Poznan University of Technology
Faculty of Computing and Information Science
Institute of Computing Science

Master's thesis

CLASSIFICATION OF MOTOR IMAGERY FOR BRAIN-COMPUTER INTERFACES

Piotr Szachewicz

Supervisor
Wojciech Jaśkowski, PhD

Poznań, 2013

Tutaj przychodzi karta pracy dyplomowej;
oryginał wstawiamy do wersji dla archiwum PP, w pozostałych kopiach wstawiamy ksero.

Abstract

Brain-computer interfaces (BCI) are systems which enable a user to control a device using only his or her neural activity [GAP10]. An important part of a brain-computer interface is an algorithm for classifying different commands that the user may want to execute.

There are several neurological phenomena that can be used in a BCI. One of them is *event-related desynchronization* (ERD), which is a temporary decrease in power of the *mu* and *beta brain waves*. This phenomenon can be registered using *electroencephalography* (EEG) and occurs when a subject performs or imagines a limb movement [CP99].

The goal of this thesis is to implement an algorithm that would be able to classify four different motor imagery tasks, in which a user imagines a movement of one of the following parts of the body: left hand, right hand, both feet, or tongue. The data and the classification task formulation are taken from the fourth edition of the BCI Competition, which is an open competition concerning BCI-related data classification [TMA+12]. Although the competition is closed and the results were already published, we try to simulate conditions similar to the ones that the competition participants had had.

Using *electroencephalogram* (EEG) data from the BCI Competition we test two feature extraction techniques: *band power features* and *time domain parameters*, and two classifiers: *linear discriminant analysis* (LDA) and *support vector machines* (SVM). We test all algorithms with different parameters on the training set using cross-validation and then we verify the best four of them on the final test set.

All four algorithms obtain results which would have enabled them to win the third place in the BCI Competition IV. This result shows that it is possible to use a fairly simple algorithm for EEG data classification and obtain good results if the parameter space for the algorithms is searched through metodically.

Contents

Chapter 1

Introduction

1.1 Scope

This thesis focuses on *brain-computer interfaces* (BCIs), *signal processing* and *machine learning*. We present an algorithm that uses machine learning techniques to classify segments of *electroencephalogram* (EEG) recordings in order to determine which *motor imagery* task the subject was executing at a particular moment of the recording — whether he or she was imagining a left hand, right hand, both feet or tongue movement. EEG signal is used as input data for the classification algorithm and signal processing methods are used to reduce the algorithm's processing time, improve signal-to-noise ratio and extract features. In order to discriminate between the four motor imagery tasks a *classifier* is used.

1.2 Motivation

Brain-computer interfaces (BCIs) are systems which enable a user to control a device using only his or her brain neural activity [GAP10]. BCIs are proposed as a communication tool for the paralyzed [WBH+00], but also have a wide range of other applications including neural prosthetics [JMM+06, MPP08], wheelchairs [RBG+07], video games and virtual reality [LLR+08], creative expression [MB05], access to the internet [MRH+10] etc. [Moo03, AGG07]. Although some of the BCI applications can also be useful for healthy people [AGG07], the main focus in BCI research is put on providing a means of communication and control to the disabled users, who otherwise would have limited — or no — means of communication with the outer world [WB06].

The first attempts in brain-computer interfaces date back to 1964 [GAP10], but the main onset of research in BCI started in the 1990s and the number of publications per year in this research area constantly grows [Ham10]. Even though brain-computer interfaces were considered to be science-fiction fifty years ago [GAP10], nowadays scientists are actually able to construct BCI systems.

Although much effort is put in BCI-related research and scientists obtain better and better results, constructing an effective brain-computer interface is still a challenge. The problems that have to be faced include choosing adequate features and classifiers, as well as using appropriate signal processing techniques. In order to improve the effectiveness of the BCI systems, scientists develop more and more complex systems (e.g. compare the work by Wolpaw from 1991 [WMNF91] to the 2012 research by Kus et al. [KVZ+12]).

One problem with publications concerning BCI systems is that very often the performance of the classifier is measured using different measures or different EEG data sets. This makes the obtained results incomparable to each other.

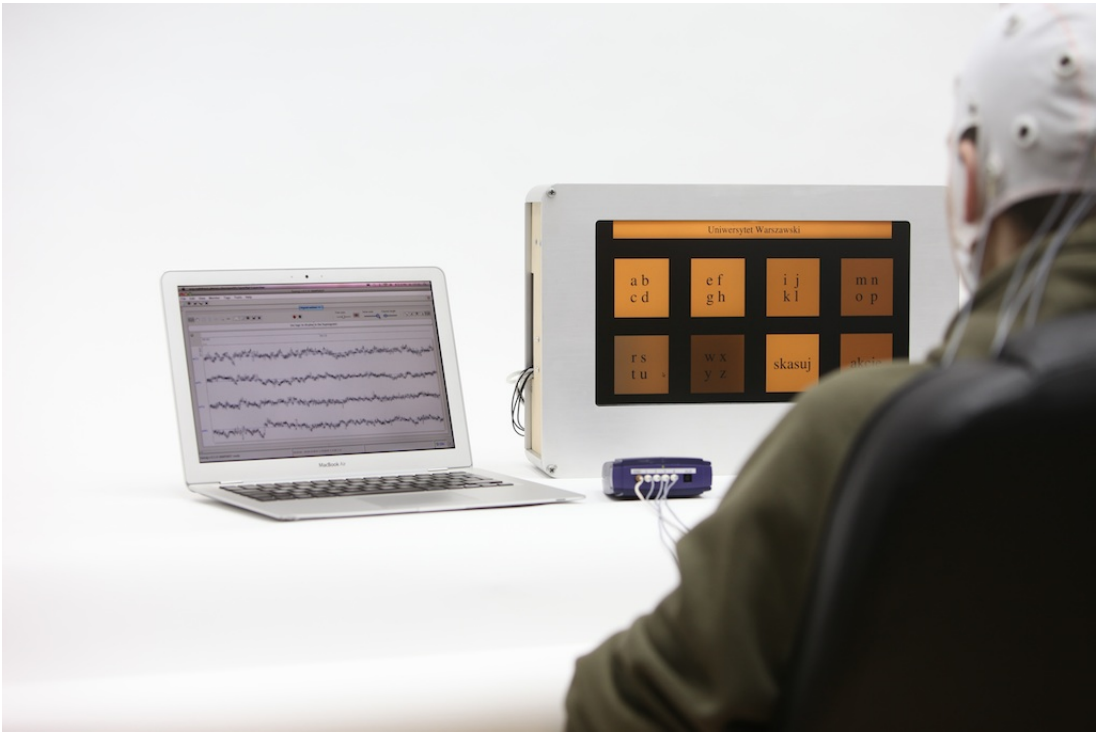

FIGURE 1.1: An example of a brain-computer interface (BCI) developed at the Faculty of Physics, University of Warsaw. This BCI uses steady-state visually evoked potentials phenomenon (cf. 2.2.6). Photography reproduced with Piotr Durka permission [DKŻ+12].

Another problem is that when BCI systems are described in papers, the authors do not always justify why did they choose the particular set of components for the system and what influence each of the components has on the performance of the presented BCI.

1.3 Objective

The goal of this thesis is to implement an algorithm that would be able to classify four different motor imagery tasks: left hand, right hand, both feet, and tongue movement. Moreover, we try to address the two drawbacks described in the previous section.

First of all we use the EEG data set and the task formulation from the BCI Competition IV (2008) [TMA+12], a competition in which researchers propose algorithms for a posed BCI-related classification tasks. The forth edition of the competition is already closed and all the data (both training and testing data sets) and the final results are now publicly available. In this thesis, we try to simulate taking part in the competition by hiding the testing set until the final evaluation and following the rules that other competitors were required to follow. Using the data from the BCI Competition IV allows us to compare the results obtained by our algorithms to the results obtained by other competitors.

We start off with a simple classification algorithm that is later improved by adding additional processing modules and choosing parameters that maximize the classifier's performance. This way we try to address the second drawback described in the previous section, as each step is throughoutly described and the classifier's performance is compared when different parameter values and signal processing techniques are used.

1.4 Thesis organization

This thesis is structured as follows. Chapter 2 describes the basic information concerning *electroencephalography* (EEG), *brain-computer interfaces* (BCI) and *event-related desynchronization* (ERD) — a neurological phenomenon that is used to classify different motor imagery classes. Chapter 3 describes the BCI Competition and one of the tasks that was posed at its fourth edition which we undertake in this thesis. Chapter 4 describes some basic methods that are used in the classification algorithms. Initial data visualizations are presented in Chapter 5 in order to show the existence of the ERD phenomenon in the given data set. In Chapter 6 the results obtained by the implemented classifiers are shown. Different feature extraction techniques, signal processing algorithms and classifiers are tested with different parameters using cross-validation. At the end of Chapter 6 the best classifiers are tested on the final test data set. The thesis ends with Chapter 7 in which we sum up the obtained results and describe the work that can be done in the future in order to further improve the accuracy of the algorithms.

Chapter 2

Background

2.1 Electroencephalography

Electroencephalography (EEG) is a technique for recording the electrical activity of the brain using electrodes placed on the scalp [Bro00]. The history of EEG dates back to 1875 when the first EEG recording from an animal was made by Richard Caton [WB06]. The first recording from human was conducted by Hans Berger in 1924 [Tep02, Bro00].

EEG signals are commonly recorded using Ag-AgCl electrodes and the signal values that can be obtained are in the range of 0.5–100 µV [Tep02]. Apart from EEG, other techniques of recording the electrical signals of the brain are also available, the main difference between them is the placement of the electrodes [WB06]:

- *electrocorticography* (ECoG) — in which the electrical signals are recorded using electrodes placed on the cortical surface (inside of the skull);
- *local field potentials* (LFP) — in which the electrodes are put inside of the brain.

The electrocorticography and the local field potential techniques provide better spatial resolution and frequency bandwidth, but on the other hand, these methods are invasive and more difficult to use than EEG [WB06]. In this thesis, we concentrate only on the EEG signals. The signal recorded using electroencephalography is called the *electroencephalogram* (EEG). An example of an electroencephalogram is shown in Figure 2.1.

2.1.1 10–20 system

In order to allow different EEG recordings to be compared, the International Federation of Societes for Electroencephalography and Clinical Neurophysiology proposed a standardized system for electrode placement called EEG 10–20 [NLdS05].

The EEG 10–20 system consists of 19 electrodes whose positions are determined in relation to specific anatomical landmarks on the head and are separated by 10% or 20% distances (Figure 2.2). The anatomical landmarks used in EEG 10–20 are the *nasion* — the indentation located just above the nose, the *inion* — the protruding bone at the back of the head, and two *preauricular points*, which are points just in front of the *pinna* [MP95, Spr10]. For example, if the distance between the left and the right preauricular point is 100%, the distances between the T3, T4 electrodes and left and right preauricular points respectively are equal to 10% of the total distance, and the distances between electrodes T3 and C3, C3 and Cz, Cz and C4, and C4 and T4 are all equal to 20% of the total distance.

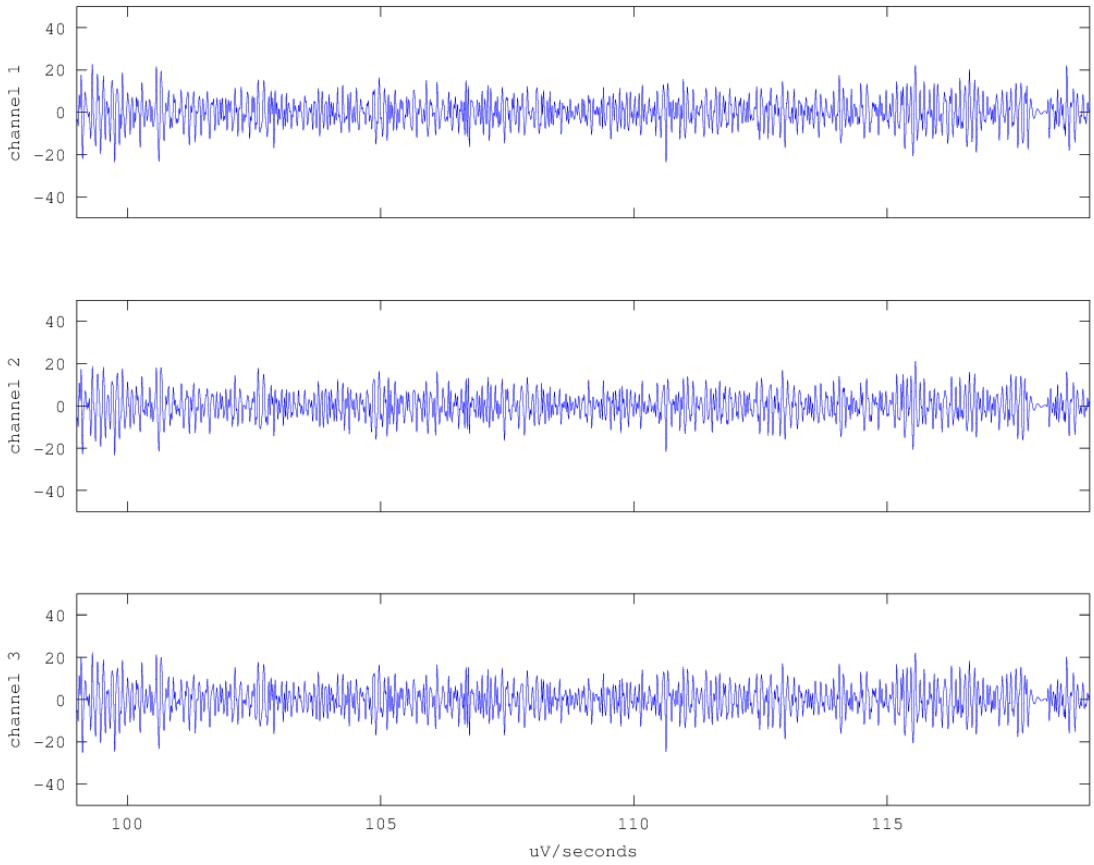

FIGURE 2.1: A segment of an electroencephalogram (EEG) that is used in this thesis.

Each electrode has a name consisting of a letter and a number. The first part of the name come from the part of the brain over which the electrode is positioned. F, T, P and O designate the *frontal*, *temporal*, *parietal* and *occipital* lobes respectively, whereas Fp and C are the *frontal pole* and the *central sulcus*. The electrodes with odd numbers are positioned over the left hemisphere, and with the even numbers — over the right hemisphere. The electrodes numbered with 0 (very often the letter "z" is used instead) are positioned on the central axis of the head [MP95].

2.1.2 Neural oscillations types

The neurons in the brain propagate information by sending short electrical pulses, called *spikes*. When spikes from a group of neurons are superimposed, the obtained signal is of an oscillatory nature. The resulting oscillations are called *neural oscillations* and are divided into several frequency ranges. This division has an important role in electroencephalography research, as each neural oscillation is connected to some type of cognitive state [BD06]:

- **delta** (0.5–4 Hz) — the slowest brain rhythm, during sleep this rhythm is the strongest one;
- **theta** (4–8 Hz) — does not occur frequently in adult humans;
- **alpha** (8–13 Hz) — predominant wave during wakefulness, present over the area of visual cortex, strongest when the eyes are closed or when a person is in state of relaxation;

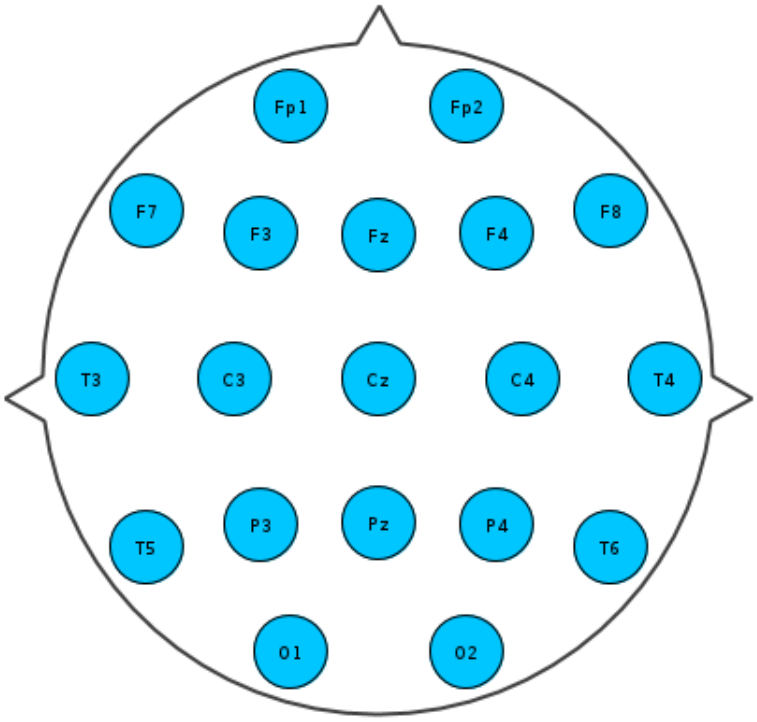

FIGURE 2.2: Electrode placement in the EEG 10-20 system (screen shot from Svarog, Signal Viewer, Recorder and Annotator on GPL, `http://www.svarog.pl`).

- **mu** (8–13 Hz) — have the same frequency range as alpha, but are present over the motor cortex;
- **beta** (13–30 Hz) — are connected to states of alertness and attention;
- **gamma** (30 > Hz) — are related to information processing.

