Master of Science Thesis

Artifact Correction for EEG Alpha Wave Measurements

Real Time Alpha Wave And Relaxation State Detection from EEG

Letian Wang B.Sc.

September 29, 2009

DCSC

Delft Center for Systems and Control

TU Delft

Delft University of Technology

Artifact Correction for EEG Alpha Wave Measurements

Real Time Alpha Wave And Relaxation State Detection from EEG

Master of Science Thesis

For obtaining the degree of Master of Science in Systems and Control
at Delft University of Technology

Letian Wang B.Sc.

September 29, 2009

Faculty of Mechanical, Maritime and Materials Engineering · Delft University of Technology

Delft University of Technology

Delft University Of Technology
Delft Center for Systems and Control

The undersigned hereby certify that they have read and recommend to the Faculty of Mechanical, Maritime and Materials Engineering for acceptance a thesis entitled **"Artifact Correction for EEG Alpha Wave Measurements"** by **Letian Wang B.Sc.** in partial fulfillment of the requirements for the degree of **Master of Science**.

Dated: September 29, 2009

Supervisor:

prof.dr.ir. P.M.J. Van den Hof

Readers:

dr.ir. X.J.A. Bombois , dr.ir. A.J. den Dekker, dr.ir. S. de Waele (Philips), dr.ir. M. Jaeger (Philips)

Abstract

The alpha wave derived from EEG signals is a useful frequency character to detect mental states of a person. However, considerable artifacts are introduced in the recording of EEG signals which leads to unreliable results of the alpha wave measurement and detection. An advanced artifact correction algorithm based on the stationary wavelet transform is proposed in the thesis to reduce artifacts from EEG signals. The algorithm employs a new method of soft substitution and threshold estimation. A real time experiment of alpha wave measurement and detection is performed to evaluate the algorithm in the thesis. The results show the algorithm can effectively reduce certain type of artifacts from the EEG signals. The reliability of alpha wave detection is improved.

Acknowledgements

I would like to thank my supervisors prof.dr.ir. P.M.J. Van den Hof dr.ir. X.J.A. Bombois , dr.ir. A.J. den Dekker, dr.ir. S. de Waele (Philips), dr.ir. M. Jaeger (Philips) and prof.dr.ir.Y. Bai for their assistance during the writing of this thesis. I am also grateful to my family and friends for their support.

Delft, University of Technology
September 29, 2009

Letian Wang B.Sc.

Table of Contents

List of Figures

List of Tables

Acronyms

ECG Electrical activity of the heart,Electrocardiography

EEG Electroencephalography

EMA Electrodes movement artifacts

EMG Activation of muscles,Electromyography

EOG Electrooculography

FFT Fast Fourier transform

GUI Graphical User Interfaces

ICA Independent component analysis

MA Muscle artifacts

MAE Mean absolute error

MRA Mallat's Multiresolution Analysis

PSD Power spectrum density

OA Ocular artifacts

SFT Short time Fourier transform

SWT Stationary wavelet transform

Index

Chapter 1

Introduction

1-1 Background

The human body is an intricate and delicate biological system. This system can be regarded as a complex nonlinear mechanical system which has tens of thousands of inputs, state variables and outputs. Interests on the human body has never decreased and research on it has never been stopped since hundreds years ago. Physiological signals are electrical or magnetic signals generated by some biological activity in the human body. They can be considered as outputs of a system. Mental states and emotions can be considered as state variables of a system. More and more physiological signals are extracted and quantitatively measured with the development of sensing related technologies. As a result, research on different mental states and emotions is improved.

A change in the mental state of a person is often reflected as a change in physiological signals. This type of change is referred to as a psycho-physiological event. Based on this fact, devices that are aware of a user's mental state can be developed for detecting psycho-physiological events. This will bring innovative products and make a revolution in both marketing and academic research. Most important is that these innovative products may improve people's life. We can imagine that people will have better awareness of their mental states and control of their emotions with the help of these kind of products which leads to better performance of working and studying. For instance, devices that assist people to reach their higher level of mental concentration will probably improve the efficiency of their studies.

Among all physiological signals, electroencephalography (EEG) signals are popular and fruitful in the application of mental state detection currently. Alpha wave is a frequency characteristic extracted from an EEG signal. It reflects the relaxation and concentration states of a person. Hence, physiological events detection related to relaxation and concentration is achieved by detecting the energy change of the alpha wave from EEG signals. Chapter 2 will give a detailed introduction of the alpha wave.

The measurements of physiological signals differ depending on the environment in which the measurements are performed. The environment includes all the conditions that influence

the measurements such as temperature, illumination, subjects(the human body that is measured), equipments and so on. According to these conditions we can divide the examination environment into two groups, namely:

- Clinical environment. Conditions in a clinical environment are strictly limited. The aim is to acquire considerable quality and content of physiological signals from the measurements.

- Non-clinical environment. Conditions in a non-clinical environment are less limited. There is a balance between the quality of physiological signals and the freedom of conditions.

Figure 1-1 is given as an example of equipments used to measure EEG signals in both clinical and non-clinical environment. The left photo is taken in a clinical environment. It shows an

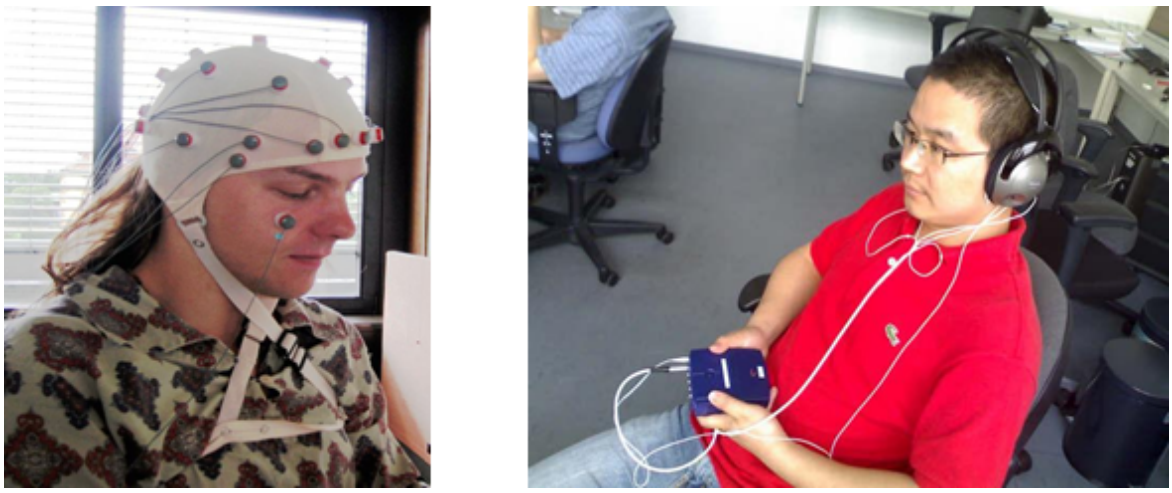

Figure 1-1: Two type of EEG measure environment. left:clinical environment right: non-clinical lifestyle environment

EEG measurement by a cap with 32 electrodes (sensors) on it. There is an extra electrode on the face of the subject. The extra electrode records the electrooculography (EOG). The right photo shows a non-clinical environment in which an EEG measurement is made by a headphone with only 2 electrodes on it. In this thesis, a non-clinical environment is also called a lifestyle measurement because it can easily be performed by non-specialized people in daily life.

Currently, most studies on the alpha wave measurement and detection are in a clinical environment. This is also the same situation for artifact detection (detecting and removing disturbance from EEG signals)[Fatourechi et al., 2007]. The ongoing research of a non-clinical alpha wave measurement and artifact detection is done by the experience processing group from Philips Research Eindhoven. A prototype of mp3 player made by Philips Research is able to detect the alpha wave of a person. In this thesis, an alternative alpha wave detection algorithm will be proposed and compared with the algorithm implemented in the prototype.

1-2 Psycho-physiological Events

As is mentioned in the previous section, psych-physiological events reflect the changes of people's mental states. Most of the mental states can be classified and projected in a 2D plane with two parameters as is shown in Figure 1-2. The two parameters are arousal and

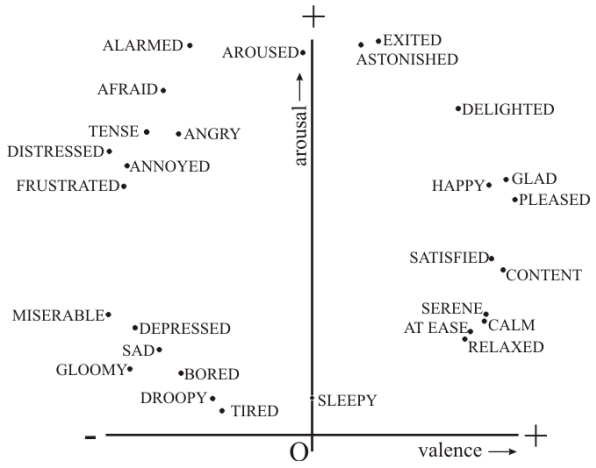

Figure 1-2: Classification of mental states in two dimensions [Healey, 2001]

valence. Arousal reflects the state of being awake of a person. With higher arousal, people are more active and excited. Valence characterizes the positive and negative mental states of a person.

Based on this classification, each mental state is presented by a point in the 2D plane in Figure 1-2. Changes of mental states can be regarded as going from one point to an other on the plane.

1-2-1 Physiological Signals

Physiological events are reflected by physiological signals. Physiological signals are electrical or magnetic signals. They are generated by some biological activities in the human body. Physiological signals have widely different sources. Some typical types of physiological signals and related examination methods are given below [Fatourechi et al., 2007].

- Activation of muscles, Electromyography (EMG)
- Conductivity of peripheral nerve, Electroneuronography (ENoG)
- Electrical activity of the heart, Electrocardiography (ECG)
- Resting potential of the retina, Electrooculography (EOG)
- Electrical activity of neurons within the brain, Electroencephalography (EEG)
- Electrical resistance of the skin, Galvanic skin response (GSR)

1-3 The Objective of the Project

1-3-1 Problem Statements

In this MSc project, focus will be put on relaxation state detection in a non-clinical environment. This will enable innovative products in the field of health care. In particular, algorithms will be developed to measure alpha wave from EEG signals. However, during the measurement of EEG signals, considerable artifacts present in EEG signal will have a large impact on the alpha wave measurement. This leads to unreliable result of relaxation state detection. To solve the problem, disturbance should be eliminated from EEG signal. In other words, artifact correction approaches should be developed to provide reliable results of the detection of the relaxation state.

1-3-2 Objective

Therefore, the objective of the project is

- Developing an alpha wave measurement algorithm including an artifact correction method for the application of relaxation state detection in a non-clinical real time environment.
- Evaluation of the algorithm mentioned above by comparing the alpha wave measurement performance with the existed algorithm in a non-clinical real time environment.

The existed algorithm is currently implemented on the prototype of mp3 player from Philips Research. The description of the existed algorithm can be found in 2-3. The algorithm proposed by this thesis will also be introduced in 2-3. The objective is illustrated in Figure 1-3

1-3-3 Emphasis

In the procedure of developing alpha wave measurement algorithm, most effort will be put on the step of developing an artifact correction method. Artifact correction is very crucial for the reliability of the relaxation states detection results. This is because, in a non-clinical measurement, plenty of artifacts are introduced to the EEG signal which causes feature changes of EEG signals. If we use these contaminated EEG signals to detect relaxation state, the results will be highly unreliable. Artifacts correction method which removes the disturbance from EEG signals will improve the reliability of relaxation state detection and ensures that the detection algorithm shows a good performance in a non-clinical environment.

1-4 Outline

Chapter 2 gives an introduction to EEG signals, artifacts and alpha wave measurement algorithms. Chapter 3 focuses on artifact correction method including the wavelet theory and its implementation on EEG artifact correction. An artifact simulation experiment is designed to

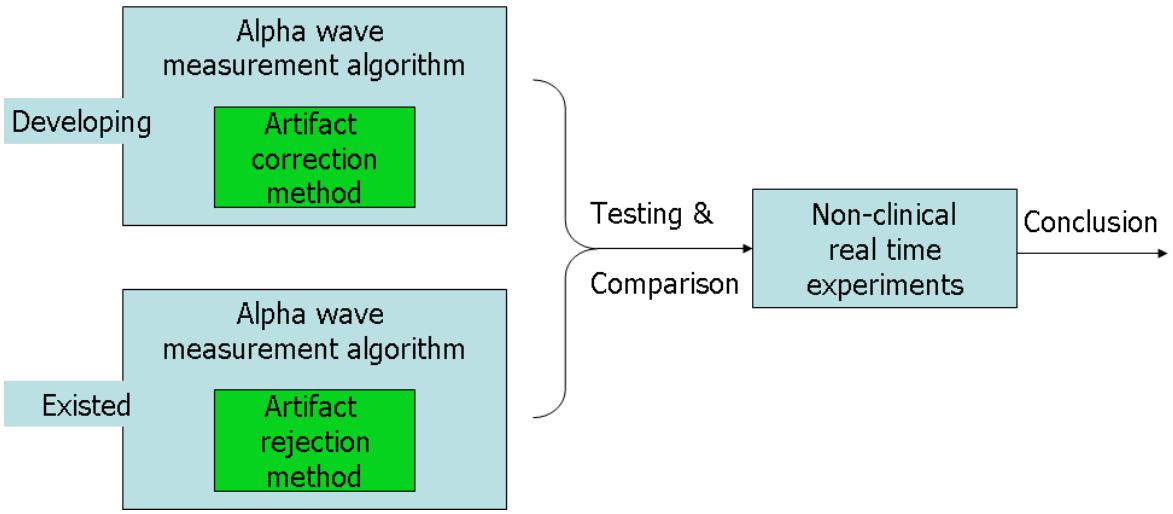

Figure 1-3: Two different alpha wave measurement algorithms will be evaluated in non-clinical real time experiments designed in this thesis.The upper block shows the algorithm that will be proposed in this thesis.

test and optimize the artifact correction method as well as the alpha wave measurement algorithm. Chapter 5 provides a real time relaxation state detection experiment including setup, results and discussion. Evaluation on alpha wave measurement and relaxation state detection will be performed based on the experiment results. The last chapter is the conclusion of the thesis including the achievements and future work.

Chapter 2

Alpha Wave Measurement

In this chapter, EEG signal and its frequency information will be introduced firstly. Then the general procedure of alpha wave measurement will be discussed. After that, different categories of artifacts and their influence on alpha power will be discussed. Finally, a first impression of the influence of artifacts on two detectors will be given.

2-1 Electroencephalography (EEG)

Chapter 1 has already given a rough introduction to EEG and its measurement. In this section, EEG will be discussed in detail.

2-1-1 EEG Signals

Electroencephalography (EEG) is the recording of electrical activity along the scalp produced by the firing of neurons within the brain [Creutzfeldt et al., 1966]. EEG signals are the electrical signals recorded as voltage changes along time. A 3-second segment of EEG signals $x(t)$ is shown in Figure 2-1. X is the time in seconds. Y is the amplitude in microvolt. As we can see in the figure, the EEG signals look like random oscillations changing along time. We can also investigate the spectrum of EEG signals in the frequency domain through the Fourier transform. Figure 2-2 depicts the spectrum of a 3-second segment of EEG signals sampled at 128Hz.

2-1-2 Mental States in EEG Signal

The EEG signals can be typically described in terms of rhythmic activity and transients. The rhythmic activity is divided into bands by frequency. Certain mental states can be reflected in these frequency bands. Table 2-1 lists 6 frequency bands named by alphabet. Frequency ranges, location in the scalp and normal mental states are listed for each frequency band.

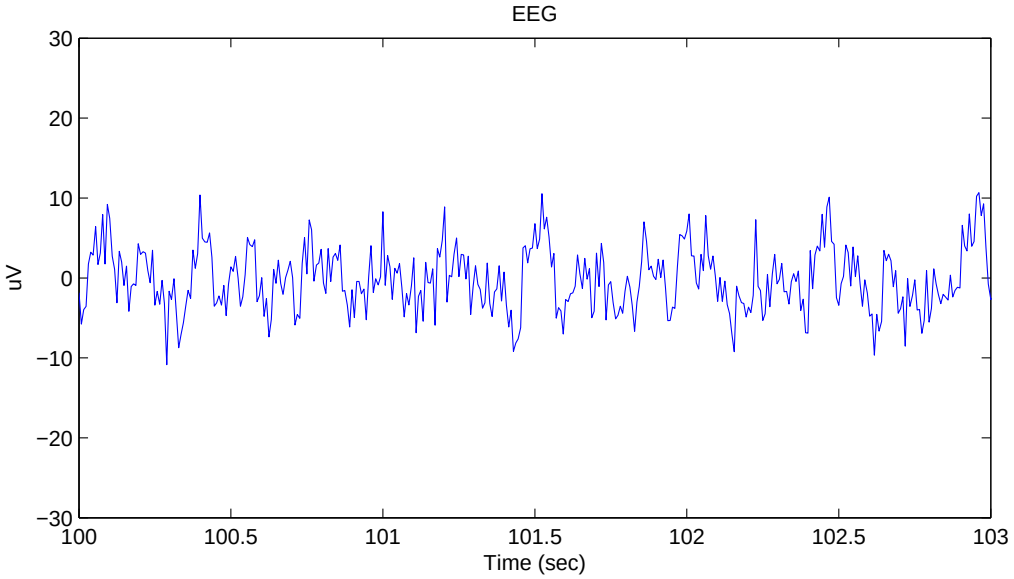

Figure 2-1: A 3-second segment of EEG signal. EEG signals look like random oscillations changing along time.

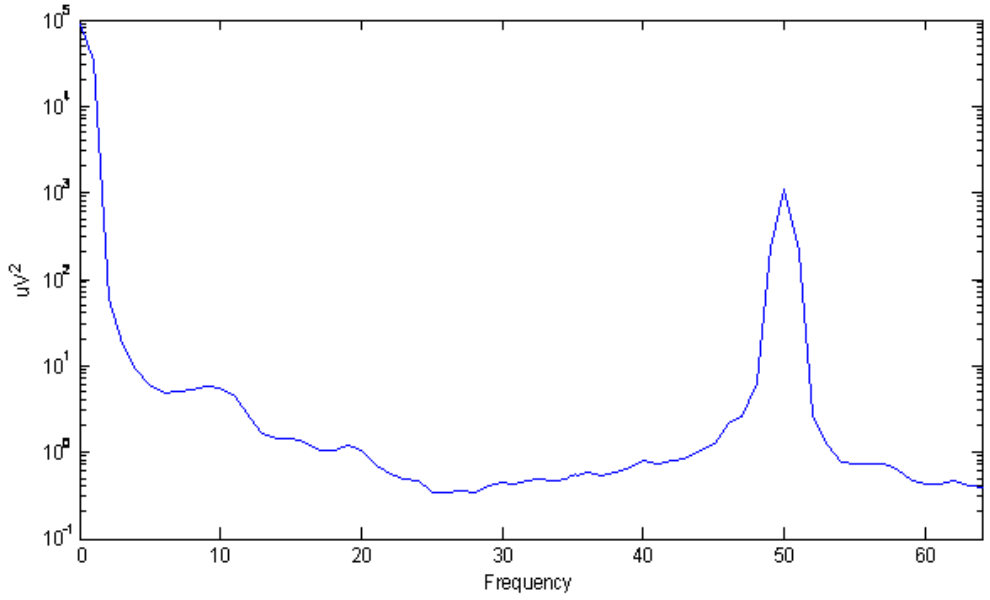

Figure 2-2: Spectrum of a 3-second segment EEG signal. Highest amplitude near 0-2Hz is caused by EEG recording equipment. The useful EEG frequency band is 2-40Hz. The peak near 50 Hz is caused by 50Hz electric interference

What we are interested in is the alpha wave. The frequency band of alpha wave is 8-12 Hz. The increase of the alpha wave is achieved by closing eyes and relaxation. Based on this fact, the relaxation state can be reflected by measuring the alpha wave level.

Table 2-1: Different frequency bands of EEG signals. The frequency band of alpha wave locates at 8-12 Hz.The increase of alpha wave is achieved by closing eyes and relaxation [E. Niedermeyer, 1999]

Bands	Frequency	Location	normal mental state
Delta	up to 4 Hz	frontally in adults, posteriorly in children; high amplitude waves	Adults slow wave sleep babies
Theta	4-7 Hz		young children drowsiness or arousal in older children and adults
Alpha	8-12 Hz	posterior regions of head, both sides, higher in amplitude on dominant side. Central sites (c3-c4) at rest .	closing the eyes and relaxation.
Beta	12-30 Hz	both sides, symmetrical distribution, most evident frontally; low amplitude waves	active, busy or anxious thinking, active concentration
Gamma	30-100 Hz		certain cognitive or motor functions

2-2 Baseline Alpha Wave Measurement

Usually, several alpha wave measurements are performed during one relaxation state detection experiment. The first measurement is called the baseline measurement. The baseline alpha wave measurement is to record the clean EEG signal and the normal alpha level of a subject. The normal alpha level will be regarded as the baseline alpha level of the subject in the following measurements. The baseline measurement is also used to initialize parameters for some alpha wave detection algorithms. The alpha wave detection algorithm proposed in this thesis is one of them.

2-3 Alpha Wave Detection Algorithm

The alpha wave level is quantized by alpha power. Alpha power is the amplitude in alpha bands in the frequency domain. The relaxation state can be detected by investigating the

changes of alpha power. The general algorithm of an alpha wave measurement categorized in this thesis consists of four steps namely

1. Pre-processing of the EEG signal
2. Fourier transformation of the EEG signal from time domain to frequency domain
3. Computation of the power of alpha waves
4. Normalization

The four steps are introduced in detail in the following subsections.

2-3-1 Pre-processing of the EEG Signal

After we acquire the raw EEG signal data, we will pre-process it to make it adapt to the algorithm that will be discussed later. The two main steps of pre-processing are re-sampling and windowing the EEG signal. Both steps will now be described in more detail.

- Re-sampling
  An EEG signal is recorded with a sampling rate of 512 Hz. It will be down sampled to 128 Hz.
- Windowing

  A time moving window is performed on the EEG signals shown in Figure 2-3. The

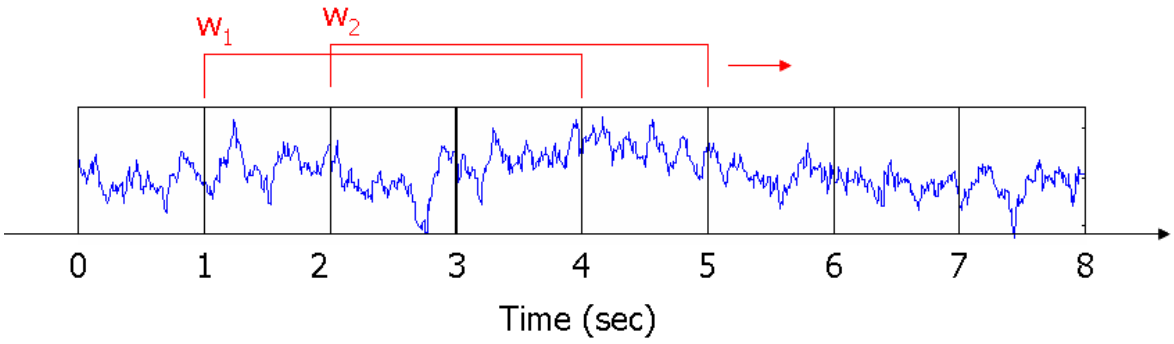

Figure 2-3: Sketch of the time moving window. The window length is 3 seconds and is updated every 1 second. The moving step is 1 second.

window length is 3 seconds. That means further processing and analysis will be done for each 3-second segment (epoch) of EEG signals. One segment of EEG signals is updated every one second by adding the new coming one-second data and throwing away one-second data. Here we define a new time variable T . The interval between two nearby T and $(T+1)$ is 1 second. At time t, the current EEG segment is

$$x(T) = \{x(t), x(t-1), x(t-2), ..., x(t-128*3+1)\}$$

After 1 second, the EEG segment is updated and becomes

$$x(T+1) = \{x(t+128), x(t+128-1), x(t+128-2), ..., x(t-128*2+1)\}$$

2-3-2 Time-Frequency Transformation

Each segment of EEG signals is transformed from the time domain $x(T)$ into frequency domain $X_T(f)$, namely

$$x(T) \rightarrow X_T(f)$$

Where f is the frequency in Hertz. The sampling frequency of f equals 128Hz. The short time Fourier transform (SFT) is employed. Particularly, the method is a matlab function pwelch – An averaged modified periodogram method of spectral estimation. The output of the function is the power spectrum density (PSD) of the EEG segment. More detail about pwelch method can be found in [Hayes, 2008]. Figure 2-2 shows the PSD of one segment of EEG signals.

2-3-3 Computation of the Power in Alpha Waves

$X_T(f)$ contains the power of all the frequency bands of one segment of the EEG signals at time T. We add up all the power that belongs to alpha wave namely

$$P(T) = \sum_{f=8}^{12} |X_T(f)|^2$$

The frequency grid is 1. Now we can plot $P(T)$against time T to see the alpha power changes along time. It can be interpreted that the higher $P(T)$ is, the more alpha power the subject has at time T. An example is given in Figure 2-4 with 90 seconds alpha power investigation from time 150 to 240 seconds. In the figure, P has higher values during 180-220 seconds time interval . Thus we can conclude that the subject has more alpha wave during that time.

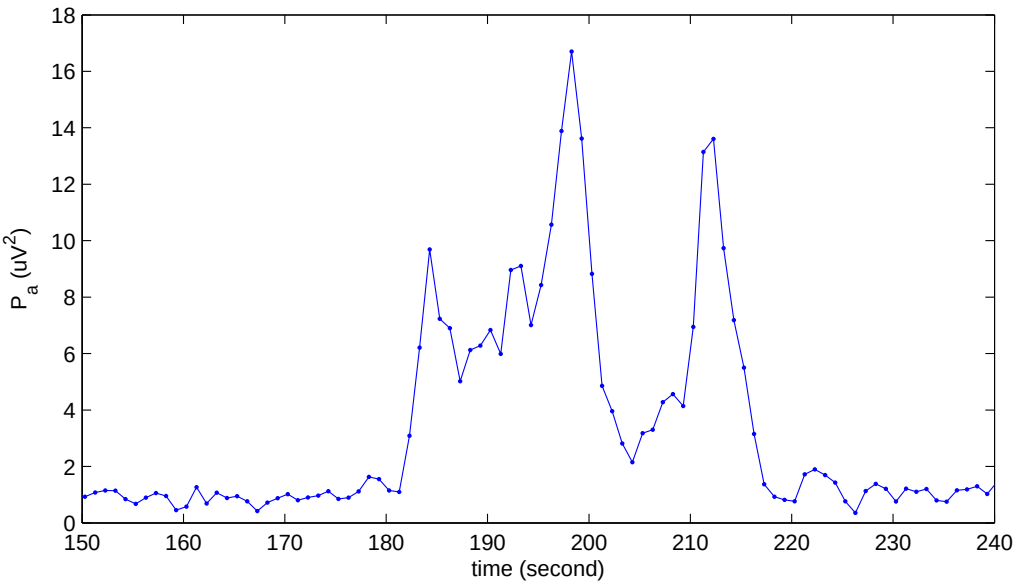

Figure 2-4: Alpha power strength changing over time, namely $P(T)$ against T. During 180-220 seconds, alpha power is high compared with other periods.

2-3-4 Normalization

Because the alpha power varies for various subjects, $P(T)$ will be normalized. Two ways of normalization of the alpha power $P(T)$ will be discussed in the followings. The first normalization is proposed by this thesis. The other one is currently implemented in the Neuro-mp3 player in Philips Research.

The first way of normalization is called the absolute method. The normalized alpha power is $P^{Abs}(T)$. The alpha power level is determined by measuring the amplitude of $P^{Abs}(T)$. $P^{Abs}(T)$ is therefore called the detector of the alpha power level. The maximum P_{max} and minimum P_{min} of the subject, recorded in the baseline measurement, are used. The normalization is described as

$$P^{Abs}(T) = \frac{P(T) - P_{min}}{P_{max} - P_{min}}$$

Because P_{max}, P_{min} are taken from the baseline measurement, it is possible that the $P(T) > P_{max}$ or $P^{Abs}(T) > P_{min}$ in the current measurement. So the range of $P^{Abs}(T)$ can be larger than [0,1]. But normally, $P^{Abs}(T) \in [0, 1]$.

The second way is called the ratio method. The alpha power detector $P^{Ratio}(T)$ is the ratio between alpha power $P(T)$ and power $P_{all}(T)$ in frequency band [5,40] Hz. The normalization is

$$P^{Ratio}(T) = \frac{P(T)}{P_{all}(T)}$$

The range of $P^{Ratio}(T)$ is always $P^{Ratio}(T) \in (0, 1]$.

2-3-5 The Proposed Algorithm in This Thesis

In this thesis, the proposed algorithm of alpha wave measurement is based on the absolute detector $P^{Abs}(T)$. The general procedure of the algorithm is shown in Figure 5-1.

