MŰEGYETEM 1782

Budapest University of Technology and Economics
Faculty of Electrical Engineering and Informatics
Department of Measurement and Information Systems

Detecting Hand Motions from EEG Recordings

Author
Kristóf Várszegi

Supervisor
Béla Pataki Ph.D.

26 October 2015

Contents

Abstract

For our daily activities, our limbs are indispensable. Unfortunately several people have lost a limb due to amputation or neurological disabilities. For them, a device replacing their lost limb would mean a complete life again. Electroencephalography (EEG) gives an insight into the electric activity of the human brain. If we analyze this activity, we can gain information about the patient's intents concerning limb movement. With the use of this, we could build thought-controlled, realistic, functional prosthetics, or even more complex Brain-Computer Interfaces (BCI-s).

However, the method of processing the EEG signal is not straightforward at all. Therefore this area is under active research worldwide. Although we can see more and more successful experiments in publications of various universities and research institutions, there are still numerous problems ahead of us on the way to a reliable solution. The first problem is accessing the sources of the signals. The sources of the EEG signal, the neurons of the human brain reside separated from the environment, inside the skull. Another problem is the low magnitude of the brain's electric signals. The EEG signal measured on the scalp is typically around 10 μV and 100 μV, hence its recording with proper resolution is not an easy task. Further problems are the limits of the recording's spatial resolution, the cross-talk between the electrodes picking up the electric signals, the environmental electromagnetic noises, and the distortions caused by other activities of the human body, such as eye movements.

A variety of approaches exist for processing the EEG signal in order to infer the motoric activity of the brain. Medicine uncovered correlation between some brain activities, such as deep sleep or strong focus, and the EEG signal's properties in certain frequency bands. Such frequency bands bear for example the name alpha, beta and mu. The power within one of such frequency bands, of a signal coming from a certain part of the brain, can be a good starting point for inferring local activity. Besides signal power, the change of electric potentials can also convey information. In the case of signals recorded from the motor cortex, the mentioned quantities help recognizing intents related to the movement of the human body.

In my paper I present the possibilities of recognizing hand motion intents, and the processing of a set of EEG recordings from 12 participants altogether. During the recordings, the participants had executed series' of hand motions, which had been partly labeled, meaning that the time of occurrence of the given hand motions are available. My task was to develop an algorithm which is capable of recognizing intent related to hand motion, and this way detecting the hand movements at the time of their occurrence during the recording from the raw EEG signal. This work is driven by the motive of developing BCI applications such as BCI-based prosthetic limbs. I examine the relevant possibilities of filtering in time- and spatial domain, the calculation of power per each frequency band, the extraction of features which help detecting movements, and the classification of data into event categories. I present my detection algorithm, the processing of the data which I used during the development process, and the testing of the method on a separated set of recordings. The procedure's success can mostly be judged by its classification accuracy, its rate of successful detections, and its amount of false indications. Starting from these, I carry out an evaluation, in which I elaborate on the procedure's everyday applicability as well.

Absztrakt | Abstract in Hungarian

Mindennapi teendőink elvégzéséhez elengedhetetlenek a végtagjaink. Sajnálatos módon számos embertársunk veszítette el valamely végtagját amputáció vagy idegrendszeri rendellenesség miatt. Az ő számukra egy elveszített végtagjukat pótló szerkezet újból a teljes életet jelentené. Az elektroenkefalográfia (EEG) segítségével betekintést nyerhetünk az emberi agy elektromos aktivitásába, melynek elemzésével információt nyerhetünk a páciens végtagjai mozgatására irányuló szándékáról. Ennek segítségével akár gondolatokkal vezérelt, valósághű, működőképes művégtagokat is készíthetnénk, nem is beszélve sokkal összetettebb agy-számítógép interfészekről.

Az EEG jelek feldolgozásának módja azonban korántsem triviális, ezért a területet világszerte aktívan kutatják. Bár egyre több biztató kísérleti eredményt láthatunk különféle egyetemek és kutatóközpontok publikációiban, egyelőre számos probléma áll még előttünk a megbízható megoldáshoz vezető úton. Elsőként a jelforrásokhoz való hozzáférés jelent problémát. Az EEG jel forrásai, az emberi agyat felépítő neuronok a külső környezettől elzárva, a koponyán belül helyezkednek el. Problémát jelent az agy által képzett elektromos jelek alacsony szintje is. A skalpon mért EEG jelek tipikusan 10 μV és 100 μV körül vannak, így megfelelő felbontással való rögzítésük nem egyszerű feladat. Problémát jelentenek továbbá a felvétel térbeli felbontásának korlátai, az elektromos jeleket felvevő elektródák közti áthallás, a környezeti elektromágneses zajok, valamint az emberi test egyéb aktivitása, például a szemmozgás által generált torzítások.

Többféle megközelítés létezik EEG felvételek feldolgozására ahhoz, hogy az agy motoros aktivitására következtessünk. Az orvostudomány ismer összefüggéseket az agy némely aktivitásai, például mély alvás vagy folyamatos koncentráció, és az EEG jel bizonyos frekvenciatartománybeli tulajdonságai között. Ilyen kitüntetett frekvenciasávok viselik például az alfa, a béta és a mü nevet. Az agy egy adott területéről származó EEG jel egy ilyen frekvenciasávba eső részének teljesítménye jó kiindulópont lehet a lokális aktivitásra való következtetéshez. A jelteljesítményen kívül az elektromos potenciálok változása is hordozhat információt. A motoros kéreg felől rögzített jelek esetén az előbbi mennyiségek az emberi test mozgására irányuló szándék detektálásához nyújtanak segítséget.

Dolgozatomban bemutatom a kézmozgatási szándék felismerésének napjainkban ismert lehetőségeit, valamint egy 12 résztvevős EEG felvétel-sorozat feldolgozását. A felvételek során a résztvevők kézmozdulat-sorozatokat hajtottak végre, melyek egy része időben megjelölésre került, azaz részben rendelkezésre állnak az egyes kézmozdulatok előfordulásának időpontjai. A feladatom egy olyan algoritmus kifejlesztése volt, amely pusztán a

rögzített EEG jelekből képes felismerni a kéz mozgatására irányuló szándékot, és ily módon kimutatni a kézmozdulatokat azok megtörténtének időpontjában a felvétel során. Az e mögötti hosszú távú motiváció BCI eszközök, például BCI alapú művégtagok fejlesztése. Megvizsgálom a feladat szempontjából releváns lehetőségeit az idő- és tértartománybeli szűrésnek, a frekvenciatartományonkénti jelteljesítmény-számításnak, a mozdulatok detektálását segítő mennyiségek származtatásának és az adatokat eseménymentes és kézmozdulat-kategóriákba soroló klasszifikációnak. Bemutatom az általam összeállított detektáló algoritmust, az ahhoz előzetes ismeretként felhasznált adatok feldolgozását, valamint a módszer tesztelését az előző halmaztól elválasztott felvételeken. Az eljárás sikere leginkább osztályozási pontossága, helyes detekcióinak aránya, és téves jelzéseinek mennyisége alapján ítélhető meg. Ezekből kiindulva végzem el az értékelést, mely során kitérek az eljárásnak a kísérletek körbehatárolt színterén kívüli, mindennapok során való alkalmazhatóságára is.

Introduction

Designing reliable Brain-Computer Interfaces (BCI-s) is the mission of numerous researchers and engineers, at institutes and companies all around the world. A BCI-driven, fully functional prosthetic hand could grant many people their lost limb again. Brain-controlled robotic arms would figuratively bring a whole new generation of medical or construction applications, augmenting doctors' or builders' physical capabilities by doubling or tripling the number of their limbs. Recently, there have been promising BCI experiments with notable results in grasping objects with a prosthetic hand (Agashe et al. 2015) and in the restoration of overground walking in a paraplegic patient (King et al. 2015).

Unfortunately, decoding the brain activity to obtain useful control signals is an extremely difficulty task. The brain is residing under the protection of the skull, which is a massive obstacle for neural electric signals. This is only one of the reasons why extracting information from non-invasively obtained EEG signal is a big challenge. Also, this topic is of rather multidisciplinary nature, as it requires certain knowledge of biology as well as engineering skills.

The long-term objective is the development of BCI-s for everyday use. As a sub-goal on the way, the objective of this study is to detect the time of hand movements from an already existing, digitalized recording of scalp EEG signal. The digital signal processing algorithm might constitute only a part of a complete BCI, but designing it is challenging enough just by itself. It requires the knowledge of the methods for motion detection from EEG signal. Therefore it is also among the objectives to review these methods and attempt to develop them. The exact detection task described in this paper serves as a measure of the developed algorithms.

Understanding the methodology with its complex mathematics is of paramount importance. There are highly adaptive algorithms with large number of parameters widely used, but many of these are like a black box. It is hard (or impossible) to see what kind of transformations such tools actually make. Therefore it is difficult to tell whether they can be applied in an unknown situation or not. This motivates the study of various, transparent mathematical approaches to the hand motion recognition problem. However, highly adaptive methods, such as Convolutional Neural Networks (CNN-s) are proven perform outstandingly in several situations, therefore they are also examined in this work.

The structure of this paper is the following. The first chapter gives an overview of the human brain activity, and it describes the properties of the EEG signal, such as its main frequency bands. It also describes how the EEG signal can be recorded, and utilized for BCI applications. This is in order to give a basis for the later chapters. The second chapter examines the EEG signal in the context of motoric intent recognition. This chapter describes the features of the EEG signal around the time of a movement. It also outlines a standard digital processing algorithm for motion detection and describes the possibilities for implementing each step. The third chapter presents notable past works related to motion detection. The fourth chapter presents the exact hand motion detection task, and also the processing algorithms which have been designed for its solution in this study. The algorithms are then evaluated based on their detection results. Finally, the last chapter summarizes the achievements and gives perspective for further development. The frequently used abbreviations in this paper are listed in Table 1.

Abbreviations	
ANN	Artificial Neural Network
AUROC	Area Under Receiver Operating Characteristic
BCI	Brain-Computer Interface
CNN	Convolutional Neural Network
CSP	Common Spatial Pattern
ECG	Electrocardiography/Electrocardiogram
EEG	Electroencephalography/Electroencephalogram
EMG	Electrooculography/Electrooculogram
EOG	Electromyography/Electromyogram
EP	Evoked Potential
EPSP	Excitatory Postsynaptic Potential
ERD	Event-Related Desynchronisation
ERP	Event-Related Potential
ERS	Event-Related Synchronisation
FBCSP	Filter Bank Common Spatial Pattern
FES	Functional Electrical Stimulation
FIR	Finite Impulse Response
FPR	False Positive Rate
ICA	Independent Component Analysis
IIR	Infinite Impulse Response
IPSP	Inhibitory Postsynaptic Potential
LDA	Linear Discriminant Analysis
MIBIF	Mutual Information-based Best Individual Feature
MIRSR	Mutual Information-based Rough Set Reduction
PCA	Principal Component Analysis
PMBS	Postmovement Beta Synchronization
QDA	Quadratic Discriminant Analysis
ROC	Receiver Operating Characteristic
SVM	Support Vector Machine
TPR	True Positive Rate
VEP	Visual Evoked Potential

Table 1: *Frequently used abbreviations.*

Chapter 1

EEG signal acquisition and analysis

1.1 The electric activity of the brain

The electrical signals generated by the human brain represent the brain function and the status of the whole body (Sanei and Chambers 2007). The first brain activity recording can be dated to 1875, when Richard Caton, a pioneer electrophysiologist from England used a galvanometer and two electrodes over the scalp of a human subject to pick up electrical signals. The term EEG combines the concepts of electro- (referring to the registration of electric signals), encephalo- (referring to emitting the signals from the head), and graphy (meaning drawing or writing), and is used to denote the signals representing the electric activity of the brain. The central nervous system consists of nerve cells (neurons), and between them, glia cells. A neuron consists of the cell body, axons, and dendrites. The structure of a neuron can be seen in Figure 1.1.

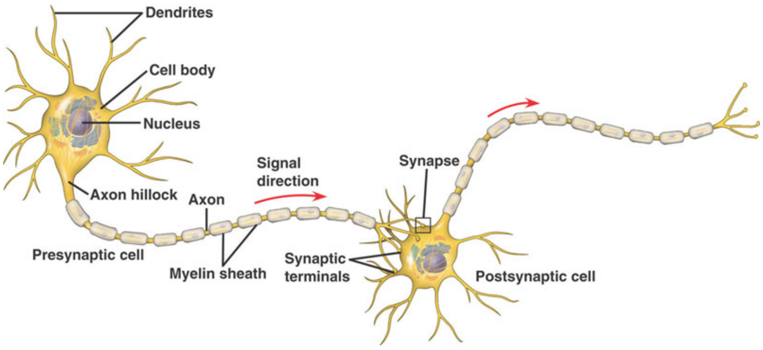

Figure 1.1: *Neurons (Cheung 2010)*

Neurons respond to stimuli and its long, cylindric axons transmit electric impulses. In humans the length of an axon can be a fraction of a millimeter to more than a meter. Dendrites receive impulses from other nerves or relay the signals to others. They are connected to either the axons or the dendrites of other cells. In the human brain each nerve is connected to approximately 10,000 other nerves, mostly through dendritic connections. The activities in the central nervous system are mainly related to the synaptic currents transferred between the junctions (called synapses) of axons and dendrites, or between dendrites of cells. Under the membrane of the cell body, electric potential of 60-70 mV with negative polarity may be recorded, which changes with variations in synaptic activities. If an action potential travels along the fibre, which ends in an excitatory synapse, an Excitatory Postsynaptic Potential (EPSP) occurs in the following neuron. If more action potentials travel along the same fibre over a short distance, the EPSP-s will add up. If such way a certain threshold of membrane potential is reached, action potential is produced on the postsynaptic neuron. In case the fibre ends in an inhibitory synapse, hyperpolarization will occur resulting in Inhibitory Postsynaptic Potential (IPSP). After the generation of an IPSP, cations and anions flow from and into the nerve cell, causing a change in the electric potential along the membrane of the nerve cell. These ion flows can generate secondary ion currents along the cell membranes in the intra- and extracellular space. The extracellular currents generate electric field potentials, which, with less than 100 Hz frequency and stationary signal mean, are the EEG-s. An example of the EEG signal is shown in Figure 1.2.

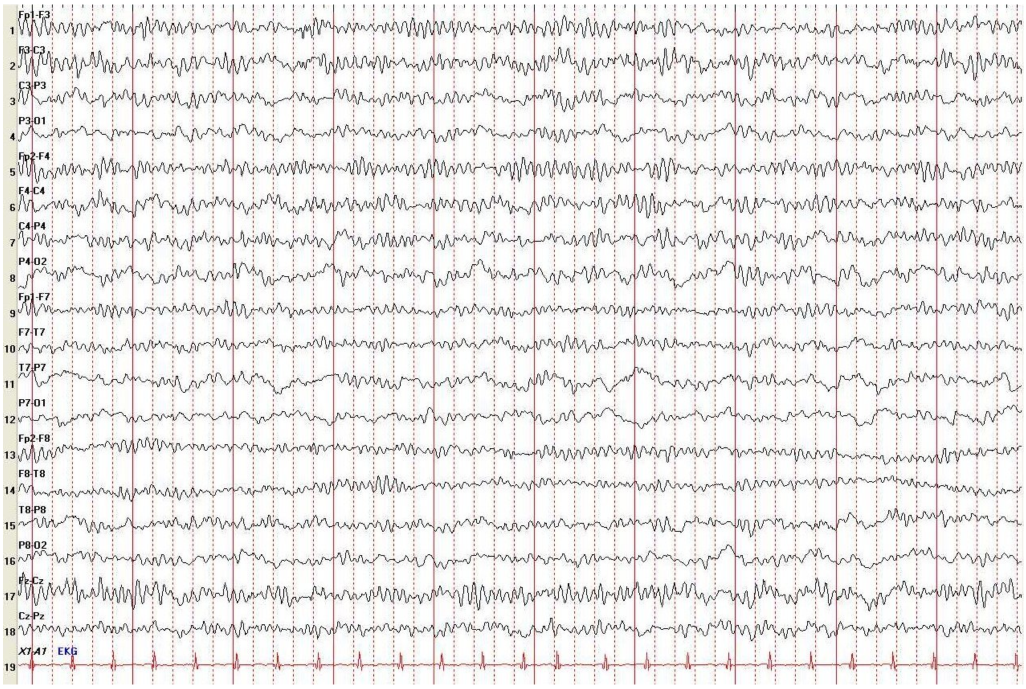

Figure 1.2: *An example of the EEG signal, along with an ECG. (Modern 2014)*

The human head comprises several layers such as the skull and the scalp. These layers significantly attenuate the EEG signal. The attenuation caused by the skull is about one hundred times more than the soft tissue. Because of the major attenuation, only large populations of active neurons can generate an electric potential high enough for recording with electrodes over the scalp. There is also a disturbing amount of noise being added to the signals, mostly generated either within the brain or over the scalp. Muscles are near the surface of the skin, and operate with strong electric signals. Especially the large muscles of the neck and the jaw may cause serious disturbance in the EEG signal, which comes from deep within the brain.

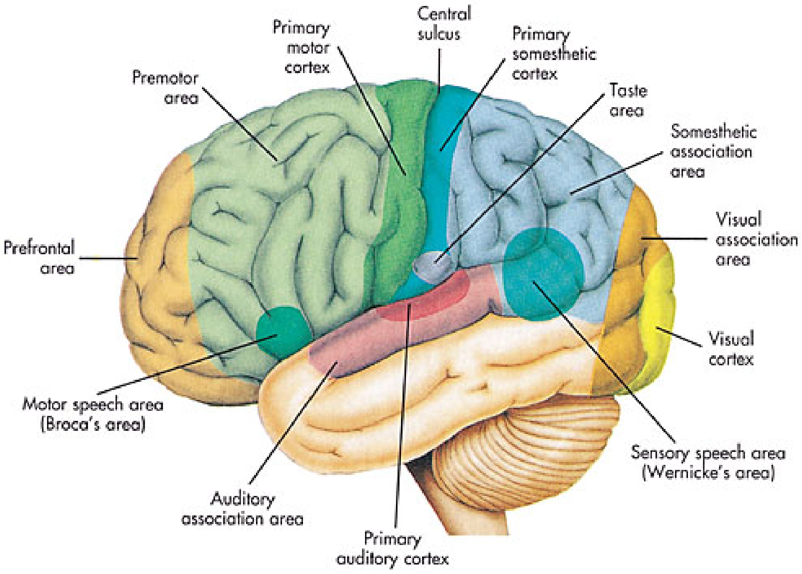
Central sulcus
Primary motor cortex
Primary somesthetic cortex
Premotor area
Taste area
Somesthetic association area
Prefrontal area
Visual association area
Visual cortex
Motor speech area (Broca's area)
Sensory speech area (Wernicke's area)
Auditory association area
Primary auditory cortex

Figure 1.3: *The structure of the human brain. (Brain Key 2012)*

Anatomically the brain can be divided into three parts: the cerebrum, the cerebellum, and the brain stem. The cerebrum consists of both the left and the right lobes of the brain with highly convoluted surface layers called the cerebral cortex. The different regions of the cerebrum are responsible for e.g. movement initiation, conscious awareness of sensation, complex analysis, and expression of emotions. The cerebellum is responsible for coordinating the voluntary movement of muscles and maintaining balance. The brain stem controls involuntary functions such as respiration and heart function regulation. Figure 1.3 depicts the basic structure of the brain.

In the view of hand motions, the most important part of the brain is the motor cortex. It is involved in the planning, control, and execution of voluntary movements. It can be divided into three areas. The first one is the primary motor cortex, which is the main contributor to neural impulses which control the execution of movements. The second one is the premotor cortex, which is responsible for some aspects of motor control, possibly including the preparation for movement, the sensory guidance of movement, the spatial guidance of reaching, or the direct control of some movements. The third area is the supplementary motor area, located on the mid-line surface of the hemisphere just in front of the primary motor cortex. Its possible functions include the internally generated planning of movement, the planning of sequences of movement, and the coordination of the two sides of the body.