2.1.3 EEG applications

Electroencephalography has been used for a wide range of applications, examples of which include:

- **scientific research** — EEG has been used in various scientific investigations in diverse research areas such as linguistics [SC11], psychology [TJK06, GSCS75], creativity [Sri07], research concerning meditation [CP06], neuromarketing [VCTB13], sex differences [DSPB76], sleep and sleep disorders [WKH74], etc.;
- **brain death diagnosis** — EEG is used as a tool to confirm brain death [CCC+08];
- **epilepsy** — the electroencephalogram of epileptic patients contains distinctive discharges of spikes and waves. Thus EEG is a valuable tool for diagnosing, classifying and monitoring epilepsy [Pan05, Smi05];
- **neurofeedback** — a type of biofeedback in which EEG signal is visualized and showed to a patient (for example in a form of a game) which helps him learn how to self-regulate his brain activity [HGS07]. Neurofeedback has been used to help patients suffering from attention deficit syndromes, alcoholism, strokes, chronic fatigue syndrome, or asthmatic conditions [EA99];

- **brain-computer interfaces (BCI)** — are systems which enable people to control computers using only EEG signals from the brain.

The following part of this thesis is focused on the BCI application of EEG.

2.2 Brain-computer interfaces (BCI)

2.2.1 Definition

When conducting research in the area of brain-computer interfaces a good definition of a brain-computer interface (BCI) is useful. Here we quote such a definition from the first international BCI conference:

> A brain–computer interface is a communication system that does not depend on the brain's normal output pathways of peripheral nerves and muscles. [WBH+00]

This definition limits the range of control signals for the BCI only to brain signals, excluding any signals obtained from muscles or peripheral nerves. This is important, because EEG contains also components related to muscle movement called *artifacts*, which could be mistakenly used to control a BCI.

For example — eye movement or blink artifacts are very strong in EEG. If these signals were used, on purpose, by mistake, or by oversight, to control the BCI, the system is not a brain-computer interface in sense of this definition, because it uses information from muscles.

2.2.2 Functional model

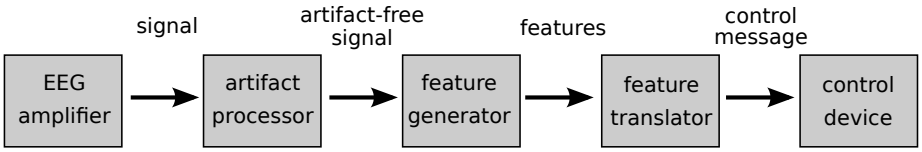

Figure 2.3: A typical model of a BCI system [BFWB07].

A typical functional model of a brain-computer interface (shown in Figure 2.3) consists of the following elements [MB03, BFWB07]:

- **EEG electrodes and amplifier** — the source of the signal that is later analyzed.
- **Artifact processor** — as mentioned in Section 2.2.1, EEG signal contains artifacts which should be removed before further processing. Examples of artifacts include eye or muscle movement, heart beat, or utility frequency (50 or 60 Hz) artifacts. Artifact processor removes or reduces the artifacts that exist in the signal.
- **Feature generator** — responsible for transforming the raw signal into a set of features that can be later used by a classifier. This module may contain several sub-modules for performing the following tasks:
  - **Signal enhancement (preprocessing)** — in this step signal processing algorithms are used in order to enhance the signal-to-noise ratio of the signal.

  - **Feature extraction** — the main element of a feature generator, contains algorithms that are used to calculate features from the given signal.
  - **Feature selection/dimensionality reduction** — may be used to select for further processing only those features that contain the most useful information which enables to discern the classification classes. Dimensionality reduction can be used to reduce the number of features in order to manage the curse of dimensionality and prevent overfitting [CP97].

- **Feature translator** — is used to translate the features obtained in the previous step to signals sent to the control device and contains the following two steps:
  - **Feature classification** — classifies — using machine learning techniques — the features into one of the control states available for a given BCI (e.g. move wheelchair to the left/right).
  - **Post-processing** — this additional step can be used after the classification is done, e.g. in order to reject the control requests having low probability in the current situation.
- **Control device** — the device that is controlled by the BCI, e.g. a wheelchair, or a computer.

2.2.3 Synchronous vs asynchronous BCI

There are two main modes of operation for a brain-computer interface: *synchronous* and *asynchronous* mode [PN01].

In a synchronous BCI a prompt cue is shown to the user periodically and during the presence of the cue on the screen the subject can generate some mental states in order to control the device. If a BCI is asynchronous, no cues are shown and the user can generate mental states aimed to control the device whenever he or she wants.

The classification algorithm implemented for this thesis works in the synchronous mode.

2.2.4 Applications

Brain-computer interfaces may have various uses. The use of BCIs have been proposed for patients suffering from the *locked-in syndrome* (LIS) [WBH+00]. People with locked-in syndrome are almost completely paralyzed (except for the eyes) as a result of a stroke, brain injury, or brain degenerative diseases. Through the use of brain-computer interfaces some of these people are able to regain means of communication with the outer world. Unfortunately, research shows that BCIs are not applicable for all patients — patients with complete locked-in syndrome, who are not even able to move their eyes, are unable to use BCI systems [KB08].

Apart from that, BCI systems were also proposed for communication (spellers) [Bir06], neural prosthetics [JMM+06, MPP08], wheelchairs [RBG+07], video games and virtual reality [LLR+08], creative expression [MB05], access to the internet [MRH+10] etc. [Moo03, AGG07].

2.2.5 Devices

Although most of the brain-computer interfaces are developed for laboratory research purposes only, there are also several BCI systems available on the market. Examples of these are Neurosky Mindwave and Emotiv EPOC headset [NAGG12]. EPOC is a 14-electrodes wireless device accompanied by software which is able to detect user's facial expressions or emotional states, and an application for manipulating a 3D object. As already mentioned in Section 2.2.1, the facial

expressions detection functionality does not account for a real BCI system, because the muscle movement is involved. The Emotiv EPOC headset is cheap and costs about 299$ [emo].

The Neurosky Mindwave device is equipped with one EEG electrode and costs 79.99$. There are ten applications and games included for monitoring and training attention and concentration levels, blink detections and meditation. More applications are available to buy in the online Neurosky store [neu].

2.2.6 Neurological phenomena

Brain-computer interfaces are categorized depending on the type of neurological phenomena they use. Currently there are four main types of neurological phenomena used by BCIs [SA13]:

- **P300** — the P300 event-related potential is a temporal growth in amplitude of the EEG signal that occurs 300 ms after a stimulus that have been awaited by a subject is presented. P300 potential occurs also after presenting a salient stimulus in a series of non-salient stimuli. An example of a BCI based on the P300 paradigm is the P300 speller. A P300 speller application shows a matrix of letters to the user and randomly highlights one row or column in the matrix. The goal of the algorithm is to determine for which column and for which row the P300 potential occurs, thus determining at which letter the user is currently looking [BGH+99].

- ***Steady-state visually evoked potentials* (SSVEP)** — if a person looks at something that flickers with some specific frequency, this particular frequency (and its upper harmonics) can be detected in his or her EEG signal. A BCI that uses this paradigm works by showing several elements flickering with different frequencies and detects at which object the user currently looks by analyzing the frequency peaks in his or her EEG [LZWG07].

- ***Slow cortical potentials* (SCP)** — humans can learn to control these potentials and produce positive and negative shifts. A negative/positive shift occurs when the EEG signal values are below/above some *baseline* (usual for the subject) level for 300 ms to several seconds. In an SCP-based BCI a user is presented with some feedback information that shows the current average value of the EEG signal. The user tries to invoke a negative or a positive potential shift while concentrating on this feedback information [HSN+04].

- ***Event-related desynchronization* (ERD)** — changes in power in the mu (8–13 Hz) and beta (13–30 Hz) bands while executing or imagining the movement. Since the EEG classifier implemented in this thesis is based on this phenomenon, ERD is described more thoroughly in Section 2.3.

New BCI paradigms are still being discovered. For example, researchers Lopez et al. and Kim et al. [LPP+09, WJH+11] have tried to use *auditory steady-state responses* (ASSR) for purpose of brain-computer interfaces. A BCI built using ASSR can work in the following way. The subject receives the following auditory stimuli: 1000 Hz sine wave (amplitude modulated with 38 Hz sine wave) to the left ear and 2500 Hz sine wave (amplitude modulated with 42 Hz sine wave) to the right ear. If the subject focuses his attention on one of the signals, its modulation frequency has a significantly larger power in his or her EEG spectrum.

2.3 Event-related desynchronization (ERD)

In 1930, in his early EEG experiments, Berger discovered an attenuation in alpha waves activity when a subject opens his or her eyes [PdS99, LMC09]. This classical phenomenon is commonly used during EEG experiments to determine the quality of the EEG signal [HCP02].

Event-related desynchronization (ERD) is a phenomenon similar to the one discovered by Berger. The term was introduced in 1970s by Gert Pfurtscheller to denote blocking of mu band waves in reaction to diverse types of stimuli, for example [LMC09]: voluntary movement of a limb [PKC99], motor imagery (imagining a movement of a limb without actually moving it) [CP99], or exercising memory tasks [Kli99].

In case of motor imagery, the event-related desynchronization starts when the subject begins to imagine a movement with a limb and manifests itself as a power decrease in mu and beta bands. After that a different phenomenon occurs, *event-related synchronization* (ERS), which is an increase in power (above the previous baseline level) in the mu and beta bands that occurs when the subject stops imagining a movement [CP99].

The phenomenon of ERD/ERS related to motor imagery actions has different spatial characteristics and strength for different limbs — it is stronger for the *contralateral* hemisphere, and weaker for the *ipsilateral* hemisphere. Contralateral hemisphere is the hemisphere on the side of the body opposite to the limb for which the motor imagery task is executed, whereas ipsilateral hemisphere is on the same side of the body. For example: if the subject imagines a movement with his or her right hand, event-related desynchronization and synchronization occurs mostly in the left hemisphere. Similarly, if the subject imagines a movement with his or her left hand, the ERD/ERS occurs mostly in the right hemisphere. Feet and arms movements are more difficult to separate, because in that case larger areas of motor cortex are affected [PKC99].

What is also important, the ERD/ERS needs some time to recover, thus approximately 10 second interval between trials is recommended [PLdS99]. Obviously, this fact has an influence on ERD/ERS brain-computer interface potential speed.

2.3.1 Detecting ERD using the band power method

In order to detect and quantify event-related desynchronization and synchronization, various methods can be utilized. Here, the most widely used technique, the band power method, is presented [PLdS99].

The first step consists of filtering the signal in two bands (e.g. 8–12 and 18–30 Hz). As a result two signals are obtained, $s_\mu(t)$ — containing the 8–12 Hz frequency components of the signal (in which the ERD effect should be the strongest), and $s_\beta(t)$ — containing 18–30 Hz components (in which the ERS effect should be visible). It should be noted however, that the particular frequency bands in which ERD/ERS occurs depend strongly on the person, thus better results can be obtained if the frequency bands are chosen for each person individually [PLdS99].

Next, signal values in these two bands are squared in order to obtain power values using the following equation:

$$p_b(t) = s_b^2(t), \tag{2.1}$$

where $b \in \{\mu, \beta\}$.

The resulting signal is averaged over trials. A trial is a time slice during the experiment when the subject performs a particular action (e.g. imagines a movement of his or her left hand). If there

are N trials related to a particular movement and each trial starts at time t_i and lasts for T seconds, then the result of averaging can be calculated as follows:

$$\overline{p_b}(\tau) = \frac{1}{N} \sum_i p_b(t_i + \tau), \tag{2.2}$$

where $i \in \{1, 2, \ldots, N\}$ and the resulting $\overline{p_b}(\tau)$ is defined for $\tau \in [0, T]$, τ denoting the time within the averaged trial.

The last step may include averaging the power over time in order to smooth the obtained signal.

2.3.2 Examples of ERD-based BCI systems

There is a great number of studies concerning ERD-based brain-computer interfaces. Here only few of them are presented, as an overview.

In 1994 Wolpaw and McFarland [WM94] developed a BCI system using the mu waves amplitude of the subject to control the position of a cursor in a two-dimensional space (previously a similar paper was published about 1-D cursor control BCI [WMNF91]). The BCI used two *bipolar channels*, FC3/CP3 and FC4/CP4. A bipolar channel represents the difference in voltage between the given two electrodes, e.g., the value of channel FC3/CP3 is equal to the difference in voltage between electrodes FC3 and CP3. Square root of power in the 7.5–12.5 Hz frequency band was used as a feature in the algorithm (the power was calculated for 200 ms segments using the Fast Fourier Transform). A very simple algorithm was used for classification — two linear equations transformed the mu power into cursor movement (up, down, left, or right). Five subjects were trained to use the BCI over 6–8 weeks. The classification accuracies obtained at the end of the experiment were in the range of 41–70%.

In 1996 Gert Pfurtscheller et al. constructed the first [WZL+04] BCI which used ERD features [PKN+96]. This system also used only two electrodes, C3 and C4, and power in frequency bands 8–10, 10–12, 16–20, 20–24 Hz was used as features. *Distinction sensitive learning vector quantization algorithm* (DSLVQA) was used for selection of the optimal frequency bands. The final classification was performed using the *learning vector quantization* (LVQ) classifier. The system was built to classify real — not imagined — left and right hand movements. The classification accuracies were reported to be 96–100% for some subjects.

The scientists continued research and developed systems which were able to classify motor imagery acts. The systems also reached almost 100% classification accuracy for classifying between left and right hand motor imagery, however, few days of training were required. Moreover, during the time of the training the classifier was adjusted to a specific subject using the rapid prototyping procedure [PN01, PNG+00, NMPSP06].

Unfortunately, there exist subjects for whom classification rates of some types of brain-computer interfaces are very low. This phenomenon is called *BCI illiteracy* and results in the fact that the subject is unable to effectively use a specific type of a BCI system [PAB+10, Asa13]. As a solution for this problem, hybrid BCI systems were proposed, which use more than one type of neurological phenomenon to operate. An example of a hybrid BCI system may be a system combining ERD with SSVEP. Using this kind of BCI, users who are unable to use the ERD-based part of the BCI can switch to the SSVEP task to increase accuracy. For subjects proficient with both types of BCIs, hybrid BCI can increase accuracy and transfer rate, which is the number of information (or commands) that can be transferred using the BCI in a given amount of time [ABK+10].

Chapter 3

Task formulation

In this chapter we present the problem that is investigated in this thesis and the data set used.

3.1 BCI Competition

BCI Competition is an open competition which aims at evaluating various approaches used in brain-computer interfaces and comparing them on the same data set in order to obtain a reliable measurement of performance for each algorithm. It is an attempt to solve the problem of comparing BCI-related signal processing methods that are published, but their accuracy was verified on different data sets or they use different performance measures, which makes the relative comparison between any two of the selected methods impossible [SGM+03].

Four editions of the BCI Competition were organized (the last one in 2008) and each edition consisted of 3–5 data sets for which a different classification task was provided. Each data set is freely available on the competition web page[1] and is divided into a training set and a test set. The goal of the competitors is to provide programs that would be able to correctly classify the data from the test set.

3.2 Data set description

In this thesis we use the 2a data set from the BCI Competition IV (2008).

This data set was described by Brunner et al. [BLMP+] and is available for download from the BCI Competition web page[2]. It contains four-class motor-imagery data which is described in the next section.

3.2.1 Experimental paradigm

The data set consists of EEG data recorded from 9 subjects. For a given subject two sessions were recorded on two different days. Each session consisted of 6 runs and each run consisted of 48 trials (12 trials for each motor imagery class). In each trial a cue was shown on the screen instructing the subject to perform one of the following motor imagery tasks: left hand, right hand, both feet, or tongue motor imagery.

Thus a single session during the experiment consisted of 288 trials, 72 trials for each motor imaginary task. Each trial started with a short sound (warning tone). At the same time a fixation cross was shown on the computer screen. After 2 seconds, in the place of the fixation cross,

[1] http://bbci.de/competition
[2] http://bbci.de/competition/iv/#datasets

a cue was shown (in a form of a small arrow) telling the subject to start the corresponding motor imagery task. After another 1.25 seconds the arrow changed back to the fixation cross. The motor imagery task should have been continued until the sixth second of the trial when the fixation cross disappeared. After that there was a short break (1.5 seconds).

The structure of a single trial is shown in Figure 3.1.

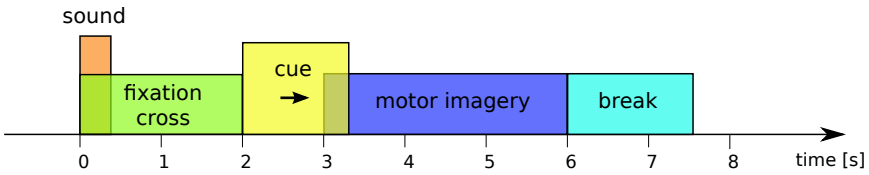

FIGURE 3.1: The time-course of a single trial in the data set 2a from the BCI Competition IV.

The data was recorded using 22 EEG and 3 *electrooculogram* (EOG, a technique for measuring the eye movements) channels. The signal was recorded at 250 Hz sampling frequency and was 0.1–100 Hz band-pass filtered and 50 Hz notch filtered. The EEG electrode placement during this experiment was a bit different than the 10-20 system shown in Figure 2.2. It consisted of the Fz, C3, Cz, C4 and Pz electrodes and additional 17 electrodes positioned at 10% distances away from the mentioned five electrodes [BLMP$^+$].

3.2.2 Task description

The data set consists of 18 files, each containing a recording of one experiment session for one subject. One of the sessions for each subject contains the class labels for each trial (i.e. we know which motor imagery task the subject was performing in each trial) and is intended to be used as a training set. The other session does not contain the class labels and the task is to infer the correct labels using the classifier trained on the training set data (first session). Correct labels for the test set are provided for one subject, as an example.

The important challenge posed in this task is that the training and the testing signal come from different recording sessions conducted on different days. The rationale behind this is that in order to use a BCI efficiently, long training is required [PN01]. Thus, it would be advantageous for a classifier to use the data collected in the previous recording sessions.

Unfortunately, data recorded on different sessions can show considerable differences. Each session requires the subject to be connected to the EEG equipment anew, which may result in different impedances of the electrodes. What is more, in each session the psycho-physical state and the motivation of the user may be different. All this may result in the fact that the signals obtained from each recording session can have different characteristics. An interesting review of intersession, intertrial and interuser variability is available in Linsey Roijendijk's master's thesis [Roi09].