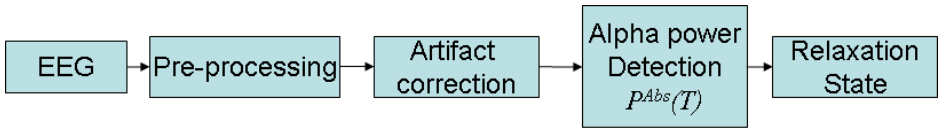

Figure 2-5: A sketch of the procedure of the algorithm based on $P^{Abs}(T)$. Raw EEG signals are transferred to alpha power and relaxation state after serval signal processing procedures

Currently, the alpha wave measurement algorithm implemented in the Neuro-mp3 player in Philips Research is based on the ratio detector $P^{Ratio}(T)$. The algorithm is shown in Figure 5-2. This algorithm will be used as a comparision with the proposed algorithm in Figure 5-1.

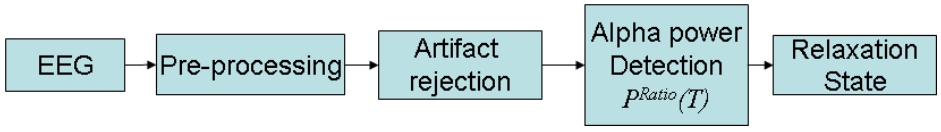

Figure 2-6: A sketch of the procedure of the algorithm based on $P^{Ratio}(T)$

The differences between the two algorithms are

- two different detectors $P^{Abs}(T)$ and $P^{Ratio}(T)$
- algorithm $P^{Abs}(T)$ involves the artifact correction procedure while algorithm $P^{Ratio}(T)$ includes artifact rejection procedure. Artifact correction and rejection will be introduced in Section 2-4-4.

Both algorithms, named by their detectors, the absolute algorithm and the ratio algorithm, will be tested and compared in Chapter 4 and Chapter 5.

2-4 Artifacts

2-4-1 Introduction

During the measurement EEG signals, other kinds of signals, such as ECG (heart beat signals), EMG (muscle signals), 50Hz electric interference, so called artifacts, are also captured by electrodes. The recorded signals are actually mixture of EEG signals and artifacts. We call this mixture contaminated EEG signals.

Artifacts are undesired signals that can introduce significant changes in neurological signals and ultimately affect the neurological phenomenon [Fatourechi et al., 2007]. Some types of artifacts will increase or decrease alpha power which leads to mistakes in the alpha wave measurement. In the application of relaxation state detection, we will have wrong detections caused by artifacts.

2-4-2 Types of Artifacts

There are many types of artifacts. Some of them will not influence the frequency band of alpha waves. For instance, if we look at Figure 2-2, we see a sharp peak around 50 Hz. This is not a feature of EEG signals, however, it is caused by the artifacts of 50Hz electric interference. Luckily this peak is located far from the alpha frequency bands. Therefore, it does not influence the power change of the alpha waves. Artifacts that have relatively considerable influence on the alpha wave can be categorized as follows:

- Ocular artifacts (OA)
  Ocular artifacts are caused by eye movements, such as blinking eyes and rolling eyeballs. Figure 2-7 shows a segment of EEG signals contaminated by eyes blinking. Figure 2-8 shows a segment of EEG signals contaminated by eyeballs rolling. In the frequency domain, ocular artifacts increase the power of EEG signals from 2Hz to 20 Hz. Unluckily, alpha waves locate between 8Hz and 12Hz. Hence, the presence of ocular artifacts in EEG signals will cause unreliable detection of alpha power.
- Electrodes movement artifacts (EMA)
  This type of artifacts is caused by the contact changes between electrodes and the skin of the subject. If the subject moves his head, the electrodes on the headphone will have relative movement on the skin of the subject. This causes the changes of conduction between electrodes and skin which disturbs the recording of EEG signals.

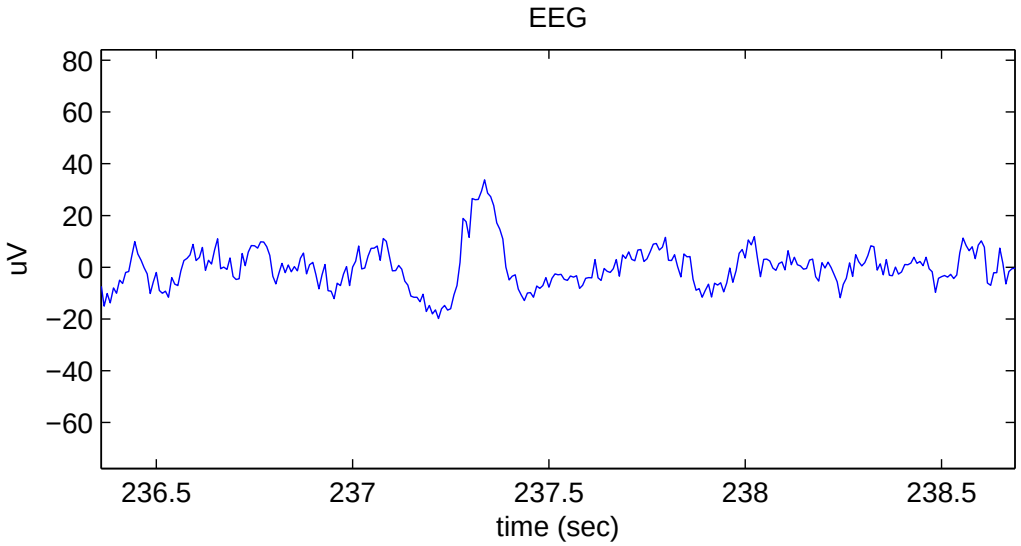

Figure 2-7: An EEG signal contaminated by eyes blinking. Between [237.2 , 237.5], an eye blink happens and reflects on the EEG signal as a small peak

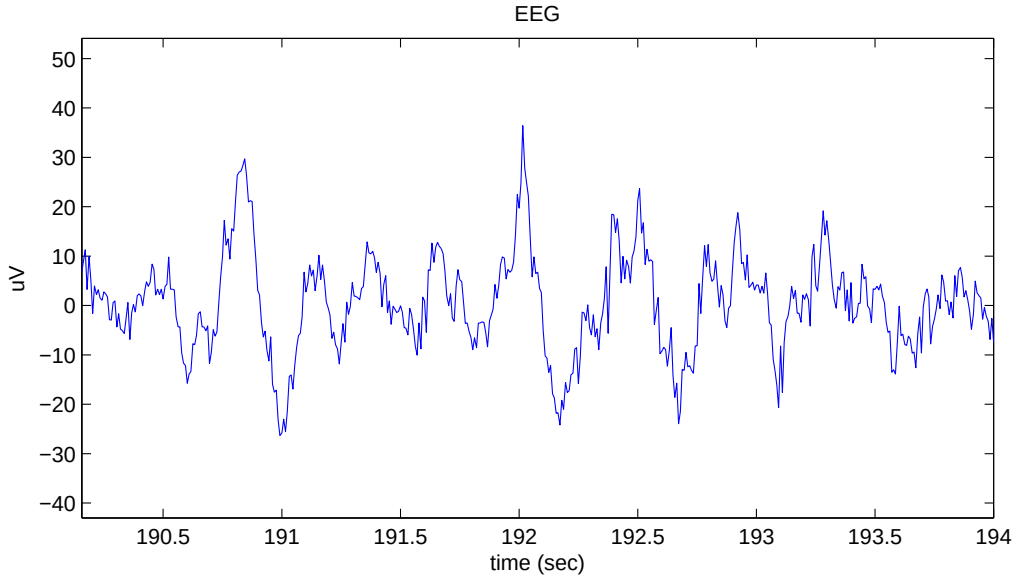

Figure 2-8: An EEG signal contaminated by eyeballs rolling. Large and slow oscillations are introduced to the EEG signal.

Electrodes movements may occur with different speed and in different directions, so the shape of electrodes movement artifacts varies a lot. Figure 2-9 shows the shape of electrodes movement artifacts caused by slowly head shaking. As we can see in the figure, the artifacts have tremendous amplitudes compared with EEG signals so that we can hardly see EEG signals when disturbed by EMA.

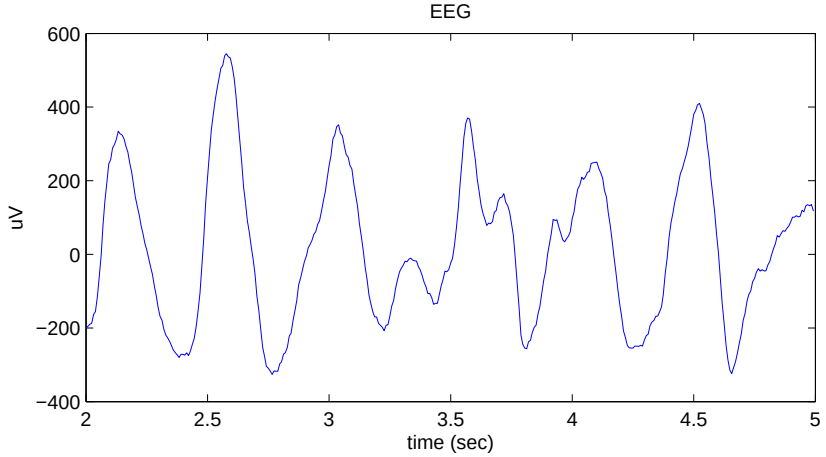

Figure 2-9: An EEG signal contaminated by the artifacts of head shaking. The artifact has tremendous amplitudes compared with normal EEG signals

In the frequency domain, EMA also tremendously increase the power of EEG signals from 0Hz to about 30Hz. Alpha wave measurement and detection will seriously be influenced by the electrodes movement artifacts.

- Muscle artifacts (MA)
  Like the brain, muscles also generate electrical signals. These signals are picked up by EEG electrodes and become muscle artifacts. MA look like fast oscillations depicted in Figure 2-10. EEG signals disturbed by MA have larger amplitudes than normal EEG signals. People have large number of muscles all over their bodies. The muscle movement that happens near electrodes, such as teeth squeezing, will have huge impact on the power spectrum of EEG signals. Usually, presence of MA in EEG signals will increase the power of EEG signals in the high frequency band from roughly 20Hz to 50Hz.

Figure 2-11 gives an illustration of the influence of artifacts on the total EEG frequency band. The x axis represents the frequency band of an EEG signal. The y axis shows how serious the power in certain frequency band are influenced by the artifacts. As is seen in the figure, electrodes movement artifact is the most serious artifact among three. It is because EMA introduces the largest artifacts to the alpha band. In a non-clinical measurement, EMA are inevitable. Therefore, it is important to remove the influence caused by EMA.

2-4-3 Influence of Artifacts on the Detectors

Two alpha power detectors were introduced in Subsection 2-3-4. Based on the information of artifacts, a first impression of how artifacts influence the two detectors will be discussed.

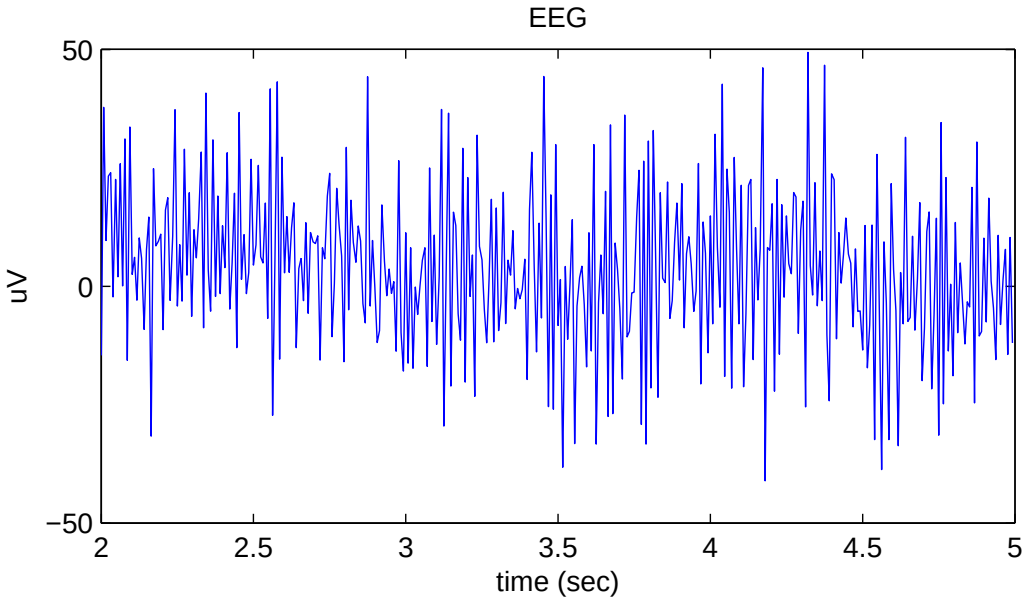

Figure 2-10: An EEG signal contaminated by the artifacts of teeth squeezing. Large and fast oscillations are introduced to the EEG signal

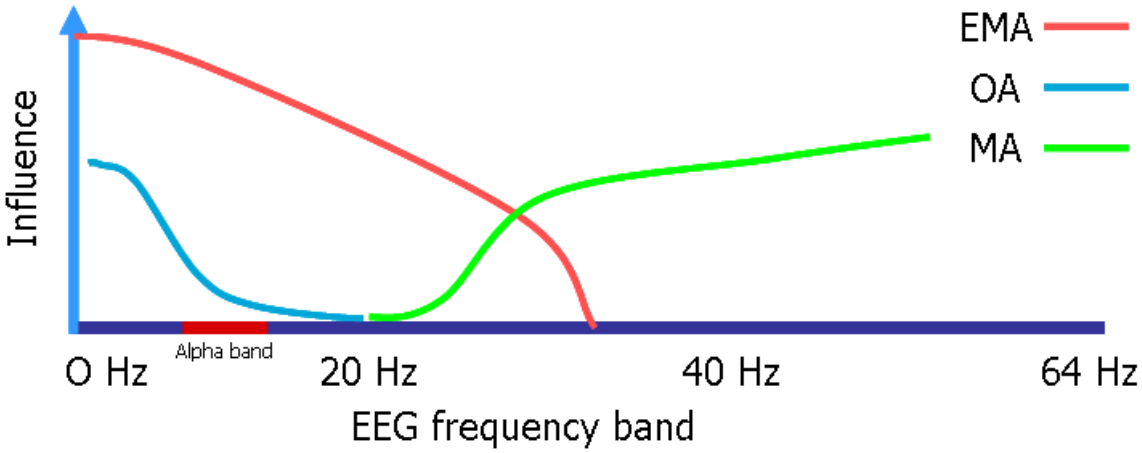

Figure 2-11: A sketch of artifacts and EEG frequency bands. The curves show the relative influence on alpha power strength on frequencies according to different type of artifacts

Before the discussion, an assumption is made below in order to introduce the quality criterion of a detector.

- Assumption
  Alpha power is contributed by the pure EEG signal and the artifacts namely

$$P(T) = P_{pureEEG}(T) + P_{artifacts}(T). \tag{2-1}$$

$P_{artifacts}(T)$ can be regarded as the error namely

$$\epsilon = P(T) - P_{pureEEG}(T). \tag{2-2}$$

When all the alpha power comes from $P_{pureEEG}(T)$, $\epsilon = 0$. This is also the same case for the normalized alpha power $P^{Abs}(T)$ and $P^{Ratio}(T)$ namely

$$\epsilon_{Abs} = P^{Abs}(T) - P^{Abs}_{pureEEG}(T).$$

$$\epsilon_{Ratio} = P^{Ratio}(T) - P^{Ratio}_{pureEEG}(T). \tag{2-3}$$

So we can regard Equation 2-3 as the quality criterion of a detector. If ϵ is close to 0, the detector will have a good performance.

The absolute detector $P^{Abs}(T)$ is only relevant to the frequency band of alpha wave [8-12]Hz. From Figure 2-11, ocular artifacts and electrode movement artifacts will have influence on $P^{Abs}(T)$. It is easy to see the detector $P^{Abs}(T)$ tends to have positive errors. The advantage of $P^{Abs}(T)$ is that artifacts from other frequency bands will not influence it.

The ratio detector $P^{Ratio}(T)$ is relevant to the frequency band of [5-40]Hz. Three types of artifacts will influence it. However, the influence is nonlinear due to the ratio. If there is one of the three types of artifacts present in the EEG signal, both $P(T)$ and $P_{all}(T)$ will increase. Because $P_{all}(T)$ will always increase more as is shown in the Figure 2-11, the ratio $P^{Ratio}(T)$ will probably decrease. This infers that the detector $P^{Ratio}(T)$ is less sensitive to the artifacts. The error of the detector $P^{Ratio}(T)$ tends to be negative.

2-4-4 Artifact Elimination

Artifact elimination can be categorized into two groups [Fatourechi et al., 2007]

- Artifact correction
  Artifact correction is to remove the artifacts from the EEG signal while keeping the pure EEG signal.
- Artifact rejection
  Artifact rejection is to remove the EEG signal which contains the artifacts.

The advantage of artifact rejection is the complete elimination of artifacts. The disadvantage is the loss of EEG signals. Relatively clean EEG signals can be obtained from artifacts contaminated EEG signal by applying an artifact correction method. However, artifacts may not be removed thoroughly. The relatively clean EEG signal may still contain artifacts.

Chapter 3 will introduce the proposed artifact correction method which is employed in the algorithm $P^{Abs}(T)$. The artifact rejection method in the algorithm $P^{Ratio}(T)$ can be refer to Section A-2 in Appendix 1.

2-5 Summary

In Chapter 2, EEG signal and its frequency information were introduced firstly. The frequency band of alpha wave is 8-12Hz. Then the general procedure of alpha wave measurement was discussed. Two algorithms based on different detector are introduced. After that, different categories of artifacts and their influence on alpha power were discussed. Three types of artifacts, namely ocular artifacts(OA), electrodes movement artifacts (EMA) and muscle artifacts (MA), are introduced. Finally, a first impression of the influence of artifacts on two detectors was given. The absolute detector $P^{Abs}(T)$ tends to have positive errors. The advantage of $P^{Abs}(T)$ is that artifacts from other frequency bands will not influence it. $P^{Ratio}(T)$ is less sensitive to the artifacts. The error of the detector $P^{Ratio}(T)$ tends to be negative. More results will be given in the later chapters. The next chapter, which is the core chapter of the thesis, will discuss the artifact correction method.

Chapter 3

Wavelet Based EEG Artifacts Correction

As discussed in the previous chapter, it is necessary to eliminate artifacts from EEG signals. Otherwise the result of the alpha wave measurement is far less reliable. Therefore, an effective approach dealing with artifacts is crucial in the whole process of alpha wave measurement and event detection from EEG. In this chapter, wavelet based EEG artifact correction method is proposed as an effective and efficient approach that eliminates artifacts from the EEG signal. Details will be introduced in the following sections.

3-1 Why Wavelets

In the literature survey [Wang, 2009] four artifacts correction methods were mentioned. These methods are summarized below. In addition, for choosing the wavelet analysis method as the method to be implemented in the artifact correction algorithm proposed in this thesis are also given.

- Linear filtering

  Linear filtering is useful for removing artifacts located in certain frequency bands that do not overlap with those of the neurological phenomena of interest. However, ocular artifacts and electrodes movement artifacts both overlap with the frequency bands where alpha waves locate. Hence this method is out of consideration.

- Independent component analysis(ICA)

  Independent component analysis(ICA) can separate different components of signals such as artifact components and desired signal components based on multichannel EEG signals. The number of channels directly influence the performance of separation between artifact and pure EEG components. There is only two channels available in the experiment conditions of this thesis due to the limitation of the EEG measure equipment

shown in Figure 1-1 in Chapter 1. Two channels will not guarantee a correct separation between artifacts and pure EEG component. Therefore ICA method is not suitable in this case.

- Regression method

  This method is useful for removing artifacts that can be separately measured from EEG signals. However, artifacts that are very difficult to separately measure from EEG, such as the artifacts caused by muscle movements and electrodes movements, cannot be removed by regression method. Even for artifacts that can be separately measured, we still need additional electrodes to measure them, which is not allowed due to a product costs increase.

The wavelets analysis method requires only single channel EEG measurement and does not need extra measurement of artifacts. Meanwhile, the wavelet analysis method performs as a filter that can remove artifacts located in frequency bands that overlap with those of the neurological phenomena of interest. Compared with the methods mentioned above, the wavelet analysis method conquers all these drawbacks. Therefore, in this thesis, this approach is chosen to remove artifacts from EEG signals.

3-2 Wavelet Analysis in the Time-Frequency Domain

Currently, analysis of frequency information of EEG signals is accepted as the most reliable and convenient approach to characterize the feature of EEG signals. It is well known that the Fourier transform is a powerful tool to perform this frequency domain analysis. Fourier analysis breaks down a signal into constituent sinusoids of different frequencies. This technique offers us a transform of view of the signal from time-based to frequency-based. However, Fourier analysis suffers a serious drawback. In transforming to the frequency domain, time information is lost. When looking at a Fourier transform of a signal, it is impossible to tell when a particular event took place. If the signal properties do not change much over time,which is the case of a stationary signal, this drawback is not very important. Unluckily, EEG signals contain numerous non-stationary and transitory characteristics due to complicated and changeable mental states of human beings. Fourier analysis is not suited to detect them. A popular solution to overcome this drawback is wavelet analysis which provides a time-frequency-based approach to analyze the non-stationary EEG signals and transient artifacts. In the following section, wavelet analysis theory will be discussed in detail.

3-3 Wavelet

It is known that Fourier analysis represents a continuous signal $x(t)$ using a linear combination of sinusoidal components with different frequencies which is

$$x(t) = \int_{\omega=-\infty}^{+\infty} X(\omega) e^{jwt} d\omega \tag{3-1}$$

Where $X(\omega)$ is the fourier coefficient with parameter of frequency: ω. Similar to Fourier analysis, wavelet analysis also breaks a signal into various frequency components by using wavelets which can be shown as

$$x(t) = \int_{a=0}^{+\infty} \int_{b=0}^{+\infty} C(a,b)\psi_{a,b}(t)dbda \tag{3-2}$$

Where the $C(a,b)$ are called wavelet coefficients. $\psi_{a,b}(t)$ is called a wavelet with parameters a and b. The difference between Fourier analysis and wavelet analysis is their basis functions. Wavelet analysis uses 'wavelets' while Fourier analysis uses sine waves. A wavelet is a waveform of effectively limited duration that has an average value of zero. An example wavelet and a sine wave as comparison are shown in Figure 3-1.

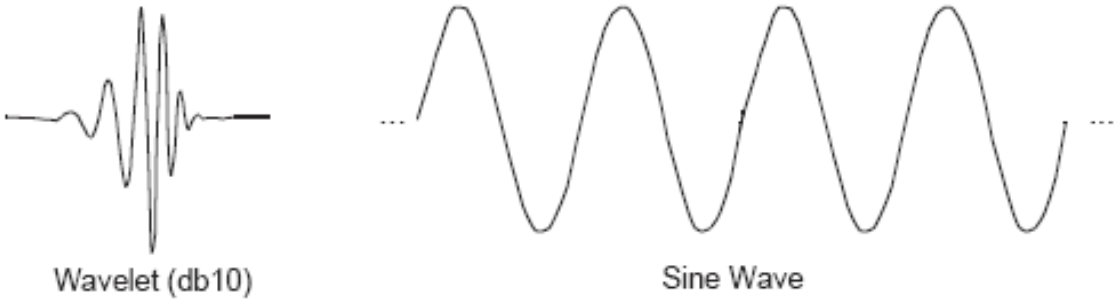

Figure 3-1: Comparison of wavelet and sine wave. Sinusoids are smooth and predictable. Wavelets are irregular and asymmetric.

Compared with wavelets, sinusoids do not have limited duration. They extend from minus to plus infinity. And where sinusoids are smooth and predictable, wavelets tend to be irregular and asymmetric. A set of wavelets that represents different frequency components are generated by one original wavelet called 'mother wavelet'.

3-3-1 Property of a Wavelet

Not all functions are a candidate for a wavelet $\psi_{a,b}(t)$. A wavelet has to satisfy the following requirements.

1. Admissibility condition

$$\int_{\omega=-\infty}^{+\infty} \frac{|\Psi(\omega)|^2}{\omega} d\omega < +\infty$$

$\Psi(\omega)$ stands for the Fourier transform of $\psi(t)$. The admissibility condition guarantees that square integrable functions $\psi(t)$ can be used to first analyze and then reconstruct a signal without loss of information [Sheng, 1996]. The admissibility condition implies that the Fourier transform of $\psi(t)$ vanishes at frequency zero,i.e.

$$|\Psi(\omega)|^2\big|_{\omega=0} = 0$$

This means that wavelets must have a band-pass like spectrum [Valens, 2004].

2. Zero mean
$$\int_{t=-\infty}^{+\infty} \psi(t)dt = 0$$
This requirement implies that $\psi(t)$ must be oscillatory. In other words, $\psi(t)$ must be a wave [Valens, 2004].

3. Square norm one
$$\int_{t=-\infty}^{+\infty} |\psi(t)|^2 dt = 1$$
This implies that the energy of a wavelet is always normalized to one.

3-3-2 Wavelet Family

Given a mother wavelet $\psi(t)$ we can generate a wavelet family that consists of one mother wavelet and its child wavelets. A set of basis functions(child wavelets) $\psi_{a,b}(t)$ are generated by shifting and scaling the mother wavelet which can be expressed by

$$\psi_{a,b}(t) = \frac{1}{\sqrt{a}}\psi(\frac{t-b}{a}), \tag{3-3}$$

where a is the scale factor, b is the translation factor and the factor $a^{-1/2}$ is for energy normalization across the different scales. An example of a mother wavelet known as Shannon wavelet is given below:

$$\psi_{shan}(t) = 2sinc(2t) - sinc(t). \tag{3-4}$$

It is a substraction of two sinc functions. Actually, $sinc(t)$ is called the scaling function of mother wavelet $\psi_{shan}(t)$. We will discuss this later. The shape of $\psi_{shan}(t)$ is shown in Figure 3-2. More shapes of other mother wavelets are shown in Figure 3-3

3-3-3 Scaling of a Wavelet

The effects of scaling a wavelet $\psi(t)$ or other functions is compressing or expanding their shapes as is shown by Figure 3-4. From Fourier theory we know that compression in time is equivalent to stretching the spectrum and vise versa. Let $\mathcal{F}$ stand for the Fourier transform operator and let $F(\omega)$ be the Fourier transform of $f(t)$. The effects of scaling on the Fourier transform of the signal can then be expressed as

$$\mathcal{F}\{f(\frac{t}{a})\} = |a|F(a\omega) \tag{3-5}$$

Using this insight we can generate child wavelets with different frequency spectra.

Next, an example based on the Shannon mother wavelet in Equation 3-4 will be given to illustrate this procedure. Figure 3-5 depicts the power spectrum density of $\psi_{shan}(t)$ estimated by Fast Fourier Transform (FFT). The signal $\psi_{shan}(t)$ is sampled at 10 Hz with 200 data points.

