Department of Electrical and Computer Engineering

Multi-channel Extraction Methods for Event-Related Potentials

Lee Wee Lih

This thesis is presented for the Degree of

Doctor of Philosophy

of

Curtin University

June 2014

Declaration

To the best of my knowledge and belief this thesis contains no material previously published by any other person except where due acknowledgment has been made.

This thesis contains no material which has been accepted for the award of any other degree or diploma in any university.

Signature:  . . . . . .

Date: 25/06/2014 . . . . . .

Acknowledgements

Firstly, I am deeply indebted to my supervisors A/Prof Tele Tan and Dr Yee Hong Leung for their continuous supervision, guidance, patience and support. This thesis would not have been possible without their detailed comments as well as their insightful discussions throughout these years.

In addition, I would also like to thank Prof Torbjörn Falkmer from Dept of Occupational Therapy for allowing me to join their fortnightly meetings. It is my privilege to meet Dr Marita Falkmer, Dr Chiara Horlin and Dr Matthew Albercht. Their enthusiasm in improving the lives of people with autism have been inspiring and infectious throughout my PhD research.

Sincere gratitude is also given to my postgraduate colleagues for their frequent discussions that keep my research environment lively. Also, special thanks to Jan Hakenberg, who taught me how to write an efficient Matlab program and Sarah Ting, for the provision of the important EEG data.

Finally I would like to thank my family and friends. To my wonderful parents, Lee Swee Kwang and Chien Mee Eng, and my sisters, Lee Yih Siew and Lee Yih Juin for their unrelenting support and encouragement on my PhD studies. Besides that, I would also like to express my gratitude to my best friends, Yong Pei Chee, Nicholas Chong and Lu Seng Kiat for their companion and encouragement throughout these years. Most importantly, I would like to thank Chen Mei for her unconditional sacrifice, love and patience and for making everything worthwhile.

Abstract

In electroencephalography (EEG), event related potentials (ERPs) can be observed as a series of positive and negative voltage deflections that are generated in response to a stimulus or event. These unique EEG signals reflect the chronological neural events that occur milliseconds to hundreds of milliseconds after a given stimulus or event. Thus, ERPs have been one of the important tools used for decades to uncover the sensory, cognitive and motor processes that underlie human thoughts and behaviour.

Extracting ERPs from the noisy background EEG activity is one of the fundamental problems in ERP analysis. Apart from the limitation of the traditional trial-averaging method, the recent advancement in EEG recording devices and Brain Computer Interfaces (BCIs) has also brought forth two important changes towards the ERP signal processing field. These changes are 1) the increasing number of signals recorded simultaneously from the scalp and 2) the requirement to extract ERPs at the single-trial level. To address these issues, the main objective of this thesis is to develop an effective single-trial multi-channel ERP extraction method, so that, all the signals recorded from the different electrodes can be fully utilized for recovering the ERP waveform at the single-trial level.

Like most existing techniques, this thesis aims to develop a new set of data-driven methods to extract ERPs without making any strong assumptions on the scalp distribution and temporal waveform of the desired ERP. To achieve this goal, this thesis seeks to provide improvement by utilising the prior information of the desired ERP in the extraction process so that the specific ERP can be extracted effectively. The first proposed method is a new linear discrimination (LD) method that performs the extraction by learning the differences between the scalp distribution of the desired and undesired ERPs. This method has the ability to extract a specific ERP under the presence of multiple ERPs. However, a closer examination shows that due to the nature of

the method, the extracted signal always experiences a slight mismatch with the actual ERP waveform.

For this reason, in the second proposed method, a temporally-constrained independent component analysis (ICA) method, namely, one-unit ICA-R is investigated. One-unit ICA-R inherits ICA's superior performance in recovering the source signals from the noisy mixed signals. The main advantage of one-unit ICA-R is its ability to avoid time-consuming full ICA decomposition and extract the desired ERP directly through the guidance of a reference signal. The design of the reference signal is, therefore, important to one-unit ICA-R. To guide the extraction successfully, a new way of generating the reference signal is developed specifically for the ERP application. By applying one-unit ICA-R with the proposed reference signal, the results demonstrate that the proposed one-unit ICA-R outperforms the traditional ICA in terms of extraction quality and computational efficiency.

Lastly, a data-driven ERP segmentation algorithm is developed to provide the ERP time region that is required for initializing the proposed LD and one-unit ICA-R methods. In this thesis, the proposed segmentation algorithm is designed specifically for a popular ERP called P300. To find the P300 time region, a new clustering technique is applied so that the P300 time region can be identified adaptively based on the provided signals. In addition, the proposed algorithm does not require any manual intervention. Thus, when applied together with the proposed ERP extraction methods, a fully-automated P300 extraction framework is also achieved.

Contents

List of Figures

List of Tables

Acronyms

a2KM	Automatic 2-cluster K-Means Algorithm
BCI	Brain Computer Interface
BLDA	Bayesian Linear Discriminant Analysis
cLD	Conventional Linear Discrimination
CSP	Common Spatial Pattern
EEG	Electroencephalography
ERP	Event-Related Potential
ENE-LD	ERP-vs-Non-ERP time region based Linear Discrimination
fMRI	functional Magnetic Resonance Imaging
fNIRS	functional Near Infrared Spectroscopy
GFP	Global Field Power
GMD	Global Map Dissimilarity
ICA	Independent Component Analysis
ICA-R	Independent Component Analysis with Reference
ICA-MR	Independent Component Analysis with Multiple Reference
KM	K-Means Algorithm
MAD	Mean Absolute Difference
MEG	Magnetoencephalography
mKM	Modified K-Means Algorithm
PET	Positron Emission Tomography
PCA	Principal Component Analysis
LD	Linear Discrimination
LDA	Linear Discriminant Analysis
LR	Logistic Regression
PI	Performance Index

SVM	Support Vector Machine
SNR	Signal-to-Noise Ratio
SSNR	Signal-to-Signal and Noise Ratio
TNTR	Target-to-Non-Target Ratio

Chapter 1

Introduction

1.1 Motivation

The brain is the most complex organ in the human body. Ever since we are born, it provides us with the abilities to think, learn, understand, feel and interact with the outside world. In our daily lives, the brain performs many tasks that are seemingly simple and yet difficult to be achieved by today's machine such as recognising a face, differentiating two different tones, understanding an emotion, etc. For centuries, understanding how the brain achieves these remarkable feats has sparked a lot of interest. However, a more comprehensive study on the brain is not made possible without the invention of Electroencephalography (EEG) by Prof Hans Berger in 1929 [1]. Since then, EEG has marked an important milestone in the history of brain research.

EEG is a technique [1] that measures the electrical activity of the brain through electrodes placed on the scalp. Today, besides EEG, many different kinds of neuro-imaging devices have also been introduced such as Magnetoencephalography (MEG), Positron Emission Tomography (PET), functional Near Infrared Spectroscopy (fNIRS) and functional Magnetic Resonance Imaging (fMRI) [2]. Although these methods served as alternative ways of investigating the brain by measuring either the blood flow or magnetic field in the brain, EEG remains the most popular method for the brain research mainly because it is relatively safe, cheap, portable, non-invasive, quiet and has a very high temporal resolution [2][3][4].

Currently, EEG has been used in different kinds of brain studies that ranged from short-term recordings such as event-related potentials (ERPs) to long-term recordings

such as in the study of sleep disorder [3][4]. Among these studies, ERPs can be regarded as one of the most important part of EEG studies because they are best observed under the fine temporal resolution of the EEG recordings. Specifically, ERPs are electrical activities that occur milliseconds to a few hundreds of milliseconds after a specific internal or external event happens [5][6][7]. They index the chronological neural processes that are responsible for how human senses, perceives, recognizes and reacts towards an event. Understanding how ERPs are generated and also how it differs between the typical and atypical population provides us with a better understanding of the neurophysiological behaviour of the brain. Most importantly, it helps us to identify the important "bio-markers" that are potentially useful for designing a more effective diagnosis and treatment to cure any illness caused by brain disorders.

In EEG recordings, ERP has a relatively small amplitude when compared to the on-going background EEG activities and noise artifacts. For this reason, ERP extraction has been an important issue in ERP analysis ever since the first ERP was discovered in 1939 [8]. Historically, ERP research did not progress very well until the computer-generated trial-averaging method was introduced in the 1960s [5][7]. Today, trial-averaging remains an important technique in ERP studies. To perform the ERP analysis successfully, it is common practice to collect a large number of ERP trials so that badly contaminated ERP trials can be rejected while the remaining effective trials are averaged. Traditionally, trial-averaging is applied mainly to suppress the random signals that are not time-locked across trials so that the signals can be smoothed and the overall representation of the ERPs can be obtained. Unfortunately, there are certain issues regarding ERPs that remain puzzling and cannot be solved by studying only the ERPs extracted from the trial-averaged signals. One of these issues is the trial-variability of ERPs that has been found in most ERP studies [9][10][11]. Understanding the degree of ERP variability at the single-trial level and the factors that cause such variability has generated an immense interest in the ERP community. Subsequently, in the last two decades, the development of single-trial ERP analysis method has been one of the hot research topics in the ERP signal processing field.

Today, technological breakthrough in electronic and computer technology have heralded a new era in ERP research. With EEG recording devices being more advanced, relatively easy-to-use and affordable, measuring EEG signals simultaneously

from different parts of the scalp is no longer an obstacle in ERP studies. The increasing trend of adopting multi-channel EEG recordings offers us an opportunity to gain new insights on the spatial characteristics of ERPs [12]. However, at the same time, the massive amount of measured signals also presents new challenges in ERP studies. As a result, the development of new ways of integrating or utilizing the signals from different EEG channels to extract ERPs in ERP studies are very essential so that the ERP-of-interest can be studied effectively. In recent years, the increasing demand for such methods can also be attributed partly to the development of brain computer interface (BCI) which is a technology that aims to utilize EEG/ERP signals to establish communication between humans and computers [13][14]. To enable a smooth and responsive BCI application, the ability to perform multi-channel extraction at the single-trial level is also crucial to ensure that each individual ERP can be detected successfully.

1.2 Objectives

As described above, there are a multitude of challenges in extracting ERPs. Specifically, a successful ERP extraction method should encompass the following characteristics so that the multi-channel ERP recordings can be utilised effectively:

1. The ability to recover the ERP-of-interest reliably from each individual trial.
2. The ability to utilize all the provided EEG channels during ERP extraction process.
3. The desire to not make too many strong assumptions on the ERP-of-interest particularly the ERP waveform and its spatial location since they may vary with subjects and experimental conditions.
4. The need to avoid or reduce the amount of human intervention since it is impractical to visually examine the huge amount of data gathered by the multi-channel ERP recordings.

To fulfill the above requirements, the objective of this thesis is to investigate and provide a new set of single-trial multi-channel ERP extraction methods so that they can facilitate the analysis of ERPs by requiring the least amount of human intervention.

1.3 Contributions of the thesis

In this thesis, the research has focused mainly on designing data-driven ERP extraction methods that are feasible for real-time implementation. The original contributions are as follows:

1. **A new linear discrimination (LD) method for extracting the desired ERP**

   A new ERP extraction method that learns the differences between the scalp distribution of the desired ERP and the background EEG/ERP is developed. The proposed LD method is designed specifically to extract an ERP-of-interest. Thus, unlike most existing data-driven extraction methods, the proposed method does not require any source signal selection before the extracted signal can be analysed. The proposed LD method also outperforms the conventional LD method in ERP extraction. Results show that the proposed LD method is able to extract small ERPs even if they are under the presence of multiple ERPs.

2. **One-unit Independent Component Analysis with reference (ICA-R) for ERP extraction**

   A temporally-constrained ICA, namely one-unit ICA-R, is proposed to overcome the source signal selection problem in the traditional ICA method. In addition, to guide the one-unit ICA-R for extracting the desired ERP directly, a new type of reference signal is also proposed to be used together with the ICA-R. The proposed method is computationally efficient and do not extract redundant signals that have to be subsequently removed. Thus, compared to the traditional ICA, one-unit ICA-R is also a more practical tool for real-time ERP applications.

3. **Data-driven method for selecting the P300 ERP time-region**

   P300 is one of the ERPs that has been widely used in different ERP applications. Defining the P300 time region is a preliminary step to initialise our proposed LD and ICA-R methods so that P300 can be extracted effectively. To estimate the time region in an adaptive and autonomous way, a data-driven ERP segmentation algorithm is developed. This method does not require any human supervision and thus avoids bias and subjective outcome when defining the P300 time

region. The overall result demonstrated how the ERP segmentation and extraction methods can be combined to provide a fully-autonomous framework in ERP extraction. However, currently the proposed method only works for P300 ERP since it exploits certain properties that are particular to P300 ERP.

1.4 Thesis outline

The focus of this thesis is ERP extraction at the single-trial level and the flow of this thesis is intended to resemble the progress of the author's research and development. Each chapter provides a stepping stone for the subsequent chapter or complements one another. The thesis is organized as follows:

Chapter 2 provides a brief background on EEG and ERP signals whereby the basic processes that govern the generation of EEG and ERP are first discussed before types of commonly encountered ERPs are reviewed.

Chapter 3 provides a brief background on the existing ERP extraction methods and also the pattern recognition methods that are popularly adapted for ERP extraction, classification and segmentation problems.

Chapter 4 proposes a new type of linear discrimination method that performs single-trial ERP extraction by learning the differences in the scalp distribution between the ERP-of-interest and irrelevant background EEG/ERPs.

Chapter 5 proposes the application of one-unit ICA-R for extracting the desired ERP at the single-trial level. The chapter begins with a simulation study to decide the best reference signal to be used with ICA-R for the ERP application. After that, the resultant one-unit ICA-R is validated on real P300 ERP datasets.

Chapter 6 proposes a new clustering-based ERP segmentation algorithm that exploits the unique characteristic of the P300 ERP to allow autonomous P300 time region identification.

Chapter 7 summarises the research and outlines recommendations for future work.

1.5 List of Peer-Reviewed Publications

The following papers are published in conjunction with this thesis:

1. W. L. Lee, Y. H. Leung and T. Tan, "P300 Response Classification in the Presence of Magnitude and Latency Fluctuations", in *Proc. Int. Conf. on Neural Information Processing (ICONIP'11)*, Shanghai, China, Nov. 2011, pp. 352–359.

2. W. L. Lee, T. Tan, T. Falkmer and Y. H. Leung, "Single-Trial Multi-channel N170 Estimation Using Linear Discriminant Analysis (LDA)", in *Proc. Int. Conf. on Neural Information Processing (ICONIP'12)*, Doha, Qatar, Nov. 2012, pp. 347–355.

3. W. L. Lee, T. Tan, and Y. H. Leung, "An improved P300 extraction using ICA-R for P300-BCI speller", in *Proc. 35th Ann. Int. Conf. of the IEEE Engineering in Medicine and Biology Society (EMBC'13)*, Osaka, Japan, July 2013, pp. 7064–7067.

Chapter 2

EEG and Event-Related Potentials

2.1 Overview

This chapter provides a brief introduction on electroencephalography (EEG) and event-related potentials (ERPs). The chapter begins by explaining the neurophysiological process that generates the observed EEG signals and also the standard procedure for collecting the EEG signals in Section 2.2. This is followed by an introduction on ERPs in Section 2.3, which includes the experiment used to generate the ERP, the way ERP is categorised, and the types of ERPs that are considered in this thesis. In Section 2.4, a brief summary of the application of ERPs in the Brain-Computer Interface is also provided.

2.2 Electroencephalography

Electroencephalography (EEG) is a method of recording the electrical activities inside the human brain through the placement of electrodes on the scalp. For several decades, it has been an essential tool that provides safe and non-invasive way of monitoring the brain activities in both research and clinical settings. Besides that, EEG is also well known for its fine temporal resolution that is essential for studying ERPs at milli-seconds time resolution. To understand the origin and the type of brain signal that can be collected from EEG, the following section provides a brief explanation on where and how the EEG signals arise neurally.

2.2.1 EEG Generation

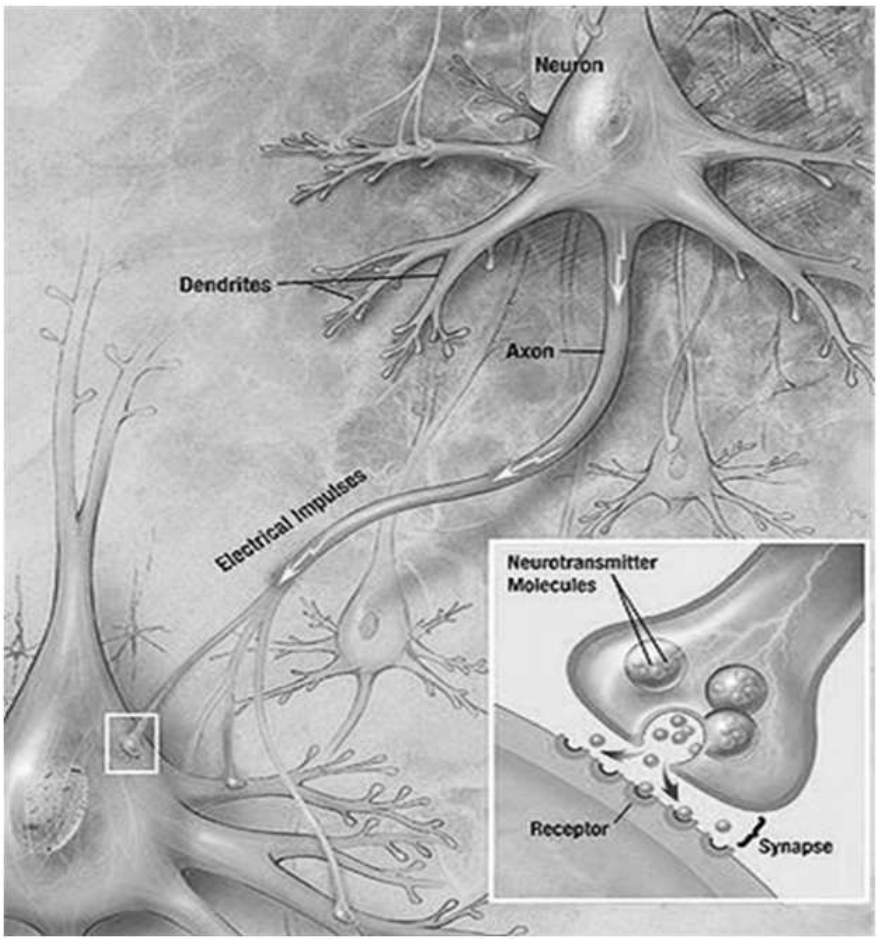

Figure 2.1: The structure of a neuron and the flow of the electrical activity from one neuron to the other neurons. Left bottom plot illustrates the structure of a synaptic cleft that is located between the presynaptic and postsynaptic terminal [17].

As shown in Fig. 2.1, the human brain is made up of billions of neurons which are interconnected to form a neural network through synapses [5][15][16]. When a neuron is activated, it triggers a series of depolarization and repolarization in part of the neural membrane along the axon. The resultant voltage change is known as the action potential. When an action potential arrives at the presynaptic terminal, it causes the release of neurotransmitters in the synaptic cleft. These neurotransmitters are then bound with the receptors and generates the potential changes in the membrane of the postsynaptic terminal. Here, these potential changes are known as postsynaptic potentials. Eventually, when a population of neurons has the same geometric orientation and experiences similar potential changes, the grand summation of these postsynaptic potentials are large enough to create an electric field that spreads to the scalp through

volume conduction. The changes of electric field that is captured by the electrode on the scalp is known as the Electroencephalography (EEG) signals.

It should be noted that the orientation of a neuron population decides whether their electrical activities cancel each other or combine to form a large signal. Thus, for this reason, EEG may not be able to capture all the electrical activities within the brain except for those that are organized with parallel orientations with each other and also perpendicular to the scalp. Based on these observations, current research suggests that EEG signals originate mostly from the pyramidal cells in the cortex of the brain [18][19][20].

2.2.2 EEG Recording

An EEG experiment setup involves the attachment of electrodes on the scalp, with conductive gel acting as a medium to provide the connectivity between the scalp and the electrodes. These electrodes are then connected to an amplifier for amplification before the sensed EEG signals are digitized through an analog-to-digital converter for display on a computer. In modern EEG recording systems, to ensure the naming and location of the electrodes are standardized across all EEG studies, the EEG electrodes are usually placed according to the international 10-20 system. The "10" and "20" normally refer to the distance between two adjacent electrodes from either front-to-back or right-to-left [4]. According to the system, the electrode location can be easily identified based on two simple rules. The first letter in each electrode represents the underlying area it covers on the scalp where F, T, C, P and O stand for the frontal, temporal, central, parietal and occipital lobes respectively. The following number or letter refers to the hemisphere location. Odd and even numbers represent the left and right hemispheres respectively while the letter 'z' (zero) represents the midline of the scalp. As there is an increasing trend to apply more electrodes in current EEG/ERP studies, an extension to this naming system is also introduced to accommodate new electrode locations as shown in Fig. 2.2 [4].

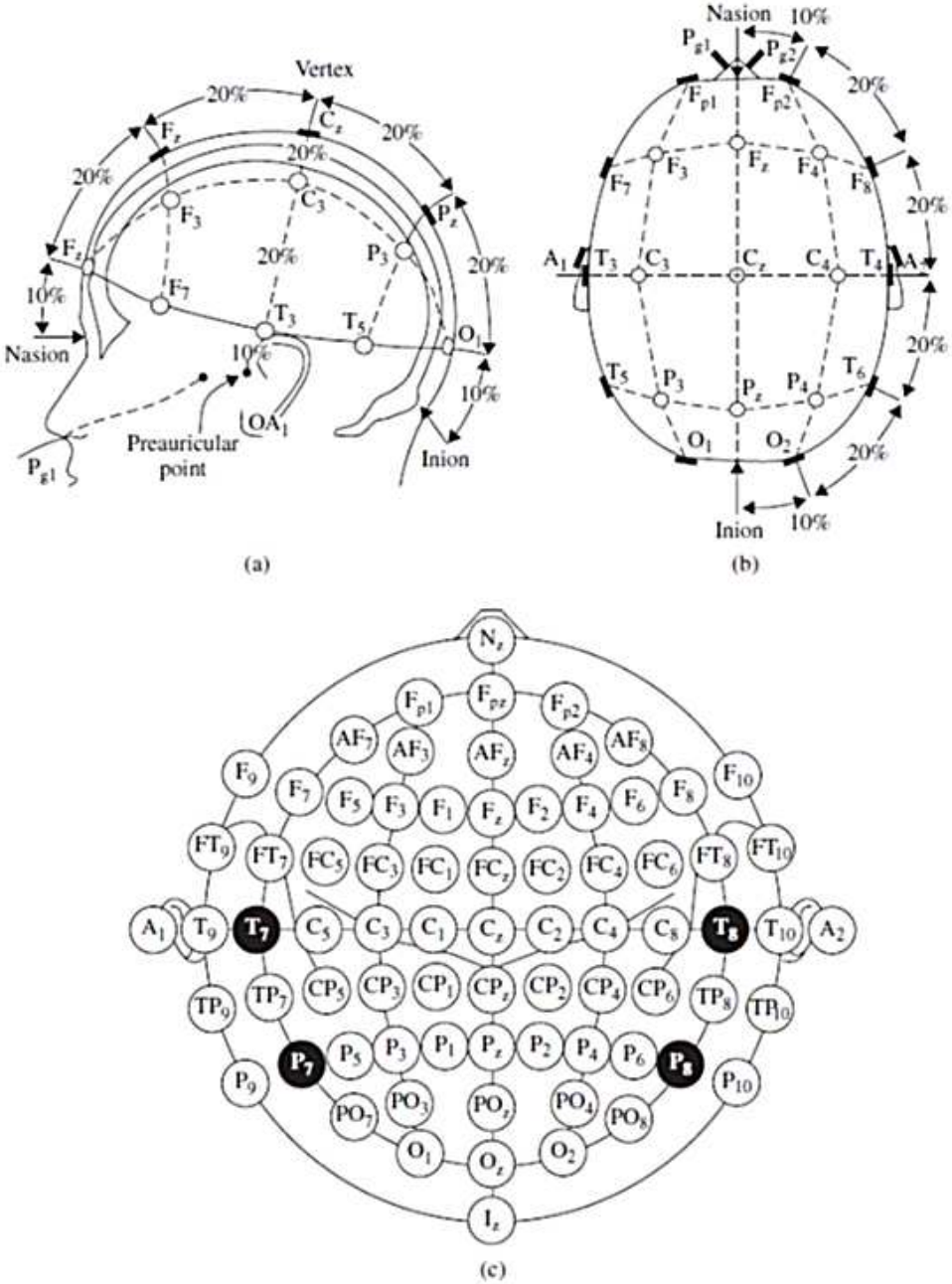

Figure 2.2: (a) Side and (b) top view of the international 10-20 system, (c) the extended version of international 10-20 system [4]

2.2.3 EEG Measurements

In the EEG recordings, the signal observed in each EEG channel is measured as the electrical potential difference between an active electrode and the reference electrode [6]. The process of subtracting the electrical activity of the reference electrode from the active electrode is known as referencing. The reference electrode is usually chosen

in such a way that it is neutral and contributes minimal amount of electrical activity. As such, it can serve as a good zero-voltage baseline for all EEG channels [21]. In the past, referencing is performed by connecting the active and reference electrodes physically to a differential amplifier. But, for modern EEG devices, such physical connection between the active and reference electrode is no longer required. In most cases, referencing is performed through software subtraction while the referencing EEG signal can be obtained in a number of ways depending on the application. In the literature, the best site for the reference electrode remains heavily debated [5][21]. As different choices of the reference electrode may alter the outcome of the observed EEG/ERP signals, it is usually advised to choose the reference electrode based on the guidelines provided by the corresponding community. In practice, the three popular methods are:

1. Electrode Cz
2. Average mastoid/ earlobe – the average signal across electrode A1 and A2
3. Average reference – the average signal across all EEG electrodes.

2.3 Event-Related Potential

Event-related potentials (ERPs) are neural activities that are generated milliseconds to hundreds of millisecond in response to a specific discrete event [5][6][16]. They are collected by recording EEG signals that begin 100 ms or more prior to the stimulus onset and terminated 500–2000 ms after the stimulus onset. By examining this segment of signal, often termed "trial" or "epoch[1]", ERPs can be observed as a series of voltage deflections that do not occur spontaneously but consistently time-locked to the stimulus onset. ERPs are important and are widely used for investigating the neural processes that govern different stages of information processing in the brain such as stimulus analysis, feature extraction, information fusion, decision making and motor response.

Fig. 2.3 shows the standard experimental procedure used to collect the ERP signals. In most ERP experiments, the stimuli are presented in a sequential order. The inter-stimulus interval is also chosen widely enough to allow the ERPs to be fully manifested before another stimulus takes place. For visual stimuli presentation, each stimulus

[1]In this thesis, the terms "trial" and "epoch" are used interchangeably.

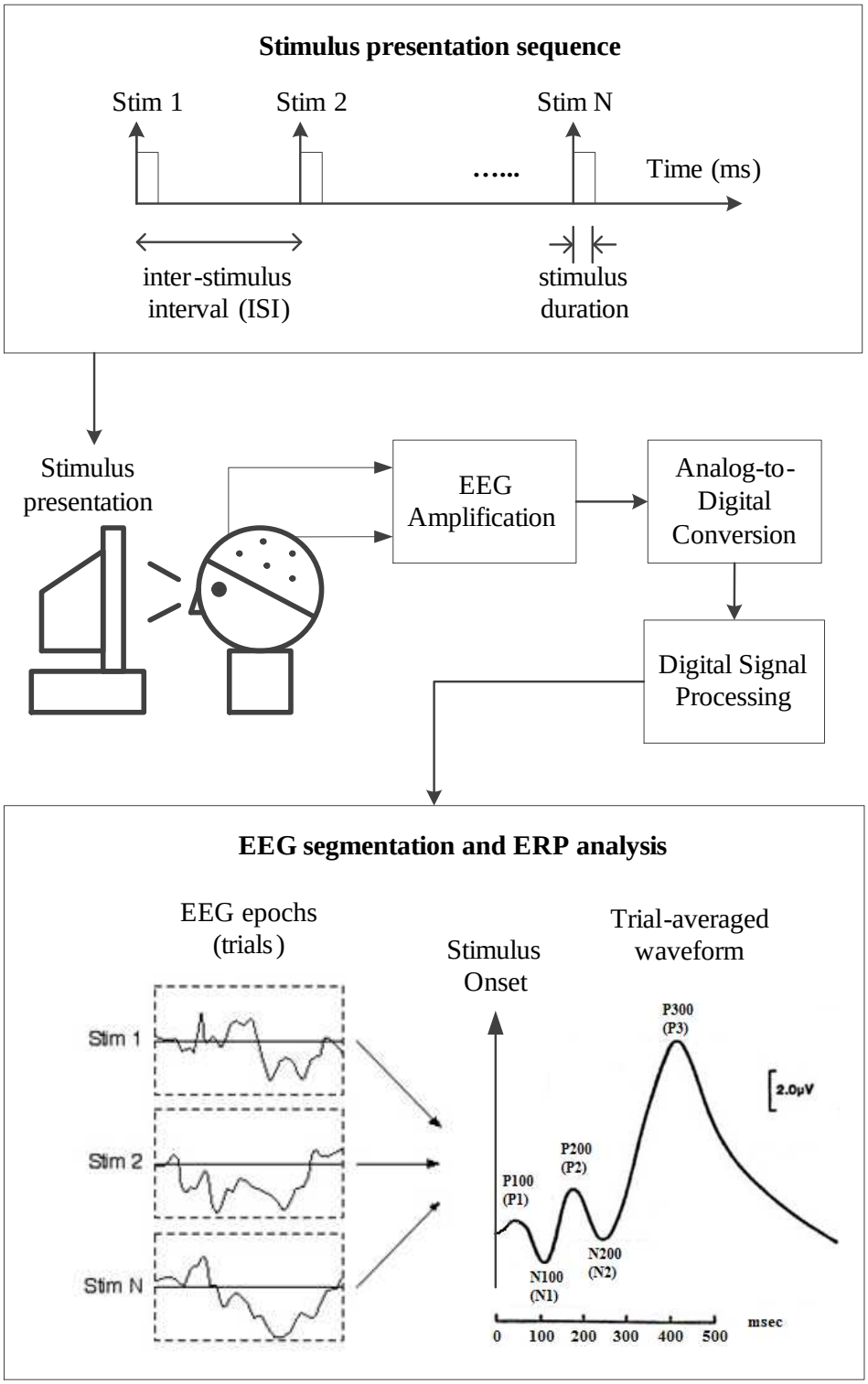

Figure 2.3: Different stages of an ERP experiment starting from the stimulus presentation, EEG recordings to the traditional ERP analysis based on a single EEG channel. (The bottom figure on the ERPs is adapted from [7].)

normally lasts around 50–150 ms, followed by a blank screen until the next stimuli begins. Since EEG signals are susceptible to other biological artifacts such as ocular and muscular artifacts, the subject are also instructed to minimize their eye and body movement throughout the experiment to prevent such artifacts from contaminating the EEG signals.

Currently, there are 10 or more different ERPs that have been discovered and studied [12][21][22]. Each of these ERPs are described based on their characteristics such as:

- polarity of the peak (positive or negative with respect to the baseline)
- amplitude (the maximum voltage difference between the baseline and the ERP peak)
- latency (the time that an ERP peak occurs after the stimulus onset)
- scalp distribution (the underlying cortex area where the ERP is pronounced)
- type of stimulus used to evoke the response (e.g. visual, auditory, somasensory)

To facilitate the studies, a common naming scheme is also introduced in which ERPs are named in terms of their polarity and latency (given either in milliseconds or as the ordinal position in the waveform after the stimulus onset). For example, a positive peak that occurs around 300 ms is called P300 or P3 while a negative peak that occurs around 200 ms is addressed as N200 or N2. However, it should be noted that the ERP peak latency can vary across trials, subjects and experiments. Thus, the time used for naming the ERP should only serve as the relative time range for an ERP.

ERPs can be divided into two major types, namely exogenous and endogenous [5]. An exogenous ERP is an ERP whose peak amplitude and latency depend on the physical characteristic of the stimulus. In contrast, the endogenous ERP is the ERP whose peak amplitude and latency depend on the internal state and capabilities of the subject (e.g. attention, arousal, memory performance, etc.). Most endogenous ERPs can be generated by different type of stimulus as long as the task condition used to generate the ERP are fulfilled. Below lists the different types of ERP that will be considered throughout this thesis.

- P100 or P1

  P100 is one of the exogenous ERPs that can often be seen as a positive peak that occurs around 90-110 ms after the stimulus onset [23]. Its occipital scalp distribution suggests that it reflects the neural activity from the visual cortex after a brief visual stimulus is presented. Experiment that study P100 usually involves the luminance change and pattern reversal of the visual stimuli. Studies showed that P100 can be recorded from infants and its latency gradually shortens to around 100 ms as the infant grows up to the age of 4-5, which suggests that the human visual system matures around the age of 4-5 [23]. However, after the age of 55, the P100 amplitude attenuates and its latency increases, suggesting that the performance of the visual system generally decrease with age.

- N170 or visual N1

  When a more complex visual stimuli such as pictures are used, another more intriguing ERP called N170 can also be observed. N170 is a negative voltage deflection which peaks at the occipital-temporal region around 170 ms of the stimulus onset. Unlike P100, instead of being affected by the low-level physical characteristic of the stimulus, N170 is typically associated with high-level processes especially the perception of the stimulus as a face. N170 research is relatively new whereby the first systematic ERP studies of face processing is conducted around 1990 by [24][25]. Numerous studies showed that N170 presents a large negative peak whenever the displayed image can be perceived as a face. This effect is normally larger when comparing between face and non-face stimulus. The association of N170 with face processing can also be observed from numerous studies which manipulated the stimulus by inverting, segmenting or isolating a particular face component such as eyes [26]. Although many researchers confirmed the relationship of N170 with face processing, many questions related to N170 remains unclear such as which facial features or properties have more weight in the face representation, how these features are integrated (either local-to-global representation or global-to-local finer representation) and what processes occur during the N170 time window [26][27].

- P300 or P3

  P300 is one of the most studied ERPs. It was first discovered by Sutton in 1965

[28]. P300 can be easily identified by its large positive peak that typically occurs around 250–500 ms after stimulus onset [29]. In terms of scalp distribution, P300 is central-parietal distributed, with the peak mostly at the parietal region. The experiment that is commonly used to generate P300 is called the oddball paradigm. It usually involves two different stimulus that are sequentially and randomly displayed on the screen. During the experiment, the subject is instructed to respond to the infrequent target stimulus by either mental counting or pressing a button while ignoring the frequently occurring non-target stimulus. Under such condition, a large P300 peak can be observed whenever the infrequent target stimulus is presented. Moreover, unlike the previously mentioned ERPs, P300 can also be observed under different presentation modalities, which is also the reason that P300 is categorised as an endogenous ERP.

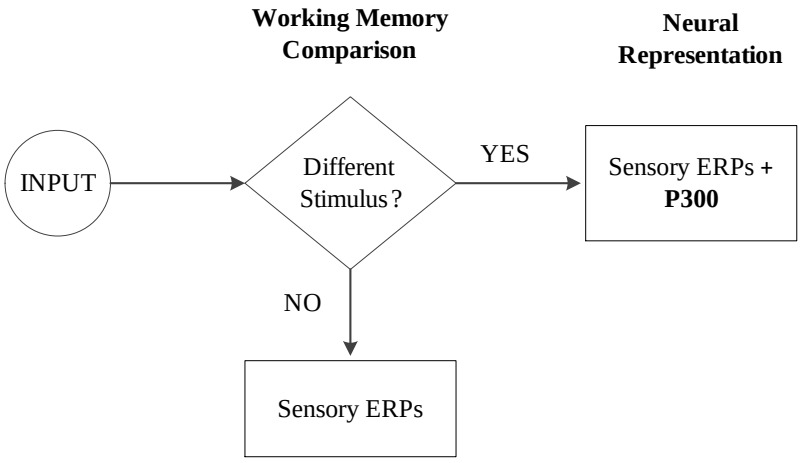

Figure 2.4: The context-updating mechanism that hypothesizes the presence of a P300 ERP together with the sensory ERPs (e.g. N100, P200, N200) in the infrequent task-relevant trials as proposed by Polich [29].

A more established hypothesis suggests that P300 reflects the context updating mechanism of the brain as shown in Fig. 2.4. Based on this theory, our brain is constantly comparing the current stimulus with the representation of the previous stimulus that has been stored in the working memory. Whenever there is a change in the context of the stimulus, an updating of the stimulus representation in the working memory is initiated and thus P300 is produced. Since our brain only engages more resources towards newly-detected changes, habituation occurs whenever the same stimulus is presented successively. This effect can be observed by manipulating the target-to-target stimulus interval [29] whereby

the peak amplitude of P300 generally decreases as the target-to-target interval decreases.

The peak amplitude of P300 is often used to index the size of the neuronal activity induced by the internal and external cognitive factors while its peak latency is often used to index the stimulus classification speed. For example, in a dual task experiment where the primary task is to perform cognition while the secondary task is to mentally count the target oddball stimuli, P300 normally experiences a decrease in its peak amplitude and also an increase in its peak latency whenever the primary task difficulty increases [29]. One possible reason for this phenomenon is that whenever more processing resources are required for task performance, the amount of resources allocated for the "updating" task reduces, thus the amount of time required to process a target stimulus increases. Apart from that, the peak amplitude and latency of P300 are also used in other studies to understand factors such as drugs, age, intelligence, personality, fatigue and psychiatric disorder, etc. [29].

It should be noted that besides the ERPs mentioned above, there are also other types of ERPs that have been extensively studied. The information regarding these ERPs as well as their applications can be obtained in [12][22][31].

2.4 Application of ERPs in Brain Computer Interface

Brain Computer Interface (BCI) is a system that provides users with a mean to communicate with and control the external world by using only their brain signals[2] [13]. Over the last two decades, the development of BCI has drawn a large amount of multidisciplinary researches where various methodologies, experimental paradigms, brain signals have been experimented for their usability and practicality as the BCI system. The emergence of BCI technology has also brought forth a significant change to the EEG signal processing and pattern recognition field where the development of new single-trial techniques have led to the significant improvement in the speed and prediction

[2]Here, the brain signals are not limited to the EEG measurements. It can be any brain activity measurement collected by other neuroimaging devices such as fMRI and fNIRS. However, EEG remains largely employed in BCI because it has been considered as the economically viable and technologically mature solution to be used in the household in the near future.

accuracy of the BCI.

Currently, most of the present BCI applications are designed primarily to assist paralyzed people[3]. In many cases, these BCI applications are designed to be either a basic word-processing system or a control system that is linked to the motorized wheelchairs, robotic arms and limb prostheses. To control these systems, the user must generate a brain signal that encode his/her intent so that the BCI translation algorithm can convert this signal into command to perform the user's intent.

Over the years, the four major types of EEG signals that have been found promising for reflecting the user's intent are steady-state visual evoked potential (SSVEP), slow cortical potential (SCP), sensorimotor rhythm and P300 ERP [32]. Among these EEG signals, the P300 ERP is one of the signals that has been extensively used by the BCI community where its applications can be found in both word-processing and control systems. P300-BCI is an attention-based BCI application which requires a user's constant attention towards the presented stimuli. In general, most P300-BCI applications are designed with slight modification on the oddball paradigm. The BCI interface usually presents a multiple number of choices on the screen. During the presentation, each choice is randomly highlighted. To select the intended choice, the user is required to stay focus on the desired choice. Whenever the desired choice is highlighted, a P300 ERP will be elicited. Thus, the goal of P300-BCI is to identify which highlighted choice evokes the P300 so that a command can be issued. In [33], with a slight modification, Donchin demonstrated that this type of presentation paradigm can be further extended to contain 36 different numbers of alphanumeric characters. As a result, his work also laid the foundation to the today's row/column based P300 spellers (which will be discussed in greater detail in Chapters 5 and 6).

Compared to other BCIs, the P300-BCI has three major advantages. Firstly, the P300-BCI is suitable for most of the people mainly because the generation of P300 ERP does not require any intensive training. Secondly, the P300 ERP can be elicited under different types of stimulus. Thus, besides the visual presentation paradigm, there are some efforts which attempt to develop an audio or tactile-based P300-BCI to benefit those patient with visual impairment [34][35]. Thirdly,the P300 ERP can be elicited within split seconds. Thus, ideally, if every individual ERP can be detected

[3]Here, the paralyzed people refers to those who suffer severe motor disorders (e.g. spinal cord injury, amyotrophic lateral sclerosis, brain stem stroke, cerebral palsy and muscular dystrophy.

successfully in each trial, the P300-BCI can achieve a remarkable communication rate when compared to other types of BCI. However, one of the problems with the P300-BCI is its fast[4] stimulus presentation rate. Sometimes, the rapid 'flashing' on the screen can strain the eyes and makes the user feel uncomfortable. For this reason, currently, there are also numerous attempts to improve the presentation paradigm by modifying the presentation rate and also the color, size and figures used to represent the choices [36][37][38][39].

2.5 Summary

ERPs are an important subclass of EEG signals that reflect neural activities which are generated in response to an event. They can be broadly categorised by their peak polarity, latency and scalp distribution. Each ERP carries important information about how humans sense, perceive and react to a changing environment under a fraction of a second. Unfortunately, ERPs are usually weaker signal compared to the on-going EEG activities. In order to analyze or detect an ERP effectively, signal processing techniques commonly used to obtain the ERP-of-interest are discussed in the next chapter.

[4]The inter-stimulus interval adopted in the ERP-based BCI is usually shorter (e.g. 175 ms) when compared to the one used in the traditional oddball paradigm (e.g. 1000 ms) [33]. As such, it allows more stimuli presented within a second while waiting the ERP to be fully manifested.

Chapter 3

Signal Processing and Pattern Recognition for ERP Application

3.1 Overview

As discussed in Chapter 1, the focus of this thesis is to develop and evaluate methods to extract ERPs. Our proposed methods are based on a number of signal processing and pattern recognition techniques that have been applied in various ERP fields such as ERP extraction, classification and segmentation. To lay the foundation for the different types of techniques used in this thesis, this chapter provides an overview of the current state of the art in each of the respective fields and also the mathematical background for some of the techniques that are used in this thesis.

This chapter is organized as follows. Section 3.2.1 gives a description of the EEG signal pre-processing techniques that are customarily applied to EEG recordings. In Section 3.2.2, the issues faced by traditional ERP studies are explained, while in Section 3.2.3, the recent development in ERP extraction methods are discussed. This is then followed by a review on the EEG model and the existing multi-channel ERP extraction methods in Sections 3.2.4 and 3.2.5 respectively.

After that, special attention is paid to various pattern recognition techniques such as classification and clustering techniques as used in ERP processing and analysis. In Section 3.3.1, an overview on the role of pattern recognition techniques in EEG signal processing field is explained, while in Section 3.3.2, a brief review of the current ERP classification system used in BCI is provided. The ERP classification system is also

used in Chapters 5 and 6 for assessing the performance of our proposed extraction methods. In Section 3.3.3, a brief introduction to the ERP segmentation problem is presented. A solution to this problem is described in Chapter 6 and this solution is critical to the development of a fully-autonomous ERP extraction method in Chapter 6. Lastly, Section 3.4 summarises the review.

3.2 Signal processing for ERP extraction

3.2.1 EEG signal pre-processing

One of the fundamental problems in ERP signal processing is to extract the waveform of the ERP-of-interest so that ERP classification and estimation (e.g. peak amplitude and latency estimation) can proceed successfully. ERP extraction remains a challenging issue mainly because ERP tends to be relatively low in voltage level, ranging between 0 and 10 μV when compared to the background EEG and noise artifacts that ranges between 0 and 100 μV [7].

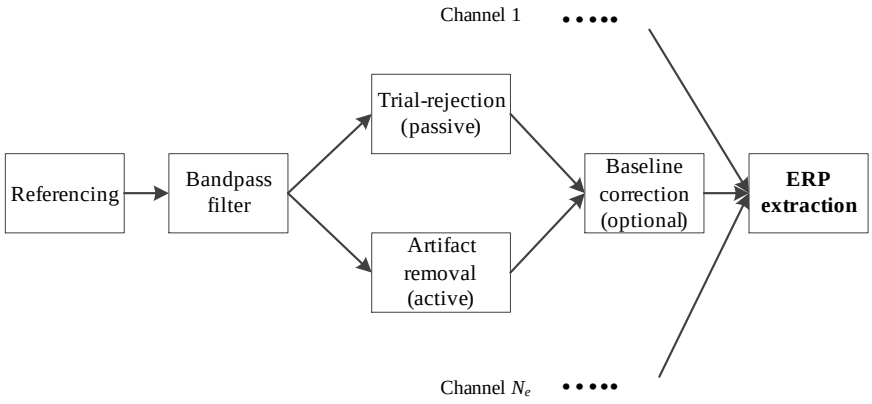

Figure 3.1: Block diagram of different commonly-used EEG pre-processing techniques

Before the EEG signals can be used to study the ERP-of-interest, they must be pre-processed. Fig. 3.1 shows the typical procedure used to extract the ERP-of-interest in most multi-channel ERP studies. Techniques such as referencing, bandpass filtering and baseline correction have become customary to minimize the noise artifacts in the EEG signals.

1. **Referencing**

The first stage of ERP signal processing normally begins with referencing. Referencing is a process of subtracting signals between the active and reference electrodes so that the common noise signals that appear in all the active electrodes and the reference electrode can be reduced [41]. As shown in Fig. 3.2(a), before referencing, EEG signals measured from the active electrodes are noisy partly due to strong interference from nearby power line. The same noisy signal is also captured by a reference electrode as shown in Fig. 3.2(c). After referencing, Fig. 3.2(e) shows that the common noise signals are greatly reduced in the measured EEG signal.

2. **Bandpass filtering**

   After referencing, some EEG artifacts such as baseline drift and power line interference may still remain in the measured EEG signal. To suppress these artifacts, bandpass filtering (e.g. 0.5–30 Hz) is applied so that any signals that are outside the frequencies of interest are removed. Fig. 3.2(g) shows an example of applying low-pass filter to suppress the remnants of power-line interference while Fig. 3.2(h) shows the EEG signal after applying both low-pass and high-pass filtering. It can be seen that after applying high-pass filter, the baseline drift is removed and the EEG signal is now settled at zero baseline.

3. **Trial rejection and artifact removal**

   Often, after these procedures are applied, the EEG signals are divided into trials (or epochs) based on the stimulus onset. At this point, if the EEG signals are still contaminated by strong noise artifacts, trial-rejection can be applied to discard the noisy trials from further analysis. However, trial-rejection reduces the number of usable trials and is a problem when the number of trial is limited in the first place. For this reason, an active artifact removal scheme is normally applied. A common example is the ocular artifact removal to suppress the noise artifacts caused by eye movements and eye blinks [42].

4. **Baseline correction**

   Before extracting the ERPs, baseline correction is commonly applied by subtracting the mean voltage of the EEG segment prior to the stimulus onset (e.g. -200 ms – 0 ms) from each trial [21]. As shown in Fig. 3.3, baseline correction

ensures the signal settled to zero baseline in every trial so that the ERP peak amplitude can be estimated correctly and is comparable across different trials.

Note that depending on the experimental design, the ways of applying the EEG pre-processing techniques may vary slightly from Fig. 3.1. After minimising the noise artifacts, additional signal processing is still required to recover the ERP-of-interest from the noisy background EEG. The process of recovering the ERP waveform is known as ERP extraction which is the main focus of this thesis.

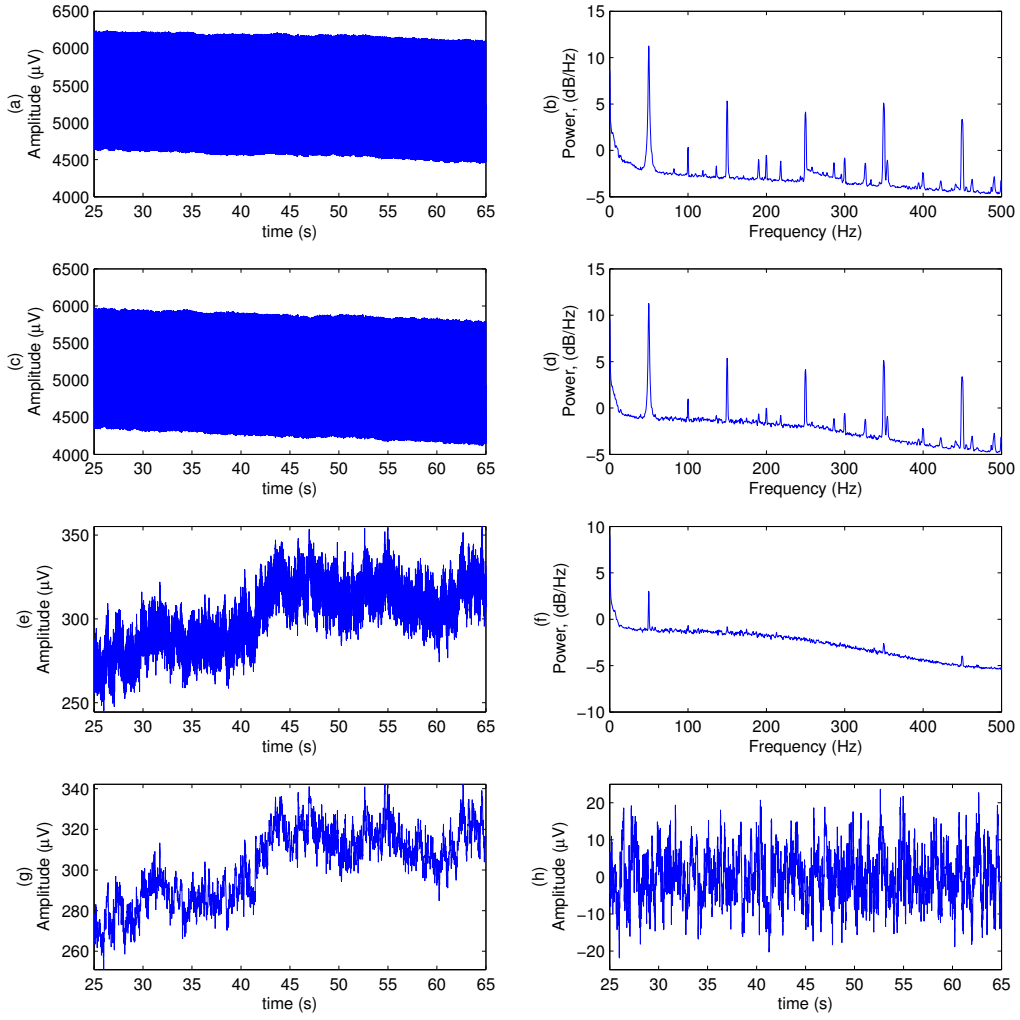

Figure 3.2: (a) EEG signal measured from an active electrode, (b) Power Spectrum of the measured EEG signal, (c) signal measured from a reference electrode, (d) Power Spectrum of the signal from reference electrode, (e) EEG signal after referencing, (f) Power Spectrum of the EEG signal after referencing, (g) EEG signal after low-pass filtering (cutoff frequency at 30 Hz), (h) EEG signal after both low-pass and high-pass filtering (cutoff frequency at 0.5 Hz)

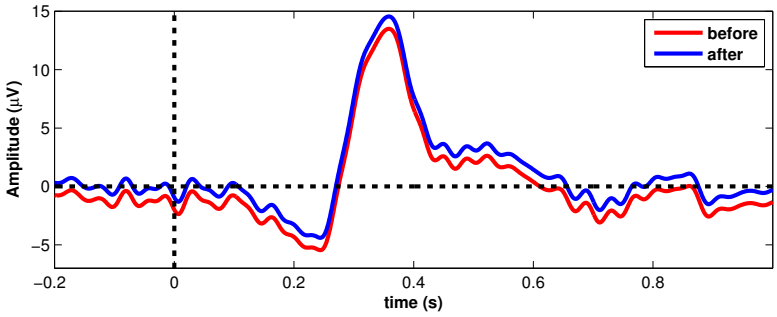

Figure 3.3: Trial-averaged ERP before and after baseline correction

3.2.2 Issues in the traditional ERP extraction methods

As discussed earlier, ERP tends to have lower signal-to-noise ratio (SNR) and thus easily affected by the background EEG signals. In order to improve the signal quality, the traditional approach is to average the ERP waveforms across trials [5][6][16]. By assuming the ERP waveform and peak latency are fixed across trials, trial-averaging technique provides an overall representation of ERPs by suppressing the irrelevant signals that are not time-locked to the stimulus. This technique has led to the discovery of many ERPs in the past and remains popular in current ERP studies.

However, there are a few drawbacks with the aforementioned trial-averaging technique. Firstly, trial-averaging assumes the ERP waveforms are invariant across trials. This assumption is not valid because numerous findings have suggested ERPs can vary greatly even under the same stimulus condition [9][10]. In addition, a person's physiological conditions, such as mental fatigue, habituation and attentiveness can also contribute towards trial variations in the ERPs. Secondly, the trial-averaging technique inhibits trial-to-trial analysis where important information, which may reflect the changing pattern of ERP amplitude and latency across trials, is lost because of the averaging process. A simple example that illustrates the importance of single-trial ERP extraction is shown in Fig. 3.4. As can be seen, the trial-averaged signal from both cases (a) and (b) are similar although the underlying ERP waveforms are different across trials. In case (a), each individual ERP has different peak amplitude but constant peak latency. In contrast, the opposite happens in case (b) whereby the peak latency varies across trials but not the peak amplitude. However, these valuable information cannot be recovered from the trial-averaged signals.

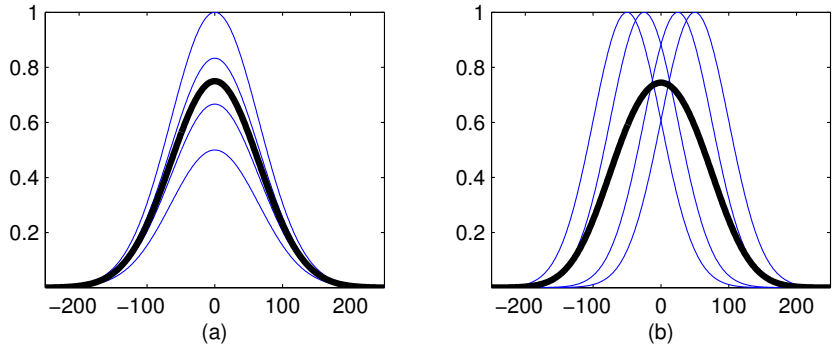

Figure 3.4: (a) the trial-averaged signal (black) when amplitude variation occurs in each trial (blue), (b) the trial-averaged signal (black) when latency variation occurs in each trial (blue),

3.2.3 Recent development in ERP extraction

Over the years, the advancement in computer technology and EEG recording devices has led to a drastic change in the way ERP is studied. For this reason, the ERP extraction methods have also experienced changes from the early trial-averaging based techniques to the newer single-trial methods. In the early days, most ERP signal processing algorithms focused primarily on improving the trial-averaging technique. To minimise the effect of noisy trials during averaging and also address the ERP peak latency variation across trials, different trial-averaging algorithms such as weighted-averaging and latency-dependent averaging were proposed[1] [11][43][44]. Soon after, to provide an in-depth study on the ERP, single-trial extraction methods began to receive attention [45][46]. The urgent need for single-trial methods can also be attributed to the newly emerged technology called brain-computer interface (BCI) which requires fast and reliable single-trial ERP extraction to enhance the ERP signals.

Among the single-trial methods, a few notable single-trial single-channel extraction methods have been proposed. For example, wavelet denoising method has been introduced in [47] and [48] to remove the spontaneous EEG so that the desired ERP can be estimated reliably. In [49][50], a framework which treats each EEG epoch as a linear combination of multiple ERP waveforms and on-going activity were proposed where the latency, waveform and amplitude of each ERP are estimated with the assumption that the waveform of each ERP is invariant across trial. In [51], the author

[1]These trial-averaging techniques are single-channel methods.

applied a subspace-regularized least square method with Gaussian-shaped basis function to recover the ERP waveform from each trial.