The output of the classifier should provide class label predictions for each sample in the trial. The aim of this requirement is to imitate the operation conditions of online, asynchronous BCI systems, which at each point of time classify the EEG signal stream in order to detect current intentions of the user [TMA$^+$12].

The last requirement is that the classification algorithm needs to have the property of causality, which means that for a given sample of EEG signal for which the classification is being made, the algorithm is allowed to use only the current and the previous samples.

3.2.3 Evaluation method

The submissions for this task are evaluated and compared using the *kappa coefficient*, which takes value of 0 for a random classifier and 1 for a perfect classifier that always classifies correctly. The value of the kappa coefficient can be calculated using the following equation:

$$\kappa = \frac{p_0 - p_e}{1 - p_e} \tag{3.1}$$

where p_0 is the classification accuracy and p_e is the hypothetical accuracy of a random classifier on the same data. In our case, $p_e = 0.25$, thus $\kappa = \frac{p_0 - 0.25}{0.75}$.

Having the classifier results for each sample in each trial, confusion matrices are constructed for each sample. From these confusion matrices the kappa value for each sample is calculated, thus a time-course of kappa values within a trial is obtained. The final measure of performance of a given algorithm is the maximum value of the kappa value from the calculated time-course [BLMP$^+$]. The implementation of the algorithm used to evaluate submissions is available in the BioSig library [SKHM07].

3.3 BCI Competition results

Five teams submitted their classification algorithms for the 2a data set. The results of the competition for this particular data set are shown in Table 3.1.

TABLE 3.1: Kappa values obtained by the competitors in the BCI Competition IV, data set 2*a* [BCI13]. Numbers 1–9 designate the 9 subjects who took part in the experiment.

no	contributor	mean kappa	subjects								
			1	2	3	4	5	6	7	8	9
1	Kai Keng Ang	**0.57**	0.68	**0.42**	**0.75**	**0.48**	**0.40**	**0.27**	**0.77**	**0.75**	0.61
2	Liu Guangquan	**0.52**	**0.69**	0.34	0.71	0.44	0.16	0.21	0.66	0.73	**0.69**
3	Wei Song	**0.31**	0.38	0.18	0.48	0.33	0.07	0.14	0.29	0.49	0.44
4	Damien Coyle	**0.30**	0.46	0.25	0.65	0.31	0.12	0.07	0.00	0.46	0.42
5	Jin Wu	**0.29**	0.41	0.17	0.39	0.25	0.06	0.16	0.34	0.45	0.37

The winning algorithm [ACW$^+$12] was provided by Kai Keng Ang with contributors: Zheng Yang Chin, Chuanchu Wang, Cuntai Guan, Haihong Zhang, Kok Soon Phua, Bahama Hamadicharef, Keng Peng Tee from Institute for Infocomm Research, Agency for Science, Technology and Research Singapore.

Chapter 4

Methods

In this chapter we describe the EEG signal processing methods used in the classifier implemented for this thesis.

4.1 Preprocessing

4.1.1 Artifacts reduction

Artifacts are unwanted signals present in the EEG. They have various origins, which include the utility frequency (50 Hz in Europe or 60 Hz in the United States), body and eye movements, or blinks [FBW+07].

The utility frequency artifacts were already removed from the data using the notch filter mentioned in Section 3.2.1. The step of eye artifacts removal is left to the competitor in order to leave the possibility of testing different artifact removal algorithms [BLMP+].

There are three main approaches for handling eye artifacts: *avoidance*, *rejection* and *removal* [FBW+07]. In artifact avoidance we simply ask the user not to execute the movements which result in EEG artifacts. This can reduce the number of artifacts, but obviously eye movements and blinks are inevitable, so sometimes these artifacts do occur. What is more, constant restraining from blinking causes fatigue to the user.

Another approach is to reject all trials which are contaminated by artifacts. This can be done either manually or automatically. In a manual artifact rejection procedure, an expert determines which trials are contaminated through visual examination. In automatic artifact rejection an algorithm is implemented which is able to determine if a trial is contaminated by artifacts or not. A simple version of such an algorithm may operate by checking if the amplitude of the *electrooculogram* (EOG, a technique for recording eye movements) channel is above the specified threshold — if that is the case we consider the trial to be contaminated. The main problem with artifact rejection is that it reduces the size of the training set, which can have consequences on the classification accuracy.

In the last approach, artifact removal, some algorithms are used in order to remove the artifact-related components from the EEG signal, but ideally leave the desired brain-originated signal intact. The simplest method for artifact removal is high-pass filtering of the signal. The eye artifacts occur in the frequency range of 0–4 Hz, so by filtering these components out, we may reduce the EOG artifacts.

Other methods for artifact removal include *linear regression*, *blind source separation*, *independent component analysis* (ICA), *principal component analysis* (PCA), *wavelet transform*, *nonlinear adaptive filtering*, or *source dipole analysis* [FBW+07].

An example of an EEG signal after eye artifact reduction is shown in Figure 4.1. The figure shows seconds 1570–1580 from the first recording session for subject one. The last three channels are EOG channels (eye movement recordings). In the EOG channels, at second 1576, we can see an eye artifact. The eye movement artifact is also visible in the EEG channel (first plot) — its amplitude is higher than the amplitude of the normal EEG signal. The second plot shows the same signal (first channel) after the artifact was removed using a 7–30 Hz band-pass filter. We use a band-pass filter instead of a high-pass filter, because we we do not need the frequencies above 30 Hz, as they do not contain any valuable ERD/ERS information.

In the further parts of this thesis all processing is done only on the signals with the eye artifacts already removed (unless noted otherwise). This allows to reduce processing time, as the algorithms do not have to remove the artifacts each time they are run.

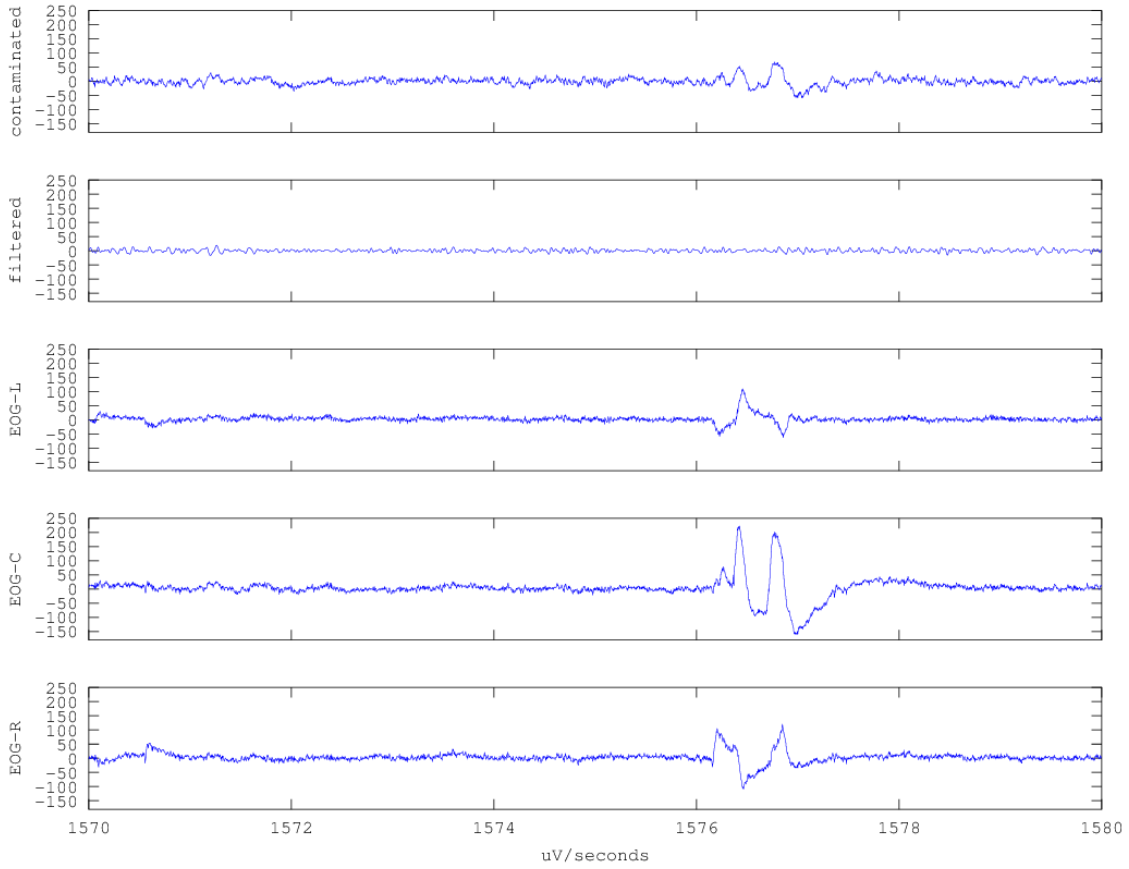

FIGURE 4.1: Eye artifacts removed using a 7–30 Hz band-pass filter (fifth order Butterworth filter). The image shows the time span between seconds 1570–1580 of the first recording session for subject one. The first plot is channel Fz without any artifact reduction. The second plot shows the same channel, but after band-pass filtering. The last three channels are electrooculogram channels (EOG).

4.1.2 Spatial filtering

Common spatial pattern (CSP) is a signal preprocessing method that can be used to enhance the discriminativity of classes [MGPF99, RMGP00]. The main concept of the algorithm can be described as follows. Let V^i be a $K \times N$ matrix in which the K-channel EEG signal data from

the i-th trial is stored (each trial is N samples long) and $c \in \{\phi, \psi\}$ — a set of two classification conditions (e.g. motor imagery of the left and right hand).

The algorithm finds a $K \times K$ transformation matrix P, which allows to transform the EEG signal from the original into a new channel space Z^i:

$$Z^i = P^T \cdot V^i. \tag{4.1}$$

Each row in P^T defines a new channel in Z^i as a linear combination of the original channels.

The covariance matrices R_c^i of the transformed signals Z^i under the condition c can be calculated as follows:

$$R_c^i = \frac{Z_c^i (Z_c^i)^T}{trace(Z_c^i (Z_c^i)^T)}, \tag{4.2}$$

where Z_c^i is the i-th trial for condition c and the trace (which is the sum of the elements on the diagonal of a square matrix) in the denominator is used for normalization [MGPF99]. The covariance matrices are then averaged across trials to obtain R_c. Each element in the covariance matrix $r_{mn} \in R_c$ is the covariance between channels m and n in all trials under the condition c.

One requirement in the common spatial patterns method is that the channels of the transformed signal are independent and uncorrelated (each channel should correspond to an independent source of signal). This means that the covariance matrices of the transformed signal R_c should be diagonal [Kuś13]. If the R_c matrix is diagonal, each channel in signal Z is correlated only with itself and the correlations between any other two channels are equal to zero.

Another requirement for the transformed signal Z is that the variance of at least one of the channels for ϕ trials is maximized and the variance for the same channel for ψ trials is minimized. At the same time the algorithm maximizes the variance of some other channel for ψ trials and minimizes it for ϕ trials. This can be be obtained if the sum of the two covariance matrices is an identity matrix [Kuś13]:

$$R_\phi + R_\psi = I. \tag{4.3}$$

In that case, if the variance for one of the channels under condition ϕ is larger, the variance for the same channel under condition ψ is accordingly smaller, because the two variances need to sum up to 1.

The result of the CSP algorithm can be calculated by means of a simultaneous diagonalization of covariance matrices for conditions ϕ and ψ, e.g. using JADiag or FFDiag diagonalization algorithms [KVZ+12]. Although CSP can also be generalized to support a larger number of conditions [GWB08], in the name of simplicity only two classes were considered in the above explanation.

4.2 Feature extraction

The following two feature extraction techniques were tested in the classifier implemented for this thesis: *band power features* and *time domain parameters*.

4.2.1 Band power

There exist several methods for band power feature extraction from EEG signals [BLL11]. In this thesis we use a method implemented in the BioSig biomedical signal processing library that calculates the band power by band-pass filtering the signal.

This method works as follows. First, the signal is filtered using a band-pass filter designed for a given frequency band. In BioSig library a 4-th order Butterworth *infinite impulse response* (IIR) filter is used.

Next, each sample of the resulting signal $x[t]$[1], which contains only the required frequency components, is squared to obtain the time course of power:

$$p[t] = x^2[t]. \tag{4.4}$$

Given the smoothing window size w, the following smoothing operation is applied to the signal obtained from the previous step:

$$\overline{p}[n] = \frac{1}{w}\sum_{k=0}^{w} p[n-k]. \tag{4.5}$$

This means that the band power for sample n is equal to the average power of w previous samples.

The final feature values are equal to $ln(\overline{p}[n])$. The logarithm is used because it can improve the performance of linear classification [VKBS09].

4.2.2 Time domain parameters

Another feature extraction technique used in this thesis are the *time domain parameters* (TDP). The original definition of time domain parameters by Carmen Vidaurre [VKBS09] described TDP as variances of the first k derivatives of the signal.

In this thesis a variant of TDP implemented in the BioSig library is used, which calculates time-varying power of the first k derivatives of the signal using the following equation:

$$p_i(t) = \frac{d^i x(t)}{dt^i}, i = 0, 1, \ldots, k. \tag{4.6}$$

The obtained values are later smoothed using an exponential moving average window filter, which is implemented with the following infinite impulse response (IIR) filter:

$$y[n] = u \cdot p_i[n] - (1-u) \cdot y[n-1], \tag{4.7}$$

where p_i is the input signal (i-th order derivative) and y is the filtering result. The u value is used as a parameter to calculate the time domain parameters. The smaller the u value is, the bigger is the moving average window. The final feature values are equal to $ln(y[n])$.

4.3 Classification

Two classifiers are tested during the experiments conducted for this thesis: *linear discriminant analysis* (LDA) and *support vector machines* (SVM).

4.3.1 Linear discriminant analysis (LDA)

Let us consider a set of n examples $\vec{a_1}, \vec{a_2}, \ldots, \vec{a_n}$ defined in a two-dimensional feature space. An example $\vec{a}$ can be projected onto a direction defined by a unit vector $\hat{w}$ using the following equation (scalar projection of a vector $\vec{a}$ onto a unit vector w):

$$b = \vec{a} \cdot \hat{w}.^2 \tag{4.8}$$

The result of this equation is a scalar, thus the projection results in a dimensionality reduction from two dimensions to one dimension.

Figure 4.2 shows two projections of the given examples onto two different vectors $\hat{w}$. Consider two classes — ϕ and ψ (denoted in the picture as white and black circles). After the first projection,

[1] Square brackets are commonly used in digital signal processing to denote discrete signals.
[2] This equation is sometimes written as $b = w^T a$.

the data from different classes is separable by means of a simple thresholding of the scalar b. If the second projection is performed instead of the first the data becomes unseparable.

Linear discriminant analysis (LDA) is a classification algorithm whose purpose is to find such a direction $\hat{w}$ that projecting the data onto $\hat{w}$ maximizes the distance between the means and minimizes the variance of the two classes. More precisely, the LDA classifier minimizes the following equation:

$$\frac{(m_\phi - m_\psi)^2}{s_\phi^2 + s_\psi^2}, \tag{4.9}$$

where m_ϕ, m_ψ are the means and s_ϕ, s_ψ are the standard deviations of both classes after projecting the examples onto $\hat{w}$ [Alp04].

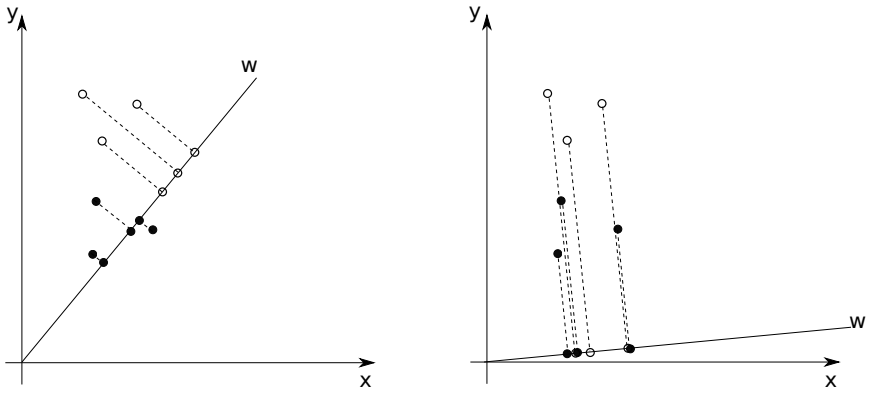

FIGURE 4.2: Two projections of the same examples onto different vectors $\hat{w}$ (as is done by the LDA classifier). The first projection makes the classification of data simple. The second projection makes the data inseparable.

4.3.2 Support vector machines (SVM)

Another classifier used in this thesis are *support vector machines* (SVM). The function used for classification in SVM has the form of a hyperplane defined by the equation:

$$y = w^T x + b. \tag{4.10}$$

The hyperplane (also called the *decision border*) divides the feature space into two parts and the classification result — i.e. whether an example is classified to be in one class or the other — depends on which side of the hyperplane the example is located.

The goal of the support vector machines is to find such a hyperplane which divides the two classes, and, moreover, for which the distances between the hyperplane and the closest examples are maximized. In SVM the distances between a hyperplane and the nearest examples are called *margins*. Figure 4.3 shows an example of two possible hyperplanes in a two-dimensional space[3]. We can see that for the first plot the margins are larger than for the second plot. In that case, the former hyperplane is preferred to the latter by the support vector machines algorithm.

The original formulation of SVM did not allow any examples to be misclassified (to be on the wrong side of the hyperplane). In this thesis we use a *soft-margin* formulation of SVM which allows some examples to be misclassified if that helps the SVM algorithm to increase the margins [BHW10]. The *soft-margin* formulation introduces a C parameter, which is used to control

[3] In a two-dimensional space a hyperplane is equivalent to a line.