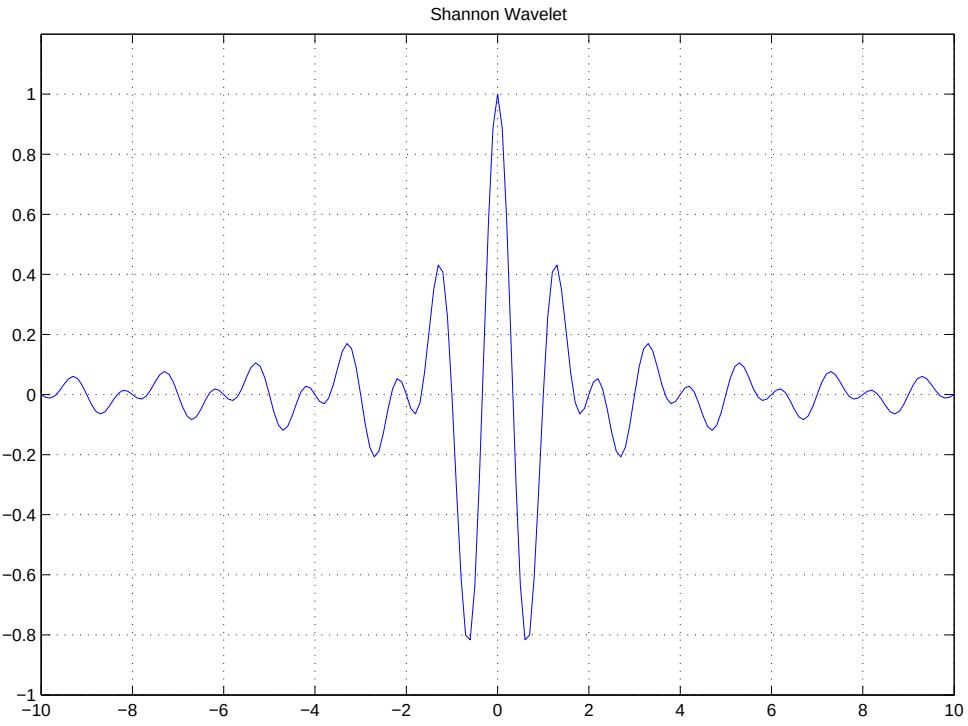

Figure 3-2: An example mother wavelet: a Shannon mother wavelet

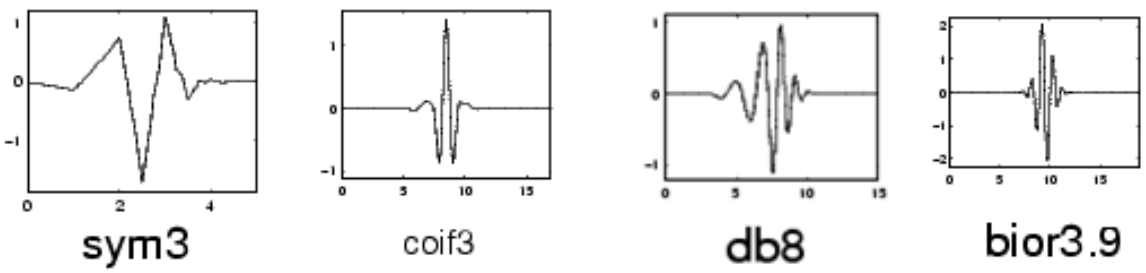

Figure 3-3: Four different shapes of mother wavelets [Taswell, 1995]

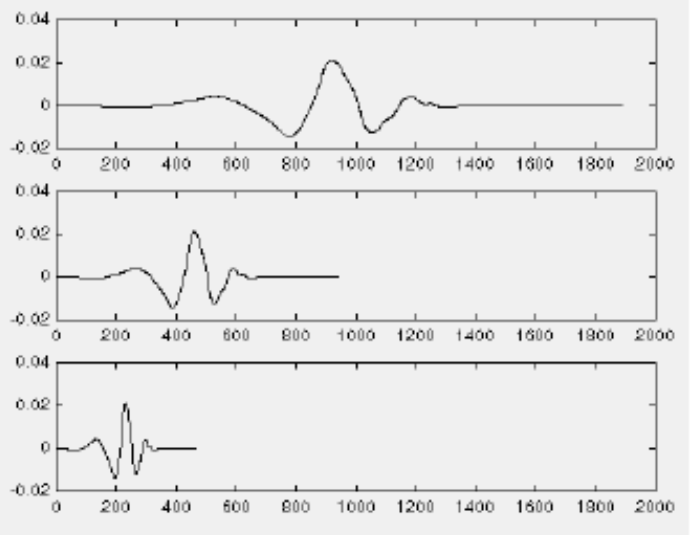

Figure 3-4: Scaling of a wavelet [Taswell, 1995]. The shape of wavelet is scaled in the time domain.

From the figure we can see that $\psi_{shan}(t)$ covers the frequency band $[0.5, 1]$ Hz. Now the Shannon mother wavelet will be scaled at $a = \{0.5, 2, 3, 4\}$ with fixed $b = 0$. With a, b determined,child wavelets $\psi_{0.5,0}(t)$,$\psi_{2,0}(t)$,$\psi_{3,0}(t)$,$\psi_{4,0}(t)$ are generated by Equation 3-3. Together with that of mother wavelet $\psi_{1,0}(t)$, the power spectrum density of the five wavelets are shown in Figure 3-6. It can be seen that the frequency bands of child wavelets are scaled version of the frequency band of the mother wavelet. The five wavelets cover the frequency band $[0.125, 2]$. What worth mentioning is that there is an overlap in the frequency band in Figure 3-6. Without $\psi_{3,0}(t)$, The four wavelets still cover the frequency band $[0.125, 2]Hz$.

Now imagine that we have a signal with limited frequency bands, then we can cover the spectrum of the signal by a wavelet family with a proper chosen set of scaling factors a. This is illustrated in Figure 3-7.

3-3-4 Shifting of a Wavelet

Shifting a wavelet simply means delaying (or hastening) its onset. Mathematically, delaying a function $f(t)$ by k is represented by $f(t - k)$. As is discussed in Section 3-3-3 the coverage of the frequency spectrum is done by scaling wavelets. In the same way, the signal can be covered in the time domain by shifting wavelets by choosing different translation factors b. For example, we have a signal of certain length and a wavelet $\psi_{a,b}(t)$ of a shorter length as shown in Figure 3-9. First b is set to 0 when the wavelet $\psi_{a,b}(t)$ locates at the start of the signal. Then we increase b with fixed step Δb until $\psi_{a,b}(t)$ moves to the end of the signal where $b = B$. If the signal is an analog signal, the fixed step $\Delta b \to 0$. If the analog signal is discretely sampled, the fixed step Δb is usually determined by the sampling frequency of the signal. Hence, the signal can be covered completely in the time domain. .

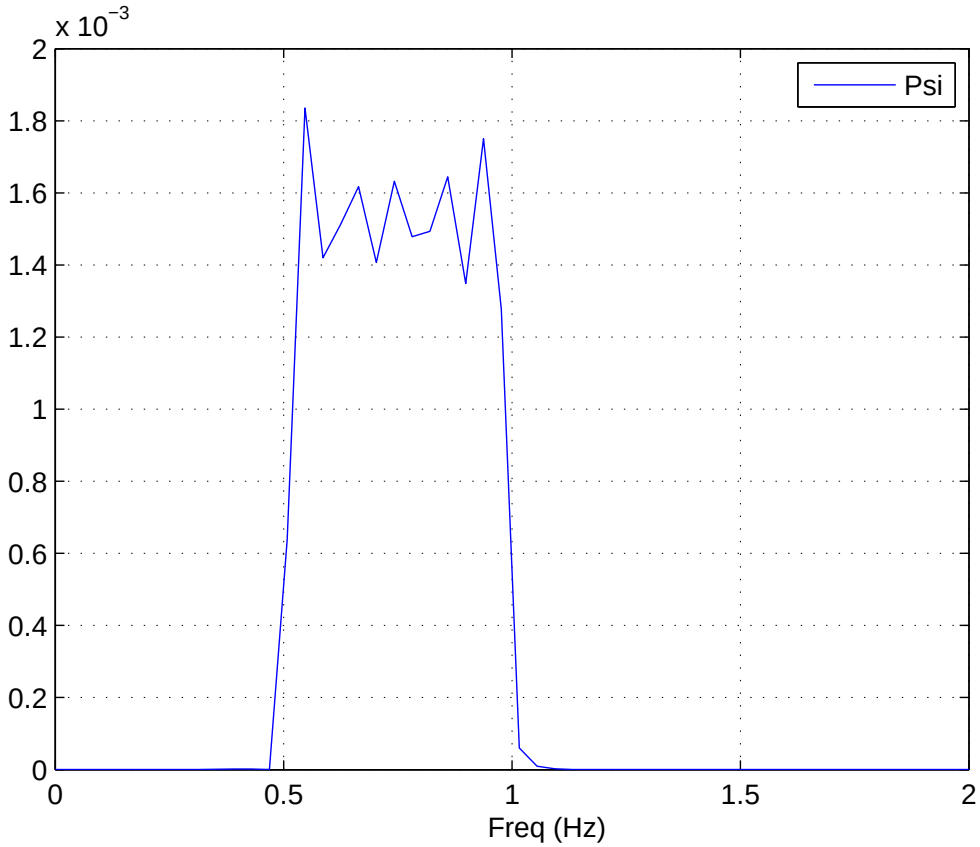

Figure 3-5: Estimation of the power spectrum density of $\psi_{shan}(t)|_{t=-10:0.1:10}$ The spectrum has a band of [0.5,1].

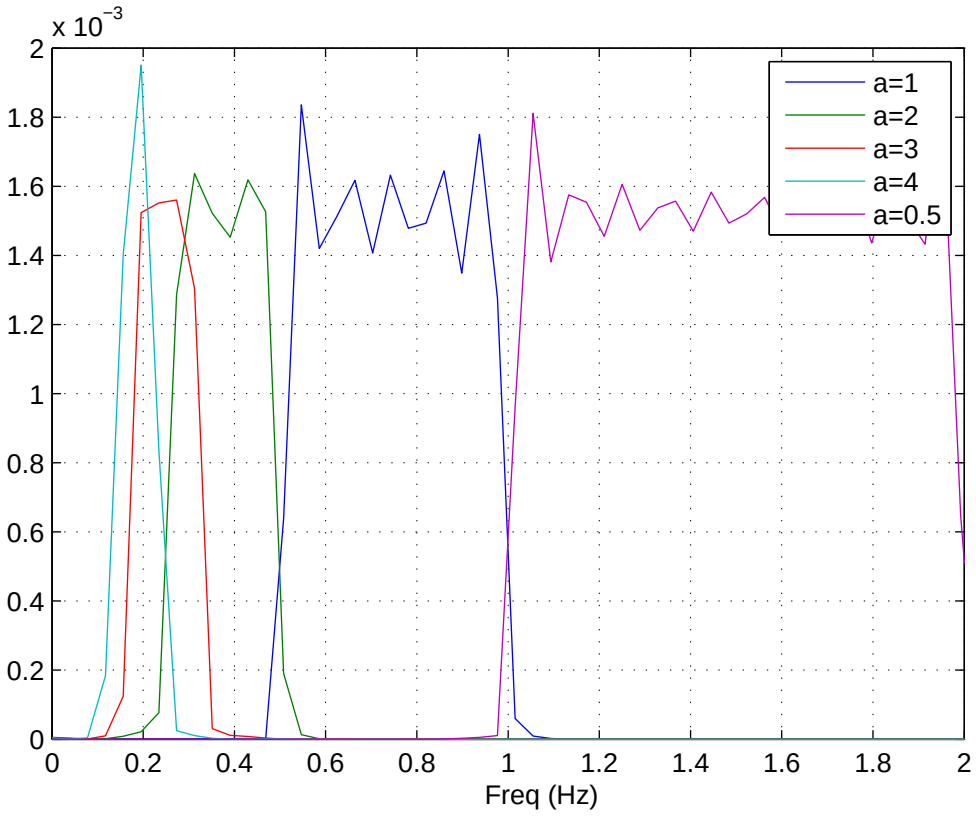

Figure 3-6: Power spectrum density of five shannon wavelets. Frequency bands of child wavelets are the scaled version of frequency band of mother wavelet. The gaps between two nearby spectra and the oscillations at the top of each spectra are caused by the spectrum estimation method. Ideally, they do not exist.

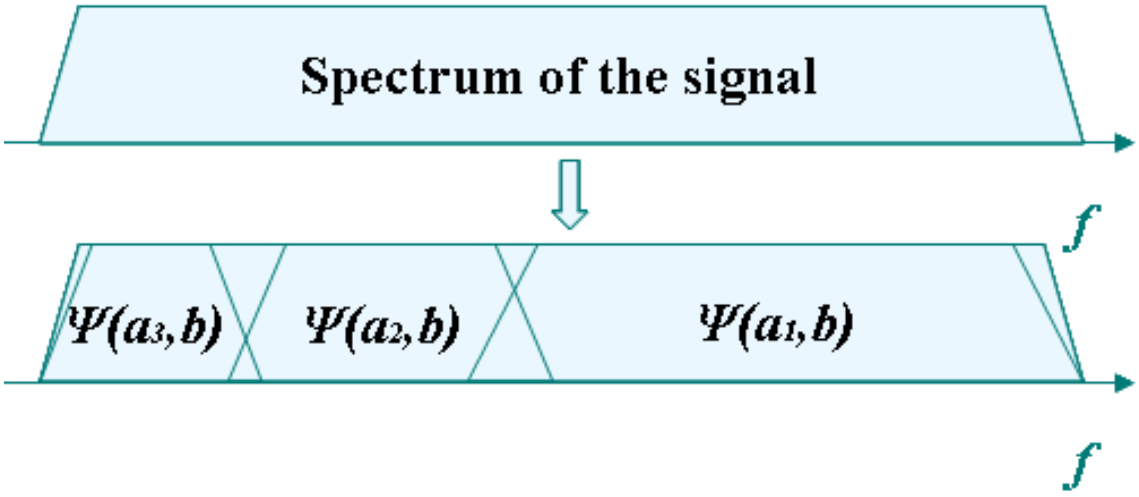

Figure 3-7: A sketch of the coverage of signal spectrum by child wavelets

Figure 3-8: shifting of a wavelet[Taswell, 1995]. The wavelet is shifted in the time domain.

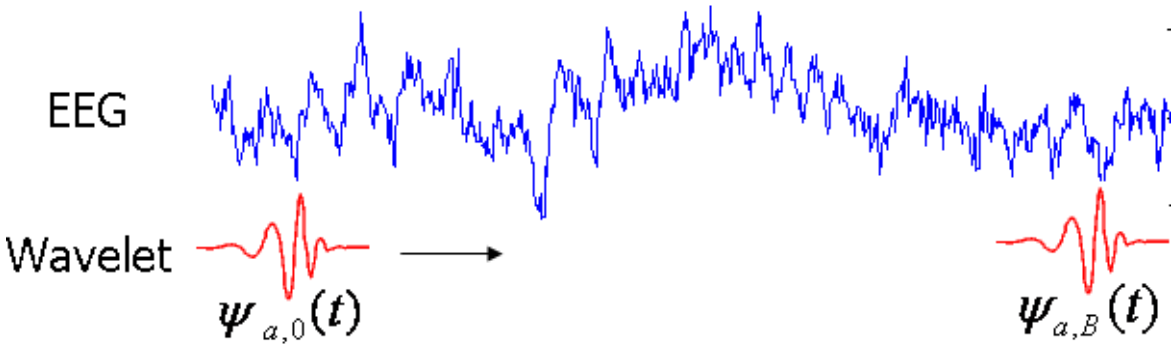

Figure 3-9: A sketch of the coverage of a signal in time domain. The wavelet is shifted from the beginning of the signal to the end by changing the parameter b.

3-4 Wavelet Transform

Wavelet transform is employed to process the EEG signals. Different with the general signal $x(t)$ for introductory use mentioned in the former sections, the signal $x(t)$ in the later sections refers in particular to the discrete EEG signal sampled at 128 Hz.

3-4-1 Classical Wavelet Transform

An EEG signal vector $x(t)$ with length N can be represented by linear combinations of the wavelet coefficients $C(a,b)$ with a set of wavelets $\psi_{a,b}(t)$. The coefficients $C(a,b)$ are obtained by the wavelet transform, which is defined as

$$C(a,b) = \sum_{t=1}^{N} x(t)\psi_{a,b}(t) = \frac{1}{\sqrt{a}} \sum_{t=1}^{N} x(t)\psi(\frac{t-b}{a}). \tag{3-6}$$

$C(a,b)$ is the wavelet coefficient with $a > 0,\ a \in R, b = 1, 2, ..., N$, where R stands for the set of real numbers. The variables a and b, scale and translation, are the new dimensions after the wavelet transform. An illustration of this transform is shown in Figure 3-10. In the figure, different scaled wavelets are shown in the vertical direction and different shifted wavelets are shown in the horizon direction. Each decomposition coefficient $C(a_i, b_i)$ is related to $\psi_{a_i,b_i}(t)$ and can be interpreted as how closely correlated the wavelet $\psi_{a_i,b_i}(t)$ is with the corresponding segment of the signal.

After the decomposition of signal $x(t)$ we get the coefficients $C(a,b)$. We can also use $C(a,b)$ to reconstruct the signal by the inverse wavelet transform. The inverse continuous wavelet transform is given by

$$x(t) = \sum_{b=1}^{N} \int_{a=0}^{+\infty} C(a,b)\psi_{a,b}(t)da \tag{3-7}$$

3-4-2 Stationary Wavelet Transform (SWT)

The classical wavelet transform suffers from two problems:

1. The frequency bands of a set of wavelets $\psi_{a,b}(t)$ overlap with each other if $a \in R$. For instance, Figure 3-6 shows the overlap of the frequency bands of three wavelets: $\psi_{2,0}(t)$,$\psi_{3,0}(t)$ and $\psi_{4,0}(t)$. Without $\psi_{3,0}(t)$, the spectrum is still covered.
2. If the frequency band of a signal starts from 0 Hz, an infinite number of wavelets have to be used to completely cover the frequency band of the signal. This is because the frequency band of a wavelet is always larger than 0. Given a wavelet $\psi_{a,b}(t)$, when $a \to +\infty$, the lower band of the frequency band of $\psi_{a,b}(t) \to 0$.

In the application of the wavelet transform, the problems listed above causes huge amount of data points and the decrease of the computational speed. The stationary wavelet transform, a discrete wavelet transform based on Mallat's Multiresolution Analysis (MRA) gives the solution to the two problems. More information about MRA and discrete wavelet transform can be found in [Mallat, 1989].

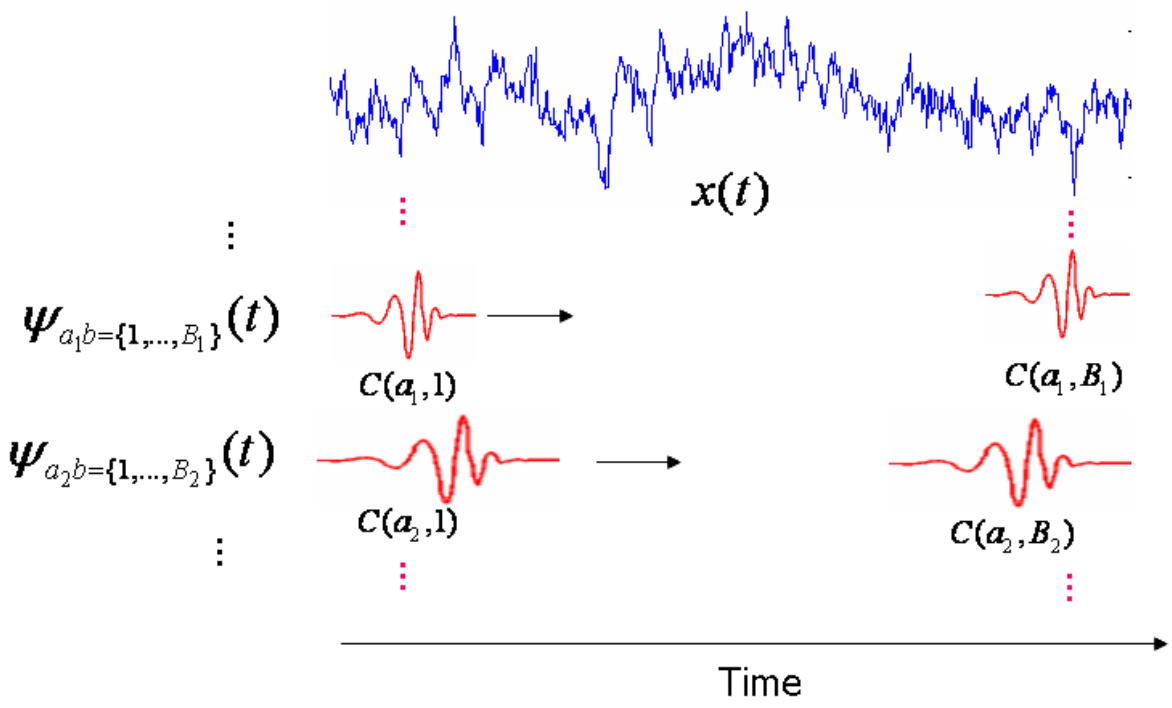

Figure 3-10: A sketch of the wavelet transform. Different scaled wavelets with the parameter a are shifted from the start to the end of the signal by changing the parameter b. For each (a, b), the coefficient $C(a, b)$ is computed.

Discretization

The solution to the first problem is the discretization of the wavelet parameter a namely

$$a = 2^j \ , \ j \in Z^+$$

Z is the set of integer numbers. As EEG signals are recorded discretely, the parameter b is already discretilized according the sampling frequency of the EEG signals namely

$$b = k \ , \ k \in Z^+$$

The fixed step Δb mentioned in Section 3-8 equals to the sampling interval of the EEG signal: 1/128 second. If an EEG segment has N data points, the corresponding parameter b covers all the signal value from 0 to $N-1$.

$$b = k, \ k = 1, 2, ..., N$$

Now we can rewrite Equation 3-3 in the discrete form namely

$$\psi_{j,k}(t) = 2^{-\frac{1}{2j}} \psi(\frac{t-k}{2^j}). \tag{3-8}$$

Scaling Function

The solution to the second problem is simply not to try to cover the spectrum all the way down to zero with wavelet spectra, but to use the spectra of another function to cover the respectively low frequency spectrum.This function is called the scaling function $\phi(t)$. The scaling function $\phi(t)$ is similar to the wavelet function $\psi(t)$. A discrete scaling function family is expressed by

$$\phi_{j,k}(t) = 2^{-\frac{1}{2j}} \phi(\frac{t-k}{2^j}) \tag{3-9}$$

An example of a scaling function is

$$\phi(t) = sinc(t).$$

Actually, $sinc(t)$ is the basis element of a shannon wavelet function in Equation 3-4. This is not a coincidence but it is a quite general relationship between wavelets and scaling function. The estimated power spectrum density of $\phi(t) = sinc(t)$ is shown in Figure 3-11. It is clear that $\phi(t) = sinc(t)$ has a low bandwidth from 0 Hz to 0.5 Hz.

Stationary Wavelet Transform

We can use finite scaled wavelets with $j = 1, 2, 3, ..., J$ to cover the high frequency band of the signal and use one scaling function with $j = J$ to cover the rest frequency band which is the low frequency band of the signal. Therefore, a discrete signal $x(t)$ can be represented by

$$x(t) = \sum_{j=1}^{J} \sum_{k=1}^{N} cD_j(k)\psi_{j,k}(t) + \sum_{k=1}^{N} cA_J(k)\phi_{J,k}(t) \tag{3-10}$$

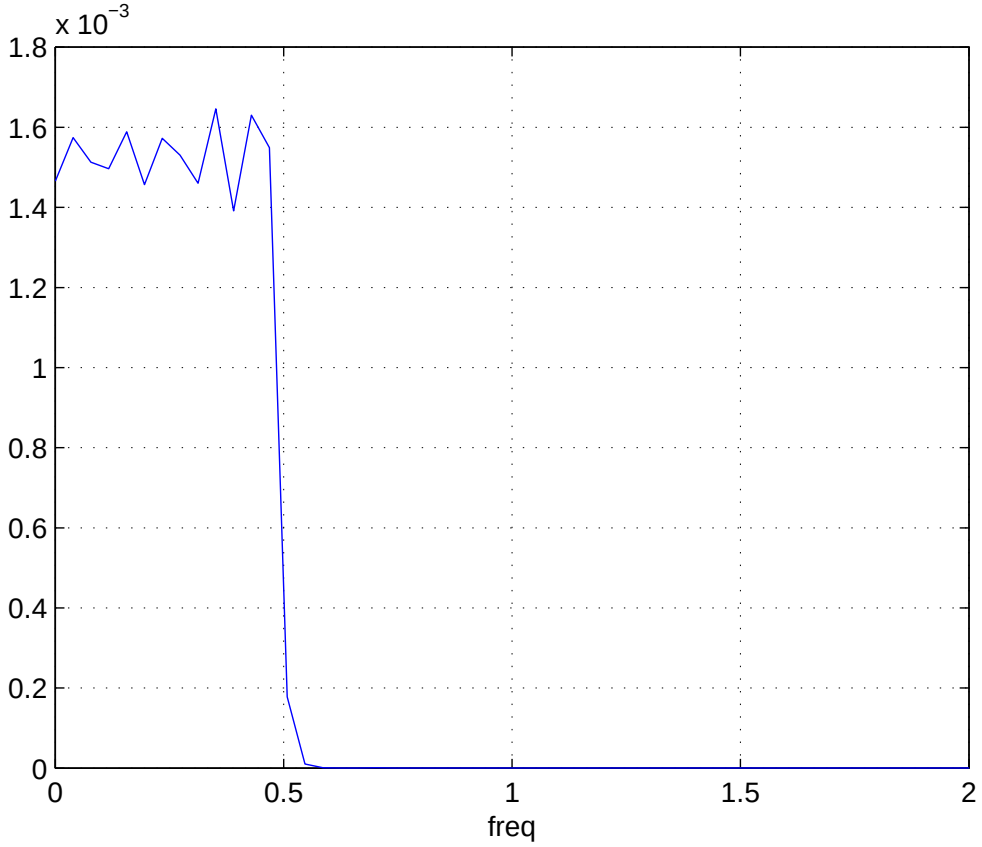

Figure 3-11: PSD of scaling function $\phi(t) = sinc(t)$. The spectrum starts from 0.

$cD_j(k)$ and $cA_J(k)$ are the stationary wavelet coefficients. They are obtained by the stationary wavelet transform which is defined as

$$\begin{cases} cD_j(k) = \sum_{t=1}^{N} x(t)\psi_{j,k}(t) \\ cA_J(k) = \sum_{t=1}^{N} x(t)\phi_{J,k}(t) \end{cases} \tag{3-11}$$

$$C(j,k) = \{cD_1(k), cD_2(k), ..., cD_J(k), cA_J(k)\} \quad \text{Where } j = 1, 2, ..., J$$

$C(j,k)$ can be regarded as the SWT wavelet coefficient matrix with size $(J+1) \times N$ which consists of two submatrixs $cD_j(k)$ $j = 1, ..., J$ with size $J \times N$ and $cA_J(k)$ with size $1 \times N$. $cD_j(k)$ is obtained by the convolution between the signal and the wavelet function $\psi(t)$. $cA_J(k)$ is obtained by convolution between the signal and the scaling function $\phi(t)$. The classical wavelet coefficients have become

$$C(a,b) \longrightarrow C(j,k) = [cD_j(k), cA_J(k)]$$

$$j = 1, 2, 3, ..., J$$

$$k = 1, 2, 3, ..., N$$

.

The actual computation of the coefficient $C(j,k)$ in practice is illustrated in Figure 3-12. More detailed theory and equations about this computation method can be found in [Nason and Silverman, 1995]. As is seen in the figure, an EEG signal $x(t)$ is decomposed by the stationary wavelet transform. The coefficients are obtained level by level. j represents the decomposition level. In level $j = 1$, the signal $x(t)$ is decomposed into two sets of coefficient vectors cD_1 and cA_1. In level $j = 2$, cA_1 is decomposed as the signal $x(t)$. Then we obtain the coefficient vectors cD_2 and cA_2. Finally in level $j = 3$, we get cD_3 and cA_3. Therefore, the coefficient matrix $C(j,k)$ is

$$C(j,k) = \{cD_1(k), cD_2(k), cD_3(k), cA_3(k)\}$$

If we look at the decomposition in the frequency domain in Figure 3-13, the bandwidth of signal is halved for each level. and it could also be found that coefficients cD_j represents the high frequency components of the signal and coefficients cA_J represents the low frequency components of the signal. This is the reason we call cD_j detail coefficients and cA_J approximation coefficients with the abbreviation cD and cA respectively.

Inverse Stationary Wavelet Transform

The inverse stationary transform is defined as

$$x(t) = \sum_{j=1}^{J}\sum_{k=1}^{N} cD_j(k)\psi_{j,k}(t) + \sum_{k=1}^{N} cA_J(k)\phi_{J,k}(t) \tag{3-12}$$

The EEG signal $x(t)$ is reconstructed with SWT wavelet coefficients. We can just inverse the arrows in Figure 3-12 to see the effect of the reconstruction.

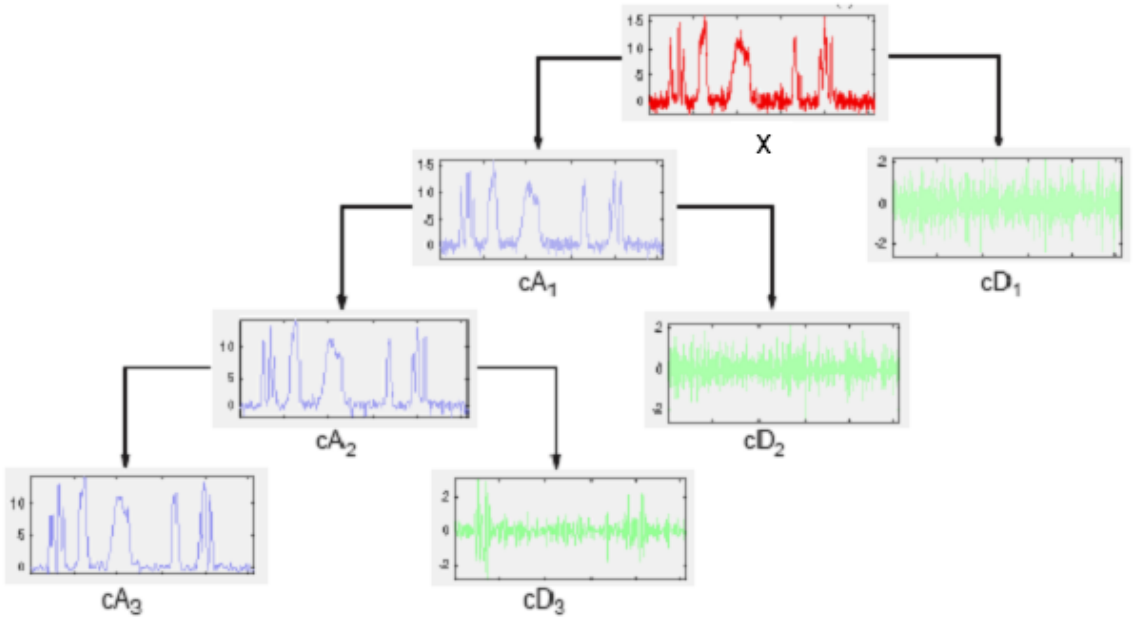

Figure 3-12: A three level SWT decomposition of signal $x(t)$. In level $j = 1$, the signal S is decomposed into the coefficients cD_1 and cA_1. In level $j = 2$, cA_1 is decomposed as the signal $x(t)$.