Single-trial single-channel extraction method remains important in ERP studies. However, over the last two decades, single-trial multi-channel extraction methods have received increasing attention mainly due to the advancement in the EEG recording device and the computer technology. The development of single-trial multi-channel extraction method is essential to ensure the multi-channel EEG recordings can be utilised effectively to examine the spatial and temporal aspects of the ERP.

3.2.4 Linear generative EEG model

The linear generative EEG model is a model that has been widely adopted[2] in most multi-channel extraction methods for representing the EEG signals [53]. Based on this model, the multi-channel EEG signals are given as:

$$\mathbf{x}(t) = \mathbf{A}\mathbf{s}(t) \tag{3.1}$$

where $\mathbf{x}(t)$ is a $N_e \times 1$ vector representing measurements from N_e electrodes, $\mathbf{s}(t)$ is a $N_s \times 1$ vector representing N_s EEG source signals and $\mathbf{A}$ is a mixing matrix with dimension $N_e \times N_s$.

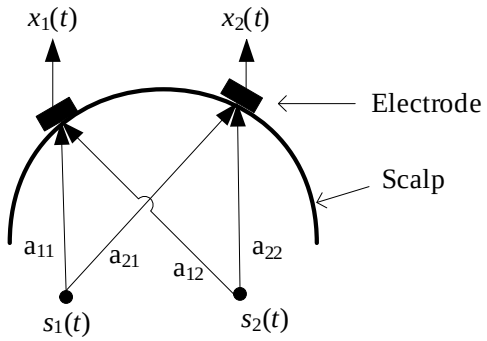

Figure 3.5: linear generative EEG model

The linear generative EEG model assumes the observed signal from any electrode is basically the linear and instantaneous combination of different available source signals. By taking Fig. 3.5 as an example, the observed signals from the two electrodes are given as:

$$x_1(t) = a_{11}s_1(t) + a_{12}s_2(t) \tag{3.2}$$

[2]Based on the author's knowledge, currently there is no other model (e.g. non-linear model) that has been developed for representing the multi-channel EEG signals.

$$x_2(t) = a_{21}s_1(t) + a_{22}s_2(t) \tag{3.3}$$

Or in the framework of (3.1),

$$\begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} s_1(t) \\ s_2(t) \end{bmatrix} \tag{3.4}$$

From Eq. (3.4), the generation of EEG signal can be explained by the mixing matrix **A** which is sometimes known as the forward model. Each row vector in **A** explains how different source signals are linearly combined to form the EEG signal as observed in any electrode. Meanwhile each column vector in **A** is the mixing vector (or scalp distribution) of each individual source signal. It explains the contribution of a source signal towards all electrodes after passing through different mediums such as brain tissue, skull and skin.

The linear and instantaneous assumptions are reasonable mainly because at the macroscopic level and EEG frequency range, the brain tissue is primarily a resistive medium governed by Ohm's law, with negligible capacitance effect [15][20]. By Maxwell's equation, the resistivity of the medium implies that [15]: (i) at any moment in time, all the fields are generated by the active current sources without significant time delay, and (ii) the potential measured at one point can be treated as the summation of potentials contributed by different sources .

3.2.5 Spatial filters

A major component of most multi-channel extraction methods is the spatial filter. Assuming the source origin of the desired signal is spatially fixed across trials, one of the methods to extract the desired signal is to find a weighting vector **w** that linearly combines information from multiple channels to obtain an aggregate estimate of the desired signal. This weighting vector is also known as the spatial filter:

$$y(t) = \hat{s}(t) = \mathbf{w}^\top \mathbf{x}(t) = \sum_{i=1}^{N_e} w_i x_i(t) \tag{3.5}$$

where $(.)^\top$ denotes transpose, $y(t)$ defines the extracted signal, i represents the channel index, N_e is the total number of electrodes, while w_i and $x_i(t)$ are elements of the vectors **w** and $\mathbf{x}(t)$ respectively.

To determine an appropriate spatial filter, different assumptions are often made on the ERP signal characteristics, which subsequently leads to data-independent and data-driven extraction methods as illustrated in Fig. 3.6.

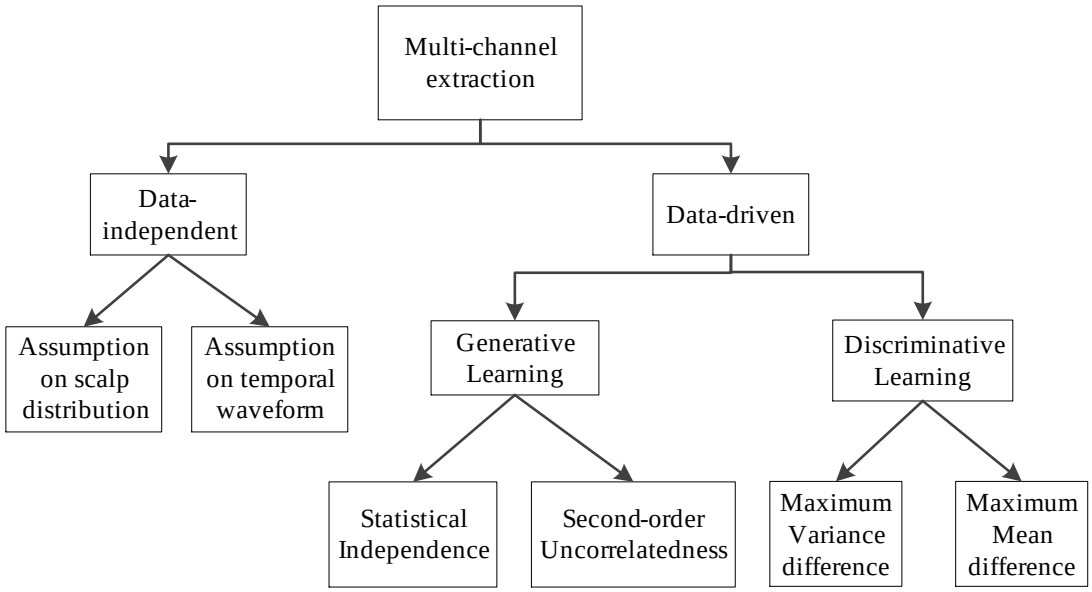

Figure 3.6: Types of multi-channel ERP extraction methods and the assumptions commonly used to determine the spatial filter.

Data-independent methods usually assume either the mixing matrix or the temporal waveform of the desired ERP is known beforehand. To apply these methods successfully, a strong background knowledge of the desired ERP is hence required. In practice, these methods can be difficult to implement since the ERP waveform and its source location in the brain may vary across subjects and experiments. To avoid such problems, data-driven methods which operate based on the statistics of the measured ERP signal have been proposed.

In general, the data-driven methods require some assumptions about the ERP signal, such as the uncorrelatedness, statistical independence, maximum variance and mean difference of the ERPs. Based on these assumptions, the methods can be further divided into *generative* and *discriminative* learning methods. For the generative learning methods, they attempt to find a spatial filter $\mathbf{w}$ that minimises the reconstruction error between the measured EEG signals and the extracted ERP source signal. Examples of generative learning methods are Independent Component Analysis (ICA) and Principal Component Analysis (PCA).

In contrast, for the discriminative learning methods, they usually assume that there

exists two sets of signals that may result from two different experimental conditions (i.e. target and non-target trials). Among these two conditions, the ERP-of-interest tend to appear stronger in one experimental condition but not another. For this reason, discriminative learning methods aim to determine a spatial filter **w** that will produce a signal whose presence indicates the presence or absence of the ERP-of-interest. Often, the spatial filter is designed on the basis of maximising the difference in some sense between two conditions. Examples of generative learning methods are Max Signal-to-Noise (SNR) Beamformer, Common Spatial Pattern (CSP), xDAWN and Linear Discrimination (LD). In practice, discriminative spatial filters are restrictive in the sense that to work effectively, it requires the ERP-of-interest to be significantly different in both experimental conditions. Thus, instead of ERP analysis, they are more commonly applied in BCI applications to extract a specific kind of ERP (e.g. P300).

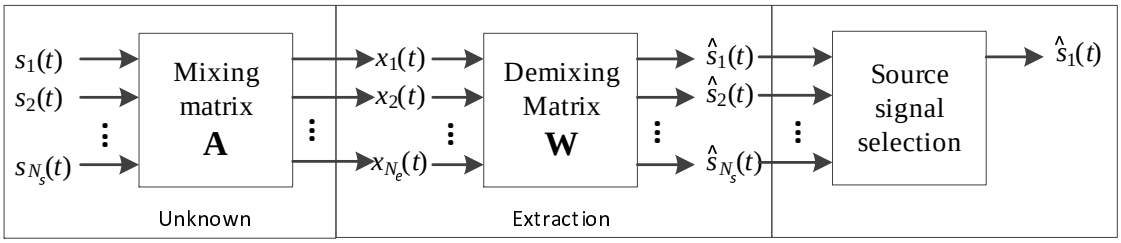

Figure 3.7: Typical procedure used by data-driven methods to extract the ERP-of-interest.

Most data-driven methods are not designed specifically for extracting the ERP-of-interest. As shown in Fig. 3.7, during extraction, a set of spatial filters **W** is usually determined to decompose the EEG signals simultaneously into a number of source signals. Consequently, an additional source signal selection stage are required before the ERP-of-interest can be obtained. It should be noted that the extraction process in Fig. 3.7 is performed under the "square-mixing" assumption, that is the number of source signal N_s is equal to the number of EEG channels N_e. Clearly, from the EEG signal perspective, the "square-mixing" assumption is not applicable since generally $N_s \neq N_e$. A further complication is that often the N_s itself is not known. To make the data-driven extraction problem more tractable, the "square-mixing" assumption is commonly employed [54][55].

In the following sections, a few popular multi-channel extraction methods will be briefly described.

3.2.5.1 Extraction based on prior knowledge of scalp distribution

Assume that the ERP-of-interest is equally mixed at different electrodes, the simplest form of the spatial filter is to average the EEG signals uniformly across every available channels:

$$\mathbf{w}_{\text{uni}} = \frac{1}{N_e}[1, 1, ..., 1] \tag{3.6}$$

where $\mathbf{w}_{\text{uni}}$ is a $N_e \times 1$ uniformly weighted vector.

Alternatively, if the most active EEG channels for the ERP-of-interest are known beforehand, the spatial filter can also be adjusted in such a way that more weight are given to these channels. An example of this approach is called Laplacian spatial filter (see [56] for more details).

3.2.5.2 Extraction based on prior knowledge of temporal waveform

Suppose the waveform of the ERP-of-interest $s(t)$ is known. Let $\boldsymbol{s}$ be the $1 \times N_T$ vector that represents the time course of $s(t)$, and $\mathbf{X}$ be the multi-channel EEG signals with dimension $N_e \times N_T$ and , the optimization problem that yields the spatial filter based on Least-Square criterion can be defined as follows [57][58]:

$$\min_{\mathbf{w}} \quad \|\mathbf{w}^\top \mathbf{X} - \boldsymbol{s}\|^2 \tag{3.7}$$

Assuming the ERP-of-interest is the major signal component in the recorded EEG signals and it is uncorrelated with other EEG source signals, the solution to the above optimization problem is given as [57]:

$$\mathbf{w}_{\text{tm}} = (\mathbf{X}\mathbf{X}^\top)^{-1}\mathbf{X}\boldsymbol{s}^\top \tag{3.8}$$

In practice, it is impossible to obtain the actual ERP source signal $s(t)$. Thus, to determine the spatial filter, a template $\boldsymbol{r}$ that resembles the ERP waveform is normally used. The design of the ERP template is very important for this method. To reduce the estimation error, there should be a minimal mismatch between the ERP template and the actual ERP waveform of any individual trial.

In this section, the extraction method proposed by [57] will be discussed in greater detail since it is chosen for comparison purposes in Chapter 4. In [57], assuming the ERP waveforms are invariant across trials, the author proposed the ERP waveform can be represented by some signal functions such as the Gaussian or Gamma functions

which resemble the actual ERP waveform. Since the ERP peak latency may vary across trials, in order to perform the extraction successfully at the single-trial level, the author proposed scanning a predefined template across time so that the time location which the pre-defined template best matches the actual ERP of each individual trial can be located. The general algorithm for Li's single-trial ERP extraction is described as follows [57]:

1. Design an ERP template $\boldsymbol{r}(\tau)$ with its peak located at time instant τ.
2. Based on the given $\boldsymbol{r}(\tau)$, determine the spatial filter $\mathbf{w}_{\text{tm}}(\tau)$ using (3.8) where $\boldsymbol{s} = \boldsymbol{r}(\tau)$
3. Calculate the least square error between the extracted signal using the cost function $J(\tau) = \|\mathbf{w}_{\text{tm}}(\tau)^{\top}\mathbf{X} - \boldsymbol{r}(\tau)\|^2$.
4. Repeat Steps 1–3 until every time instant is investigated.
5. Find the value of τ where $J(\tau)$ is a minimum. Let τ_0 denote the τ thus found. τ_0 corresponds to the estimated peak latency of the ERP of that trial while the respective $\mathbf{w}_{\text{tm}}(\tau_0)$ is the spatial filter for extracting the ERP-of-interest of that trial.

As discussed in Chapter 2, while ERPs are characterised by their polarity, scalp distribution and latency, they cannot be categorised by their temporal waveforms. Thus, in practice, the above method often experiences difficulty in locating the temporal waveform of the ERP-of-interest. This problem is mentioned in [57] and is also demonstrated in part of the comparison study in Chapter 4.

3.2.5.3 Independent Component Analysis

Independent Component Analysis (ICA) is a type of generative learning method that attempts to decompose the noisy mixed signals into a set of source signals. ICA is widely used in different signal processing fields and has also been described in detail in many publications [54][55][59][60][61]. In EEG application, ICA is widely used because of its good performance in removing artifacts and extracting source signals [62][63][64][65][66].

Similar to the linear generative EEG model, the basic[3] ICA model assumes the observed signals are linear combinations of the source signals [59]. In order to recover the original source signals, ICA makes the key assumption that the source signals are statistically independent, that is the time course of $s_i(t)$ does not provide any information about the time course of $s_j(t)$ for $j \neq i$. Driven by this assumption, ICA tries to find a set of spatial filters[4] $\mathbf{W}_{\text{ica}}$ such that the resulting signals $\hat{\mathbf{s}}_{\text{ica}}(t) = \mathbf{W}_{\text{ica}}^{\top}\mathbf{x}(t)$ are maximally independent. To estimate statistical independence, different measures such as kurtosis, negentropy and mutual information are proposed and further led to the variants of ICA algorithms such as FastICA [59], JADE [67] and Infomax ICA [68].

Despite the ability to recover individual source signals, ICA has two limitations [59]. These are (i) the sign and scaling ambiguity, and (ii) the permutation ambiguity in its extracted signals. First, since both the mixing matrix $\mathbf{A}$ and actual source signals $\mathbf{s}(t)$ are unknown, any scalar multiplier in one of the source signal $s_i(t)$ can always be canceled by the division of the corresponding column $\mathbf{a}_i$ of $\mathbf{A}$ by the same scalar, without making any changes on $\mathbf{x}(t)$. For example, $\mathbf{x}(t) = (\frac{1}{\alpha}\mathbf{A})(\alpha\mathbf{s}(t))$. In practice, this problem is usually fixed by maintaining each source to have unit variance: $\mathbb{E}\{s_i^2(t)\} = 1$. However, this sign ambiguity remaining unsolved whereby both $\mathbf{A}$ and $\mathbf{s}(t)$ can be multiplied by -1 without affecting the model. In practice, if the sign of the ERP is known beforehand, the sign of the respective source signal can also be changed manually [69]. Similarly, the permutation ambiguity also arises because the order of rows in $\mathbf{s}(t)$ and columns in matrix $\mathbf{A}$ can be randomly changed without any effect on the result. Formally, it means that a permutation matrix $\mathbf{P}$ and its inverse can be substituted in the model to give $\mathbf{x}(t) = \mathbf{A}\mathbf{P}^{-1}\mathbf{P}\mathbf{s}(t)$. This ambiguity is less problematic for most applications as long as the signals are extracted correctly. In fact, although less mentioned in the literature, the aforementioned ambiguities are also applied to other data-driven methods such as PCA, max SNR beamformer, common spatial pattern and xDAWN that will be discussed in the following sections.

[3]There exists other type of ICA called convolutive ICA which assumes that the mixing of different source signals involves convolution and time delays. This type of ICA is often applied in speech signal processing field [60].

[4]In the ICA literature, the resultant set of spatial filters is also known as demixing matrix.

3.2.5.4 Principal Component Analysis

In EEG application, principal component analysis (PCA) is a method which attempts to decorrelate a set of measured signals into a set of uncorrelated signals called principal components through orthogonal transformation [70]. Without loss of generality, let the multi-channel signals $\mathbf{x}(t)$ be centered to have zero mean across all EEG channels and $\mathbf{R}_{xx} = \frac{1}{N_T}(\mathbf{X}\mathbf{X}^\top)$ be the estimated covariance matrix of the signals $\mathbf{x}(t)$ where $\mathbf{X}$ is a $N_e \times N_T$ matrix represents multi-channel EEG signals, with N_e EEG electrodes and N_T time samples.

One popular method to solve the PCA problem is to perform eigenvalue decomposition on the covariance matrix $\mathbf{R}_{\text{xx}}$ so that $\mathbf{R}_{\text{xx}}$ can be expressed in terms of its eigenvectors and eigenvalues as [71][72]:

$$\mathbf{R}_{\text{xx}} = \mathbf{W}_{\text{pca}}\boldsymbol{\lambda}\mathbf{W}_{\text{pca}}^\top \tag{3.9}$$

where $\mathbf{W}_{\text{pca}}$ is a matrix with each column representing a normalised eigenvector while $\boldsymbol{\lambda}$ is a diagonal matrix with its diagonal element representing the eigenvalue associated with its corresponding eigenvector in $\mathbf{W}_{\text{pca}}$. After the eigen-decomposition, the eigenvectors in $\mathbf{W}_{\text{pca}}$ is usually arranged according to their respective eigenvalues in descending order.

In the EEG signal analysis, the obtained $\mathbf{W}_{\text{pca}}$ is the set of spatial filters that decompose the multi-channel signals into different source signals $\mathbf{y}(t)$.

$$\mathbf{y}(t) = \hat{\mathbf{s}}_{\text{pca}}(t) = \mathbf{W}_{\text{pca}}^\top\mathbf{x}(t) \tag{3.10}$$

In PCA, the first principal component $y_1(t)$ is usually the signal that contributes the most variance towards the measured signals $\mathbf{x}(t)$ while the second component captures the next largest and so on. Ideally, assuming the ERP-of-interest contributes the most variance towards $\mathbf{x}(t)$, one can retain only the first principal component for subsequent analysis. The process of removing the unwanted components are also known as source rejection, or dimension reduction in the machine learning context.

Another application of the PCA is to whiten the signals so that the signals are decorrelated and normalised to be unit variance. The whitening transformation matrix is given as [71]:

$$\tilde{\mathbf{W}}_{\text{pca}} = (\mathbf{W}_{\text{pca}}\boldsymbol{\lambda}^{-\frac{1}{2}}) \tag{3.11}$$

In the literature, the performance of PCA and ICA are often compared [66][73]. In general, PCA attempts to use second-order statistics to decorrelate the measured signals in order to extract the source signals, while ICA make uses of the stronger condition that the source signals are statistical independent to extract them. For ICA, maximisation of statistical independence is usually performed by exploiting the higher-order statistics and other non-Gaussianity features of the source signals [59]. Thus, when compared to PCA that relies only on the second-order statistics, ICA can be a better method at capturing non-Gaussian signals (which is more likely in the case of EEG signals). In practice, the ability to capture non-Gaussian signals can also be partly the reason why ICA has been a more popular method in the EEG signal processing field [3][54][64][74]. However, it should be noted that since ICA involves higher-order statistics, the method also requires more amount of signals to work effectively [73].

3.2.5.5 Extraction based on maximum variance difference

In general, a spatial filter that enhances the ERP-of-interest in one condition, or class, but not another is called a *discriminative* spatial filter. It can be obtained either by maximising the difference in variance (that is power of the extracted signal) or the mean between the two experimental conditions. A few examples of techniques that maximise the difference in variance are Max SNR (Signal-to-Noise Ratio) beamformer, common spatial pattern (CSP) and xDAWN; while techniques that maximise the mean difference involve classification techniques such as logistic regression (LR) and linear discriminative analysis (LDA). In this section, a brief review will be presented on the max SNR beamformer, CSP and xDAWN to outline their similarities and differences.

1. **Max SNR beamformer**

   Let $\mathbf{R}_1$ and $\mathbf{R}_2$ represent the trial-averaged sample covariance matrices from EEG epochs of class 1 and 2 respectively. Mathematically, the goal of finding a spatial filter $\mathbf{w}$ that maximises the variance of class 1 and minimising the variance of class 2 is given as [75]:

   $$\mathbf{w}_{\text{SNR}} = \arg\max_{\mathbf{w}} \quad \frac{\mathbf{w}^\top \mathbf{R}_1 \mathbf{w}}{\mathbf{w}^\top \mathbf{R}_2 \mathbf{w}} \tag{3.12}$$

   Note that the objective function of the above optimization problem is a form of

Rayleigh quotient [75]. This allows (3.12) to be reformulated as a constrained optimization problem as follows:

$$\begin{aligned} \max_{\mathbf{w}} \quad & \mathbf{w}^\top \mathbf{R}_1 \mathbf{w} \\ \text{subject to} \quad & \mathbf{w}^\top \mathbf{R}_2 \mathbf{w} = 1 \end{aligned} \tag{3.13}$$

By solving the above constrained optimization problem using Langrange multiplier, the solution can be obtained as [76]:

$$\mathbf{R}_1 \mathbf{w} = \boldsymbol{\lambda} \mathbf{R}_2 \mathbf{w} \tag{3.14}$$

where the solution has the form of a generalized eigenvalue problem, in which, the eigenvector with the largest eigenvalue represents the spatial filter that maximises the variance of class 1 while minimises the variance of class 2.

In the case of a P300 experiment, the above discriminative method can be applied by treating the P300 trials as class 1 and the non-P300 trials as class 2. Since P300 is only observed in class 1, $\mathbf{R}_1$ and $\mathbf{R}_2$ can also be regarded as the signal and noise covariance matrix respectively. As a result, this technique is also called max SNR beamformer. This technique is also used in array signal processing [77] and was introduced to the P300-BCI application by [75].

2. **CSP**

   Other discriminative methods that share an optimization problem similar to (3.12) are CSP and xDAWN. CSP is originally developed by [78] for a study on the normal and abnormal EEG [79] before it was introduced to the BCI community by [80]. Currently, it is one of the popular signal extraction tools in BCI applications [76][81][82][83]. In general, CSP contains two computational stages. The first stage involves the whitening transformation of the composite covariance matrix $\mathbf{R}_s = \mathbf{R}_1 + \mathbf{R}_2$ while the second stage involves the generalized eigendecomposition of $\mathbf{R}_1$. The CSP spatial filters are then given as $\mathbf{W}_{csp} = \mathbf{BV}$ where $\mathbf{B}$ is the whitening matrix from the first stage and $\mathbf{V}$ is the eigenvectors matrix from the second stage. Mathematically, the optimization problem of CSP can be also expressed as a form of Rayleigh Quotient [81]:

$$\mathbf{w}_{csp} = \arg\max_{\mathbf{w}} \quad \frac{\mathbf{w}^\top \mathbf{R}_1 \mathbf{w}}{\mathbf{w}^\top \mathbf{R}_s \mathbf{w}} \tag{3.15}$$

except that $\mathbf{R}_2$ in (3.12) is now replaced by the composite covariance matrix $\mathbf{R}_s$. Based on this optimization problem, CSP can be considered as a form of spatial filter that maximises the "signal-to-signal plus noise ratio" (SSNR). Similarly, it can also be solved as a generalized eigenvalue problem.

3. **xDAWN**

   With the same motivation as CSP, xDAWN[5] is another popular discriminative spatial filter which attempts to maximise the SSNR between the signals. It is proposed by [84] specifically for ERP applications. By taking the stimulus presentation timing into consideration, the author suggest a new representation model for the EEG signals and a new way of pre-filtering the ERP signals so that the covariance matrix $\mathbf{R}_1$ is estimated based on a cleaner ERP signal. The optimization problem for xDAWN is given as:

$$\mathbf{w}_{\text{xDAWN}} = \arg\max_{\mathbf{w}} \quad \frac{\mathbf{w}^\top(\mathbf{A}^\top\mathbf{D}^\top\mathbf{D}\mathbf{A})\mathbf{w}}{\mathbf{w}^\top(\mathbf{X}\mathbf{X}^\top)\mathbf{w}} \tag{3.16}$$

   where $\mathbf{X}$ is the multi-channel EEG signals, with N_e EEG channels and N_T time samples over the entire EEG recording. Let N_l define the number of time samples in each ERP trial, $\mathbf{D}$ is a $N_T \times N_l$ Toeplitz matrix whose elements of the first column is defined such that $\mathbf{D}(t_k, 1) = 1$, where t_k is the stimulus onset of the k-th target stimulus. $\mathbf{A}$ is a $N_l \times N_e$ dimensional matrix that represents a matrix of ERP signals and is estimated as:

$$\mathbf{A} = (\mathbf{D}^\top\mathbf{D})^{-1}(\mathbf{X}\mathbf{D})^\top. \tag{3.17}$$

   It is worth noting that in cases where the interval between t_k and t_{k+1} is very large, $(\mathbf{D}^\top\mathbf{D})^{-1}$ will be reduced to a diagonal matrix. In such situations, the estimated $\mathbf{A}$ can be observed as the traditional multi-channel trial-averaged signals. On the other hand, it is also proposed in [84] that the above optimization problem can be solved through QR decomposition. More details on xDAWN can be found in [84].

In practice, there can be situations where the spatial filter that provides the maximum variance difference as in max SNR beamformer (3.12), CSP (3.15) and xDAWN (3.16) do not extract the source signal that corresponds to the ERP-of-interest [53]. To

[5] xDAWN is actually named after a signal model $\mathbf{X} = \mathbf{D}\mathbf{A}\mathbf{W}^\top + \mathbf{N}$ [84].

address this problem, generalized eigenvalue decomposition is usually performed to determine the set of spatial filters **W** simultaneously. To recover the ERP-of-interest, it has been suggested to also consider the first few spatial filters that provide the largest variance differences [80].

In the literature, another issue that is less discussed is the estimation of the signal covariance matrix $\mathbf{R}_1$. Let $\mathbf{R}_{1,k}$ represent the signal covariance matrix estimated from k-th epoch in class 1. In practice, $\mathbf{R}_1$ is obtained by averaging every $\mathbf{R}_{1,k}$ in class 1 [80], while each $\mathbf{R}_{1,k}$ is estimated from the entire EEG segment of the epoch (e.g. 0–1000 ms after stimulus onset) [75][82][83][84]. For ERP application, the approach of using a long EEG segment to compute $\mathbf{R}_{1,k}$ remains questionable. This is because unlike other signals (e.g. sensorimotor rhythm) used in BCI applications, ERP-of-interest (e.g. P300) normally appears only within a specific time range (e.g. 300–600 ms) of the epoch segment. Thus, when estimation is based on a longer EEG segment, the signal covariance matrix $\mathbf{R}_1$ is corrupted easily by noises. This problem is demonstrated in Chapter 6 and we also demonstrated how proper time selection may improve the performance of xDAWN.

3.2.5.6 Extraction based on maximum mean difference

Suppose that the ERP-of-interest only occurs within a time interval [t_1, t_2] and has different peak amplitudes under two different task conditions. Let $\mathbf{X}_{1,k}$ and $\mathbf{X}_{2,k}$ denote k-th EEG epoch under condition 1 and 2 respectively where $\mathbf{X}_{1,k}$ and $\mathbf{X}_{2,k}$ both contain a multi-channel EEG signals, with N_e channels and N_t time samples. The mean amplitude of ERP-of-interest across the time interval [t_1, t_2] under class 1 and class 2 can be described respectively as:

$$\bar{\mathbf{x}}_1 = \frac{1}{N_k} \sum_k \sum_{t=t_1}^{t_2} \mathbf{x}_{1,\mathrm{k}}(t) \tag{3.18}$$

$$\bar{\mathbf{x}}_2 = \frac{1}{N_k} \sum_k \sum_{t=t_1}^{t_2} \mathbf{x}_{2,\mathrm{k}}(t) \tag{3.19}$$

where N_k is the number of epochs, $\mathbf{x}_{1,k}(t)$ and $\mathbf{x}_{2,k}(t)$ represent multi-channel measurement vector at time instant t of k-th epochs under condition 1 and 2 respectively.

To capture the major signal that contributes to the difference, a discriminative spatial filter can be determined by maximising the mean difference between the two conditions [53]. There are a variety of ways to achieve such a goal. Assuming the signals across the time interval [t_1, t_2] for both condition 1 and 2 [i.e. $\mathbf{x}_{1,\mathrm{k}}(t)$ and $\mathbf{x}_{2,\mathrm{k}}(t)$] are Gaussian distributed, have identical spatial covariance matrices $\mathbf{R}$ but differing in their means, a discriminative spatial filter can be given as [53]:

$$\mathbf{w}_{\mathrm{LD}} = \mathbf{R}^{-1}\Delta\mathbf{x} \tag{3.20}$$

where $\Delta\mathbf{x} = \bar{\mathbf{x}}_1 - \bar{\mathbf{x}}_2$. To achieve maximum mean difference between two conditions, the above spatial filter is determined by considering the direction of $\Delta\mathbf{x}$ and the correlated activities in the electrodes. This spatial filter can be identified as the projection vector of the classifier called linear discriminant analysis (LDA) [85].

Equation (3.20) shows that the goal of maximising the separation between the mean difference is generally similar to the goal of training a classifier (which will be discussed in Section 3.3.2). Thus, different discriminative projection vectors of linear classifiers such as logistic regression [86][87][88], LDA [89][90][91] and their variants [75][92] are often employed as the spatial filter for ERP extraction. Often, these spatial filters are called Linear discrimination (LD) [86][88] to avoid confusion with the classifiers used in the ERP classification.

3.3 Pattern recognition for ERP classification and segmentation

3.3.1 Overview

Processing and analysing a large amount of EEG signals can be challenging and sometimes time-consuming especially when the analysis is performed using multiple channels at the single-trial level. For this reason, there is an increasing demand for pattern recognition algorithms to assist in finding and labeling meaningful structures in multi-channel EEG recordings. In general, pattern recognition is a discipline in which one of the goals is to develop data-driven algorithms to assign a group of objects into a number of classes [85][93][94]. Classification is the supervised learning process of assigning a new object to a class based on a given set of labeled objects (training data). In contrast, the unsupervised learning process of grouping the objects into classes without the training data is usually known as clustering.

Both classification and clustering techniques are important for ERP studies. In recent years, classifiers are mostly employed in BCI applications. They are usually trained to learn and detect a specific EEG signal so that the detection outcome can be used as a feedback for control and communication with the BCI [95][96][97][98]. In some cases, these classifiers are also adapted to determine the spatial filter used in ERP extraction as mentioned in Section 3.2.5.6. In contrast, clustering techniques are mostly used for grouping EEG signals with either similar spatial, temporal or frequency characteristic into different segments. As such, they can reduce the rich EEG information into its compact form, thus further simplifying the analysis of EEG. In the following sections, different types of classifier and clustering techniques will be described briefly in the context of ERP classification and segmentation respectively.

3.3.2 ERP Classification

For ERPs, most classification methods have been applied mainly to BCI applications such as P300-BCI as mentioned in Chapter 2. Most P300-BCI systems treat the ERP detection problem as a binary classification problem where the goal is to assign EEG epochs that contain a P300 ERP to a target class and vice versa.

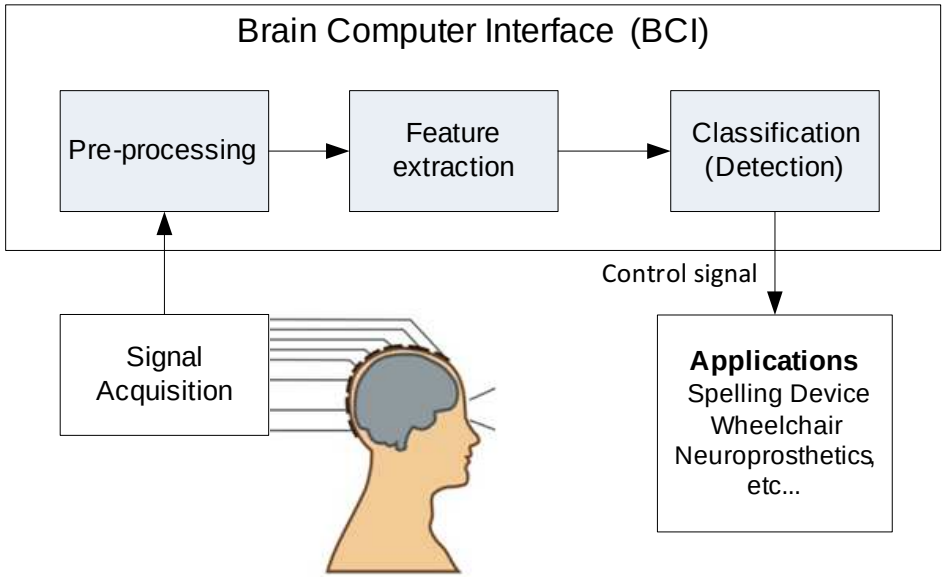

Figure 3.8: The processes involved in a BCI system

ERP classification is a very challenging problem due to the high inter-trial variability and the low SNR of the ERPs. Before ERP classification can be carried out, the overall BCI system usually involves three main processing stages, that is, preprocessing, feature extraction, and classification as shown in Fig. 3.8. In the preprocessing stage, techniques such as bandpass filtering and artifact removal are usually performed before the recorded EEG signals are divided into epochs. After that, each EEG epoch is processed to extract a set of features that best describe the ERP-of-interest. In ERP classification, the most common approach is to directly use every EEG measurement within an epoch as the features [97][98]. However, by doing so, a very large amount of features is usually obtained. For this reason, to reduce the number of features in each epoch, it is also popular to downsample the EEG signals after bandpass-filtering [97][98]. Alternatively, other approach to reduce the number of features is to apply spatial filter(s) to extract either one or a number of signals that are usually less than the number of EEG channels in an epoch [75][83][82][84][96]. After extracting the features, all features are then concatenated to form an one-dimensional feature vector for classification. However, before that, a classifier has to be constructed by training it with a set of labeled training samples (or feature vectors).

In P300-BCI, different types of classifiers have been introduced. Among these classifiers, LDA and support vector machine (SVM) are the two major types of classifiers [95][97][98]. In this section, only LDA and its regularized variant are discussed.

For more information about other popular classifiers (i.e. Bayesian LDA and SVM), see Appendix A.

3.3.2.1 Linear discriminant analysis

Linear discriminant analysis (LDA) is a method that projects a high-dimensional feature vector onto lower-dimensional space so that the projected data can be easily separated. The optimization problem that defines the LDA is formulated as follows [85][93][99]:

$$\max_{\mathbf{u}} \quad \frac{\mathbf{u}^\top \mathbf{S}_b \mathbf{u}}{\mathbf{u}^\top \mathbf{S}_w \mathbf{u}} \tag{3.21}$$

where the goal is to find a projection vector $\mathbf{u}$ that maximise the ratio of the between-class variance $\mathbf{S}_b$ to the within-class variance $\mathbf{S}_w$.

Suppose there are two classes (i.e. class 1 and class 2) where both classes have equal number of training samples N. Additionally, suppose class 1 is the positive class while class 2 is the negative class. The $\mathbf{S}_b$ and $\mathbf{S}_w$ matrices can be calculated as:

$$\mathbf{S}_b = (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)^\top \tag{3.22}$$

$$\mathbf{S}_w = \mathbf{S}_1 + \mathbf{S}_2 \tag{3.23}$$

where $\boldsymbol{\mu}_1$, $\boldsymbol{\mu}_2$, $\mathbf{S}_1$ and $\mathbf{S}_2$ are defined as follows. For each class c, c=1,2, the sample mean $\boldsymbol{\mu}_c$ and covariance matrix $\mathbf{S}_c$ are computed as:

$$\mathbf{S}_c = \sum_{i=1}^{N} (\mathbf{v}_{c,i} - \boldsymbol{\mu}_c)(\mathbf{v}_{c,i} - \boldsymbol{\mu}_c)^\top \qquad c = 1, 2 \tag{3.24}$$

$$\boldsymbol{\mu}_c = \frac{1}{N} \sum_{i=1}^{N} \mathbf{v}_{c,i} \qquad c = 1, 2 \tag{3.25}$$

where $\mathbf{v}_{c,i}$ denotes the i-th feature vector in class c.

Since the optimization problem in (3.21) is expressed as a form of Rayleigh quotient, its solution has the form of a generalized eigenvalue problem [85]:

$$\mathbf{S}_b \mathbf{u} = \lambda \mathbf{S}_w \mathbf{u} \tag{3.26}$$

Note that the rank of $\mathbf{S}_b$ is at most one since it is the outer product of the same vectors. Assuming $\mathbf{S}_w$ is invertible, Eq. (3.26) can be rewritten as:

$$\mathbf{S}_w^{-1} \mathbf{S}_b \mathbf{u} = \lambda \mathbf{u} \tag{3.27}$$

$$\mathbf{S}_w^{-1}[(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)^\top]\mathbf{u} = \lambda\mathbf{u} \tag{3.28}$$

By substituting (3.22) into (3.27) and replacing $(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)^\top\mathbf{u}$ in (3.28) with α, a closed form solution for LDA can be given as:

$$\mathbf{u} = \frac{\alpha}{\lambda}\mathbf{S}_w^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2) \tag{3.29}$$

where for any scaling factor $\frac{\alpha}{\lambda}$, $\mathbf{u}$ is always in the direction of $(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)$.

In practice, the scaling factor is of less concerned as long as $\mathbf{u}^\top\mathbf{v}_{1,i} > \mathbf{u}^\top\mathbf{v}_{2,i}$. Thus, for convenience, $\frac{\alpha}{\lambda}$ is usually maintained as 1 and the popular form of the solution for LDA is given as [85]:

$$\mathbf{u} = \mathbf{S}_w^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2) \tag{3.30}$$

To assign a test sample $\tilde{\mathbf{v}}$ to its respective class, the classification procedure is often performed as follows. Firstly, by applying $\mathbf{u}$ to $\tilde{\mathbf{v}}$, the projected value (or the classifier score) z is computed as:

$$z = \mathbf{u}^\top\tilde{\mathbf{v}} - b \tag{3.31}$$

where the classifier threshold or bias term b is usually given as the average of the projected mean from both classes.

$$b = \frac{1}{2}[\mathbf{u}^\top(\boldsymbol{\mu}_1 + \boldsymbol{\mu}_2)] \tag{3.32}$$

Secondly, the classification rule is applied as:

$$q = \text{sgn}(z) = \text{sgn}(\mathbf{u}^\top\tilde{\mathbf{v}} - b) \tag{3.33}$$

where $\text{sgn}(.)$ is a signum function and the class label is given as $q \in \{+1, -1\}$. If $q = +1$, $\tilde{\mathbf{v}}$ is assigned to class 1. Likewise, if $q = -1$, $\tilde{\mathbf{v}}$ is assigned to class 2.

From the Bayes decision theory point of view, LDA has an implicit assumption that the feature vectors of both classes are Gaussian distributed and share the same covariance matrix. Although this assumption may not be necessarily true, massive studies have shown that LDA produces significantly good results in most BCI applications [95][97]. Besides that, it is also popular because it is easy to implement and can be computed fast enough for online classification. As a result, many algorithms have been developed based on LDA such as regularized LDA [76], stepwise LDA [97] and Bayesian LDA [101].

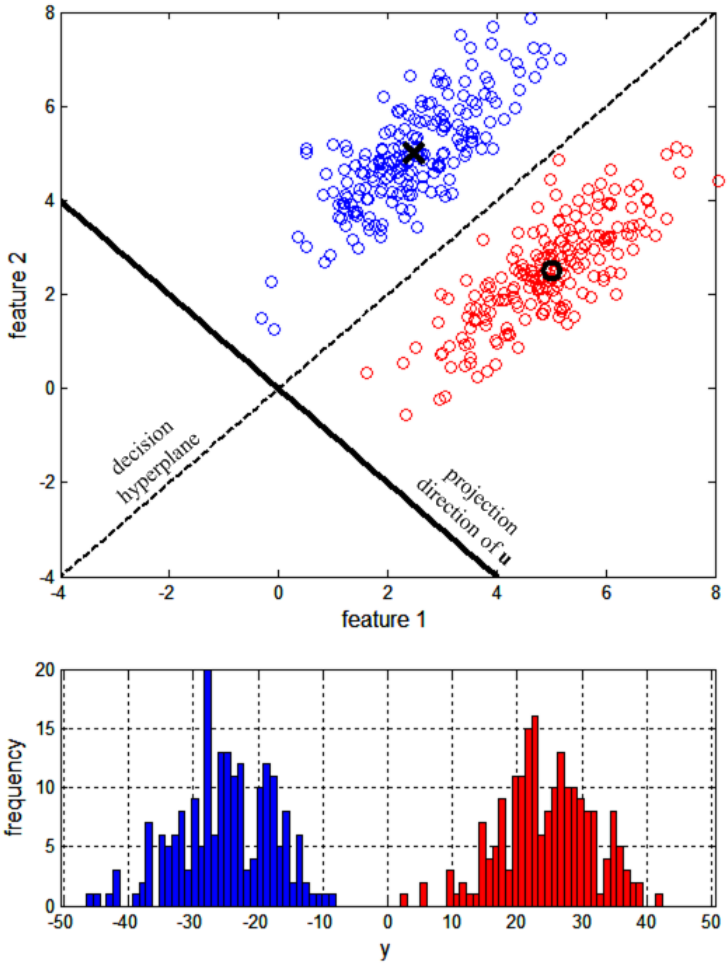

Figure 3.9: Top: Scatterplots of two-dimensional data from two different classes, Bottom: LDA projection on the given data using equation (3.31).

3.3.2.2 Regularized linear discriminant analysis

In BCI application, when the number of available training samples are less than the number of features, over-fitting problem may occur as there is insufficient amount of samples for estimating the mean and covariance matrix for each class. As a result, the covariance matrix $\mathbf{S}_w$ may be singular and cannot be inverted. To avoid the over-fitting issue, a regularization parameter $\lambda \in [0, 1]$ can be added into $\mathbf{S}_w$ in order to promote the generalization and robustness of LDA against outliers [76][100]:

$$\mathbf{S}_w = (1 - \lambda)\mathbf{S}_w + \lambda\mathbf{I} \tag{3.34}$$

where $\mathbf{I}$ is the identity matrix. This variant of LDA is also known as regularized LDA (RLDA). Choosing an suitable λ is an important issue in RLDA. This is because a large λ may significantly change the information in $\mathbf{S}_w$, while a small λ may not be

effective enough to resolve the singularity problem in $\mathbf{S}_w$. For this reason, in practice, the optimal λ is usually obtained through cross-validation by testing a finite range of λ.

3.3.3 ERP segmentation

Let $\mathbf{X} = [\mathbf{x}(t_1), \mathbf{x}(t_2), ..., \mathbf{x}(t_{N_t})]$ be a $N_e \times N_t$ matrix representing the multi-channel EEG recordings. By mapping every channel measurement to their relative spatial locations on the scalp, each $\mathbf{x}(t)$ can be observed as a snapshot of the scalp distribution at different time instant. Subsequently, when analysing $\mathbf{x}(t)$ across time, an evolution of the scalp distributions can be observed. analysing the trial-averaged ERP signals in this way has led to an important discovery that each ERP tends to exhibit a unique and stable scalp distribution over a specific time duration. In the literature, this meta-stable time region is sometimes known as microstates [102]. This observation has led to interesting questions such as the starting time of each specific ERP and the duration of their scalp distribution in the EEG signals.

Identifying the time region of the ERP-of-interest is essential for many ERP signal processing techniques since this time information is often required for designing the ERP template [57][58], constraining the ERP estimation [49][50][57], computing the signal statistics [53][87][88][89], and finding the best EEG segment for ERP detection [97]. In practice, to determine the ERP time region manually can be laborious and time consuming when there is a large amount of scalp distributions to be analysed. Apart from that, the process of manual segmentation can be subjective since ERPs usually vary across subjects, trials and experimental conditions. Thus, a solution to the problem of identifying the ERP time region is to perform ERP segmentation with the pattern recognition techniques. In the literature, the two ERP segmentation techniques commonly used in ERP analyses are: (i) the combination of Global Field Power (GFP) and Global Map Dissimilarity (GMD) [103][104] and (ii) their extension based on the modified K-means clustering algorithm (mKM) [105].

3.3.3.1 Global Field Power and Global Map Dissimilarity

Global Field Power (GFP) is a method originally proposed by [102] to facilitate ERP analysis by transforming the multi-channel EEG signals into one-dimensional measurements that reflects the activity of each individual ERP. Mathematically, GFP attempts to measure the standard deviation of the scalp distribution at each time instant given as:

$$GFP(t) = \sqrt{\frac{1}{N_e}\sum_{i=1}^{N_e}[x_i(t) - \bar{x}(t)]^2} \tag{3.35}$$

where $x_i(t)$ represents the measurement for i-th channel while $\bar{x}(t) = \frac{1}{N_e}\sum_{i=1}^{N_e} x_i(t)$ is simply the mean of multi-channel measurements at time t.

Global Map Dissimilarity (GMD) is a method that attempts to measure the difference between two scalp distributions [103]. When applied together with GFP, the method is usually used to measure the difference between two successive scalp distributions in time:

$$GMD(t) = \sqrt{\frac{1}{N_e}\sum_{i=1}^{N_e}[x_i^*(t) - x_i^*(t-1)]^2} \tag{3.36}$$

where x_i^* is the i-th channel measurements from a scalp distribution that are normalised to have zero-mean and unit variance. Technically, GMD can be considered as a first-order derivative edge-detection method whose goal is to identify the transition time between two different scalp distributions based on the measured edge strength [106]. In ERP analysis, GMD is usually inversely correlated with GFP. The time region of an ERP is usually selected to be the region where two successive peaks of GMD are aligned to the troughs of GFP.

3.3.3.2 Modified K-Means Clustering

Segmentation based on the GFP and GMD can be bias and subjective since it still requires human supervision. To automate the segmentation process, a clustering-based segmentation algorithm is therefore proposed by [105]. In the literature, the technique proposed by [105] is often referred as the K-means (KM) clustering algorithm although it has a different structure to the standard KM algorithm in terms of the distance measure between samples and the way of assigning the samples into representative clusters. To avoid confusion, this technique will be referred as modified K-means clustering

(mKM) in this thesis.

The original KM algorithm is a clustering technique that attempts to group the objects by assessing their squared euclidean distances [93][107]. The clustering process normally begins by selecting K number of points as initial cluster centroids. After that, each sample is assigned to the nearest cluster based on their distances with the centroids. When all samples have been assigned, the positions of the centroids are re-estimated based on their members. The process of assigning each sample and re-estimating the centroids is repeated until there are no changes in the positions of the centroids. mKM algorithm shares similar structure with KM except that the samples [$\mathbf{x}(t)$ in this thesis] are assigned based on spatial correlation. In addition, instead of computing the mean across the members, mKM performs PCA on each cluster so that the resultant largest principal component can be taken as the re-estimated centroid.

Both KM and mKM shares similar inherent problems during the segmentation process. The main problem is to define the number of clusters for representing the ERPs [93]. Since the best number of cluster is usually unknown, the algorithm has to be repeated by testing different number of clusters. The second problem is initialization of the cluster centroids [93]. In practice, the centroids are often initialized by randomly selecting a number of $\mathbf{x}(t)$ from the given signals. Since different starting points will yield different results, the algorithm has to be repeated for multiple times to ensure the consistency of the result. The third problem is that the clustering algorithms usually treat the segmentation problem globally by ignoring the time information in each individual samples. Consequently, the final segmentation result may contain a number of spurious short and meaningless segments [104]. To deal with this issue, a further post-processing on the segmentation result is often necessary to remove these short and meaningless time segments [104][105].

3.4 Summary

In this chapter, the multi-channel ERP extraction problem is reviewed. In addition, different signal processing and pattern recognition techniques used in various ERP fields are also described. Currently, the single-trial multi-channel ERP extraction problem has been addressed differently depending on the assumptions posed on the ERPs. Each

method has its own merits and drawbacks. In general, these methods can be divided into data-independent and data-driven methods. Data-independent methods are usually designed specifically for ERP extraction. However, to apply these methods successfully, a strong background knowledge on the ERP waveform and the mixing matrix is required. In contrast, data-driven methods do not require any such assumptions, and thus the extracted signal is more adaptive towards the given ERP dataset. However, most existing data-driven methods are inefficient since they cannot extract the ERP directly. Therefore, the development of a new data-driven method that is dedicated for ERP extraction is necessary to facilitate the studies of ERPs. Designing such extraction method is difficult without combining techniques from other ERP fields. Thus, in Section 3.3.2 and 3.3.3, the problem and techniques related to ERP classification and segmentation problem are also discussed. More details on how these techniques can be utilised for ERP extraction will be discussed in detail in the following chapters.

Chapter 4

A New Method to Extract ERP through Discrimination Between ERP and non-ERP Time Regions

4.1 Introduction

The ability to examine the dynamics of the ERPs at single-trial level is one of the ultimate goals in the ERP signal processing field. The time course of the ERP often provides richer and valuable picture of the neural activity, their activation time and also duration across trials. Moreover, better understanding on the trial-to-trial characteristics of the ERP also helps us to overcome certain limitation in the conventional trial-averaging method and further provides insight on the factor that leads to peak amplitude and latency difference as observed at trial-averaged signals. As mentioned in Chapter 3, the single-trial ERP analysis has been addressed in several significantly different ways. Since the modern EEG recordings are usually flooded with huge collection of data captured from different electrodes on the scalp, it is increasingly important to utilise the available spatial information to provide a better and more reliable extraction on the ERP-of-interest.

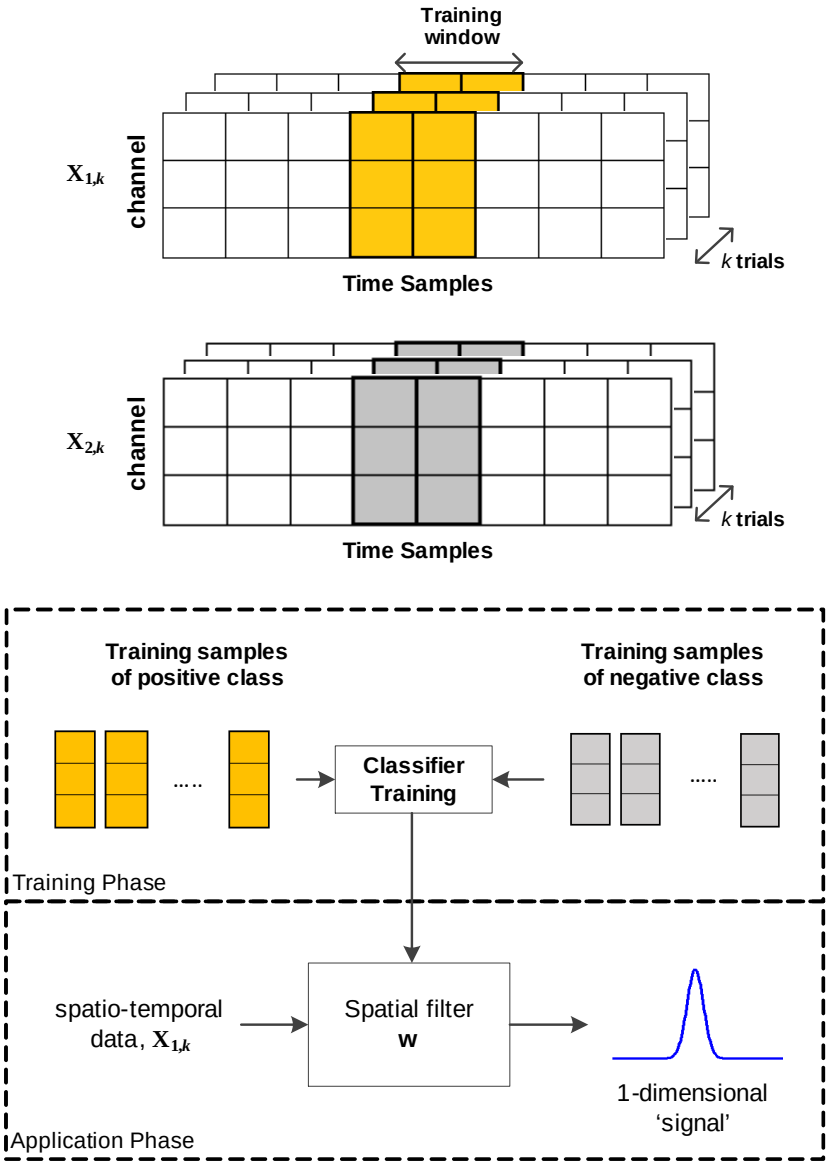

Figure 4.1: The classifier training process that is normally used in the conventional LD to obtain the spatial filter $\mathbf{w}$ that extracts the ERP-of-interest, which causes the mean differences between two experimental conditions. Let $\mathbf{X}_{1,k}$ and $\mathbf{X}_{2,k}$ be the EEG epochs taken from the 'ERP' and 'non-ERP' trials respectively. During training, for each condition, the multi-channel measurement vectors from the window-of-interest is taken as an individual training sample for each class respectively.

In the literature, different methods have been proposed to exploit the specific feature of the ERP for a successful signal extraction. Recently, besides the commonly used methods such as PCA and ICA, LD method is also getting increasingly popular to facilitate the ERP analysis. In general, LD is a method that tries to extract a signal that contributes the maximum mean difference between two experimental conditions.

In [53], it is demonstrated that one of the solutions of finding such spatial filter is generally the same as finding the discriminative projection vector of the binary linear classifier (See Chapter 3.2.5.6). Fig. 4.1 shows the classifier training that is commonly used to obtain the LD spatial filter. From the figure, it can be easily observed that since only multi-channel measurements are involved, the conventional LD is also a method that attempts to train a classifier to learn the scalp distribution difference between the two conditions. Depend on the class labels, the desired ERP are usually projected to the positive side of the 'extracted' signal and vice versa.

In practice, the application of LD has shown promising results on the P300 ERP studies. For example, it has been used to extract P300 ERP for analysing the possible relationship between P300 and the stimulus response time [87], for measuring the video and speech quality perception [89][91] and as the feature extraction tool for P300-BCI [75][92]. In these cases, such extraction is generally possible because LD takes advantages of the fact that there exists a time window which P300 ERP causes large activity difference between P300 and non-P300 trials. Unfortunately, if the ERP-of-interest appeared in both experimental conditions and has minimal activity difference between them, the conventional LD may no longer work effectively for extracting the ERP-of-interest. This kind of situation is most likely encountered when studying the exogenous ERPs such as P100 and N170.

To address the above issues, a more generic type of LD method called ENE-LD (ERP-versus-Non-ERP time region-based Linear Discrimination) is presented in this work. In the literature, studies have shown that each ERP tends to exhibit a unique scalp distribution that spans across a specific time region [5][7][6][12][21]. Thus, assume that the time region of the ERP-of-interest can be roughly estimated, we propose that a classifier can be trained to learn the relevant and irrelevant scalp distributions simply from the ERP and non-ERP time regions within the same EEG segment. By doing so, it also avoids the restrictive requirement that the ERP-of-interest must have large difference in the window-of-interest of the two different conditions. The proposed way of obtaining the spatial filter for the ENE-LD is illustrated in Fig 4.2.

