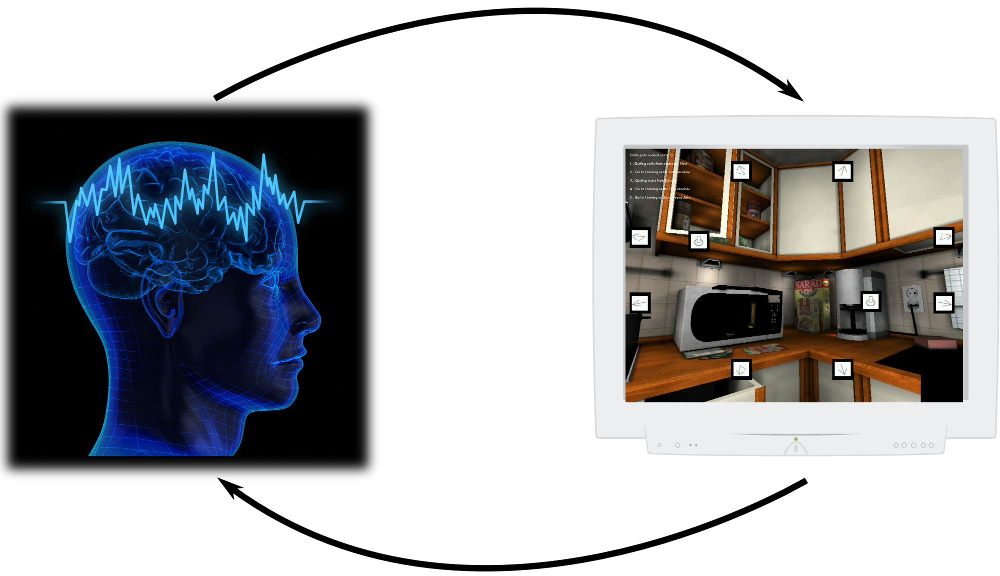
Exploiting Code-Modulating, Visually-Evoked Potentials for Fast and Flexible Control via Brain-Computer Interfaces

Hannes Riechmann, M.Sc.
Bielefeld

2015



Vom Promotionsausschuss der Technischen Fakultät der Universität Bielefeld
zur Erlangung des akademischen Grades
Doktor der Ingenieurwissenschaften (Dr.-Ing.)
genehmigte Dissertation
1. Gutachter: Prof. Dr. Helge Ritter
2. Gutachter: apl. Prof. Dr. Franz Kummert

Gedruckt auf alterungsbeständigem Papier ISO 9706

Abstract

Brain-Computer Interfaces (BCIs) promise to provide a communication channel for patients who have lost motor control. By directly measuring the activity of the user's brain, all traditional neuromuscular communication paths are circumvented. This provides a method of communication to patients who are unable to communicate by other means and hands-free communication for healthy users. But after two decades of intensive research, these devices are still quite slow for practical applications and the complexity of the applications is limited.

Currently, the most popular technique to measure the brain activity is the electroencephalogram (EEG), which is also used in this work.

Ideally a brain-computer interface is reliable, i.e. correctly detects the user's intention, fast, i.e. transmits enough information for fluent interaction and flexible, i. e. offers enough different commands to control the application conveniently. Constructing such a BCI is hard because the human brain is far from being understood and the activity of the brain can only be measured partially and indirectly with a low signal-to-noise ratio.

This dissertation presents a new BCI design for the control of robotic devices. Specifically, I show the first use of a code-modulating, Visually-Evoked Potential (cVEP) for a navigation and control task. The few studies, which have been done so far using cVEP, show high information-transfer rates and low latencies in spelling systems compared to other EEG-based BCIs. Still, there is no previous work on transferring the cVEP to applications beyond spelling, such as control of robotic devices.

This work, first, presents three exploratory experiments where I investigated stability of the cVEP potential in different conditions. Afterwards, I used the findings of these experiments to build a complete EEG-based BCI using cVEP for the control of a devices. For the evaluation subjects had to control a virtual agent in a kitchen task using the cVEP-BCI system. The evaluation shows that the system achieves an average accuracy above 80% and an average latency of about 2 s per command. All but one subjects were able to control the agent and complete the task.

Thus, this work shows for the first time that the cVEP potentials is indeed very flexible without loosing much speed or reliability. Although this system has only been evaluated in the lab it opens a new research track towards fast and flexible BCIs for diverse applications.

Contents

List of Figures

List of Tables

Acronyms

BAP Brain Activity Pattern.

BCI Brain-Computer Interface.

BMI Brain-Machine Interface.

BOLD Blood Oxygen Level Dependent.

CCA Canonical Correlation Analysis.

cVEP code-modulating, Visually-Evoked Potential.

EEG Electroencephalography.

ERD Event-Related Desynchronization.

ERP Event-Related Potential.

fMRI functional Magnetic Resonance Imagery.

ITR Information Transfer Rate.

LDA Linear Discriminant Analysis.

LIS Locked-in Syndrome.

MRT Magnetresonancetomography.

SCP Slow Cortical Potential.

SMR Sensorimotor Rhythms.

SSVEP Steady-State, Visually-Evoked Potential.

SVM Support Vector Machine.

VEP Visually-Evoked Potential.

1. Introduction

1.1. The Vision of BCIs

Cogito ergo sum, as stated by Descartes, is a fundamental element of modern philosophy. As long as one can think, one does exist, one is alive. Still, humans need a body and functional muscles to express their thoughts. Without muscles and neuromuscular control one is still caught in ones own thoughts. Therefore, the idea to communicate purely by thought (telepathy) or manipulate the environment by thought (telekineses) is a common theme in human culture, from religious texts to modern science-fiction literature.

Technical devices have brought humans quite close to these dreams, enabling live communication over thousands of miles or remote clinical operations where human motion is transfered to a robot on another continent. Still, all these devices need muscular activity to activate and control them. On the contrary, a *Brain-Computer Interface (BCI)* allows communication and manipulation by pure thought. For example, patients could control a service robot in a hospital by pure thoughts. Thus, BCIs enable us to really “live“ as long as we think, instead of only existing.

In todays research, a Brain-Computer Interface is a device or system which analyzes the activity of the human brain to provide a direct communication channel from the brain to a machine. Thus, a BCI circumvents all traditional neuromuscular paths, like speech or gestures. No muscular activity is needed. The only prerequisite is functional brain activity. The user volitionally alters his/her brain activity either by doing a specific mental task or with the help of external stimuli provided by the BCI system. The brain activity is analyzed using computer algorithms from machine learning and pattern recognition. The output of these algorithms is translated into commands matching the application.

Research on BCIs touches different disciplines from psychology and neuroscience over electrical engineering to computer science. Psychology and neuroscience can identify and describe patterns in the brain activity which are suitable for ”thought-control“. Electrical engineering builds devices to measure the brain activity. Finally, computer science, especially machine learning and pattern recognition is used to identify the intended pattern on-line, in real-time in the measured data.

The new communication channel provided by a BCI can be used for different applications. Three areas of applications are most important for this thesis:

First, BCIs are beneficial for patients suffering from *Locked-in Syndrome (LIS)*. Patients are fully awake and conscious, but loose complete control of their muscles, not only in the limbs but all muscles including facial and phonatory muscles. As muscular activity is needed for all kinds of communication, patients not only loose the ability to move, but also the ability to communicate. LIS is caused by different neurological diseases and some lesions which disrupt the neuromuscular pathways between brain and muscles, for example amyotrophic lateral sclerosis (ALS), well known because of Stephen Hawking who suffers from ALS. There a different stages of LIS. Most patients retain some control of the extraocular muscles [Ahmadi et al., 2010], allowing them limited communication using eye movements. Others are completely locked-in loosing even voluntary control of the eye movements. Giving these patients even limited means of communication does greatly improve their life [Münßinger et al., 2010]. Especially, as it is also very important that patients can communicate their wishes and problems with their doctors and care-givers [Smith and Delargy, 2005]. Today, in the USA alone, about 6000 persons per year are diagnosed with ALS, of these 90% decide to die instead of using ventilation. “The anticipated loss of the ability to communicate is a major factor in the decision to decline ventilation” [Sellers et al., 2010].

Second, there is a growing interest in using BCIs for *hands-free control* of a computer. For these applications the user is supposed to be healthy, but manual control is difficult or the hands are needed for other tasks, for example in space [Menon et al., 2009; Poli et al., 2013]. Control by BCI might not only provide higher accuracy and lower latency than the difficult manual control in space, it also eases collaborative control compared to joysticks for example.

Third, BCIs could also be used as new and challenging input modality for the control of *video games* [Nijholt et al., 2009; Lotte, 2011]. Especially in combination with virtual reality, BCIs could provide higher immersion than other modalities [Lotte, 2011]. Additionally, BCIs can provide new possibilities for interaction between players [Maby et al., 2012].

Other areas of research include BCIs for *rehabilitation* of patients and *passive* BCIs, which monitor the subjects mental state. Both types of BCIs are not strictly used for communication. This leads to quite different requirements. Therefore, those two areas are not considered within this work.

Regardless of the application, the purpose of any kind of communication is to convey information from one entity to another [Shannon, 2001]. A communication channel like a BCI merely transmits information, usually segmented into messages, from a sender to a receiver. Thus, three measures are very common to evaluate BCIs. Namely: First, the amount of information it can transfer in a fixed period of time. Second, the probability of failure. Third, the number of different available messages. Of course there are numerous other measures, but in the following I will mainly relate to these three measures, as they are the one investigated the most in research and the most suitable ones to illustrate the problems of current BCIs. A given application of communication has certain requirements on these three criteria [Tonet et al., 2008]. For example, when communicating with another person, this person might loose interest if communication is too slow, i.e. too little information can be transfered in a given time. Or, when controlling a wheelchair transmitting "Drive left" erroneously, when the user wanted to drive to the right, might lead to a crash. In contrast, a wrong letter in a transmitted sentence might not be a problem as the sentence is still understood.

A visionary, perfect BCI would meet the communication requirements of the application and, additionally, be affordable and usable by non-technical personnel.

This perfect BCI has not yet been realized. So, I proceed by analyzing the gaps between the current state of the research and this vision, including the major challenges to close those gaps. Afterwards, I introduce the contributions of this thesis and how these contributions bring BCI research closer to the vision behind BCIs. Finally, I outline the structure of this thesis.

1.2. Current Challenges Towards this Vision

Despite being valuable in different areas and under research for decades, the perfect brain-computer interface has not yet been realized. Even worse, BCIs still have not moved from the research labs to hospitals and homes. In this section, I will briefly investigate what the obstacles are and why they are difficult to overcome.

Current BCI system often do not really satisfy the *communication requirements* of their intended application well, even in the idealized lab scenarios where the studies are performed. Before they can be transferred to hospitals and homes, communication speed and reliability (i.e. accuracy in the correct detection of the users intention) have to be enhanced [McFarland and Wolpaw, 2011; Tonet et al., 2008]. This is a very difficult task as BCI researchers deal with a noisy measurement of the human brain and need to gather and analyze data about the internal state of this very complex system. The brain activity one uses to infer the user's intention is hidden in a lot of internal noise, i.e. other brain activity, and external noise, e.g. electrical fields of near electrical devices when using Electroencephalography (EEG). Additionally, in many cases it is difficult to model the target signals, as the structure of the target signal is often unknown and subject- and session-specific. Furthermore, researchers face a traditional dilemma when trying to improve speed and reliability. If one wants high reliability one needs to analyze the brain activity of the user over a longer period, loosing speed. This hold regardless of the kind of brain activity which is used [Zhao et al., 2009]. To improve both, one needs better signals or better recognition algorithms.

Communication speed and reliability are not the only problems though. Depending on the type of brain activity, which is analyzed, the number of distinct commands, which are offered to the user, is also often limited. 2-3 for some systems [Galán et al., 2008], 4-8 for others [Riechmann et al., 2011]. Usually, the number of distinct commands is limited as this eases the recognition of the correct command. For many applications this leads to quite complex selection procedures, for example choosing a letter by binary choice. There, one has to narrow down the group of possible letters to one letter by several binary choices. This not only lowers speed, but also feels unnatural to the user. For other applications this restricts the users options. For example a spelling application with word completion forces to use words from a given dictionary and a semi-autonomous wheelchair which users can only command to turn left or right at the next crossing does not allow to influence the path through the room.

Additionally, many BCIs today are optimized for a specific application. For example there are numerous studies on optimizing the user interface for spelling applications to evoke better discriminable brain activity. This close coupling between interface and application greatly restricts applicability of BCI systems. Imagine one had to use a different keyboard for each task such as browsing or typing a letter. Ideally, one would have one or two BCI systems with different user interfaces for different applications. Still, more general and flexible BCIs often achieve lower speed and reliability, thus receive less interest.

Furthermore, current approaches need tedious preparations, user training and often expensive hardware and equipment. All three issues apply both to real-world application of BCI and research. The preparation time mainly consists of applying the sensor to the user. For example, to study or use an EEG-based BCI the EEG sensors have to be attached to the user's head. In laboratory research this consumes valuable experimentation time, in real-world application it is expensive and reduces independence of the user because some helper personnel is needed. Depending on the used brain activity, several training sessions for each user may also be necessary to allow them to successfully evoke the brain activity which is to be detected. This is less of a problem for real-world applications, but increases research effort, if, for example, ten training sessions are necessary and ten users take part, then 100 sessions have to be conducted before evaluable data can be recorded at all. For most real-world applications the necessary sensors are also still quite expensive, a medical EEG device costs somewhere between € 10,000 and € 250,000, a Magnetresonancetomography (MRT) device some magnitudes more.

1.3. Contributions of this Thesis

Of the aforementioned issues this thesis mainly concentrates on four: *speed, reliability, number of commands and flexibility.* Improving speed means here to increase the amount of successfully transmitted information from the BCI user to the target application in a given time. As stated in the previous section it is closely linked to reliability, where reliability means the percentage of successful transmissions. Both cannot be considered separately. Additionally, the system will offer more commands than comparable systems, allowing more fine-grained and more natural control for the user. To some extent this is also linked to the first two. On the one hand, more distinct commands means one needs less transmissions in many situations (e.g.: one choice out of eight instead three choice out of two). On the other hand, more distinct commands increases the difficulty of the classification problem. Finally, the system is more flexible than most BCI systems so far, allowing to easily adapt it to different applications.

Over the years several different approaches have been studied to evoke brain activity which can be discriminated and used as input for the BCI. For the remainder of this thesis they will be called *Brain Activity Pattern (BAP)*. One example is the imagination of movement of the hands. All the approaches have different advantages and limitations. I will describe them in detail in section 2.3.

In the last few years, some studies investigated the *code-modulating, Visually-Evoked Potential (cVEP)* [Bin et al., 2011; Spüler et al., 2012a]. For this BAP the activity of the visual cortex is measured. A set of visual stimuli is presented on the screen. To evoke discriminable brain activity all stimuli flicker simultaneously, each with a different on/off pattern. The user gazes at a specific stimulus to select the corresponding command. Then, a brain wave can be detected using EEG which corresponds to the stimulus the subject gazes at.

Previous studies have shown that high information transfer rates (up to > 100 bit/minute) with high reliability ($> 90\%$ accuracy) are possible with cVEP-based BCIs. The communication speed which was reached in those studies outperforms all other approaches for surface-EEG BCIs [Spüler et al., 2012a]. As a result, cVEP-BCIs are named as a promising new research direction in "Brain Computer Interfaces, Principles and Practice" [Wolpaw and Wolpaw, 2012, p. 241]. So far, however, only spelling tasks have been tried, using a fixed number of targets at fixed positions on a black screen.

To develop a system which improves the four above-mentioned issues, I investigate the relation between the signal strength of the code-modulating, Visually-Evoked Potential (cVEP) potential and the characteristics of the stimulus presentation. The goal is to create a BCI system with the speed and reliability of the cVEP which gives the user a high number of commands and is flexibly usable for different applications, especially for applications navigating and controlling some robotic device.

This leads to the over-arching research question: **How to develop and evaluate a flexible BCI system for a wide range of applications while preserving the speed and reliability of the cVEP potential known from spelling systems?**

To investigate this question, I divide it into four sub-questions:

1. **Investigate the cVEP pattern in general.**

   So far, evidence about the properties of cVEP-BCIs is limited, in terms of studies published and subjects tested. New studies and new subjects will allow me to provide further evidence about the advantages of cVEP-BCIs.

2. **Evaluate different designs of the cVEP-BCI in general and specifically the user interface. Investigate the strength of the cortex response and the resulting accuracy for these different designs.**

   As a visually-evoked potential (VEP), the signal strength of the cVEP highly depends on the exact characteristics of the stimulus presentation and, more generally, the complete user interface. While there is some previous work on other VEPs about such influences, there is no previous work about this for cVEP-BCIs. I will investigate parameters such as size, color and shape of the stimuli and possible effects of a distracting background.

   Additionally, I investigate influences of the specific code which is used to generate the pattern for the stimuli and the users' mental state on the signal strength.

3. **Implement and evaluate a mechanism to balance speed and accuracy of the cVEP-BCI.**

   Repetitions are a common approach to improve the accuracy in BCIs. This allows to trade speed for accuracy. Essentially, one wants to repeat as much as necessary to achieve a reliable classification and as little as possible to achieve the lowest possible latency. The issue is further complicated by the fact that the signal strength varies during a session, which means that the ideal number of repetitions also varies constantly. So, I design and evaluate a mechanism to on-line asses the current signal strength and, based on that, dynamically adapt the number of repetitions.

4. **Implement and evaluate a complete cVEP-BCI for navigation and control.** Finally, a complete cVEP-BCI for navigation and control is to be implemented under consideration of the results of the previous questions. This system shows the feasibility of cVEP-BCIs for command and control of devices. It is also thoroughly evaluated in a practical study with healthy subjects and compared with existing cVEP-BCIs for spelling and other BCI approaches for control.

Four studies have been conducted to answer those four questions. Still, there is no one-to-one correspondence between questions and studies, instead each study provides results which form the answers for the questions together.

The first two studies are off-line, preliminary studies. In the first I tested the effect of color on the signal strength by comparing the classification accuracy of standard black/white flickering and a red/green flickering. To use different colors would later allow to embed the stimuli more naturally into a user interface. Additionally, I investigated the feasibility of a hierarchical flickering pattern. Here, hierarchical means that the stimuli are divided into groups. In the first half of each flickering sequence only the groups flicker differently and the items within the groups flicker to the same pattern, in the second it is the other way round. This would allow to detect the group of the attended stimulus very early.

In the first study, the stimuli were quite large and there were additional, so called non-target stimuli, which the subject never attends to, but which are intended to increase signal strength. As a major drawback the stimuli occupied nearly all the space on the screen.

So, in the second study I removed the non-target stimuli and reduced the stimuli in size, creating space to later show the scene. With these small stimuli I again compared two conditions: One with standard black background and one with a fixed image of our lab previously recorded with a robot camera.

The third study was an on-line spelling experiment to test our setting in an on-line run and to evaluate our mechanism for dynamic repetitions. As no BCI approach is absolutely reliable some misinterpretation of the brain activity is unavoidable. One option to lower the probability of misinterpretation is to repeat the decision multiple times. Of course this lowers communication speed. The idea of dynamic repetitions is to analyze the data, not only to infer the user's intention, but also to analyze how confident this inference is, how probable a misinterpretation is. When this can be derived well from the data, then one can repeat only decisions when it was quite probable that the inferred command does not match the real one. This has already been examined for different BAPs, but not yet for cVEP.

Finally, in the fourth study I evaluated the complete, flexible cVEP-BCI controlling a virtual agent. As an example application I built a system where the user navigates and controls a virtual avatar. The virtual avatar was controlled in a first person perspective on a screen. The stimuli for the cVEP were overlaid over the scene, i.e. the scene was drawn first and then the stimuli were drawn on top of the scene. The number and positions of the stimuli varied depending on the scene. As an example scenario a virtual kitchen was chosen. Transferred to a real humanoid robot this would allow the robot to act as a kind of a surrogate body, as in [Do et al., 2013; Hachmeister et al., 2011]. For a real robot the BCI system would be the same instead of a virtual environment one would record the robots real environment and display it on-screen for the user.

1.4. Structure of the Thesis

Here, the remainder of this thesis is outlined.

- **Chapter 2 Brain-Computer Interfaces:** Chapter 2 gives a general introduction into the current research on BCIs providing the necessary background for the rest of the thesis. This chapter introduces the terms, presents different recording techniques and BAPs and shows different BCI applications.
- **Chapter 3 Visually-Evoked Potentials:** In Chapter 3 I give the necessary context to understand and asses my research. Thus, after a general introduction in chapter 2, chapter 3 gives a more detailed introduction for the techniques used in my experiments, namely visually-evoked potentials (VEPs). VEPs are a group of brain activity patterns which are used in BCIs. I will describe their neural foundations, characteristics and different variants, especially the cVEP I use in my experiments.
- **Chapter 4 Prerequisites for the cVEP-BCI:** Chapter 4 introduces in detail some methods with are central for my experiments and usually used in more than one. This includes some signal processing methods, like Canonical Correlation Analyses and Support Vector Machines as well as the software framework and some common parts of the experimental setup.
- **Chapter 5 Influence of Codebook and Color in cVEP-BCIs:** This chapter describes the study where I investigated the feasibility of using a hierarchical codebook and non-standard colors for cVEP-BCIs. A hierarchical codebook splits the stimuli into groups and the codebook into a first part that encodes the group and a second that encodes a specific item within the group. This allows to very rapidly detect the group. One could, for example start a robotic movement roughly into the correct directions already after the first half of the epoch. Additionally, I investigate how red/green flickering stimuli compare to traditional black/white ones in terms of signal strength and user comfort. The results of this study were partly published in [Riechmann et al., 2013].
- **Chapter 6 Influence of Stimulus and Background Characteristics in cVEP-BCIs:** In the second experiment, the user interface is still static, but several stimulus characteristics are already adapted for the need of a dynamic BCI for navigation and control. First, the stimuli are

much smaller and rearranged in such way that one has enough space on the screen to later display the scene. Second, I evaluate if the signal strength decreases, when using a lively background. In particular, a background which resembles a scene in a navigation and control scenario.

- **Chapter 7 An On-line cVEP Speller with Dynamic Repetitions:** This experiment serves three purposes. First, testing my BCI system in an on-line setting. Second, evaluating dynamically changing stimuli. Based on previous research one can assume that the shape of the stimulus does not influence the signal strength. This is to be evaluated in this experiment. Third, testing of a dynamic repetition mechanism for cVEP-BCIs. As stated above this allows to dynamically balance reliability and speed.

- **Chapter 8 An On-line cVEP-BCI for Fast and Flexible Control:** This chapter describes and discusses the final experiment. For this final experiment, I built a BCI system designed for the control of a virtual agent based on the cVEP potential. This system provides a flexible design, easily adaptable to different control tasks, a high number of commands (up to 16 for now) and a low latency of about 2 seconds. Using dynamic repetitions of epochs the system provides asynchronous control and adapts to the signal strength of each user.

  Based on a cVEP-BCI the system enables the user to move the virtual agent freely within a virtual environment, where the agent can perform some actions. As example scenario I use a virtual kitchen with actions like getting coffee from the cupboard or switching on the coffee-machine. The BCI gives fine-grained control where desirable (e.g. for navigation) and semi-autonomous actions, like "Switch on coffee-machine", where appropriate. The results of this experiment are partly published in [Riechmann et al., 2014].

- **Chapter 9: Discussion** After each study has been presented separately, this chapter will summarize all four studies, relate the results from the different studies to the four research questions and embed the results into the research context, along with a comparison with other state-of-the-art BCIs and a discussion about advantages, limitations and prospects of my approach. Additionally, I discuss directions for future work. Especially, how to further improve speed and reliability and which further applications of the cVEP in general and my final cVEP-BCI system specifically could be interesting.

2. Brain-Computer Interfaces

This section gives a thorough introduction to the topic of *Brain-Computer Interfaces*. The term itself already says that a BCI is an interface between brain and computer (or more general: a computer-controlled device). While there are many interfaces for communication between humans and technical devices, *a BCI directly connects* a brain with a computer, bypassing all traditional neuromuscular paths. In contrast to BCIs, all other means of communication between humans or from a human to a machine, like speech, gestures, keyboard rely on the correct and often very complex activation of muscles.

As with almost any form of communication, the direct communication from a brain to a computer needs several translation steps. Commonly, a BCI consists of around four parts representing different translation steps [Wolpaw et al., 2002]. This scheme is depicted in Figure 2.1.

At first, the brain activity has to be digitalized, using some kind of sensor and *digital data acquisition*. After that, some *signal processing* is needed. Often this is divided into two steps: feature extraction and classification. In the feature extraction step some meaningful features out of the high-dimensional data are derived. In the classification step these features are classified. Next, some *output device* is needed. On one hand, this output device is the effector of the communication (might be a computer screen to communicate with other people or a robot that is controlled). On the other hand, this output device needs to provide some kind of feedback. This is essential and leads to two systems, which have *to adapt to each other*, the user's brain and the signal processing system. Last, a *communication protocol* is needed which defines the process of communication.

In the next subsection, I give a brief taxonomy of BCIs, introducing the necessary terms for the rest of this chapter. After that, the different parts of a BCI are discussed in detail. I start with the different recording techniques, with a focus on EEG. Then, the most common protocols for BCIs are discussed and, last, i give an overview over the different applications which are currently studied by researchers. No overview over signal processing techniques will be given in this section. This is part of chapter 3, where I will focus on the specific brain activity pattern used in my research.

2.1. Taxonomy

BCIs can be clustered according to several properties, for example depending on the neural activity they use to infer the user's intention. Here, I will introduce general properties by which researchers cluster BCIs. The different data recording approaches and neural activities will then be presented and compared in the next section. Examples for the different classes of BCI will also be given in the next sections.

i. Voluntary vs non-voluntary On a very general level BCIs can be divided into voluntary and non-voluntary approaches [Pasqualotto et al., 2012]. In the first case the user is able to elicit the neural activity by himself or can be trained to do so. Usually, the user learns different neural activities and the BCI infers which if any the user elicited at a specific time. In the non-voluntary case the neural activity is elicited by external stimuli and the user conveys his intention by attending to one specific out of a set of stimuli. The BCI infers which was the attended stimulus.

ii. Dependent vs independent Non-voluntary BCIs can be further divided into dependent and independent BCIs [Wolpaw et al., 2002]. Dependent means that some muscular activity is needed to attend to the stimulus and elicit the neural activity. Usually this means that the stimulus needs to be fixated with the gaze, which means the user needs oculomotor control. On the first sight, this seems to undermine the central idea of BCIs, to communicate without movement. Still, there are many patients who cannot move their limbs and phonatory muscles at all, but still retain oculomotor control

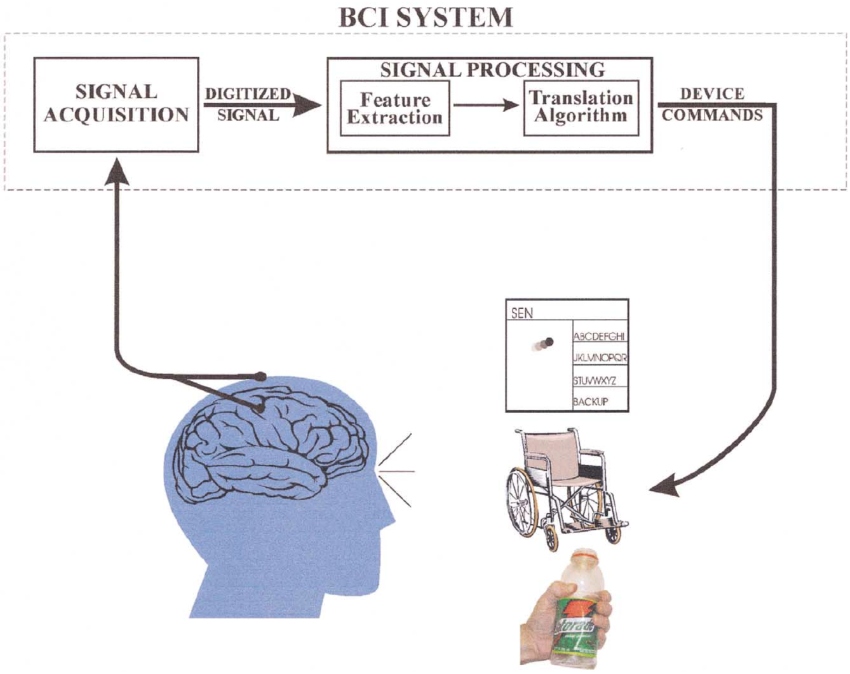


Figure 2.1.: General BCI scheme from [Wolpaw et al., 2002]. Basic design of any BCI system. Some recording device measures the neuronal activity of the user. This data is digitized and analyzed. In a first step, useful features are derived from the high-dimensional, noisy data. In a second step this data is translated into commands for an output device. The user observes the reaction of the output device, thus closing the loop.

[Ahmadi et al., 2010]. Thus, it is worth to do research on these BCIs. Of course, one should always check whether an eye-tracker could be an alternative for a specific patient.

iii. Synchronous vs asynchronous This distinction refers to the communication protocol [Nooh et al., 2011]. Can information be transmitted continuously (asynchronous BCI) or only at specific points in time (synchronous BCI). An asynchronous BCI gives the users more freedom, but it is often more difficult to realize. Usually an asynchronous BCI is realized by always analyzing the data and in addition to the control commands adding a “No command” condition [Finke et al., 2012].

iv. Active vs passive So far, I considered BCI as a means of active communication, i.e. the user actively wants to convey information. Recently, BCIs have also been used for passive communication, i.e. the interface infers passively information about the state of the user, e.g. his alertness [Zander and Kothe, 2011]. These passive BCIs system do not aim at replacing traditional communication channels such as speech, but to provide additional information which cannot by transmitted by other, active communication channels, as the user cannot asses this information himself. This makes BCIs an interesting technique for healthy users. In [Rötting et al., 2009] a passive BCI monitored users when playing a simple computer game. The BCI detected a so-called error potential, when the user was sub-consciously aware that he had made a mistake and the game reacted accordingly. In [Reissland and Zander, 2009] a BCI was used to recognize when a user bluffed in a dice game showing that complex mental states can be detected in the EEG. In [Schmidt et al., 2009] drivers drowsiness was assessed by EEG and automated data analysis during driving a car on a public motorway.

As passive BCIs have quite different technical requirements and potential applications compared to active BCIs, I will not further consider them in this thesis.

v. Brain-Machine Interfaces BCIs are often investigated as a method of controlling some technical device. These BCIs are intended to help patients as well as provide a means of hands-free control for healthy users. The controlled devices range from TVs and music players to wheelchairs and robotic devices. These systems are sometimes also called Brain-Machine Interface (BMI) in the literature. I will usually refer to them as BCIs for navigation and control.

vi. Hybrid BCIs To augment communication speed and offer more different options, it has also been proposed to construct hybrid BCIs. According to [Pfurtscheller et al., 2010], “a typical hybrid BCI is [...] composed of one BCI and another system (which might be another BCI)“. Often two different neural activities are combined [Pfurtscheller et al., 2010; Riechmann et al., 2011]. Sometimes, also two different data recording techniques are combined [Fazli et al., 2012]. Finally, some studies have been done to use non-BCI information such as residual muscular activity as complementary input [Carlson et al., 2013].

2.2. Data Acquisition

As a first step to interpret the user's brain activity and to infer the correct information, the brain activity has to be measured. There are several techniques to do so, which are commonly grouped into *invasive* and *non-invasive* approaches [Pfurtscheller and Brunner, 2008]. *Invasive* techniques place the sensors beneath the scalp, right on or within the brain. *Non-invasive* techniques place them on the scalp. Here, I will deal with invasive approaches only briefly and then show different non-invasive ones. The main focus is on EEG, as this is used by most BCI studies and is also used for my experiments.

2.2.1. Invasive Methods

Invasive data acquisition methods for BCI research have the advantage of a better signal-to-ratio as the data is acquired close to its origin. The disadvantage is that the sensors have to be placed within the scalp via surgery. This imposes financial and ethical problems for research on this type of interfaces. Therefore, many studies use trained monkeys or patients who have to undergo scalp surgery as part of their diagnosis or therapy, mostly epilepsy patients [Schalk and Leuthardt, 2011].

There are different approaches, mostly differing on the type of sensor and the magnitude of neurons recorded. Older studies often recorded single neurons or groups of neurons [Aggarwal et al., 2008; Mollazadeh et al., 2008; Vargas-Irwin et al., 2010]. Instead, todays most popular approach, Electrocorticography (ECog) uses electrodes on the cortex which record larger population of neurons, but still offer much higher spatial resolution than surface EEG [Fifer et al., 2011]. ECog is often referred to as intracranial EEG (iEEG), which it technically is. Using ECog the electrical fields emitted by the neurons are measured. Compared to EEG, ECog offers higher spatial resolution (in the range of millimeters instead of several centimeters), higher amplitudes and less sensitivity to artifacts from not-brain-related electrical fields produced by movement from the subject or electrical devices in the surroundings [Schalk and Leuthardt, 2011]. Another important aspect is the availability of higher frequencies. Above 40 Hz EEG amplitudes are very small and few studies examined the information contained in those frequencies [Schalk and Leuthardt, 2011], using ECog frequencies up to 500 Hz are accessible and can be analyzed [Staba et al., 2002].

Using these frequencies allows to built BCIs for applications which are not possible with non-invasive measures. In [Fifer et al., 2011] the authors derived the parameters of a grasping movement from the ECog with high precision. A twelve- year-old, human boy was instructed to do reach-and-grasp movements towards various objects with different sizes and shapes. A trial began when the subject started to move from a fixed, neutral position towards an object and ended when the fingertips touched the object. During the movement the ECog was recorded. Using Generalized Linear Models the grasp aperture was derived from the data and compared with the real grasp aperture. They could show that it is possible to reconstruct the grasp aperture well. This was evaluated by computing the Pearson correlation between the real aperture and the reconstruction. The best frequency bands are in the high gamma range (> 70 Hz), using only data from either high gamma 1 (70-100) or high gamma 2: they resulted in a Pearson correlation of 0.74 and 0.75 respectively, using all frequencies and all electrodes they reached 0.8.

While being interesting for prosthesis control, invasive techniques could not show great advantages in traditional BCI protocols for spelling, when compared with sensors on the scalp [Sellers, 2013]

In remains to be seen, how far these methods work with patients who have no arm/hand control. BCI systems using ECog for motor control usually derive the intended movement directly from the sensorimotor cortex. Depending on the reason for the impairment the sensorimotor cortex of the patient might not generate usable activity. In 2012, Yanagisawa et al. compared performance in controlling a hand prostheses using ECog between healthy user's and paralyzed patients who had no or little hand control [Yanagisawa et al., 2012]. Three out of five healthy patients were able to control the prosthetic hand, but only one out of four mildly impaired patients and no one out of three severely impaired patients.

So, while invasive methods are an interesting field of research, their financial and ethical problems will restrict them to severely impaired patients. There, some promising results have been found, but research is only at the beginning and it remains to be seen whether invasive approaches can be helpful for this patient group.

2.2.2. Functional Magnetic Resonance Imagery

functional Magnetic Resonance Imagery (fMRI) is another method for measuring human brain activity. Using strong, external magnetic fields, the blood oxygen level is measured. This blood oxygen level is coupled with neuronal activity. Neuronal activity needs oxygen hence on a time-scale of seconds the area where neurons are highly active becomes more saturated with oxygen, leading to a higher blood oxygen level which can be measured [Huettel et al., 2009]. This is called a *Blood Oxygen Level Dependent (BOLD)* measurement. As the BOLD response is slow, the temporal resolution of fMRI is relatively bad. The spatial resolution depends on the strength on the magnetic field and can be pushed quite low. When using real-time fMRI the spatial solution is usually in the range of a few millimeters [Sitaram et al., 2008].

Research on using fMRI for BCIs has been relatively scarce, because of the low temporal resolution and the costs of fMRI scanners. Currently, most studies do not focus on conveying information from the user to other people or devices, but to convey information of his brain activity to the user himself. Thus, providing highly localized, immediate (in terms of a few seconds) feedback on neural activity. This can be used to train users to actively modulate brain regions which normally cannot be activated voluntarily. This is both helpful for research as well as rehabilitation [Sitaram et al., 2008]. A special case of rehabilitation is described in [Liberati and Birbaumer, 2012], where the authors vote for using fMRI-BCIs to reactivate goal-directed thinking in Minimally Conscious State patients.

2.2.3. Functional Near-Infrared Spectroscopy

functional Near-Infrared Spectroscopy (fNIRS) uses light in the near-infrared range (650 to 900 nm) to measure cerebral blood flow [Obrig and Villringer, 2003]. This means that, like fMRI, fNIRS is a BOLD measure, offering a temporal resolution in the range of seconds. The sensors are affordable, offer a reasonable spatial resolution and are very simple and fast to mount. Therefore, they gained some interest for BCI research [Sitaram et al., 2007; Coyle et al., 2007; Luu and Chau, 2009]. In [Sitaram et al., 2007] a fNIRS-based system was able to identify whether subjects imagined left or right hand motor imagery with similar speed and accuracy as EEG-based system. In [Luu and Chau, 2009] user's had to mentally evaluate two possible drinks and the fNIRS successfully detected the preference with 80% accuracy. As a fNIRS system can be mounted very easily, e.g. via a headset [Luu and Chau, 2009], it is particularly interesting as an additional measure in addition to, for example, EEG or as a measurement technique for passive BCIs [Zander and Kothe, 2011].

For this work it is not suitable as communication speed is the central aspect of this thesis and this is where fNIRS-based systems are not as good as EEG-based systems.

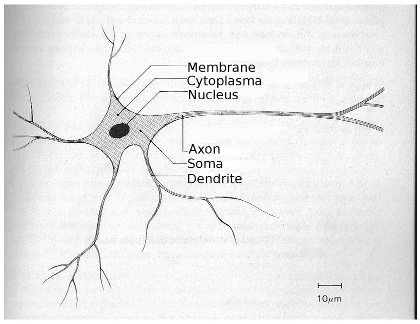


Figure 2.2: Schema of a neuron. The neuron receives current from its input neurons through synapses which end near its soma or its dendrites. When the accumulated current at the axon reaches the threshold of the neuron, an action potential is generated and current flows through this neuron's axon to it's output neurons. Figure from [Schmidt and Dudel, 1987, p. 1]. Labels translated.

2.2.4. Electroencephalography

EEG is by far the most popular data recording technique for BCI [Nicolas-Alonso and Gomez-Gil, 2012]. Its relative cheapness eases research, but there are also several reasons in handling and data quality that make EEG as popular as it is. As EEG is also the data recording technique used in my studies I will give a more thorough introduction here. In short, to record an EEG, one places electrodes on the scalp which measure electrical current between themselves and a reference electrode. The neurons in the brain produce electrical fields. These fields get measured at the electrodes and allow to derive information about the brain activity. Of course, the electrical fields from the neurons are very small and the electrodes pick up all kinds of other, often much stronger, electrical fields from the surroundings, which have to be filtered somehow. This results in a very poor signal-to-noise ratio compared to other recording techniques [Nicolas-Alonso and Gomez-Gil, 2012].

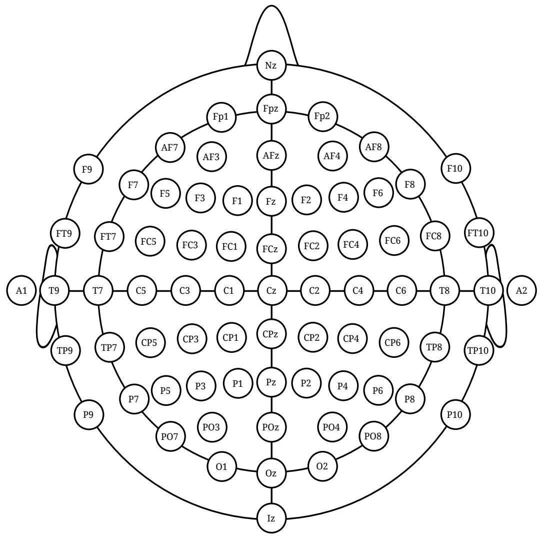
In the following, I will talk about the neuronal prerequisites which allow us to record brain activity via electrical fields at all. Then, I will describe the recording steps necessary to obtain a reasonable signal-to-noise ratio and present the most prominent rhythms found in the EEG data, along with some typical artifacts. Additionally, I will present some advantages and limitations of surface EEG and, finally, show some new sensors which try to overcome some of the limitations.

2.2.4.1. Electrophysiological basis

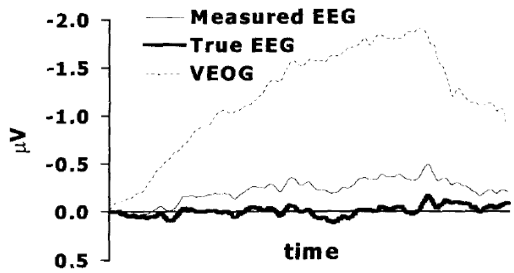
Neurons are the basic computation units of brains [Schmidt and Dudel, 1987], one neuron is schematically depicted in Figure 2.2. Neurons communicate via the exchange of charged ions [Hodgkin, 1971]. When a neuron is excited by other neurons via action potentials, it releases charged ions at its own axon (primary current) and, after that, attracts charged ions from the outside to conserve electric charges (secondary current). This flow of current generates electrical fields. They are very weak and not detectable outside the skull, but if huge number of neurons (thousands to millions) fire simultaneously, the generated electrical fields can sum up [Nunez and Srinivasan, 2006, p. 59]. The structure of the cortex is ideal for this to happen, as "Macrocolumns of tens of thousands of synchronously activated large pyramidal cortical neurons" [Baillet et al., 2001] do produce fields which sum up to magnitudes which can be picked up in the EEG and these structure are at the outer rim of the brain [Teplan, 2002], both favorable to be picked up by surface electrodes and likely to contain interesting information as these brain areas are evolutionary young [Nunez and Srinivasan, 2006, p. 59].

2.2.4.2. Recording basics

To do a EEG recording first the electrodes have to be placed on the scalp, sometimes they are glued, more often they are kept in place by a cap [Baillet et al., 2001]. Electrodes are typically made of silver-chloride (AgCl) [Teplan, 2002]. For wet electrodes one needs to apply some conductive gel or fluid to the position where the electrode is mounted. This lowers conductance. To further lower conductance one typically cleans the mounting site with alcohol before applying the gel. Dry electrodes do not need this time-consuming procedure, they are considered later in this section. One considers only a resistance of below 10 kΩ as applicable for passive electrodes in a BCI setting [Usakli, 2010]. If this resistance is not reached one has to clean the recording site again and apply the gel more thoroughly.



(a) Electrode positions and names of the 10-20 system.
License: Creative Commons.
Author: Marius 't Hart



(b) Example of ocular artifacts in EEG measurements from [Croft and Barry, 2000]. In this example the ocular artifacts enhances the negative potential, i.e. the measured effect is much stronger in the measured as in the real cortical activity.

Figure 2.3.

The procedure of preparing the site and attaching the electrodes needs some experience and is one of the reasons it is difficult to move from the scientific laboratory to patient homes.

For a common placement of electrodes the standardized, international 10-20 system is used [Teplan, 2002; Jasper, 1958]. The head is divided according to distances from prominent skull features. From front to back the skull is measured from the upper point of the nose (nasion) to the occipital bone at the back (inion). From left to right the distance is measured between the two ears (more precisely the preauricular points). These two distances are divided into 10% or 20% portions. The electrode positions are named by one or two letters indicating the front/back position and a number or small z for their left/right position. Odd numbers, are for the left hemisphere, even for the right and z for the central positions. The standard positions are shown in Figure 2.3a. Additional positions can be inserted as needed, for example the position in the middle between Fz and Cz is called FCz and the position in the middle of F3 and F7 is called F5.

For my experiments I used a standard medical EEG amplifier, the gTec gUSBAmp with wet, passive electrodes. The gUSBAmp supports up to 16 electrodes and delivers a relatively good signal-to-noise ratio. This amplifier is used in numerous surface EEG studies on BCIs, e.g. [Guger et al., 2009; Pires et al., 2011; Raif et al., 2013].

2.2.4.3. Rhythms in human EEG

“The normal EEG is a tapestry of overlapping, waxing-and-waning rhythms generated by different cortical regions.” [MacDonald, 2003]. Here, I briefly describe the most prominent rhythms found in the EEG of healthy adults. Typically, the normal EEG is categorized into four or five rhythms [Teplan, 2002; MacDonald, 2003], which are shown in Table 2.1. For BCI research α, μ and β are most interesting.

The α-rhythm was the first rhythm which was found by Hans Berger in his first EEG experiments. It is the most prominent rhythm in the adult human EEG and usually attributed to relaxation during wakefulness. Amplitude rises in eye-closed situation. It is strongest over the occipital cortex. For BCIs it is usually used in passive BCIs to monitor drowsiness [Hung et al., 2010], but it can be modulated volitionally after some feedback training and has been used as additional input for video-games, see section 2.4.2 p. 20ff.

Table 2.1.: Standard rhythms found in the EEG of healthy human adults

Name	Frequency	Description
β	> 13	activity, in particular in sensorimotor cortex [Pfurtscheller and Lopes da Silva, 1999]
α	8-13	awake relaxation
μ	8-13	“”idling” motor-sensory cortex in wakefulness“ [MacDonald, 2003]
θ	4-8	sleep
δ	0.5-4	deep sleep

The μ-rhythm was found 1952 by Henri Gastaut and is located over the sensorimotor cortex. It indicates an ”idling“ sensorimotor cortex [MacDonald, 2003] and, depending which limbs are being moved or not, its amplitude can vary substantially between the hemispheres. It is very popular in BCIs as it can be easily modulated volitionally. BCIs based on it are described in subsection 2.3 p. 18ff.

The β rhythms are usually correlated with activity. They are used often in BCIs based on sensorimotor rhythms in combination with the μ-rhythm [Pfurtscheller and Lopes da Silva, 1999].

Finally, the θ and δ rhythms are most prominent during sleep.

Additional rhythms can be caused by additional stimuli. These are usually located at the regions where the external stimulus is processed, for example the visual cortex for visual stimuli.

2.2.4.4. Artifacts

”According to the glossary of the International Federation of Clinical Neurophysiology (IFCN) the term *artifact* is described as ”any potential difference due to an extracerebral source, recorded in EEG tracings”“ [Klass, 1995]. As the cerebral electrical fields are very weak, these artifacts are often magnitudes stronger. Artifacts can be categorized in ”Subject related“ and ”Technical“ [Teplan, 2002], sometimes the technical category is further divided into ”technological“ (arising from the electrodes, the amplifier, cables and so on) and ”external“ (power lines and so on) [Klass, 1995].