1.2 The main frequency bands of the EEG signal

The amplitudes and frequencies of EEG signal, during a given state such as alertness or deep sleep, vary over people. The characteristics of the EEG waves also change with age. Rhythms generated by the same physiological machinery at different ages often fall into different bands with different names.

The categorization of different brain rhythms begun with a German psychiatrist, Hans Berger naming the approximately 10 hertz oscillation the "alpha" rhythm. The reason of Berger naming it "alpha" is because this was the first rhythm he observed. The subsequently discovered frequency bands were also labeled with Greek letters. Around the time of these discoveries, the EEG recording technology confined the frequency of the observable brain rhythms (Buzsáki 2006). The upper border of frequencies was limited by the widely used mechanical pen recorders, and the lower border by the electrode polarization and movement artifacts.

There are five major brain rhythms distinguished by their different frequency ranges. These are the delta (δ, 0.5-4 Hz), theta (θ, 4-8 Hz), alpha (α, 8-12 Hz), beta (β, 12-30 Hz), and gamma (γ, above 30 Hz, up to 80 Hz) rhythms. The beta range is sometimes subdivided into three sub-bands, namely the low beta (or "sigma", 12-15 Hz), beta 2 (15-20 Hz) and high beta (or beta 3, 20-30 Hz). Above the gamma frequencies, a "fast" (80-200 Hz) and "ultra-fast" (200-600 Hz) band are also defined. The borders of the frequency ranges slightly vary over books and articles on the topic. Table 1.1 summarizes the main frequency bands.

Brain rhythm	Frequency
delta	0.5-4 Hz
theta	4-8 Hz
alpha	8-12 Hz
beta	12-30 Hz
gamma	30-80 Hz

Table 1.1: *The main brain rhythms and their frequency bands.*

The delta rhythm is associated mainly with deep sleep and fatigue. It has also been found during some continuous-attention tasks (Kirmizi-Alsan et al. 2006). Delta-band oscillations in the primary visual cortex might also rise as a response for task-relevant events (Lakatos et al. 2008). The theta waves are related to lapses in attention, drowsiness, memory consolidation, inhibition of elicited responses, creative inspiration and deep meditation. A theta wave is often accompanied by other frequencies and seems to be related to the level of arousal. The alpha band is connected to relaxation, readiness and inactive cognitive processing. The beta range is associated with relaxation and calm focus (low beta), intense focus and cognitive processing (beta 2), and anxiety and distractibility (beta 3). The gamma rhythm can be observed during short-term memory matching of recognized objects, sounds, or tactile sensations. There is another brain rhythm of possible interest called mu, having oscillation in the 8-12 Hz band, sharing it with alpha. It is closely related to somatosensory cortex activity, and to the resting state of motor neurons. Figure 1.4 shows examples of the waveforms observable in the main frequency bands.

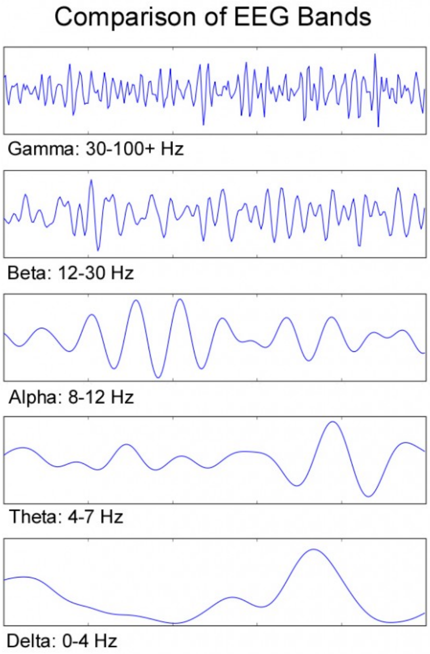

Figure 1.4: *Comparison of EEG Bands. (Kent 2010)*

1.3 The technique of recording EEG

An EEG recording system mainly consists of electrodes, amplifiers with filters, analog-digital converters and a recording computer (Teplan n.d.). The electrodes pick up the signal from the scalp, and the amplifiers magnify the microvolt magnitude signals into a range where they can be digitalized with proper accuracy. The A/D converters then transform the signals into digital form, and a computer (desktop, embedded etc.) stores and processes the obtained data according to the purpose. Scalp recordings made with such systems allow the measurement of potential changes over time between active electrodes and a reference electrode. Active electrodes are placed at the parts of interest (e.g. the primary motor cortex), and the reference electrode is usually placed on the top of the head, the ear or the mastoid. There can be more reference electrodes, in which case their voltages are averaged. A ground electrode is also needed for being able to subtract the common voltages present at the active and reference points. A minimal mono-channel configuration for an EEG measurement consists of one active electrode, one reference and one ground electrode. A multi-channel measurement can use up to 128 or 256 active electrodes.

The electrodes and their proper function are critical for acquiring appropriately high quality data for interpretation. There are many types of electrodes: disposable electrodes, saline-based electrodes, reusable disc electrodes (gold, silver, stainless steel or tin), needle electrodes (inserted invasively under the scalp), headbands and electrode caps. For multichannel montages, electrode caps are preferred, with a number of electrodes installed. Commonly used scalp electrodes consist of Ag-AgCl disks, with 1 to 3 mm diameter. In order to minimize signal distortions, impedances of electrode-scalp contacts should be below 5 kΩ, and balanced within 1 kΩ of each other. Some EEG devices include impedance monitors to help ensuring this. The required preparation of the skin varies with the type of electrode. Generally the cleaning of the skin surface from oil and brushing from dried parts is needed. Since the electrodes may scrape the skin the right hygiene must be ensured when recording EEG.

In 1958, the International Federation in Electroencephalography and Clinical Neurophysiology had standardized the EEG electrode placement (Jasper 1958). This system, called 10-20 electrode placement, divides the head into proportional distances from prominent skull landmarks to provide adequate coverage of all regions of the brain. This standardized placement is depicted in Figure 1.5.

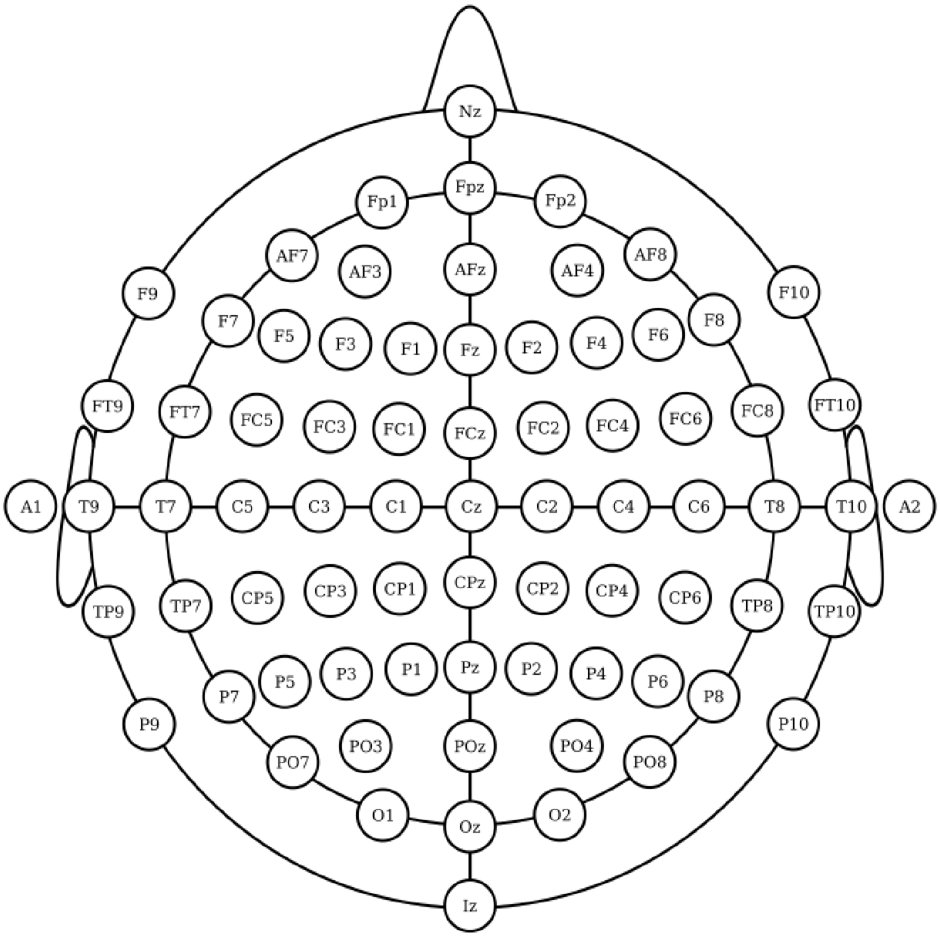

Figure 1.5: *The international 10-20 electrode position system. (Hart 2008)*

The "10" and "20" in the standard's name mean that the actual distances between adjacent electrodes are either 10% or 20% of the total front-back or right-left distance of the skull. Each site has a letter to identify the lobe and a number to identify the hemisphere location. The letters F, T, C, P and O stand for frontal, temporal, central, parietal, and occipital lobes, respectively. The letter "C" is used only for identification purposes, as no central lobe exists. A "z" (zero) refers to an electrode placed on the mid-line of the scalp. Even numbers (2,4,6,8) refer to electrode positions on the right hemisphere, and odd numbers (1,3,5,7) refer to those on the left hemisphere. The left and right side is considered from the point of view of the subject by convention.

Amplifiers adequate to measure the EEG signal must satisfy specific requirements. The EEG amplifier's input signal consists of five components: the desired biopotential, the undesired biopotentials, the power line interference signal and its harmonics, interference signals generated by the tissue-electrode interface, and other noise. The amplifier must provide amplification selective to the physiological signal, reject superimposed noise and interference signals, and guarantee protection from damages through voltage and current surges for both patients and electronic equipment. Electrically shielded rooms minimize the impact of the urban electric background, in particular the 50/60 Hz power line noise. A shielded room is not necessary for usual medical measurements, but it is recommended for research purposes when the maximum possible data quality is desired. The amplification unit should have analog filters built in. A high-pass filter (with cutoff frequency around 0.1-0.7 Hz) is required to reduce low frequency components coming from bioelectric flowing potentials (breathing, etc.), that remain in the signal after subtracting the voltages of the ground electrode. To limit the frequency band of the signal, a low-pass filter is also needed with cutoff at the highest frequency of interest.

The recording device (e.g. a desktop computer) repeatedly samples the channels of the analog signal at a fixed time interval, and each sample is converted into a digital representation by an analog to digital converter. The sampling rate should be high enough so the Nyquist frequency is above the frequency of the signal of interest. The A/D converter's resolution (0.5 μV or more precise is recommended) is determined by the smallest amplitude that can be sampled, which depends on the number of bits of the device. The digitalized signal data can be then processed with digital filtering. This is usually needed for extracting relevant information. Such filters should be designed in a way to minimize their influence on the useful signal properties. The recording computer must also have enough storage space, as for example 1 hour of eight channels 14-bit signal sampled with 500 Hz occupies 200 MB of the memory. Today's desktop computers usually have enough storage, but this may be a challenge in embedded devices.

The recorded EEG signal usually contains artifacts which should be removed either manually by experts or automatically. These undesired components are usually higher in amplitude and have different shape than a clean signal. Artifacts can be either patient-related or technical. The patient related artifacts are minor body movements, the electric activity of muscles, ECG (pulse, pace-maker), eye movements, and sweating. The technical artifacts are related to the 50/60 Hz power line interference, impedance fluctuation, cable movements, broken wire contacts, too wet or too dry skin surfaces pieces or low battery. Additional electrodes for monitoring eye movement, ECG, and muscle activity can be used for better discrimination of the physiological artifacts.

1.4 What the EEG signal is good for

The EEG signal can be used for many purposes. They can be used to locate areas of damage following head injury, stroke, tumor etc., to investigate epilepsy and locate seizure origin, to investigate sleep disorders, and to diagnose coma or brain death. They can also be used to monitor sleep, alertness or cognitive engagement (alpha rhythm), control anaesthesia depth ("servo anaesthesia"), and to monitor brain development. Many brain disorders are diagnosed by visual inspection of the EEG signal, done by clinical experts in the field who are familiar with the manifestation of brain rhythms.

Brain-Computer Interfaces (BCI-s) are subjects of great interest because of multiple reasons. A Brain-Computer Interface is a system that recognizes the user's command only from the EEG signal and reacts accordingly. For example in a simple task, an arrow is displayed on a computer screen, which is to be moved only by the subject imagining the motion of the left or right hand. During the imagining process, certain characteristics of the brainwaves are extracted and can be used for recognizing the user's commands. Looking at complex tasks, EEG-driven artificial limbs could grant people the function of their lost or paralyzed body part (due to stroke or spinal cord injury, for example).

The EEG signal can be also used for communication. For the latter purpose, the so-called P300 spellers are rather popular and are under intense development. Figure 1.6 shows a basic P300 speller in operation.

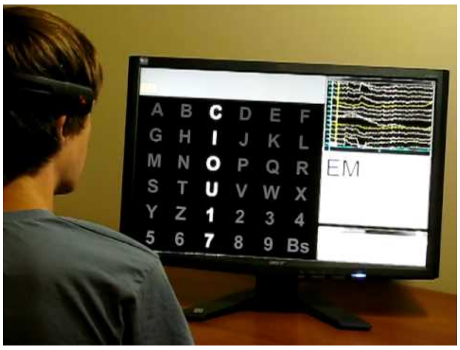

Figure 1.6: *A P300 speller (snapshot). (Wadsworth Center 2010)*

Such devices exploit a special change in potential, called P300, which is related to a person's reaction to a stimulus. In the EEG signal, this manifests as a positive deflection in voltage with a latency of roughly 250 to 500 ms between the stimulus and the response (Polich 2007). In a P300 speller, a matrix of letters is shown to the user, who would like to spell out words of choice, and is watching for the next letter in line. The letters are flashing up according to an order specific to the device type. When the corresponding letter is flashing up, a P300 potential is evoked inside the brain. The signal processing algorithm analyzes the EEG data and selects the letter with the highest P300 component, which is then written on the screen. Normally between 2-20 flashes per letter are required to achieve a high accuracy. The effectiveness depends mainly on the electrode position and the individual amplitude of the P300 response of the subject.

Recently, University of Houston reported that they have created an algorithm that allowed a man to grasp a bottle and other objects with a prosthetic hand. The algorithm was measuring the EEG signal and hand joint angular velocities (Agashe et al. 2015). In the demonstration of their technique, a 56-year-old man, whose right hand had been amputated, grasped objects including a water bottle and a credit card, as depicted in Figure 1.7. The subject managed to grasp the selected objects 80 percent of the time using a bionic hand fitted to the amputee's stump.

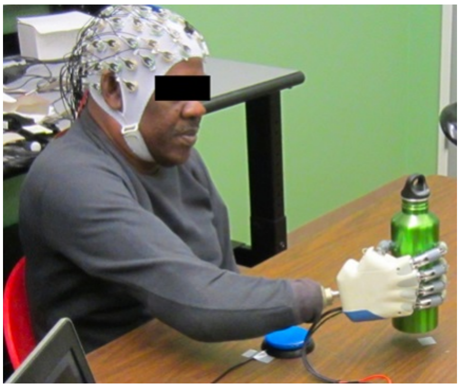

Figure 1.7: *New UH research has demonstrated that an amputee can grasp with a bionic hand, powered only by his thoughts. (University of Houston 2015)*

The research team first recorded brain activity and hand movement in able-bodied volunteers as they picked up five objects, each chosen to illustrate a different type of grasp: a soda can, a compact disc, a credit card, a small coin and a screwdriver. The recorded data were used to create decoders of neural activity into motor signals, which successfully reconstructed the grasping movements. This experiment showed that it is feasible to extract detailed information on intended grasping movements from the EEG signal along with hand joint angular velocity. It also provided evidence that the acquired signals predicted the movement, rather than reflecting it.

Another recent study (King et al. 2015) demonstrates an individual, with paraplegia due to spinal cord injury, purposefully operating a noninvasive BCI-FES (Brain-Computer Interface, Functional Electrical Stimulation) system for overground walking in real time.

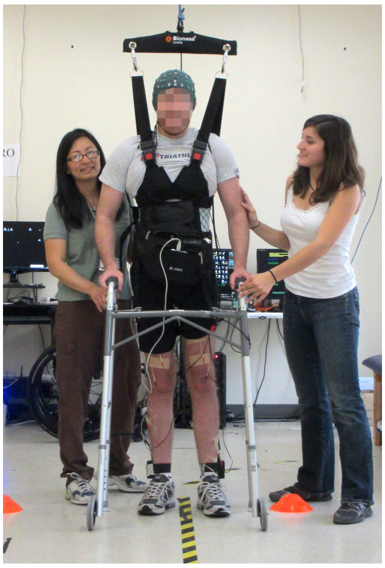

Figure 1.8: *A man whose legs had been paralyzed for five years walks along a 12-foot course using UCI-developed technology that lets the brain bypass the spinal cord to send messages to the legs. (University of California, Irvine 2015)*

The participant initially operated the system while being completely suspended to an overhead rail, and subsequently translated this skill to an overground walking condition. The setup is depicted in Figure 1.8. The subject achieved a high level of control and maintained this level of performance during a 19-week period. These results provide a proof-of-concept for direct brain control of a lower extremity prosthesis to restore basic overground walking after paraplegia due to spinal cord injury. The participant achieved and maintained a high level of performance during the experiment. In comparison to the suspended walking conditions, there was a notable increase in the false alarm rate during overground walking. This is explained by an increase in EEG noise produced by movements, such as postural stabilization or weight shifting. The false alarm rate decreased toward the end of the study, presumably due to the practice in operating the BCI. The participant was also able to carry a light conversation during these experiments without interfering with the function of the system. This robustness in real-time control, together with a high-level of performance sustained across months, indicates that BCI-FES mediated restoration of basic walking function after spinal cord injury is feasible.

Motion restoration using the EEG signal looks promising from these results. However, thinking about artificial limbs does not need to stop at replacing lost body parts with realistically functional, biomechanic devices. Besides our natural limbs, additional body parts could be of great use in numerous tasks and environments, where several objects are to be handled with physical motion (e.g. medicine or construction). Moreover, direct control of computers by a BCI would largely augment the possibilities of user operation. Aircraft pilots, whose hands are occupied on the handles, could give various commands to the airplane machinery by thought. A door would open for a new generation of communication methods, office computer applications, and generally machinery control systems.

Chapter 2

Processing the EEG signal for motion recognition

2.1 The manifestation of motion in the EEG signal

There is a number of features of the EEG signal used for operating a BCI. In this chapter, the Evoked Potentials (EP), the Event-Related Desynchronization (ERD) and Synchronization (ERS), and the Bereichtshaftspotential (readiness potential) are presented.

2.1.1 Evoked potentials

Evoked potentials or event-related potentials (ERP-s) are voltage fluctuations resulting from neural activity (Teplan n.d.). They are initiated by an external or internal stimulus. ERP-s provide a suitable handle for studying cognitive processes. Mental operations, such as those involved in perception, selective attention, language processing, and memory, proceed over time ranges in the order of tens of milliseconds. PET and MRI is capable of localizing regions of activation during a given mental task, ERP-s can help defining the time course of these activations. Amplitudes of ERP components are often much smaller than spontaneous EEG components, which makes them nearly impossible to be recognized from the raw EEG signal. ERP-s are extracted from a set of single recordings by digital averaging of epochs (recording periods) of EEG, time-locked to repeated occurrences of sensory, cognitive, or motor events. This way the background voltage fluctuations (which are assumed to be random relatively to the time of the stimuli) are theoretically averaged out, leaving only the event-related brain potentials. These electrical signals reflect only the activity which is consistently associated with the stimulus. Thus the ERP reflects the temporal pattern of neuronal activity evoked by a stimulus.

