg3 = A1(:, 6);
elseif channel3 ==7
    dataseg3 = A1(:, 7);
elseif channel3 ==8
    dataseg3 = A1(:, 8);
elseif channel3 ==9
    dataseg3 = A1(:, 9);
elseif channel3 ==10
    dataseg3 = A1(:, 10);
elseif channel3 == 11
```

```
    dataseg3 = A1(:, 11);
elseif channel3 ==12
    dataseg3 = A1(:, 12);
elseif channel3 ==13
    dataseg3 = A1(:, 13);
elseif channel3 ==14
    dataseg3 = A1(:, 14);
elseif channel3 ==15
    dataseg3 = A1(:, 15);
elseif channel3 ==16
    dataseg3 = A1(:, 16);
elseif channel3 ==17
    dataseg3 = A1(:, 17);
elseif channel3 ==18
    dataseg3 = A1(:, 18);
elseif channel3 ==19
    dataseg3 = A1(:, 19);
elseif channel3 ==20
    dataseg3 = A1(:, 20);
elseif channel3 ==21
    dataseg3 = A1(:, 21);
end

if channel4 ==1
    dataseg4 = A1(:, 1);
elseif channel4 ==2
    dataseg4 = A1(:, 2);
elseif channel4 ==3
    dataseg4 = A1(:, 3);
elseif channel4 ==4
    dataseg4 = A1(:, 4);
elseif channel4 ==5
    dataseg4 = A1(:, 5);
elseif channel4 ==6
    dataseg4 = A1(:, 6);
elseif channel4 ==7
    dataseg4 = A1(:, 7);
elseif channel4 ==8
    dataseg4 = A1(:, 8);
elseif channel4 ==9
    dataseg4 = A1(:, 9);
elseif channel4 ==10
    dataseg4 = A1(:, 10);
elseif channel4 == 11
    dataseg4 = A1(:, 11);
elseif channel4 ==12
    dataseg4 = A1(:, 12);
elseif channel4 ==13
    dataseg4 = A1(:, 13);
elseif channel4 ==14
    dataseg4 = A1(:, 14);
elseif channel4 ==15
```

```
   dataseg4 = A1(:, 15);
elseif channel4 ==16
   dataseg4 = A1(:, 16);
elseif channel4 ==17
   dataseg4 = A1(:, 17);
elseif channel4 ==18
   dataseg4 = A1(:, 18);
elseif channel4 ==19
   dataseg4 = A1(:, 19);
elseif channel4 ==20
   dataseg4 = A1(:, 20);
elseif channel4 ==21
   dataseg4 = A1(:, 21);
end
%dataseg1 = A(:, 1);
%dataseg2 = A(:, 2);%the column of data containing blood flow velocity in cm/s is the 2nd
column. Seperated from the rest of data%
eventstarttime=str2double(eventstarttime1);
eventendtime=str2double(eventendtime1);
%dataseg1 = num2str(dataseg1d);
%dataseg3 = A(:, 3);
%dataseg4 = A(:, 4);
%time1(1,:)=[];  %deleting first few rows containig patient name and details
%time1(1,:)=[];
%time1(1,:)=[];
%time1(1,:)=[];
%time1(1,:)=[];
%time1(1,:)=[];
[m,n]=size(dataseg1);
% for i=1:1:m
%     if eventstarttime== time1(i)
%         TF(i) =1;
%     else
%         TF(i) = 0;
%     end
% %TF=strcmp(eventstarttime,dataseg1);
%     if eventendtime== time1(i)
%         TF1(i) =1;
%     else
%         TF1(i) = 0;
%     end
% end
% %TF=strcmp(eventstarttime1,dataseg1);
% %TF1=strcmp(eventendtime1,dataseg1);
% for i=1:1:m
%     if TF(i) ==1
%          break;
%     end
%   end
% n=i;
% for i=1:1:m
%     if TF1(i) == 1
```

```
%         break;
%     end
% end
% e=i;
% for i=1:1:l
%     time(i)=time1(i+l1-l2+n+l1-1);
%     dataseg11n(i)=dataseg1(i+l1-l2+n+l1-1);
%     dataseg12n(i)=dataseg2(i+l1-l2+n+l1-1);
%     dataseg13n(i)=dataseg3(i+l1-l2+n+l1-1);
%     dataseg14n(i)=dataseg4(i+l1-l2+n+l1-1);
% end
g=g+1;

%[dataseg13n,w]= pwelch(dataseg12,1024,[],4096,1000);
% t=[1:1:l];
 %d=dataseg11n;
cla(handles.axes1, 'reset');
axes(handles.axes1);
plot(time1, dataseg1, 'b');
hold on;
cla(handles.axes2, 'reset');
axes(handles.axes2);
plot(time1, dataseg2, 'r');
hold on;
cla(handles.axes3, 'reset');
axes(handles.axes3);
plot(time1, dataseg3, 'g');
hold on;
cla(handles.axes4, 'reset');
axes(handles.axes4);
plot(time1, dataseg4, 'y');
hold on;

guidata(hObject, handles);