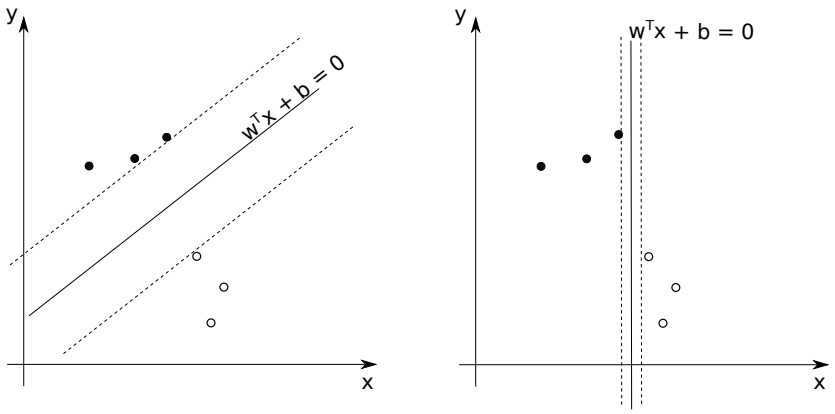

FIGURE 4.3: The figure shows two possible linear decision boundaries for a support vector machines (SVM). The decision boundary with a larger margin (plot on the left) is preferred by the SVM classifier.

how important is the proper classification to the classifier. If the value of the C parameter is large, the algorithm does not allow the examples to be misclassified. The smaller the value of the C parameter is, the less important it is to correctly classify the examples. In that case the algorithm may find a decision border for which the margins are larger, even though some examples are misclassified. Thus, the C parameter may be used as a tool to control overfitting, as the more examples are misclassified (for example because they are outliers), the less the the overfitting effect is prominent.

Support vector machines can be transformed into a non-linear method if a non-linear *kernel function* is used. A kernel function is a function that defines a new feature vector for a given example x by calculating the similarity between the example x and some other example y. There are several similarity functions, also called *kernel functions*, defined. In this thesis we use a Gaussian kernel which is defined as follows:

$$K(x, y) = e^{-\frac{\|x-y\|^2}{2 \cdot \sigma^2}} \tag{4.11}$$

and takes the values between 0 and 1 depending on the similarity of x and y (1 if the two examples are equal). If the features are calculated using the Gaussian kernel, non-linear class boundaries can be learned by the SVM classifier. The σ parameter (standard deviation) controls the width of the kernel function (the larger the σ value, the wider the function). For a wider Gaussian kernel functions examples far away from each other are considered more similar to each other than for a narrower Gaussian. Thus, for larger kernel widths the overfitting effect of the algorithm can be minimized.

The description presented here discussed only two-class versions of the classifiers, however both methods can be generalized to address multiclass problems.

4.4 Cross-validation

With the purpose of having similar chances to the BCI competitors, the testing set was hidden for the most part of the experiments and it was only used to ultimately check the final performance of the few chosen classifiers.

In order to compare different classifiers cross-validation on the training set was used instead. Cross-validation is a machine learning technique which can be used to estimate the optimal param-

eters for a classifier while preventing overfitting the classifier to the training data [Mit97]. In our experiments we used k-fold cross validation, which works by dividing the training set into k parts of equal size, then using $k-1$ parts as a training set and check the classification rate on the one remaining part (validation set). This operation has to be repeated k times (folds), so that each part of the training set is used for evaluating the quality of the classifier exactly once. The final result is obtained by calculating the average of k classification rates obtained for k validation sets.

In our case, the training data set for one subject consisted of 288 trials, 72 trials for each motor imagery task. In order to ensure that at each fold is of equal size, we used 8-fold cross-validation. What is more, the cross-validation algorithm was designed in such a way, that in each fold the validation set (and thus the training set) contained equal number of trials for each motor imagery task. This means that each validation set consisted of 36 trials: 9 trials for each motor imagery task (imagining left hand, right hand, feet or tongue movements).

4.5 Tools

We used GNU Octave [EBH97] programming language to implement the classification and evaluation algorithms used in this thesis. In addition, two libraries were used: BioSig library and Shogun machine learning toolbox.

BioSig[4] [SB08, VSS11] is an open-source library for processing of biomedical signals and implements many methods for EEG signal processing, e.g. removal of eye artifacts and utility frequency, histograms, common spatial patterns, several feature extraction techniques, methods for cutting signal into segments (so that each segments contains one trial) and several classification and visualization algorithms.

Shogun [SRH+10] is an open-source machine learning toolbox available on GNU GPL license. It contains multiple machine learning algorithms including several implementations of the SVM algorithm. For the classification algorithms, the BioSig LDA and Shogun SVM implementations were used.

[4]The following version was used: `https://github.com/piotr-szachewicz/biosig`.

Chapter 5

Data visualization

In this chapter we present some analysis and visualizations of BCI Competition IV 2a data.

5.1 Signal frequency analysis

First the frequency content of the signal was analyzed. Figure 5.1 shows the FFT plots drawn for each motor imagery class (left hand, right hand, feet and tongue). The FFT was calculated by cutting out trials related to a specific motor imagery class, calculating FFT for each trial separately and then averaging the results.

We can see that the energy in the lower frequency bands is higher than in the higher frequency bands. The peak at the 50 Hz is a residue of the utility frequency. This was partly removed by a 50 Hz notch filter (as described in Section 3.2), but some remainings of this frequency can still be visible in the spectrum (as well as the dampening effect of the filter on the adjacent frequencies).

Figure 5.2 shows the frequency content of the signal after applying a 7–30 Hz fifth-order Butterworth band-pass filter used to remove the eye artifacts. This filter removes from the signal also other unnecessary frequency components, as the ERD/ERS phenomenon occurs in the the mu and beta frequency bands (8–30 Hz) only.

Interestingly, at this stage it is already possible to see that differences exist between different motor imagery tasks for some frequency ranges around 8–12 and 20–23 Hz.

5.2 Event-related desynchronization quantization

In this section the ERD/ERS phenomenon will be showed by using the quantification method described in Section 2.3.1. Figure 5.3 illustrates the time courses of mu and beta bands from electrodes C3 and C4 while subject 7 was performing left hand motor imagery task. In each trial the cue was shown at second 2 and the start of the event-related desynchronization can be seen around second 2.5–3 in both frequency bands. All power values were calculated in relation to a baseline power extracted from range 1–2 seconds after the trial start, thus the percent value informs how much stronger or weaker the band power was comparing to the baseline value.

According to the literature, the strongest ERD/ERS effect should be visible on the contralateral hemisphere [CP99]. In case of left hand motor imagery the contralateral hemisphere is the right hemisphere, represented on the plot by the C4 electrode. Unexpectedly, in the presented plots, the ERD and ERS effects were stronger for the ipsilateral hemisphere.

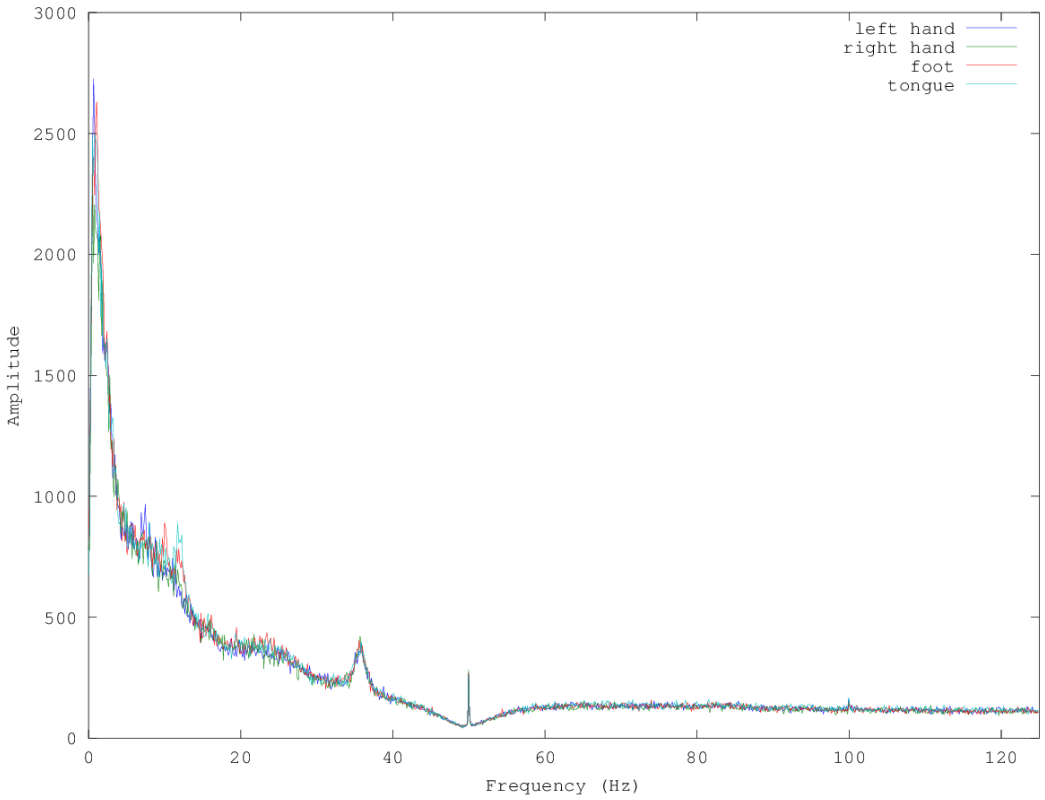

FIGURE 5.1: FFT plots calculated for each motor imagery class (subject 1, electrode $C4$). FFT was averaged over trials. No artifact removal was performed on the data.

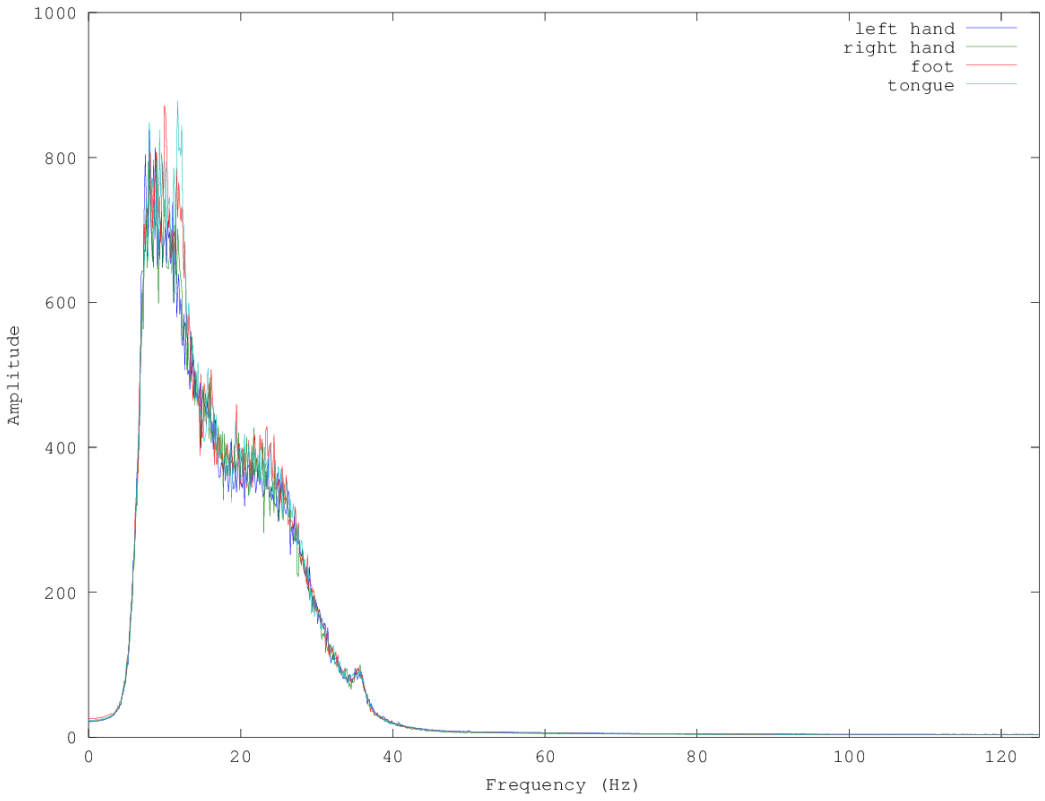

FIGURE 5.2: FFT plots calculated for each motor imagery class (subject 1, electrode $C4$). FFT was averaged over trials. The signal was filtered using a 7–30 Hz band-pass filter.

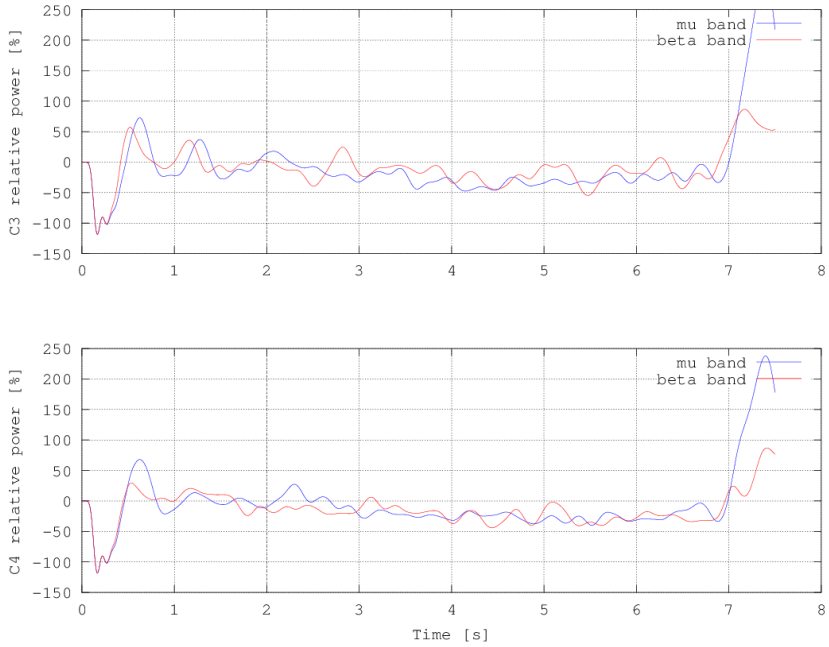

FIGURE 5.3: The event-related desynchronization plotted using the method described in Section 2.3.1. This plot has been done for subject 7, left hand motor imagery, electrodes C3 and C4. The mu band ranges and beta band were chosen to be 8–12 and 14–18 Hz respectively.

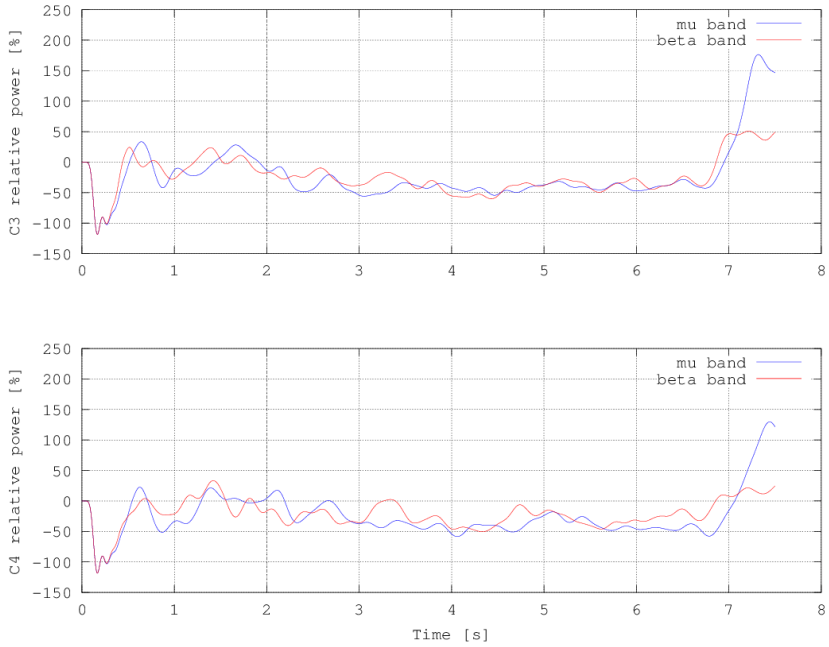

FIGURE 5.4: The event-related desynchronization plotted using the method described in Section 2.3.1. This plot has been done for subject 7, right hand motor imagery, electrodes C3 and C4. The mu band ranges and beta band were chosen to be 8–12 and 14–18 Hz respectively.

Figure 5.4 shows an analogous plot for right hand motor imagery. We can see the ERD and ERS effects on both plots, the beta-band ERD and ERS being stronger for the contralateral hemisphere (electrode $C3$).

5.3 Time-frequency maps for event-related desynchronization

Another way to visualize the ERD/ERS phenomenon are time-frequency maps. Figures 5.5-5.8 shows examples of such maps for subject 7 for left hand, right hand, feet and tongue motor imagery.

Figure 5.5 shows time-frequency changes related to left hand movement imagery. The red colors mark the event-related desynchronization (decrease in power) and the blue ones mark the event-related synchronization (increase in power). All changes are calculated in relation to a baseline period (time interval between seconds 1 and 1.5). This means that if at some moment the power for a specific frequency range is larger than the baseline power for the same frequency, it was marked with blue color.

We can see that for left hand motor imagery (Figure 5.5), the time-frequency maps for both electrodes are fairly similar, the ERD effect being almost equal for both electrodes. The main difference is stronger desynchronization for frequencies 17–18 Hz and 12–13 Hz visible on the C4 electrode.

For right hand motor imagery (Figure 5.6), the ERD phenomenon is clearly stronger for the C3 electrode for a wide range of frequencies (8–12 and 16–24 Hz).