2.1.2 Event-related synchronization and desynchronization

Another category of interesting features of the EEG signal contains the Event-Related Synchronization (ERS) and Desynchronization (ERD) (Sanei and Chambers 2007). The main difference between these and the ERP is that the ERP is a stimulus-locked or, more generally, a phase-locked reaction, while the ERD/ERS is a non-phase-locked response. A frequent example of these is finger movement, in which case prominent negative voltage prior to movement onset, and beta oscillations immediately after the movement offset are respectively phase-locked (evoked) and non-phase-locked processes. The ERD is measured in terms of the peak power of the averaged power map of the period of the event, compared to the average power during the reference time, all this at a given frequency. In ERS, the amplitude enhancement is based on the synchronized behavior of a large number of neurons, making it easy to measure field potentials using scalp electrodes.

Voluntary movement results in a circumscribed desynchronization in the upper alpha and lower beta bands over the brain's sensorimotor regions. The ERD starts over the contralateral (opposite side of that of the movement) part of the central sulcus (see Figure 2.1), and during the movement, becomes symmetrical bilaterally (including both sides) with the time of execution of the movement.

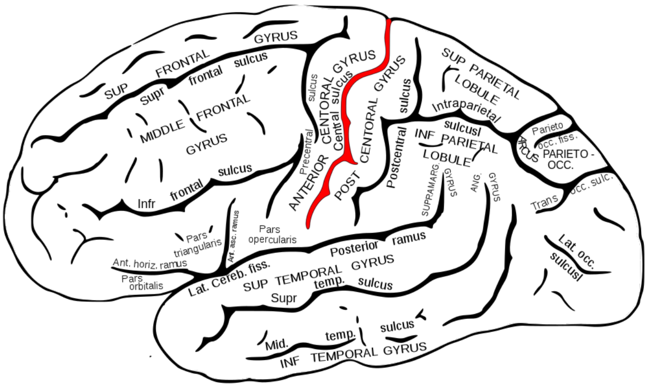

Figure 2.1: *The central sulcus (highlighted in red). (Gray 2009)*

In the case of finger motion, fast and slow movements have different corresponding neural processes. Fast movement is preprogrammed and the afferents are delivered to the muscles as bursts. On the other hand, slow movement depends on the reafferent input from kinaesthetic receptors (the ones in muscles and other body parts which sense the position and movement of the body) evoked by the movement itself. The finger movement of the dominant hand is accompanied by a pronounced ERD in the ipsilateral side (the side of the movement), whereas movement of the non-dominant finger is preceded by a less lateralized ERD. Similar phenomenon can be found in the case of hand movement.

The transient beta activity after the movement (also called Postmovement Beta Synchronization, PMBS for short), is another interesting phenomenon which starts during the movement and continues for about 600 ms. It is found after finger or foot movement over both hemispheres without any significant bilateral coherence. Its frequency band varies from subject to subject. For finger motion the range is around 16-21 Hz whereas for foot movement it is around 19-26 Hz. The amplitude of the PMBS is similar for both fast and slow finger motion, despite that these movements involve different neural pathways. Moreover, this activity is significantly larger with hand movement than in the case of finger movement. Wrist movement also comes with larger beta oscillations compared to finger movement. These lead to the assumption that during the movements with larger beta oscillations, a bigger population of motor cortex neurons change from an increased neural discharge to a state of cortical disfacilitation or cortical idling. Therefore the movement of more fingers results in a larger beta wave. Beta activity is also important in the generation of a grasp signal, since it has less overlap with other frequency components.

ERS within the gamma frequency band (35-45 Hz) can also carry information. Such activity is present after visual stimuli or just before movement. Gamma, together with other activities in the alpha and lower beta bands, can be observed around the same time after movement, but gamma ERS manifests itself just before the motion, whereas beta ERS occurs immediately after the event.

Besides alpha, beta, and gamma ERS and ERD activities, a long delta oscillation starts immediately after finger movement and lasts for a few seconds. This can be a prominent feature in distinguishing between movement and non-movement states.

2.1.3 Bereichtshaftspotential

The Bereichtshaftspotential (BP for short, or readiness potential) is a measure of activity in the motor cortex and supplementary motor area of the brain leading up to voluntary muscle movement (Kornhuber and Deecke 1965).

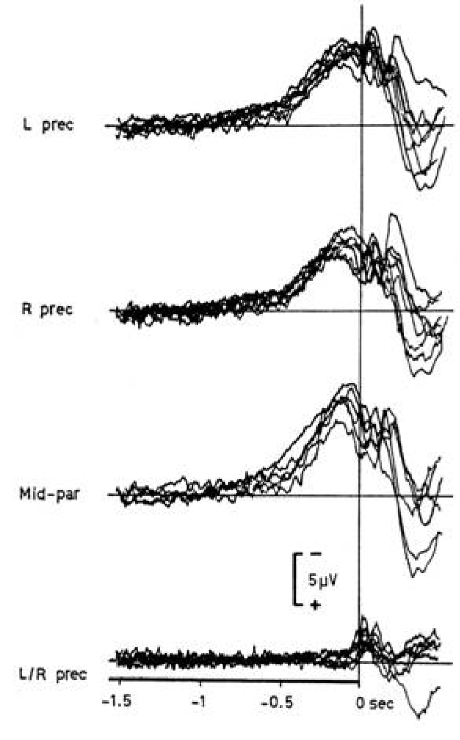

Figure 2.2: *Bereitschaftspotential (also pre-motor potential or readiness potential). (Deecke 2005)*

The BP is a manifestation of cortical contribution to the pre-motor planning of intentional movement. The BP is ten to hundred times smaller than the alpha rhythm of the EEG. It becomes apparent only by averaging the periods of a recording around the onset of a movement. The BP has two components: the early one lasting from about -1.2 to -0.5

sec and the late component from -0.5 to shortly before 0 sec relative to movement onset (see Figure 2.2). The early component (BP1) is a stage with moderate steepness, whereas the late component (BP2) is much steeper.

Artifacts due to head-, eye-, lid-, or mouth-movements and respiration must be eliminated before averaging, because these artifacts may have a magnitude which makes it difficult to render them negligible even after hundreds of sweeps. In the case of eye movements, eye muscle potentials must be distinguished from cerebral potentials.

In addition to finger and eye motion, the BP has been recorded accompanying willful movements of the wrist, arm, shoulder, hip, knee, foot and toes, and also prior to speaking, writing, and swallowing.

2.1.4 Expectations considering hand motion

In the context of hand motion detection, all of these features might be of help. ERP may be present around the time of movements because of the subjects' attention. ERD/ERS may also be seen at the execution of a movement. The BP is not likely to be recognizable due to its tiny amplitude relative to other rhythms. Still, if it is identifiable, it is may prove as a great handle for the movement detection.

2.2 The outline of a standard digital processing algorithm

The task of a BCI is to generate control signals by exploiting the behavior of the EEG signal before, during, and after the imaginary movement, or after certain brain stimulation. This chapter describes the outline of extracting such control signals from the raw EEG signal data. As the signal acquisition has already been presented, this outline deals only with the procedure after the A/D conversion. The outline of a general digital signal processing procedure for motoric intent recognition is shown is Figure 2.3.

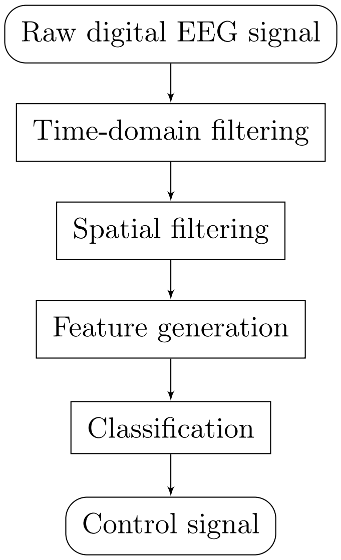

Figure 2.3: *The outline of a general digital signal processing procedure for motoric intent recognition.*

The purpose of this section is to give an overview of common methods for each step, but not to present their compete mathematical formulations, solutions and proofs (for these, see the references). Furthermore, examples of the usage of each technique are not described here to ensure the conciseness of this chapter. Instead, past motion recognition related BCI attempts involving such methods are presented in the next chapter.

2.3 Time-domain filtering

Usually the first transformation applied to a raw EEG signal is a time-domain filter. The purpose of this pass on one hand is to eliminate (to the maximal possible extent) DC and high frequency noise and power line (50/60 Hz) harmonic interference. DC and high-frequency noise can be reduced using a band pass filter, while the power line harmonic interference is usually cut out with a notch filter. For these purposes, FIR or IIR digital filters are used. The advantage of FIR filters over IIR ones is their inherent stability and linear phase characteristics. However, their operation is more costly computationally than that of IIR filters.

Besides the elimination of noise, the time-domain filter may also serve the purpose of separating the components of the EEG signal which fall into different frequency bands (alpha, beta, gamma etc.), to help the extraction of relevant features later on. Such multi-band filters are called filter banks.

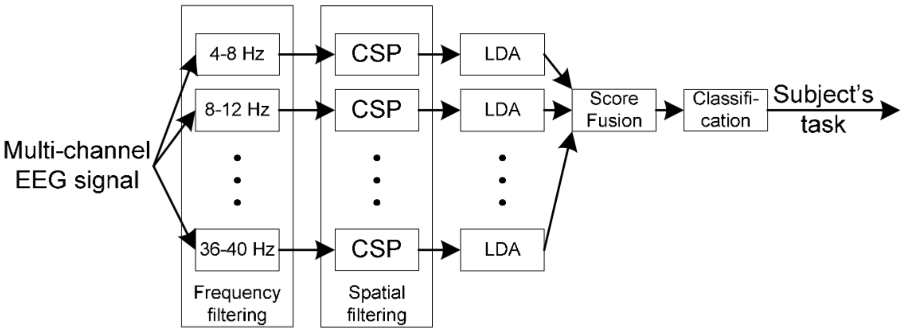

Figure 2.4: *An example of a filter bank applied in a BCI. (Ang et al. 2008)*

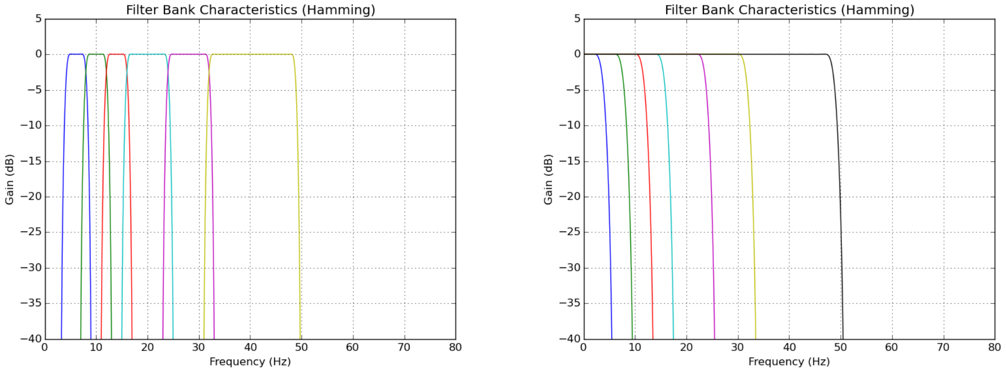

Figure 2.5: *Examples of filter bank characteristics (left: band-pass, right: low-pass).*

Figure 2.4 shows an example of a process involving a time-domain filter bank, and Figure 2.5 the magnitude plots of two examples. All of the filters within a bank are applied to the signal, and each result is examined separately afterwards.

When analyzing the EEG signal, the power carried in the different frequency components may be of interest. Power calculations can also be carried out in this stage using a spectral density estimation method. A basic way to do this is using the periodogram technique, and taking the squared magnitude of the discrete Fourier transform of the signal:

$$S(f) = \frac{\Delta t}{N} \left| \sum_{n=0}^{N-1} x_n e^{-i2\pi n f} \right|^2, \qquad -\frac{1}{2\Delta t} < f \leq \frac{1}{2\Delta t} \tag{2.1}$$

where N is the number of samples taken from the signal, uniformly spaced by Δt, having value x_n, and $1/(2\Delta t)$ is the Nyquist frequency. The periodogram is a simple tool, but it is an inconsistent estimator because it does not converge to the true spectral density as $N \to \infty$. It also exhibits high spectral leakage.

Bartlett's spectral density estimation method (also known as the method of averaged periodograms) provides a way to reduce the variance of the periodogram in exchange for a reduction of resolution (Bartlett 1948), compared to standard periodograms. A final estimate of the spectrum at a given frequency is obtained by averaging the estimates from the periodograms at that same frequency, derived from non-overlapping portions of the original series. First, the original N point data segment is split up into K (non-overlapping) data segments, each of length M. Then for each segment, the periodogram is calculated by computing the discrete Fourier transform (without dividing by M), then the squared magnitude of the result is taken and divided by M. Afterwards, the result of the periodograms are averaged for the K data segments. This way the variance is reduced due to the averaging.

Welch's method for spectral density estimation is also based on the concept of using periodogram spectrum estimates, mostly on Bartlett's method. In Welch's estimation method, the portions of the series contributing to each periodogram are allowed to overlap (Welch 1967). The signal is split up into overlapping segments. The original data segment is split up into L data segments of length M, overlapping by D points. The overlapping segments are then windowed. After the data is split up into overlapping segments, a window is applied to the individual L data segments in time domain. Most window functions influence the data at the center more than the data at the edges, causing loss of information. To minimize this loss, the individual data sets are overlapped in time. Welch's method is often desired to reduce the noise due to the imperfection and finiteness of the data.

The signal processing procedure sometimes involves the downsampling of the input data because of computational capacity constraints. Before such step, a low-pass filter (designed according to the factor of downsampling) should be applied to the signal to prevent aliasing.

2.4 Spatial filtering

The EEG signal is usually acquired from multiple electrodes spread over the scalp. The signal picked up by the electrodes will inevitably contain some degree of redundancy. A potential wave occurring at one point of the scalp will be sensed at other points as well. The purpose of spatial filters is to remove this crosstalk between the electrodes, and leave only the local signal component in the channels. Common techniques for spatial filtering are the Principal Component Analysis (PCA), the Independent Component Analysis (ICA), the Common Spatial Pattern, the XDAWN, and the Surface Laplacian methods.

2.4.1 Principal Component Analysis

Principal Component Analysis (PCA) uses an orthogonal transformation to convert a set of observations of possibly correlated variables into a set of values of linearly uncorrelated variables called principal components (Pearson 1901). The number of principal components is less than or equal to the number of original variables. The first principal component has the largest possible variance, and each succeeding component in turn has the highest variance possible under the constraint that it is orthogonal to the preceding components. The resulting vectors form orthogonal basis, as they are the eigenvectors of the covariance matrix.

Matrix $\mathbf{X}$ stores the acquired signal data in a way that each column corresponds to a different channel (meaning each row represents samples from a different point of time), and that the values are translated so that each column has zero mean. Let i be the index of the time point, and k the index of the principal component. The transformation is defined by a set of p-dimensional vectors of weights $\mathbf{w}_{(k)} = (w_1, \ldots, w_p)_{(k)}$ that project each row vector $\mathbf{x}_{(i)}$ of the data matrix $\mathbf{X}$ to a vector of principal component scores $\mathbf{t}_{(i)} = (t_1, \ldots, t_p)_{(i)}$, given by $t_{k(i)} = \mathbf{x}_{(i)} \cdot \mathbf{w}_{(k)}$ in a way that the individual variables of $\mathbf{t}$ successively inherit the maximum possible variance from $\mathbf{x}$ over the dataset. Each loading vector $\mathbf{w}$ is constrained to be a unit vector. This way the first weight vector $\mathbf{w}(1)$ must satisfy

$$\mathbf{w}_{(1)} = \text{arg max} \left\{ \frac{\mathbf{w}^T \mathbf{X}^T \mathbf{X} \mathbf{w}}{\mathbf{w}^T \mathbf{w}} \right\} \tag{2.2}$$

where the quantity to be maximized is a Rayleigh quotient. The quotient's maximum possible value is the largest eigenvalue of the matrix $\mathbf{X}^T\mathbf{X}$, which occurs when $\mathbf{w}$ is the corresponding eigenvector. The second principal component is given by the second largest eigenvalue and it's eigenvector, and third is by the third one, and so on. The complete principal component decomposition of $\mathbf{X}$ is given as $\mathbf{T} = \mathbf{X}\mathbf{W}$ where $\mathbf{W}$ is a p-by-p matrix whose columns are the eigenvectors of $\mathbf{X}^T\mathbf{X}$.

PCA is the simplest of the eigenvector-based multivariate analyses. After performing the analysis, using only the first few principal components provides the data with reduced dimensionality, in exchange for loss of information. A drawback of PCA is that it is sensitive to the relative scaling of the original variables.

2.4.2 Independent Component Analysis

Independent Component Analysis (ICA) assumes that behind the data from direct observations, there are independent signal sources which are being mixed as they get to the spectator (Hyvärinen and Oja 2000). A typical illustration of this is the cocktail-party problem, in multiple people are speaking simultaneously in a room equipped with microphones in different locations. The objective is to obtain the original speech signals of each person from the recorded signals, which are the mixtures of the intact ones. For this there must be at least as many microphones as the number of people.

The model of ICA is as follows. Let $\mathbf{x}$ denote a vector of directly observable variables $(x_1, \ldots, x_n)$, and $\mathbf{s}$ the vector of statistically independent sources of the signal $(s_1, \ldots, s_n)$. This way the model of ICA is

$$\mathbf{x} = \mathbf{A}\mathbf{s} \tag{2.3}$$

where A is the matrix mixing the intact signals into the observed data. The independent components are latent variables as they cannot be directly observed, and the mixing matrix is also unknown.

It is assumed that the components s_i are statistically independent. By estimating the matrix $\mathbf{A}$ and calculating its inverse $\mathbf{W}$ the independent components are obtained as $\mathbf{s} = \mathbf{W}\mathbf{x}$. The key to estimating the ICA model is the non-gaussianity of the original signal sources. ICA is very closely related to the method called blind source separation (BSS) or blind signal separation. In BSS, "blind" means that we know little about the mixing matrix, and make few assumptions about the source signals. Involving the noise in the model would make it more realistic, but the estimation of the noise-free model is difficult enough already, and it is sufficient for many applications.

2.4.3 Common Spatial Pattern

The Common Spatial Pattern method (CSP) takes two sets of spatial patterns representing two classes, into which other sets of spatial patterns are later to be classified (Müller-Gerking, Pfurtscheller and Flyvbjerg 1999). An element of a set (a spatial pattern), will be the amplitudes of a multi-channel EEG signal at a given point of time. A set of patterns consists of spatial patterns that make up the EEG recording. From a recording of many movement trials, the sets are used to calibrate the method by extracting the spatial features. For the distinction of the classes, covariances are used.

As output, the calibration gives an ordered list of the characteristic spatial patterns. These patterns define directions in the pattern space that are optimally suited for distinguishing between the two classes. After the appropriate transformation, a time-series of patterns that belong to the first class will scatter maximally along the first direction and minimally along the last (and vice versa in the case of the other class). The second and the second-to-last directions are the second best directions for the distinction, and so on.

The mathematical formulation begins with $\mathbf{V}_a^i$ denoting the matrix containing the data of the period of the i-th execution of the movement belonging to class a (Müller-Gerking, Pfurtscheller and Flyvbjerg 1999). Each row of this matrix corresponds to one channel of the EEG. The mean of the signal is taken to be zero, as it is usually removed by a previous time-domain band-pass filter. The characteristic information is in the second moments, or the covariance matrix, which is to be normalized with the total variance:

$$\mathbf{R}_a^i = \frac{\mathbf{V}_a^i \mathbf{V}_a^{iT}}{\mathrm{trace}(\mathbf{V}_a^i \mathbf{V}_a^{iT})} \tag{2.4}$$

for class a and $\mathbf{R}_b^i$ for class b. These matrices are then averaged for all the trials within each class. The method continues by the simultaneous diagonalization of the averaged matrices $\mathbf{R}_a$ and $\mathbf{R}_b$. The matrix of the resulting eigenvectors, $\mathbf{P}^{\mathrm{T}}$ will be the matrix of the projective transformation (the eigenvectors are the rows of $\mathbf{P}^{\mathrm{T}}$). This is applied to the EEG signal as

$$\mathbf{Z} = \mathbf{P}^{\mathrm{T}}\mathbf{V} \tag{2.5}$$

where $\mathbf{Z}$ is the resulting spatially filtered signal. The EEG signal should be projected to the most discriminatory characteristic patterns (vectors in $\mathbf{P}^{\mathrm{T}}$). Then the variance of these values within a time window should be calculated. These variances are the basic features on which the classification is based. Picking only a few characteristic patterns are usually sufficient for the task. These patterns are practically spatial filters which select the most relevant spatial aspects for the classification. This also reduces the dimensionality of the problem, while making use of the information carried by all channels.