An overview of the most common artifact sources is given in [Teplan, 2002] as follows (some additional remarks by me):

Patient related:

- any minor body movements
- EMG (electromyography, electric fields of muscles, especially the jaw muscles)
- ECG (pulse, pace-maker)
- eye movements
- sweating

Technical:

- 50/60 Hz (power lines)
- impedance fluctuation
- cable movements
- broken wire contacts
- too much electrode paste/jelly or dried pieces
- low battery

Some of the artifacts can be reduced or completely circumvented by careful preparation, such as impedance fluctuation, cable movements, broken wire contacts or sweating. Other artifacts can be reduced by careful instruction of the subject, such as minor body movements or EMG by jaw muscles. It has to be noted though that almost no subject can sit completely motionless for hours. Some artifacts cannot be reduced but filtered well, especially the net hum produced by the main power lines in the room. In Europe this induces a strong sinus-curve around 50 Hz, fortunately this frequency area does not contain valuable information for most BCIs. It is normally completely suppressed by a notch filter in the recording device. Other artifacts, especially eye movements, cannot be suppressed though. Even if the protocol does not need eye movements, involuntary eye movements are very difficult to suppress and produce quite large artifacts, an example can be seen in Figure 2.3b.

Unfortunately, most BCI studies do not deal with artifacts at all [Fatourechi et al., 2007], which means many results are questionable as the classifier might classify artifacts instead of brain activity [Vaughan et al., 2003; Fatourechi et al., 2007]. When artifacts are dealt with, there are two common strategies: artifact rejection and artifact removal.

"Artifact rejection refers to the process of rejecting the trials affected by artifacts."[Fatourechi et al., 2007]. Often this is done manually by visual inspection of the EEG by an expert. Every trial is inspected and the contaminated ones are removed for the further analysis [Ramoser et al., 2000]. Of course this is an expensive method, requiring at lot of human labor and it is very subjective. Additionally, it is by design not suitable for on-line systems. Because of these disadvantages some research has been done on automatic artifact rejection. For example in [Croft and Barry, 2000] the authors propose to reject all trials were the maximal amplitude exceeds a certain threshold (e.g. 50 μV). Still, all automatic rejection mechanism suffer from a dilemma. If the definition of artifact-free is very strict many trials are rejected and the interface gets sometimes temporarily stalled. If the definition is very broad many contaminated trials are still analyzed.

To reduce this dilemma, different groups have tried to do artifact removal, i.e. somehow removing the artifacts from the data and analyzing the then artifact-free data. One approach to do so would be a frequency filter. Often only certain frequency bands are of interest for the BCI. Artifacts in other, non-overlapping bands can be removed by simple filtering. One example for this technique would be the notch filter mentioned earlier. Often, one also filters lower frequencies to remove artifacts caused by movement [Fatourechi et al., 2007]. Another idea is to measure or model the artifact and to remove it from the data. This has been proposed especially for oculomotor artifacts (EOG artifacts). These artifacts are the most problematic as they are quite strong and difficult to suppress. Additionally, they can be measured quite well by measuring the EOG, i.e. placing electrodes around the eyes, which directly measure the electrical fields produced by the ocular muscles [Croft and Barry, 2000]. There exist different approaches to derive a good approximation of the EEG artifacts from the measured EOG [Croft and Barry, 2000].

Finally, different approaches using Blind Source Separation exist. Blind Source Separation in general tries to "demix" the recorded channels and find the underlying components which generate the measured data. After finding these components, one can usually identify one which is related to ocular artifacts. This component is then removed and the components are "mixed" again to restore the original EEG data without the artifact. The advantage is that the EOG is not needed. Unfortunately, the components usually have to be inspected manually to find the artifact component(s). The most popular algorithm for BSS is the Independent Component Analysis (ICA), several studies tried to use ICA for artifact removal [Jung et al., 2000; Delorme et al., 2007; De Clercq et al., 2006]. Still, it is difficult to ensure that the artifact is removed as completely as possible without loosing non-artifact data.

In total, literature strongly suggests that artifacts need to be considered much more, but cannot be avoided completely. For on-line BCI, especially when tested on healthy subjects, one should at least proof that the information is truly conveyed by the cortical activity and not by eye or body movement.

2.2.4.5. Advantages

Basically, there are three main reasons why EEG is as popular as it is in the neuroscience and the BCI community. It is cheap, easy to use, and offers a very good temporal resolution. While medical EEG devices still cost above €10000 this is feasible for many research labs and much less than a real-time fMRI scanner. The same holds for the necessary preparations, depending on the type and number of electrodes used and the conductivity needed it takes from about 15 minutes up to an hour or more, but compared to invasive methods or fMRI this is still feasible.

The last point mentioned I want to examine in more detail. The presented recording techniques can be grouped into techniques using the electrical activity directly (EEG and ECog) and indirectly through BOLD measures (fMRI and NIRS). One important difference is the temporal resolution. The electrical fields are directly coupled to the brain activity and using modern EEG devices can be sampled at 2000 Hz and more. In contrast, the blood oxygen level needs several seconds to adjust to changed activity [Weiskopf et al., 2004]. For communication and control in BCI setting this is an essential point. BOLD-measures need to wait a few to several seconds between two commands to allow the blood oxygen level to readjust. This means a hard theoretical limit in transmittable information. For EEG the limit is the time-course of the considered brain activity and the low signal quality which often makes repetitions necessary. As there are many brain activities to be considered and many approaches to deal with the low signal quality these practical boundaries can be pushed quite far.

2.2.4.6. Limitations

The electrical fields of the neurons are very weak and the recording site on the scalp is relatively far away. Additionally, between the neurons and the electrodes there are different fluids, the skull, the skin all with different and sometimes very high resistance. This leads to two major limitations.

First, the signal gets “smeared” [Nunez and Srinivasan, 2006, p.41]. This means independently of the number and density of electrodes it is very difficult to obtain a high spatial resolution [Le and Gevins, 1993] or to reconstruct neural sources of the signal [Pascual-Marqui et al., 2002].

Second, the signal-to-noise ratio of EEG is very low. This means that some tedious preparation is necessary to get a usable signal at all, but it also means that only certain brain activity, which generates a strong enough signal, is accessible. For example, it is known from ECog recordings that certain parameters about the movement of the upper limbs can be extracted directly from the electrical fields of the brain, but the low signal-to-noise ratio prevents to do this with surface EEG.

2.2.4.7. Mobile EEG

The general principle of EEG has not changed since the first human EEG in 1929 [Berger, 1929]. Over time the amplifiers and electrodes have been improved gradually. In the last years some new ideas have also come up though. Especially, the concepts of active and dry electrodes have paved the way to a new class of EEG applications: mobile EEG.

One intuitive idea is to move the amplification step closer to the recording site, a concept called active electrode. This should improve signal-to-noise ratio as less external noise is picked up. One of the first, small studies about active electrodes already showed the advantage of having very little external noise. Even the 50 Hz interference produced by power lines vanished [MettingVanRijn et al., 1996]. With passive electrodes this interference is magnitudes stronger than the actual brain signal in unshielded environments. Today, active EEG electrodes are developed by scientists [MettingVanRijn et al., 1996; Mitra et al., 2012; Morikawa et al., 2013] and private corporations (e.g. gTec SAHARA system, Brain Products actiCAP).

Other then better signal quality, another major motivation of developing active electrodes was the ability to use dry electrodes. As described before, wet electrodes need some tedious preparation. Additionally, they dry out over time, which makes it unfeasible to record EEG for more than a couple of hours [Lee et al., 2013]. Dry electrodes are directly applied to the skin with minimal preparations. Modern systems even record well with a lot of different hair types and haircuts.

With only a short preparation and most of the electronics integrated in the electrodes it becomes possible to built mobile EEG systems. With a mobile EEG and small, wearable computers to process the data, new applications are possible, e.g auto-adjustment of a smart-living environment [Lin et al., 2010] or monitoring of driver drowsiness [Hung et al., 2010]. In the first system, different power spectra (namely α and θ) were used to asses the user’s alertness. Depending on the user’s state, parameters of the air-conditioning and the lightning system of the room were adjusted. In the second system, a

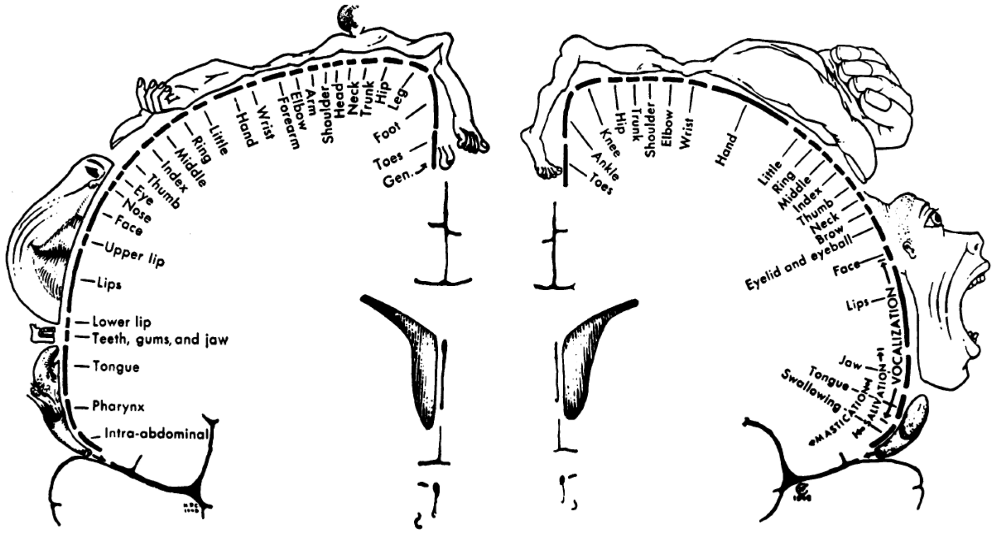


Figure 2.4.: Human sensory (on the left) and motor (on the right) homunculus. It can be seen that the hands are represented by the most neurons on the motor cortex and left and right hand are lying quite far apart. The feet are also prominently represented, but the two feet are lying close to each other. Third-strongest represented is the face, then the tongue. Figure from [Schott, 1993], originally from [Penfield and Rasmussen, 1950].

mobile EEG along with a small, mobile signal processing system was used to detect drowsiness during driving. When drowsiness was detected a warning tone was issued.

2.3. Brain Activity Pattern

To use a BCI as a communication channel one needs some kind of brain activity which can be manipulated intentionally by the subject. Here, I will present four different classes of these activity patterns, which are used in current BCI system based on surface EEG [Pasqualotto et al., 2012; Ortiz-Rosario and Adeli, 2013; Nicolas-Alonso and Gomez-Gil, 2012]. All have different advantages and disadvantages and should be chosen carefully depending on the target group and the target application.

2.3.1. Slow Cortical Potentials

Slow Cortical Potentials (SCPs) were used in one of the first BCI systems, the “Thought Translation Device” (TTD) [Birbaumer et al., 1999]. These potentials are defined as positive or negative deflections of the EEG which last several hundred milliseconds up to some seconds. The Tübingen group which developed the TTD showed that these potentials can be generated voluntarily after training with feedback. The TTD was tested with several ALS patients and successfully gave them a means of communication, but the training for gaining voluntary control takes several months and not all subjects gained control. Additionally, the number of distinct commands is low. Typical SCP systems use a “No condition” and a “Subject evoked SCP” condition as a binary switch. As one of these binary decisions takes seconds, the amount of information which can be communicated is rather low. It is reported that subjects reached a speed of 36 words per hour [Birbaumer and Cohen, 2007]. Because of these disadvantages research on SCP has rather diminished today [Pasqualotto et al., 2012].

2.3.2. Sensorimotor Rhythms

One prominent rhythm in the human EEG is the *μ-rhythm (*$8 - 12$ Hz*)* over the sensorimotor areas of the cortex and the associated *β rhythm (*$18 - 26$ Hz*)*. Their amplitude is maximal when the subject

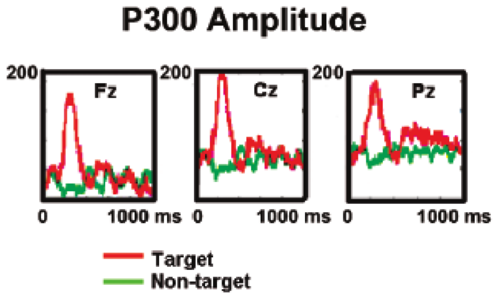


Figure 2.5: An exemplary P300 waveform at three different electrode sites from [Birbaumer, 2006]. EEG data was averaged over many trials. There is a clear difference between attended (target) stimulus and non-attended (non-target).

is not involved in sensory or motor processing and decreases during processing [Pfurtscheller and Lopes da Silva, 1999]. Additionally, the sensor and motor cortex are strongly localized [Penfield and Rasmussen, 1950]. Finally, the decrease in amplitude over the motor cortex, called Event-Related Desynchronization (ERD), occurs not only during executed movements of the corresponding body part, but also during imagined movement [Wolpaw et al., 2002]. After several training sessions most subjects can generate a signal strong enough to be detected automatically in single trials using imagined movement. This allows to built independent BCIs using Sensorimotor Rhythms (SMR). Today, most SMR-BCIs use left and right hand motor imagery along with imagined movements of the feet or the tongue. These body parts are represented largest on the human cortex, as can be seen in Figure 2.4.

For a two-command BCI 90% of the users reach an accuracy of at least 90% after a few training sessions [Wolpaw et al., 2002]. When more commands are used accuracy usually decreases. Still, although the number of distinct commands is quite limited, this brain activity pattern has been very popular, mostly because it is so far the only feasible solution for a truly independent and voluntary BCI. A lot of research has been done, especially by the Wadsworth group [Wolpaw et al., 2002] and the Graz group [Neuper et al., 1999]. Due to this research the transmittable amount of information and the reliability of BCIs based on ERDs could be steadily improved. In addition to the two mentioned groups the group in Lausanne did a lot of research on ERD-BCIs, especially on controlling a semi-autonomous wheelchair via a BCI [Carlson and Millán, 2013; Millán et al., 2009], I will get back to that work in section 2.4 p. 18ff.

Still, for this work a BCI using sensorimotor rhythms seems not feasible. For the control of more complex (robotic) devices the limited number of commands is very difficult to overcome.

2.3.3. Event-Related Potentials

Event-Related Potentials (ERPs) are positive or negative deflections from the baseline activity which are time-locked to an event [Laureys et al., 2009]. The most famous is the *P300*, a positive deflection roughly 300 ms after the event. It is elicited by the so-called oddball paradigm [Pritchard and Walter, 1981; Polich, 2012]. When a subject expects a certain event out of a series of random events the occurrence of the specific event, the oddball, elicits a P300. Usually in the visual P300 paradigm, one looks at some stimuli, e.g. numbers or letters, on a computer screen which flash consecutively in a random order. If one concentrates on a specific stimulus, e.g. by mentally counting the flashes of it, the flashing of that stimulus elicits the P300. An example P300 waveform can be seen in Figure 2.5. The P300 was first used for a BCI system by Donchin and Farwell [Farwell et al., 1986]. The central idea has stayed the same since then. A subject perceives a series of random events, most often a series of different, flashing stimuli on a screen. The subject concentrates on one specific event (one stimulus), usually by mentally counting it's occurrence. Whenever this specific event occurs, a P300 is elicited and the BCI system tries to recognize this. By comparing the timestamps of the events and the occurrence of the P300 the corresponding event can be computed and thus the subject's intention be inferred. As a single occurrence of a P300 potential cannot be detected reliably in the EEG, one usually needs few to several repetitions.

Soon, they developed the row/column mode [Farwell and Donchin, 1988], which is still used today and commonly referred to as the Farwell/Donchin scheme. When the number of stimuli is high, one can arrange them in a grid and always flash whole rows and columns. Then, the chosen stimulus is the intersection of the chosen row and the chosen column. This lowers the numbers of event drastically for a high number of stimuli. A BCI speller might use 26 letter plus some special characters resulting in 36 stimuli. In a 6 times 6 grid, one needs 12 flashes per repetition using the row/column scheme or 36 flashes when every stimulus flashes on its own. An example for a P300 speller will be given in section 2.4 p. 18ff.

To improve the reliability of the P300 detection several different feature extraction and classification algorithms have been tried [Meinicke et al., 2003; Krusienski et al., 2008]. A new approach to enhance the signal quality is to use faces as stimuli for the P300 [Onishi et al., 2011]. So far, evidence suggests that humans are so tuned for processing of faces that the P300 is stronger when the stimuli are faces instead of numbers or letters. In [Onishi et al., 2011] the authors built a BCI system for navigation. The stimuli were 8 arrows pointing in different directions. Instead of flashing them, there were temporarily replaced by a human face. Using only two repetitions all four healthy subjects reached accuracies of more than 90%. Still, one decision took around 3 s.

A P300-BCI is usually considered to be independent, but research has shown that patients without oculomotor control have difficulties in using a visual P300-BCI [Kaufmann et al., 2013]. While concentrating on a stimulus is enough to elicit a P300, it is an open debate how much low-level visual processing contributes to the classification [Kaper et al., 2004]. Additionally, it might not always be feasible to present visual stimuli. For those reasons, it has been tried to elicit P300 with auditory [Kübler et al., 2009; Klobassa et al., 2009] or somatosensory [Lugo et al., 2014; Brouwer and Van Erp, 2010] stimuli. Up to now BCIs using non-visual stimulation seem to be slower than visual P300-BCIs, though.

In total, BCIs using P300 offer a high number of discrete commands, but are quite slow.

2.3.4. Visually-Evoked Potentials

Visually-Evoked Potentials are another approach to detect which stimulus a subject focuses on. There, one directly analyzes the activity of the visual cortex. Using suitable visual stimuli one can elicit Visually-Evoked Potentials (VEPs) that can be detected in the EEG data. Depending on the kind of stimulation one distinguishes different VEP. The most popular is the Steady-State, Visually-Evoked Potential (SSVEP). A stimulus flickering with a constant frequency in the range of roughly $5 - 40$ Hz elicits a response of the visual cortex at the same frequency. This response can be picked up by EEG [Zhu et al., 2010]. This allows to built quite fast BCIs with four to eight commands. Because the target stimulus needs to be fixated with the gaze all VEP-based BCIs are considered to be dependent.

More on VEPs, especially the new cVEP will be explained in chapter 3 p. 29ff.

2.4. BCI Applications

After information about the user's intention has been derived from the recorded data by well-designed signal processing, the acquired information can be used for various kinds of application. Roughly, there are three categories of applications. *Spelling, gaming and device control* [Nicolas-Alonso and Gomez-Gil, 2012]. Additionally, the use of BCI as a rehabilitation device, especially for stroke patients has gained some interest recently, but this is outside the scope of this work. Spelling systems are always designed for some patient group, normally patients with LIS or tetraplegic patients, as healthy subjects have faster, more reliable means of communication. In contrast, when used for gaming, BCIs can be an interesting input modality for healthy subjects and patients, just by being a new, challenging input modality, but also because emotional conditions such as alertness can be measured and used as input for the game. The last category contains a vast range of different applications, from BCIs, to control a smart home [Lin et al., 2010] over BCIs controlling prosthesis for patients [Fifer et al., 2011] to robots which allow bedridden patients some kind of telepresence elsewhere [Leeb et al., 2013]. Actually, as my work deals with controlling (robotic) devices via BCI, the third category was divided into general devices and robotic devices. In the following I will present example studies for all four categories of applications, with a focus on applications controlling (robotic) devices.

Figure 2.6.: User interface for free spelling with predictive text-entry from [Kaufmann et al., 2012]

2.4.1. Spelling Systems

The first motivation for research on BCIs was to bring patients, who are unable to communicate otherwise, a means of communication (with other people). One paper with ignited BCI research was the P300 speller [Farwell and Donchin, 1988]. Ever since then, spelling systems using BCIs have been extensively researched. Today, research focuses on approaches how to move from the laboratory to the clinic, to built systems which are suitable for the end-users. Additionally, spelling often acts as a sample applications for new BAPs and signal processing approaches as the user interface can be maximized for information throughput and the parameters of different spellers (like information throughput, ease of use, and so on) can be more easily compared than other applications.

i. Auto-calibrated speller As an example study to move out of the lab, I want to present a recent study by Kaufmann et al. [Kaufmann et al., 2012]. They programmed a simple-to-use user interface along with a parameterless auto-calibration. The only choice needed from the user was whether to use a predictive dictionary-based text-entry known from modern smart-phones or normal character-by-character spelling. They tested the system with 19 subjects, who had no prior BCI experience. For data recording surface EEG with normal wet electrodes was used. Those had to be set-up by the experimenter, but, as clinical personnel is used to EEG studies, this would not be a major hindrance for a clinical study. Then, after program start, a calibration session was automatically started, where the users had to “spell“ a predefined word to allow the system to acquire training data. Training of the signal processing system and derivation of all system parameters was then done in the background without user or experimenter input. After the classifier has been computed, the system automatically switches to free spelling mode. The user interface with the speller matrix is depicted in Figure 2.6.

For the evaluation, subjects were given a copy-spell task, i. e. they had to spell a predefined sentence. The sentence had 45 characters and had to be spelled twice. Overall the system reached a good accuracy of 91% correctly recognized character in the character-by-character condition. Accuracy was not significantly different in the predictive text-entry condition. Still, compared to other means of communication, this BCI speller was very slow. To spell the two sentences the subjects needed an average of 22.3 min without and 12.4 min with the predictive text-entry.

ii. Spelling for locked-in patients While needing almost no experimenter input, the previous system was still evaluated with healthy subjects. There is growing understanding, that to really develop BCIs suitable for patients the systems have to be tested with these patients. Unfortunately, this makes it even more difficult to test several subjects. As a result, most studies evaluating BCIs with locked-in patients are case studies such as [Sellers et al., 2010; Kaufmann et al., 2013].

In [Sellers et al., 2010] the authors report about a patient suffering from ALS, who had used an eye-gaze device to interact with family and friends and to run his own NIH-funded research laboratory.

When the disease progressed he could no longer use the eye-gaze device. The authors of the study developed a BCI for independent use at home. The caregivers of the patient were instructed how to set-up electrode cap, electrodes and software. The software was connected to several applications including word-processing, e-mail, TV, environmental control. The BCI was a standard P300-BCI using the row/column scheme for up to 72 commands. Data was acquired using a medical EEG amplifier with eight scalp electrodes. The patient used the system for 2.5 years $6 - 8$ h each day. The time for a selection was 26 s on average including a 9 s pause. The patient reached an accuracy of 83%. His P300 potential and the BCI accuracy were stable over the complete time. In total, the system allowed the subject to continue an independent life: "The user and his family state that the BCI has restored his independence in social interactions and in scientific research"[Sellers et al., 2010].

2.4.2. Games

Modern computer games try to give players an increased feeling of immersiveness [Rollings and Adams, 2003, p. 58]. Therefore, the use of BCIs as an (additional), immersive input for video games has gained some interest. The requirements are somewhat different than for BCIs in a clinical settings, for example medical EEG amplifier are too expensive and the usual preparation needed for wet electrode is too cumbersome for gamers. But the recent developments in mobile and sometimes wearable EEG recording systems along with dry electrodes have brought BCIs for gamers much closer to realizability. Two different research lines can be seen. First, traditional BCI layouts can be used as input to control a game completely or partly [Lotte et al., 2010; Maby et al., 2012]. Second, information about the user's emotional state can be used as a supplementary input [van de Laar et al., 2013].

i. BCI as part of the challenge As an example for the first research direction I briefly present the study from Maby et al. about a game they called "BCI Connect Four" [Maby et al., 2012]. The well-known Connect Four is altered by using a P300-BCI for the selection of the column to place the next coin. As an additional challenge they implemented a "Contest Mode" where always both players try to select the column where the next coin is placed. The player with the "stronger" brain activity wins and the coin is placed in the column that player selected. Stronger in this context means, that the classifiers for the two players evaluate the uncertainty of the current classification. The player corresponding to the classifier with less uncertainty wins. This allows to place the coins of the competitor into columns where they are helpful to oneself. A shortcoming of the BCI, it's unreliability, is thus turned into a feature, making the game more interesting. The game was only evaluated with two players. The average accuracy was 81.7% correctly classified columns, which seems rather low for a one-out-of-seven choice, but was enough to fluently play Connect Four. Actually, the authors reported that the game takes similar time with a BCI as with manual control. I think Connect Four is a fitting choice for a BCI game as the only user choice can be translated well into a BCI decision, because there are only seven options. Additionally, these decisions make up only a little portion of the whole gaming time, most time is spent thinking about the best column anyhow.

ii. BCI as supplementary input As an example for the second research direction I want to briefly present α-*WoW*, an adaption of the popular video-game "World of Warcraft" [van de Laar et al., 2013]. In World of Warcraft (WoW) each player controls a character in a fantasy settings. These characters are from one of several different, predefined classes. One such class, the druid, has the ability to transform between two shapes. In the bear shape the character is very strong and can take a lot of damage. In the human shape the character is weaker and can take less damage, but is able to cast powerful magical spells. In standard WoW users change their shape via a key press. In their study, the authors recorded the user's alpha level and the character transformed depending on the alpha level. User's got feedback about their current alpha level (see Figure 2.7) and the threshold for transformation, thus learned to control their alpha level while gaming. The authors did a study with 42 participants using a consumer-grade, mobile EEG (Emotiv EPOC). For the evaluation they relied mainly on questionnaires and reported that users found the BCI condition "a novel and interesting modality" [van de Laar et al., 2013], but they also admit that is remains to be seen whether people loose interest in using the BCI for the game after the novelty wears off.

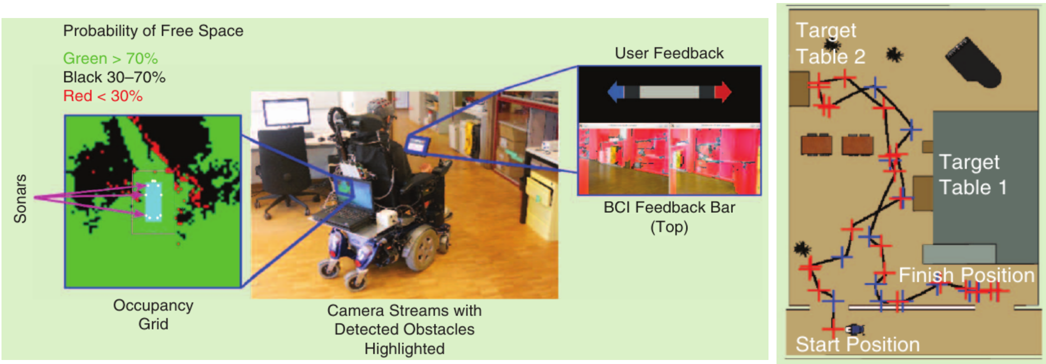
Figure 2.7.: User interface for α-WoW from [van de Laar et al., 2013]. The orange bar on the top left provides feedback on the alpha level.

2.4.3. Device Control

For many disabled patients being able to control simple devices improves autonomy in the daily life a lot. Thus, there has been quite some research how to control different devices with the low information transfer rate and unreliability of BCIs. I will present three areas of this broad research direction. Namely, controlling devices in a smart-home, devices improving mobility (e.g. brain-controlled wheelchairs) and applications in space for healthy users.

i. Smart-home Smart-home, as a topic within BCI research, is a broad topic, including home automation, smart TVs and all other devices at home which can be controlled by a computer interface. As an example I want to examine a study from Aloise et at. [Aloise et al., 2011]. They built a P300-BCI system to control various devices at home, such as the phone, the inter-phone, the DVD-player and a fan. Most of these domotic devices need few commands with little pressure of time, e.g. when switching on the DVD player it does matter little whether this takes 5 s or 15 s and one transmitted command, e.g. "Play" is enough in most situations. But, these devices also rarely need commands. This is a requirement that is seldom considered in BCI research. The BCI has to detect when it is not needed. In the examined study the authors recorded training data for a "No Control" condition as well as normal training data for P300 containing target and non-target epochs. They trained a classifier with P300 targets and non-targets and could show that the classifier scores of the classifier for the "No control" epochs differed from P300 targets, thus the classifier scores could indicate well when the user did not want to issue commands at all. In their evaluation, the system only wrongly issued commands in 1.5% of epochs in the "No control" condition and correctly classified 88.73% of the "Control" trials. These values allowed their subjects to control their home environment reliably. As they tried the system with eleven healthy subjects it remains to be seen how the results can be transferred to patients.

ii. Mobility Many studies try to offer immobile patients a means of mobility, e.g. by means of a BCI controlled wheelchair. In a recent study by Carlson et al. [Carlson and Millán, 2013] the authors present a wheelchair employing a "shared control" strategy. This means the user only controls the high-level behavior. The wheelchair itself is occupied with some sensors and deals with low-level behavior, such as obstacle avoidance. Their BCI system conveys three different commands to the wheelchair "Turn left", "Turn right", or "No command". To issue these commands they use an SMR-BCI with Motor Imagery. They trained the user's with three classes: Left hand, right hand or both feet and then select the two best-discriminable ones to control the wheelchair. As long as the user issues "No command" by not imaging one of the two commands the wheelchair drives forward, while avoiding obstacles. When the user employs Motor Imagery the corresponding command gets sent to the wheelchair and the wheelchair turns and then drives forward again.



(a) The semi-autonomous wheelchair

(b) Schematics of the office with two example BCI trajectories

Figure 2.8.: Both figures from [Carlson and Millán, 2013]

Figure 2.8a shows the wheelchair along with the user interface for the user and the user interface for the experimenter, which shows the wheelchairs knowledge about the surroundings. The authors did an evaluation with four healthy subjects. For the on-line evaluation users had to drive with the wheelchair through a standard office. Figure 2.8b shows a schematic of the office along with an example trajectory. The users had to drive from the entrance through a narrow passage between two desks to a third desk, turn around, drive again through that passage and drive to the exit of the room. All subjects managed to do so in all runs. They needed in average 417.6 s ± 108.1 s. In average this was 160 s more than the same subjects needed in a manual control condition. They had a very reliable control, but they also had many training sessions and very good Motor Imagery classification rates in general. In a cursor-control task, which they did before each wheelchair session, the worst subject reached an accuracy of 90%. It is questionable if patients could reach such good classification accuracy with a Motor Imagery BCI.

Other groups did research on driving a car by a BCI [Göhring et al., 2013; Zhao et al., 2009]. Again, the low information transfer rate gets accounted for by moving as many decisions as possible from the user to the device. In [Göhring et al., 2013] the car was equipped for fully autonomous driving. The user only decides in which direction to drive. Actually, they implemented two modes, one where the car's direction was directly controlled through four mental commands for accelerate, brake, left, right and one where the user only made decisions at crossroads. Despite the even lower signal-to-noise ratio the authors did not use a medical EEG, but a mobile, consumer-grade EEG the Emotive EPOC. As brain activity four mental, motor imagery commands were used ("left“,"right“,"pull“,"push“). Subjects were able to roughly follow the predefined trajectory with an average deviation of 1.8 m when only controlling direction (speed was fixed) and 2.8 m when controlling speed and direction.

When only having to choose the direction at the next crossroad (shared control), subjects had 10 s to generate to correct command and succeeded in more than 90% of the trials.

In [Zhao et al., 2009] a medical EEG device was used to control a virtual car. They also used Motor Imagery of left hand, right hand, both feet, but they not only considered the location of the resulting ERD, but also the time duration of the ERD. Additionally, their BCI did not control the car directly, but the steering wheel and the accelerator. Subjects where seated in front a standard computer monitor and saw a rendered 3D scene of a car on a lane, with feedback about the status of the steering wheel and the accelerator. The system was evaluated with four subjects, two of those successfully managed to drive on a straight lane an S-pattern around several cones and drive on a curved road. Here success means that the car did not hit a cone or leave the road. Unfortunately, no quantitative evaluation was done.

iii. Space Besides devices designed for patients, BCIs could also be useful as a hands-free means of controlling devices. One area where this could be useful is research in space. Manual control of

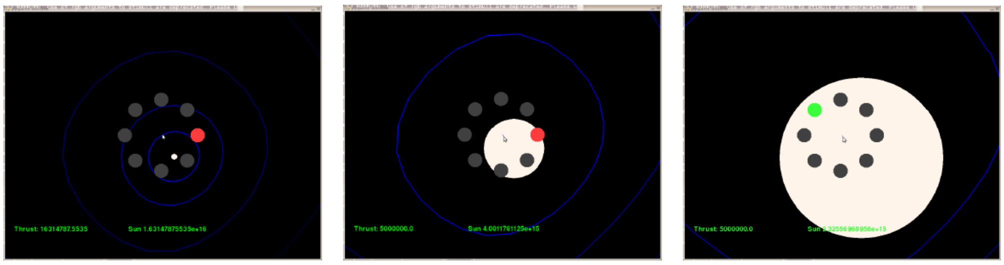


Figure 2.9.: SpaceCommander interface adapted for BCI from [Poli et al., 2013].

computers and devices in space is slow and error-prone [Poli et al., 2013]. In their recent study Poli et al. built a system where two user's control a virtual spacecraft collaboratively [Poli et al., 2013]. Both users together control a 2D mouse pointer with a P300-BCI. Around the pointer eight circles flash in random order, the user selects one by concentrating on it, as with the usual P300 oddball paradigm, the cursor then moves into the direction of the detected circle. As a virtual spacecraft the authors used the "SpaceCommander" game. The spacecraft in the game follows the mouse pointer and had to be steered to do a fly-by maneuver around a star without planets.

The user interface with the circles to steer the spacecraft can be seen in Figure 2.9. Subject pairs where able to complete the task in 66.7% of trials. As with many studies controlling devices, the authors did some off-line evaluation runs to analyze the accuracy of their system, but have little data to confirm that the systems works on-line.

2.4.4. Control of Robotic Devices

Several studies have also tried to control robotic devices with a BCI. The central idea is to have the robot execute some of the tasks the patients themselves cannot do. Often the robot is used as a kind of telepresence device so patients can virtually leave their hospital room [Carlson et al., 2013]. Additionally, a robot might be able to fetch things for the patient or help them with daily activities like cooking or coffee-making.

i. Robots doing pick-and-place The first study to draw attention to the idea of controlling humanoid robots by BCI was done by Bell et al. 2008 [Bell et al., 2008]. They built a BCI to navigate a humanoid robot (a Fujitsu HOAP-2 humanoid robot with 25 degrees-of-freedom) through a room and to choose objects to be picked up by the robot. The BCI was a P300 system with up to six different commands. The user could see the scene through the head camera of the robot. The robot would walk to a predefined table which had some colored cubes lying on it. In the image from the head camera the objects on the table would flash as necessary for the oddball paradigm to elicit the P300. The user would concentrate on the flashes of one of the objects to select it for pick-up. By detecting the P300 the system would derive which object to pick-up and sent the command to the robot. Then the system would present possible target locations for the picked-up object in a similar way and the robot would walk to the desired target and put the object down. The room with two target objects and two target locations is depicted in Figure 2.10. From the BCI perspective this is a rather simple setting, just two standard P300 decisions are used, but, in total, the setting accomplishes a complete pick-and-place task. Of course, this heavily relies on suitable robotics where the robot can handle most aspects of the task itself and the user only needs to specify the goals. On the one hand, this is probably the only way a complex, humanoid robot with many degrees of freedom can be controlled over a communication channel as slow and unreliable as a BCI. Nearly all studies on BCI for robot control apply this approach. On the other hand, it severely restricts the user in the task he or she can accomplish with the robot. Every action to be taken by the robot must be programmed in advance.

The authors evaluated the performance of the BCI by indicating the target actions to the user. Using five repetitions an accuracy of 95% was reached for a four command situation. To ensure that it is possible to use a flexible number of stimuli, the authors also tried to train the classification system using data of the four command situation and classify data of a six command situation. The averaged

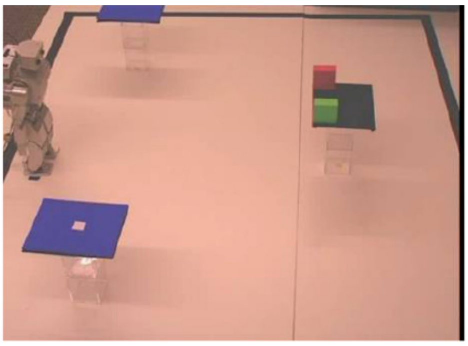


Figure 2.10: Scenario of BCI-controlled, humanoid robot doing pick-and-place. At the right table two target objects to be picked can be seen. On the left the robot is shown along with two possible target locations to place objects. Image from [Bell et al., 2008].

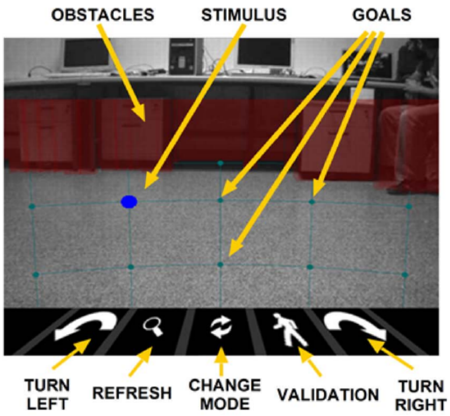
classification accuracy was roughly 85% with five repetitions or 93% with ten repetitions. In total, the interface offers limited interaction and it takes several seconds to issue a command, but the accuracy is very high and the study showed that it is possible to control a humanoid with a BCI.

Another study with a pick-and-place task gradually increased the amount of control of the user [Finke et al., 2012]. The humanoid robot Nao could be moved freely through the room by means of a hybrid P300 / SMR BCI (P300 for going/stopping, SMR for turning left/right). Whenever a target object was within the view of the robot, a P300 decision was done whether to pick up the object and, in case multiple objects were in view, which object to pick up. The system worked asynchronously, i.e it detected when the user did not want to issue commands. Two users, each controlling one robot worked together or competed with each other. In the competitive condition, the task was to navigate the Nao through the room, pick-up one of three colored cubes and bring that cube to ones home position. In the collaborative task each Nao had to move the cube half of the complete distance. The system was evaluated with six pairs. The users were able to control the robots quite well, but due to the slowness of the BCI and the slowness of the robot, this simple task took on average more than 9 minutes.

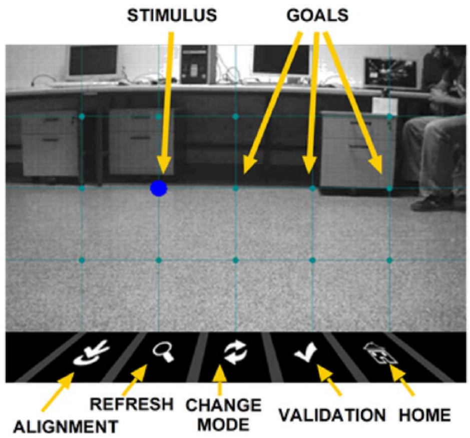
ii. A P300 BCI controlling a telepresence robot When a system allows the user to navigate a robot and activate some autonomous behaviors of it, one gets close to a telepresence robot, which would, in some sense, allow patients to leave their beds. One study evaluating a simple telepresence system controlled by a BCI was done by Escolano et at. [Escolano et al., 2012]. They used a P300-BCI to steer a remotely located telepresence robot. The user interface consists of a live image of the camera and the P300 stimuli, which are directly overlaid on the image from the robot. It is depicted in Figure 2.11. The BCI had two modes: navigation and camera control. In both modes there were some general menu buttons below the camera image. For every command it was necessary to first activate the command and then activate a "Validation" command. This was a precaution against wrong commands. Additionally, a grid was overlaid over the robot's camera image indicating positions where the robot could drive to. When one of those stimuli was selected and validated the robot autonomously drove to that new position. In camera mode, the overlaid stimuli would turn the camera into the direction of the stimulus, such that the center of the camera view would point at the selected stimulus. The whole interface, except the stimuli, was kept in black-white to have a high and stable contrast between background and stimuli regardless of the surroundings. With these stimuli a standard Donchin scheme was used for the P300 stimulation process to lower the number of epochs needed for one decision. The menu bar below and the overlaid grid where separated in rows and columns and flashed according to the Donchin scheme. The number of repetitions was subject-dependent.

As with most of the BCI controlled robots, the authors employed a "shared control". Here, the user can freely turn the camera in camera mode and tell the robot where to drive next in navigation mode. When driving the robot does autonomous path planning and obstacle avoidance.

The system was evaluated by a user study with five healthy subjects. At first, each participant did a performance assessment without the robot, then each participant had to carry out two tasks on-line with the robot. In the assessment phase, participants first did a screening to check whether the P300 potential was elicited, all participant succeeded. Then, each participant did a training phase where training data was acquired for the classification system and the trained classifier was evaluated on-line with the user interface, but without the robot. All participants reached an accuracy of more than 93%.

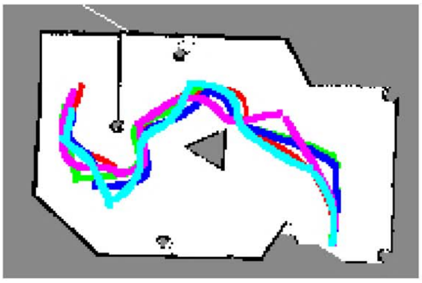


(a) Navigation mode

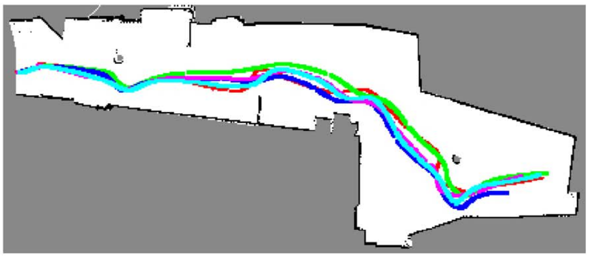


(b) Camera control mode

Figure 2.11.: User interface for the two modes of the BCI controlled telepresence robot, from [Escolano et al., 2012]. The figures show the menu bar below and the stimulus grid overlaid on the image from the robot. For the images a single stimulus was highlighted, in the real application a row-column scheme was applied.



(a) Task one: Constrained environment, two targets.



(b) Task two: Open space. One target

Figure 2.12.: The two robotic tasks for the evaluation of the BCI controlled telepresence robot. Figures from [Escolano et al., 2012].

After that, the two robotic tasks were done. In both tasks the robot had to be navigated from a start to a goal position and some target along the route had to be looked at with the robot's camera. In task one, the robot had to be steered through a constrained environment and visually search for two targets. In task two, the robot had to be steered through open space with one visual target. Both scenarios can be seen in Figure 2.12.

The results of the study were very thoroughly analyzed in the paper, including the feasibility of the shared control and so on. Here, I will concentrate on the results of the BCI. The main metrics can be seen in Table 2.2. The accuracy of the BCI is analyzed with two different metrics, the real and practical accuracy. This accounts for the fact that in a navigation setting sometimes when the correct command is not inferred by the BCI the executed command still moves the robot roughly into the correct direction (called a reusable error in the paper). To compare the BCI with others, the real accuracy still seems more suitable as the amount of reusable errors is determined to a high extent by pure chance. Notably, the real accuracy varies from 73% to 100% although the number of repetitions was adjusted in such way that all participant reached an accuracy above 93% in the preliminary on-line phase without the robot. The real task setting seems to be cognitively demanding enough that the accuracy drops. Still, 73% and better is enough to control the robot quite reliably especially with the additional validation phase. This validation phase seems to be necessary, though. If the validation phase would always just confirm the previous command the number of selections per

Table 2.2.: Results for the BCI evaluation of the two tasks where the real robot was controlled. Table from [Escolano et al., 2012].

	Task 1				Task 2			
	min	max	mean	std	min	max	mean	std
# total errors	0.00	7.00	3.50	2.88	0.00	11.00	4.90	3.70
# reusable errors	0.00	2.00	0.60	0.84	0.00	5.00	1.20	1.81
Real BCI acc.	0.81	1.00	0.90	0.08	0.73	1.00	0.86	0.09
Practical BCI acc.	0.83	1.00	0.92	0.07	0.78	1.00	0.89	0.07
# selections/min	3.39	5.49	4.41	0.72	3.40	4.77	4.16	0.46
# selections/mission	2.11	3.08	2.54	0.34	2.36	3.40	2.80	0.39
# missions/min	1.17	2.27	1.77	0.39	1.00	2.02	1.53	0.33
# sequences	6.00	10.00	8.00	1.33	8.00	10.00	8.40	0.84
ITR (bits/min)	9.97	21.73	16.05	3.83	9.86	20.62	14.32	3.33
# misunderstandings	0.00	0.00	0.00	0.00	0.00	1.00	0.10	0.32

command (called mission in the paper) should be roughly 2.0, but the average values for this metric are 2.5 for the first task and 2.8 for the second task. With several repetitions (six to ten, depending on the subject) and the additional validation phase the speed of the interface is quite slow even for a BCI. With the validation phase only 1.0 to 2.27 commands per minute could be sent to the robot. The whole evaluation session with the robot lasted about 4 h per participant.

In total, this study on one hand illustrates well the vision behind BCIs for navigation and control, the robot can serve as some kind of avatar or surrogate body. For a full telepresence system a means of communication is missing of course, but could be easily added by adding one of the presented P300 spellers as a third mode and having some speakers and microphones in addition to the camera. On the other hand this study also illustrates the difficulties. Accuracies are often even lower than for BCIs in spelling systems and the number of commands per minute is very low.

iii. Hybrid BCI for robot control To augment the amount of transmittable information and the number of distinct commands, one can also use a hybrid BCI [Choi and Jo, 2013; Riechmann et al., 2011; Hachmeister et al., 2011]. In [Choi and Jo, 2013] a mobile EEG, the Emotive EPOC was used to navigate the humanoid robot Nao. The authors combined a one-class SMR and a two class SSVEP-BCI to have three classes in total. The poor signal quality of the consumer EEG device seems not to permit more classes per condition. Even with a small number of classes one decision lasted 5 s, but at least this led to an average accuracy of more than 80% in an off-line evaluation before the robot experiment. When controlling the robot, the SMR was used to start and stop the robot's forward movement and the two SSVEP classes to turn the robot left or right. For the evaluation three subjects had to navigate the robot through a small maze by BCI and by keyboard control. The authors then compared the two conditions using different metrics, like total time, number of forward and turning steps, and others. They could, for example, show that on average the total time in the BCI condition is only 1.2 times higher than in the keyboard condition. This seems to be a very good value, but this kind of metrics should be regarded with caution as long as one does not know how the ratio of time needed for the robot to move and time needed to wait for commands is. The quotient of the number of forward and turning step of the two conditions is close to one, indicating a good accuracy of the BCI and a reliable control. In total, it is interesting that a humanoid can be controlled by a consumer-grade EEG device, but the number of distinct commands is low and certainly not high enough for real applications.

In [Riechmann et al., 2011; Hachmeister et al., 2011] our working group presented a P300/SMR hybrid system to control humanoid robots. The idea was to use the strength of the ERD produced by left hand or right hand motor imagery as a continuous control dimension and up to five P300 stimuli as a discrete control dimension. Depending on the situation in the scenario, one or both of them might be suited well. When both are used simultaneously one needs to detect which one was used by the subject, an issue we investigated in [Riechmann et al., 2011]. As the P300 potential can be detected more reliably, we decide to use the presence or absence of the P300 as indicator whether P300 or ERD was intended by the subject. We found, that using ten repetitions the correct pattern was detected with an accuracy of 95%. In [Hachmeister et al., 2011] the hybrid BCI was used to control the humanoid iCub in an example scenario. The task was to take control by activating one button in the continuously running P300 stimulus presentation, then to turn the head towards a target position. Left and right Motor Imagery was used to turn the head of the robot. At the same time a five stimuli P300 stimulus was running, where one of the five stimuli acted as stop button for the head motion. When the subject was satisfied with the head orientation and had stopped the head movement, another P300 decision was started to select one out of five facial expressions. The system was evaluated with one healthy participant. She successfully completed the task in four out of five runs, but again the interaction was quite limited.