The rest of this chapter is organized as follows. First, the signals extracted from ENE-LD are potentially useful for estimating the ERP parameters such as peak latency, amplitude and scalp distribution. For this reason, in this work, we also evaluate the

proposed ENE-LD on one of the existing estimation framework proposed by Li *et al.* [57] whereby the original template-matching based technique used in the framework is replaced by the extracted signals from our ENE-LD to represent the desired ERP waveform. The procedure of using the proposed ENE-LD method for ERP extraction and estimation is described in Section 4.2. In Section 4.3, a simulation study is conducted to assess the overall strength and weakness of the proposed ENE-LD from the extraction and estimation perspective whereby its performance is also compared to Li's template-matching based estimation method. After that, the performance of the proposed method is examined on a real N170 ERP dataset in Section 4.4 whereby the proposed ENE-LD is compared to the conventional LD before the conclusions are drawn in Section 4.5.

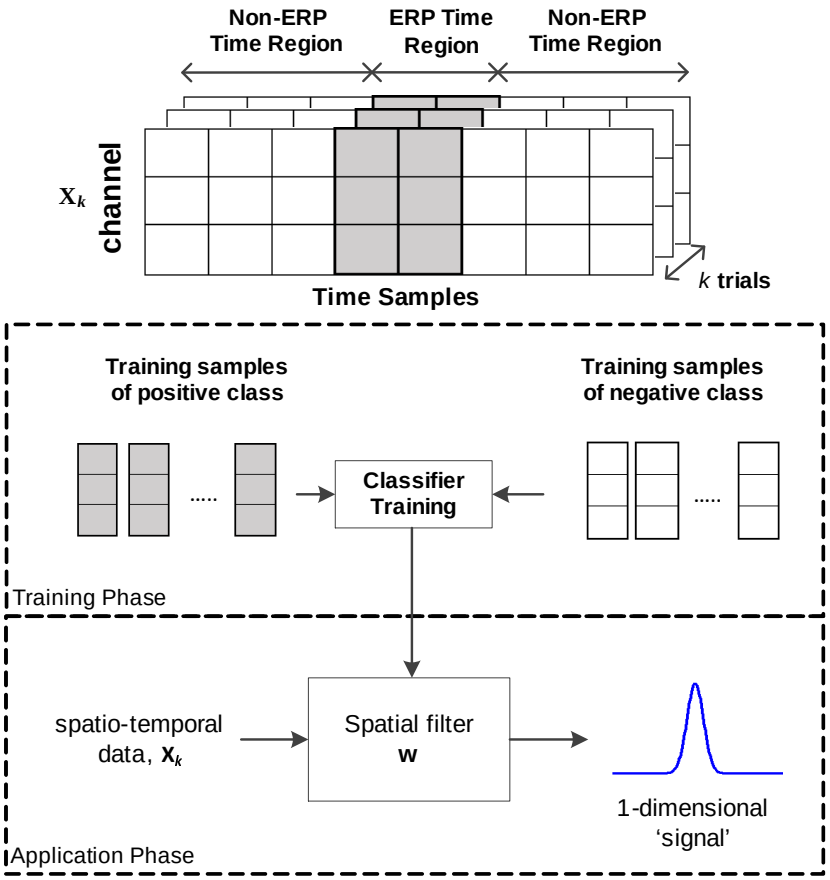

Figure 4.2: The proposed ENE-LD implementation for extracting the ERP-of-interest activity from the irrelevant ERP activity.

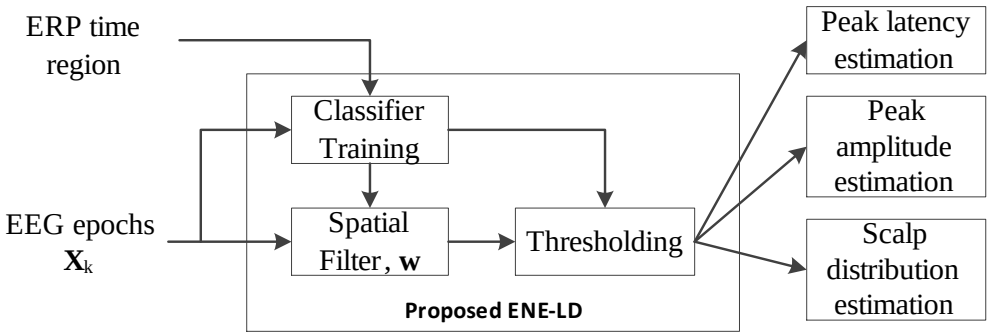

Figure 4.3: The overall process of using the proposed ENE-LD method for extraction and estimation.

4.2 Methodology

4.2.1 Overview of ENE-LD extraction method

Fig. 4.3 shows the overall process that is used in Section 4.3 and 4.4 to evaluate our proposed ERP extraction method. For the proposed ENE-LD method, the process of obtaining the spatial filter is described in Fig. 4.2. During the training phase, the time regions that contain the scalp distribution of the ERP-of-interest are assigned to the positive class, while the remaining time regions are assigned to the negative class. After that, a binary linear classifier is constructed based on these training samples while its discriminative projection vector is applied as a spatial filter to separate the ERP and non-ERP time region in an EEG epoch. Due to the nature of the classifier, since the scalp distribution of the ERP-of-interest are given the positive class label, the ERP-of-interest is usually projected to the positive side of the extracted signal and vice versa. Thus, thresholding is applied to remove the negative part of the extracted signal before it is used for the ERP parameter estimation.

4.2.2 Spatial filter and peak latency estimation

4.2.2.1 ENE-LD spatial filter through LDA classifier

In the literature, the two popular classifiers that are commonly used in the conventional LD are logistic regression and LDA. In this work, LDA is chosen for the proposed ENE-LD module because of its closed-form solution. For the proposed ENE-LD, the procedure to obtain the spatial filter from the LDA classifier is summarised as follows. Suppose the EEG equipment has N_e measurement channels, and the measurements

captured at time index t during the k-th trial are stored in the $N_e \times 1$ vector $\mathbf{x}_k(t)$. Let N_T be the number of snapshots taken in each EEG epochs (or trial), N_k be the total number of trials, and the sets $\mathcal{T}_1$ and $\mathcal{T}_2$ contain, respectively, the time indices of the positive class and those of the negative class. Let T_1 and T_2 be the number of elements in $\mathcal{T}_1$ and $\mathcal{T}_2$, respectively. To enable the classifier to learn the scalp distribution of the ERP-of-interest, we first compute the following two $N_e \times 1$ vectors which represent the mean scalp distribution of the positive and negative classes:

$$\boldsymbol{\mu}_c = \frac{1}{T_c N_k} \sum_{t \in \mathcal{T}_c} \sum_{k=1}^{N_k} \mathbf{x}_k(t) \qquad , c = 1, 2 \tag{4.1}$$

and the $N_e \times N_e$ covariance matrix for both classes as:

$$\mathbf{S}_c = \frac{1}{T_c N_k} \sum_{t \in \mathcal{T}_c} \sum_{k=1}^{N_k} (\mathbf{x}_k(t) - \boldsymbol{\mu}_c)(\mathbf{x}_k(t) - \boldsymbol{\mu}_c)^\top \qquad , c = 1, 2 \tag{4.2}$$

To maximise the separability of the positive and negative classes, the closed-form solution of the spatial filter as given by LDA is computed as:

$$\mathbf{w}_{\text{eld}} = \mathbf{S}_w^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2) \tag{4.3}$$

where by accounting the prior class probability, the within-class scatter matrix $\mathbf{S}_w$ is calculated as:

$$\mathbf{S}_w = \frac{1}{N_T}(T_1 \mathbf{S}_1 + T_2 \mathbf{S}_2) \tag{4.4}$$

In practice, to avoid $\mathbf{S}_w$ becomes singular, $\mathbf{S}_w$ can be regularized using the method as mentioned in Chapter 3.3.2.2.

4.2.2.2 Estimating the ERP peak latency

To determine whether a time region is correlated to the ERP-of-interest, the spatial filter $\mathbf{w}_{\text{eld}}$ is applied to an EEG epoch and the classifier output $y_k(t)$ is calculated as follow:

$$y_k(t) = \mathbf{w}_{\text{eld}}^T \mathbf{x}_k(t) - b \tag{4.5}$$

where $b = \mathbf{w}_{\text{eld}}^T(\boldsymbol{\mu}_1 + \boldsymbol{\mu}_2)/2$ is the threshold of the classifier. From the classifier point of view, $y_k(t)$ describes the possibility of a ERP-of-interest occurrence with positive

sign indicating the ERP-of-interest is present and vice versa. Accordingly, the latency of the ERP-of-interest can be taken to be the time when $y_k(t)$ is largest:

$$\hat{\tau}_k = \arg\max_t \{y_k(t)\} \tag{4.6}$$

Since only the positive part of signal $y_k(t)$ belongs to the desired ERP activity, the negative part of the signal can be eliminated:

$$\tilde{y}_k(t) = \max(y_k(t), 0) \tag{4.7}$$

Lastly, by normalising $y_k(t)$ to have unit variance, the resultant normalised signal $\tilde{y}_k(t)$ can be used to represent the desired ERP source signal for estimation in the following section. The advantage of using this signal in Li's estimation framework method is that ENE-LD does not involve the modeling of the ERP waveform. Thus, unlike the template-matching technique used in the original estimation method, the proposed method also avoid making any strict assumption that the ERP waveforms must be invariant across trials.

4.2.3 Scalp distribution and amplitude estimation

4.2.3.1 The EEG model for estimation

For scalp distribution and amplitude estimation, the method described by [57] is adopted. In [57], the estimation method is developed based on the following motivation. First, the linear EEG model for each EEG epoch can be rewritten as [57]:

$$\mathbf{x}_k(t) = \mathbf{a}_k s_k(t) + \sum_{i=1}^{N} \mathbf{b}_{k,i} n_{k,i}(t) \tag{4.8}$$

where $s_k(t)$ and $n_{k,i}(t)$ represents the desired and irrelevant source signals at k-th EEG trial respectively. $\mathbf{a}_k$ and $\mathbf{b}_{k,i}$ are the mixing vectors which explain the contribution of the corresponding source signal towards each electrode on the scalp. Alternatively, they convey the scalp distribution of the corresponding source signal.

Now, suppose that each desired ERP source signal are from the same neural generators in the brain and passing through the same mixing medium, the linear EEG model can be rewritten again as:

$$\mathbf{x}_k(t) = \sigma_k \mathbf{a}_0 \tilde{s}_k(t) + \sum_{i=1}^{N} \mathbf{b}_{k,i} n_{k,i}(t) \tag{4.9}$$

where $\mathbf{a}_0$ and $\tilde{s}_k(t)$ are the normalised common mixing vector and desired source signal in each EEG trial respectively while the dimensionless scaling factor σ_k can be interpreted as the amplitude of the source signal.

4.2.3.2 Estimating the common scalp distribution $\mathbf{a}_0$

Assume that the desired ERP source signal is uncorrelated with other ERP source signals $n_i(t)$ and it is estimated as $\tilde{y}_k(t)$ where $\tilde{y}_k(t)$ is normalised to have unit-variance, the scalp distribution of the desired ERP source signal in each trial can be estimated through minimum mean-square error (MMSE) criterion as:

$$\min_{\mathbf{a}_k} \quad \mathbb{E}\{\|\mathbf{x}_k(t) - \mathbf{a}_k\tilde{y}_k(t)\|^2\} \tag{4.10}$$

where the optimal solution for $\mathbf{a}_k$ is given as:

$$\hat{\mathbf{a}}_k = \frac{\mathbb{E}\{\mathbf{x}_k(t)\tilde{y}_k(t)\}}{\mathbb{E}\{\tilde{y}_k^2(t)\}} \tag{4.11}$$

where $\mathbb{E} = \frac{1}{N_T}\sum_t\{.\}$ is the sample mean.

Based on equations (4.8) and (4.9), and by absorbing the scaling factor σ_k for the desired ERP source signal into $\mathbf{a}_k$, the relationship between $\mathbf{a}_k$ and $\mathbf{a}_0$ is given as:

$$\mathbf{a}_k = \sigma_k\mathbf{a}_0 \tag{4.12}$$

To provide a robust estimation of $\mathbf{a}_0$, the estimation can be performed by first normalising $\mathbf{a}_k$ to drop the term σ_k and then averaging these normalised mixing vectors across trials [57]:

$$\hat{\mathbf{a}}_0 = \frac{1}{N_k}\sum_{i=1}^{N_k}\frac{\hat{\mathbf{a}}_k}{\|\hat{\mathbf{a}}_k\|} \tag{4.13}$$

Subsequently, the $\hat{\mathbf{a}}_0$ is the estimated common scalp distribution for the ERP-of-interest.

4.2.3.3 Estimating the amplitude σ_k

Ideally, the two vectors $\mathbf{a}_0$ and $\mathbf{a}_k$ are identical except for a scaling factor, which is exactly the unknown amplitude σ_k associated with the desired ERP source signal in the k-th EEG trial. Based on the estimated $\hat{\mathbf{a}}_k$ and $\hat{\mathbf{a}}_0$, the amplitude σ_k can be estimated through least-square fit as:

$$\min_{\sigma_k} \quad \|\hat{\mathbf{a}}_k - \sigma_k\hat{\mathbf{a}}_0\| \tag{4.14}$$

The solution for estimating $\hat{\sigma_k}$ is simply the projection of $\hat{\mathbf{a}}_k$ on the unit vector $\hat{\mathbf{a}}_0$ [57]:

$$\hat{\sigma_k} = \hat{\mathbf{a}}_k^\top \hat{\mathbf{a}}_0 \tag{4.15}$$

Subsequently, the estimated $\hat{\tau}_k$, $\hat{\mathbf{a}}_0$ and $\hat{\sigma}_k$ are used to parameterize the extracted signal during ERP analysis.

4.3 Simulation Study

4.3.1 Simulation scenario

In the literature, the nature of ERPs at single-trial level such as the level of amplitude, latency and waveform variations remains largely unknown. Thus, to quantify the performance of the proposed method, it is of interest to address its performance under the possible worst case scenarios. These scenarios are 1) when multiple ERPs occur and they are possibly overlapped with the ERP-of-interest; 2) when the ERP-of-interest experiences large latency variations across trials. Detecting ERP under such conditions is challenging and to examine its performance, preliminary assessment is performed on simulated ERP datasets. For the simulation study, in each run, a dataset of 100 simulated trials were created using the linear generative EEG model as follows. In each trial, the EEG epoch is computed to contain four different ERP source signals:

$$\mathbf{x}_k(t) = \sum_{i=1}^{4} \sigma_i \mathbf{a}_{0i} s_{k,i}(t) \tag{4.16}$$

where $\mathbf{a}_{0i}$ represents the common mixing vector of each ERP source signal. Each ERP source signal is defined as a normalised Gaussian waveform that has a width of δ_i and peaks at latency $\tau_{k,i}$:

$$s_{k,i}(t) = \exp\left(\frac{-(t-\tau_{k,i})^2}{2\delta_i^2}\right) \tag{4.17}$$

In this experiment, these four ERP source signals are chosen to represent the P100, N170, P200 and P300 ERPs in the real studies.

In the literature, N170 is one of the important ERPs, which has been extensively studied [27]. It is a negative peak which occurs around 170 ms after stimulus onset and is strongest in the occipital-temporal region. Previous studies have shown that N170 is often associated with the neural processing of face and object images, whereby the

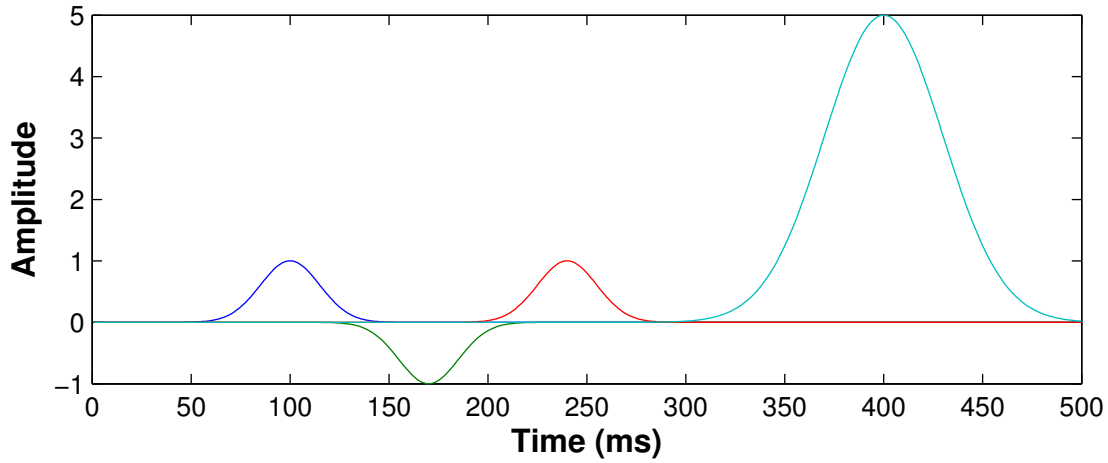

Figure 4.4: An example of the ERP source signals used in the simulation (when σ_i is absorbed into s_i instead of $\mathbf{a}_{0i}$)

N170 amplitude is larger when faces are presented as stimuli [27][108]. Accordingly, N170 has been extensively used in the study of face processing deficits that are characteristic of Autism Spectrum Disorder (ASD) [109]. Our goal in this simulation study is to extract the N170 that usually occurs between P100 and P200. The reason of choosing N170 ERP is that its peak amplitude is normally smaller compared to the commonly-examined P300 ERP. Thus, in practice, detecting the peak of N170 is generally harder since it is easily affected by the background EEG and also the possible overlap from other adjacent two ERPs such as P100 and P200. Meanwhile, although the EEG epochs used in the real N170 study usually do not involve P300 ERP, the P300 ERP is still generated in this study mainly to observe whether the performance of the proposed method is affected by the presence of larger ERP within the EEG epoch.

Table 4.1: Parameters used to generate a simulated ERP dataset

Parameters	P100	N170	P200	P300
σ_i	1	-1	1	5
δ_i (ms)	15	15	15	30
$\tau_{k,i}$ (ms)	$100 \pm \sigma_{\tau 1}$	$170 \pm \sigma_{\tau 2}$	$240 \pm \sigma_{\tau 3}$	400

Note: $\sigma_{\tau 1}$,$\sigma_{\tau 2}$ and $\sigma_{\tau 3}$ depends on the conditions.

The ERP parameters used to generated the simulated dataset is shown in Table 4.1. In each trial, the ERPs are generated to have fixed peak amplitudes and widths, but variable peak latencies (except for P300) which are chosen to be Gaussian distributed. In terms of $\mathbf{a}_{0i}$, the mixing vector for every ERP is fixed across trials. However, in each

run, they are randomly generated with uniform distribution [0,1] to represent different ERP scalp distributions. Each vector dimension is 40×1 so that each trial contains a 40-channel EEG segment. Also, note that these mixing vectors are normalised before generating the simulated data. Lastly, the whole process is repeated for 200 runs to test the overall performance under different mixing matrices.

In this stimulation study, for comparison purposes, the original template-matching (TM) based method proposed by Li *et al.* [57] is applied for single-trial ERP estimation (see Chapter 3.2.5.2 for more details). The parameters used in ENE-LD and TM method are described as follows. For our proposed method, to identify the N170 ERP, the time region for positive training samples were selected manually from 150 ms to 190 ms while the remaining time regions were set as negative training samples. During the estimation, the process described in Section 3.2.5.2 was performed. For TM, assume that the actual source signal for N170 ERP is known, an exact Gaussian template which matches the N170 waveform was used. To determine the peak latency, the predefined Gaussian template was scanned across time. The time which produces minimum error when reconstructing the EEG epoch with the time-shifted template was selected as the N170 peak latency. Based on our testing, Li's TM method are susceptible to the presence of large ERP. To generate minimum error, the method tends to reconstruct the largest ERP within the EEG epoch. Thus, for fair comparison, the template-matching method is constrained to search the N170 ERP within the time between 120 ms and 220 ms. Meanwhile, the best time-shifted template from the peak-latency estimation is also used for the scalp distribution and amplitude estimation.

Table 4.2: Parameters used to generate a simulated ERP dataset

Latency variation	$\sigma_{\tau 1}$ (ms)	$\sigma_{\tau 2}$ (ms)	$\sigma_{\tau 3}$ (ms)
condition 1	$\pm$ 5	$\pm$ 5	$\pm$ 5
condition 2	$\pm$25	$\pm$25	$\pm$25
condition 3	$\pm$25	$\pm$50	$\pm$25

To examine the performance of the proposed methods, three conditions were considered and were used to represent different levels of latency variations and degree of overlapping with the other ERPs. In the first condition, all the P100, N170 and P200 ERPs experience minimal latency variations across trials and has minimal chance of overlap with one another. In the second condition, these ERPs experience medium

latency variations and medium chances of overlap while the third condition has the N170 ERP experiences large latency variations and huge chances to overlap with other ERPs. The parameters for each conditions are shown in Table 4.2. In addition, to examine the proposed method in the noisy situation, these three conditions are also tested with the additive white noises under two different SNR conditions: -10 dB and 10 dB. In practice, the peak-picking technique used in the proposed ENE-LD is susceptible to random noises. Thus, assume that signal filtering is allowed, both TM and ENE-LD methods is tested on the moving-average filtered signals using a 20 ms window segment for averaging. Fig. 4.5 shows the examples of the ERP waveforms taken from one of the simulation runs for each condition in Table 4.2 under two different SNR levels. Note that in each simulation run, the mixing vectors of the ERPs are randomly generated and thus the strongest channel for each ERP can be different from each other. Also, due to this reason, it is normal for the plots in Fig. 4.5 to present weaker P100, P200 and P300 in the N170 strongest channel.

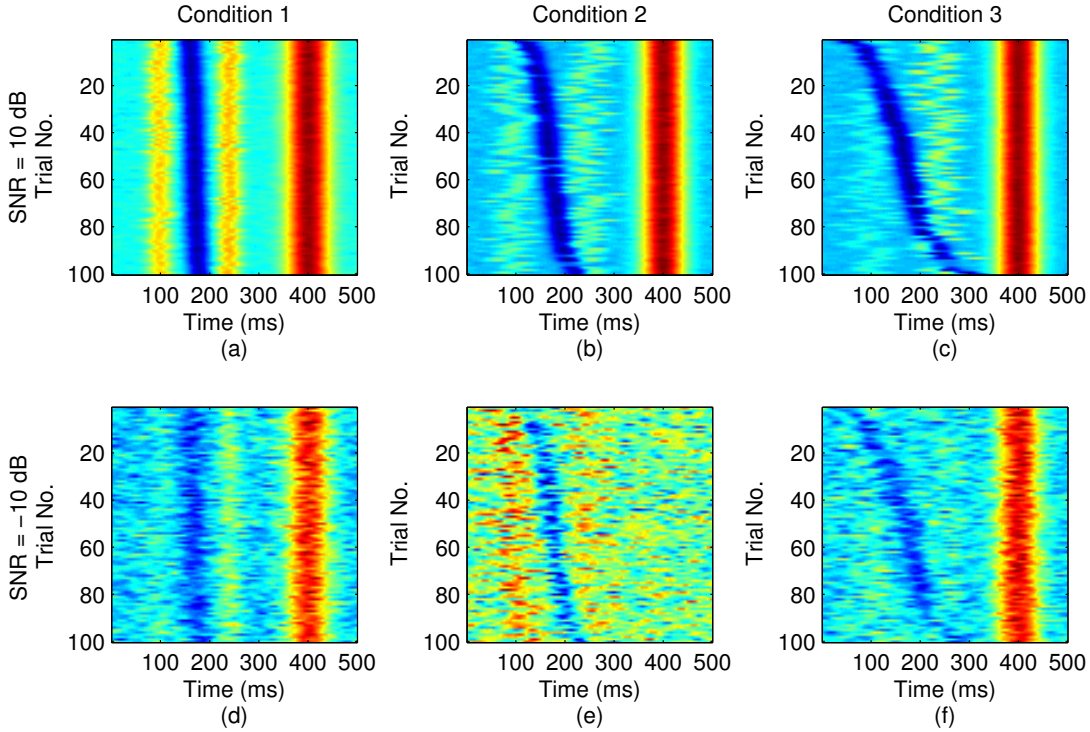

Figure 4.5: (a)–(c) Examples of ERP waveform taken from different simulation runs used in each condition when SNR = 10 dB, (d)–(f) when SNR = -10 dB. (Note that: each plot represents the waveforms from each EEG epoch that are taken from one of the channel which the N170 ERP is the strongest. These waveforms are also latency sorted based on their N170 peak latency.)

4.3.2 Simulation results and discussion

As latency estimation is crucial to the accurate estimation of amplitude and scalp distribution, one of the goals of this study is to examine the accuracy of peak latency estimated using our proposed method. Before examining the estimation results, the first assessment is to examine whether the proposed ENE-LD method is able to extract the desired ERP activity. The example of the extraction results in one of the simulation runs from each condition are shown in Fig. 4.6. On the first row of Fig. 4.6, the results show the simulated N170 waveforms generated for different conditions at 10 dB SNR. In each plot, the green-dotted line represents the boundary of the positive training window for the proposed ENE-LD. On the second row of Fig. 4.6, the results show the extracted waveforms provided by the proposed ENE-LD.

From these results, it shows that the proposed ENE-LD method is able to selectively extract the ERP-of-interest even under the presence of other ERPs. More importantly, these extracted ERPs capture the latency variations experienced by the simulated N170 ERP. In conditions 2 and 3, it is interesting to observe that although not all the training samples $\mathbf{x}_k(t)$ within the positive training window belong to the N170 ERP, ENE-LD is still able to extract the desired ERP activity accordingly. From the LDA classifier point of view, this is possibly attributed to the fact that after averaging the positive training window across trials, the mean scalp distribution $\boldsymbol{\mu}_1$ within this region still represent the scalp distribution of desired N170 ERP and thus allowing the ENE-LD to detect the desired N170 correctly. Similar results can also be observed in Fig. 4.7 when SNR = -10 dB. Except that, the signal extracted in condition 3 is much noisier in this case because the mean scalp distribution estimate is affected by the background noise and the limited amount of correct training samples within the training window. Overall, these empirical results show that it is plausible to train a classifier for extraction by dividing the EEG segment into ERP and non-ERP time region. Besides that, the result also demonstrates the robust extraction performance by the proposed ENE-LD whereby as long as the mean scalp distribution is estimated correctly, it can always separate the ERP-of-interest from other ERPs successfully.

Although the proposed ENE-LD methods managed to extract the ERP-of-interest accordingly, it is worth noticing that its extracted waveform does not match exactly with the desired ERP waveform. By taking condition 1 as an example, Fig. 4.8(a)

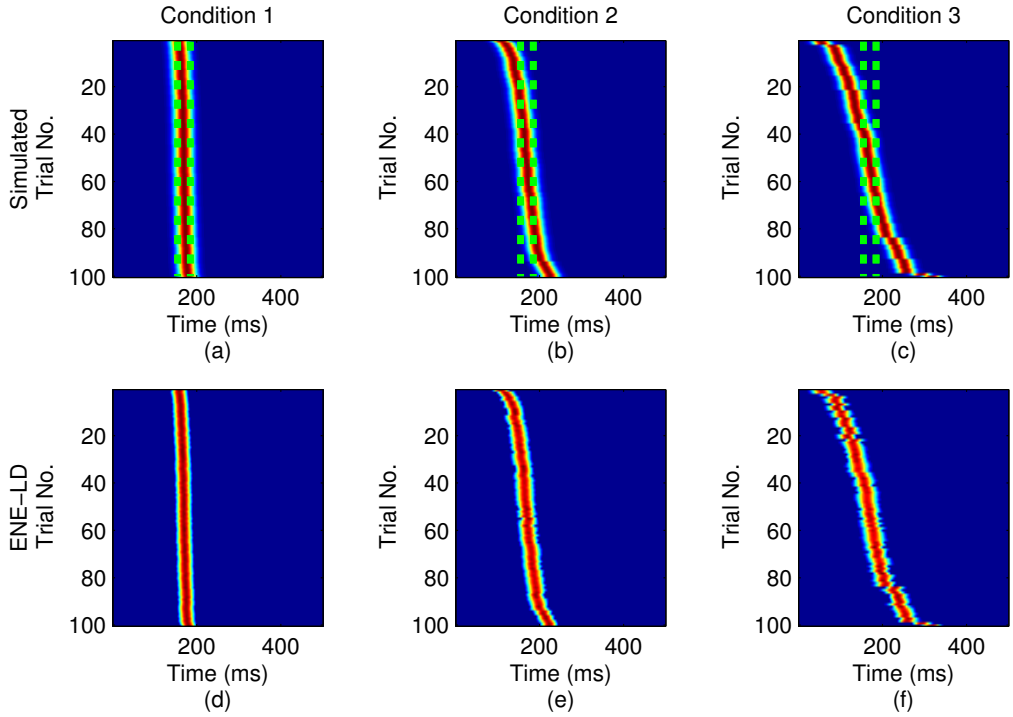

Figure 4.6: An example run from each condition when SNR = 10 dB: (a)–(c) Simulated N170 ERP waveforms (d)–(f) Estimated ERP waveforms by the proposed ENE-LD. Green dotted-lines represent the training window for ENE-LD method.

and (b) shows the trial-averaged signals from the ENE-LD method before and after removing the negative part of the signals. In Fig. 4.8(a), the result demonstrated how the proposed ENE-LD method projects the relevant ERP activity to the positive part of the signal and vice versa so that the relevant ERP activity can be traced easily. However, after removing the negative part of the signals, Fig. 4.8(b) shows that the trial-averaged signal from ENE-LD does not match exactly with the simulated N170 waveform. In general, mismatch between the original and extracted waveform have an adverse effect on the the scalp distribution and amplitude estimation. To understand the impact, we examine equation (4.11) that is used in the estimation process.

Equation (4.11) or its similar form based on Least-Square criterion are commonly used in various studies to estimate the scalp distribution of the extracted signal [86][87][88]. In Li's estimation framework, a good estimate on scalp distribution also leads to accurate amplitude estimation. For convenience, the trial number k is removed temporarily for the following analysis while the scalp distribution $\mathbf{a}_k$ is replaced temporarily as $\mathbf{a}$.

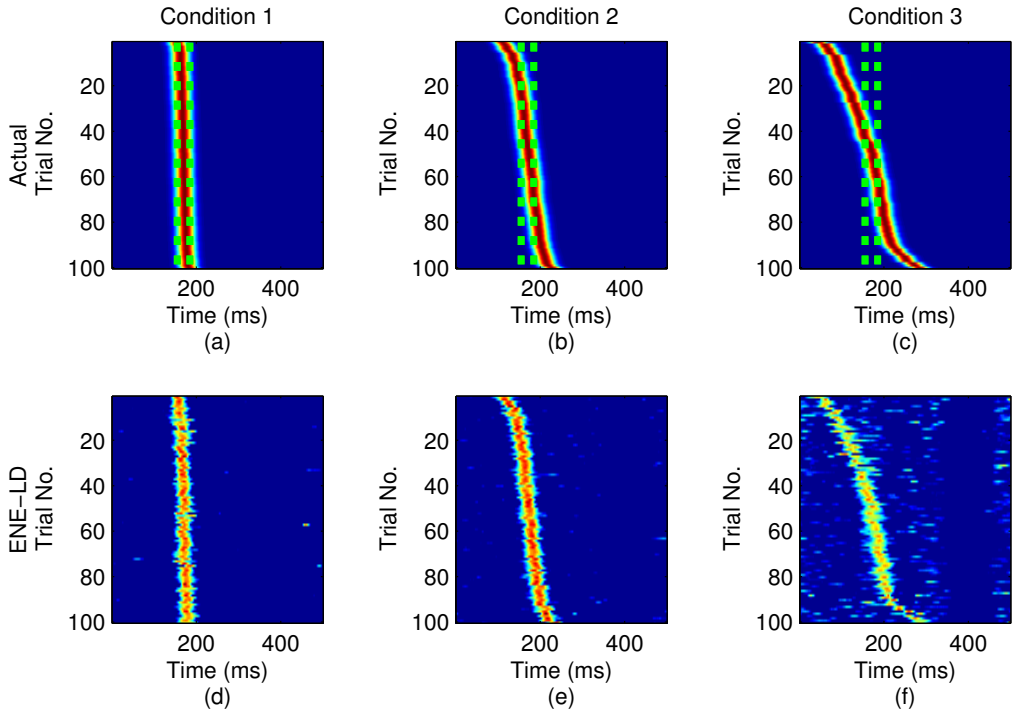

Figure 4.7: An example run from each condition when SNR = -10 dB: (a)–(c) Simulated N170 ERP waveforms (d)–(f) Estimated ERP waveforms by the proposed ENE-LD. Green dotted-lines represent the training window for ENE-LD method.

By replacing $\mathbf{x}(t)$ with (4.9), equation (4.11) can be rewritten as:

$$\begin{aligned}\hat{\mathbf{a}} &= \frac{\mathbb{E}\{\mathbf{x}(t)\tilde{y}(t)\}}{\mathbb{E}\{\tilde{y}^2(t)\}} \\ &= \frac{\mathbb{E}\{[\sigma\mathbf{a}_0\tilde{s}(t) + \sum_{i=1}^{N}\mathbf{b}_i n_i(t)].\tilde{y}(t)\}}{\mathbb{E}\{\tilde{y}^2(t)\}} \\ &= \frac{\mathbb{E}\{\sigma\mathbf{a}_0\tilde{s}(t)\tilde{y}(t) + \sum_{i=1}^{N}\mathbf{b}_i n_i(t)\tilde{y}(t)\}}{\mathbb{E}\{\tilde{y}^2(t)\}}\end{aligned} \tag{4.18}$$

Assume that the estimated ERP waveform $\tilde{y}(t)$ are uncorrelated to other source signals $n_i(t)$ whereby $\mathbb{E}\{n_i(t)\tilde{y}(t)\} = 0$, the solution for estimating $\mathbf{a}$ can be approximated as:

$$\hat{\mathbf{a}} = \frac{\mathbb{E}\{\sigma\mathbf{a}_0\tilde{s}(t)\tilde{y}(t)\}}{\mathbb{E}\{\tilde{y}^2(t)\}} \tag{4.19}$$

In the above equation, it can be observed that to obtain a good estimate on the common scalp distribution $\mathbf{a}_0$ which is the normalised version of $\mathbf{a}$, the estimated ERP waveform $\tilde{y}(t)$ does not necessarily have to be exactly the same as $\tilde{s}(t)$ as long as it is uncorrelated with other ERP source signal. This is mainly because the scalar term $\sigma\tilde{s}(t)\tilde{y}(t)$ can be dropped after the normalisation of $\mathbf{a}$. However, the exact amplitude σ of the desired ERP cannot be recovered when there is a mismatch between the original and estimated

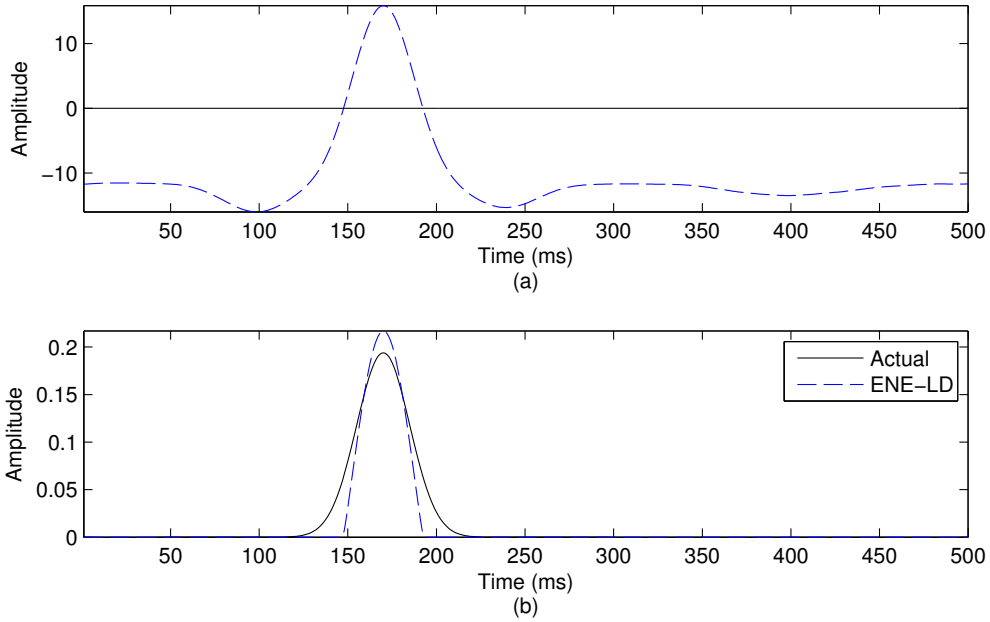

Figure 4.8: (a) Trial-averaged signal from ENE-LD in condition 1 before removing the negative part of the signal. (b) normalised trial-averaged signal from ENE-LD after removing the negative part of the signal. For comparison, the normalised trial-averaged signal of the simulated N170 ERP is also plotted. Note that: All the trial-averaged signals are averaged with peak-aligned at 170 ms.

ERP waveforms. Moreover, the estimated amplitude is biased since $\mathbb{E}\{\tilde{s}(t)\tilde{y}(t)\} \neq 1$. The above situation is also an issue faced by the TM method when designing the template for the real ERP application. To understand the performance of the proposed ENE-LD method in single-trial estimation, a further analysis on its estimation results proceeds as follows.

Fig. 4.9 and 4.10 show the examples of peak latency estimation performance by TM and ENE-LD respectively under different SNR conditions. From the second row of Fig. 4.9 and 4.10, the results show that the estimated peak latency by the proposed ENE-LD are quite consistent to the simulated N170 peak latency. Besides that, the proposed ENE-LD method has a large peak detection range that allows them to search the N170 peaks successfully even if the underlying N170 ERP experiences large latency variations as shown in Fig. 4.9(f). However, TM has a shorter detection range partly due to the constraint we imposed on the time-searching range. Thus, when the underlying N170 ERPs experience large latency variation, TM's performance degrades significantly as shown in plot (c) of Fig. 4.9 and 4.10. In the study, for fair comparison, the template matching technique is excluded from condition 3 in the following analysis.

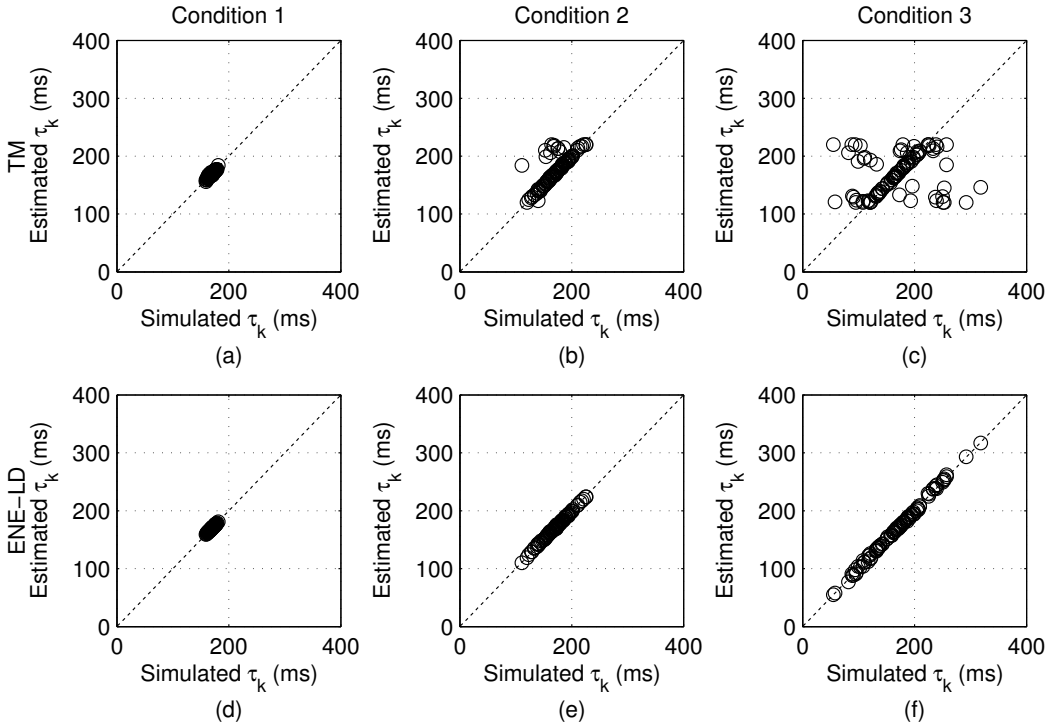

Figure 4.9: (a)–(c) shows the examples of scatterplots of estimated latency vs simulated latency for TM method under different conditions when SNR = 10 dB. Similar graphs are also plotted for the ENE-LD method in (d)–(f).

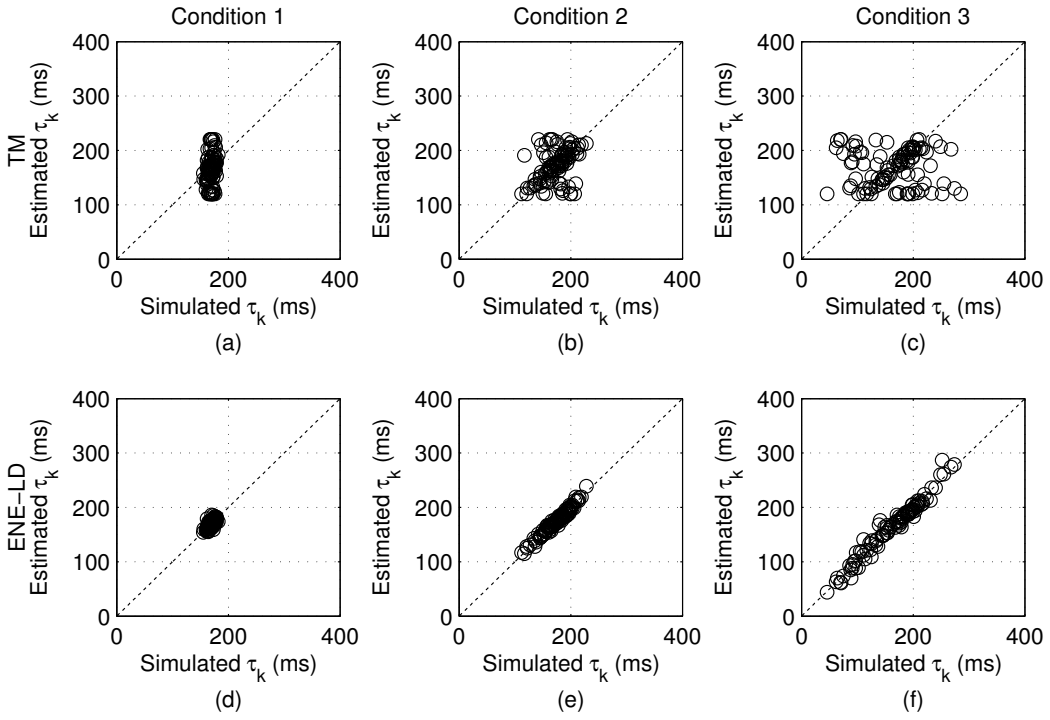

Figure 4.10: (a)–(c) shows the examples of scatterplots of estimated latency vs simulated latency for TM method under different conditions when SNR = -10 dB. Similar graphs are also plotted for the ENE-LD method in (d)–(f).

Table 4.3: The average of mean absolute difference between the simulated and estimated peak latencies of N170 ERP after 200 runs with different mixing vectors.

SNR	Method	Condition 1	Condition 2	Condition 3
10 dB	TM	0.97 ± 0.15	10.02 ± 2.33	–
	ENE-LD	0.31 ± 0.06	0.80 ± 0.15	2.04 ± 0.49
-10 dB	TM	18.39 ± 2.13	24.54 ± 3.07	–
	ENE-LD	3.85 ± 1.04	4.82 ± 1.62	11.21 ± 5.52

Table 4.3 shows the average of mean absolute difference (MAD) between the simulated and estimated peak latencies for the two methods after 200 runs where MAD is computed as:

$$\text{MAD (ms)} = \frac{1}{N_k} \sum_{i=1}^{N_k} \mid \tau_k - \hat{\tau}_k \mid \tag{4.20}$$

where τ_k and $\hat{\tau}_k$ are the simulated and estimated peak latency respectively. From the table, the results show that SNR and the chances of overlap are the factor that affects TM and ENE-LD during the peak latency estimation. In general, both TM and ENE-LD performance are better when SNR is higher and the chances of overlap are minimal. When compared between TM and ENE-LD, the table shows that ENE-LD has better peak latency estimates where its MAD results are lowest in all conditions. It is also worth noting that TM is more sensitive to the noisy background signals. By taking condition 1 as an example, its MAD in the peak latency estimates increases dramatically from 0.97 ms at 10 dB SNR to 18.39 ms at -10 dB SNR. In contrast, for the same condition, the MAD of the proposed ENE-LD only increases slightly from 0.31 ms to 3.85 ms.

Table 4.4: Average correlation coefficient between the simulated and estimated scalp distribution by different methods after 200 runs with different mixing vectors

SNR	Method	Condition 1	Condition 2	Condition 3
10 dB	TM	0.997 ± 0.001	0.942 ± 0.024	–
	ENE-LD	0.999 ± 0.001	0.985 ± 0.060	0.971 ± 0.011
-10 dB	TM	0.952 ± 0.021	0.884 ± 0.048	–
	ENE-LD	0.991 ± 0.004	0.978 ± 0.008	0.954 ± 0.018

Table 4.4 shows the average correlation coefficient between the simulated and estimated scalp dsitribution for TM and ENE-LD respectively. The results show that despite the mismatch, the scalp distributions by ENE-LD are highly correlated to the simulated scalp distribution whereby it also managed to achieve the average correlation coefficient of at least 0.95 under SNR = -10 dB. However, the correlation coefficient

decreases slightly across conditions when there are higher latency variations and increased chances of overlapping with the other ERPs.

Table 4.5: Paired sample t-test between correlation coefficient achieved by ENE-LD and TM

SNR	Condition 1	Condition 2	Condition 3
10 dB	$t(199)$=32.467, p <.001	$t(199)$=29.355 , p <.001	–
-10 dB	$t(199)$=28.442, p <.001	$t(199)$=31.726 , p <.001	–

In comparison to TM, the correlation coefficient achieved by both ENE-LD and TM are close to each other in some conditions. For this reason, paired sample t-tests was performed for each condition to compare the correlation coefficient achieved by ENE-LD and TM. Table 4.5 revealed that in all cases, p-value from every paired sample t-test are less than 0.001, suggesting that the performance of both ENE-LD and TM are statistically different to each other. In addition, since t-value are always positive, the results also reveal that ENE-LD are statistically better than TM in all conditions.

Table 4.6: Average estimated amplitude by different methods after 200 runs with different mixing vectors (Simulated amplitude σ = 1)

SNR	Method	Condition 1	Condition 2	Condition 3
10 dB	TM	0.912 ± 0.019	0.397 ± 0.278	–
	ENE-LD	0.871 ± 0.012	0.729 ± 0.160	0.664 ± 0.167
-10 dB	TM	0.286 ± 0.237	0.163 ± 0.145	–
	ENE-LD	0.735 ± 0.090	0.603 ± 0.149	0.504 ± 0.154

Table 4.6 shows the average estimated amplitude for the two methods. First, the results show that neither TM nor ENE-LD managed to obtain an exact amplitude estimate of 1 for all conditions. This result is reasonable because when there is overlap between ERPs, the uncorrelated assumption no longer holds in (4.18) and thus it affects the amplitude estimation. From the above conditions, it can be observed that both TM and ENE-LD does not work very well in condition 2 and 3 when the chances of overlapping are higher. Not only the amplitude estimates deviate largely from one, the estimates also experience higher variance compared to condition 1. For TM, another reason that it works poorly in amplitude estimation is due to the poorly estimated peak latency. This effect can be observed in condition 1 of Table 4.6 when SNR = -10 dB. As the peak latency decides the location of the template, incorrect peak latency generally affects the choice of template used for amplitude estimation.

In the condition 1 where SNR = 10 dB and the chances of overlapping are minimal, both TM and ENE-LD obtain an average estimate of 0.912 and 0.871 respectively. This result suggests that the mismatch from the proposed methods are small and thus the estimates are close to the simulated value of 1. The small variance in the amplitude estimates also suggests that both methods may have converged. However, due to the bias from the mismatch, the exact amplitude cannot be obtained. When compared between TM and ENE-LD, TM shows a better performance in this case. One of the main reasons is that TM employs the template that match exactly with the ERP-of-interest for estimation. In practice, designing the suitable template for ERP is usually difficult and the strict assumption that the ERP waveform are fixed across trials is often required. Thus, by taking these considerations into account, the performance in amplitude estimation by our data-driven extraction method is still considered superior when compared to TM.

Overall, from the above results, there are two observations that can be summarised from the simulation study. Firstly, when the time region of the ERP-of-interest is known, ENE-LD can be implemented to selectively extract the desired ERP activity even under the presence of other ERPs. From our simulation study, the results also show that the ERP time region does not have to be exactly correct as long as it is representative and allows ENE-LD to learn the scalp distribution of the ERP-of-interest correctly. Once this condition is fulfilled, ENE-LD is able to detect the ERP-of-interest reasonably well at single-trial level even if the latency variation is large. Subsequently, it also provides ENE-LD a better performance in peak latency estimation when compared to the original TM method. Secondly, although the extracted ENE-LD waveform is not exactly the same as the simulated ERP waveform, the result shows that this mismatch does not affect ENE-LD's performance in scalp distribution estimation. Except that, the amplitude estimated by ENE-LD is always biased. Thus, unless the the mismatch is small and consistent across trials, ENE-LD may not be the best tool for amplitude estimation.

4.4 Experiment with a real ERP dataset

4.4.1 Dataset and experimental procedure

The analysis of N170 remains a challenge especially for the conventional LD method mainly because N170 usually occurs with similar peak amplitude in two different experimental conditions. Thus, for the conventional LD, the small differences in N170 make it relatively harder to be exploited when compared to the large differences commonly observed between P300 and non-P300 trials. For this reason, our objective in this section is to examine the performance of the proposed ENE-LD method in extracting the N170. As a comparison, the conventional LD methods are also evaluated. For convenience, they will be referred as cLD.

The selected N170 dataset is part of the publicly available multi-modal face dataset contributed by Henson *et al.* [111]. The original dataset contains EEG, MEG and fMRI recordings of a healthy subject for the purpose of measuring and studying the neurocognitive responses when performing face perception tasks. Two sessions were conducted on the same subject using the same stimuli presentation paradigm, based on Phase 1 of a previous study by Henson *et al.* [112]. In each of these sessions, a total of 86 faces and 86 scrambled faces were randomly presented [111]. The measurements were made by a 128-channel BioSemi ActiveTwo EEG system with a sampling frequency of 2048 Hz. All electrodes were later re-referenced to the common-average reference.

During the signal pre-processing stage, EEG signals from both sessions were band-pass filtered[1] between 0.1-40 Hz. In addition, a forward-backward moving-average filter which uses 20 ms averaging window is also applied. For our study, we combined these two sessions to extract trials for both face and non-face experimental conditions. For each trial, a 400 ms segment of each EEG signal was extracted starting from stimulus onset. Baseline correction was performed using a 200 ms segment of the EEG signal prior to stimulus onset. A total of 150 and 151 trials were extracted respectively for both the faces and non-face dataset. The trial-averaged signals for faces and non-faces (i.e. scrambled faces) stimuli are shown in Fig. 4.11.

In order to examine our method, we applied the proposed ENE-LD to perform

[1] A combination of a forward-backward 2nd order elliptic highpass and 9th order elliptic lowpass filters.

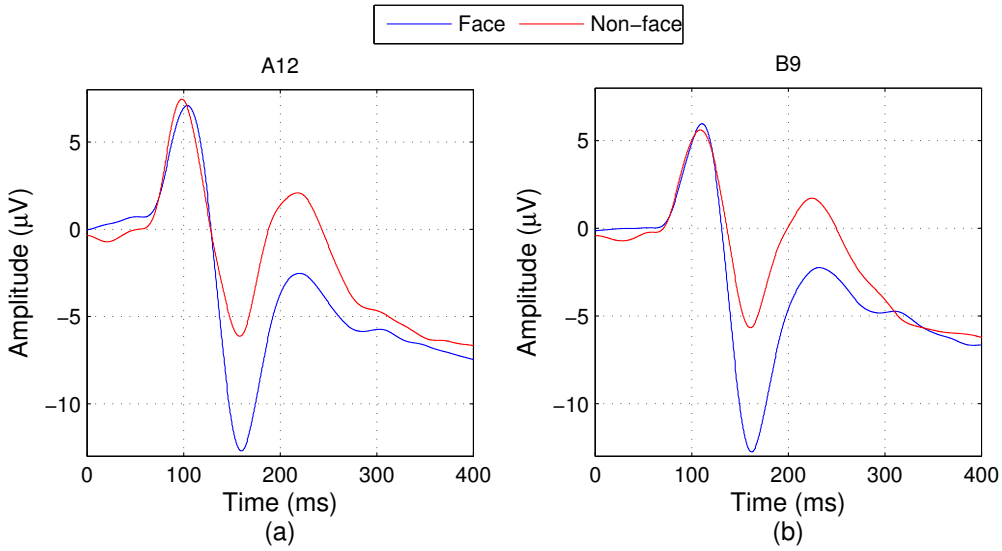

Figure 4.11: Trial averaged signals from channel A12 and B9 for face (blue) and non-face (red) conditions. Note: The channel labels are based on 128-channel BioSemi headcap and they represent the occipital-temporal region at left and right hemisphere of the brain.

single-trial N170 extraction for both face and non-face conditions. To apply ENE-LD, the positive training window for N170 is defined as 140 – 180 ms. This training window is selected because our observation on Fig. 4.11 suggests that the N170 peak latency of this subject is located around 163 ms. For peak latency and scalp distribution estimation, the procedure described in Section 4.2 was used. For comparison, the cLD implementations that are based on logistic regression (LR) [53] and on LDA [89] were also applied whereby the classifier was trained using the procedure as shown in Fig. 4.1 where

1. the training window is the same as the N170 time window used in the ENE-LD method.

2. The positive and negative training samples were taken respectively from the face and non-face trials.

4.4.2 Results and discussion

4.4.2.1 Results from N170

Fig. 4.12(a) and (b) shows the trial-averaged signals extracted from ENE-LD before the negative part of the signal was removed. From the figure, the result shows that

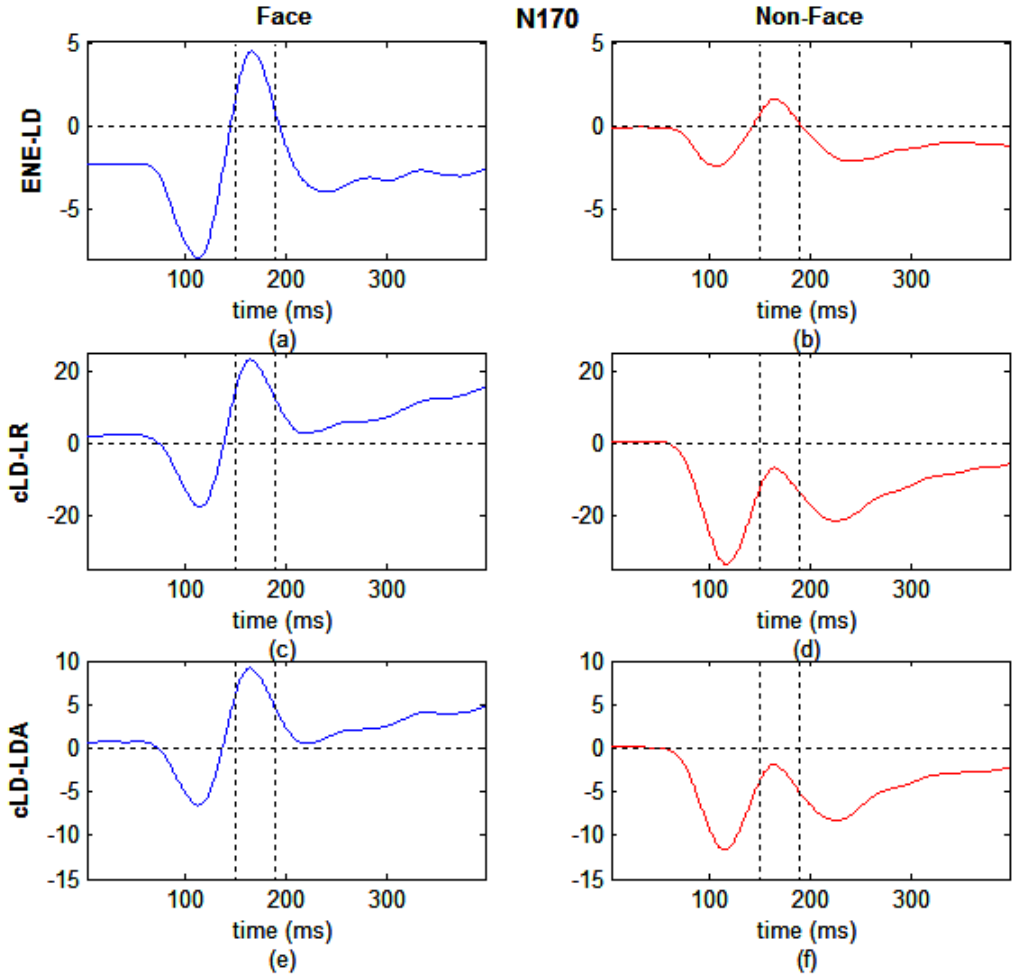

Figure 4.12: Each row represents the trial-averaged extracted signals for N170 from ENE-LD, cLD-LR and cLD-LDA respectively while each column represents the results taken from 'Face' and 'Non-Face' condition respectively. The vertical dotted-line represents the defined ERP time region.