2.4.4 xDAWN algorithm

The purpose of the xDAWN algorithm (Rivet et al. 2009) is to generally maximize the signal-to-noise ratio (SNR). As the operators it provides calculate a linear combination of the EEG channel values at a given point of time, xDAWN can be considered a spatial filtering method. The name of the algorithm came from the mathematical description of its final model, which comprises matrices denoted by the letters 'X', 'D', 'A', 'W', and 'N'. The basic model assumed by the algorithm is written as

$$\mathbf{X} = \mathbf{DA} + \mathbf{N} \tag{2.6}$$

where $\mathbf{X}$ is the matrix containing the complete recorded EEG signal (with each column corresponding to one channel). $\mathbf{A}$ is the matrix of the reference ERP signals of one trial (single execution of the subject's task). $\mathbf{D}$ is a Toeplitz matrix which actually schedules the reference signals into the complete recording at the times of the task execution. Matrix $\mathbf{N}$ represents the on-going activity of the subject's brain, the artifacts etc., generally the noise. The least square estimation of $\mathbf{A}$ is simply given by

$$\hat{\mathbf{A}} = (\mathbf{D}^{\mathrm{T}}\mathbf{D})^{-1}\mathbf{D}^{\mathrm{T}}\mathbf{X}. \tag{2.7}$$

The extended model involves spatial filters, which are to enhance the intact response in the signal. This is written as $\mathbf{XU} = \mathbf{DAU} + \mathbf{NU}$, where the columns of matrix $\mathbf{U}$ are the spatial filters. The spatial filters should be defined in the following way to ensure maximal signal to signal plus noise ratio:

$$\hat{\mathbf{U}} = \arg\max_{\mathbf{U}} \frac{\text{trace}(\mathbf{U}^{\mathrm{T}}\hat{\mathbf{A}}^{\mathrm{T}}\mathbf{D}^{\mathrm{T}}\mathbf{D}\hat{\mathbf{A}}\mathbf{U})}{\text{trace}(\mathbf{U}^{\mathrm{T}}\mathbf{X}^{\mathrm{T}}\mathbf{X}\mathbf{U})} \tag{2.8}$$

which leads to a Rayleigh quotient. The complete description of the solution can be found in (Rivet et al. 2009).

To obtain the optimal spatial filters, first the QR factorization of the matrices $\mathbf{X}$ and $\mathbf{D}$ need to be calculated. From the singular value decomposition $\mathbf{Q}_{\mathbf{D}}^{\mathrm{T}}\mathbf{Q}_{\mathbf{X}} = \mathbf{\Phi}\mathbf{\Lambda}\mathbf{\Psi}^{\mathrm{T}}$, the desired number of singular vectors ψ_i should be selected which are associated with the largest singular values. The spatial filter vectors are then obtained as $\hat{\mathbf{u}}_i = \mathbf{R}_{\mathbf{X}}{}^{-1}\psi_i$. With these spatial filter vectors stored in a matrix $\mathbf{U}$, the estimation of the original signals for the complete recording is given by

$$\hat{\mathbf{S}} = \hat{\mathbf{U}}^{\mathrm{T}}\mathbf{X} \tag{2.9}$$

2.4.5 Surface Laplacian

The surface Laplacian technique is based on Ohm's law, which establishes a local relationship between the surface Laplacian of the scalp potential function and the underlying flow of electric current caused by the activity of the brain activity (Carvalhaes and de Barros 2015). This local relation is assumed to improve the spatial resolution of the EEG data.

The most common assumption within this technique is that the signal's sources of interest are inside the skull, and that there are no sources in the scalp itself, where then we have the Laplace-equation

$$\Delta V = 0 \tag{2.10}$$

with V denoting the electric potential. For small local areas, the scalp can be approximated with a plane (x, y). Explicating the Laplace-equation in Cartesian coordinates, taking the z component to the other side, and using the relationship between the electric field and the potential we get

$$\frac{\partial^2 V}{\partial x^2} + \frac{\partial^2 V}{\partial y^2} = \frac{\partial E}{\partial z} \tag{2.11}$$

which can be rewritten using the relation between the electric field and the current density $\mathbf{E} = \rho\mathbf{j}$, ρ being the resistivity. The left side of the equation is defined as the surface Laplacian of V, denoted by $\Delta_S(V)$. The resulting equation is

$$\Delta_S(V) = \rho\frac{\partial \mathbf{j}}{\partial z} \tag{2.12}$$

which represents the relation between the surface Laplacian of V and the change of the current density along direction perpendicular to the surface. A non-zero value of the surface Laplacian at a point on the surface indicates to radially diverging current flow under the scalp, which is associated with the presence of a source of currents inside the skull. This is a direct relationship between the surface Laplacian and the neural currents under the scalp.

The starting point for computing the surface Laplacian is the recorded EEG data. There are a couple of techniques for the estimation. The Finite Difference Method operates on a discrete-spaced grid fit to the electrode locations. However, mesh-free methods allow for a much more flexible configuration of electrodes and are not restricted to the planar scalp model. In such techniques, the Laplacian differentiation is performed analytically on a continuous function built from the data, either by interpolation or a parametrization procedure. Two examples of such methods are smoothing thin-plate spline Laplacians and smoothing spherical splines.

2.5 Feature generation

After applying time-domain and spatial filters of choice, we would like to transform the resulting signal into quantities, by which the instances belonging to different classes can be best discriminated. An example of an instance is the EEG signal's values in the window of the last 1 second during the operation of a BCI.

A trivial shortcut at this step is to not calculate any features at all, and simply feeding the filtered EEG signal to a classification algorithm. Interestingly, this may be appropriate when using highly adaptive classification methods such as Convolutional Neural Networks, in which case the classifier actually finds the best suitable transformation for the input data. Features can also be the amplitudes of the oscillations within given frequency bands.

A common feature generation approach is calculating the power of the signal and averaging it over a time period, for every channel. Especially when applied after a time-domain filter bank, this means quantifying the neural activity of the brain in a given frequency range, or more ranges.

Another possible way is to look for certain trends in the signal, and representing these trends by fitting a line on the values within a time window. After this linear regression, the steepness of such line may carry the information of interest.

A more complex feature is the covariance matrix of the multi-channel values during an epoch. After calculating the sample covariance matrix of the current time window, different kinds of auxiliary transformations can be applied such as normalization with the matrix trace. The advantage of using a covariance matrix is that it contains both per-channel and inter-channel information.

Before being fed to a classifier, the signal can also be normalized by mean and standard deviation values obtained from a training dataset. This may make the convergence of an adaptive classifier faster and also the detection accuracy more precise.

2.6 Classification

Classification is the problem of identifying the category to which of a new observation belongs, on the basis of a training set of data instances whose membership category is known. Some classifier algorithms are designed to discriminate between only two classes, and some to between more. A binary classifier can be extended to handle more categories with methods such as pair-wise, divide and conquer, or one versus rest classification. There are algorithms which involve multiple classifiers estimating the same variable, in which case the some kind of combination of their output values (e.g. weighted average, majority voting) is formed as the final classification output. Such ensembles may to perform better than one classifier on its own.

In performing a prediction of a given movement from an EEG signal, the last step is to tell the right class for each feature vector. The above described processing procedure contains a classifier, which is not capable of correct operation without its internal parameters being set, or in other words the adaptive classifier being trained. The basic scheme for a training procedure is shown in figure 2.6. A set of signal records is required for this phase as training data, along with the explicitly known class labels for each element of the dataset.

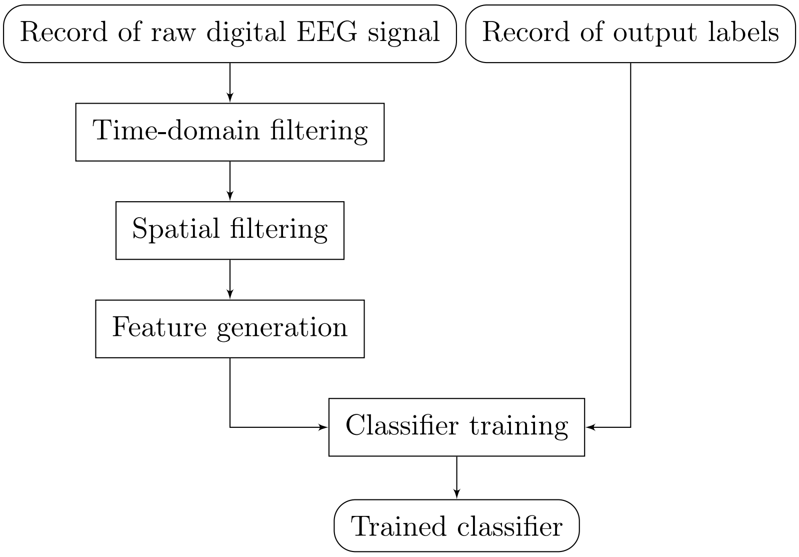

Figure 2.6: *The outline of a general procedure for classifier training.*

It is important to note that this training data should be separated into a learning set and a validation set. In each iteration, the classifier's parameter setting algorithm would examine only the learning set, and testing its performance on the validation set. This is called cross-validation, and is required as we want the classifier to perform on a set of data which it hasn't ever seen yet, and not on the data on which it has been trained.

It is also important to avoid the overfitting of the classifier. Overfitting means that the classifier has gotten suited for the training data too much, even if it has been trained with cross-validation. This usually happens when an adaptive algorithm adjusts its parameters over too many iterations, until it becomes refined too much. In such case, it will mislabel incoming data instances which are just slightly different than the data it has been trained on. In order to prevent this, the training procedure of the classifier should not consist of too many iterations of parameter refinement.

The available amount of training data is often not enough for suitably fitting the classifier. A clever trick for increasing the number of training instances is called data augmentation. This means generating new data elements from the existing ones with some kind of transformation. A great example of data augmentation is in the case of image recognition, when new data instances are created by mirroring the existing ones.

2.6.1 Simple threshold

The simplest way to sort the incoming elements into classes is to define a threshold value based on experience. Above this value, the instance is considered to be a member of one class, and below the value, to the other class. The optimal value of the threshold in this method is defined based on experience, which is usually gained by examining an available training dataset.

2.6.2 Naive Bayes classifier

Bayesian classifiers assign the most likely class to a given data instance described by its feature vector. In the case of the naive approach, the features are assumed to be independent:

$$P(\mathbf{x}|C) = \prod_{i=1}^{n} P(x_i|C) \tag{2.13}$$

where $\mathbf{x} = (x_1, \ldots, x_n)$ denotes the feature vector and C the class label. $P(\mathbf{x}|C)$ is the probability of the feature vector taking the value of $\mathbf{x}$ with the condition that it belongs to class C. This assumption may be unrealistic, but the method has proven successful in numerous applications.

The Bayesian classifier calculates for each class the probability of a given feature vector belonging there. According to Bayes' rule, this probability is given by

$$P(C|\mathbf{x}) = \frac{P(\mathbf{x}|C)P(C)}{P(\mathbf{x})} \tag{2.14}$$

where $P(\mathbf{x})$ can actually be omitted, as it has the same value for all the classes and this way does not influence the comparison. The objective is then to find $h_C(\mathbf{x}) = \arg\max_C P(\mathbf{x}|C)P(C)$ which is the maximum *a posteriori* probability hypothesis for the given feature vector $\mathbf{x}$. The difficulty is in estimating the class-conditional probability distribution $P(\mathbf{x}|C)$. For this, there are numerous kinds of approximations in practice. One of these is the already mentioned inter-feature independency, with which we obtain the naive Bayes classifier having the discrimination function

$$f_C^{NB}(\mathbf{x}) = \prod_{i=1}^{n} P(x_i|C)P(C). \tag{2.15}$$

The vector $\mathbf{x}$ is sorted to that class C which yields the highest discriminant value.

2.6.3 Linear Discriminant Analysis

Linear Discriminant Analysis (LDA) is a generalization of Fisher's linear discriminant method. It assumes that the feature vectors are linearly separable (Teknomo 2015). Given a feature vector, LDA also aims to choose the class for which the $P(C|\mathbf{x})$ probability is maximal, similarly to the Bayes classifier. However, LDA assumes that the data elements within each class have multivariate normal distribution, and the covariance matrix of all classes are identical. The discriminant function (based on which the classification decision will be made) of LDA is

$$f_C^{LDA}(\mathbf{x}) = \mu_C \Sigma^{-1} \mathbf{x}^{\mathrm{T}} - \frac{1}{2} \mu_C \Sigma^{-1} \mu_C^{\mathrm{T}} + \ln(P(C)) \tag{2.16}$$

where μ_C is the multivariate mean vector of class C, Σ^{-1} is the common shared covariance matrix, $\mathbf{x}$ is the feature vector, and $P(C)$ is the probability of the occurrence of class C. For the feature vector instance $\mathbf{x}$, the class with the highest discriminant value is chosen.

2.6.4 Quadratic Discriminant Analysis

Quadratic Discriminant Analysis (QDA) is a generalized version of LDA in the sense that instead of a linear one, the separating surface between the classes is quadratic, providing a more complex border. In the case of a quadratic classifier, the class label for a feature vector $\mathbf{x}$ is decided based on $\mathbf{x}^{\mathrm{T}} \mathbf{A} \mathbf{x} + \mathbf{b}^{\mathrm{T}} \mathbf{x} + c$.

QDA also assumes that the data elements in each class are normally distributed, but it does not assume that the covariance matrices of the classes are identical. The class of a given feature vector is decided with likelihood ratio test, from which it can be shown that the resulting separating surface between the classes is quadratic.

2.6.5 Logistic Regression

The task is still the estimation of the conditional probability $P(C|\mathbf{x})$. The basic idea is to model this quantity with a chosen best fitting function of $\mathbf{x}$. In the case of logistic regression, the probability value (which is bounded between 0 and 1) is undergoing logistic transformation, which way the resulting quantity can be an unbounded, linear function of $\mathbf{x}$ (Shalizi 2012). The univariate model of logistic regression is

$$\log \frac{P(x)}{1 - P(x)} = \beta_0 + x\beta_1 \tag{2.17}$$

which after solving for $P(x)$ gives

$$P(x) = \frac{e^{\beta_0 + x\beta_1}}{1 + e^{\beta_0 + x\beta_1}} = \frac{1}{1 + e^{-(\beta_0 + x\beta_1)}}. \tag{2.18}$$

where x is the single feature, e is Euler's number, and $\beta_{0,1}$ are the model parameters. In the multivariate case, the model is stated as

$$P(\mathbf{x}) = \frac{e^{b + \mathbf{x} \cdot \mathbf{w}}}{1 + e^{b + \mathbf{x} \cdot \mathbf{w}}} = \frac{1}{1 + e^{-(b + \mathbf{x} \cdot \mathbf{w})}} \tag{2.19}$$

where $\mathbf{x} = (x_1, \ldots, x_n)$ is the vector of features, b is constant parameter (also called offset or bias), and $\mathbf{w} = (w_1, \ldots, w_n)$ is the vector of the linear coefficients (also called weights) of the individual features (elements of $\mathbf{x}$). The probability given by the above expression is the basis of the classification decision with the threshold of 0.5 for minimal mis-classification rate.

Logistic regression does not only define the boundary between classes, but also makes the class probabilities dependent on the distance from that boundary. The parameters of the model can be fitted using likelihood or various numerical optimization techniques such as Newton's method.

2.6.6 Support Vector Machine

The definition of a Support Vector Machine (SVM) is as follows. Let $\mathbf{x}$ denote the feature vector and y the class label, with possible numeric values -1,+1 (Burges 1998). Let's assume there is a hyperplane which separates the positive from the negative instances. The points $\mathbf{x}$ which lie on the hyperplane satisfy $w \cdot x + b = 0$, where $\mathbf{w}$ is the normal vector of the hyperplane. In a linearly separable case, the support vector algorithm looks for the separating hyperplane which ensures the largest distance from the hyperplane to the closest positive and the closest negative instances (these $\mathbf{x}$-es are called the support vectors). The problem is formulated as

$$\mathbf{w} \cdot \mathbf{x}_k + b \geq +1 \quad \text{for } y_k = +1 \tag{2.20}$$

$$\mathbf{w} \cdot \mathbf{x}_k + b \leq -1 \quad \text{for } y_k = -1 \tag{2.21}$$

where $\mathbf{x}_k$ and y_k are the feature vector and th corresponding class label for the k-th data instance, respectively. Furthermore, $\mathbf{w}$ is the vector of weights, and b is the bias. The above expressions combined in one inequality are written as

$$y_k(\mathbf{w} \cdot \mathbf{x}_k + b) \geq 0 \quad \forall t \tag{2.22}$$

The optimization is performed with the Lagrangian formulation of the problem, on one hand because Lagrange multipliers make it easier to handle. On the other hand because this way the training data will only appear in the form of dot products between vectors, which allow for the generalization of the procedure to nonlinear case. The Lagrangian expression reads as

$$L_P = \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{k=1}^{K} \alpha_k y_k (\mathbf{w} \cdot \mathbf{x}_k + b) + \sum_{k=1}^{K} \alpha_k \tag{2.23}$$

where α_k are the Lagrange-multipliers, K is the total number of data instances, and $\|.\|$ denotes the Euclidean norm. The objective is to minimize L_P with respect to $\mathbf{w}, b$, and maximize it with with respect to the α_k-s (subject to the constraints $\alpha_k \geq 0$ $(\forall k)$).

The SVM can be extended to nonlinear case with applying a transformation to the feature vectors, mapping them to another Euclidean space. For this purpose, a "kernel" function is used, which can be for example a Gaussian radial basis function, a polynomial (homogeneous or inhomogeneous), or a hyperbolic tangent.

2.6.7 Artificial Neural Network

The Artificial Neural Network (ANN, or neural network for short within the field) is a type of learning model which is inspired by biological neural networks (the brain). The ANN is a system of interconnected "neurons". It is used to approximate functions that may have a large number of inputs and are generally unknown. Their connections have numeric weights that can be adjusted with learning algorithms. ANN-s are widely used in computer vision and speech recognition. A layout of an ANN is shown in Figure 2.7.

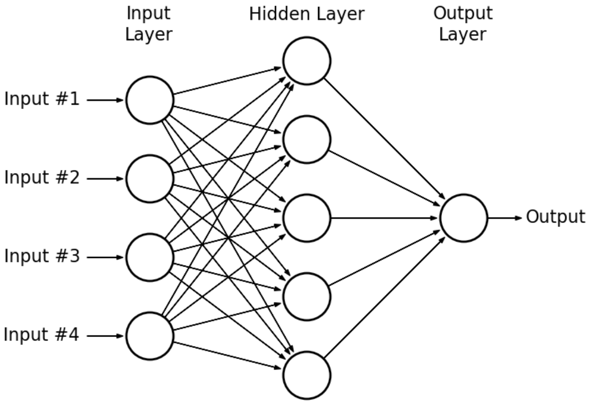

Figure 2.7: *An example of a simple Artificial Neural Network (Minnaar 2015).*

In the sense of data flow, an ANN consists of three kind of layers. The first is the input layer, which serves as a multi-channel port for the incoming feature vectors. The second type is the hidden layer, in which each artificial neuron takes the linear combination of the input vector $\mathbf{x}$ with the weight values of the connections between the hidden neuron and the input neuron, represented by $\mathbf{w}$. Besides the linear combination, usually a bias value is added as well. Afterwards, an univariate transformation of choice is applied to this weighted sum. The complete transformations made by one layer of neurons on the incoming data vector can be written as

$$\mathbf{a} = f(\mathbf{W}\mathbf{x} + \mathbf{b}) \tag{2.24}$$

where $f(.)$ is the output transfer function, $\mathbf{a}$ denotes the output of each neuron in the layer (in column vector form), $\mathbf{W}$ the matrix of connection weights between the previous and the current layer (each row corresponds to one neuron's incoming connections), $\mathbf{x}$ is the (column) vector of the input values fed to the layer, and $\mathbf{b}$ is the (column) vector of the biases of each neuron. There are several different types of hidden layers, applying various transformations at this stage. The resulting value is fed into the next layer, which can be another hidden layer or an output layer. The third type (in the sense of data flow) is the output layer, in which the neurons execute the same scheme of transformation as the hidden layer's neurons, but their results are fed into the further steps of the processing algorithm.