2.5. Evaluating BCIs

To evaluate a certain BCI system one needs to compare the results between different studies. There are many metrics to evaluate BCI systems. Most studies give some results concerning accuracy and latency of their BCI system. Other examples are: time to complete a given task, specific accuracy per class (confusion matrix), κ-coefficient, ROC-curve, signal-noise-ratio, mutual information criteria and so on. A well-written summary can be found in [Dornhege, 2007, chapter 19, p. 327]. Although many studies use similar metrics, it is often still difficult to compare the results of different studies [Billinger et al., 2013]. Here, I want to examine some common metrics, which I also used to evaluate my own experiments.

For spelling BCIs the most important metric is the information transfer rate (ITR) [Spüler et al., 2012a; Billinger et al., 2013]. First derived by Wolpaw et al. [Wolpaw et al., 2000] it allows to incorporate accuracy, latency and number of commands into one formula, which computes the amount of information that can be transferred in a given time:

$$ITR = log_2(M) + p * log_2(p) + (1-p) * log_2(\frac{1-p}{M-1}) \tag{2.1}$$

here M is the number of different commands or letters to choose from and p is the classification accuracy. This equation gives the ITR in bits per trial. This allows a comparison between different BCI spelling systems and comparisons between BCIs and other communication methods. Often the ITR in bits per trial is multiplied with the number of trials per minute to give the ITR in bits per minute. This allows for a fairer comparison as different BCI approaches have very different trial length and for the user the information transfer per time is much more important.

In spelling applications the amount of transferable information is often considered the most important factor, although, especially with patients, other factors, for example convenience of use, arise as well.

For the evaluation of BCI systems for navigation and control no standard metric exists [Thomas et al., 2013]. As I already presented when I discussed the examples, most authors give some accuracy metric and some metric dealing with time, like the time needed to complete a given task. Still, the time needed to complete a given task is mainly dependent on the given task, thus cannot be compared between studies. Additionally, the accuracy strongly depends on the latency as it can be pushed easily by using more repetitions. The cost of a misclassification also depends strongly on the situation it happens in. When a light bulb is turned on by mistake, it can be easily turned off again. To revert a robotic action is often more time consuming. This is taken into account by a new metric called *efficiency*, proposed in [Quitadamo et al., 2012]. It takes into account accuracy and cost of misclassification. Still, the cost of a misclassification has to computed by the designer and the metric does not take into account the latency of decisions.

So, to obtain comparable metrics beyond basic accuracy and latency, I will focus on two approaches.

First, the time needed in terms of the task, i.e. the amount of time needed by the BCI in relation to the total amount of time needed for the task. For example, when the device needs 2 min for the completion of a specific command, it is not an issue whether the user needs 10 s or 20 s to send this command via the BCI. In contrast, when doing navigation where a step of a robot might take for example 1 s the latency of the BCI is important. This metric is also used for example in [Escolano et al., 2012].

Second, I will compare the BCI with a manual keyboard control. First, in terms of latency and, second, also in the number of commands needed to complete a given task. Of course, compared to the keyboard the BCI always needs more time and commands. Still, the time and commands needed with the keyboard give a comparable measure of the complexity of the task and thus helps in comparing the BCI with other approaches. This measure was also used for example in [Chae et al., 2012; Legény et al., 2011].

For more metrics concerning BCI systems for navigation and control see [Escolano et al., 2012].

3. Visually-Evoked Potentials

A Visually-Evoked Potential (VEP) is an occipital potential, which is measurable by surface EEG roughly over the visual cortex as a reaction to visual stimuli. It has long been used in clinical neuroscience as "important diagnostic information regarding the functional integrity of the visual system" [Odom et al., 2004]. It is also an interesting input modality for BCI. The visual cortex reacts very fast to changing stimuli, it's activity can be measured well using surface EEG and it's method of operating is quite well understood [Celesia, 2005]. This allows to build *faster and more reliable BCIs* compared to EEG-based BCIs using other brain activities [Bin et al., 2009a]. The major drawback of BCIs using VEPs is their dependence on eye-gaze, I will get back to this later in this chapter.

In general, a VEP-BCI user interface is similar to a visual P300-BCI display. Different stimuli are presented on the screen and the user attends to one of them. In a VEP-BCI these stimuli are designed to elicit different activities in the visual cortex. Although usually more than one stimulus is within the subject's field of view, the focused stimulus dominates the activity of the visual cortex.

There are different ways of stimulating the visual cortex. For BCIs two approaches are most popular, SSVEP and cVEP [Bin et al., 2009a]. SSVEP-BCIs display flickering stimuli at different, fixed frequencies. This elicits periodic brain activity with the same frequency as the flickering of the focused stimulus. *cVEP* uses complex codes to modulate on and off phases of the different stimuli. This allows for more stimuli at the same time, but elicits more complex brain activity.

In addition to these two, some research has been done on transient VEP (tVEP) which occur at the start of a flickering period [Cui et al., 2004] and motion on-set VEPs, which occur when stimuli are moved [Guo et al., 2008]. These last two are not considered in this work, as BCIs based on them are rather slow.

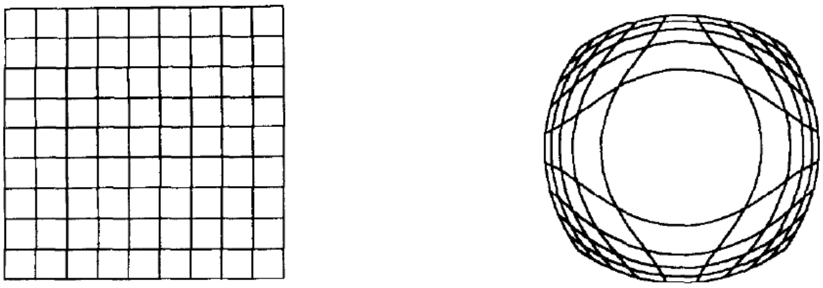
Next, I introduce the electro-physiological basis of visually-evoked potentials and explain why they are suited well for EEG-based BCIs. Then SSVEP and cVEP are examined and the current status of the research on both of them is presented. Last, I compare VEP-BCIs with other BCIs and input interfaces using eye-tracking, explaining what the advantages and limitations of VEP-BCIs are.

3.1. Electrophysiological Basis

The human visual cortex has some properties which make it an interesting input for EEG-based BCIs. First, it can be measured very well using EEG. It lies directly under the occipital scalp and it's pyramidal structure allows the action potentials to sum up [Celesia, 2005, p.118].

Second, it reacts very fast to changing input of the retina. When using SSVEP for diagnosing visual disorders, only 250 ms of data are needed for the analysis [Celesia, 2005, p.118]. One can expect the start of a response to be in the range of 40 ms [Spüler et al., 2012b] to 80 ms [Sutter, 1992]. For comparison, the P300 component starts after $300 - 400$ ms.

Third, there is the effect of *cortical magnification*. Although humans have a broad field of vision, the density of sensory cells in the fovea region of the retina is much higher than in the periphery and this distribution is reflected in the visual cortex. This means that the brain activity of the visual cortex is mainly influenced by the stimulation of the fovea region [Sutter, 1992]. An illustration of this effect is given in Figure 3.1. Some experiments to quantify it can be found in [Sutter, 1992], he found that the influence of the visual input on the measured activity "decreases almost exponentially" from the fovea to the periphery. This effect is essential to build BCIs using VEPs as it means that the brain activity is almost completely determined by the stimulus the subjects focuses on. Thus, the subject can convey information by modulating his brain activity through gazing at a specific stimulus.



(a) (b)

Figure 3.1.: Illustration of the effects of cortical magnification. (b) Represents the approximate image on the surface of the primary visual cortex when a subject fixates the center of the grid shown in (a). Figure and description from [Sutter, 1992].

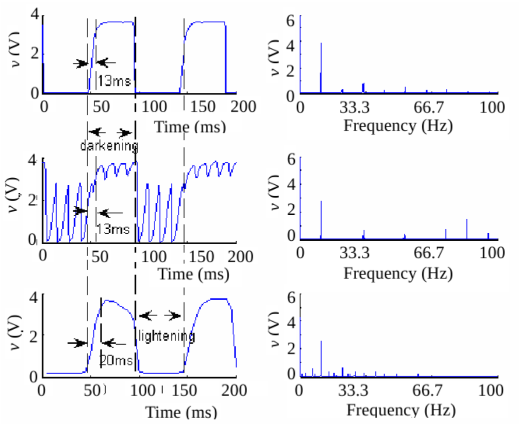
3.2. Steady-State, Visually-Evoked Potentials

The most popular and best researched visually-evoked potential is the SSVEP. When a subject gazes at a stimulus, which flickers with a fixed frequency between 5 Hz and 100 Hz sinusoidal brain activity at the same frequency and its harmonics can be observed [Wu and Yao, 2009; Herrmann, 2001]. This can be used to construct BCIs. Some stimuli are flickering at *different, fixed frequencies*. The subject selects one of these stimuli by directly gazing at it. This induces a brain activity in the visual cortex at the frequency of that stimulus and its harmonics. Using band power analysis this frequency can be isolated from the EEG measurement and thus the attended stimulus be recognized. Typically, between two and eight different commands are used. There exists also some studies which use stimuli flickering at the same frequency, but with different phases, these are not considered here [Kluge and Hartmann, 2007; Wu and Yao, 2008].

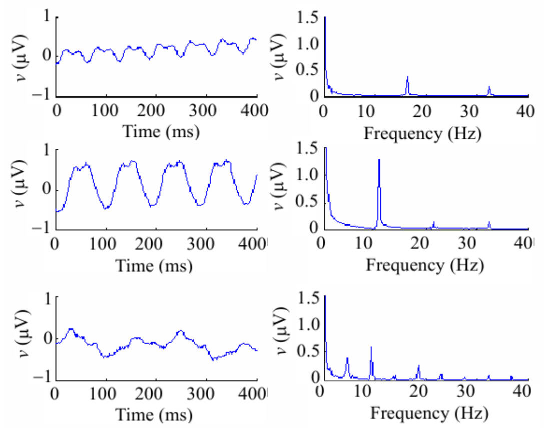
3.2.1. Interface Design

When designing such a BCI one has to decide for a stimulation method first. So far, studies mainly used LED-arrays [Zhu et al., 2010], but CRT monitors have been used as well [Bakardjian et al., 2010; Wang et al., 2008; Kelly et al., 2005]. LCDs seem not to be suitable as they cannot produce a sharp change from black to white. As of 2013, no studies using LCDs for SSVEP BCIs have been reported [Amiri et al., 2013]. There is only one notable exception, a group from Bremen did several successful studies using an LCD screen, e.g [Volosyak, 2011]. When comparing LED-arrays with CRT displays both have advantages. In an LED-array every LED can be switched on and off with very exact timing allowing to use every possible frequency for the stimulation. CRT can only use frequencies which are integer factors of their nominal frequency. Additionally, LED-arrays produce better flickering, as was examined in [Wu and Yao, 2009] and is depicted in Figure 3.2a. CRT displays by design induce a high frequency flicker at their operating frequency on top of the desired signal. When these disadvantages can be coped with, CRTs have the advantage that the user interface for the BCI user and the stimuli can be integrated in one interface. This is quite important as it is more comfortable for the user and generates better signals as the user needs less time looking back-and-forth between interface and stimuli.

The next necessary decision is to decide which frequencies to use. Several studies found that the signal strength of the SSVEP response varies greatly between different frequencies [Zhu et al., 2010; Kuś et al., 2013], see also Figure 3.2b. Additionally, user comfort also depends on the frequency range used [Zhu et al., 2010]. Currently, a lot of different frequencies are studied. Commonly, they are roughly divided into three bands: Low ($1-12$ Hz), medium ($12-30$ Hz) and high ($30-60$ Hz) [Regan, 1989; Zhu et al., 2010]. The results which frequencies produce the strongest signal are not fully in agreement [Kuś et al., 2013; Regan, 1975; Pastor et al., 2003; Herrmann, 2001], partly because all studies examined different frequencies regions under different conditions. Three conclusions are possible though. First, the best frequencies are subject specific. Second, some frequencies are suitable



(a) Qualitative comparison of different stimulation methods. From top to bottom: LED-array, CRT monitor, LED monitor. Left: signal waveform, right: frequency spectrum. Stimulus frequency was 10.8 Hz. The LED-array produce the cleanest waveform and spectrum, the CRT produces a high frequency jitter at its operating frequency, the LCD cannot produce sharp edges.



(b) Qualitative comparison of different frequencies. From top to bottom: 16.1 Hz, 10.8 Hz, 4.6 Hz. Left: signal waveform, right: frequency spectrum. Both are the average of the EEG of one subject at one electrode. 10.8 Hz produces the strongest signal. The signal generated by the other frequencies is much weaker. For all frequencies the harmonics are also present in the spectrum.

Figure 3.2.: Both figures from [Wu and Yao, 2009].

for many subjects, particular frequencies around $10 - 15$ Hz. Third, frequencies in the higher bands evoke smaller amplitudes, but are less annoying and tiring for the users.

When using CRT monitors for the stimulation the user interface and the stimuli can be integrated, but one has to keep in mind that the interface design also has an influence on signal strength, for example it is important to have enough contrast between stimuli and background [Gergondet et al., 2011]. Parameters to be considered when designing the stimuli are flickering method, contrast, color [Mouli et al., 2013], size [Bakardjian et al., 2010], shape [Ng et al., 2013], pattern [da Cruz et al., 2013].

An example interface can be in Figure 3.3. It is quite typical. Similar interfaces can be found for example in [Kuś et al., 2013; Kelly et al., 2005; Bin et al., 2009b]. Some stimuli, in this case five, are positioned around the actual user interface, in this case the letters to be chosen, all on one computer screen. The stimuli are quite large, rectangular areas and flicker black/white. Some more diverse interfaces will be examined in section 3.2.3.

3.2.2. Classification

As mentioned earlier, the flickering frequency of the attended stimulus can be found in the occipital EEG. In Figure 3.2b this effect seems to be very prominent, but that data was averaged over many trials. To detect the frequency in a single trial of just one or a few seconds needs some effort in signal processing. In earlier studies the preferred method was *power spectral density analysis (PSDA)* [Hakvoort et al., 2011]. The power spectral density for a given time window is estimated by computing the Fourier transform and analyzing the coefficients. The output of the PSDA can then be classified further, in the simplest case the frequency with the maximal coefficients is chosen. There are mainly two issues to be considered when using the PSDA for SSVEP classification. First, one needs to combine the data from the different electrodes either before or after the PSDA and second, one needs to deal with harmonics somehow.

Often the different electrodes are combined before the PSDA by *spatial filtering*. The idea is to combine all relevant channels into one. An study comparing different approaches was done in [Garcia-Molina and Zhu, 2011]. Mainly, there are methods which select the one or two best channels just by

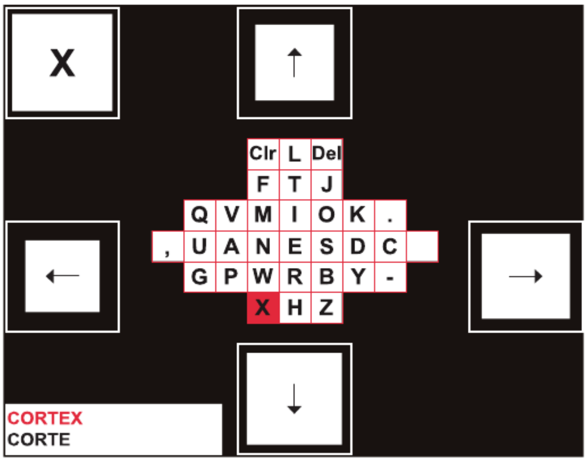


Figure 3.3.: User interface for the Bremen SSVEP speller from [Volosyak, 2011]. The four arrows allow to move the cursor, the stimulus in the upper left corner allows to select the letter under the cursor. In the lower left the text to spell is depicted together with the text spelled so far.

comparing the signal quality in all channels, statistical approaches like Principal Component Analysis (PCA) and Independent Component Analysis (ICA), and approaches which estimate or model the SSVEP and/or the noise and/or the signal to noise ratio, like Minimum Energy Combination (MEC), Common Spatial Filter (CSP), Maximum Contrast Combination (MCC). The authors found that the combination of the two best channels and Maximum Contrast Combination worked best with differences up to 10% in classification accuracy between the different filters. In contrast the Bremen work-group very successfully uses Minimum Energy Combination [Volosyak, 2011].

For the second issue, the harmonics, an exemplary study [Müller-Putz et al., 2005] examined their importance. They used a PSDA to estimate the band-power at the stimulation frequencies and up to the third harmonic. These features were combined in one vector which was classified using distinction sensitive learning vector quantization (DSLVQ) [Pregenzer and Pfurtscheller, 1999]. DSLVQ is an enhancement to Learning Vector Quantization (LVQ). In LVQ each class is represented by codebook vector or prototypes, which are derived from the training data. For classification the distance of the new item to the prototypes is calculated, usually using Euclidean distances. The class label of the nearest prototype is used as class label for the new item. DSLVQ additionally weights the dimension, thus computing a weighted Euclidean distance. The weights are derived during training. Dimensions which positively influence the classification get higher weights, dimension which are responsible for misclassification get lower weights. Using DSLVQ on the band-power coefficients the authors performed an off-line study to asses the contribution of harmonics[1]. Nine subjects participated and did 240 trials without feedback. Their results show that all three harmonics contribute the classification accuracy, i. e. classification with three harmonics is significantly better than with two which is already significantly better than using only one. With all three harmonics an average accuracy of 63.8%±9.3% was reached in a 4-class problem using epochs of 1 s length.

Another approach for spatial filtering and classification of SSVEPs is the *Canonical Correlation Analysis (CCA)*. A common procedure in multivariate statistics, it is useful when one has two sets X and Y which are correlated somehow. CCA then finds linear projections A and B to maximize the correlation between AX and BY. I will describe CCA in detail in section 4.1.2 as I use it as well for my experiments. When classifying SSVEP the measured EEG is set as X and Y gets constructed as "ideal" brain response to a stimulus, i.e. a sinusoidal waveform of the frequencies and the harmonics. One constructs one such Y for every stimulus frequency and performs the CCA between the EEG data and every Y. The Y which gets the highest correlation is chosen and the system outputs the command

[1] In the paper a second on-line study is also presented, but that does not evaluate harmonics and is therefore not discussed here.

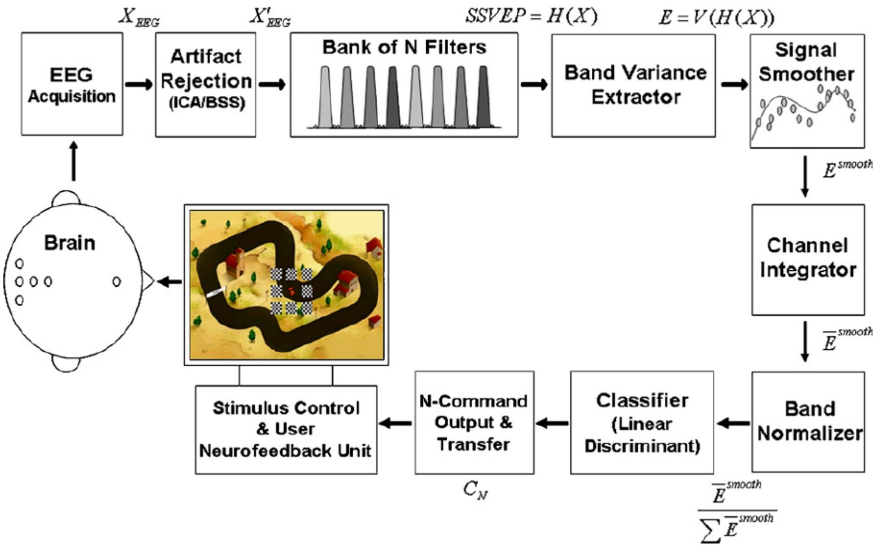


Figure 3.4.: User interface and signal processing pipeline for the 2D control of a virtual car from [Bakardjian et al., 2010]. In the user interface the car moves along the track. The track is stationary and always completely in view. The eight stimuli move along with the car.

of the corresponding frequency. This method was introduced in [Lin et al., 2006; Bin et al., 2009b] and improved in [Zhang et al., 2011, 2013a].

3.2.3. Example Studies

Here, I present some examples for SSVEP studies, which illustrate the advantages and limitations of SSVEP.

i. SSVEP for control In one interesting study the authors present an SSVEP system for the *2D on-line control* of a virtual car [Bakardjian et al., 2010]. The task was to drive a virtual car on a computer screen around a 2D racing course. For the interface (depicted in Figure 3.4) 8 stimuli were placed around the car, allowing to steer the car up, down, left, right in 45° steps. The stimuli flickered in a checkerboard pattern, at frequencies of 6.0, 7.3, 8.4, 11.2, 12.9, 14, 15.3 and 16.8 Hz on a CRT screen running at 170 Hz. In contrast to most SSVEP studies the stimuli were quite small to leave enough space on the screen that the users can perceive the car and the course well. The size of the stimuli was set such that they cover the whole fovea (approximately 2.5 visual degree). Additionally, the stimuli were not stationary, but moved along with the car. The authors performed a study with four subjects. Each subject did two sessions with the car interface: one evaluation run with a predetermined path and one free run. Unfortunately, the free run is not evaluated in the paper. In the evaluation run the next direction to steer was given acoustically. Each subject did 48 trials, six per target. With an epoch length of 3.4 s they reached an accuracy of 98%.

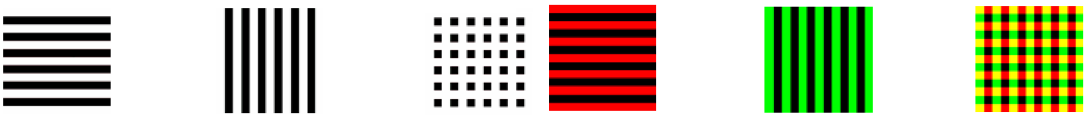
Although the interface is still rather simple the speed and accuracy are superior to all other approaches for navigation or control in BCIs using surface EEG.

ii. SSVEP for robot control SSVEP-BCIs have also been studied for the control of robotic devices [Faller et al., 2010; Gergondet et al., 2011]. In [Gergondet et al., 2011] a humanoid robot was navigated using a four command, SSVEP-based BCI. Again the user was presented with the view of the robot's camera and the stimuli were overlaid. To have a high contrast or salience the camera image was black/white again. Four red squares were used as stimuli, flickering at 5, 7, 9 and 12 Hz. The interface can be seen in Figure 3.5a. The four stimuli allowed to move the robot forward and backward and to turn it left and right. The main contribution of the paper was to compare to different interface

(a) The interface with the four stimuli and feedback that the "Forward" command was triggered.

(b) The adaptive interface with the four stimuli after the "Forward" command was triggered.

Figure 3.5.: User interface for an SSVEP-BCI controlled, humanoid robot. Figures from [Gergondet et al., 2011].



(a) Black/white condition

(b) Red/green condition

Figure 3.6.: Stimuli for the independent SSVEP-BCI [Allison et al., 2008]. The two left most pictures for each condition represent the two stimuli, in frames where both stimuli were active the rightmost picture was rendered.

modes. In the static mode each triggering of a command would move or turn the robot by a fixed amount. Even with an SSVEP-BCI this means that the robot is often standing and waiting for new commands. To have a more fluent movement the authors tried an *adaptive mode*. Two things were changed in comparison with the static mode. First, each command would change the velocity of the robot, so after the first "Forward" command the robot would start moving forward and go on until the next command. When the next command was again "Forward" it would move faster, when the next command was "Backward" it would stop. This way, it is no longer necessary to continuously activate the "Forward" command to move forward smoothly. Second, the position of the stimuli would change to have the focus on the last triggered command. This can be seen in Figure 3.5b, where the adaptive interface is shown after the "Forward" command was triggered.

To evaluate the system five users had to accomplish a simple "slalom" task. They had to navigate the robot from position A to position B while passing through one narrow "door" represented by two poles. Unfortunately, little quantitative data about the BCI performance was given. When comparing the two conditions the authors show that the adaptive interface indeed does produce much less unwanted stops. In contrast, the opportunity to accelerate was used very little, perhaps the path was to short or the users felt not confident enough to accelerate. Additionally, the adaptive interface felt kind of cluttered to some subjects. When looking at the figures, the stimuli do indeed seem quite large compared to the rest of the scene.

iii. Independent SSVEP VEP-BCIs are largely considered *dependent*. When a user wants to select a stimulus, he or she has to gaze at it. This means VEP-based BCIs are unusable for patients without oculomotor control. Some studies investigated the feasibility of independent VEP-BCIs to solve this issue [Allison et al., 2008; Kelly et al., 2005; Zhang et al., 2010]. As these studies are conducted with healthy subjects it is difficult to rule out eye movements. The central idea in [Allison et al., 2008] is to have two overlapping stimuli, both in the focus of the user. This way eye movements should not be beneficial. The user selects one of the stimuli by concentrating on it. For the presentation a CRT monitor was used, running at 60 Hz. There were three conditions, two with overlapping stimuli, one of these with black/white lines, one with red/green lines, the third condition was a control condition with two separate stimuli. The stimuli for the two independent conditions are depicted in Figure 3.6.

The two left images of both conditions are the two stimuli. The left one flickered with 10 Hz and the right one with 12 Hz. In frames where both stimuli were active, the rightmost picture was rendered. The authors did only an evaluation without classification. They compared the frequency spectrum for epochs where subjects attended the 10 Hz stimulus with epochs were subjects attended the 12 Hz stimulus. In general, there was a SSVEP signal depending on a the attended stimulus even in the overlapping conditions, but the generated SSVEP was much stronger in the control condition. In about half of the subjects the signal was strong enough to control a BCI, but with only two commands and not so good accuracy. In total, the study shows that independent SSVEP-BCIs are possible, but seem to loose their advantages compared to other BCIs, e.g. P300.

iv. SSVEP stimulation tuned to cortex structure Another study tried to boost the signal-to-noise ratio by tuning the stimuli to the structure of the primary visual cortex (V1) [Vanegas et al., 2013]. The human primary visual cortex roughly has a *cruciform* layout. This means that a simultaneous stimulation of left and right eye and upper and lower half of the retina leads to similar brain activity thus similar electrical fields but in opposite orientation. Thus, the fields tend to cancel out. The authors tried to *phase-shift* the different octants of the stimulus. They aim to stimulate the different areas of V1 with adequate delays to achieve a constructive interference. In total, they compared four conditions: A) Traditional stimulation without phase-shift. B) A fixed phase-shift with fixed octants for all subjects fitting the structure of V1. C) A fixed phase-shift with subject-specific octants. D) Subject-specific phase assignment for 32 sections of the stimulus.

For the subject-specific conditions the activity of V1 was measured using pattern-pulse multi-focal visual evoked potential (PPMVEP).

For all conditions one single large circle was flickering in a checkerboard pattern. The signal strength was measured by computing the averaged band-power at the stimulation frequency without harmonics using FFT. On average the signal strength for the three conditions with phase-shift was B) 202% C) 383% and D) 300% larger than in the control condition A. This means that the signal strength of the SSVEP response to a single stimulus can be boosted by phase-shifting different parts of the stimulus. In principle, this might lower recognition times of SSVEP-BCIs drastically, thereby enhancing speed. Due to the novelty of this study, no study yet exists trying this stimulation procedure in a real BCI setting.

3.3. Code-Modulating, Visually-Evoked Potential

SSVEP-BCIs use stimuli which flicker at different, constant frequencies. One can depict their flickering by a binary vector. 0 for frames where the stimulus is inactive (usually black) and 1 for frames where the stimulus is active. For example, for a 10 Hz flicker rendered on a CRT which runs at 60 Hz this vector would start with: 111000111000111000. In all SSVEP-BCIs the binary vector corresponding to the flickering of a stimulus contains groups of 0s and 1 of the same fixed length to generate a constant frequency. For *code-modulating, visually-evoked potentials (cVEPs)* this restriction is abolished. In principle, the visual cortex can be stimulated with arbitrary sequences. One needs one sequence (called *codebook vector*) for each stimulus. For a successful BCI, one has to chose these sequences to stimulate distinct brain activity.

3.3.1. First cVEP study by E.E. Sutter

One possibility is to generate the codebook vectors using *m-sequences*. This has been used in all cVEP studies so far since the first by Sutter [Sutter, 1992]. M-sequences are pseudo-random, binary sequences which have very low auto-correlation. To generate codebook vectors, one generates an m-sequence for the first stimulus and shifts it by a distinct amount for all the other stimuli. An example is depicted in Figure 3.7.

M-sequences have some favorable properties for the generation of the codebooks: Solomon Golomb called them “Balance property”, “Run property” and “Correlation property” [Golomb, 1981]. Balance

Figure 3.7.: Example codebook for a cVEP-BCI using four stimuli and a shift of two frames on the left. The rows are the vectors for the different stimuli. The column are the frames. 1 indicates that the stimulus is rendered in that frame. 0 that it is not. Middle and right picture show the first and fifth frame using that codebook.

property means the number of 0 and 1s is balanced. The run property extends this. It says the number of runs is balanced, where a *run* is a subsequence containing only 0s or only 1s. In an m-sequence half of the runs are of length 1, one quarter of the runs are of length 2, one eighth are of length 3 and so on. But it is their third property which is most important for the cVEP-BCIs. The "Correlation property" means that the correlation of an n bit m-sequence is n for a shift of 0 and -1 for all other shifts [Helleseth and Kumar, 1999]. M-sequences can be generated for every $n = 2^d - 1 \forall d \in \mathbb{N}$. One needs at least two frames shift between the stimuli as the lag between last and first shift is one less and must be greater zero. This means that for any number of stimuli s one computes the next power of 2 with $2^{p-1} < s \leqslant 2^p$. Then one needs $2^{p+1} - 1$ frames for one stimulation sequence with shift 2. For example, for a spelling system with 32 symbols one needs 63 frames. On a 60 Hz monitor one epoch would then take only 1.05 s.

When using many stimuli in a VEP-BCI, they are quite close to each other on the screen. This means in addition to the attended stimulus its direct neighbors also have some influence on the measured brain activity [Sutter, 1992]. When using shifted version of the same sequence for all stimuli this issue can be resolved. When the different shifts are assigned in ascending order to the stimuli as in the example in Figure 3.7, all stimuli have the same shift distance to their neighbor, which means that independent of the attended stimulus the neighbors contribute in an equal manner to the brain activity. Still, this does not hold for the stimuli at the border. Therefore, Sutter added additional "neighbor" stimuli which are never attended directly, but contribute to the activity when the border stimuli are attended. This layout with consistent neighbors is depicted in Figure 3.8a.

Of course to construct a fast BCI with this technique one needs also to be able to discriminate the evoked brain activity well. Sutter built a prototype system with special hardware and evaluated it with 60 healthy and 20 disabled subjects, but gives no quantitative results. He also tested the system with one ALS patient using intracranial electrodes. About this patient, Sutter reports that he reached an average of 10-12 words per minute and a mean access time of approximately 1.2 s. Compared to BCIs using surface EEG this is much faster. Unfortunately, most studies reviewing Sutter's work concentrate on the one reported case using intracranial electrodes [Bin et al., 2009a; Spüler et al., 2012a] and his research was not really pursued. Some work was done by K. Momose [Momose, 2007, 2008], but in terms of detection time and accuracy that work was not very successful.

3.3.2. Improvements by Bin et al.: Filtering by CCA

cVEP-BCIs regained interest after Bin et al. did a comparison of different VEP-BCIs including cVEP [Bin et al., 2009a]. In particular, they did an experiment to compare an SSVEP-BCI with a cVEP-BCI. For both conditions they used a CRT monitor running at 60 Hz. The SSVEP condition used six targets

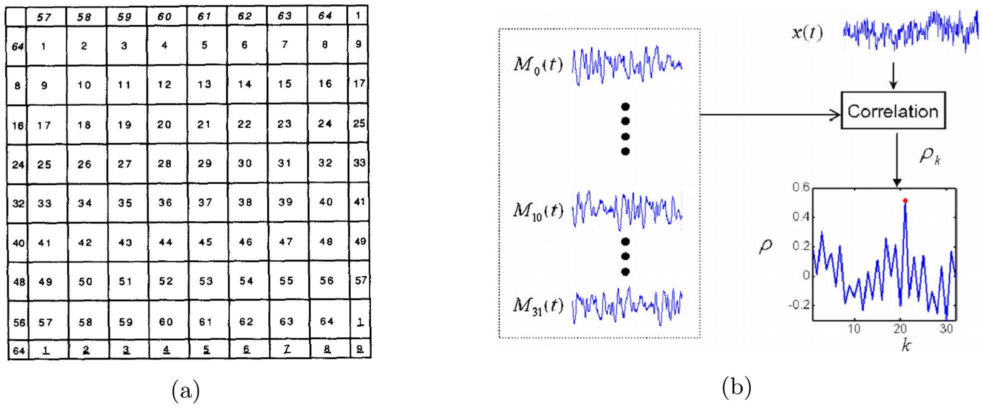


Figure 3.8.: A: Stimuli layout with consistent neighbors from [Sutter, 1992]. The circular shift in the codebook increases by four frames per stimulus. B) For classification copies with different shifts are correlated with training data. Highest correlation wins. Figure from [Bin et al., 2011]

with the frequencies 15 Hz, 12 Hz, 10 Hz, 8.6 Hz, 7.5 Hz, 6 Hz. The classification of the SSVEP data was done using a power spectral analysis using FFT, details were not given in the paper. Training data was acquired by instructing the subjects to gaze sequentially at all targets. The training data was only necessary for the channel selection.

For both conditions the best channel was used for classification. This channel was computed by testing all channels and choosing the channel with the best accuracy on the training data.

For the cVEP condition 16 targets were used with a shift of four frames meaning 63 frames per epoch (consistent neighbors were added around the target stimuli according to Sutter's layout). For the classification, training data was acquired by instructing the subject to gaze at a specific target for some time. The data was averaged and used as a template. This template was circular shifted to obtain templates for all classes. As the codebook sequence was shifted by four frames per stimulus, the EEG data has to be shifted by $(4/60 * 1000)$ ms per class. To classify new items the correlation coefficients between the new item and each template are computed. Then, as depicted in Figure 3.8b, the class of that template which has the highest correlation with the new item is chosen as the class of the new item.

Both conditions were evaluated with the same ten subjects. After the training, all subjects had to input 32 letters in each condition. Classification accuracy was high for both systems (91% for the cVEP condition, 85% for the SSVEP condition). The number of targets was much higher for the cVEP condition (16 vs. 6) resulting in a much higher Information Transfer Rate (ITR) (92.8 ± 14.1 bits/min for the cVEP system vs. 39.7 ± 7.8 bits/min for the SSVEP system).

The same group improved their cVEP-BCI two year later [Bin et al., 2011]. They adapted the spatial filtering and classification using *CCA* for their cVEP BCI. As stated above, CCA is a well known, popular mechanism for SSVEP-BCIs. It finds linear projections for two data sets to maximize the correlation between them. For SSVEP one data set is the EEG data and the other is an artificial sinusoidal curve of the stimulus frequency and its harmonics. For a cVEP-BCI this is not as easy as for SSVEP as the brain response to the cVEP stimulation is a complex non-linear transformation of the codebook [Sutter, 1992]. Thus, it is currently impossible to construct an ideal response. Instead, Bin et al. chose to construct a response with less noise by averaging over all the training data items. Suppose c is the number of channels, s is the number of samples in one epoch. The authors recorded N epochs where the subject had to look at a specific reference target k. Then, a reference template was computed by averaging over the N training epochs. For the CCA, X is the set of stacked training data items with dimensions $(N * s) \times c$. To construct Y the reference template is stacked N times resulting in a matrix with dimensions $(N * s) \times c$ as well. With these two matrices the CCA is performed which computes linear projections A and B to maximize correlation $p = \frac{A^T X Y^T B}{\sqrt{A^T X X^T A * B^T Y Y^T B}}$.

A can then be used as a linear spatial filter. By multiplying the data with A, the channels are recombined with different weights to rotate the data closer to the average. As noise cancels out in the average, the noise in one data item is reduced by rotating it closer to the average. Classification was done by computing the correlation between the spatially filtered data and the class templates, which were constructed by shifting the reference template.

For the evaluation, the authors first compared the results of their 16-target cVEP-BCI from [Bin et al., 2009a] with the same data sets spatially filtered using the CCA, which increased the average accuracy from 95% to 98% which is nearly significant ($p = 0.06$). Then, they performed a study with five subjects and two conditions. First, a 16-target cVEP BCI using a shift of four bits. Second, a 32-target cVEP BCI using a shift of only two frames, both with the CCA and classification by comparing correlation coefficients. Each subject did about 200 epochs in the trainings phase and 64 letters with on-line feedback in the evaluation phase. Although the accuracy of the 32-target system was slightly lower ($85 \pm 0.05\%$ accuracy in the evaluation of the 32-target system vs. $92 \pm 0.03\%$ for the 16-target system), the ITR was higher for the 32-target system (108 ± 12 bits/min vs. 96 ± 6.3 bits/min).

3.3.3. The Work of Spüler et al.: Classification by OCSVM

Classification was further improved by the Tübingen group [Spüler et al., 2012a]. They implemented two enhancements on top of the work by Bin et al: Namely, preselection of channels to be used for the CCA template and the classification by a *One-Class SVM* instead of simple comparison of the correlation.

The important task for the CCA is the construction of the *template* Y. As done by Bin et al., Spüler et al. use the average of the training data as template, but they use only certain channels[2]. In total, four alternatives for the construction of Y were implemented: A) The best overall channel is computed on the training data and used as the only channel for the CCA template. All channels are used for X B) The same channels as in [Bin et al., 2011] were used for X and Y C) The best channel per subject is derived on the training data and used as Y. All channels are used for X. D) Each channel is tested on the training data per subject. All channels reaching an accuracy $p_c \geq 0.9 \cdot p_{best}$ are used for Y. p_{best} is the accuracy of the best channel. Again, all channels are used for X.

The classification is further improved by using a One-Class SVM (OCSVM). I present the general idea here. The mathematical details of this algorithm will be examined in chapter 4. A one-class SVM decides if a new data item fits to the items that were used for training the SVM. For the classification in the cVEP-BCI, several training epochs, where the subject attended the reference target, are used to train the OCSVM. Traditionally, one computed the correlation of the data item with the templates for all classes and these templates were shifted versions of the reference template. When using the OCSVM, the new data item is copied once per class and each copy is shifted backwards corresponding to the shift of the class. Then, the different copies are classified by the OCSVM. The copy with the smallest distance to the training data as computed by the OCSVM is chosen. By its shift the class of the new item can be derived. The advantage of the OCSVM is that different dimensions of the input are weighted according to their discriminative capabilities.

For the evaluation, Spüler et al. first did an off-line study with eight subjects. Each subject attended each target 20 times resulting in 640 trials. A ten-fold cross-validation was done to compute the classification accuracy. To obtain the reference template, the training data items were shifted backwards according to their class to obtain 64 epochs of class 0. The different spatial filters resulted in the following average accuracies when using the OCSVM for classification: A)85.80% B) 92.32% C) 93.32% D) 96.29%. Classification by correlation was also tried and reached an accuracy of 89.90% with spatial filter method B). The authors also did an on-line proof of concept with one subject. This subject reached an average accuracy of 92.71%. One epoch took 1.05 s and between the epochs there was a pause of 0.85 s to display feedback and allow the subject to switch attention to the next letter. This results in an ITR of 133.6 bits/min which makes this the *fastest BCI speller* using surface EEG.

These results could be confirmed in a second study with more subjects [Spüler et al., 2012b]. For this second study they used the same spelling system setting as in the first. Spatial filtering method C was applied, i.e. choosing the subject-specific best channel for the construction of the template.

[2] For the CCA the number of observations (the number of rows) of X and Y have to be the same, but the number of variables (the number of columns) may vary.

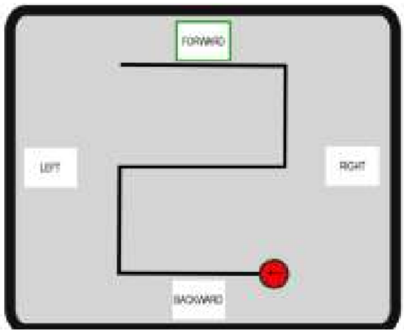


Figure 3.9: User interface for navigation of a robot using a cVEP-BCI from [Kapeller et al., 2013]. The four rectangles around the scene are the stimuli, the black line is the target path and the red dot symbolizes the robot.

For the classification they used an OCSVM again and tested different adaption mechanisms. The EEG data changes over time during the session and between sessions due to changing impedance, small changes in electrode locations and changes in the brain activity. Thus, the classification system should adapt itself to these changes. An example for adaption in an ERD-based BCI can be found in [Riechmann and Finke, 2012]. Spüler et al did the first study about classifier adaption for cVEP systems. Basically, their adaption system guesses the class labels for new data items and retrains the whole classification system including the CCA and the OCSVM periodically, including the new items. They tried two approaches to guess the labels. First, they just used the label the new item got from the current classifier. Second, they analyzed the data for so-called Error Potentials (ErrP). After the subject got feedback, a potential similar to a P300 can be found in the EEG when the feedback shows that the classifier chose the wrong class. This can be classified as well and when that classifier finds an ErrP the last trial is repeated and not added to the training data set for the next retraining. They found that both adaption mechanisms improve accuracy. The unsupervised adaption is more suitable for subjects who have a high accuracy from the beginning, the ErrP-based adaption is more suitable for subjects who have a not so good accuracy.

For the evaluation a study with nine subjects was performed. Data was recorded from 30 electrodes. In copy-spelling mode (spelling a predefined sentence) an average accuracy of 96% was reached, leading to an average ITR of 144 bits/min, which is even faster than their first system and much faster than BCIs for spelling which are not based on VEPs. Additionally, they also did a free spelling session with some subjects and the accuracy dropped a little, probably because the subjects sometimes failed to find the correct stimulus in time before the next trial was started.

3.3.4. cVEP for Robot Control

At the same time as I did my studies, gTec, the company behind the gUSBAmp amplifier, also did an experiment about a navigation and control application using a cVEP-BCI [Kapeller et al., 2013]. In their study, subjects had to navigate a small, wheeled robot along a predefined path. The subjects saw the robot and the path from above through a camera on their screen. There were four stimuli overlaid over the frames from the camera, see Figure 3.9. The four stimuli allowed to drive the robot forward and backward and to turn right and left. They compared the performance of a SSVEP and a cVEP approach. For the SSVEP approach they used a steady flickering at 8.57 Hz, 10 Hz, 12 Hz and 15 Hz for stimulation and a Minimum Energy algorithm for spatial filtering. After the spatial filtering of the EEG data they estimated the signal-to-noise ratio and classified these ratios by Linear Discriminate Analysis. One epoch lasted 2 s. For the cVEP an m-sequence of 63 frames was used for stimulation (resulting in epochs of 1.05 s length), CCA for spatial filtering and LDA for the classification of the correlation coefficients. To allow asynchronous control of the robot, a fifth class was implemented as a "no operation“ command. This was done by transforming the LDA output into probabilities and only issuing commands when the highest probability was above a predefined threshold.

For the evaluation data of eleven subjects was acquired, using eight electrodes. After some training, data was acquired to train the two classification systems, each subject did an accuracy evaluation and the robot control scenario. For the accuracy test, each subject did 20 trials. In each trial one of the four stimuli was highlighted as target using a small green border. Without the zero class an average accuracy of 94.51% for the c-VEP BCI and 84.18% for the SSVEP BCI was reached. Adding the zero class decreased the accuracy by 20 to 30%. In the robot control scenario each subject drove the robot three times along the path, which is depicted in Figure 3.9. One time with each of the two BCI conditions and another time using a keyboard control. The data of four subjects was excluded either because their results differed more than two standard deviations from the rest or because they could not complete the task. On average the remaining seven subjects needed 93 s using the keyboard, 573 s using the SSVEP system and 223 s using the cVEP system.

This result indicates that the cVEP system is better suited for this simple navigation scenario then an SSVEP system, both in terms of accuracy and speed. Interestingly, both conditions suffered a huge accuracy drop when adding the zero class, this might be because of a non-optimal implementation, but it might also indicate that the detection of the "no control" state is particularly difficult in VEP-BCIs. In total, the system shows some hints for the feasibility of controlling a robotic devices with a cVEP-BCI, but the scenario is very simple and it remains to be seen how the results can be transferred to more complex tasks. Especially, as there are few stimuli, which are also quite large and relatively far away from the possibly distracting scene.

3.4. Comparison with Other BCIs

Here, I compare the VEP-based BCIs with other BCI approaches, with a focus on applications for navigation and control of robotic devices. Currently, three types of BCIs are popular in research about how to control a robotic device with a BCI, namely BCIs using *P300*, BCI using *sensorimotor rhythms (SMR)* and *VEP* BCIs. All three have different strength and weaknesses, but in my opinion there are good reasons why one should do research on using VEP-BCIs for this control task.

i. Sensorimotor rhythms BCIs using sensorimotor rhythms have the advantage that they are *completely voluntary*, thus no stimuli are needed. This means that the system is much simpler and more convenient for the user as no device for the stimulus presentation is needed. Additionally, these type of BCIs can be used asynchronously quite easily. A lot of work on navigation and control has been done by the Lausanne group with their semi-autonomous wheelchair [Carlson and Millán, 2013]. As presented in the last section, their wheelchair is equipped with several sensors, but looks quite ordinary otherwise, e.g. no bulky monitor. But their work also shows the limitations of SMR-BCIs. The number of distinct commands is very small, typically two or three. This allows to indicate the desired direction to the wheelchair and, combined with the autonomous intelligence of the wheelchair, a suitable control. For more complex scenarios where a robot might be desired to fetch something, the limited number of command cannot be resolved well. To choose one out of sixteen objects, one would need four binary decisions and the probability for a misclassification cumulates. An SMR-BCI with two commands and a binary accuracy of 80% would only have an accuracy of 41% for such a decision. Additionally, that decision would take a lot of time. Therefore, I think SMR-BCIs are not well suited for the control of robotic devices.

ii. P300 Currently, the preferred option for BCI control of robotic devices is the P300 potential. The possible number of commands is high, i.e. 30 different commands is no problem. Additionally, the classification is very stable across subjects, especially compared to SMR-BCIs. There are few subjects who cannot produce a good signal. Finally, the P300 is quite a late component, something one might call high level brain activity. After the sensory information is processed the P300 is elicited if the event is recognized as target event. Therefore, the characteristics of the stimulus do influence the latency and amplitude only little. This allows to use the P300 in a wide range of applications. The stimulus presentation can be tuned to the demands of the applications without much concern about the signal strength.

In the first section of this chapter, I presented a study about a BCI-controlled telepresence robot [Escolano et al., 2012]. The authors placed stimuli within the scene on positions where the robot could drive to, without considering contrast or size of the stimuli and got a signal strength comparable to P300 spelling systems where the bare stimuli are placed on a black background. In another interesting study a P300-BCI was used to control an avatar in a virtual apartment [Bayliss, 2003]. Stimuli were placed in the scene at some objects the avatar could interact with, like television, radio and so on.