3-5 Implementation of SWT on Artifact Correction

Stationary wavelet transform (SWT) will be employed in EEG artifact correction. The general idea is, instead of operating an original EEG signal, we process the corresponding wavelet coefficients from the original EEG signal.

The sampling frequency of the EEG signal is 128 Hz. Each segment of the windowed EEG signal is 3 seconds and the segment is updated once every second. The whole algorithm can be divided into two procedures with a baseline measurement and an actual measurment.

1. Baseline measurement
   The aim of this measurement is to acquire prior statistical knowledge of wavelet coefficients from clean EEG signals. The variance and the mean of the coefficients $C(j, k)$ are estimated over k. Section 3-6-2 will give the detailed estimation methods.

   During this measurement, the subject should prevent any body movements to provide an relatively artifacts-free EEG signal. The procedure can be illustrated in Figure 3-14

2. Actual measurement
   This measurement is the real EEG signal measure procedure. The artifacts affected EEG signal will be corrected in this procedure. Figure 3-15 gives an illustration.

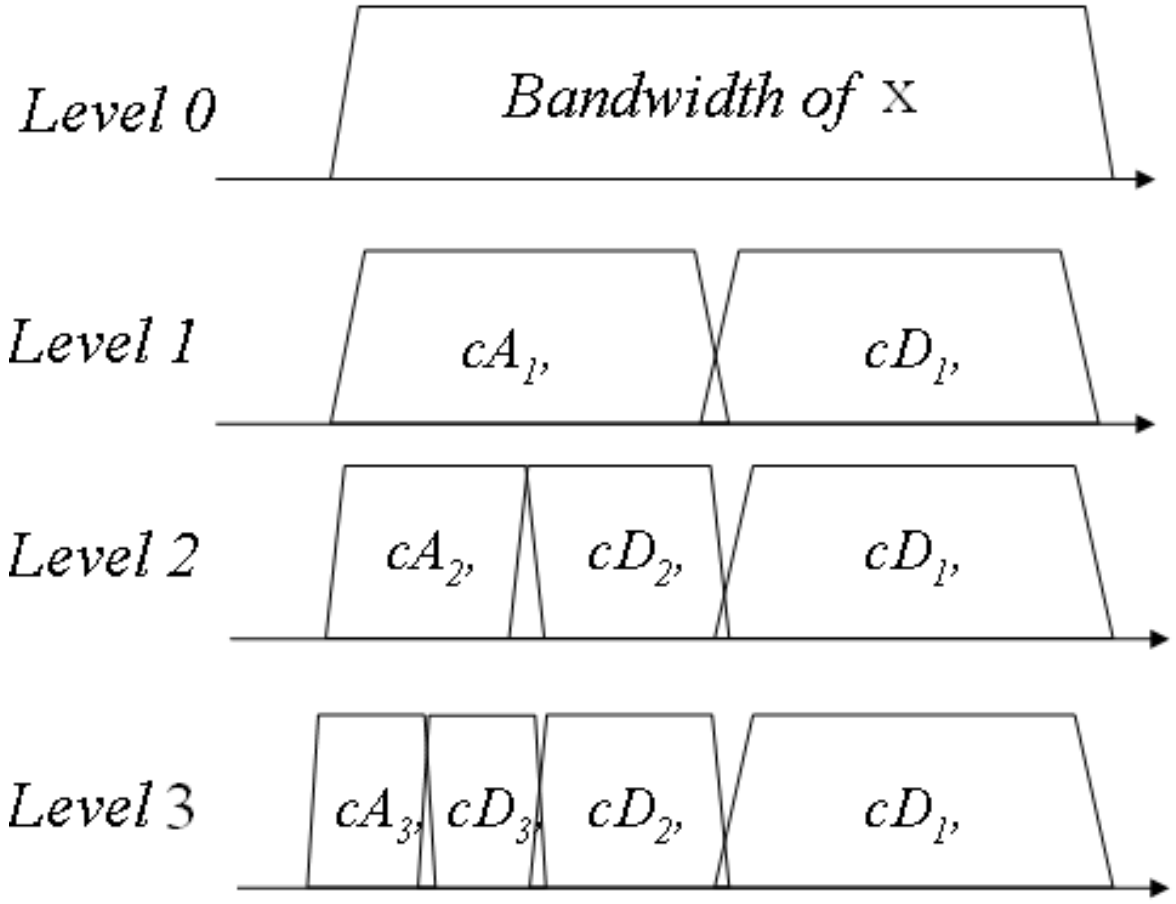

Figure 3-13: A three level SWT decomposition of signal S in the frequency domain

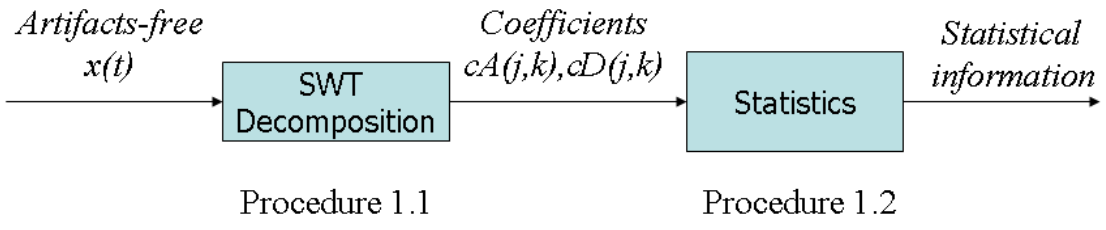

Figure 3-14: Baseline measurement. Prior statistical knowledge of wavelet coefficients from clean EEG signals are acquired.

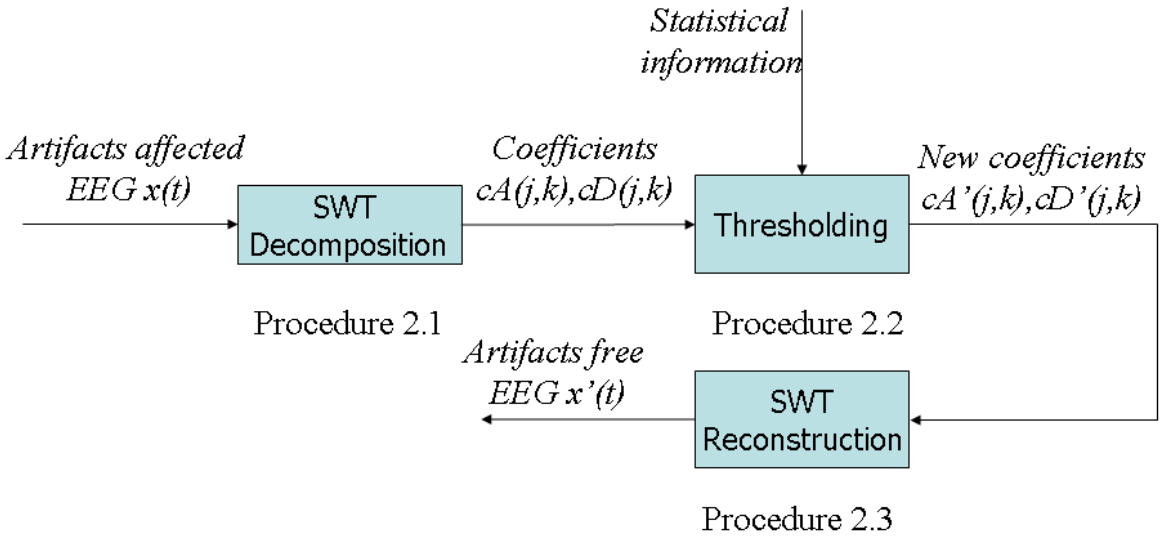

Figure 3-15: Procedure of actual measurement. Artifacts affected EEG signal are corrected in this procedure.

3-5-1 Aim of the Algorithm

The aim of the artifacts correction algorithm is to remove the artifacts components from EEG signals while keeping as many EEG components as possible. Therefore, the key step is the thresholding procedure 2.2 in Figure 3-15.

3-5-2 Decomposition

Procedure 1.1 and 2.1 is the SWT decomposition step. Given a three-second EEG signal $x(t)$ vector sampled at 128Hz, we know the length of $x(t)$ is $128 * 3 = 384$ and $x(t)$ contains frequency information from 0-64Hz. A level $j = 5$ SWT decomposition is applied to $x(t)$, we will get the wavelet coefficient matrix which is made up of 6 coefficient vectors

$$C(j,k) = \{cA_{j=5}(k); cD_{j=5}(k); cD_{j=4}(k); cD_{j=3}(k); cD_{j=2}(k); cD_{j=1}(k)\}$$

where $k = 1, 2, ..., 384$, so the coefficient sequences share the same length of 384 with the EEG signal $x(t)$. $C(j,k)$ has thus the size of 6*384. Each coefficient sequence also represents different frequency bands of $x(t)$ as is shown in Figure 3-16.

Considering the frequency bands of alpha waves and three types of artifacts, decomposition level 5 is already sufficient enough to distinguish the high frequency artifacts and the low frequency artifacts. There is no need to apply a higher level decomposition. The reason why not perform a 4 level decomposition is that we want to remove the electrical trend of EEG signal located at 0-2Hz to get a better plotting quality. An example of a 3-second EEG signal and its wavelet coefficients is depicted in Figure 3-17. Between time 0.5-1 second in Figure 3-17 we can see a clear eye blink from $x(t)$ which introduces an amplitude peak. It can also be seen in Figure 3-17 that higher level decomposition coefficients cA_5, cD_5, cD_4, cD_3 are larger than those without blink artifact while lower level decomposition coefficients are not affected.

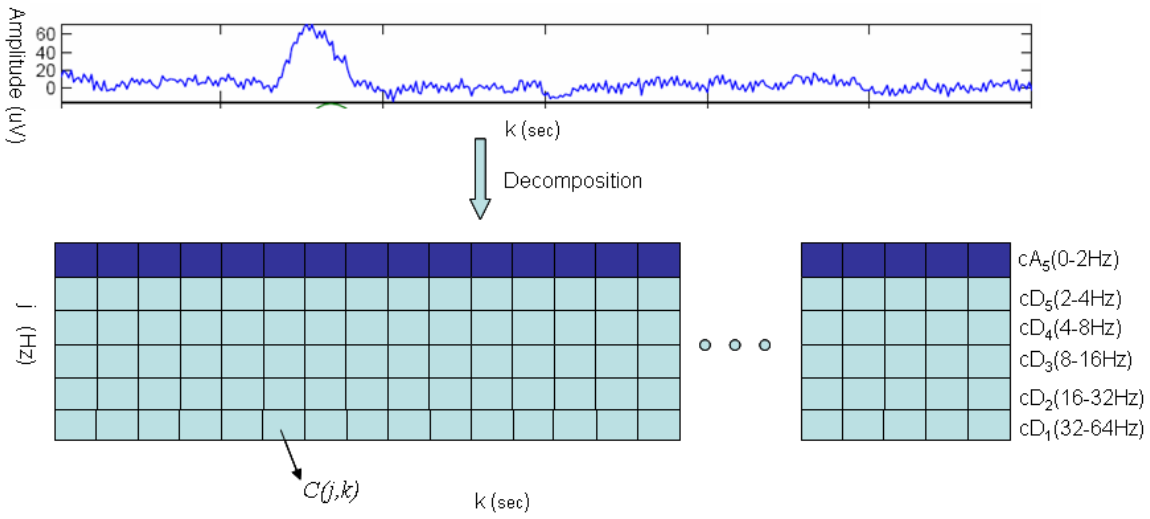

Figure 3-16: 5 level wavelet decomposition of $x(t)$. The signals in the time domain is decomposed to the coefficients in the time-frequency domain. Each block stands for one coefficient with the parameter (j, k). From the top to bottom, coefficients represent the low frequency components to high frequency components.

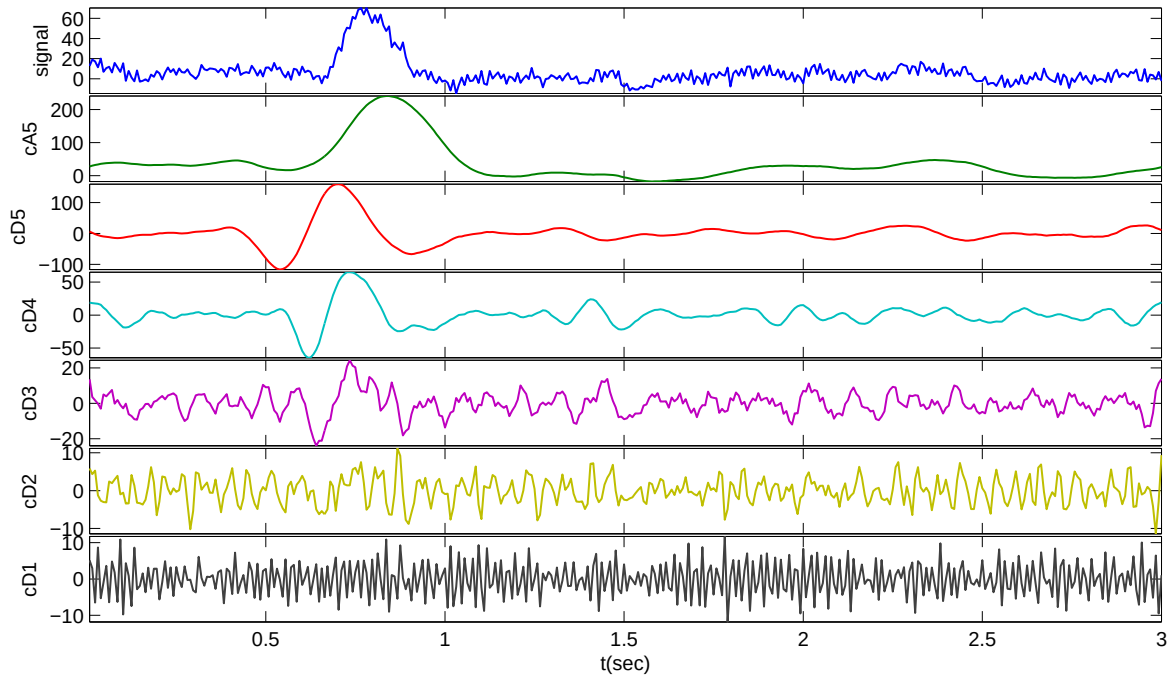

Figure 3-17: The signal $x(t)$ is decomposed by SWT using the wavelet 'db2'. $x(t)$ and six wavelet coefficients sequences share the same length. Example: An artifact of an eye blink contaminates the signal as well as the coefficients during the time interval [0.5 1] second.

3-5-3 Reconstruction

As is shown in Figure 3-15, after passing through the threshold (Procedure 2.2), new decomposition coefficients will be relatively more artifact-free compared with the original ones. We can reconstruct the new decomposition coefficient by Equation 3-12 to acquire cleaned EEG signal for the alpha wave measurement.

3-6 Thresholding

As is shown in the example in Figure 3-17,ocular artifact (OA) coefficients have larger amplitude than that of clean EEG coefficients. This is the same case for other types of artifacts (MA, EMA). Therefore, artifacts can be separated from EEG signals by thresholding the decomposition coefficients $C(j,k)$. Thresholding refers to Procedure 2.2 in Figure 3-15. If we zoom in this procedure, we see detailed steps as is shown in Figure 3-18. In this fig-

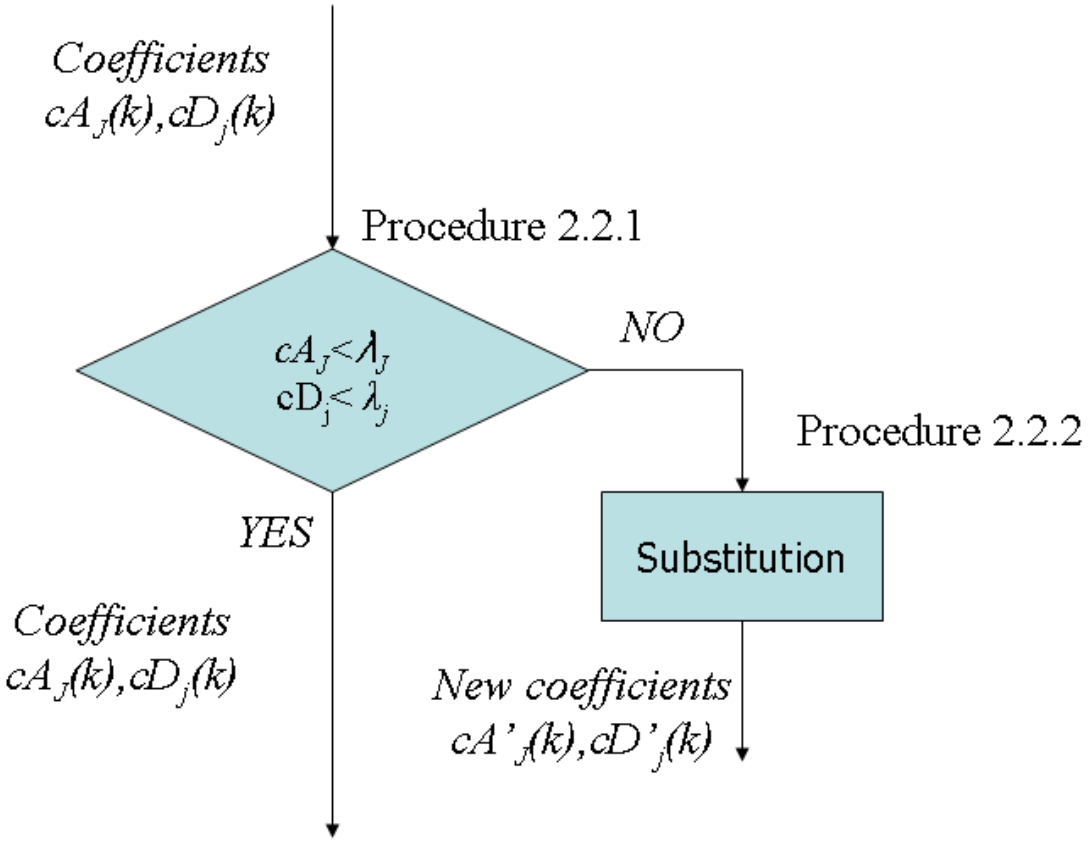

Figure 3-18: Detail steps of thresholding. If the coefficients are larger than the thresholds, the coefficients will be substituted. If the coefficients are smaller than the thresholds, they will be kept.

ure, Procedure 2.2.1 is to judge whether the coefficient $C(j,k)$ is larger than threshold λ_j. Coefficients $C(j,k)$ that do not exceed λ_j will be kept and those who do exceed λ_j will be

substituted by other values. Figure 3-19 shows an example. The green dash lines are the thresholds. Coefficient sequences cD_j and cA_J have different λ_j. Procedure 2.2.2 provides

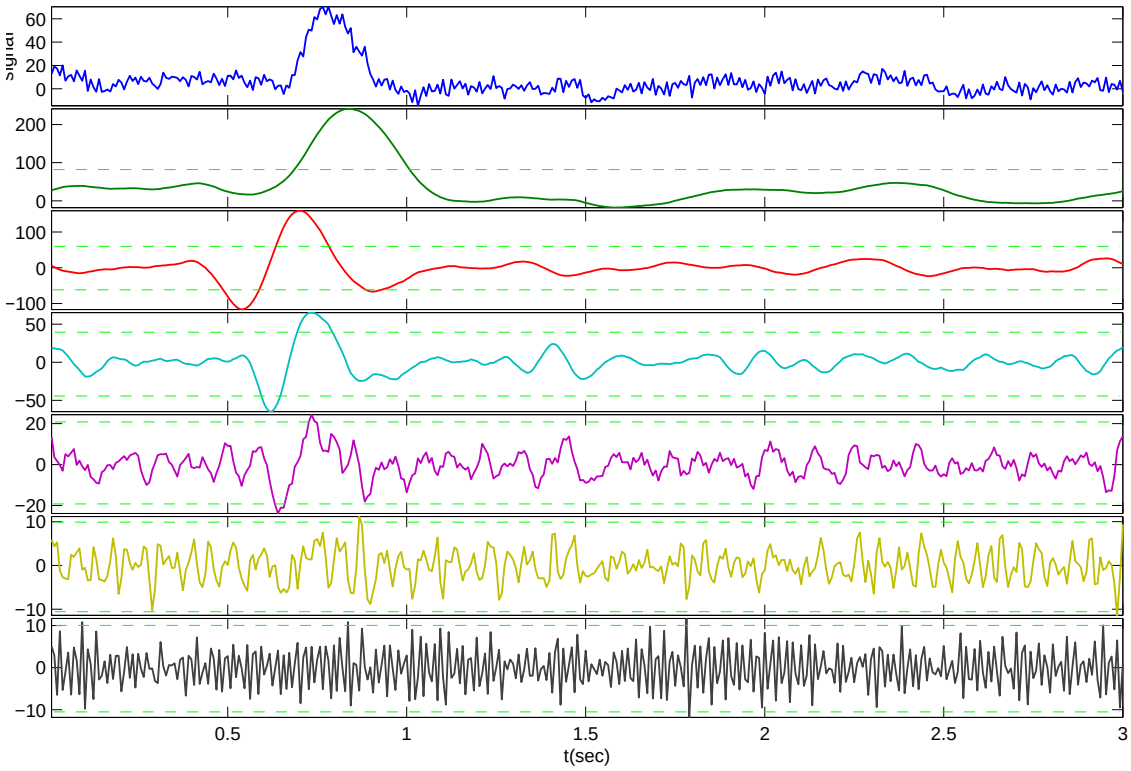

Figure 3-19: Thresholding the coefficients. The green dash lines are the thresholds. Threshold λ_j differs for different coefficient sequences cD_j and cA_J.

the substituted values that become the new coefficients $C'(j,k)$.

3-6-1 Choosing Proper Wavelet Families

Before performing the threshold procedure, we need to choose a proper wavelet family in Procedure 2.1 to maximize the amplitude of coefficients from artifacts compared with coefficients from clean EEG signals. This is crucial because ,with maximized amplitude of artifacts coefficients and normal amplitude of clean EEG coefficients, it is easier to determine a good threshold to distinguish artifacts from clean EEG. Hence the aim mentioned in subsection 3-5-1 can be fulfilled.

A maximal amplitude for coefficients for artifacts is achieved when the shape of the mother wavelet resembles the shape of the artifacts. Recall the wavelet transform equation 3-6, we can interpret this equation in the following way: the wavelet coefficients are acquired by convolving $x(t)$ with $\psi_{a,b}(t)$ of different scale and position. If the shape of $x(t)$ at certain position b is similar to the wavelet at certain scale a and the same position b, we will get

larger amplitude of coefficients $C(a, b)$. This also holds for the SWT. Therefore, the wavelet family whose shape in the time domain is similar to that of the artifacts will be chosen. In this thesis, different types of the artifacts have different shapes. It is imposable to choose one type of wavelet family that satisfied all the artifacts. We chose the wavelet family 'db2' whose shape is the most similar to that of the electrodes movement artifacts (EMA) as a compromise. Because EMA introduces the most noise to alpha waves. The shape of 'db2' is shown in Figure 3-20.

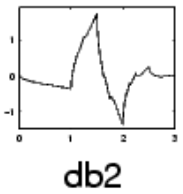

Figure 3-20: The shape of wavelet family 'db2' in the time domain. It resembles the shape of the electrodes movement artifacts.

3-6-2 Determination of the Threshold λ

We will get relatively artifacts-free EEG signals and coefficients during the baseline measurement. We use these coefficients to estimate the threshold λ_j at level j for the vectors cA_j and cD_j during the actual measurement.

We know that we will never get absolute artifact-free EEG signals to compute λ_j. In addition, an EEG signal is non-stationary and varies according to different subjects. Thresholds for coefficients of artifacts and clean EEG also change with time and subject. We propose the following protocol:

- Assume that the baseline measured EEG signal is clean enough to compute proper threshold values.
- Assume that the basic feature of EEG from one person does not change too swiftly within hours
- Assume that the EEG signal is stationary enough compared with artifacts
  It is true that the variance of the awake EEG signal can significantly change from one second to another. But this change is small compared with the variance of the EEG signal corrupted by artifacts.

Under the assumption given above, we will compute the threshold values based on the statistical information from baseline measurement. One of the information is the maximum absolute value of the coefficients $M_j(T)$ at level j at time T. $M_j(T)$ is defined as

$$M_j(T) = \max_{arg\ k} |C_T(j, k)|$$

where $C_T(j, k)$ stands for the coefficient matrix obtained from the EEG segment $x(T)$ in the baseline measurement.

The threshold λ_j for each level has been calculated as

Method 1 Mean+2std [Zikov et al., 2002]

$$\lambda_j = mean(M_j) + 2 \cdot std(M_j)$$

$mean(M_j)$ is defined as the average of M_j from N segments namely

$$mean(M_j) = \frac{1}{N} \sum_{T=1}^{N} M_j(T)$$

$std(M_j)$ is defined as the standard deviation of M_j from N segments namely

$$std(M_j) = \sqrt{\frac{1}{N} \sum_{T=1}^{N} (M_j(T) - mean(M_j)}$$

Where N is the duration of the baseline measurement.

Method 2 1.5std [Krishnaveni and Anitha, 2006]

$$\lambda_j = 1.5 \cdot std(M_j)$$

Method 3 Coifman [Coifman and Donoho, 1995]

$$\lambda_j = \sqrt{2\sigma^2 ln(N \cdot fs)}$$

fs stands for the sampling frequency. $fs = 128Hz$ in this thesis. σ^2 is the noise variance. Since the noise variance cannot be estimated from the contaminated EEG signal, it can be estimated by using the median absolute deviation given by $\frac{median(|cD_j|)}{0.6745}$, where cD_j are the detail coefficients at level j

Method 4 Quantile method (the proposed method)

$$\lambda_j = 2 \cdot quantile(|C_{all}(j,k)|, 0.9)$$

where

$$C_{all}(j,k) = \{C_1(j,k); C_2(j,k), ..., C_N(j,k)\}$$

The p-quantile of the distribution of a vector X can be defined as the value x such that

$$x = P(X < x) \leq p$$

Methods 1-3 have been taken from literature and analyzed in a clinical environment. During the baseline computation, subjects are strictly controlled in order not to introduce artifacts into EEG signals. EEG signals with long-period-inevitable artifacts such as eye blink are manually removed from baseline EEG signals. However in a life style environment, subjects cannot be strictly controlled and it is impossible to perform manual artifacts rejection. For methods 1-3, it can be imagined that presence of a few artifacts in the baseline will influence the estimation of the threshold, to put it more precisely, artifacts affected thresholds will be larger than non-artifacts affected thresholds. The fourth method does not use extreme coefficients to estimate the threshold. Hence the estimation of the threshold will not be affected by a small number of artifacts in the baseline EEG signals.

3-6-3 Determine the Value of New Coefficients

This step refers to Procedure 2.2.2 –substitution. The question is how to deal with the coefficients that exceed the threshold. Current methods can be divided into two categories: hard substitution and soft substitution.

- Method 1 Hard substitution
  Let $(T)(\cdot, \lambda)$ denote the substitution operator with threshold λ. Hard substitution is defined as [Donoho, 1995]

$$\mathcal{T}(C(j,k),\lambda) = \begin{cases} C(j,k) & if|C(j,k)| < \lambda \\ 0 & otherwise \end{cases} \tag{3-13}$$

  Hard substitution will completely eliminate the coefficients larger than the threshold. The advantage is that it perform a complete removal of artifacts components. The drawback is that clean EEG components affected by artifacts are also removed. If there are too many artifacts, too much information of the EEG signal will be lost.

- Method 2 Soft substitution
  Soft substitution is to replace coefficients that exceed the threshold by non-zero values. A type of soft substitution from literature is defined as [Kumar et al., 2008]

$$\mathcal{T}(C(j,k),\lambda) = \begin{cases} C(j,k) & if|C(j,k)| < \lambda \\ sign(C(j,k))0.7 \cdot \lambda & otherwise \end{cases} \tag{3-14}$$

  This method restores the artifacts affected coefficients with fixed values to prevent a complete loss of clean EEG components. However, the constant value $sign(C)0.7 \cdot \lambda$ may still be a bad estimation for the original value.