Figure 5.7 shows that the event-related desynchronization is much weaker on both electrodes for feet motor imagery than in case of left and right hand motor imagery. Some small differences between the two electrodes can be visible, e.g. at frequency ranges 8–12, 22–23 and 27–29 Hz.

The time-frequency map for tongue motor imagery (Figure 5.8) shows that the ERD effect for tongue is the least intense among all motor imagery tasks.

Another interesting, but unexpected, observation is that event-related synchronization does not seem to be useful to differentiate between motor imagery tasks for subject 7, because in all cases the ERS effect was stronger in electrode $C3$.

In order to show that the ERD/ERS effect varies among different subjects, we also present a time-frequency plot for subject 1 for left hand motor imagery tasks (Figure 5.9). Comparing to subject 7 (Figure 5.5) we can see that the ERD effect is much weaker for subject 1 and the ERS effect is not visible at all.

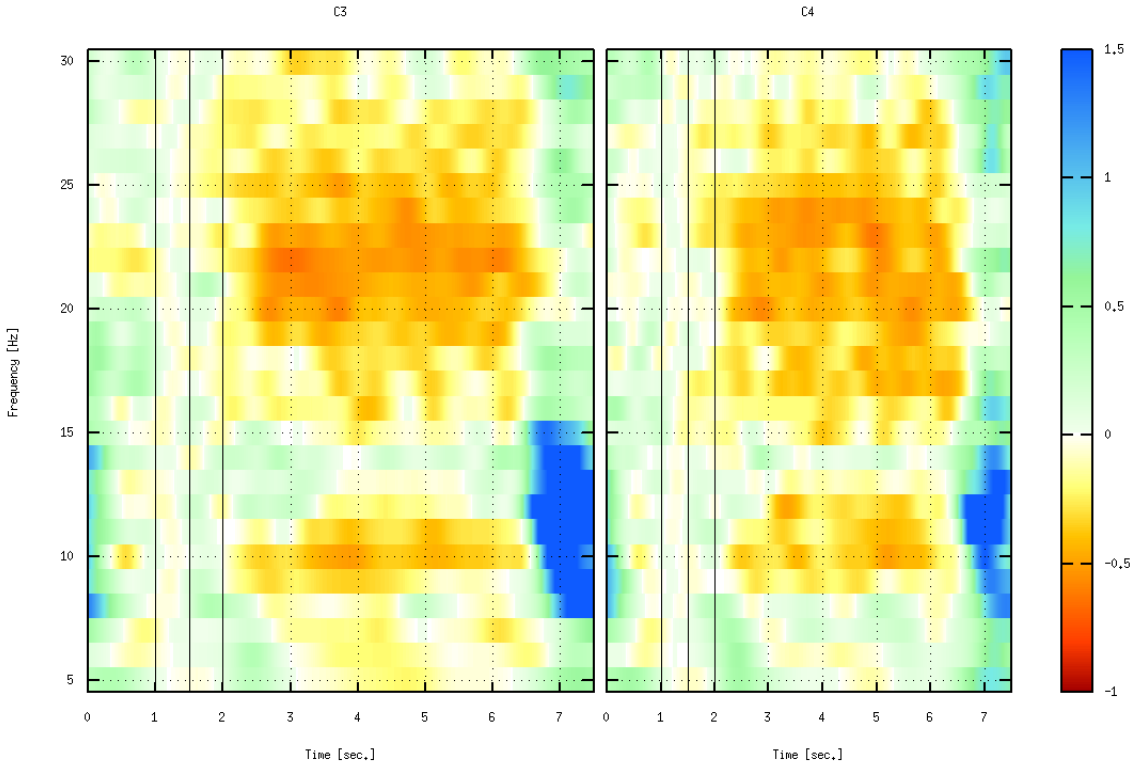

FIGURE 5.5: Time-frequency map of ERD/S, subject 7 imagining a movement with his left hand. Red color designates event-related desynchronization, blue — event-related synchronization. As a baseline the power in seconds $1.0 - 1.5$ was used.

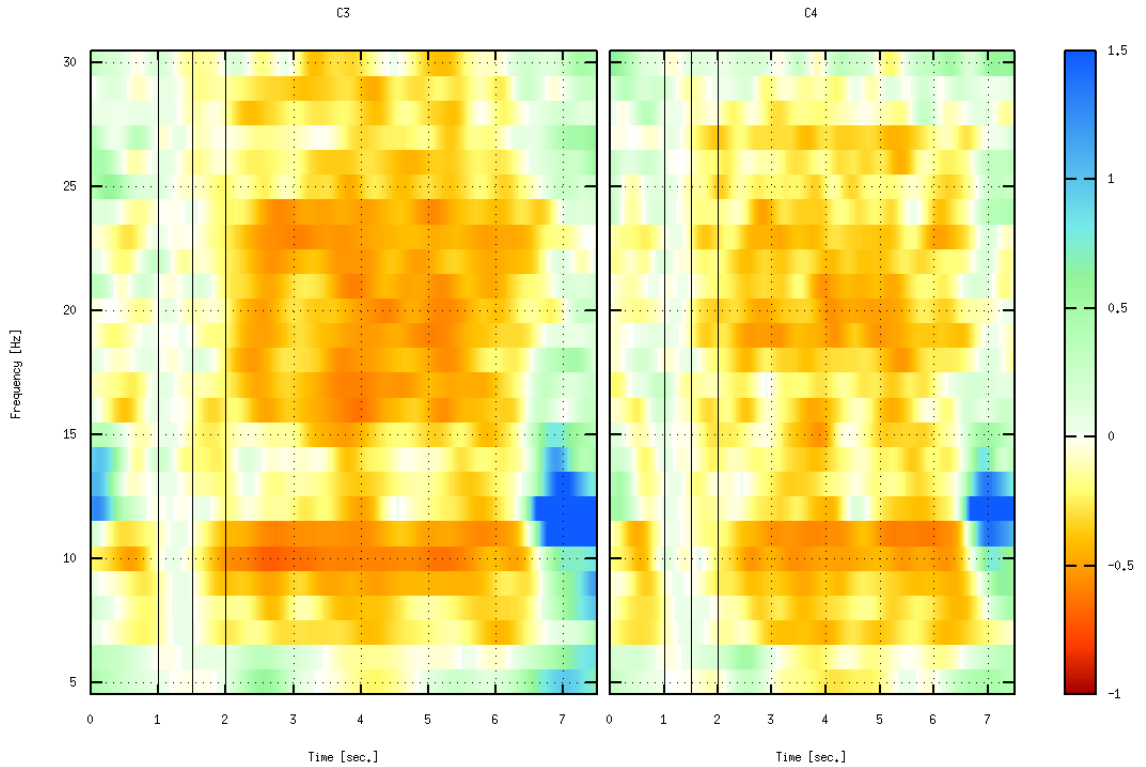

FIGURE 5.6: Time-frequency map of ERD/S, subject 7 imagining a movement with his right hand.

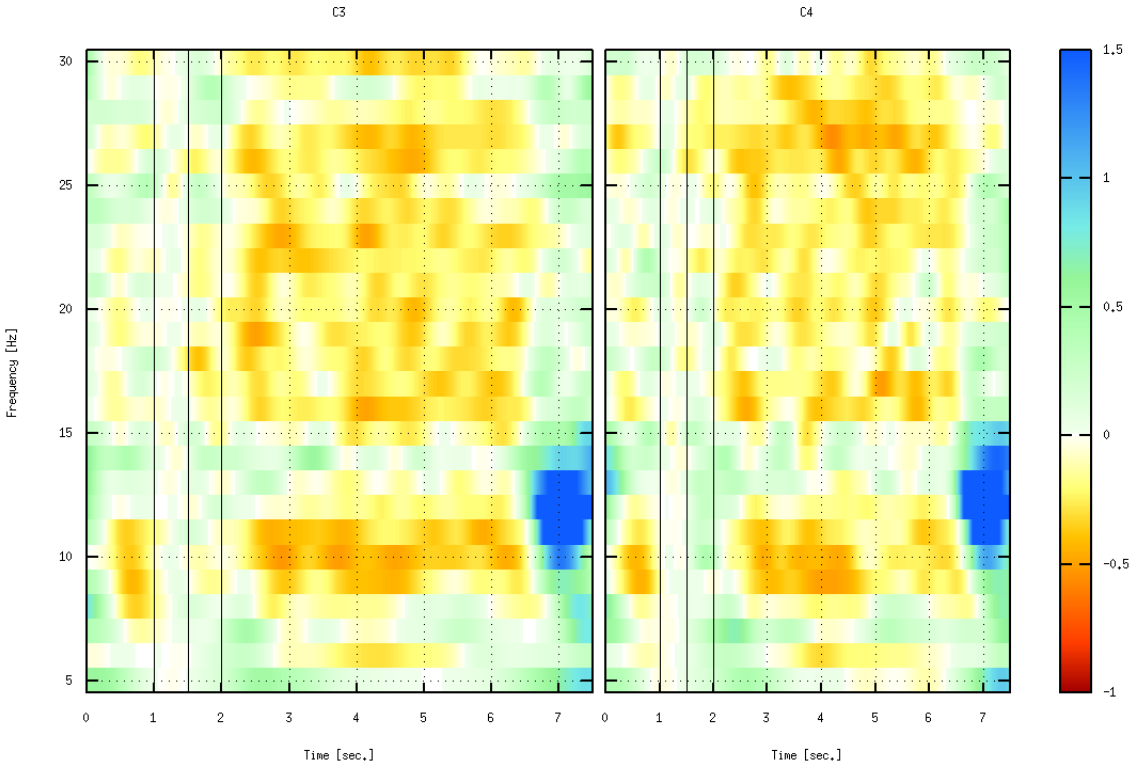

FIGURE 5.7: Time-frequency map of ERD/S, subject 7 imagining a movement with his feet.

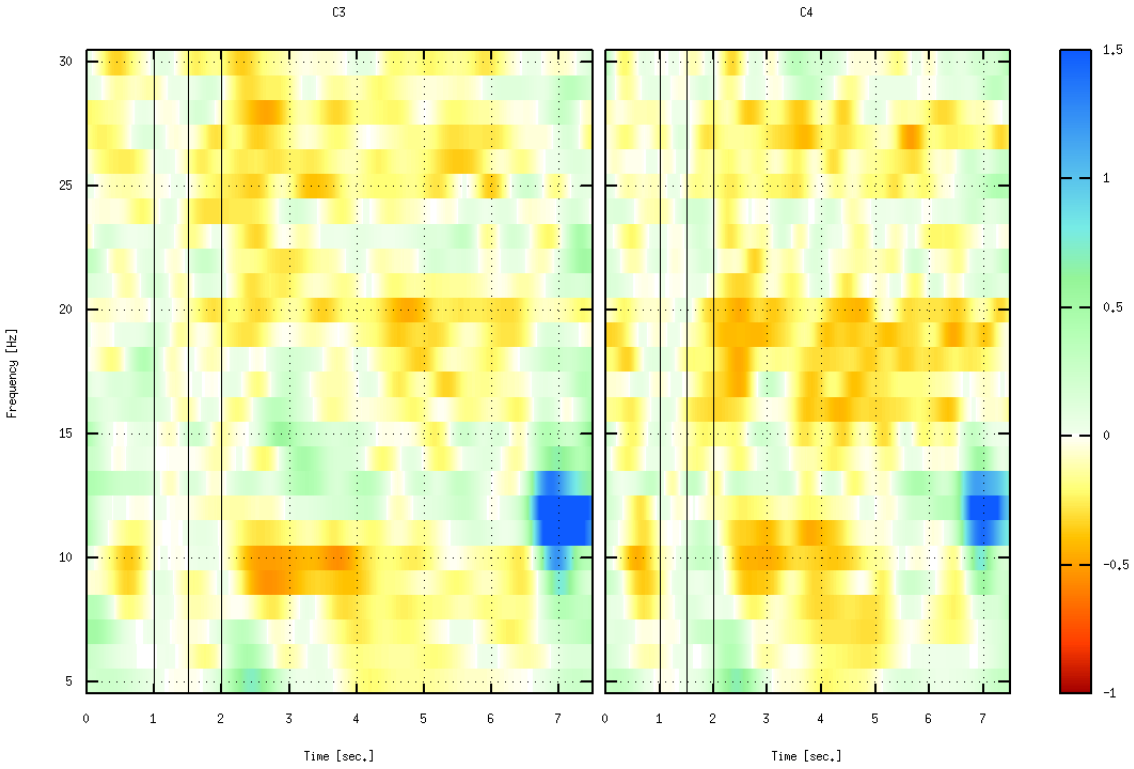

FIGURE 5.8: Time-frequency map of ERD/S, subject 7 imagining a movement with his tongue.

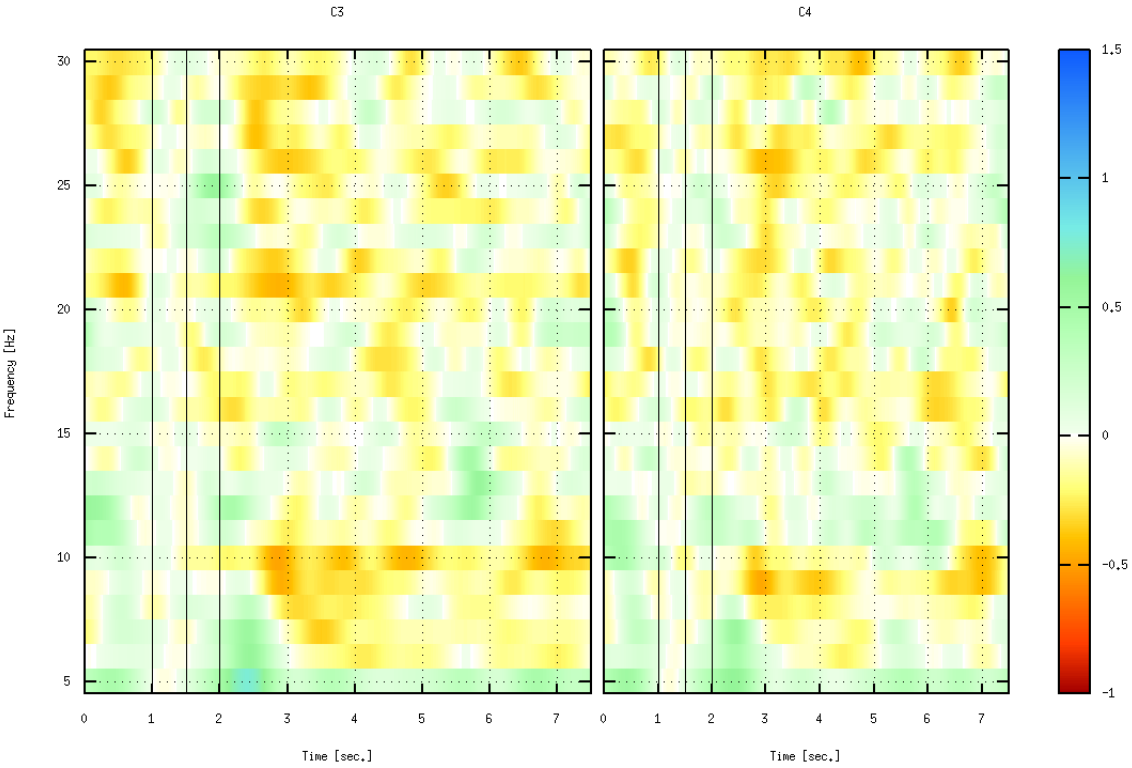

FIGURE 5.9: Time-frequency map of ERD/S, subject 1 imagining a movement with his left hand.

Chapter 6

Experiments and results

In this chapter we discuss the results obtained using different variations of classification algorithms. In the first and second section we present the cross-validation results obtained using the band power and time domain parameters features. For each feature extraction technique we test and compare two classifiers: *linear discriminant analysis* (LDA) and *support vector machines* (SVM). In the last section we show the results obtained on the test set using the best classifiers from each group.

6.1 Band power features

6.1.1 Linear discriminant analysis (LDA) classifier

For the first experiment we implemented a simple classification algorithm which uses band power features and a *linear discriminant analysis* (LDA) classifier. The algorithm consisted of the following steps:

1. Preprocessing:
    a) EOG channels removal — according to the BCI Competition rules, the EOG channels should not be used for classification, thus these channels were ignored in further processing. EOG channels could have been used to reduce artifacts, e.g. when removing artifacts with the linear regression method, but with the band-pass filtering artifact removal method these channels are useless.
    b) Artifact reduction using band-pass filtering (a 7–30 Hz fifth-order Butterworth band-pass filter was used).
2. Feature extraction — band power in 8–12 and 14–18 Hz frequency bands were used as features (smoothing window size — 1 second).
3. Classification using the linear discriminant analysis classifier.

Tables 6.1-6.3 show the results obtained for the algorithm when we run it for each subject using the 8-fold cross-validation described in section 4.4.

Table 6.1 shows that this algorithm obtained 0.34 kappa by cross-validation and 0.49 kappa on the training set. The total processing time of the experiment (loading the signal, extracting the features, performing cross-validation) was 69.07 minutes. Since there were 22 channels (3 EOG channels were excluded) and for each channel two frequency bands were calculated, there were 44 features used as an input for the classifier.

Table 6.2 shows the kappa values for each subject. There are strong intersubject differences visible, e.g. the best classification rates were obtained for subject 9, and the worst — for subject 5. This can be explained by the BCI illiteracy phenomenon described in Section 2.3.2.

