بسم الله الرحمن الرحيم

IMPROVED FEATURE EXTRACTION ALGORITHM FOR BRAIN COMPUTER INTERFACE

By

Sami N. Alrabie

A thesis submitted for the requirements of the degree

Of Master of Science in Computer Science

Supervised By
Dr. Anas M. Ali Fattouh
Dr. Fadi F. Fouz

COMPUTER SCIENCE DEPARTMENT
FACULTY OF COMPUTING AND INFORMATION TECHNOLOGY
KING ABDULAZIZ UNIVERSITY
JEDDAH – SAUDI ARABIA
Rabi'I 1436H – January 2015G

IMPROVED FEATURE EXTRACTION ALGORITHM FOR BRAIN COMPUTER INTERFACE

By

Sami N. Alrabie

This thesis has been approved and accepted in partial fulfillment of the requirements for the degree of Master of Science in Computer Science

EXAMINATION COMMITTEE

	Name	Rank	Field	Signature
Internal Examiner	Dr. Abdullah Saad AL-Malaise AL-Ghamdi	Associate Prof	Software Engineering	
External Examiner	Dr. Elsayed Abdel RazekElfar	Associate Prof	Electrical Engineering	
Co-Advisor	Dr. Fadi F. Fouz	Prof	Parallel Computing	
Advisor	Dr. Anas M. Ali Fattouh	Associate Prof	Automatic Control	

KING ABDULAZIZ UNIVERSITY

Rabi'I 1436H – January 2015G

Dedication

To my beloved parents, wife, and teachers, who taught me to be ambitious…

To all who supported me to complete this work….

ACKNOWLEDGEMENT

First, I am thankful to Allah for giving me the opportunity to study for my master degree, for giving me the strength to complete this thesis, and for his endless blessing that kept me feeling all the time that he is organizing everything for me.

I would like to express my deepest sense of gratitude to my supervisor Dr. Anas Fattouh and Dr. Fadi Fouz for their patient guidance, encouragement and excellent advice throughout this study. I appreciate their assistance in writing this thesis. Without his help, this work would not be possible.

I would also like to thank my brothers, sisters for the support they provided me through my entire life and in particular, I must acknowledge my parents and my wife Noof, without whose love, encouragement, support and assistance, I would not have finished this work.

I admire the help of many people who offered me valuable support throughout my study. I would like to express my special thanks to Dr. Abdulrahan Hila Altahi, the dean of the Faculty of Computing and Information Technology, and Dr. Aiiad Albeshri, the head of the Computer Science Department, for their encouragement and support.

I express my deepest thanks to my manager Mr. Abdulaziz Ali Alayafi, my colleagues and my friends who support and assistance during this study.

IMPROVED FEATURE EXTRACTION ALGORITHM FOR BRAIN COMPUTER INTERFACE

Abstract

Brain-computer interfaces (BCIs) provide a direct communication between the brain activities and the computer. BCIs are based on detecting and classifying specific activities patterns among brain signals that are associated with specific task or event. However, brain activity patterns are considered as dynamic stochastic processes due both to biological and to technical factors. Therefore, the time course of the generated electroencephalography (EEG) signal should be taken into account during the feature extraction stage. To use this temporal information, three main approaches have been proposed, concatenation of features from different time segments, combination of classifications at different time segments, and dynamic classification. Dynamic classification consists in extracting features from several time segments to build a temporal sequence of feature vectors that can be classified using a dynamic classifier.

In this research work, we propose an improved feature extraction algorithm using Kalman filtering technique. The EEG signal is firstly modeled by a harmonic sum of sinusoidal signals. Then the weights are estimated using a Kalman filter.

TABLE OF CONTENTS

List of Figures

Figure **Page**

List of Tables

List of Symbols and Terminologies

ALS	Amyotrophic Lateral Sclerosis**.**
AR	Autoregressive.
ARMA	Combination of AR & MA.
BCI	Brain Computer Interface.
CWT	Continuous Wavelet Transform**.**
CLIS	Locked-In Syndrome**.**
DWT	Discrete Wavelet Transform.
ECG	Electrocardiograms.
ECoG	Electrocardiogram.
EEG	Electroencephalogram.
EKF	Extended Kalman Filter**.**
EKU	Ensemble Kalman Filter.
EMG	Electromyography.
EOG	Electrooculography**.**
ERDs	Event-Related de-Synchronizations**.**
ERP	Event Related Potential.
ERSs	Event-Related Synchronizations.
FFT	Fast Fourier Transform.
FLDA	Fisher's Linear Discriminate Analysis**.**
FMRI	Functional Magnetic Resonance Imaging.
HMM	Hidden Markov Model.
ICA	Independent Component Analysis**.**
IEKF	Iterated Extended Kalman Filter.

ISI	Inter stimulus interval**.**
KF	Kalman Filter.
LDA	Linear Discriminate Analysis**.**
MA	Moving Average.
MVAAR	Multivariate Adaptive AR**.**
MEG	Magnetoencephalography
MSR	Magnetically Shielded Room.
MRPs	Motor-Related Potentials.
NIRS	Near Infrared Spectroscopy.
PE	Permutation Entropy**.**
PCA	Principal Component Analysis.
PKF	Perturbation Kalman Filter.
PKF	Perturbation Kalman Filter**.**
PSD	Power Spectral Density**.**
SCP	Slow Cortical Potentials.
SNR	Signal-to-Noise Ratio**.**
SSVEP	Steady State Visual Evoked Potentials.
SVM	Support Vector Machines.
SWLDA	Stepwise Linear Discriminate Analysis.
SQUID	Superconducting quantum interference device**.**
UKF	Unscented Kalman Filter.

Chapter 1

Introduction

1.1 An Overview of Brain Computer Interface

The goal of a direct brain–computer interface (BCI) is to allow an individual with severe motor disabilities to have effective control over devices such as computers, speech synthesizers, assistive appliances and neural prostheses [1]. Such an interface would increase an individual's independence, leading to an improved quality of life and reduced social costs [1]. A BCI system detects the presence of specific patterns in a person's ongoing brain activity that relates to the person's intention to initiate control [2]. The BCI system translates these patterns into meaningful control commands. The BCI system has steps or components to interpret signal, which are signal acquisition, feature extraction, feature selection, classification, application and feedback. Feature extraction as the basis of mental pattern is the main content [3].Figure 1.1 shows the stages of a typical BCI system. We give now a short brief for each step and they will be covered in detail in Chapter 2.

- **Signal acquisition**: In this step the brain activities is recorded. The brain activities can be measured in an invasive or non-invasive manner (see types of BCIs next section). Brain activity can be recorded as Electroencephalographic signal (EEG), functional Magnetic Resonance Imaging (fMRI), Positron Emission Tomography (PET) or through other methods. In this thesis, we use scalp EEG measured with an electrode cap. It is the most common acquisition methods. After the acquisition of the signals, the signals are sampled and digitized [4].

- **Signal preprocessing**: Raw EEG data are very noisy signal. The goal of this step is to increase the Signal-to-Noise Ratio (SNR). Preprocessing can include re-referencing, artifact rejection and band-pass filtering [5].

- **Feature Extraction**: We want to extract the features of the signal. These should contain the proper information of the signal. A common procedure during feature extraction is spatial filtering. Feature Extraction reduces the dimensionality of the problem. The main goal of this thesis to improve features extraction method. To select the most appropriate classifier for a given BCI system, it is necessary to simply understand what features are used, what their properties are and how they are used. The design of a BCI system, some crucial properties of these features must be taken into accounts: noise and outliers, high dimensionality, time information, non-stationary, small training sets [6].

- **Classification**: Based on the features a decision regarding the intention of the user has to be made in the final classification step. The classifier will translate the feature vector into a simple command [7].

- **Applications and feedback**: Based on the classification outcome we can now give an instruction to an external device as shown in Figure 1.1.

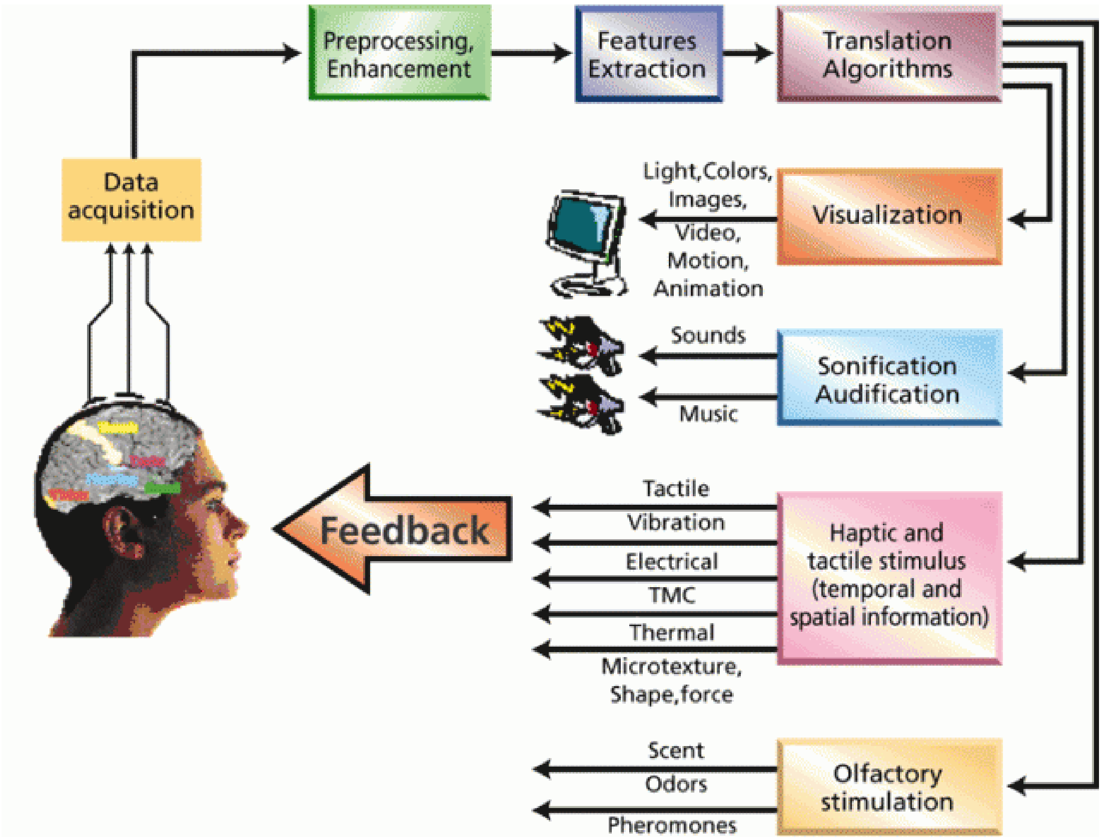

Figure 1.1. General signal processing flowchart of a brain–computer interface [4].

1.2 Types of Brain Computer Interface

There are three types of Brain Computer Interface (BCIs) as shown in Figure 1.2. BCI depends on many factors such as the acquisition method, how the subjects are trained, how the signal is processed or based on the output.

1. **Invasive BCIs**: The electrodes are placed directly in the grey matter. These BCIs are thought to record the most pure signals, since they are directly connected to single neurons. The direct connection ensures that there will be no attenuation nor spreading of the signal. Indeed, in practice some good results have been obtained concerning vision

repair. However, in case an invasive BCI is applied, there is a high risk of creating scar tissue around the electrodes that might lead to malfunction. Because of the invasive procedure and the need for a personalized system, the overall cost will be much higher than the cost of a non-invasive BCI [8].

2. **Partially Invasive BCIs**: The electrodes are still placed under the skull. Instead of placing them inside the grey matter, they are now placed at the surface of the grey [8].

3. **Non-Invasive BCIs**: The interfaces used nowadays are in most cases non-invasive methods. These use an electrode cap placed over the head to record the brain potentials. This reduces the risk of medical problems significantly. The high temporal resolution is preserved, making real time applications possible. On the contrary, the spatial resolution of non-invasive BCIs is quite low. This is because the signals now first have to pass the low conductive skull before being measured. The system however is wearable and not too expensive with no medical risks. One of the main disadvantages is the extensive training often necessary before the user can use the interface optimally. Even after training, accuracy might still leave much to be desired. In this thesis, we will only address non-invasive BCIs based on scalp EEG [9].

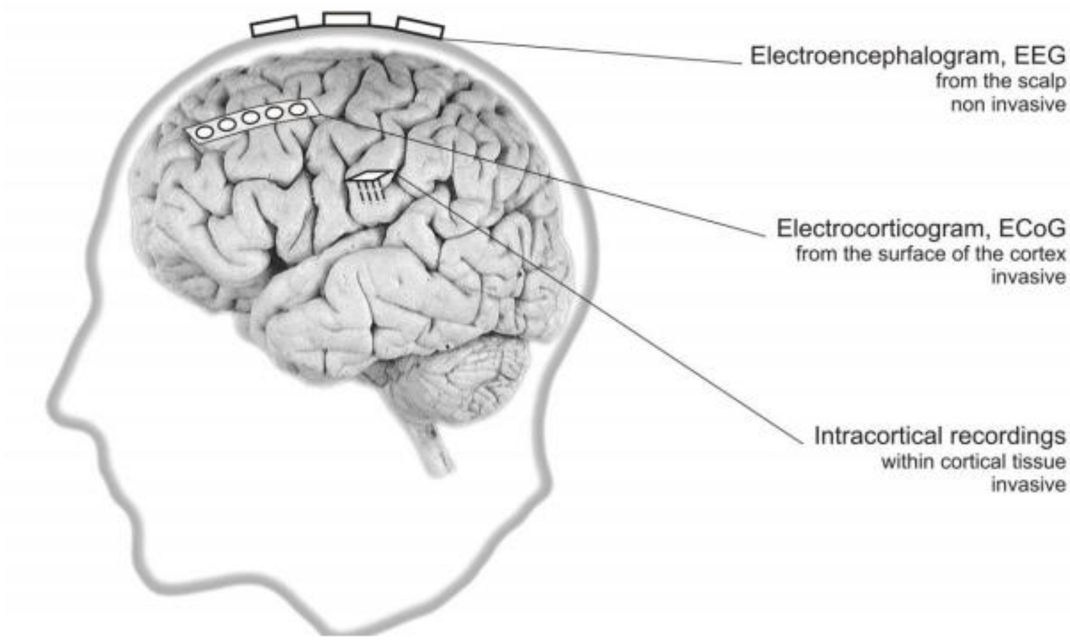

Figure 1.2. Types of detect the brain's electrical activity: EEG, ECoG, and intracranial recordings [2].