- Method 3 (the proposed method)
  Soft substitution based on Kalman 1-step-ahead Auto Regressive (AR) parameter estimation
  The artifacts affected coefficients will not be substituted by the fixed values. Instead, we use non-constant values to substitute them. The non-constant values are estimated from the previous sets of coefficients. This estimation is done by Kalman 1-step-ahead estimator. Therefore, the error between the estimated coefficients and the original coefficient is minimized compared with the two methods mentioned above . Because the Kalman estimator requires parametric model of the signals, coefficients of clean EEG signals are assumed to be modeled by an adaptive AR model namely

$$C(j,k) = \sum_{i=1}^{p} \alpha_i(j,k)C(j,k-i) + \omega(j,k) \tag{3-15}$$

  where $\alpha_i(j,k)$ are the parameters of the model at time k, p is the model order. p is chosen as 6 in the thesis. $\omega(j,k)$ is the white noise. Hence the method 3 can be described as

$$\mathcal{T}(C(j,k),\lambda) = \begin{cases} C(j,k) & if|C(j,k)| < \lambda \\ \sum_{i=1}^{p} \alpha_i(j,k-1)C(j,k-i) & otherwise \end{cases} \tag{3-16}$$

  In the equation, AR model parameter $\alpha_i(j,k)$ is estimated from $\alpha_i(j,k-1)$ by a Kalman filter. The process can be described as follows:

– Innovation of residual between the current coefficient and the coefficient estimated from $(k-1)$

$$\varepsilon(j,k) = \mathbf{C}(j,k) - \hat{\alpha}_(j,k-1)^T \mathbf{C}(j,k-1)$$

– Innovation of residual covariance

$$v(j,k) = (1-a)v(j,k) + a \cdot \varepsilon(j,k)^2$$

– Computation of Kalman gain

$$\mathbf{K}(j,k) = \mathbf{A}(j,k-1)\mathbf{C}(j,k-1)/(\mathbf{C}(j,k-1)^T \mathbf{A}(j,k-1)\mathbf{C}(j,k-1) + v(j,k))$$

– Update of AR parameters

$$\hat{\alpha}(j,k) = \hat{\alpha}(j,k-1) + \mathbf{K}(j,k)\varepsilon(j,k)$$

– Update of the covariance matrix

$$\mathbf{A}(j,k) = \mathbf{A}(j,k-1) - (1+a)\mathbf{K}(j,k)\mathbf{C}(j,k)^T \mathbf{A}(j,k-1) + a^2\mathbf{I}$$

Where the parameter vector $\hat{\alpha}(\mathbf{j},\mathbf{k}) = [\hat{\alpha}_\mathbf{1}(\mathbf{j},\mathbf{k}); \cdots ; \hat{\alpha}_\mathbf{p}(\mathbf{j},\mathbf{k})]$ and the coefficient vector $\mathbf{C}(\mathbf{j},\mathbf{k}) = [\mathbf{C}(\mathbf{j},\mathbf{k}); \cdots ; \mathbf{C}(\mathbf{j}-\mathbf{k}+\mathbf{1})]$. $\mathbf{K}(\mathbf{j},\mathbf{k})$ is the Kalman gain and $\mathbf{A}(\mathbf{j},\mathbf{k})$ is the covariance matrix for the estimates of $\alpha(\mathbf{j},\mathbf{k})$. The initial values $\mathbf{A_0}$ and $\alpha_\mathbf{0}$ are an identity matrix $\mathbf{I_{p\times p}}$ and a zero vector respectively, of order p. a is the adaptive factor.

The illustration is shown in Figure 3-21 Together with the choice of a wavelet family for decomposition and methods on estimation of the threshold, three kinds of substitution methods will be tested in the next chapter.

3-7 Summary

EEG artifacts correction based on wavelet analysis has been discussed in this chapter. Wavelet analysis has more advantages in EEG artifacts correction than the other methods mentioned in the first section for life style measurement . Then wavelet theory was introduced in detail in the following sections. To implement the theory into practice with acceptable computational efficiency, the stationary wavelet transform, a modified version of Mallat's DWT , was employed in the artifacts correction approach. Afterwards, details about implementation of SWT on artifacts correction were discussed. Several threshold and substitution methods were compared theoretically. The actually performance of different threshold and substitution methods will be analyzed in the next chapter.

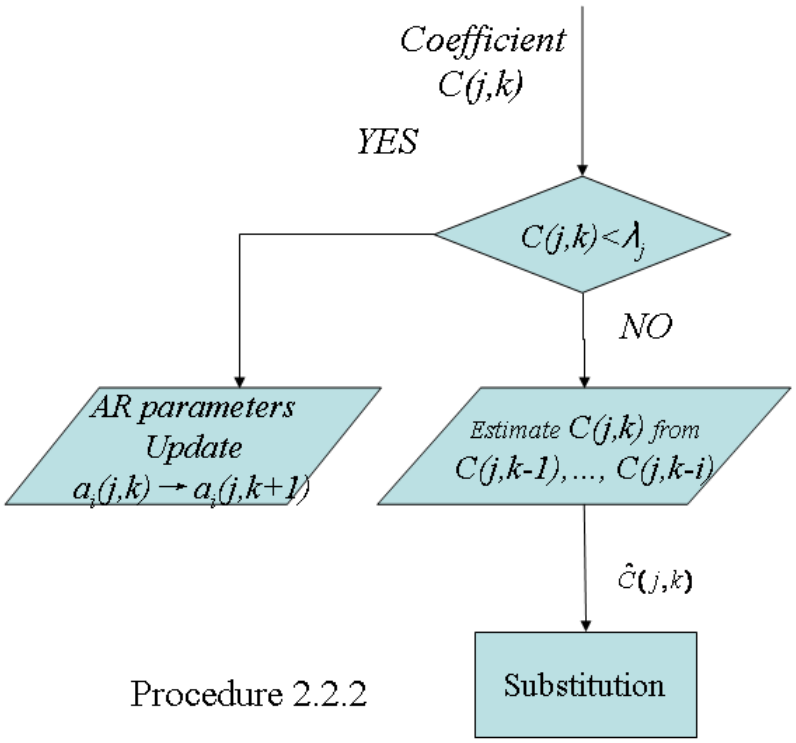

Figure 3-21: Illustration of the proposed method 3. If the coefficient $C(j,k)$ exceeds the threshold, it will be substituted by the estimated coefficient $\hat{C}(j,k)$ by kalman 1-step-ahead estimation. If the coefficient $C(j,k)$ is smaller than the threshold, it will be used to update the parameters of the AR model

Chapter 4

Simulation of Artifacts and Evaluation

In Chapter 3, four threshold computation methods and three decomposition coefficients substitution methods were introduced. Their performance on artifacts correction will be validated and compared in this chapter. The best methods will be employed in the artifact correction method and tested in the real time alpha wave measurement in Chapter 5.

4-1 Simulation of Artifacts Contaminated EEG Signals

4-1-1 Significance of the Simulation

When we remove the artifacts from artifacts contaminated EEG signals, we do not know how many artifacts are still left. We are also short of knowledge on how similar the modified EEG signals and the pure EEG signals look like. Thus it is hard to validate the performance of the artifacts correction algorithm without knowing the original pure EEG signals.

To solve this problem, simulations of artifacts contaminated EEG signals will be made by mixing artifacts with clean EEG signals. Once we perform the artifacts correction algorithm to the simulated EEG signals, we can compare the modified EEG signals yielded by the algorithm with the clean EEG signals to get a validation.

4-1-2 Assumption of the Simulation

We cannot create artifacts contaminated EEG signals without knowing how to mix artifacts with clean EEG signals. Although there are some studies on certain types of artifacts such as transfer of the EOG (ocular electric signals) activity into the EEG [Verleger et al., 1982], a totally accepted model of all kinds of artifacts is still missing. Therefore two assumptions are made to build the model of contaminated EEG signals.

- Assumption 1
  In this thesis, an assumption of a linear model of artifacts contaminated EEG signals is given as
  $$x_c(t) = x_p(t) + w_i \cdot n_i(t) \tag{4-1}$$
  where $x_c(t)$ refers to contaminated EEG signals, $x_p(t)$ refers to pure EEG signals, $n_i(t)$ is the i-th type of artifacts and w_i is the weight of $n_i(t)$.

- Assumption 2
  As pure EEG signals can not be acquired, hence relatively clean EEG signals with no obvious artifacts are regarded as pure EEG signals.

4-1-3 Realization of the Simulation

Based on the assumptions, we can generate artifacts contaminated EEG signals $x_c(t)$ by a linear combination of measured clean EEG signals and measured artifacts. As is introduced in Chapter 2, these types of artifacts can be distinguished as OA,EMA and MA. Therefore, three types of contaminated EEG signals will be generated by the same clean EEG signal with different artifacts. The parameters of clean EEG signals $x_p(t)$ shown in Figure 4-1 are given below

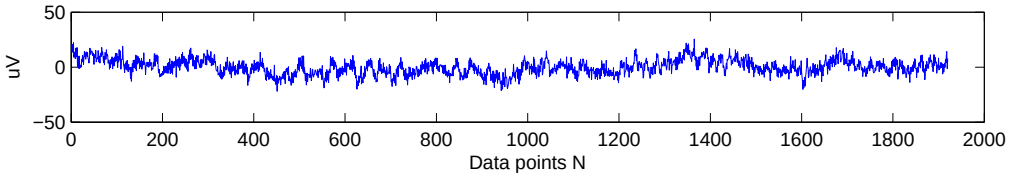

Figure 4-1: A set of relatively clean EEG signals are assumed to be the pure EEG signal $x_p(t)$

- Sampling frequency: 128Hz
- Length: 15 seconds
- Location: measured by electrodes C3

Now the generation of three type of artifacts is described below

1. Ocular artifacts (OA)contaminated EEG signals
   Ocular artifact signals, namely EOGs, are separately measured by electrodes near the eyes. Three blinks with different blinking speed are selected as the seeds of ocular artifacts. The weight w_{OA} is determined by comparing the amplitudes of blinks in EOG and similar blinks in EEG namely
   $$w = \frac{h_2}{h_1}$$
   where h_1 is the amplitude of blinks in EOG and h_2 is the amplitude of blinks in EEG which is shown in Figure 4-2. Hence the ocular artifacts contaminated EEG signals obtained by Equation 4-1 can be shown in Figure 4-3

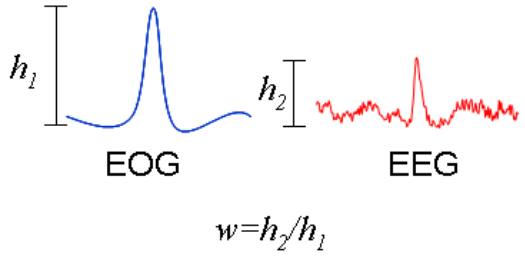

Figure 4-2: Determine of weight w_{OA} for ocular artifacts. The weight w_{OA} is determined by comparing the amplitudes of blinks in EOG and similar blinks in EEG

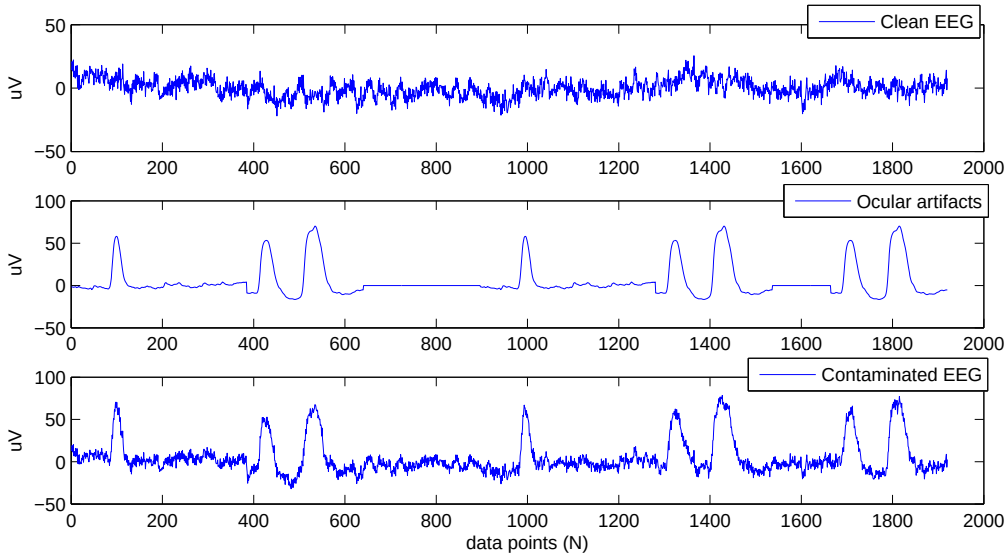

Figure 4-3: Generation of OA contaminated $x_{cOA}(t)$

2. Electrode movement artifacts (EMA) contaminated EEG signals
   Electrode movement artifacts are measured by slowly shaking and moving the electrodes far from the head. Electrical signals due to resistance change are recorded. Due to the distance between the electrode and brain, only a negligible amount of EEG signals are recorded. The weight w_{EMA} is determined in the same way in OA contaminated EEG signals. Hence the electrode movement artifacts contaminated EEG signal obtained from Equation 4-1 can be shown in Figure 4-4. We can see from the figure that EMA artifacts have really high amplitudes compared with EEG signals and the EEG signals are hardly recognizable in the presence of EMA.

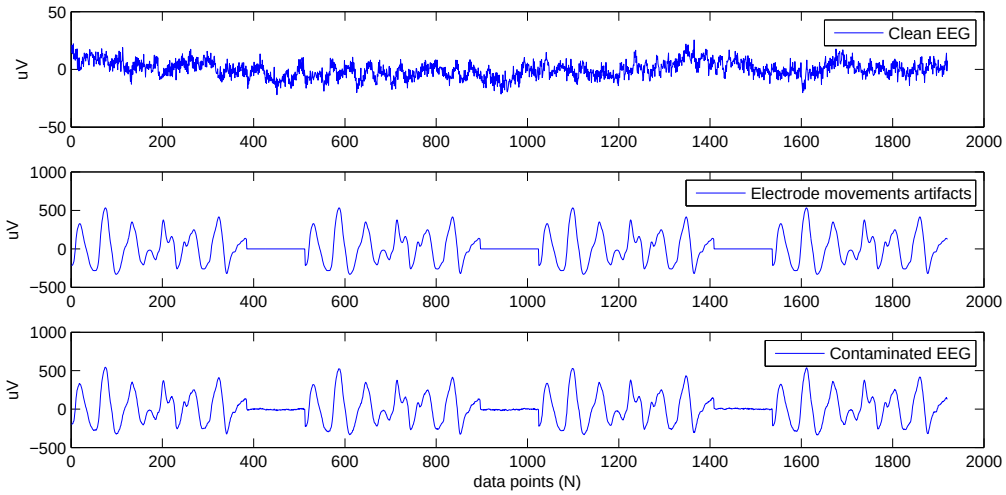

Figure 4-4: Generation of EMA contaminated EEG signal $x_{cEMA}(t)$

3. Muscle artifacts (MA) contaminated EEG signals
   Muscle artifacts, namely EMG signals, are separately measured by the electrodes near the jaw in which the muscle provides the most MA to EEG signals. When the subject is clenching his/her teeth, EMG signals are recorded. The weight w_{MA} is selected in the same way in OA contaminated EEG signals. Figure 4-5 depicted the generation of MA contaminated EEG signals. The EEG signal in $x_{cMA}(t)$ is also hardly recognizable in the presence of muscle artifacts in the time domain.

Given simulated contaminated EEG signals shown above, we can perform the validation of the artifacts correction algorithm with different thresholding and substitution methods.

4-2 Validation

4-2-1 Introduction

In this section, two detectors, $P^{Abs}(T)$ and $P^{Ratio}(T)$ introduced in Section 2-3-4, together with two artifact elimination methods, artifact correction and artifact rejection introduced in Section 2-4-4, will be tested with the simulated EEG signals and the artifacts.

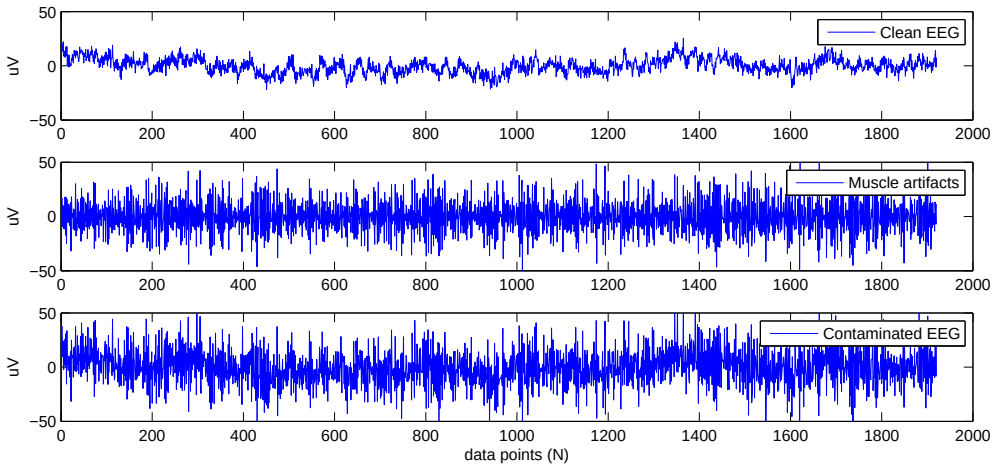

Figure 4-5: Generation of MA contaminated EEG signal $x_{cMA}(t)$

After that, we will perform further evaluation on the artifact correction method. In Chapter 3, four threshold computation methods and three substitution methods were introduced. Threshold computation methods:

1. mean+2std
2. 1.5std
3. Coifman
4. quantile

Substitution methods:

1. Hard substitution
2. Soft substitution
3. Soft substitution based on Kalman 1-step-ahead Auto Regressive (AR) parameter estimation

Combinations of each computation method and each substitution method forms an entire thresholding procedure (Procedure 2.2) which is shown in Figure 3-18. Each combination will be tested and compared to give a results to see which combination is the best in the artifact correction. The combination of the methods can be symbolled as Method(i,n) and M(i,n). i refers to the i-th computation method and n refers to the n-th substitution method.

4-2-2 Quality Criterion

Recall the quality criterion in Equation 2-3 in Chapter 2, we will compute the normalized mean absolute error (MAE) of the artifacts removed EEG alpha power $P_n(T)$ and the pure

EEG alpha power $P_p(T)$ namely

$$MAE| = \frac{1}{N}\sum_{T=1}^{N}\frac{|P_n(T) - P_p(T)|}{P_p(T)}$$

to see the alpha spectrum difference between pure EEG signals and artifacts removed EEG signals. N refers to the total numbers of analysis segments of the signal. Each segment is a 3-second-length EEG signal.

4-2-3 Artifacts Correction Validation

Two detectors, $P^{Abs}(T)$ and $P^{Ratio}(T)$, together with two artifact elimination methods, artifact correction and artifact rejection, are tested with the simulated EEG signals in the artifact correction validation. The combination of threshold computation and coefficient substitution method employed in the artifact correction is the method(4,3) which is the quantile method and the Kalman filter AR model method. The reason of choosing them is that both of them are the proposed methods in this thesis.

Method of artifact correction (corr), artifact rejection (rej) and no treatment (non) on artifact will be respectively combined with the two detectors. Therefore, six combinations will give six results. The result of each combination is shown in Table 4-1.

Table 4-1: MAE of two detectors with no artifact treatment (non), artifact correction (corr) and artifact rejection (rej) on three type of artifacts. The MAE values are the normalized difference between alpha power $P_n(T)$ and pure EEG alpha power$P_p(T)$ in percentage of $P_p(T)$. $P_n(T)$ is derived from contaminated EEG signals.

Method	OA (%)	MA (%)	EMA (%)	Average (%)
$P^{Abs}(T)$ non	24.7782	2.2284	14136.4326	4721.1464
$P^{Abs}(T)$ corr	24.2347	1.9952	34.0061	20.0266
$P^{Abs}(T)$ rej	16.1935	11.5301	2648.4392	892.0542
$P^{Ratio}(T)$ non	15.1018	78.0657	45.0599	46.0758
$P^{Ratio}(T)$ corr	10.8617	26.4531	33.9524	23.7557
$P^{Ratio}(T)$ rej	22.706	79.2853	28.7387	42.91

Firstly, the two detectors are compared. From the MAE results of $P^{Abs}(T)$ non, for the three example types of artifacts introduced in the previous section, EMA introduces the most disturbance (14136% larger than the pure EEG alpha wave) to alpha power . OA makes small changes on alpha power (24.7%) and the influence of MA is the smallest (2.2%). For the results of $P^{Ratio}(T)$ non, MA introduces the most disturbance to alpha power (78.1% ,the other two are 15.1% and 45.1%). The results support the content in section 2-4-3 in Chapter 2. If we look at the average , $P^{Ratio}(T)$ (46.1%)is much better than $P^{Abs}(T)$ (4721%).

Secondly, two artifact elimination methods are compared. Comparing $P^{Abs}(T)$ corr with $P^{Abs}(T)$ rej, artifact correction works for the three types of artifacts because the average influence caused by artifacts is decreased from 4721% to 20.1%. Artifact rejection (16.2%) did better in OA than Artifact correction (24.2%). However, for the results of MA, $P^{Abs}(T)rej$ (11.5%)has larger error than $P^{Abs}(T)non$ (2.2%). This infers that extra artifacts are introduced to the EEG signal by the artifact rejection. Compared $P^{Ratio}(T)$ corr with $P^{Ratio}(T)$

rej, artifact correction method is better for OA and MA. Artifact rejection method performs better for EMA. Furthermore, artifact rejection method with $P^{Ratio}(T)$ performs the best for EMA among all the methods (average 23.7%, the lowest of all).

As a general view and conclusion, $P^{Ratio}(T)$ is better than $P^{Abs}(T)$ if no artifact treatment is performed. Artifact rejection method trends to introduce artifacts when the frequency bands of artifacts (such as MA) is not or less overlapped by that of the alpha waves. Sometimes the influence from the rejection is even larger than the artifact itself. But for the artifacts overlapped by alpha waves, artifact rejection is effective. The average MAE for artifact correction method is the smallest which infers that artifact correction method can stably remove artifacts. However, artifact correction is not always better than artifact rejection, which means artifacts are not completely removed.

Now a deep investigation on the EEG signals and their spectrum is performed in order to see how it works by the combination of the detector $P^{Abs}(T)$ and the artifact correction. Figure 4-6 shows a segment of EMA contaminated signals and its spectrum. In the top plot,

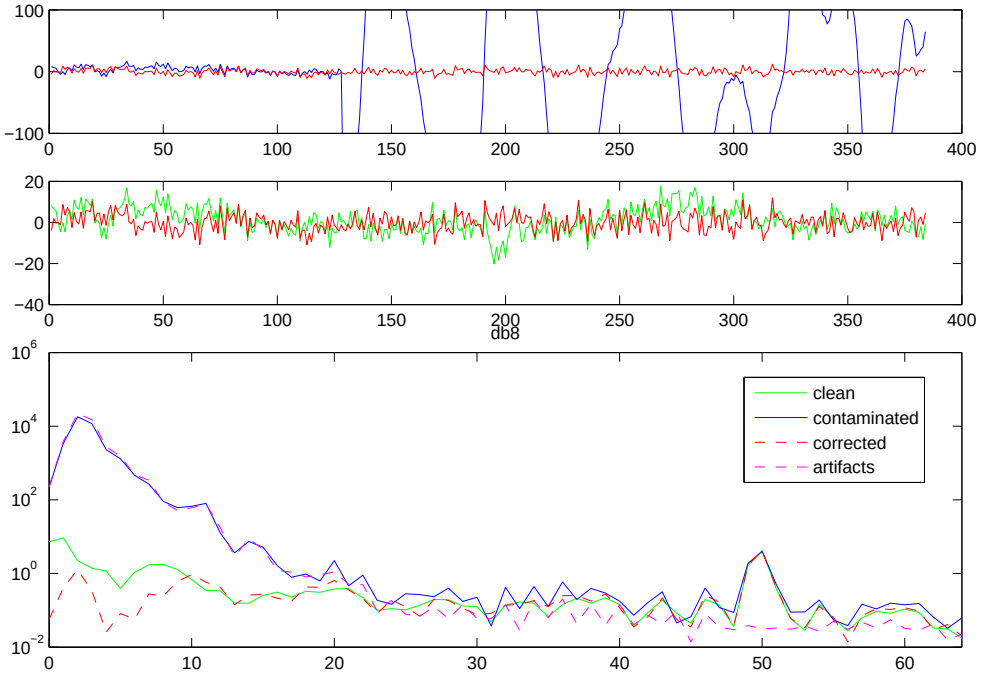

Figure 4-6: A segment of EMA contaminated signals and its spectrum. Top: Pure EEG $x_p(t)$ (blue) and artifact-contaminated EEG $x_c(t)$ (red); Middle: pure EEG $x_p(t)$ (green) and artifact-corrected EEG $x_n(t)$ (red); Bottom: Spectrum of both the signals and the electrodes movement artifacts.

we can see EEG signal (blue) with big oscillation (150-350 data points) caused by artifacts. The big oscillation was removed in the corrected EEG signal (red). In the middle plot, artifacts corrected EEG signal (red)is compared with the original clean EEG signal (green). In the bottom plot, the low frequency spectrum(0-25 Hz) of $x_c(t)$ shifted up by artifacts was corrected by the algorithm especially for the alpha bands(8-12 Hz).

However, in the frequency bands about (2-6 Hz), the spectrum of $x_n(t)$ has a noticeable error with the spectrum of $x_p(t)$. A possible explanation is that there originally exist some artifacts in pure EEG $x_p(t)$. These artifacts increase the spectrum power between (2-6 Hz). After passing the algorithm, these artifacts were removed therefore the error appeared. The support for this explanation is as follows. In the middle plot of Figure 4-6, we can see small oscillations of the signals between (25,75) and (190,300) data points of pure EEG $x_p(t)$ (green). These oscillations are low frequency components and usually are caused by artifacts because the trend of clean EEG signals does not change so fast. If we look at the same period of artifacts removed EEG $x_n(t)$ (red), the small oscillations were removed.

Figure 4-7 shows the decomposition coefficients. We can also see the big oscillation (150-350

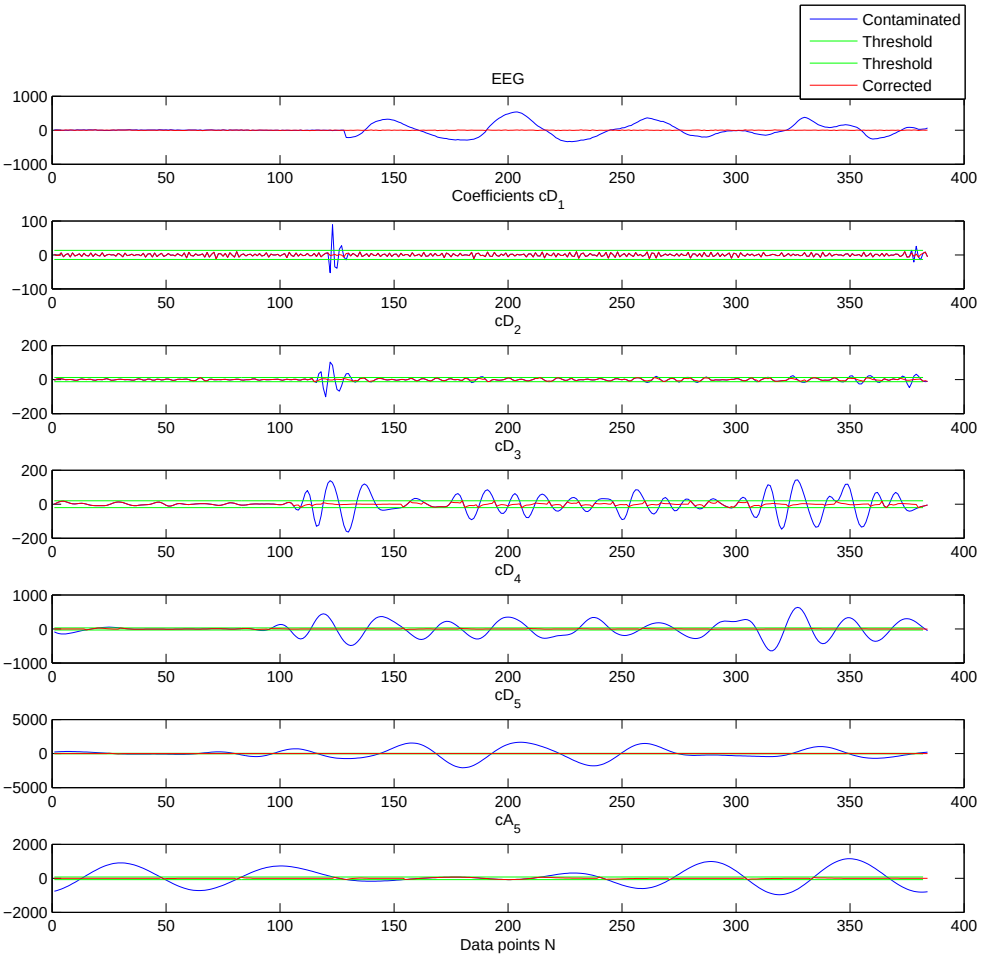

Figure 4-7: Decomposition coefficients of contaminated EEG signal, corrected EEG signal and thresholds.Six bottom plots are the raw coefficients(blue) and thresholds(green). If the coefficients exceed the threshold, they are substituted by the estimated coefficients.The new coefficients(red) are within the thresholds

data points) in coefficients $cA_5, cD_5, cD_4, cD_3, cD_2$ which represents the frequency bands of 0-32 Hz. The corrected new coefficients in red are much smaller and more comparable with the

normal EEG coefficients(0-100 data points). Therefore,most electrodes movement artifacts are eliminated from the signal $x_n(t)$ reconstructed by the new coefficients.