The parameters of an ANN are the weights of the connections and the biases of the artificial neurons. These parameters are tuned iteratively with the gradient descent and the backpropagation algorithms (Nielsen 2015). In the learning process, each neuron's contribution to the output error is calculated, and the parameters are adjusted accordingly.

2.6.8 Convolutional Neural Network

The Convolutional Neural Network (CNN) is the improved version of the ANN. In the CNN, the neurons are tiled in such a way that they respond to overlapping, partial regions of the input data. Compared to the basic ANN, in a CNN the neurons inside one layer share the same weights for the connections over the neurons' own receptive field. For example, the weight of the connection at the top-right corner of one neuron's receptive field has the same value as the top-right weight of the neighboring neuron. These convolutional type of layers are accompanied with various other kinds inside the network, serving auxiliary functions. CNN-s are widely used in the fields of image and video recognition (see Figure 2.8 for an example).

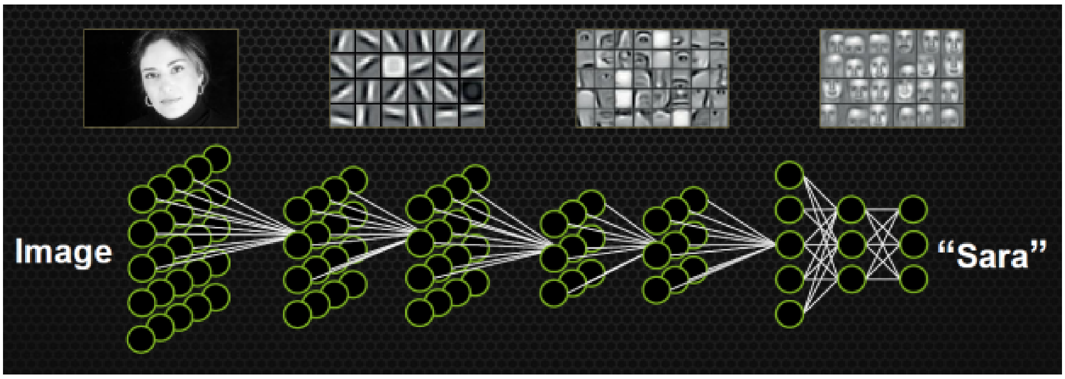

Figure 2.8: *An example of a Convolutional Neural Network designed for image recognition. (Brown 2014)*

Several types of layers are applied in CNN-s. The basic layer is the above mentioned convolutional type, in which the neurons process a partial region of the input data. Another is the default type of the simple ANN, often called dense layer, in which all neurons are connected to every neuron in the previous layer (this was the case of the simple ANN). Pooling layers compute the maximal or average value of a feature over a subset of the input vector's components, making the network less sensitive to small offsets in the feature values. Dropout layers are often included with the purpose to prevent overfitting the network. In this kind of layer, in every batch of iteration individual neurons are either omitted from the net with probability (1-P) or kept with probability P, and only the reduced network is trained on the data in the given cycle of learning. After every such cycle, the removed neurons are reinserted into the network with their original weights.

2.6.9 About the generality of classifiers

Generally there is no method which can be considered the best for every possible classification task. SVM-s and various kinds of ANN-s may perform very well in solving high dimensional problems, such as image recognition (LeCun, Cortes and Burges 1998). However, these methods come with high computational cost. In an application where there is little computational resource available, which can be the case for many embedded systems, methods such as Naive Bayes, LDA, QDA or Logistic Regression might be more suitable. These simpler methods might bring the results with a precision high enough for a given task. Figure 2.9 shows a visual comparison of the performance of some classifiers, and the next chapter presents a couple of applications in BCI-s.

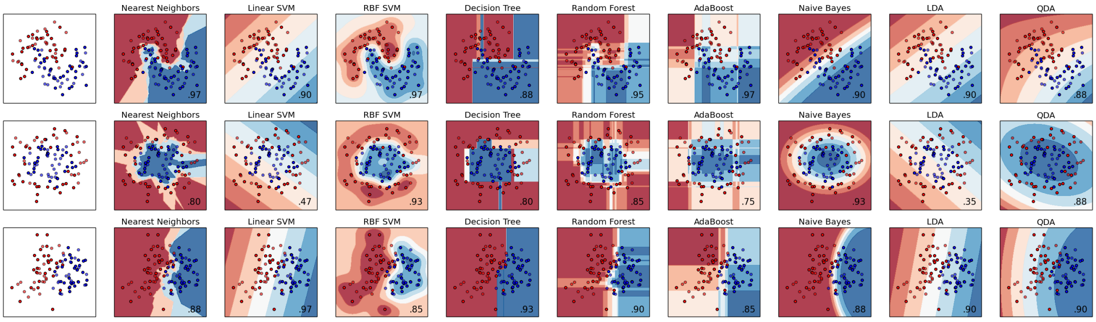

Figure 2.9: *Comparison of some classifiers on three kind of datasets. (scikit-learn developers 2014)*

Chapter 3

Past studies related to motion detection

This chapter presents a couple of highly interesting previous attempts of using BCI for motoric intent recognition (not necessarily limited to hand motion detection), made by various scientists. The first chapter has already presented two remarkable experiments with complete BCI systems, one with grasping trials using a prosthetic hand (Agashe et al. 2015), and the other with restoring overground walking function (King et al. 2015). Presenting these works was more about giving an overview on the architecture of two complete BCI systems. The current chapter focuses on presenting the signal processing algorithms in relevant works.

3.1 A case of restoring hand grasp function

Pfurtscheller et al. (2005) achieved noninvasive restoration of hand grasp function in a tetraplegic patient, with functional electrical stimulation (FES). The patient in the experiment was able to induce bursts of beta oscillations by imagining foot movement. The EEG signal was recorded with a monochannel configuration.

The EEG was recorded bipolarly from two gold electrodes, with the active electrode being at the Cz position (in the international standard 10-20 system). The EEG signal was amplified between 0.5 and 30 Hz with an EEG-amplifier and sampled with 128 Hz. It was then processed online by bandpass filtering (15-19 Hz), squaring, averaging over 128 samples, and applying logarithm. The resultant feature value was passed to a threshold detector. If passing, a trigger pulse was generated by the BCI, followed by a disabled output period of 3 seconds. The threshold was empirically selected by comparing the band power values obtained from resting and motoric imagery periods.

The output of the BCI was then used as a trigger signal for switching between states in a predefined sequence, corresponding to different grasp phases. These phases were the following: no stimulation (phase 0), opening hand (phase 1), grasping (phase 2), releasing (phase 3). After phase 3, the system returned to phase 0. This means that the same motor imagery was used for triggering the transition between each consecutive states.

Moving the hand was achieved by functional electrical stimulation. For example, the opening of the hand by the extension of all fingers joints and the thumb could be achieved by stimulation of the finger extensors and the thumb extensor muscle with electrodes on the radial side of the proximal forearm.

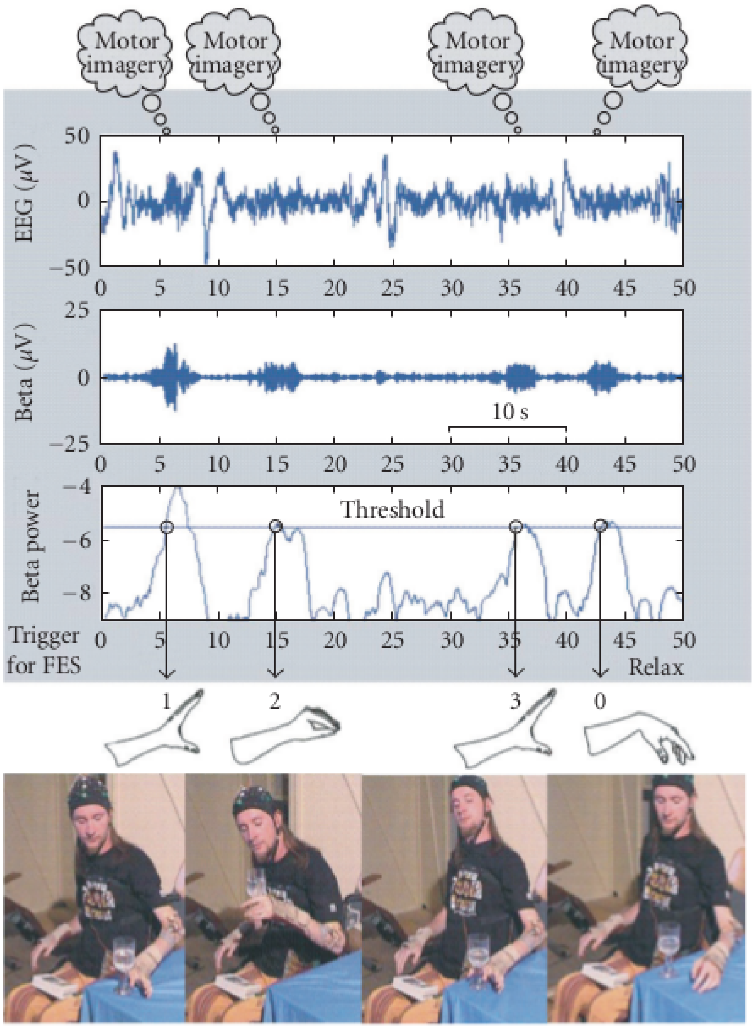

Figure 3.1: *Example of the raw and the bandpass filtered EEG signal (in the top and the middle graphs, respectively), the band power time course (bottom graph). Photographs of the grasping (bottom). (Pfurtscheller et al. 2005)*

The subject using the BCI was a 29-year old man suffering from a traumatic spinal cord injury. Several types of imaginations were tried to increase the classification accuracy. First, imaginations of left versus right hand movements were carried out. The accuracy was higher with single-foot motor imageries versus relaxing or hand movement imagination. With this BCI, the patient was able to grasp a glass with the paralyzed hand completely on his own without additional help or further technical aid. An arising question is the subject-wise generality of the device, as it has only been tested with one person. The result is still remarkable and gives valuable hints for BCI design.

3.2 A case of two-dimensional movement control

The generation of a two-dimensional movement control signal was reportedly achieved in (Wolpaw and McFarland 2004). The experiment was carried out with four subjects with various disabilities. The subjects sat in front of a computer screen, on which a target point appeared randomly in one of eight possible positions. The subjects were to learn the control of a cursor via BCI by a reinforcement method and to drive it to the target point. If the cursor reached the target within 10 seconds after its appearance, it flashed as a reward. In case of failure, the cursor and the target simply disappeared. In either case, the screen was blank for 1 second before the beginning of the next trial.

The EEG signal was recorded from 64 standard electrode locations over the scalp. All channels were referenced to the right ear, amplified by a gain of 20,000 within the band 0.1-60 Hz, and digitalized at 160 Hz. Each dimension of the cursor movement was controlled by a linear equation in which the independent variable was a weighted combination of the amplitudes of a 8-12 Hz mu rhythm or 18-26 Hz beta band over the right and left sensorimotor cortices. The weights in this feature variable were updated after each trial by an adaptive algorithm to optimize the translation of the subject's EEG activity into cursor control. During the session, the last 400 milliseconds of spatially filtered EEG signal (Laplacian filter from C4 and C3 electrodes over the sensorimotor cortex) underwent autoregressive frequency analysis to determine the amplitudes in specific mu and beta oscillations. For the adaptive control algorithm, the least- mean-square method was used in adjusting the weights to minimize the difference between the actual target location and the predicted target location (by the subjects' control signal) for past trials.

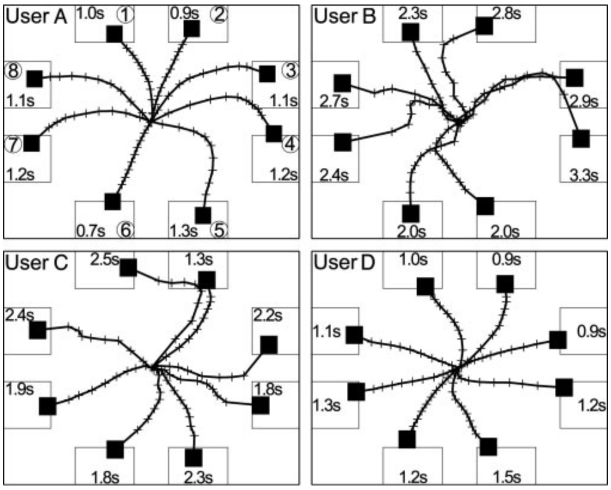

Figure 3.2: *Cursor trajectories. (Wolpaw and McFarland 2004)*

This study connects to motion intent recognition in terms that the subjects tended to use motor imagery to control the cursor. Figure 3.2 shows the trajectories of the cursor while reaching for the target during operation. The four subjects reached the target within the 10 secs allowed in 89%, 70%, 78%, and 92% of the trials respectively, and their average movement times for these trials were 1.9, 3.9, 3.3, and 1.9 secs. The results of this study show that people can learn to use non-invasively recorded EEG signal to control rapid and accurate movement of a cursor in two dimensions.

3.3 A case of single-trial EEG classification in a movement task with CSP

The classification of single-trial EEG-s recorded during preparation for movements of the left or right index finger or the right foot is described in (Müller-Gerking, Pfurtscheller and Flyvbjerg 1999). This study builds on the ERD in the alpha and beta frequency bands at the planning and the execution of the movement. In the case of finger or hand movement, the desynchronization starts in the contralateral (the side opposite to that of the motion) sensorimotor cortex during the planning phase and stays asymmetrical over both hemispheres until the movement onset.

Three subjects were asked to perform one of four movements after a series of stimuli, by pressing a micro-switch with left or right index finger, flexing the toes of the right foot, or moving the tongue to the upper gum. Each trial started with a short warning tone, followed by a visual cue (after 1 sec) on a computer screen, indicating a randomly chosen movement (from the four mentioned ones) which was to be performed. The actual movement was to be performed only after a third, acoustic stimulus occurring 2 seconds after the cue.

The EEG signal was recorded from 56 electrodes, with a reference electrode on the tip of the nose. The electrodes were arranged in a grid of 2.5 cm spacing covering the pre- and post-central areas. The recorded signal was filtered between 0.15-60 Hz, and sampled at 128 Hz. Electrooculograms (EOG-s) were also recorded to detect eye-movements.

The digitalized signal had been filtered to 6 different frequency bands: alpha (8-12 Hz), lower alpha (8-10 Hz), upper alpha (10-12 Hz), beta (19-26 Hz), gamma (38-42 Hz), and to a 8-30 Hz broad band. This was done with FIR filters.

Afterwards, the data from each filter bank underwent the Common Spatial Pattern method as spatial filtering. The data features of choice were $2m$ number of variance values var_p^i for $p = 1 \ldots m$ and from $(N - m + 1) \ldots N$ (m is the number of discrimination patterns of choice, i denotes the i-th trial, p is the index sweeping the feature vector, and N is the number of electrodes), which were normalized by the total variance of the projections on the retained patterns, and finally log-transformed:

$$f_p^i = \log\left(\frac{\mathrm{var}_p^i}{\sum_{p=1}^{2m} \mathrm{var}_p^i}\right) \tag{3.1}$$

where f_p^i is the p-th element of the feature vector within the current frequency band. The log-transformation is applied to make the distribution of the features normal.

The feature vectors extracted from the training data were used to calibrate the parameters of a linear (or quadratic) Bayesian classifier. Since the CSP method only works for two classes, pairwise classifications were performed between all the conditions. A trial was recognized for a given class only if that class obtained the majority in all of the pairwise classifications. Otherwise, the trial is classified as undecidable.

The best classification rates for 3 subjects were 94%, 90% and 84%, respectively. The best results were obtained with the 8-30 Hz wide band-pass filter before applying CSP. The high recognition rates and the CSP method's procedural and computational simplicity make it a promising method for BCI-s.

3.4 A case of motor imagery classification with Filter Bank CSP

Kai Keng Ang et al. (2012) developed a rather complex method for the classification of two of the BCI Competition IV datasets. These recordings contain 4 classes of motor imagery EEG trials: left hand, right hand, foot, and tongue. The dataset 2a was recorded with 22 EEG channels, along with 3 monopolar electrooculogram (EOG) channels, and used the left mastoid serving as reference. Dataset 2b also used 3 monopolar EOG-s, and comprises 3 bipolar recordings (C3, Cz, and C4).

The processing algorithm, named Filter Bank Common Spatial Pattern (FBCSP) consists of four progressive stages of signal processing and machine learning on the EEG data: a time-domain filter bank, a CSP spatial filter, feature selection, and classification of the selected CSP features.

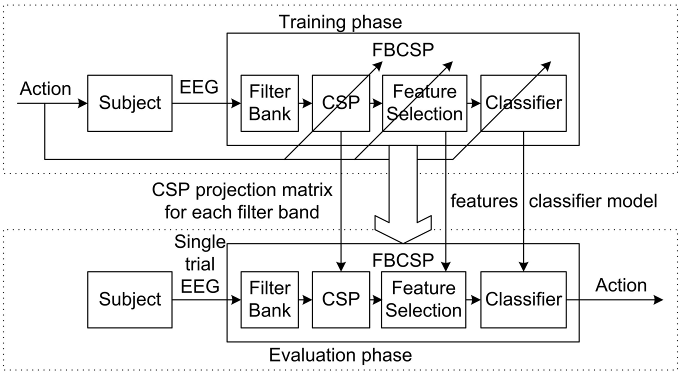

Figure 3.3: *Architecture of the filter bank common spatial pattern (FBCSP) algorithm for the training and evaluation phases. (Kai Keng et al. 2012)*

The time-domain filter bank decomposes the EEG into multiple frequency pass bands using multiple Chebyshev Type II band-pass filters. Altogether 9 band-pass filters are used with pass-bands 4-8, 8-12, . . . , 36-40 Hz. These frequency ranges are used because they yield a stable frequency response and cover the range of 4-40 Hz.

The spatial filter stage uses the CSP algorithm. Each pair of band-pass and spatial filter in the first and second stage performs spatial filtering on the EEG signal, and thus calculates CSP features that are specific to that frequency range.

For the third stage, the feature selection, the Mutual Information-based Best Individual Feature (MIBIF) and the Mutual Information-based Rough Set Reduction (MIRSR) algorithms are used. These 2 algorithms perform feature selection only on the training data by selecting the discriminative CSP features based on the mutual information computed between each feature and the corresponding motor imagery classes.

The last stage is the classification, for which a Naive Bayesian Parzen Window classifier is used. The CSP algorithm is for binary classification, but there have been several proposals for its multi-class extension, for example the Divide-and-Conquer, Pair-Wise, and One-Versus-Rest method, or the simultaneous diagonalization of covariance matrices from the multi-class data.

For the dataset 2a, the best results were reached by the Pair-Wise extension to FBCSP, yielding a kappa value (an indicator that measures the agreement between two raters classifying a set of items into mutually exclusive categories) of 0.572. Compared to this, the One-Versus-Rest method gave a result with no significant difference, and the Divide-and-Conquer scored worse. In the case of dataset 2b, FBCSP using the MIRSR algorithm for feature selection reached a better kappa value (0.599). The other feature selection algorithm's result had no significant difference in comparison. All in all the results of the Filter Bank Common Spatial Pattern algorithm makes this method promising for motor imagery classification.