ENE-LD can be applied to extract N170 under both face and non-face conditions. Fig. 4.12(c)–(f) shows the trial-averaged signals extracted from the cLD methods. When comparing the results from cLD-LR and cLD-LDA, the trial-averaged signals from both method are very similar except for a scaling difference. By first examining the extraction outcome of both cLD methods, the result shows that the cLD methods fail to isolate the irrelevant ERPs from their extracted signals in both face and non-face conditions. In addition, since the cLD methods do not apply thresholding on the signal, all the extracted signals can eventually lead to significant estimation errors in the scalp distribution. Thus, for the rest of the analysis, only the results from ENE-LD are used for single-trial estimation.

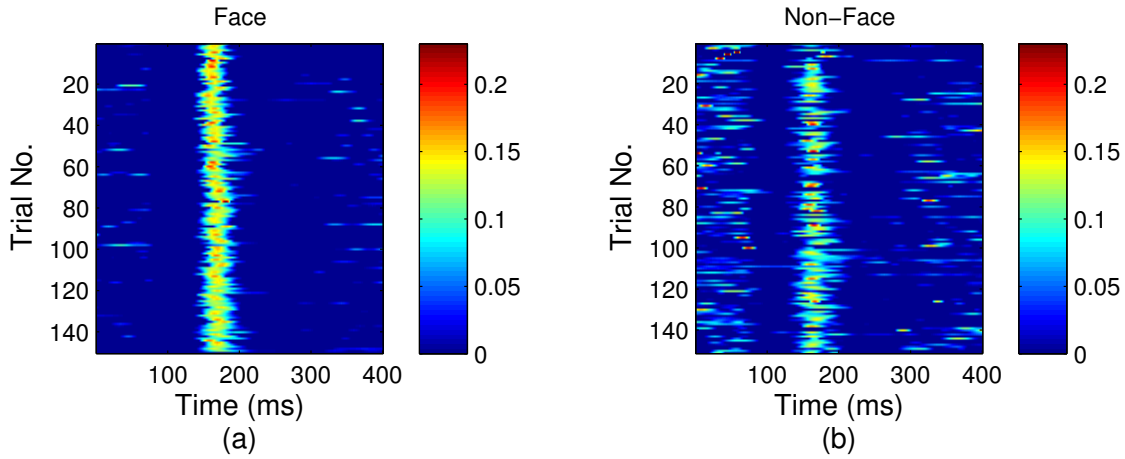

Figure 4.13: The normalised $\tilde{y}_k(t)$ for N170 from (a) face conditions and (b) non-face conditions.

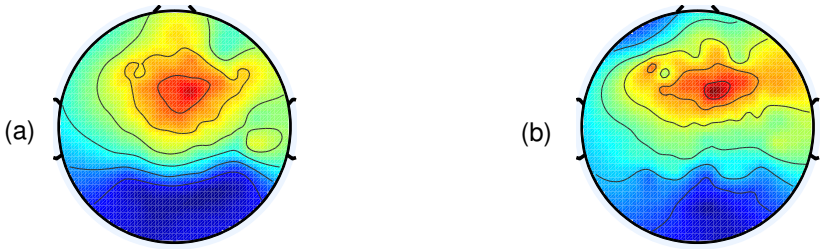

Figure 4.14: The estimated scalp distribution $\hat{\mathbf{a}}_0$ for N170 from (a) face and (b) non-face conditions.

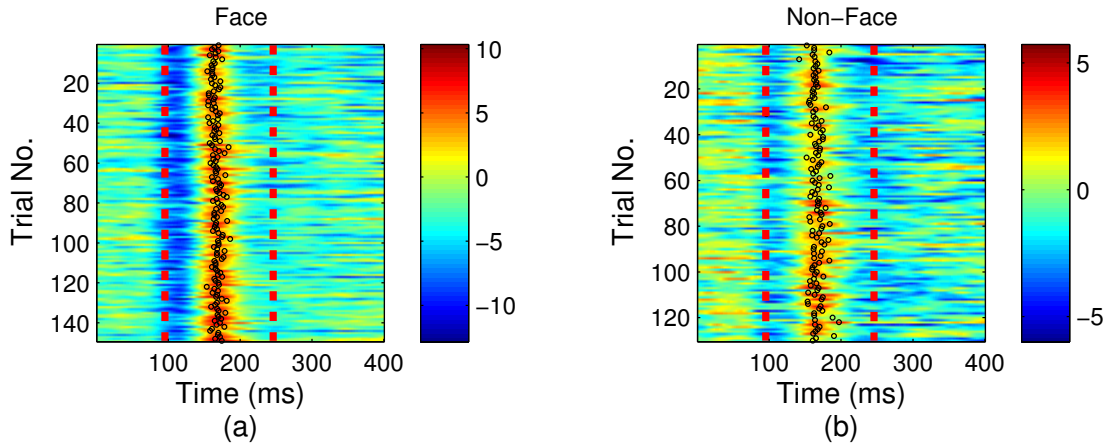

Figure 4.15: The estimated peak latency (black circle) for N170 from (a) face and (b) non-face conditions. The red-dotted lines show the time range for searching the peak latency. For better visualization, the results are plotted on the original $y_k(t)$.

Fig. 4.13 shows the single-trial N170 extraction results under face and non-face conditions. Overall, the results show that the proposed ENE-LD method is able to separate the ERP-of-interest reasonably well from the irrelevant EEG activities. Interestingly, a visual examination of the plots shows that regardless of face or non-face conditions, N170 experiences minimal latency variations with consistent time regions across the trials. This characteristic is similar to Condition 1 in the simulated study where there is no sign of overlapping among the ERPs. Fig. 4.14(a) and (b) shows the common scalp projection estimated from both face and non-face condition respectively. The figure reveals that both scalp projections for N170 have occipital-temporal distributions, which matches the findings reported in [108] and [110]. Interestingly, in the literature, for face condition, N170 usually has a more profound negativity at the right hemisphere. However, our results show that the differences between the hemispheres are less pronounced for this subject.

Overall, the results show that the extraction from ENE-LD are not entirely free from noise as can be seen from Fig. 4.13(b). In this figure, although ENE-LD managed to extract the N170 across the trials, the extracted signals still contain remnants of irrelevant EEG activities. From our examination, a possible reason is the weak N170 activities under the non-face condition as depicted in the trial-averaged signals of Fig. 4.11. In order to provide a more accurate latency estimation, those trials that do not have positive $y_k(t)$ (see section 4.2) within the defined time region were removed from the analysis while latency searching range is limited to between 95 ms and 245 ms.

Fig. 4.15 shows the peak latencies (black circles) for N170 as estimated by ENE-LD from the constrained time window (red-dotted line). For better visualization, the results were plotted over $y_k(t)$ for all face and non-face trials. Fig. 4.15 suggests that N170 can be readily observed in a single trial, with its latencies stable around 166 ± 6 ms and 165 ± 10 ms respectively for face and non-face conditions.

4.4.2.2 Additional Results on P100 and P200

In the literature, N170 usually appears in the P100-N170-P200 complex as shown in the raw trial-averaged signals depicted in Fig. 4.11 [26][27]. Thus, as the additional tests, we applied the ENE-LD and both cLD methods to extract the P100 and P200

ERPs. The ERP time windows for P100 and P200 are defined as 90–130 ms and 200–240 ms respectively.

Fig. 4.16 and 4.17 show the trial-averaged extracted signals for P100 and P200 respectively. Similar to the N170 results, Fig. 4.16 shows that ENE-LD outperforms cLD-LR and cLD-LDA in extracting the P100 from both face and non-face conditions. However, in Fig. 4.17, the result shows that two distinct positive peaks appear in the positive part of the extracted signals. This suggests that our ENE-LD and both the cLD methods fail to discriminate P200 from the presence of P100 under both face and non-face conditions. A possible reason that ENE-LD fails to operate properly is because P200 in this dataset has an unusually small peak amplitude when compared to P100. This is contrary to the results in the literature where P200 are commonly found to have same peak amplitude as P100 [26][27][108][110]. For this reason, P200 is removed from the following single-trial peak latency and scalp distribution estimation.

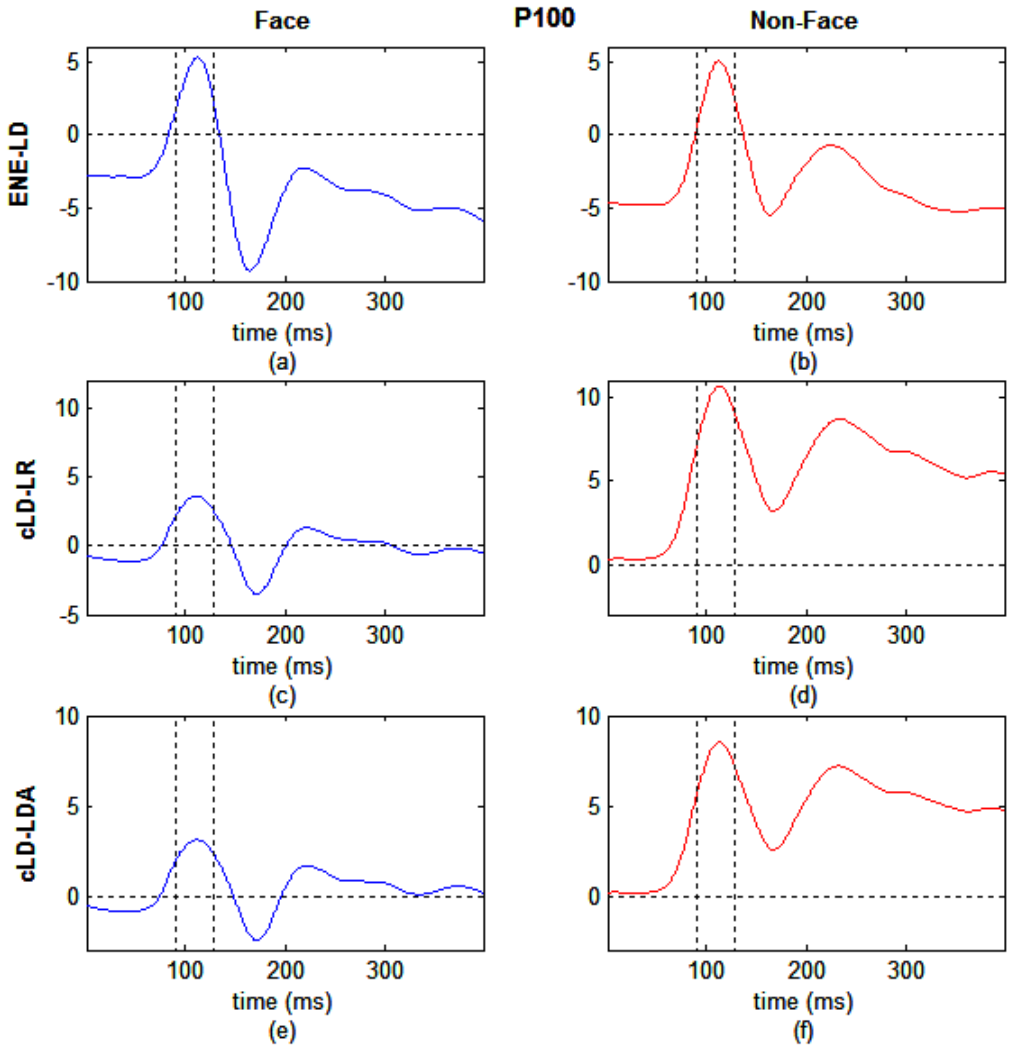

Figure 4.16: Each row represents the trial-averaged extracted signals for P100 from ENE-LD, cLD-LR and cLD-LDA respectively while each column represents the results taken from 'Face' and 'Non-Face' condition respectively. The vertical dotted-line represents the defined ERP time region. The horizontal dotted-line represents the baseline.

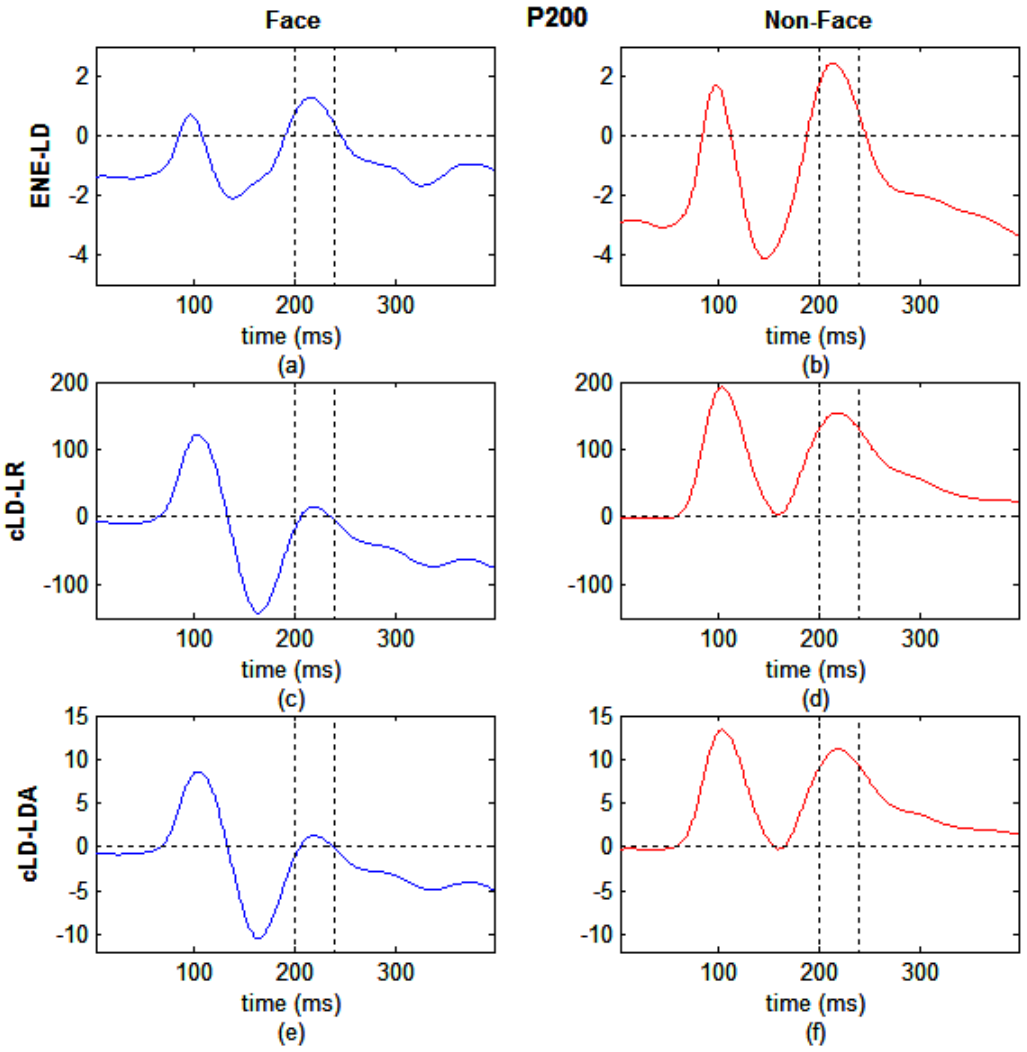

Figure 4.17: Each row represents the trial-averaged extracted signals for P200 from ENE-LD, cLD-LR and cLD-LDA respectively while each column represents the results taken from 'Face' and 'Non-Face' condition respectively. The vertical dotted-line represents the defined ERP time region. The horizontal dotted-line represents the baseline.

Similar to the N170 case, Fig. 4.18(a) and (b) show that our ENE-LD can extract P100 from both face and non-face conditions reliably. The scalp distribution given by the extracted signals in Fig. 4.19 also matches the typical P100 occipital-temporal distribution given in [108] and [110]. Lastly, by constraining the peak latency searching range to between 25 ms and 175 ms, Fig. 4.20(a) and (b) show that P100 are stable around 112 $\pm$ 6 ms and 113 $\pm$ 5 ms respectively for face and non-face conditions. From the additional tests, our results suggest that both P100 and N170 are readily

observed at single-trial level using ENE-LD.

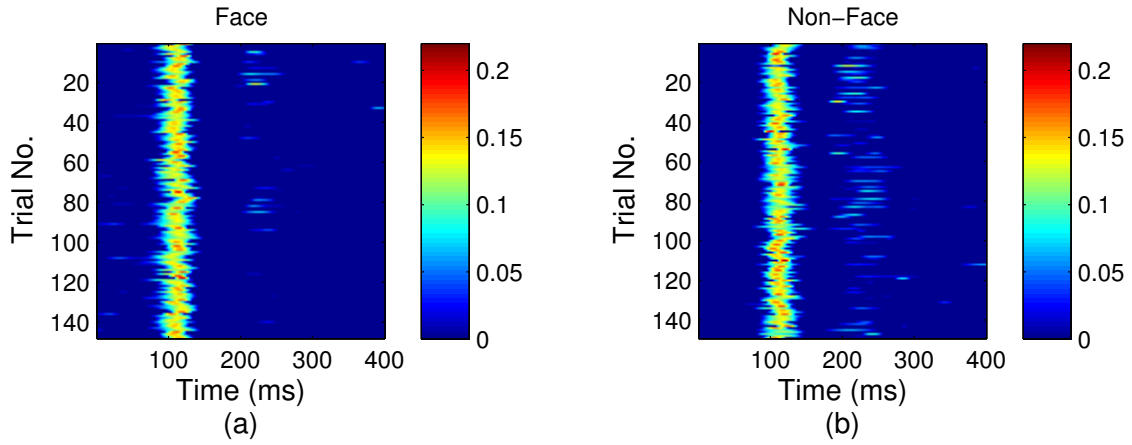

Figure 4.18: The normalised $\tilde{y}_k(t)$ for P100 from (a) face conditions and (b) non-face conditions.

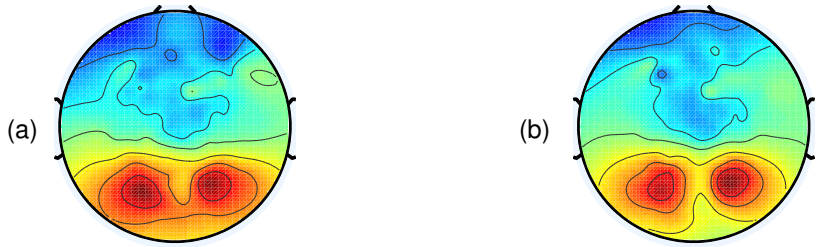

Figure 4.19: The estimated scalp distribution $\hat{\mathbf{a}}_0$ for P100 from (a) face and (b) non-face conditions.

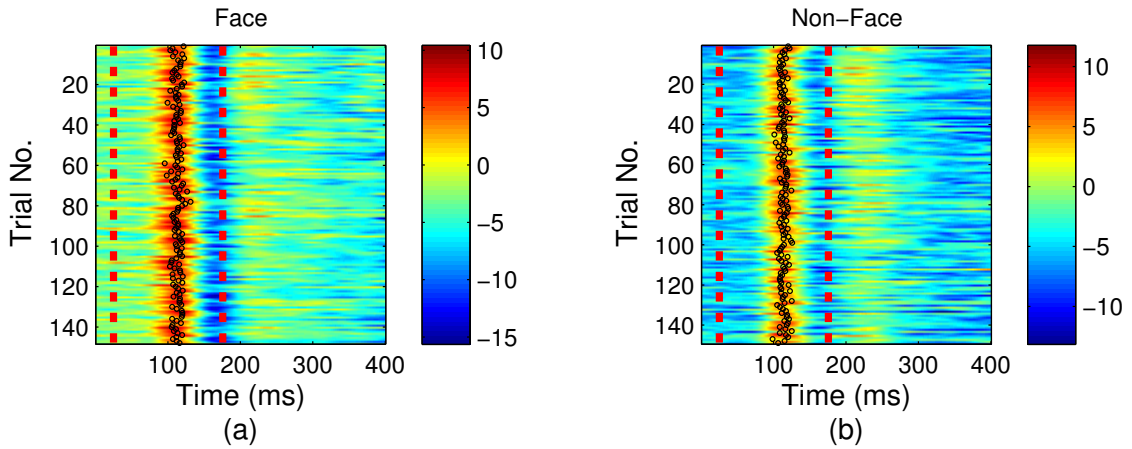

Figure 4.20: The estimated peak latency (black circle) for P100 from (a) face and (b) non-face conditions. The red-dotted lines show the time range for searching the peak latency. For better visualization, the results are plotted on the original $y_k(t)$.

4.5 Conclusion

In this chapter, motivated by the consistent finding that each ERP has their own unique scalp distribution, a new extraction method called ENE-LD is developed. This method performs the separation through learning the differences of the mean scalp distribution between the ERP and non-ERP time regions. Similar to the conventional LD method, to perform the separation successfully, it utilises the classifier's discriminative projection and threshold to extract the time course of the ERP-of-interest. Unlike PCA and ICA, ENE-LD does not require any source selection. In cases where the time-region of the ERP-of-interest is roughly known, the method allows one to perform a direct ERP extraction so that only the dynamics of the ERP-of-interest is examined at the single-trial level.

The simulation study which involved the mixed signals of four different ERPs demonstrated that ENE-LD can be applied to extract the ERP-of-interest even under the presence of other ERPs which may be larger or overlapping with the ERP-of-interest. When compared to the existing template-matching method, the proposed ENE-LD has better performance in peak latency and scalp distribution estimation. In this study, the proposed ENE-LD method is also tested on a real N170 ERP dataset whereby the method is applied to three different ERPs (i.e. P100, N170 and P200). Our results showed that the ERP extraction using ENE-LD is generally feasible for P100 and N170 whereby the extraction results allow us to understand the scalp distribution and the consistency of the P100 and N170 ERP time regions at the single-trial level. Our results also demonstrated that the SNR of the ERP-of-interest remains an important factor to the successful application of ENE-LD.

Chapter 5

One-unit Independent Component Analysis with Reference

5.1 Overview

Among the multi-channel extraction methods, ICA is one of the more successful techniques which has been used extensively in different signal processing fields. The early efforts of applying ICA for single-trial ERP analysis has been demonstrated successfully in [63] and [64]. However, the extraction of ERP using ICA is not always straightforward because traditional ICA recovers all the independent source signals from the mixtures simultaneously. After the decomposition, the task-related source signal is still required to be selected before the desired ERP can be studied [62][114][115][116].

Since not all source signals are eventually of interest, one of the increasingly popular methods is the application of temporally constrained ICA, also known as ICA-with-reference (ICA-R) [122]. ICA-R is an extension of traditional ICA. It attempts to perform signal extraction and source selection jointly by adding prior information as a constraint. By incorporating the prior temporal information into one or a set of reference signals, ICA-R utilises these reference signals to search for one or a subset of relevant independent source signal directly without performing a full ICA decomposition. Specifically, the ICA-R that uses one reference signal to extract one desired source signal is often known as one-unit ICA-R while the ICA-R that uses multiple reference signals to extract multiple source signals is known as ICA-MR.

Currently, one-unit ICA-R has been applied successfully in various biomedical signal processing applications such as artifact removal [117] and rhythmic activities extraction [118][119]. Despite its success, to the best of our knowledge, the application of one-unit ICA-R remains unexplored in ERP studies while a recent attempt of ICA-R was based on ICA-MR [121] where it is employed to extract the P300 ERP for the P300-BCI applications. Unfortunately, the approach by [121] is (i) restrictive since it is only developed for the P300-BCI application, and (ii) inefficient since it still requires a set of reference signals before P300 can be extracted.

In this work, to address the aforementioned issue, we present an implementation of one-unit ICA-R that employs only one designated reference signal to extract the desired ERP. The advantages of using the proposed one-unit ICA-R are as follows. Firstly, the method is developed for general ERP applications. Secondly, the proposed ICA-R do not extract redundant signals that have to be subsequently removed. Thus, unlike the traditional ICA and the one proposed by [121], source signal selection is not required after applying the proposed one-unit ICA. Consequently, it is also computationally more attractive than the existing ICA methods mentioned above.

The remainder of this chapter is organized as follows. The procedure for implementing the one-unit ICA-R is described in Section 5.2.1. The successful application of ICA-R relies heavily on the design of the reference signal. Thus, to examine the type of reference signal that is most suitable for ERP extraction, three different types of reference signals are proposed in Section 5.2.4. The proposed reference signals are then examined with a simulation study in Section 5.3. In Section 5.4, since our proposed ICA-R is motivated by the ICA-MR, the performance of our proposed ICA-R is compared against the ICA-MR using the P300-BCI speller dataset. Lastly, Section 5.5 examines the the proposed ICA-R on a larger P300 ERP dataset collected from the traditional oddball paradigm before conclusion is drawn in Section 5.6.

5.2 Proposed method

5.2.1 Traditional ICA

Traditional ICA is a blind source separation technique whose aim is to recover a set of source signals $\mathbf{s}(t)$ from a set of observed signals $\mathbf{x}(t)$ [59] (see (3.1), repeated below):

$$\mathbf{x}(t) = \mathbf{A}\mathbf{s}(t) \tag{5.1}$$

where the source signals $\mathbf{s}(t)$ is assumed to have zero-mean and with unit variance. Suppose the EEG source signals are all temporally independent[1] and spatially-fixed (or mixing matrix $\mathbf{A}$ is stationary) [54][55], the objective of ICA is to find a $N_s \times N_e$ demixing matrix[2] $\mathbf{W}$ such that the recovered source signals are maximally independent:

$$\mathbf{y}(t) = \hat{\mathbf{s}}(t) = \mathbf{W}^\top \mathbf{x}(t) \tag{5.2}$$

where the multi-channel EEG signals are represented as $\mathbf{x}(t)$.

Most traditional ICA performs the extraction by first decomposing the raw EEG signals simultaneously into a number of source signals that are equal to the number of EEG channels. Thus, for EEG recordings that have large number of EEG channels, the extraction process can be time-consuming while the selection of the desired source signals can be a complicated process.

5.2.2 One-unit ICA-R

In situations where the ERP-of-interest is known beforehand, it seems a more efficient method is to apply the one-unit ICA-R since it requires only one reference signal and hence yields only one extracted signal. In this work, the one-unit ICA-R method proposed by [122] is employed and our focus in this chapter is to determine a suitable reference signal for ERP extraction. For convenience, the time index t is omitted from $\mathbf{x}(t)$, $y(t)$ and $r(t)$ in the following equations of this section.

In the literature, the ICA implementation that aims to extract one source signal is usually known as one-unit ICA [120]. The optimization problem that defines the one-unit ICA is formulated as follows:

$$\max_{\mathbf{w}} \quad J(y) = [\mathbb{E}\{G(y)\} - \mathbb{E}\{G(v)\}]^2 \tag{5.3}$$

[1] In Chapter 4, we observed that both P100 and N170 ERPs reside closely within their specific time regions across trials and do not overlap with each other despite being close in terms of their peak latencies. In addition, since ERPs are originated from different part of the brain, these observations suggest that the assumption of temporally independence and uncorrelatedness between EEG/ERP source signals can be relevant.

[2] In ICA literature, the demixing matrix also means the set of spatial filters used to recover the source signals.

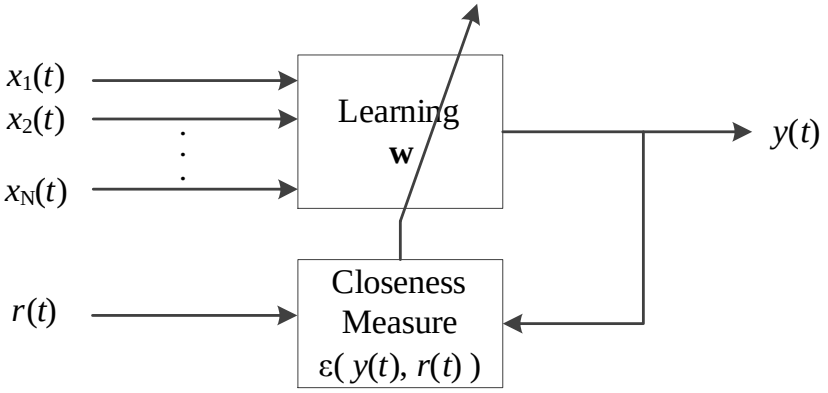

Figure 5.1: Block diagram of the one-unit ICA-R algorithm

where $J(y)$ is a contrast function known as approximated negentropy as proposed by [59], $y = \mathbf{w}^T\mathbf{x}$ represents the extracted signal, $\mathbf{w}$ is the spatial filter, v is a zero mean and unit variance Gaussian random variable [3], $\mathbb{E}\{.\}$ is the expected value of a variable and $G(.)$ can be any convex non-quadratic function (a number of practical functions for $G(.)$ are also suggested by [59]). In practice, $\mathbb{E}\{G(y)\}$ is estimated by the sample mean of $G(y)$ given as $\mathbb{E}\{G(y)\} = \frac{1}{N_t}\sum_{t\in T} G(y(t))$ where N_t is the total number of time samples.

However, when using the above one-unit ICA method, the extracted signal which is maximally independent is not necessarily guaranteed to be the desired source signal [122]. Thus, to guide the extraction, some prior information of the desired signal is required. One of the prior information can be provided is a reference signal $r(t)$. Subsequently, this leads to the one-unit ICA-R [122]. The constrained optimization problem that defines the one-unit ICA-R is formulated as follows:

$$\begin{aligned} \max_{\mathbf{w}} \quad & J(y) = [\mathbb{E}\{G(y)\} - \mathbb{E}\{G(v)\}]^2 \\ \text{subject to} \quad & g(\mathbf{w}) \leq 0, \quad h(\mathbf{w}) = \mathbb{E}\{y^2\} - 1 = 0 \end{aligned} \tag{5.4}$$

where $g(\mathbf{w}) = \varepsilon(y, r) - \xi$, $\varepsilon(y, r)$ is a the closeness measure between y and r while ξ defines the threshold of the closeness measure and $h(\mathbf{w})$ is a constraint that ensures the output y has unit variance.

In [122], the constrained optimization problem (5.4) was solved through the augmented Lagrangian approach, with the inequality constraint $g(\mathbf{w})$ transformed into

[3] A fundamental result in information theory is that a Gaussian random variable has the largest entropy among all random variables of equal variance. Thus, by comparing the extracted signal (or more precisely a random variable y) with the Gaussian random variable v, the degree of negentropy in y can be estimated. In one-unit ICA-R, the spatial filter $\mathbf{w}$ is successfully estimated whenever y achieved maximum negative entropy (or sometimes known as non-gaussianity).

equality constraint through the introduction of a slack variable. The augmented Lagrangian function for (5.4) is given as follows [122]:

$$L(\mathbf{w}, \mu, \lambda) = J(y) - \frac{1}{2\gamma}[\max^2\{\mu + \gamma g(\mathbf{w}), 0\} - \mu^2] - \lambda h(\mathbf{w}) - \frac{1}{2}\gamma\|h(\mathbf{w})\|^2 \quad (5.5)$$

where μ and λ are Lagrange multipliers for the constraints $g(\mathbf{w})$ and $h(\mathbf{w})$, respectively. γ is a scalar penalty parameter. A Newton-like learning algorithm to solve (5.5) can be derived by finding the maximum of the above augmented Lagrangian function as follows [122]:

$$\mathbf{w}_{j+1} = \mathbf{w}_j + \eta \mathbf{R}_{\mathbf{xx}}^{-1} L'_{\mathbf{w}_j} / \delta(\mathbf{w}_j) \quad (5.6)$$

where j denotes the iteration index, η is the learning rate, $\mathbf{R}_{\mathbf{xx}}$ is the covariance matrix of the observed signals $\mathbf{x}(t)$ and

$$L'_{\mathbf{w}_j} = \beta \mathbb{E}\{\mathbf{x}G'(y)\} - \frac{1}{2}\mu \mathbb{E}\{\mathbf{x}g'(\mathbf{w}_j)\} - \lambda \mathbb{E}\{\mathbf{x}y\} \quad (5.7)$$

$$\delta(\mathbf{w}_j) = \beta \mathbb{E}\{G''(y)\} - \frac{1}{2}\mu\{g''(\mathbf{w}_j)\} - \lambda \quad (5.8)$$

where $\beta = \text{sgn}(\mathbb{E}\{G(y)\} - \mathbb{E}\{G(v)\})$, sgn(.) is the signum function, $G'(y)$ and $G''(y)$ are the first and the second derivatives of $G(y)$ with respect to y, and $g'(\mathbf{w}_j)$ and $g''(\mathbf{w}_j)$ are the first and the second derivatives of $g(\mathbf{w}_j)$ with respect to y. The Lagrange multipliers μ and λ are learned by the following gradient ascent methods [122]:

$$\mu_{j+1} = \max\{0, \mu_j + \gamma g(\mathbf{w}_j)\} \quad (5.9)$$

$$\lambda_{j+1} = \lambda_j + \gamma h(\mathbf{w}_j) \quad (5.10)$$

In this work, the following measures are taken when one-unit ICA-R is applied:

- Mean square error (MSE) is chosen as the closeness measure where $\varepsilon(y, r) = \mathbb{E}\{[y - r]^2\}$.
- $G(.) = \log(\cosh(.))$ is selected as the non-quadratic function [59].
- Centering and pre-whitening (see Chapter 3.2.5.4) are applied as a signal preprocessing step throughout this study before applying ICA-R [59]. The reason

for applying pre-whitening is that it reduces the number of parameters to be estimated and can make the algorithm faster[4].

- The scalp distribution, or mixing vector $\mathbf{a}$[5] of the extracted signal is estimated through minimum mean square error as [123][124]:

$$\hat{\mathbf{a}} = \frac{\mathbb{E}\{\mathbf{x}y\}}{\mathbb{E}\{y^2\}} \tag{5.11}$$

5.2.3 Practical issues when applying one-unit ICA-R

It should be noted that threshold ξ is an important parameter in one-unit ICA-R. In general, the value of the threshold ξ depends on the choice of the closeness measure $\varepsilon(y, r)$. In our case where MSE is used, if the threshold ξ is chosen to be too large, there may exist multiple source signals whose closeness measures fulfill the threshold condition. As a result, the ICA-R may not converge to the desired source signal successfully. In contrast, if the threshold ξ is too small, the algorithm may fail to converge. Thus, the threshold should be selected carefully to ensure the one-unit ICA-R works effectively. Unfortunately, to the best of our knowledge, there are no other better option other than trying with different value of ξ [118][122]. In Section XXX, we observe that the threshold $\xi = 10^{-3}$ is suitable in our studies and will be used throughout this chapter.

Another practical issue that needs some attention is the training of the one-unit ICA-R. Similar to other data-driven ERP extraction methods, ICA-based methods require a large amount of training samples to obtain a reliable spatial filter. However, unlike the ENE-LD method in the previous chapter, during training, the one-unit ICA-R requires a large segment of multi-channel EEG signals as the input signal. This large segment of EEG signals can be taken from a segment of the EEG recording or formed by the concatenating trials[6] [64]. Also, for this reason, to train the one-unit ICA-R, a

[4]In [119], a faster variant of ICA-R is proposed by implementing the pre-whitening step. As such, during the ICA-R training, the step such as $\mathbf{R}_{\mathbf{xx}}^{-1}$ [see (5.6)] can be avoided since $\mathbf{R}_{\mathbf{xx}}$ has been whiten to an identity matrix $\mathbf{I}$.

[5]In traditional ICA, under the square mixing assumption (i.e $N_e = N_s$), the resultant $\mathbf{W}$ is an invertible square matrix that also represents the mixing matrix $\mathbf{A}$ when it is inverted. The column vectors of $\mathbf{A}$ potentially convey the topographic information of a source signal. However, unlike traditional ICA, this scalp distribution of the extracted source signal cannot be obtained by inverting $\mathbf{w}$ from the ICA-R.

[6]It should be noted that concatenation of trials from different time is allowed under the linear and instantaneous mixing assumption as long as the mixing process $\mathbf{A}$ remains stationary.

reference signal is also required to be constructed with the same signal length as the input signal.

5.2.4 Generating a new reference signal for ERP application

The reference signal plays a vital role in one-unit ICA-R. It has to be constructed carefully so that it incorporates the appropriate prior information. Previous studies have shown that the reference signal is not required to be exactly similar to the desired signal as long as it carries the important characteristics of the desired signal. One of the common methods to generate the reference signal is to create a synthetic signal that represents the time location of the desired signal. For example, to extract rhythmic activities such as fetal ECG from the maternal ECG, a rectangular pulse train whose frequency is similar to the fetal ECG can be used [119]. Alternatively, if the desired signal is strong and readily observed in a certain EEG channel, a reference signal can be constructed by converting the signal segment that exceeds a given detection threshold into a series of rectangular pulses that represent the time location of the desired signal [117]. Unfortunately, these methods are not applicable for ERPs since ERPs are weak and non-periodic. Therefore, a new way of generating the reference signal are required for ERP application.

5.2.4.1 Fixed reference signal

As described earlier, the major problem in generating a reference signal for the desired ERP is to determine the ERP time region of each individual trial. In practice, these ERP time regions are very difficult to be obtained even by a simple threshold. Thus, one of the more feasible solutions is to estimate the ERP time region from the trial-averaged signal and assume that the desired ERP in all trials reside mainly in the estimated time region.

The first reference signal proposed for our one-unit ICA-R is constructed based on the above idea. Suppose the input signal $\mathbf{x}(t)$ of the one-unit ICA-R is taken from a large segment of the multi-channel EEG recording and may contain target and non-target trials. Let t_k be the stimulus onset of the k-th target trial, and the time interval $[t_{start}, t_{end}]$ is the ERP time region estimated from the trial-averaged target signal. The first reference signal $r_1(t)$ is constructed by generating a series of rectangular pulses

on [t_{start}, t_{end}] after the stimulus onset t_k of each trial:

$$r_1(t) = \begin{cases} 1 & \text{for } t \in \mathcal{T} \\ 0 & \text{otherwise} \end{cases} \tag{5.12}$$

where the set $\mathcal{T} = \bigcup_k [t_k+t_{start},\ t_k+t_{end}]$ represents the overall estimated ERP time region [See Fig. 5.2(a)].

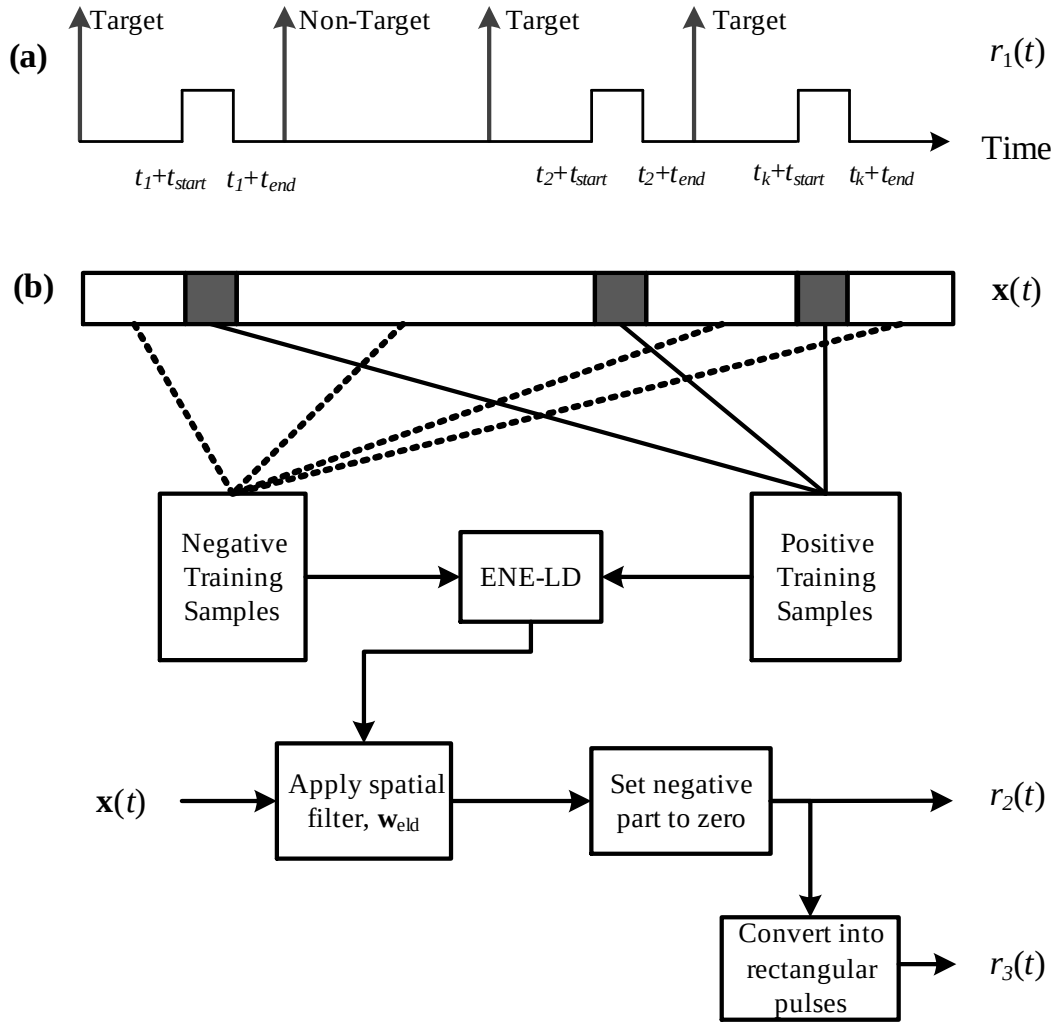

Figure 5.2: Procedure for generating different types of reference signal for ERP extraction. (a) the fixed reference signal $r_1(t)$, (b) the ENE-LD based reference signal $r_2(t)$ and its reshaped version $r_3(t)$.

5.2.4.2 ENE-LD based reference signal

The underlying ERPs are unlikely to be stationary across trials. As a result, the mismatch information in $r_1(t)$ may prevent one-unit ICA-R from extracting the desired source signal accurately. Since the ERPs are non-stationary, a new type of reference

signal is required to identify the trial-to-trial variations in the ERP. In the previous chapter, it is noted that although the ERP extracted by ENE-LD are not exactly similar to the source signal, it is potentially useful for indicating the time location of the desired ERP at the single-trial level. Thus, instead of imposing the strict assumption that ERP time regions are fixed across trials, another approach is to use the ENE-LD extracted signal to reflect the ERP time region at the single-trial level.

Another motivation for using ENE-LD based reference signal in one-unit ICA-R is to combine the strengths of both the one-unit ICA-R and ENE-LD methods. Specifically, as a discriminative-learning method, ENE-LD is good at maximizing the differences between ERP and non-ERP time regions. However, as can be seen in the previous chapter, they are less suitable to recover the original waveform of the source signal. Thus, as a generative-learning method, the strength of one-unit ICA-R for signal reconstruction is utilised to reduce the mismatch between the exact and estimated source signal. As a result, the combination of one-unit ICA-R and ENE-LD is proposed to deal with the respective weaknesses of these methods.

Similar to the procedure used in the previous chapter, ENE-LD is tasked to learn the differences between the scalp distribution of the relevant and irrelevant ERP time regions. As shown in Fig. 5.2(b), the procedure to obtain the ENE-LD based reference signal is as follows :

1. Those multi-channel measurement vectors whose time indices are within the set $\mathcal{T}$, $\mathbf{x}(t)|_{t\in\mathcal{T}}$, are taken as the positive training samples while the remaining measurement vectors are taken as the negative training samples.

2. Based on these training samples, the ENE-LD spatial filter $\mathbf{w}_{\text{eld}}$ and its threshold b is obtained by using the procedure described in Section 4.2.2.1.

3. By applying the ENE-LD spatial filter and its threshold, the second reference signal $r_2(t)$ is given as:

$$r_2(t) = \max(\mathbf{w}_{\text{eld}}^{\top}\mathbf{x}(t) - b, 0) \tag{5.13}$$

5.2.4.3 Binary form of ENE-LD based reference signal

In practice, the rectangular pulse train is often selected for constructing the reference signal. Thus, for comparison purposes, the third reference signal $r_3(t)$ is obtained by

reshaping $r_2(t)$ into rectangular pulses as follows:

$$r_3(t) = \begin{cases} 1 & \text{for } r_2(t) > 0 \\ 0 & \text{otherwise} \end{cases} \tag{5.14}$$

5.3 Simulation Study

5.3.1 Semi-simulated EEG data

In this section, our first objective is to examine whether the one-unit ICA-R can be used to extract the desired ERP directly. After which, our next objective is to understand which reference signal proposed in Section 5.2.4 would be most suitable for the one-unit ICA-R. The ability of ICA-R to recover the desired independent source signal has been verified in many works based on mixtures of pure synthetic waveforms [119][122][117]. However, in this work, a set of semi-simulated EEG datasets are generated to examine the ICA-R extraction performance under noisy EEG background activities. In addition, the study is also designed to mimic the amplitude, latency and width variations as observed in recorded ERPs so that the impact of the three different proposed reference signals on ICA-R can be examined.

Without loss of generality, in this study, special attention is paid to the P300 ERP mainly because it is one of the most extensively studied ERPs and has vast applications in clinical diagnoses and BCI applications [22][127]. As discussed in Chapter 2, the P300 ERP is usually observed as a large positive peak that occurs between 250-500 ms after a stimulus presentation [29] although some research suggest that its latency may range widely from 250 ms to 800 ms [30]. In terms of scalp distribution, the P300 ERP tends to be central-parietal distributed, with peak mostly at the parietal region. The simulation process used to generate the P300 ERP in this study is described as follows.

The overall process for generating the semi-simulated EEG dataset is shown in Fig. 5.3. Mathematically, the linear EEG model for generating the P300 ERP is given as follows:

$$\begin{aligned} \mathbf{x}(t) &= \alpha \mathbf{x}_{\text{sim}}(t) + \mathbf{x}_{\text{EEG}}(t) \\ &= \alpha \mathbf{a}^{\top} s_{\text{sim}}(t) + \mathbf{x}_{\text{EEG}}(t) \end{aligned} \tag{5.15}$$

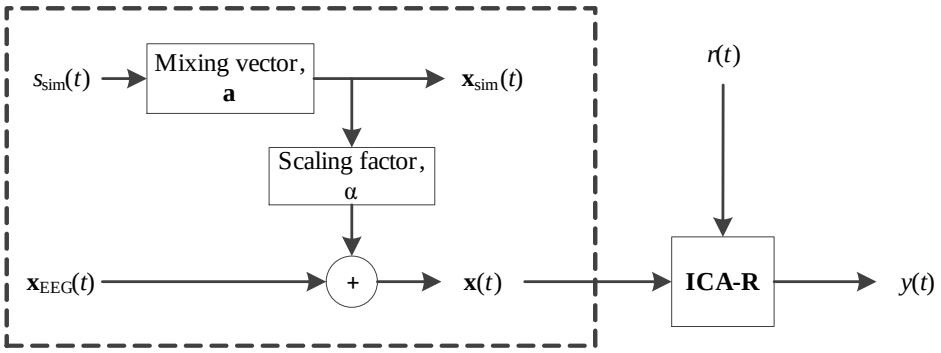

Figure 5.3: An overview of the procedure for the simulation study where the process of generating a semi-simulated multi-channel signals $\mathbf{x}(t)$ is shown in the dotted-line box.

where α is a scaling parameter to adjust the SNR between $\mathbf{x}_{\text{sim}}(t)$ and $\mathbf{x}_{\text{EEG}}(t)$, $\mathbf{a}$ is the mixing vector and $s_{sim}(t)$ is the simulated source signal of the P300 ERP. The above model is similar to the one that is used in the previous chapter except that the irrelevant signals are now replaced by $\mathbf{x}_{\text{EEG}}(t)$, which is taken from a recording of a subject who was instructed to sit and watch a program on television for 40 mins.

Firstly, assuming a sampling frequency of 1 kHz, the source signal $s_{\text{sim}}(t)$ is generated as a series of 100 P300 ERPs, with each ERP appearing in every 800 time samples. In other words, the inter-stimulus interval is assumed to be 800 ms. Secondly, for each P300 ERP, it is generated as a positive Gaussian pulse as shown in Fig. 5.4(b):

$$s_k(t) = c_k \exp\left\{-\frac{(t-\tau_k)^2}{2\delta_k^2}\right\} \tag{5.16}$$

where c_k, τ_k and δ_k are the peak amplitude, latency and width of the k-th ERP respectively. In this study, the width δ_k of each ERP is a random number uniformly distributed between 50 ms and 100 ms; the amplitude c_k of each ERP is uniformly distributed between 0 and 1; while the latency τ_k is normally distributed with a mean latency $\bar{\tau}$ of 350 ms and whose standard deviation σ_τ is set to represent different experimental conditions. In addition, all the generated ERPs were latency-sorted so that the range of latency variation can be observed easily. Finally, the mixing vector $\mathbf{a}$ for the source signal was fixed and generated based on the typical scalp distribution of the P300 ERP as shown in Fig. 5.4(a). Observe that similar to real P300 ERPs, the largest signal in $\mathbf{x}_{\text{sim}}(t)$ will originate from channel Pz.

In this study, $\mathbf{x}_{\text{EEG}}(t)$ were measured with a 40-channel NuAmps EEG system (34 EEG channels, 2 reference channels and 4 EOG channels). As a pre-processing step,

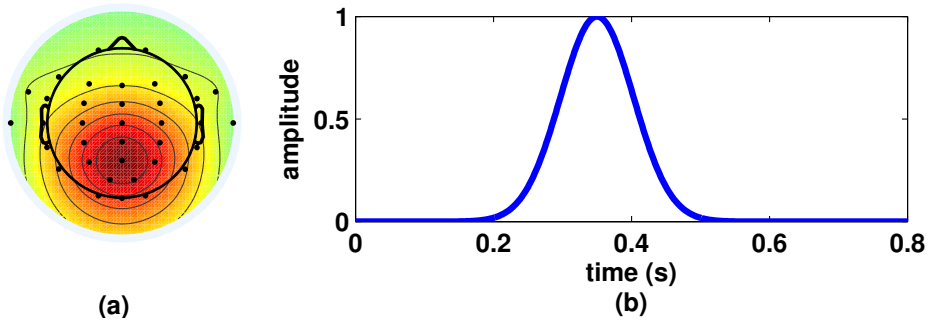

Figure 5.4: (a) The normalised mixing vector **a** and (b) waveform used to generate the P300 ERP in this study

the EEG signals were bandpass filtered[7] between 0.5 Hz - 35 Hz while eye-blinks were removed using the method of [117] by taking the electrode VEOG as a reference. During simulation, each semi-simulated EEG segment $\mathbf{x}(t)$ was generated by adding $\mathbf{x}_{\text{sim}}(t)$ into the real EEG signal $\mathbf{x}_{\text{EEG}}(t)$ where α is adjusted according to the signal-to-noise (SNR) ratio between channel Pz of $\mathbf{x}_{\text{sim}}(t)$ and $\mathbf{x}_{\text{EEG}}(t)$ so that SNR is maintained at 0 dB. For evaluation, the experiment involves testing 11 different values of latency standard deviation ranging from $\sigma_\tau = \pm$ 0 ms to $\sigma_\tau = \pm$ 100 ms, while for each σ_τ, a total of 300 different semi-simulated signals were generated.

To recover the simulated source signal, the one-unit ICA-R was applied to each semi-simulated signal with the three different reference signals $r_1(t)$, $r_2(t)$ and $r_3(t)$. During the training of the one-unit ICA, the estimated ERP time region $\mathcal{T}$ is set between 275 ms and 425 ms after each stimulus onset. The extraction quality of the proposed one-unit ICA-R was examined by measuring the performance index (PI) as defined below [118]:

$$\text{PI} = -10\log_{10} E\{[s_{\text{sim}}(t) - y(t)]^2\} \tag{5.17}$$

To use this metric, both the simulated source signal $s_{\text{sim}}(t)$ and extracted signal $y(t)$ were normalised to be zero mean with unit variance. This metric can be seen as the inverse of the MSE that describes the similarity between the simulated source signal and the extracted signal in decibel (dB). Generally, values larger than 20 dB indicate good extraction quality [118]. In this study, due to its popularity, the Infomax-based ICA [63][64] is selected to represent the traditional ICA and served as a baseline comparison. For this study, the method as implemented in the well-known EEGLAB toolbox

[7] A combination of a forward-backward 2nd order elliptic highpass and 8th order elliptic lowpass filters.

[126] was employed. For traditional ICA, its performance was evaluated by choosing the source signal with the highest PI [8].

5.3.2 Results and discussion

5.3.2.1 Qualitative assessment

Fig. 5.5(a) depicts the simulated source signal $s_{\text{sim}}(t)$ while Fig. 5.5(b) shows the semi-simulated signal from channel Pz of the corresponding signal $\mathbf{x}(t)$. Fig. 5.5(c), (e) and (g) show the reference signals $r_1(t)$, $r_2(t)$ and $r_3(t)$ respectively while their corresponding extracted signals $y(t)$ (solid line) with ICA-R are shown in Fig. 5.5(d), (f) and (h) respectively. In addition, the dotted line in Fig. 5.5(d), (f) and (h) represent the simulated source signal which is normalised to have zero-mean and unit variance. From these figures, it can be observed that one-unit ICA-R is robust against the design of the reference signal since the results show that all reference signals managed to guide the one-unit ICA-R to extract the desired source signal effectively. However, in terms of the amplitude range, the extracted signals are slightly different from the original source signals mainly because of the scaling ambiguity and the zero-mean and unit-variance source signal assumption as mentioned in Section 3.2.5.3. In any event, without loss of generality, the extracted signals closely resemble the normalised source signal as observed in Fig. 5.5(d), (f) and (h). In addition, from these figures, it can be observed that the signal from $r_2(t)$ resembles the normalised source signal the closest. By examining the temporal waveform of each reference signal, it can be seen that the performance difference in extraction is caused mainly by the characteristic of the reference signal as shown in Fig. 5.5(c), (e) and (g).

To examine the characteristic of each reference signal in detail, 2D graphs were plotted in Fig. 5.6 to reveal how each reference signal acts at the single-trial level. By dividing the signals into trials, each plot in Fig. 5.6(a) shows one of the 300 simulated source signals $s_{\text{sim}}(t)$ with varying peak amplitudes, latencies and widths. Meanwhile each plot in Fig. 5.6(b) shows channel Pz from the corresponding semi-simulated signal $\mathbf{x}(t)$. The corresponding responses given by each reference signal were shown

[8] The Infomax-based ICA decomposes the ERP data into a number of source signals equal to the number of EEG channels. As a result, source selection is necessary before the ERP-of-interest can be studied.

in Fig 5.6(c), (d) and (e). In Fig. 5.6(c), it is easily observed that the fixed reference signal $r_1(t)$ does not change with respect to any amplitude and latency fluctuations. In contrast, due to the ability of ENE-LD to learn the differences between ERP and non-ERP time regions, Fig. 5.6(d) and (e) show that both reference signals $r_2(t)$ and $r_3(t)$ managed to pick up the changes in the ERP peak latencies and adapted their waveforms accordingly.