% --- Executes on button press in previousclip1.
% function previousclip1_Callback(hObject, eventdata, handles)
% % hObject    handle to previousclip1 (see GCBO)
% % eventdata  reserved - to be defined in a future version of MATLAB
% channel = get(handles.channels, 'String');
% channels = str2num(channel);
% channel1=channels(1)
% channel2=channels(2)
% channel3=channels(3)
% channel4=channels(4)
%  filename = get(handles.input_filename, 'String');
% % compstr = strrep(filename, 'txt', 'mat');
% % c = strcmp(filename, compstr)
% % if c==1
% %     A1=load(filename, '-mat');
% %     A=(A1.Data);
```

```
% % else
% %     A = load(filename);
% %
% %     %A=cell2mat(A1);
% % end
% global A
% global g
% global d
% global Flag
% global XY1
% global XY2
% %A = load(filename); %importing the data file (.exp)%
% %A = A;   %the data is imported into 1 variable A with 2 sub variables data and textdata
which are seperated in this step%
% %time1= A.textdata;
%
% eventstarttime1=get(handles.input_edit1, 'String');
% eventendtime1=get(handles.input_edit2, 'String');
% lengthofplot1=get(handles.input_edit6, 'String');
% lengthofplot=str2double(lengthofplot1);
% if lengthofplot == 0
%      l=e-n;
%
% else
%     l=lengthofplot*2000;
% end
% l1=(l*(g+1));
% l2=l*(g);
% fid=fopen(filename);
% [e]=textscan(fid, '%f %f %f %f %f', l1-l2+n+l1);
% A=cell2mat(e);
% time1=A(:, 1);
% if channel1 ==1
%     dataseg1 = A(:, 1);
% elseif channel1 ==2
%     dataseg1 = A(:, 2);
% elseif channel1 ==3
%     dataseg1 = A(:, 3);
% elseif channel1 ==4
%     dataseg1 = A(:, 4);
% elseif channel1 ==5
%     dataseg1 = A(:, 5);
% elseif channel1 ==6
%     dataseg1 = A(:, 6);
% elseif channel1 ==7
%     dataseg1 = A(:, 7);
% elseif channel1 ==8
%     dataseg1 = A(:, 8);
% elseif channel1 ==9
%     dataseg1 = A(:, 9);
% elseif channel1 ==10
%     dataseg1 = A(:, 10);
```

```
% elseif channel1 == 11
%     dataseg1 = A(:, 11);
% elseif channel1 ==12
%     dataseg1 = A(:, 12);
% elseif channel1 ==13
%     dataseg1 = A(:, 13);
% elseif channel1 ==14
%     dataseg1 = A(:, 14);
% elseif channel1 ==15
%     dataseg1 = A(:, 15);
% elseif channel1 ==16
%     dataseg1 = A(:, 16);
% elseif channel1 ==17
%     dataseg1 = A(:, 17);
% elseif channel1 ==18
%     dataseg1 = A(:, 18);
% elseif channel1 ==19
%     dataseg1 = A(:, 19);
% elseif channel1 ==20
%     dataseg1 = A(:, 20);
% elseif channel1 ==21
%     dataseg1 = A(:, 21);
% end
%
% if channel2 ==1
%     dataseg2 = A(:, 1);
% elseif channel2 ==2
%     dataseg2 = A(:, 2);
% elseif channel2 ==3
%     dataseg2 = A(:, 3);
% elseif channel2 ==4
%     dataseg2 = A(:, 4);
% elseif channel2 ==5
%     dataseg2 = A(:, 5);
% elseif channel2 ==6
%     dataseg2 = A(:, 6);
% elseif channel2 ==7
%     dataseg2 = A(:, 7);
% elseif channel2 ==8
%     dataseg2 = A(:, 8);
% elseif channel2 ==9
%     dataseg2 = A(:, 9);
% elseif channel2 ==10
%     dataseg2 = A(:, 10);
% elseif channel2 == 11
%     dataseg2 = A(:, 11);
% elseif channel2 ==12
%     dataseg2 = A(:, 12);
% elseif channel2 ==13
%     dataseg2 = A(:, 13);
% elseif channel2 ==14
%     dataseg2 = A(:, 14);
```

```
% elseif channel2 ==15
%     dataseg2 = A(:, 15);
% elseif channel2 ==16
%     dataseg2 = A(:, 16);
% elseif channel2 ==17
%     dataseg2 = A(:, 17);
% elseif channel2 ==18
%     dataseg2 = A(:, 18);
% elseif channel2 ==19
%     dataseg2 = A(:, 19);
% elseif channel2 ==20
%     dataseg2 = A(:, 20);
% elseif channel2 ==21
%     dataseg2 = A(:, 21);
% end
%
% if channel3 ==1
%     dataseg3 = A(:, 1);
% elseif channel3 ==2
%     dataseg3 = A(:, 2);
% elseif channel3 ==3
%     dataseg3 = A(:, 3);
% elseif channel3 ==4
%     dataseg3 = A(:, 4);
% elseif channel3 ==5
%     dataseg3 = A(:, 5);
% elseif channel3 ==6
%     dataseg3 = A(:, 6);
% elseif channel3 ==7
%     dataseg3 = A(:, 7);
% elseif channel3 ==8
%     dataseg3 = A(:, 8);
% elseif channel3 ==9
%     dataseg3 = A(:, 9);
% elseif channel3 ==10
%     dataseg3 = A(:, 10);
% elseif channel3 == 11
%     dataseg3 = A(:, 11);
% elseif channel3 ==12
%     dataseg3 = A(:, 12);
% elseif channel3 ==13
%     dataseg3 = A(:, 13);
% elseif channel3 ==14
%     dataseg3 = A(:, 14);
% elseif channel3 ==15
%     dataseg3 = A(:, 15);
% elseif channel3 ==16
%     dataseg3 = A(:, 16);
% elseif channel3 ==17
%     dataseg3 = A(:, 17);
% elseif channel3 ==18
%     dataseg3 = A(:, 18);
```