1.3 Motivation and Problem Statement

Brain-computer interfaces (BCIs) provide a direct communication between the brain activities and the computer [2]. BCIs are based on detecting and classifying specific activities patterns among brain signals that are associated with specific task or event [10]. However, brain activity patterns are considered as dynamic stochastic processes due both to biological and to technical factors [11]. Therefore, the time course of the generated electroencephalography (EEG) signal should be taken into account during the feature extraction stage. To use this temporal information, three main approaches have been proposed, concatenation of features from different time segments [12], combination of classifications at different time segments [7], and dynamic classification [2]. Dynamic classification consists in extracting features from several time segments to build a temporal sequence of feature vectors that can be classified using a dynamic classifier.

In this research work, we propose an improved feature extraction algorithm using Kalman filtering technique. The EEG signal is firstly modeled by a harmonic sum of sinusoidal signals. Then the weights are estimated using a Kalman filter

1.4 Research Objectives

The main objective of this work is to improve feature extraction algorithm using Kalman filtering technique. The proposed algorithm will be implemented on binary steady-state visual evoked potentials (SSVEP) BCI system. Thus the research objectives are:

1. Understanding in detail the feature extraction algorithms of EEG signals.
2. Developing an improved feature extraction algorithm.
3. Implementing a prototype as a proofing of the concept.
4. Compare the performance of a BCI-based system proposed feature extraction algorithm using Kalman filter technique with other algorithm Fast Fourier Transform (FTT).

1.5 Thesis Organization

The rest of this thesis is organized in five chapters as follows. **Chapter 2** will be an introduction to Brain Computer Interface. This will include in detail feature extraction algorithm BCIs. **Chapter 3** will be dedicated to Kalman filter technique in estimating the state of a noisy system.

Chapter 4 describes how to employ the Kalman filter technique in extracting the features of a SSVEP based BCI. **Chapter 5** will present and discuss the results of applying the proposed method on a SSVEP based BCI. **Chapter 6** gives a conclusion and an outlook on future work.

Chapter 2

Review of Literature

2.1. Introduction

In this chapter, we want to provide a detailed background of the mechanism used in BCI applications. Figure 2.1 shows a typical BCI system framework. In general, the sequence of events in a BCI system is as follows. The brain signal is recorded employing a signal acquisition device. These signals are then converted from analog to digital using an amplifier and feed to a computer. After that, pre-processing is performed to get rid of unnecessary data like noises and artifacts. Features that are relevant for recognizing different mental activities are then extracted, and classification algorithms are used to recognize that activity is performed by the user. The result of the classification is then translated into commands and is employed to regulate an application [13].

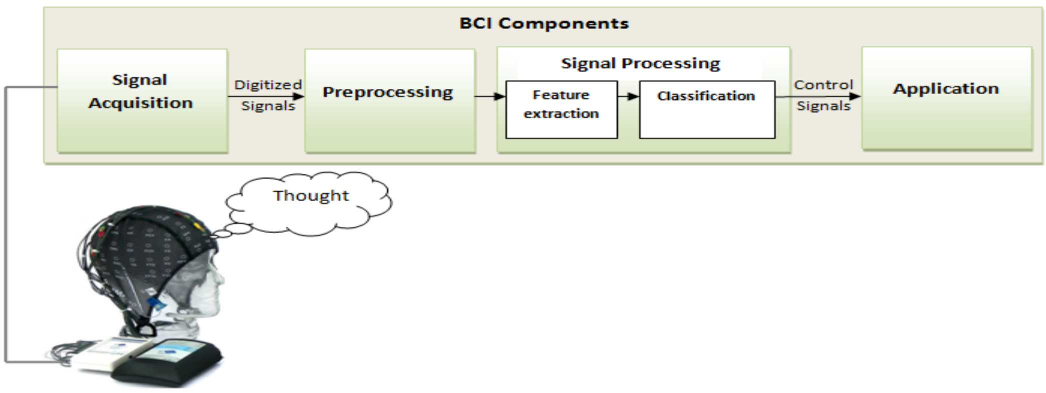

Figure 2.1: Basic block diagram of BCI system.

As mentioned in the chapter 1, the BCI system has stages to interpret signal, which are signal acquisition, feature extraction, feature selection, classification, application and feedback. Therefore, in Section 2.2, we give the neuroimaging methods use in BCIs. Then, in Section 2.3, we analyze the most neuroimaging method, which is EEG in BCI systems. After that, we review signal acquisition stage used for recording brain activities in Section 2.4. In addition, we analyze EEG signal in Subsection 2.4.1, Steady State Visual Evoked Potentials (SSVEP) in Subsection 2.4.2 and we discussed Oscillatory Brain Activity in Subsection 2.4.3. Pre-processing stage are studied in Section 2.6.An outline of the method feature extraction stage and its methods are studied in Section 2.7.

2.2 Neuroimaging Methods in BCIs

Physiological activities in the human body, including those occurring in the brain, can be directly measured through electrophysiological signals such as those caused by the aforementioned action potentials. Those include electrocardiography (ECG, heart), electroencephalography (EEG, brain), electromyography (EMG, brain and muscular system), magnetoencephalography (MEG, brain), electrogastrography (EGG, stomach) and electrooptigraphy (EOG, eye dipole field). Neuroscientists use a type of sensing methods to measure brain signals. Some of methods, which are usually used, are EEG (invasive and noninvasive), magnetoencephalography (MEG), positron emission tomography(PET), function magnetic resonance imaging (fMRI) and functional Near Infrared (fNIR) .The three techniques which are used to measure brain activity (as opposed to brain structure) are MEG, fMRI and EEG. Each of these methods has its own unique advantages and disadvantages. We give short description for MEG, fMRI provided and full description of EEG method because it most common used for BCI and we used it in this thesis [2, 4]:

- MEG maps brain activities by recording magnetic fields produced by the electrical activities in the brain. MEG needs expensive and intensive low noise amplifier called superconducting quantum interference device(SQUID), furthermore the measurements are sensible to ferromagnetism therefore MEG equipment should be isolated inside Magnetically Shielded Room (MSR) where MSR will isolate SQUID from all external magnetic field even Earth's magnetic field which is billion time stronger than the raw MEG. MEG is known for having very high temporal and spatial resolution and can be useful for studying activities that take less than 10 milliseconds. Unfortunately, in terms of BCI, MEG has two very serious problems. Firstly, it is extremely expensive, with MEG devices often costing hundreds of thousands of dollars or more. Secondly, MEG devices are very big and are not suitable for ambulatory applications such as BCI [2].fMRI (functional magnetic resonance imaging) uses nuclear magnetization of the hydro atoms in the fluids, mainly the blood, to adjust a powerful magnetic field. Because fMRI depends on the fluids moves in the body tissues, it will be more helpful for slow events around many hundred milliseconds. Since of this and other reasons, fMRI is unusually used for BCIs [2].

- EEG signals are obtained by recording fluctuations in the local electric potentials on the surface of the scalp, where it is assumed that these fluctuations originate from the underlying human brain activity. Although EEG contains more noise, EEG signal has low SNR, than MEG and fMRI, EEG is the most used techniques in BCI that represents more than 80% of BCI published work where EEG has very low setup cost and is very portable. The EEG rhythm contains much interesting information. For example, each frequency band of the EEG signal is associated with certain brain activities. Neuroscientists have associated each of these frequency bands with a specific set of mental activities or states [2].The next section EEG will be explained in detail.

2.2.1 EEG analysis

EEG is a non-invasive recording method in which electrical components of the electromagnetic domain of the brain generated by neuronal activity are measured. Since its discovery by Hans Berger [6], the EEG has been used to evaluate neurological disorders in the clinic and to investigate brain function in the laboratory. Over this time, people have speculated that the EEG could have a fourth application as it offers the possibility of a new non-muscular communication and control channel (a practical BCI). The most important advantages of the EEG method that also make it commonly used in BCI are it’s relatively short time constants, its functionality in most environments, and its relatively simple and inexpensive equipment [2, 7].

The EEG signal is usually recorded at many brain locations simultaneously using one electrode (sensor) at each position (the term channel is often used to refer to a recording position). These electrodes are stuck to the scalp with a conductive gel in order to improve the contact impedance between the skin and the electrodes. A set of differential amplifiers (one for each channel) are then used to digitize the signals [10]. For the application of a larger number of electrodes, an electrode cap is often used Figure 2.2. The distance between neighboring electrodes is usually in the range of one to a few centimeters and available EEG caps can record up to 128 channels.

EEG recordings exhibit adequate time resolution but suffer from disadvantages that have mostly caused by the skull bone, the meanings, and the intra-cerebral liquor. These layers cause the signals from a local ensemble of neurons to spread to scalp electrodes that are up to 10 cm away from the recording electrode. A very effect of these layers is that a low-amplitude activity at frequencies of more than 40 Hz is practically invisible in the EEG. Therefore, it is difficult to use the EEG to record the activity of single neurons or even of a small brain region. Moreover, the analysis of the EEG is also complicated due to the presence of artefacts that are signal components picked up by EEG electrodes and are not caused by neural activity. Typical artefacts in EEG comprise muscle activity, movements of the eyeball, eye blinks and the stray pick-up from exterior signal sources [13].

Figure 2.2: An EEG cap for the use of a large number of electrodes.

As artefacts have much larger amplitude than the signals of interest, it has to be removed before EEG signals analysis. The fact that artefacts are picked up with highest intensity at electrodes closest to their origin can help in identifying them. Most artefacts can be controlled using additional control electrodes close to possible artefact locations, by proper frequency filtering of the recorded signals, and by using digital signal processing algorithm [12].

Another important issue with the EEG signals that must be considered is its non-stationary. Non-stationary of the signal is a considerable variation in its statistics at different time lags. In general, during normal brain condition the multichannel EEG distribution is considered as multivariate Gaussian. However, the mean and covariance statistics change from segment to segment, and this is the first symptom of non-stationary. The second symptom appears due to the change in the distribution (itself) of signal segments (i.e. Away from Gaussian). This can be observed, for example, during the changes in the oscillatory brain activity, during the transition between physiologic states, during eye blinking, and in the event-related potential (ERP) signals. The non-Gaussianity of the signals can be checked by some measures such as skewness, negentropy, kurtosis, and Kulback-Laibler (KL) distance [7]. Even with the aforementioned shortcomings, EEG is still the most interesting recording method of BCI systems and other clinical and research applications [2, 10, 13].

2.3 Signal Acquisition Stage

There are different types of features of the ongoing EEG signals, relying on different physiological activities related to human brain. There are two main classes of these features. The first is time- and phase-locked (evoked) to an externally or internally paced event. This class is based on the responses of the subject to some stimuli and it is known as Event Related Potentials (ERPs), including the P300, steady-state visual evoked potentials (SSVEPs), and Motor-Related Potentials (MRPs). The second class is also time-locked but not phase-locked (induced) where the subject regulates the brain activity by concentrating on specific mental tasks. For example imagination of hand movement which can be applied to modify activity in the motor cortex. This class includes the event-related de-synchronizations (ERDs) and event-related synchronizations (ERSs). These two classes as well as the most frequently used features (for BCI purpose) which are firs Event Related

Potentials (ERPs) are specific patterns generated by the brain of the subject after or during the presentation of preselected visual and/or audio stimuli. These patterns can be detected by analyzing the recorded EEG signals and can be specified which stimulus among a larger set of possible stimuli has drawn the subject's attention. ERPs were initially developed for environment control. They are mainly proposed for disabled subjects who are unable to interact with outside world thoroughly their neuromuscular pathways. ERPs include P300 patterns, Steady State Visual Evoked Potentials (SSVEP) and motor-related potentials (MRPs), which also known as slow negative potentials or slow cortical potentials (SCP). However, only the SSVEP type of patterns will be described here.

2.3.1 Steady State Visual Evoked Potentials

Steady-state visual evoked potentials (SSVEPs) are oscillations in the EEG that are generated in the visual cortex when a subject views a periodically flickering stimulus. An interesting characteristic of these oscillations is their amplitude, which can be modulated by visual attention. Subjects can increase the amplitude of the SSVEPs by concentrating on the stimulus or decrease the amplitude by ignoring it. Hence, SSVEP is employed in BCI applications by the presentation of several flickering light sources with different frequencies. In such a paradigm, the focused light elicits a signal pattern of the same frequency or harmonics with that of the source. Therefore, an SSVEP based BCI system can be realized by the detection of the focused light sources from these signal patterns. As an example, a wheelchair can be controlled by using only four light sources to perform a movement on the main directions [8].

2.3.2 Oscillatory Brain Activity

Physiologically significant signal features can be extracted from changes in the oscillatory brain activity. Such changes can be evoked by presentation of stimuli by concentration of the user on a specific mental task. Various frequency bands are related to changes in the amplitude of oscillatory activity. These frequency bands are shown in Table 2.1. For example, in systems based on motor imagery, movement or preparation for movement is typically accompanied by a power decrease in mu and beta frequency bands, particularly contra lateral to the movement. This means that imagination of left hand movement corresponds to a decrease in mu-band amplitude over the right sensorimotor cortex, whereas imagination of the right hand movement corresponds to a decrease in mu-band amplitude over the left sensorimotor cortex. This decrease in the band power has been labeled as event-related de-synchronization (ERD). In contrast, the increase in the amplitude of mu and beta bands after a movement indicates relaxation and is due to synchronization in firing rates of large populations of cortical neurons. This increase has been labeled as event-related synchronization (ERS) see Figure (2.3) [2, 5].

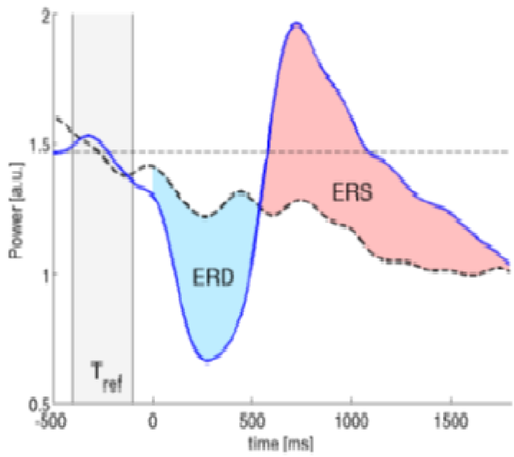

Figure 2.3: ERD and ERS [2].