They specifically designed the stimuli to be unobtrusive for the user. They also examined whether there is a difference between P300 amplitude and shape when subjects were seated in front of a monitor or perceived the scene in a Virtual Reality setting using a Head-Mounted Display (HMD). 9 subjects participated in the study, each did one 5 min session in front of a 21 inch CRT monitor and one using an HMD. The grand average waveform showed no significant difference. Additionally, there was no significant difference in the number of repetitions needed to activate the different objects.

While being very popular among researchers P300-BCIs also have one considerable disadvantage. They are *slow*. First, because P300 is a late component, the response from the brain itself is quite slow (roughly 300 ms). Second, the stimuli have to be activated one after the other. The information is purely contained in the point of time when the P300 happens, which means one cannot activate more than one stimulus at a time. Third, several repetitions are necessary to get an acceptable accuracy. For example in the telepresence system only four selections per minute could be carried out. In experiments of our own group we needed less repetitions, but still one trial takes $6 - 10$ s [Riechmann et al., 2011].

iii. VEP Visually-evoked potentials are early responses, or low-level brain activity. The reaction to the stimulus starts about 80 ms after stimulation begin [Sutter, 1992]. Additionally, it is possible to use different stimuli, for example different flickering frequencies simultaneously. Finally, the signal is quite strong and can be classified without repetitions. This allows to build BCI systems which are much *faster* than other BCIs. 30 and more stimuli can be discriminated well using only a single epoch of 1 s. Spelling system using VEP-based BCI therefore allow for quite fluent spelling [Spüler et al., 2012b; Volosyak, 2011].

Still, the signal strength in VEP-BCIs is highly dependent on stimulus characteristics such as size or contrast. For spelling systems this is not a problem, the stimuli characteristics are tuned for optimal performance. For navigation and control tasks it imposes a dilemma. One has to balance between tweaking the user interface for usability and signal strength.

Most studies so far concentrated more on getting a good signal strength. For example, the interface in [Gergondet et al., 2011] was only black-and-white to maximize contrast. There were also only four stimuli for navigate the robot and still these stimuli also occluded most of the scene.

In other studies, the scene and the stimuli get displayed on the same screen, but are located in separated areas [Kapeller et al., 2012, 2013]. Again, this allows to maximize contrast and increase the size of the stimuli, but also uses up much of the visual space. Additionally, this forces the user to look back-and-forth between scene and stimuli often. Finally, these studies also use only few stimuli.

Another disadvantage of VEP-BCIs is their dependence on *gaze control*, this will be discussed in the next subsection.

In total, VEPs allow for very fast BCIs, which could even be useful for healthy users, but so far little research exists for navigation and control tasks using VEP-BCIs. Additionally, the existing studies do not use the advantage of having many commands and improving user experience by integrating scene and stimuli.

3.5. Comparison with Eye-Tracking

BCIs using a visually-evoked potential are considered to be dependent BCIs, because they require the subjects to have oculomotor control. I discussed an SSVEP-BCI which did not need oculomotor control, but it remains to be seen to what extent the advantages of VEPs can be preserved. For patients still having oculomotor control, systems tracking the eye-gaze could be an alternative to a BCI. In fact, E. E. Sutter originally invented cVEP-BCIs as an alternative to eye-tracking interfaces [Sutter, 1992].

In the spelling domain eye-tracking systems are currently roughly a factor two faster [Spüler et al., 2012b; Urbina and Huckauf, 2007]. For navigation and control this is not clear currently. Eye-tracking systems have been built for navigation in virtual environments [Stellmach and Dachselt, 2012] or to control a first-person shooter games [Sko et al., 2013], but these studies give little quantitative data about reliability and speed, mostly they evaluated user experience.

So, research on VEP-BCIs is valuable. There are subjects who have problems with eye-tracking systems, due to glasses or excessive head movements [Sutter, 1992]. Then, eye-tracking systems usually require specific light conditions and it is not clear in how far eye-tracking systems depend on stimulus characteristics. Additionally, research on VEP-BCIs especially on cVEP-BCIs is only at the beginning. Their speed might increase to be comparable with eye-tracking system or they might prove to be independent in the end. Finally, one might also integrate both, for example to boost accuracy as done for P300 and eye-tracking in [Momose, 2008].

4. Prerequisites for the cVEP-BCI

This chapter deals with the most central methods I used in my experiments and explains them in detail. The first part is dedicated to the signal processing methods used for the off- and on-line analyses of the EEG data. The second section deals with the software framework used for the data processing. In the last section common parts of the experimental set-up, like the EEG data acquisition, are presented.

4.1. Signal Processing

In this section, I cover the algorithms used in the signal processing of the recorded EEG data. Signal processing has been a major topic in BCI research since the beginning and today a myriad of approaches exist. Typically, there are three to four steps in signal processing for BCIs: *Preprocessing*, in particular band-pass filtering and sometimes artifact rejection or removal, *feature extraction* usually combined with dimensionality reduction and finally *classification*, sometimes expanded by a *post-processing* step. A review of even the most popular approaches would exceed the scope of this work. Therefore, I focus on the algorithms used in my experiments and the reasons why they were used.

I start with a short discussion on *band-pass filters* for cVEP-BCIs, then I describe *Canonical Covariance Analysis* and its use for feature extraction. Next, I present *SVM* and *LDA* and how they are used for classification in my experiments. Finally, I discuss dynamic repetitions.

4.1.1. Band-Pass Filter

As a first step in signal processing, EEG data is usually band-pass filtered, see e.g. [Wolpaw et al., 2002; Wang et al., 2006].

For cVEP-BCIs no narrow frequency band of interest can be defined. Due to the complex code-modulating of the stimulus and the non-linear response of the visual cortex, a wide frequency band contains the information.

Some band-pass filtering has to be done, though. There are at least two filters used in nearly all BCI recordings including cVEP. As the main power lines in the lab emit strong electrical fields (magnitudes stronger than anything which can be picked up from the brain) their frequency (around 50 Hz in Europe) must be suppressed by a notch filter. The second filter is a high-pass filter at a low frequency to compensate the slow drift. Over time (in range of several minutes) the EEG slowly drifts which distorts the data. A high-pass filter at 0.1 Hz or 1 Hz keeps the data always zero-centered thereby removing the drift. One can safely assume that below 1 Hz there is no usable information as for cVEP one epoch takes around 1 s which is less than frequencies below 1 Hz need for one cycle.

A bit more on this topic can be found in [Spüler et al., 2012b]. There, they used the notch filter at 50 Hz and the high-pass filter (at 0.5 Hz) plus an additional low-pass filter at 60 Hz. Still, preliminary tests showed that the low-pass filter does not increase signal strength, thus it is omitted in my work.

4.1.2. Feature Extraction by Canonical Correlation Analysis

Canonical Correlation Analysis (CCA) has proven quite successful as feature extraction tool for cVEP-BCIs. In fact, it has been used most studies with cVEP-BCIs in the last years. I use it as well for my experiments, therefore it is explained in detail here.

4.1.2.1. Mathematical Foundation

CCA maximizes the *linear correlation* between two sets of variables X and Y for the same observations. Let the dimensions of the matrices be $X : n \times k, Y : n \times l$. This means every row is one observation or one data sample and each column is one measured variable. For the variables x_i and y_j the linear correlation can be computed [Weenink, 2003]:

$$p_{ij} = \frac{\Sigma_{ij}}{\sqrt{\Sigma_{ii} * \Sigma_{jj}}} \tag{4.1}$$

where Σ_{ij} is the covariance between x_i and y_j:

$$\Sigma_{ij} = \frac{1}{n-1} \sum_{a=1}^{n} (X_{ai} - \mu_i)(Y_{aj} - \mu_j) \tag{4.2}$$

and μ_i and μ_j are the averages. One can compute this correlation coefficient for any i and j, thus construct a correlation matrix.

The values in this correlation matrix depend very much on the coordinate system used, any rotation on X and Y changes the values of the matrix. For example, rotating the data onto its principal components would result in correlations equal zero for any i, j with $i \neq j$.

In contrast to the pure correlation coefficient, which deals with the correlation between two variables, CCA deals with the correlation between two sets of variables. The goal of the CCA is to find projections A_1 and B_1 to maximize this correlation:

$$argmax_{A_1,B_1} \quad p = \frac{A_1^T X^T Y B_1}{\sqrt{A_1^T X^T X A_1 \cdot B_1^T Y^T Y B_1}} \tag{4.3}$$

A_1 and B_1 have the dimension $k \times 1$ respectively $l \times 1$ and are called a canonical function, $A_1^T X^T$ and YB_1 are the canonical variates. Several canonical functions can be computed. The relationship of the two sets that is covered by the previous canonical functions can be removed and the procedure is repeated to get the next canonical function. At most $min(rank(X), rank(Y))$ canonical functions can be derived. By design all these canonical functions are orthogonal and (linearly) statistically independent.

4.1.2.2. Computation of the CCA

Several approaches for the computation of the CCA exist. The most common is the algorithm of Golub [Björck and Golub, 1973], which is briefly explained here. This section is based on [Press, 2011].

The key idea of the algorithm is to not compute the maximization in equation 4.3 directly.

Let $d = min(rank(X), rank(Y))$ be the maximal number of canonical functions which can be computed and $A_1, B_1, ..., A_d, B_d$ the different canonical functions and $p_1, ..., p_d$ the corresponding correlation coefficients, with $p_1 > p_2 > ... > p_d$. Additionally, we denote $U_1^T = A_1^T X^T, V_1 = YB_1$ and $U_2, V_2, ..., U_d, V_d$ respectively. Finally, we denote U, V the matrices with the columns $U_1, ..., U_d$ and $V_1, ...V_d$ respectively.

Now, consider the correlation between U_1 and any V_j with $j \neq 1$. Assuming one of these correlations is non-zero, a weighted linear projection of V_1 and this V_j would exist which would have a higher correlation with U_1 than V_1. Such projection must not exist therefore the correlations of U_1 and any V_j with $j \neq 1$ are zero and by extension of this argument: $corr(U_i, V_j) = 0$ for any $i \neq j$. Thus:

$$U^T V = D \tag{4.4}$$

where D is a diagonal $d \times d$ matrix.

Next, we can scale U and V w.l.o.g. to ensure:

$$U^T U = I, V^T V = I \tag{4.5}$$

where I is the $d \times d$ identity-matrix.

In the case where X and Y have the same number of columns and maximal rank, we can formulate the following uniqueness argument:

The last two equation give $d^2 - d$ and $2 \cdot d(d+1)/2$ constraints, so we have $2d^2$ constraints and the two matrices A and B have $2d^2$ degrees of freedom. This means that any solution A^*, B^* satisfying equations 4.4 and 4.5 is unique (ignoring column permutations).

Thus, one can solve equations 4.4 and 4.5 to compute the canonical functions which satisfy the maximization in equation 4.3:

First, X and Y are QR-factorized:

$$X = Q_X \cdot R_X, Y = Q_Y \cdot R_Y \tag{4.6}$$

where Q_X, Q_Y are column orthogonal and R_X, R_Y are upper triangular matrices. Now, $Q_X^T \cdot Q_Y$ is computed and SVD-factorized:

$$Q_X^T \cdot Q_Y = LSM^T \tag{4.7}$$

where S is diagonal and contains the eigenvalues of $Q_X^T \cdot Q_Y$ and L, M are column orthogonal. Next we define:

$$A = R_X^{-1} \cdot L, B = R_Y^{-1} \cdot M \tag{4.8}$$

and thats it, A and B are the complete canonical functions.

To show this, we check that they solve the equations 4.4 and 4.5:

$$\begin{aligned}
D &= U^T V \\
&= A^T X^T Y B \\
&= (L^T R_X^{-1T})(R_X^T Q_X^T)(Q_Y R_Y)(R_Y^{-1} M) \\
&= L^T (R_X^{-1T} R_X^T)(Q_X^T Q_Y)(R_Y R_Y^{-1}) M \\
&= L^T (Q_X^T Q_Y) M \\
&= L^T (LSM^T) M \\
&= (L^T L) S (M^T M) \\
&= S
\end{aligned}$$

As S is diagonal by construction, D is diagonal as well and equation 4.4 fulfilled.

Next, we check equation 4.5:

$$\begin{aligned}
U^T U &= A^T X^T X A \\
&= (L^T R_X^{-1T})(R_X^T Q_X^T)(Q_X R_X)(R_X^{-1} L) \\
&= L^T (R_X^{-1T} R_X^T)(Q_X^T Q_X)(R_X R_X^{-1}) L \\
&= L^T L \\
&= I
\end{aligned}$$

A corresponding argument can be given to show that $V^T V = I$. So, the proposed solution also fulfills equation 4.5 and as a result of the uniqueness argument this solution also maximizes the correlation.

In the case where X and Y do not have maximal rank or not the same number of columns the uniqueness does not hold. In this case, one takes only those columns of L and M for the computation of A and B which correspond the highest eigenvalues of $Q_X^T Q_Y$. The prove that this maximizes the correlation becomes more complex, but that proof is beyond the scope of this work. In fact, in my experiments X and Y do have the same number of columns and maximal rank.

4.1.2.3. CCA in Statistics

In statistics Canonical Correlation Analysis is a well known tool for several years [Thompson, 2005]. It is used whenever one has more than one dependent and also more than one independent variable and both sets contain variables which are metric. In those cases Canonical Correlation Analysis is the only tool to analyze the relation between the groups simultaneously. For other cases CCA is not so popular as other methods for analyses make stronger assumptions and tend to produce stronger results [Thompson, 2005].

4.1.2.4. CCA for BCIs

CCA is popular as a feature extraction technique for VEP-BCIs for some time now. Originally, it was used as a spatial filter for SSVEP-BCIs, see also section 3.2.2 p. 31f. For SSVEP-BCIs one can easily built an approximate ideal response of the brain for each stimulus. This ideal response typically is a combination of sine and cosine functions with the frequency of a stimulus and its harmonics. Their amplitude does not matter for the correlation. Then, one can compute the CCA between the EEG data and the ideal response and compute the resulting correlation coefficients for the different stimuli. For classification, the coefficients are compared and the class corresponding to the highest coefficient is chosen.

CCA is also used for cVEP-BCIs as introduced in section 3.3 p. 35ff. For the cVEP an ideal response cannot be constructed easily. Because of the more complex stimulation pattern the brain response is also more complex. So far, one just uses the average over epochs as a reference [Spüler et al., 2012a; Bin et al., 2009a]. So, to use CCA for cVEP let X be the trainings epochs stacked along the time dimension. For n trainings epochs with s samples and c channels X has the dimensions $(n*s)\times c$. The reference or template Y is built using the average over all trainings samples. This average is stacked n-times to get the suitable number of observations. The number of channels for the template can be varied, usually a subset of the recording channels is selected to maximize accuracy [Spüler et al., 2012a]. When the CCA has been performed, one can built a spatial filter by stacking all projections A of the resulting canonical functions.

Essentially, the resulting spatial filter rotates a given item in such a manner that its correlation with the average of all training data items is enhanced. To understand the motivation behind this, one must consider the classification as well. In the work by Bin et al., classification was based on correlation. For each class one copy of the new item was created and shifted with the shift factor of that class. The copy which has the highest correlation with the average of the training data is chosen as label for the new item. When the CCA is applied as spatial filter, this correlation is increased more for the correct copy as for the incorrect copies, thus enhancing accuracy.

4.1.2.5. CCA in my Experiments

For my experiments, CCA is used as *spatial filter* similar as in [Spüler et al., 2012a], with the distinction that I do not use any of the channel selection strategies presented there. I perform the CCA with all channels, because I recorded only 10 channels in my experiments.

The MATLAB function *canoncorr* is used for the computation of the CCA. It computes the two complete canonical functions A and B using the Golub algorithm described above. A is then used as spatial filter by matrix multiplying it with each incoming EEG epoch.

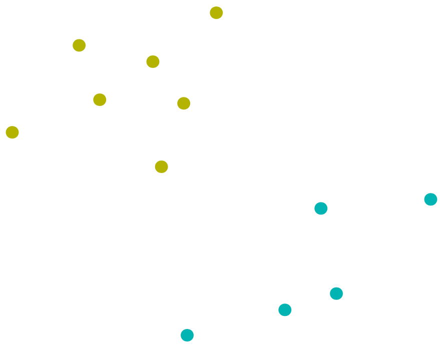
4.1.3. Classification by Support-Vector Machine

The classification of the EEG data has always been a major topic in research on BCIs. Several approaches have been examined. In general, linear methods have been more successful then non-linear methods, because non-linear methods tend to suffer from over-fitting [Manyakov et al., 2011]. For linear methods Linear Discriminant Analysis is the most popular approach. Its use for BCIs will be discussed in the next subsection. Additionally, *linear Support Vector Machines (SVMs)* are also used [Manyakov et al., 2011; Spüler et al., 2012a]. For non-linear methods, SVMs using Gaussian kernels are quite popular. As a linear SVM is used in [Spüler et al., 2012a,b] and my own experiments I will cover SVMs in detail.

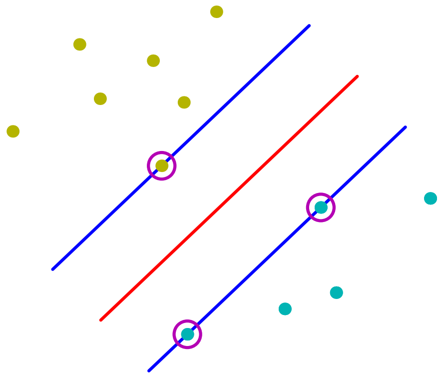
Here, I present the initial concept and implementation of SVMs first, then I present some additions, the 'kernel-trick' and the idea of slack-variables. Next, I discuss the usage of SVMs for BCIs and finally I introduce the One-Class SVM and its usage as classifier in cVEP-BCIs. The parts about the concept and the implementation are based on [Bishop et al., 2006, chapter 7].

4.1.3.1. Concept

To illustrate the concept, I consider the following simple classification problem. There are data samples with class labels i.e. $(x_i, t_i) \in D$ with $x_i \in \mathbb{R}^2, t_i \in \{-1, 1\}$, i.e the data samples are from two different



(a) A 2-class problem which is linear separable.



(b) Red line: Classification boundary with maximal margin. Violet circles: Support vectors, i.e. vector with minimal distance to decision hyperplane.

Figure 4.1.: Example scenario for classification by Support Vector Machine. Figures from [Bishop et al., 2006, chapter 7], adapted.

classes and have two dimensions. The data samples are to be classified by a linear classifier, i.e a classifier of the form

$$y(x_i) = w^T x_i + b \tag{4.9}$$

where x_i is a data sample, $y(x_i)$ is the class label output by the classifier and w and b describe the hyperplane, which is used as decision boundary. It is assumed that the data samples are linearly separable, i.e there exists at least one linear classifier which can classify the training samples without error. The example situation is depicted in Figure 4.1a. From the precondition that at least one linear classifier can separate the training samples without error, it directly follows that an infinite number of possible linear decision boundaries would correctly separate the two sets of training samples. One has to make further assumptions to chose one of them.

The idea is to find the decision boundary which *generalizes best*, i.e. whose boundary classifies new data, not seen during the training, best. For example, Linear Discriminant Analysis(LDA) assumes that both classes follow a Gaussian distribution and places the decision boundary in such way to separate the two assumed Gaussian distributions best. For Support Vector Machines one assumes, that the best classifier is the one that *maximizes the margin*, where the margin is the minimal distance of any data sample to the decision boundary. This means SVM computes the decision boundary that has the maximal distance to those samples which lie minimally close to the border. One can see that the position and direction of the classification border then only depends on those points with minimal distance to the classification border. These points are called *support vectors*. The example situation with classifier and support vectors is depicted in Figure 4.1b.

4.1.3.2. Implementation

The distance of a point x to the decision boundary can be expressed as:

$$\frac{|y(x)|}{\parallel w \parallel} \tag{4.10}$$

Because we assume that all data samples are classified correctly and the classifier is linear, this can be transformed:

$$\frac{t_i y(x_i)}{\parallel w \parallel} = \frac{t_i(w^T x_i + b)}{\parallel w \parallel} \tag{4.11}$$

To find w and b which maximize the margin, one has to solve:

$$argmax_{w,b}(\frac{1}{\parallel w \parallel} min_i t_i(w^T x_i + b)) \tag{4.12}$$

A direct solution of this optimization problem is not feasible.

To transform this problem into a easier-to-solve one, one must observe that rescaling w and b by the same scalar s, $s \neq 0$ does not change the distances of the points to the decision boundary, thus also not the margin. Without loss of generality one can thus define that for the points closest to the separating hyperplane the following holds:

$$t_i(w^T x_i + b) = 1 \tag{4.13}$$

This means that for all data samples it holds that:

$$t_i(w^T x_i + b) \geq 1 \tag{4.14}$$

To maximize the margin, one now needs only to maximize $\frac{1}{\|w\|}$ with the constraints from equation 4.14. This is equivalent to minimize $\parallel w \parallel^2$ with the constraints, thus the optimization problem to solve is now:

$$argmin_{w,b} \frac{1}{2} \parallel w \parallel^2 \tag{4.15}$$

again with above constraints. The factor $\frac{1}{2}$ was added for later convenience. Although b is not directly part of the optimization function it is implicitly defined through the constraints. The corresponding Lagrange function is:

$$L(w,b,a) = \frac{1}{2} \parallel w \parallel^2 - \sum_{i=1}^{N} a_i(t_i(w^T x_i + b) - 1) \tag{4.16}$$

where $a = (a_1, ...a_n)$ are the Lagrange multipliers. Setting the corresponding derivatives equal to zero gives the following conditions:

$$w = \sum_{i=1}^{N} a_i t_i x_i \tag{4.17}$$

$$0 = \sum_{i=1}^{N} a_i t_i \tag{4.18}$$

Inserting these conditions into the Lagrange function leads to:

$$L^{'}(a) = \sum_{i=1}^{N} a_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} a_i a_j t_i t_j x_i x_j \tag{4.19}$$

which has to be maximized with the constraints:

$$a_i \geq 0 \tag{4.20}$$

$$\sum_{i=1}^{N} a_i t_i = 0 \tag{4.21}$$

Solving this quadratic optimization problem leads to $a_i > 0$ for all data samples with minimal distance to the hyperplane, the *support vectors*, and $a_i = 0$ for all other data samples. To minimize space and time requirements one usually adopts a technique called Sequential Minimal Optimization (SMO) [Platt, 1999], where only two Lagrange multipliers at a time are optimized. This can be solved analytically.

4.1.3.3. Not-Linearly Separable Data

Still, for most classification tasks the data is not linearly separable and the above procedure would not converge to any decision boundary. One approach to solve this issue is to transfer the data to a *(usually higher dimensional) feature space* where it is linearly separable. For this, one constructs a non-linear transformation function $\Phi(x)$ and uses $\Phi(x)$ during training and classification. SVMs are particularly suitable for this approach because the input data samples are not directly needed during the computation of the decision boundary. As can be seen in Equation 4.19 only the scalar product between the samples is needed. This is called the "Kernel" trick. For the remainder of this section, we define $k(x_i, x_j)$ as the scalar product of x_i and x_j. For many high dimensional feature spaces the scalar product between two vectors in the feature space can be handled much easier than the vectors themselves. One particular example is the feature space induced by the Gaussian Radial Basis Function, where $k(x_i, x_j) = e^{\frac{-(\| x_i - x_j \|)^2}{2\sigma^2}}$ [Vert et al., 2004, p. 7]. This is also called the Gaussian kernel. It can be shown that the corresponding feature space has infinite dimension. Still, the scalar product depends only the distance in the input space and some predefined σ[1]. Using a sufficiently small σ all classification problems become separable[2].

However, due to noisy input data it is often not desirable to choose a decision boundary that classifies all training samples correctly. Usually this leads to severe *over-fitting* and poor generalization. Therefore, one needs an SVM model that allows to tolerate a certain amount of misclassification of the training samples. This is done by introducing *slack variables* ξ_i, $i = 1, ..., N$ with $\xi_i \geq 0$. They are defined as $\xi_i = 0$ for correct data points, i.e. points that satisfy equation 4.14 and $\xi_i = |t_i - y(x_i)|$ for incorrect points, i.e. points which are misclassified or lie too close to the decision hyperplane. The hard margin constraints are then replaced by, so called *soft margins*:

$$t_i(w^T x_i + b) \geq 1 - \xi_i \tag{4.22}$$

which leads to the optimization problem:

$$argmin_{w,b} C \sum_{i=1}^{N} \xi_i + \frac{1}{2} \parallel w \parallel^2 \tag{4.23}$$

where $C > 0$ is called the cost parameter and balances the influence of the slack variables. Using again a Lagrange approach, one obtains this optimization problem:

$$L^{'} = \sum_{i=1}^{N} a_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} a_i a_j t_i t_j x_i x_j \tag{4.24}$$

with the constraints:

$$C \geq a_i \quad \geq 0 \tag{4.25}$$

$$\sum_{i=1}^{N} a_i t_i \quad = 0 \tag{4.26}$$

which is the same problem as without the slack variables except the first group of constraints where the cost parameter limits the Lagrange multipliers. As the Lagrange multipliers define the influence of their corresponding data sample on the decision hyperplane, C, in effect, limits the influence of single data samples on the decision hyperplane.

[1] Essentially, when using a Gaussian kernel, the probability distribution of the class likelihood is described by a sum of Gaussian distributions. One Gaussian distribution is placed at each support vector and all these Gaussian distributions have the standard deviation given by σ.

[2] Except of course for cases where two data points with different labels have the same coordinates, but these cases have 0 probability when the samples are drawn from some underlying distribution.

4.1.3.4. SVM for BCIs

SVMs have been used in BCI research in quite some studies [Kaper et al., 2004; Lotte et al., 2007; Jrad et al., 2011; Yeh et al., 2013; De Massari et al., 2013; Resalat and Setarehdan, 2012; Resalat et al., 2012; Dehzangi et al., 2013; Sakurada et al., 2013]. An interesting comparison was done already 2005 in [Lee et al., 2005]. They compared three classifiers: LDA, linear SVM, and Nearest Neighbors Classifier (NCC) on different features of EEG data, namely feature extraction using Adaptive Auto Regression (AAR) and feature extraction plus dimension reduction by Common Spatial Pattern (CSP). They show that the linear SVM provides better results on the high-dimensional AAR features, but LDA performs slightly better on the low dimensional CSP features. Both outperform NCC in general. In total, the classification using CSP clearly outperforms the AAR features with means the combination of CSP plus LDA achieves the highest accuracy.

Their results confirm a general trend in BCI research. When the EEG features are well understood and can be modeled well or transformed into a low-dimensional feature space LDA often performs best. When high-dimensional data with little or no preprocessing has to be classified directly SVM often performs better. This is consistent with pattern recognition theory where SVM is considered to handle situation with many dimensions, but little training data quite well. This makes it an interesting tool for the analyses of cVEP data, because we cannot (yet) model the response of the visual cortex.

4.1.3.5. One-Class SVM

A classifier generally finds a decision boundary to separate two or more classes, but there are also *one-class classifiers*. "A one-class-classifier attempts to find a separating boundary between a data set and the rest of the feature space."[Roth, 2004]. They are usually used for *novelty* or *outlier* detection. A *One-Class SVM (OCSVM)* is a variation of the SVM, proposed in [Schölkopf et al., 2001]. The central idea is to learn a hyper-sphere which includes a given percentage of the data. More precisely, for a given probability distribution P in some input space I one searches for a subspace S of I, such that a data point drawn from P lies within S with an a-priori specified probability of p. The OCSVM learns a function f defined on I, which has positive values for points within S and negative for points outside of S.

Of course, many such functions can be found. The approach of the OCSVM is to define f as the function which separates the training data set from the origin. Suppose, $I = \mathbb{R}^D$ and there is a (possibly non-linear) transformation function Φ, then f can be written as:

$$f(x) = sgn((w\Phi(x)) - b) \tag{4.27}$$

To find f based on a training data set, the maximum-margin approach known from standard SVM is used. Let there be training data samples x_i with $x \in \mathbb{R}^D$ and $i = 1, ..., N$. Then, the following optimization problem is solved:

$$argmin_{w,b} \frac{1}{2} \parallel w \parallel^2 + \frac{1}{\nu D} \sum_i \xi_i - b \tag{4.28}$$

with the constraints:

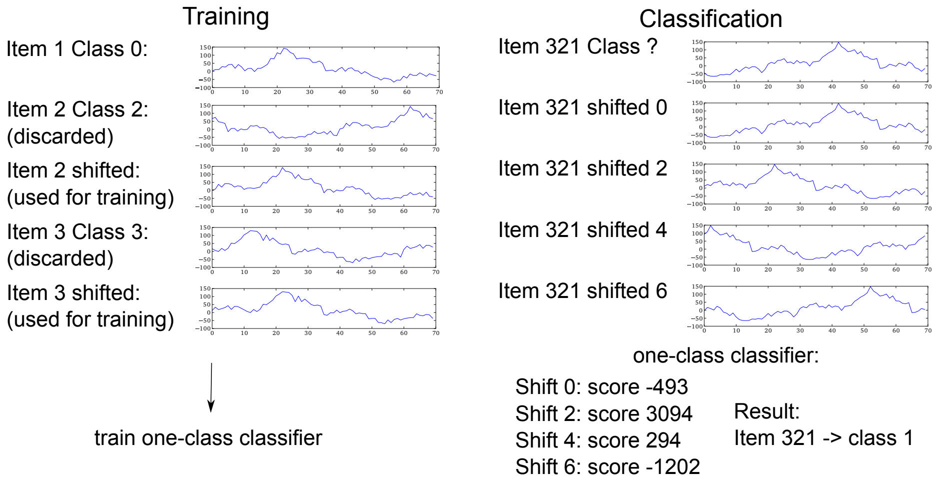
$$w \cdot \Phi(x_i) \geq \quad b - \xi_i \tag{4.29}$$

$$\xi_i \geq \quad 0 \tag{4.30}$$

where ξ_i is again a slack variable tolerating training data vectors where the function is not positive. This constrained optimization problem balances two conflicting goals. First, the function should be positive for the training data samples. Second, $\parallel w \parallel$ should be small (basically this means regularization). The second goals is similar as in the derivation of the SVM. Without it, the margin could grow infinitely. ν is set by the experimenter to balance between those two goals.

Solving the corresponding Lagrange function and substituting leads to:

$$f(x) = sgn(\sum_i \alpha_i k(x_i, x) - b) \tag{4.31}$$



(a) All training vectors are back-shifted according to the distance in the codebook to obtain one class.

(b) For classification of new vectors, copies of the vector with all possible shifts are classified.

Figure 4.2.

where α_i is the Lagrange coefficient for x_i and $k(x, y) = \Phi(x) * \Phi(y)$ is the kernel corresponding to Φ. Accordingly, one can derive:

$$b = \sum_j \alpha_j k(x_j, x_i) \tag{4.32}$$

for any x_i with $\alpha_i \neq 0$, i.e. for any support vector.

So, for a given training data set, a function f can be derived which separates most of the training vectors from the origin and can be expressed using only scalar products of the training vectors. Using the kernel trick, this simple geometric approach of separating the training set from the origin in the feature space leads to complex pattern in the input space.

4.1.3.6. One-Class SVM for cVEP-BCIs

For cVEP-BCIs, the classification task can be described as follows: There is a training data set containing items with the same shift (usually the shift of target 0). When a new data item arrives for classification, copies of that item are produced and shifted for each possible class. Now, the classifier has to detect which one of these copies fits the best with the training data set. This was done by computing the correlation between each copy and the training data set (usually the mean of the training data set was used) [Bin et al., 2009a]. But, this scenario can be viewed as a typical situation for a one-class classification [Spüler et al., 2012a]. The properties of this class can be learned using the training data set. After training, the classifier can compute for every copy the likeliness that this copy belongs to the learned class, as illustrated in Figure 4.2. This idea was realized in [Spüler et al., 2012a] using a linear OCSVM. The OCSVM was trained using a training data set of one target and later each copy of the new data item was classified. The output of the OCSVM is the binary class label (-1 or 1 depending on whether or not the item is supposed to belong to the class) as well as a score which is the distance to the decision boundary multiplied with the class label. This means one can just select the copy with the highest score. Similar to the correlation approach before, where one selects the copy with the highest correlation. Still, the OCSVM approach gives better results [Spüler et al., 2012a].

To understand the difference between the two approaches one needs to analyze them more deeply. In the correlation approach one chooses the copy which has the highest linear correlation with the mean of the training data set. This is the copy where the scalar product between the copy and the mean of the training data is maximal. In the OCSVM a linear kernel is used. This means the OCSVM finds a hyperplane which separates the training data from the origin. Then, the copy with the highest score is the copy where the scalar product between the vector w, which describes the hyperplane, and the copy is maximal. So, one approach chooses the copy which fits best to the mean of the training data and one approach chooses the copy which fits best to some vector w which describes the training data. Empirical evidence from Spüler et al. suggests that the second approach works better. To find the reason a thorough comparison of both approaches is needed, which is out of the scope of this work.

Still, one can speculate that the OCSVM produces better results because it produces a weighted classifier, i. e. dimensions which provide a larger amount of information for the classification can get higher weights. In contrast, the correlation approach considers all dimensions equally.

For the actual training procedure two parameters have to be set for the OCSVM: ν which balances the regularization and the cost parameter c which limits the influence of single training data vectors. In [Spüler et al., 2012a] both were set to the defaults of libsvm [Chang and Lin, 2011], i.e. $\nu = 0.5$ and $c = 1$.

In my experiments I use libsvm as well with the same parameters. I tested a wide range for both parameters off-line, but achieved no significant improvement.

4.1.4. Classification by Linear Discriminant Analysis

So far, the most successful classification approach for cVEP-BCIs seems to be the linear OCSVM. Because it is so simple and successful for other BCIs I want to also test the *Linear Discriminant Analysis (LDA)*. Therefore, LDA is introduced here as well, based on the LDA introduction in [Hastie et al., 2009, p. 106ff.].

First, I introduce the common LDA concept, then an enhancement for high-dimensional data, the Regularized LDA.

4.1.4.1. Concept

LDA is based on a Bayesian approach. Suppose an arbitrary classifier, which has to assign class labels $C_1, ..., C_N$ to a data item x. Then, *Bayesian law* states that the probability that the item has class label C_k is:

$$P(C_k|x) = \frac{P(x|C_k)\pi_k}{\sum\limits_{i=0}^{N} P(x|C_i)\pi_i} \tag{4.33}$$

where π_k is the class prior and $P(x|C_k)$ is the likelihood, i.e. the probability that one would draw item x from the class distribution C_k. The denominator can be ignored for the classification problem as it is equal for every $P(C_k|x)$. It remains to model the prior and the likelihood. To do this, one needs to make assumptions about the class distribution. For LDA one assumes a multivariate Gaussian distribution with mean μ and standard deviation Σ, this gives the following likelihood:

$$P(x|C_k) = \frac{1}{(2\pi)^{(D/2)}|\Sigma_k|^{1/2}} exp(-\frac{1}{2}(x-\mu_k)^T\Sigma_k^{-1}(x-\mu_k)) \tag{4.34}$$

where D is the dimensionality of the data vectors.

LDA further assumes that the covariances of all classes are equal, i.e $\Sigma_i = \Sigma_j \forall i, j \in 1, ..., N$.

To compute whether class C_i or C_j is more probable, one can now compute $\frac{P(x|C_i)\pi_i}{P(x|C_j)\pi_j}$ and test whether this expression is greater than 1. This equivalent to $log(\frac{P(x|C_i)\pi_i}{P(x|C_j)\pi_j})$ greater 0.

$$log(\frac{P(x|C_i)log(\pi_i)}{P(x|C_j)log(\pi_j)}) = log(P(x|C_i)) + log(\pi_i) - log(P(x|C_j)) - log(\pi_j) \tag{4.35}$$

$$\begin{aligned} = log(\pi_i) - log(\pi_j) + &log(\frac{1}{(2\pi)^{(D/2)}|\Sigma_i|^{1/2}}) - \frac{1}{2}(x-\mu_i)^T\Sigma_i^{-1}(x-\mu_i) \\ &-log(\frac{1}{(2\pi)^{(D/2)}|\Sigma_j|^{1/2}}) + \frac{1}{2}(x-\mu_j)^T\Sigma_j^{-1}(x-\mu_j) \end{aligned} \tag{4.36}$$

$$= log(\pi_i) - log(\pi_j) - \frac{1}{2}(\mu_i+\mu_j)^T\Sigma^{-1}(\mu_i-\mu_j) + x^T\Sigma^{-1}(\mu_i-\mu_j) \tag{4.37}$$

where the non-linear terms in x cancel out due to the shared covariance. Note that only the last term depends on x. This criterion could therefore be rewritten as:

$$x^T w + b \geq 0 \tag{4.38}$$

with $w = \Sigma^{-1}(\mu_i - \mu_j)$ and $b = log(\pi_i) - log(\pi_j) - \frac{1}{2}(\mu_i + \mu_j)^T \Sigma^{-1}(\mu_i - \mu_j)$. The same projection, i.e. the same w was also derived by Fisher, as the most discriminative projection, see [Bishop et al., 2006, section 4.1.4], which is why the same approach is sometimes called Linear Discriminant Analysis and sometimes Fisher Discriminant Analysis.

In a practical classification task, the parameters of the class distributions are usually not known, instead they are derived from some training data. Usually, one uses the fraction of training data samples as class prior. Mean and covariance are computed from the statistics of the training samples of the corresponding class.

4.1.4.2. Regularized LDA

In cases where the number of dimensions is high or the number of training data items is low, the parameters of the distributions cannot be estimated accurately. Especially, the covariance is problematic as it has D^2 parameters for a D-dimensional problem. To compensate for this, one can shrink the learned covariance matrix towards a multiple of the identity matrix. [Hastie et al., 2009, p. 112]. I will call this Regularized LDA[3] (RLDA) [Friedman, 1989]. Assume a training data set $A = (x_1, y_1)...., (x_N, y_N)$ of labeled training samples x_i and class labels $y_i \in (1, ..., M)$.

In standard LDA the covariance matrix is estimated based on the training data according to:

$$\hat{\Sigma} = \sum_{m=1}^{M} \sum_{i=1, y_i = m}^{N} (x_i - \mu_m)(x_i - \mu_m)^T / (N - M) \tag{4.39}$$

For Regularized LDA this term is augmented:

$$\hat{\Sigma}' = (1 - \lambda)\hat{\Sigma} + \lambda a I \tag{4.40}$$

where λ is a user-specified value balancing the regularization, a is some computed multiplier and I is the identity matrix. λ has to be given by the user or computed using cross-validation. Different approaches exist for deriving a, for example a can be computed as the average eigenvalue of $\hat{\Sigma}$ [Friedman, 1989]. For exploring whether RLDA is feasible for cVEP at all, I just set $a = 1$, this is also the standard option in the used software package mlpy [Albanese et al., 2012]. Afterwards, Regularized LDA is just like standard LDA, but every $\hat{\Sigma}$ replaced by $\hat{\Sigma}'$.

4.1.4.3. LDA for BCIs

As mentioned in the SVM section, LDA is very successful for the classification of EEG data for BCIs [McFarland and Wolpaw, 2011; Lotte et al., 2007]. As stated before, more complex classification approaches always suffer the problem that one needs either a good model of the data or a lot of training data to estimate the parameters. EEG data is high-dimensional, difficult to model [Hema et al., 2007; Zhang et al., 2008] and training data is expensive to acquire, especially because one usually needs specific training data per subject and session [Riechmann and Finke, 2012]. A linear hyperplane is probably not the best possible classification boundary, but the only one that can be estimated quite accurately under these restrictions.

Even the parameters for the LDA can be difficult to estimate, leading to the use of Regularized LDA for BCIs. Surprisingly, this is much less common though [Lotte et al., 2007].

For cVEP classification it might be interesting, because there the number of dimensions is very high and LDA without regularization fails. I will investigate this in the first study, chapter 5.

[3] This approach is also called Regularized Discriminant Analysis (RDA). Unfortunately, RDA is used ambiguously. First, for balancing between different covariances for each class (QDA) and shared covariance (LDA). Second, for balancing between the estimated covariance and a multiple of the identity matrix [Friedman, 1989]. For clarification I use the term RLDA for the approach to balance the estimated covariance and the multiple of the identity matrix.

4.1.5. Enhancing Accuracy Using Repetitions

Despite all efforts to maximize accuracy by enhancing signal strength and signal processing strategies EEG-based BCIs are still not completely reliable. Another method to improve accuracy is to combine more then one epoch to derive the users intention. To ease this discussion, I introduce a new term, the *decision*. Basically one *decision* consists of one or more epochs and results in at most one command. One can imagine this in terms of a dialogue. The application asks a question, like "Where to drive next", and the BCI runs one or more epochs to derive the user's intent. Either a command is issued to answer the question or no command is issued. One of these question-answer procedures is called a *decision*. Now let us assume, there is a suitable way to combine the information from the different epochs. For example, choosing the command whose class is assigned most often to one of the epochs. Then, a higher number of repetitions or epochs per decision allows to increase accuracy. Of course, more repetitions also means increased latency and less information throughput.

For P300-based BCIs repetition is almost always used. For ERD-based systems and SSVEP systems one can also improve accuracy by increasing the length of an epoch. Here, I only describe repetitions, because that is also suitable for cVEP-based BCIs. A broader analysis of post-classification measures to improve accuracy can be found in [Plass-Oude Bos et al., 2012].

Generally, approaches to increase accuracy by repetitions can be grouped as either using a fixed and static number of repetitions or using a dynamic number of repetitions.

i. Static number of repetitions Most studies concerning P300-based BCIs use a fixed number of repetitions. The number of repetitions usually is chosen using results of preliminary tests or experience of the experimenter. Early studies used up to 80 repetitions [Farwell and Donchin, 1988]. Nowadays 4 to 10 repetitions are usually used [Lotte et al., 2007]. The exact number depends on the signal processing, some experimental parameters like the time between the flashing of two stimuli, and the accuracy one wishes to achieve.

So, basically when choosing the number of repetitions one balances accuracy and speed based on the signal strength. As the signal strength varies during the experiment one often has to do more repetitions than strictly necessary to achieve a constant, high accuracy.

ii. Dynamic number of repetitions Alternatively, one can try to tune the number of repetitions dynamically to the current signal strength to improve speed when signal strength is high and still be accurate even when signal strength is low. This allows to preserve the maximal possible speed for varying signal strengths. An evaluation of different ways to adapt the number of repetitions dynamically can be found in [Schreuder et al., 2013].

Several approaches exists to dynamically adapt the number of repetitions. The major challenge is to estimate the current signal strength. Often, the information which is used to combine the result of the different epochs is also used to estimate the signal strength. When using the example above, i.e. a voting scheme for the class labels, one might repeat until the votes for the highest-ranked class exceed the votes for all other classes by a certain amount.

Depending on the classifier which is used, more sophisticated approaches can be applied. The classifier most often used in P300 systems is the Linear Discriminant Analysis (LDA). In addition to the class label one also gets a score value for each epoch, which is the distance of the new data vector to the decision hyperplane. In [Lenhardt et al., 2008] this score is summed up across epochs for each possible class, for epochs classified as negative, i.e. not containing the P300 signal, their score is considered as negative. Then, the sum of all scores is computed to estimate the signal strength. When this sum is lower than a given threshold, the decision is considered as classified with enough certainty and no more repetitions are started. Setting the threshold allows to balance accuracy and speed. With a suitable threshold, the mean bitrate could be boosted from 32 bits/min to 45 bits/min with a slight accuracy loss of 5% compared to a system using a static number of repetitions which was optimized per subject.

In [Riechmann et al., 2011] we also implemented and evaluated a P300 system using a dynamic number of repetitions. We summed the score output of the LDA across epochs for each class. Then, the highest score-sum was checked against a threshold. The threshold was computed subject-specifically on the training data as the lowest threshold which achieved the maximal possible accuracy on the training data. With an average of 4.88 repetitions an accuracy of 97% for a 5-class P300 system was achieved in the on-line run after the training.

For my cVEP-BCI two approaches for the dynamic repetitions will be implemented and evaluated in the experiments.

4.2. Modern Software for BCI Experiments - The UBiCI Framework

For the recording, processing and analysis of the EEG data a suitable software framework is necessary, which is also able to interface with the target device of the BCI. For my research I mainly use the UBiCI framework, which is developed completely by our working group, partly also by myself.

In this section, I want to present this software. First, the design goals, especially under the assumption that the software is developed for a scientific context. In my opinion, this imposes some special requirements on the software. Next, I present some other frameworks, which are available and the reasons, why we chose to implement a new framework. Then, I describe the global architecture and some more detailed aspects which show how we address the challenges. Finally, I describe some points which are open for improvement.

4.2.1. Challenges for Research Software for BCIs

Here, I want to describe the requirements of a software framework for the analysis of EEG data in a scientific context. Some of these are general requirements for software which is used in a research context. Others are more specific requirements for software for BCIs. First, I describe some typical use cases to derive the requirements. At the beginning, one needs to remember the different parts of a BCI, which are mentioned in section 2 p. 7, because the different parts have somehow different requirements. These parts are data acquisition, signal processing, output device and communication protocol.

For a typical BCI scenario, the data acquisition needs to be able to communicate to the driver of an EEG amplifier, typically these drivers are closed source and only available for certain architectures and platforms [Venthur and Blankertz, 2012]. The data acquisition encodes the data somehow and sends it to the signal processing. The signal processing analyses the data on-line. For this, it needs to integrate implementations of several algorithms. As one does not want to code all these algorithms, a possibility to interface with a wide variety of libraries is needed. Again, most implementations are only available for certain architectures or programming languages/environments. Then, the result of the signal processing has to be converted into a command for the output device and sent to it.

Another typical use-case is to try different signal processing algorithms off-line after the experiment and to do statistical analyses on the data. For this one needs again to integrate a lot of libraries, but one needs also to store the data in a way to easily run analysis on all the data-sets of one experiment.

Next, often users, especially under-graduate students, use such a framework for a short project, for example as part of their master thesis. They have limited time and often also limited coding experience. So, the first-time use of the framework has to be as easy as possible.

Last, one sometimes wants to give a demonstration of the work to outsiders. Usually the work being demonstrated is not only the work, one is currently working on, but also older work. So, one needs a mechanism to either use the old version of the software or to ensure that the current version is still usable for the old demo.

From these use-cases I derive the following general requirements:

1. **Modularity** The first and most important requirement is modularity. Several components have to work together, the three major parts mentioned earlier, but also within these parts different processing steps have to be put in independent software components. When doing research, the software is always changing and, to the extent possible, one has to ensure that changes in one part do not affect another.

2. **Extensibility** Somehow connected to modularity, extensibility means it has to be easy to integrate new functionality into the code. For this, the framework has to be modular, but that is not

enough. Additionally, for example for the integration of extended types, the existing code has to ignore the additional fields internally, but to preserve them for the outside. The framework needs to support and enforce this, for example through the use of polymorphism.

3. **Connectivity** A BCI framework needs to work with a lot of external software and libraries. For external software one needs methods to easily convert the data types to other formats and support efficient and commonly used inter-process communication mechanisms. For easy integration of external libraries one needs interfaces to other programming languages and software environments.

4. **Platform independence** Platform independence means that the framework has to run on different platforms and architectures, partly this is a direct consequence of the requirement of connectivity. As most softwares and libraries are tied to a particular platform, our framework needs to support all these platforms to interact with these external softwares and libraries. Additionally, most users are used to a particular platform, where they feel most comfortable and it greatly eases first-time use, when a new framework works in the environment one is used to.