```
% elseif channel3 ==19
%     dataseg3 = A(:, 19);
% elseif channel3 ==20
%     dataseg3 = A(:, 20);
% elseif channel3 ==21
%     dataseg3 = A(:, 21);
% end
%
% if channel4 ==1
%     dataseg4 = A(:, 1);
% elseif channel4 ==2
%     dataseg4 = A(:, 2);
% elseif channel4 ==3
%     dataseg4 = A(:, 3);
% elseif channel4 ==4
%     dataseg4 = A(:, 4);
% elseif channel4 ==5
%     dataseg4 = A(:, 5);
% elseif channel4 ==6
%     dataseg4 = A(:, 6);
% elseif channel4 ==7
%     dataseg4 = A(:, 7);
% elseif channel4 ==8
%     dataseg4 = A(:, 8);
% elseif channel4 ==9
%     dataseg4 = A(:, 9);
% elseif channel4 ==10
%     dataseg4 = A(:, 10);
% elseif channel4 == 11
%     dataseg4 = A(:, 11);
% elseif channel4 ==12
%     dataseg4 = A(:, 12);
% elseif channel4 ==13
%     dataseg4 = A(:, 13);
% elseif channel4 ==14
%     dataseg4 = A(:, 14);
% elseif channel4 ==15
%     dataseg4 = A(:, 15);
% elseif channel4 ==16
%     dataseg4 = A(:, 16);
% elseif channel4 ==17
%     dataseg4 = A(:, 17);
% elseif channel4 ==18
%     dataseg4 = A(:, 18);
% elseif channel4 ==19
%     dataseg4 = A(:, 19);
% elseif channel4 ==20
%     dataseg4 = A(:, 20);
% elseif channel4 ==21
%     dataseg4 = A(:, 21);
% end
% %dataseg1 = A(:, 1);
```

```
% %dataseg2 = A(:, 2);%the column of data containing blood flow velocity in cm/s is the 2nd
column. Seperated from the rest of data
% %dataseg1=num2str(dataseg1d);
% eventstarttime=str2double(eventstarttime1);
% eventendtime=str2double(eventendtime1);
% %dataseg3 = A(:, 3);
% %dataseg4 = A(:, 4);
% %time1(1,:)=[];  %deleting first few rows containig patient name and details
% %time1(1,:)=[];
% %time1(1,:)=[];
% %time1(1,:)=[];
% %time1(1,:)=[];
% %time1(1,:)=[];
% [m,n]=size(dataseg1);
% % for i=1:1:m
% %     if eventstarttime== time1(i)
% %         TF(i) =1;
% %     else
% %         TF(i) = 0;
% %     end
% % %TF=strcmp(eventstarttime,dataseg1);
% %     if eventendtime== time1(i)
% %         TF1(i) =1;
% %     else
% %         TF1(i) = 0;
% %     end
% % end
% % %TF=strcmp(eventstarttime1,dataseg1);
% % %TF1=strcmp(eventendtime1,dataseg1);
% % for i=1:1:m
% %     if TF(i) ==1
% %          break;
% %     end
% %   end
% % n=i;
% % for i=1:1:m
% %     if TF1(i) == 1
% %         break;
% %     end
% % end
% % e=i;
% % if lengthofplot == 0
% %      l=e-n;
% %
% % else
% %     l=lengthofplot*2000;
% % end
% g=g+1;
%
% % for i=1:1:l
% %     time(i)=time1(i+l1-l2+n+l1-1);
% %     dataseg11n(i)=dataseg1(i+l1-l2+n+l1-1);
```

```
% %     dataseg12n(i)=dataseg2(i+l1-l2+n+l1-1);
% %     dataseg13n(i)=dataseg3(i+l1-l2+n+l1-1);
% %     dataseg14n(i)=dataseg4(i+l1-l2+n+l1-1);
% % end
% %[dataseg13n,w]= pwelch(dataseg12,1024,[],4096,1000);
% % t=[1:1:l];
%  d=dataseg11n;
% cla(handles.axes1, 'reset');
% axes(handles.axes1);
% plot(time1, dataseg1, 'b');
% hold on;
% cla(handles.axes1, 'reset');
% axes(handles.axes2);
% plot(time1, dataseg2, 'r');
% hold on;
% %cla(handles.axes1, 'reset');
% axes(handles.axes3);
% plot(time2, dataseg3, 'g');
% %hold on;
% % cla(handles.axes1, 'reset');
% axes(handles.axes4);
% plot(time1, dataseg4, 'y');
% hold on;
%
% guidata(hObject, handles);