5.3.2.2 Quantitative examination

Fig. 5.7(a) shows the averaged PI scores achieved by the different ICA methods when different latency variations in the ERP are encountered. When comparing the different reference signals, the result shows that the adaptive ability of reference signals $r_2(t)$ and $r_3(t)$ provides the one-unit ICA-R with a better overall performance in extraction.

Table 5.1: Grand average PI score across all level of latency variations

Method	Reference signal	PI (dB)
one-unit ICA-R	$r_1(t)$	14.66 $\pm$ 1.01
	$r_2(t)$	23.33 $\pm$ 0.64
	$r_3(t)$	20.16 $\pm$ 1.42
Infomax ICA		19.63 $\pm$ 0.09

Specifically, as shown in Table 5.1, $r_2(t)$ achieved the best performance, with a grand average PI score of 23.33 $\pm$ 0.64 dB across all level of latency variations, followed by 20.16 $\pm$ 1.42 dB for $r_3(t)$ and 14.66 $\pm$ 1.01 dB for $r_1(t)$. The signal quality extracted by using $r_2(t)$ is also better when compared to the traditional Infomax-based ICA which has a grand average PI score of 19.63 $\pm$ 0.09 dB. For this result, our observations are similar to the finding based on pure simulated signals [125], in which, the one-unit ICA-R may outperform the traditional ICA when the reference signal are designed correctly. Interestingly, a comparison between $r_2(t)$ and $r_3(t)$ also shows that the conventional approach of using rectangular pulses in the reference signal $r_3(t)$ does not guarantee a better extraction performance. From the analysis, we conclude that it may be important to include the amplitude variation of the ERPs in the design of the reference signal.

To understand the computational efficiency of ICA-R with different reference signals, the convergence time required by each combination of ICA-R was measured and tabulated in Fig. 5.7(b). For this test, a CPU with an Intel Core 2 Quad Q9650 3 GHz processor and 4 Gb memory RAM was used as the test bench. For a simulated data that contains 34 channels and 80,000 time samples, all three variants of the proposed one-unit ICA-R only takes an average of less than 2 seconds to converge and complete the extraction. Among the reference signals, the ICA-R that uses $r_2(t)$ is the most efficient with an average convergence time of 0.30 ± 0.16 s, followed by 1.17 ± 0.03 s for $r_3(t)$ and 1.35 ± 0.12 s for $r_1(t)$. In contrast, Infomax-based ICA as implemented in the popular EEGLAB toolbox [126] took an average of 96.75 ± 5.95 s to complete the full ICA decomposition. From these empirical results, one can conclude that a reference signal that contains more reliable information on the desired signal such as $r_2(t)$ will help ICA-R to converge quicker and also extract better source signals. In addition, compared to the traditional ICA that performs full source extraction, one-unit ICA-R requires far lesser amount of time since the extraction only involves one source signal.

5.3.2.3 Robustness of the proposed reference signal

In Fig. 5.7(a), the results show that compared to the traditional ICA, the performance of all the one-unit ICA-R decrease when the latency variation of the ERPs increases. This indicates that apart from the choice of reference signals, another factor that affects the performance of the one unit ICA-R is the overall estimated time region $\mathcal{T}$. In practice, there will be situations when the time region of a specific trial is not covered well by $\mathcal{T}$. When this situation occurs, it introduces mismatch between $r_1(t)$ and the original source signal, and subsequently reduces the performance of the corresponding one-unit ICA-R. Likewise, for $r_2(t)$ and $r_3(t)$, when $\mathcal{T}$ introduces the mislabeled training samples (i.e. the positive training samples are not taken from the correct ERP time region of each respective trial and vice versa), the performance of the one-unit ICA-R is also reduced. However, interestingly, from Fig. 5.7(a), a closer examination reveals that among all three reference signals, the performance of $r_2(t)$ is less affected by the mismatch, with only $\sim$2 dB difference between PI at σ_τ=0 ms and σ_τ=100 ms. This is mainly because as can be seen by comparing Fig. 5.6(a) and (d), the reference signal $r_2(t)$ is more robust and is able to adapt to the amplitude, width and latency variations

of the underlying ERPs.

5.3.2.4 Effect of parameter ξ on one-unit ICA-R

In this section, the above simulation study is repeated by using different values of ξ for one-unit ICA-R with $r_2(t)$. Fig. 5.8(a) and (b) shows that one-unit ICA-R achieved similar PI scores and convergence times under different values of ξ. A closer examination reveals that smaller ξ value allows one-unit ICA-R to converge faster at larger latency variations and also achieve a slightly better PI score. This result is probably because at larger latency variations, a lower threshold (i.e ξ=0.1) allows one-unit ICA-R to accommodate better the mismatch that appeared in the designated reference signal. However, comparison between Fig. 5.7(a) and 5.8(a) suggests that the overall effect of ξ is insignificant when compared to the design of reference signal.

5.3.3 Summary

In this section, a simulation study was conducted based on semi-simulated EEG datasets where the simulation attempts to encompass the possibility of amplitude, latency and width variations in the ERP-of-interest at the single-trial level. In practice, the actual time region of the desired ERP is difficult to identify at the single-trial level and the only estimates that can be possibly obtained is at the trial-averaged level. As a result, the ability to design a representative reference signal based on this limited information becomes an important factor that decides whether the proposed one-unit ICA-R can be potentially used for ERP extraction. For this reason, three types of reference signals are proposed. A rectangular pulse train which assumes that the time region and amplitude of the ERP-of-interest are invariant across trials was used as the first reference signal. The second reference signal was taken from the signal extracted by ENE-LD method. The third reference signal is the binary version of the second reference signal.

Overall, our empirical results show that it is possible to extract the desired ERP correctly using the one-unit ICA-R. As expected, since the extraction does not involve full signal decomposition, the one-unit ICA-R is more efficient in extraction compared to the traditional ICA method. However, trial-to-trial variations in the amplitude, width

and latency remain a potential issue for the one-unit ICA-R. These variations will inevitably degrade the performance of the one-unit ICA-R depending on the level of mismatch between the reference signal and the actual source signal. To address this issue, our empirical results show that the ENE-LD based reference signal, that is $r_2(t)$, is a feasible solution since it is able to reflect well the changes in the amplitude, latency and waveform variations of the underlying ERPs across trials. We have also demonstrated that the performance of the one-unit ICA-R with $r_2(t)$ is comparable to the existing Infomax-based ICA. Thus, for the rest of this chapter, the ENE-LD based ICA-R is evaluated as the suitable method for extracting the P300 ERP from the P300 speller and oddball paradigm dataset. For convenience, in the following section, the proposed ENE-LD based one-unit ICA-R will be referred as one-unit ICA-R.

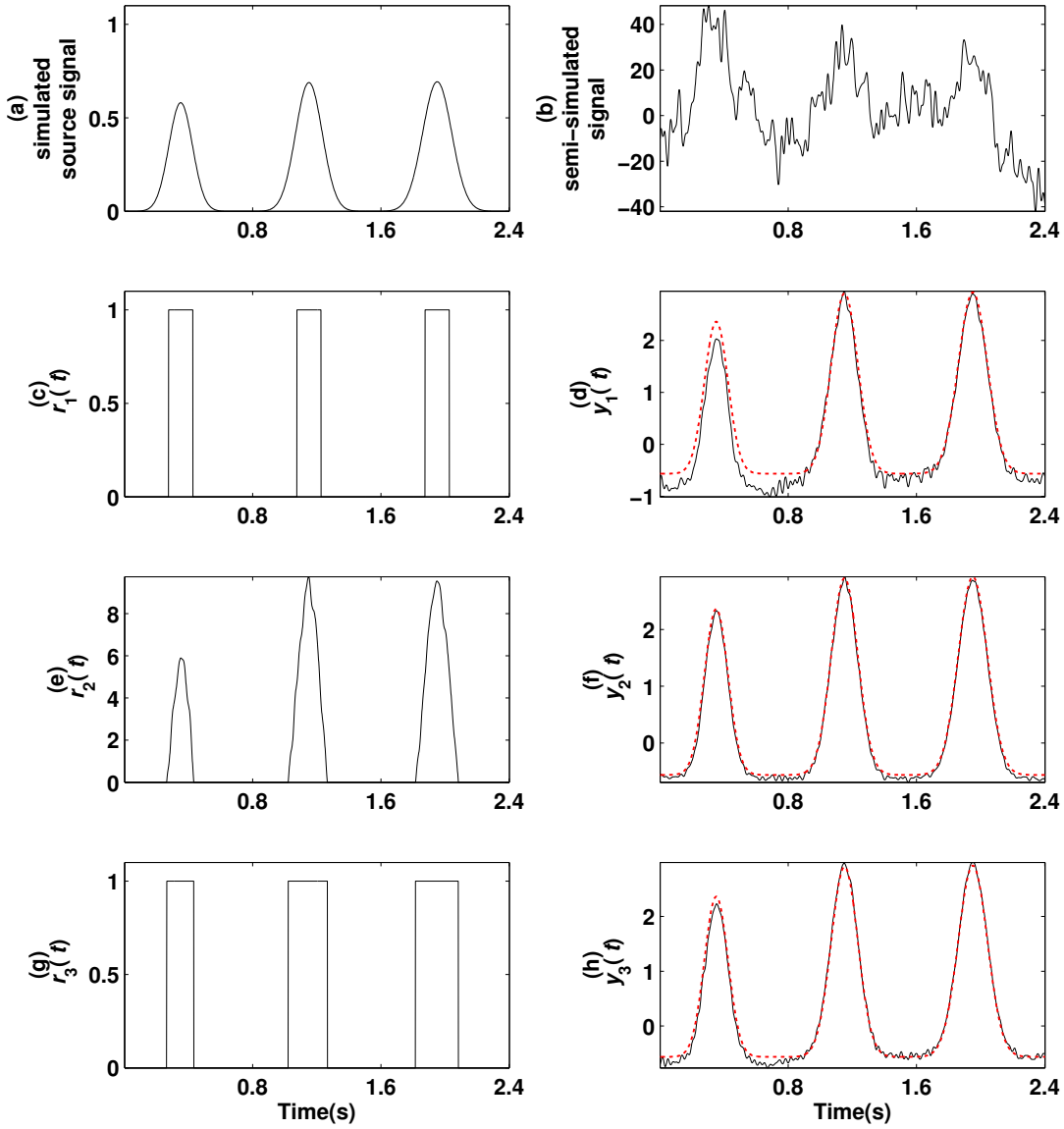

Figure 5.5: (a) A segment of source signal $s_{\text{sim}}(t)$ and (b) semi-simulated signal from Channel Pz of the corresponding semi-simulated signals $\mathbf{x}(t)$; (c), (e) and (g) are the reference signals that were used to recover the source signal while the resultant signals $y(t)$ extracted from ICA-R are plotted as solid lines in (d), (f) and (h). For comparison, the normalised source signal is also plotted as dotted lines in (d), (f) and (h).

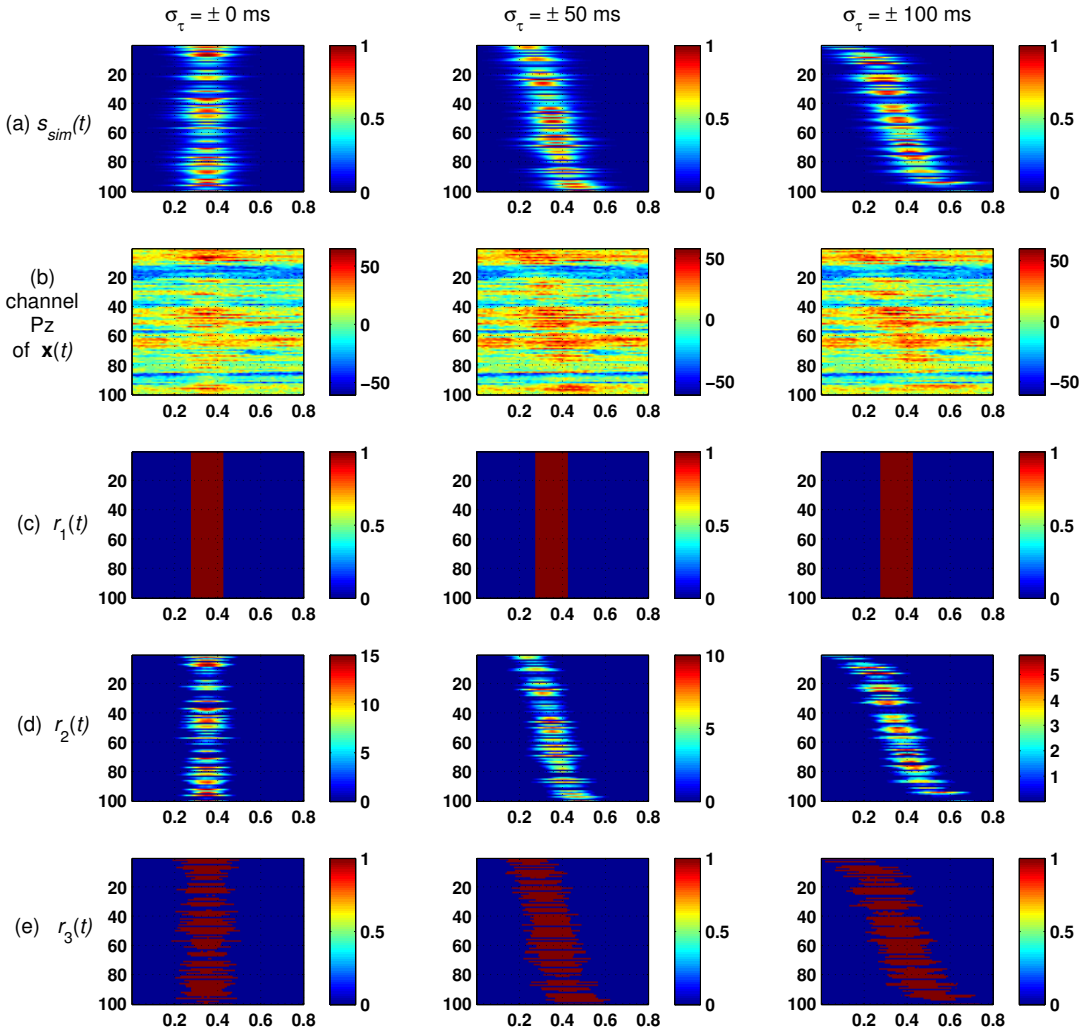

Figure 5.6: Visualization of different signals at single-trial level. (a) The examples of simulated ERPs with varying peak amplitude, width and also different level of latency variations σ_τ (plotted in different columns); (b) noisy 'mixed' signals from Channel Pz of the corresponding semi-simulated signals $\mathbf{x}(t)$; (c)-(e) The responses given by the reference signal $r_1(t)$, $r_2(t)$ and $r_3(t)$ respectively. For all plot, the y-axis represents the trial number and the x-axis gives the time in second.

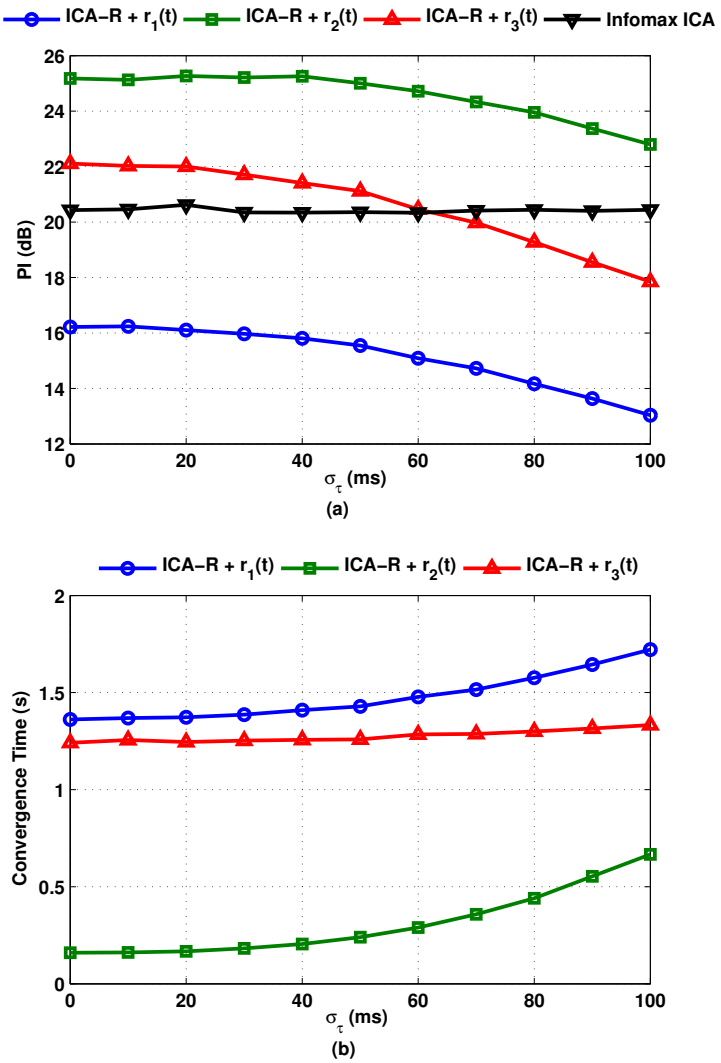

Figure 5.7: (a) Performance Index PI and (b) the convergence time of ICA-R when different reference signals were used.

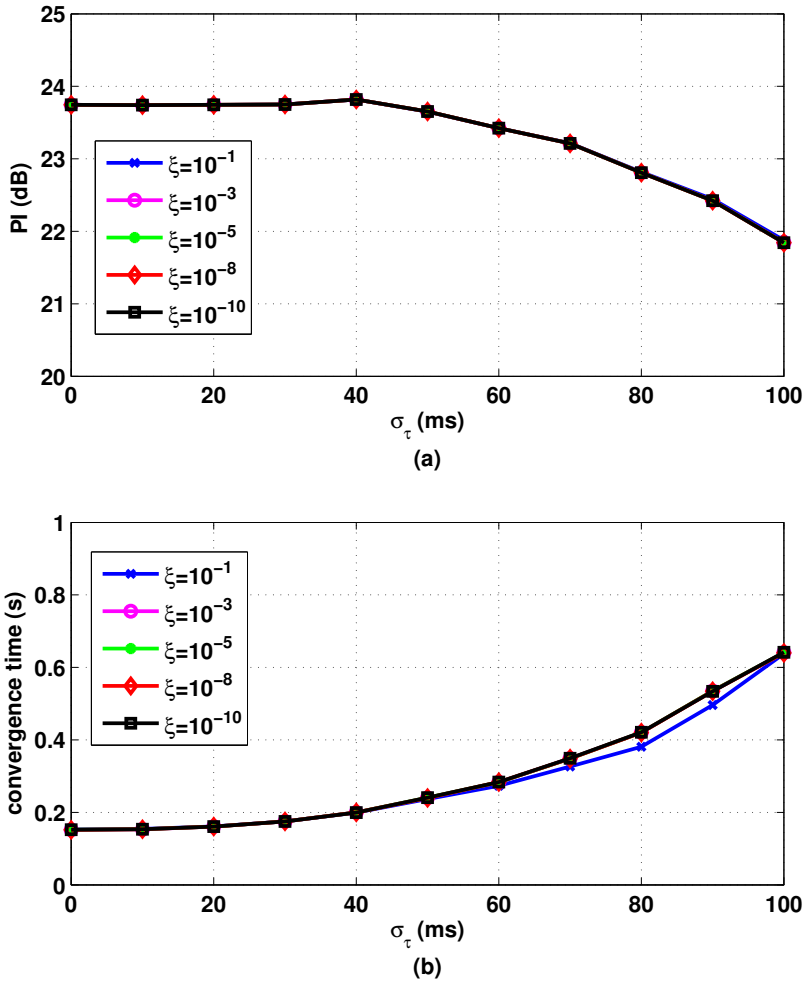

Figure 5.8: (a) Performance Index PI and (b) the convergence time of ICA-R with $r_2(t)$ when different value of ξ were used as threshold.

5.4 P300 Speller - BCI Competition 2003 Dataset IIB

5.4.1 Overview

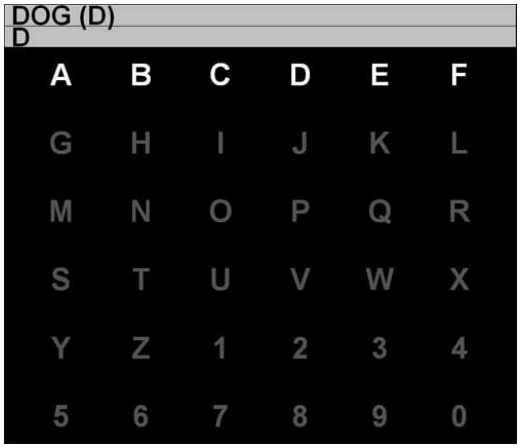

Figure 5.9: Traditional 6 $\times$ 6 character matrix used in the P300 speller from the BCI competition 2003 dataset IIb.

The P300 speller is one of the important application of P300. It is a type of Brain Computer Interface (BCI) that aims to assist patients with locked-in syndrome to utilise their P300 ERPs for spelling words. A traditional P300 speller paradigm usually consists of a predefined set of alphanumeric characters arranged in a 6 $\times$ 6 matrix and presented in front of the user as shown in Fig. 5.9. Each row/column of characters is highlighted in a random order, resulting up to 12 stimulation in each presentation run. To use this speller, the user has to concentrate on a desired character during the presentation. When the desired character is highlighted, a P300 is elicited. Thus, the task of the P300 speller is to identify the row and column that elicits the P300 (See Appendix B.1 for more details on how the P300 speller works).

ICA is one of the earliest feature extraction techniques used in the P300 speller [113]. Unfortunately, compared to other ERP extraction methods, the implementation of ICA in the P300 speller remains limited mainly because its source selection stage often requires human intervention [113][114][128]. Beside being impractical for real-time applications, the task of selecting the independent source signal-of-interest is often heuristic and subjective [129]. To address this problem, different constrained ICA methods which attempt to utilise prior information such as the spatial pattern [130] and the time course of the desired ERP [121] have been proposed to assist the source selection process. Among these methods, the ICA-R with multiple reference

(ICA-MR) method proposed by [121] is very attractive because of its ability to avoid a full ICA decomposition which can be time-consuming. However, ICA-MR by [121] remains ineffecient in extracting the P300 ERP because it still requires a set of source signals to be extracted before the P300 can be selected.

Thus, different to the approach by [121], we propose to apply our one-unit ICA-R to extract the P300 ERP directly. In this study, to evaluate the performance of our proposed one-unit ICA-R, P300 speller dataset from BCI Competition II [127] was selected. Our objectives in this study are:

1. To evaluate whether the one-unit ICA-R can be applied to extract the P300 ERP directly without any human intervention.

2. To examine the prediction accuracy of using the signals extracted by one-unit ICA-R and compare it with the existing ICA approaches.

3. To compare the computational efficiency of one-unit ICA-R against the existing ICA approaches.

5.4.2 Procedure

5.4.2.1 Data Preparation

For the selected dataset, the EEG recordings were collected from one subject using 64 EEG electrodes with a sampling frequency of 240 Hz. During the experiment, the subject is instructed to concentrate on a series of characters. For each character, a total of 15 presentation runs were performed and thus, a total of 180 epochs were collected (30 target and 150 non-target epochs). During the presentation, an inter-stimulus interval of 175 ms was employed where each row/column was highlighted for 100 ms, followed by an idle period of 75 ms.

In this work, to allow comparisons with other existing ICA algorithms as reported by [113] and [121], the dataset was divided into three different sets: training, *Test1* and *Test2* respectively. The number of characters contained in each set is tabulated in Table 5.2. (See Appendix B.2 on how training and testing sets were generated for this study).

Table 5.2: The total number of characters that are assigned for training and testing in this work

Set	Number of Characters
Training	6
Test1	36
Test2	31

5.4.2.2 Training phase

Fig. 5.10 shows the overall process used in the P300 speller dataset to extract the P300. Firstly, all the EEG recordings from different sets were bandpass-filtered[9] between 2-10 Hz. During the ICA-R training, EEG recording from the training set was centered and pre-whitened before the one-unit ICA-R was applied to estimate the spatial filter, $\mathbf{w}$. As shown in Fig. 5.11, to generate the ENE-LD based reference signal, the time window between 250 ms and 400 ms after each target stimulation onset was defined as the positive ERP time regions. During training, the extracted signal was segmented into EEG epochs of 525 ms starting from stimulus onset. For classifier training, these EEG epochs are taken as the training samples (126 features) and used to train a linear SVM classifier [152]. Specifically, those EEG epochs from the target row and column stimulations are taken as positive training samples while the remaining epochs are taken as negative training samples.

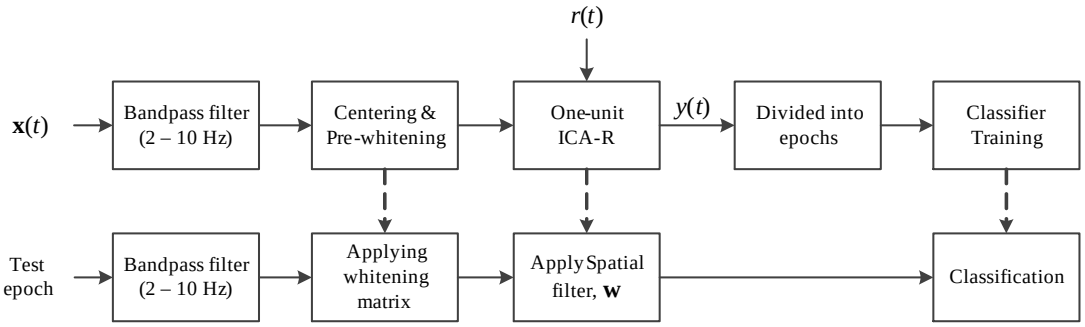

Figure 5.10: Block diagram of the training and testing phase of the one-unit ICA-R specifically for the P300-BCI application.

5.4.2.3 Testing phase

To extract the P300s in the testing set, each test epoch was applied with the whitening matrix and spatial filter that were obtained in the training phase. By applying the

[9]A combination of a forward-backward 2nd order elliptic highpass and 8th order elliptic lowpass filters.

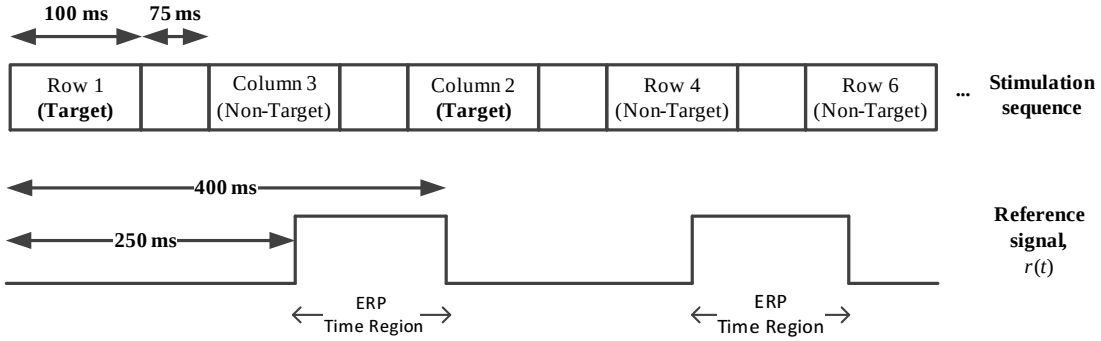

Figure 5.11: The procedure for defining the ERP time region for the generation of ENE-LD based reference signal for the given P300-BCI dataset.

classifier on the newly obtained epochs, the row and column epoch with the highest accumulated classifier score after each run were selected to predict the character [98]. It should be noted that predicting a character within a single run remains difficult, which is why the evaluation is normally performed by examining the prediction accuracy by slowly including more epochs from different presentation runs.

5.4.3 Results and discussion

5.4.3.1 Qualitative assessment

In this section, before examining the classification performance of the signal extracted from the proposed one-unit ICA-R, its extraction performance was first evaluated qualitatively with the following two criteria: 1) Is the proposed ICA-R able to extract the desired P300? and 2) Is the whitening and spatial filter from the training phase valid for extracting P300 from the unseen data during the testing phase?

To answer these questions, the scalp distribution of the signals extracted from all the training and testing sets were estimated using (5.11) in Chapter 4. After that, these extracted signals were separated into target and non-target epochs before trial-averaging was performed to obtain the trial-averaged target and non-target signals.

Firstly, from the trial-averaged signals in Fig. 5.12, it can be observed that the extracted signals exhibit the expected P300 characteristics where only the trial-averaged target signals contains positive peaks around 330 ms. In contrast, the trial-averaged non-target signals do not show any significant activity at around 330 ms. Secondly, from Fig. 5.13(a)–(c), the normalised scalp distribution of the extracted ICs from these

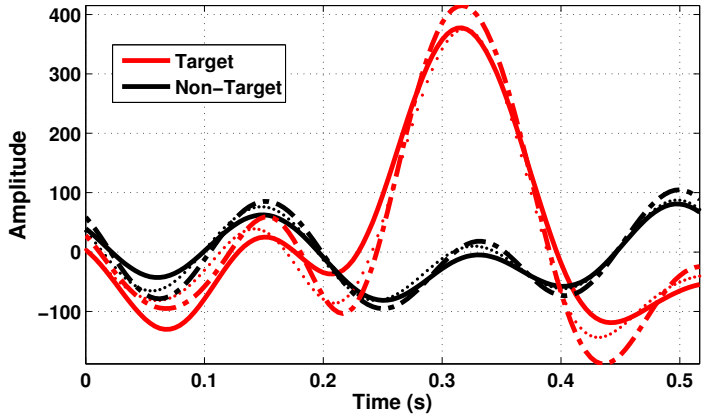

Figure 5.12: Trial-averaged target and non-target signals segmented from the extracted signals $y(t)$ of Training data (Solid line), *Test1* (Dotted line) and *Test2* (Dash-Dotted line) respectively when ICA-R with the proposed reference signal $r_2(t)$ was applied.

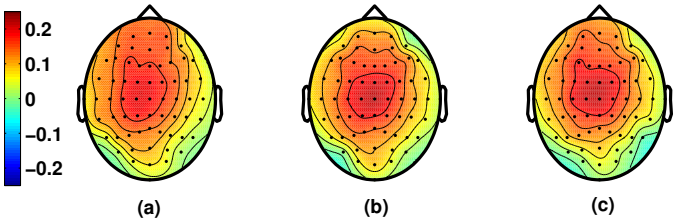

Figure 5.13: normalised scalp distributions represented by the extracted signals $y(t)$ of (a) Training data, (b) *Test1* and (c) *Test2* respectively when ICA-R was applied with the proposed reference signal $r_2(t)$.

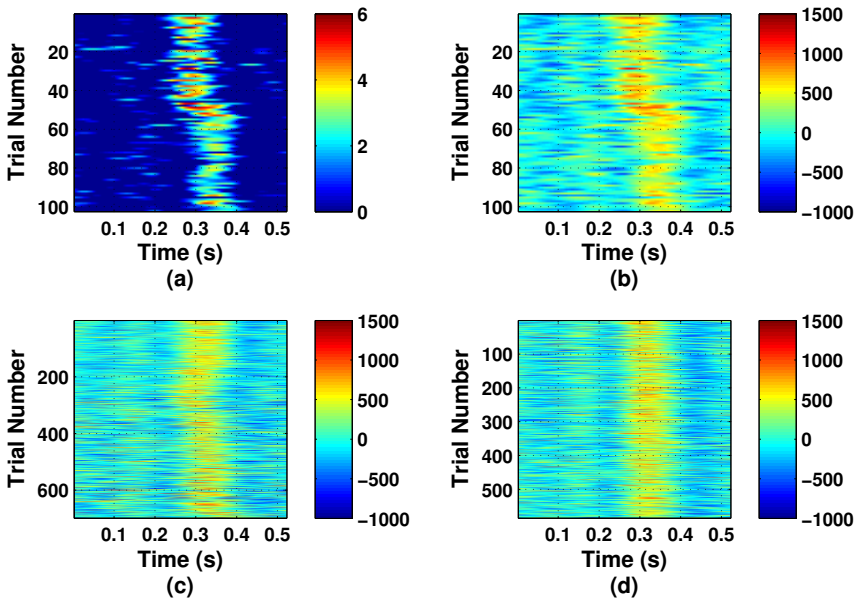

Figure 5.14: Single-trial target responses extracted from: (a) ENE-LD reference signal on Training data, (b)–(d) one-unit ICA-R from Training data, *Test1* and *Test2* respectively. (Note: only target row/column epochs whose previous and upcoming two stimulations belonging to non-target were used.)

sets also matches with the spatial characteristics of the P300 ERP as reported in other work [131]. Thirdly, in Fig. 5.14(b)–(d), by plotting those target row/column epochs whose previous and upcoming two stimulations are non-target, the result at the single-trial level reveals that the extracted signals contain the P300 ERP almost consistently around 300–400 ms. On a different note, it is also worth mentioning that the proposed one-unit ICA-R is able to extract P300 correctly partly due to the ability of the ENE-LD reference signal to capture the time region of the P300 beforehand as shown in Fig. 5.14(a). Lastly, the similarity between the trial-averaged signals and scalp distribution across all training and testing sets suggests that the P300 is stable in terms of its source location. Thus, it is viable to use the spatial filter from the training phase to extract the P300 in the subsequent testing session. Overall, the above results show that the signals extracted from all training and testing sets are similar in terms of their trial-averaged signals, scalp distributions and even signals at single-trial level.

5.4.3.2 Comparison with the existing ICA methods

Table 5.3 shows the prediction accuracy achieved by the ICA-R and ICA-MR at 5, 10 and 15 runs when all 64 EEG channels were used for ERP extraction and only six characters were used for training. The results show that the proposed one-unit ICA-R has comparable or better results when compared to the ICA-MR as reported in [121].

Table 5.3: Prediction accuracy on *Test1* when 64 EEG channels were used for extraction while six characters were used for training.

Method	Accuracy (%)		
	5 runs	10 runs	15 runs
ICA-R	94	97	97
ICA-MR	90	96	97

Next, when examining the ICA-R performance on *Test2* in Fig. 5.15, our result shows that similar to the traditional ICA-based method used by the BCI competition winner, the proposed one-unit ICA-R also uses eight runs to reach perfect prediction. Besides that, the method also nearly achieves perfect prediction at 5 runs with only one mistake. From these results, it can be seen that although only one reference signal was used to extract the P300 ERP, the proposed one-unit ICA-R method has a comparable performance to the existing ICA methods found in the literature.

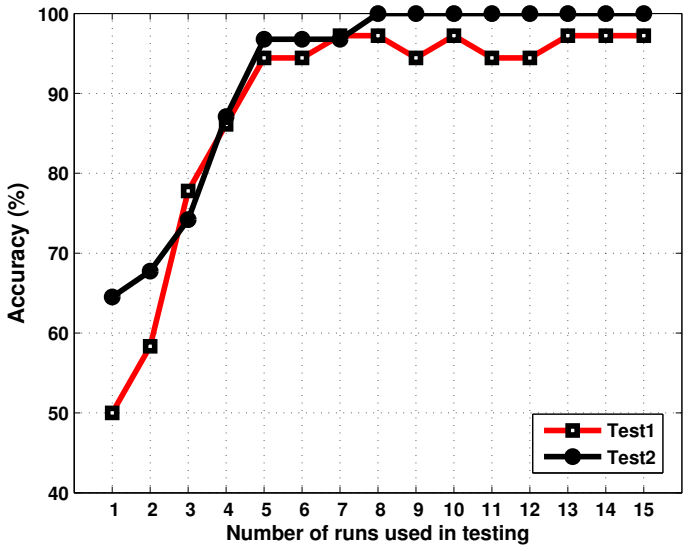

Figure 5.15: Prediction accuracy of ICA-R on *Test1*(36 characters) and *Test2*(31 characters) of the BCI competition 2003 dataset when all 64 EEG channels were used for extraction.

Table 5.4: Computational time spent by ICA-R, ICA-MR and the traditional Infomax-based ICA when used to perform the P300 extraction on a 64×32100 dimension EEG signals.

Method	Computational Time (s)
ICA-R	0.54 ± 0.03
ICA-MR	3.02 ± 0.05
Infomax-ICA	58.06 ± 13.00

In terms of the computational time required by each ICA method, a further test is performed by running each method on a segment of continuous EEG signals that is taken from the four characters 'A' in *Test1*. This unique step is mainly for ICA-MR because during extraction, ICA-MR uses the 12 reference signals to represent the row/column stimulation. Thus, changes in the target row/column stimulation is not allowed so that the extraction can be performed successfully. The resulting EEG segment used for this test consists of 64 channels and 32,110 time samples. The test is conducted using an Intel Xeon E3-1230 processor running at 3.2 GHz with 8 Gb memory ram. The computational time spent by each method for P300 extraction after 50 iterations are shown in Table 5.4. It can be clearly seen that the computational time increases whenever the partial and full ICA decomposition step are required by the ICA-MR and the traditional ICA-based method. In contrast, since only one source signal is extracted, the proposed one-unit ICA-R is the most computation efficient among these three methods, taking an average of 0.54 s to complete the extraction.

5.4.4 Summary

In this section, the proposed one-unit ICA-R has been implemented on a P300-BCI dataset for ERP extraction. By using the extracted signal for prediction, the prediction accuracy of the proposed one-unit ICA-R relative to the existing ICA methods suggests that the proposed one-unit ICA-R is able to perform the extraction reasonably well. Besides being more computationally efficient, one-unit ICA-R also avoids any post source selection stage that often requires human intervention. Thus, its fast and reliable extraction makes one-unit ICA-R a more practical tool for online training and prediction in P300-BCI. In addition, the ability of the method to visualize signals in the single-trial level also makes it a potential tool for understanding the characteristic of P300 across trials not only in the P300 speller, but also other ERP studies which will be shown next.

5.5 P300 - Oddball Paradigm

5.5.1 Overview

In this section, the application of the proposed one-unit ICA-R was further evaluated on a larger dataset to understand its effectiveness when ERPs from different subjects were encountered. For this reason, the publicly available P300 dataset contributed by Ting *et al.* [133][134] was selected for the study. This dataset was originally collected for a study that utilises P300 to provide a quantitative assessment between 2D and 3D visualization modalities. Different from the BCI application, this dataset was collected using the traditional oddball paradigm. With longer inter-stimulus interval, the P300 ERPs observed in this kind of experiment is less affected by the flashing frequency. The characteristic of P300 generated from the traditional oddball paradigm is also well documented in [29]. Unlike P300-BCI, P300 in the oddball paradigm usually occurs together with other early ERPs such as N1, P200 and N200 [29]. Thus, in order to study the contribution of P300 in the given EEG signals, our goal for applying ICA-R is to separate the P300 from other ERPs.

5.5.2 Dataset

The procedure used by [133] for collecting this dataset is described as follows. During the presentation, each participant was randomly presented with the image of a cube and sphere for 100 ms, followed by a black screen for 1000 ms. The number of cube and sphere used in this experiment were 320 and 80 respectively. In addition, the cube was treated as non-target while the sphere was treated as target. Whenever a sphere was presented, the participant was required to respond by pressing a key on a keyboard. To understand the impact of stereoscopic vision on the P300, the stimulus presentation was carried out under both 2D and 3D visualization modalities. Furthermore, the task was also made increasingly difficult by using the same stimulus but under three levels of occlusion, which were quantified in terms of percentage occluded (0%, 30% and 70% respectively). As a result, a total of six different EEG sessions were conducted in a random order for each participant. These sessions will be identified as 2d00, 2d30, 2d70, 3d00, 3d30 and 3d70 respectively in this work. For this dataset, all EEG measurements were recorded using a 40-channel Nuamps system with a sampling frequency of 1000 Hz. The stimuli were presented with a True3Di display system.

5.5.3 Procedure

5.5.3.1 Data pre-processing

For each EEG session, the EEG segment from 0.2 s before the first stimulus to 1 s after the last stimulus was extracted for the ICA-R. This segment of EEG signals contains all the target and non-target trials. Thus, our objective in this section is to apply the proposed one-unit ICA-R to extract a source signal from this EEG segment such that after dividing into epochs, the source signal should only contain P300 in the target trial and no P300 activity for the non-target trials.

The data preprocessing step is as follows. First, the EEG segment was referenced to the average of channel A1 and A2 before being bandpass-filtered[10] between 0.5 - 30 Hz. Since the subjects were not restricted from blinking their eyes, an artifact removal technique described in [117] was applied. Similar to [117], the average of channel Fp1 and Fp2 was taken as the reference since these channels were nearest to the eyes. In

[10]A combination of a forward-backward 2nd order elliptic highpass and 8th order elliptic lowpass filters.

addition, any problematic EEG channels were also removed from the analysis.

5.5.3.2 Reference signal and ICA-R

In each session, the P300 time region was identified by using a data-driven time selection method that will be explained in the next chapter. The identified time region was used to represent the ERP time region of every target trial so that the ENE-LD based reference signal can be constructed. After generating the reference signal, the proposed one-unit ICA-R was applied on the EEG segment to extract a source signal. For analysis, the resultant extracted signal was divided into single-trial target and non-target epochs, with each epoch contains a 0–1000 ms EEG segment after stimulus onset. Here, the target group represents the correctly identified target trials. Baseline correction was performed using a 200 ms time segment prior to the stimulus onset. For comparison, the traditional ICA method is applied on the given EEG segment whereby the source signal that is relevant to the P300 is manually selected for comparison.

5.5.4 Results and discussion

5.5.4.1 Qualitative examination

Isolating P300 from other ERPs is very important for P300 studies to ensure that the individual contribution of P300 towards the EEG signals can be examined carefully at the single-trial level. To understand the efficacy of our proposed ENE-LD based ICA-R, evaluation was conducted by examining how well the source signal extracted from ICA-R resembles P300 but not other ERPs in terms of scalp distribution, waveform and time region.

Fig. 5.16 shows the trial-averaged target and non-target signals from Channel Pz of each EEG sessions. In this figure, we can observe that most subjects have strong P300 activity at around 300–600 ms with its peak exceeding 10 μV except for subjects 2, 3 and 6. For most subjects, the early ERPs that elicits before 300 ms can also be clearly observed in each session.

Fig. 5.17 shows the trial-averaged target and non-target signals from the source signal extracted by the proposed one-unit ICA-R. Before analysing this figure, it is worth mentioning that as described in Chapter 3, due to the scaling ambiguity, all ICA

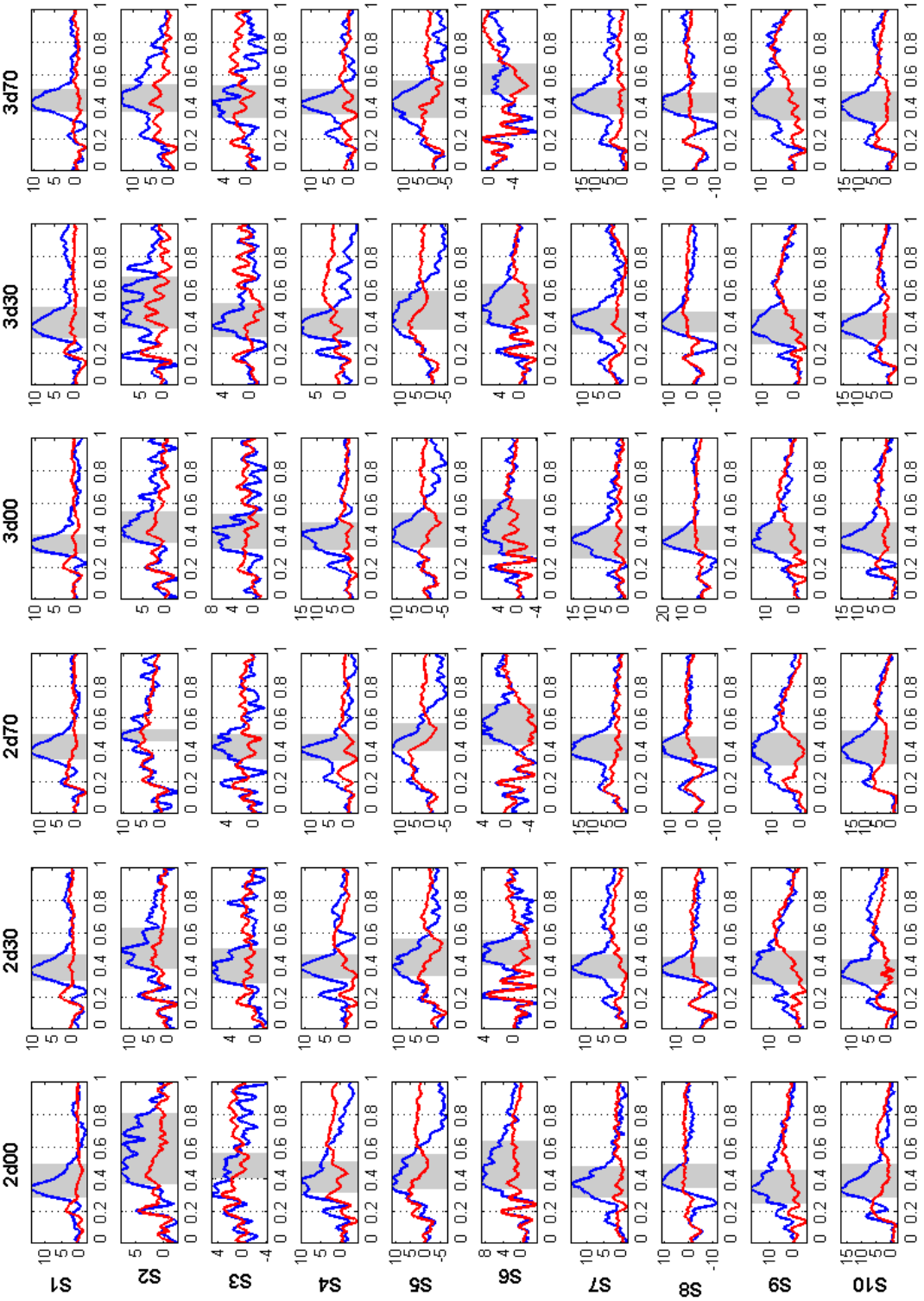

Figure 5.16: Trial-averaged target (blue) and non-target (red) signals from channel Pz of the raw EEG signals from each session. (Note: In each plot, the y-axis represents the signal amplitude in micro-volt while x-axis represents the time in second. In addition, the grey region represents the positive training window for generating the ENE-LD based reference signal.)

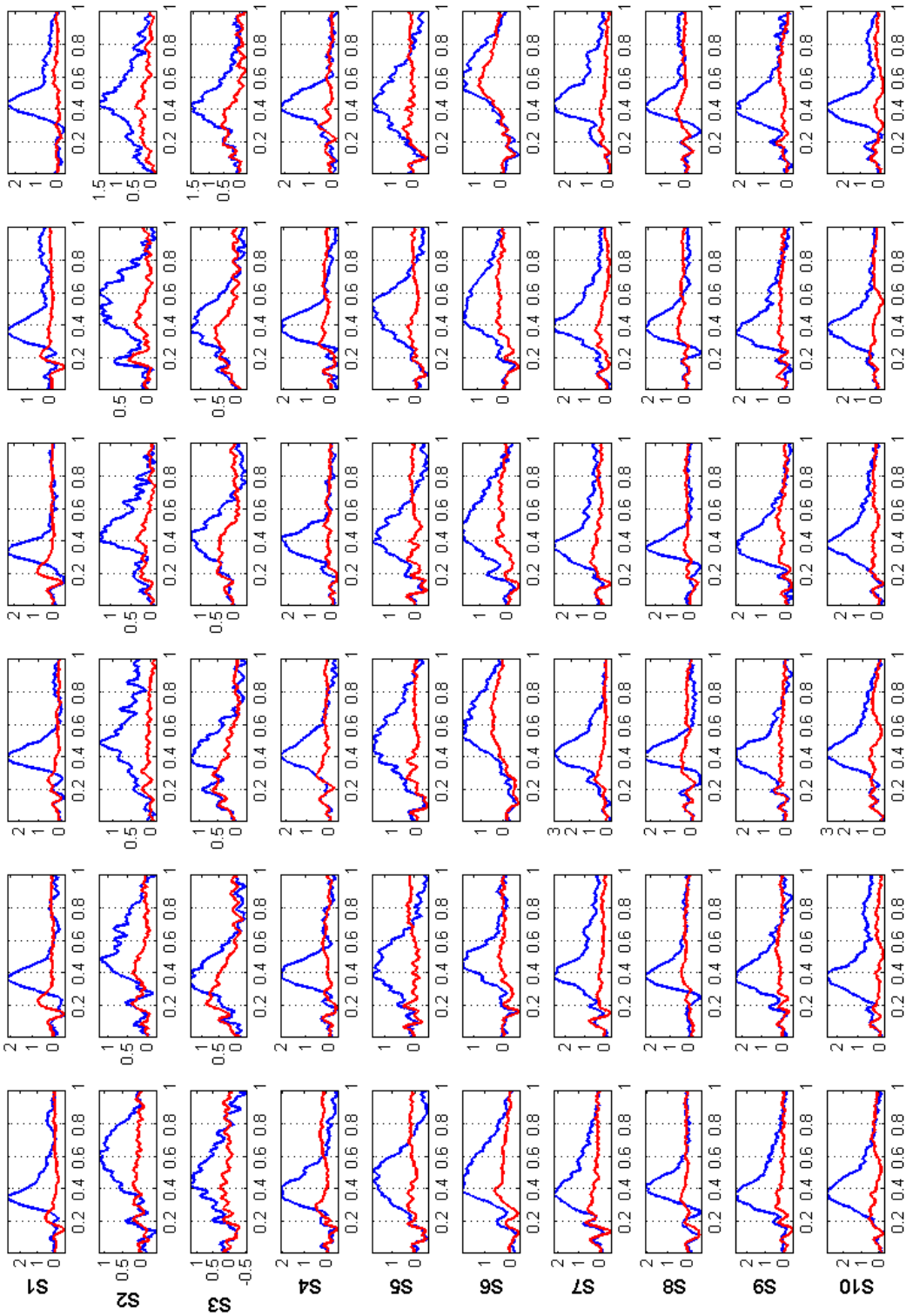

Figure 5.17: Trial-averaged target (blue) and non-target (red) signals segmented from the source signal extracted by our proposed one-unit ICA-R. (Note: In each plot, due to the nature of ICA and normalisation effect, the scale in y-axis may not be the actual signal scale. The x-axis represents the time in second.)

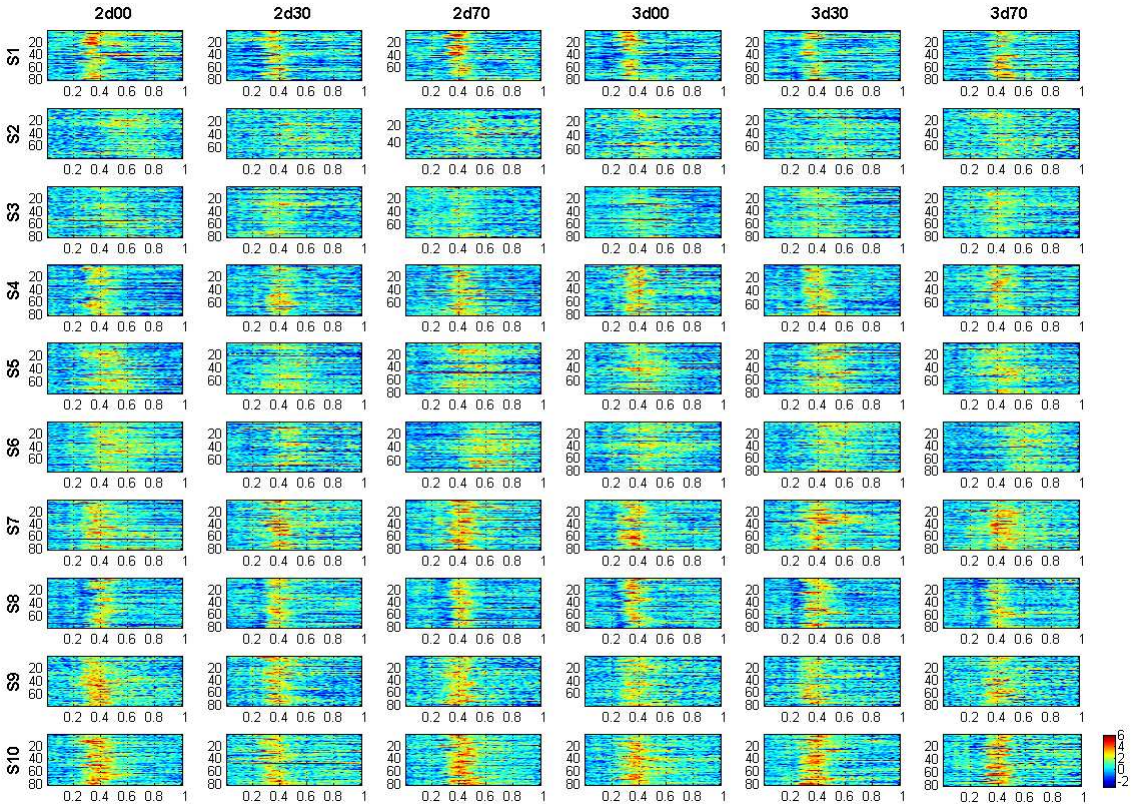

Figure 5.18: Visualization of target epochs segmented from ICA-R of different EEG sessions at single-trial level where y-axis and x-axis represents the trial number and time in second respectively.

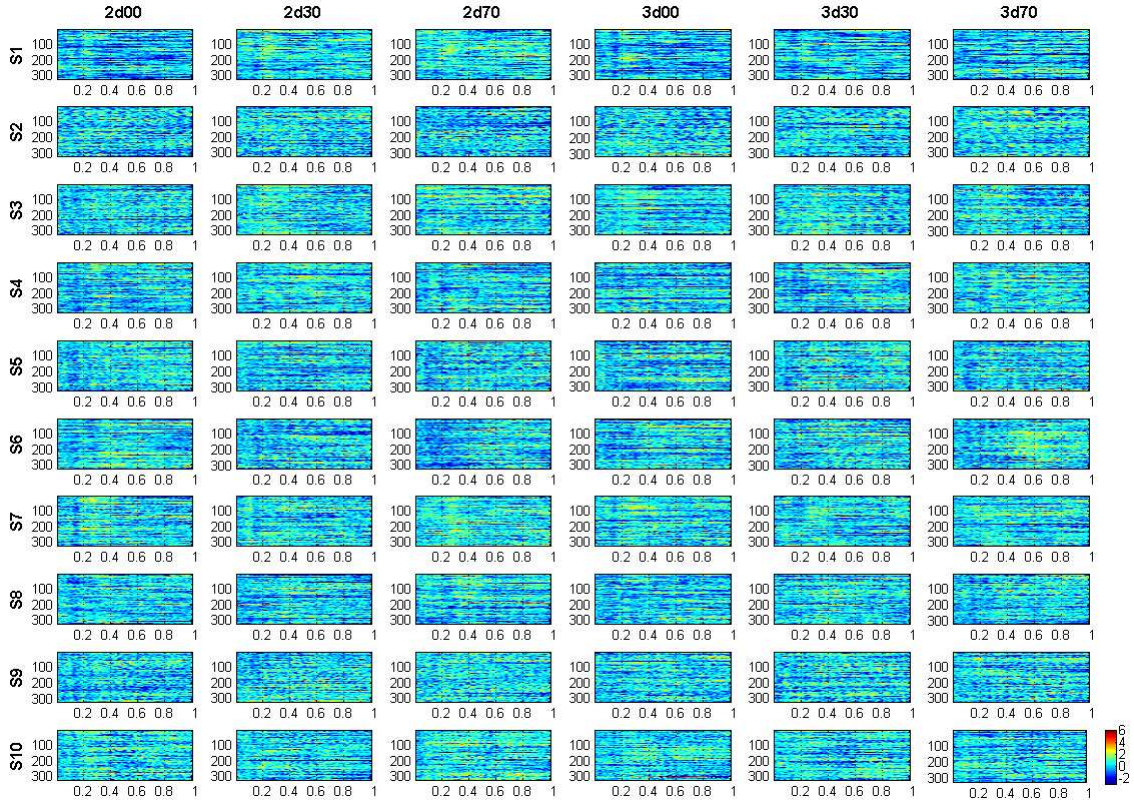

Figure 5.19: Similar plots for non-target epochs at single-trial level.