5. **Rapid prototyping** In a research context, it is absolutely necessary to offer a way to test a new idea rapidly. Partly, this done by high extensibility and connectivity, but still rapid prototyping is a major goal on its own.

6. **Ease of use** As mentioned, there are always cases where users want to use the framework for a small project, for example, to acquire some EEG data, which they analyze later using their own set of tools. This is both very beneficial for research as well as for the further development of the software, as new users find bugs and missing features more often and easily as users who work with it everyday. To support this, a framework has to offer an easy start.

7. **Backwards compatibility** The last one of the use-cases I mentioned above, imposes a special requirement. To the extent possible the software should be backwards compatible. One can argue against this. With modern source code management it is quite easy just to switch back to an older version for a demo. Still, then one has to deal with all the bugs present in the old version, which were fixed in the current one. Additionally, to the extent possible scientific experiments should be reproducible. For this, one not only needs to store the data safely, but also somehow tag the software.

4.2.2. Other Available Frameworks

There are several frameworks available for software development in the context of BCIs and EEG data analysis. So, we had to decide whether to use and possibly extend one of the existing frameworks or to develop our own from scratch. When looking at the competitors one might be tempted to look at the situation back in the year 2010, when this decision was taken in our working group. Still, I will do this only briefly and then look at the situation today.

In 2010, Renard et al. published a paper, where they advertised their framework OpenViBE and described the state of other frameworks at that time [Renard et al., 2010]. At that time there were four frameworks for BCIs available for free. BioSig [Schlögl et al., 2007], BCI2000 [Mellinger et al., 2007], BCI++ [Maggi et al., 2008] and OpenViBE. None of those met the requirements above.

1. **BioSig** BioSig is a toolbox written for MATLAB and Octave and is bound to these platforms by design. This means that they are not suited to be connected well with external libraries. Additionally, MATLAB is not free and Octave, a free MATLAB alternative does not provide the complete API of MATLAB.

2. **BCI2000** The authors of BCI2000 restricted the use of the free version to non-profit research and educational purposes. It was difficult to judge in how far this would be suitable as a starting point for our own extensions.

3. **BCI++** An interesting project, integrating a C++ core with a MATLAB binding. Again the license was unclear. Additionally, the project seemed not to be developed actively.

4. **OpenViBE** This is complete, open source framework for BCI experiments, tuned towards the integration of BCI and Virtual Reality.

In 2013 a new overview of BCI frameworks was presented in [Brunner et al., 2013]. I briefly summarize this study here to have a base for comparison with the UBiCI. Seven frameworks were presented:

1. **BCI2000** The license model of BCI2000 was changed to an open-source license, the GPL. Additionally, a lot of effort has been invested by different labs in the past years. BCI2000 now offers a wide range of components for data acquisition from different devices, signal processing, storage, data export, audio and video stimulation. It is written in C/C++. The modules for a particular experiment can be arranged via a GUI without programming knowledge. For easy connection of external software and other libraries thorough Python and MATLAB interfaces exist, additionally BCI2000 exposes its state through a simple UDP interface.

   BCI2000 is used by laboratories and clinics all over the world. Development particularly focuses on ease of use, especially to provide BCIs for application on patients. There, software has to be usable for non-technical personnel like doctors and nurses.

   Originally BCI2000 is designed for use on Microsoft Windows, although unsupported versions for Mac and Unix seem to exist.

   Although, it is the most complete and sophisticated framework for BCIs, it is tuned towards bringing BCIs to the clinics not towards research. In particular, it is not really platform independent.

2. **OpenViBE** OpenViBE also offers a wide range of components for all parts of a BCI. The OpenViBE developers explicitly try to balance the needs of developers, researchers and clinicians. As such, it supports easy embedding of new components by providing an easy interface for developers. Additionally, it is more platform independent, Windows and Linux versions are supported by the official maintainers, additional platforms are supported by the community. As in the BCI2000, the components can be arranged in a graphical way without programming knowledge, additionally complete interaction scenarios can be built in a GUI editor without programming.

   A particular focus is on visualization, especially real-time visualization of the brain activity in 3D and the possibility to interface with Virtual Reality appliances.

3. **TOBI Common Implementation Platform (CIP)** The TOBI CIP is not a real framework for BCIs, but a set of interfaces designed to ease interaction of different frameworks. These interfaces describe different data types for storing and sending EEG data, class labels, timing information and so on. Additionally, TOBI CIP provides a network-based, inter-process communication based on these types.

4. **BCILAB** BCILAB is a MATLAB-based toolbox, which provides a lot of signal processing components and interacts well with EEGLAB, a well-known MATLAB-based toolbox for analysis and visualization of brain activity data.

5. **BCI++** Another C++-based, modular set of components for BCI research. Although it is presented in the 2013 survey, active development seems to have ceased quite some time ago, the news about a new version to be released is from May 2010[4]

6. **xBCI** Seems also to be no longer maintained, although there is a paper from 2010 describing the framework [Susila et al., 2010], the latest version of the software was released in 2008[5].

7. **BF++** This is an interesting tool-suite designed to evaluate and compare different human computer interaction systems and BCIs based on several long established and some self-developed metrics. Unfortunately, it is currently only provided as Windows binary, without options to extend it.

[4]http://www.sensibilab.lecco.polimi.it/
[5]http://sourceforge.net/projects/xbci/files/

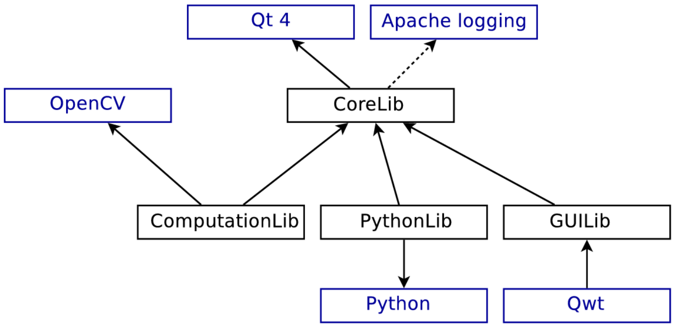


Figure 4.3.: Structure of UBiCI libraries and dependencies. Components are grouped into libraries depending on their function and external dependencies. Therefore, one only needs to install external dependencies for components one really needs.

In summary, there are two widely used BCI frameworks today, the BCI2000 and the OpenViBE. Additionally, there is the MATLAB-bound BCILAB and several smaller frameworks, which were developed at a single institute and are either no longer maintained or only partly accessible to the public.

Both of the major frameworks offer a rich set of well-documented components to do BCI research. Still, both are not perfect. BCI2000 is tuned more towards clinical use than towards research and it mostly tied to the Microsoft Windows platform. OpenViBE lacks bindings to other programming languages and architectures.

4.2.3. Structure of the UBiCI

To have a truly multi-platform framework for research on BCIs our working group implemented the UBiCI framework. Here, I will present the guidelines for the implementation and show how the requirements from above are met by the framework using these guidelines.

To realize the implementation we use *C/C++* and *QT*[6]. C/C++ is still a widely used and distributed programming language, allowing the inclusion of many external libraries easily. QT is a C/C++ framework which supports a very wide range of platforms and architectures. Additionally, it allows to very easily realize the communication between different modular units with its signal/slot mechanism.

The central concept of the UBiCI is called a *component*. Each component represents one processing step, this can be a signal processing step like a bandpass filter or a more general processing step, like storing the data on the disk. Each component announces its possible in- and outputs in its signature. To set-up an experiment, the user of the framework only needs to define which components to use and the connections between these components. This can be done either in a simple text-file or by visual programming in the GUI. One such set-up is called a *deployment*. Additionally, components and connections can be grouped into so called *modules* to create and share bigger building blocks for common tasks such as data recording and storage.

To be as platform independent as possible, the core classes, which are needed to set-up the components only depend on C/C++ standard features and QT. These core classes are tested and maintained for Windows and Unix with compilation in GNU gcc[7], Visual Studio[8] and LLVM/clang[9]. Still, some components have additional dependencies, for example components for data acquisition which depend on the driver for the specific hardware. Therefore, components are grouped into different libraries according to their dependencies. At start, the framework loads only those libraries which are necessary for the given deployment.

[6] www.qt-project.org
[7] gcc.gnu.org
[8] www.visualstudio.com
[9] clang.llvm.org

The framework also contains a mechanism to transparently connect processes running on different hosts. This allows to run each component on the platform which is most easily supported by its dependencies. Additionally, one can distribute the computational workload easily on different machines.

The framework allows to code components in Python[10] and use these components completely transparently. This eases rapid prototyping and enhances connectivity, as today many libraries offer python bindings. MATLAB[11] code can also be called directly from within the framework using a Python-to-MATLAB library. This point enhances productivity in a research environment much, because today most scientists, if they publish their implementation of a new approach, they mostly do it as a small MATLAB toolbox or Python script. Additionally, it eases collaboration as the data can be easily converted to different formats, i.e. my own datasets are published in the UBiCI format and as MATLAB/eeglab datasets.

For ease of first time use we deliver pre-built binaries, which already include example deployments, for example for data acquisition and storage. Additionally, some tutorials explain the first steps.

All components and types can be sub-classed for easy extensibility. Both new components and types can be defined in additional libraries. New types can be transparently used in the framework and easily send over network or stored on the hard-disk. In addition to enhancing extensibility, this feature also greatly helps to improve backwards compatibility. Instead of changing an existing component one can just create a sub-class and add the new features there.

Another method to improve backwards compatibility are regression-tests. This is especially useful to ensure the reproducibility of experiments. Using an automated build-server, it is automatically checked after a software modifications is checked into the source code management that this change does not break existing deployments.

4.2.4. Limitations

The framework has been used successfully for several studies in our working group. It has been used by PhD students as well as by under-graduate students for their bachelor or master theses. Still, there are some points open for improvement.

First, at the moment, the framework is not yet published. This is planned, but needs some time.

Second, the number of components is limited in comparison to BCI2000 or OpenViBE. This is caused already by a much smaller number of developers and users and can only be remedied by publishing and advertising it. One should not overestimate this point, though. Everything needed for the research in our group is already there and new components can be easily added.

Third, it would be nice to implement the TOBI interfaces. This would allow closer interaction with other frameworks and remedy the second issue.

4.3. Common Experimental Setup

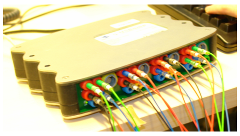
In addition to the UBiCI framework, some parts of the experimental setup are more or less equally used in all of my studies. Thus, they are described here to avoid duplication.

These are parts of the stimulus presentation, my method to synchronize stimulus presentation and EEG data, and the data acquisition. The signal processing differed between experiments and is described per experiment.

4.3.1. Stimulus Presentation

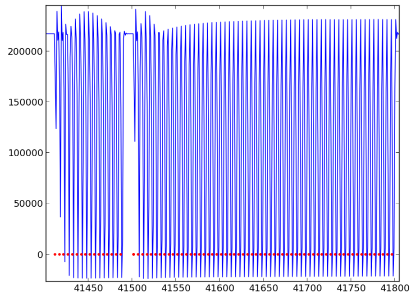
The stimulus presentation for a cVEP-BCI is more of a technical challenge than for others BCIs. It is absolutely necessary to ensure that each frame is displayed exactly once. Otherwise the brain activity is distorted and does no longer fit to the learned pattern. First, one has to decide what kind of displaying device should be used. The different options can be evaluated similar as for SSVEP-BCIs. One could use LED arrays, CRT monitors and LCD monitors. LED arrays are not suitable as I want to embed

[10] https://www.python.org/
[11] www.mathworks.com/products/matlab/



(a) gTec gUSBAmp

(b) Photo of the experimental set-up with the CRT monitor, the attached photo-resistor and the EEG device.



(c) Example response of the photo-resistor. The red dots mark the recognized white frames. The gap around 45000 marks the start of the epoch.

Figure 4.4.

stimuli and scene and this cannot be done with LED arrays[12]. LCD monitors cannot produce sharp changes between the frames and cannot provide the necessary high frame rates. This means one can only use CRT monitors at the moment. They can render the whole scene, produce sharp transitions of the frames and run at the necessary frequencies. The only drawback is that they produce slightly weaker responses of the visual cortex than LED arrays, at least in SSVEPs, as discussed in section 3.2.1 p. 30f.

The monitor I use is a 19" CRT, the Samsung SyncMaster 1000MB, running at a resolution of 800×600 pixel. V-Sync is enabled as well as the Double-Buffer of the graphics card.

For the first three experiments stimuli and scene were rendered using a self-written OpenGL application. Every frame of the codebook was rendered beforehand and saved as a bitmap. During the studies the OpenGL application just rendered one rectangle over the whole screen, using the correct bitmap as texture.

Before the studies I checked the correct timing of the rendering using some high speed cameras from our working-group.

4.3.2. Synchronization of Stimuli and EEG

Precise timing is not only necessary for the rendering of the stimuli, but one needs also a precise synchronization of rendering and EEG data. The difference between two stimuli in the codebook is only two frames or, when running at 60 Hz, 33.33 ms. This means that for a deviation of about 16 ms it becomes impossible to recognize the correct target.

To ensure very precise timing I attached a photo resistor to the screen. The photo resistor is connected to a battery and an input channel of the EEG device. Underneath the photo resistor a small white rectangle is rendered. For the first frame of every epoch the white rectangle is replaced by a black one. The photo-resistor has a higher resistance when it is stimulated by light. Using a small electrical circuit this resistance is transformed into a voltage[13]. The voltage output is connected to an input of the EEG device. This means that the voltage which is measured in the connected EEG channel drops every time the monitor draws the white rectangle, an example is shown in Figure 4.4c. For the first frame of each epoch this rectangle is not rendered, thus no drop in the photo-resistor channel is registered. This missing drop indicates the begin of an epoch. Because the photo-resistor is connected to same EEG device as the head electrodes, the beginning of the epoch in the photo-resistor channel also precisely marks the beginning of the epoch in the channels of the head electrodes.

A photo of the set-up of monitor and photo resistor is given in Figure 4.4b.

[12]OLED displays would be an interesting alternative, but as of now they are too expensive.
[13]The complete circuit can be found in the appendix, Figure A.1 p. 120

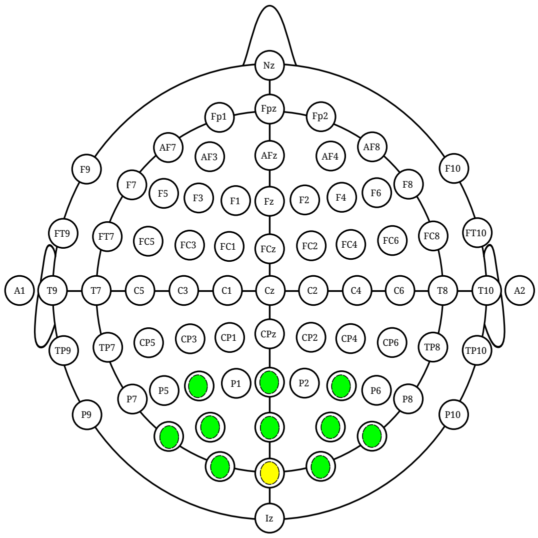


Figure 4.5.: Head model with electrode positions for my studies. Adapted version of Figure 2.3a

4.3.3. EEG Recordings

Data was acquired using a gUSBamp (Guger Technologies) amplifier, depicted in Figure 4.4a. 10 channels were equipped with Ag/AgCl electrodes placed at P3, Pz, P4, PO3, POz, PO4, PO7, O1, O2, PO8 according to the extended international 10-20 system and referenced at Oz. The electrode set-up is depicted in Figure 4.5. Electrode sites were cleaned with a medical alcohol and prepared with some medical EEG paste. Electrode impedances were kept below 5 kΩ. The amplifier sampled the EEG data at 600 Hz, performed high-pass filtering at 1 Hz and notch filtering at $48 - 52$ Hz.

5. Influence of Codebook and Color in cVEP-BCIs

So far, cVEP systems had only been studied for spelling systems. These approaches have in common that they utilize only spelling matrices with a fixed number of targets. The stimuli are flickering black/white and occupy the complete screen. This maximizes signal strength, but also means that these system are rather restricted in terms of possible applications beyond spelling. In my first two experiments I investigate the influence of different stimulation parameters on the classification accuracy, as a preparation for the navigation system later.

In this chapter, the first experiment is presented in detail. First, the objectives are discussed. Then the methods, including experimental setup and procedure, are described. In the results section the initial results of the experiment in terms of classification accuracy and user acceptance are given first, these were also published in [Riechmann et al., 2013]. Then, some more analysis on the data is done. At last, a discussion and conclusion of this experiment is given.

The data which was recorded during the study is published under the DOI: 10.4119/unibi/2695379

5.1. Objectives

As a first step towards more flexibility, I investigate three questions.

First, can the *codebook vector* be constructed in such a way that it is possible to very rapidly (250 ms) detect the general area at which the subject gazes, while preserving the classification accuracy? This would, for example, make it possible to initiate robot movement roughly in the intended direction right at the beginning of detecting user's intention. Additionally, one could also render the detected group better for the rest of the epoch, for example increase their size. To accomplish this rapid detection of the general area, I split the decision of the classifier system into two parts. First, the general area and second the specific item within the area.

Second, can a *green/red flicker* be detected as reliably as a black/white flicker in terms of classification accuracy? This would allow to use different colors for the stimuli wherever appropriate.

And third, is one of the two conditions less exhausting for the subjects? This will be measured a) by a questionnaire answered by each subject directly after the experiment. This provides a subjective measure of exhaustiveness; and b) by the Galvanic Skin Response (GSR) of the subject, which is recorded during the experiment. GSR is often used as a physiological, objective measure of the level of excitation and stress [Cacioppo et al., 2007]. Psychologists assume that a green/red flicker is less exhausting [Sutter, 1992], but still this has not been tested in cVEP-based BCIs.

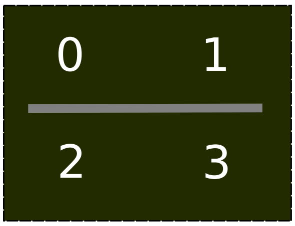
I use 16 targets for this study and split them into 4 areas or groups, which means that both classifier stages have to solve a 4-class problem. This allows for more flexibility when constructing the codebooks. I discuss this, along with the classification scheme, in the next section.

5.2. Methods

This section describes the methods used for the experiment, especially the *hierarchical codebook* and the necessary adaptations to the signal processing sequence. The experimental setup and procedure are also given.

5.2.1. Hierarchical Codebook

The first objective of this experiment is to evaluate a hierarchical codebook. In contrast, I call the traditional approach a flat codebook. The hierarchical approach groups the stimuli. The aim is to



(a) Example stimuli

Stimuli

[0,1,0,1, 1,0,0,1, 0,1,0,1 1,0,0,1]
[0,1,0,1, 0,1,1,0, 0,1,0,1, 0,1,1,0]
[1,0,1,0, 1,0,0,1, 1,0,1,0, 1,0,0,1]
[1,0,1,0, 0,1,1,0, 1,0,1,0, 0,1,1,0]

Frames

(b) Example codebook

Figure 5.1.: Hierarchical codebook example. 4 Stimuli in 2 groups. In this example only 4 bits are used in each segment. The two codebook blocks are repeated once, i.e. segments 1 and 3 and segments 2 and 4 are always equal within a given target. Target 0 and 1 are in the same group and thus differ only in segment 2 and 4. Target 2 is the first item of the second group, thus its segment 1 and 3 differ in relation to the two targets from the first group. Target 3 accordingly is the second target of the second group.

detect the group early, even before the end of the epoch. To do so n-targets are divided into m groups with n/m items per group. Then, one codebook for m targets and one for m/n targets are designed and combined. The two parts of the codebook are called blocks for the remainder of this chapter. An example for four targets in two groups is depicted in Figure 5.1. The first block differs between the upper and the lower group, but is equal for target 0 and 1 and also for target 2 and 3. The next block is different between the items per group, but equal for all groups. This way the first block encodes information about group membership and the second block about the specific item within the group. Both blocks together encode a specific target. The two blocks are then repeated to increase accuracy.

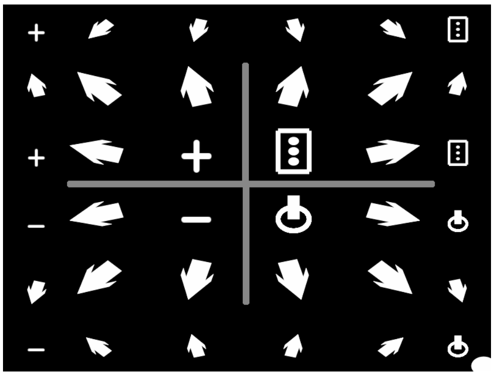
For the experiment I use 16-targets: 12 arrows and 4 navigation symbols (start/stop, faster, slower, menu). These targets are divided into 4 groups, according to Figure 5.2. In addition to the 16 target-stimuli the figure also shows a gray cross and several non-target stimuli at the border of the screen. The gray cross indicates how the targets are divided into groups, it is not part of the real stimulus presentation. In fact, the subjects did not know about the hierarchical codebook. Additionally, the figure shows the non-target stimuli in the surroundings which are matching neighbors for the second classification stage. E.E. Sutter speculated in [Sutter, 1992] that the first neighbors also contribute non-marginally to the activity of the visual cortex in addition to the real target. He proposed each stimulus should have an equal number of first neighbors with an equal lag in the codebook. For the hierarchical codebook this can only be done for one of the blocks. I chose the second block, this will be discussed later in section 5.3.

5.2.2. Signal Processing

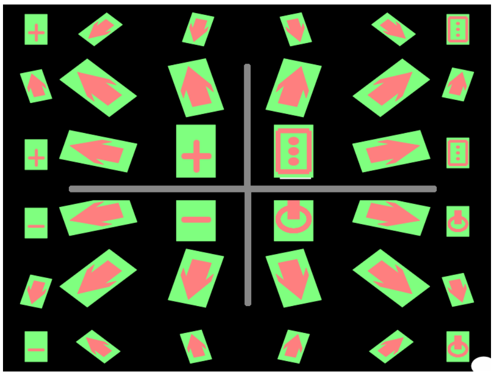
The signal processing has to be adapted as well to cope with the hierarchical codebook. I propose a hierarchical two-stage system for classification. The first classifier detects the group or area, the second classifier the specific item within the group. As described, the first block of the codebook is different between groups and equal within the groups and the second block is constructed the other way round. To create the codebooks I use m-sequences according to [Bin et al., 2009a] and chose a 4-bit shift between items resulting in a 15 bit m-sequence for the first block and accordingly another 15 bit for the second block. I repeated this 30 bit codebook once and ended up with a 60 bit codebook.

In current cVEP systems the data is shifted to obtain a one-class problem, for example in [Spüler et al., 2012a]. For the hierarchical codebook this is not necessary, though. On both classification stages only a 4 class classification problem has to be solved. This can be done efficiently by a multi-class Support-Vector Machine (SVM). For this study the SVM from libsvm [Chang and Lin, 2011] was used with a linear kernel and the cost parameter set to 1. These parameters were found experimentally. The advantage of not shifting the data is that it allows to use arbitrary codebooks. In principle, every set of binary codebook vectors would do.

Before the data is classified by the multi-class SVM, a Canonical Correlation Analysis (CCA) is applied as a spatial filter, as described in 4.1.2. The CCA finds, for given data matrices X, Y, the



(a) Black/white condition



(b) Green/red condition

Figure 5.2.: Stimulus presentation of the cVEP system for navigation tasks. The 16 target symbols in the middle area show 12 navigation arrows and 4 four typical menu items: accelerate, open menu, decelerate, on/off. The thin gray lines indicate the division of the items into 4 general areas. These lines are not part of the real stimulus presentation. The surrounding stimuli are non-target stimuli which are added to have correct direct neighbors for each target stimulus.

projections A, B which maximize the correlation $p = corr(AX, BY)$. The EEG data was set as X, in the form of $(trials \cdot samples) \times channels$ and the repeated average of the EEG data as Y, in the form of $(k * samples) \times channels$ where k is the number of training items. After performing the CCA the projection matrix A was used as spatial filter.

5.2.3. Galvanic Skin Response

The two color conditions examined in this experiment are to be evaluated regarding their signal strength, but also regarding the mental effort they demand from the subjects. To evaluate this, I measured the stress level of the subjects during the experiment using the Galvanic Skin Response.

Galvanic Skin Response (GSR), also known as Electrodermal Activity (EDA) refers to the skin conductance of the human hands. Since the end of the 19th century it is known that the electrical conductance of the skin at the hands is almost entirely determined by the activity of the sweat glands [Cacioppo et al., 2007, chapter 7]. Generally, the sweat glands are primarily involved in thermoregulation, i.e. humans sweat to cool the body. However, the sweat glands located on the palmar and plantar surfaces, i.e. on hands and feet, are thought to be more related to grasping behavior, i.e. they modulate the friction. Additionally, they are influenced by psychological relevant stimuli, more specifically the activity of the sweat glands is modulated by the sympathetic nervous system.

The sympathetic nervous system influences the stress reaction of the body. It counter-acts the parasympathetic nervous system which maintains the body during rest [Brodal, 2004]. In contrast to other organs, the sweat glands are exclusively influenced by the sympathetic system, without any influence of the parasympathetic system [Boucsein, 2012]. Thus, they are ideally suited to measure the stress response. Other measures, e.g. heart rate, are simultaneously influenced by sympathetic and parasympathetic system [Setz et al., 2010].

One usually distinguishes two kinds of Galvanic Skin Response. First, a short-time reaction to discrete stimuli, the skin conductance response (SCR) [Cacioppo et al., 2007, p. 168 ff.]. This is not of interest for my experiment and is, therefore, not described here.

Second, the long-term level of conductance, called the Skin Conductance Level (SCL) [Cacioppo et al., 2007, p. 170 ff.]. Basically, the SCL rises whenever the subject anticipates or performs a task. This holds even for purely mental tasks without external stimuli, for example solving mathematical equations. More generally it is modulated by activity, stress and emotion [Turpin and Grandfield, 2010].

Most importantly for my experiments the SCL rises during stressful situations.

Figure 5.3.: Picture of the gTEC GSR sensor from http://www.gtec.at/Products/Electrodes-and-Sensors/g.Sensors-Specs-Features/

For the measurement, one attaches two electrodes at the hands or two different fingers. Usually, one induces a small, constant voltage and measures the induced current flow. The conductance is computed using Ohm's law. It is also possible to induce a constant current and measure the voltage, but this is uncommon today. There also exists a variety of electrode systems usually wet systems with paste. The used electrodes and paste differ considerably from EEG systems, because for GSR recordings one has to ensure that paste and preparation do not interfere with the natural sweat system. The cleaning procedure usually used for EEG recordings would disturb the natural resistive properties of the skin.

For these reasons I use a commercially sensor, specifically designed for GSR measurements, the gTec Galvanic Skin Response (GSR) Sensor, depicted in Figure 5.3. The sensor consists of two dry electrodes, one box with the circuit and battery and two cables for the output. The electrodes are attached to the fingers using a hook-and-loop fastener. The output is compatible with the gTec gUSBAmp and can be used just as an EEG electrode, thus the GSR recording is effortlessly synchronized with the EEG data.

5.2.4. Experimental Setup

EEG data was acquired as described in section 4.3.

I used 16 navigation symbols as stimuli, instead of letters for a spelling task. The stimuli were presented at a frequency of 60 Hz. In the black/white condition the stimulus was rendered bright white when the codebook vector was 1 for this frame and not rendered when the codebook vector was 0. The background was completely black. In the green/red condition the background was also black. Around each stimulus, though, a green rectangle was rendered regardless of the codebook. If the codebook was 1 for the actual frame the stimulus was rendered in red onto the rectangle (see Figure 5.2). Otherwise the stimulus was not rendered.

The Galvanic Skin Response was measured at the volar surfaces of index and middle finger of the left hand to asses which condition (one black/white, one green/red) was more exhausting.

In addition, the participants were asked to fill in a small questionnaire that contained questions on their subjective experience with the task. The specific questions are given later together with the results.

5.2.5. Experimental Procedure

10 healthy subjects completed 2 blocks each, one black/white, one green/red, in alternating order. Each block consisted of 480 trials. Subjects were acquired from the local student population and our working group. All subjects had little or no BCI experience and no experience with cVEP-BCIs. All subjects signed a written consent and were paid for their expenditure of time.

5.3. Results

In this section the results of the experiment are given. First, the classification accuracy is given, then the effect of the hierarchical codebook on the accuracy is examined. Next, the effect of the two color conditions on classification accuracy and user experience is analyzed. Then, I test whether Linear Discriminant Analysis (LDA) can be used instead of the OCSVM for classification. As the

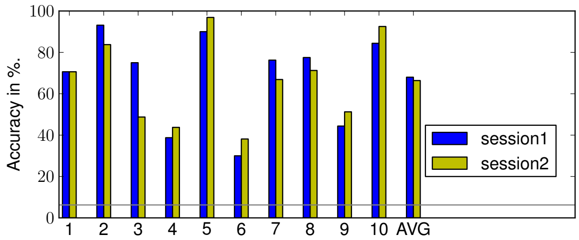


Figure 5.4.: Classification accuracy for the two sessions for all subjects and the average. The gray line indicates the chance level.

accuracy between subjects varies greatly I examine the EEG data also to find possible predictors for cVEP performance. Finally, I analyze for a possible correlation between the probability for a correct classification and the distance to the separating hyperplane.

5.3.1. General Classification Accuracy

First, I evaluated the study with regard to the classification accuracy. To do so, the data set was split into a training set, which contained the first 320 epochs of one session and a testing set, which contained the last 160 epochs of the same session. The accuracy was measured as the percentage of correctly classified items on the testing set given the spatial filter and the SVM trained on the training set. Training and testing was performed on a per-subject basis. Figure 5.4 shows the overall classification accuracies per subject and session.

Subjects 2, 5 and 10 achieve good classification rates (>80%) and four other subjects reach usable classification rates of about 70%. Three subjects do not achieve control over the system, however their classification results are still way better than chance. In total, the system correctly classifies 67% of the trials on average. If one excludes the three non-performing subjects the average increases to 78%. There is a high variance between subjects, but low variance between the two sessions of one subject.

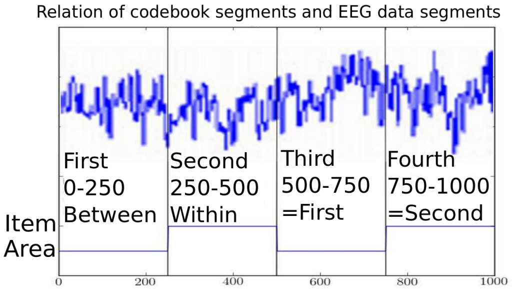
5.3.2. Hierarchical Codebook

For further analysis, each EEG data epoch is split into four segments according to the four blocks of the codebook as illustrated in Figure 5.5a. Classifier training and evaluation is then done for each segment separately, i.e. the classification system is trained using the first segment of the first 320 epochs and the accuracy is calculated using the first segment of the last 160 epochs, and so on for the other segments. For each segment the classification system now has to solve a 4-class problem, either classifying for the general area (segment 1 and 3) or the specific item (segment 2 and 4). The accuracies of the four distinct segments are plotted in Figure 5.5b.

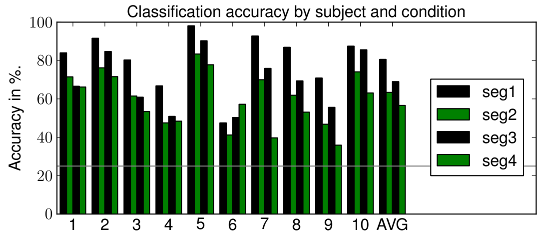
The first segment classifies best with an average of 81%, then the third with an average of 69%, the second with 63.5% and the fourth with 55%. This gives rise to two trends: the first repetition of a block produces better data than the second and the general area is detected better than the specific item.

For further examination of the second trend, the classification per classification stage (general area and specific item) of the whole epochs is examined. Figure 5.6a shows the separate accuracies of the two classifiers for the whole trials. Overall the general area is classified significantly more accurately than the specific item within the area (average 88.2% vs. average 73.4%, paired t-test $p < 0.001$). This is quite unexpected, as the two classification stages should be equally difficult. Both classifiers need to solve a 4 class problem and both parts use two codebooks segments of 15 bits each. Additionally, the neighbor stimuli were rendered to fit classification stage two, which should further improve signal strength for that stage.

Figure 5.5b shows the individual accuracies of the data segments. Note that the respective codebooks and thus the (stimulus) flickering patterns are identical for segment 1 and 3 and also for segment 2 and 4. In theory, the corresponding segments should classify equally well. In practice, however, this is not the case. Segment 1 classifies significantly better than segment 3. The same is true for segments 2

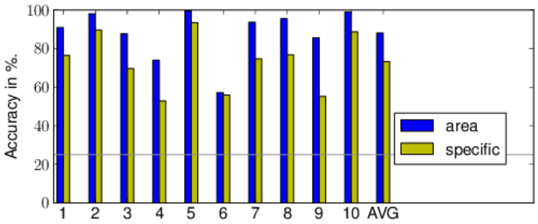


(a) The EEG data of each trial is split into four segments according to the four segments of the hierarchical codebook.

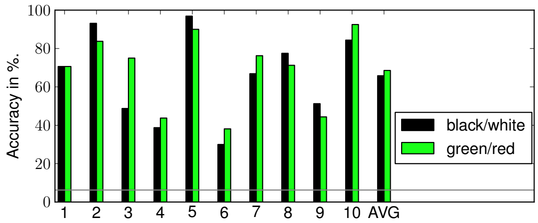


(b) Classification accuracy of the four distinct segments for all subjects and the average. The gray line indicates chance level. The first and third segment contain information about the general area, the second and fourth about the specific item within the area.

Figure 5.5.: Classification per segment



(a) Classification accuracy of the two stages separately for all subjects and the average. The gray line indicates the chance level.



(b) Classification accuracy sorted by condition for all subjects and the average. The gray line indicates chance level.

Figure 5.6.

and 4. The reason for this degradation is not yet clear, but is likely related to the structure of a trial (see Fig. 5.5a). Each segment, that is, the particular flickering pattern described by the corresponding codebook, triggers a unique activation pattern in the visual cortex. The exact time course of this activation is yet unknown. As I use very short segments (≈ 250 ms), the visual cortex activation pattern might not instantly switch from one pattern to the other upon the start of a new segment. There might be a smooth transition from one pattern to the other in terms of neural activity. Such an effect would then cause the signal of a consecutive segment to be a mixture of the true activity corresponding to the current flickering pattern and the activity corresponding to the previous pattern. In other words, a segment would contain also information on its predecessor. If so, a classifier trained on segment 2 data should be able to cope with segment 3 data, at least to a certain degree (i.e., better than chance, $> 25\%$). I evaluated this hypothesis by training a classifier only on segment 2 data and applying it to the data of segment 3. The accuracy (66%) was considerably above chance, which supports the hypothesis. On the other hand, segment 3 classifies generally better than segment 2, but should be equally effected by the mixture problem. Likely, the mixture problem is not the only effect that is present in the data and not the only reason for the worse performance of the second classification stage.

5.3.3. Evaluation of the Questionnaire

The questions and answers of the questionnaire are given in Table 5.1[1]. Unfortunately, subject 1 refused to answer the questionnaire and subject 6 had to be excluded from the analysis, because she marked all answers either 1 or 6 in a non-consistent manner. The questions cover three areas. The

[1]The actual questionnaire was German. Questions were translated by the author. The original questionnaire can be found in the appendix as Figure B.1.

Table 5.1.: Question and answers of the questionnaire, which was filled out by the subjects directly after the experiment. Possible answers ranged from 1 (very low) to 6 (very high). The average in the last column was computed excluding subject 6.

Question	2	3	4	5	6	7	8	9	10	AVG
Before the experiment I felt stressed.	1	1	1	1	1	3	2	2	1	1.5
Before the experiment I felt relaxed.	5	6	2	5	6	3	5	5	5	4.5
The experiment was exhausting.	3	2	4	4	1	6	5	2	4	3.8
The experiment was exciting.	4	3	3	3	1	1	3	3	5	3.1
The experiment was tiring.	4	2	4	5	6	6	4	3	5	4.1
The green/red condition was exhausting.	3	3	5	2	1	6	4	2	4	3.6
The black/white condition was exhausting.	3	1	6	5	1	4	5	4	4	4

state of the subjects before the experiment (relaxed/stressed), the exhaustiveness of the experiment in general, the exhaustiveness of the two color conditions. Each question had to be answered with a score from 1 (very low) to 6 (very high).

In general, subjects felt relaxed before the study. The exceptions are subject 2, who did feel neither stressed (score 1) nor relaxed (score 2) and subject 5, who marked a 3 for these two questions. Both subjects had good classification results, so there seems to be no correlation between feeling relaxed/stressed and signal strength.

The study was quite exhausting and tiring for all subjects. The average questionnaire score was 3.75 points out of 6 possible points for "The experiment was exhausting" and 4.13 out of 6 for "The experiment was tiring". When asked, subjects told that the primary causes were the flickering of the stimuli, which was perceived quite annoying and exhausting and the monotony of the task. The second reason is also supported by the relatively low score for the question "The experiment was exciting" which got 3.1 points in average.

The last two question dealt with the exhaustiveness of each of the two color conditions. In average the green/red condition was perceived a little less exhausting (score 3.6 vs 4.0), but the difference is not significant (paired t-test $p \approx 0.57$). Individually, there are quite some differences between the color conditions, I investigate this further in the next section.

5.3.4. Green/Red Flickering

The two color conditions are compared with regard to two metrics: The classification accuracy and the user experience.

Figure 5.6b shows the accuracies for the two different conditions per subject. On average there is no advantage in classification for one condition over the other. Individual differences exists up to 10% except for subject 3 were the difference amounts to 26%. This subject was very tired during the second session, which was the black/white one, where the accuracy was bad. For subject 1 the difference is zero. For the other eight subjects there are differences between 5% and 10% in both directions. The two conditions were done in an alternating order, so order might have an influence, but the accuracy per session in recording order from Figure 5.4 does not support this. So, for some subjects there seems to be a difference in signal strength depending on the color of the stimuli.

The user experience related to the color conditions is evaluated using two metrics. First the questionnaire and second the GSR measurement. As stated in the questionnaire evaluation, there was no significant difference between the two color conditions. Still, for some subjects individual preferences could be observed. Differences up to three points in the score could be observed. Again, these differences go equally in both directions. Additionally, when considering the data in the order of recording no correlation between experienced exhaustiveness and the order of the session can be observed.

For the evaluation of the skin conduction, I compared the average measured GSR value during the trials of the two sessions for each subject. As expected there was a trend correlated with the session order. The GSR sensor data showed a reduction in skin conductance between 2% and 30% for 8 subjects from the first to the second session. This is consistent with the literature, as subjects become accustomed to a task the GSR level gradually declines. There were two exceptions: one subject showed an increase of 2% and for one subject the measurement was broken. Regarding the two color conditions no effect could be found. The decline in conductance from the first to the second session is unaffected by the order of the color conditions.

As there are individual differences in accuracy and self-assessed user experience, one might ask whether these are correlated or not. To evaluate this, the Pearson correlation between the difference in classification accuracy and the difference in user score for the two conditions was computed. Subject three was left out because of the aforementioned problem that he was very tired during the second session. Subjects 1 and 6 were left out because I have no valid questionnaire answers for them. For the remaining seven subjects the Pearson correlation between the two differences is quite considerable (0.63), but due to the low number of subjects it is not significant ($p \approx 0.126$). So there might be a correlation that the condition which is perceived more exhausting classifies better, perhaps due to better concentration.

It might also be beneficial to mix different color schemes in one application, for example to maximize contrast between stimuli and background in a dynamic environment. To evaluate this possibility, I trained a classification system with the first 320 items of one session of one user and evaluated it on the last 160 items of the other session of the same user. The classification accuracy is better than chance, but very low. The average accuracy across all subjects was 23%.

5.3.5. Classification by LDA

In chapter 4, I already introduced the use of Linear Discriminant Analysis (LDA) for classification in BCIs. Here, I want to evaluate the use of LDA for cVEP classification. Current cVEP studies have all used SVM classification, mainly due to the very high number of dimensions and a relatively low number of training items. In the current experiment, data was measured with 10 electrodes, one epoch lasted 1 s and the sampling rate was 600 Hz. This means there are 10×600 dimensions. On the other hand, I recorded 480 epochs per session, of which 360 are used for training. The SVM copes relatively well with such situations. Standard LDA does not, because the number of parameters to estimate for the cross-correlation matrix grows quadratically with the number of dimensions and the already unstable matrix also has to be inverted.

Using well-tuned regularization this issue can be remedied. The regularization approach used here was described in detail in section 4.1.4.2 p. 53. The general idea is to tune the cross-correlation matrix by adding a scalar multiplicative of the identity matrix to it before inversion. As implementation I use the regularized LDA implemented in *mlpy* [Albanese et al., 2012].

The parameter λ, which adapts the influence of the regularization term was set to 0.1, meaning quite little regularization. This value was found through manual exploration. The classification accuracy does not vary much with the exact value, although classification fails without regularization.

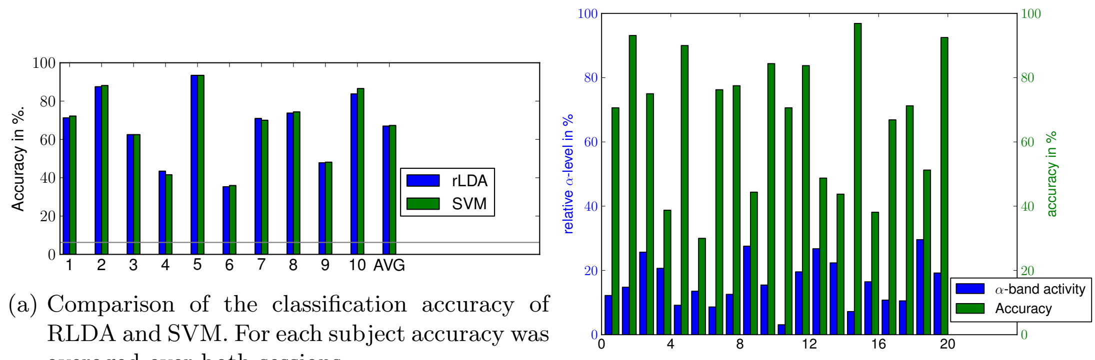
Figure 5.7a shows the classification accuracy using RLDA compared to the results using SVM. The results of the two sessions per subjects were averaged. For all subjects the RLDA achieves very similar results compared to the SVM. On average, the difference is less than one percent, the maximal differences occur for subjects 4 where LDA outperforms SVM by 1.8% and subject 10 where SVM outperforms LDA by 2.8%, all other individual differences are also below 1%.

Additionally, both classifier usually agree on the individual epochs, thus a combination of both does not improve classification.

In summary, classification by LDA is possible and it might be interesting to investigate this further. Still, for the work of this thesis it does not give immediate benefits, so I use the SVM for remaining experiments.

5.3.6. Alpha-Rhythm Amplitude as Indicator of Performance

Still, there is no explanation for the high variance between subjects. To some extent such a variance is normal in BCI studies, but the variance in this experiment is higher than usual. One reason might be different levels of attention and concentration. The questionnaire could not reveal a correlation



(a) Comparison of the classification accuracy of RLDA and SVM. For each subject accuracy was averaged over both sessions.

(b) Comparison of relative α-band activity and classification accuracy for all sessions.

Figure 5.7.

between tiredness before the experiment or interest in the experiment and the classification accuracy. So, I tried to find a correlation between neural indicators of attention and classification accuracy.

There are different neural indicators of attention and a lot of approaches to estimate them based on EEG data, e.g. in [Hung et al., 2010]. I decided to compute the band power of different bands in a short window before each epoch and test whether the band power of one these bands correlates with classification accuracy.

The exact procedure was as follows. A time-window of 0.83 s right before each epoch was extracted from the EEG. Band power was estimated using the norm of the squared FFT coefficients for each of the following frequency bands: $\theta(4-8$ Hz$)$, $\alpha(8-13$ Hz$)$, $\beta(13-30$ Hz$)$,$\gamma(30-70$ Hz$)$. Then the relative band power was calculated as ratio between each band power and the band power of the complete frequency range $(1-70$ Hz$)$. The relative band power is considered more robust, as it is invariant to some degree to general fluctuations in the EEG activity. To compute one value per session, this pre-epoch, relative band power was averaged for all trainings and test epochs of one session. The resulting values for the different channels were inspected and the relative band power of the PZ electrode was chosen for further analysis, as the observed effects are strongest there.

Of greatest interest is the relative α-band activity as this is closely related to mental attention (for more details see section 2.2.4 p. 11ff.). Figure 5.7b shows the relative α-activity and the classification accuracy for each of the twenty sessions. Indeed for many sessions there is a negative correlation, i.e. sessions with relatively low α activity have a higher classification accuracy. For example, for sessions 5, 9, 15 this correlation is rather strong. For some other sessions this correlation cannot be found, for example 2 and 6. Additionally, I computed the Pearson coefficient between the relative α-activity and the classification accuracy. It shows a moderate linear correlation of -0.44 with $p = 0.05$. This negative correlation is consistent with neurophysiological theory, where low α activity means high attention.

5.3.7. Probabilistic Classifier Output

The accuracy in this first study indicates that a cVEP system for navigation and control will need to repeat epochs to reach a reliable control, as it is common in P300 systems. Still, repetitions lower the speed of the system, both in terms of ITR and latency. One does want to repeat as little as possible. This raises the question when to repeat and how often to repeat, as also discussed in section 4.1.5 p. 54. Repeating too little or not repeating incorrectly classified epochs means the system is not reliable, repeating too often or epochs which are classified correctly, lowers speed unnecessarily. To repeat well-directed and purposeful, one needs to asses the probability of a correct classification for a given epoch.

Different approaches exists, for example estimating the SNR using the amplitude of the EEG data. Still, the most straightforward is to use the classification system. The SVM is particularly useful for this, as the distance to the decision hyperplane can be translated to a true probability measure. The

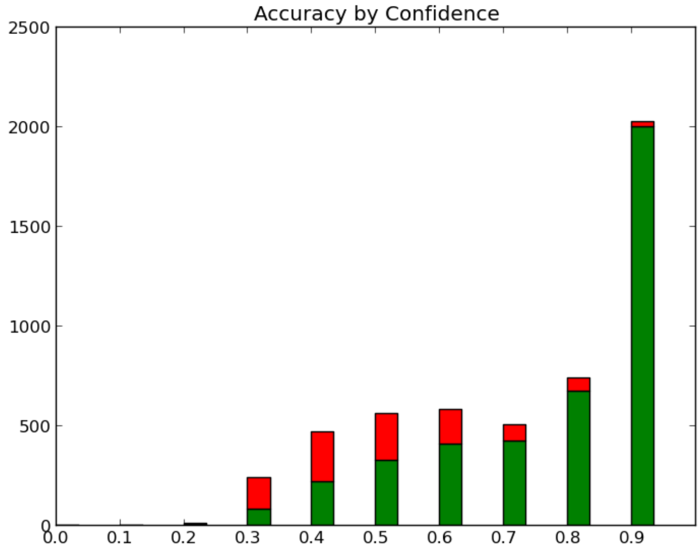


Figure 5.8.: Correlation of classifier confidence and accuracy. Data items are grouped based on the output probability of the SVM. X-axis: output probability. Y-axis: Number of correctly classified epochs in green and incorrectly classified epochs in red.

approach for this translation, which is used in the libsvm implementation is described in [Wu et al., 2004].