% --- Executes on button press in pushbutton6.
function pushbutton6_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton6 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
channel = get(handles.channels, 'String');
channels = str2num(channel);
channel1=channels(1)
channel2=channels(2)
channel3=channels(3)
channel4=channels(4)
 filename = get(handles.input_filename, 'String');

global A
global g
global d
global Flag
global XY1
global XY2
%A = load(filename); %importing the data file (.exp)%
%A = A;   %the data is imported into 1 variable A with 2 sub variables data and textdata which
are seperated in this step%
%time1= A.textdata;
```

```
eventstarttime1=get(handles.input_edit1, 'String');
eventendtime1=get(handles.input_edit2, 'String');
lengthofplot1=get(handles.input_edit6, 'String');
lengthofplot=str2double(lengthofplot1);
if lengthofplot == 0
     l=e-n;

else
    l=lengthofplot*1000;
end
l1=(l*(g));
l2=(l*(g-1));
lg=l1-l2-l;
 compstr = strrep(filename, 'txt', 'mat');
 c = strcmp(filename, compstr)
 if c==1
     A=load(filename, '-mat');
     A1=(A.savedata);
 else
%     A = load(filename);
     fid=fopen(filename);
    [vg]=textscan(fid, '%f %f %f %f %f %f %f %f %f %f %f', l, 'HeaderLines',lg);
    A1=cell2mat(vg);
%     %A=cell2mat(A1);
%
 end

time1=A1(:, 1);
if channel1 ==1
    dataseg1 = A1(:, 1);
elseif channel1 ==2
    dataseg1 = A1(:, 2);
elseif channel1 ==3
    dataseg1 = A1(:, 3);
elseif channel1 ==4
    dataseg1 = A1(:, 4);
elseif channel1 ==5
    dataseg1 = A1(:, 5);
elseif channel1 ==6
    dataseg1 = A1(:, 6);
elseif channel1 ==7
    dataseg1 = A1(:, 7);
elseif channel1 ==8
    dataseg1 = A1(:, 8);
elseif channel1 ==9
    dataseg1 = A1(:, 9);
elseif channel1 ==10
    dataseg1 = A1(:, 10);
elseif channel1 == 11
    dataseg1 = A1(:, 11);
elseif channel1 ==12
    dataseg1 = A1(:, 12);
```

```
elseif channel1 ==13
   dataseg1 = A1(:, 13);
elseif channel1 ==14
   dataseg1 = A1(:, 14);
elseif channel1 ==15
   dataseg1 = A1(:, 15);
elseif channel1 ==16
   dataseg1 = A1(:, 16);
elseif channel1 ==17
   dataseg1 = A1(:, 17);
elseif channel1 ==18
   dataseg1 = A1(:, 18);
elseif channel1 ==19
   dataseg1 = A1(:, 19);
elseif channel1 ==20
   dataseg1 = A1(:, 20);
elseif channel1 ==21
   dataseg1 = A1(:, 21);
end

if channel2 ==1
   dataseg2 = A1(:, 1);
elseif channel2 ==2
   dataseg2 = A1(:, 2);
elseif channel2 ==3
   dataseg2 = A1(:, 3);
elseif channel2 ==4
   dataseg2 = A1(:, 4);
elseif channel2 ==5
   dataseg2 = A1(:, 5);
elseif channel2 ==6
   dataseg2 = A1(:, 6);
elseif channel2 ==7
   dataseg2 = A1(:, 7);
elseif channel2 ==8
   dataseg2 = A1(:, 8);
elseif channel2 ==9
   dataseg2 = A1(:, 9);
elseif channel2 ==10
   dataseg2 = A1(:, 10);
elseif channel2 == 11
   dataseg2 = A1(:, 11);
elseif channel2 ==12
   dataseg2 = A1(:, 12);
elseif channel2 ==13
   dataseg2 = A1(:, 13);
elseif channel2 ==14
   dataseg2 = A1(:, 14);
elseif channel2 ==15
   dataseg2 = A1(:, 15);
elseif channel2 ==16
   dataseg2 = A1(:, 16);
```

```
elseif channel2 ==17
    dataseg2 = A1(:, 17);
elseif channel2 ==18
    dataseg2 = A1(:, 18);
elseif channel2 ==19
    dataseg2 = A1(:, 19);
elseif channel2 ==20
    dataseg2 = A1(:, 20);
elseif channel2 ==21
    dataseg2 = A1(:, 21);
end