Table 2- 2: Characteristics of normal EEG rhythms

Brain Rhythm	Frequency	When and where it can be found.
delta - δ	1 - 4 Hz	• Dominant in infants, during deep stages of adult sleep, and serious organic brain disease. • Observed in central cerebrum and parietal lobes.
theta - ϑ	4 - 8 Hz	• In drowsy normal adult, and in children when awake. • Observed in frontal, parietal and temporal regions.
alpha - α	8 – 13 Hz	• The most prominent rhythm in the normal alert adult brain. • Most prominent at occipital and parietal regions and about 25% stronger over the right hemisphere.
mu - μ	8 – 13 Hz	• These oscillations are decreased in amplitude when movements of body parts are imagined or performed. • It can be localized over the part of the sensorimotor cortex corresponding to the body part.
beta - β	13 – 25 Hz	• As mu oscillations, beta oscillations are also decreased in amplitude when movements of body parts are imagined or performed. • It can be also localized over the part of the sensorimotor cortex corresponding to the body part.
gamma - γ	25 – 40 Hz	• It has a role in sharing information and in segmenting information during a working memory task. • It can be localized over different parts of the visual cortex.

Moreover and mainly related for BCI use, ERD and ERS do not require actual movement; they occur also with motor imagery (i.e. imagined movement). Thus, they might support an independent BCI [2]. However, these systems require a long training period for the subject to obtain a successful performance. The subject is required to learn to regulate his brain activity with feedback mechanisms in these training sessions [2,10,13].

2.4 Preprocessing Stage

The raw EEG signals usually contain frequency components of up to 300 Hz due to noise and artefacts. However, neural information often lies below 100 Hz (and in many application lies below 30 Hz). Hence, components above these frequencies are considered as undesired components and must be filtered out. By removing the undesired frequencies, we retain the effective information in the signal, reduce the noise, and make the signals suitable for processing and classification. The undesired frequencies or components in EEG signal are usually due to noise and artefacts associated with the signal. EEG noise and artefacts are generated either within the brain (patient-related or internal artefacts) or over the scalp (system or external artefacts). The internal artefacts are usually related to EOG signals (electro-oculogyric) which monitor eye blinking, the ECG signals (electrocardiograms) which monitor heart electrical activity, the EMG signals (electromyogram) which monitor muscles electrical activity, and possibly the sweating process. On the other hand, the system or external artefacts include the 50/60 Hz power supply interference, electrical noise from the electronic components, cable defects, unbalanced impedances of the electrodes, and impedance fluctuation. Most of these artefacts are filtered out by the hardware provided in new EEG machines. However, usually a remaining part of artefacts needs to be removed [2].

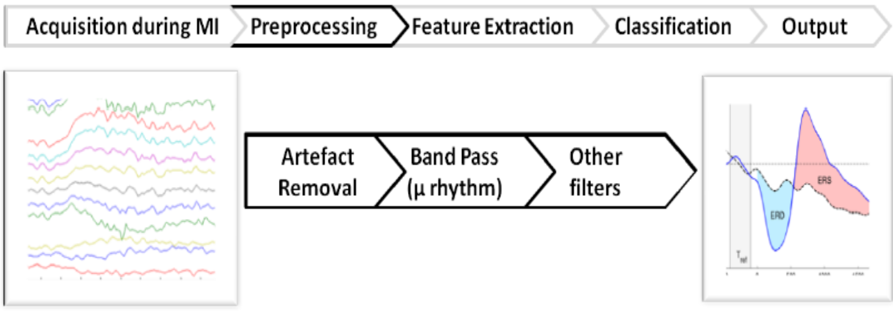

Figure 2.4: preprocessing stage [2].

In general, the filtering algorithms can be divided into adaptive and non-adaptive filters. The main examples of the non-adaptive filters are high pass filters, low pass filters, and Notch filters. The high pass filters with a cut-off frequency of usually less than 0.5 Hz are used to remove the very low frequency noise such as those of breathing. On the other side, high frequency components are reduced by using low pass filters with a cut off frequency of approximately 50-70 Hz. Notch filters, however, with a null frequency of 50 Hz are usually necessary to ensure removing of the strong 50 Hz power supply [13].

The adaptive noise filters are also used by many researchers to remove noise and artefacts from the EEG signals. However, an effective adaptive filter requires usually reference electrodes during the EEG recordings. The reference electrodes carry significant information about the noise or artefact. For example, in the removal of eye blinking and (EOG) artefacts, a signature of eye blink and (EOG) signals can be captured from the FP1 and FP2 electrodes. In the detection of possible jaw and neck muscle activity, as another example, the (EMG) signal can be captured from the two front-temporal electrodes (FT9, FT10) and the two occipital electrodes (O9, O10). The most fundamental type of adaptive filters is the Wiener filter [5, 7 , 13].

2.5 Feature Extraction Stage

Different thought actions produce in varying patterns of brain signals. BCI is recognized as a pattern recognition system that assigns each pattern in a class corresponding to its features. BCI extracts some features from brain signals that reveal similarities to a certain class as well as contrast from the rest of the classes. The features are measured of the attributes of the signals that contain the discriminatory data interested to separate their different kinds. The design of a proper set of features is a challenging issue. The data of interest in brain signals is hidden in a highly noisy environment, and brain signals comprise a huge number of Synchronous sources. A signal that interested may be overlapped in time and space by many signals from several brain tasks. Because of this reason, in more than cases, it is not just to use Easy methods as a band pass filter to select the desired band power. Brain signals measure in many channels. No need for all information provided by the measured channels is generally appropriate for now the underlying events of interest. Dimension reduction methods such as principal component analysis or independent component analysis can be used to decrease the dimension of the real data, remove the unnecessary and irrelevant information. Computational costs are then reduced. Brain signals are naturally non-stationary. Time information about when a certain feature occurs should be taken. Some approach divides the signals into short segments and the parameters can be estimated from each segment. However, the segment length influences the accuracy of estimated features. Multiples features are extracted from many channels and from many time segments before being concatenated into one feature vector. The main difficulties in BCI design is selecting relevant features from the large number of possible features. High dimensional feature vectors are not desirable because of the "curse of dimensionality" in training classification algorithms [11].

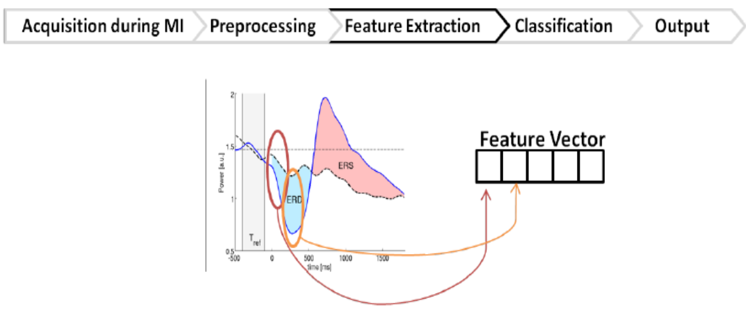

Figure 2.5: Feature Extraction [2].

2.5.1 Features Extraction Methods

As described, above the neurophysiologic features of the brain signals. In order to control a BCI system, these features have to be mapped to values that allow for easy discrimination of different classes of brain signals. The classified signals in turn should be translated into simple commands for a computer or other devices. However, if more than one feature is used for the discrimination, it is impossible for a human to specify an optimal mapping between signals and commands. Furthermore, neurophysiologic signals vary from person to person. Hence, it is necessary to specify mapping rules for each subject, wants to use a BCI, individually [11, 13].

To solve these problems, most BCI systems acquire labeled training data from a subject. Then, a computer is used to learn from a set of training examples how to map signals to desired commands. This technique called supervised machine learning. The term "supervised learning" comes from the idea that a teacher or supervisor indicates the desired output, or command, for each training input example. Machine learning algorithms are usually divided into feature extraction and classification modules. The feature extraction module aims to transform raw EEG signals from time series into another representation that makes classification easy. The new representation usually removes unnecessary information from the signals and retains information that is important to discriminate different classes of signals. After feature extraction, machine-learning algorithms are used to infer specific mapping between the labeled feature vectors, produced by the feature extraction module, and classes. We only consider supervised machine learning algorithms. All feature extraction methods summaries in Tables based on its domain, So Table 2.2 shows dimensional reduction methods, like principal component analysis or independent component analysis are explained. In a Table 2.3 time and/or frequency methods, like matched filtering or wavelet transform, and parametric modeling, like autoregressive component. In Table 2.4, spatial pattern algorithms are an explained. Feature extraction methods are one of the main themes of this thesis [11].

Table 2.2: Summary of Feature extraction Method Spatial Domain [11].

Method	**Properties**	**References**
PCA (Principal Component Analysis)	• Linear transformation • Set of possibly correlated observations is transformed into a set of uncorrelated variables • Optimal representation of data in terms of minimal mean-square-error • No guarantees always a good classification • Valuable noise and dimension reduction method. PCA requires that artifacts are uncorrelated with the EEG signal	[14]
ICA (independent component analysis)	• Splits a set of mixed signals into its sources Mutual • statistical independence of underlying sources is assumed • Powerful and robust tool for artifact removal. Artifacts are required to be independent from the EEG signal • May corrupt the power spectrum	[15]

Table 2.3: Summary of Feature extraction Method Time Frequency Domain [11].

AR (Autoregressive Components)	• Spectrum model • High frequency resolution for short time segments • Not suitable for non-stationary signals • Adaptive version of AR: MVAAR	[16]
MF(Matched Filtering)	• Detects a specific pattern on the basis of its matches with • predetermined known signals or templates • Suitable for detection of waveforms with consistent temporal characteristics	[17]
CWT (Continuous Wavelet Transform)	• Provides both frequency and temporal information • Suitable for non-stationary signals	[18]
DWT (Discrete Wavelet Transform)	• Provides both frequency and temporal information • Suitable for non-stationary signals • Reduces the redundancy and complexity of CWT	[19]

Table 2.4: Summary of Feature extraction Method Space Domain [11].

Method	Properties	References
CSP (Common Spatial Pattern)	• Spatial filter designed for 2-class problems. Multiclass extensions exist • Good result for synchronous BCIs. Less effective for asynchronous BCIs • Its performance is affected by the spatial resolution. Some electrode • locations offer more discriminative information for some specific brain activities than others Improved versions of CSP	[20]

In Section 2.3, we described the neurophysiologic features of the brain signals. In order to control a BCI system, these features have to be mapped to values that allow for easy discrimination of different classes of brain signals. The classified signals in turn should be translated into simple commands for a computer or other devices. However, if more than one feature is used for the discrimination, it is impossible for a human to specify an optimal mapping between signals and commands. Furthermore, neurophysiologic signals vary from person to person. Hence, it is necessary to specify mapping rules for each subject wants to use a BCI individually. We will explain domains as follows [7, 13]:

- **Spatial Domain Analysis**

Most BCI systems work with multivariate time series, i.e. data from more than one electrode is available for analysis. Therefore, the features extracted from those electrodes should be combined efficiently for the discrimination of a given set of cognitive task. Thus, the goal of spatial domain analysis methods is to find efficient combinations of features from more than one electrode. Actually, there are two main ways for performing spatial domain analysis. The first way is to use a subset of all available electrode positions that carry the informative features for a classification task. This approach depends on the fact that changes in neurophysiologic features (such as changes in SSVEP peaks) are often stronger at electrodes over brain regions implying a related cognitive task. Optimal electrode subset can then be selected manually (without performing any computations), or by using one of the expert algorithms developed in the literature [7].

The second way to perform spatial analysis, instead of choosing a subset of electrode position, consists of applying spatial separating (filtering) algorithms. The most common separating algorithm is the independent component analysis (ICA). ICA algorithm is an iterative technique used to separate multichannel signals in to several components corresponding to statistically independent sources (brain or noise). Hence, by retaining only components that have informative features, classification accuracy can be improved. The obvious drawback of this method is when the number of sources becomes more than the number of electrodes or observations (known as underdetermined systems). In such a system, the ICA method cannot be applied, and generally, the original sources cannot be extracted. One solution to this problem is to utilize clustering based methods when the signals are sparse [9, 10, 13].

- Frequency Domain Analysis

Changes in oscillatory activity discussed in Section 2.3.2 are usually not time-locked to the presentation of stimuli or to actions of the user. Hence, time domain analysis methods cannot be used to reveal this kind of features. Instead, methods that are invariant to exact temporal evolution of signals should be used. Therefore, signals should be transformed from time domain to frequency domain representation. This representation is useful for estimating the power spectral density (PSD) of the signal that is an important characteristic that can be used to identify oscillatory activity components. The two main groups of methods for frequency transformation are developed in the literature include Fourier methods and parametric methods [9, 10, 13].

The Fourier group contains methods that are based on the fast Fourier transform (FFT) such as the periodogram, the Welch method, and the multi-taper method [8]. However, these methods are not practical for BCI systems. This is because time series analyzed for such systems are typically very short, where Fourier methods can give reasonable results only for long signal sequences and the performance usually deteriorates with shorter sequences [8]. On the other hand, the parametric group contains methods such as autoregressive (AR) method, the moving average (MA) method, or the combination of these two methods

(ARMA). However, the autoregressive method is often applied in BCI systems since it seems to be sufficiently powerful to model typical rhythmic and broadband brain activity [9, 10, 13].

The idea behind all parametric methods is to employ priori assumptions regarding the generating random process. Depending on these prior assumptions, a model class and model order can be chosen in order to estimate the PSD, and hence capture the signal characteristics. In general, parametric methods are superior for estimating PSD than Fourier methods since they can work efficiently even with short time series. Moreover, modeling of a time series using a parametric method itself is a strong reduction in dimensionality as well as the noise of the EEG signals. However, some informative data may be lost during this modeling process, which is considered as a drawback of the parametric methods. Furthermore, the training of the AR model, which often be used with BCI systems as mentioned above, does not incorporate knowledge about the discriminative value of the information. This may, in principle, case a problem for a following classification task. To avoid this problem, the optimal AR model order and, therefore, the compression rate, have to be determined using validation techniques [9, 10, 13].

- Time Domain Analysis

We often choose to analyze EEG signals in time domain if the amplitude of the neurophysiologic signals changes over time. Such change usually occurs time-locked to the presentation of stimuli or time-locked to actions of the user of a BCI system. SSVEP and MRPs are two valid examples for signals that can be characterized with the help of time domain features. Analyzing an EEG signal in time domain in order to reveal neurophysiologic changes is straightforward. Time series features, such as the following, can easily be computed:

- The average of the signal (offset).
- The linear trend of the signal.
- Absolute minimum and maximum values.
- Number and order of local minimum and maximum values.
- Weight factors describing the matching and positions of predefined patterns.
- Slopes/steepness/height/width of predefined patterns.

Most of these time domain features cannot be observed in single trial studies and can be clearly extracted only by averaging many trials over temporal windows or channels. In addition, the averaging strategy helps to reduce dimensionality and noise from EEG signals. However, averaging, particularly over channels, shift the analysis away from the brain enforcing inferences about summary measures. This leads to uncertainty about how signals should be analyzed and generated, and what they tell us about the underling system. Therefore, time domain features that depend on averaging methods can be useful for BCI only in combination with good classification algorithms. [9, 10, 11].