To test whether estimating the probability of a correct classification using the SVM works for the cVEP data, I binned the data according to their probability. Ten bins were created, one for epochs with probability less than 0.1, one for epochs with probabilities at least 0.1, but less than 0.2, and so on. Each epoch is classified twice because of the hierarchical codebook. This means, the complete probability would be some combination of the two probabilities from the two classification stages. Here, to test the feasibility of using the SVM output at all, I take each classifier decision individually, which means every epoch is considered twice.

Figure 5.8 shows the result. On the X-axis the different bins are depicted. On the Y-axis the number of epochs is given. For each bin one bar is plotted. The green part shows the number of correctly classified epochs in that bin. The red part shows the number of incorrectly classified epochs in that bin. Two trends show the effectiveness of the SVM probability estimation for the cVEP data. First, most of the epochs get a high probability. Second, when an epoch gets a high probability it is really classified correctly nearly always.

One might impose a threshold at a probability of 70% for example. Repeating every epoch with a probability less than this threshold would mean that few epoch are repeated, but nearly all misclassified epochs.

5.4. Discussion and Conclusion

There are two aspects I want to discuss here. First, did the study fulfill the objectives? Second, what else can the detailed analysis of the data reveal which should be taken into account for the next studies towards a general cVEP-BCI for navigation and control.

Concerning the specific objectives of the study, the results show that the codebook vector can be chosen in a way that makes it possible to very rapidly (250 ms) detect the general area at which the subject gazes, without making the specific classification impossible. The classifier accuracy in detecting the general area in this short time windows was 81%. As stated in the objectives this rapid detection of the general area of the target is very useful for control of semi-autonomous devices like a

BCI-controlled wheelchair. What is more, it greatly increases flexibility in constructing the codebook. There might be situations where it is useful to increase the Hamming distance of the codebook vector of a specific item to all items in exchange for a decrease of the Hamming distances within the other items. For example, when combining navigation symbols with menu items such as "stop".

Still, the total classification accuracy is lower than in other recent cVEP studies. On average the correct target was detected with an accuracy of 67%. In the latest cVEP study regarding spelling the accuracy was about 95% [Spüler et al., 2012b]. I strongly suspect that this is mainly due to the overlapping effect between the different segments of the hierarchical codebook. Additionally, the questionnaire showed that the subjects had to make an effort to keep gazing at the targets without blinking. It is difficult to check whether all subjects were motivated to do so, especially as such an off-line study is not very exciting for the subjects. This might be one reason for the high variance. The result from the α-rhythm analysis support this theory to some degree, as I could show a negative correlation of relative α band power and classification accuracy, which indicates that high mental attention (which results in low α activity) is beneficial for the classification. Another reason might also be a non-optimal flickering pattern.

Next, the results show that a green/red flicker can be detected as reliably as a black/white flicker in terms of classification accuracy. On average both color conditions classify equally well. When looking at the neurophysiology of the human visual system, this is rather unexpected. As early as the retina itself, there are much more cells concerned with black/white vision (about 75 to 150 million rods) as with color vision (7 million cones). This suggests that other color combination could also be possible. I chose red/green because the red/green pathway in the visual system gives the strongest response when measuring brain activity of color pathways [Engel et al., 1997]. Still, one could try the blue/yellow pathway as well or mix colors of two pathways.

Finally, on average both color conditions are rated equally exhausting by the users, a result that is confirmed by the GSR measurements. Still, there are differences for individual subjects in accuracy and questionnaire rating. Interestingly, these results are correlated, i.e. when a subject rates one color condition more exhausting than the other, then the more exhausting condition also classifies slightly better. For practical applications it might be useful to test for individual preferences before fixing the parameters of the system.

When considering the next studies, some additional aspects are also of interest. First, the result when cross-classifying the two color conditions indicates that it is difficult to mix different color schemes. This is consistent with the neurophysiology, as different colors are processed in different areas of the visual cortex [Engel et al., 1997]. But it also means that it is difficult to ensure a high contrast between stimuli and background everywhere. Second, the general classification accuracy hints that either the chosen codebook structure or the hierarchical approach in general lowers classification accuracy compared to the flat codebooks used in [Bin et al., 2009a; Spüler et al., 2012b]. Third, the confidences of the libsvm seems to be very suitable for the implementation of a dynamic repetition scheme. Fourth, replacing the SVM by a regularized LDA would be possible without loosing accuracy. Thus, the better classifier could be chosen, for example in terms of run-time, which is important for an on-line system.

6. Influence of Stimulus and Background Characteristics in cVEP-BCIs

In this second experiment, the user interface is still static, but several stimulus characteristics are already adapted for the need of a dynamic BCI for navigation and control.

Again, I first describe the specific objectives of the experiment, then the methods. Next, the results are presented and finally the results are discussed and some conclusions are given.

Actually, two experimental series were done for this experiment. After the interface was implemented and tested, one experimental series with three subjects was done, but the results were not satisfying. So, I made some modifications on the interface and ran another experimental series. Both versions of the interface are described in this chapter, but only the data of the second series is analyzed.

The data which was recorded during the study is published under the DOI: 10.4119/unibi/2695465

6.1. Objectives

The general objective of this experiment is to implement and evaluate a static cVEP-BCI that, except being static, fulfills the requirements of a BCI interface for navigation and control.

Compared to the first experiment, this general objective, can be divided into two points, which have to be changed: First, the stimuli have to be much *smaller* and rearranged in such way that one has enough space on the screen to later display the scene. Second, one has to evaluate if the signal strength decreases, when using a *non-black background*, which resembles a scene in a navigation and control scenario. Essentially, this background should be non-uniformly colored and look like a lively scene.

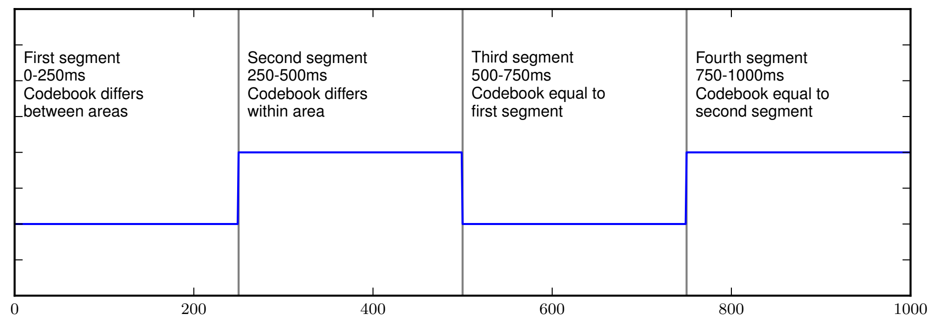
In addition to these changes towards an unconstrained control interface, the results from the first experiment also need to be taken into account. Especially, there seems to be an issue with the hierarchical codebook, which decreases signal strength: The EEG activity containing information about the group membership gets mixed with the EEG activity containing information about the item within the group. Probably this effect cannot be removed completely, but optimization of the codebook might be helpful.

6.2. Methods

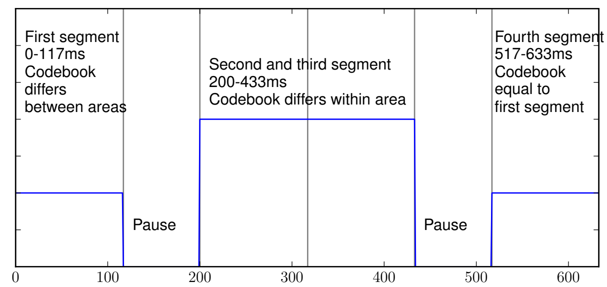
In this section, the specific methods for the second experiment are described. First, the new codebook, the new stimulus presentation and a new flickering method which were derived to build an unconstrained stimulus presentation for control without loosing too much signal strength. Then, the signal processing, experimental setup and experimental procedure are described.

6.2.1. Codebook

In accordance with the first experiment, I used a hierarchical codebook also for the second experiment, but the used m-sequence(s) and the structure of the codebook were changed. For the hierarchical codebook a two-stage classification is necessary. In the first experiment the first stage, i.e. the recognition of the general area or group, performed much better than the second stage, the recognition of the item within the group. In theory, both classification tasks should perform similar. Closer investigation of the data brought up the hypothesis that brain activity containing information for the first stage got mixed with data from the second stage. To mitigate this effect the structure of the codebook was adapted as illustrated in Figure 6.1. For the first experiment the structure was group – item within group – group – item within group. Now, the structure is group – pause – item within group – item



(a) Codebook of first experiment



(b) Codebook of second experiment

Figure 6.1.: The codebook structure for the first two experiments. For the second experiment there was a pause before switching from general area to specific item and back again, and the two segments containing information about the specific items were put together to mitigate the mixing of brain activity observed in experiment 1.

within group – pause – group. By reordering of the blocks and the introduction of some pause frames, I hope to mitigate the mixing problem, which caused a lower classification accuracy of the specific item within a group in the first experiment.

Additionally, I decreased the lag between two stimuli, in my first experiment this lag was four frames per stimuli, now I use two. Preliminary tests showed that there is no considerable difference in signal strength, but the epochs become much shorter thus the latency is decreased.

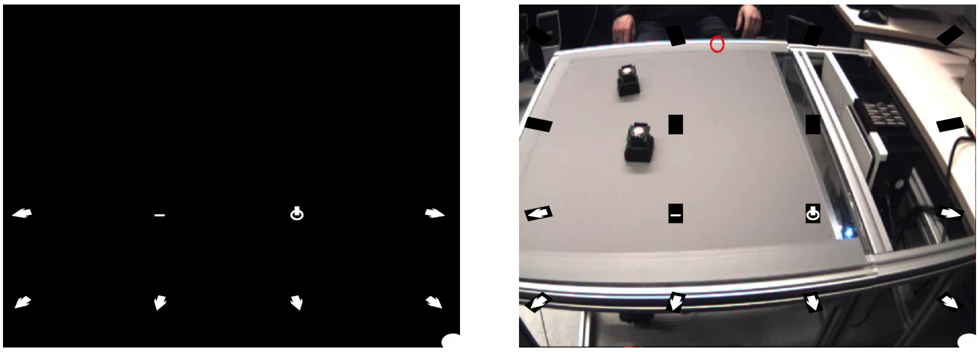
6.2.2. Stimulus Presentation

For the presentation of the stimuli the same software as in the first experiment is used. The arrangement of the stimuli was altered to fulfill the objectives. First, the size of the stimuli was decreased to occupy approximately 2° on the retina, as this is the size of the foveal area [Fairchild, 1998, p.5]. To result in 2° on the retina, the stimuli occupy $\approx 2 \times 2$ cm on the screen. The subjects were seated at a distance of ≈ 60 cm from the screen. This gives:

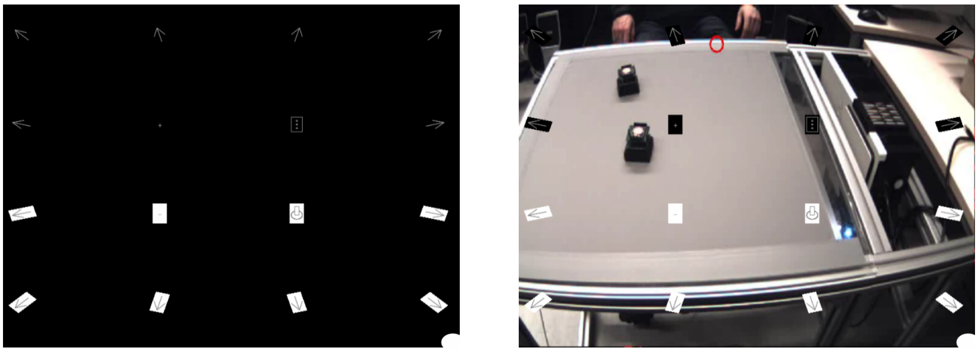
$$arctan(2/60) = 0.0333 \approx 2^\circ \tag{6.1}$$

Additionally, the neighbor stimuli are removed. According to Sutter [Sutter, 1992] they enhance the signal strength, but they occupy too much space. So, this potential drop in signal strength has to be accepted. Removing the neighbor stimuli allows to move the target stimuli to the border. This results in the stimulus presentation which is depicted in Figure 6.2a.

To evaluate the effect of a non-uniform background, I build a second condition where the background is not just black, but a static image which we recorded for another experiment. The image shows the view of a robot during an navigation experiment, effectively simulating a static, but lively background and depicted in Figure 6.2b. One can assume that the background decreases signal strength. The subject allocates resources both on the retina and in the visual cortex to process the background. This probably interferes with the brain activity that contains the information about the stimulus.



(a) No-background condition (b) Background condition



(c) No-background condition (d) Background condition

Figure 6.2.: User interface with the old flickering (a and b) and the new flickering technique (c and d). In the new interface the number of pixels which flicker is increased by far without using more space, at least in the background condition.

This effect is worsened by the properties of a CRT-monitor. Due to the rendering on the monitor the static background is not static, but flickers with the refresh rate of the monitor. This creates a rhythmic brain activity with a frequency equal to the refresh rate of the monitor. To mitigate this effect, I set the refresh rate of the monitor to 120 Hz and render the stimuli twice with each entry from the codebook. Thus, in terms of the codebook and the cVEP, the refresh rate is still 60 Hz, but the background is rendered with 120 Hz.

With the stimulus presentation just described, the results in terms of classification accuracy were not satisfying. The signal strength was too low. To increase signal strength one can increase the number of activated receptors on the retina, by increasing the number of pixels of a stimulus. Still, this should be done without increasing the size of the stimuli. To accomplish this, I do not longer flicker the symbol, i.e. the arrow, but the rectangle around the symbol. The new stimulus presentation is shown in Figure 6.2c and 6.2d. In the condition without background the new stimuli occupy some more space as before. For the overall goal the condition with background is more important though. In this condition a rectangle around the stimulus is needed anyhow to control the contrast locally.

6.2.3. Signal Processing

Signal processing was performed off-line after the sessions. Again, data was segmented, spatially filtered using the CCA and classified by the SVM. Due to the shorter codebook the epochs were also shorter, now each epoch consisted of 633 ms.

6.2.4. Experimental Setup

The experimental setup was similar to the first experiment. EEG data was acquired as described in section 4.3.

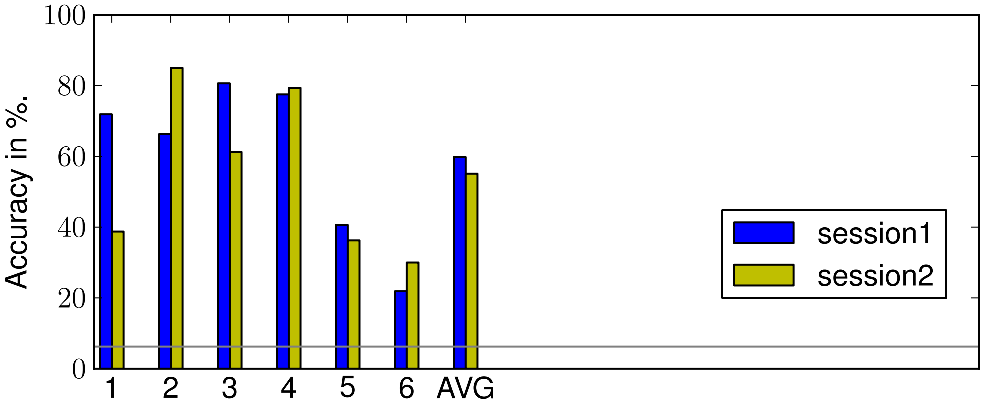


Figure 6.3.: Classification accuracy for the two sessions for all subjects and the average. The gray line indicates the chance level.

Again, subjects were seated in front of the monitor. There, the new stimulus presentation was used to first display the next target stimulus and then render one epoch. No feedback was given. There were two conditions. In the no-background condition the stimuli were rendered over a uniformly black background. In the background condition stimuli were rendered on a static scene image from a robot.

6.2.5. Experimental Procedure

6 healthy subjects completed 2 blocks each, one without background, one with background, in alternating order. Each block consisted of 480 trials. The target was shown for 0.85 s, one epoch lasted 633 ms. No feedback was given. Subjects were acquired from the local student population and our working group. All subject had little or no BCI experience. All subjects signed a written consent and were paid for their expenditure of time.

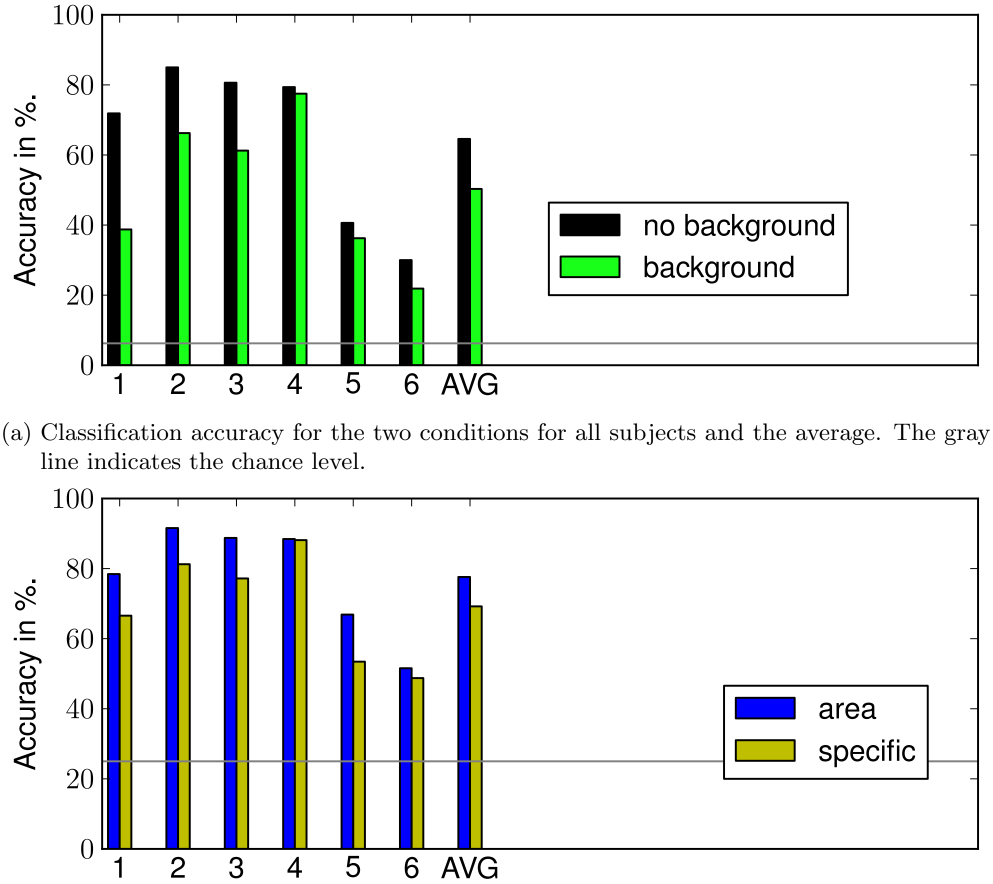
6.3. Results

In this section, the results of the second experiment are presented. The experiment is evaluated based on the classification accuracy. To compute the classification accuracy the first 320 trials of one session are used again to train the classification system of spatial filter and SVM. Next, the classifier is evaluated on the last 160 trials of the same session.

First, the general accuracy is examined, then the accuracy per condition. Finally, the accuracy per classification stage, i.e. detection of the general area versus the item within the area, is analyzed.

6.3.1. General Accuracy

The accuracies of all sessions are depicted in Figure 6.3. In average, an accuracy of about 60% is reached, which is lower than in the first experiment. There is some variance between subjects, in particular between subjects 1 to 4 and subjects 5 and 6. Subjects 1 to 4 all reach an accuracy of 70% to 80% in at least one session, which would be usable in a control interface. For subjects 5 and 6 the accuracy is less than 40% which is still way above chance, but would not be usable in an on-line interface. In contrast to the first experiment, there is also some variance between the two sessions of one subject. In particular, for subjects 1 to 3. For subject 1 the difference is more than 30%. In average, the first and the second session perform equally, so the difference is probably related to the two conditions.



(a) Classification accuracy for the two conditions for all subjects and the average. The gray line indicates the chance level.

(b) Classification accuracy per classifier stage for all subjects and the average. The gray line indicates the chance level.

Figure 6.4.

6.3.2. Background Condition

To evaluate the variance between the two sessions of one subject further, the classification accuracies are grouped by condition as shown in Figure 6.4a. Here, one can clearly see two results. First, in the no-background condition the classification accuracy is similar to the first experiment. Second, the classification accuracy drops in the background condition. The average accuracy drops significantly from 65% to 50% (paired t-test $p \approx 0.03$). The difference is even more pronounced when looking at the well performing subjects 1 to 4, who have an average accuracy of 79% in the condition without background and 61% in the condition with background. For one subject, namely subject 4, the classification accuracy is slightly higher in the condition with background. For this subject the background seems not to decrease signal strength.

6.3.3. Classification Accuracy per Stage

As mentioned in the methods section, the structure of the codebook was changed to improve the classification of the second state in the hierarchical system. To evaluate this, I computed the accuracies of the two stages separately as depicted in Figure 6.4b. Still, the classification accuracy of the first stage is higher than that of the second stage for all subjects. In average, the first stage achieves an accuracy of 77% and the second stage an accuracy of 69%, so the difference is much lower than in the first experiment.

6.4. Discussion and Conclusion

The objective of the second experiment was foremost to have a static BCI interface that combines the speed of the cVEP with a user interface that is flexible and unconstrained enough for applications like navigation and control of robotic devices, except of course being static.

To fulfill this goal the stimuli size decrease to approximately 2 cm on the screen. With this size there is enough space on the screen to display the scene and the stimuli should still cover the 2° on the retina belonging to the fovea. Additionally, the neighbor stimuli were removed.

Both measures lead to a decrease in signal strength. This could be compensated by a new flickering mechanism. Instead of flickering the symbol, for example arrows, a small rectangle around each symbol is flickered, this activates more pixels, which leads to more activated receptors in the retina and higher signal strength. The small rectangle around the stimuli is needed anyhow to locally enhance contrast on a non-uniform background.

The background was the second major step towards an unconstrained interface. In addition to the standard condition with black background I evaluated a condition with a static color image from a robot camera. This leads to a significant decrease in signal strength, which must be compensated in the final interface somehow.

In total, on the one hand this experiment shows the encouraging result that smaller stimuli and the removal of the non-target stimuli can be compensated well. The resulting accuracy would be fine for an on-line interface. On the other hand, a background that emulates a scene of a robotic control scenario decreases signal strength considerably. The resulting accuracy would be barely useful in an on-line interface. Thus, a way to mitigate the negative effect of the background is needed.

7. An On-line cVEP Speller with Dynamic Repetitions

In this third experiment, my cVEP-BCI system is evaluated in an on-line spelling experiment.

The software implementation for this experiment as well as the execution of the evaluating study was done by Ingo Killmann as part of his bachelor thesis. All the results which are presented here are calculated by myself on that data set.

In agreement with Ingo Killmann, I decided to build an on-line spelling system as a further step towards an on-line system for navigation and control.

Next, the objectives are elaborated in more detail, then the methods for this experiment are described, especially the on-line stimulus presentation and the dynamic repetitions. Finally, the results of the study are analyzed and discussed.

For the explanations of the dynamic repetition mechanism, it is helpful to remember a term, which was introduced in section 4.1.5 p. 54, the *decision*. Basically, a *decision* results in one command sent from the BCI to the target system, in this case one letter sent to the speller. One decision consists of one or more epochs. The results of these epochs are combined to compute the command of the decision.

7.1. Objectives

This experiment serves three purposes. First, a first test of my BCI system in an *on-line* setting. Second, an evaluation of *dynamically changing stimuli*. Third, a test of a *dynamic repetition mechanism*.

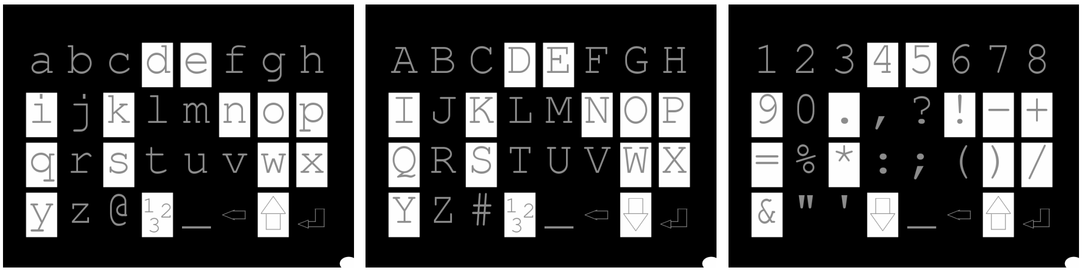
For a cVEP-BCI the brain activity depends on the flickering pattern of a stimulus. So far, my experiments explored different influences on signal strength, like color and size. Another question is the influence of shape. Due to the new flickering mechanism introduced in the last experiment, the number of pixels which are flickering does not depend on the shape of the stimulus. Thus, one can assume that the shape of the stimulus does not influence the signal strength. This is to be evaluated in this experiment.

Next, a mechanism for using dynamic repetitions is to be implemented and evaluated. The motivation to use dynamic repetitions is three-fold. First, reliability. Classification of EEG data never works perfectly. Repeating epochs increases classification accuracy. Second, adaptivity, repetitions increase latency and lower bandwidth. So, we try to only repeat when necessary, i.e when the epoch cannot be classified with high certainty, for example because of external noise or fading concentration of the subject. Third, asynchrony. We do not want to issue commands when the user is not gazing at a stimulus at all, e.g. when thinking about what to do next. So, when classification does not reach high certainty even with some repetitions, the system assumes that the user does not want to issue commands at the moment.

Finally, the study is to be done on-line. So far, my cVEP-BCI studies were off-line studies. Before building the final system, an on-line study is needed to evaluate the effects of on-line feedback.

7.2. Methods

Here, the methods for this experiment are described. First, the stimulus presentation, which uses three different pages of symbols. Then, the mechanism for the dynamic repetitions and, finally, the experimental setup and procedure.



(a) First page of symbols (b) Second page of symbols (c) Third page of symbols

Figure 7.1.: All three pages of the stimulus presentation for the spelling experiment.

7.2.1. Codebook

For the first two experiments I used a hierarchical codebook. After careful consideration, I decided to switch to a *flat codebook*. Although the classification accuracies of the first two experiments were in the 60% to 70% range an increase in signal strength is needed. Especially, because signal strength was much lower when using a non-uniform background.

When looking at previous studies from other groups, one can assume that a flat codebook increases signal strength in comparison with a hierarchical codebook.

The lag between the stimuli was set to two frames. This results in a system comparable to [Spüler et al., 2012b].

7.2.2. Stimulus Presentation

To evaluate the influence of different stimuli shapes in a speller system, the system has different pages of symbols. This is similar to a virtual keyboard on current smart-phones. In total, the system allows to input 82 different symbols on three pages with 32 stimuli each (see Figure 7.1). Every page contains stimuli to change to the other pages, the underscore which inputs a space, backspace to delete the last character and enter to end a sentence. Additionally, the first page contains small letters (a-z) and the @-symbol, the second page contains capital letters and the #-symbol, the third page contains numbers and several special characters.

Except for the new symbols, the stimulus presentation works as in the experiments before. The size of the stimuli was increased again to be comparable with other cVEP-BCI spelling systems.

During the on-line phase, the system showed the text which was input so far on the screen, above the stimuli.

7.2.3. Signal Processing

Classification of the individual cVEP epochs was done in the usual three steps: First, the EEG-data was bandpass filtered. Second, it was spatially filtered. Thirdly, it was classified by a linear OCSVM.

Bandpass filtering and spatial filtering by the CCA were done as in the first and second experiment.

The classification is adapted to the flat codebook. It is now done as described in section 4.1.3.6 p. 51. This means all the training data vectors are first converted to class 0. For example, the codebook-vector of the stimulus of class 5 is shifted by 10 frames, compared to the codebook vector of stimulus of class 0. This means the corresponding brain response is shifted by a time-lag of 167 ms. To convert an EEG data vector of class 5 to class 0 it is circular back-shifted by this amount of time. To classify a new data vector, one copy of the data vector is created for every class and back-shifted by the lag of that class. The copy that is considered to be closest to the trained class 0 by the OCSVM gives the label of the epoch. The distance to the decision hyperplane of each copy is forwarded as 'score' to the dynamic repetitions module, which computes the label of the decision.

7.2.4. Dynamic Repetitions

To implement a dynamic repetitions mechanism ons needs to asses the certainty of the classification. I decided to use the *distance to the decision hyperplane* of the OCSVM.

Using the distance to the hyperplane is motivated in general by the maximum-margin approach of the SVM. The main assumption to motivate the maximum-margin approach is that a higher distance between the hyperplane and the nearest points of the training data increases generalization, i.e. the classification accuracy of new data. By extending this assumption, one can argue that the classification of new points is correct with higher probability when these points are further away from the decision boundary. For the cVEP hierarchical codebook I presented also some experimental validation in my first experiment (section 5.3.7 p. 71). Probably these results also hold for the flat codebook.

As we have more than two classes, the classification score of an epoch is calculated as difference of the distances to the hyperplane of the best and second-best class.

When an epoch has been classified and this difference computed, this difference is called the *score* of the best class. When this score reaches a predefined threshold, the decision is considered to be complete and the command corresponding to the best class is sent to the speller. When the threshold is not reached, the next epoch is rendered and classified. The scores are summed up over the epochs per class until the threshold is reached for one class or the maximal number of epochs is reached. When the threshold is reached the corresponding command is sent. When the maximal number of epochs is exceeded the decision is dropped, i.e no command is sent and the scores are reset.

7.2.5. Experimental Setup

The experimental setup was similar to the first and second experiment. EEG data was acquired as described in section 4.3. Again, subjects were seated comfortably in front of the monitor.

During training, the new stimulus presentation was used to display the next target stimulus and then render one epoch. No feedback was given. After each 32 items the system switched to pause mode and the subjects had time to relax.

During the on-line run an epoch was rendered and classified. As long as the threshold of the dynamic repetitions mechanism was not reached new epochs were rendered and classified until either the threshold was finally reached or the maximal number of repetitions was reached. In both cases the scores were reset and a new decision started. When the threshold was reached, a pause of 1.66 s was inserted before the next decision, so the subject could observe the feedback and gaze at the next letter to type. This cycle was repeated until the end of the experimental block.

When the threshold of the dynamic repetitions mechanism was reached the decision was considered to be successful. When the chosen stimulus was one of the two “page switch” stimuli the page of the stimulus presentation was changed and the next decision started. When the chosen stimulus was the backspace the last letter of the text was deleted. Otherwise the new letter was added to the text.

7.2.6. Experimental Procedure

6 healthy subjects were recruited for the study. Subjects were acquired from the local student population and our working group. All subject had little or no BCI experience. All subjects signed a written consent. One data set had to be discarded because the photo-sensor failed due to a low battery. Each subject completed 3 blocks, one off-line training block, two on-line evaluation blocks with different thresholds for the dynamic repetitions.

The training block consisted of 480 trials. The target was shown for 0.83 s, one epoch lasted 1.05 s. No feedback was given.

In each on-line block each subject had to write two sentences three times each. The two sentences were “Mama backt 1A Kuchen!” and “Ein V8-Motor ist laut.”. One of the two blocks was done with a threshold of 3000 for the dynamic repetitions mechanism, the other one with a threshold of 5000. The order of thresholds was alternated between subjects.

Table 7.1.: General results for the speller experiment for all subjects and the average.

Subject	1	2	3	4	5	AVG
Off-line accuracy (%)	91.3	89.4	95.0	100.0	93.8	93.9
On-line accuracy (%)	89.5	93.7	98.3	94.7	98.8	94.8
Time per sentence (min)	3.05	2.75	2.47	1.63	2.18	2.42
Time per letter (sec)	7.0	6.4	5.7	3.8	5.0	5.6
Error-free letter per minute	8.5	9.4	10.5	16.0	11.9	10.8

7.3. Results

The on-line blocks are of main interest to evaluate the study in terms of the objectives. Still, the off-line data is analyzed as well, mainly for comparison with other studies.

So, I first look briefly at the off-line data, then the on-line data is analyzed in detail. There, the general accuracy is analyzed. Then, the efficiency of the dynamic repetitions mechanism is analyzed. Finally, the accuracy for the different pages is compared to evaluate the effect of different shapes.

7.3.1. Off-line Accuracy

For the off-line training block, I computed the classification accuracy. The first 360 trials were used to train the classifier system. The accuracy of this classifier on the last 180 trials are depicted in Table 7.1 - First row. The average accuracy is 93.9%. With 1.88 s per trial, this would mean a theoretical ITR of 139.11 bit/min. There is moderate variance across subjects. For all subjects the accuracy is above 90%.

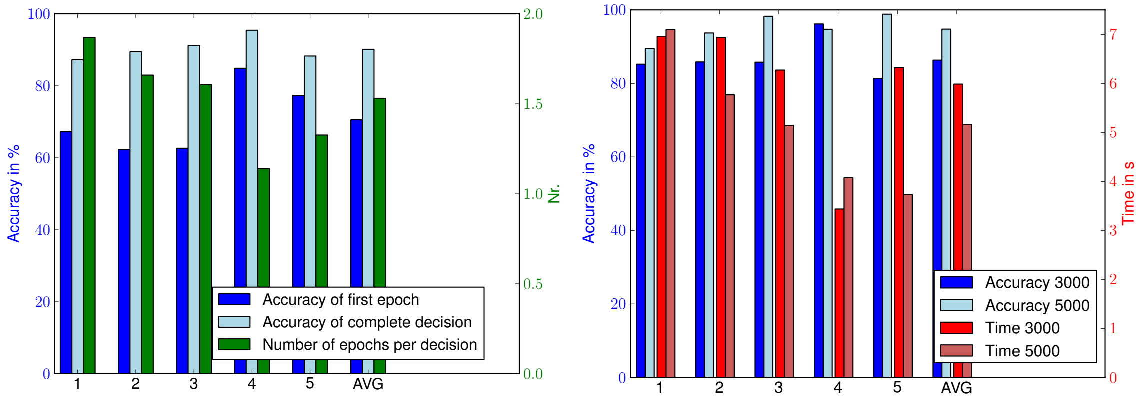
7.3.2. On-line Accuracy

For the on-line blocks, subjects were instructed to write error-free sentences. Two kinds of errors can occur. First, the system can make a classification error. Second, the subject can gaze at an incorrect stimulus. Only the first kind of errors were to be corrected. The second kind had to be indicated to the experiment supervisor. For the evaluation of the accuracy only the first kind, i.e. errors caused by the BCI were considered. Additionally, due to the instruction to write error-free sentences the number of trials per subject varies, as every system error makes two additional decisions necessary, one to delete the wrong letter, one to write the correct one.

In Table 7.1 - Second row the accuracy during the on-line run is depicted. The average accuracy is nearly 95%. All subject have an accuracy above 90%, except for subject 1 who has an accuracy of nearly 90%. In total, the variance is quite low.

When looking at the average time needed per sentence for each subject, the result is somewhat different. The times vary between 1.6 min per sentence and 3.1 min per sentence. This indicates that the dynamic repetition mechanism partly compensates for differences in signal strength by using more repetitions. Based on the time per sentence one can also compute the time per letter. For this metric the page-switches are also counted as letters. The time per letter or command varies between 3.8 s and 7 s. For latter comparison I computed also the opposite, the number of error-free letters per minute, which has an average of nearly 11 letters per minute.

Compared to the off-line block, the on-line condition is much more difficult. Several epochs are classified incorrectly because of the timing. Subjects have to think which symbol has to be activated next and have to find that symbol within the 1666 ms after feedback is given and the next epoch is started. Still, the on-line accuracy is similar and sometimes even higher than the off-line accuracy. To understand this effect one has to analyze the effects of the dynamic repetitions.



(a) Classification accuracy for the first epoch and the complete decision, in relation to the average number of repetitions per decision.

(b) Classification accuracy for the two different thresholds (3000 and 5000) in relation to the average time per letter.

Figure 7.2.: Results of the dynamic repetition mechanism.

7.3.3. Dynamic Repetitions

To evaluate the dynamic repetitions mechanism, two things are to be investigated. First, I compare the effects of the dynamic repetitions vs. a theoretical system with no repetitions. Second, I analyze the influence of the threshold.

For the first part, I compare the classification accuracy of the first epoch with the accuracy of the complete decision and the number of epochs per decision, all depicted in Figure 7.2a.

The data shows that the accuracy of the first epoch alone is much lower than the off-line accuracy, including quite some variance. The dynamic repetition mechanism correctly compensates for this by using more epochs per decisions for subjects with low signal strength. This can be seen well, when comparing subject 3 and 4. Subject 3 has a low accuracy on the first epoch, thus a high number of epochs per decision (1.6) and reaches more than 90% for the accuracy of complete decisions. Subject 4 has a high accuracy of the first epoch, thus only needs 1.14 decisions on average. Still, there is no direct correlation between accuracy of the first epoch and number of decisions. Subject 1 has a slightly higher accuracy on the first epoch, but slightly lower number of epochs than subject 3. Subject 1 also has a quite low accuracy on the complete decisions.

The efficiency of the dynamic repetitions can also be seen when examining the scores of the first epochs. On the one hand, 79% of the correctly classified first epochs had a high score and were accepted without further repetitions. On the other hand, 75% of the incorrectly classified first epochs had a low score and additional epochs were added.

To compare the efficiency of the two thresholds, the accuracies for the complete decision per threshold and the resulting time per letter are depicted in Figure 7.2b.

As can be expected, the accuracies are higher when using the higher threshold, especially for subjects with relatively low signal strength, for example subject 5. On average the accuracy is at 86% for the threshold 3000 and 95% for the threshold 5000. The only exception is subject 1 where accuracies for both thresholds are relatively low. Probably an even higher threshold would be necessary for him.

When considering the time per letter the situation is more complex. For most subjects the higher accuracy of the higher threshold also means less time per letter, despite the higher number of repetitions. This is understandable as each error results in two complete, additional decisions (one to delete the wrong letter, one to type the correct one again). This needs quite much time compared to an additional epoch. Still, for subjects 1 and 4 the time per letter is lower for the lower threshold. For subject 1 this is an effect of the little increase in accuracy. For subject 4, who has a very good signal strength, the accuracy using the lower threshold is already very high ($> 96\%$), so additional repetitions cannot increase the accuracy much, they increase only the time per letter.

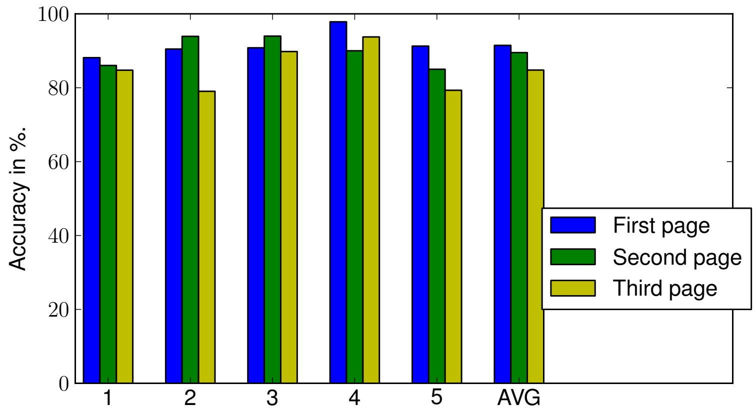


Figure 7.3.: On-line accuracy for each of the three pages separately. Classifier training was done on data of first-page stimuli only.

7.3.4. Influence of Shape

This study was also done to evaluate the effect of different stimulus shapes. Therefore, during acquisition of the training data only the stimuli on the first page were used. Now, the on-line accuracy is computed separately for each of the three pages and depicted in Figure 7.3. On average, stimuli on the first page have a slightly higher accuracy than stimuli on the second page, but this difference is not significant (paired t-test: $p > 0.45$). On average, the stimuli on the third page have a lower accuracy than those on the other pages (85% vs 91% and 90%) and this effect is narrowly significant (paired t-test first page vs third page: $p \approx 0.046$). Still, both effects are not consistent across subjects. Especially the accuracy of stimuli on the third page has a quite high variance. Partly, this might be attributed to the relatively low usage of the third page, subjects might fail to find the correct symbol in time. For subject 2 for example the accuracy of the second page is higher than the first page, but the accuracy of the third page is much lower than the other two. Then, for subjects 1 and 3 all threes accuracies are close together.

7.4. Discussion and Conclusion

For this study an on-line cVEP spelling system was implemented and evaluated.

During the off-line block the average accuracy was 93.9% for epochs of only 1.05 s. Compared to the first two experiments the signal strength was improved by far. Two changes in the experiment design can be accounted for this. First, compared to the second experiment the size of the stimuli was increased again, roughly to the stimulus size of the first experiment. This was done to be comparable with other cVEP spelling studies, but also increases signal strength. Additionally, a flat codebook was used for this experiment. This increases signal strength a lot. Also, letters were used instead of navigation symbols, but the effect of shape is supposed to be negligible. Compared to other cVEP-BCI studies on spelling systems, the system proposed here, reaches similar speed and accuracy during the off-line phase, as state-of-the art systems. For example, Spüler et al. had an average accuracy of 92% without special, subject-specific optimization of the spatial filtering and 96% with it [Spüler et al., 2012a].

During the on-line spelling of predefined sentences a similar accuracy of 94.8% was accomplished, allowing for fluent typing. Still, the speed decreased. One decision (letter or special character) took between 4 s and 7 s. This results from the repetitions which are necessary to accomplish the same accuracy in the more difficult on-line setting.In particular, subjects have to think about the next letter to type and find the matching stimulus in time.

Compared to other non cVEP-BCIs this is still fast. For example, the auto-calibrated P300-speller [Kaufmann et al., 2012] which was discussed in section 2.4.1 needed 22.3 min to spell one sentence of 45-characters (12.4 min with predictive text entry). Here, the average time for a 21-character sentence was 2.42 min.

The only other study doing on-line spelling with a cVEP-BCI was faster though. With their system around 20 letters per min can be written. One should note though that their system is designed to maximize speed, where as I wanted to test different prerequisites for the system for navigation and control. Some easy-to-implement measures could improve the speed of our system, like training the subjects more, so they intuitively know where which stimulus is placed. Additionally, a reduction of the pause after each input would improve speed much.

In particular, the study tested the feasibility of dynamic repetitions for cVEP. Actually, this mechanism has to fulfill two contrasting objectives. It has to ensure a high accuracy despite varying signal strength, and the maximal possible communication speed. As mentioned, the accuracy of the on-line with an average of 94.8% is very good. Even the subject with the lowest signal strength has an accuracy of nearly 90%. Additionally, only 21% of the correctly classified first epochs were repeated, but 75% of the incorrectly classified first epochs. This shows that the mechanism is also effective in terms of speed. An open issue is the selection of the threshold. Here, two fixed thresholds were tested, but fixed thresholds are clearly suboptimal. A subject-specific threshold, maybe based on the accuracy during the off-line run, would be better.

Finally, another objective of this study also was to investigate the influence of shape on the signal strength. Here, the result is less clear. I investigated the accuracy of the stimuli of the three different pages. The classifier was only trained on stimuli of the first page. The accuracies of stimuli on the first and second page is equal, indicating that the shape of the stimulus does not influence the signal. Still, the accuracy of the stimuli on the third page is lower. This could be caused by two factors. First, the stimuli on the third page are more distinct in shape, but, second and probably more importantly, the special characters on the third page are in no self-explanatory order and a relatively seldom used. Both increases the probability of the subject not finding the correct stimulus in time. So, in terms of stimulus shape, one can cautiously say that the study supports the theory that shape has little influence on the brain activity.

8. An On-line cVEP-BCI for Fast and Flexible Control

For this final experiment I built a BCI system designed for the control of a virtual agent based on the cVEP. This system provides a flexible design, which is easily adaptable to different control tasks, a high number of commands (up to 16 for now) and low latencies around 2 s. Using dynamic repetitions of epochs the system provides asynchronous control and adapts to the signal strength of each user.

Based on the cVEP-BCI, the system enables the user to move the virtual agent freely within a virtual environment, where the agent can perform some actions. As example scenario I use a virtual kitchen with actions like getting coffee from the cupboard or switching on the coffee-machine. This interfaces gives fine-grained control, where desirable (e.g. for navigation) and semi-autonomous actions, like "Switch on coffee-machine", where appropriate. The results of this experiment are also partly published in [Riechmann et al., 2014].

The data which was recorded during the study is published at pub.uni-bielefeld.de

8.1. Objectives

The objectives for this experiment can be grouped into two parts. Objectives to be met by the developed BCI system in general and the specific objections for the scenario which I use as an example to evaluate the BCI system on.

8.1.1. cVEP-BCI System for Navigation and Control

First, I want to implement a *flexible and general* cVEP-BCI system for navigation and control that is as fast as possible.

In fact, due to using the cVEP the system is supposed to be faster than any other BCI for navigation and control. Still, it will be slower than a cVEP system for spelling. This holds for all kinds of brain activity, spelling systems have a higher information throughput than the more complex systems to control devices. For navigation and control speed from a users perspective is mainly related to latency. How long does it take for the device to react, after the user has made a decision on the next action to take? For this system one epoch should take less than one second and a complete decision about two seconds. Again, a decision means one or more epochs which results in one command sent from the BCI to the target system, as introduced in section 4.1.5 p. 54.

Next, the system should support a wide-range of scenarios. Current BCI systems for navigation and control are often tweaked towards a particular scenario. For example, the scene might be mostly static or only rendered in black-white. For this new system every kind of scene can be used. It renders the scene image and, on top of this, the stimuli, without any restriction about color, contrast and so on.

To support different scenarios the number of different commands needs to be sufficiently high and needs to be dynamic. Current BCI systems for navigation and control often support only 4-6 commands, whereas even in the limited interaction of current computer games far more commands are used. For now, I think that 10-20 commands are necessary to support different scenarios in a flexible manner. Additionally, the shape of the commands should also be dynamic, i.e. it should not be necessary to acquire training data for each command. Ideally, the brain activity does only depend on those characteristics of the stimuli which are not used by the user to identify the command corresponding to the stimulus. Then, one can acquire training data on one (small) set of commands and use the classifier on another (larger) set, like I did in the previous experiment.

There is one last aspect to address about flexibility concerning the scenario: An adaptable degree of shared control. Here, shared control means that the BCIs user and the internal control logic of the robotic device share control as discussed in paragraph 2.4.3.ii p. 21. There is always a dilemma when dealing with shared control. When the device has little autonomy the user can precisely control it, but has to issue many commands which makes interaction slow. When the device has high autonomy the interaction is faster, but the user looses precise control. This dilemma has to be dealt with mainly when designing a specific application for a specific scenario. Still, the system in general supports this by offering dynamic commands.