3.5 A case of an ensemble classifier applied in motor imagery classification

An interesting improvement of the CSP method is described in (Lei et al. 2009). In this study, the outputs from different CSP subspaces are combined by majority voting. The main advantage of such classifier ensemble is that a combination of similar classifiers is very likely to outperform one of the classifiers on its own.

The EEG signal was recorded from three healthy, right-handed participants, between 22-26 years of age. During the recording, motor imagery tasks were to be performed and the discrimination of two classes were studied. The recording was made with two EOG and 127 EEG channels. As reference, the Cz electrode is used. The signal was sampled with 250 Hz and filtered between 0.1-48 Hz with a 8th order Butterworth filter. In off-line analysis, the data was down-sampled to 100 Hz and re-referenced to the common average.

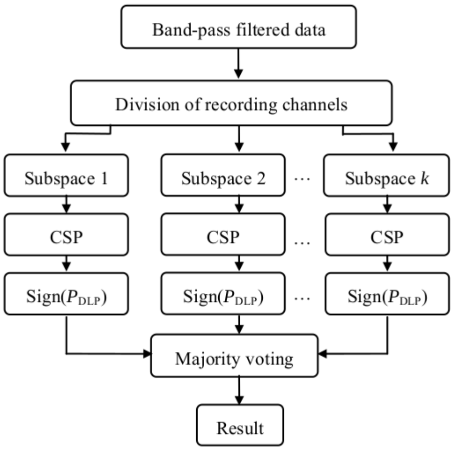

Figure 3.4: *The diagram of the classifier for EEG signal classification. (Lei et al. 2009)*

A subject-specific frequency band was selected based on the difference of the means of the two classes and the standard deviation of the union of the two classes for all participants.

This was usually close to the frequency band between 8-35 Hz. The signal was then filtered for this pass-band.

Afterwards, multiple channel selections were tried such as the full-channel set containing all 127 EEG channels, the sensorimotor channels, or a channel bank (comprising 10 different channel sets located in different areas of scalp). Then CSP filters were calculated for 10 individual datasets, each containing 2 patterns.

The features are obtained by projecting the signal using the spatial filters, computing their power, applying logarithm, and calculating the difference of such log-power values coming from two tasks:

$$P_{DLP}(\mathbf{S}) = \log(\mathbf{w}_a^{\mathrm{T}}\mathbf{S}\mathbf{S}^{\mathrm{T}}\mathbf{w}_a) - \log(\mathbf{w}_b^{\mathrm{T}}\mathbf{S}\mathbf{S}^{\mathrm{T}}\mathbf{w}_b) \tag{3.2}$$

where $\mathbf{w}_{a,b}$ denote the spatial filters corresponding to classes a, b, and $\mathbf{S}$ is the matrix of the signal from one trial.

The sign of P_{DLP} is interpreted as the predicted class in case of a single classifier. With the 10 different spatial filters, 10 feature values are computed, and summed to represent the majority voting method. The sign of the final sum is the basis of decision between classes a and b. According to the results in the study, this CSP ensemble method outperformed LDA classifiers and support vector machines, making it a promising method for BCI applications.

Chapter 4

Designing the hand motion detection algorithms

This paper so far presented the basics of the EEG signal and its nature during movements, and the basics of digital signal processing algorithms for the purpose of motoric intent recognition. Some interesting past works, related to this purpose, were also presented. The primary objective of this study is to design digital signal processing algorithms capable of detecting hand movement. The current chapter presents the development and the evaluation of these algorithms. They were designed for hand motion detection generally, but for the development, a certain dataset with a limited variety of hand motions was used.

4.1 The dataset and the task

The series of EEG recordings used for the detection task was derived from the WAY-EEG-GAL dataset (Luciw, Jarocka and Edin 2014), transformed by Kaggle Inc. (Kaggle Inc. 2015a). During the recordings, twelve healthy participants performed grasp-and-lift series in which the object's weight were changed unpredictably between trials, thus enforcing changes fingertip force coordination.

In each of the trials, the participant was cued to reach for the object, grasp it with the thumb and index finger, lift it and hold it for a couple of seconds, put it back on the support surface, release it, and, lastly, to return the hand to a designated rest position. The beginning of the reach and the lowering of the object was cued by a LED, otherwise the pace of the task was up to the participant. In total there were 12 subjects, 10 series of trials for each subject, and approximately 30 trials within each series. The training dataset contains the first 8, and the test set contains the 9th and 10th series for each subject. The true labels for series 9 and 10 are unfortunately not available, as these are reserved as test sets for competitions. It is possible to get a solution file for series 9-10 evaluated as well, by manually uploading it to a submission portal on Kaggle's website.

The EEG was recorded with 32 channels (the electrode locations used are shown in Figure 4.1), marked with numbers (not by the black/white fill). The signal was first sampled at 5 kHz and band-pass filtered between 0.016-1,000 Hz. The signal was finally sampled with the rate of 500 Hz preceded by an adapted low-pass filter to prevent aliasing.

Using the dataset, the task was to detect 6 events with labels "HandStart" meaning the start of reaching for the object, "FirstDigitTouch" the moment of contact between the finger and the object, "BothStartLoadPhase" the appearance of the lifting force, "LiftOff" the object leaving the support surface, "Replace" the object being replaced on the platform, and "BothReleased" meaning the releasing of the object with all fingers. The participants were healthy, and they actually carried out the motion. Even though motoric intent detection and motion detection mean slightly different things, in the current context they can be considered the same. This is because the input dataset provides only EEG values, and it does not contain other measurements like arm angles or EMG.

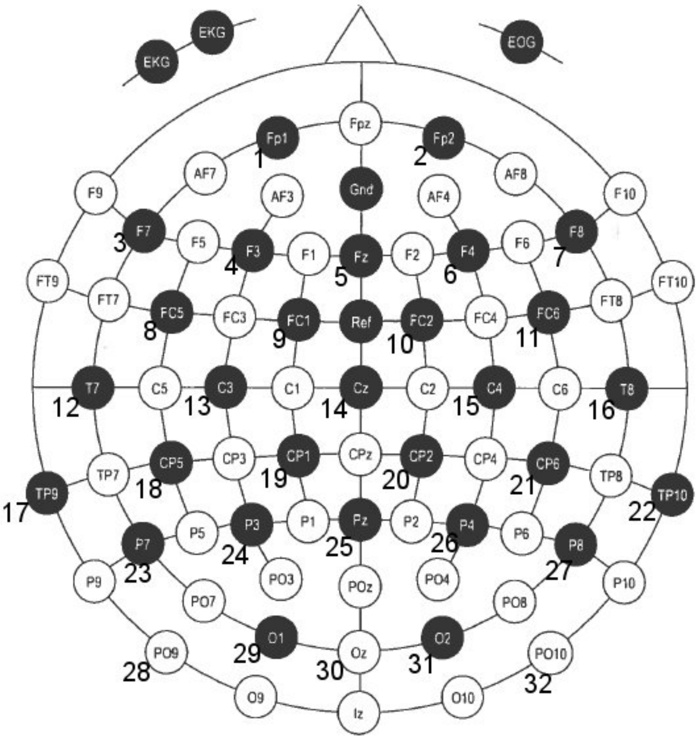

Figure 4.1: *The 32 electrode locations used for the recording, marked with numbers (not by the black/white fill). (Kaggle Inc. 2015b)*

In the dataset, the events always occur in the same sequence. The event files for the training data contains truth values for all events. The six label columns are either zero or one, depending on whether the corresponding event has occurred within ±150 milliseconds. For the detection task, the expectation was to predict a probability of one for the entire window, and zero outside it. The time series' of detections were evaluated by calculating the area under the receiver operating characteristic curve (AUROC), which is described well in (Tape 2015). This is calculated for each event, then the event-wise AUROC values are averaged over all 6 events. In the ideal case, the AUROC is 1, when regardless of decision threshold (between 0 and 1), all the events are detected, and no false detections are made. A serious difficulty in achieving high detection results in a BCI is that the classification decisions are not made between classes with similar population size. The events which we are looking for only occupy a tiny percentage of the time. This makes seemingly low False Positive Rates (even 1-2%) still practically unacceptable. They still mean a large number of false detections compared to the true events.

4.2 The rules

The procedure must be causal. It must not use data originating from the time after the moment under examination. This is for imitating an on-line BCI processing algorithm, where data from the future is really not available. Training the processing models on other subjects and series outside of the currently analyzed series is permitted. This was not of great help, as the participants' brain activity has different personal characteristics.

My personal goal takes the basic task one step further. My aim is to design an algorithm, which is generally capable of detecting the hand movements involved in the dataset. This means some important constraints. The first is that even though the order of the movements is known and guaranteed to be fixed, I do not consider it so. This means that when analyzing a certain point of time in the signal, I do not scan for the last couple events which happened just recently. A state-machine approach, always logging which event happened last, could increase detection accuracy in a sequential trial like this, if it does not miss any event (otherwise it would get lost in the network of states). However, in real life, the order of the movements of a person is rather arbitrary. Therefore I assume that the order of the hand motion events are arbitrary.

Another constraint I set for my solution is that it may not measure the time elapsed since the end of trials. Consecutive trials during the experiment happened in similar distances from each other in time. This could also be used as information for estimating the probability of the beginning of the next trial at a given point of time. For real-life reasons similar to what I have described at the previous constraint, I do not use this information in the design of my detection algorithms.

The last limitation I set for the solution is that I do not explicitly look for the occurring of a Visual Evoked Response (VEP) when detecting the first hand movement in a trial. In the experiment, the participants were to start a trial when a LED light flashed up in front of them. This presumably triggered a VEP in the visual cortex of their brains. The detection method of the first event in the sequence could then build on this fact and directly look for the VEP in the channels which are positioned over the visual cortex. In natural, real-life motion, there are no visual indicators lighting up to call for movement. Therefore, I do not design my algorithms in a way that they directly search for features in the visual cortex electrodes' signal. Adaptive algorithms probably find parts of this feature in the complete multi-channel signal and use it for the detection task. However, not ordering them explicitly to do so is a step towards making the detection procedure more lifelike.

The algorithms designed during this study have been evaluated on series 1-8. These were separated into a training set and a test set. As the true event labels are available for these series, evaluating the algorithms could be carried out on a local computer.

4.3 The solution approach

The main approach for designing the algorithms was to first analyze the most popular methods and working solutions, then to improve them at some points, and finally to combine them into a new one which presumably aggregates the strengths of the individual ones.

The algorithms designed in this study create models for each subject individually. A global model, which instantly fits all the subjects would be a great achievement, and this is the future direction of the research. Unfortunately, such universal approach does not yield remarkable accuracy in detecting the movements, or it is too complex computationally to be applicable as of today. Therefore the current approach is to create subject-specific models for each participant.

The approach for writing a computer program for the procedures was to make it modular in terms of the different steps in the digital processing algorithm. In the end, one main Python script has been prepared, along with two auxiliary files containing class definitions and functions. The script follows the scheme of the procedure outline. It has been written in a way that different methods for each step (time-domain filtering, classification etc.) can be chosen simply by setting flags and adjusting parameters. At the end of every run, the script exports the current configuration of the procedure. With such files it is easy to track the process of experimenting with parameters, as well as to compare different models.

Before starting to write a digital processing algorithm for the data, one must estimate the specificities of the brain activity during the recording. I took into consideration the possible activity of each frequency band. The participants were properly seated during the experiment, and they were awake. Therefore I eliminate the possibility of a participant being in the state of deep sleep during the experiment. However, the delta activity may be interesting if the participants focused with continuous attention, which might be shown in channels coming from the frontal parts. During the pause between two grasp-and-lift trials, there were possibly lapses in attention. This makes the theta band interesting at channel locations not related to the task. The alpha band may show signs of readiness just before the beginning of a trial. The beta (more precisely the beta 2) band is the most interesting because of presumable intense focus and cognitive processing going on in the participants' brain during the task. The gamma range over the somatosensory cortex might be active because of the short-term memory matching of the object's touch when the participant have reached it. I expect the motor neurons to show high activity during the hand movement. Therefore, the mu rhythm will probably also carry information in the channels coming from over the sensorimotor cortex. Mu-range power is expected to decrease over contralateral motor cortex (on the opposite side of the moving hand). It is also expected to increase over the ipsilateral motor-cortex (on the same side as the moving hand). Figure 4.2 shows CSP patterns of the subjects as an illustration of the brain activity at a motion event.

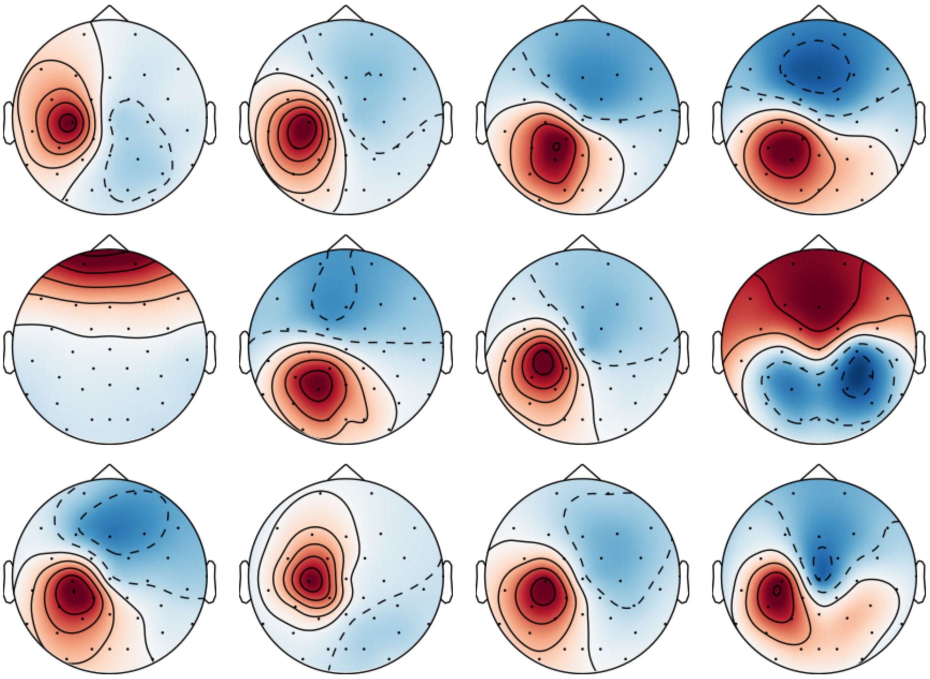

Figure 4.2: *CSP patterns of the subjects. (Barachant 2015)*

Besides ERP-s in the above mentioned frequency bands, the presence of Bereitschaftspotential right before a motion event (especially the first one) is also expected. Even if this has a tiny amplitude and may be easily lost in noise, I tried feeding its frequency range into the classifiers in some of the cases described later in this chapter.

Because of a LED light triggering the execution of each trial, a Visual Evoked Response is presumably present in the signal just before the first event ("HandStart"), in the channels coming from over the visual cortex. This causes a little corruption in detecting the first movement, which unfortunately affects the lifelikeness of the task, even if the algorithm does not explicitly look for the VEP in the visual cortex.

It is expected that the signal is polluted with artifacts, particularly over the frontal lobe because of inevitable eye movement such as blinking. Such artifacts manifest in the signal as highly salient spikes, as it is shown in Figure 4.3.

The voltage levels of these is much larger than those of natural brain activity. Thus it is a good approach for reject these artifacts by applying a threshold to the signal values. If a voltage value higher than this threshold is measured, then the signal is silenced (forced to be zero) for that moment or for a short amount of time. The silencing period may be as long as the average time of a blink, between 100-400 milliseconds. (Milo et al. 2010).

The digital processing algorithms have been written in Python 3.4.3. The main Python modules used were:

- `pandas` for file operations,
- `math` and `numpy` for matrix calculations,
- `scipy` for time-domain filtering,

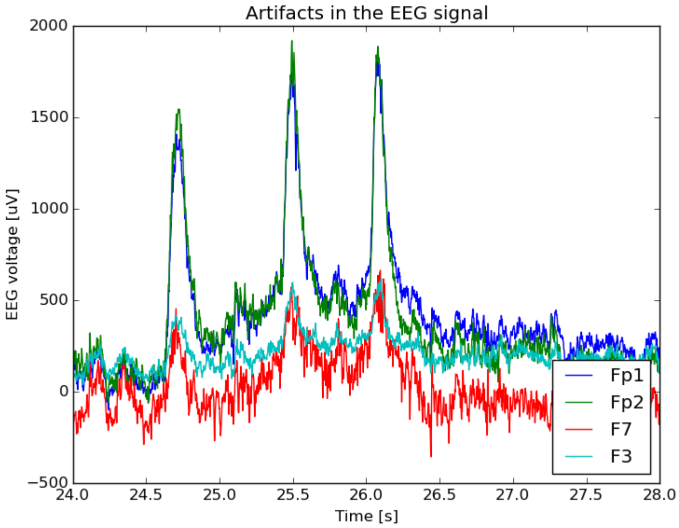

Figure 4.3: *Artifacts in the EEG signal. The spikes probably correspond to blinks.*

- `mne` for epoching and spatial filtering,
- `sklearn` for basic machine learning,
- `lasagne` and `nolearn` for Neural Networks,
- `joblib` for parallel computing,
- and `matplotlib` for plotting.

I coded several previously mentioned methods in Java to better understand their underlying math and to see their computational requirements when using the CPU or the GPU. For writing the actual digital processing program, I switched to Python for its widely used and well-tested libraries.

Time-domain filters are rather easy to create using the `scipy.signal` module. For my algorithms, the filter design methods tried were Remez (using the Parks-McClellan algorithm), Kaiser window, and Hamming for FIR filters, and Butterworth and Chebyshev type II. for IIR filters (Oppenheim, Schafer and Buck 1999). The first reason for choosing these methods is that they were elaborately described in the literature. Another reason is that they are popular in applications. The third reason is that they can be designed easily with the available programming libraries. Examples of the frequency responses of these filters can be seen in Figure 4.4.

The digital processing can be rather time-consuming. PCA can be used to reduce the data dimensionality, but using it did not cause any significant speedup in the process. It may function as spatial filter as well, but for that purpose I preferred CSP. The reason for this is that CSP brings better results as it uses the information of the training data labels as well, not just the signals.

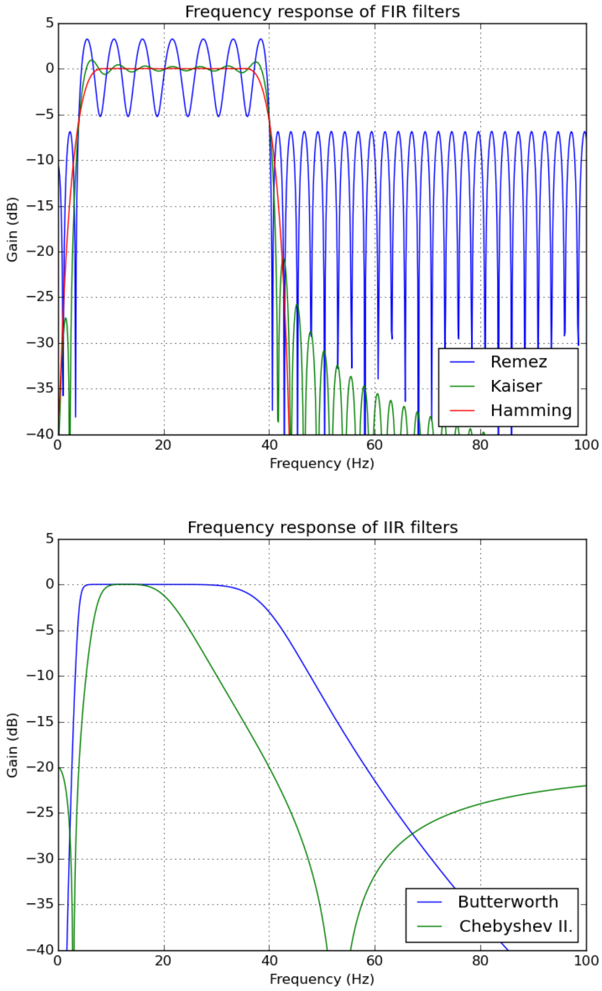

Figure 4.4: *Examples of FIR and IIR filter frequency responses (band-pass between 4-40 Hz).*

The first kind of feature generation consisted of calculating the channel variances over a time window starting before and ending right at the examined moment. In other cases, simple normalization was used. This was carried out using signal mean and standard deviation values estimated from the training data. Theoretically, normalization does not affect the operation of most of the classification methods. However, depending on the initialization of a classifier's parameters, and hence their initial magnitude distribution, a method might converge faster for normalized signal values.