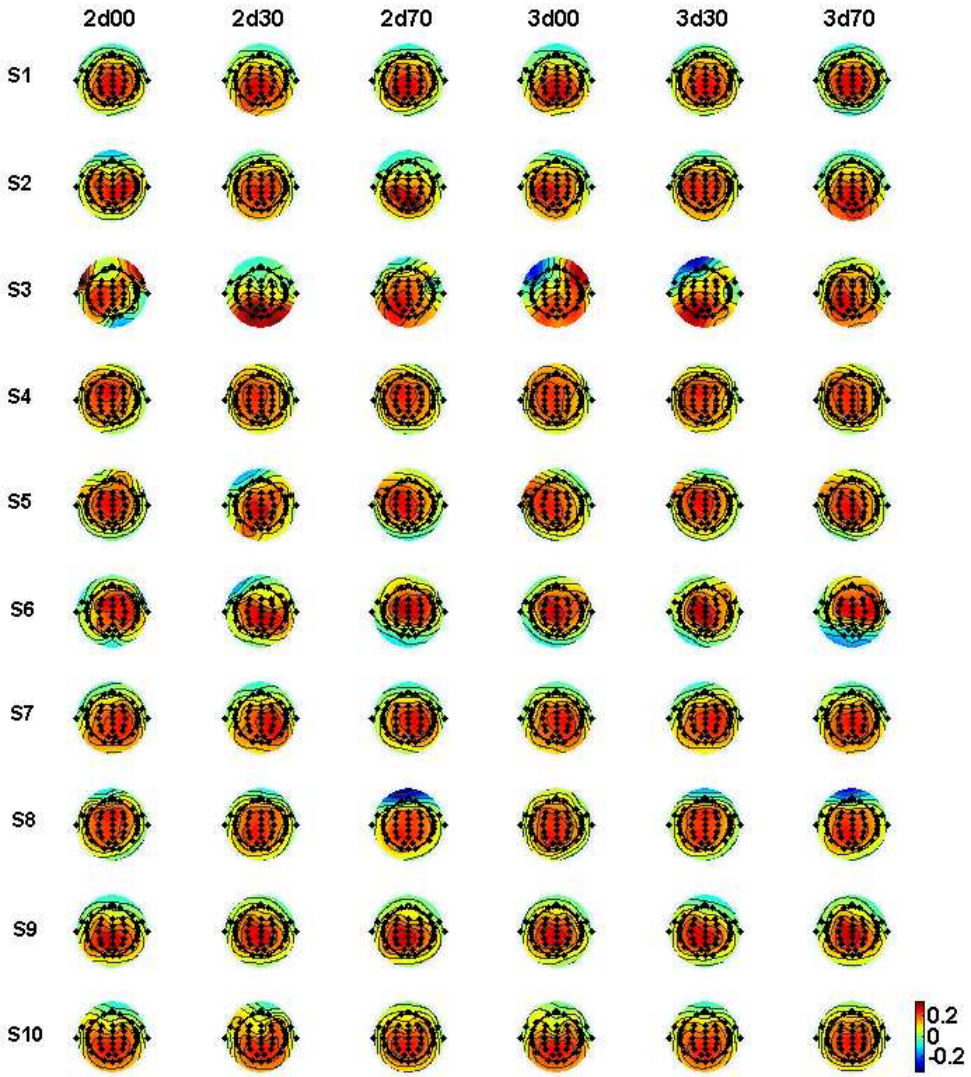

Figure 5.20: normalised scalp distributions represented by the source signals from the one-unit ICA-R for each session

methods usually normalise their source signals to have unit variance. Thus, the scale of the source signal from ICA-R in Fig. 5.17 or that from the traditional ICA method in Fig. 5.21 cannot be directly compared to those raw EEG signals in Fig. 5.16. Nevertheless, the scaling of the extracted signal is usually less important in the ERP studies as long as the signals reflect the dynamics of the desired ERP at the single-trial level.

When comparing the trial-averaged signals from the proposed one-unit ICA-R in Fig. 5.17 with the raw EEG signals in Fig. 5.16, two observations can be made. Firstly, all the extracted signal contains a large positive peak that only occurs between 300 and 600 ms of the trial-averaged target signal. When compared to Fig. 5.16, their peak latencies are also similar to those in the raw trial-averaged signals. Secondly, most of

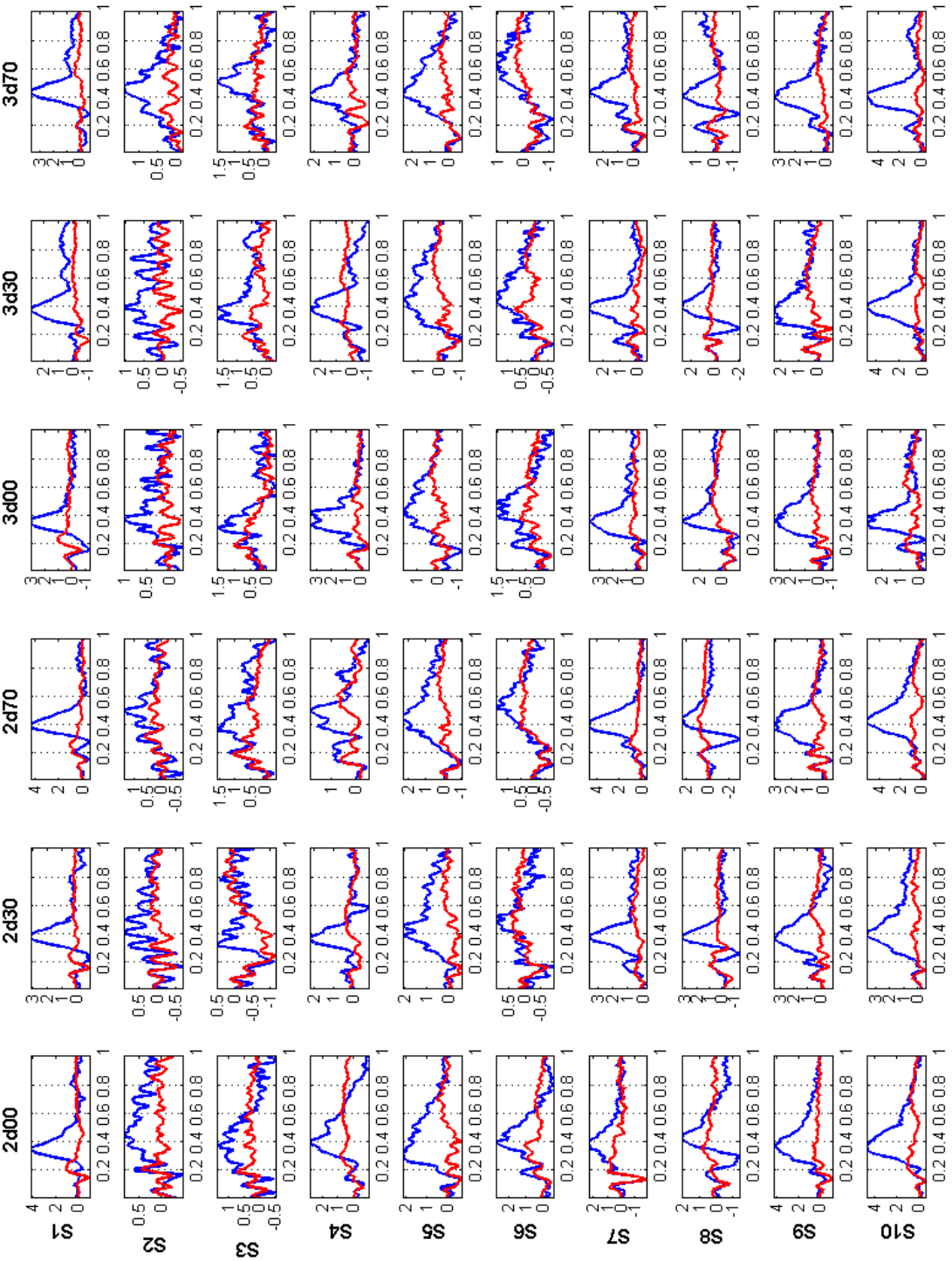

Figure 5.21: Trial-averaged target (blue) and non-target (red) signals segmented from the source signal selected from traditional ICA method. (Note: In each plot, due to the nature of ICA and normalisation effect, the scale in y-axis may not be the actual signal scale. The x-axis represents the time in second.)

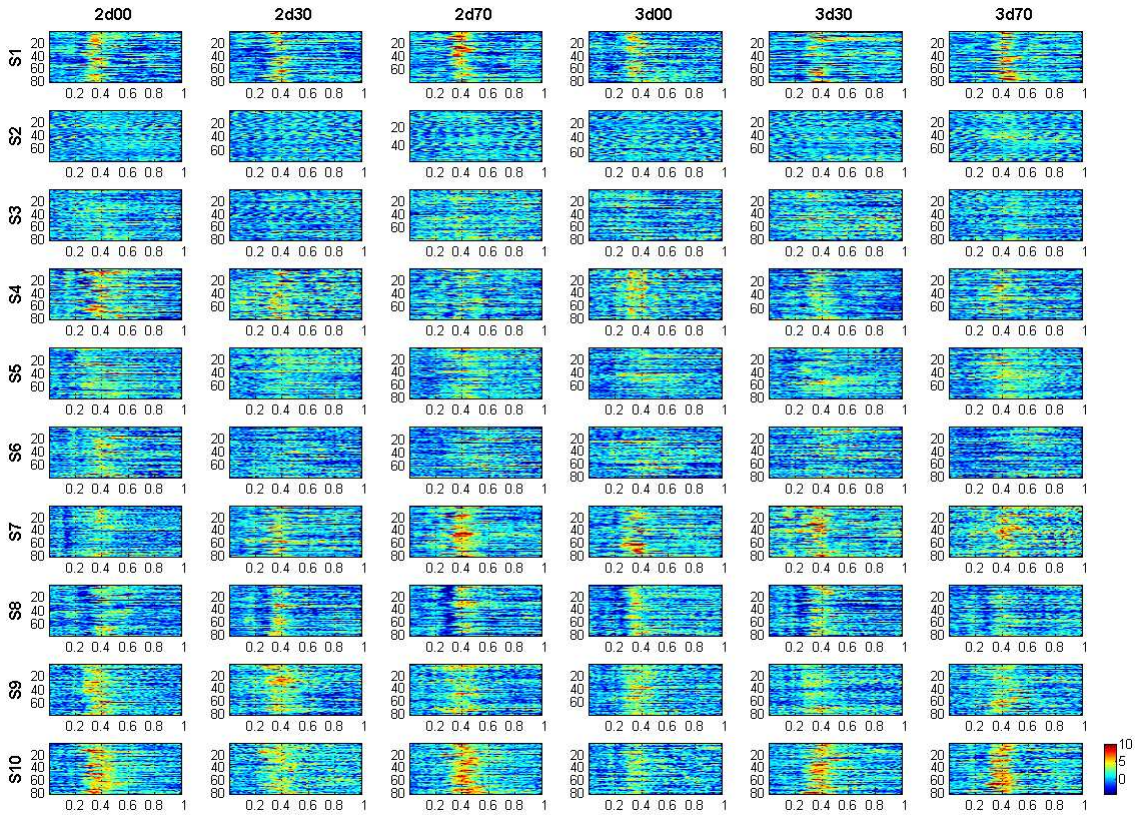

Figure 5.22: Visualization of target epochs segmented from traditional ICA method of different EEG sessions at single-trial level where y-axis and x-axis represents the trial number and time in second respectively.

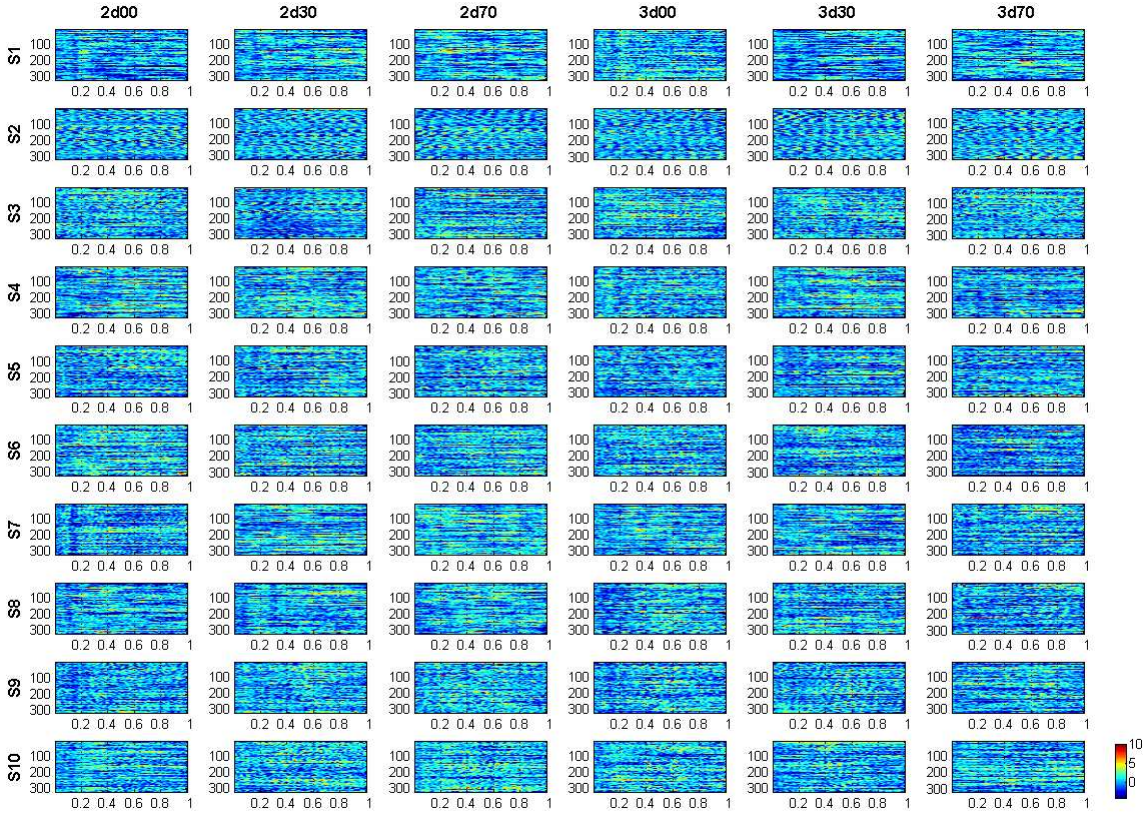

Figure 5.23: Similar plots for non-target epochs at single-trial level.

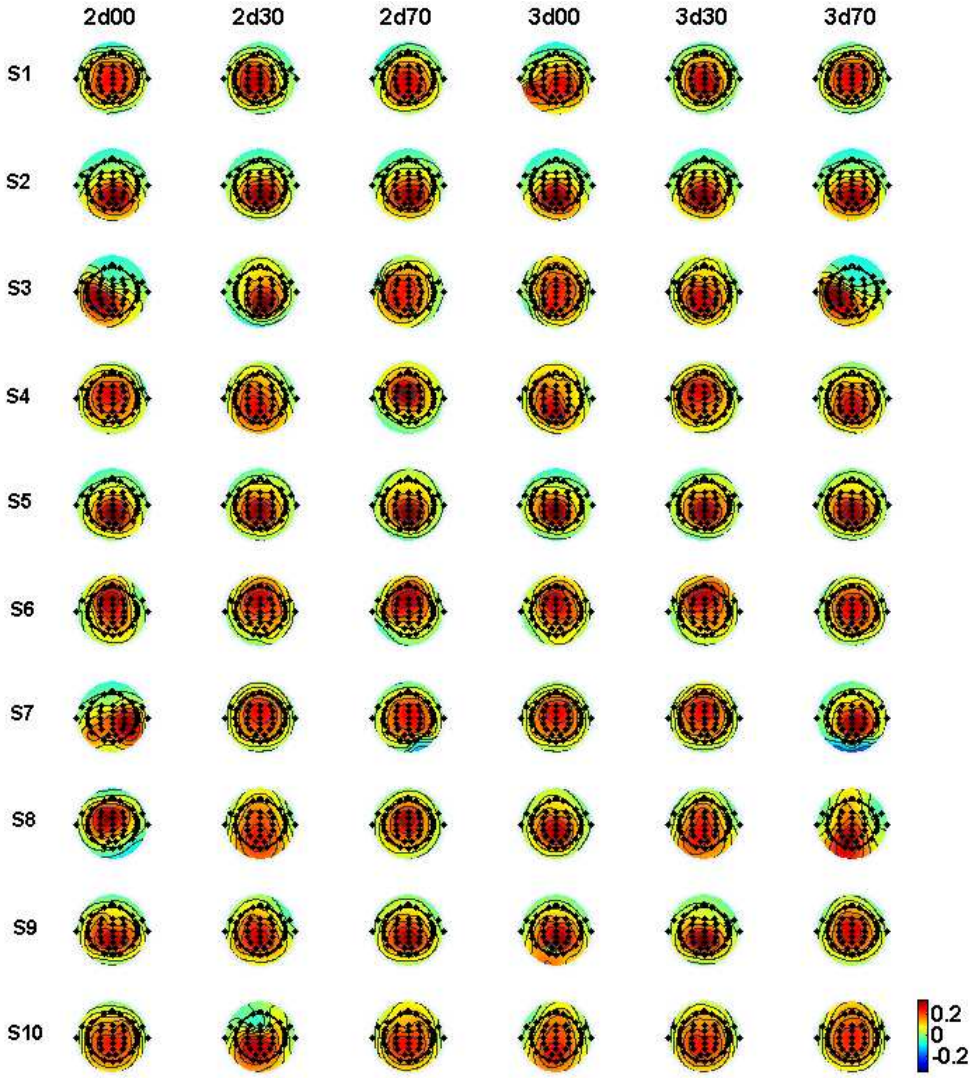

Figure 5.24: normalised scalp distributions represented by the source signals selected from the traditional ICA for each session

the trial-averaged non-target signals segmented from the extracted signals shows no significant activity across time. Unlike the non-target signals in the raw EEG signals, Fig. 5.17 shows that the early ERPs have been suppressed after the extraction by the one-unit ICA-R. These observations can also be made at the target and non-target epochs at single-trial level as shown respectively in Fig. 5.19 and 5.20 respectively. In addition, Fig. 5.20 also shows that almost all the extracted signals resemble the P300 ERP, from which their scalp distributions were found to be central-parietal distributed except for a few sessions with subjects 2, 3, and 6.

For traditional ICA method, in each session, the source selection is performed by selecting one source signal that is central-parietal distributed and has similar peak latency to the original EEG signals. Similar to the one-unit ICA-R, the resultant signals

are examined by its trial-averaged signals, single-trial target and non-target epochs, and scalp distribution respectively in Fig. 5.21, 5.22, 5.23 and 5.24. In general, when comparing the trial-averaged signals of the source signal from both the traditional ICA and one-unit ICA-R methods, one can observe that both methods provide very similar trial-averaged signals and scalp distributions. However, when examining their overall results from Subjects 2, 3 and 6, one can observe that both methods do not work particularly well and this is probably because these subjects have weaker P300s compared to the rest of the subjects. Nonetheless, by putting these sessions aside, the similarity between the results of these two methods suggests that it is viable to perform ERP extraction through ICA-R directly without manual intervention.

5.5.4.2 Quantitative examination

In the previous section, although both ICA and one-unit ICA-R methods have quite similar results in most of the EEG sessions, a closer examination suggests that the signals from traditional ICA method perform poorer in suppressing the irrelevant ERPs. These observations can be made from the trial-averaged signals of some sessions from Subject 7 and 8 as shown in Fig. 5.21. Thus, in this section, to quantify the extraction performance of the traditional ICA and our ICA-R method, we define a metric called Target-to-Non-Target Signal Ratio (TNTR) as follows:

$$TNTR = 10\log_{10}\left(\frac{\sigma^2_{target}}{\sigma^2_{non-target}}\right) \tag{5.18}$$

where σ^2_{target} and $\sigma^2_{non-target}$ are the variance of trial-averaged target and non-target signals respectively. The defined TNTR is similar to the common SNR metric except that the signal and noise variance are now replaced by the variance of the trial-averaged target and non-target signals. The motivation behind this metric is that studies have shown that both target and non-target signals are similar in terms of the early ERPs, except that P300 only occurs in the target signal [29]. Thus, if ICA-R is able to extract only P300, the extracted signal should contain P300 in the target signal while no activity in the non-target signal. Subsequently, if the extracted signal fulfills this requirement, its TNTR value will be large.

To calculate the TNTR for the raw EEG signals, three midline EEG channels (i.e. Channel Fz, Cz and Pz) which are normally considered to have strong P300 ERPs

Table 5.5: Average TNTR across sessions for channel Fz, Cz Pz and also signal extracted from ICA-R and traditional ICA method (without subjects 2, 3 and 6).

Signal	Average TNTR (dB)
Channel Fz	5.02 ± 3.11
Channel Cz	7.35 ± 2.91
Channel Pz	8.84 ± 2.84
Traditional ICA	10.95 ± 2.86
ICA-R	13.48 ± 1.78

were selected. By excluding sessions from Subjects 2, 3 and 6, their averaged TNTR across the remaining sessions were computed and shown in Table 5.5. Similarly, the average TNTR for the source signals from both the traditional ICA and the cascaded ICA-R were also computed and tabulated in Table 5.5. Based on Table 5.5, since P300 tends to be stronger in channel Pz [29], it can be observed that the average TNTR is higher in channel Pz compared to channel Fz and Cz. By comparing these values to those signals from the ICA methods, the results show that the source signals from both ICA methods have higher TNTR compared to channel Pz, with the TNTR from our proposed ICA-R being the highest with 13.48 ± 1.78 dB, followed by traditional ICA with 10.95 ± 2.86 dB. Thus, from these results, it can be seen that the signal from ICA-R is more suitable for studying the P300 since it suppresses the irrelevant ERPs much effectively than the traditional ICA.

5.5.4.3 Analysis of P300 latency

By examining the extracted signal either though the temporal waveform or the scalp distribution, the results suggest that the P300 ERP can be successfully extracted by the proposed one-unit ICA-R. Given the success, the one-unit ICA-R provides the opportunity for studying the characteristic of P300 at the single-trial level. Detailed study of this dataset using the single-trial estimation technique is beyond the scope of this work. However, to understand whether the signal from our ICA-R method provides a meaningful result, a simple analysis was performed by comparing the average median latency of P300 under different experimental conditions (See Appendix C for more details on how median latency of each session was computed.). Fig. 5.25 shows the P300 latency against different levels of visual processing task which involves varying occluded objects under 2D and 3D visualization.

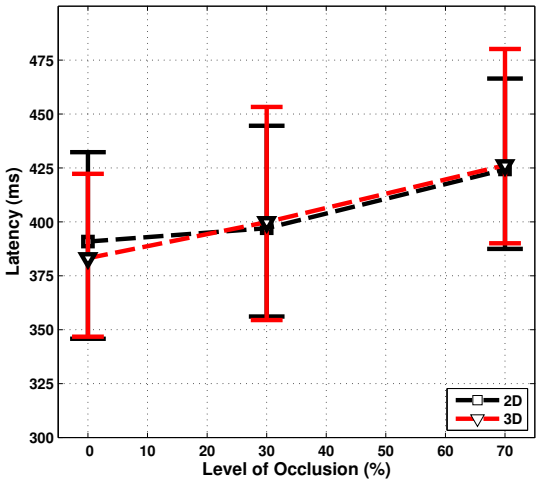

Figure 5.25: Average median latency of P300 when different visualization modalities and level of occlusion were used. The errorbars show the averaged first and third quartile of the ERP latency across subjects (without subject 2,3 and 6).

Our results are quite similar to the observations made in [133]. The peak latency of P300 increases as the level of visual occlusion increases. This is probably because as the level of occlusion increases, the task becomes more difficult and more processing time is required by the brain to recognize the target stimulus. This observation matches the literature finding about the impact of the task difficulty has on the P300 ERP [29]. However, in contrast, our results showed that there is no significant difference between the peak latency of P300 ERP under the 2D and 3D display. This is probably because the information provided under both 2D and 3D display are similar and does not affect the subjects' performance in recognizing the objects.

5.6 Conclusion

ICA is one of the widely adopted EEG signal processing techniques which has been found promising in various ERP studies. It is a kind of blind source separation method which attempts to recover a set of source signals from a set of mixed signals simultaneously without making any prior assumption on the source signal's mixing matrix and waveforms. Since the extraction are performed "blindly", ICA could not provide the source-of-interest directly and source selection is required after the ICA signal decomposition. Since not all the source signals extracted from ICA are eventually of interest

and in most cases, the desired ERP is known beforehand, a form of temporally constrained ICA namely one-unit ICA-R is proposed in this thesis to aid ERP extraction.

For the one-unit ICA-R, it is a method that attempts to extract the signal-of-interest directly through the guidance of a reference signal. The design of a reference signal plays a major part in the success of extracting the ERP using this method. Due to this reason, three types of reference signals were proposed in this chapter whereby our simulation study shows that the previously proposed ENE-LD method is a potential candidate for generating the reference signal mainly because of its ability to reflect the ERP's trial-to-trial variations effectively. Thus, the adaptability of the ENE-LD based signal also helps the ICA-R to achieve a better and consistent extraction performance that is comparable to the existing traditional ICA. By further applying the proposed ICA-R on two real ERP datasets taken from the BCI competition and oddball paradigm, the test results demonstrated the advantages of using the proposed ICA-R for ERP extraction. In BCI application, not only its prediction accuracy comparable to the existing traditional ICA-based methods, our result also demonstrates the proposed one-unit ICA-R is more computationally efficient and does not require any manual source selection. Thus it is a practical tool for real-time BCI application. Apart from that, the proposed one-unit ICA-R is also successfully applied to a larger P300 dataset where the results show that the overall extraction performance by the proposed one-unit ICA-R is better than the traditional ICA method.

Chapter 6

Data-driven Method for P300 Time Region Selection

6.1 Introduction

P300 is one of the ERPs that has been used extensively in many clinical researches to understand different neurological disorders such as dementia, schizophrenia and alcohol dependence [22][29][30]. In recent years, it is also gaining popularity in Brain-Computer Interface as a medium to establish the communication pathway between humans and computers [97][101][113][140].

Similar to other ERPs, identifying the P300 time region is the preliminary step in the ERP studies. This time region often conveys important information for the subsequent signal processing techniques to either extract, detect or analyse the P300 successfully at single-trial level. As shown in the previous two chapters, defining a proper time region is also crucial for our proposed extraction methods to ensure that the characteristic of the ERP can be learned effectively. In practice, obtaining a precise P300 time region at the single-trial level is difficult. Thus, during initialization of most P300 extraction methods, the common practice is to approximate the P300 time region from the trial-averaged signals.

However, as shown in Chapter 5, the task of finding the P300 time region remains challenging even at the trial-averaged signal level mainly because the P300 waveforms are very dependent on the selected channel, subject and experimental conditions. Apart from that, the task is time-consuming and sometimes subjective especially when it

involves visual examination of large amount of signals from multiple electrodes. Other issues such as meeting the real-time requirement like P300 speller also poses further practical complications. To address these issues, a data-driven method that defines the P300 time region objectively and automatically is therefore necessary.

In the literature, the problem of identifying an ERP time region is usually known as ERP segmentation. Its introduction is mainly motivated by the observation that each individual ERP exhibits (i) a stable and unique scalp distribution that (ii) it spans across a specific time region. In the past, several algorithms have been developed for identifying the P300 time region. These algorithms can be divided into two major types, namely adaptive-time and fixed-time segmentation.

As described in Chapter 3, the two adaptive-time segmentation algorithms that are widely used in ERP analysis are 1) the combination of GFP and GMD [103][104], and 2) the clustering-based method called mKM [105]. These two algorithms are data-driven and they are able to identify different ERP time regions simultaneously and adaptively according to the given trial-averaged signal. For GFP and GMD, each ERP time region is identified by finding the time region where the amount of 'activity' is consistently higher and also when there is a sharp transition between two successive scalp distributions. However, this method still requires human intervention. Thus, its outcome can sometimes be biased and subjective when the transition of scalp distribution is not obvious. To automate the ERP segmentation process, mKM is introduced. mKM is a clustering algorithm that attempts to group the similar scalp distributions into clusters based on their spatial correlation. The algorithm begins by selecting a number of scalp distributions randomly across time as the initial cluster representative and then adjusting these cluster representatives iteratively so that each cluster representative closely represents different ERP scalp distributions within the EEG segment. mKM is a potential tool for finding the P300 time region. However, the method suffers from two major drawbacks. Firstly, the starting point for the cluster representatives and the correct number of clusters used to represent the ERPs are two crucial factors for mKM. However, determining a good initialization and the correct number of cluster is a non-trivial problem by itself. To overcome this problem, multiple runs using different starting points and number of clusters are often necessary [103]. Secondly, mKM is not designed specifically for determining the P300 time region. Since not all the

clusters are associated with the P300 ERP, cluster selection is necessary so that the P300 time region can be determined. However, in current mKM implementations, this selection can only be achieved through visual examination.

Due to the complexity of the aforementioned techniques, another approach to determine the P300 time region is through fixed-time segmentation. In this approach, the P300 time region is assumed to have fixed time length regardless of the subject and experimental conditions. In addition, most fixed-time segmentation algorithms require two trial-averaged signals (one obtained from the target (P300) trials, and the other from non-target (non-P300) trials) to perform the ERP segmentation successfully. To identify the P300 time region, the common approach is to employ a sliding fixed-time window (e.g. 50, 100, 200 ms) to scan across the trial-averaged signals for the most relevant time region. In [53], the author proposed to measure the discriminativity of each candidate P300 time window by comparing it between the target and non-target signals so that the best performing time region can be selected. However, the suggested discriminativity information does not provide any conclusive results [53]. Manual selection is still required before the relevant time window can be selected. In [135], the selection of the ERP time region is based on the spatial filter called xDAWN. Assuming that the desired ERP always contributes the most variance towards the EEG signals, the method attempts to identify the ERP time region by finding a time window that provides the highest eigenvalue for the first component extracted by xDAWN. Different to the sliding window approach, another fixed-time segmentation is also proposed based on xDAWN [136]. The algorithm starts by first finding a set of p largest amplitudes in the first component extracted by xDAWN. The P300 time region is then given by the earliest and latest time in the given set. The algorithm is repeated by training xDAWN with the newly identified time region. The iteration stops when the selected time region converges. In general, most fixed-time segmentation algorithms are designed specifically for identifying the P300 time region. Since no manual intervention is required, they are more practical and often used in real-time application. However, the fixed-time segmentation algorithms also experience two major drawbacks. Firstly, the algorithm is not adaptive and has the tendency to under- or over-represent a P300 time region. Secondly, as described above, most fixed-time segmentation algorithms cannot work independently. Instead of examining the underlying scalp distribution

structure, these algorithms normally have to combine with the subsequent extraction or classification techniques in order to work effectively. As a result, most fixed-time segmentation algorithms are often restricted to a specific technique and cannot be applied as a general tool to extract the time information for various ERP signal processing techniques.

As mentioned above, although there are increasing numbers of applications on the P300 ERP and the P300 time region has an influential impact on most of the P300 algorithms, a suitable method for determining the P300 time region remains an open question. Moreover, the problem is addressed differently depending on the targeted P300 applications. Clearly, adaptive-time segmentation is better than the fixed-time segmentation since they can reflect the changes in the P300 waveform. The obtained time region is also more precise and reliable for ERP extraction and classification. However, issues regarding the number of cluster for representing the underlying ERPs and also the cluster selection can be problematic when applying adaptive-time segmentation such as mKM without human intervention.

To overcome the above issues, a new clustering approach for P300 ERP segmentation namely the automatic 2-cluster K-means algorithm (a2KM) is presented in this chapter. Firstly, we propose to overcome the issue of number of cluster by reducing the ERP segmentation task into a simpler form in which the signal is only divided into two regions (i.e. P300 and non-P300 time regions). As such, the number of cluster is restricted to only two which further simplifies the cluster selection process. Secondly, we also propose a new way of selecting the cluster so that the P300 time region can be defined automatically without any human intervention and therefore practical for real-time P300 applications.

The organization of this chapter is as follows. The detail of the proposed algorithm is first described in Section 6.2. For evaluation, the proposed method was tested on different EEG sessions from the P300 oddball paradigm dataset whereby the results of two major examinations are discussed in Section 6.3. In the first examination, the P300 time regions identified by the proposed method were visually examined and compared to the mKM algorithm. For the quantitative examination, different P300 time regions were also used to train the spatial filters for extracting the P300. The resultant extracted signals are then measured for their quality through a ERP classification method. After

that, in Section 6.4, the method was applied on the P300 speller dataset to examine the prediction accuracy of the epochs extracted based on the time regions given by the proposed method. Lastly, the conclusions are drawn in Section 6.5.

6.2 Proposed algorithm

6.2.1 Problem formulation

P300 is a type of endogenous ERP whose elicitation relies heavily on the context of the given stimulus. Similar to other ERPs (e.g. mismatch negativity, N400 and Error-Related Negativity), the P300 experiment usually involves at least two different types of stimulus which are either task-relevant (target) or task-irrelevant (non-target). Often, the target and non-target signals contain the same early sensory ERPs such as N1, P200 and N200. The only difference is that P300 is only present in the target signal but absent in the non-target signal.

Based on these observations, the trial-averaged multi-channel target $\mathbf{X}_{\text{T}}$ and non-target signal $\mathbf{X}_{\text{NT}}$ can be formulated as below:

$$\mathbf{X}_{\text{T}} = \mathbf{X}_{\text{P300}} + \mathbf{X}_{\text{CE}} + \mathbf{N}_1 \tag{6.1}$$

$$\mathbf{X}_{\text{NT}} = \mathbf{X}_{\text{CE}} + \mathbf{N}_2 \tag{6.2}$$

whereby $\mathbf{X}_{\text{T}}$, $\mathbf{X}_{\text{NT}}$, $\mathbf{X}_{\text{P300}}$, $\mathbf{X}_{\text{CE}}$, $\mathbf{N}_1$ and $\mathbf{N}_2$ are $N_e \times N_T$ dimensional spatio-temporal matrices, N_e represents the number of channels while N_T represents the total number of time samples. $\mathbf{X}_{\text{P300}}$ and $\mathbf{X}_{\text{CE}}$ represents the multi-channel P300 and common early ERP respectively, while $\mathbf{N}_1$ and $\mathbf{N}_2$ represent different background noise observed in the target and non-target signal.

Let $x_{\text{P300}}(e,t)$ be the element of $\mathbf{X}_{\text{P300}}$ which represents an EEG measurement from electrode e at a particular time t. Since P300 ERP only occurs within a certain time interval $[t_1, t_2]$, $x_{\text{P300}}(e,t)$ is only non-zero when $t \in [t_1, t_2]$, i.e.,

$$x_{\text{P300}}(e,t) = \begin{cases} 0 & \text{for } t \notin [t_1, t_2] \\ z(t) & \text{for } t \in [t_1, t_2] \end{cases} \tag{6.3}$$

where $z(t)$ represents a P300 ERP waveform. Thus, principally, the task of determining the P300 time region is equivalent to determining the time interval $[t_1, t_2]$ where $x_{\text{P300}}(e, t)$ is non-zero.

Assuming that $\mathbf{N}_1$ and $\mathbf{N}_2$ are negligible after trial averaging, an approximation of $\mathbf{X}_{\text{P300}}$ can be obtained by using $\mathbf{X}_{\text{NT}}$ as a baseline to compute the differential signal $\mathbf{X}_{\text{d}}$:

$$\mathbf{X}_{\text{d}} = \mathbf{X}_{\text{T}} - \mathbf{X}_{\text{NT}} = \tilde{\mathbf{X}}_{\text{P300}} \tag{6.4}$$

After subtraction, the common and irrelevant ERPs are removed. Subsequently, the segmentation problem is reduced to a simpler task with only P300 and non-P300 time regions remained.

Thus, in this work, we proposed to perform our segmentation algorithm on the differential signal $\mathbf{X}_{\text{d}}$ instead of the target signal $\mathbf{X}_{\text{T}}$. To perform the segmentation successfully, our proposed algorithm involves two major processing stages. In the first stage, we deal with the remnants of the common ERPs in $\mathbf{X}_{\text{d}}$ through signal conditioning. In the second stage, the automatic 2-cluster KM (a2KM) algorithm is carried out to find the P300 time region. An overview of our proposed segmentation algorithm is shown in Fig. 6.1.

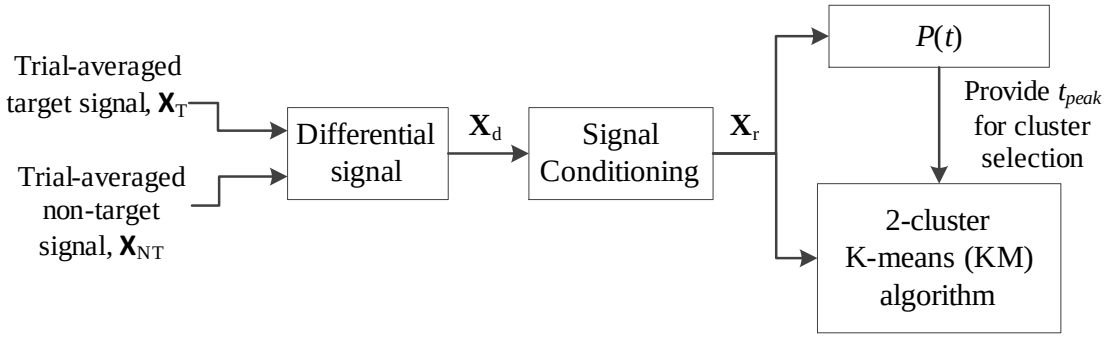

Figure 6.1: The schematic diagram for our proposed a2KM algorithm for identifying the P300 time region

6.2.2 Signal conditioning

In general, the mismatch between the common ERPs from the target and non-target signals are fairly common. As a result, there will be remnants of the irrelevant ERPs remaining in $\mathbf{X}_{\text{d}}$ after subtraction. For example, Fig. 6.2(a) shows the difference between

the trial-averaged target and non-target signals from an EEG channel. Due to the mismatches in the P200 and N200, the resultant signal partially contains these two ERPs as shown in Fig. 6.2(b). Therefore, further noise reduction is required.

First, since the P300 is a positive signal, the differential signal $\mathbf{X}_{\mathrm{d}}$ should only contain positive signals. By half-wave rectifing $\mathbf{X}_{\mathrm{d}}$ so that all the negative element is zeroed, the rectified signals $\mathbf{X}_{\mathrm{r}}$ can be computed as:

$$x_{\mathrm{r}}(e,t) = \max\{x_{\mathrm{d}}(e,t), 0\} \tag{6.5}$$

where $x_{\mathrm{d}}(e,t)$ and $x_{\mathrm{r}}(e,t)$ are denoted as an element of $\mathbf{X}_{\mathrm{d}}$ and $\mathbf{X}_{\mathrm{r}}$ respectively.

After half-wave rectification, some positive residual signals remain as shown in Fig. 6.2(c). To minimise these residual signals and only emphasize the large P300 ERP, we borrow an idea from conventional electrocardiography (ECG) signal processing [137][138], in which a nonlinear amplification is performed on $\mathbf{X}_r$ by squaring each element and averaging across the channels at every time instant. The resultant channel-free signal $P(t)$ is computed as below:

$$P(t) = \frac{1}{N_e}\sum_{e=1}^{N_e} x_{\mathrm{r}}^2(e,t) \tag{6.6}$$

However, $P(t)$ can still be noisy after non-linear amplification. Thus, a forward-backward moving-average filter is introduced to smooth $P(t)$. In this chapter, a moving-average window of 50 ms is used. An example of $P(t)$ and the smoothed $P(t)$ are shown in Fig. 6.2(d) and (e) respectively.

Alternatively, $P(t)$ can be interpreted as the channel-averaged instantaneous power of $\mathbf{X}_r$ whereby it measures the 'activity' of the P300 at each time instant. Ideally, the P300 time region can be selected by choosing the non-zero segment in $P(t)$. However, as shown in Fig. 6.2(e), signal conditioning can only attenuate but not completely remove the remnant ERPs. Therefore, a proper method to select the time region is required. Although thresholding is a viable solution, determining the threshold on $P(t)$ can be difficult since the range of $P(t)$ varies across sessions as well as the residue noise level. To address this issue, a segmentation algorithm is proposed in the next section.

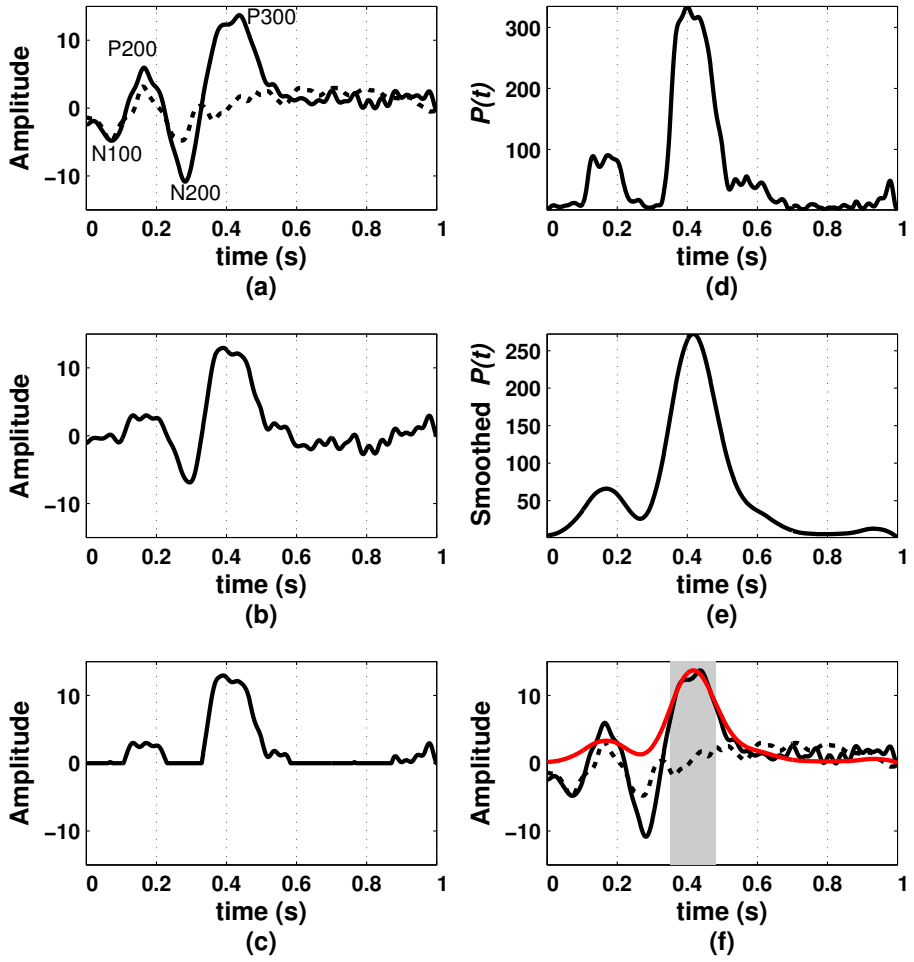

Figure 6.2: An example of signals observed at different stages of the proposed algorithm. (a) Trial-averaged target (solid) and non-target (dotted) signal from channel Pz of an EEG session; (b) the differential signal $\mathbf{X}_d$ and (c) half-wave rectified signal $\mathbf{X}_r$ from the same EEG channel; (d) the squared and channel-averaged signal $P(t)$; (e) the smoothed $P(t)$; and (f) the plot in (a) is re-plotted together with a re-scaled (e) in red line while the grey area represents the P300 time region identified by the proposed a2KM algorithm.

6.2.3 Signal segmentation

After signal conditioning, the second stage of our proposed algorithm is to perform segmentation so that the P300 time region can be identified. As described earlier, since there are only P300 and non-P300 time regions remaining in $\mathbf{X}_{\mathrm{r}}$, the P300 and non-P300 time regions can be considered as the active and quiet stages in the multi-channel signals $\mathbf{X}_{\mathrm{r}}$. During active stage or when P300 occurs, it is assumed that there will be a change in the mean 'activity' when compared to the quiet stage. Thus, for this reason, we apply the standard squared euclidean KM algorithm with 2 clusters on $\mathbf{X}_{\mathrm{r}}$ so that the resultant clusters represent two different level of mean 'activity' as found in the P300 and non-P300 time region.

Let $k \in \{1, 2\}$ be the cluster number, $\mathbf{x}_{\mathrm{r}}(t)$ be the multi-channel measurement vector (or column vector) in $\mathbf{X}_{\mathrm{r}}$, S_k represents the cluster, $\mathbf{c}_k$ represents the centroid of S_k and $m(t)$ be the cluster label for each $\mathbf{x}_{\mathrm{r}}(t)$. The pseudocode for the standard 2-cluster KM algorithm is as follows [106]:

1. Randomly select two $\mathbf{x}_{\mathrm{r}}(t)$ as the initial centroids $\mathbf{c}_1$ and $\mathbf{c}_2$ for clusters S_1 and S_2 respectively.

2. For each time instant t, assign $\mathbf{x}_{\mathrm{r}}(t)$ to the nearest cluster S_k based on their squared euclidean distance to the cluster centroid $\mathbf{c}_k$ as follows:

$$m(t) = \underset{k \in \{1,2\}}{\operatorname{argmin}} \quad \|\mathbf{x}_{\mathrm{r}}(t) - \mathbf{c}_k\|^2 \tag{6.7}$$

3. Let $S_k = \{\mathbf{x}_{\mathrm{r}}(t)|_{m(t)=k}\}$ contains the set of $\mathbf{x}_{\mathrm{r}}(t)$ that are nearest to the cluster centroid $\mathbf{c}_k$.

4. Update each cluster centroid $\mathbf{c}_k$ with the available members in the cluster S_k

$$\mathbf{c}_k = \frac{1}{N_k} \sum_{\mathbf{x}_r(t) \in S_k} \mathbf{x}_{\mathrm{r}}(t) \tag{6.8}$$

5. Calculate the total distance-to-centroid

$$err = \sum_{k} \sum_{\mathbf{x}_r(t) \in S_k} \|\mathbf{x}_{\mathrm{r}}(t) - \mathbf{c}_k\|^2 \tag{6.9}$$

6. Repeat Step 2–5 until there is no membership changes in S_k or when err converged.

After applying the above KM algorithm, the cluster S_k that is relevant to the P300 ERP has to be identified before the P300 time region can be retrieved. To perform the cluster selection automatically, we propose that the time location of $P(t)$ peak can be a potential candidate for determining the P300 cluster. To locate the peak of $P(t)$, a simple peak detection is applied. Suppose that the time location of the peak is given as t_{peak}, the cluster whose members contain the measurement vector $\mathbf{x}_{\mathrm{r}}(t_{peak})$ is selected as the P300 cluster while the time indices correponding to the members of P300 cluster are retrieved as the P300 time region as follows:

$$\mathcal{T} = \{t|_{m(t)=m(t_{peak})}\} \tag{6.10}$$

where $\mathcal{T}$ represents the P300 time region. Fig. 6.2(f) shows an example of the P300 time region identified by the proposed automatic 2-cluster KM algorithm.

6.3 Experimental study on oddball paradigm dataset

6.3.1 Overview

As explained earlier, in practice, it is very difficult to determine the exact P300 time region even at the trial-averaged signal level. Thus, to examine whether our proposed a2KM algorithm is able to select a quality P300 time region, two examinations (i.e. qualitative and quantitative) were carried out in this section. For the qualitative examination, our objective is to examine the segmentation results from the following perspectives:

1. By using trial-averaged target signals as reference, are the selected time regions located within the acceptable P300 time range of every session?

2. Mislabeling – Does the segmentation algorithm falsely include any time region that is less likely to be the P300 time region?

3. Fragmentation – Does the selected time region contains disjointed time segments within the expected P300 time region?

4. Undersegmentation – Does the selected time region covers an acceptable amount of time within the expected P300 time region?

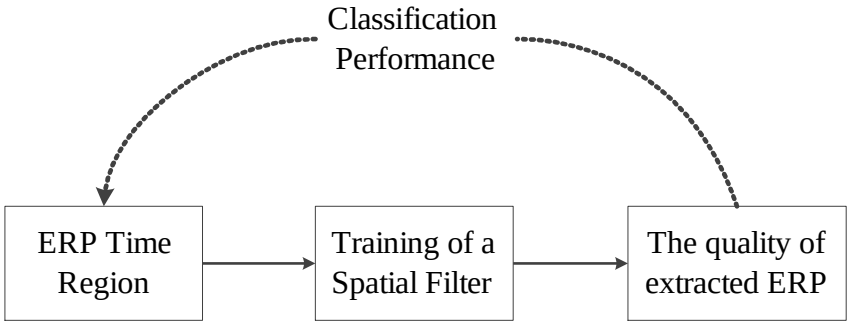

Figure 6.3: The cause-effect relationship between a ERP time region and the ERP extraction method. The dotted-line represents the method used to measure the quality of a selected time region in our quantitative examination.

Fig. 6.3 shows the cause-effect relationship between a ERP time region and the ERP extraction method. In practice, most data-driven ERP extraction methods require a proper ERP time region to find the relevant samples for computing the statistics on the desired ERP. As a result, depending on the selected ERP time region, the resultant samples will affect the training of a spatial filter and further its performance in extracting the ERPs. Specifically, in this case, a good P300 ERP time region will ensure that the spatial filter extracts every P300 ERP successfully from the target trials, and thus creates discrimination between the target and non-target trials.

For this reason, in the second examination, we evaluate the performance of our proposed a2KM algorithm quantitatively by examining the classification performance of the signals extracted by a spatial filter when the time region given by a2KM is applied. For a comprehensive study, to understand how a selected time region has an influence on the ERP extraction methods, different time regions and spatial filters are also selected for comparison.

6.3.2 Procedure for qualitative examination

Before performing the segmentation, the EEG signals of each session was pre-processed and prepared by using the same procedure applied in Section 5.5.3.1. For each EEG session, the multi-channel signals were referenced to the average of channel A1 and A2. Then they were bandpass filtered[1] between 0.5 and 30 Hz by applying forward-backward equiripple filter. Ocular artifacts were removed using the method in [117]

[1]A combination of a forward-backward 2nd order elliptic highpass and 8th order elliptic lowpass filters.

while in some sessions, the bad EEG channels were also removed from the analysis. For each EEG session, an EEG epoch of 1000 ms duration after the stimulus onset was extracted, followed by a baseline correction using a 200 ms EEG segment prior to the stimulus onset. After that, the epochs were divided into target and non-target groups before averaging across trials to form the trial-averaged target $\mathbf{X}_{\mathrm{T}}$ and non-target signal $\mathbf{X}_{\mathrm{NT}}$ respectively.

Based on the procedures elaborated in Section 6.2, the proposed a2KM algorithm is applied to identify the P300 time region for each session in the oddball paradigm dataset. To obtain a reliable result, a2KM was repeated with five different random starting points and the outcome with the minimum total distances-to-centroid err was chosen (see the steps in Section 6.2.3).

For comparison, the traditional mKM algorithm was implemented as follows [104][105]:

1. Concatenate the trial-averaged target and non-target signals, $\mathbf{X}_{\mathrm{T}}$ and $\mathbf{X}_{\mathrm{NT}}$ respectively, across time [105]. In particular, the newly formed multi-channel signals $\mathbf{X}_{\mathrm{TNT}}$ is obtained as: $\mathbf{X}_{\mathrm{TNT}} = [\mathbf{X}_{\mathrm{T}}\ \mathbf{X}_{\mathrm{NT}}]$.

2. Remove the mean of every multi-channel measurement vector in $\mathbf{X}_{\mathrm{TNT}}$ [2].

3. Set the number of cluster to 2 and perform the mKM segmentation on $\mathbf{X}_{\mathrm{TNT}}$.

4. Based on the segmentation results, calculate a total distance-to-centroid measure known as modified cross-validation criterion as proposed in [105].

5. Repeat Step 3 and 4 with 20 different starting points and also different number of clusters ranging from 3 to 20.

6. Among all the segmentation results obtained in Step 5, select the one with the least modified cross-validation score.

7. Retain only the first half of the selected segmentation result since only the first half of $\mathbf{X}_{\mathrm{TNT}}$ represents the trial-averaged target signals.

8. From the final segmentation result, select the cluster that is most relevant to P300 ERP by using our proposed cluster selection method.

[2] This step is also equivalent to the step of "re-referencing the EEG signals to average channel reference" mentioned in the original article [105]

6.3.3 Procedure for quantitative examination

As discussed earlier, in this study, our objective is to evaluate the impact a selected P300 time region has on a spatial filter. For this reason, four different time regions were chosen as the P300 time region. Each of these time regions was taken either from fixed time regions such as (i) 0–1000 ms[3] and (ii) 300–600 ms, or from the segmentation algorithms using (iii) the conventional mKM and (iv) the proposed a2KM algorithm. In addition, three spatial filters are selected to evaluate the quality of the time region. These spatial filters are our proposed ENE-LD based one-unit ICA-R, conventional LD (cLD) [89] and xDAWN [84].

In this study, EEG recording from each session was prepared by using the procedure similar to the previous section, except that, for classification purposes, the EEG recordings were bandpass-filtered[4] to between 0.5 and 12 Hz. This was then followed by the application of ocular artifact removal. Once completed, every target and non-target epoch was extracted, as a 1000-ms EEG segment starting from the stimulus onset, followed by a baseline correction using the 200 ms EEG segment prior to the stimulus onset. Lastly, every epoch was downsampled from 1000 Hz to 50 Hz[5].

Fig. 6.4 illustrates an overview of the overall ERP classification system designed for evaluating the quality of a selected P300 time region. The detail procedure is as follows:

1. In this study, a 5-fold cross-validation method was employed, in which, all the target and non-target epochs were divided into five partitions. In each round of validation, four partitions were used as training data while the remaining partition was selected as test data.

2. In the cases that apply the mKM and a2KM algorithm, the P300 time region was decided by performing the segmentation with the available training epochs. In particular, these training epochs were first used to obtain the trial-averaged

[3]In practice, spatial filters such as CSP, xDAWN, and some variants of LD are commonly trained based on the samples taken from the entire training epochs [82][84][75]. In this dataset, this step is also equivalent to selecting a time region that covers from 0 to 1000 ms.

[4]A combination of a forward-backward 2nd order elliptic highpass and 10th order elliptic lowpass filters.

[5]It is a common practice to perform downsampling and keeping the number of features less than the number of training epochs during ERP classification so that the problem of overfitting in the classifier, or also known as the curse of dimensionality, can be avoided [139].