2.5.2 Dynamic Systems

A dynamical system is defined as the system that changes its state over time, frequently in a rather complex manner. Understanding, processing, and classifying such changes is of greatest importance for the analysis of EEG signals. Formally, a dynamical system is given by a phase space, a continuous or discrete time, and a time-evolution law (also called system dynamics).The elements or points that represent possible states of the system are called state variables and the space made up of the state variables is called phase space or state space. The state of a system may be described by $\boldsymbol{m}$ variables, and thus it can be represented by a point in an $\boldsymbol{m}$-dimensional phase space. Let us assume that the state of such a system at a fixed time $\boldsymbol{t}$ can be specified by $\boldsymbol{m}$ variables. These parameters can be considered to form a vector

$$\vec{x}(t) = (x_1(t), x_2(t), \dots \dots, x_m(t))^T \tag{2.1}$$

Time-evolution law allows calculating all future states given a state at any particular moment. For time-continuous systems, the time evolution equations consists of a system of coupled differential equations, one for each of the systems variables.

$$\dot{\vec{x}}(t) = \frac{d\vec{x}(t)}{dt} = \vec{F}\big(\vec{x}(t)\big), \quad F: R^d \to R^d \tag{2.2}$$

The vectors $\vec{x}(t)$ (i.e. the line connecting system states) define a trajectory in phase space, which is a path followed by a dynamical system as time progresses [9, 19]. A dynamical system may be a linear system if all the equations describing its dynamics are linear; otherwise, it is nonlinear. On the other hand, a dynamical system can be deterministic if the equations of motion (which every future state of the system must follow) are predefined and stochastic otherwise. However, the neural networks of the brain, which is of prime concern to us in the present context, are likely to be a chaotic system [19]. The important features of such a system is its nonlinearity and deterministic. Although chaotic systems are kind of systems that are deterministic, their behavior shows sustained irregularity.

An important property of the chaotic systems is that, after long observation, the trajectory will converge to a subspace of the total phase space. This subspace is called the attractor of the system since it 'attracts' trajectories from all possible initial conditions .The attractor, in chaotic systems, is a very complex object with fractal geometry [9, 19].

2.6 Signal Classification Stage

The features extracted in the previous stage are the input for a classifier. The goal of the classification step is to determine the mental state of an individual. Based on that classification a command can be given to an external device. Therefore, the classification algorithm takes the abstract feature vector that reflects specific aspects of the current state of the user EEG and transforms that vector into an application-dependent device command. In certain cases, the classification can simply be done by comparing the signal resultant from the preprocessing step to a threshold. Other possibilities are the use of linear classifiers such as Linear Discriminate Analysis (LDA) or Fisher LDA classifiers. Another very popular method is to use neural network methods. These are more complex and non-linear techniques. The most common examples are Support Vector Machines (SVMs) and Hidden Markov Models (HMMs). Moreover, one can choose between an adaptive and a non-adaptive classifier. We will discuss simpler Bayesian linear discriminate analysis (BLDA) algorithm, as we use it for classification in this thesis [2].

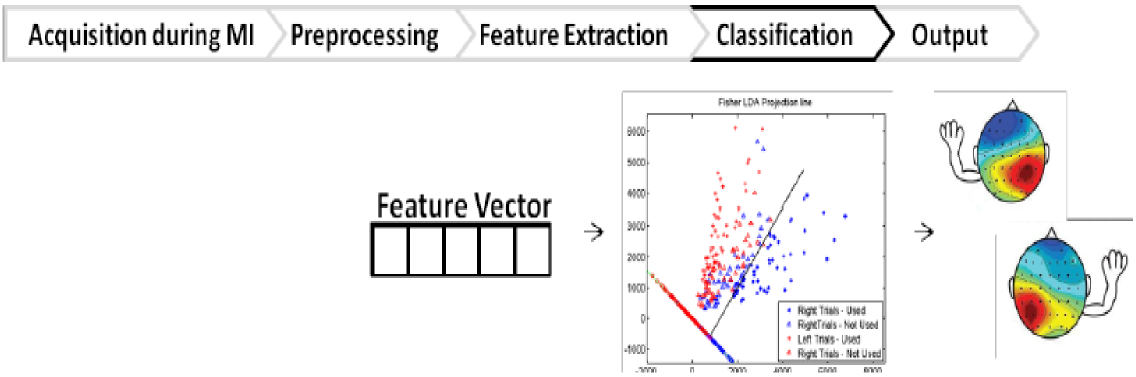

Figure 2.6: Classification stage [2].

2.6.1 Fisher's LDA

The main goal in Fisher's linear discriminate analysis (FLDA) is to compute a discriminate vector that separates two or more classes as accurate as possible [9]. In this thesis, we only consider the two-class case because our aim in SSVEP-based BCI applications is to discriminate between EEG signals contain SSVEP property and EEG signals do not contain it. We are given a set of input vectors $\mathbf{x_i} \in \mathbb{R}^{\mathbf{D}}, \mathbf{i} \epsilon \{\mathbf{1}, \dots \dots \dots, \mathbf{N}\}$ and corresponding class-labels$\mathbf{y_i} \in \{-\mathbf{1}, \mathbf{1}\}$. Denoting by $\mathbf{N_1}$ the number of training examples from the first class (for which $\mathbf{y_i} = \mathbf{1}$), by $\mathbf{C_1}$ the set of indices $\mathbf{i}$ belonging to the first class, and using analogous definitions for $\mathbf{N_2}$, $\mathbf{C_2}$, the objective function for computing a discriminant vector $\boldsymbol{\omega} \in \mathbb{R}^{\mathbf{D}}$ is

$$J(\omega) = \frac{(\mu_1 - \mu_2)^2}{\sigma_1^2 + \sigma_2^2} \tag{2.1}$$

where

$$\mu_k = \frac{1}{N_k} \sum_{i \epsilon c_k} \omega^T x_i \,, \qquad \sigma_k^2 = \sum_{i \in c_k} (\omega^T x_i - \mu_k)^2 \tag{2.2}$$

This means that we are searching for a discriminate vector that yields a large distance between the projected means and small variance around the projected means (small within-class variance). Matrix equations for the quantities $(\mu_1 - \mu_2)^2$ and $\sigma_1^2 + \sigma_2^2$ can be used in order to compute the optimal discriminant vector for a training data set. Hence, we need first to define the class means $\mathbf{m_k}$ as following:

$$m_k = \frac{1}{N_k} \sum_{i \in C_k} x_i \tag{2.3}$$

Then, we can define the between-class scatter matrix $\mathbf{S_B}$ and the within-class scatter matrix $\mathbf{S_W}$.

$$S_B = (m_1 - m_2)(m_1 - m_2)^T \tag{2.4}$$

$$S_W = \sum_{k=1}^{2} \sum_{i \in C_k} (x_i - m_k)(x_i - m_k)^T \tag{2.5}$$

With the help of these two matrices, the objective function for computing the discriminate vector can be written as

$$J(\omega) = \frac{\omega^T S_B \omega}{\omega^T S_W \omega} \tag{2.6}$$

Then, by computing the derivative of **J** and setting it to zero, we can show that the optimal solution for **ω** satisfies the following equation:

$$\omega \propto S_W^{-1}(m_1 - m_2)$$

The main advantages of FLDA are its conceptual and computational simplicity, especially for the situation in which the number of training examples **N** is large and the number of features **D** is small (i.e. $\mathbf{D} < N$). However, we run into problems if other cases occur. If the number of training examples **N** becomes smaller than the number of features **D** (i.e.$\mathbf{D} > N$), then the within-class scatter matrix $\mathbf{S_W}$ becomes singular and cannot be inverted. A simple solution for this problem is to replace the inverse $\mathbf{S_W^{-1}}$ by the Moore-Penrose pseudo-inverse $\mathbf{S_W^{\dagger}}$ [10], and the optimal solution for ω then reads:

$$\omega \propto \mathrm{S}_{\mathrm{W}}^{\dagger}(\mathrm{m}_1 - \mathrm{m}_2) \quad (2.8)$$

On the other hand, if the number of features **D** is nearly as big as the number of training examples **N** over-fitting occurs. This situation is often found in BCI applications [1], because data from BCI experiments usually contains outlier, resulting from, for example, muscle activity or eye-blinks, and therefore there is an increased tendency for over-fitting. One solution to this problem is to use a regularized version of FLDA [13].

2.7 BCI Applications

The main objective of a BCI is to detect small differences in brain signals and use these to steer an external device. In principle this external device can be anything, as so can be the input causing the change in brain signal. However, the input is generally limited to some typical tasks intended for subject training. These tasks include (limited) cursor control, motor imagery, tracking a moving object or selecting a target. The results of these tasks can then be translated into more useful applications in the field of communication environmental control or neural prosthetics. As shown in Figure 1.10, the kind of application will on the one hand depend on the severity of the locked-in state. A distinction is made between Complete

Locked-In Syndrome (CLIS) and LIS patients, and healthy subjects. On the other hand, it will depend on the Information Transfer Rate (ITR) of the BCI-system. This is a measurement for how often in time an accurate decision can be made [2].

Chapter 3

Kalman Filter

3.1 Introduction

This chapter covers Kalman Filter (KF) from all aspects. It gives an overview of Kalman filter, its advantages, its applications and an example of Kalman filter. Kalman filter will be used in this thesis for features extraction.

3.2 Kalman Filter Definition

Kalman filter is invented by Rudolf E. Kalman in 1969 and it became one of the most filtering algorithms today because of its small computational requirements. G. Welch and G. Bishop [8] defined Kalman filter as “set *of mathematical equations that provides an efficient computational (recursive) means to estimate the state of a process, in a way that minimizes the mean of the squared error* ". Also Grewal and Andrews [22] defined Kalman filter as "*Theoretically Kalman Filter is an estimator for what is called the linear-quadratic problem, which is the problem of estimating the instantaneous “state” of a linear dynamic system perturbed by white noise" by using measurements linearly related to the state but corrupted by white noise. The resulting estimator is statistically optimal with respect to any quadratic function of estimation error*".

3.3 Kalman Filter Advantages

Kalman Filter considers the greatest achievement in estimation theory of the twentieth century. It enabled technology for Space Age. It made the precise of navigation of spacecraft through the solar system efficient and powerful. Today it used in modern control systems; tracking and navigation of all types of vehicles and predictive design estimation of and controlled systems. Some of its advantages are:

- Efficient because it use least-square method.
- It estimates past, present, future and estimates missing states with inequality measure.
- Powerful and robust because it forgives in many ways and stable.
- Can be implemented in the form of an algorithm for digital computer. It makes capable of much greater than analog filters.
- No need for deterministic dynamics or the random processes have stationary properties, and many applications of importance include non-stationary stochastic processes as EEG signal.
- Compatible with state space formulation of optimal controllers for dynamics systems and it prove useful dual properties of estimations and control.
- Provides the necessary information for mathematically sound, statistically based decision methods for detecting anomalous measurements [23].

3.4 Kalman Filter Applications

The KF has been used in a wide range of applications. Control and prediction of dynamic systems are the main areas. When a KF controls a dynamic system, it is used for state estimation.When controlling a system, it is important to know what goes on in the system. In complex systems, it is not always possible to measure every variable that is needed for controlling the system. A KF provides the information that cannot directly be measured by estimating the values of these variables from indirect and noisy measurements. A KF can for example be used to control continuous manufacturing processes, aircrafts, ships, spacecraft, and robots When KFs are used as predictors, they predict the future of dynamic systems that are difficult or impossible for people to control. Examples of these systems are the flow of rivers during flood, trajectories of celestial bodies, and prices of traded goods [24].

As mention above, KF Kalman filter is the most common today and can be used in many fields but its main goals estimate and perform analysis of estimators. We choose some applications use Kalman filter. Some of KF applications are listed below to prove its importance and ability:

- Phase- locked loops in radio equipment.
- Smoothing the output from laptop trackpads.
- Autopilot.
- Brain-computer interface.
- Chaotic signals.
- Tracking and vertex fitting of charged particles in particle detectors.
- Tracking of objects in computer vision.
- Dynamic positioning [22].

3.5 Kalman Filter Example

To understand Kalman Filter (KF) we give this example to get an idea how the KF work. Suppose, there is a robot moves around in place and need to localize itself. Of course, a robot is subject to sources of noise when it drives around. To estimate its location we suppose that the robot has access to absolute measurement σ_1^2

- **Model**. We model the system of a navigating robot .We suppose robot drive at constant speed s. for this we have system model describes the right locations of robot over time,

$$x_k = x_{k-1} + s + w_k \tag{3.1}$$

Where new location x_k depends on previous location x_{k-1}, speed constant per time step s, and a noisew_k. We suppose the noise is zero mean random noise, and Gaussian distributed. This means that on average the noise is zero sometimes more or less. We present the deviation in the noise byσw.

To use absolute measurements in estimating the location, we have to describe how these measurements are related to the location. We suppose a measurement model that describes how measurementsz_k depend on the location x_k of the robot,

$$z_k = x_k + v_k \tag{3.2}$$

Sensor in this case give measurement $z_k$of location of the robotx_k, it corrupted by measurement noisev_k. We suppose this noise is zero mean on average Gaussian distributed, and it has a deviation of σ_v.

- **Initialization**. Suppose the initial estimate of the location of the robot $\hat{x}_0$and the uncertainty, that is, variance, of σ_1^2 this is the true location.
- **Prediction**. Suppose the robot drives for one time step. As we know the from system model, the location will on average change with about s. Therefore, we can update the

estimate of the location with this information. We can predict what the location of the robot most likely is after one-step. We calculate the new location $\hat{x_1}$ at step k = 1 as

$$\hat{x_1} = \hat{x_0} + s + 0 \qquad (3.3)$$

We took the noise in the system equation as zero. From equation (4.1) we know that the state is corrupted by noise, we do not know the exact amount of noise at a certain time. Since we know the noise on average is zero, we used $w_k = 0$ in calculating the new location estimate.

As we know noise varies around zero, we can update the uncertainty in the new estimate. We calculate the uncertainty σ_1^2 . We have in a new estimate:

$$\sigma_1^2 = \sigma_0^2 + \sigma_w \qquad (3.4)$$

- **Correction**. If the robot keeps on driving without getting any absolute measurements, the uncertainty in the location given by equation (3.5) will increase more and more. If we do make an absolute measurement, we can update the belief in the location and reduce the uncertainty in it. That is, we can use the measurement to correct the prediction that we made.