For classification, I tried the Naive Bayes, LDA, QDA, Support Vector Machine, Logistic Regression, and Neural Network methods. Logistic Regression was preferred over Naive Bayes, LDA, and QDA based on evaluations over small datasets. SVM-s were also tried, but due to my little knowledge about their parameter optimization method, they weren't applied in the final algorithms. The major difference between my processing algorithms are the examined frequency ranges and the classifier of choice. The classification method strongly determines the kind of preprocessing to be applied. The final output needed from the classifier (or ensemble of classifiers) is not a class label, but a probability of a certain class. With the packages used for the program, this is easy to carry out. The probability values calculated for each time point in the series were the final outputs then.

4.4 Algorithm #1: Band-pass Filter Bank Common Spatial Pattern with Logistic Regression

This algorithm builds on the CSP method to enhance signal variances at the motion events. Its flow is shown in Figure 4.5.

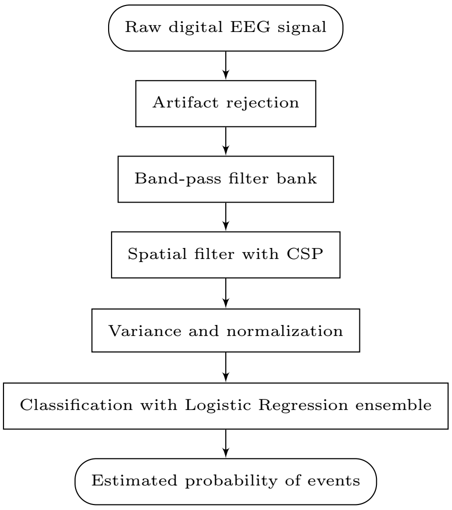
Raw digital EEG signal
Artifact rejection
Band-pass filter bank
Spatial filter with CSP
Variance and normalization
Classification with Logistic Regression ensemble
Estimated probability of events

Figure 4.5: *The procedure of Algorithm #1.*

The artifact rejection is done by zeroing out signal values above an experimental threshold. Afterwards, a filter bank consisting of two FIR band-pass filters (using Hamming window) is applied to the signal. FIR filters are preferred here because of their linear phase characteristics. The characteristics of the filter bank are shown in Figure 4.6.

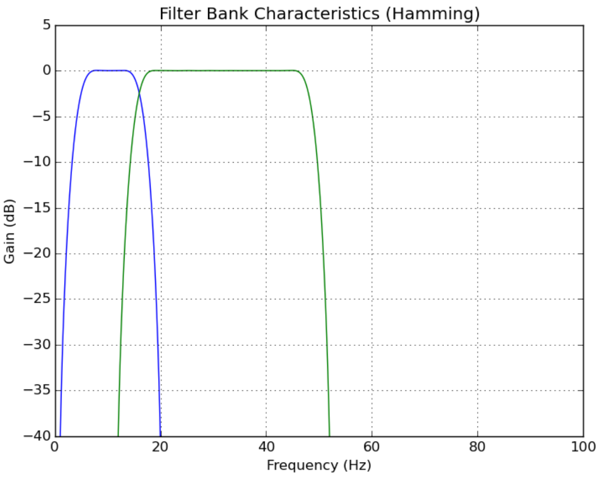

Figure 4.6: *The filter bank characteristics used in Algorithm #1.*

The frequency bands of the filters overlap. If they didn't, the frequencies at their border would be attenuated (or cut out) in both of the filter outputs. This would mean a loss of information in the global frequency range of interest. A couple of more complex time-domain filter banks have been tried. Experimenting with them took a lot of computational time, and using them did not yield better results. Thus, the basic two-filter bank is kept.

The signals are then spatially filtered using the CSP method. The spatial filters are calculated separately for each event. Features are generated then by calculating the signal variance within a time window. This is actually carried out by convolving the signal with a boxcar function, which is set to be as wide as the window. The variance values are then normalized. For classification, Logistic Regression ensembles are used, separately for each event. Within one event, the probabilities estimated by individual classifiers (each analyzing one time-domain filter output) are combined using arithmetic mean. This mean value is taken as the final estimated probability value for given event. The parameters of the algorithm can be found in Table 4.1.

Algorithm parameters	
Time-domain filter	Band-pass filter bank, FIR (Hamming) filters with frequency bands 4-17 and 15-49 Hz, order 251
Spatial filter	CSP, with 0.5 sec window, using the first and last 8 components
Feature generation	Variance within 0.5 sec window, normalization
Classification	Logistic Regression ensemble for each event separately

Table 4.1: *The parameters of Algorithm #1.*

The algorithm has been executed for each participant separately. The result was an event-wise AUROC of 0.722 over all of the subjects. Figure 4.7 shows the ROC curves for each event resulted from this evaluation.

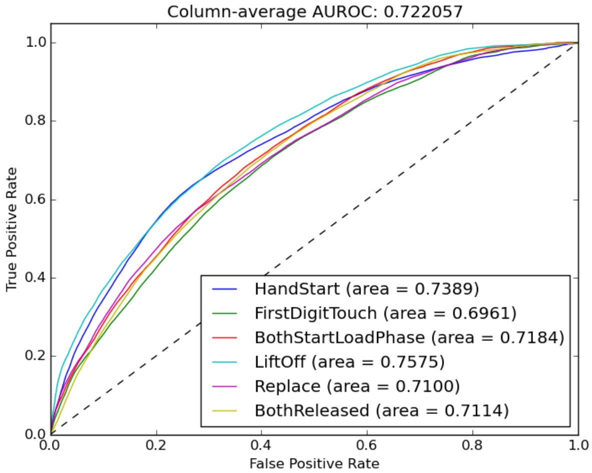

Figure 4.7: *The ROC curves resulted from Algorithm #1.*

4.5 Algorithm #2: Low-pass Filter Bank with Logistic Regression

This procedure focuses on the differences between frequency ranges, rather than examining bands separately. The filter bank concentrates on the low frequencies, below the beta range.

This frequency range seems to carry valuable information for the detection task. Therefore, this algorithm performs a detailed examination of the signal's features in that range. The flow of the process is shown in Figure 4.8.

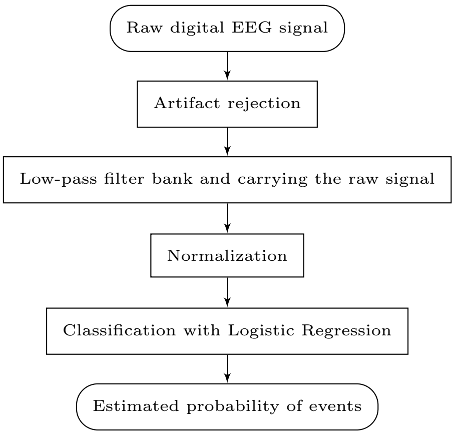
Raw digital EEG signal
Artifact rejection
Low-pass filter bank and carrying the raw signal
Normalization
Classification with Logistic Regression
Estimated probability of events

Figure 4.8: *The procedure of Algorithm #2.*

Artifact rejection is applied by zeroing out signal values above an experimental threshold. Afterwards, the time-domain filtering is done by a filter bank with fourteen IIR filters. In this algorithm, IIR filters were preferred because they take less time than with FIR filters. Because of this, they made experimenting with model parameters much faster. In the current case when we have this many filters, there is a significant difference between the computational time of IIR and FIR types. The cutoff frequencies are densely defined in the filter bank below the beta band. The characteristics of the low-pass filter bank are shown in Figure 4.9.

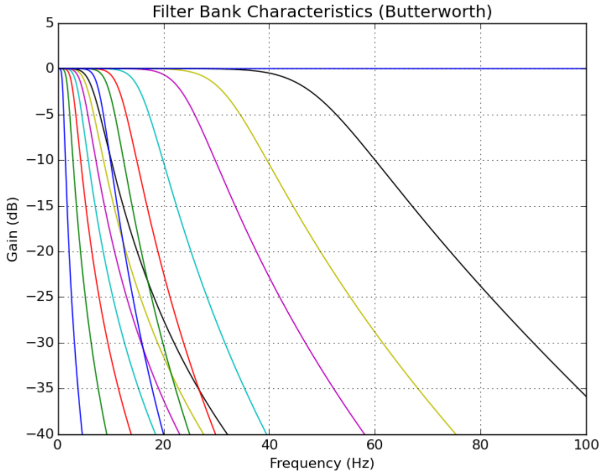

Figure 4.9: *The low-pass filter bank characteristics used in Algorithm #2.*

There are filters with cutoff frequencies in the beta and gamma bands as well. The filter bank could contain even more filters, which would probably lead to slightly better results. Experimenting with 18-20 cutoff filters on smaller parts of the dataset yielded a little increase in the detection accuracy. Unfortunately, the available RAM imposes limits on the number of filters. Therefore, the final processing went with fewer filters.

The filtered signals, along with the raw signal are then normalized using mean and standard deviation estimates calculated from the training data. Afterwards, each filtered signals, each channels are inserted next to each other in one feature matrix. The classification is done using Logistic Regression, separately for each event. The classifiers' input is the vector of the raw and the low-pass filtered, then normalized signal values at the time point of interest. The parameters used in the algorithm are summarized in Table 4.2.

Algorithm parameters	
Time-domain filter	Low-pass filter bank, IIR (Butterworth) filters with cutoff frequencies at 1, 2, 3, 4, 5, 6, 7, 8, 10, 12, 16, 24, 32, and 49 Hz, order 3 below 4 Hz and order 5 above
Spatial filter	-
Feature generation	Taking the values of the signals from every filter and the raw signal as well, normalization per channels
Classification	Logistic Regression for each event separately

Table 4.2: *The parameters of Algorithm #2.*

This algorithm also has been executed for the data from each participant separately. It has reached an event-wise average AUROC result of 0.796 over all of the subjects. The ROC curves for each event resulted from this evaluation are shown in Figure 4.10.

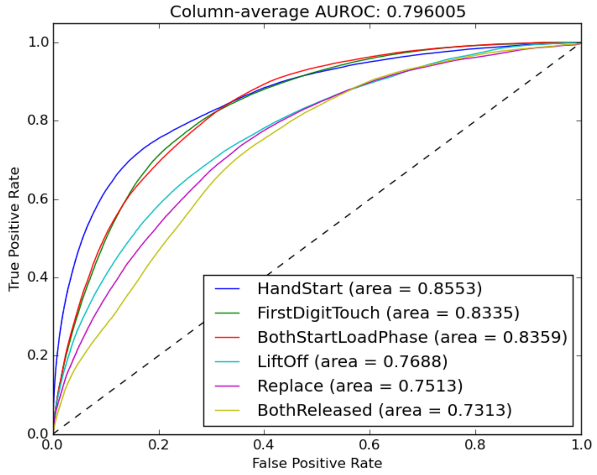

Figure 4.10: *The ROC curves resulted from Algorithm #2.*

4.6 Algorithm #3: Normalization and a Convolutional Neural Network

This third algorithm builds on the highly adaptive nature of Convolutional Neural Networks. The CNN used in this algorithm is designed to have a large amount of parameters. Little preprocessing is done before the classification of features. Figure 4.11 shows the flow of this algorithm.

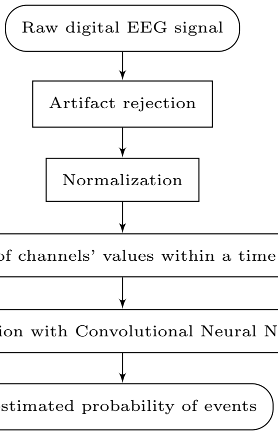
Raw digital EEG signal
Artifact rejection
Normalization
Extraction of channels' values within a time window
Classification with Convolutional Neural Network
Estimated probability of events

Figure 4.11: *The procedure of Algorithm #3.*

The algorithm first rejects the artifacts with a simple threshold. The signal's channels are then normalized using the mean and standard deviation estimated from the training data. Afterwards, the normalized signals are fed into the CNN directly, without any further temporal or spatial preprocessing.

The network's inputs are the values of all the EEG channels within the 2 second time window just preceding the moment of interest. The convolution inside the network happens along the time dimension only. The CNN consists of 6 layers, including the input and output ones, and it has 3,487,814 internal parameters altogether. The structure of the network is shown in Figure 4.12.

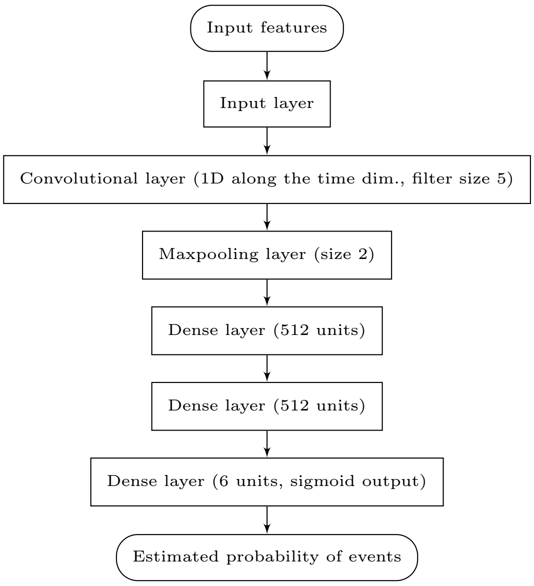
Input features
Input layer
Convolutional layer (1D along the time dim., filter size 5)
Maxpooling layer (size 2)
Dense layer (512 units)
Dense layer (512 units)
Dense layer (6 units, sigmoid output)
Estimated probability of events

Figure 4.12: *The structure of the CNN used in Algorithm #3.*

The sequence of layers in the network is the following: input layer, 1-dimensional convolutional layer, maxpooling layer, dense layer, dense layer, dense layer. The last dense layer serves as the output layer. The purpose of the convolutional layer is to filter the input data in several different ways. The maxpooling layer is there to make the network more robust against feature offsets. The dense layers further combine the pooling layer's output values. There is a moderately high number of internal parameters in the convolutional and the dense layers. This is to ensure a refined CNN which still does not require too much computational capacity.

As the CNN is trained on a large amount of data, a batch iterator object is used to generate the individual learning samples during the training, instead of passing a large matrix in every turn. This batch iterator stores all the data before the CNN's learning phase, and only indices to data samples are passed to it at the time of training. Using these indices, the batch iterator can select and feed samples quickly to the CNN. The algorithm separates 20% of the training data, and uses it only as cross-validation set. For the duration of the training, dropout layers are inserted into the CNN after each convolutional and dense layers (except the output layer) in order to mitigate the risk of overfitting. The output error's measure, the loss function is the mean of the output channels' individual cross-entropy:

$$C = \sum_{i=1}^{N_o} [-t_i \log(p_i) - (1 - t_i) \log(1 - p_i)] \tag{4.1}$$

where N_o denotes the number of outputs, and i is the index of the output channel. For adjusting the CNN's parameters, the Adam stochastic optimization method is used (Kingma and Ba 2014). The attributes of the network used for this algorithm are summarized in Table 4.3.

CNN attributes	
Layer types	Input, convolutional (1D), max-pooling, dense, dense, dense
Loss function	Mean cross-entropy
Internal output nonlinearity of layers	None, linear transformation is used
Nonlinearity of the output layer	Sigmoid
Dropout probability	0.5
Parameter optimization method	Adam with learning rate 0.001

Table 4.3: *The attributes of the CNN.*

The training of the CNN took around 1600 seconds per subject. Figure 4.13 shows the training and validation losses over time during the learning phase, of one CNN.

Only one CNN is used for the classification of all six motion events, per each subject. The evaluation was performed in the same way as for the two other processing algorithms described in this chapter. This third algorithm, using a CNN, yielded an event-wise AUROC of 0.829. The resulting ROC curves are shown in Figure 4.14 for each event.

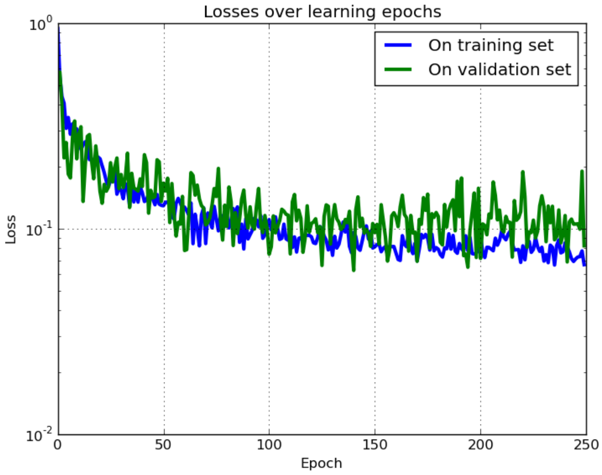

Figure 4.13: *The training and validation losses over time during the learning phase of a CNN.*

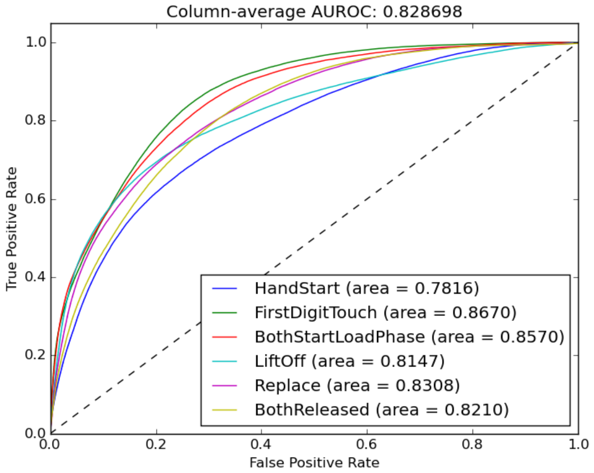

Figure 4.14: *The ROC curves resulted from Algorithm #3.*

4.7 Algorithm #4: Hybrid

The final algorithm simply merges the strengths of the previously ones. It selects the best algorithm from previous ones for each event separately. Based on the AUROC results, Algorithm #2 is used for the first event (HandStart), and Algorithm #3 for the rest of the events. For 5 out of the 6 motion events, the third algorithm brought the best results. For the first event (HandStart), the second event reached the highest AUROC. The hybrid approach achieved an event-wise averaged AUROC 0.841. It is not a surprise that this is the best result among those of the described algorithms. The ROC curves obtained from this result are shown in Figure 4.15.

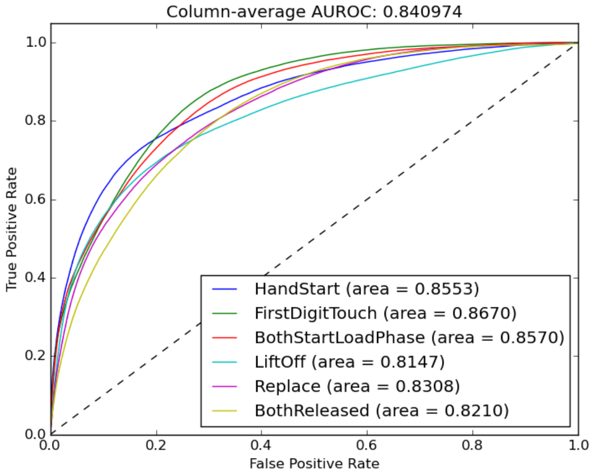

Figure 4.15: *The ROC curves resulted from the final, hybrid algorithm.*

4.8 Evaluation

The AUROC results of the above described algorithms are summarized in Table 4.4. The results of algorithms #1, #2, and #3 are also represented by bar charts in Figure 4.16 for visual comparison. The AUROC of the last, hybrid algorithm are visualized separately in Figure 4.17, as this simply reproduces the best results of the first three method per each event.