4-2-4 Validation of Different Thresholding Methods

Method(4,3) employed in the artifact correction has already been tested in the previous section. But the other combinations of the method (i,n) are not validated yet. Four threshold computation methods and three substitution methods introduced in 4-8 are tested and validated in this section. The algorithm of alpha wave measurement is the absolute algorithm based on $P^{Abs}(T)$. The best combination of the methods (i,n) will be determined by comparing their MAE results.

Before showing the results of the simulation, an example will be given to illustrate the modification of coefficients by the threshold computation and coefficients substitution. A fragment of OA contaminated signals and its decomposition coefficients cD_4 and thresholds are depicted in Figure 4-8. We can see that method 1 mean+2std and method 4 quantile have similar threshold values and method 3 Coifman gives a more aggressive value. The threshold of method 2 1.5std is small in the sense that coefficients of normal EEG signals exceed the threshold.

Now we look into the effects of substitution methods with the fragment of signals and coefficients. Figure 4-9 shows new coefficients computed by 3 substitution methods. As we can see in Figure 4-9, the coefficients that exceed the threshold are substituted by a constant value for method 1 hard substitution and 2 soft substitution and a non-constant for method 3 Kalman AR estimation.

MAE results of the combination method(i,n) employed in the artifact correction in the alpha wave measurement are shown in Figure 4-10.

For the artifacts of OA and MA, it is clear that threshold computation method(2,n)–1.5std is the worst due to bad estimation of thresholds shown in Figure 4-8. Method (1,n),(3,n),(4,n) are comparable with each other. For these methods, substitution method (i,3)–Kalman AR estimation always gives the smallest MAE compared to method (i,1) and (i,2).

For the artifacts of EMA, four threshold computation methods are comparable. The reason is that estimation error is too small to be comparable with the coefficients changes caused by the artifacts. For example, coefficients cA_5 of normal clean EEG are in the range of -100 to 100, however, artifacts coefficients cA_5 are in the range of -2000 to 2000. Substitution method (i,3)–Kalman AR estimation still gives the smallest MAE when i=1,3,4. And method (i,2)–soft substitution failed for EMA.

Figure 4-11 ,4-12 and 4-13 provide the details about the distribution of the alpha power error for each segment of signals for three type of artifacts respectively. The results is accordance with the results in Figure 4-10. From the three figures we can see Method(1,3) and Method(4,3) always have a small distribution of the errors.

As a conclusion, the best substitution method is method(i,3) which is soft substitution based on Kalman 1-step-ahead AR parameter estimation. All threshold computation methods perform stable and comparable for three type of artifacts except the 2nd method: $\lambda_j = 1.5 \cdot std(M_j)$.

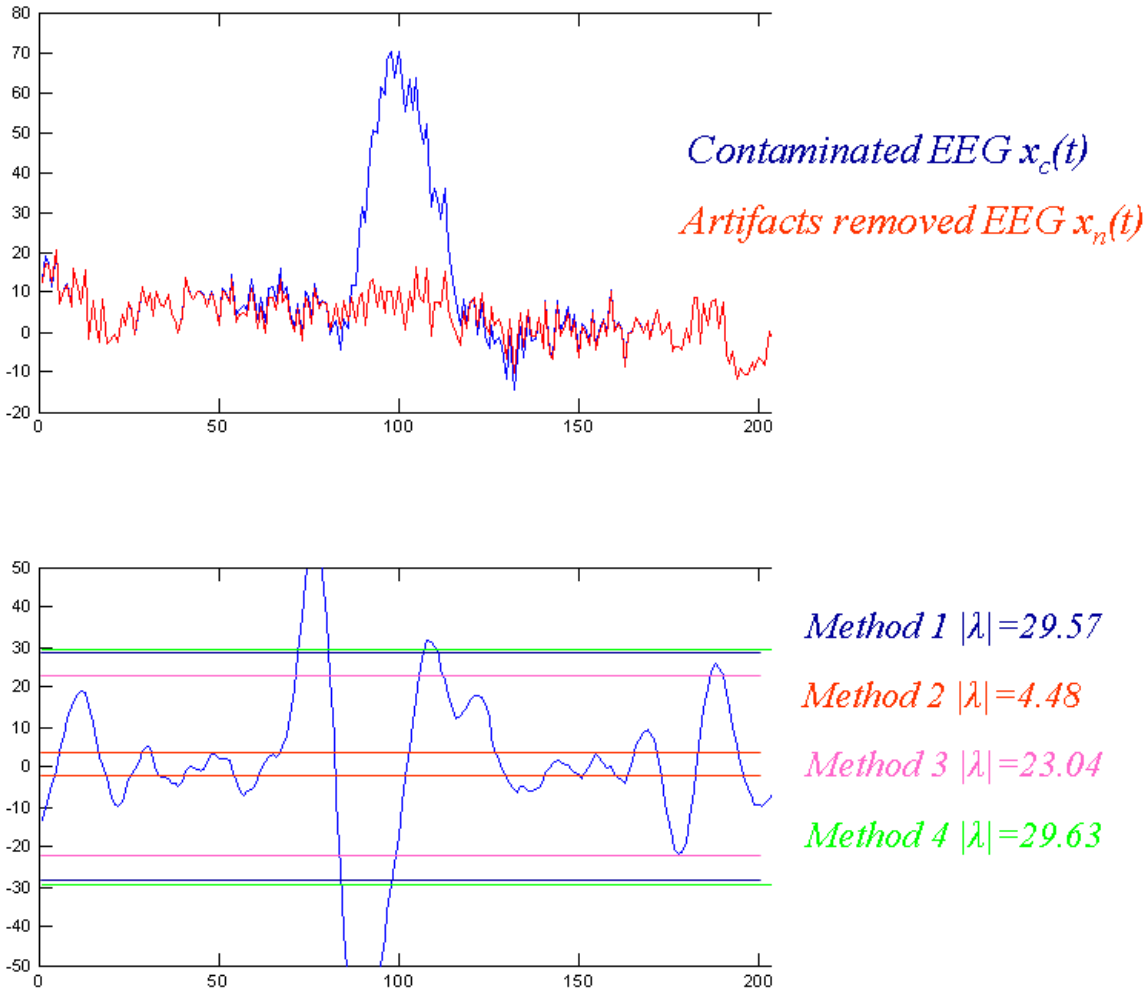

Figure 4-8: Four thresholds for the same vector of coefficients. The top plot shows the EEG signal in the time domain. The bottom plot shows the detailed coefficient vector cD_4 of the signal. Different thresholds are estimated by different methods.

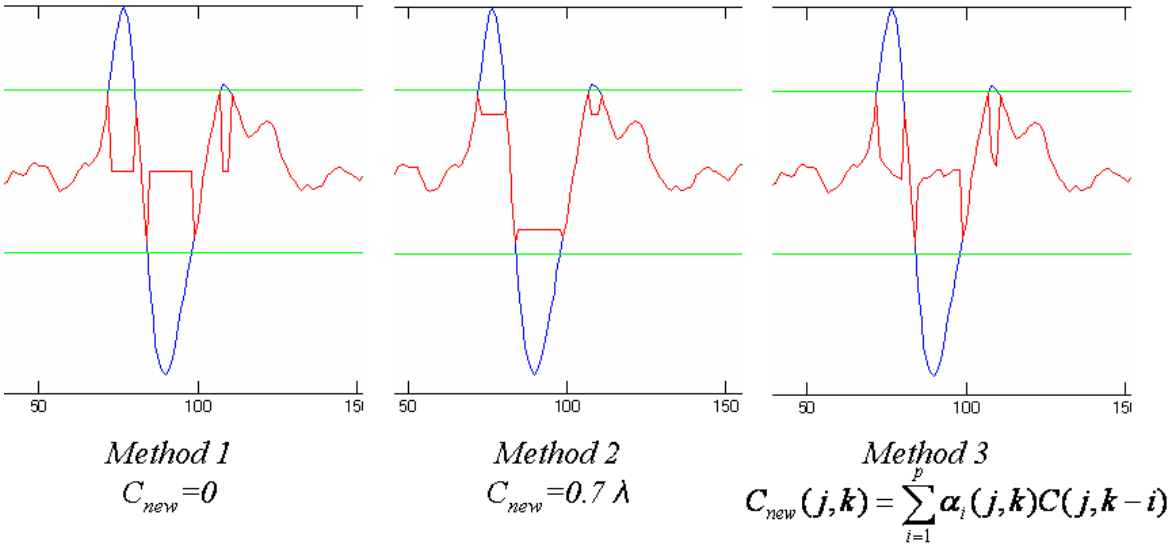

Figure 4-9: New coefficients by 3 substitution methods. Method 1 hard substitution gives constant substitution 0. Method 2 soft substitution provides a constant value with changed sign. Method 3 Kalman AR estimation gives a non-constant substitution.

Hence, substitution method 3 will be employed in the real time relaxation state detection experiment. Threshold computation method 4 will be also used. The reason is that the signals for the estimation of threshold in the baseline measurement are clean EEG signals. In the real time experiment, if we want to perform longer time baseline measurement like 30-60 seconds, artifacts such as eye blinks will inevitably influence EEG signals. Threshold computation method 1 and 3 will also be influenced while method 4 will not be influenced as was explained in Section 3-6-2.

4-3 Summary

In this chapter, firstly, we discussed the simulation of three types of artifacts contaminated EEG signals. Then the validation of artifacts correction algorithm proposed in Chapter 3 was performed and three type of artifacts can be effectively removed from contaminated EEG signals. Finally, the performance of four threshold computation methods and three coefficients substitution methods were tested in terms of mean absolute error (MAE) of the alpha power. The results show that the method(4,3) has the best performance for the simulation. Therefore, method (4,3) - Quantile method and the Soft thresholding based on Kalman estimation, will be employed in the real time relaxation state detection experiment which will be introduced in the next chapter.

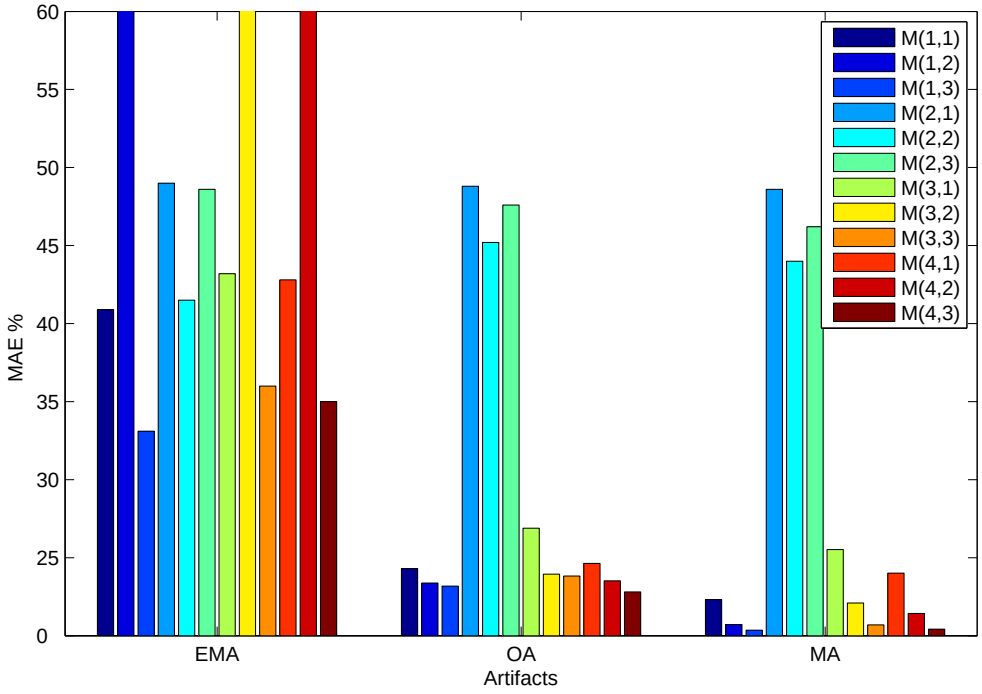

Figure 4-10: Performance of different combination of methods. The values are the normalized MAE between artifact-corrected EEG alpha power $P_n(T)$ and pure EEG alpha power $P_p(T)$ in percentage of $P_p(T)$. The lower the MAE, the better the performance. Substitution method (i,3) Kalman AR estimation has the lowest MAE. Threshold estimation method (1,n) mean+2std and (4,n) quantile are comparable and are better than the other two method (2,n) 1.5std and (3,n) Coifman.

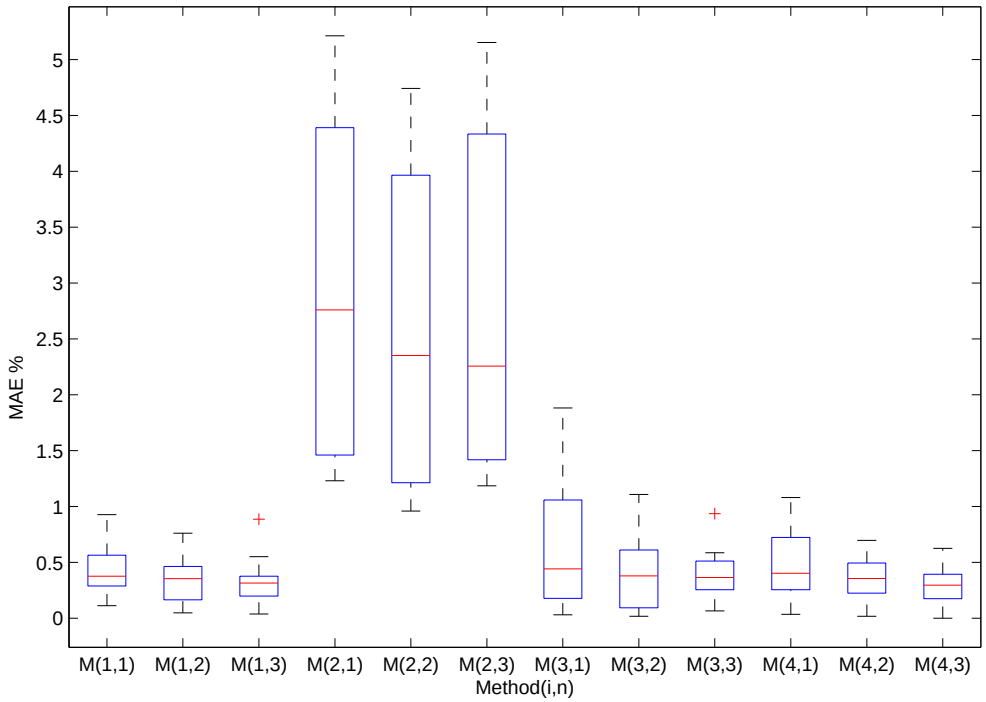

Figure 4-11: Spread of MAE from 12 segments of signals for OA. The dash line infers the spread of the error from 12 segments.The blue rectangular involves the 50% quantile of the errors. The red line is the median of the MAE. The red cross is the extreme value. Method with lower median and smaller spread performs better.

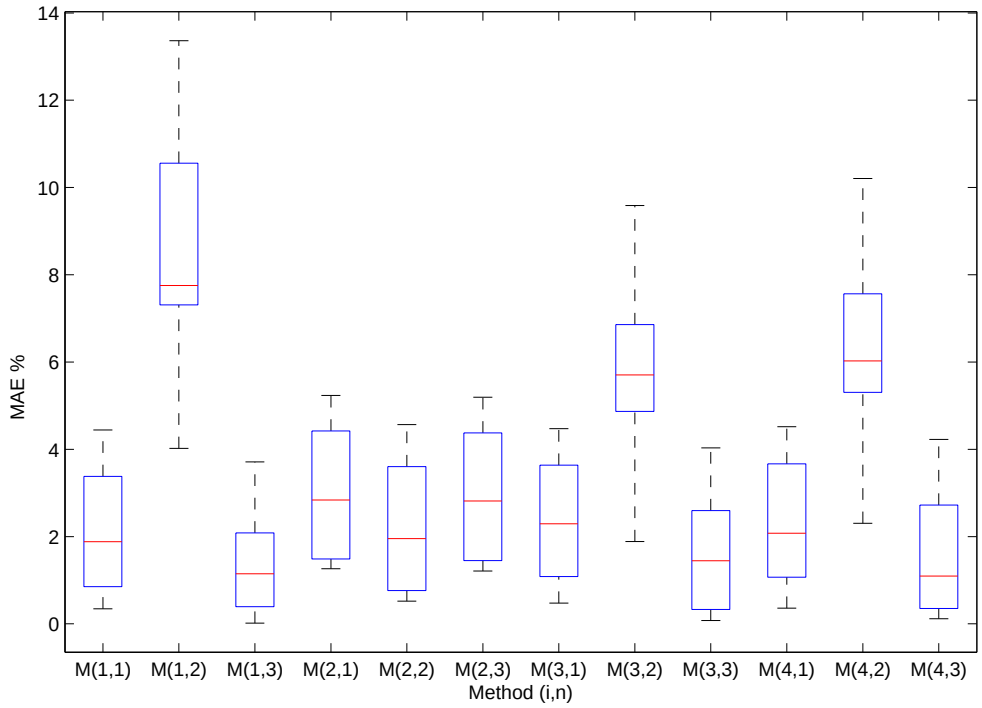

Figure 4-12: Spread of MAE for 12 segments of signals for EMA. Method with lower median and smaller spread performs better.

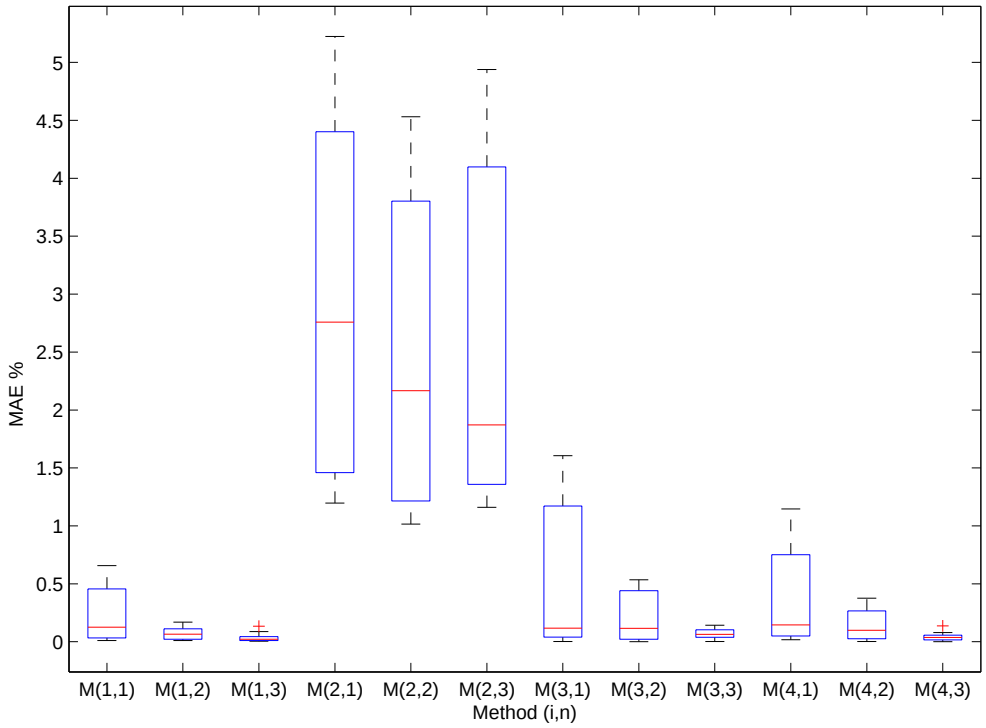

Figure 4-13: Spread of MAE for 12 segments of signals for MA. Method with lower median and smaller spread performs better.

Chapter 5

Experiment of Real Time Alpha Wave Measurement

5-1 Introduction

In this chapter, the proposed algorithm of alpha power detection combined with the artifact correction method will be implemented in an experiment of real time alpha wave measurement. We can evaluate the performance of the proposed algorithm in a real time non-clinic environment. Meanwhile, the alpha power detection algorithm in current applications will also be tested to make a comparison with the proposed algorithm. The two alpha power detection algorithms is reviewed as

- Algorithm 1
  Absolute alpha power detector with artifact correction

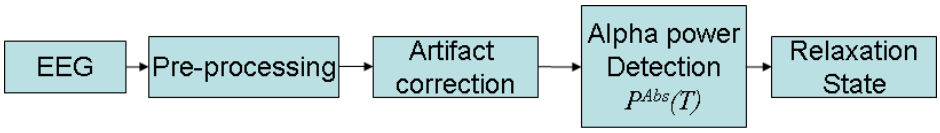

Figure 5-1: Sketch of procedure of the absolute algorithm P^{Abs}

- Algorithm 2
  ratio alpha power detector with artifact rejection (implement on the prototype of neuro-mp3 in Philips Research)

In the experiment, values of two detectors $P^{Abs}(T)$ and $P^{Ratio}(T)$ will be derived from the recorded EEG signals during the experiment. And during each experiment event from T_1 to T_2 shown in Figure 5-3 (the event here is keeping eyes open), the mean alpha power $\overline{P}$ will

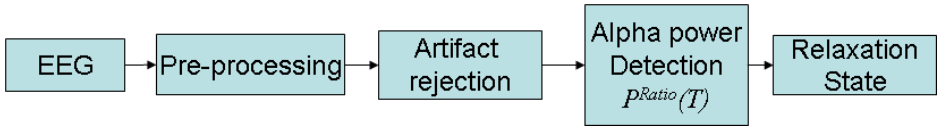

Figure 5-2: Sketch of procedure of the ratio algorithm P^{Ratio}

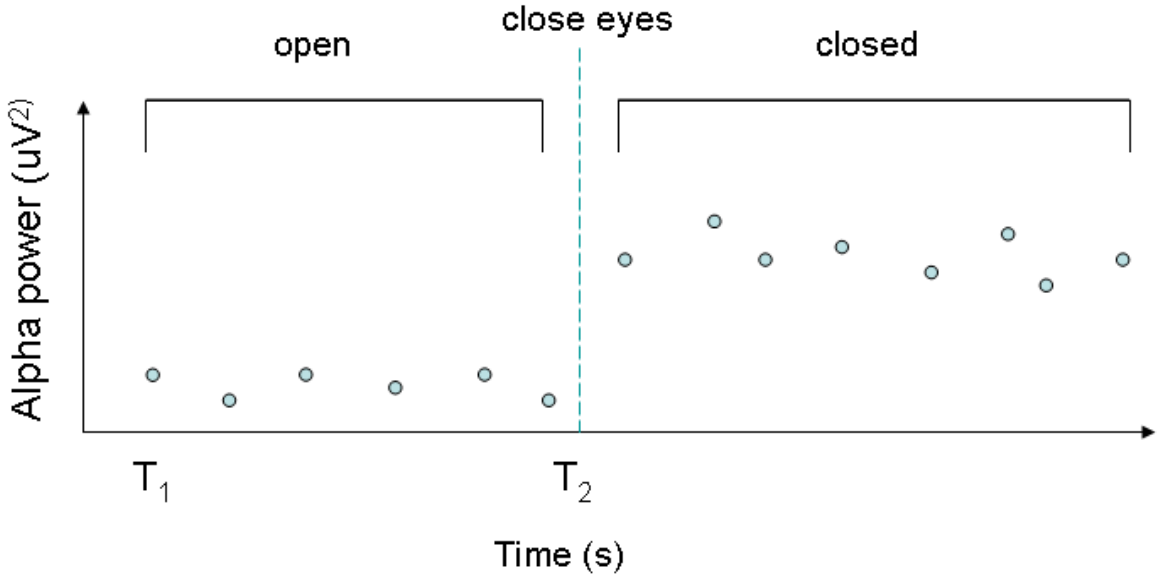

Figure 5-3: A sketch of an alpha power plot of two events: eyes open and eyes closed. Average of alpha power $\overline{P}$ is computed for each event namely $\overline{P}_{eyes\ open}$ and $\overline{P}_{eyes\ closed}$

be computed to describe the statistics of alpha power for this event, namely

$$\overline{P} = \frac{1}{N} \sum_{T=T_1}^{T_2} P(T)$$

$\overline{P}$ also reflects the relaxation state during certain event. If $\overline{P}$ is larger, subjects are more relaxed during a certain event.

5-2 Experiment Design

5-2-1 Experiment Equipments

The experiment should be done in real time. The corresponding equipments and interfaces that realize the real time experiment are shown in Figure 5-4. The subject wears a headphone with electrodes on it. The EEG signals are measured by electrodes and wirelessly sent to the computer by the Nexus (blue box) in subject's hands. The operate system (Linux) passes the data to the Graphical User Interfaces(GUI) of Matlab which is shown in the bottom of the figure. The GUI shown in Figure 5-4 provides a tool for real time monitoring and recording EEG signals. The alpha wave strength and spectrum of EEG signals can also be directly inspected. Meanwhile, the GUI can also detect artifacts within different frequency bands. A real time demonstration can be easily performed with the help of GUI.

5-2-2 Test Design

In order to have a complete evaluation and a clear comparison of the two algorithms,the experiment should contain the following tests.

- Tests in the presence of different artifacts.
  Types of artifacts are grouped into ocular artifacts (OA), electrodes movement artifacts (EMA) and muscle artifacts (MA). By moving certain body parts, subjects can achieve these kind of artifacts. For example, if the subject blinks his eyes, ocular artifacts are triggered. However, actions of the body will inevitably trigger different types of artifacts at the same time. For instance, it is easy to imagine that eyes blinks also cause muscle movements and slightly movements of electrodes.

  In the experiment, five events are included to achieve the threes type of artifacts mentioned above. The events are

  1. Fast eyes blink.
     The interval between one blink to another is less than 1 second
  2. Eyeballs rolling.
     The rolling speed is about π rad/s.
  3. Discrete teeth squeezing.
     Each impulse of teeth squeezing lasts for 1 second and the interval between two nearby impulses is about 4 seconds.

Figure 5-4: The realization of real time alpha wave measurement. Top left: subject with EEG measuring headphone and nexus. Bottom: Graphical User Interface to monitoring and recording EEG signals.

4. Continuous teeth squeezing.
   Subject are asked to keep on squeezing the teeth.
5. Discrete head shaking. Like discrete teeth squeezing, each impulse is 1 second and the interval is 4 seconds.

Each event contains a mixture of different sources of artifacts. For example, discrete teeth squeezing probably yields muscle artifacts and electrodes movement artifacts.

- Tests in the condition of different mental states of subjects.
  One of the purposes of the algorithm is to detect the alpha power change to reveal the change of mental states. As is introduced in Chapter 2, alpha power varies according different mental states. If the subject keeps his/her eyes open and mentally active, the alpha power is low. If the subject keeps his/her eyes closed and mentally relaxed, the alpha power will be high.

  1. eyes open and mental active (Mental activeness is achieved by doing mental arithmetic or reading on the screen.)
  2. eyes closed and mental relaxed

- Tests with different subjects.
  EEG differs from different subjects. EEG signals will also be influenced by age, skin, and especially the hair condition of the subject. Experiment with different subjects will test the robustness of the algorithms.

5-2-3 Experiment Procedure and Events

Based on the tests above, the experiment procedure with different events is listed below.

1. Baseline measurement
   - Training: eyes closed for 30 seconds (Computing the thresholds λ for the artifact correction parameter)
   - Eyes open for 60 seconds
   - Eyes closed for 60 seconds
   - Eyes open for 60 seconds
   - Eyes closed for 60 seconds
2. Artifacts related to eyes
   - Fast blink 20 times
   - Rolling eyeballs for 20 seconds during which eyes are closed
3. Artifacts related to teeth
   - Discrete teeth squeezing 20 times with eyes open
   - Discrete teeth squeezing 20 times with eyes closed
   - Continuous teeth squeezing 20 seconds with eyes open

- Continuous teeth squeezing 20 seconds with eyes closed

4. Artifacts related to head
   - head shaking 20 times with eyes open
   - head shaking 20 times with eyes closed

To summarize, the events and their average alpha power $\overline{P}$ during each event are listed below:

Table 5-1: Summary of the events and their label for alpha power in the test

Eyes open	Eyes closed
no actions of body $\overline{P}_{bsl,\ open}$	no actions of body $\overline{P}_{bsl,\ closed}$
Eyes blinks $\overline{P}_{eyeblinks,\ open}$	Eyeballs rolling $\overline{P}_{eyes\ rolling,\ closed}$
Head shaking $\overline{P}_{head,\ open}$	Head shaking $\overline{P}_{head,\ closed}$
Discrete teeth squeezing $\overline{P}_{teeth_d,\ open}$	Discrete teeth squeezing $\overline{P}_{teeth_d,\ closed}$
Continuous teeth squeezing $\overline{P}_{teeth_c,\ open}$	Continuous teeth squeezing $\overline{P}_{teeth_c,\ closed}$

5-3 Evaluation

5-3-1 Subjects

Five subjects are tested in the experiments. The information of the subjects are listed below. If the hair of a subject is thicker, the contact between the electrodes and the scalp will be worse. More artifacts are likely be introduced during the experiment.