  Suppose that we make an absolute measurement z1.We want to combine this measurement into our estimate of the location. We include this measurement in the new location estimate using a weighted average between the uncertainty in the observed location from the measurement z_1 and the uncertainty in the estimate that we already had x1

$$\hat{x}\,k^{-} = \frac{\sigma_v^2}{\sigma_1^2+\sigma_2^2} + \frac{\sigma_v^2}{\sigma_1^2+\sigma_2^2} z_1 = \hat{x}\,\frac{\sigma_v^2}{\sigma_1^2+\sigma_2^2}\left(z_1 - \hat{x_1}\right) \qquad (3.5)$$

This way of including the measurement has as consequence that if there is relatively much uncertaintyσ_1^2in the old location estimate, that we then include much of the measurement. On the other hand, if there is relatively much uncertainty σ_v^2in the measurement, then we will not include much of it. Absolute measurements do not depend on earlier location estimates; they provide independent location information. Therefore, they decrease the uncertainty in the location estimate. Realize that probabilities represent populations of samples in a way like mass represents populations of molecules. With this, we notice that we can combine the uncertainty in the old location estimate with the uncertainty in the measurement.

This gives us the uncertainty

$$\frac{1}{\sigma_1^{2+}} = \frac{1}{\sigma_1^2} + \frac{1}{\sigma_v^2}$$

We can rewrite into

$$\sigma_1^{2+} = \sigma_1^2 - \frac{\sigma_1^2}{\sigma_1^2+\sigma_1^2}\sigma_1^2 \tag{3.6}$$

Notice in this equation that incorporating new information always results in lower uncertainty in the resulting estimate. The uncertainty $\sigma^{2,+}$ is smaller than or equal to both the uncertainty in the old location estimate σ_1^2 and the uncertainty in the measurementσ_V^2.Note also that we use in (3.5) and (3.6) same weighting factor. We introduce a factor K representing this weighting factor and rewrite (4.5) and (4.6) into

$$x_1^{\wedge -} = x_1^{\wedge} + K\,(z_x - x_1^{\wedge}) \tag{3.7}$$

$$\sigma_1^{2+} = \sigma_1^2 - K\,\sigma_1^2 = (1 - K) \tag{3.8}$$

where

$$K = \frac{\sigma_1^2}{\sigma_1^2+\sigma_V^2} \tag{3.9}$$

Factor K is a weighting factor that determines how much of the information from the measurement should be taken into account when updating the state estimate. If there is almost no uncertainty in the last location estimate, that is, if $\sigma_1^2$is close to zero, then K will be close to zero. This has consequently that the received measurement is not taken into great account. If the uncertainty in the measurements is small, that is, if σ_v^2 is small, then K will approach one. This implies that the measurement will in fact be taken into account.

In summary, we have in essence shown the equations that the Kalman Filter uses when the state and measurements consist of one variable. The Kalman Filter estimates the state of a system that can be described by a linear equation like (3.1). For reducing the uncertainty, the Kalman Filter uses measurements that are modeled according to a linear equation like (3.2). Starting from an initial state, the Kalman Filter incorporates relative information using equations (3.3) and (3.4). To include absolute information, the Kalman Filter uses equations (3.7) and (3.8) with means of the K factor from the equation (3.9). In the following sections, we will formalize the concepts that we used here and derived the general Kalman Filter equations that can also be used when the state we want to estimate consists of more than one variable [23].

3.6 Kalman Filter Process

The Kalman filter discusses the general problem of trying to estimate the state x $\in k^n$ of a discrete-time controlled process that is governed by a linear stochastic difference equation

$$xk = Ax_{k-1} + Bu_{k-1} + w_{k-1} \tag{3.10}$$

with a measurement, that is $z \in k^m$

$$zk = Hxk + vk \tag{3.11}$$

The random variables wk and vk represent the process and measurement noise (respectively). They are assumed to be independent (of each other), white and with normal probability distributions

$$p(w) \sim N(0, Q) \tag{3.12}$$

$$p(v) \sim N(0, R) \tag{3.13}$$

The process Q noise covariance and R measurement noise covariance matrices might vary with each time step or measurement, but here we consider they are constant. The n × n matrix A in the difference equation (3.10) describes the state at the previous time step k – 1 to the state at the current step k, in the absence of both a driving function and process noise. See that A might vary with each time step, but here we assume it is constant. The n × l matrix B describes the optional control input u∈ $k^{\,n}$to the state x. The m × n matrix H in the measurement equation (3.12) describes the state to the measurement z_k. H might vary with every time step or measurement, but here we assume it is constant [22, 23].

3.7 Kalman Filter Computational Origins

Let x̂ k¯∈k^n be a priori state estimate at step k given information of the process prior to step k and x̂ k ∈k^n be a posteriori state estimate at step k addressed measurementz_k. We also can then define a priori and a posteriori estimate errors as:

$$e_k^- = xk - x\hat{\ } k^- \tag{3.14}$$

$$e_k = xk - x\hat{\ } k \tag{3.15}$$

The *a priori* estimate error covariance is then

$$p_k^- = E[e_k^- e_k^- \; T] \tag{3.16}$$

$$p_k = E[e_k e_k^T] \tag{3.17}$$

In deriving the equations for the Kalman filter, our aim to find an equation computes an a posteriori state estimate xˆ k as a linear compound of an a priori estimate xˆ k¯ and a weighted difference between an real measurement z_k and a measurement prediction H xˆ k ¯ as seen below in (3.18). Some justification for (3.18) is given in “The Probabilistic Origins of the Filter” found below

$$x\hat{}k = x\hat{}k^{-} + \mathrm{k}\,(zk - H\mathrm{x}\hat{}\ \mathrm{k}^{-}) \tag{3.18}$$

The difference (z_k- H xˆ k ¯) in (3.18) is named the measurement innovation or the residual. The residual indicates the difference between the predicted measurement H xˆ k ¯ and the real measurement z_k. A residual of zero means that the two are in full agreement [30, 33].

The (n × m) matrix K in (3.18) is the gain or mixing factor to minimize a posteriori error covariance in equations (3.17). This will achieve first change in equations (1.7) in the above defined for k. when substitute into (3.17), will perform the indicate expectations. When derive of the track of the result with respect to K making result equal to zero, and then solving for K. One method of the Resulting K that minimizes (3.17) is given by

$$Kk = P_K^- H^T (H\, P_K^- H^T + R)^{-1} = \frac{P_K^- H^T}{H\, P_K^- H^T + R} \tag{3.19}$$

From (3.19) we can see that as the measurement error covariance R equals zero, the gain K weights the residual more heavily. Clearly,

$$\lim_{R \to 0} K_k = H^{-1}$$

On the other hand, as the *a priori* estimate error covariance P_K^- approaches zero, the gain *K* weights the residual less heavily. Specifically

Another way of thinking about the weighting by *K* is that as the measurement error covariance approaches zero, the actual measurement zk is “trusted” more and more, while the predicted measurement H xˆ k ¯ is trusted less and less. On the other hand, as the *a* *priori* estimate error covariance p_k^- approaches zero the actual measurementzk is trusted less and less, while the predicted $H\,\hat{x}_k^-$ measurement is trusted more and more[30,33].

3.8 Kalman Filter Operation

The Kalman filter uses feedback control to estimates a process. It estimates the process state at any time and takes feedback from (noisy) measurements. Kalman filter equations classify into two groups: time update equations and measurement update equations. Time update equations project forward (in time) the current state and error covariance estimates to get the a priori estimates for the next time step. The measurement update equations are held for the feedback, i.e. for joining a new measurement into the a priori estimate to get an updated a posteriori estimate. The time update equations can also be considered of as predictor equations, while the measurement update equations can be considered of as corrector equations. Really, the final estimation algorithm resembles that of a predictor-corrector algorithm for solving numerical problems. In Figure 3.1, the time update projects the current state estimate ahead in time. The measurement update adjusts the projected estimate by an actual measurement at that time [22, 23].

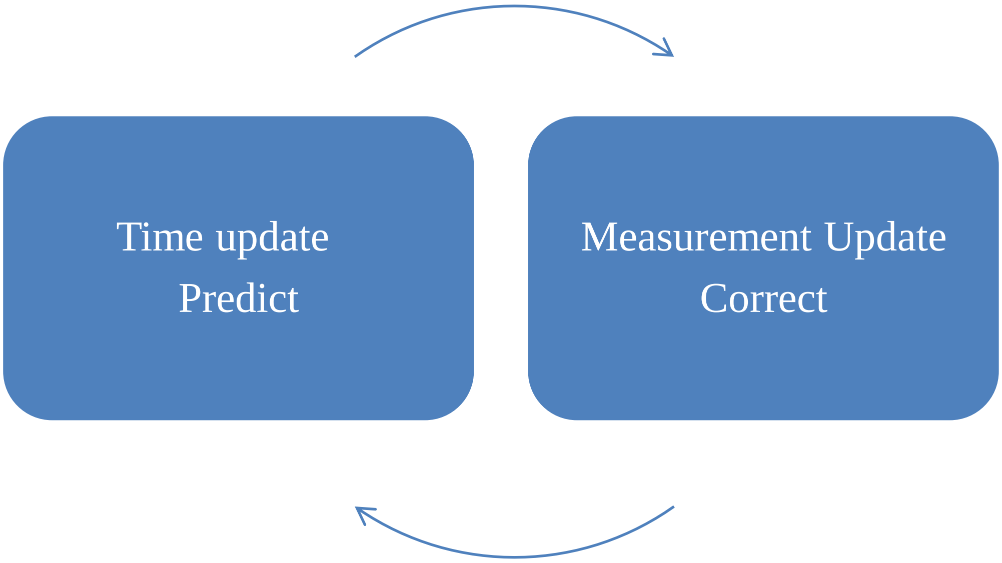

Figure 3.1: Kalman Filter Cycle [22].

Table 3.1: Kalman filter time update equations [21].

$\hat{x}_k^- = Ax_{k-1} + Bu_{k-1}$	(3.20
$P_K^- = AP_{k-1}A^T + Q$	(3.21)

From Table 3-1:

- Project the state and covariance estimates forward from time step $k-1$ to step k.
- Calculate A and B are from (3.10).
- Calculate Q from (3.11).

Table 3.2: Kalman filter update equations [21].

$K_k = P_K^- H^T (H P_K^- H^T + R)^{-1}$	(3.22)
$x\hat{}\, k = x\hat{}\, k^- + \mathrm{k}\,(zk - H\, \mathrm{x}\hat{}\, \mathrm{k}^-)$	(3.23)
$p_k = K_k(I - K_k H) P_K^-$	(3.23)

From Table 3-2:

- First step during the measurement update is to compute the Kalman gain K_k .
- Next step is to actually measure the process to obtain zk.
- Final step is to obtain an *a posteriori* error covariance estimate via (3.23)

Next, each time and measurement update set, the process is returned with the previous a posteriori estimates related to forecast the new a priori estimates. This recursive view is one of the every interesting features of the Kalman filter it makes efficient implementations much more available than (for example) an implementation of a Wiener filter which is designed to work on all of the data directly for all estimate. The Kalman filter instead recursively conditions the current estimate on all of the past measurements. Figure 1-2 below offers a full picture of the operation of the filter, joining the high-level design of Figure 3-1 with the equations from Table3-1 and Table 3-2 [22].

In the real implementation of the filter, the measurement noise covariance R is usually measured before operation of the filter. Including the measurement error covariance R is usually practical (possible) because we want to be ready to measure the process anyway (while operating the filter), so we should generally be able to take any off-line sample measurements in order to manage the variance of the measurement noise. The judgment of the process noise covariance Q is generally higher complex as we typically do not can quickly observe the process we are estimating. Sometimes an almost easy (poor) process model can give satisfactory results if one "injects" enough uncertainty into the process through the selection of Q. Certainly in this case one would hope that the process measurements are reliable. In both case, whether or not we have a reasonable basis for taking the parameters, often-superior filter performance (statistically speaking) can be achieved with setting the filter parameters Q and R. The tuning is regularly done off-line, usually with the help of another (distinct).Kalman filter in the process usually referred to as system identification [22].

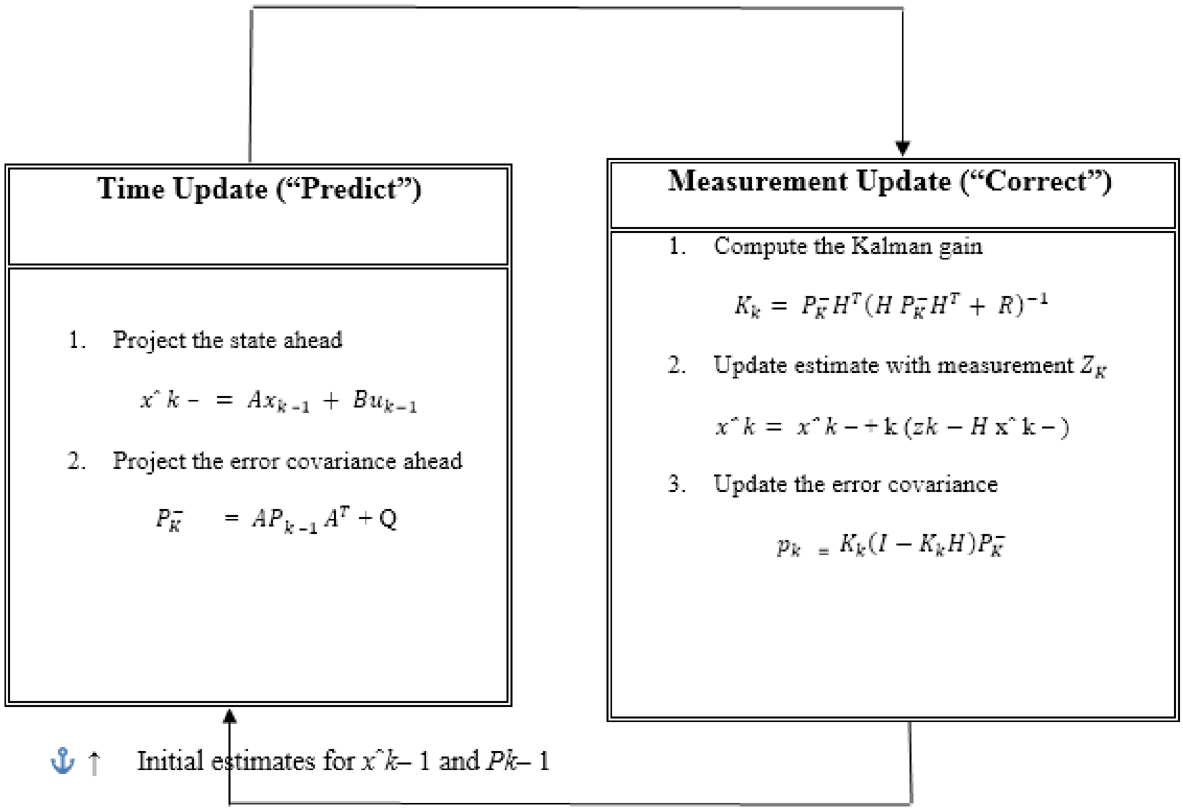

Figure3.2: Kalman filter Operation [21].