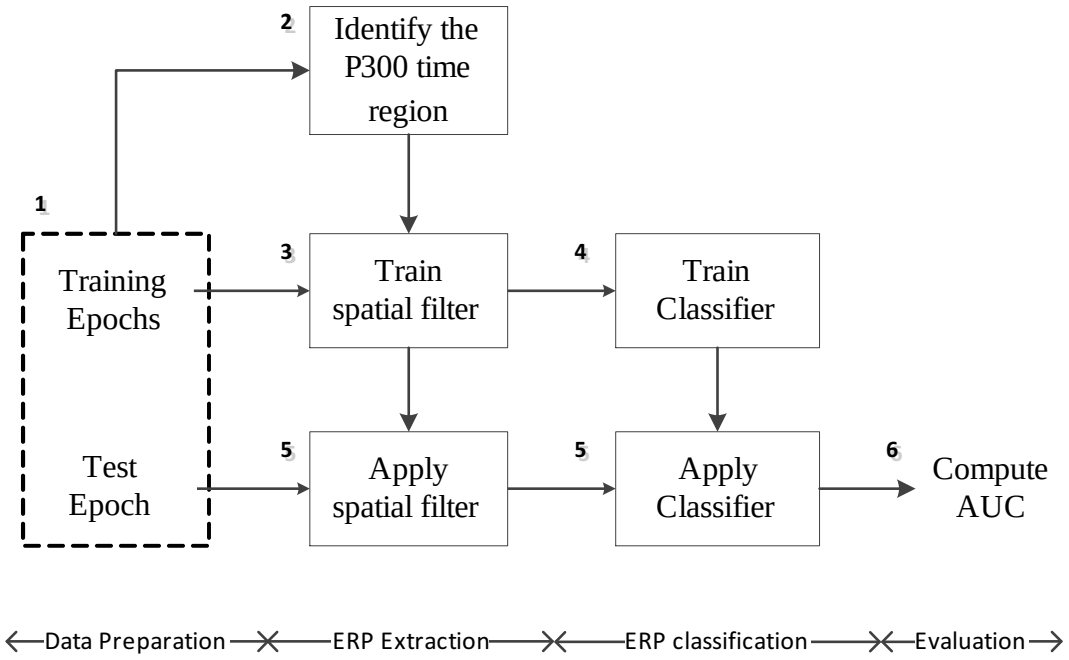

Figure 6.4: The overall ERP classification system used to examine the quality of a selected P300 time region.

signals $\mathbf{X}_{\mathrm{T}}$ and $\mathbf{X}_{\mathrm{NT}}$. After that, the same ERP segmentation procedure described in Section 6.3.2 was applied to determine the P300 time region.

3. After obtaining the P300 time region, all multi-channel measurement vectors from the selected time region of the training epochs were used to train a spatial filter. This newly obtained spatial filter was then applied on the training epochs to extract the P300 ERPs. Subsequently, at this stage, each epoch was an extracted signal with a dimension of 1×51.

4. All the training epochs were taken as feature vectors for training the popular Bayesian Linear Discriminant Analysis (BLDA) classifier. The reason for using BLDA is mainly because unlike SVM, it does not require any further cross-validation to fine-tune the parameters [101].

5. For classification, the constructed spatial filters and classifiers were applied to the test epochs. It should be noted that for all different combinations of time region and spatial filter, ERP classifications are performed with the same number of features (or the same time length for the extracted signal).

6. Lastly, the classifier scores obtained from all test epochs were used to construct a Receiver-Operating-Characteristic (ROC) curve whose Area-Under-Curve (AUC)

was computed to represent the classification performance of a given combination of P300 time region and spatial filter.

6.3.4 Results and discussion

6.3.4.1 Qualitative examination

In this section, for the ease of examination, the signals and segmentation results from the different EEG sessions were concatenated in the order that every six rows of signals represent respectively sessions 2d00, 2d30, 2d70, 3d00, 3d30, and 3d70 from a specific subject. For comparison, channel Pz is selected for evaluating the performance of the proposed algorithm.

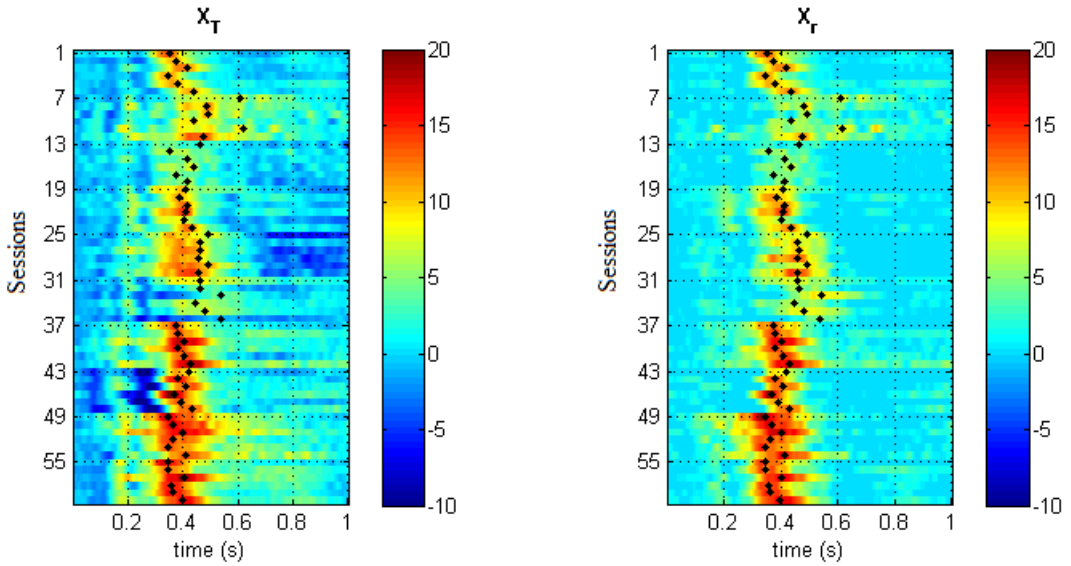

Figure 6.5: (a) The trial-averaged signals given by channel Pz from $\mathbf{X}_{\mathrm{T}}$ of different EEG sessions (b) The trial-averaged signals given by channel Pz from the rectified signals $\mathbf{X}_{\mathrm{r}}$ of different EEG sessions. The black dots represent the location of $P(t)$ peaks (see Section 6.2.2). Note that, the EEG sessions from 10 different subjects are concatenated in the order that every six sessions represent the EEG session 2d00, 2d30, 2d70, 3d00, 3d30, and 3d70 from a specific subject respectively (e.g. session no. 1–6 referred to subject 1, session no. 7–12 referred to subject 2 and so on so forth.)

Firstly, Fig. 6.5(a) and (b) shows respectively the heatmap of the concatenated signals from channel Pz of $\mathbf{X}_{\mathrm{T}}$ and $\mathbf{X}_{\mathrm{r}}$ from different EEG sessions. Fig. 6.5(a) shows that the P300 is readily observed between 300 ms and 600 ms of the trial-averaged target signals $\mathbf{X}_{\mathrm{T}}$. However, its time location and duration varies across different subjects and experiment sessions, which is also the reason why an adaptive segmentation algorithm

is crucial for obtaining the P300 time region. To perform our proposed segmentation algorithm successfully, Fig. 6.5(b) shows the importance of the signal conditioning step in eliminating the irrelevant ERPs. In Fig. 6.5(b), it can be clearly seen that the traces of early ERPs around 200 ms are significantly reduced in the rectified signal $\mathbf{X}_{\mathrm{r}}$. Eventually, when compared to the original target signals $\mathbf{X}_{\mathrm{T}}$, the P300 and non-P300 time regions are also more distinct in the rectified signals $\mathbf{X}_{\mathrm{r}}$.

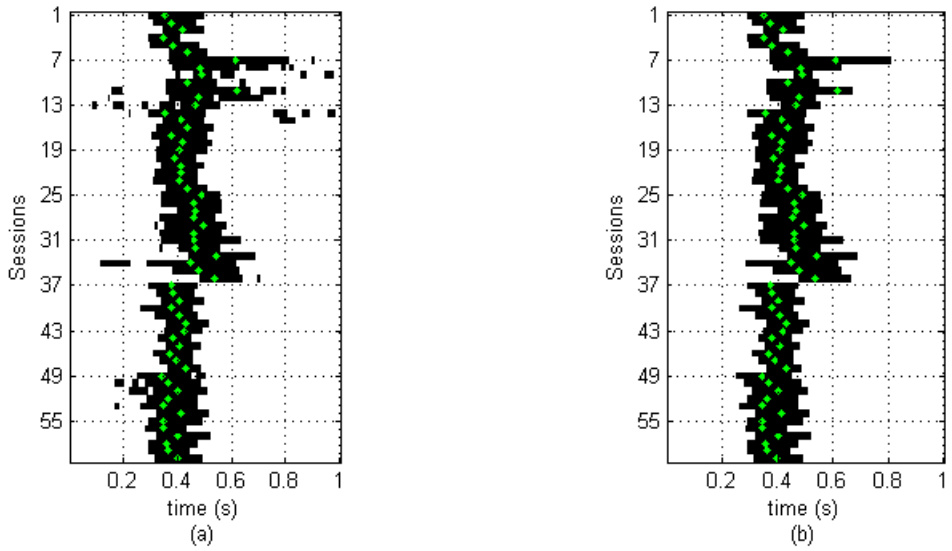

Figure 6.6: (a) The P300 time regions (black) identified in each EEG sessions through the proposed a2KM algorithm. The green dots represent the location of $P(t)$ peaks which are used for cluster selection. (b) The P300 time regions given by a2KM when the constraint is imposed to retain only the connected time region that contains the $P(t)$ peak.

Fig. 6.6(a) shows the P300 time region identified by our proposed automatic segmentation algorithm. The results reveal that the identified P300 time regions are similar to the P300 time regions observed in both $\mathbf{X}_{\mathrm{T}}$ and $\mathbf{X}_{\mathrm{r}}$ as seen in Fig. 6.5(a) and (b). A closer examination indicates that the success to find the P300 time region automatically can be also attributed to the reliable P300 time location provided by the peak of $P(t)$ [see (6.6)]. By examining the location of the $P(t)$ peak in Fig. 6.5(a) and (b), the results show that its location is always situated within the P300 time regions. Subsequently, this also helps our proposed a2KM to select the correct cluster for representing the P300 time region.

However, it should be noted that in some occasions, mislabeling does happened

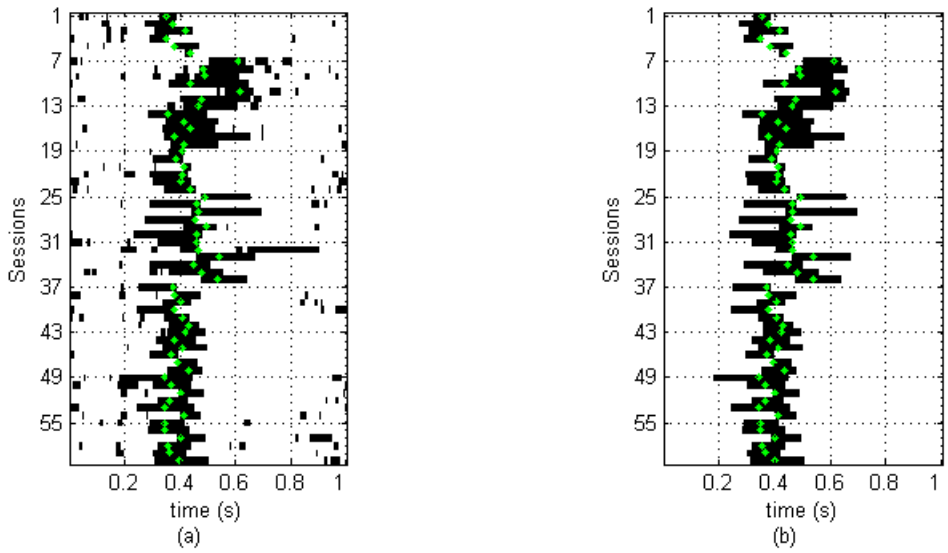

Figure 6.7: (a) The P300 time regions (black) identified in each EEG sessions through the conventional mKM algorithm. The green dots represent the location of $P(t)$ peaks which are used for cluster selection. (b) The P300 time regions given by mKM when the constraint is imposed to retain only the connected time region that contains the $P(t)$ peak.

in our proposed a2KM where short and meaningless time segments[6] were found in the identified P300 time regions. Our observations suggest that there are two major reasons why such anomalies occurred. Firstly, as discussed in Section 5.5.4.1, certain subjects such as subject 2, 3 and 6 [or row 7-12, 13-18 and 31-36 in Fig. 6.5(a)] tends to have weaker P300 waveforms when compared to the other subjects. As a result, these EEG sessions are prone to mislabeling. Secondly, we observe that for those subjects with strong P300 ERPs, occasionally, their identified P300 time regions may also contain mislabeled segments at around 200 ms [see sessions 36, 50, 53 of Fig. 6.6(a)]. These results suggest that in these sessions, the residuals of the earlier positive P200 ERPs were not fully suppressed and thus they were falsely included in the P300 time region. To eliminate these short and meaningless time segments, we therefore constrained the segmentation results by only retaining the connected time region that covers the peak of $P(t)$. In this case, our results in Fig. 6.6(b) shows that the identified P300 time regions are better represented with less noises. A more detailed illustration on the P300 time region with respect to the trial-averaged target and non-target signals

[6]As mentioned in Section 3.3.3.2, the clustering algorithms usually treat the segmentation problem globally by by ignoring the temporal relationships between different multi-channel measurement vectors $\mathbf{x}(t)$. As a result, it causes short and meaningless time segments to arise.

is shown in Fig. 5.16 of the previous chapter.

Fig. 6.7(a) shows the P300 time regions found by the conventional mKM algorithm when our proposed cluster selection method is applied. In this study, mKM uses an average of 7.3 $\pm$ 1.9 clusters to represent the underlying ERPs from target and non-target signals. In Fig 6.7(a), the results show that the P300 time regions determined by mKM are different from our proposed algorithm. The differences are possibly caused by the different distance measures employed by both methods. By examining the P300 time regions found by mKM in Fig 6.7(a), the results indicate that these time regions suffered from a greater number of mislabeled time segments. Moreover, Fig. 6.7(b) reveals that in some sessions, the mKM segmentation results also display sign of under-segmentation when the constraint is applied to remove the short time segments.

6.3.4.2 Quantitative Examination

AUC is a common measure for evaluating the performance of a classifier [141][142]. In general, an AUC of 1 reflects perfect classification while an AUC of 0.5 is equivalent to classification by chance. In this study, the AUC[7] performance of the different combination of spatial filters and time regions are tabulated in Table 6.1.

Table 6.1: The mean classification performance (in AUC) of the spatial filter across 60 EEG sessions when the samples from different time regions of the target and non-target epochs were used to train the spatial filter.

	0–1000 ms	300–600 ms	mKM	a2KM
ICA-R[1]	0.812 $\pm$ 0.110	0.875 $\pm$ 0.097	**0.888 $\pm$ 0.093**	**0.888 $\pm$ 0.095**
cLD[2]	0.812 $\pm$ 0.106	0.874 $\pm$ 0.095	0.886 $\pm$ 0.089	**0.891 $\pm$ 0.090**
xDAWN[3]	0.832 $\pm$ 0.106	0.861 $\pm$ 0.106	0.859 $\pm$ 0.109	**0.867 $\pm$ 0.107**
Average	0.819 $\pm$ 0.107	0.870 $\pm$ 0.099	0.878 $\pm$ 0.097	**0.882 $\pm$ 0.097**

[1] ICA-R can also be trained by using the time region of 0–1000 ms, but this is not the intended method (see Section 5.2.4).
[2] conventional LDA-based LD spatial filter (See Chapter 4).
[3] Assuming that the P300 is the largest component in the signal [135][136], the eigenvector with the largest eigenvalue is taken as the spatial filter (See Chapter 3).

By examining the overall AUC performance achieved by using the entire epoch (0-1000 ms), the results show that this time region does provide a good classification performance with AUC of at least 0.81. However, when compared to other shorter and yet more representative time regions, it leads to poorer performance. From the

[7]For more information about the typical AUC range for P300 ERP classification, see [143].

results, it can be easily seen that the estimated statistics of the P300 ERP (e.g. mean spatial pattern and spatial covariance matrix) is not optimal for training a spatial filter to extract the P300 ERP. The main reason is that this time region covers all ERPs within the epochs and thus the estimated statistics about P300 are corrupted by the irrelevant ERPs. However, if a time region is carefully selected to cover the desired P300 ERP, one can observe that the overall AUC performance for all spatial filters increases substantially from 0.819 in 0–1000 ms to 0.870 in 300-600 ms.

As observed above, defining a time region that is limited to the desired ERP time region is very important. To determine whether the mKM and the proposed a2KM are able to identify the P300 ERP time region correctly, the AUC performance by the time region of 300–600 ms is taken to be the baseline for examining the AUC performance of both segmentation algorithms. Overall, the results suggested that although the time selection is performed in an automated way, both methods are able to identify the P300 time region correctly. In addition, the results from Table 6.1 also suggest that both algorithms can provide a representative P300 time region. Therefore, when the spatial filters were trained by using the samples from these identified time region, they achieve better AUC performance compared to those which use a the fixed time region. However, when comparing the conventional mKM and the proposed a2KM, our results showed that the proposed a2KM algorithm is better in that it achieves an average AUC of 0.882 ± 0.097 across all spatial filters compared to 0.878 ± 0.097 achieved by the mKM algorithm.

6.4 Experimental study on P300 speller dataset

6.4.1 Overview

The P300 speller is a kind of BCI that assists the user to spell a character through their P300 ERPs. The traditional P300 speller usually presents a 6×6 character matrix on the screen. During each presentation run, 6 rows and 6 columns of characters are highlighted, and only those row and column of characters which contains the desired character will evoke a P300. Subsequently, the task of P300 speller is to predict the users' intended character by identifying the row and column that evokes the P300.

P300 speller is another potential application for the proposed ERP segmentation

algorithm. Although many studies have been conducted on the P300 speller, investigations on the automatic time selection method for P300 ERP are rarely studied. Often, the P300 time regions that have been used are fixed and are selected to be as large as possible to ensure the P300 ERP are always captured. The fixed time regions that have been used in various studies are shown in Table 6.2.

Table 6.2: The choices of fixed time region used in various published works on P300 speller during the extraction of the target and non-target epochs

Year	Author(s)	Selected Time Region (ms)
2003	Xu *et al*. [113]	0–650
2004	Kaper *et al*. [144]	0–600
2006	Mirghasemi *et al*. [145]	0–600
2008	Hoffmann *et al*. [101]	0–1000
	Krusienski *et al*. [97]	0–800
	Rivet *et al*. [84]	0–1000
	Rakotomanonjy *et al*. [140]	0–667
2009	Chumerin *et al*. [146]	0–1000
2011	Jin *et al*. [147]	0–500
	Pires *et al*. [75]	0–1000
2013	Yin *et al*. [148]	0–800

Choosing a suitable sample size for each EEG epoch is important for the P300 speller since it determines the classification performance and also the amount of features to be used. A longer epoch segment can guarantee the P300 ERP is always captured and thus a good classification performance can be achieved. However, it is also inefficient because the extended time region will increase the features but without necessarily improving the prediction accuracy.

In this study, to demonstrate that a shorter and more representative epoch segment can achieve a good prediction accuracy, we first performed an exhaustive search by evaluating every possible combination of starting and ending time for an epoch segment. From there, we identified the epoch segments that should be extracted to achieve a good prediction accuracy in the P300 speller, in which, the result forms the basis of our evaluation. Next, the proposed a2KM algorithm is applied to the selected dataset. To examine the possibility of using the proposed a2KM algorithm for finding a reliable epoch segment for prediction, its identified P300 time regions are compared to the results obtained from the baseline exhaustive search method. Note that, unlike the work reported in Section 6.3.4.1, in this section, the ERP classification is performed in

a conventional way without involving any ERP extraction method. The main reason is to demonstrate that the proposed a2KM algorithm is a general tool that is suitable for initializing both ERP extraction and classification methods.

6.4.2 Dataset

For this study, the P300 speller dataset from BCI competition III was chosen. Similar to the dataset from BCI competition II (see chapter 5), this dataset was recorded under the same recording conditions, that is, there are 64 EEG electrodes and at a sampling frequency of 240 Hz . The only difference is that this dataset contains EEG recordings of two different subjects. For each subject, training and testing data were provided, each containing 85 and 100 characters respectively. In this dataset, for each character, a total of 15 different presentation runs were performed. As a result, there are a total of 180 responses (30 targets and 150 non-targets) collected for each individual character.

6.4.3 Classifier training and testing

Firstly, the training and testing data were bandpass-filtered[8] to between 1-10 Hz before being downsampled from 240 Hz to 20 Hz. During the training phase, a 1000 ms epoch was extracted after every stimulus onset. Using these epochs, a trial-averaged target $\mathbf{X}_{\mathrm{T}}$ and a non-target $\mathbf{X}_{\mathrm{NT}}$ response were computed and fed into the proposed segmentation algorithm to search for the P300 time region. Then, based on the outcome, those EEG measurements that were not within the time region were removed while the rest were reshaped into a one-dimensional feature vector by concatenating the EEG measurements from different channels. All 64 EEG channels were used in this study. As a result, each feature vector has a dimension of $(64N_T) \times 1$ where N_T is the number of time samples in each channel.

With the provided class label of target and non-target, a total of 15300 (12 flashes $\times$ 15 runs $\times$ 85 characters) feature vectors were obtained and fed into the BLDA classifier for training. In this study, instead of applying the ensemble SVM used by the competition winner [140], the BLDA classifier is used because of its ability to be fine-tuned automatically without any cross-validation.

[8] A combination of a forward-backward 2nd order elliptic highpass and 8th order elliptic lowpass filters.

During testing, instead of using 1000 ms EEG segment, each epoch was extracted only from the identified time region after stimulus onset. Similarly, each test epoch was reshaped into a feature vector for classification. By applying the classifier, the row and column epoch with the highest accumulated classifier score after each run were chosen to predict the character [144].

Meanwhile, for the exhaustive search method, in each iteration, the epoch segment of different starting and ending times was extracted for classifier training and testing. After examining its prediction accuracy through the BLDA classifier, the process was iterated until every possible combination of starting and ending time for epoch segment was examined.

6.4.4 Results and discussion

6.4.4.1 Conditions that an epoch segment should fulfill

For the exhaustive search method, a total of 231 different combinations of starting and ending time were tested. For Subject A, the prediction accuracy of every epoch segment that were extracted under these combinations of starting and ending times were plotted in Fig. 6.8(a) and (b). Specifically, Fig. 6.8(a) and (b) show the prediction accuracies after 5 runs and 15 runs respectively, where every non-white square corresponds to a valid epoch segment (i.e. starting time $t_1 \leq$ ending time t_2).

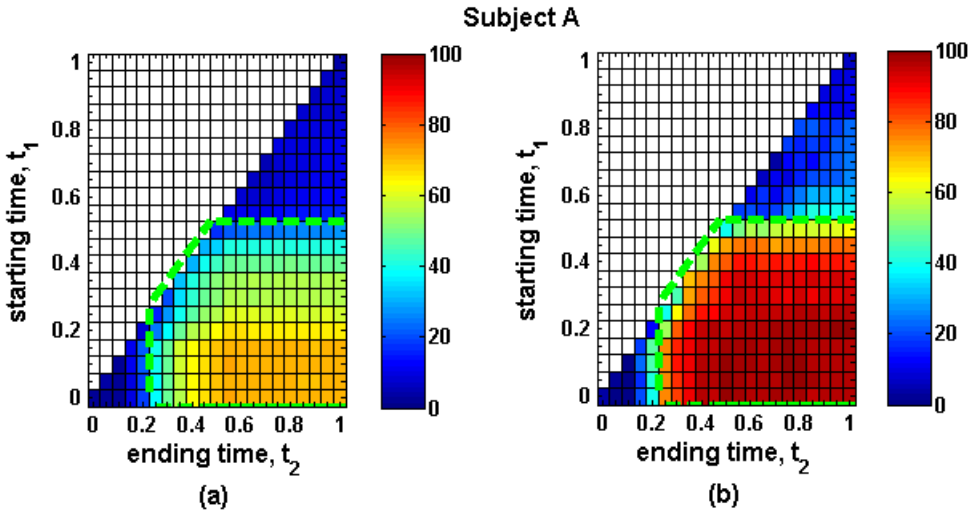

Figure 6.8: (a) Prediction accuracies after 5 runs and (b) 15 runs for different epoch segments in Subject A dataset.

Similar to the observations reported in [150], it can be seen from Fig. 6.8(a) and (b) that there are two minimum conditions that an epoch should fulfill in order to achieve an acceptable prediction accuracy in the P300 speller. By visually examining both Fig. 6.8(a) and (b), the results show that regardless of the number of epochs used to predict a character (i.e. 5 runs or 15 runs), these two conditions are (i) the starting time t_1 of an epoch which must be less than or equal to 0.50 s and (ii) the ending time t_2 of an epoch which must be greater than or equal to 0.25 s.

Firstly, by taking Fig. 6.8(b) as an example, the results indicate that those epoch segments that do not meet the two conditions (i.e. either the ending time is less than 0.25 s or the starting time is greater than 0.50 s) always yield poor performance, with average prediction accuracy less than $\sim$50%. In other words, these results imply that the measurements taken from these time range are generally less useful for prediction.

Table 6.3: The minimum, mean and maximum prediction accuracies of those epoch segments in Subject A dataset that fulfill the conditions of $t_1 \leq$0.50 s, $t_2 \geq$0.25 s and $t_1 \leq t_2$.

	min (%)	mean (%)	max (%)
5 runs	16.0	54.7	72.0
15 runs	39.0	87.6	99.0

In contrast, Table 6.3 shows the minimum, mean and maximum prediction accuracies achieved by those epoch segments that fulfill the aforementioned conditions. The results show that if the two aforementioned conditions are fulfilled, an epoch segment can achieve an average of 54.7 $\%$ and 87.6 $\%$ prediction accuracy after 5 and 15 runs respectively. However, as shown in Table 6.3, it should be noted that even if they fulfill these two conditions, an epoch segment can still yield poor performance with minimum 16 $\%$ and 39 $\%$ prediction accuracy after 5 and 15 runs respectively. This situation occurs mostly with epoch segments that do not cover significant amount of time samples between 250 and 500 ms. For example, a poor performing epoch segment may have a time region of 0–300 ms or 450–800 ms.

In other words, besides fulfilling the two conditions, an epoch segment must also cover a significant portion of EEG samples between 250 ms and 500 ms to achieve a good prediction accuracy. By examining this time region (i.e. 250–500 ms) with the trial-averaged target signal waveform from Subject A in Fig. 6.9(a), this time region

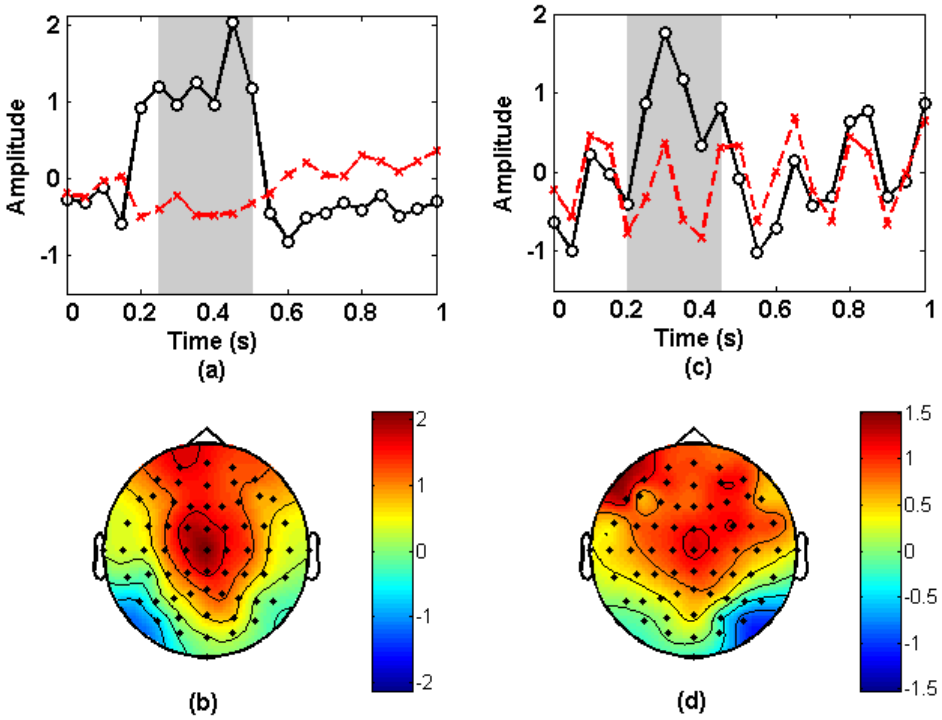

Figure 6.9: (a) Trial-averaged target (solid) and non-target (dotted) signal from Channel Pz of subject A. The grey region represent the P300 time region identified by the proposed a2KM algorithm while the green line represent. (b) The mean scalp distribution across the identified P300 time region. Similar graphs are plotted for subject B in (c)-(d).

is highly correlated with the P300 ERP time region. Thus, it also implies that the discriminative time region possibly lies within the P300 ERP time region. To find this discriminative P300 time region, another potential way is therefore to perform the proposed a2Km algorithm. Interestingly, by applying our proposed a2KM algorithm, the identified P300 time region for Subject A is also 250–500 ms, which coincides with the discriminative time region from the exhaustive search method. Our results are similar to the exhaustive search method possibly because the ERP segmentation is performed on the downsampled EEG training data. Nonetheless, from these results, it suggests that in practice, we may avoid the time-consuming exhaustive search method and adopt the proposed a2KM segmentation algorithm to provide a suggestion on the time region for extracting the epochs.

Similarly, the same observations can also be made on the Subject B dataset in Fig. 6.10(a) and (b). However, with slight differences, visual examination from the exhaustive search results suggest that the two conditions that an epoch segment should fulfill are (i) $t_1 \leq 450$ ms and (ii) $t_2 \geq 200$ ms. Table 6.4 shows the minimum, mean

and maximum prediction accuracies achieved by those epoch segments that fulfill the aforementioned conditions. For Subject B, those epoch segments that fulfill the two conditions can yield an average of 65.0 % and 87.2 % prediction accuracies after 5 and 15 runs.

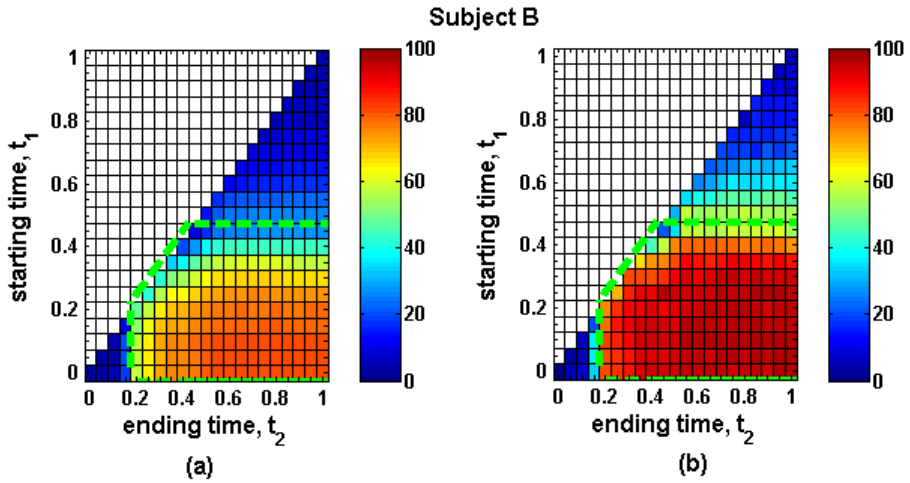

Figure 6.10: (a) Prediction accuracies after 5 runs and (b) 15 runs for different epoch segments in Subject B dataset.

Table 6.4: The minimum, mean and maximum prediction accuracies of those epoch segments in Subject B dataset that fulfill the conditions of t_1 $\leq$450 ms, t_2 $\geq$200 ms and $t_1 \leq t_2$.

	min (%)	mean (%)	max (%)
5 runs	11.0	65.0	82.0
15 runs	21.0	87.2	97.0

For Subject B dataset, based on the exhaustive search method, the results imply that the P300 ERP time region are from 200 to 450 ms. This result is similar to the P300 time region found by our proposed a2KM algorithm as shown in Fig. 6.9(b). Again, this result suggests that the proposed a2KM algorithm is potentially useful for finding the relevant epoch segment that should be included in the P300 speller.

6.4.4.2 Prediction accuracy based on the proposed a2KM algorithm

In recent years, there is an increasing trend of selecting a generous epoch segment covering more than 700 ms after stimulus onset for the P300 speller to ensure the P300 ERP is always included in the target epochs. However, as shown in the previous

Table 6.5: Subject-averaged prediction accuracy achieved by different classifiers after 5 and 15 runs for P300 speller from the BCI competition III.

Classifier	Type	5 runs(%)	15 runs(%)	features
BLDA	a2KM	60.5	90.0	384
	a2KM $\pm$ 50 ms	71.0	94.5	512
	a2KM $\pm$ 100 ms	**76.0**	**96.5**	**640**
	a2KM $\pm$ 150 ms	76.0	96.0	768
BLDA	fixed (0–650 ms)	76.0	97.0	896
BLDA	fixed (0–1000 ms)	76.5	96.5	1344
BLDA	exhaustive search	77.0	98.0	928/640[2]
sSVM[1]	fixed (0–667 ms)	69.5	96.5	896
eSVM[1]	fixed (0–667 ms)	73.5	96.5	896

[1] The prediction accuracy of single SVM (sSVM) and ensemble SVM (eSVM) as reported by the BCI competition III winner in [140].
[2] The best epoch segment for subject A and B after 5 runs are 150–950 ms (72%) and 100–650 ms (82%) respectively. With slight difference, the best epoch segment for subject A and B after 15 runs are 150–500 ms (99%) and 100–650 ms (97%) respectively.

section, the results from the exhaustive search method suggest that the time regions that cover the signal samples outside the 200 ms and 500 ms interval are generally less useful for prediction. Based on the literature, these observations are reasonable mainly because the P300 ERP appears less frequently outside this time interval [29]. For this reason, although a longer epoch segment guarantees a good prediction accuracy, the method can be inefficient. This is mainly because a larger epoch segment may include the irrelevant time regions and subsequently increase the number of features[9] without necessarily improving the prediction accuracy. Thus, an intuitive idea to strike a balance between the prediction performance and the number of features is to choose an epoch segment that is sufficiently wide enough to cover the P300 time region.

Table 6.5 shows the average prediction accuracy achieved when using the time region provided by the proposed a2KM algorithm. As a baseline comparison, the table also includes the performance achieved by the BCI competition winner who uses ensemble SVM[10] (eSVM) with a time region of 0-667 ms [140]. Besides that, since our prediction is based on BLDA classifier, the performance of BLDA with epochs taken from 0–650 ms and 0–1000 ms were also included. First, when the epoch segment

[9]The number of features is an important factor that affects the time required to train the classifier, the amount of stored data and the computational load required before an epoch can be used for prediction.
[10]This approach involves the generation of 17 SVM classifiers that are trained on different partitions of the training data.

is taken from 0 to 650 ms, both eSVM and BLDA achieve almost identical prediction accuracy, but with BLDA performed slightly better than eSVM. Besides that, for BLDA, the results also show that there is no significant performance difference for using the epoch segments taken either from 0–650 or 0–1000 ms. This suggests that the extended time region does not necessarily provide better performance.

From the table, it can be seen that when using the P300 time region found by the proposed a2KM algorithm, the average prediction accuracy for both Subjects A and B is 90%. Although this accuracy is 6.5% lower compared to the eSVM, the amount of features extracted from the identified P300 time region is only 384, which is 512 features less than the one used by the eSVM in [140]. Thus, this result implies that most of the discriminative features are concentrated within the estimated P300 time region. By expanding the width of the estimated time region 100 ms to the left and 100 ms to the right of the time axis, it is found that the expanded time region can help BLDA achieving the prediction accuracy similar to the eSVM in [140]. Most importantly, the expanded time region only requires an average of 640 features which is still 256 features less than the one used in the eSVM.

6.5 Conclusion

P300 is one of the important ERPs that have been extensively used in both ERP studies and BCI applications. The importance of identifying the P300 time region can be observed in the two different experiments conducted in this chapter. In the first experiment that involves the oddball paradigm, the result shows that the generation of a reliable P300 spatial filter relies heavily on the P300 time region. A more representative P300 time region is always necessary to ensure a quality P300 extraction. In the second experiment with the P300 speller dataset, the results showed that the accuracy of the P300 speller is governed by the time region used for P300 detection. Ideally, an epoch segment should encompass a significant portion of the P300 time region since they represent the most discriminative time region in the P300 speller.

In practice, determining the P300 time region at trial-averaged signal level remains difficult since P300 ERPs normally varies with subjects and experimental conditions. Thus, one of the most effective ways of determining the P300 time region accurately

is through solving the ERP segmentation problem. For this reason, in this chapter, a new segmentation algorithm namely a2KM that combines signal-conditioning and the clustering technique is developed. Based on our study, the proposed a2KM algorithm is able to identify the P300 time region autonomously and adaptively. Therefore, when combined with the existing ERP extraction methods, not only a good extraction performance can be achieved, a fully-autonomous P300 extraction framework can also be obtained. In addition, when compared to the conventional mKM algorithm, the proposed algorithm does not require any re-estimation on the number of clusters used to represent the underlying ERPs. Thus, this also makes it a more computationally efficient and practical tool for real-time applications. On the other hand, the proposed a2KM algorithm is also a potential tool for identifying the P300 time region which is necessary to achieve a good prediction accuracy in the P300 speller.

Chapter 7

Conclusion and Future Works

7.1 Summary

Today's technological advancement has offered us the ability to measure EEG signals simultaneously from multiple sites of the scalp. In turn, this provides us with new opportunities to study both the spatial and temporal characteristic of ERPs. To analyze a specific ERP without interference from other ERPs and background noises, this thesis explored new ways of extracting the ERP-of-interest at single-trial level by utilising the signals from multiple EEG channels.

In Chapter 2, the basics of EEG/ERP were introduced and the characteristics of the ERPs were explained. Current literature suggests that besides polarity, every ERP tends to present itself as a unique scalp distribution that spans across a specific time region. This finding was found to be highly relevant and has been exploited throughout this thesis.

In Chapter 3, the existing single-trial multi-channel ERP extraction methods were reviewed. In general, the existing multi-channel extraction methods can be divided into data-independent and data-driven methods. Each method has its own merits and drawbacks. Most data independent methods are designed specifically for ERP extraction. However, their strong assumption on the scalp distribution and the temporal waveform of the ERP-of-interest is less desired in practice since these ERP characteristics may vary between subjects with experimental conditions. In contrast, although data-driven methods can avoid such assumptions, most of them cannot extract the ERP-of-interest directly. Often, human intervention is required to complete the extraction. As a result,

the extraction procedure is usually subjective and not replicable.

Accordingly, to address the aforementioned issues, the main contribution of this thesis is to develop new data-driven extraction methods that extract the ERP-of-interest directly without human intervention.

In Chapter 4, a new LD method namely ENE-LD was developed based on the concept that each ERP has its unique scalp distribution that appears over a particular time region. To perform the extraction, ENE-LD is trained to learn the differences in the scalp distribution between the ERP and non-ERP time regions. The simulation study showed that the method is able to differentiate and extract the desired ERP-of-interest in the presence of multiple ERPs. Consequently, the extracted signal is useful for examining the peak latency, scalp distribution and dynamics of the ERP at the single-trial level. However, a closer inspection showed that the non-linear operation (i.e. zero-ing the negative part of the signals) has caused the extracted signal to exhibit an inevitable mismatch with the actual ERP waveform. As such, it introduces bias when used to estimate the peak amplitude. Despite that, for general applications, the results showed that the ENE-LD is still useful since it can extract weak ERPs (e.g. P100 and N170) which is not possible with the conventional LD methods.

In Chapter 5, to overcome the problem that the ENE-LD method will always exhibit a mismatch between the extracted and the actual ERP, one-unit ICA-R is proposed. One-unit ICA-R is an extension to the traditional ICA method. Therefore, it inherits the ICA's well known ability to recover the EEG source signals. The main advantage of the one-unit ICA-R is its ability to recover the ERP-of-interest directly through the guidance of a reference signal. However, before the extraction can be performed successfully, a new reference signal design is required. The main challenge is to build a reference signal that resembles the actual ERP source signal. Based on our examination, the signal extracted by the proposed ENE-LD method is useful since it is able to approximate closely the trial-to-trial amplitude, latency and width variations of the ERP-of-interest. Thus by combining both ICA-R and ENE-LD, a more accurate ERP waveform can be extracted. Evaluations with the real P300 datasets revealed that the proposed one-unit ICA-R can extract the P300 ERP successfully without any human intervention. In addition, it also outperformed the traditional ICA methods in terms of computational efficiency and extraction quality.

As discussed in Chapter 6, ERP time region is important for initializing most ERP extraction methods (e.g. the proposed one-unit ICA-R, xDAWN, cLD etc.). In practice, this time region is usually obtained from the trial-averaged signal. In Chapter 6, a new ERP segmentation algorithm, a2KM, is proposed specifically to identify the P300 ERP time region. By finding the differences between trial-averaged target (P300) and non-target (non-P300) signals, a2KM is able to determine the P300 time region adaptively and autonomously. Evaluations using real ERP datasets suggest that when the proposed a2KM is applied together with an ERP extraction method, not only a more reliable P300 ERP can be extracted, a fully-autonomous P300 extraction framework can also be obtained. Most importantly, this study demonstrates that (i) how ERP segmentation problem can be addressed by the pattern recognition approach, and (ii) how its solution to the ERP segmentation problem can be potentially used to initialize an ERP extraction method. Additional tests also suggested that the proposed a2KM is a potential tool for identifying the P300 time region in the P300 speller.

7.2 Future Works

1. **Choices of classifier for the ENE-LD method**

   ENE-LD is a method that utilises the classifiers' discriminative projection vector and threshold to perform the extraction. In this thesis, as a proof of concept, a simple binary linear classifier namely LDA was employed. However, to ensure a more reliable ERP extraction, other binary linear classifiers which are well-known for their robustness and also superior discriminativity such as BLDA and SVM are potential candidates to replace the LDA method.

2. **The effect of imbalanced data during ENE-LD training**

3. **Choices of the $G(y)$ and $\varepsilon(y, r)$ for the one-unit ICA-R**

   In the literature, the three popular non-quadratic convex function $G(y)$ that can be used for estimating the negentropy [122] are: (i) $G_1(y) = \log\{\cosh(y)\}$, (ii) $G_2(y) = \exp(-\frac{y^2}{2})$ and (iii) $G_3(y) = y^4/4$. In this thesis, the general purpose function $G_1(y)$ was chosen. However, to optimize the one-unit ICA-R, an

investigation can be conducted to find the optimum function $G(y)$ for ERP application. Similarly, another popular choice for the closeness measure $\varepsilon(y, r)$ is by finding the correlation between the extracted signal and the reference signal, i.e. $\varepsilon(y, r) = -\mathbb{E}\{yr\}$. Further investigation is still required to understand the impact a closeness measure has on the performance of the one-unit ICA-R.

4. **A better reference signal design for one-unit ICA-R**

   In Chapter 5, our simulation study shows that a good reference signal should approximate the actual source signal as close as possible. Although our results demonstrated that the ENE-LD based reference signal is a feasible solution to guide the one-unit ICA-R, finding a better design for the reference signal can be an interesting addition to this thesis. One possible solution is to use the extracted signal as the new reference signal in an iterative manner. However, this gives rise to the question of whether the method would converge and whether the new reference signal would give a significant improvement in extraction.

5. **A better ERP segmentation algorithm**

   Besides P300 ERP, the proposed a2KM algorithm is potentially useful for determining the time regions of those ERPs which have large differences between their target and non-target trials (e.g. N400, Mismatch Negativity, Error-Related Potential etc.). Thus, an interesting extension to this thesis is to examine its performance on the aforementioned ERPs. However, for other type of ERPs, further investigation is still required.

6. **Neurofeedback and neurotherapy applications**

   A promising future direction of this thesis is the application of ERP in the neurofeedback and neurotherapy systems. In current ERP-based BCI, the ERP is only used in a binary control system although its peak amplitude and latency often provide richer information about the user's ability to perform a specific task (i.e. face recognition). As observed, to fully utilise these information, an ERP extraction method that can recover the desired ERP waveform directly and automatically is a stepping stone for the neurofeedback and neurotherapy systems. As demonstrated in this thesis, the proposed extraction methods hold all these promising characteristics and therefore an investigation towards this direction

should be conducted.

7. **Investigation based on low-cost EEG devices**

   Over the last few years, increasing number of low-cost EEG devices (e.g. Emotiv) are introduced and foreseen as potential self-diagnostic tools in the near future. However, these EEG devices usually come with less number of EEG channels, lower resolution in analog-to-digital converter, lower grade of EEG electrodes and sampling rate. Since the effect these limitations have on the ERP extraction methods remains largely unexplored, evaluating the impact of these low-cost EEG devices on the proposed ERP methods can be seen as an interesting extension to this thesis.

Appendix A

Other Linear Classifiers

Two popular linear classifiers such as Support Vector Machine (SVM) and Bayesian LDA (BLDA) were employed in the P300-BCI datasets used in Chapter 5 and 6 respectively. A brief summary on these techniques are provided in the following sections.

A.1 Support Vector Machine

SVM is a linear discriminative classifier whose objective is to find the 'best' decision hyperplane that not only separates both classes, but also maximizes the distance between the hyperplane and the samples closest to the hyperplane (the distance is also known as margin) [151][152].

Let $\mathbf{v}_i$ denotes the training samples with class labels $q_i \in \{+1, -1\}$. Given the discriminative hyperplane $\mathbf{u}$ and bias term b, assuming the samples of both classes can be linearly separated by the margins parallel to the hyperplane, the projected samples can be written as [151]:

$$\mathbf{u}^\top \mathbf{v}_i + b \geq +1 \quad \text{if } q_i = +1 \tag{A.1}$$

$$\mathbf{u}^\top \mathbf{v}_i + b \leq -1 \quad \text{if } q_i = -1 \tag{A.2}$$

The two inequalities above can be combined into one inequality as:

$$q_i(\mathbf{u}^\top \mathbf{v}_i + b) \geq 1 \tag{A.3}$$

The distance between two corresponding margins are then given as:

$$\gamma(\mathbf{u}, b) = \min_{\mathbf{v}|q=+1} \frac{\mathbf{u}^\top \mathbf{v}}{\|\mathbf{u}\|} - \max_{\mathbf{v}|q=-1} \frac{\mathbf{u}^\top \mathbf{v}}{\|\mathbf{u}\|} = \frac{2}{\|\mathbf{u}\|} \tag{A.4}$$

Maximizing the above distance is also similar to minimizing $\|\mathbf{u}\|$. Thus, for binary separable classes problem, the optimization problem for SVM can be formulated as [151]:

$$\min_{\gamma,\mathbf{u},b} \quad \frac{1}{2}\|\mathbf{u}\|^2 \quad \text{s.t} \quad q_i(\mathbf{u}^\top \mathbf{v}_i + b) \geq 1 \quad \text{for all } i \tag{A.5}$$

By solving the above problem as the generalized Lagrangian, for any test vector $\tilde{\mathbf{v}}$, the resultant classification rule for SVM is given as [151]:

$$\begin{aligned} q &= \text{sign}(\mathbf{u}^\top \tilde{\mathbf{v}} + b) \\ &= \text{sign}\left[\left(\sum_{i=1}^{N_s} \alpha_i q_i \mathbf{v}_i \right)^\top \tilde{\mathbf{v}} + b \right] \end{aligned} \tag{A.6}$$

where α_i is the Lagrangian multiplier. Those vectors with nonzero coefficient α_i are called support vectors. The bias term b can be calculated as:

$$b = -\frac{1}{2}\left(\min_{\mathbf{v}_i|q_i=+1} \mathbf{u}^\top \mathbf{v}_i - \max_{\mathbf{v}_i|q_i=-1} \mathbf{u}^\top \mathbf{v}_i \right) \tag{A.7}$$

In some cases, the 'best' hyperplane which gives the least classification error may not provide a clean margin between two classes. Or in other words, some outliers may fall within the margins between two classes. For these type of linearly inseparable cases, a soft margin is therefore introduced by adding the slack variable ξ_i into (A.5). As a result, the optimization problem for the soft-margin SVM can be formulated as [151]:

$$\begin{aligned} \min_{\gamma,\mathbf{u},b} \quad & \tfrac{1}{2}\|\mathbf{u}\|^2 + C\textstyle\sum_{i=1}^{N_c} \xi_i \\ \text{subject to} \quad & q_i(\mathbf{u}^\top \mathbf{v}_i + b) \geq 1 - \xi_i \\ & \xi_i \geq 0, \qquad \text{for all } i \end{aligned} \tag{A.8}$$

where C is the penalty parameter and N_c is the number of training samples in each class c.

A.2 Bayesian Linear Discriminant Analysis

Bayesian LDA (BLDA) is a variant of linear regression method [94] which has been adapted and introduced to the BCI application by [101]. The adaptation is possible

mainly because under specific condition, LDA and linear regression shares similar solutions. A proof of their relationship can be found in [94]. The basic idea behind Bayesian regression and consequently behind BLDA is to interpret the objective function of ridge regression as the exponent of a probability distribution from the exponential family. As such, the posterior distribution of the projection vector $\mathbf{u}^*$ and the predictive distribution of the estimated value $\hat{z}$ given any new input feature vector can be estimated.

Let N denote the total number of training samples, $\mathbf{v}^*$ denotes the modified feature vector where $\mathbf{v}^*$ has an additional feature of value '1' in the last element of the vector so that the bias term b is integrated in the last element of $\mathbf{u}^*$. $\mathbf{V} = [\mathbf{v}_1^*, \mathbf{v}_2^*, ..., \mathbf{v}_N^*]$ denotes the horizontal concatenation of all the modified features vectors while $\mathbf{z} = [z_1, z_2, ...z_N]$ represents the 'class labels' for the feature vectors where instead of $\{+1, -1\}$, the value for each class is given as $\{+\frac{N}{N_+}, -\frac{N}{N_-}\}$ with N_+ and N_- represent the total number of training samples in class $+1$ and -1 respectively. By using $\mathbf{D}$ to represent the training samples and its class labels as the pair $\{\mathbf{V}, \mathbf{z}\}$, the posterior distribution of $\mathbf{u}^*$ can be computed using Bayes rule as :

$$p(\mathbf{u}^*|\beta, \alpha, \mathbf{D}) = \frac{p(\mathbf{D}|\beta, \mathbf{u}^*)p(\mathbf{u}^*|\alpha)}{\int p(\mathbf{D}|\beta, \mathbf{u}^*)p(\mathbf{u}|\alpha))d\mathbf{u}^*} \tag{A.9}$$

and the probability distribution of the predicted value y given any new input feature vector $\mathbf{v}^*$ is given as:

$$p(z|\beta, \alpha, \mathbf{v}^*, \mathbf{D}) = \int p(z|\beta, \mathbf{v}^*, \mathbf{u}^*)p(\mathbf{u}^*|\beta, \alpha, \mathbf{D})d\mathbf{u}^* \tag{A.10}$$

where β and α are the unknown hyperparameters that has to be estimated iteratively.

Most importantly, by assuming $p(\mathbf{D}|\beta, \mathbf{u}^*)$ and $p(\mathbf{u}^*|\alpha)$ are Gaussian and only the mean values of $p(\mathbf{u}^*|\beta, \alpha, \mathbf{D})$ and $p(z|\beta, \alpha, \mathbf{v}^*, \mathbf{D})$ are of interest, the resultant projection vector $\mathbf{u}^*$ can be estimated as:

$$\mathbf{u}^* = \beta(\beta\mathbf{V}\mathbf{V}^\top + \mathbf{I}'(\alpha))^{-1}\mathbf{V}\mathbf{z} \tag{A.11}$$

where $\mathbf{I}'(\alpha)$ is a diagonal matrix with all its diagonal elements equal to α except the last diagonal element which is a very small positive value. Meanwhile, the predicted value z of any test vector $\tilde{\mathbf{v}}$ is simply given as:

$$z = \mathbf{u}^{*\top}\tilde{\mathbf{v}}^* \tag{A.12}$$

More detail information on BLDA can be found in [101].

Appendix B

P300 Speller

B.1 How the P300 speller works

In general, the row/column based P300 speller contains a predefined set of alphanumeric characters, which are arranged in a 6×6 matrix as shown in Fig. B.1(a). As shown in Fig. B.1(b), in each presentation run, 6 rows and 6 columns of characters are randomly highlighted without repeating. During the presentation, the user is required to focus on one desired character. As such, every time whenever the desired character is highlighted, a P300 ERP will be elicited.

From the P300 speller perspective, the main objective for the P300 speller is to identify the row and column that elicit the P300 ERP so that the user's desired character can be spelt correctly. At the low level, the process of predicting the user's desired character usually involves three major steps as shown in Fig. B.1(c). These three major steps are performed as follows:

1. *Epoch extraction*

   After each presentation run, 12 EEG epochs are extracted according to the stimulus onset from each highlighted row and column. After that, every epoch goes through feature extraction process and usually transformed into feature vectors before fed into a classifier.

2. *Classification*

   During classification, each epoch is evaluated and given a classifier score.

3. *Character Prediction*

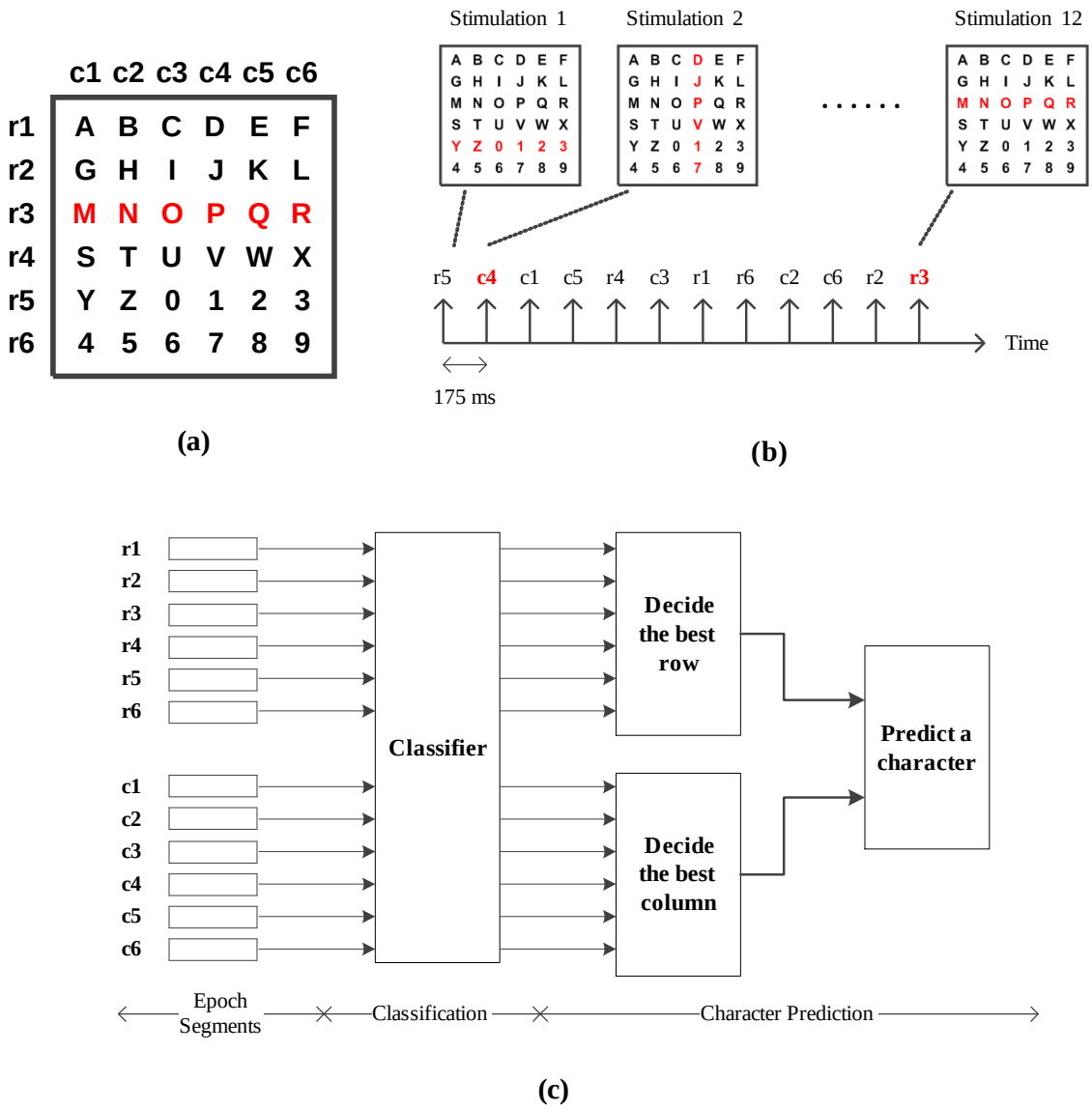

Figure B.1: (a) A traditional row/column based P300 speller, (b) The process of how each presentation run is performed in the traditional P300 speller, (c) The low-level processes which are involved to predict a character after each presentation run.