TABLE 6.1: The overall statistics for the first implemented algorithm which used band power features (frequency range 8–12, 14–18 Hz) and an LDA classifier).

number of features	training set kappa	cross-validation kappa	processing time [min]
44	0.49	**0.34**	69.07

TABLE 6.2: The kappa values for each subject for the first implemented classification algorithm (band power features, LDA classifier) obtained using the cross-validation technique on the training set.

subjects								
1	2	3	4	5	6	7	8	9
0.38	0.22	0.48	0.22	0.13	0.18	0.39	0.47	0.58

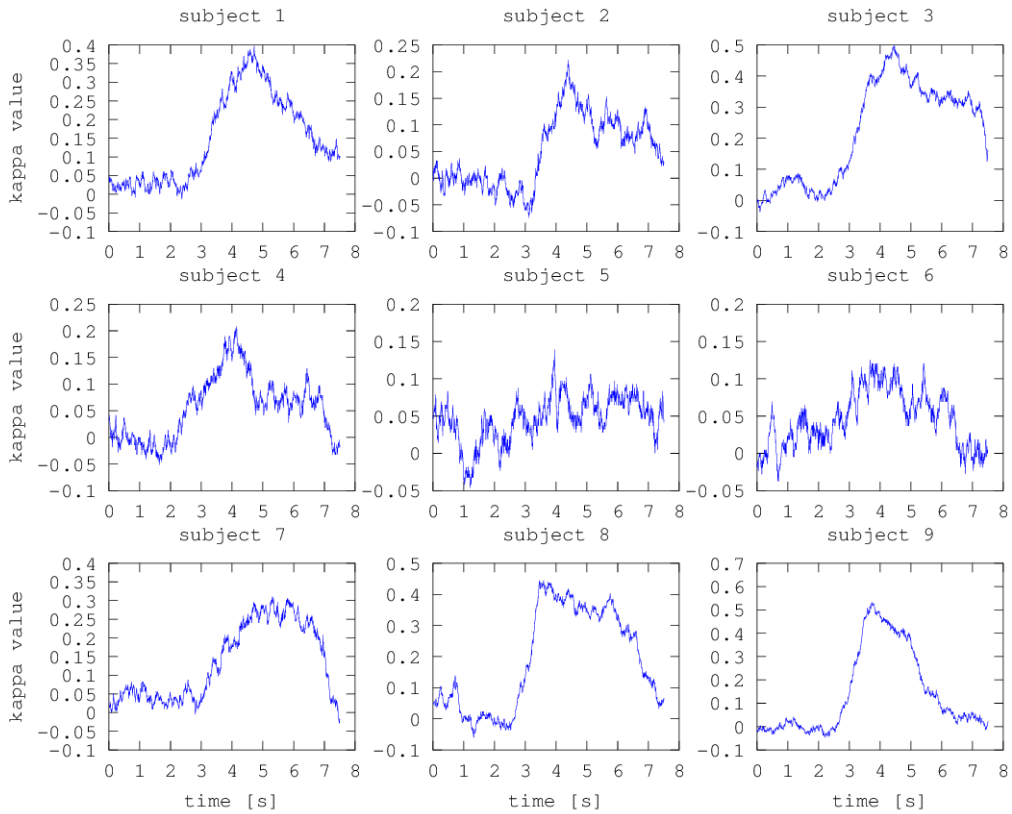

FIGURE 6.1: The time-courses of kappa values for each subject.

Figure 6.1 shows the time courses of kappa values for each subject. The time at which kappa reaches its maximum value for each subject is presented in Table 6.3. The kappa values for all subjects oscillate around zero for seconds between 0 and 3 and reach their maximum points somewhere between the third and the sixth second of the trial. This happens because at the beginning of each trial only a fixation cross is shown on the screen and the subject does not even know which

Table 6.3: The times (in seconds) at which the kappa value was the largest for each subject.

subjects									
1	2	3	4	5	6	7	8	9	mean
4.72	4.40	4.46	4.14	3.94	3.67	5.30	3.46	3.74	4.2

motor imagery he will be requested to perform in a moment. Thus, neither the user, nor the classifier should at this time be able to respond accordingly to the cue which will be presented in the future. On the other hand, the cue was presented to the subject at second 2, so in seconds 3–6 the subject was probably executing the motor imagery task, which resulted in some event-related desynchronization. In that case, the features extracted from that range contain some ERD-related information which could have been used to correctly classify between the four motor imagery tasks.

An important lesson learned from this step is that we do not need to include the whole duration of the trials in the training set, because some time ranges do not contain any information that could be used to properly classify the motor imagery tasks, and what is more, may introduce some noise in the training data set.

Trimming

In this step, the effect that trimming the learning signal has on the classification accuracy is examined. Six different time ranges were checked for their influence on classification rates. For example, if the signal is trimmed in 4–5 seconds range it means that only features calculated for the samples that are within this range are used as learning examples by the classifier.

Table 6.4 shows the results obtained by trimming the trial signal to different time ranges. The best results were obtained if the signal was trimmed to 3–6 seconds time range (or 3.5–6.0) and the worst — when the signal was trimmed to 4.5–5.0 seconds. The conclusion is that removing unnecessary, noisy data (e.g. seconds from zero to three) from the trial improved the classification accuracy only slightly (although it is possible that this effect may be due to different assignment of trials to groups when performing the cross-validation procedure). On the other hand, removing too much information is disadvantageous. The latter results from the fact, that each subject has his strongest ERD effect for a different moment of time in the trial, because each subject could start (or end) performing the motor imagery task at a slightly different time.

In the further experiments we will trim the trials to 3.5–6.0 seconds intervals. Although we are not sure about whether or not it improves the classification accuracy, it is visible that it significantly reduces the processing time of the algorithm.

Table 6.4: The effect of trimming on the classification accuracy.

trimming range [sec]	number of features	train kappa	**cross-validation kappa**	processing time [min]
no trimming	44	0.41	**0.32**	69.07
3–6.5	44	0.47	**0.35**	32.87
3.5–6.0	44	0.49	**0.35**	23.90
4.0–6.0	44	0.49	**0.33**	19.48
4.0–5.5	44	0.51	**0.32**	14.97
4.0–5.0	44	0.54	**0.31**	10.50
4.5–5.0	44	0.54	**0.27**	6.07

Downsampling

In the previous step, two best trim ranges were found. for further processing we prefer the shorter one — although both seem to give similar classification results, the processing time is shorter if the 3.5–6.0 seconds time interval is used.

Downsampling is another technique for reducing the amount of data in the signal. Downsampling reduces the sampling frequency of the signal by removing some samples from it. How many samples are removed is described by downsampling factor. For example, if the signal with its sampling frequency equal to 200 Hz is downsampled by a factor of $M = 4$, the sampling frequency of the resulting signal will be equal to 50 Hz. Downsampling achieves that by leaving only every M-th sample in the signal.

The classification results for signals downsampled with different downsampling factors is shown in Table 6.5. The classification for this step was performed on the signal which trials were already trimmed to 3.5–6.0 seconds.

It can be seen that increasing the downsampling factor does not have any significant effect on the classification rates and it reduces the processing time significantly, e.g., increasing downsampling factor from 1 (no downsampling) to 2 reduces the processing time by almost 50%.

On the other hand, increasing the downsampling factor should be done with caution, as it reduces the sampling rate of the signal, what in turn reduces the maximum representable frequency in the signal. For example if the sampling frequency of a signal is equal to 250 Hz, the maximum frequency that can be correctly represented in that signal (the Nyquist frequency) is equal to 125 Hz. If the signal is downsampled by a factor of 4, its sampling frequency is reduced to $\frac{250}{4} = 62.5$ Hz, in which case the Nyquist frequency is equal to 31.25 Hz.

Since the event-related desynchronization occurs in mu (8–13 Hz) and beta (13–30 Hz) bands, we do not want to reduce the sampling frequency any further in order not to lose the ERD-related information contained in these bands.

Table 6.5: The effect of downsampling on the classification accuracy. Before downsampling, the data was trimmed to 3.5–6.0 seconds range.

downsampling factor	number of features	train kappa	**cross-validation kappa**	processing time [min]
1	44	0.50	**0.34**	24.00
2	44	0.49	**0.34**	12.97
3	44	0.49	**0.34**	8.73
4	44	0.49	**0.34**	6.62
5	44	0.49	**0.35**	5.35
6	44	0.48	**0.34**	4.54

Smoothing window size

The band power feature extraction algorithm has a smoothing window size parameter, which describes the length of the window in which the band power is averaged (see Section 4.2.1).

Here we modify the value of this parameter in order to determine how does it affect the kappa classification rate. In these experiments, we used the trimming range 3.5–6 seconds and the downsampling factor of 4 as these parameters performed best in the previous steps.

The obtained results are presented in Table 6.6. The classification algorithm performed best when it used a window that was two seconds long. The smoothing window can reduce local fluctuations of power and, what is more, it can also reduce the intersubject and intertrial variations.

We know that all subjects perform the motor imagery task at some moment between the third and sixth second of each trial, but the exact onset and length of the motor imagery execution is dependent on the person. In case of a two-second long window, if it is calculated for samples around 5.5–6 second, it is able to gather some ERD-related information from all subjects.

Table 6.6: The effect of smoothing window size on the classification accuracy.

window size	number of features	train kappa	**cross-validation kappa**
0.5	44	0.39	**0.28**
1.0	44	0.49	**0.33**
1.5	44	0.56	**0.38**
2.0	44	0.59	**0.39**
2.5	44	0.60	**0.38**
3.0	44	0.59	**0.36**

Frequency bands

In this step we modify the ranges of the frequency bands in order to determine which frequency bands are the most useful for the classifier. As for the other parameters, the best values obtained in the previous steps were used.

Firstly, the performance of the classifier was tested when only one frequency band was used. The results for different frequency bands are presented in table 6.7. It can be seen that the best results are obtained when the frequency band is around 8–14 Hz. Interestingly, the result for this frequency band is better than the results obtained when two frequency bands were used in the previous steps. The algorithm used 22 features because there were 22 EEG channels and for each channel the band power in the given frequency band was used as a feature.

Table 6.7: The effect of frequency band ranges selection on the classification accuracy when only one frequency band is used.

frequency band	number of features	train kappa	**cross-validation kappa**
8–12	22	0.49	**0.37**
7–12	22	0.49	**0.36**
8–11	22	0.43	**0.29**
8–13	22	0.52	**0.40**
8–14	22	0.53	**0.42**
8–15	22	0.53	**0.42**
9–14	22	0.53	**0.41**
14–18	22	0.34	**0.18**
16–20	22	0.36	**0.20**
18–22	22	0.39	**0.24**
20–24	22	0.40	**0.24**
22–26	22	0.38	**0.23**
24–28	22	0.32	**0.15**
26–30	22	0.29	**0.11**

Next, a second frequency band was added to the feature set to test if using more than one frequency bands improves the performance of the classifier. Results presented in Table 6.8 show that, in the best case, the cross-validation kappa did increase only by 0.02 when a second band around 21–25 Hz was added.

This result was further investigated (Table 6.9) in order to check if it is possible to tune up the second band's frequencies in order to obtain better classification results. A slightly better value was obtained if the second band was equal to 20–26 Hz, however this difference may be as well a result of chance (different assignment of trials to k cross-validation folds).

TABLE 6.8: The effect of frequency band ranges selection on the classification accuracy when two frequency bands are used.

frequency bands	number of features	train kappa	**cross-validation kappa**
8–14, 14–18	44	0.62	**0.41**
8–14, 16–20	44	0.62	**0.41**
8–14, 18–22	44	0.63	**0.42**
8–14, 20–24	44	0.63	**0.43**
8–14, 21–25	44	0.63	**0.44**
8–14, 22–26	44	0.63	**0.43**
8–14, 24–28	44	0.61	**0.41**
8–14, 26–30	44	0.59	**0.39**

TABLE 6.9: The effect of frequency band ranges selection on the classification accuracy when two frequency bands are used. The second band frequencies are variations around 21–26 Hz.

frequency bands	number of features	train kappa	**cross-validation kappa**
8–14, 21–25	44	0.63	**0.44**
8–14, 22–25	44	0.63	**0.44**
8–14, 21–24	44	0.62	**0.42**
8–14, 20–26	44	0.64	**0.45**

Table 6.10 shows results obtained when more than two frequency bands were used. We can see that the best result is obtained if the 14–20 Hz band is added to the best so far frequency bands. However the cross-validation kappa for this example is equal regardless of the 14–20 Hz band being used or not. Thus, we assume that the additional frequency band does not add any valuable information for the classification algorithm.

TABLE 6.10: The effect of frequency band ranges selection on the classification accuracy when three frequency bands are used.

frequency bands	number of features	train kappa	**cross-validation kappa**
8–11, 11–14, 20–26	66	0.71	**0.44**
8–14, 20–23, 23–26	66	0.68	**0.42**
8–14, 14–20, 20–26	66	0.71	**0.45**
8–14, 20–26, 26–30	66	0.70	**0.43**
8–11, 11–14, 20–23, 23–26	66	0.71	**0.44**
8–14, 14–20, 20–26, 26–30	88	0.77	**0.45**

Spatial filtering

The last step consisted of adding a CSP spatial filter to the algorithm. Table 6.11 shows the classification results if a CSP filter was used with only one frequency band. Comparing to results presented in Table 6.7 we can see that the cross-validation kappa for all frequency bands increased by 10% if the filter was used.

Moreover, the CSP algorithm reduced the number of features from 22 to 8. This is because the CSP transformation matrix reduced the 22 channels to 8 best CSP components that maximize differences between each motor imagery class. This may have helped to reduce the overfitting effect, as the difference between the training set and validation set kappa value was also reduced.

Tables 6.12–6.13 show the results for two and three frequency band classification when a CSP filter is used. These are also significantly better than in the case when no spatial filter was used. The best result is obtained with three frequency bands (8–15, 19–24, 24–30 Hz).

Table 6.11: The effect of frequency band ranges selection on the classification accuracy when only one band, a CSP filter and an LDA classifier are used.

frequency bands	number of features	train kappa	**cross-validation kappa**
7–12	8	0.50	**0.45**
8–11	8	0.45	**0.40**
8–13	8	0.54	**0.49**
8–14	8	0.55	**0.50**
8–15	8	0.55	**0.51**
9–14	8	0.54	**0.49**
11–15	8	0.51	**0.46**
12–16	8	0.44	**0.38**
14–18	8	0.34	**0.26**
16–20	8	0.36	**0.30**
18–22	8	0.41	**0.35**
20–24	8	0.42	**0.36**
22–26	8	0.40	**0.34**
24–28	8	0.36	**0.29**
26–30	8	0.30	**0.23**

Table 6.12: The effect of frequency bands selection when two bands, a CSP filter and an LDA classifier are used.

frequency bands	number of features	train kappa	**cross-validation kappa**
8–15, 16–20	16	0.61	**0.55**
8–15, 18–22	16	0.63	**0.56**
8–15, 19–23	16	0.63	**0.56**
8–15, 19–24	16	0.63	**0.57**
8–15, 19–25	16	0.64	**0.56**
8–15, 20–23	16	0.62	**0.55**
8–15, 20–24	16	0.63	**0.56**
8–15, 20–25	16	0.63	**0.56**
8–15, 21–23	16	0.61	**0.53**
8–15, 21–24	16	0.62	**0.55**
8–15, 21–25	16	0.62	**0.55**
8–15, 22–26	16	0.62	**0.55**
8–15, 24–28	16	0.61	**0.54**
8–15, 26–30	16	0.61	**0.53**

6.1.2 Support vector machines (SVM) classifier

In this section we check if changing the classifier can further improve the classification results obtained so far. We decided to try the support vector machines (SVM), because it is a very popular algorithm and currently it is considered to be a state-of-the-art classification method [BHW10].

TABLE 6.13: The effect of frequency bands selection when three or more bands, a CSP filter and an LDA classifier are used.

frequency bands	number of features	train kappa	**cross-validation kappa**
8–15, 19–22, 22–24	24	0.65	**0.56**
8–12, 12–15, 19–24	24	0.65	**0.56**
8–15, 15–19, 19–24	24	0.66	**0.56**
8–15, 19–24, 24–30	24	0.67	**0.58**
8–12, 12–15, 19–22, 22–24	32	0.68	**0.56**
8–15, 15–19, 19–24, 24–30	32	0.70	**0.58**

The SVM algorithm is also widely used in EEG signal classification [BFWB07], for example Schlögl et al. compared several classification algorithms on motor imagery data and among all classifiers that were tested SVM performed best [SLBP05].

In our preliminary experiments with the SVM algorithm using a linear kernel we did not obtain any interesting results. Thus in this section we present results obtained using the SVM with a Gaussian kernel.

The classifier parameters (C parameter and the kernel width σ) were fine-tuned using a grid-search method which tests the performance of all possible combinations from a given sets of both parameters [HCL+03]. Due to long learning time of the SVM classifier, only every 30-th sample from each trial was used as a training example for the classifier (as the amount of data was already reduced with downsampling by a factor of four, this means that actually only every 120-th sample from each trial was used for training).

Table 6.14 shows the kappa values for different values of kernel width (σ) and C parameter. We can observe the standard SVM behavior while changing the parameters values, that is — when C parameter is decreased, or kernel width is increased, the overfitting effect is reduced (a decrease of test set kappa is visible). The best values of kappa were obtained for three (C, σ) parameters pairs: (1, 100), (10, 100) and (10, 1000). We can see that in our case using the SVM gave 0.02 worse results than the LDA classifier.

TABLE 6.14: The effect the kernel width and C parameter have on classification performance (measured using the kappa coefficient) using band power features and an SVM classifier. The values in the round brackets are classification results for the training set, the values without brackets were obtained through cross-validation.

	C parameter					
kernel width	0.01	0.1	1	10	100	1000
10	0.49	0.53	0.56	0.52	0.45	0.44
	(0.54)	(0.62)	(0.75)	(0.91)	(0.99)	(1.00)
100	0.43	0.47	**0.56**	**0.56**	0.55	0.51
	(0.45)	(0.50)	(0.62)	(0.68)	(0.76)	(0.89)
1000	0.37	0.37	0.47	**0.56**	0.56	0.56
	(0.38)	(0.38)	(0.49)	(0.62)	(0.66)	(0.69)
10000	0.36	0.36	0.36	0.46	0.55	0.56
	(0.37)	(0.38)	(0.38)	(0.49)	(0.62)	(0.65)
100000	0.36	0.35	0.36	0.36	0.46	0.55
	(0.37)	(0.37)	(0.37)	(0.37)	(0.49)	(0.62)

6.2 Time domain parameters features

In this section time domain parameters were used instead of band power features. Optimal parameters for the method were chosen and the examples were classified using the LDA and SVM classifiers.