We see that under requirements where Q and R .are in fact constant, both the estimation error covariance p_k and the Kalman gain K_K will stabilize fast and then wait constant as we saw in the filter update equations. If this is the case, these parameters can be pre-computed by either running the filter off-line, or by managing the steady-state value. It is often the case but that the measurement error (in fact) does not remain constant. For example, when sighting beacons in our optoelectronic tracker ceiling panels, the noise in measurements of nearby beacons will be smaller than that in far-away beacons. In addition, the process noise is seldom modified dynamically through filter operation becoming in order to set to different dynamics. For example, in the problem of tracking the head of a user of a 3D virtual environment we might reduce the magnitude of if the user shows to be going slowly, and increase the magnitude if the dynamics start changing rapidly. In so cases might be taken to account for both uncertainty of the user's intentions and uncertainty in the model [22, 23].

3.9 Nonlinear Dynamic Systems

Many dynamic system and sensor models are not linear as EEG, but not far from it either. This means that the functions that describe the system state and measurements are nonlinear, but approximately linear for small differences in the values of the state variables. Instead of assuming a linear dynamic system, we now consider a nonlinear dynamic system, consisting of a nonlinear system and a nonlinear measurement model. Nonlinear System Model. The system of which we want to estimate the state is no longer governed by the linear equation from (3.1), but by a nonlinear equation.

We have

$$x_k = f(xk-1) + w_{k-1} \tag{3.24}$$

where f is a nonlinear system function relating the state of the previous time step to the current state, and where w_{k-1} represents the noise corrupting the system. The noise is assumed independent, white, zero-mean, and Gaussian distributed. Nonlinear Measurement Model. We also no longer assume that the measurements are governed by a linear equation as in (3.2). Instead, we have that

$$z_k = h(x_k) + v_k \tag{3.25}$$

Where$h$ is a nonlinear measurement function relating the state of the system to a measurement, and where v_k is the noise corrupting the measurement. This noise is also assumed independent, white, zero-mean, and Gaussian distributed [23].

3.10 Extended Kalman Filter (EKF)

The Kalman filter addresses the general problem of trying to estimate the state $x \in k^n$ of a discrete-time controlled process that is ruled by a *linear* stochastic difference equation. However, what happens if the process to be estimated and (or) the measurement relationship to the process is non-linear. Some of the most interesting and successful applications of Kalman filtering have been such situations. A Kalman filter that linearizes about the current mean and covariance is referred to as an Extended Kalman Filter or EKF. In something akin to a Taylor series, we can linearize the estimation around the current estimate using the partial derivatives of the process and measurement functions to compute estimates even in the face of non-linear relationships. To do so, we must begin by modifying some of the material presented in Section 4.1. Let us assume that our process again has a state vector, but that the process is now governed by the *non-linear* stochastic difference equation [21]:

$$x_k = f(x_{k-1}, u_{k-1}, w_{k-1}) \tag{3.26}$$

And measurement $z \in k^m$

$$z_k = h\,(x_k, v_k) \tag{3.27}$$

Where the random variables $w_k$and v_k represent the process and measurement noise as in (1.3) and (1.4). In this case, the non-linear function $f$in the difference equation (2.1) relates the state at the previous time step $k-1$ to the state at the current time step k. It includes as parameters any driving function u_{k-1} and the zero-mean process noise w_K. The non-linear function h in the measurement equation (2.2) relates the stateXw_K to the measurement z_k.

In use, sure one does not know the original values of the noise $w_k$andv_k at any time step. However, one can close the state and measurement vector without them as

$$k^{\sim} = f\,(x_{k-1}^{\sim}, u_{k-1}, 0) \tag{3.28}$$

$$z_k^{\sim} = h(\,x_k^{\sim}, 0) \tag{3.29}$$

where $x_k^{\sim}$ is some a posteriori estimate of the state (from a previous time step k). It is necessary to see that a primary flaw of the EKF is that the distributions (or densities in the continuous case) of the several random variables are no longer common after undergoing their own nonlinear transformations. The EKF is easily an ad hoc state estimator that only approximates the optimality of Bayes' rule by linearization. The complete set of EKF equations is shown below in Table 3-3 and Table 3-4. Note that we have substituted $X_K^{\sim^-}$ for $x_k^{\sim}$to remain consistent with the earlier "super minus" a priori notation, and thatwe now attach the subscript k to the Jacobians matrices A, W,H,V and , to reinforce the notion that theyare different at (and therefore must be recomputed at) each time step.

Table 3.3: Extended Kalman filter time update equations [21].

$X_K^{\sim^-} \quad = f\,(x_{k-1}^{\sim}, u_{k-1}, 0)$	(3.30)
$P_K^{-} \quad = A_K P_{k-1} A_K^T + W_K\, Q_{K-1}$	(3.31)

As with the basic Kalman filter, the time update equations in Table 3.3 project the state and covariance estimates from the previous time step $K-1$ to the current time step K. A_K and W_K are the process Jacobians at step *k*, and Qk is the process noise covariance at step *k*.

Table 3.4: Extended Kalman filter update equations [22].

$K_k = P_K^- H_K^T (H_K P_K^- H_K^T + V_K R_K V_K^T)^{-1}$	(3.32)
$x\hat{}k = \tilde{X}_K^{-} + k_K (Z_K - h(\tilde{X}_K^{-}, 0))$	(3.33)
$p_k = (I - K_k H) P_K^-$	(3.34)

As with the basic Kalman filter, the measurement update equations in Table 3.4 correct the state and covariance estimates with the measurement Z_K . Again h in (3.33) comes from (3.29), Hk and V are the measurement Jacobians at step *k*, and R_K is the measurement noise covariance at step *k*. Note we now subscript R allowing it to change with each measurement.

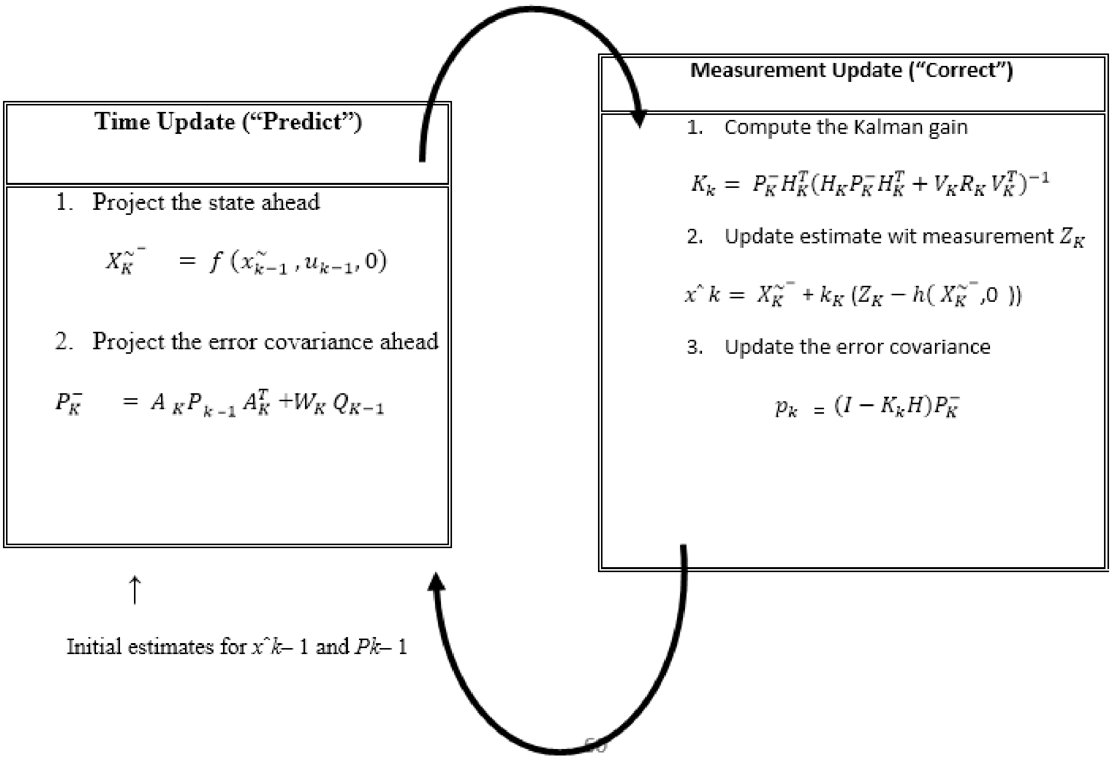

Figure 3.3: An operation of the Extended Kalman Filter [21].

An important feature of the EKF is that the Jacobian$H_K$in the equation for the Kalman gain$K_k$serves to correctly propagate or "magnify" only the relevant component of the measurement information. For example, if there is not a one-to-one mapping between the measurement Z_K andthe state through h , the Jacobian H_K affects the Kalman gain so that it only magnifies the portion of the residual $(Z_K - h(X_K^{\sim -}, 0\ ))$that does affect the state. Of course if overall measurements there is one one-to-one mapping between the measurement Z_K and the state viah , then as you might expectthe filter will quickly diverge. In this case, the process is unobservable [21, 22].

The extended Kalman filter (EKF) is presumably the common generally applied estimation algorithm for nonlinear systems. But , higher than 35 years of experience in the estimation society has revealed that is difficult to implement, difficult to tune, and just reliable for systems that are almost linear on the time scale of the updates. Many of these difficulties arise from its use of linearization [22, 23].

3.11 Perturbation Kalman Filter

Linearized or Perturbation Kalman Filter (PKF) estimates the state of nonlinear dynamic systems by linearizing its nonlinearities. Linearization techniques simulate linear behavior locally at a point or along a small interval. The results of this simulation are then extrapolated to the general domain. The extrapolation depends on the direction of the linearity, that is, the direction of the derivatives at a point on a surface. Linearization around a point x means approximating the function at a very small distance from$x$ [24].

3.12 Iterated Extended Kalman Filter

The EKF linearizes the nonlinear system and measurement function, redefining the nominal trajectories using the latest state estimates once. When there are significant nonlinearities, it can be beneficial to iterate the nominal trajectory redefinition a number of times using the new nominal trajectory. The idea of the Iterated Extended Kalman Filter (IEKF) is to use all information in a measurement by repeatedly adjusting the nominal state trajectory [24].

3.13 Unscented Kalman Filter

A recursive estimator uses knowledge from the previous period in extension to the current observation measurement to produce an estimate of the current state. Unlike the Kalman Filter though, EKF and UKF are designed for non-linear systems. In difference, UKF uses unscented transformation technique, which measures the statistics of a stochastic variable that undergoes non-linear transformation .It is perfect up to the second order and needs fewer samples compared to an alike particle filter. The performance of UKF under certain conditions and showed that it performed robustly in general tracking applications of non-linear systems. Figure 1 shows the overview of the UKF process, which is composed of two main parts, similar to the KF. First is the time-update, where in the initial state estimate is calculated by choosing sigma points and solving for its mean and covariance. The observation is also propagated in this step and its mean and covariance are calculated. The second part is the measurement update. The Kalman gain and cross-covariance of the propagated state and the propagated observation are measured and used to update the state and its covariance [25].

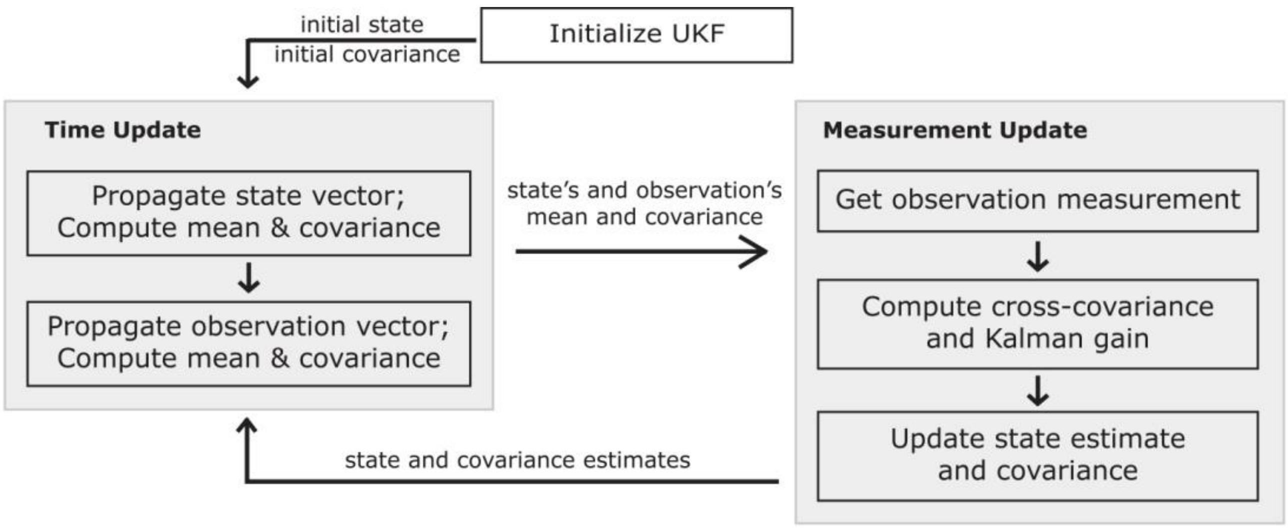

Figure 3.4: Unscented Kalman Filter process [25].

3.14 Particle filters

Particle filters are an alternative technique for state estimation. Particle Filters represent the complete posterior distribution of the states. Therefore, they can deal with any nonlinearities and noise distributions. Particle filter have been combined with the Unscented Kalman Filter in the Unscented Particle Filter [24].

3.15 Ensemble Kalman Filter

Ensemble Kalman Filter allows for states with huge amounts of variables. Due to the computations involved in propagating the error covariance in the KF, the dimension of the states is restricted to no more [24].

Chapter 4

Proposed Feature Extraction Method

4.1 Introduction

Steady-state visual evoked potentials (SSVEP) are periodic change in brain signals as a response to repetitive visual stimuli. The frequency of repetitive visual stimulus and its harmonics appear in the recoded Electroencephalography (EEG). Thus, the recorded EEG signal can be modeled as a weighted sum of stimulus frequency and its harmonics. The weights can be estimated using Kalman filter.