Based on the given scores, the P300 speller is then required to decide the row and column epochs that are mostly likely to contain the P300 ERP. In this case, assuming that only one presentation run is involved, the selection can be performed easily by selecting the row and column epochs which contain the best classifier score. Subsequently, their corresponding row and column are used to predict the user's intended character.

In practice, sometimes, it can be difficult to predict the user's desired character by running only one presentation run. Thus, multiple presentation runs are normally

required to improve the prediction accuracy. In cases where mutliple runs were performed, the two common approaches used to select the best row and column are either: (i) to average each row/column epoch across multiple runs before evaluating them using a classifier [113][130], or (ii) to decide based on the accumulated classifier score for each row/column epoch from every run [98][140][144].

Also, for the above reason, in the P300-BCI literature, the performance of an ERP classification system is usually evaluated by examining the prediction accuracy obtained after every run (or selected number of runs).

B.2 Data structure for BCI competition dataset II

The P300 speller dataset provided from BCI competition dataset II contains EEG recordings collected from one subject under three different sessions (i.e Session 10, 11 and 12). The data structure for this dataset is as follows:

Table B.1: Data structure of the BCI competition II P300 speller dataset

Session	Part	Target Word	Set
10	1	CAT	Training
	2	DOG	
	3	FISH	*Test1*
	4	WATER	
	5	BOWL	
11	1	HAT	
	2	HAT	
	3	GLOVE	
	4	SHOES	
	5	FISH	
	6	RAT	
12	1	FOOD	*Test2*
	2	MOOT	
	3	HAM	
	4	PIE	
	5	CAKE	
	6	TUNA	
	7	ZYGOT	
	8	4567	

As shown in Table B.1, Session 10, 11 and 12 contains 5, 6 and 8 parts respectively where different words were spelt in each part. For every character in each part, 15 presentation runs were performed. As a result, for each character, a total of 180 epochs (12 stimulations $\times$ 15 runs) can be extracted.

For evaluation in Chapter 5, all the parts from different sessions are re-arranged and concatenated into three different sets (i.e. Training, *Test1* and *Test2*) as shown in Table B.1.

Appendix C

Additional Results from Chapter 5

In Chapter 5, an analysis was performed to evaluate the median latency of P300 ERP under different visual stimuli conditions by using our proposed one-unit ICA-R method. For each session, the signal extracted from every target epoch was smoothed with a forward-backward moving-average filter using 20 ms time window. This was then followed by a simple peak detection to search for the maximum point in every target epoch. Fig. C.1 shows the peak latency identified from the target epochs of every EEG session. From the figure, the results show that most P300 peaks occur between 300 ms and 600 ms. Table C.1 shows the median latency of each individual sessions (without sessions from subject 2, 3 and 6).

Table C.1: Median latency estimated under different visual stimuli conditions (without sessions from subject 2, 3 and 6.)

Subjects	**Median Latency (ms)**					
	2d00	**2d30**	**2d70**	**3d00**	**3d30**	**3d70**
S1	345	366	403	348	368	438
S4	375	396	406	403	391	407
S5	491	446	476	430	503	439
S7	371	409	425	364	401	444
S8	414	383	408	359	383	418
S9	366	400	416	394	365	426
S10	375	379	436	384	387	411

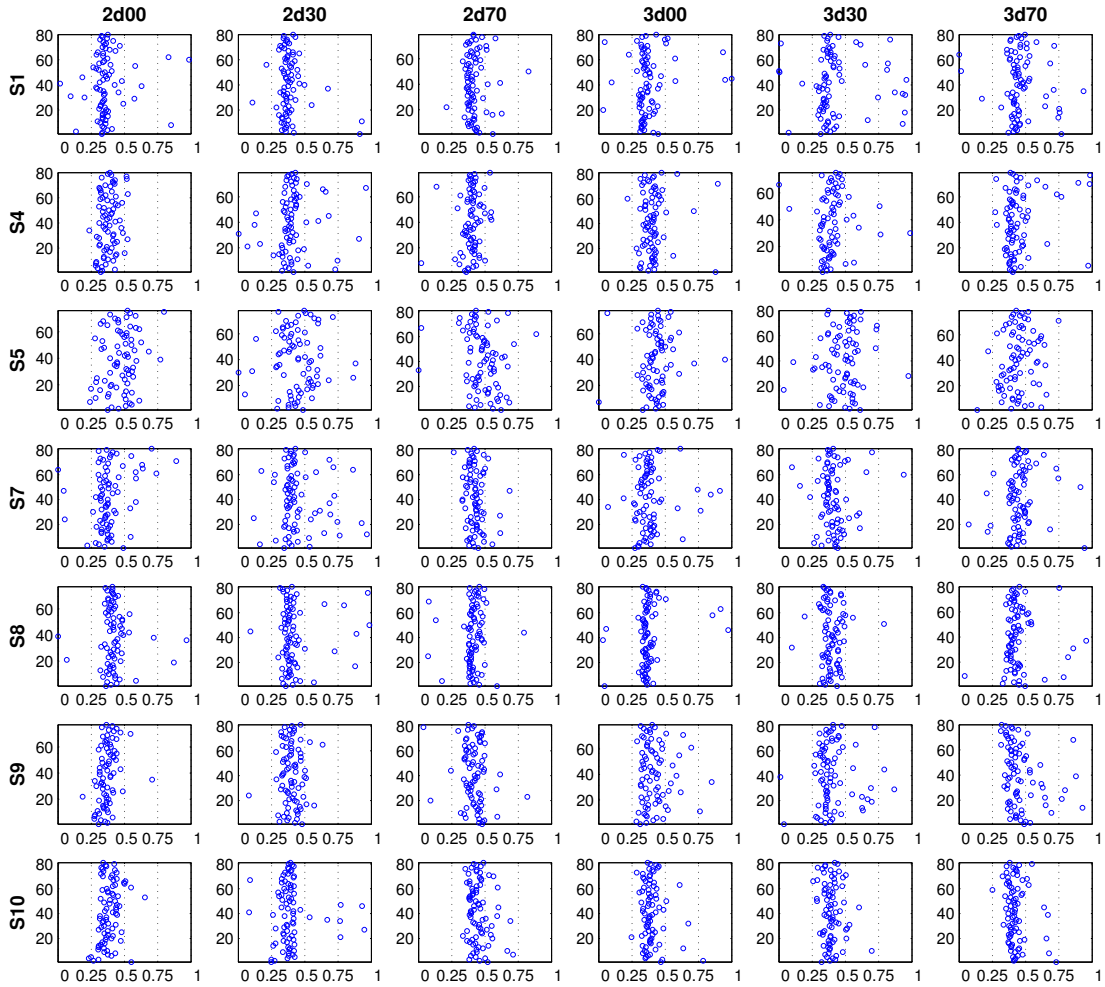

Figure C.1: P300 peak latency identified from the target epochs of each session after applying one-unit ICA-R (without sessions from subject 2, 3 and 6) where y-axis and x-axis represents the trial number and time in second respectively.

Appendix D

Additional Results from Chapter 6

In Chapter 6, different time regions were compared for their qualities based on the ERP classification performance (in AUC). The following figures show the detail AUC value for different extraction methods (i.e. one-unit ICA-R, conventional LD and xDAWN) when different time regions (i.e. 0–1000 ms, 300-600 ms, mKM and a2KM) were used.

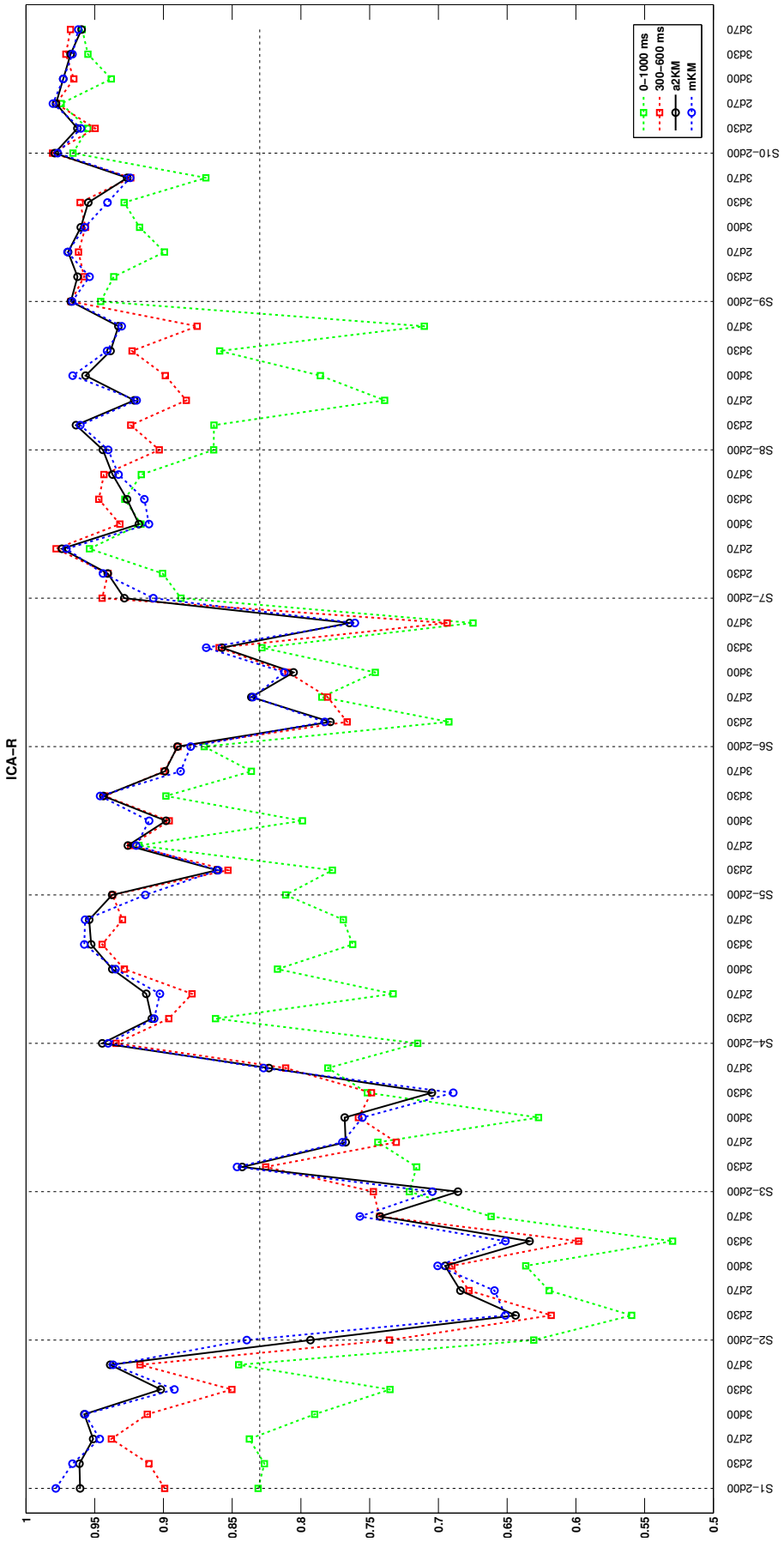

Figure D.1: AUC performance for different time regions when used together with one-unit ICA-R across different EEG sessions. (Note that: As a reference, the horizontal black dotted line represents the best AUC value achieved in IEEE MLSP'09 competition for a different type of P300 stimulus presentation.)

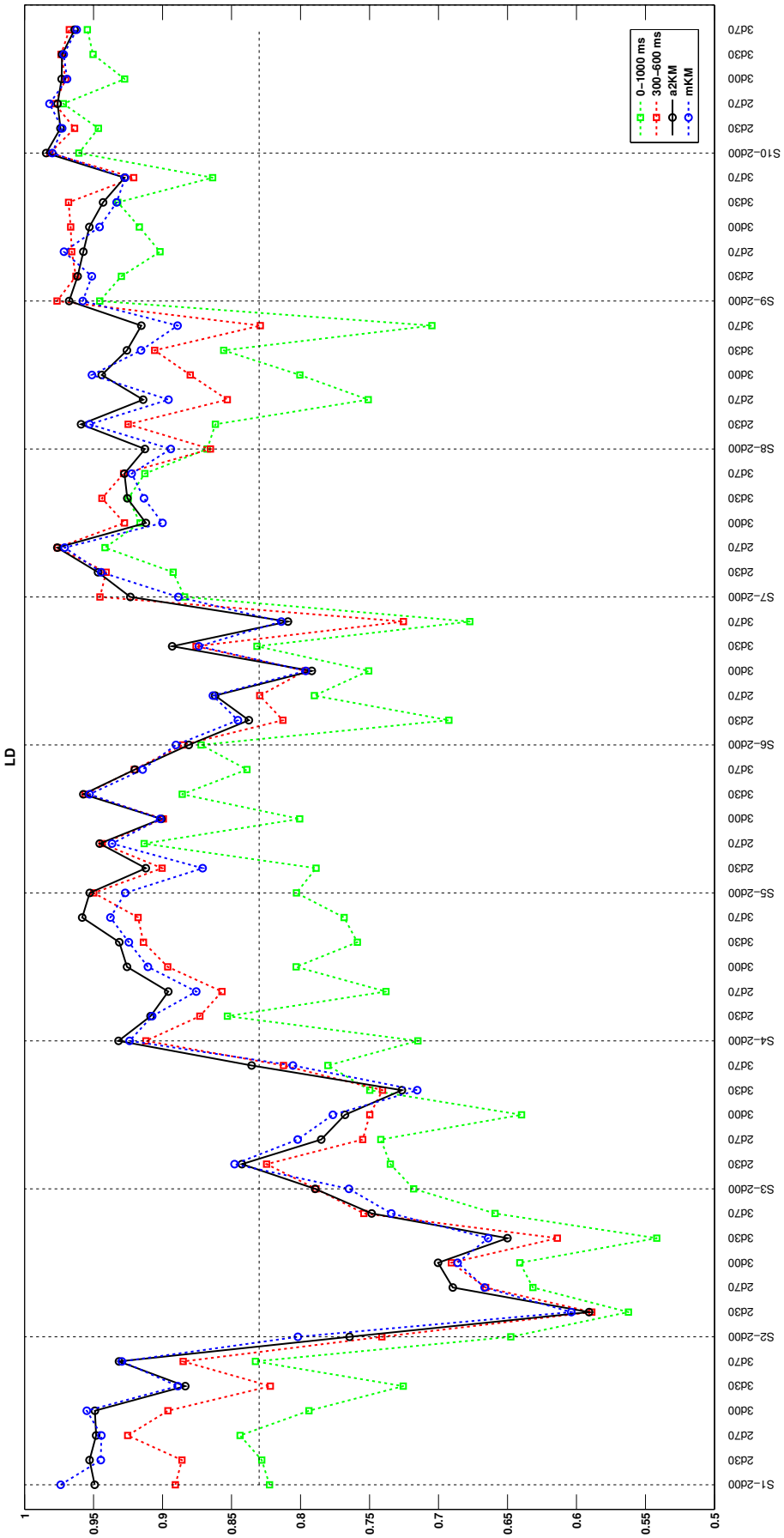

Figure D.2: AUC performance for different time regions when used together with the conventional LD (LDA-based) across different EEG sessions. (Note that: As a reference, the horizontal black dotted line represents the best AUC value achieved in IEEE MLSP'09 competition for a different type of P300 stimulus presentation.)

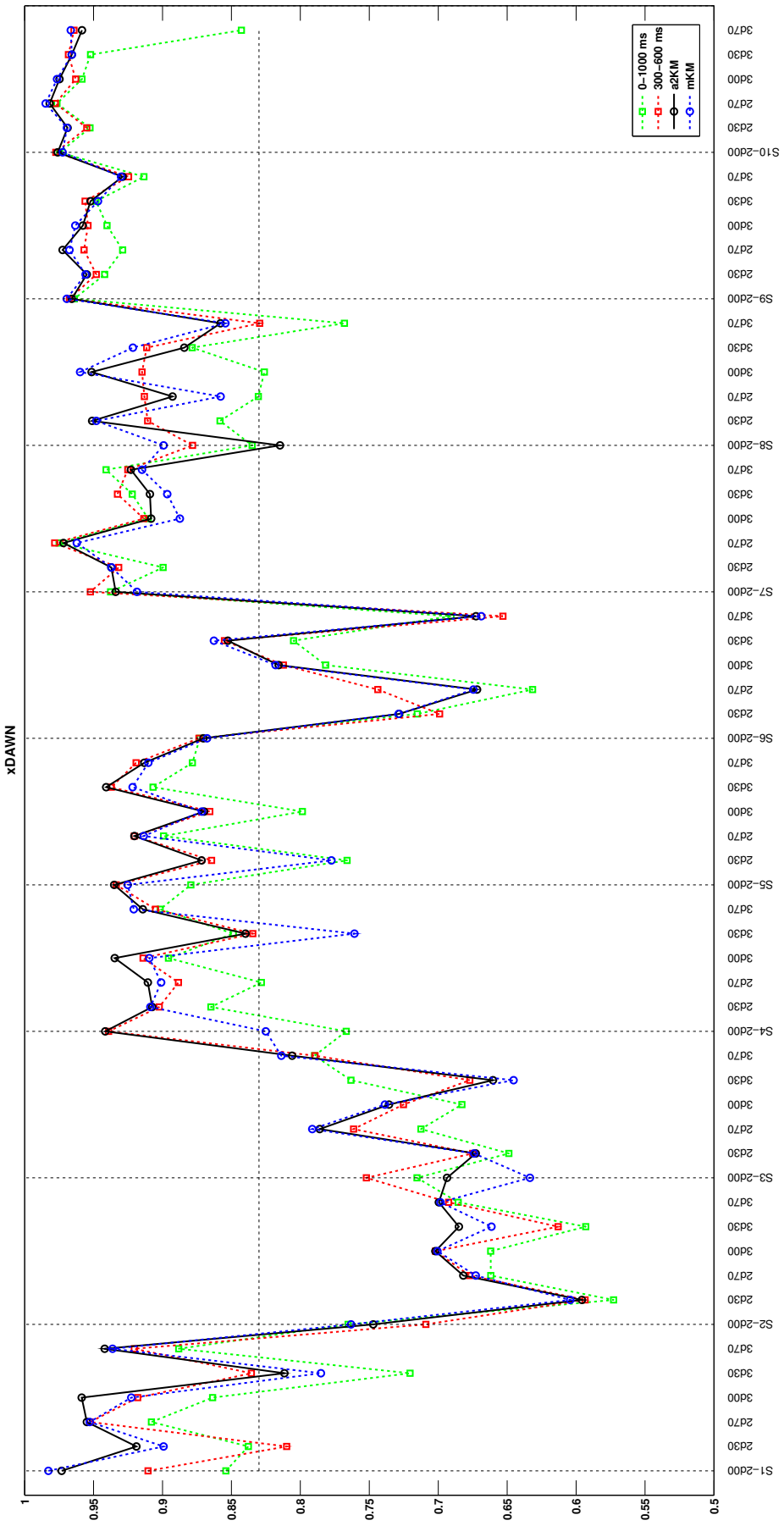

Figure D.3: AUC performance for for different time regions when used together with xDAWN across different EEG sessions. (Note that: As a reference, the horizontal black dotted line represents the best AUC value achieved in IEEE MLSP'09 competition for a different type of P300 stimulus presentation.)

References

[1] H. Berger, "Ueber das Elektrenkephalogramm des Menschen", *Archives fur Psychiatrie Nervenkrankheiten*, vol. 87, 1929, pp. 527-570.

[2] J. D. Kropotov, "Introduction: Basic Concepts of QEEG and Neurotherapy", *Quantitative EEG, Event-Related Potentials and Neurotherapy*, San Diego: Academic Press, 2009, pp. xxxi–iviii.

[3] S. Tong and N. V. Thakor, *Quantitative EEG Analysis: Methods and Clinical Applications* (Engineering in Medicine and Biology Series), Norwood: Artech House, 2009.

[4] S. Sanei and J. A. Chambers, *EEG Signal Process.*, West Sussex: Wiley, 2007.

[5] S. J. Luck, *An Introduction to the Event Related Potential Technique*, Cambridge: MIT Press, 2005.

[6] M. G. H. Coles and M. D. Rugg, "Event-related brain potentials: an introduction", in *Electrophysiology of Mind: Event-related Brain Potentials and Cognition*. Oxford: Oxford University Press, 1996, pp. 1–26.

[7] M. Fabiani, G. Gratton and K. D. Federmeier, "Event-related potential - methods, theory, and applications", in *Handbook of Psychophysiology*, 3rd Ed, J. T. Cacioppo, L. G. Tassinary and G. Berntson Eds. Cambridge: Cambridge University Press, 2007, pp. 85–119.

[8] H. Davis, P. A. Davis, A. L. Loomis, E. N. Harvey and G. Hobart, "Electrical reactions of the human brain to auditory stimulation during sleep", *J. Neurophysiology*, vol. 2, pp. 500-514, Apr. 1939.

[9] B. I. Turetsky, J. Raz and G. G. Fein, "Estimation of trial-to-trial variation in evoked potential signals by smoothing across trials", *Psychophysiology*, vol. 26, no. 6, pp. 700-712, Nov. 1989.

[10] J. Möcks, T. Gasser, D. T. Pham and W. Köhler, "Trial-to-trial variability of single potentials - methodological concepts and results", *Int. J. Neuroscience*, vol. 33, pp. 25-32, Mar. 1987.

[11] C. D. McGillem and J. I. Aunon, "Measurements of signal components in single visually evoked brain potentials", *IEEE Trans. on Biomedical Eng.*, vol. BME-24, no. 3, pp. 232–241, May 1977.

[12] C. A. Nelson and J. P. McCleery, "Use of Event-Related Potentials in the Study of Typical and Atypical Development", *J. the Amer. Academy of Child and Adolescent Psychiatry*, vol. 47, no. 11, pp. 1252-1261, Nov. 2008.

[13] J. R. Wolpaw *et al.*, "Brain-computer interfaces technology: a review of the first international meeting", *IEEE Trans. on Rehabilitation Eng.*, vol. 8, no. 2, pp. 767–791, Mar. 2002.

[14] S. Makeig, C. Kothe, T. Mullen, N. B. Shamlo, Z. Zhang and K. Kreutz-Delgado, "Evolving signal processing for Brain-Computer Interfaces", *Proceedings of the IEEE 100 (Centennial-Issue)*, pp. 1567–1584, 2012.

[15] P. L. Nunez and R. Srinivasan, *Electric fields of the brain: The Neurophysics of EEG*, 2nd ed. New York: Oxford University Press, 2006.

[16] F. Lopes da Silva, "EEG: Origin and measurement", in *EEG - fMRI: Physiological Basis, Technique, and Applications*, C. Mulert and L. Lemieux, Eds. Berlin: Springer, 2010, pp. 19–38.

[17] Neuron [Online]. Available: http://en.wikipedia.org/wiki/Neuron.

[18] D. A. Pizzagalli, "Electroencephalography and high-density electrophysiological source localization", in *Handbook of Psychophysiology*, 3rd Ed, J. T. Caciopp o, L. G. Tassinary and G. Berntson Eds. Cambridge: Cambridge University Press, 2007, pp. 56–84.

[19] G. Buzsáki, C. A. Anastassiou and C. Koch, "The origin of extracellular fields and currents - EEG, ECoG, LFP and spikes", *Nature Reviews Neuroscience*, vol. 13, no. 6, pp. 407–420, May 2012.

[20] H. Hallez *et al.*, "Review on solving the forward problem in EEG source analysis", *J. Neuroengineering and Rehabilitation*, vol. 4, no. 46, Nov. 2007. doi: 10.1186/1743-0003-4-46

[21] G. F. Woodman, "A brief introduction to the use of event-related potentials in studies of perception and attention", *Attention, Perception & Psychophysics*, vol. 72, no. 8, pp. 2031–2046, Nov. 2010.

[22] C. C. Duncan *et al.*,"Event-related potentials in clinical research: Guidelines for eliciting, recording, and quantifying mismatch negativity, P300, and N400", *Clinical Neurophysiology*, vol. 120, no. 11, pp. 1883–1908, Nov. 2009.

[23] H. Pratt, "Sensory ERP components", in *The Oxford Handbook of Event-Related Potential Components*, S. J. Luck and E. S. Kappenman, Eds. Oxford: Oxford University Press, 2012, pp. 89–114.

[24] K. Bötzel and O. J. Grusser, "Electric brain potentials evoked by pictures of faces and non-faces - a search for face-specific EEG potentials", *Experimental Brain Research*, vol. 77, no. 2, pp. 349–360, Sept. 1989.

[25] D. A. Jeffreys, "A face-responsive potential recorded from the human scalp", *Experimental Brain Research*, vol. 78, no. 1, 193–202, Nov. 1989.

[26] B. Rossion and C. Jacques, "The N170: Understanding the time course of face perception in the human brain", in *The Oxford Handbook of Event-Related Potential Components*, S. J. Luck and E. S. Kappenman, Eds. Oxford: Oxford University Press, 2012, pp. 115–142.

[27] M. Eimer, "The face-sensitive N170 component of the event related brain potential", In *The Oxford Handbook of Face Perception*, A. J. Calder, G. Rhodes, M. Johnson, J. Haxby, Eds. Oxford: Oxford University Press, 2011 pp. 329–344.

[28] S. Sutton, M. Braren , J. Zubin and E. John, "Evoked potential correlates of stimulus uncertainty", *Science*, vol. 150, no. 3700, pp. 1187–1188, Nov. 1965.

[29] J. Polich, "Updating P300: an integrative theory of P3a and P3b", *Clinical Neurophysiology*, vol. 118, no. 10, pp. 2128–2148, Oct. 2007.

[30] S. H. Patel and P. N. Azzam, "Characterization of N200 and P300: Selected studies of the event-related potential", *Int. J. Medical Sci.*, vol. 2, no. 4, pp. 147–154, Oct. 2005.

[31] E. S. Kappenman and S. J. Luck, *The Oxford Handbook of Event-Related Potential Components*, Oxford: Oxford University Press, 2011.

[32] J. R. Wolpaw, N. Birbaumer, D. J. McFarland, G. Pfurtscheller and T. M. Vaughan, "Brain-computer interfaces for communication and control", *Clinical Neurophysiology*, vol. 113, no. 6, pp. 767-791, Jun. 2002.

[33] E. Donchin, K. M. Spencer and R. Wijesinghe, "The mental prosthesis: assessing the speed of a P300-Based brain-computer interface", *IEEE Trans. on Rehabilitation Eng.*, vol. 8, no. 2, pp. 174–179, Jun. 2000.

[34] D. S. Klobassa, T. M Vaughan, P. Brunner, N. E. Schwartz, J. R. Wolpaw, C. Neuper and E. W. Sellers, "Toward a high-throughput auditory P300-based brain-computer interface", *Clinical Neurophysiology*, vol. 120, no. 7, pp. 1252–1261, Jul. 2009.

[35] A-M. Brouwer, J. B. F Van ERP, "A tactile P300 brain-computer interface", *Frontiers in Neuroscience*, vol. 4, no. 19, May 2010. doi: 10.3389/fnins.2010.00019

[36] E. W. Sellers, D. J. Krusienski, D. J. McFarland, T. Vaughan and J. R. Wolpaw, "A P300 event-related potential braincomputer interface (BCI): The effects of matrix size and inter stimulus interval on performance", *Biological Psychology*, vol. 73, no. 3, pp. 242–252, Oct. 2006.

[37] D. J. Mcfarland, W. A. Samacki, G. Townsend, T. Vaughan and J. R. Wolpaw, "The P300-based brain-computer interface (BCI): Effects of stimulus rate", *Clinical Neurophysiology*, vol. 122, no. 4, pp. 731-737, Apr. 2011.

[38] K. Takano, T. Komatsu, N. Hata, Y. Nakajima and K. Kansaku, "Visual stimuli for the P300 brain-computer interface: A comparison of white/gray and green/blue

flicker matrices", *Clinical Neurophysiology*, vol. 120, no. 8, pp. 1562-1566, Aug. 2009.

[39] Y. Zhang, Q. Zhao, J. Jin, X. Wang and A. Cichoki, "A novel BCI based on ERP components sensitive to configural processing of human faces", *J. Neural Eng.*, vol. 9, no. 2, Mar. 2012. doi:10.1088/1741-2560/9/2/026018

[40] B. Blankertz, S. Lemm, M. Treder, S. Haufe and K-R. Müller, "Single-trial analysis and classification of ERP components - a tutorial", *NeuroImage*, vol. 56, no. 2, pp. 814–825, Jun. 2010.

[41] E. S. Kappenman and S. J. Luck, "The effects of electrode impedance on data quality and statistical significance in ERP recordings", *Psychophysiology*, vol. 47, no. 5, pp. 888-904, Sept. 2010.

[42] R. J. Croft and R. J. Barry, "Removal of ocular artifact from the EEG: a review", *Clinical Neurophysiology*, vol. 30, no. 1, pp. 5–19, Feb. 2000.

[43] C. D. Woody, "Characterization of an adaptive filter for the analysis of variable latency neuroelectric signals", *Medical and Biological Eng.*, vol. 5, no. 6, pp. 539–553, Nov. 1967.

[44] C. E. Davila and M. S. Mobin, "Weighted averaging of evoked potentials", *IEEE Trans. on Biomedical Eng.*, vol. 39, no. 4, pp. 338–345, Apr. 1992.

[45] J. C. Woestenburg, M. N. Verbaten, H. H. Van Hees and J. L. Slangen, "Single trial ERP estimation in the frequency domain using orthogonal polynomial trend analysis (OPTA): Estimation of individual habituation", *Biological Psychology*, vol. 17, pp. 173–191, May 1983.

[46] J. L. Kenemans, M. N. Verbaten, J.-W. Roelofs and J. L. Slangen, "Initial- and change-orienting reactions: An analysis based on visual single-trial event-related potentials", *Biological Psychology*, vol. 28, pp. 199-226, Jun. 1989.

[47] R. Q. Quiroga and H. Garcia, "Single-trial event-related potentials with wavelet denoising", *Clinical Neurophysiology*, vol. 114, no. 2, pp. 376-390, Feb. 2003.

[48] Z. Wang, A. Maier, D. A. Leopold, N. K. Logothetis and H. Liang, "Single-trial evoked potential estimation using wavelets", *Computers in Biology and Medicine*, vol. 37, no. 4, pp. 463-473, Apr. 2007.

[49] W. Truccolo, K. H. Knuth, A. Shah, S. L. Bressler, C. E. Schroeder and M. Z. Ding, "Estimation of single-trial multicomponent ERPs: Differentially variable component analysis (dVCA)", *Biological Cybern.*, vol. 89, no. 6, pp. 426–438, Dec. 2003.

[50] L. Xu, P. Stoica, J. Li, S. L. Bressler, X. Shao and M. Ding, "ASEO: A method for the simultaneous estimation of single-trial event-related potentials and ongoing brain activities", *IEEE Trans. Biomed. Eng.*, vol. 56, no. 1, pp. 111–121, Jan. 2009.

[51] P. A. Karjalainen, J. P. Kaipio, A. S. Koistinen and M. Vauhkonen, "Subspace regularization method for the single-trial estimation of evoked potentials", *IEEE Trans. Biomed. Eng.*, vol. 46, no. 7, pp. 849–860, Jul. 1999.

[52] C. Sielużycki, R. König, A. Matysiak, R. Kuś, D. Ircha and P. J. Durka, "Single-Trial evoked brain responses modeled by multivariate matching pursuit", *IEEE Trans. Biomed. Eng.*, vol. 56, no. 1, pp. 74–82, Jan. 2009.

[53] L. C. Parra, C. D. Spence, A. D. Gerson and P. Sajda, "Recipes for the linear analysis of EEG", *NeuroImage*, vol. 28, no. 2, pp. 326-341, Aug. 2005.

[54] C. J. James and C. W. Hesse, "Independent component analysis for biomedical signals", *Physiological Measurement*, vol. 26, no. 1, pp. R15-R39, Feb. 2005.

[55] R. Vigário and E. Oja, "BSS and ICA in neuroinformatics: From current practices to open challenges", *IEEE Reviews in Biomedical Eng.*, vol. 1 pp. 50-61, Dec. 2008.

[56] D. J. Mcfarland, L. M. McCane, S. V. David and J. R. Wolpaw, "Spatial filter selection for EEG-based communication", *Electroencephalography and Clinical Neurophysiology*, vol. 103, no. 3, pp. 386–394, Mar. 1997.

[57] R. Li, J. C. Principe, M. Bradley and V. Ferrari, "A spatiotemporal filtering methodology for single-trial ERP component estimation", *IEEE Trans. Biomed. Eng.*, vol. 56, no. 1, pp. 83-92, Jan. 2009.

[58] L. Spyrou, S. Sanei and C. C. Took, "Estimation and location tracking of the P300 subcomponents from single-trial EEG", in *IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP'07)*, Honolulu, Apr. 2007, pp. 1149–1152

[59] A. Hyvaärinen and E. Oja, "Independent component analysis: algorithms and applications", *Neural Networks*, vol. 13, pp. 411–430, Jun. 2000.

[60] S. J. Choi, "Blind source separation and independent component analysis: a review", *Neural Inform. Process. - Lett. and Reviews*, vol. 6, pp. 1–57, Jan. 2005.

[61] M. Ullsperger and S Debener, *Simultaneous EEG and fMRI: Recording, Analysis, and Application*. New York : Oxford University Press, 2010.

[62] F. Cong, I. Kalyakin, T. Huttunen-Scott and T. Ristaniemi, "Single-trial based independent component analysis on mismatch negativity in children", *Int. J. Neural Syst.*, vol. 20, no. 4, pp. 279–292, Aug. 2010.

[63] T. P. Jung, S. Makeig, M. Westerfield, J. Townsend, E. Courchesne and T. J. Sejnowski, "Analysis and visualization of single-trial event-related potentials", *Human Brain Mapping* vol. 14, no. 3, pp. 166-185, Nov. 2001.

[64] S. Makeig, T. P. Jung, A. J. Bell, D. Ghahremani and T. J. Sejnowski, "Blind separation of auditory event-related brain responses into independent component", *Proc. Nat. Academy of Sci.*, vol. 94, no. 20, pp. 10979-10984, Sept. 1997.

[65] S. Makeig, M. Westerfield, T. P. Jung, J. Covington, J. Townsend, T. J. Sejnowski and E. Courchesne, "Functionally Independent Components of the Late Positive Event-Related Potential during Visual Spatial Attention", *J. Neuroscience*, vol. 19, no. 7, pp. 2665-2680, Apr. 1999.

[66] A. Delorme, T. Sejnowski and S. Makeig, "Enhanced detection of artifacts in EEG data using higher-order statistics and independent component analysis", *Neuroimage*, vol. 34, no. 4, pp. 1443-1449, Feb. 2007.

[67] J. F. Cardoso, "High-Order Contrasts for Independent Component Analysis", *Neural Computation*, vol. 11, no. 1, pp. 157–192, Jan. 1999.

[68] A. J. Bell and T. J. Sejnowski, "An information-maximisation approach to blind separation and blind deconvolution", *Neural Computation*, vol. 7, no. 6, pp. 1129–1159, Nov. 1995.

[69] F. Cong, I. Kalyakin and T. Ristaniemi, "Can back-projection fully resolve polarity indeterminacy of independent component analysis in study of event-related potential?", *Biomedical Signal Process. and Control*, vol. 6, no. 4, pp. 422–426, Oct. 2011.

[70] A. Cichocki and S. Amari, "Blind decorrelation and SOS for Robust Blind Identification", in *Adaptive blind signal and image Processing*, West Sussex, England: John Wiley & Sons, 2002, ch. 4, pp. 129–176.

[71] S. V. Vaseghi, "Eigenvector analysis, principal component analysis and independent component analysis", in *Multimedia Signal Process.: Theory and Applications in Speech, Music and Communications*. West Sussex, England: John Wiley & Sons, 2007, pp. 381–414.

[72] R. M. Chapman and J. W. McCrary, "EP component identification and measurement by principal component analysis", *Brain and Cognition*, vol. 27, no. 3, pp. 288–310, Apr. 1995.

[73] S. Makeig and J. Onton, "ERP features and EEG dynamics: an ICA perspective", in *The Oxford Handbook of Event-Related Potential Components*, S. J. Luck and E. S. Kappenman, Eds. Oxford: Oxford University Press, 2012, pp. 51–86.

[74] T. P. Jung, S. Makeig, C. Humphries, T.-W. Lee, M. J. Mckeown, V. Iragui and T. J. Sejnowski, "Removing electroencephalographic artifacts by blind source separation", *Phychophysiology*, vol. 37, no. 2, pp. 163–178, Mar. 2000.

[75] G. Pires, U. Nunes, and M. Castelo-Branco, Statistical spatial filtering for a P300-based BCI: Test in able-bodied, and patients with cerebral palsy and amyotrophic lateral sclerosis, *J. Neuroscience Methods*, vol. 195, no. 2, pp. 270-281, Feb. 2011.

[76] B. Blankertz, R. Tomioka, S. Lemm, M. Kawanabe and K-R. Müller, "Optimizing spatial filters for robust EEG single-trial analysis - Revealing tricks of the trade", *IEEE Signal Processing Mag.*, pp. 41–56, Jan. 2008

[77] B. D. V. Veen and K. M. Buckley, “Beamforming: a versatile approach to spatial filtering”, *IEEE ASSP Mag.*, pp. 4–24. Apr. 1988.

[78] Z. J. Koles, M. S. Lazar and S. Z. Zhou, ”Spatial patterns underlying population differences in the background EEG”, *Brain Topography*, vol. 2, no. 4, pp. 275–284, Jun. 1990.

[79] Z. J. Koles, J. C. Lind, P. Flor-Henry, ”Spatial patterns in the background EEG underlying mental disease in man”, *Electroencephalography and Clinical Neurophysiology*, vol. 91, no. 2, pp. 319-328, Jun. 1994.

[80] H. Ramoser, J. Müller-Gerking, G. Pfurtscheller, “Optimal spatial filtering of single trial EEG during imagined hand movement”, *IEEE Trans. Rehabil. Eng.*, vol. 8, no. 4, pp. 441–446, dec. 2000.

[81] W. Samek, C. Vidaurre, K. R. Muüller and M. Kawanabe, “Stationary common spatial patterns for brain-computer interfacing”, *J. Neural Eng.*, vol. 9, no. 2, 2012. doi: 10.1088/1741-2560/9/2/026013

[82] G. Pires, U. Nunes, and M. Castelo-Branco, “P300 spatial filtering and coherence-based channel selection”, in *4th Int. IEEE EMBS Conf. on Neural Eng. (NER’09)*, Antalya, April 2009.

[83] K. Yu, K. Shen, S. Shao, W. C. Ng, X. Li, and K. Kwok, “Single-trial event-related potential based rapid image triage system”, *J. Systemics, Cybern. and Informatics*, vol. 9, no. 3, pp. 8-14, May 2011.

[84] B. Rivet, A. Souloumiac, V. Attina and G. Gibert, ”xDAWN algorithm to enhance evoked potentials: application to braincomputer interface”, *IEEE Trans. Biomed. Eng.*, vol. 56, no. 8, pp. 2035-2043, 2009.

[85] R. O. Duda, P. Hart and D. Stork, *Pattern Classification*, 2nd ed. New York: John Wiley & Sons, 2001.

[86] L. Parra, C. Alvino, A. Tang, B. Pearlmutter and N. Yeung, A. Osman and P. Sajda, “Linear Spatial Integration for Single-Trial Detection in Encephalography”, *NeuroImage*, vol. 17, pp. 223-230, 2002.

[87] A. D. Gerson, L. C. Parra and P. Sajda, "Cortical origins of response time variability during rapid discrimination of visual objects", *NeuroImage*, vol. 28, no. 2, pp. 342–353, Sept. 2005.

[88] A. Luo and P. Sajda, "Using single-trial EEG to estimate the timing of target onset during rapid serial visual presentation", in *28th Int. Conf of the IEEE Eng. in Medicine and Biology Soc. (EMBC'06)*, New York, Sept. 2006, pp. 79–82.

[89] S. Scholler, S. Bosse, M. S. Treder, B. Blankertz, G. Curio, K. R. Müller and T. Wiegand *et al.*, "Toward a Direct Measure of Video Quality Perception Using EEG", *IEEE Trans. Image Process.*, vol. 21, no. 5, pp. 2619-2629, May 2012.

[90] A. K. Porbadnigk, J. N. Antons, B. Blankertz, M. Treder, R. Schleicher, S. Möller and G. Curio *et al.*, "Using ERPs for Assessing the (Sub)Conscious Perception of Noise", In *32nd Int. Conf of the IEEE Eng. in Medicine and Biology Soc. (EMBC'10)*, Buenos Aires, Aug., 2010.

[91] A. K. Porbadnigk, M. S. Treder, B. Blankertz, J. N. Antons, R. Schleicher, S. Möller, G. Curio and K. R. Müller, "Single-trial analysis of the neural correlates of speech quality perception", *J. Neural Eng.*, vol. 10, pp. 056003, Jul. 2013.

[92] U. Hoffmann, J. M. Vesin and T. Ebrahimi, "Spatial filters for the classification of event-related potentials", in *European Symposium on Artificial Neural Network (ESANN'06)*, Bruges, Apr. 2006, pp. 26–28.

[93] S. Theodoridis and K. Koutroumbas, *Pattern Recognition*, 4th ed., Burlington: Academic Press, 2008.

[94] C. M. Bishop, *Pattern Recognition and Machine Learning*, Cambridge: Springer, 2006.

[95] F. Lotte, M. Congedo, A. L*é*cuyer, F. Lamarche and B. Arnaldi, "A review of classification algorithms for EEG-based brain-computer interfaces", *J. Neural Eng.*, vol. 4, no. 2, pp. R1-R13, Jan. 2007.

[96] M. Akcakaya *et al.*, "Noninvasive brain computer interfaces for augmentative and alternative communication", *IEEE Reviews in Biomed. Eng.*, vol. 7, pp. 1937-3333, Dec. 2013.

[97] D. J. Krusienski, E. W. Sellers, D. J. McFarland, T. M. Vaughan and J. R. Wolpaw, "Toward enhanced P300 speller performance", *J. Neuroscience Methods*, vol. 167, no. 1, pp. 15-21, Jan. 2009.

[98] D. J. Krusienski, E. W. Sellers, F. Cabestaing, S. Bayoudh, D. J. McFarland, T. M. Vaughan and J. R. Wolpaw, "A comparison of classification techniques for the P300 Speller", *J. Neural Eng.*, vol. 3, no. 4, pp. 299–305, Oct. 2006.

[99] H. Kim, B. L. Drake and H. Park, "Multiclass classifiers based on dimension reduction with generalized LDA", *Pattern Recognition*, vol. 40, no. 11, pp. 2939-2945, Nov. 2007.

[100] R. P. N. Rao and R. Scherer, "Statistical pattern recognition and Machine learning in brain-computer interfaces", in *Statistical Signal Processing for Neuroscience and Neurotechnology*, G.O. Karim, Ed. Oxford: Academic Press, 2010, pp. 335–367.

[101] U. Hoffmann, J. M. Vesin, T. Ebrahimi and K. Diserens, "An efficient P300-based braincomputer interface for disabled subjects", *J. of Neuroscience Methods*, vol. 167, no. 1, pp. 115125, 2008.

[102] D. Lehmann and W. Shandies, "Spatial analysis of evoked potentials in man - A review", *Prog. Neurobiol.* , vol. 23, no. 3, pp. 227-250, 1984.

[103] M. M. Murray, D. Brunet and C. M. Michel, "Topographic ERP Analyses: A Step-by-Step Tutorial Review", *Brain Topography*, vol. 20, no. 4, pp. 249-264, Mar. 2008.

[104] D. Brunet, M. M. Murray and C. M. Michel, "Spatiotemporal Analysis of Multichannel EEG: CARTOOL", *Computational Intell. and Neuroscience*, vol. 2011, Nov. 2010. doi:10.1155/2011/813870

[105] R. D. Pascual-Marqui, C. M Michel and D. Lehmann, "Segmentation of Brain Electrical Activity into Microstates: Model Estimation and Validation", *IEEE Trans. Biomed. Eng.*, vol. 42, no. 7, pp. 658-665, Jul. 1995.

[106] O. Marques, *Practical Image and Video Process. Using Matlab*. New Jersey: John Wiley & Sons, 2011.

[107] S. P. Lloyd, "Least squares quantization in PCM", *IEEE Trans. Inform. Theory*, vol. 28, no. 2, pp. 129-137, Mar. 1982.

[108] D. Kuefner, A. D. Heering, C. Jacques, E. Palmero-Soler and B. Rossion, "Early visually evoked electrophysiological responses over the human brain (P1, N170) show stable patterns of face-sensitivity from 4 years to adulthood", *Frontiers in Human Neuroscience*, vol. 3, no. 67, pp. 1–22, Jan. 2010.

[109] G. Dawson, S. J. Webb and J. McPartland, "Understanding the nature of face Process. impairment in autism: insights from behavioral and electrophysiological studies", *Developmental Neuropsychology*, vol. 27, no. 3, pp. 403–424, 2005.

[110] L. Boutsen, G. W. Humphreys, P. Praamstra and T. Warbrick, "Comparing neural correlates of configural Process. in faces and objects: an ERP study of the thatcher illusion", *NeuroImage*, vol. 32, no. 1, pp. 352–367, Aug. 2006.

[111] N170 EEG Dataset Website [Online]. Available: http://www.fil.ion.ucl.ac.uk/spm/data/mmfaces/.

[112] R. N. Henson, Y. Goshen-Gottstein, T. Ganel, L. J. Otten, A. Quayle and M. D. Rugg, "Electrophysiological and Haemodynamic Correlates of Face Perception, Recognition and Priming", *Cerebral Cortex*, vol. 13, no. 7, pp. 793–805, Jul. 2003.

[113] N. Xu, X. Gao, B. Hong, X. Miao, S. Gao and F. Yang, "BCI competition 2003-data set IIb: enhancing P300 wave detection using ICA-based subspace projections for BCI applications", *IEEE Trans. Biomed. Eng.*, vol. 51, no. 6, pp. 1067–1072, Jun. 2004.

[114] H. Serby, E. Yom-Tov and G. F. Inbar, "An improved P300-based brain-computer interface", *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 13, no. 1, pp. 89–98, Mar. 2005.

[115] J. Onton, M. Westerfield, J. Townsend and S. Makeig, "Imaging human EEG dynamics using independent component analysis", *Neuroscience and Biobehavioral Reviews*, vol. 30, no. 6, pp. 808–822, 2006.

[116] D. Iyer and G. Zouridakis, "Single-trial evoked potential estimation: comparison between independent component analysis and wavelet denoising", *Clinical Neurophysiology*, vol. 118, no. 3, pp. 495–504, Mar. 2007.

[117] C. J. James and O. J. Gibson, "Temporally constrained ICA: An application to artifact rejection in electromagnetic brain signal analysis", *IEEE Trans. Biomed. Eng.*, vol. 50, no. 9, pp. 1108–1115, Sept. 2003.

[118] Z. L. Zhang, "Morphologically constrained ICA for extracting weak temporally correlated", *Neurocomputing*, vol. 71, pp. 1669–1679, Mar. 2008.

[119] Q. H. Lin, Y. R. Zheng, F. L. Yin, H. Liang and V. D. Calhoun, "A fast algorithm for one-unit ICA-R", *Inform. Sci.*, vol. 177, no. 5, pp. 1265-1275, Sep. 2006.

[120] A. Hyvärinen, "One-unit contrast functions for independent component analysis: a statistical analysis", In *Proc. 1997 IEEE Workshop on Neural Networks for Signal Process.*, Amelia Island, Sept. 1997, pp. 388–397.

[121] O. I. Khan, S.-H. Kim, T. Rasheed, A. Khan, and T.-S. Kim, "Robust extraction of P300 using constrained ICA for BCI applications", *Med. Biology Eng. Comput.*, vol. 50, pp. 231-241, Jan. 2012.

[122] W. Lu and J.C. Rajapakse, "ICA with reference", in *Proc. 3rd Int. Conf. Independent Component Analysis and Blind Signal Separation (ICA'01)*, San Diego, pp. 120-125.

[123] V. Zarzoso and A. Hyvärinen, "Iterative algorithms", in *Handbook of Blind Source Separation*, P. Comon and C. Jutten Eds. Oxford: Academic Press, 2010, ch. 6, sec. 6.9.2, pp. 207.

[124] A. Cichocki and S. Amari, "Sequential Blind Signal Extraction", in *Adaptive blind signal and image Processing*. West Sussex, England: John Wiley & Sons, 2002, ch. 5, sec. 5.2.5, pp. 191–193.

[125] W. Lu and J. C. Rajapakse, "ICA with Reference", in *Neurocomputing*, vol. 69, pp. 2244–2257, Feb. 2006.

[126] A. Delorme and S. Makeig, "EEGLAB: an open source toolbox for analysis of single-trial EEG dynamics including independent component analysis", *J. Neuroscience Methods*, vol. 134, no. 1, 9–21, Mar. 2004.

[127] B. Blankertz *et al.*, "The BCI competition 2003: Progress and perspectives in detection and discrimination of EEG single trials", *IEEE. Trans. Biomed. Eng.*, vol. 51, no. 6, pp. 1044-1051, Jun. 2004.

[128] K. Li, R. Sankar, Y. Arbel and E. Donchin, "Single trial independent component analysis for P300 BCI system", in *31st Int. Conf of the IEEE Eng. in Medicine and Biology Soc. (EMBC'09)*, Minneapolis, pp. 4031-4034, Sept. 2009.

[129] Y. Wang and T. P. Jung, "Improving brain-computer interfaces using independent component analysis", in *Towards Practical Brain-Computer Interfaces*, B. Z. Allison *et al*, Ed. Heidelberg: Springer-Verlag Berlin, 2012, pp. 67–83.

[130] S. G. Wang and C. J. James, "Enhancing evoked responses for BCI through advanced ICA techniques" *Proc. IET 3rd Int. Conf. on Advances in Medical, Signal and Inform. Process. (MEDSIP'06)*, Stevenage, pp. 1–4, Jul. 2006.

[131] S. Halder, E. M. Hammer, S. C. Kleih, M. Bogdan, W. Rosenstiel, N. Birbaumer and A. Kübler, "Prediction of accuracy and visual P300 brain-computer interface aptitude", *PLoS ONE*, vol 8, Feb. 2013. doi: 10.1371/J..pone.0053513.

[132] J. N. Mak, Y. Arbel, J. W. Minett, L. M. McCane, B. Yuksel, D. Ryan, D. Thompson, L. Bianchi and D. Erdogmus, "Optimizing the P300-based brain-computer interface: current status, limitation and future directions", *J. Neural Eng.*, vol. 8, no. 2, 2011. doi: 10.1088/1741-2560/8/2/025003.

[133] S. Ting, T. Tan, G. West, A. Squelch and J. Foster, "Quantitative assessment of 2D versus 3D visualization modalities", in *Visual Communications and Image Process. (VCIP'11)*, Tainan, pp. 6-9, Nov. 2011.

[134] T. Tan, "Brain responses to 2d versus 3d images using EEG" [Online]. Available: http://researchdata.ands.org.au/brain-responses-to-2d-versus-3d-images-using-eeg.

[135] S. Rousseau, C. Jutten and M. Congedo, "Time window selection for improving error-related potential detection", in *Proc. 4th Int. Joint Conf. on Computational Intell. (IJCCI'12)*, Barcelone, Oct. 2012, pp. 715–720.

[136] H. Cecotti, R. Phlypo, B. Rivet, M. Congedo, E. Maby and J. Mattout, "Impact of the time segment analysis for P300 detection with spatial filtering", in *3rd Int. Symp. on Applied Sci. in Biomed. and Comm. Tech. (ISABEL'10)*, Rome, Nov. 2010.

[137] P. S. Hamilton and W. J. Tompkins, "Quantitative investigation of QRS detection rules using the MIT/BIH Arrhythmia Database", *IEEE Trans. Biomed. Eng.*, vol. BME-33, pp. 1157–1165, Dec. 1986.

[138] J. Pan and W. J. Tompkins, "A real-time QRS detection algorithm", *IEEE Trans. on Biomed. Eng.*, vol. BME-32, pp. 230–236, Mar. 1985.

[139] A. K. Jain, R. P. W. Duin and J. Mao, "Statistical Pattern Recognition: a review", *IEEE Trans. Pattern Analysis and Mach. Intell.*, vol. 22, no. 1, pp. 4–37, Jan. 2000.

[140] A. Rakotomamonjy and V. Guigue, "BCI Competition III: dataset II ensemble of SVMs for BCI P300 speller", *IEEE Trans. Biomed. Eng.*, vol. 55, no. 3, pp. 1147–1154, Mar. 2008.

[141] T. Fawcett, "An Introduction to ROC analysis", *Pattern Recognition Lett.*, vol. 27, no. 8, pp. 861-874, Jun. 2006.

[142] A. P. Bradley, "The use of the area under the ROC curve in the evaluation of Machine learning algorithms", *Pattern Recognition*, vol. 30, no. 30, pp. 1145–1159, Jul. 1997.

[143] K. E. Hild II, M. Kurimo and V. D. Calhoun, "The sixth annual MLSP competition, 2010", in *IEEE Int. Workshop on Mach. Learning for Signal Process. (MLSP'10)*, Kittila, Aug. 2010, pp. 107-111.

[144] M. Kaper, P. Meinicke, U. Grossekathoefer, T. Lingner and H. Ritter, "BCI Competition 2003Data Set IIb: Support vector Machines for the P300 speller paradigm" *IEEE Trans. Biomed. Eng.* vol. 51, no. 6, pp. 1073–1076, 2004.

[145] H. Mirghasemi, R. Fazel-Rezai and M. B. Shamsollahi, "Analysis of P300 Classifiers in Brain Computer Interface Speller", *28th Int. Conf. of the IEEE Eng. in Medicine and Biology Soc. (EMBC'06)*, New York, Aug. 2006, pp. 6205–6208.

[146] N. Chumerin, N. V. Manyakov, A. Combaz, J. A. K. Suykens and M. M. Van Hulle, "An application of feature selection to on-line P300 detection in brain-computer interface", in *IEEE Int. Workshop on Mach. Learning for Signal Process. (MLSP'09)*, Grenoble, Sept. 2009, pp. 1–6.

[147] J. Jin, B. Z. Allison, E. W. Sellers, C. Brunner, P. Horki, X. Wang and C. Neuper, "An adaptive P300-based control system", *J. Neural Eng.*, vol. 8, no. 3, Jun. 2011. doi: doi:10.1088/1741-2560/8/3/036006

[148] E. Yin, Z. Zhou, J. Jiang, F. Chen, Y. Liu and D. Hu, "A novel hybrid BCI speller based on the incorporation of SSVEP into the P300 paradigm", *J. Neural Eng.*, vol. 10, no. 2, Apr. 2013. doi:10.1088/1741-2560/10/2/026012

[149] B. Blankertz *et al.*, "The BCI Competition III: Validating Alternative Approaches to Actual BCI Problems", *IEEE. Trans. Neural Syst. Rehabil. Eng.*, vol. 14, no. 2, pp. 153–159, Jun. 2006.

[150] N. V. Manyakov, N. Chumerin, A. Combaz and M. M. Van Hulle, "On the selection of time interval and frequency range of EEG signal preProcess. for P300 brain-computer interfacing", *Proc. XII Mediterranean Conf. on Medical and Biology Eng. and Comp. (MEDICON'10)*, Chalkidiki, May 2010, pp. 57–60.

[151] W-H. Steeb, "Optimization", in *The Nonlinear Workbook*, 5th ed. Singapore: World Scientific Publishing, 2011, pp. 297–322.

[152] C. J. C. Burges, "A Tutorial on Support Vector Machines for Pattern Recognition", *Data Mining and Knowledge Discovery*, vol. 2, no. 2, pp. 121-167, Jun. 1998.

Every reasonable effort has been made to acknowledge the owners of copyright material. I would be pleased to hear from any copyright owner who has been omitted or incorrectly acknowledged.