6.2.1 LDA classifier

First, the classification accuracy was compared for different trimming ranges. Table 6.15 shows that the best results were obtained if the trimming range for the time domain parameters was 4.0–6.0 seconds. The processing time of the algorithm is similar as in the case of band power features (cf. Table 6.5).

Table 6.15: The effect of trimming on the kappa classification rate (TDP features with 9 derivatives and $u = 0.0085$, LDA classifier, downsampling factor: 4).

trimming range [s]	number of features	train kappa	**cross-validation kappa**	processing time [min]
3–6	220	0.78	**0.45**	9.66
3.5–6	220	0.79	**0.46**	8.27
4–6	220	0.81	**0.47**	7.01
4–5	220	0.82	**0.40**	4.43

Next, the effect that the number of derivatives calculated for the time domain parameters have on the classification rates was estimated (table 6.16). The best results were obtained for five derivatives and more.

For further experiments we decided to use five derivatives in order to keep the number of features at a minimum level (the more derivatives are calculated, the bigger the number of features is). For 5 derivatives, the number of features is equal to 48, because there are 8 channels obtained as a result of using the CSP filter and five derivatives plus the original signal are used as features.

Table 6.16: The effect of number of derivatives on the classification result (TDP features, $u = 0.0085$, LDA classifier, CSP filter).

number of derivatives	number of features	train kappa	**cross-validation kappa**
1	16	0.62	**0.53**
2	24	0.66	**0.55**
3	32	0.68	**0.56**
4	40	0.69	**0.56**
5	48	0.70	**0.57**
6	56	0.70	**0.56**
7	64	0.70	**0.57**
8	72	0.71	**0.57**
9	80	0.71	**0.57**
10	88	0.71	**0.57**
11	96	0.71	**0.57**
12	104	0.71	**0.57**

Table 6.17 shows how the classification accuracy changes, when the exponential window size is changed. The best results were obtained for $u = 0.015$. As explained in section 4.2.2 the u parameter is responsible for the size of the moving average filter that is applied after the power of TDP derivatives is calculated. For $u = 1$ the result is as if no filter was applied (see equa-

tion 4.7). As u decreases, the moving average window size increases. We can see that the optimal kappa values are obtained for values near $u = 0.015$.

TABLE 6.17: The effect of the exponential window constant u on the classification accuracy (TDP features with five derivatives, LDA classifier, CSP filter).

u	number of features	train kappa	**cross-validation kappa**
0.0001	48	0.43	**0.10**
0.005	48	0.68	**0.51**
0.01	48	0.70	**0.56**
0.0125	48	0.69	**0.58**
0.015	48	0.69	**0.58**
0.0175	48	0.68	**0.58**
0.2	48	0.40	**0.37**
0.3	48	0.36	**0.33**
0.4	48	0.34	**0.31**
0.5	48	0.33	**0.30**
0.6	48	0.31	**0.29**
0.7	48	0.30	**0.27**
0.8	48	0.30	**0.28**
0.9	48	0.29	**0.26**
1.0	48	0.25	**0.24**

6.2.2 SVM classifier

The TDP features were also tested using an SVM classifier with a Gaussian kernel. The results obtained by grid-searching the C and σ parameter values are shown in table 6.18. The SVM classifier performed best, reaching 0.55 kappa value through cross-validation procedure, when $C = 100$ and $\sigma = 100$. We can see that, like in case of band power features, the SVM was unable to perform better than the simpler LDA classifier.

TABLE 6.18: The effect the kernel width and C parameter have on classification performance (measured using the kappa coefficient) using TDP features and an SVM classifier. The values in the round brackets are classification results for the training set.

	C parameter						
kernel width	0.01	0.1	1	10	100	1000	10000
10	0.44	0.50	0.53	0.52	0.47	0.44	0.42
	(0.49)	(0.56)	(0.68)	(0.82)	(0.95)	(1.00)	(1.00)
100	0.41	0.45	0.52	0.53	**0.55**	0.52	0.48
	(0.44)	(0.48)	(0.57)	(0.64)	(0.72)	(0.81)	(0.92)
1000	0.35	0.34	0.43	0.52	0.53	0.54	0.54
	(0.36)	(0.36)	(0.46)	(0.57)	(0.61)	(0.67)	(0.73)
10000	0.34	0.33	0.33	0.44	0.53	0.54	0.54
	(0.35)	(0.35)	(0.35)	(0.46)	(0.57)	(0.61)	(0.65)
100000	0.34	0.34	0.34	0.33	0.43	0.53	0.53
	(0.35)	(0.35)	(0.35)	(0.35)	(0.45)	(0.57)	(0.61)

6.3 Final results on the test set

6.3.1 Our results

To this point we did not use the test set and all the results presented so far were obtained using cross-validation on the training set. To compare our algorithms with the BCI Competition results we chose the four best algorithms from each group: band power features with LDA and SVM classifiers and time domain parameters, also with both classifiers.

The band power features (BP) were used with the following parameters:

- trial trimmed to 3.5–6.0 seconds,
- smoothing window size — 2 seconds,
- frequency bands 8–14, 19–24 and 24–30 Hz.

The time domain parameters (TDP) used:

- trials trimmed to 4.0–6.0 seconds,
- five derivatives,
- moving average exponential window of size $u = 0.015$.

In all cases the data was downsampled by a factor of 4 and filtered using a 7–30 Hz band-pass filter. The SVM classifier used the following (C, σ) pairs:

- $C = 1$, $\sigma = 100$, when used with band power features,
- $C = 100$, $\sigma = 100$, when used with time domain parameters.

TABLE 6.19: The final evaluation results for the chosen algorithms that were tested in this chapter.

number of features	features	classifier	training kappa	cross-validation kappa	test kappa
24	BP	LDA	0.61	0.58	**0.45**
24	BP	SVM	0.56	0.56	**0.44**
48	TDP	LDA	0.62	0.58	**0.41**
48	TDP	SVM	0.67	0.55	**0.34**

The final results for the algorithms are shown in Table 6.19. We can see that for each algorithm the kappa on the training set was the highest, the cross-validation kappa was lower, and the kappa on the test set was the lowest. The situation that the training set kappa was the highest is normal in machine learning and comes from the fact that the classifier uses the training set to learn how to classify data. Thus, it is able to classify better the data that it has already seen than the data that it did not yet see. The difference between the cross-validation kappa and the test set kappa can be explained by the fact that the test set comes from a different recording session than the training and the cross-validation sets (the problem behind this was already explained in Section 3.2.2).

The best results on the test set were obtained when band power features were used. In case of using the LDA classifier the kappa value on the testing set was equal to 0.45, in case of SVM — 0.44. The worse results were obtained if TDP features and the SVM classifier were used (0.34). Interestingly, this algorithm obtained the best results on the training set, which shows that the problem with this algorithm is overfitting.

Although the results obtained when testing all algorithms using cross-validation were very similar to each other (ranging from 0.55 to 0.58 kappa), the results on the final test set show a much wider variability (from 0.34 to 0.45 kappa). This shows that different algorithms are able to generalize the information learned from the training set to the test set to various degrees.

TABLE 6.20: The final evaluation kappa values for each subject.

features	classifier	subjects								
		1	2	3	4	5	6	7	8	9
BP	LDA	**0.74**	0.11	0.53	0.34	**0.18**	**0.31**	**0.62**	**0.61**	**0.63**
BP	SVM	0.73	**0.29**	**0.56**	**0.37**	0.13	0.29	0.51	0.55	0.57
TDP	LDA	0.70	0.20	0.49	0.31	**0.18**	0.29	0.52	0.49	0.54
TDP	SVM	0.60	0.08	0.48	0.25	0.17	0.20	0.42	0.42	0.46

The results for each subject for all four algorithms are presented in Table 6.20. We can see that the best results for most subjects were achieved when using the LDA classifier with band power features. However, for three subjects better results were obtained with the BP+SVM algorithm, and in case of subject 2 the SVM classifier performed over two times as well as the LDA classifier. This fact could possibly be used to further improve the classification accuracy of the algorithm if — at the cross-validation step — the classifier giving the best accuracy would be chosen for each subject.

TABLE 6.21: Kappa values obtained by the competitors in the BCI Competition IV, data set 2*a* [BCI13] with our algorithm included on the third place.

no	contributor	mean kappa	subjects								
			1	2	3	4	5	6	7	8	9
1	Kai Keng Ang	**0.57**	0.68	**0.42**	**0.75**	**0.48**	**0.40**	0.27	**0.77**	**0.75**	0.61
2	Liu Guangquan	**0.52**	0.69	0.34	0.71	0.44	0.16	0.21	0.66	0.73	**0.69**
3	Piotr Szachewicz	**0.45**	**0.74**	0.11	0.53	0.34	0.18	**0.31**	0.62	0.61	0.63
4	Wei Song	**0.31**	0.38	0.18	0.48	0.33	0.07	0.14	0.29	0.49	0.44
5	Damien Coyle	**0.30**	0.46	0.25	0.65	0.31	0.12	0.07	0.00	0.46	0.42
6	Jin Wu	**0.29**	0.41	0.17	0.39	0.25	0.06	0.16	0.34	0.45	0.37

Comparing with the final BCI competition results (Table 3.1) we can see that all classifiers tested on the final testing set would have been able to reach the third place in the competition. Table 6.21 shows the final results of the BCI Competition for data set 2a including the BP+LDA algorithm implemented in this thesis. We can see that the result obtained by our algorithm was close to the two best algorithms and, what is more, for two out of nine subjects our algorithm performed better than all other algorithms.

6.3.2 Other competitor's methods

As an example, we shortly describe the two best algorithms for the BCI Competition IV.

The algorithm by K. K. Ang et al. used the *Filter Bank Common Spatial Patterns* (FBCSP) method that was developed by them [ACZG08]. This algorithm, instead of applying the common spatial patterns method to one frequency band (in our case it was 7–30 Hz), applied it several times to smaller frequency bands, e.g. 4–8, 8–12, 12–16 Hz. After that the most discriminative features were selected using Mutual Information Best Individual Features. The algorithm consisted of four Naive Bayes Parzen Window classifiers and the final classification was based on the probability values provided by these classifiers (the classification used the one-vs.-rest strategy).

The algorithm by Liu Guangquan et al. used a CSP filter applied to 8–30 Hz filtered signal. *Log-variances* of the signals were used as features and linear discriminant analysis was used to reduce the dimensionality of the feature vector. The final classification was done using a Bayesian classifier.

Chapter 7

Conclusion

In this thesis we implemented an algorithm for four-class motor imagery *electroencephalogram* (EEG) signal classification based on the *event-related desynchronization* (ERD) phenomenon. We tested two types of feature extraction techniques: *band power features* and *time domain parameters* (TDP), and two classifiers: *support vector machines* (SVM) and *linear discriminant analysis* (LDA). We tested four classification algorithms using all four combinations of the aforementioned feature extraction techniques and classifiers. We used the 8-fold cross-validation method on the training set and through this procedure we chose optimal parameter values.

We discovered that trimming the trials to 3.5–6.0 seconds ranges slightly improves the classification accuracy. We used a downsampling factor of 4, because it reduced the processing time of the algorithm and did not change the algorithm's classification results. Better classification results were obtained when the band power features were used (comparing to time domain parameters). The classification accuracy for time domain parameters (TDP) was slightly worse when calculated using cross-validation and significantly worse when calculated on the test set.

We hypothesize that the TDP features are not very effective when the training and testing data come from different recording sessions, as was the case in the BCI Competition IV. We can also see that the fact that the training and testing data sets come from different days had also a significant influence on other algorithms — for every feature extraction technique/classifier pair the test kappa was at least 12% smaller than the kappa obtained through cross-validation.

The *common spatial patterns* (CSP) algorithm increased the classification accuracy by about 10% in case of both feature extraction techniques. Thus, we verified that the CSP algorithm is a very powerful method in BCI signal classification.

Unexpectedly, for both feature extraction techniques, the linear discriminant analysis (LDA) classifier performed better than the support vector machines (SVM) even though we grid-searched the SVM parameters (the C parameter and σ), searching for values that give the best classification results.

The fine-tuned algorithms were evaluated on the testing data set and all of them reached classification rates well above chance level. The kappa value for the best algorithm (which used the LDA classifier with band power features) was equal to 0.45 and the worst one (which used the SVM classifier with TDP features) — 0.34. All of the algorithms would have reached the third place in the BCI Competition IV (data set 2a).

Several improvements may be proposed for the algorithms presented in this thesis. First of all, the parameters for each classifier were selected according to the cross-validation results. The results obtained through cross-validation could be different for each run of the procedure, as the assignment of trials to specific folds is done randomly, thus different assignment may result in a slightly different

kappa value. In order to estimate the classifier's performance more accurately, the experiments should have been executed several times and the results of the experiments should have been averaged.

Another problem with the model selection procedure used in this thesis is that it used a greedy parameter selection algorithm, which can be suboptimal. At each step the algorithm selected a parameter value for which the best classification results were obtained. However, if parameters would have been selected in a different order, possibly a different model would have been chosen.

The third drawback of the presented algorithms is that they did not use any of the *event-related synchronization* (ERS) information that occurred after the motor imagery task was terminated. It is possible that this information would improve the accuracy of the classifier, however no successful results were obtained throughout this thesis (preliminary results, not presented in this thesis, were unsatisfactory).

Future work may include a better algorithm for choosing the frequency bands for which the band power features were computed. In our case, several bands were analyzed from the 7–30 Hz frequency band, in which the ERD/ERS effect is the strongest, and the ones that gave the best classification results were chosen. However, some more automatic procedure would be useful for this task, as for each subject the ERD/ERS phenomenon influences different frequency bands.

Interesting lessons can be also learned from the results obtained by the other competitors. Especially promising is the *filter bank common spatial patterns* (FBCSP) algorithm developed by K. K. Ang et al. [ACZG08], which applies the common spatial patterns (CSP) method to several smaller frequency bands. Another thing that can be tried is using some dimensionality reduction method to reduce the number of features as in the Liu Guangquan's algorithm which reached the second place in the BCI Competition.

Appendix A

Code

The CD attached to this thesis contains all the code used to conduct the experiments described in this work.

The part of the code that was used for the final experiments on the test set data is available in the following public code repository:

`https://github.com/piotr-szachewicz/event-related-desynchronization`.

All the information necessary to install and run the code is available in the `README` file in the root directory of the repository.

Bibliography

[ABK+10] Brendan Z. Allison, Clemens Brunner, Vera Kaiser, Gernot R. Müller-Putz, Christa Neuper, and Gert Pfurtscheller. Toward a hybrid brain–computer interface based on imagined movement and visual attention. *Journal of Neural engineering*, 7(2), 2010.

[ACW+12] Kai Keng Ang, Zheng Yang Chin, Chuanchu Wang, Cuntai Guan, and Haihong Zhang. Filter bank common spatial pattern algorithm on BCI competition IV datasets 2a and 2b. *Frontiers in Neuroscience*, 6, 2012.

[ACZG08] Kai Keng Ang, Zhang Yang Chin, Haihong Zhang, and Cuntai Guan. Filter bank common spatial pattern (fbcsp) in brain-computer interface. In *Neural Networks, 2008. IJCNN 2008.(IEEE World Congress on Computational Intelligence). IEEE International Joint Conference on*, pages 2390–2397. IEEE, 2008.

[AGG07] Brendan Allison, Bernhard Graimann, and Axel Gräser. Why use a BCI if you are healthy. In *ACE Workshop-Brain-Computer Interfaces and Games*, pages 7–11, 2007.

[Alp04] Ethem Alpaydin. *Introduction to machine learning*. The MIT Press, 2004.

[Asa13] Vahid Asadpour. A review of hybrid brain-computer interface systems. *Advances in Human-Computer Interaction*, 2013, 2013.

[BCI13] BCI Competition IV results. `http://www.bbci.de/competition/iv/results/`, March 2013.

[BD06] Katarzyna Blinowska and Piotr Durka. Electroencephalography (EEG). *Wiley Encyclopedia of Biomedical Engineering*, 2006.

[BFWB07] Ali Bashashati, Mehrdad Fatourechi, Rabab K. Ward, and Gary E. Birch. A survey of signal processing algorithms in brain–computer interfaces based on electrical brain signals. *Journal of Neural engineering*, 4(2):32–57, 2007.

[BGH+99] Niels Birbaumer, Nimr Ghanayim, Thilo Hinterberger, Iver Iversen, Boris Kotchoubey, Andrea Kübler, Juri Perelmouter, Edward Taub, and Herta Flor. A spelling device for the paralysed. *Nature*, 398(6725):297–298, 1999.

[BHW10] Asa Ben-Hur and Jason Weston. A user's guide to support vector machines. In *Data mining techniques for the life sciences*, pages 223–239. Springer, 2010.

[Bir06] Niels Birbaumer. Breaking the silence: brain–computer interfaces (BCI) for communication and motor control. *Psychophysiology*, 43(6):517–532, 2006.

[BLL11] Nicolas Brodu, Fabien Lotte, and Anatole Lécuyer. Comparative study of band-power extraction techniques for motor imagery classification. In *Computational Intelligence, Cognitive Algorithms, Mind, and Brain (CCMB), 2011 IEEE Symposium on*, pages 1–6. IEEE, 2011.

[BLMP+] Clemens Brunner, Robert Leeb, Gernot R. Müller-Putz, Alois Schlögl, and Gert Pfurtscheller. BCI Competition 2008 — Graz data set A. `http://www.bbci.de/competition/iv/desc_2a.pdf`. Accessed March 1, 2013.

[Bro00] Joseph D. Bronzino. Principles of electroencephalography. In Joseph D. Bronzino, editor, *The biomedical engineering handbook*. CRC Press, 2000.