Table 5-2: Information of five subjects

Subject	Age	Gender	Hair condition
1	25	male	thin
2	28	male	thin
3	23	male	thick
4	22	male	thick
5	24	female	normal

5-3-2 Performance of Artifacts Elimination

In this subsection, the goal of the evaluation is to see which algorithm is able to eliminate more artifacts in the alpha wave measurement and detection. The two algorithms are shown in Figure 5-1 and Figure 5-2.

Recall the assumption in section 2-4-3.

- Assumption
  Alpha power is contributed by the pure EEG signal and the artifacts namely

$$P(T) = P^{pureEEG}(T) + P^{artifact}(T).$$

$P^{artifact}(T)$ can be regarded as the error namely

$$\epsilon = P(T) - P^{pureEEG}(T).$$

Then we can obtain the indicator of the performance of artifact elimination in the artifact measurement in the case of eyes open namely

$$\epsilon = \frac{|\overline{P}_{atfct,\ open} - \overline{P}^{pureEEG}_{atfct,\ open}|}{\overline{P}^{pureEEG}_{atfct,\ open}} \times 100\%$$

$\overline{P}_{atfct,\ open}$ stands for the total mean alpha power in the artifact measurement during the event of eyes open. $\overline{P}^{pureEEG}_{atfct,\ open}$ stands for the mean alpha power contributed by pure EEG signal in the artifact measurement during the event of eyes open. However, $\overline{P}^{pureEEG}_{atfct,\ open}$ is always unknown in the experiment. The baseline total mean alpha power $\overline{P}_{bsl,\ open}$ will be assumed as the approximation of $\overline{P}^{pureEEG}_{atfct}$. That is

$$\overline{P}_{bsl,\ open} \approx \overline{P}^{pureEEG}_{atfct,\ open} \tag{5-1}$$

The reason is as follows. Firstly, in the baseline measurement, alpha power contributed by artifact is almost zero, that is

$$\overline{P}_{bsl,\ open} \approx \overline{P}^{pureEEG}_{bsl,\ open}(T) + \dot{0}$$

Secondly, in the case of eyes open and mentally active, the alpha power contributed by pure EEG $P^{pureEEG}(T)$ always stays relatively low and varies within a small range whenever the subject stays still or moves his/her body. So the mean alpha power from pure EEG $\overline{P}^{pureEEG}$ will remain roughly the same in both baseline measurement and artifact measurement namely

$$\overline{P}^{pureEEG}_{bsl,\ open} \approx \overline{P}^{pureEEG}_{atfct,\ open}$$

Therefore the equation 5-1 holds. Now we get the artifact elimination performance indicator

$$\epsilon = \frac{|\overline{P}_{atfct,\ open} - \overline{P}_{bls,\ open}|}{\overline{P}_{bls,\ open}} \times 100\% \tag{5-2}$$

For example, in the case of eyes blinks events, it is

$$\epsilon_{eyes} = \frac{|\overline{P}_{eyes\ blinks,\ open} - \overline{P}_{bss,\ open}|}{\overline{P}_{bsl,\ open}} \times 100\%$$

What worth mentioning is that the above equations is is only valid with eyes open. In the case of eyes closed and mentally relaxed, movements of the body will decrease the relaxation of mental states. Alpha power contributed by pure EEG signals will also decrease. So the mean alpha power from pure EEG signals will not be the same.

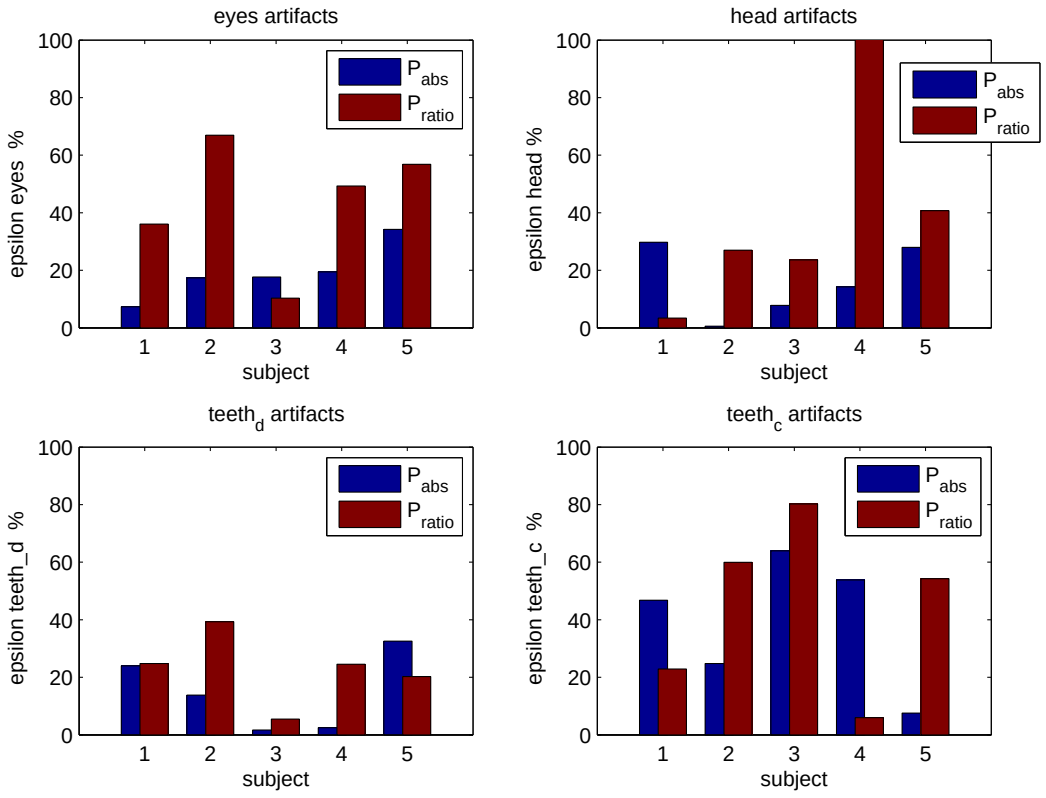

Figure 5-5: Normalized absolute error in the form of deviation between $\overline{P}_{bsl,\ open}$ and $\overline{P}_{atfct,\ open}$ from the assumed ground truth. Four events, two algorithms and five subjects are tested. A small deviation infers better artifact elimination performance

Based on the equation 5-2, we are able to analyze the performance of the algorithms by computing the difference ϵ. Figure 5-5 shows five subjects' results of artifact elimination performance. As we can see in the figure, the algorithm $P^{Abs}(T)$ has less artifacts influence than the algorithm $P^{Ratio}(T)$ in the most cases(15 from 20 cases).Therefore, we can conclude that $P^{Abs}(T)$ algorithm is better in eliminating artifact in this experiment. Moreover, both algorithms perform well for the artifact of discrete teeth squeezing because the deviation is the smallest among all the artifact events. And continuous teeth squeezing introduces the most artifacts to the alpha power. Both algorithms are highly influenced compared with the other three artifact events.

5-3-3 Performance on the Relaxation State Detection

In this subsection, the evaluation will be made to see if the relaxation state is still detectable by the two algorithms in the presence of artifacts.

Because of the lack of the information of pure EEG signals, the real alpha power level of a subject is never available when there exist artifacts. That means, the real mental states of a subject is unknown. However, we can still get a relative mental states level between the events of eyes open and eyes closed.

The following relation of mean alpha power contributed by pure EEG signals among different events is obvious:

$$\overline{P}^{pureEEG}_{bsl,\ closed} > \max(\overline{P}^{pureEEG}_{bsl,\ open}, \overline{P}^{pureEEG}_{atfct,\ open}, \overline{P}^{pureEEG}_{atfct,\ closed}) \tag{5-3}$$

Another inequality we can deduce from the experiment conditions is

$$\overline{P}^{pureEEG}_{atfct,\ closed} > \overline{P}^{pureEEG}_{bsl,\ open} \tag{5-4}$$

The reason is as follows. In the event of eyes open in the baseline measurement, subjects were asked to keep mentally active. So the average alpha power $\overline{P}^{pureEEG}_{bsl,open}$ is small. In the event of eyes closed in the artifact measurement, subjects were asked to produce certain artifacts while trying to stay mentally idle. So the average alpha power $\overline{P}^{pureEEG}_{atfct,\ closed}$ should be larger than $\overline{P}^{pureEEG}_{bsl,\ open}$. For example, in the event of discrete teeth squeezing with eyes closed, subjects may be mentally active when generating the squeezing impulse. They were probably relaxed in the 4 seconds interval (the interval is introduced in Section 5-2-2). Therefore, the average of alpha power $\overline{P}^{pureEEG}_{atfct,\ closed}$ should be always larger than $\overline{P}^{pureEEG}_{bsl,\ open}$. However, the inequalities 5-3 and 5-4 may not hold for the total alpha power contributed by pure EEG signals and artifacts. Therefore, based on the inequalities 5-3 and 5-4, we can evaluate which algorithm performs better on relaxation state detection in the presence of artifacts. We define miss detection by

- *The relaxing state of subjects is not detected while subjects are relaxed.*

The objective function that can reflect the miss detection of the relaxation state is

$$\epsilon_{miss\ detection} = \frac{(\overline{P}_{atfct,\ closed} - \overline{P}_{bsl,open})}{\overline{P}_{bsl,\ open}} \times 100\%$$

If $\epsilon \leq 0$, two mental states are totally undistinguishable which will cause miss detection of the relaxation state. And if $\epsilon > 0$, the larger ϵ is, the more distinguishable two mental states are. In other words, the risk of miss detection is lower with larger ϵ.

Now we define the false alarm as

- *The relaxing state of subjects is detected while subjects are not relaxed.*

The objective function that can reflect the false alarm of the relaxation state is

$$\epsilon_{false\ alarm} = \frac{(\overline{P}_{bsl,\ closed} - \overline{P}_{atfct,\ open})}{\overline{P}_{bsl,\ open}} \times 100\%$$

Similar to the miss detection objective function, if $\epsilon \leq 0$, two mental states are totally undistinguishable which will cause false alarm of the relaxed state. And if $\epsilon > 0$, the larger ϵ is, the more distinguishable two mental states are. The risk of false alarm is lower with larger ϵ.

To show how distinguishable two mental states are if there are no artifacts, ϵ in the baseline measurement is computed by

$$\epsilon_{bsl} = \frac{(\overline{P}_{bsl,\ closed} - \overline{P}_{bsl,\ open})}{\overline{P}_{bsl,\ open}} \times 100\%$$

Figure 5-6 shows the results of the detection performance of two algorithms in the case of no artifacts in the baseline measurement. Figure 5-7 shows the results of miss detection objective function in the case of artifacts in the artifacts measurement. From Figure 5-6, the proposed

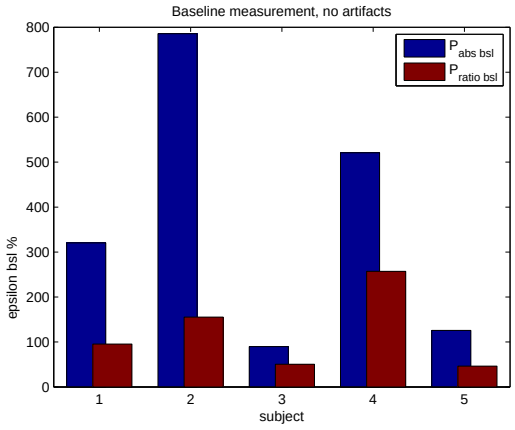

Figure 5-6: Mean alpha power difference between the period of eyes open and eyes closed for the 2 algorithms and 5 subjects when there is no artifacts. Larger difference ϵ means more detectable alpha power difference between eyes open and eyes closed.

algorithm with detector $P^{Abs}(T)$ always provides a larger difference between $\overline{P}_{bsl,\ open}$ and $\overline{P}_{bsl,\ closed}$. And we can also infer that this difference depends on subjects. If we compare the two figures 5-6 and 5-7, the alpha power difference in the presence of artifacts shown in

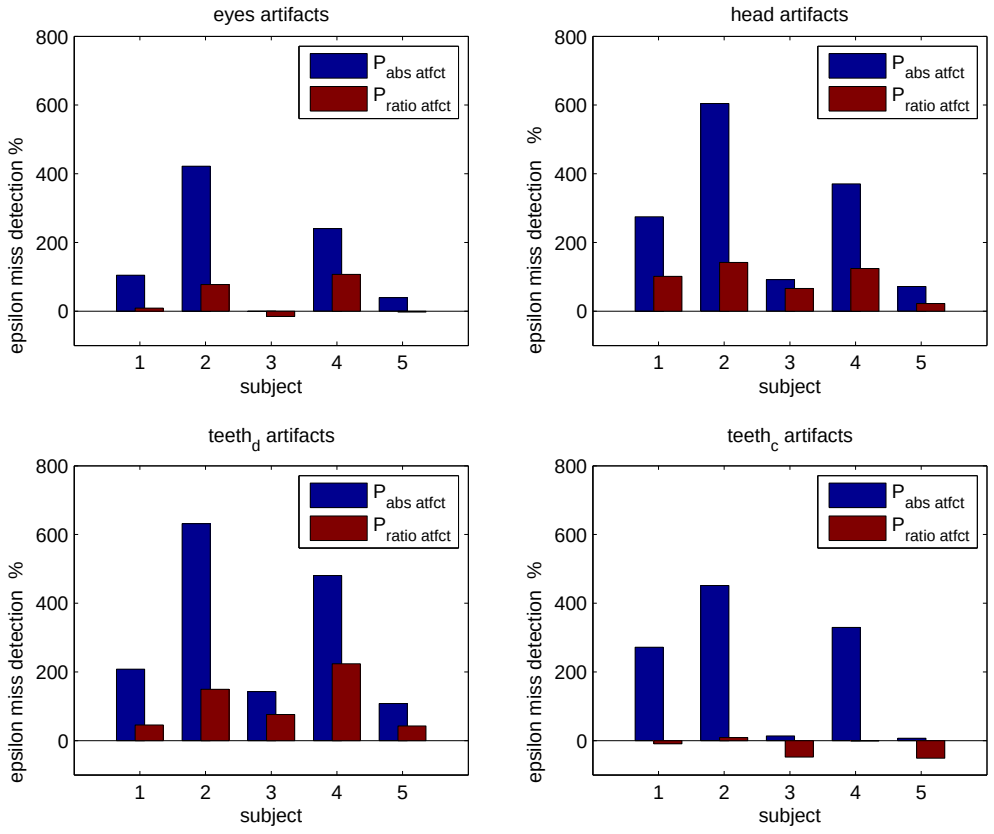

Figure 5-7: Mean alpha power difference between $\overline{P}_{bsl\ eyes\ open}$ and $\overline{P}_{atfct\ eyes\ closed}$ for 4 artifact events, 2 algorithms and 5 subjects when artifacts are introduced. With larger difference ϵ,alpha power between two mental states is more detectable. So the risk of miss detection is lower with a larger difference.

Figure 5-7 is always smaller than that in the absence of artifacts shown in Figure 5-6. This verifies

$$\overline{P}^{pureEEG}_{bsl,\ closed} > \overline{P}^{pureEEG}_{atfct,\ closed}$$

in Inequality 5-3.

If the difference is below zero, miss detection happens. From Figure 5-7,the miss detection rate of the proposed algorithm $P^{Abs}(T)$ is 1/20, the only miss detection lies in the top left plot for eyes artifacts, Subject 3. The miss detection rate of the algorithm $P^{ratio}(T)$ is 5/20.

Figure 5-8 shows the results obtained for the false alarm objective function in the case of artifacts. If the difference is below zero, a false alarm occurs. It follows from Figure 5-8 that

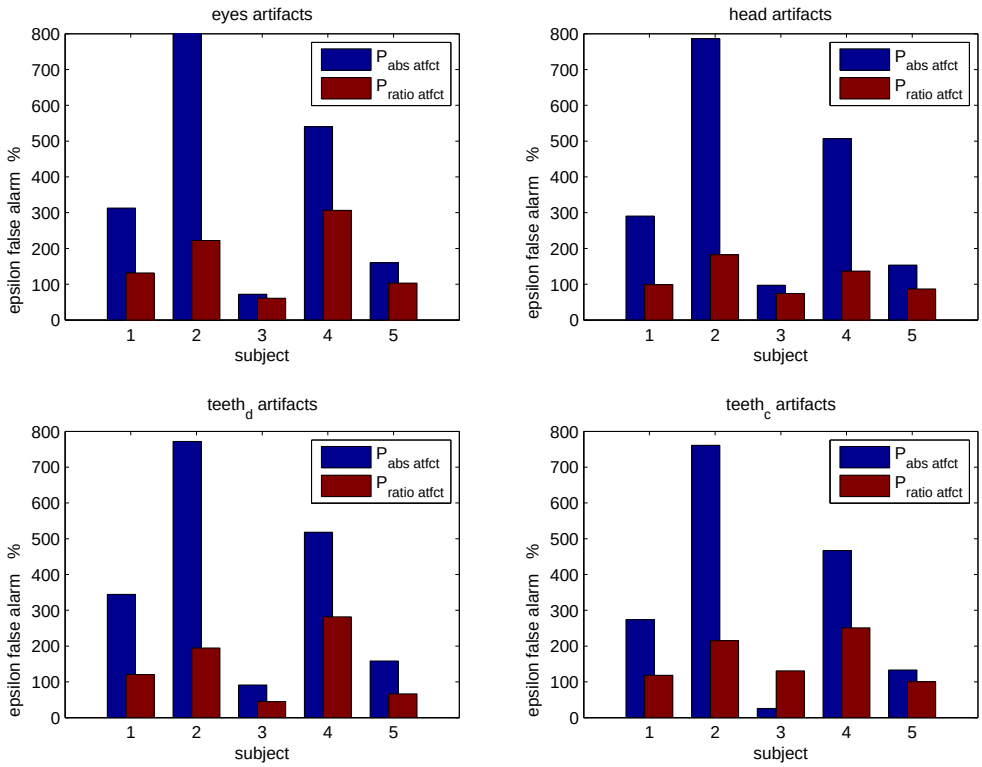

Figure 5-8: Mean alpha power difference between $\overline{P}_{bsl\ eyes\ closed}$ and $\overline{P}_{atfct\ eyes\ open}$ for 4 artifact events, 2 algorithms and 5 subjects when artifacts are introduced. With larger difference ϵ, alpha power between two mental states is more detectable. So the risk of false alarm is lower with a larger difference.

both algorithms have no false alarms. Compared Figure 5-6 with Figure 5-8, alpha power difference of the same subject and event from two figures are close to each other. It infers that the risk of false alarm is lower compared with the risk of miss detection. From the results in Figure 5-7, we can conclude that the proposed algorithm $P^{Abs}(T)$ has better performance when it comes to relaxation state detection compared with the other algorithm $P^{Ratio}(T)$ in the absence as well as in the presence of artifacts.

5-4 Summary

In this chapter, two alpha power detection algorithms were applied in a real time alpha wave measurement experiment. The experiment was designed to include four artifact events, two mental state events and five subjects. Based on the results of the experiment, two evaluations were performed to test the performance of the sensitivity to artifact disturbance and the performance relaxation state detection. Algorithm P^{Abs} with artifact correction shows good results in both evaluations. Algorithm P^{Ratio} with artifact rejection shows a good result in the evaluation of false alarms. On the other hand, the most serious artifact event is continuous teeth squeezing. For this event, both algorithms have the lowest performance among all the artifact events.

Chapter 6

Conclusion

6-1 Conclusion

The goal of this thesis is to find a solution for a reliable real time alpha wave detection from EEG measurements in a non-clinical environment. Once that reliable detection is achieved it can be applied to real time relaxation state detection in people's daily life. For example, the algorithm to detect alpha waves can be implemented in the neuro-mp3 player which detects people's mental states of relaxation.

As a solution, an artifact correction algorithm based on stationary wavelet transform (SWT) has been developed to enhance the reliability of alpha wave detection. The algorithm could effectively reduce artifacts such as ocular artifacts, muscle artifacts and electrodes movement artifacts from EEG signals in the experiment conditions introduced in this thesis. This improves the robustness of the alpha wave detection algorithm. Meanwhile, the availability of the artifact correction algorithm enlarges the application field of alpha wave detection and reduces the restriction of experiment conditions. For example, the subject can be more free to have body movements during the alpha wave measurement.

The proposed alpha wave detection algorithm with artifact correction was compared with the alpha wave detection algorithm which is currently implemented in the prototype of neuro-mp3 player in Philips Research. The results show that the newly proposed alpha wave detection method outperforms the current employed one in the experiment conditions mentioned in this thesis.

6-2 Summary of the Achievements

The main contributions that can be extracted from this work are

- Better solution to reduce the artifacts in the alpha wave measurement. An improved artifact correction algorithm has been developed. The algorithm improves the current

artifact correction algorithm based on the stationary wavelet transform in a non-clinical environment in the following respects

1. The algorithm can be used to reduce different types of artifacts from EEG signals.
2. A method based on quantiles improves the estimation of thresholds on wavelet coefficients according to the result in 4-10.
3. Soft substitution based on Kalman 1-step-ahead AR parameter estimation is applied in the thresholding of contaminated wavelet coefficients. It outperformed the other two methods introduced in 3-6-3 according to the result in 4-10.

- Better solution to the detection of alpha waves. A new detector of alpha power in the measurement of alpha waves was proposed. The detector uses historical information to normalize the alpha power. Compared with the current detector implemented in the prototype of neuro-mp3 player, it increase the delectability of alpha wave in the relaxation state detection experiment based on the result in Figure 5-7.

- A user interface is developed to realize the real time alpha wave measurement in matlab. The interface monitors the EEG signals, alpha wave level and relaxation state. The interface based on GUI of Matlab builds the connection between the EEG measuring equipment and Matlab which offers real time recording and processing of EEG signals. A demonstration on EEG artifact correction and relaxation state detection detection can be easily performed based on the interface.

- A classification of EEG artifacts in the non-clinical environment is made and the influence of artifacts on EEG signals is analyzed and reported.

- An experiment on artifact correction and relaxation state detection is designed. The experiment is designed to test the detection of alpha wave level in the presence of different artifacts.

6-3 Future Work

The proposed alpha wave detection and artifact correction algorithm still has errors that cannot be ignored in the alpha wave detection and measurement. This is probably caused by the estimation error of the thresholds of wavelet coefficients. The current estimation of thresholds is based on 30 to 60 seconds baseline measurement. Once the thresholds are determined, they are not changed until the experimental condition change, for example, the change of different subjects. It could be an improvement to develop an adaptive threshold estimation method to decrease the estimation error of the thresholds in order to decrease the errors in the alpha wave detection.

The demonstration of artifact correction and relaxation state suffers from a running speed problem. The running speed of artifact correction algorithm will slow down if we increase the AR model order of the soft thresholding method. The improvements could be done by optimizing the platform on which Matlab is running. For example, we can use faster PC instead of the current one and optimize the operate system. On the other hand, a Matlab code could also be optimized to achieve a higher running speed of the algorithm.

In this thesis, EEG signals are recorded by a single channel. Moreover, the alpha wave detection and artifact correction algorithm can only deal with one channel data currently. The algorithm could be improved to adapt to multi-channel data based on the increase of running speed.

Some other studies can be done. For example, during the experiments, 5 subjects were measured, three of them were Chinese. For these subjects the alpha wave is very strong and easy to detect while the other 2 Caucasian subjects always have an unstable alpha wave level which is not easy to detect. The same phenomenon was mentioned in [D.Chestakov, 2008]. Therefore there could be a race dependency in the strength of alpha wave. An easy experiment can be done by taking a significant number of subjects of each population and analyzing their alpha wave strength with the experiment performed in the thesis.

Bibliography

[Coifman and Donoho, 1995] Coifman, R. R. and Donoho, D. L. (1995). Translation-invariant de-noising. *LECTURE NOTES IN STATISTICS-NEW YORK-SPRINGER VERLAG*, pages 125–125.

[Creutzfeldt et al., 1966] Creutzfeldt, O. D., Watanabe, S., and Lux, H. D. (1966). Relations between eeg phenomena and potentials of single cortical cells. i. evoked responses after thalamic and erpicortical stimulation. *Electroencephalography and clinical neurophysiology*, 20(1):1–18.

[D.Chestakov, 2008] D.Chestakov, D. I. S. G. G. M. (2008). Detection of Steady-State Visual Evoked Potentials in the EEG. *Philips Research Europe*, pages 61–62.

[Donoho, 1995] Donoho, D. L. (1995). De-noising by soft-thresholding. *IEEE transactions on information theory*, 41(3):613–627.

[E. Niedermeyer, 1999] E. Niedermeyer, L. d. S. (1999). *Electroencephalography*. Williams & Wilkins, Baltimore, MD.

[Fatourechi et al., 2007] Fatourechi, M., Bashashati, A., Ward, R. K., and Birch, G. E. (2007). Emg and eog artifacts in brain computer interface systems: a survey. *Clinical Neurophysiology*, 118(3):480–494.

[Hayes, 2008] Hayes, M. H. (2008). *Statistical digital signal processing and modeling*. Wiley India Pvt. Ltd.

[Healey, 2001] Healey, R. W. P. E. V. J. (2001). Toward Machine Emotional Intelligence: Analysis of Affective Physiological State. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(10):1175–1191.

[Krishnaveni and Anitha, 2006] Krishnaveni, V. J. S. and Anitha, L. R. K. (2006). Removal of ocular artifacts from eeg using adaptive thresholding of wavelet coefficients. *Journal of Neural Engineering*, 3(4):338–346.

[Kumar et al., 2008] Kumar, P. S., Arumuganathan, R., Sivakumar, K., and Vimal, C. (2008). Removal of ocular artifacts in the eeg through wavelet transform without using an eog reference channel. *Int. J. Open Problems Compt. Math*, 1(2).

[Mallat, 1989] Mallat, S. G. (1989). A theory for multiresolution signal decomposition: The wavelet representation. *IEEE transactions on pattern analysis and machine intelligence*, 11(7):674–693.

[Nason and Silverman, 1995] Nason, G. and Silverman, B. (1995). The stationary wavelet transform and some statistical applications. *LECTURE NOTES IN STATISTICS-NEW YORK-SPRINGER VERLAG-*, pages 281–281.

[Sheng, 1996] Sheng, Y. (1996). Wavelet transform. *The transforms and applications handbook*, pages 747–827.

[Taswell, 1995] Taswell, C. (1995). Wavbox 4: A software toolbox for wavelet transforms and adaptive wavelet packet decompositions. *LECTURE NOTES IN STATISTICS-NEW YORK-SPRINGER VERLAG*, pages 361–361.

[Valens, 2004] Valens, C. (2004). A really friendly guide to wavelets. *Available in: http://perso. wanadoo. fr/polyvalens/clemens/wavelets/wave lets. html.*

[Verleger et al., 1982] Verleger, R., Gasser, T., and Mocks, J. (1982). Correction of eog artifacts in event-related potentials of the eeg: Aspects of reliability and validity. *Psychophysiology*, 19(4):472–480.

[Wang, 2009] Wang, L. (2009). Psycho-physiological event detection from ECG and EEG. pages 20–27.

[Zikov et al., 2002] Zikov, T., Bibian, S., Dumont, G. A., Huzmezan, M., and Ries, C. R. (2002). A wavelet based de-noising technique for ocular artifact correction of the electroencephalogram. volume 1. Engineering in Medicine and Biology, 2002. 24th Annual Conference and the Annual Fall Meeting of the Biomedical Engineering Society] EMBS/BMES Conference, 2002. Proceedings of the Second Joint.