if channel3 ==1
    dataseg3 = A1(:, 1);
elseif channel3 ==2
    dataseg3 = A1(:, 2);
elseif channel3 ==3
    dataseg3 = A1(:, 3);
elseif channel3 ==4
    dataseg3 = A1(:, 4);
elseif channel3 ==5
    dataseg3 = A1(:, 5);
elseif channel3 ==6
    dataseg3 = A1(:, 6);
elseif channel3 ==7
    dataseg3 = A1(:, 7);
elseif channel3 ==8
    dataseg3 = A1(:, 8);
elseif channel3 ==9
    dataseg3 = A1(:, 9);
elseif channel3 ==10
    dataseg3 = A1(:, 10);
elseif channel3 == 11
    dataseg3 = A1(:, 11);
elseif channel3 ==12
    dataseg3 = A1(:, 12);
elseif channel3 ==13
    dataseg3 = A1(:, 13);
elseif channel3 ==14
    dataseg3 = A1(:, 14);
elseif channel3 ==15
    dataseg3 = A1(:, 15);
elseif channel3 ==16
    dataseg3 = A1(:, 16);
elseif channel3 ==17
    dataseg3 = A1(:, 17);
elseif channel3 ==18
    dataseg3 = A1(:, 18);
elseif channel3 ==19
    dataseg3 = A1(:, 19);
elseif channel3 ==20
    dataseg3 = A1(:, 20);
```

```
elseif channel3 ==21
   dataseg3 = A1(:, 21);
end

if channel4 ==1
   dataseg4 = A1(:, 1);
elseif channel4 ==2
   dataseg4 = A1(:, 2);
elseif channel4 ==3
   dataseg4 = A1(:, 3);
elseif channel4 ==4
   dataseg4 = A1(:, 4);
elseif channel4 ==5
   dataseg4 = A1(:, 5);
elseif channel4 ==6
   dataseg4 = A1(:, 6);
elseif channel4 ==7
   dataseg4 = A1(:, 7);
elseif channel4 ==8
   dataseg4 = A1(:, 8);
elseif channel4 ==9
   dataseg4 = A1(:, 9);
elseif channel4 ==10
   dataseg4 = A1(:, 10);
elseif channel4 == 11
   dataseg4 = A1(:, 11);
elseif channel4 ==12
   dataseg4 = A1(:, 12);
elseif channel4 ==13
   dataseg4 = A1(:, 13);
elseif channel4 ==14
   dataseg4 = A1(:, 14);
elseif channel4 ==15
   dataseg4 = A1(:, 15);
elseif channel4 ==16
   dataseg4 = A1(:, 16);
elseif channel4 ==17
   dataseg4 = A1(:, 17);
elseif channel4 ==18
   dataseg4 = A1(:, 18);
elseif channel4 ==19
   dataseg4 = A1(:, 19);
elseif channel4 ==20
   dataseg4 = A1(:, 20);
elseif channel4 ==21
   dataseg4 = A1(:, 21);
end
%dataseg1 = A(:, 1);
%dataseg2 = A(:, 2);%the column of data containing blood flow velocity in cm/s is the 2nd
column. Seperated from the rest of data
eventstarttime=str2double(eventstarttime1);
eventendtime=str2double(eventendtime1);
```

```
%dataseg1=num2str(dataseg1d);
%dataseg3 = A(:, 3);
%dataseg4 = A(:, 4);
%time1(1,:)=[];  %deleting first few rows containig patient name and details
%time1(1,:)=[];
%time1(1,:)=[];
%time1(1,:)=[];
%time1(1,:)=[];
%time1(1,:)=[];
[m,n]=size(dataseg1);
% for i=1:1:m
%     if eventstarttime== time1(i)
%         TF(i) =1;
%     else
%         TF(i) = 0;
%     end
% %TF=strcmp(eventstarttime,dataseg1);
%     if eventendtime== time1(i)
%         TF1(i) =1;
%     else
%         TF1(i) = 0;
%     end
% end
%TF=strcmp(eventstarttime1,dataseg1);
%TF1=strcmp(eventendtime1,dataseg1);
% for i=1:1:m
%     if TF(i) ==1
%          break;
%     end
%   end
% n=i;
% for i=1:1:m
%     if TF1(i) == 1
%         break;
%     end
% end
% e=i;

g=g-1;

% for i=1:1:l
%     time(i)=time1(i+l1-l2+n+l2-1);
%     dataseg11p(i)=dataseg1(i+l1-l2+n+l2-1);
%     dataseg12p(i)=dataseg2(i+l1-l2+n+l2-1);
%     dataseg13p(i)=dataseg3(i+l1-l2+n+l2-1);
%     dataseg14p(i)=dataseg4(i+l1-l2+n+l2-1);
% end
%[dataseg13p,w]= pwelch(dataseg12,1024,[],4096,1000);
% t=[1:1:l];
% d=dataseg11p;
cla(handles.axes1, 'reset');
axes(handles.axes1);
```

```
plot(time1, dataseg1, 'b');
hold on;
cla(handles.axes2, 'reset');
axes(handles.axes2);
plot(time1, dataseg2, 'r');
hold on;
cla(handles.axes3, 'reset');
axes(handles.axes3);
plot(time1, dataseg3, 'g');
hold on;
cla(handles.axes4, 'reset');
axes(handles.axes4);
plot(time1, dataseg4, 'y');
hold on;

% handles    structure with handles and user data (see GUIDATA)
guidata(hObject, handles);


% --- Executes on button press in pushbutton7.
function pushbutton7_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton7 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)