4.2 SSVEP Modeling

Any periodic signal can be decomposed into a set of Fourier series. As the brain dynamics perform as a low-pass filter [26, 27], high harmonic components will be filtered. Therefore, a preprocessed SSVEP signal generated from stimulus with frequency *f* can be decompose into the Fourier series of its harmonics as follows [28]:

$$y(t) = \sum_{i=1}^{n} w_{1i} \sin(2i\pi ft) + w_{2i} \cos(2i\pi ft) + e_i \tag{4.1}$$

Where f is the base frequency, $t = \frac{1}{s}, \frac{2}{s}, \cdots, \frac{T}{s}$, T is the number of samples and s is the sampling rate (128 Hz in our case), n is the number of harmonics and e_i is a Gaussian noise with zero mean and σ^2 variance. We assume that the time segment is short enough for the noise to be stationary within this segment [29].

4.3 Estimation of Model Parameters

In order to estimate the parameters of recorded EEG signal modeled by equation (4.1), Kalman filter described in Figure 3.2 is employed. To this end, the system (4.1) should be rewritten in the form of equation where the system parameters are the state of the new system.

$$W_k = W_{k-1} + E_k \tag{4.2}$$

$$y_k = HW_k + v_k \tag{4.3}$$

where $W_k = [w_{11} \quad w_{21} \quad \cdots \quad w_{1n} \quad w_{2n}]$, E_k is the covariance matrix of the process noise of zero mean, $H = \left[\sin(2\pi ft) \quad \cos(2\pi ft) \quad \cdots \quad \sin(2n\pi ft) \quad \cos(2n\pi ft)\right]^T$ and v_k is the measurement noise with zero mean.

The parameter vector W_k can be estimated using Kalman filter described in Figure 3.2. The initial values of can be set as described in [29, 30]. The estimation process is shown in the following figure:

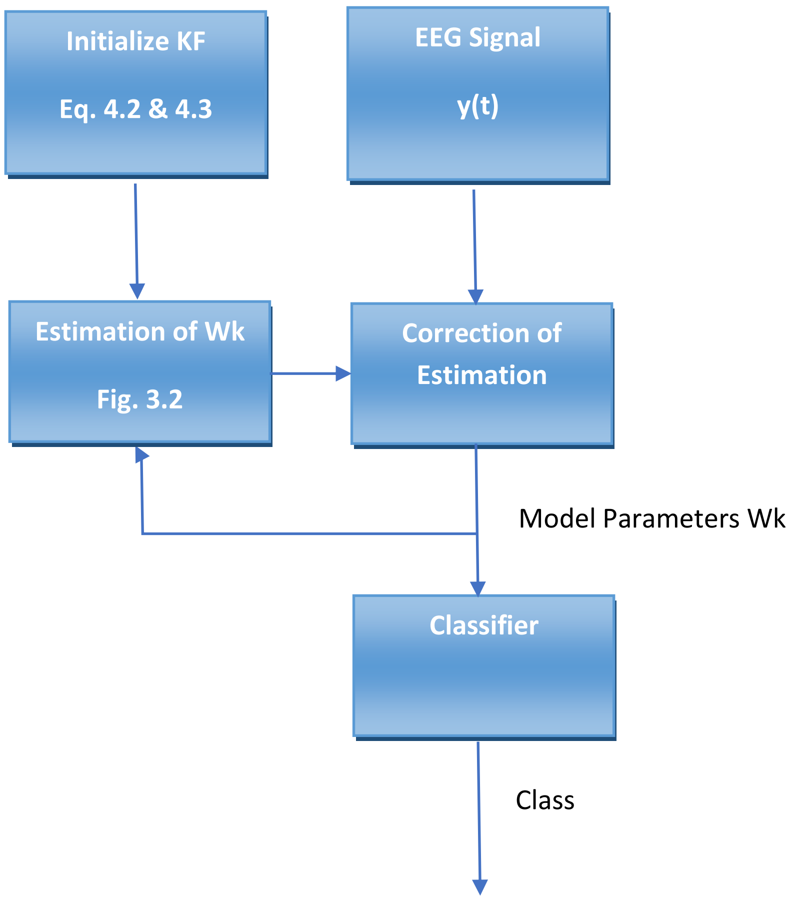

Figure 4.1: Proposed estimation process

Chapter 5

Results and Discussion

5.1 Introduction

In order to evaluate the developed feature extraction method, a SSVEP experiment is built. The experiment is run using a predefined procedure where the user has to look at each stimulus with a specific frequency and time. The recorded signals are preprocessed and two methods are employed to extract the features: Fast Fourier Transform approach and the proposed method. A linear discriminate classifier is used to classifier the two sets of features and the results are compared.

5.2 SSVEP Experiment

The proposed SSVEP system consists of two checkerboards working at different frequencies as shown in Figure 5.1.

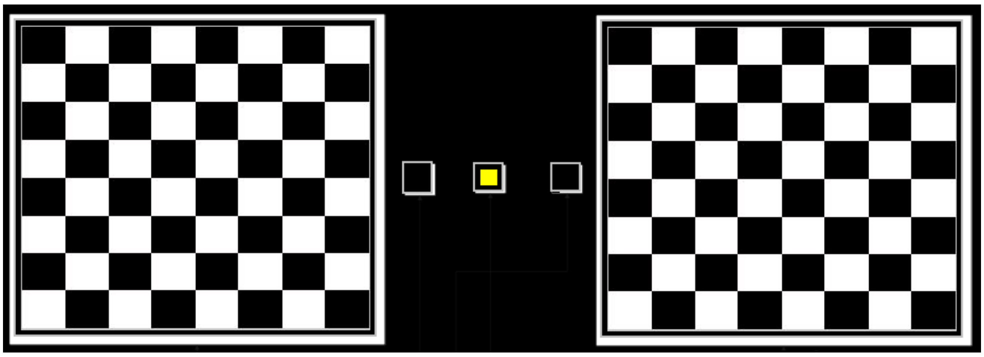

Figure 5.1: Proposed 2-class visual stimulation system

A subject looks at specified checkerboard indicated by the yellow square beside it. The generated EEG signal is recorded using EPOC Emotiv headset with fourteen sensors distributed over the scalp as shown in Figure 5.2 .

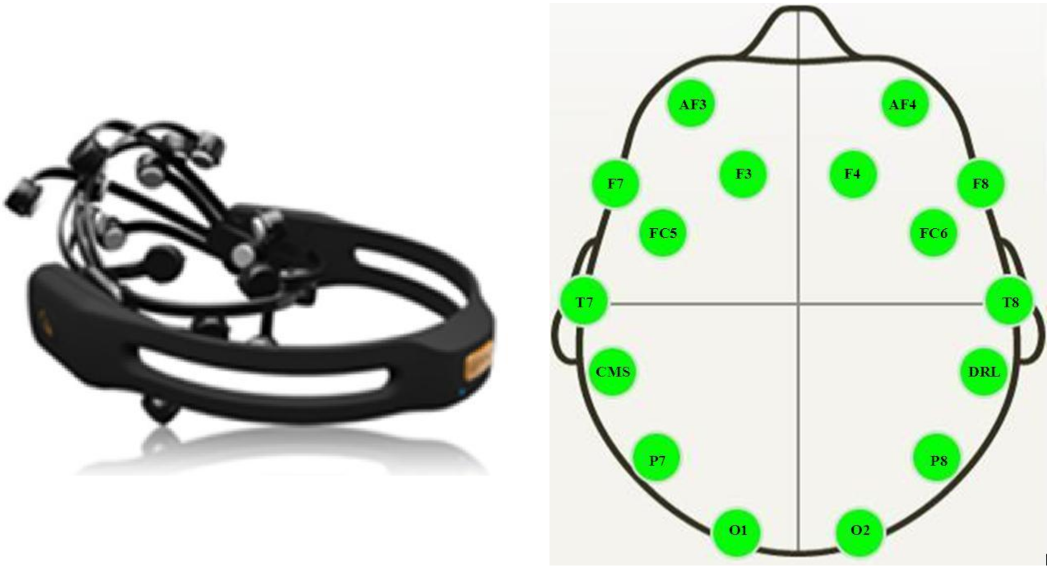

Figure 5.2: Signal acquisition unit: the Emotiv EPOC headset (Left) and the location of electrodes relative to the head (Right).

In order to extract features from recorded EEG signal, the recorded EEG signal is firstly filtered by a fourth order Butterworth filter between 2 Hz and 30 Hz. Then two channels are constructed from the fourteen EEG signals using a correlation method. EEG segments correspond to left and right flickers are extracted from constructed channels. Each segment is divided into 1 second segments and Fast Fourier Transform (FFT) is applied on each 1 second segments. Finally, the values of FFT of each 1 second segment at working frequencies and their harmonics are extracted to form the feature vector as shown in Figures 5.3 and 5.4.

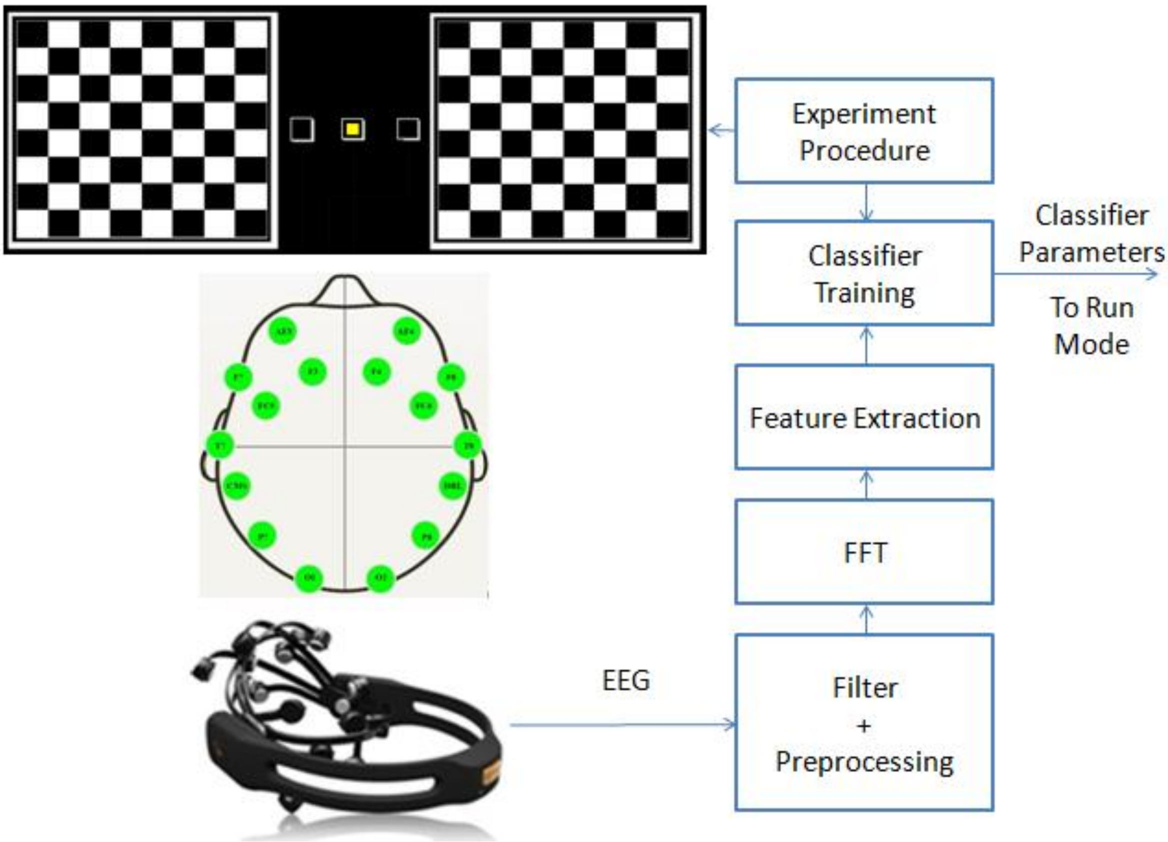

Figure 5.3: Training Mode SSVEP Experiment using FFT.

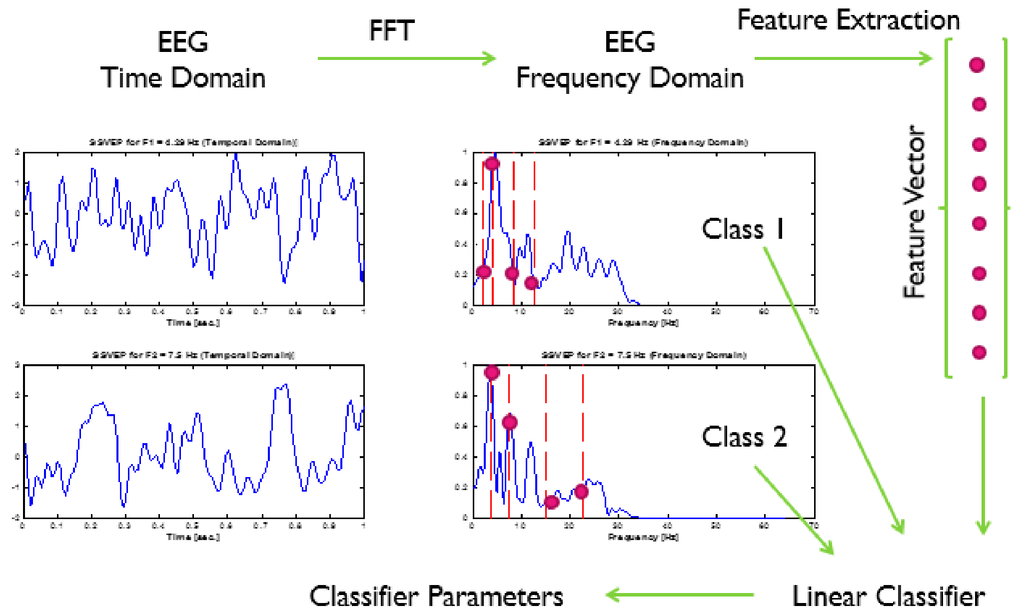

Figure 5.4: Signals in Training Mode using FFT.

The obtained samples, feature vectors and their classes are divided into training and test groups using 10-fold cross-validation method. The training samples are used to train linear classifier and the test samples are used to test the trained classifier error rate.

Same above experiment is performed but using the proposed Kalman filter instead of the FFT. The obtained results will be presented in next section as shown in Figures 5.5 and 5.6.