Finally, the system needs to adapt for varying signal strength. When the signal strength is high and the classification is very reliable, control is relatively easy, but in BCIs using surface EEG, there are almost always sessions and subjects where the signal strength is low. Partly, this has to be dealt with using suitable classification algorithms, but another important measure is dynamic repetition. This allows to lower speed and increase latency to compensate for low signal strength, without decreasing accuracy too much.

8.1.2. The Test Scenario

Second, I want to implement one *show-case scenario* for the general system and evaluate the feasibility of the system using this scenario. Mainly, this scenario has to be designed to evaluate the system and show in how far the system meets the described objectives.

To ease implementation and more importantly evaluation, the scenario consists of a virtual agent who does navigation and manipulation in a virtual environment. From the BCI perspective there is not much difference whether to control a real robot or a virtual agent. In general, the virtual agent is more detached from or not as immersive to the user as a concrete, real device and the virtual agent makes less errors. When using a BCI the control is always kind of indirect, though. The BCI user usually sees the robot and/or scene only indirectly through cameras and monitors. Additionally, the control strategy usually decouples BCI and device also on implementation level. The BCI analyzes the brain activity and sends commands to the device. The device sends, in response or asynchronously, part of its state and the state of its surroundings back to the BCI. So, errors in one system do not interfere directly with the other system.

As a future goal, the developed system should assist patients with their daily life. So, the scenario has to evaluate the system in a simulated, everyday-life situation, where an avatar could perform manipulations the patient cannot do himself. The scenario also has to test different levels of shared control showing that the system is suitable for this.

Finally, the scenario should include different situations in general. For example, in terms of the surrounding scene (colors and contrast), but also in the accuracy needed to fulfill a sub-task. First, this serves to show the robustness of the system. Second, this allows to examine in more detail the effect of color and contrast of the background on the signal strength.

8.2. Methods: The cVEP-BCI

In this section, the cVEP-BCI for navigation and control is described in detail. As mentioned before (section 2 p. 7) a BCI system usually consists of four parts: Data acquisition. signal processing, output device and communication protocol. For BCIs using external stimuli one also needs a stimulus presentation, usually combined with the output device somehow.

The actual system structure of my system is depicted in Figure 8.1. There is the data acquisition, the user interface, signal-processing and protocol. The user interface displays the scene with the device. When a new decision starts, the stimuli are rendered on top of the scene. The user looks at the scene, decides for a command to send and gazes at the corresponding stimulus. This evokes a certain brain activity. The data acquisition sends EEG data, which is acquired from electrodes over the visual cortex, to the signal processing. Additionally, the user interface sends the time of the start of the epoch, and information about which stimuli are currently visible to the signal processing. The signal processing derives the intended command from the EEG data and sends the command to the device. The device executes the command which induces changes in the scene. The new state of the scene is processed by the user and the next decision starts.

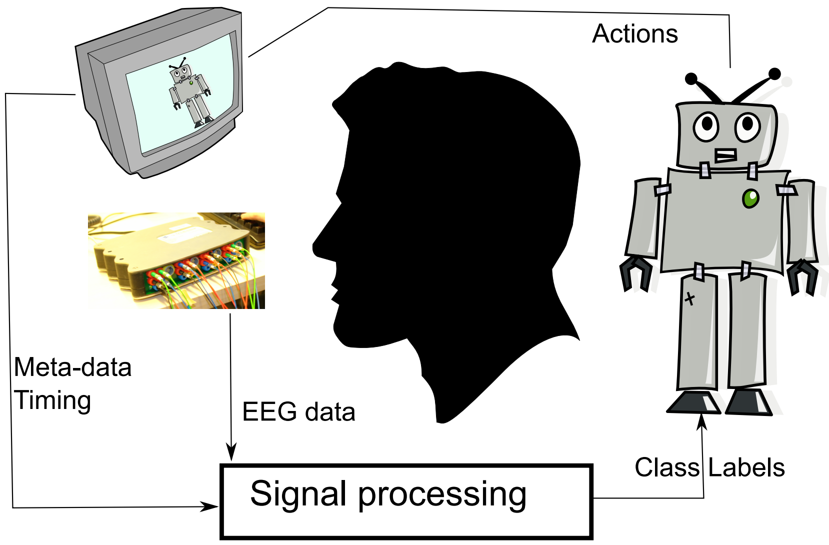


Figure 8.1.: System structure and data flow of the final cVEP-BCI for navigation and control. Head, monitor and robot from openclipart.org

The user interface is described in detail in the next two subsections. The signal processing is divided into two parts. As usual there is feature extraction and classification. Additionally, there is the dynamic repetitions mechanism. These two parts are described after the user interface. Finally, in the last subsection, the communication protocol is described. The protocol defines the time-course when using the system. At which time epochs are generated, how the data flows, when commands are sent to the device and so on. The data acquisition is not covered here as its exact details are not relevant for the system.

8.2.1. Codebook

At first, one needs to design a codebook for the cVEP stimuli, because the codebook indirectly defines several parameters for the system like the length of the epochs.

As mentioned in chapter 5, a hierarchical codebook would be helpful for navigation, because knowing very early the group of the attended stimulus could be used to initiate movement preparation early. Unfortunately, the results of experiments 1 to 3 show that a hierarchical codebook decreases the signal strength strongly. So, I decided to use a flat codebook. A hierarchical one might be interesting future work. The next decision is how many frames to shift between stimuli. Here, I chose a lag of 2 frames. Experiment 3 and preliminary tests of the navigation system indicate that the signal strength for 2 or 4 frames lag does not differ much. So, a lag of 2 results in shorter epochs with similar signal strength.

As I needed at most 16 classes and decided for a shift of 2 bits, the codebook vectors need 31 bits. The codebook vectors for the first two classes which were actually used in the study are shown in Figure 8.2, the complete codebook is presented in the appendix Figure A.2.

8.2.2. User Interface

The user interface consists of the scene, where the to-be-controlled device operates in, and the stimuli which are drawn on top of the scene. Scene and stimuli were presented on a 19" CRT running at a resolution of 800x600 pixels. For the rendering the irrlicht[1] engine was used. In other cVEP systems each codebook frame is rendered once, usually at a monitor refresh rate of 60 Hz. Here, the monitor is set to a refresh rate of 120 Hz and every codebook frame was rendered twice. Otherwise the 60 Hz flickering of the scene would create interfering brain activity and distract the subject, as was discussed in section 6.2.2 p.76.

[1] irrlicht.sourceforge.net

[1, 1, 0, 0, 1, 1, 0, 1, 0, 0, 1, 0, 0, 0, 0, 1, 0, 1, 0, 1, 1, 1, 0, 1, 1, 0, 0, 0, 1, 1, 1]
[1, 1, 1, 1, 0, 0, 1, 1, 0, 1, 0, 0, 1, 0, 0, 0, 0, 1, 0, 1, 0, 1, 1, 1, 0, 1, 1, 0, 0, 0, 1]

Figure 8.2.: The codebook vectors for the first two classes used in the experiment.

To balance signal strength and flexibility the stimuli have to be as large as necessary for a good signal, but also as small as possible. To accomplish this, the stimuli have to occupy about 2° on the retina. When subjects are seated at a comfortable distance of 60 cm from the screen, this gives:

$$sin(2^{\circ}) * 60cm \approx 2.09cm \tag{8.1}$$

So, the stimuli are built to occupy 2×2 cm on the screen.

Two kinds of stimuli are supported. Stimuli with a fixed 2D position on the screen and stimuli which are fixed to positions in the 3D scene of the irrlicht engine. The first kind of stimuli is useful for commands that are always available and independent of the situation in the scenario, for example general commands like start, stop or commands for navigation. The second kind is useful, when some action of the device can only be used in certain situations. For example, the device might have some preprogrammed autonomous actions like open the window, or there might be commands like switch on TV which are only useful when the TV is in view. These kind of stimuli are called *action stimuli* for the remainder of this chapter. For every epoch and every action stimulus, the system calculates whether the corresponding position in the 3D scene is currently within the camera view of the user. The stimulus is only rendered when this is the case. It is rendered at the corresponding position of the object. The matching 2D position on the screen is calculated using the 3D engine.

8.2.3. Signal Processing

Classification of the individual cVEP epochs is done in three steps: First, the EEG-data is bandpass filtered. Second, it is spatially filtered and, thirdly, classified by a linear OCSVM. All three steps are performed as it was done in the third experiment.

Bandpass filtering is done in the acquisition hardware, using a 1 Hz high-pass filter and a notch filter around 50 Hz.

To construct a spatial filter a Canonical Correlation Analysis (CCA) is performed, as described in detail in section 4.1.2 p. 43. CCA is designed to find projections for two datasets in such way that the projections have maximal correlation [Weenink, 2003]. For the spatial filtering of cVEP data, one finds projections to maximize the correlation between the EEG data and the average of the EEG data to use this projections as a spatial filter which cancels noise, originally proposed in [Bin et al., 2009a].

This was implemented as follows. Let there be n vectors of training data, each with m samples and k channels. The data matrix X is constructed by stacking all training data vectors and has the dimensions $(n \cdot m) \times k$. The template matrix Y is constructed by averaging over all trials and stacking the average n times. Now, the CCA is used to find the projections a, b with $argmax_{a,b} corr(Xa^t, Yb^t)$. I used the MATLAB implementation of the CCA for this. a^t is used as a spatial filter, which means for every data epoch x, $x \cdot a^t$ is computed and used for the classification.

Classification is done by a linear one-class SVM (OCSVM), according to section 4.1.3.6 p. 51. As each stimulus is flickering according to a shifted versions of the codebook for class 0, the brain activity produced by attending the different stimuli is also a shifted version of the brain activity when attending to the stimulus of class 0. Thus, by shifting the data back, one can obtain a one-class problem. Each training data vector is back-shifted according to its label to obtain a vector of class 0. Then, an OCSVM is trained on all these vectors. When a new vector arrives for classification copies of this vector are created and shifted for each possible class. Each copy is then classified by the OCSVM. Finally, the distance to the decision hyperplane of each copy is forwarded as 'score' to the dynamic repetitions module, which computes the label of the decision.

In the actual implementation I use the libsvm [Chang and Lin, 2011] with a linear kernel and the cost parameter set to 1.0.

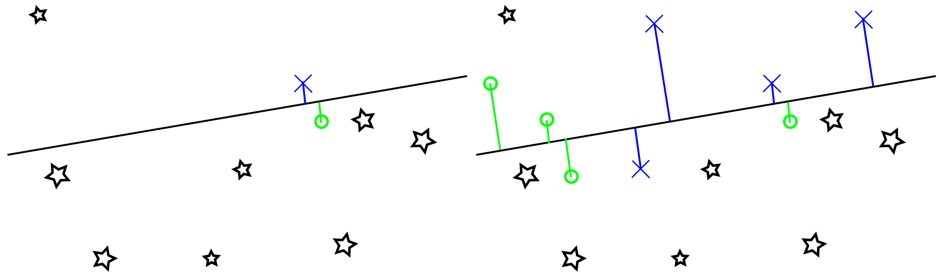


Figure 8.3.: Example of the classification using an OCSVM and dynamic repetitions. Training data depicted by star. Two possibles class, i.e. shift factors depicted by crosses respectively circles. Left: situation after one epoch. Right: after four epochs.

8.2.4. Dynamic Repetitions

My motivation to use dynamic repetitions is three-fold. Reliability, i.e. increasing accuracy by repetition. Adaptivity, i.e. only repeat when necessary, to preserve speed. Third, asynchrony, i.e. no commands are to be issued when the user does not want to issue commands. A more detailed motivation is given in section 4.1.5 p. 54.

For these three goals I need to asses the certainty of the classification. I decided to use the distance to the decision hyperplane of the OCSVM. An illustration is given in Figure 8.3. The OCSVM finds the decision hyperplane which encapsulates a given percentage of the training data (the stars in the figure). After the training, one decision is classified. Two different shifting factors (i.e. two different classes) are considered, depicted by crosses respectively circles. At the moment of the left illustration, one epoch has arrived. The two copies of this epoch with different shift factors are classified by the OCSVM. Then, the system need to asses which copy fits better to the one class learned from the training data. In the example this would be the copy depicted by a cross. Both copies are close to the decision hyperplane though. Therefore, the probability of a misclassification is considered quite high. The system waits for more epochs. After four epochs the situation is illustrated again. To combine the results for the different epochs, all the distances to the hyperplane for all epochs are summed up per class.

The distance is considered as positive for copies on the correct side of the hyperplane and negative for copies outside the class boundary. This sum is called the score of the class. Figure 8.4 illustrates possible results. On the left, after one epoch, both classes have a similar score, which means the system waits for more epochs. In the middle, after four epochs, the difference between the classes is high and the system outputs class 1 as label for the current decision. On the right, another decision is depicted after five epochs. Here, the scores are similar even after five epochs. This can have two possible reasons, either both classes get a similar score due to artifacts in the data or because the subject did not attend a stimulus at all. In either case the system drops the current decision, all scores are reset, no label is output and the next decision is started.

In my application the system has to classify more than two classes. To copy with that, I use the difference between the best and the second-best score to asses the classification certainty. When this difference is higher as a subject-specific threshold, the system considers an epoch to be classified with enough certainty.

Naturally, the success of this mechanism strongly depends on the choice of a fitting threshold. For a similar mechanism, which I used for the third experiment, I evaluated two different, subject-independent thresholds. The results showed that the resulting accuracy and speed strongly depend on the chosen threshold, but also indicated that a fitting threshold has to be subject-specific. Still, the problems remains how to rate the fitness of a threshold. Somehow one needs to balance accuracy and speed. For a spelling system one could use the threshold that maximizes the ITR on some subject-specific benchmark data.

For the current system I designed the following mechanism to compute the optimal threshold: A data set of labeled data is needed. This dataset has to consist of decisions with a fixed number of repeated epochs. Basically, every possible threshold is tested to find the one threshold that maximizes

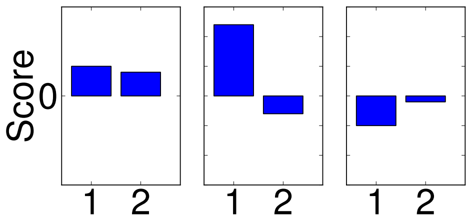


Figure 8.4.: Three possible situations when comparing the score of two classes. Left: after one epoch the difference is low. Middle: after four epochs the difference is high and the classification is considered certain enough. Right: in this decision, the difference is low even after five epochs, no labels is output for this decision.

the accuracy of the system. To compute the accuracy for each threshold, that threshold is used as parameter for the dynamic repetition mechanism and the classification accuracy of the system on the labeled decisions is computed. A decision that does not reach the threshold after the maximal number of repetitions is classified as "no command" and counted as incorrect classification for that threshold. Thus, a higher threshold means more repetitions, resulting in higher accuracy values, but when the threshold becomes too high, several trials are considered as "no command" which lowers the accuracy value of that threshold.

To sum it up, when the classification works well, the differences are high and few repetitions are necessary. When the classification does not work very well, repetitions are necessary but the user can still exercise control over the system. And, when the user does not want to issue commands, the differences of the scores are small and commands are generated rarely.

8.2.5. Communication Protocol

The communication protocol defines the interaction cycle between the user, the BCI system and the target device as well as within the BCI. On the highest level the interaction consists of an ongoing sequence of *decisions*. For each decision the user gazes at the stimulus which corresponds to the command he or she wants to send. The BCI analyzes the brain activity the whole time, discarding decisions as long as the user does not gaze at a stimulus. When the user gazes at a stimulus, the BCI detects which stimulus the users gazes at and sends the corresponding command to the target device. The device realizes the command, resulting in a change within the scene. This is seen by the user who initiates the next decision based on the current situation of the device. To improve signal strength the user interface only initiates a new decision after the movement of the device is finished, i.e. the protocol defines that either the stimuli are rendered and brain activity is analyzed or the device is moving, never both together at the same time.

On a lower level, each decision consists of one or more *epochs*. For each decision an epoch, i.e. one cycle through the codebook, is rendered with all stimuli which are currently active. The epoch is classified and assessed by the dynamic repetitions mechanism. As a result, the dynamic repetition mechanism can order another epoch for this decision, end the decision and send the command or end the decision and drop it. When another epoch is ordered that epoch is rendered, classified, and so on. When the decision ends, the next decision is started.

8.3. Methods: Kitchen Study

To evaluate the system a specific scenario is needed. As outlined in the objectives section, the scenario consists of an avatar which is controlled by the user and an every-day-life situation, where the avatar performs a task for the user. To avoid the need of having a humanoid robot to serve as avatar, the scenario is completely virtual, i.e. a virtual agent acts in a virtual surrounding.

Next, I describe the actual scenario in detail, then I describe how I deal with artifacts in this study. Afterwards the set-up and procedure of the study are presented.

Figure 8.5.: A screen-shot of the user interface. It shows the eight navigation arrows and two action stimuli, one at the cupboard and one at the coffee-machine. Each stimulus consists of the actual symbol in gray, which is always rendered and a white rectangle which is only rendered for frames where this stimulus is active. Each stimulus is surrounded by a small black rectangle to maximize contrast and to have similar conditions regardless of the position in the kitchen.

8.3.1. Setting

As an example situation, I chose a virtual kitchen. The kitchen was designed based on a free kitchen model[2] to resemble an ordinary kitchen. The task of the avatar is to produce some virtual coffee in this kitchen.

The scene is seen from a first-person perspective of the virtual agent as depicted in Figure 8.5. For the navigation within the scene there are 8 arrow stimuli on fixed 2D positions. 4 to move around (forward, backward, right, left) and 4 to look around (up, down, left, right). When a movement command is given, the virtual avatar moves a short distance in the corresponding direction. When a look-around command is given, the camera is rotated by about 20° in that direction. This interface allows to navigate with great flexibility similar as in first person computer games.

Additionally, 4 action stimuli are used. These stimuli are attached to objects in the virtual world, where the agent can perform autonomous actions. In the kitchen these objects are: oven, coffee in cupboard, coffee-machine and sink. The corresponding actions are "get cake from oven","get coffee from cupboard", "put something in coffee-machine" or "start coffee-machine" (depending on the state of the machine) and "get water from the sink". When the user selected an action stimulus the virtual agent moves to the corresponding object and executes the action autonomously[3].

8.3.2. Artifact Detection

EEG data is always contaminated with artifacts, where artifact means electrical sources which contribute to the signal, but which are not related to the target brain activity. This can be either external sources of electrical fields, like electrical devices in the surrounding or muscles of the subject, or internal sources in the brain like the motor cortex. The problem of artifacts was discussed in detail in section 2.2.4 p. 13ff.

Very few studies on visually-evoked potentials do consider artifacts, because artifacts are considered less of a problem for VEP-BCIs than for other types of BCIs. This is also why I did not consider artifacts for the first three experiments. For this last experiment I want to at least show that ocular and movement artifacts do not contribute to the classification, i.e. there might be artifacts, but they are not helpful for the detection of the correct target.

[2]The model can be found at: http://lvlworld.com/media/id:1868

[3]In this virtual scenario, the autonomous execution of the action is just indicated by a text message displayed to the user.

To identify epochs which are contaminated with noise, a simple threshold approach is adopted. All epochs containing samples which are outside of a ± 50 mV range are considered mildly contaminated. Epochs with samples outside a ± 75 mV range are considered highly contaminated. Despite its simplicity, a simple threshold based detection is still the most commonly used technique [Fatourechi et al., 2007]. The threshold of ± 75 mV is also taken from the literature [Nolan et al., 2010].

It is important to emphasize that I do not use artifact rejection during the experiment, only artifact detection later off-line to check for possible influence of artifacts.

8.3.3. Experimental Setup

The experimental setup was similar to the previous experiments. EEG data was acquired as described in section 4.3 p. 59ff. A gUSBAmp (Guger Technologies) amplifier was used. 10 channels were equipped with Ag/AgCl electrodes placed at P3, Pz, P4, PO3, POz, PO4, PO7, O1, O2, PO8 according to the extended international 10-20 system and referenced at Oz. Electrode impedances were kept below 5 kΩ. The amplifier sampled the EEG data at 600 Hz, performed high-pass filtering at 1 Hz and notch filtering at $48 - 52$ Hz.

Subjects were seated comfortably in front of the monitor at a distance of ≈ 60 cm.

After the experiment each subject filled out a small questionnaire, the original questionnaire is given in the appendix Figure B.2 p. 124. An English translation is given with the answers in the result section.

8.3.4. Experimental Procedure

To evaluate the on-line control system a study with eight healthy subject was conducted. All had normal or corrected-to-normal vision and all were recruited from the local student population or our working group. All had little or no BCI experience. Each subject did one session with three blocks: The off-line block, the on-line evaluation block and the free run block.

i. Off-line block In the off-line block 480 trials were recorded. Before each trial, the target stimulus (one of the eight navigation arrows) was shown for 1.25 s. Then, one epoch was rendered. Afterwards, the corresponding command was executed. No on-line classification was done. The virtual agent always executed the correct command, subjects were informed about this. This way, they could see the effects of the navigation commands during training. Additionally, the targets were chosen in such way, that for every 32 trials the agent moves a certain path through the kitchen. This not only showed the subject the kitchen. It also ensures that the contrast and color of the background differ for the different training data vectors. Thus, the classification system can find features in the data which are independent of the characteristics of the background.

After every 32 trials the system switched to a pause mode and started no more trials. To start the next run of 32 trials the subject had to press the space key. Thus, subjects could pace the training themselves and relax from time to time.

The first 320 of the 480 trials were used to calculate the CCA projection and train the OCSVM. The last 160 trials were used to evaluate the off-line performance. The trained classification system was also used for the other two blocks.

ii. On-line evaluation block In the on-line evaluation block 160 decisions were recorded. Again, one of the navigation arrows was shown for 1.25 s as the target. Then, five epochs were rendered with a gap of 0.05 s between the epochs. All five epochs were classified and their scores were summed up as described in section 8.2.4 p. 93. The command with the highest score was executed by the interface providing on-line feedback. Again a pause was done every 32 trials.

This block serves three purposes. First, to gather some labeled data to evaluate the on-line accuracy, i.e. to examine the effect of feedback. Second, some subject-specific, labeled data with repetitions is necessary to compute the threshold for the dynamic repetitions. Third, I also need labeled data with repetitions to evaluate the accuracy and effectiveness of the dynamic repetition mechanism.

iii. On-line free-run block In the third block, the subjects could move around freely in the virtual scene. No targets were given. Subjects were only instructed to produce cups of virtual coffee. For each coffee, a sequence of five actions had to be executed by activating the corresponding action stimuli (grab coffee from cupboard; put coffee into coffee-machine; get water from sink; put water into coffee-machine; switch on coffee-machine). The objects and action stimuli were shown to the subjects beforehand, but no further explanation was given, in particular there was no predefined path to follow. In effect, subjects had to move and look around in the scene to make the next action stimulus appear. Then, the stimulus could be activated and the virtual avatar would perform the action autonomously. After the action had been performed, the system would start the next decision and subjects would steer the agent to the next corresponding object. Users were not instructed how to solve the task. They could look around to see the next object and activate it. The virtual agent would then move autonomously to the object. Alternatively, they could move to the object using the navigation controls and then activate it.

In the BCI system, the dynamic repetition mechanism was activated for the third block. Its threshold was computed using the 160 trials from the second block as input. The maximal number of repetitions was set to five, i.e when the threshold was not reached after five repetitions the decision was dropped and no command was issued. Additionally, a minimal number of repetitions was used. This means that the threshold was first checked after the minimal number of epochs was rendered and classified. In effect, this avoids wrong classification when the first epoch is classified wrongly and also has high confidence. Even when this happens for more than one epoch, it is unlikely that the wrong labels agree. When the labels do not agree the resulting difference in score would be low and no command issued. The minimal number of decisions was set to two.

When the last action of the five-action sequence was activated by the user and executed by the agent, one run, i.e. one virtual coffee was completed and the system switched to the pause mode and reset the agents position. To complete the study each subject had to do five runs.

iv. Keyboard control As no predefined targets are given during the third block, it is not possible to directly asses the classification accuracy. To have a baseline I constructed a comparable keyboard interface for the coffee scenario. The twelve possible BCI commands were replaced by twelve keys on the keyboard. An action command could only be triggered when the corresponding object in the scene was within the view of the virtual agent, as in the BCI condition. The keyboard control condition was not used by the subjects, it was only used to compute the baseline.

8.4. Results

When analyzing the results, I use the same block structure as in the experiment. So, I first briefly look at the off-line data investigating the classification accuracy and possible effects of the background. Then, the data from the second block is evaluated with a focus on the efficiency of the mechanism for dynamic repetitions. Next, I deeply investigate the data from the free run. Additionally, I present the results of the questionnaire and some analysis on possible artifacts in the data.

One subject had to be left out of the analysis, because his data did not contain classifiable signals. Later analysis indicated that the electrode cap was positioned incorrectly and no activity of the visual cortex could be picked up.

8.4.1. Off-Line Block

During the whole off-line block the target is predefined and the virtual agent always executes the command which corresponds to the target. This means the virtual agent navigates on a predefined path through the virtual kitchen. Therefore, the data used for the training of the classifier consists of epochs with varying background. This also holds for the last third of the epochs which are used for an off-line evaluation of the classification accuracy. Here, two aspects are investigated. First, the general accuracy, than the influence of background on the accuracy.

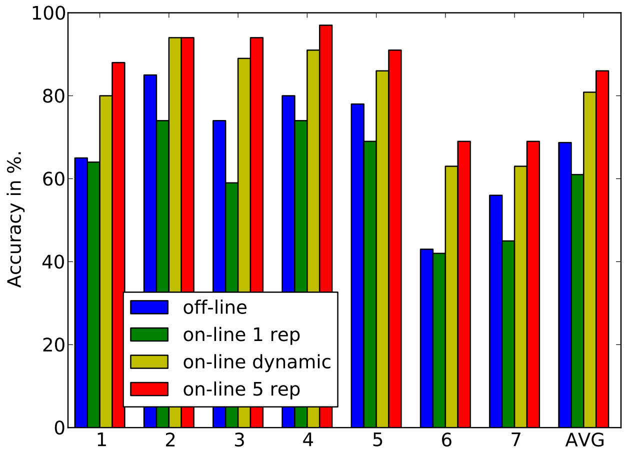


Figure 8.6.: Classification accuracy for the first and second block. Four conditions are depicted for each subject and the average: Accuracy during first block without feedback in blue. Classification accuracy during the second block for the first epoch in green, (hypothetical) accuracy if the dynamic repetitions mechanism had been used in yellow and actual classification accuracy with five repetitions in red. Feedback in the second block was given based on all five repetitions.

i. Classification accuracy The classification accuracies of the first and second block are depicted in Figure 8.6. In the off-line block an eight-class problem had to be solved by the classifier, because the predefined target would always be one of the navigation arrows and never an action stimulus. In average the accuracy was 69%, but there was a high variance across subjects. Subjects 2 to 5 reached good classification accuracies around 80%. Subject 1 had 65% accuracy and the other two had accuracies below 60% where no control a of a system is considered possible, i.e for them repeating epochs is essential.

ii. Influence of contrast and color To investigate the influence of contrast and color, I investigate the accuracy of the different epochs of one run. As mentioned before, the training stops and goes into a pause mode after each 32 trials. Each of these runs of 32 trials starts at the same position and follows the same path. Within one run each position and orientation is used twice, because the agent goes back and forth, but the indicated target is different for the two times the agent visits a certain position. The complete accuracies per position averaged over subjects are given in the appendix Table C.3 p. 126.

Here, I search for correlations between color and contrast of the background and the accuracy. Actually, I compute the Pearson correlation between the average accuracy on a given position and different statistics about the image shown on the screen during that epoch[4].

First, the standard deviation of the gray-scale value across the pixels of the image. The resulting Pearson coefficient is -0.39 with $p \approx 0.028$, i.e. there is a moderate negative correlation. So, a high variance in luminance lowers the signal strength and the accuracy.

For further investigation, I transformed the image into the HSV color space and compute the Pearson correlation between the standard deviation of the pixels for each of the three values and the accuracy. The Pearson values are -0.42, $p \approx 0.016$ for the hue, -0.31, $p \approx 0.086$ for the saturation and -0.40, $p \approx 0.024$ for the value. So again, less variance in the image means higher accuracy of the classification.

I also computed the Pearson coefficient between the mean of the gray-scale image and the means of the H, S and V value of the one side and the accuracy on the other side, all these correlations are weak and not significant.

Finally, I also investigated whether the specific color has an influence. Again, I computed the Pearson coefficient between mean and standard deviation of the image and the accuracy, only this time mean

[4]These statistics are computed on a screen-shot taken from the first frame of the corresponding epoch.

and standard deviation are computed on the color channels of the RGB image. The mean values show no significant correlation with the accuracy, but the standard deviation does. The coefficients are -0.39, $p \approx 0.026$ for red, -0.38, $p \approx 0.03$ for green and -0.37, $p \approx 0.039$ for blue. These values indicate that the signal strength is lower when there is high variance in a color channel, but all the colors have a similarly strong effect.

8.4.2. On-Line Block

In the second block, the users have somehow to follow a predefined path. Before each decision, a target is announced. When the classification is successful, the virtual agent follows the same path as in the off-line block. When a classification error occurs the robot leaves the specified path, but the subject still gets the same target stimuli to attend to. This means that the scenario is much more restricted than the third block, but it allows to compute the classification accuracy and to evaluate the dynamic repetitions.

i. Accuracy First, I calculate the accuracy during the second block considering only the first epoch of each decision (depicted as second bar in Figure 8.6). The average accuracy is 61%. Again, there is some variance across subjects. The considered accuracy roughly correlates with the accuracy during the off-line block, but is slightly lower.

To compute the feedback during the block, all five epochs per decision were combined using the dynamic repetition mechanism, i.e. the mechanism was used without threshold by setting the minimal and maximal number of repetitions to five and the threshold to zero. The resulting accuracy is depicted as the fourth bar in Figure 8.6. The classification accuracy is around 90% for subject 1 to 5 and 70% for subjects 6 and 7, the average is 86%.

ii. Dynamic repetitions Based on the data of the second block, the best possible threshold for the dynamic repetitions is computed. This is later used for the third block. Additionally, it allows to evaluate the effectiveness of the mechanism. First, I computed the accuracy and number of repetitions using the computed threshold on the second block[5]. This accuracy is depicted as third bar in Figure 8.6. The average accuracy is 81%, the average number of epochs per trial is 2.6. So, with roughly half the number of repetitions the average accuracy drops only by 6%. Approximately 1.5 s are necessary per decision (in average 2.6 epochs of length 0.52 s plus some computational time for the classification).

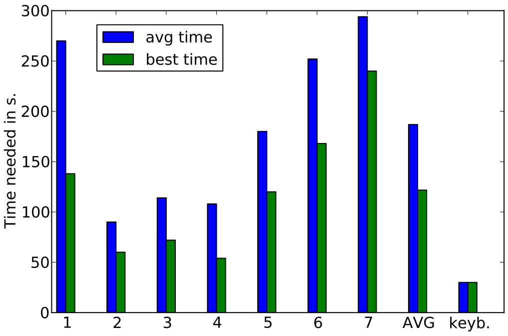
Again, the mechanism is quite effective in recognizing difficult decisions. After two epochs, 75% of the correctly classified decisions are accepted and 70% of the incorrect classified decisions are not accepted. After three epochs already 84% of the correctly classified decisions are accepted and still 63% of the incorrect classified decisions are not.

8.4.3. Free Run

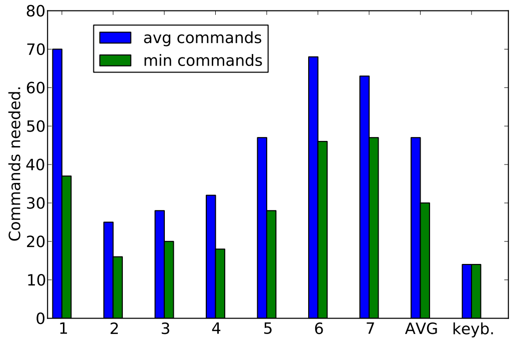
To show that an interface works one has to show that the subjects can achieve some kind of task with it. After the previous results showed that a predefined target stimulus can be identified correctly by the system, this section deals with the evaluation of the third block, where the subjects decided themselves which commands they wanted to emit to fulfill the given task. The values for the keyboard control were acquired as the best possible run executed by myself.

The focus for the evaluation of the third-block data is on the time which was needed to accomplish the task. This measure balances accuracy and latency, similar to the ITR for spelling system. Still, it is not comparable between studies as the time also incorporates several parameters which depend on the scenario, for example the time which is needed by the agent to execute the actions. Thus, I also look at the number of commands which were sent in a given run from the BCI to the virtual agent, which gives kind of an accuracy-like metric. Additionally, I investigate whether the data shows a learning curve per subject and what influence contrast and color might have.

[5]One has to be careful here, as the threshold is optimized and evaluated on the same data, but the analysis still shows the potential of dynamic repetitions with an optimal threshold.



(a) Average and best time per run for every subject and a theoretical keyboard control.



(b) Average and minimal number of commands per run for every subject and a theoretical keyboard control.

Figure 8.7.

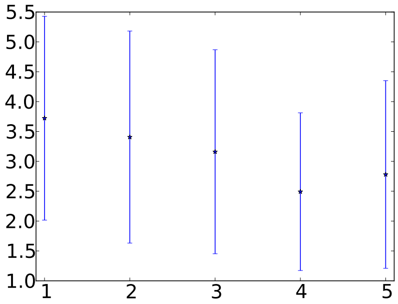
i. Time needed and number of commands Figure 8.7a shows the time needed for a run per subject and for the keyboard interface. For every subject the average and the lowest time is depicted. For the keyboard best time and average time are set to the best time achieved by me.

When using the keyboard, the activation of commands takes virtually no time, the 30 s stem from the time the virtual agent needs to execute the commands. Naturally, these 30 s are also part of the time needed by the subjects in the BCI condition. In average, 3 min were needed with the BCI and 2 min was the average for the best run of each subject. Again, the variance is quite high. Subjects 2 to 4 achieve very good results of less than two minutes in average and around one in the best run. So, these three subjects need only half a minute to issue the necessary BCI commands to move around in the kitchen and activate the five actions[6]. The other subjects need some more time, but all seven subjects successfully completed all five runs.

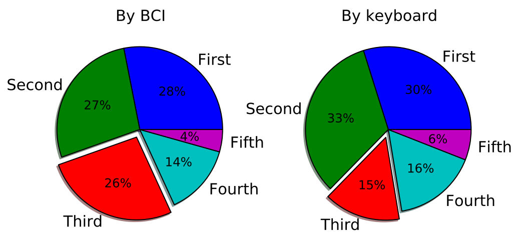
To squeeze out the effect of the movement time of the agent, I also examined the number of commands sent using the BCI interface and the keyboard control, see Figure 8.7b. With the keyboard control 14 commands are necessary, five for the five actions, nine to move and look around. Using the BCI, subjects 2 to 4 needed less than 20 commands for the best run and less than 30 in average. Subjects 1 and 5 needed less than 40 commands in their best runs and subject 6 and 7 around 45. The difference to the keyboard condition is the summation of not-optimal commands issued by the subject and wrongly recognized commands by the BCI. For subjects 2 to 4 this difference is small, especially in their best runs, when they had some experience with the task. For the other subjects the difference also decreases when the subjects become accustomed to the task, but even in their best run the BCI made some errors in recognizing the command.

ii. Learning effect In general, the difference between best and average time for producing a virtual coffee is quite high, especially for subjects 2 to 5. To asses this, I computed the time per run averaged over subjects for each of the five runs and depict this in Figure 8.8a. A clear trend can be seen that later runs take less time, with the first run taking one and a half times more time than the fourth run. The fifth run takes a bit more time than the fourth, but still less than the rest. In total, there seems to be some kind of learning effect, as the subjects have to get used to the task and the unfamiliar interface. To examine this statistically I computed the ratio of time needed for each subject, i.e. for every subject the times of all five runs are divided by the time of the first run. A paired t-test on the transformed times shows that the fourth run took significantly less time than the first ($p \approx 0.028$). For the fifth run vs. the first the t-test shows a trend ($p \approx 0.07$). The complete original and transformed data can be found in the appendix tables C.1 and C.2 p. 125.

[6] As with the keyboard control, another half a minute is needed by the agent to move.



(a) Time per run in minutes averaged over all subjects with standard deviation.



(b) Commands needed per action compared between BCI control and manual control. The third action was highlighted because it takes more commands in the BCI condition as one would expect from the keyboard condition.

Figure 8.8.

iii. Effects of background contrast and color I already did some evaluation about the effect of the background on the data from the off-line block. Still, the path of the virtual agent through the kitchen during the off-line run does not cover all positions and rotations which are needed during block three for the production of the virtual coffee. In this unconstrained setting, some subjects had the impression that the BCI worked better in some areas of the kitchen and not so good in others. So, I analyzed the number of commands spent for the 5 distinct actions of each run to evaluate this. Figure 8.8b shows the number of commands needed for the five actions, compared between BCI control and keyboard control, in percentage. When looking at the percentages of the keyboard control condition, one sees, for example, that fewest commands are needed for the fifth action (actually only one), because after the fourth action “put water into coffee-machine” the action button of the coffee-machine is already visible and just needs to be activated again. On the other hand, for the first action the virtual agent has to be turned several times in order to be able to activate the “grab coffee from cupboard” action. If the probability of a correct classification of the interface had been equal everywhere in the kitchen scene, the distribution of commands over the five actions in the BCI condition would have been similar to the distribution in the keyboard condition. Instead, the figure shows that action three (getting water from the sink) needs a much higher percentage of commands in the BCI condition compared to the keyboard condition. Visual inspection shows that the sink and the area around this are much darker than the rest of kitchen, a screen-shot is depicted in the appendix Figure A.3 p.122. Additionally, this part of the kitchen is not part of the path the agent moves during the training. One might speculate that the low performance of the BCI in this area is caused by the classifier not having training examples for this background condition.

iv. Effects of shape During training, only a subset of the stimuli were used, only the navigation stimuli. The action stimuli were added later in the free run only. So, the stimulus set used during training and the set used during the free-run differ, allowing some analysis of possible of effects of stimulus shape on signal strength. No quantitative evaluation can be done, but the analysis of commands per action can be used for a qualitative analysis. The last action only consisted of one command: “Activating the coffee-machine”. This action does need as much commands as expected, thus at least this action stimulus classifies with a similar accuracy as the navigation symbols.

8.4.4. User Experience

Each subject had to fill out a small questionnaire directly after the experiment. The questions and answers can be seen in Table 8.1, for every question the subjects had to give an answer from 1 (very low) to 6 (very high)[7].

The first two questions asked about the mental condition of the subjects before the experiment. A

[7]The original questions were in German, for this work they were translated by the author. The original questionnaire is given in the appendix Figure B.2.

Table 8.1.: Answers to a questionnaire, which was filled out by the subjects directly after the experiment. Possible answers ranged from 1 (very low) to 6 (very high).

Questions/ Subjects	1	2	3	4	5	6	7	AVG
Before the experiment I felt stressed.	2	1	1	2	3	1	2	1.7
Before the experiment I felt relaxed.	5	5	1	4	3	6	4	4.0
The experiment was exhausting.	3	4	2	5	2	4	2	3.1
The experiment was exciting.	6	5	5	5	5	6	6	5.4
The experiment was tiring.	2	2	3	4	2	6	4	3.3
I had the impression that I could control the system reliably.	4	5	4	4	4	5	4	4.3
When the preparations were not so time-consuming I could image to control PC/Video-games with this system.	4	5	5	3	6	5	6	4.9

feeling of being stressed might lower concentration and signal strength, but our results do not support this. There is no correlation between users mental state before the experiment and signal strength.

The next three questions dealt with the mental state during the experiment. The answers show that the experiment was not too exhausting (mean: 3.1) and not too tiring (mean: 3.3). When answering these questions several subjects stated that the first phase without feedback was more exhausting than the two phases with feedback. Additionally, the experiment was rated quite interesting (mean: 5.4).

For the question regarding the impression of control (mean: 4.3) the scores are all above medium, even for subjects were the BCI worked not very reliable.

An even higher score (mean: 4.9) was given to the question whether subjects could imagine to play games using a BCI. Of course, this question is highly biased as all subjects participated because of interest in BCIs, but still the question shows that their interest was not dimmed after their first-hand experience with BCIs.

8.4.5. Artifacts

Here, I asses a possible influence of artifacts on the signal strength by computing a separate accuracy for contaminated epochs from the off-line block. Actually, there are very few epochs which exceed even the lower threshold of ± 50 mV. Of all the 7×160 trials for the evaluation of the off-line accuracy only 42 are considered as being contaminated. These all belong to the data of subject 4. In general, subject 4 has a good signal strength. The off-line accuracy is 80%. This is good, but not excessively good. The accuracy of the 42 epochs is 78%. So, there is no notable effect of artifacts on the signal strength. Only 4 epochs exceed the second threshold of ± 75 ms. All of these are classified correctly, but the small number prohibits any statistical evaluation.

8.5. Discussion

As in the result section, I structure the discussion along the three different blocks of the study. Additionally, after the discussion of the study, I also discuss the system in general.

8.5.1. Off-Line Block

The results of the off-line block give rise to two discussion points. First, they allow to compare the current experiment with my previous experiments and other off-line experiments. Second, they allow to asses the effect of the background further.

I start by comparing the current result with my spelling experiment. The classification accuracy is much lower as in that experiment. Even though the classification system now needs to solve an 8-class problem instead of a 32-class problem in the speller experiment. This shows the negative impact of the smaller stimuli and the changing background on the signal strength. Both issues were also evaluated in my second experiment. There the stimuli were also small and in one of the two conditions there was a static image as background. This condition resulted in a very low signal strength, much lower than in the current experiment. Thus, the change from the hierarchical codebook, which was used in the first two experiments, to the flat codebook, which I use now, seems to increase the signal strength a lot. The classification accuracy in experiment 2 with the static background image was too low for on-line control. Now, even with a varying background an average accuracy of 69% is reached. This is better than the single-trial accuracy of comparable P300 systems, which use somewhat longer epochs, e.g. [Ganin et al., 2013].

Due to the varying background, the current experiment delivers also additional information about the influence of the background. The second experiment showed that a non-uniform background has a negative effect on the signal strength. The current experiment showed that this effect is stronger when the background has a high variance in the luminance, as derived by the variance in gray-scale values. Additionally, I could show that the specific color has little effect.

8.5.2. On-Line Block

The results from the on-line block show that the classification of cVEPs with a varying background is feasible, even when the agent does not follow the same path it followed during the acquisition of training data. This happens always when a trial was incorrectly classified. Despite this effect the classification accuracy using one epoch is only slightly lower than in the off-line condition (61% on average vs. 69% in the off-line condition).

Additionally, the results show the effectiveness of the repetitions mechanism for a cVEP for navigation and control. With a static number of five repetitions an average accuracy of 86% is reached. Although five repetitions is quite a lot, due to the short epoch length, the whole decision needs only 2.8 s.

With dynamic repetitions this can be further improved. With a subject-specific threshold an average accuracy of 81% is reached for decision length of about 1.5 s.

Finally, there are two subjects with a noticeable lower signal strength. Even with five repetitions the system is not able to detect the correct target with more than 70% accuracy. For comparability between the subjects I kept the parameters fixed even for these subjects. In a real daily-life application one would need further checks. Maybe another electrode arrangement would improve their signal strength. Alternatively, one could increase the maximal number of repetitions.

8.5.3. Free Run

The free run was intended to show that the subjects can achieve a daily-life goal with the system. Indeed, all seven subjects were able to complete the five runs of producing virtual coffee.

Three of these seven subjects had a reliable control over the system and, after some runs, needed only marginally more commands as in an ideal run with a comparable keyboard interface. Additionally, the system worked really fast for these subjects, they needed about 30 s to issue the about 15 commands to reach the goal. For a twelve-command BCI in a completely unrestricted settings, this is much faster than other approaches.

Additionally, a learning effect could be observed for all seven subjects. They needed some runs to understand the task, find a suitable solution and to get used to the unusual interface. With some more training two more subjects would probably have achieved reliable control. For the last two subjects other parameters would have been needed. This would lower speed, but the increase in reliability would be worth it.

Finally, the free run results also indicate an influence of the contrast of the background on the signal strength.

8.5.4. The System in General

The study mainly served to evaluate the cVEP-BCI system for navigation and control in general. Here, I discuss the results in terms of what they indicate for the system as a whole. Four points are to be discussed. First, reliability and speed, i.e is the system as reliable and fast as needed? Second, a comparison with other BCI approaches for navigation and control. Third, what can be inferred about the user experience? And finally, how to deal with artifacts in cVEP systems.

i. Reliability and speed For a rather complex situation, the system reached a high reliability and speed for three out of seven subjects. Two others would need some more time to get accustomed to the system and possibly slightly different parameters for the dynamic repetitions, but their performance in the second block strongly indicates that high reliability and speed would have been possible. Even with the used parameters their performance was okay. For the last two subjects some more tuning would be necessary, but the general approach still worked, they were able to complete the task. So, except for one subject who did not produce a cVEP at all, all subjects achieved control over the system and could emit commands faster as in other BCIs for navigation and control.

ii. Comparison with other BCIs Compared to BCIs for control using P300 the cVEPs system offers higher information throughput and much lower latencies.

In a P300 system, presented in [Escolano et al., 2012] roughly half a minute was needed per command. Based on our own experience, P300-BCIs with 12 commands can offer latencies as low as $6 - 10$ s, but that is still 3-5 times slower than the cVEP system.

On the other hand, SSVEP-based approaches offer comparable speed, but much less flexibility, for example in terms of number of commands and are more restricted in terms of scenario and interface, as discussed in paragraph 3.4.iii p. 41.

The classification accuracy of my system is comparable to these well-known approaches at least for some subjects, but not yet for all. This needs further investigation. In real-world applications it might be sensible to test different approaches for a given subject and use which works best.

iii. User experience It is always difficult to asses the user experience. Indeed, the small questionnaire used for the experiment can only provide hints about the user experience, a deeper assessment of the user experience would need a more detailed questionnaire and possible other measures as well. Still, the questionnaire indicates that the final system is less tiring and more interesting than the off-line systems used in the first two experiments. Additionally, in the off-line experiments most subjects complained about the annoying flickering, for the current experiment nobody did.

iv. Artifacts The analysis of artifacts was done by a simple, but very common threshold mechanism. With this mechanism only very few epochs are identified as possibly contaminated. This means, either artifacts are not a concern in cVEP systems or the threshold mechanism is not suitable to detect artifacts in cVEP systems. First, one can argue that artifacts are least less of a concern in all VEP-based BCIs as in others. This is caused by the electrode montage. A major source of artifacts in the EEG measurements are eye movements, but these are much more prominent in the frontal electrodes, which are not used. Additionally, the reference electrode is relatively close to the measurement electrodes. This means that the sources of most external artifacts, like eye movements, body movements, and so on, have a similar distance to reference and measurement electrode, which means that their contribution almost vanishes when the reference signal is subtracted. Second, there might be other artifacts not so common in other BCIs which are of concern for VEPs. For example, Sutter mentions artifacts from the neck muscles in [Sutter, 1992]. Still, in the current experiment no neck muscle artifacts happened, they would have been found by the threshold mechanism.

8.6. Conclusion

In this chapter, a complete BCI system for navigation and control was presented and evaluated. Based on the cVEP, the system reaches comparable reliability, but much lower latencies than other BCIs for navigation and control and offers more commands. The system was evaluated using a scenario, where the subjects had to command a virtual agent to perform a daily-life task, consisting of five sub-task in an unconstrained environment.