% --------------------------------------------------------------------
% function uitoggletool1_ClickedCallback(hObject, eventdata, handles)
% % hObject    handle to uitoggletool1 (see GCBO)
%
% datacursormode on;
% %dcm_obj = datacursormode(clippeddataviewer__original12);
% global Flag;
% global XY1
% global XY2;
%
% Flag1 =  Flag;
% menu=findall(get(gcf, 'Children'),'Type','uicontextmenu');
% menuCallback=get(menu, 'Callback');
% dataCursor=menuCallback{2}
% info = getCursorInfo(dataCursor)
% if ~isempty(info)
% disp(info.Position)
% pos={info.Position}
% XYpoint=cell2mat(pos)
% end
%
%     if  Flag1 == 1
%         global XY1;
%          XY1 = cell2mat(pos)
%
%         Flag = 2;
```

```
%     else
%         global XY2;
%          XY2 = cell2mat(pos)
%         % XY2(2) = XYpoint(:,2);
%          Flag = 1;
%     end
%
%
% guidata(hObject, handles);


% --------------------------------------------------------------------
 function uitoggletool1_OnCallback(hObject, eventdata, handles)
% % % hObject    handle to uitoggletool1 (see GCBO)

%dcm_obj = datacursormode(clippeddataviewer__original12);
 global Flag
 global XY1
 global A
 global XY2
 y= [0:0.01:1]
 Flag1 =  Flag;
 if  Flag1 == 1
    menu=findall(get(gcf, 'Children'),'Type','uicontextmenu');
     menuCallback=get(menu, 'Callback');
    dataCursor=menuCallback{2}
    info = getCursorInfo(dataCursor)
    if ~isempty(info)
        disp(info.Position)
        pos={info.Position}
        XYpoint=cell2mat(pos)
    end


    XYpoint=cell2mat(pos)
    global XY1
    XY1(1) = XYpoint(: ,1)
    XY1(2) = XYpoint(:,2)
    Flag = 2;
    x1=XY1(1)
    axes(handles.axes1);
    plot(x1, y, 'b');
    hold on;
    axes(handles.axes2);
    plot(x1,y,  'b')
    hold on;
    axes(handles.axes3);
    plot(x1,y,  'b')
    hold on;
    axes(handles.axes4);
    plot(x1,y,  'b')
    hold on;
```

```
else
   menu=findall(get(gcf, 'Children'),'Type','uicontextmenu');
   menuCallback=get(menu, 'Callback');
   dataCursor=menuCallback{2}
   info = getCursorInfo(dataCursor)
      if ~isempty(info)
         disp(info.Position)
         pos={info.Position}
         XYpoint=cell2mat(pos)
      end
   XYpoint=cell2mat(pos)
   global XY2
   XY2(1) = XYpoint(:,1)
   XY2(2) = XYpoint(:,2)
   x2=XY2(1)
   axes(handles.axes1);
   plot(x2, y, 'b');
   hold on;
   axes(handles.axes2);
   plot(x2,y,  'b')
   hold on;
   axes(handles.axes3);
   plot(x2, y, 'b');
   hold on;
   axes(handles.axes4);
   plot(x2,y,  'b')
   hold on;
   Flag = 1;
end



%
%
 guidata(hObject, handles);
%
%
% ------------------------------------------------------------------
% function uitoggletool1_ButtonDownFcn(hObject, eventdata, handles)
% % hObject    handle to uitoggletool1 (see GCBO)
% % eventdata  reserved - to be defined in a future version of MATLAB
% % handles    structure with handles and user data (see GUIDATA)
% datacursormode on;
% global Flag;
% global XY1
%
% global XY2;
%
% Flag1 =  Flag;
% menu=findall(get(gcf, 'Children'),'Type','uicontextmenu');
% menuCallback=get(menu, 'Callback');
% dataCursor=menuCallback{2}
```

```
% info = getCursorInfo(dataCursor)
% if ~isempty(info)
% disp(info.Position)
% pos={info.Position}
% XYpoint=cell2mat(pos)
% end
%
%     if  Flag1 == 1
%         global XY1;
%          XY1 = cell2mat(pos)
%          %XY1(2) = XYpoint(:,2);
%          Flag = 2;
%     else
%         global XY2;
%          XY2 = cell2mat(pos)
%          %XY2(2) = XYpoint(:,2);
%          Flag = 1;
%     end
%
%
% guidata(hObject, handles);


% --- Executes during object creation, after setting all properties.
 function uitoggletool1_CreateFcn(hObject, eventdata, handles)
% % hObject    handle to uitoggletool1 (see GCBO)

global Flag
 global XY1

 global XY2
% global Flag;
% global XY1
%
% global XY2;
%
% Flag1 =  Flag;
% menu=findall(get(gcf, 'Children'),'Type','uicontextmenu');
% menuCallback=get(menu, 'Callback');
% dataCursor=menuCallback{2}
% info = getCursorInfo(dataCursor)
% if ~isempty(info)
% disp(info.Position)
% pos={info.Position}
% XYpoint=cell2mat(pos)
% end
%
%     if  Flag1 == 1
%         global XY1;
%          XY1 = cell2mat(pos)
%          %XY1(2) = XYpoint(:,2);
%          Flag = 2;
```