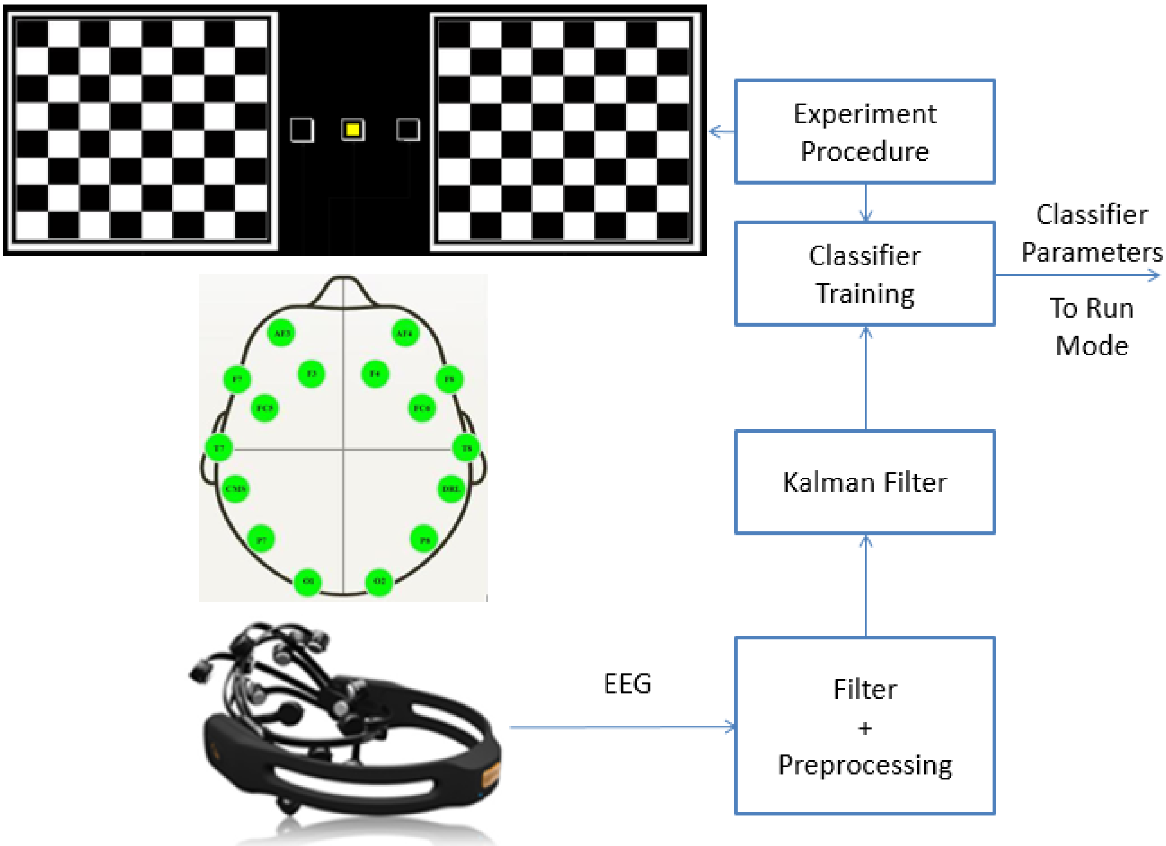

Figure 5.5: Training Mode SSVEP Experiment using KF.

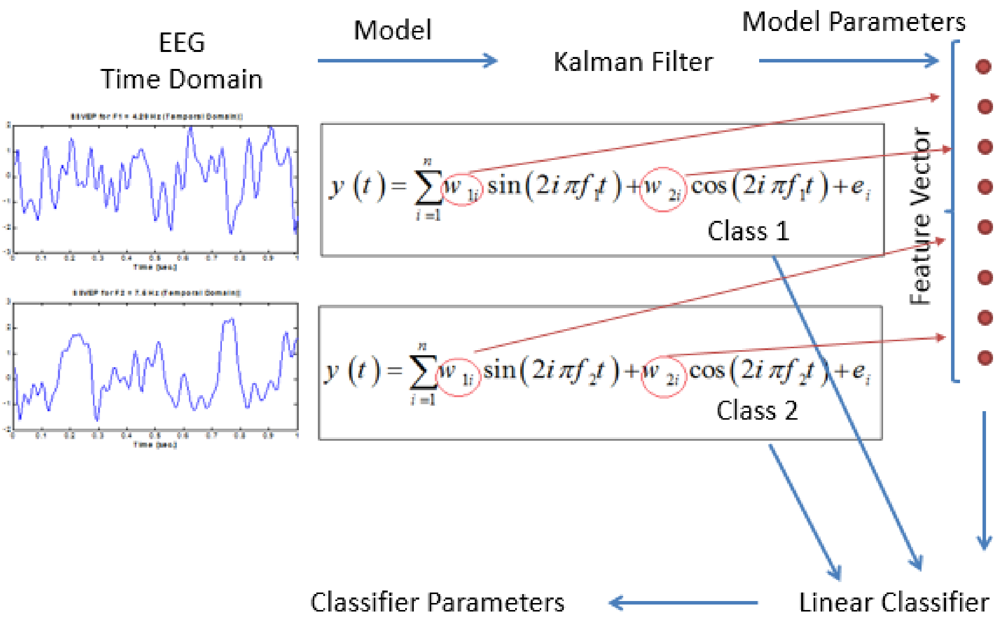

Figure 5.6: Signals in Training Mode using KF.

5.3 Results and Discussion

The FFT method produced an average error rate 35%. Figure 5.6 shows the Classified and misclassified samples.

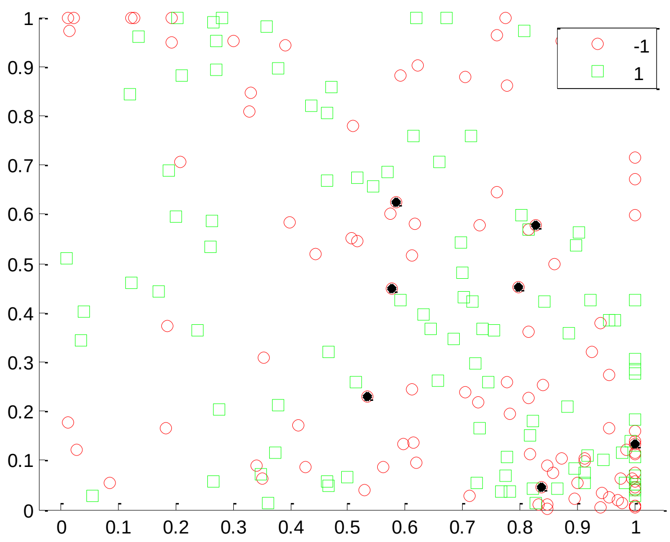

Figure 5.7: Classified and misclassified samples (black samples are misclassified).

The Kalman method produced an average error rate 20%. Figure 5.7 shows the Classified and misclassified samples.

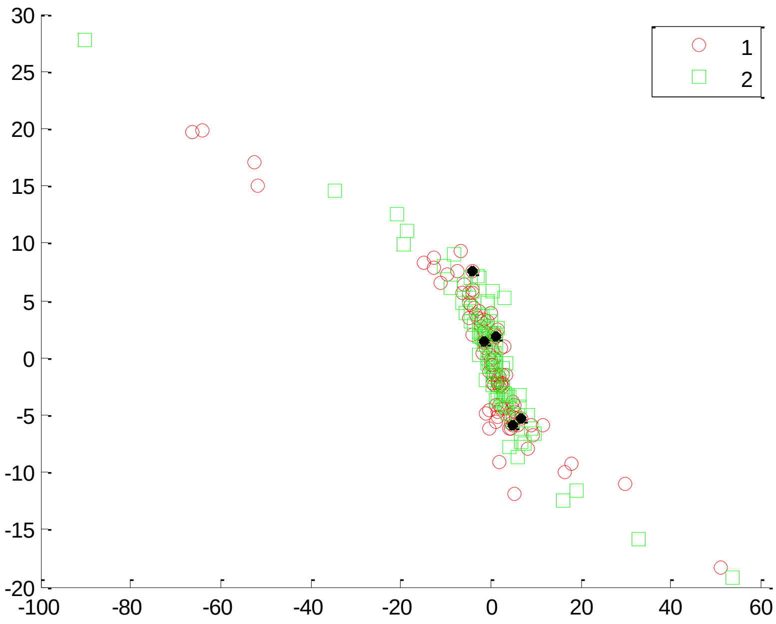

Figure 5.8: Classified and misclassified samples (black samples are misclassified).

5.4 Conclusion

A feature extraction method is proposed in this master research. The proposed method is based on modeling the short-time preprocessed SSVEP signal as weighted sum of sinusoidal signals with frequency equal to the stimulus frequency and its harmonics. Then a Kalman filter is employed to estimate the weights of this sum.

The proposed methods is applied in a binary SSVEP experiment and it showed better classification accuracy comparing with other methods.

5.5 Future Work

As a future work, the number of harmonics used in the SSVEP signal model need to be optimized. More experiments need to be carried out with different number of harmonics and the optimal value should be defined.

In addition, the initial values used in the Kalman filter need to be determined in a more accurate way.

LIST OF REFERENCES

1. Ali Bashashati, MehrdadFatourechi, Rabab K Ward , Gary E Birch , A survey of signal processing algorithms in brain–computer interfaces based on electrical brain signals, journal of neural engineering , Published 27 March 2007.

2. Lotte, Fabien, et al. "A review of classification algorithms for EEG-based brain–computer interfaces." Journal of neural engineering 4 (2007).

3. Schalk, Gerwin, et al. "BCI2000: a general-purpose brain-computer interface (BCI) system." Biomedical Engineering, IEEE Transactions on 51.6 (2004): 1034-1043.

4. Celine De Vreese, Brain Computer Interfaces based on imaginary handmovement using EEG beamforming, Master of Science in Biomedical Engineering, University of Ghent, June 2012.

5. Pfurtscheller, Gert, et al. "Current trends in Graz brain-computer interface (BCI) research." IEEE Transactions on Rehabilitation Engineering 8.2 (2000): 216-219.

6. McFarland, Dennis J., et al. "BCI meeting 2005-workshop on BCI signal processing: feature extraction and translation." IEEE transactions on neural systems and rehabilitation engineering 14.2 (2006): 135.

7. Bashashati, Ali, et al. "A survey of signal processing algorithms in brain–computer interfaces based on electrical brain signals." Journal of Neural engineering 4.2 (2007): R32.

8. Mason, S. G., et al. "A comprehensive survey of brain interface technology designs." Annals of biomedical engineering 35.2 (2007): 137-169.

9. DEL R. MILLÁN, J. O. S. É., et al. "Non-invasive brain-machine interaction." International Journal of Pattern Recognition and Artificial Intelligence 22.05 (2008): 959-972.

10. Haselsteiner, E. &Pfurtscheller, G. (2000). Using time-dependent neural networks for EEG classification. IEEE Trans. on Rehabilitation Engineering, Vol. 8, pp. 457-463.

11. Luis Fernando, Nicolas-Alonso, Jaime Gomez-Gil, Review Brain Computer Interfaces, www.mdpi.com/journal/sensors, Sensors2012, 12, 121112doi:10.3390/s12020121,

Published 31 January 2012.

12. Obermeier, B.; Guger, C.; Neuper, C. & Pfurtscheller, G. (2001). Hidden Markov models for online classification of single trial EEG. Pattern recognition letters, pp. 1299-1309.

13. Dalal Mohammed Bakheet , (2014) ;P300 Quran Player Based On Ordinal Analysis OF Time Series, 2014**.**

14. Lin C.J., Hsieh M.H. Classification of mental task from EEG data using neural networks based on particle swarm optimization. Neurocomputing. 2009; 72:1121–1130.

15. Kun L., Sankar R., Arbel Y., Donchin E. Single trial independent component analysis for P300 BCI system. Proceedings of the 31th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBCS'09); Minneapolis, MN, USA. September 2009; pp. 4035–4038.

16. Krusienski D.J., McFarland D.J., Wolpaw J.R. An Evaluation of Autoregressive Spectral Estimation Model Order for Brain-Computer Interface Applications. Proceedings of the 28th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBS'06); New York, NY, USA. September 2006; pp. 1323–1326.

17. Krusienski D.J., Schalk G., McFarland D.J., Wolpaw J.R. A mu-rhythm matched filter for continuous control of a brain-computer interface. IEEE Trans. Biomed. Eng. 2007;54:273–280.

18. Bostanov V. BCI competition 2003-data sets Ib and IIb: Feature extraction from event-related brain potentials with the continuous wavelet transform and the t-value scalogram. IEEE Trans. Biomed. Eng. 2004;51:1057–1061.

19. Mason S.G., Birch G.E. A brain-controlled switch for asynchronous control applications. IEEE Trans. Biomed. Eng. 2000;47:1297–1307.

20. Ramoser H., Muller-Gerking J., Pfurtscheller G. Optimal spatial filtering of single trial EEG during imagined hand movement. IEEE Trans. Rehabil. Eng. 2000;8:441–446.

21. Welch, Greg, and Gary Bishop. "An introduction to the Kalman filter." (1995).

22. Grewal, Mohinder S., and Angus P. Andrews. *Kalman filtering: theory and practice using MATLAB*. John Wiley & Sons, 2011.

23. Negenborn, Rudy. *Robot localization and Kalman filters*. Diss. Utrecht University, 2003.

24. Strid, Ingvar, and Karl Walentin. "Block Kalman filtering for large-scale DSGE models." *Computational Economics* 33.3 (2009): 277-304.

25. Nunez, Paul L., and Ramesh Srinivasan. Electric fields of the brain: the neurophysics of EEG. Oxford university press, 2006.

26. Bédard, Claude, Helmut Kröger, and Alain Destexhe. "Modeling extracellular field potentials and the frequency-filtering properties of extracellular space." Biophysical journal 86.3 (2004): 1829-1842.

27. Lin, Zhonglin, et al. "Frequency recognition based on canonical correlation analysis for SSVEP-based BCIs." Biomedical Engineering, IEEE Transactions on 53.12 (2006): 2610-2614.

28. Friman, Ola, Ivan Volosyak, and A. Graser. "Multiple channel detection of steady-state visual evoked potentials for brain-computer interfaces." Biomedical Engineering, IEEE Transactions on 54.4 (2007): 742-750.

29. A. Schlögl, “The electroencephalogram and adaptive autoregressive model: theory and applications,” Ph.D. dissertation, Technischen University at Graz, 2000.

30. C. W. Anderson, E. A. Stolz, and S. Shamsunder, “Multivariate autoregressive models for classification of spontaneous electroencephalogram during mental tasks,” IEEE Trans. Biomedical Eng., vol.45, no.3, pp.277-286, 1998.