Appendix A

Related Matlab Code for Alpha Wave Measurement Algorithm

A-1 Artifact Correction Method

```
function [xnew,pCo,lasts,SWC_th,ARall] = wl_artifactAAR(x,motherwavelet,thre,buffer,last,ARall)
% [xnew,pCo,lasts,SWC_th,ARall] = wl_artifactAAR(x,motherwavelet,thre,buffer,last,ARall)

%% parameters
p=6; %AR model order
AR=1; %AR=1, AR method, AR=0, soft thresholding
Uc=0.001; %innovation factor
thn=1; %artifact coefficient spread factor, if c(k)>thre, c(k) and c(k-1) are both substituded.
tha=1; %%artifact coefficient spread factor, if c(k)>thre, c(k) and c(k+1) are both substituded.
pCo=zeros(6,1);
%% Mirror data and SWT decomposition
if length(x)==128*buffer-10
    xn = [fliplr(x(2:22));x;fliplr(x(end-21:end-1))];
end
if length(x)==128*buffer
    xn = [flipdim(x(2:17),1);x;flipdim(x(end-16:end-1),1)];
end
[SWC] = swt(xn,5,motherwavelet);
for i=1:6
    pCo(i,1)=max(abs(SWC(i,:)));
end
SWC_th=SWC;
%% Initalize AR parameters
if isempty(ARall)
    ARall.AR=zeros(6,p);
    ARall.Vt=0;

    ARall.At=zeros(p,p);
    ARall.I=eye(p);
end

STEMP=[];

if isempty(last)
    last=randn(p,6);
end

%% Thresholding
for i=1:6
    for j=1:length(SWC(i,:))
        if AR
            if j>p
                Stemp=SWC(i,j-p:j-1)';
            else
                Stemp=[last(end-p+j:end,i);SWC(i,1:j-1)'];
            end
            STEMP=[STEMP Stemp];
        end
```

```
        if abs(SWC(i,j))>thre(7-i)
            if AR
                SWC(i,j)=ARall.AR(i,:)*Stemp;
                if j>thn
                    for ka=1:thn
                        if abs(SWC(i,j-ka))>0.8*thre(7-i)
                            SWC(i,j-ka)=ARall.AR(i,:)*STEMP(:,end-ka);
                        end
                    end

                end
                if j<length(SWC(i,:))-thn
                    for kb=1:thn
                        if abs(SWC(i,j+kb))>0.8*thre(7-i)
                        SWC(i,j+kb)=ARall.AR(i,:)*[Stemp(2:end);SWC(i,j)];
                        end
                    end
                end
            else
                SWC(i,j)=0;
                if j>thn
                    for ka=1:thn
                        if abs(SWC(i,j-ka))>0.8*thre(7-i) && SWC(i,j-ka)~=0
                         SWC(i,j-ka)=sign(SWC(i,j-ka))*0.5*thre(7-i); %soft thresholding
%                             SWC(i,j-ka)=0; %hard thresholding
                        end
                    end

                end
                if j<length(SWC(i,:))-tha
                    for kb=1:tha
                        if abs(SWC(i,j+kb))>0.8*thre(7-i)
                         SWC(i,j+kb)=sign(SWC(i,j+kb))*0.5*thre(7-i);
%                             SWC(i,j+kb)=0;
                        end
                    end
                end
            end
        end
%% update AR parameters , estimation by Kalman gain
        if AR
            if abs(SWC(i,j))<=thre(7-i)
                Et=SWC(i,j)-ARall.AR(i,:)*Stemp;
                if Et==0
                    Et=0.01;
                end
                ARall.Vt=(1-Uc)*ARall.Vt+Uc*Et^2;
                kt=ARall.At*Stemp/(Stemp'*ARall.At*Stemp+ARall.Vt);
                ARall.AR(i,:)=ARall.AR(i,:)+kt'*Et;
                ARall.At=ARall.At-(1+Uc)*kt*[Stemp(1:p-1);SWC(i,j)]'*ARall.At+Uc^2*ARall.I;
            end
            if  abs(SWC(i,j))>thre(7-i) %if Kalman estiamtion goes wrong, reset AR parameters
                SWC(i,j)=0;
%                   ARall.AR=zeros(6,p);
%                   ARall.Vt=0;
%                   ARall.At=zeros(p,p);
%                   ARall.I=eye(p);
            end
        end
    end
end
for i=1:6
    lasts(:,i)=SWC(i,end-p+1:end)';
end
%% reconstruction
xs = iswt(SWC,motherwavelet)';
xnew = xs(17:128*buffer);
```

A-2 Artifact Rejection Method

Introduction: For each EEG segment with 128 data points, if the amplitude of 25% of the data points exceeds the threshold, the entire segment will be rejected. Instead, the previous EEG segment will be regarded as the current EEG segment.

```
function [std_EEG,rejection]=arti_rejection(x,ns,std_EEG,artifact_mag)
x_now=x(end-127:end);
% artifact_mag=4.2;
artifact_length=0.04;
factor_avg=0.05;
flag_dummy=0;
rejection=0;
th_length=artifact_length*ns;
mean_EEG=mean(x_now);
temp_std=std(x_now);
```

```
% std_EEG=std(x_now);
if std_EEG==0;
    std_EEG=temp_std;
end
for i=1:ns
    if abs(x_now(i)-mean_EEG)/std_EEG >= artifact_mag
        flag_dummy = flag_dummy+1;
    end
end
if flag_dummy >= th_length
    rejection=1;
else
    std_EEG  = sqrt((temp_std *temp_std * factor_avg) + (std_EEG * std_EEG * (1 - factor_avg)));
    if std_EEG>= 10
        std_EEG=10;
    end
end
```

A-3 Alpha Wave Measurement Algorithm Implemented in GUI

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

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

% Last Modified by GUIDE v2.5 23-Jun-2009 14:15:24

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
    'gui_Singleton',  gui_Singleton, ...
    'gui_OpeningFcn', @AlphaWaveDetectionRT_OpeningFcn, ...
    'gui_OutputFcn',  @AlphaWaveDetectionRT_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 AlphaWaveDetectionRT is made visible.
function AlphaWaveDetectionRT_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 AlphaWaveDetectionRT (see VARARGIN)

% Choose default command line output for AlphaWaveDetectionRT
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

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

```
% --- Outputs from this function are returned to the command line.
function varargout = AlphaWaveDetectionRT_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure                                                    68
% 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;                                                 73


function tYlim_Callback(hObject, eventdata, handles)
% hObject    handle to tYlim (see GCBO)                                          78
% 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 tYlim as text
%        str2double(get(hObject,'String')) returns contents of tYlim as a double 83


% --- Executes during object creation, after setting all properties.
function tYlim_CreateFcn(hObject, eventdata, handles)
% hObject    handle to tYlim (see GCBO)                                          88
% 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.                                                   93
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

                                                                               98

function motherWavelet_Callback(hObject, eventdata, handles)
% hObject    handle to motherWavelet (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)                 103

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

                                                                               108
% --- Executes during object creation, after setting all properties.
%%
function motherWavelet_CreateFcn(hObject, eventdata, handles)
% hObject    handle to motherWavelet (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB             113
% 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'))  118
    set(hObject,'BackgroundColor','white');
end


% --- Executes on button press in start.                                       123
function start_Callback(hObject, eventdata, handles)
% hObject    handle to start (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
%% inital setting                                                              128
%javaaddpath('/home/biosignal/biosignal/common/data_convert/SAND/matlabEmoServer/libSocket.jar');
% tYlim = str2double(get(handles.tYlim,'String'));
% motherWavelet = get(handles.motherWavelet,'String');
set(handles.run_t,'string','A');
set(handles.saves,'string','B');                                               133
set(handles.records,'string','B');
set(handles.sign,'Visible','off');
set(handles.start,'Visible','on');
set(handles.stop,'Visible','off');
guidata(hObject,handles);                                                      138
set(handles.baselineC,'BackgroundColor',[0.702 0.702 0.702]);
set(handles.normalize,'BackgroundColor',[0.702 0.702 0.702]);
guidata(hObject,handles);
alphabands=[8 12];
totalbands=[4 40];                                                             143
%load arcoeff.mat
cla(handles.rawData);
cla(handles.filteredData);
cla(handles.ratio);
set(handles.sign,'Visible','off');                                             148
kb=1;
compute_th=false;
kn=1;
kt=1;
kb=1;                                                                          153
fs=128;
```

```
% pa_high=40;
% pa_low=0;
% pafft_high=0.7;
% pafft_low=0;
[ctx,TE,T,X,TS,XS,tend,xend,pend,ttend,xxend,pendFFT,pEOG,pEMG,pCoSWT,Pa_norm,Pafft_norm]=deal([]);
[Data_raw,Data_time,Data_new,Data_ntime,Data_ratio_time,Data_norm_wl,Data_ratio_FFT,Data_pa_wl,
    Data_pa_FFT,Data_ratio_SWT]=deal([]);
run=true;
wv_co_th=eval(get(handles.wv_th,'string'));
t_swt = 0;
x_swt = 0;
pad = 0;
pend=0;
paend=0;
paendFFT=0;
PA=200;
PAFFT=200;
ratioFFT=0;
ratioSWT=0;
ratioFFTend=0;
ratioWTend=0;
set(handles.baselineC,'BackgroundColor',[0.702 0.702 0.702]);
set(handles.Recording,'BackgroundColor',[0.702 0.702 0.702]);
[c,d]=butter(3,4/512,'high'); % lowcut-off freq.=1 Hz
compute_th=0;
fq=512;
saveData=false;
recordData=false;
denoise_plot=false;
%power_buf=3;
last=[];
overlap=0;
ARall=[];
plotwindow= str2double(get(handles.sPlotwindow,'String'));
%% Conection and data conversion
client = Client('LocalHost', 16000);%conect to the server
while run
    wv_co_th=eval(get(handles.wv_th,'string'));
    set(handles.start,'Visible','off');
    set(handles.stop,'Visible','on');
    buffer = str2double(get(handles.epoch,'string'));
    motherWavelet = get(handles.motherWavelet,'String');
    Ylim_th = eval(get(handles.tYlim,'String'));
    EMG_th = str2double(get(handles.th_EMG,'String'));
    EOG_th = str2double(get(handles.th_EOG,'String'));
    MOV_th = str2double(get(handles.th_MOV,'String'));
    noise_th = str2double(get(handles.th_noise,'String'));
    plotwindow= str2double(get(handles.sPlotwindow,'String'));
    if get(handles.run_t,'string')=='B'
        run = false;
    set(handles.start,'Visible','on');
    set(handles.stop,'Visible','off');
    end
    if get(handles.baselineT,'string')=='A'
        compute_th = true;
    end
    if get(handles.schange,'string')=='A'
        denoise_plot = true;
    end
    if get(handles.schange,'string')=='B'
        denoise_plot = false;
    end
    if get(handles.sNorm,'string')=='B'
        norml = false;
    end
    if get(handles.sNorm,'string')=='A'
        norml = true;
    end

    str = client.getMessage();%read message from client
    msg=char(str);
    data=readstr(msg,'A',fq);%transfer msg in char to double
    tbyte=data.t;
    xbyte=data.A;
    TS=[TS;tbyte];XS=[XS;xbyte];%store the buffered data to analyze
    if length(TS)==fq
        T=[T;TS];
        X=[X;XS];
        if length(T)==fq*buffer
%% Following is Alpha detection algorithm
            if kt~=1
                tend=t_swt(end);
                paend=pa_abs_SWT;
                xend=x_swt(end);
                pend=pa_norm_SWT;
                xxend=x_original(end);
                ttend=t_original(end);
%                 pendFFT=ratio_n;
                paendFFT=pa_no_method;
                ratioFFTend=ratioFFT;
```

```
            ratioWTend=ratioSWT;
        end
        t_original=T;xx=X;
        r=fq/fs;
        xtemp=512;
        if kt==1
            xx = [flipdim(xx(1:xtemp),1);xx;flipdim(xx((end-xtemp+1):end),1)];
        else
            xx=[X_last;xx;flipdim(xx((end-xtemp+1):end),1)];
        end
        xx=filtfilt(c,d,xx);
        xx=xx((xtemp+1):(end-xtemp));
        noise=0;
        for o=1:length(xx)
            if xx(o)==0
                noise=1;
            end
        end
%% fft method
        xFFT=detrend(xx);
        xFFT=(xFFT-mean(xFFT));

        %             [freqFFT,powerFFT]=fftspec(xFFT,fq);
        [powerFFT,freqFFT] = pwelch(xFFT,fq,floor(fq * overlap),fq,fq);
        %             [freqFFT,powerFFT]=arspec_armasel(xFFT,sPlotwindow,tYlim);
        %             powerFFT=powerFFT*var(powerFFT);
        pa_abs_FFT=sum(powerFFT(logical(freqFFT<alphabands(2) & freqFFT>=alphabands(1))));
        ptotalFFT=sum(powerFFT(logical(freqFFT<totalbands(2) & freqFFT>=totalbands(1))));
        ratioFFT=pa_abs_FFT/ptotalFFT;
%         pafft_abs=ratioFFT;
%% Pre-process
        [y,xx,ctx]=resample_rt(ctx,xx,1,r);
        tx=t_original(1:r:end);
        t_original=tx(1:end-10);
        te=tx(end-9:end);
        TE=[TE te];
        if kt>1;
            t_original=[TE(:,kt-1);t_original];
        end
        x_original=detrend(xx);
        x_original=x_original-mean(x_original);

        if kt~=1
            [powerr,freqr] = pwelch(x_original,fs,floor(fs * overlap),fs,fs);
        else
            [freqr,powerr]=fftspec(x_original,fs);
        end
        pa_no_method=sum(powerr(logical(freqr<alphabands(2) & freqr>=alphabands(1))));

        power_n=powerr*var(xx);
        par=sum(power_n(logical(freqr<53 & freqr>=47)));
        if length(xx)>fs
            xnew=xx(end-fs+1:end);
        else
            xnew=xx;
        end
        xnewd=detrend(xnew);
        xnewMax=abs(max(xnewd));
%% Noise & electrods movement detection
        MOV=0;
        color_n='b';
        if xnewMax>MOV_th
            MOV=1;
            color_n='r';
        end

        set(handles.nMOV,'string',xnewMax);
        color_EMG='g';
        color_EOG='g';
        set(handles.nNoise,'string',par);
        if par>noise_th || noise==1
            color_n='r';
            set(handles.tNoise,'BackgroundColor','r');
        end
        if par<=noise_th && noise==0 && par>=0.5*noise_th
            set(handles.tNoise,'BackgroundColor','y');
        end
        if par<=0.2*noise_th && noise==0
            set(handles.tNoise,'BackgroundColor','g');
        end
%% DWT method
        t_swt=t_original;
        [sig_dwt,pCo]=wvlt_correction_s(xnew,motherWavelet,wv_co_th);
%% SWT method, artifact correction
        [sig_swt,pCo_swt,last,swc,ARall] = wl_artifactAAR(x_original,motherWavelet,wv_co_th,
            buffer,last,ARall);
        x_swt=sig_swt;
        pCo_swt=flipdim(pCo_swt,1);
```

```
            swc=flipdim(swc,1);
            set(handles.pCot,'string',pCo_swt);
%% power computation
            if kt~=1
                [powerSWT,freqSWT] = pwelch(x_swt,fs,floor(fs * overlap),fs,fs);
            else
                [freqSWT,powerSWT]=fftspec(x_swt,fs);
            end
            pa_abs_SWT=sum(powerSWT(logical(freqSWT<alphabands(2) & freqSWT>=alphabands(1))));
            ptotalSWT=sum(powerSWT(logical(freqSWT<totalbands(2) & freqSWT>=totalbands(1))));
            ratioSWT=pa_abs_SWT/ptotalSWT;

%% Baseline threshold computation
            if compute_th
                CO=[];
                pCoSWT=[pCoSWT abs(swc)];
                pEMG=[pEMG;pCo(2)];
                pEOG=[pEOG;pCo(5)];
                boundary=[0.9;0.9;0.9;0.9;0.9;0.9];
                weights=[2;2;2;2;2;2;];
                set(handles.sign,'Visible','on');
                Pa_norm=[Pa_norm;pa_abs_SWT];
                if kb==60
                    for i=1:6
                            CO(1,i)=weights(i)*quantile(pCoSWT(i,:),boundary(i));
                    end

                    str_CO=['[',num2str(CO),']'];

                    EMG_th=mean(pEMG)+3.5*std(pEMG);
                    EOG_th=mean(pEOG)+4*std(pEOG);
                    pa_high=max(Pa_norm);
                    pa_low=0; %Currently, for convenience, set 0.
                    pa_range=[pa_low pa_high];
                    str_pa=['[',num2str(pa_range),']'];
                    set(handles.tYlim,'String',str_pa);
                    set(handles.th_EMG,'String',EMG_th);
                    set(handles.th_EOG,'String',EOG_th);
                    set(handles.wv_th,'String',str_CO);

                    set(handles.baselineC,'BackgroundColor',[0.702 0.702 0.702]);
                    pEMG=[];
                    pEOG=[];
                    pCoSWT=[];
                    Pa_norm=[];
                    set(handles.sign,'Visible','off');
                    kb=1;
                    compute_th=false;
                    set(handles.baselineT,'string','B');
                end

                kb=kb+1;
            end
%% Normalize
            test_time=60;
            if norml
                if kn<test_time/2
                    set(handles.sign,'Visible','off');
                else
                    set(handles.sign,'Visible','on');
                end
               Pa_norm=[Pa_norm;pa_abs_SWT];
               Pafft_norm=[Pafft_norm;pa_abs_FFT];
                if kn==test_time/2

                    beep;
                    pause(0.5);
                    beep;

                end
                if kn==test_time
                    beep;
                    pause(0.5);
                    beep;
%                     pa_high=1.2*quantile(Pa_norm,0.98);
%                     pa_low=0.8*quantile(Pa_norm,0.02);
                    pa_high=max(Pa_norm);
                    pa_low=min(Pa_norm);
                    pa_range=[pa_low pa_high];
                    str_pa=['[',num2str(pa_range),']'];
                    pa_abs_th=1.1*roc(Pa_norm(2:test_time/2-1),Pa_norm(test_time/2+1:test_time-1));
%                     pafft_abs_th=1.3*roc(Pa_norm(2:29),Pa_norm(31:59));
                    pa_norm_th=(pa_abs_th-pa_low)/(pa_high-pa_low);
%                     pafft_norm_th=(pafft_abs_th-pafft_low)/(pafft_high-pafft_low);
                    set(handles.tYlim,'String',str_pa);
                    Pa_norm=[];
                    hold off;
                    axes(handles.ratio);
                    hold on;plot(1:5000:10001,[pa_norm_th pa_norm_th pa_norm_th],'-m');hold off;
                    hold off;
```

```
                    axes(handles.filteredData);
                    hold on;plot(1:5000:10001,[pa_abs_th pa_abs_th pa_abs_th],'-m');hold off;
                    norml=false;
                    set(handles.sNorm,'string','B');
                    set(handles.normalize,'BackgroundColor',[0.702 0.702 0.702]);
                    kn=1;
                end
                kn=kn+1;
            end
%% EMG EOG artifacts detection
            if length(t_swt)>length(x_swt)
                t_swt=t_swt(1:length(x_swt));
            end
            if length(t_swt)<length(x_swt)
                t_swt=[t_swt;t_swt(2*end-length(x_swt)+1:end)];
            end

            if pCo(2)>EMG_th
                color_EMG='r';
            end
            set(handles.nEMG,'string',pCo(2));
            if pCo(5)>EOG_th %&& pCo(5)<1500
                color_EOG='r';
            end
            if pCo(5)>EOG_th*3
                MOV=1;
%                 color_MOV=[0.702 0.702 0.702];
                color_n='r';
            end
            set(handles.nEOG,'string',pCo(5));

%% prepare data for ploting

            pa_norm_SWT=(pa_abs_SWT-Ylim_th(1))/(Ylim_th(2)-Ylim_th(1));

            pa_norm_plot_SWT=[pend;pa_norm_SWT];
            t_pa_plot=[tend;t_swt(end)];
            pa_abs_plot_SWT=[paend;pa_abs_SWT];
            ratio_plot_FFT=[ratioFFTend;ratioFFT];
            ratio_plot_SWT=[ratioWTend;ratioSWT];
            if buffer~=1 && kt~=1
                t_swt_plot=[tend;t_swt((end-fs+1):end)];
                x_swt_plot=[xend;x_swt((end-fs+1):end)];
                t_original_plot=[ttend;t_original((end-fs+1):end)];
                x_original_plot=[xxend;x_original((end-fs+1):end)];
            else
                t_swt_plot=[tend;t_swt];
                x_swt_plot=[xend;x_swt];
                t_original_plot=[ttend;t_original];
                x_original_plot=[xxend;x_original];
            end
            pa_plot_no_method=[paendFFT;pa_no_method];
            kt=kt+1;
%% plot results
            if MOV==0
                set(handles.tEMG,'BackgroundColor',color_EMG);
                set(handles.tEOG,'BackgroundColor',color_EOG);
                set(handles.tMOV,'BackgroundColor','g');
            end
            if MOV==1
                set(handles.tMOV,'BackgroundColor','r');
                set(handles.tEMG,'BackgroundColor',color_EMG);
                set(handles.tEOG,'BackgroundColor',color_EOG);
            end
            set(gcf,'CurrentAxes',handles.rawData);
            hold on;plot(t_original_plot,x_original_plot,color_n,t_swt_plot,x_swt_plot,'g');xlim([(
                t_original_plot(end)-mod(t_original_plot(end),plotwindow)-buffer) (t_original_plot(
                end)-mod(t_original_plot(end),plotwindow)+plotwindow)]);ylim([-70 70]);
            set(gcf,'CurrentAxes',handles.filteredData);
            if denoise_plot
                hold on;semilogy(t_pa_plot,ratio_plot_SWT,['-*' color_n],t_pa_plot,ratio_plot_FFT,'-
                    ok');xlim([(t_swt(end)-mod(t_swt(end),plotwindow)-buffer) (t_swt(end)-mod(t_swt(
                    end),plotwindow)+plotwindow)]);ylim([0 1]);
            else
                hold on;semilogy(t_pa_plot,pa_abs_plot_SWT,['-*' color_n],t_pa_plot,
                    pa_plot_no_method,'-ok');xlim([(t_swt(end)-mod(t_swt(end),plotwindow)-buffer) (
                    t_swt(end)-mod(t_swt(end),plotwindow)+plotwindow)]);ylim(Ylim_th);
            end
            set(gcf,'CurrentAxes',handles.ratio);
            hold on;plot(t_pa_plot,pa_norm_plot_SWT,['-*' color_n],t_pa_plot,ratio_plot_FFT,'-ok');
                xlim([(t_swt(end)-mod(t_swt(end),plotwindow)-buffer) (t_swt(end)-mod(t_swt(end),
                plotwindow)+plotwindow)]);ylim([0 1]);
            set(gcf,'CurrentAxes',handles.power);
            semilogy(freqSWT,powerSWT,'-g',freqFFT,powerFFT);
            xlim([0 64]);
            ylim([0.001 50]);grid on;
            drawnow;
%% clear the buffer and save data;
            if kt~=1 && buffer~=1
                T_last=T(1:fq);
```

```
                X_last=X(1:fq);
                T=T(fq+1:end);
                X=X(fq+1:end);
            end
            if kt~=1 && buffer==1
                T_last=T;
                X_last=X;
                T=[];
                X=[];
            end

        end
        TS=[];XS=[];
        if get(handles.saves,'string')=='A'
            saveData = true;
        end
        if get(handles.records,'string')=='A'
            recordData = true;
        end
        if  recordData && buffer~=1 && kt~=1
            set(handles.Recording,'BackgroundColor','g');
            Data_raw=[Data_raw;x_original((end-fs+1):end)];
            Data_time=[Data_time;t_original((end-fs+1):end)];
            Data_new=[Data_new;x_swt((end-fs+1):end)];
            Data_ntime=[Data_ntime;t_swt((end-fs+1):end)];
            Data_norm_wl=[Data_norm_wl;pa_norm_SWT];
            Data_ratio_FFT=[Data_ratio_FFT;ratioFFT];
            Data_ratio_time=[Data_ratio_time;t_swt(end)];
            Data_ratio_SWT=[Data_ratio_SWT;ratioSWT];
            Data_pa_wl=[Data_pa_wl;pa_abs_SWT];
            Data_pa_FFT=[Data_pa_FFT;pa_no_method];

            if saveData
                Data.x_raw=Data_raw;
                Data.t_raw=Data_time;
                Data.x_corr=Data_new;
                Data.t_corr=Data_ntime;
                Data.norm_SWT=Data_norm_wl;
                Data.ratio_FFT=Data_ratio_FFT;
                Data.ratio_SWT=Data_ratio_SWT;
                Data.t_alpha=Data_ratio_time;
                Data.abs_SWT=Data_pa_wl;
                Data.abs_FFT=Data_pa_FFT;
                Data.thresholds=wv_co_th;
                Data.wavelet=motherWavelet;
                Data.buffer=buffer;
                Data.EOG_EMG_th=[EOG_th;EMG_th];
                Data.paMAXMIN=Ylim_th;
                savefile = get(handles.filename,'String');
                file = ['/home/biosignal/hdrive/bio_rt/' savefile '.mat'];
                save(file,'Data');
                saveData=false;
                recordData=false;
                [Data_raw,Data_time,Data_new,Data_ntime,Data_pa_wl,Data_pa_FFT,Data_ratio_time,
                    Data_norm_SWT,Data_ratio_FFT]=deal([]);
                set(handles.Recording,'BackgroundColor',[0.702 0.702 0.702]);
                set(handles.records,'string','B');
                set(handles.saves,'string','B');
            end
        end
    end
end
%%

% --- Executes on button press in stop.
function stop_Callback(hObject, eventdata, handles)
% hObject    handle to stop (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
%handles.run_t=false;
set(handles.run_t,'string','B');
set(handles.start,'Visible','on');
set(handles.stop,'Visible','off');
guidata(hObject,handles);

% --- Executes on button press in baselineC.
function baselineC_Callback(hObject, eventdata, handles)
% hObject    handle to baselineC (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
set(handles.baselineT,'string','A');
set(handles.baselineC,'BackgroundColor','g');
guidata(hObject,handles);


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

```
set(handles.saves,'string','A');
guidata(hObject,handles);


% --- Executes on button press in Recording.
function Recording_Callback(hObject, eventdata, handles)
% hObject    handle to Recording (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
set(handles.records,'string','A');
guidata(hObject,handles);



% --- Executes on button press in change.
function change_Callback(hObject, eventdata, handles)
% hObject    handle to change (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
set(handles.schange,'string','A');
set(handles.change,'BackgroundColor','g');
set(handles.pa,'BackgroundColor',[0.702 0.702 0.702]);
guidata(hObject,handles);


% --- Executes on button press in pa.
function pa_Callback(hObject, eventdata, handles)
% hObject    handle to pa (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
set(handles.schange,'string','B');
set(handles.pa,'BackgroundColor','g');
set(handles.change,'BackgroundColor',[0.702 0.702 0.702]);
guidata(hObject,handles);



function filename_Callback(hObject, eventdata, handles)
% hObject    handle to filename (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 filename as text
%        str2double(get(hObject,'String')) returns contents of filename as a double


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

function th_noise_Callback(hObject, eventdata, handles)
% hObject    handle to filename (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 filename as text
%        str2double(get(hObject,'String')) returns contents of filename as a double


% --- Executes during object creation, after setting all properties.
function th_noise_CreateFcn(hObject, eventdata, handles)
% hObject    handle to 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
function th_EOG_Callback(hObject, eventdata, handles)
% hObject    handle to filename (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 filename as text
%        str2double(get(hObject,'String')) returns contents of filename as a double


% --- Executes during object creation, after setting all properties.
```

```
function th_EOG_CreateFcn(hObject, eventdata, handles)
% hObject    handle to 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
function th_EMG_Callback(hObject, eventdata, handles)
% hObject    handle to filename (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 filename as text
%        str2double(get(hObject,'String')) returns contents of filename as a double


% --- Executes during object creation, after setting all properties.
function th_EMG_CreateFcn(hObject, eventdata, handles)
% hObject    handle to 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
function th_MOV_Callback(hObject, eventdata, handles)
% hObject    handle to filename (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 filename as text
%        str2double(get(hObject,'String')) returns contents of filename as a double


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



function epoch_Callback(hObject, eventdata, handles)
% hObject    handle to epoch (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 epoch as text
%        str2double(get(hObject,'String')) returns contents of epoch as a double


% --- Executes during object creation, after setting all properties.
function epoch_CreateFcn(hObject, eventdata, handles)
% hObject    handle to epoch (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 wv_th_Callback(hObject, eventdata, handles)
% hObject    handle to wv_th (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 wv_th as text
%        str2double(get(hObject,'String')) returns contents of wv_th as a double


% --- Executes during object creation, after setting all properties.
function wv_th_CreateFcn(hObject, eventdata, handles)
% hObject    handle to wv_th (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');                                          778
end


function sPlotwindow_Callback(hObject, eventdata, handles)                           783
% hObject    handle to sPlotwindow (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 sPlotwindow as text               788
%        str2double(get(hObject,'String')) returns contents of sPlotwindow as a double


% --- Executes during object creation, after setting all properties.
function sPlotwindow_CreateFcn(hObject, eventdata, handles)                          793
% hObject    handle to sPlotwindow (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.                    798
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
                                                                                     803

% --- Executes on button press in normalize.
function normalize_Callback(hObject, eventdata, handles)
% hObject    handle to normalize (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB                  808
% handles    structure with handles and user data (see GUIDATA)
set(handles.sNorm,'string','A');
set(handles.normalize,'BackgroundColor','g');
guidata(hObject,handles);
```

A-3-1 Data conversion

```
 function data = readstr(msg,channels,fq)
% function data = readstr(msg,channels,fq)
% convert data in char to double                                                     3
% input:
% msg is 1X1 char
% format example:
%    t1.000A0Z5231.3432A1Z5231.3432A2Z5231.3432...A15Z5231.3432 , where 'Z'
%    is the separating character.                                                    8
% channels: recordded channels,example 'ABCM'
% fq: sampling frequency
% output:
% data: struct with  sampling time,channel names and signals
format_part='';                                                                     13
N_channels=length(channels);
N_data = N_channels*16;
t = sscanf(msg,'%*c %e %*s',[1 inf]);
t = linspace(t,t+15/fq,16)';
data.t=t;                                                                           18
for k=1:N_data
    format_part=[format_part '%*c %*e %*c %e '];
end
format_channel = ['%*c %*e' format_part];
x_total= sscanf(msg,format_channel,[1 inf])';                                       23
if length(x_total)<N_data
    x_total=zeros(N_data,1);
end
xm=reshape(x_total,16,N_channels);
i=1:N_channels;                                                                     28
data.(channels(i))=xm(:,i);
```