```
%     else
%         global XY2;
%          XY2 = cell2mat(pos)
%         % XY2(2) = XYpoint(:,2);
%          Flag = 1;
%     end
%
%
 guidata(hObject, handles);

% --- Executes during object deletion, before destroying properties.
function uitoggletool1_DeleteFcn(hObject, eventdata, handles)
% hObject    handle to uitoggletool1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)


% ------------------------------------------------------------------------
% function uitoggletool1_OnCallback(hObject, eventdata, handles)
% % hObject    handle to uitoggletool1 (see GCBO)
% datacursormode on;
% global Flag;
% global XY1
%
% global XY2;
%
% Flag1 =  Flag;
% menu=findall(get(gcf, 'Children'),'Type','uicontextmenu');
% menuCallback=get(menu, 'Callback');
% dataCursor=menuCallback{2}
% info = getCursorInfo(dataCursor)
% if ~isempty(info)
% disp(info.Position)
% pos={info.Position}
% XYpoint=cell2mat(pos)
% end
%
%     if  Flag1 == 1
%         global XY1;
%          XY1 = cell2mat(pos)
%          %XY1(2) = XYpoint(:,2);
%          Flag = 2;
%     else
%         global XY2;
%          XY2 = cell2mat(pos)
%          %XY2(2) = XYpoint(:,2);
%          Flag = 1;
%     end
%
%
% guidata(hObject, handles);
```

```
% --- Executes during object creation, after setting all properties.
function channels_CreateFcn(hObject, eventdata, handles)
% hObject    handle to channels (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called


% --- Executes during object deletion, before destroying properties.
function channels_DeleteFcn(hObject, eventdata, handles)
% hObject    handle to channels (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)




function channels_Callback(hObject, eventdata, handles)
% hObject    handle to channels (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of channels as text
%        str2double(get(hObject,'String')) returns contents of channels as a double

guidata(hObject, handles);




function savedata_CreateFcn(hObject, eventdata, handles)
% hObject    handle to savedata (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
   set(hObject,'BackgroundColor','white');
end



function savedata_Callback(hObject, eventdata, handles)
% hObject    handle to savedata (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of savedata as text
%        str2double(get(hObject,'String')) returns contents of savedata as a double
```

```
guidata(hObject, handles);


% --- Executes on button press in pushbutton9.
function pushbutton9_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton9 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
[FileName,PathName,FilterIndex] = uigetfile('*.*')
h=strcat(PathName,FileName);
set(handles.input_filename,'String',h);
guidata(hObject, handles);


% --- Executes on button press in pushbutton10.
function pushbutton10_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton10 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