Event-wise AUROC results

Event	Algorithm #1	Algorithm #2	Algorithm #3	Algorithm #4
Handstart	0.739	0.855	0.782	0.855
FirstDigitTouch	0.696	0.834	0.867	0.867
BothStartLoadPhase	0.718	0.836	0.857	0.857
LiftOff	0.758	0.769	0.815	0.815
Replace	0.71	0.751	0.831	0.831
BothReleased	0.711	0.731	0.821	0.821
Average	0.722	0.796	0.829	0.841

Table 4.4: *The event-wise AUROC results of the algorithms.*

The first algorithm reached the lowest score. Using CSP did increase the detection accuracy, as a notable AUROC value has been achieved this way, with a time-domain filter bank comprising only two band-pass filters. The accuracy could be increased with a more complex filter bank, which could be further improved with adaptively finding the cutoff frequencies. The accuracy could also be increased with using weighted averaging when forming the output of the classifier ensembles. It could also be increased with using more complex features, such as the covariance matrix as a whole, not only the set of variances.

The second algorithm performed surprisingly well, considering that it used very simple features with a basic type of classifier. This algorithm might have exploited the information from the sub-beta frequency ranges very well. It could probably be improved by refining the time-domain filter bank in the beta and gamma ranges, and by using more complex features and classification methods.

The third algorithm brought the best results among the first three. This method relied on the adaptivity of the CNN, which demonstrated notable performance. Studying and applying feature generation methods suitable for CNN-s would probably increase the detection accuracy. Using a time-domain filter bank could also raise the performance, but this would require high computational capacity. The results could be improved with the use of a more complex CNN, or an ensemble of CNN-s as well.

The last algorithm only combined the previous ones. The principle of this hybrid approach is rather simple. It takes the best method for detecting a certain motion event. This algorithm can be thought of as an ensemble of complex methods.

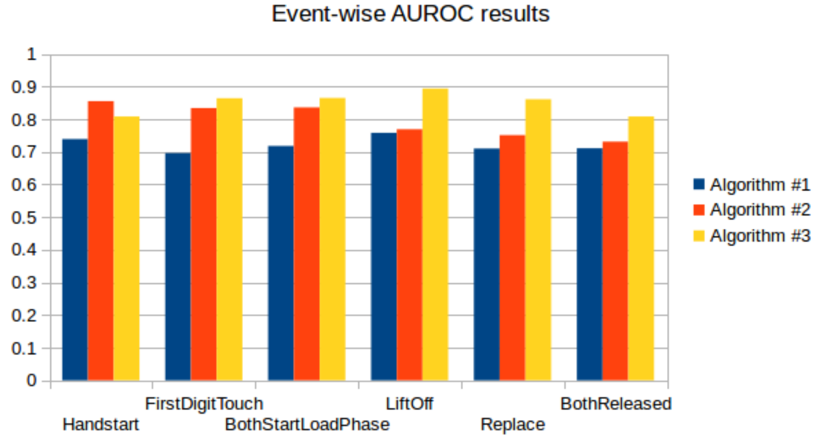

Figure 4.16: *The event-wise AUROC results of the motion detection algorithms #1, #2, and #3.*

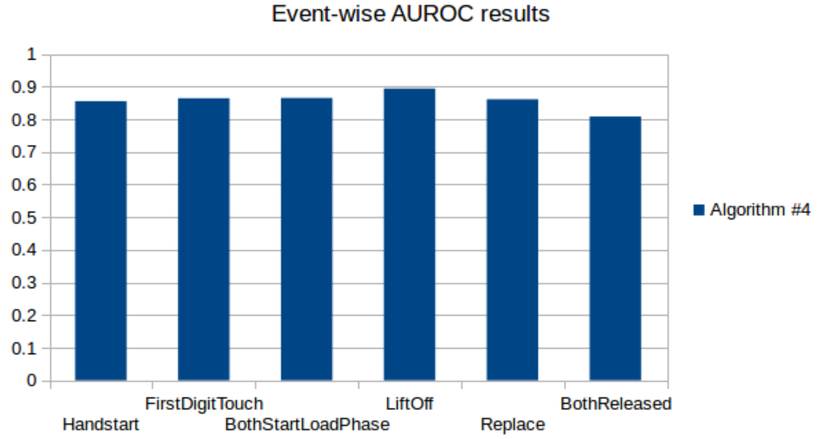

Figure 4.17: *The event-wise AUROC results of the hybrid motion detection algorithm.*

The execution time of each algorithm can be found in Table 4.5. The algorithms were run on a computer equipped with an Intel Core i5 4-core CPU and 4 Gb of RAM.

Execution times		
Algorithm	Training	Testing
#1	3.37 hrs	25 mins
#2	4.04 hrs	4 mins
#3	5.33 hrs	60 mins
#4	6.01 hrs	61 mins

Table 4.5: *The execution time of the algorithms. The testing phase means the motion detection from the test data.*

In a real-life BCI application, it is obviously expected from the digital processing algorithm to perform one detection between the arrival of two consecutive samples. Otherwise, the process couldn't keep up with the incoming samples, and it would be useless. It is not a serious issue if the process gets into a slight delay after 1 sample, if it manages to get back in sync with the next samples. The total testing time then serves as a rather good basis for comparison to other algorithms.

However, comparing the testing time to the total time of the EEG recordings is the most interesting, when evaluating the applicability. The EEG recordings used for testing the algorithms is 53 minutes long in total. Algorithms #1 and #2 clearly fall within the time constraint, as they took half the time as the recordings. Algorithms #3 and #4 on the other hand take longer than the recordings, by just a few minutes. This might be troublesome on first sight, but it does not mean that these algorithms may not perform well computationally.

Note that the algorithms were developed and tested on a consumer category computer. There are two reasons why the last two algorithms can satisfy the time constraint. The first reason is that the program written for the digital signal processing is not optimized in terms of CPU and RAM usage. After a fine optimization, the execution time of the last two algorithms might drop right below the constraint. The other reason is that the computer's hardware was not designed specifically for complex and large computations. Installing the program on digital signal processing hardware, or exploiting the performance of a GPU could give significant boost to the speed of execution. Nowadays such devices are becoming more and more available, and their prices are decreasing to more affordable levels. Also, they can be manufactured in sizes appropriate for embedded systems, so they are suitable for everyday BCI applications. Based on these thoughts, I consider algorithms #3 and #4 applicable as well in real-life in terms of speed.

The questionable part is the detection accuracy. The AUROC values measure the quality of a classifier without a fixing the decision threshold at an exact value. It is up to the engineer to determine the threshold according to the acceptable rate of false positives. If we look at the ROC curves obtained by the algorithms, we can see that 50% True Positive Rate still comes with more than 2% False Positive Rate. Converting these to absolute measures, the number of correct detections for the current amount of data is a couple hundred, while the number of false ones is still a few thousand.

A BCI with such rates is clearly not reliable. There are a couple of promising ideas for increasing the detection accuracy. The first one is to silence the output of the detector for a short amount of time after a positive detection. This would decrease the potential disturbance caused by false positives coming right after a correct detection. Another idea is using the P300 phenomenon to cancel action triggered by a false detection. For example, noticing the movement of a prosthetic hand would act as a stimulus, if such motion was not intended. This would evoke a P300 response in the brain. The BCI would recognize this response, and stop the prosthetic limb's movement.

Despite that these results do not enable applicability, the final products of this study have great value. The variations of existing processing methods provided new information through the experiments. Besides, a Python program was written for the processing algorithms in a modular structure, which then can be easily fitted for BCI applications or other signal processing tasks.

For the grasp-and-lift dataset, Kaggle hosted a competition, which did not have that kind of rules regarding the everyday applicability like the ones I declared for this work. The best submissions achieved around 0.95 or even higher AUROC values. The winner achieved AUROC above 0.981 (Barachant and Cycon 2015). The publicly released solutions used large and complex Convolutional Neural Networks, with 11 or even more layers. Some used ensembles of CNN-s. There was a solution which was said to take 4 days to run even on a computer with a modern high performance GPU. The results of these methods are outstanding with today's measures. However, my personal approach is that we should increase the focus on developing algorithms which are transparent as much as possible. It is difficult to apply such highly adaptive methods to new tasks if the internal transformations are not known analytically. However, the fact is that CNN-s perform very well in numerous data processing applications. In my opinion, the next step in the development of CNN-s and similar highly adaptive methods (such as Recursive CNN-s) is to analytically uncover the effects of utilizing the different kind of building blocks more deeply.

Summary

The objective of this paper was on one hand to review the main methods of motion detection from scalp EEG signal. The other objective was to design digital signal processing algorithms which are capable of detecting hand motions from a given dataset. The deeper motivation for these objectives was the development of reliable Brain-Computer Interfaces.

In the first chapter, the acquisition and the analysis of the EEG signal were described, along with the basics of the human brain activity. This chapter also elaborated on the possible uses of the EEG signal. The second chapter described the EEG signal's nature in the context of motion detection. Important phenomena such as EP-s, ERS/ERD, and BP were presented. The chapter then outlined a standard digital processing algorithm for motion detection and described the possibilities for implementing each step in the procedure. The third chapter presented remarkable works and results from researchers and engineers in various institutes and companies. The fourth chapter described the motion detection task targeted by this study, and the algorithms which were designed to solve it. The chapter then evaluated these algorithms based on their detection results.

The results achieved by the algorithms can be characterized as promising. They ended up not too far from the best results in terms of AUROC. Unfortunately, setting a decision threshold high enough to ensure a feasible True Positive Rate would also permit a large number of false detections, as it was shown. However, by implementing a simple artifact rejection and a hybrid combination of rather different algorithms, the initial methods were successfully improved, even if only by a little bit.

A couple of ideas were described for increasing the detection accuracy with auxiliary methods, such as output silencing after action, or canceling with P300. These constitute one of the future directions towards reliable BCI-s. Another direction is the improvement of analytical classification methods and the analysis of highly adaptive numerical ones.

From this study it is visible that there is still a long road ahead of us towards reliable BCI-s, applicable in everyday use. The development in the field in the past decade projects good perspective, and numerous researchers pursue the objective presently. The futuristic vision of BCI-based fully functional prosthetics and physical augmentation might be far, but it is certainly in our line of sight.

Bibliography

Agashe, Harshavardhan Ashok et al. (2015). 'Global cortical activity predicts shape of hand during grasping'. In: *Frontiers in Neuroscience* 9.121.

Ang, Kai Keng et al. (2008). 'Filter Bank Common Spatial Pattern (FBCSP) in Brain-Computer Interface'. In: *IEEE International Joint Conference on Neural Networks*, pp. 2390–2397.

Barachant, Alexandre (2015). *Common Spatial Pattern with MNE.* URL: `https://www.kaggle.com/alexandrebarachant/grasp-and-lift-eeg-detection/common-spatial-pattern-with-mne` (visited on 19/10/2015).

Barachant, Alexandre and Rafal Cycon (2015). *Code and documentation for the winning solution in the Grasp-and-Lift EEG Detection challenge.* URL: `https://github.com/alexandrebarachant/Grasp-and-lift-EEG-challenge` (visited on 25/10/2015).

Bartlett, M. S. (1948). 'Smoothing Periodograms from Time-Series with Continuous Spectra'. In: *Nature* 161, pp. 686–687.

Burges, Christopher J.C. (1998). 'A Tutorial on Support Vector Machines for Pattern Recognition'. In: *Data Mining and Knowledge Discovery* 2, pp. 121–167.

Buzsáki, György (2006). *Rhythms of the Brain.* Oxford University Press, Inc.

Carvalhaes, Claudio and J. Acacio de Barros (2015). 'The surface Laplacian technique in EEG: Theory and methods'. In: *International Journal of Psychophysiology* 97, pp. 174–188.

Hyvärinen, Aapo and Erkki Oja (2000). 'Independent Component Analysis: Algorithms and Applications'. In: *Neural Networks* 13, pp. 411–430.

Jasper, H.H. (1958). 'The ten-twenty electrode system of the International Federation'. In: *Electroencephalography and Clinical Neurophysiology* 10, pp. 367–380.

Kaggle Inc. (2015a). *Grasp-and-Lift EEG Detection.* URL: `https://www.kaggle.com/c/grasp-and-lift-eeg-detection/data` (visited on 21/10/2015).

Kai Keng, Ang et al. (2012). 'Filter bank common spatial pattern algorithm on BCI competition IV Datasets 2a and 2b'. In: *Frontiers in Neuroscience* 6.39.

King, Christine E. et al. (2015). 'The feasibility of a brain-computer interface functional electrical stimulation system for the restoration of overground walking after paraplegia'. In: *Journal of NeuroEngineering and Rehabilitation* 12, p. 80.

Kingma, Diederik P. and Jimmy Ba (2014). 'Adam: A Method for Stochastic Optimization'. In: *CoRR* abs/1412.6980. URL: `http://arxiv.org/abs/1412.6980`.

Kirmizi-Alsan, Elif et al. (2006). ‘Comparative analysis of event-related potentials during Go/NoGo and CPT: Decomposition of electrophysiological markers of response inhibition and sustained attention’. In: *Brain Research* 1104, pp. 114–128.

Kornhuber, Hans H. and Lüder Deecke (1965). ‘Hirnpotentialänderungen bei Willkürbewegungen und passiven Bewegungen des Menschen: Bereitschaftspotential und reafferente Potentiale’. In: *Pflügers Archiv* 284, pp. 1–17.

Lakatos, Peter et al. (2008). ‘Entrainment of neuronal oscillations as a mechanism of attentional selection’. In: *Science* 320, pp. 110–113.

LeCun, Yann, Corinna Cortes and Christopher J.C. Burges (1998). *The MNIST database of handwritten digits.* URL: http://yann.lecun.com/exdb/mnist (visited on 26/10/2015).

Lei, Xu et al. (2009). ‘Common Spatial Pattern Ensemble Classifier and Its Application in Brain-Computer Interface’. In: *Journal of Electronic Science and Technology of China* 7.1.

Luciw, Matthew D., Ewa Jarocka and Benoni B. Edin (2014). ‘Multi-channel EEG recordings during 3,936 grasp and lift trials with varying weight and friction’. In: *Scientific Data* 1, p. 140047.

Milo et al. (2010). *Nucl. Acids Res.* URL: http://bionumbers.hms.harvard.edu (visited on 23/10/2015).

Müller-Gerking, Johannes, Gert Pfurtscheller and Henrik Flyvbjerg (1999). ‘Designing optimal spatial filters for single-trial EEG classification in a movement task’. In: *Clinical Neurophysiology* 110, pp. 787–798.

Nielsen, Michael A. (2015). *Neural Networks and Deep Learning.* Determination Press.

Oppenheim, Alan V., Ronald W. Schafer and John R. Buck (1999). *Discrete-time Signal Processing (2nd ed.)* Upper Saddle River, NJ, USA: Prentice-Hall, Inc. ISBN: 0-13-754920-2.

Pearson, K. (1901). ‘On Lines and Planes of Closest Fit to Systems of Points in Space’. In: *Philosophical Magazine* 2, pp. 559–572.

Pfurtscheller, Gert et al. (2005). ‘EEG-Based Asynchronous BCI Controls Functional Electrical Stimulation in a Tetraplegic Patient’. In: *EURASIP Journal on Applied Signal Processing* 19, pp. 3152–3155.

Polich, John (2007). ‘Updating P300: An Integrative Theory of P3a and P3b’. In: *Clinical Neurophysiology* 118.10, pp. 2128–2148.

Rivet, Bertrand et al. (2009). ‘xDAWN Algorithm to Enhance Evoked Potentials: Application to Brain Computer Interface’. In: *IEEE Transactions on Biomedical Engineering* 56, pp. 2035–2043.

Sanei, Saeid and J. A. Chambers (2007). *EEG Signal Processing.* John Wiley & Sons Ltd.

Shalizi, Cosma Rohilla (2012). *Advanced Data Analysis from an Elementary Point of View.* notes for course 36-402, Advanced Data Analysis, at Carnegie Mellon University.

Tape, Thomas G. (2015). *The Area Under an ROC Curve.* URL: http://gim.unmc.edu/dxtests/roc3.htm (visited on 26/10/2015).

Teknomo, Kardi (2015). *Linear Discriminant Analysis Tutorial.* URL: http://people.revoledu.com/kardi/tutorial/LDA/LDA.html (visited on 19/10/2015).

Teplan, Michal. ‘Fundamentals of EEG Measurement’. In: *In: Measurement Science Review, Volume 2, Section 2*, p. 2002.

Welch, P. D. (1967). ‘The Use of Fast Fourier Transform for the Estimation of Power Spectra: A Method Based on Time Averaging Over Short, Modified Periodograms’. In: *IEEE Transactions on Audio Electroacoustics* AU-15, pp. 70–73.

Wolpaw, Jonathan R. and Dennis J. McFarland (2004). ‘Control of a two-dimensional movement signal by a noninvasive brain-computer interface in humans’. In: *PNAS, Proceedings of the National Academy of Sciences of the United States of America* 101, pp. 17849–17854.

Online image and video references

Brain Key (2012). *Brain Structure.* URL: http://www.brain-key.com/our%5C_brain.html (visited on 17/10/2015).

Brown, Larry (2014). *Schematic representation of a deep neural network, showing how more complex features are captured in deeper layers.* URL: http://devblogs.nvidia.com/parallelforall/accelerate-machine-learning-cudnn-deep-neural-network-library (visited on 20/10/2015).

Cheung, Joshua (2010). *(untitled).* URL: http://biomedicalengineering.yolasite.com/neurons.php (visited on 24/10/2015).

Deecke, Lüder (2005). *Bereitschaftspotential.* URL: https://commons.wikimedia.org/wiki/File:Bereitschaftspotenzial%5C_fig1.jpg (visited on 18/10/2015).

Gray (2009). *Central Sulcus.* URL: https://upload.wikimedia.org/wikipedia/commons/8/8e/Gray726%5C_central%5C_sulcus.svg (visited on 18/10/2015).

Hart, Marius 't (2008). *The 10-20 system.* URL: http://www.mariusthart.net/?e=200 (visited on 17/10/2015).

Kaggle Inc. (2015b). *(untitled).* URL: https://www.kaggle.com/c/grasp-and-lift-eeg-detection/data (visited on 21/10/2015).

Kent, James L. (2010). *Comparison of EEG Bands.* URL: http://psychedelic-information-theory.com/eeg-bands (visited on 17/10/2015).

Minnaar, Alex (2015). *(untitled).* URL: http://alexminnaar.com/implementing-the-distbelief-deep-neural-network-training-framework-with-akka.html (visited on 26/10/2015).

Modern, John (2014). *(untitled).* URL: http://somatosphere.net/2014/01/eeg.html (visited on 24/10/2015).

scikit-learn developers (2014). *Classifier comparison.* URL: http://scikit-learn.org/stable/auto%5C_examples/classification/plot%5C_classifier%5C_comparison.html (visited on 20/10/2015).

University of California, Irvine (2015). *A man whose legs had been paralyzed for five years walks along a 12-foot course using UCI-developed technology that lets the brain bypass the spinal cord to send messages to the legs.* URL: http://news.uci.edu/health/uci-brain-computer-interface-enables-paralyzed-man-to-walk (visited on 17/10/2015).

University of Houston (2015). *New UH research has demonstrated that an amputee can grasp with a bionic hand, powered only by his thoughts.* URL: http://www.uh.edu/news-events/stories/2015/March/0331BionicHand (visited on 17/10/2015).

Wadsworth Center (2010). *A P300 speller.* URL: https://www.youtube.com/watch?v=08GNE6OdNcs (visited on 18/10/2015).