[CCC+08] Zhe Chen, Jianting Cao, Yang Cao, Yue Zhang, Fanji Gu, Guoxian Zhu, Zhen Hong, Bin Wang, and Andrzej Cichocki. An empirical EEG analysis in brain death diagnosis for adults. *Cognitive neurodynamics*, 2(3):257–271, 2008.

[CP97] Miguel A. Carreira-Perpinan. A review of dimension reduction techniques. *Department of Computer Science. University of Sheffield. Tech. Rep. CS-96-09*, pages 1–69, 1997.

[CP99] Neuper Christa and Gert Pfurtscheller. Motor imagery and ERD. In Gert Pfurtscheller and Fernando H. Lopes da Silva, editors, *Event-related desynchronization: Handbook of electroencephalography and clinical neurophysiology, revised series (Vol. 6)*. Elsevier, 1999.

[CP06] B. Rael Cahn and John Polich. Meditation states and traits: EEG, ERP, and neuroimaging studies. *Psychological bulletin*, 132(2):180–211, 2006.

[DKŻ+12] Piotr Durka, R Kuś, J Żygierewicz, Magda Michalska, Piotr Milanowski, Maciej Łabęcki, Tomasz Spustek, Dawid Laszuk, Anna Duszyk, and Mateusz Kruszyński. User-centered design of brain-computer interfaces: OpenBCI.pl and BCI Appliance. 2012.

[DSPB76] Richard J. Davidson, Gary E. Schwartz, Eric Pugash, and Edward Bromfield. Sex differences in patterns of EEG asymmetry. *Biological Psychology*, 4(2):119–137, 1976.

[EA99] James R. Evans and Andrew Abarbanel. *Introduction to quantitative EEG and neurofeedback*. Academic Press, 1999.

[EBH97] John W. Eaton, David Bateman, and Søren Hauberg. *Gnu octave*. Free Software Foundation, 1997.

[emo] Emotiv website. `http://www.emotiv.com/`. Accessed June 13, 2013.

[FBW+07] Mehrdad Fatourechi, Ali Bashashati, Rabab K. Ward, Gary E. Birch, et al. Emg and eog artifacts in brain computer interface systems: A survey. *Clinical neurophysiology*, 118(3):480–494, 2007.

[GAP10] Bernhard Graimann, Brendan Allison, and Gert Pfurtscheller. Brain–computer interfaces: A gentle introduction. In *Brain-Computer Interfaces*, pages 1–27. Springer, 2010.

[GSCS75] Anthony Gale, Graham Spratt, Antony J. Chapman, and Adrian Smallbone. EEG correlates of eye contact and interpersonal distance. *Biological Psychology*, 3(4):237–245, 1975.

[GWB08] Moritz Grosse-Wentrup and Martin Buss. Multiclass common spatial patterns and information theoretic feature extraction. *Biomedical Engineering, IEEE Transactions on*, 55(8):1991–2000, 2008.

[Ham10] Brahim Hamadicharef. Brain-Computer Interface (bci) literature — a bibliometric study. In *Information Sciences Signal Processing and their Applications (ISSPA), 2010 10th International Conference on*, pages 626–629. IEEE, 2010.

[HCL+03] Chih-Wei Hsu, Chih-Chung Chang, Chih-Jen Lin, et al. A practical guide to support vector classification, 2003.

[HCP02] C. J. Harland, T. D. Clark, and R. J. Prance. Remote detection of human electroencephalograms using ultrahigh input impedance electric potential sensors. *Applied Physics Letters*, 81(17):3284–3286, 2002.

[HGS07] Hartmut Heinrich, Holger Gevensleben, and Ute Strehl. Annotation: Neurofeedback – train your brain to train behaviour. *Journal of Child Psychology and Psychiatry*, 48(1):3–16, 2007.

[HSN+04] Thilo Hinterberger, Stefan Schmidt, Nicola Neumann, Jürgen Mellinger, Benjamin Blankertz, Gabriel Curio, and Niels Birbaumer. Brain-computer communication and slow cortical potentials. *Biomedical Engineering, IEEE Transactions on*, 51(6):1011–1018, 2004.

[JMM+06] Andrew Jackson, Chet T Moritz, Jaideep Mavoori, Timothy H Lucas, and Eberhard E Fetz. The Neurochip BCI: towards a neural prosthesis for upper limb function. *Neural Systems and Rehabilitation Engineering, IEEE Transactions on*, 14(2):187–190, 2006.

[KB08] Andrea Kübler and Niels Birbaumer. Brain-computer interfaces and communication in paralysis: extinction of goal directed thinking in completely paralysed patients? *Clinical neurophysiology: official journal of the International Federation of Clinical Neurophysiology*, 119(11):2658, 2008.

[Kli99] W. Klimesch. EEG alpha and theta oscillations reflect cognitive and memory performance: a review and analysis. *Brain Research Reviews*, 29(2-3):169–95, 1999.

[Kuś13] Rafal Kuś. Common spatial filtering (CSP). [on-line] `http://brain.fuw.edu.pl/edu/EEG:Pracownia_EEG/CSP`, 2013.

[KVZ+12] Rafal Kus, Diana Valbuena, Jaroslaw Zygierewicz, Tatsiana Malechka, Axel Graeser, and Piotr Durka. Asynchronous BCI based on motor imagery with automated calibration and neurofeedback training. *IEEE transactions on neural systems and rehabilitation engineering: a publication of the IEEE Engineering in Medicine and Biology Society*, 2012.

[LLR+08] Anatole Lécuyer, Fabien Lotte, Richard B Reilly, Robert Leeb, Michitaka Hirose, and Mel Slater. Brain-computer interfaces, virtual reality, and videogames. *Computer*, 41(10):66–72, 2008.

[LMC09] Steven Lemm, Klaus-Robert Müller, and Gabriel Curio. A generalized framework for quantifying the dynamics of EEG event-related desynchronization. *PLoS Computational Biology*, 5(8), 2009.

[LPP+09] M. A. Lopez, H. Pomares, F. Pelayo, J. Urquiza, and J. Perez. Evidences of cognitive effects over auditory steady-state responses by means of artificial neural networks and its use in brain–computer interfaces. *Neurocomputing*, 72:3617–23, 2009.

[LZWG07] Zhonglin Lin, Changshui Zhang, Wei Wu, and Xiaorong Gao. Frequency recognition based on canonical correlation analysis for SSVEP-based BCIs. *Biomedical Engineering, IEEE Transactions on*, 54(6):1172–1176, 2007.

[MB03] Steven G. Mason and Gary E. Birch. A general framework for brain-computer interface design. *Neural Systems and Rehabilitation Engineering, IEEE Transactions on*, 11(1):70–85, 2003.

[MB05] Eduardo Reck Miranda and Andrew Brouse. Interfacing the brain directly with musical systems: on developing systems for making music with brain signals. *Leonardo*, 38(4):331–336, 2005.

[MGPF99] Johannes Müller-Gerking, Gert Pfurtscheller, and Henrik Flyvbjerg. Designing optimal spatial filters for single-trial EEG classification in a movement task. *Clinical neurophysiology*, 110(5):787–798, 1999.

[Mit97] Tom M. Mitchell. *Machine learning*. McGraw-Hill Boston, 1997.

[Moo03] Melody M. Moore. Real-world applications for brain-computer interface technology. *Neural Systems and Rehabilitation Engineering, IEEE Transactions on*, 11(2):162–165, 2003.

[MP95] Jaakko Malmivuo and Robert Plonsey. *Bioelectromagnetism: principles and applications of bioelectric and biomagnetic fields*. Oxford University Press, USA, 1995.

[MPP08] Gernot R. Muller-Putz and Gert Pfurtscheller. Control of an electrical prosthesis with an SSVEP-based BCI. *Biomedical Engineering, IEEE Transactions on*, 55(1):361–364, 2008.

[MRH+10] Emily M. Mugler, Carolin A. Ruf, Sebastian Halder, Michael Bensch, and Andrea Kubler. Design and implementation of a P300-based brain-computer interface for controlling an internet browser. *Neural Systems and Rehabilitation Engineering, IEEE Transactions on*, 18(6):599–609, 2010.

[NAGG12] Luis Fernando Nicolas-Alonso and Jaime Gomez-Gil. Brain computer interfaces, a review. *Sensors*, 12(2):1211–1279, 2012.

[neu] Neurosky store website. `http://store.neurosky.com/products/mindwave-1`. Accessed June 13, 2013.

[NLdS05] Ernst Niedermeyer and Fernando H. Lopes da Silva. *Electroencephalography: basic principles, clinical applications, and related fields*. Lippincott Williams & Wilkins, 2005.

[NMPSP06] Christa Neuper, Gernot R. Muller-Putz, Reinhold Scherer, and Gert Pfurtscheller. Motor imagery and EEG-based control of spelling devices and neuroprostheses. *Event-related dynamics of brain oscillations*, 159:393, 2006.

[PAB+10] Gert Pfurtscheller, Brendan Z. Allison, Clemens Brunner, Gunther Bauernfeind, Teodoro Solis-Escalante, Reinhold Scherer, Thorsten O. Zander, Gernot Mueller-Putz, Christa Neuper, and Niels Birbaumer. The hybrid BCI. *Frontiers in neuroscience*, 4, 2010.

[Pan05] Chrysostomos P. Panayiotopoulos. Optimal use of the EEG in the diagnosis and management of epilepsies. In *The Epilepsies: Seizures, Syndromes and Management*. Bladon Medical Publishing, 2005.

[PdS99] Gert Pfurtscheller and Fernando Lopez de Silva. EEG event-related desynchronization (ERD) and event-related synchronization (ERS). *Electroencephalography: Basic Principles, Clinical Applications and Related Fields*, 1999.

[PKC99] Gert Pfurtscheller, Pichler-Zalaudek Karin, and Neuper Christa. Erd and ers in voluntary movement of different limbs. In Gert Pfurtscheller and Fernando H. Lopes da Silva, editors, *Event-related desynchronization: Handbook of electroencephalography and clinical neurophysiology, revised series (Vol. 6)*. 1999.

[PKN+96] Gert Pfurtscheller, J. Kalcher, Christa Neuper, D. Flotzinger, and M. Pregenzer. On-line EEG classification during externally-paced hand movements using a neural network-based classifier. *Electroencephalography and Clinical Neurophysiology*, 99:416–425, 1996.

[PLdS99] Gert Pfurtscheller and Fernando H. Lopes de Silva. Event-related EEG/MEG synchronization and desynchronization: basic principles. *Clinical Neurophysiology*, 1999.

[PN01] Gert Pfurtscheller and Christa Neuper. Motor imagery and direct brain-computer communication. *Proceedings of the IEEE*, 89(7):1123–1134, 2001.

[PNG+00] Gert Pfurtscheller, Neuper Neuper, C. Guger, WAHW Harkam, Herbert Ramoser, Alois Schlogl, BAOB Obermaier, and MAPM Pregenzer. Current trends in graz brain-computer interface (BCI) research. *Rehabilitation Engineering, IEEE Transactions on*, 8(2):216–219, 2000.

[RBG+07] Brice Rebsamen, Etienne Burdet, Cuntai Guan, Haihong Zhang, Chee Leong Teo, Qiang Zeng, Christian Laugier, and Marcelo H Ang Jr. Controlling a wheelchair indoors using thought. *Intelligent Systems, IEEE*, 22(2):18–24, 2007.

[RMGP00] Herbert Ramoser, Johannes Muller-Gerking, and Gert Pfurtscheller. Optimal spatial filtering of single trial EEG during imagined hand movement. *Rehabilitation Engineering, IEEE Transactions on*, 8(4):441–446, 2000.

[Roi09] Linsey Roijendijk. Variability and nonstationarity in brain computer interfaces. Master thesis, Radboud University, 2009.

[SA13] Leila Azinfar Reza Fazel-Rezai Setare Amiri, Ahmed Rabbi. Brain-computer interface systems – recent progress and future prospects. In *BCI Integration: Application Interfaces*. 2013.

[SB08] Alois Schlogl and Clemens Brunner. Biosig: a free and open source software library for BCI research. *Computer*, 41(10):44–50, 2008.

[SC11] Brigitte Stemmer and John F. Connolly. The EEG/ERP technologies in linguistic research: An essay on the advantages they offer and a survey of their purveyors. *The Mental Lexicon*, 6(1):141–170, 2011.

[SGM+03] Paul Sajda, Adam Gerson, K.-R. Muller, Benjamin Blankertz, and Lucas Parra. A data analysis competition to evaluate machine learning algorithms for use in brain-computer interfaces. *Neural Systems and Rehabilitation Engineering, IEEE Transactions on*, 11(2):184–185, 2003.

[SKHM07] Alois Schlogl, Julien Kronegg, Jane E. Huggins, and Steve G. Mason. Evaluation criteria for BCI research. *Toward brain-computer interfacing*, pages 327–490, 2007.

[SLBP05] Alois Schlögl, Felix Lee, Horst Bischof, and Gert Pfurtscheller. Characterization of four-class motor imagery EEG data for the BCI-competition 2005. *Journal of Neural Engineering*, 2(4):L14, 2005.

[Smi05] S. J. M. Smith. EEG in the diagnosis, classification, and management of patients with epilepsy. *Journal of Neurology, Neurosurgery & Psychiatry*, 76(suppl 2):ii2–ii7, 2005.

[Spr10] William Spriggs. *The Sleep Technician's Pocket Guide*. Jones & Bartlett Learning, 2010.

[SRH+10] Sören Sonnenburg, Gunnar Rätsch, Sebastian Henschel, Christian Widmer, Jonas Behr, Alexander Zien, Fabio de Bona, Alexander Binder, Christian Gehl, and Vojtěch Franc. The shogun machine learning toolbox. *The Journal of Machine Learning Research*, 99:1799–1802, 2010.

[Sri07] Narayanan Srinivasan. Cognitive neuroscience of creativity: EEG based approaches. *Methods*, 42(1):109–116, 2007.

[Tep02] Michal Teplan. Fundamentals of EEG measurement. *Measurement science review*, 2(2):1–11, 2002.

[TJK06] Ryan Thibodeau, Randall S. Jorgensen, and Sangmoon Kim. Depression, anxiety, and resting frontal EEG asymmetry: A meta-analytic review. *Journal of Abnormal Psychology*, 115(4):715, 2006.

[TMA+12] Michael Tangermann, Klaus-Robert Müller, Ad Aertsen, Niels Birbaumer, Christoph Braun, Clemens Brunner, Robert Leeb, Carsten Mehring, Kai J. Miller, Gernot R. Müller-Putz, et al. Review of the BCI Competition IV. *Frontiers in Neuroscience*, 6, 2012.

[VCTB13] Giovanni Vecchiato, Patrizia Cherubino, Arianna Trettel, and Fabio Babiloni. Methodology of a typical "neuromarketing" experiment. In *Neuroelectrical Brain Imaging Tools for the Study of the Efficacy of TV Advertising Stimuli and their Application to Neuromarketing*, pages 9–31. Springer, 2013.

[VKBS09] Carmen Vidaurre, Nicole Krämer, Benjamin Blankertz, and Alois Schlögl. Time domain parameters as a feature for EEG-based brain-computer interfaces. *Neural Networks*, 22(9):1313–1319, 2009.

[VSS11] Carmen Vidaurre, Tilmann H. Sander, and Alois Schlögl. Biosig: The free and open source software library for biomedical signal processing. *Computational intelligence and neuroscience*, 2011, 2011.

[WB06] Jonathan R. Wolpaw and Niels Birbaumer. Brain-computer interfaces for communication and control. In M. E. Selzer, S. Clarke, L. G. Cohen, P. Duncan, and F. H. Gage, editors, *Textbook of neural repair and rehabilitation: Neural repair and plasticity*, pages 602–614. Cambridge University Press, 2006.

[WBH+00] Jonathan R. Wolpaw, Niels Birbaumer, Wiliam J. Heetderks, Denis J. McFarland, P. H. Peckham, G. Schalk, E. Donchin, L.A. Quatrano, C. J. Robinson, and T. M. Vaughan. Brain-computer interface technology: a review of the first international meeting. *Rehabilitation Engineering, IEEE Transactions on*, 8(2):164 –173, jun 2000.

[WJH+11] Kim D. W., Hwang H. J., Lim J. H., Lee Y. H., Jung K. Y., and Im C. H. Classification of selective attention to auditory stimuli: toward vision-free brain-computer interfacing. *Journal of Neuroscience Methods*, 197:180–185, 2011.

[WKH74] Robert Leon Williams, Ismet Karacan, and Carolyn J. Hursch. *Electroencephalography (EEG) of human sleep: clinical applications.* Wiley New York, 1974.

[WM94] Jonathan R. Wolpaw and Denis J. McFarland. Multichannel EEG-based brain-computer communication. *Electroencephalography and Clinical Neurophysiology*, 90:444–449, 1994.

[WMNF91] Jonathan R. Wolpaw, Dennis J. McFarland, Gregory W. Neat, and Catherine A. Forneris. An EEG-based brain-computer interface for cursor control. *Electroencephalography and clinical neurophysiology*, 78(3):252–259, 1991.

[WZL+04] Yijun Wang, Zhiguang Zhang, Yong Li, Xiaorong Gao, Shangkai Gao, and Fusheng Yang. BCI competition 2003-data set IV: an algorithm based on CSSD and FDA for classifying single-trial EEG. *Biomedical Engineering, IEEE Transactions on*, 51(6):1081–1086, 2004.

Poznan University of Technology
Faculty of Computing and Information Science
Institute of Computing Science

Typeset using LaTeX in Computer Modern.

BibTeX:

```
@mastersthesis{ key,
    author = "Piotr Szachewicz",
    title = "{Classification of Motor Imagery for Brain-Computer Interfaces}",
    school = "Poznan University of Technology",
    address = "Pozna{\'n}, Poland",
    year = "2013",
}
```