[FileName,PathName,FilterIndex] = uigetfile('*.*')
h=strcat(PathName,FileName);
set(handles.savedata,'String',h);
guidata(hObject, handles);
```

REFERENCES

1. American Sleep Apnea Association – Information
2. National Heart, Lung and Blood Institute – Basic Summary for Sleep Apnea
3. Ambrosio C., Bowman T., Mohsenin V., Quality of life in patients with obstructive sleep apnea, Chest Jan 99, Vol. 115, no. 1 123-129.
4. National Institue of Health – National Center on Sleep Disorders Research
5. Barclay L., Patients with obstructive sleep apnea have less gray matter, Medscape news, Nov. 21, 2002.
6. Operating Instructions – Doppler Box – DWL Compumedics Germany
7. Johnson J.T., Gluckman J. L., Sanders M.H., Management of Obstructive Sleep Apnea, Martin Dunitz Ltd., 2002.
8. http://en.wikipedia.org/wiki/Sleep_apnea
9. Bishop C.C.R., Powell S., Rutt D., Browse N.L., Transcranial Doppler measurement of middle cerebral artery blood flow velocity: a validation study, Stroke Vol 17 No. 5
10. Bradley D.T., Floras J.S., Sleep apnea: implications in cardiovascular and cerebrovascular disease, Lung Biology in Health and Disease, Volume 146.
11. Findley L.J., Barth J.T., Powers D.C., Wilhoit S.C., Boyd D.G., Surratt P.M., Cognitive impairment in patients with obstructive sleep apnea and associated hypoxemia, Chest, Nov. 1986.
12. Zhang R., Zuckerman J.H., Giller C.A., Levine B.D., Transfer Function Analysis of dynamic autoregulation in humans, Am J Physol Heart Circ Physiol 274:233-241, 1998.

13. Fisher J.P., Ogoh S., Young C.N., Raven P.B., Fadel P.J., Regulation of middle cerebral artery blood velocity during dynamic exercises in humans, J Appl Physiol, 105(1):266-273, July 1, 2008.
14. Hayakawa T, Terashima M., Kayukawa Y., Ohta T., Okada T., Changes in cerebral oxygenation and hemodynamics during obstructive sleep apneas, Chest 1996; 109:916-21.Meyer J.S., Ishikawa Y., Hata T., et al, Cerebral blood flow in normal and abnormal sleep and dreaming. Brain cogn 1987; 6:266-94.
15. Hajak G., Klingelhofer J., Schulz-Varszegi M., Sander D., Ruther E., Sleep apnea syndrome and cerebral hemodynamics, Chest 1996; 110:670-79.
16. Aaslid R., Transcranial Doppler sonography. New York: Springer, 1986; 39-59.
17. Fischer A.Q., Chaudhary, B.A., Taormina M. A., Akhtar B.A., Intracranial hemodynamics in sleep apnea, Chest 1992; 102:1402-06.
18. Jenum P., Borgesen S., Intracranial pressure and obstructive sleep apnea, Chest 1989; 95:279-83.
19. Markwalder T., Grolimund P., Seiler R., Roth F., Aaslid R., Dependence of blood flow velocity in the middle cerebral artery on end-tidal CO2 partial pressure: a transcranial ultrasound Doppler study. J Cereb B F and Metab 1984; 4:368-72.
20. Seibler M., Daffertschofer M., Hennerici M., Freud J.T., Cerebral blood flow velocity alterations during obstructive sleep apnea syndrome, Neurology 1990; 1461-62.
21. Newell D.W., Aaslid R., eds. Transcranial Doppler. New York: Raven Press, 1992.
22. Sorteberg W. Cerebral artery blood velocity and cerebral blood flow. In: Newell D.W., Aaslid R. eds. Transcranial Doppler, Chapter 6, New York, Raven press, 1992.
23. Droste D.W., Harders A.G., Rastogi E., A transcranial Doppler study of blood flow velocity in middle cerebral arteries performed during rest and during mental activities.
24. Townsend R.E., Prinz P.N., Obrist W.D., Human cerebral blood during sleep and waking. J Applied Physiol 1973; 35:620-25.

25. Loeppky L.A., Miranda F.G., Elridge M.W., Abnormal cerebrovascular responses to CO2 in sleep apnea patients. Sleep 1984; 7:97-109.
26. Netzer N., Werner P., Jochums I., Lehmann M., Strohl K.P., Blood flow of the middle cerebral artery with sleep-disordered breathing: correlation with obstructive hypopneas. Stroke 1998; 29:87-93.
27. Somers V.K., Dyken M.E., Mark A.L., Abboud F.M., Sympathetic nerve activity during sleep in normal subjects. N Engl J Med. 1993; 328:303-307.
28. Guyton A.C., Cerebral blood flow, the cerebrospinal fluid and brain metabolism. In: Guyton A.C., Basic Neuroscience: Anatomy and Physiology. 2nd ed. Philadelphia, Pa: WB Saunders; 1992:285-295.
29. Dahl A., Lindegaard K-F, et al., A comparison of transcranial Doppler and cerebral blood flow studies to assess cerebral vasoreactivity, Stroke 1992; 23:15-19.
30. Sorteberg W. Cerebral artery blood velocity and cerebral blood flow. In: Newell D.W., Aaslid R. eds. Transcranial Doppler, Chapter 6, New York, Raven press, 1992.
31. Droste D.W., Harders A.G., Rastogi E., A transcranial Doppler study of blood flow velocity in middle cerebral arteries performed during rest and during mental activities.
32. Townsend R.E., Prinz P.N., Obrist W.D., Human cerebral blood during sleep and waking. J Applied Physiol 1973; 35:620-25.
33. Loeppky L.A., Miranda F.G., Elridge M.W., Abnormal cerebrovascular responses to CO2 in sleep apnea patients. Sleep 1984; 7:97-109.
34. Netzer N., Werner P., Jochums I., Lehmann M., Strohl K.P., Blood flow of the middle cerebral artery with sleep-disordered breathing: correlation with obstructive hypopneas. Stroke 1998; 29:87-93.
35. Somers V.K., Dyken M.E., Mark A.L., Abboud F.M., Sympathetic nerve activity during sleep in normal subjects. N Engl J Med. 1993; 328:303-307.

36. Guyton A.C., Cerebral blood flow, the cerebrospinal fluid and brain metabolism. In: Guyton A.C., Basic Neuroscience: Anatomy and Physiology. 2nd ed. Philadelphia, Pa: WB Saunders; 1992:285-295.

37. Dahl A., Lindegaard K-F, et al., A comparison of transcranial Doppler and cerebral blood flow studies to assess cerebral vasoreactivity, Stroke 1992; 23:15-19.

38. Bishop C.C.R., Powell S., Rutt D., Browse N.L., Transcranial Doppler measurement of middle cerebral artery blood flow velocity: a validation study, Stroke Vol 17 No. 5

BIOGRAPHICAL INFORMATION

Gauri Bhave was born in Pune, India on the 2nd of March, 1986. She graduated with a Bachelor's degree in Instrumentation and Control Engineering from the University of Pune, India in June 2007. She then worked in a software company, Tata Consultancy Services in India for one year. In 2008 she received a Dean's Fellowship at the Bioengineering department which gave her an impetus to pursue her Master's degree at the University of Texas, Arlington. Her research interests include signal processing, instrumentation and electronic system design, diagnostic imaging, physiology and life sciences. She is planning to pursue a PhD degree in Biomedical Engineering.