9. Discussion

This chapter collects the results of the experiments which were presented in the previous chapters and discusses them in regard to the research question. As stated in the introduction in section 1.3 p. 3 the examined research question can be divided into four sub-questions. First, a general investigation of the code-modulating, Visually-Evoked Potential (cVEP), a new approach for EEG-based BCIs. Second, a closer investigation about possible parameters that influence the signal strength of the cVEP. Third, finding and evaluating a method to balance speed and accuracy. Finally, building on the answers to the three previous questions, the implementation and evaluation of a cVEP-BCI for navigation and control of devices.

This discussion follows the same structure. The next four sections discuss my results in terms of these four research questions. Additionally, a fifth section discusses further research and, finally, the last section summarizes the thesis and gives an outlook.

9.1. cVEP: A new Approach for EEG-based BCIs

Here, I discuss the advantages and disadvantages of cVEP-BCIs in general and show the contributions of this thesis to the investigation of cVEP-BCIs in general.

Current BCIs provide only limited means of communication. Three key factors to be improved are speed, accuracy and number of commands. All three are deeply coupled though. More commands usually mean less accuracy, or more accuracy less speed, and so on.

BCIs based on the code-modulating, Visually-Evoked Potential seem to outperform other paradigms in those three key factors. Still, there are a limited number of studies[1]. Already decades ago E.E. Sutter did the first studies using cVEP [Sutter, 1992]. Still, his research was not carried on until recently. Then, the groups of Bin et al. [Bin et al., 2009a, 2011] and Spüler et al. [Spüler et al., 2012a,b] both realized spelling systems using the cVEP. These outperform other paradigms in terms of speed and accuracy. In [Spüler et al., 2012a] the systems reaches an ITR of 133.6 bit/min in an off-line task which was the highest ITR reached by a BCI employing surface EEG at that time. In [Spüler et al., 2012b] subjects could write on average 21.35 error-free letters per minute. This would allow patients already quite fluent communication. Furthermore, there have, so far, been no reports of “BCI illiteracy” for cVEP-BCIs. BCI illiteracy means, that some subjects are unable to control the system at all, an effect known from ERD-based BCIs as well as from high frequency SSVEP-BCIs [Nijholt and Tan, 2008; Volosyak et al., 2011].

Despite the promising results, few studies have examined the cVEP so far. To support the existing results, I also did one spelling experiment (see chapter 7 p. 81ff.) similar to the work of Bin et al. and Spüler et al. In my spelling study, an average ITR of 139.11 bit/min was achieved during the off-line block and all subjects reached good accuracies, confirming the previous results. In the on-line blocks subjects reached 8-16 error-free letter per minute. This is slower than in the previous work, but still consistent. While the previous work was tuned to maximize on-line speed, my system was mainly intended for evaluation of the dynamic repetition mechanism, which is discussed later in this chapter. With little optimization towards on-line spelling speed the system could reach 20 letters per minute as well.

Another aspect, which has not been examined at all for cVEP-BCIs, is flexibility in terms of applications. The famous P300 paradigm or the SSVEP paradigm both have been used to build all kinds of applications. Several examples are described in chapter 2, including BCIs for the control of humanoid robots, different devices at home such as light or TV, wheelchairs and so on. For cVEP-BCIs only spelling applications have been investigated so far. Research on other BCI approaches suggests that

[1] A deeper investigation of the previous work on the cVEP is done in section 3.3 p. 35ff.

more demanding applications, like navigating a robot, decrease communication speed and reliability. In this thesis I give first insights into using cVEP-BCIs for applications beyond spelling. I show that the accuracy really drops, but also how this can be compensated. The details are discussed in the next two sections, but the essence is that cVEP can be used for applications beyond spelling and also outperforms P300 and SSVEP in terms of speed and reliability for such applications.

Finally, cVEP-BCIs are dependent BCIs by design. This means, they only work as long as there is residual muscular activity in the oculomotor muscles. Thus, they are not suitable for completely locked-in patients. This leads to a discussion which is also ongoing for SSVEP. In essence, a wide range of applications remains, but one must asses carefully from case to case, if other input modalities, such as eye-tracking devices might be better suited than a dependent BCI.

Possible applications might include patients with locked-in syndrome (LIS) who still have control over the oculomotor muscles. Additionally, other patient groups, for example patients with quadriplegia might benefit from a BCI. Finally, there is also some research about BCIs for healthy users for hands-free applications. This includes applications in space or for gaming, several examples were presented in chapter 2 of this thesis.

A detailed discussion comparing eye-tracking devices and dependent BCIs was also done in section 3.5 p. 41. In summary, further research on both input modalities is needed. Currently, eye-trackers are faster, but depend on careful calibration and controlled lighting conditions. Additionally, they can only partly deal with glasses and head movement, which was the original reason why E.E. Sutter investigated the cVEP. On the other hand, BCIs currently are slower, but especially for the cVEP there is a lot of room for improvement. Finally, for many applications it might be useful to combine both, either simultaneously to enhance accuracy or alternately depending on the user's current liking. One example, of such alternating between eye-gaze system and a P300-BCI is a painter suffering from ALS. She uses eye-gaze technology for communication and the BCI for painting [Holz et al., 2013; Kübler et al., 2013].

9.2. Different Influences on cVEP Signal Strength

My next research question was to asses the influence of different parameters on the signal strength of the cVEP. Previous work focused on finding a suitable spatial filter [Bin et al., 2009a, 2011] and a suitable classifier [Spüler et al., 2012a]. Still, there a several parameters, which influence the signal strength and thus also the classification accuracy. Especially, with regard to cVEP-BCIs for more complex applications than spelling it is absolutely necessary to asses the influence of some of these parameters.

In this thesis I investigated three groups of parameters: the codebook structure, several details of the stimulus presentation and the users' mental state.

9.2.1. Codebook Structure

The codebook plays a central role in the design of cVEP systems. The codebook defines for every stimulus a binary vector. This vector defines for every frame, whether this stimulus is active, i.e. rendered, or not. This means, the codebook has to be designed to maximize the mutual Hamming distances between the different codebook vectors. In previous work, the systems always used plain m-sequences as originally proposed by E.E. Sutter. M-sequences have the property of a low auto-correlation. Thus, circular shifted version of one m-sequence have high Hamming distances to each other.

Different other sequences might offer high Hamming distances as well, for example almost perfect sequences as proposed in [Bin et al., 2011].

A thorough evaluation of possible sequences is out of the scope of this thesis. Instead, I tried a modification of m-sequences to enhance functionality. The idea is to separate the recognition of the attended stimuli in two parts. First, the general area or group. Second, the item within the group. I call this a hierarchical codebook and introduce it thoroughly at the beginning of chapter 5 p. 63ff. One could also use more than two layers, but I worked with two, to asses the feasibility.

A hierarchical codebook has two advantages. First, the stimuli can be grouped according to their function. This allows to deliver the group information to the application already after the first half of the complete epoch. For example, for a navigation application the stimuli might be grouped according to their general direction and the device might start movement towards a roughly correct direction already after the first half of the epoch. Second, the epochs can be shorter. Two easier classification problems need less stimulation frames than the original complete problem.

So, for the first experiment, which is presented in chapter 5 p.¸63ff., a hierarchical codebook with the following structure was used. 15 bits encoding the group, 15 bits encoding the item within the group, again the first 15 bits and again the second 15 bits, so 60 bits in total. The repetition was done to increase accuracy. Still, the results of the first study were mixed. Recognizing the group membership after only 15 frames was quite successful, with an average accuracy of 81%. The detection of the item within the group was more difficult though, so the accuracy in detecting the complete target after all 60 frames was only 67%. Three out of ten subjects did not reach usable classification rates, but if one excludes those three, the average is still only 78%. Compared to previous work this is rather low. Deeper analysis revealed that there might be some kind of interference effect, i.e. the brain activity encoding the group gets mixed with the brain activity encoding the item within a group and the signal processing system is not able to separate this.

For the second experiment the shifting factor was set to two and the structure was changed: First 7 bits encoding the group, a pause of 5 bits 7 bits encoding the item, again the 7 bits encoding the item, again the pause and finally again the 7 bits encoding the group. It was hoped, that the new structure would mitigate the mixing problem, but again the item within the group was detected with much lower accuracy than the group.

As a consequence I used a flat codebook with a shift of two for the third and fourth experiment. For the third experiment, which was the spelling experiment an accuracy of 93% was reached in the off-line block. For comparison, Spüler et al. achieved 92% without subject-specific filters and 96% with subject-specific filters [Spüler et al., 2012a].

In summary, the hierarchical codebook has advantages, but decreases classification accuracy too much. One needs to find a way to solve the mixing problem before it can be used for practical application.

9.2.2. Stimulus Presentation

As it is a visually-evoked potential the signal strength of the cVEP depends strongly on the characteristics of the stimulus presentation, similar to SSVEPs. For SSVEPs this was investigated for example in [Wang et al., 2010; Zhu et al., 2010; Byczuk et al., 2012]. This research is summarized in section 3.2 p. 30ff.

For cVEP-BCIs no such investigation existed. Again, there are many parameters of the stimulus presentation, which might have an influence on the signal strength and I concentrate on four which are important for a cVEP-BCI for navigation and control. These are shape, color and size of the stimuli and the background of the interface.

i. Shape First, I consider the shape of the stimuli. Depending on the implementation of the stimulus presentation the shape can influence the number of pixels, but this effect is considered as part of the discussion concerning the stimulus size. So, basically when discussing the influence of shape, one question is one about generalization capabilities of the system. When the system is trained on “A” and “B” does it also correctly discriminate data where “C” and “D” are used as stimuli, when the same codebook vectors are used for both stimuli sets.

In [Spüler et al., 2012b] this was found to be the case and even more. The shifting property when using plain m-sequences allows to train only on few stimuli, but then recognize much more.

In my experiments the issue of shape was touched twice. In the third experiment (chapter 7 p. 81ff.), subjects used a spelling system with three pages of characters: minuscule letters, majuscule letters and special characters and symbols. During the training block only data from the first page was recorded. Later, the subjects had to input complete sentence where all three pages were necessary. The classification accuracy of first and second page was nearly identical, the accuracy of the third was somewhat lower, but not significantly, probably it was less used and subjects sometimes failed to find the correct character in time.

In the fourth experiment (chapter 8 p. 89ff.), the training block only used the navigation stimuli, but not the action stimuli. Additionally, the action stimuli had different shifting factors and were later used simultaneously together with the navigation stimuli. No quantitative evaluation is possible as the subjects were free to choose their path of action, but the qualitative analysis suggests that the action stimuli were classified with the same accuracy as the navigation stimuli.

So, empirical evidence supports the theory that the stimulus shape has little to no effect on the brain activity.

ii. Color Two questions arise about the effects of color stimulus . First, does the choice of stimulus color have an effect on the signal strength? Second, is it possible to mix different color schemes within one classifier? For the first question some previous work on SSVEPs exists, which indicates that color does have an influence. In [Mouli et al., 2013] three LEDs were used alternatively as stimuli for an SSVEP-BCI. One red, one green, one blue. Signal strength was significantly higher for the green LED in four out of four subjects for most of the investigated frequencies.

For cVEP, all previous studies used to render the stimulus white when it was active and not render it when it was not active. I call this black/white flickering as this was done on a black background. In my first experiment I compare this black/white pattern, with a green/red pattern. There, a green rectangle is rendered for each stimulus as a local background. When a stimulus is active in a given frame, it is rendered in red onto the green rectangle. Classification accuracy for these two conditions was identical on average, but differed for some subjects.

To investigate the second question I trained the classifier system on data from one condition and evaluated it on the data from the other color condition of the same subject. Theoretically, this is a difficult problem as the black/white information is processed in a different area of the visual cortex as the red/green information. This means the spatial characteristics of the signal probably depend on the color scheme. Indeed, in the mixed scenario the classification accuracy was very low (around 20%).

So, using different color schemes seems feasible for cVEP-BCIs. For mixing them one would need to cope somehow for the different spatial distribution of the signal, maybe by training multiple classifiers.

iii. Size Size is a very important parameter, both for the signal strength and for the application. The number of neurons which are activated by a visual stimulus directly depends on the number of activated sensory cells in the retina and this is directly dependent on the size of the stimulus. For SSVEPs this is for example discussed in [Regan, 1977]. One important aspect is the fact that the sensory cells in the retina are all but equally distributed, as discussed in section 3.1 p. 29. Most of the sensory cells are concentrated in a very small area, the fovea, which occupies roughly 2° of the visual field.

In [Bakardjian et al., 2010] the authors find that, for SSVEPs, the signal strength is not affected too much when the stimuli occupy $1.8^\circ \times 1.8^\circ$. In previous cVEP systems, size was ignored, the stimuli were scaled such that all stimuli and the currently written text fit on the screen, i.e. stimuli occupied much more than 2°.

In my experiments I also use only two different sizes. In experiment 1 and 3 stimulus size was maximized as in previous work. In experiment 2 and 4 the stimuli occupied roughly 2° of the visual field. The data is not directly comparable, because other parameters also changed. But in somewhat similar conditions of experiment 1 and 2 also similar accuracies were measured. 67% on average for the first experiment and 65% in the corresponding condition of the second experiment. In the fourth experiment the stimuli were also small and the classification accuracy was acceptable, despite several other parameters which probably also have a negative effect on the signal strength, such as a changing background.

So, the effect of stimulus size is not yet exactly quantified, but it seems small as long as the whole visual field of the fovea is covered. This finding is exactly what is needed for the construction of a cVEP-BCI for navigation and control. Still, a more systematic investigation would be interesting.

iv. Background For a BCI for navigation and control one would like to draw the stimuli onto a scene image. Of course, one could propose other layouts, for example attaching LED arrays at the border of the screen. Still, all other layouts mean the user has to look back and forth between stimuli and scene.

When one wants to draw the stimuli on a scene, one must investigate the effect of background on the signal strength. For SSVEP-BCIs some systems for navigation and control exist which draw the stimuli onto the scene image [Bakardjian et al., 2010; Gergondet et al., 2011]. Both are analyzed in detail in section 3.2.3 p. 33ff. In [Bakardjian et al., 2010] a small virtual car is steered along a track on the screen. The track is always completely visible and the stimuli are drawn around the car and follow it. The static scene around the track is mostly yellowish, the track is black and the stimuli are drawn in a black-white checkerboard pattern. No evaluation of the effect of the background was done, but a reliable control and good accuracy is shown. Additionally, the authors argue that putting the stimuli near the point of interest instead at the edges is beneficial for the signal strength as it helps to avoid large eye-movements. This is in accordance with the results of [Trejo et al., 2006]. In [Gergondet et al., 2011] a humanoid robot is steered using a 4-command SSVEP-BCI. The images from the robots camera are directly streamed to the user interface and the stimuli are drawn onto them. The scene image is rendered only as gray-image though. The stimuli are drawn in red. Again, no quantitative evaluation on the effect of the background was done, but a reliable control was achieved. Still, the problem of background and contrast is circumvented to some extent. The scene is rendered as a gray-scale image and the red stimuli ensure a high and to some extent stable contrast. For cVEP-BCIs no such evaluation is done in previous work. In my experiments the effect of background is considered twice:

First, in the second experiment a quantitative comparison between a condition with standard, pure black background (called "no-background" condition) and a condition with a static scene image from a camera (called "background" condition) is done. In a preliminary test, classification with background was not possible at all, when the background and the stimuli are rendered with 60 Hz. So, for the actual experiment background and stimuli are rendered with 120 Hz, but each codebook frame is rendered twice. In effect, it looks as if the background is rendered with 120 Hz and the stimuli with 60 Hz, which seems to be less distracting. Each subject did one off-line block with each condition. The average accuracy of the "no-background" condition blocks was 65%, the average accuracy of the "background" condition blocks was 50%. For five out of six subjects a significant drop in accuracy was observed.

Second, in the fourth experiment, the background is the colored and changing background of the virtual kitchen where the virtual agent moves around. A deep analysis of correlations between parameters of the background and the resulting accuracy was done in section 8.4 p. 97ff. In particular, there is a negative, linear correlation between the standard deviation of the gray-scale image at the start of an epoch and the classification accuracy. Here, the standard deviation of the gray-scale image was used as measure of variance in the luminance. In contrast, specific colors in the background or the overall level of luminance showed no correlation with the accuracy.

So in general, I found indication that a non-uniform background lowers the signal strength. This is in accordance with the neurophysiology, as the visual processing of the non-uniform background produces more neural activity than the processing of pure black background. This activity contributes to the EEG signal, but does not contain information about the stimulus, thus reduces signal-to-noise ratio. Additionally, this effect seems to be stronger when the background is more complex, i.e. high variance in luminance or contrast across the image. Still, even for very complex backgrounds the intention of the user can be inferred reliably when using some repetitions.

9.2.3. User's Mental State

Another parameter which might influence the signal strength is the user's mental state. In particular, when there is high variance in the accuracy across subjects as in my experiment 1 and 2, the mental state can be investigated as a possible source of this variance. Interesting factors might be tiredness and level of attention.

I analyzed this in detail for the first experiment. First, subjects were asked about their mental state. They had to rate statements like "I felt stressed before the experiment" and "I felt relaxed before the experiment" on a scale from 1 to 6. The rating of these question was correlated with the classification accuracy, but no linear correlation could be found. Additionally, I analyzed the relative bandwidth of the α-rhythm of the subjects right before the start of each epoch. The α-rhythm is a standard measure of attention and concentration [van de Laar et al., 2013], it was briefly presented in section 2.2.4 p. 11ff. A high relative band power of the *alpha*-rhythm indicates relaxed attention whereas a low band power indicates stress or agitation. I could show a negative, linear correlation between pre-epoch, relative α-level and classification accuracy (for more details see section 5.3.6 p. 70). This means that subjects with lower α level, i.e. higher mental involvement, or higher stress had a better signal strength, i.e. higher accuracy.

So, while I could not find correlations between self-assessed mental state and accuracy, there is some correlation between mental state assessed by the EEG and the accuracy.

9.3. Balancing Speed and Accuracy

The next research question was to balance speed and accuracy. In particular by repeating epochs. The results of experiment 1 and 2 indicate that classification based on single epochs is not reliable enough. The signal strength, especially with a lively background, is too low. A common concept to increase accuracy is to combine the classifier output of multiple epochs before issuing commands. The group of epochs which are combined to determine one command is then called a *decision*. The concept of repetitions is introduced in section 4.1.5 p. 54f. Especially, I discuss the issue of balancing speed and accuracy. Essentially, one wants to repeat as much as necessary to achieve a reliable classification and as little as possible to achieve the lowest possible latency. This issue is further complicated by the fact that the signal strength varies during a session, which means that the ideal number of repetitions also varies constantly.

This can be partly compensated by using a dynamic number of repetitions. The idea is to asses the combined signal strength of the epochs of the current decision directly after the classification of each epoch. Based on this combined signal strength the system decides whether another epoch is needed or the command can be issued. Still, this creates the new issues how to asses the signal strength on-line, without the true label of the decision, and how to decide when the signal strength is high enough.

In previous work on cVEPs no repetitions are used, because in cVEP spelling system the accuracy is usually high enough with only one epoch.

My first approach to balance speed and accuracy using dynamic repetitions was implemented and evaluated in experiment 3 chapter 7 p. 81ff. BCI systems using dynamic repetitions are currently mostly used in P300 systems. There, one usually uses the score of the LDA classifier, i.e. the distance of an epoch to the decision hyperplane, as metric for the signal strength. So, I use the score of the OCSVM which is also the distance to the decision hyperplane as basis to asses the signal strength. As I have more than two classes, the classification score of an epoch is calculated as difference of the distances to the hyperplane of the best and second-best class. When an epoch has been classified and this difference computed, this difference is called the *score* of the best class. When this *score* reaches a predefined threshold, the decision is considered to be complete. Two different thresholds were evaluated in the study. Using this mechanism an average on-line accuracy of 94.8% was achieved. Additionally, only 21% of the correctly classified first epochs were repeated, but 75% of the incorrectly classified first epochs. Thus, this scheme was also effective in terms of speed.

For the navigation system in experiment 4 the scheme was slightly adapted. There, I summed the *score* of each class per epoch and calculated the difference after that. This difference was then checked against a subject-specific threshold. The results of the evaluation of this mechanism have to be interpreted cautiously, because the threshold is computed on the same data block which is also used for the evaluation. This is the only block, where epochs were repeated and true labels are known. Still, on this dataset the accuracy for the first epoch was 61% on average. The average accuracy using the score-sum mechanism with a fixed number of five repetitions was 86%. Using dynamic repetitions the average number of repetitions was 2.6. The average accuracy was 81%. So, with roughly half the number of repetitions the average accuracy drops only by 6%.

In summary, my studies show that dynamic repetitions are useful to balance speed and accuracy in cVEP-BCIs and demonstrate a mechanism to realize them.

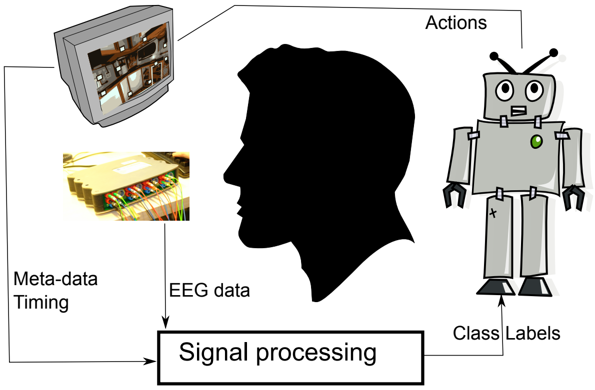


Figure 9.1.: System schematics of the final cVEP-BCI for navigation and control. On the monitor the user interface for the virtual kitchen study is indicated. Head, monitor and robot from openclipart.org

9.4. A cVEP-BCI for Navigation and Control

The final research question deals with the implementation and evaluation of a complete cVEP system for navigation and control, taking into account the results of the previous questions. Since many years BCIs for navigation and control have been the second major application besides BCIs for spelling. Here, I briefly summarize the properties of the system, compare it with other BCIs and eye-tracking systems for navigation and control and summarize my results about the user experience.

9.4.1. The System

When designing BCIs for navigation and control, one faces a special issue, namely balancing signal strength and flexibility of the application. As such the user interface and the signal processing have to be integrated closely. My system has three key features to cope with this issue. First, the system uses the cVEP for a high signal strength. Second, a flexible scene presentation with the stimuli dynamically overlaying the scene ensures the flexibility of the application. Finally, the dynamic repetitions mechanism connects both. The different parts are also shown in Figure 9.1.

The general cVEP approach has already been discussed in section 9.1. For the final system I again use a flat codebook. This time a codebook was chosen that supports up to 16 targets with an epoch length of 0.52 s.

The design of the user interface is particularly challenging. The signal strength of VEPs is greatly influenced by the stimulus characteristics and the stimuli which maximize signal strength are often not well suited for a good user experience. So the design needs to balance signal strength and user experience. In my system, the size of the stimuli was chosen according to the findings of experiment 2. So, they occupy slightly more than 2° of the visual field. The navigation stimuli were positioned on fixed positions on the border of the screen according to the direction they would steer the agent. The fixed positions allow the user to jump with the gaze rapidly towards the fitting stimulus. The action stimuli, which allow to activate certain actions on corresponding objects, are rendered on the object they correspond to. This is supposed to ease interaction as well. The user needs to find the object anyhow, then it is easy to focus the gaze on the stimulus. To ensure a high local contrast a small black rectangle is rendered around each stimulus.

The dynamic repetitions mechanism was adapted slightly. Compared to the spelling system in the third experiment the signal strength in the navigation system is lower and the dynamic repetitions system has to compensate for it. Additionally, the system needs to support asynchronous control. Thus, I introduced a minimal and maximal number of repetitions. The minimal number of repetition increases accuracy. The maximal number allows some kind of asynchronous control, i. e. no command is issued when the user does not gaze at any stimulus.

In the evaluation I could show that the system is indeed fast, flexible and allows fluent interaction.

When following an a-priori defined path an average classification rate of 81% with a latency of around 1.5 s was reached. In the free-run, all but one subjects were able to complete the task. Three of eight subjects had a very fast and reliable control. They needed a comparable number of commands with the BCI as in the baseline with keyboard control and achieved a latency of about 2 s per command. Four other subjects were also able to complete the task, but needed a bit more time. The last subject did not produce a cVEP, probably the electrodes were misplaced.

9.4.2. Comparison with Other Input Modalities

As mentioned before, several BCI approaches for the control of devices exist. Additionally, eye-tracking devices would be an alternative, as a cVEP-BCI still needs oculomotor control. Here, I compare my system with previous P300-BCIs, SSVEP-BCIs and eye-tracking systems. A general comparison of cVEP-BCIs versus theses alternative was done in section 3.4 p. 40ff.

i. P300 P300 is widely used in BCIs for navigation and control. It is flexible, because its signal strength does not depend much on the stimulus characteristics, and well-investigated. Some examples are presented and discussed in section 2.4.4 p. 23ff.

The main advantages of P300-BCIs are: They allow to use many different commands, there are no P300-BCI "illiterates", it can be detected reliably when using a sensible number of repetitions and one can design the user interface quite freely.

The disadvantage is speed. The brain signal is generated quite late in the processing of stimulus thus it is quite slow, i.e. it takes 300 ms from stimulus onset for the P300 to appear. Additionally, all stimuli have to be presented one after the other. Finally, some repetitions are necessary to detect the signal reliably. As a result it took about 30 s on average per command in a telepresence system [Escolano et al., 2012] with 20 different commands. In work from our working group each decision took about 5.5 s in a 5-command P300-BCI with dynamic repetitions [Riechmann et al., 2011].

In contrast, my current system allowed subjects to issue commands with a latency of $1.5 - 2$ s in a 12-command system. Additionally, further optimization is possible, as this is the first cVEP attempt, whereas P300 systems have been optimized for decades.

ii. SSVEP SSVEP- and cVEP-based BCIs share some properties, because both are VEPs. This means both need oculomotor control of the subject and the signal strength depends on stimulus characteristics. The work on SSVEP-BCIs is summarized in section 3.2.3 p. 33ff.

For navigation of a robotic device there are two interesting studies. In [Gergondet et al., 2011] a real robot is steered using a 4-command SSVEP. Each decision took about 2 s. Stimuli are drawn onto a scene image from the robot. The image was rendered in gray-scale and stimuli in red. All subjects were able to steer the robot reliably and quite fluently. In [Bakardjian et al., 2010] an 8-command BCIs was used to steer a virtual car on a static track. Each decision took in average 3.4 s.

So, SSVEP-based BCIs are faster than P300-BCIs, but usually do not offer as many commands as cVEP-BCIs and current SSVEP studies do not provide the same flexibility as my cVEP-based system.

iii. Eye-tracking Some comparison of cVEP-BCIs and eye-tracking system was done in section 9.1.

Still, a detailed comparison of the final system with eye-tracking systems is difficult. In general, it is assumed that systems based on eye-tracking are faster than BCIs. Eye-tracking systems have been built for navigation in virtual environments [Stellmach and Dachselt, 2012] or to control a first-person shooter game [Sko et al., 2013], but these studies give little quantitative data about reliability and speed of command recognition, mostly they evaluated user experience.

I would argue that to some extend the question is still open, i.e. one does not know which system ends up being better. Additionally, it highly depends on the user and the application what "being better" means in the first place. So, as of today research on both input modalities is necessary.

iv. Other cVEP-BCIs for control There is one recent study using a cVEP system for the navigation of a small robot. I discussed it in section 3.3.4 p. 39. Basically a 4-command BCI was used to steer a wheeled robot along a line. The authors compared SSVEP and cVEP and found that the cVEP system outperformed the SSVEP system both in speed and reliability. In the cVEP condition each decision took about 3 s. Still, compared to my final system the user interface is static and the system is slower and offers less commands.

9.4.3. User Experience

Evaluation of the user experience was not in the focus of this thesis. Still, there are two interesting aspects in the results derived from the questionnaires. First, a common criticism made against VEP-BCIs states that the flickering of the stimuli is annoying and tiring for the users. This was indeed indicated by the questionnaire results from my first experiment. For the final system, no user complained about the flickering and the system was not rated tiring by any user. The embedding of the stimuli in the scene and the interesting task seem to increase acceptance. Second, the subjects liked the system. To some part this is just the results of the system being new, but also subjects are fascinated by the idea of controlling a game or a robot just with their thoughts.

9.5. Future Work

In this work, I present my dissertation project on cVEP-BCIs. The final system already allows for quite fluent control in a flexible scenario. Still, there is always room for improvement. Three areas of possible improvements are discussed here. First, several approaches to further improve accuracy and speed, second an outlook on more complex applications, third some thoughts on a mobile cVEP system.

9.5.1. Enhance Accuracy

When looking at related research and the results of my experiments, several approaches can be identified which might increase the accuracy of the cVEP-BCI. Increasing the accuracy would also increase speed as one can work with less repetitions and shorter epochs. Possible modifications can be grouped according to the part of the BCI which is optimized. Here, I discuss modifications to the user interface, the data acquisition, the signal processing and the post-processing. For each modification I discuss the possible impact, the chance of success and the effort which is needed to implement it.

9.5.1.1. Modifications to the User Interface

The properties of the user interface or stimulus presentation have a strong influence on the signal strength, as I have shown in this work. When regarding previous work on other BCI systems, three major modifications are of immediate interest.

i. Checkerboard cVEP For SSVEPs there exists a common alternative to the flickering of complete stimuli, the pattern-reversal checkerboard pattern. The idea is to have a black-white stimulus which is divided into small patches. These patches are alternately black or white colored. This looks like a checkerboard as shown in section 3.2 p. 30ff. When flickering the stimulus, the black and white patches are switched for each flicker. Several researchers claim that the signal strength is higher using this checkerboard pattern compared to single-switch stimuli [Lalor et al., 2005; Zhu et al., 2010].

This could be easily tested for a cVEP interface as well, but there might be an even better alternative.

ii. Facial cVEP The human visual system is highly tuned towards processing of (human) faces. For BCIs using a visual P300 some research has been done to exploit this property. In [Onishi et al., 2011] the authors show a P300-BCI which uses images of faces as the P300 stimuli. With only 2 repetitions they had an accuracy of more than 90%. In [Bakardjian et al., 2011] the effect of emotional faces was also investigated for an SSVEP-BCI. They show an increase in information transfer rate from 50 to 64 bits/minute for emotional faces (both joy and anger) compared to a blurred face or a checkerboard pattern.

A similar experiment could be easily run for a cVEP-BCI and one can expect similar results as with the facial SSVEP.

iii. Adapt to cortex structure Another attempt to better use the properties of the human visual system for the stimulus presentation was presented in [Vanegas et al., 2013] and discussed in section 3.2.3 p. 33f. The central idea is to flicker different parts of the stimulus at different times. In the primary visual cortex (PVC) the different areas which process the information from different areas of the visual field are positioned vertically to each other. Thus, when stimulated at the same time, their generated electrical fields cancel out partially. When this is circumvented by flickering different stimulus parts at slightly different times, the signal strength can be enhanced significantly. For SSVEP, Vanegas et al. found a 200% increase in signal strength for a not-subject-specific adaption and up to 383% increase for a subject-specific adaption.

The not-subject-specific pattern could be realized with manageable effort for my system as well. Main limitation is the frame rate of the screen. To allow for four different flicker on-sets for different parts of the stimulus one needs a frame rate which is four times the normal stimulus flickering rate. Still, the potential benefit is very high, doubling the signal strength could boost accuracy considerably.

9.5.1.2. Data Acquisition – Better Electrodes

Another method to improve signal strength would be to use better electrodes, especially active electrodes, as discussed in section 2.2.4 p. 11ff. In an active electrode system a small amplifier is directly attached to each electrode, thus much less external noise is picked up. This not only concerns noise from other electrical devices, but also movements artifacts and so on.

One should note though that the positive effect of active electrodes might be less strong for cVEP-BCIs as for other BCIs, because the cVEP is less influenced by artifacts [2].

9.5.1.3. Improve Signal Processing

Next, one can think about different approaches to improve the signal processing. Here, I discuss three approaches for improving feature extraction and classification.

i. Enhancements for the CCA In 2013, two approaches to enhance CCA as feature extraction for SSVEPs where presented [Zhang et al., 2013a,b]. The central idea in [Zhang et al., 2013a] is to do the CCA not only along the spatial dimension, but also along the temporal dimension. By doing a CCA in both directions alternatively, a filtered, subject-specific response is built iteratively. This response is then used as reference signal for a standard CCA analysis. In [Zhang et al., 2013b] the authors try to incorporate also the information of different trials when building the reference signal.

Both measure improved the accuracy of their SSVEP system slightly, but significantly. To use these approaches for cVEP, one would need to adapt them. Especially, because for SSVEP one can easily generate a reference signal. For cVEP, one must either find an appropriate substitute, like the average over trials which is used in standard CCA for cVEP or find a suitable artificial reference signal.

ii. Channel specific off-set The brain response does not start directly synchronously with the start of the flickering. At the very least the information has to be sent from the retina to the visual cortex[3]. Additionally, it might take some time for the visual cortex to start processing the information. There are estimates in the literature at which time the brain activity starts to reflect the cVEP activity, but they differ from 40 ms [Spüler et al., 2012b] to 80 ms [Sutter, 1992]. It would be beneficial to not start classifying the EEG data before on-set of the activity, especially because the circular shifting impedes the classification system in disregarding the first, unrelated part.

I would also assume that the perfect off-set differs between channels. To asses this, one would only need some data where there is not visual input after each epoch. Some preliminary tests indicate a potential increase in accuracy by 5 to 10 percentage points.

[2] The issue of artifacts in cVEP-BCIs was discussed shortly in section 8.4 p. 97ff.
[3] Of course the light also needs time to reach the retina, but that very small amount of time can be disregarded.

iii. Classification using LDA For the data of the first experiment I show that classification using Multi-Class Linear Discriminant Analysis performs similar to classification by SVM. This should be investigated deeper, especially it might decrease the time needed for classification thus decreasing system latency. Currently, classification takes about $100 - 200$ ms for epochs of 500 ms. So reducing this to nearly zero would mean a significant speed-up.

Unfortunately, there is little work on LDA for one-class classification as needed by the flat codebook system. Maybe one could add copies with a wrong shift as negative sample and obtain a two-class problem.

9.5.1.4. Enhance Post-Processing

There are several ways to improve accuracy after classification, for example the dynamic repetitions mechanism I use. Several other measures are discussed in [Plass-Oude Bos et al., 2012]. Three post-processing measures to improve my cVEP-BCI are discussed here.

i. Use probabilities for dynamic repetitions At the moment the score, i.e. the distance to the hyperplane, is used to asses the signal strength for the dynamic repetition mechanism. The motivation is that this score relates to the probability of a correct classification, i.e. data vectors with higher distance are considered to be classified correctly with higher probability. This assumption is based on the maximal-margin approach of SVMs. Still, this relation is probably non-linear. There are approaches to transform the score into a true probability, at least approximately, for example using binning techniques [Zadrozny and Elkan, 2002; Rüping, 2004].

The dynamic repetition mechanism might work more effectively, but it is already quite effective. So, the potential benefit is limited.

ii. Include error potentials For a higher increase in accuracy one could integrate error potentials. When feedback indicates that the BCIs detected the wrong command, the subject elicits a error potential (ErrP). This can be detected from the EEG data and the decision repeated.

The integration does need some work, as one needs a complete, new classification pipeline to recognize the ErrP. Additionally, one has to be careful not to lower system performance when the accuracy of detecting the ErrP is lower than the accuracy of the cVEP detection, as discussed in [Lenhardt, 2011]. Still, especially in a robotic application the benefit of avoiding some false classifications can be huge.

The integration of cVEP and ErrP has also been shown in [Spüler et al., 2012b].

iii. Threshold adaption At the moment, the threshold for the dynamic repetitions is fixed. By analyzing the score and the occurrence of error potentials, one could adaptively tune the threshold on-line. For example, when the label of the first epochs agrees with the final label for several decisions one might lower the threshold. On the other hand, when many error potentials are detected one might increase the threshold.

9.5.2. More Complex Applications

Another line or direction of future work is to integrate the cVEP with more complex scenarios. The cVEP system which I presented in this work is quite flexible and is supposed to be suitable for a broad range of applications. Two possible applications are discussed here. Integration of a real robot, and possible gaming scenarios.

9.5.2.1. Real Robot

My work showed the feasibility of controlling a virtual agent with a cVEP-BCI. A natural extension would be the integration of a real robot. The main issues are to find a suitable scenario and the integration of two systems which are both error-prone in themselves.

A suitable scenario extending the kitchen scenario might be cooking or baking. The user would select the next ingredient and amount. Then, the robot would put them into the cooking pot or baking bowl.

This would give rise to several dynamic BCI-driven interactions. Again, the user would see the scene from a robot camera and one would render stimuli onto the scene.

The scenario could be realized like this:

1. User selects a recipe or chooses to do without.
2. User selects an ingredient from a list of words or from bowls in front of the robot.
3. User selects intended amount.
4. Robot grabs some of the ingredient, trying to match the desired amount.
5. User checks amount visually and commands the robot to adjust when necessary.
6. Robot puts the ingredient into a main bowl.
7. If not finished go back to step 2.

This could easily be extended by having some electronically controllable kitchen device, for example mixers and a stove to allow real cooking or baking of simple recipes.

Mainly, this kind of scenario would allow to show the strength of the cVEP in a real application. When selecting a recipe or an ingredient the user has to choose from quite large number of possibilities. When the user selects the amount the number of options is smaller, but this should be fast to allow smooth interaction. When checking the amount, the latency is even more important. Ideally, the robot would start moving towards the main bowl after it grabbed some amount and the user would interfere only when the robot did not grab the correct amount. To realize this one needs a fast and reliable BCI.

9.5.2.2. Gaming

Several studies investigated the use of BCI for games [Nijholt et al., 2009; Lotte, 2011; van de Laar et al., 2013] with different BCI approaches. Using BCIs for gaming is an interesting option. Some disadvantages of BCIs are perceived as an interesting challenge by gamers, for example that control via a BCI is unreliable and unfamiliar. Using a cVEP for gaming would be interesting, because it allows to built a fast interface with many commands compared to other BCIs. In most studies so far, the game is simple or has to be simplified because of the low number of commands, this might not be necessary when using cVEP.

The virtual kitchen scenario in my last experiment could be easily converted to a game. For example, one might tell subjects that the aim is to produce virtual coffee as fast as possible and display the fastest time so far. One could even do a two player game, where two virtual agents produce coffee in one kitchen.

9.5.3. Mobile cVEP

Finally, it would also be interesting to transfer the cVEP to a mobile setting. Mobile BCIs have recently gained some interest. Some of the examples are discussed in section 2.4 p. 18ff. Especially, when investigating BCIs for healthy users, it would be beneficial to be mobile, for example for space applications or mobile games.

In principle, a mobile cVEP setting could be realized with an EEG headset, for example the Emotive EPOC, and a smart-phone. One would need to investigate how to ensure the necessary frame rate for the stimulus presentation and how to synchronize stimulus presentation and EEG.

9.6. Summary and Outlook

I started in the introduction by identifying some issues of current BCIs, which impede their use in everyday-life application. Central issues are low speed of communication and low reliability or accuracy. These two are also deeply intertwined.

I proposed that code-modulating, Visually-Evoked Potentials (cVEP) are a promising approach for faster and still reliable BCIs. There are some studies showing cVEP-BCIs to be faster than all other EEG-based BCIs. Still, there are few studies on cVEP for spelling systems and none for applications beyond spelling.

Thus, in my thesis I investigated the use of cVEP for EEG-based BCIs with a focus on BCIs for navigation and control. Four central research questions have been identified in the introduction and investigated throughout the thesis.

1. Investigating the cVEP in general.
2. Evaluate different designs of the cVEP-BCI in general and specifically the user interface. Investigate the strength of the cortex response and the resulting accuracy for these different designs.
3. Implement and evaluate a mechanism to balance speed and accuracy of the cVEP-BCI.
4. Implement and evaluate a complete cVEP-BCI for navigation and control.

To investigate these question I analyzed and discussed the current state-of-the art for BCIs in general and the cVEP specifically. Then I performed four experiments to investigate these research questions. Finally, the results of these experiment were discussed in this chapter with regard to the research questions.

In general, I found further evidence that cVEP outperforms other EEG-based BCI approaches in speed and reliability. It is quite robust to changes in certain stimulus characteristics like size and color, but influenced by the properties of the background. The best signal strength is achieved by a pure black background. For a navigation and control scenario I display the scene and the device on the screen and draw the stimuli onto them. A significant drop in signal strength can be observed, but the users' intention can still classified.

These findings allowed to build a complete BCI system for navigation and control, which offers up to 12 commands and a latency of about 2 s in a very flexible setting. Other BCIs either are much slower (those based on P300) or are quite restricted in terms of the setting or the user interface (those based on SSVEP).

So, this thesis showed the great potential of cVEP-based BCIs. cVEP-BCIs for spelling outperform other EEG-based BCIs in terms of information transfer rate and error-free letters per minute and my cVEP-based BCI for navigation and control also outperforms other approaches in terms of flexibility and speed.

The current system can serve as a basis for daily-life BCIs for patients and for healthy users. For patients one mainly needs a study with some patient groups to check in how far the advantages of cVEP-BCIs are preserved when transferring the system to patients. For healthy users, the next steps would be to further optimize speed and evaluate the system in more complex scenarios.

It is my belief that cVEP-BCIs can significantly shrink the gap in communication speed between BCIs and other input modalities such as eye-tracking systems. Thus, I hope that in the future the *code-modulating, Visually-Evoked Potential* opens up a range of new applications for brain-computer interfaces. As such, the results of this thesis bring current BCIs closer to the vision of BCIs: To allow real "thought control" of computers and robotic devices.

A. Supplementary Material

Enclosed Data Medium

The enclosed DVD contains the following items:

1. UBiCI version used for the experiment as source and binary for Ubuntu Precise.
2. All data sets from the experiments.
3. Several scripts to derive the results from the data sets.
4. The videos of the last two experiments.
5. A README describing structure and file formats of the content.

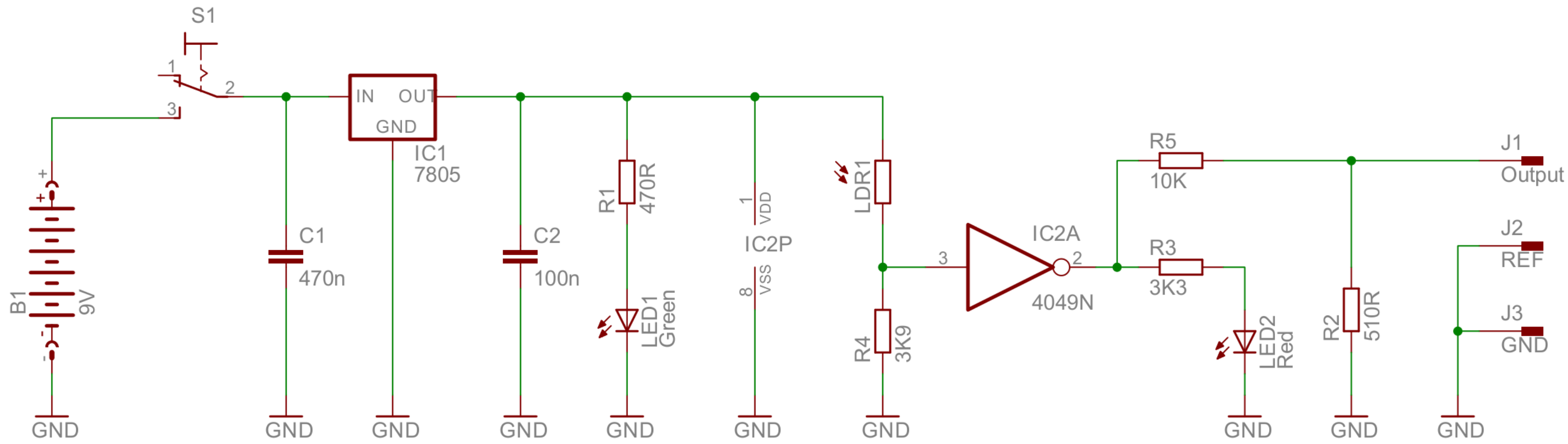


Figure A.1.: Electrical circuit which connects the Light-Dependent-Resistor with the EEG device.

Class 00:	[0, 0, 1, 1, 0, 1, 0, 0, 1, 0, 0, 0, 0, 1, 0, 1, 0, 1, 1, 1, 0, 1, 1, 0, 0, 0, 1, 1, 1, 1, 1]	not used
Class 01:	[1, 1, 0, 0, 1, 1, 0, 1, 0, 0, 1, 0, 0, 0, 0, 1, 0, 1, 0, 1, 1, 1, 0, 1, 1, 0, 0, 0, 1, 1, 1]	look upward
Class 02:	[1, 1, 1, 1, 0, 0, 1, 1, 0, 1, 0, 0, 1, 0, 0, 0, 0, 1, 0, 1, 0, 1, 1, 1, 0, 1, 1, 0, 0, 0, 1]	move forward
Class 03:	[0, 1, 1, 1, 1, 1, 0, 0, 1, 1, 0, 1, 0, 0, 1, 0, 0, 0, 0, 1, 0, 1, 0, 1, 1, 1, 0, 1, 1, 0, 0]	not used
Class 04:	[0, 0, 0, 1, 1, 1, 1, 1, 0, 0, 1, 1, 0, 1, 0, 0, 1, 0, 0, 0, 0, 1, 0, 1, 0, 1, 1, 1, 0, 1, 1]	rotate left
Class 05:	[1, 1, 0, 0, 0, 1, 1, 1, 1, 1, 0, 0, 1, 1, 0, 1, 0, 0, 1, 0, 0, 0, 0, 1, 0, 1, 0, 1, 1, 1, 0]	get cake
Class 06:	[1, 0, 1, 1, 0, 0, 0, 1, 1, 1, 1, 1, 0, 0, 1, 1, 0, 1, 0, 0, 1, 0, 0, 0, 0, 1, 0, 1, 0, 1, 1]	get coffee
Class 07:	[1, 1, 1, 0, 1, 1, 0, 0, 0, 1, 1, 1, 1, 1, 0, 0, 1, 1, 0, 1, 0, 0, 1, 0, 0, 0, 0, 1, 0, 1, 0]	rotate right
Class 08:	[1, 0, 1, 1, 1, 0, 1, 1, 0, 0, 0, 1, 1, 1, 1, 1, 0, 0, 1, 1, 0, 1, 0, 0, 1, 0, 0, 0, 0, 1, 0]	move left
Class 09:	[1, 0, 1, 0, 1, 1, 1, 0, 1, 1, 0, 0, 0, 1, 1, 1, 1, 1, 0, 0, 1, 1, 0, 1, 0, 0, 1, 0, 0, 0, 0]	coffee-machine
Class 10:	[0, 0, 1, 0, 1, 0, 1, 1, 1, 0, 1, 1, 0, 0, 0, 1, 1, 1, 1, 1, 0, 0, 1, 1, 0, 1, 0, 0, 1, 0, 0]	get water
Class 11:	[0, 0, 0, 0, 1, 0, 1, 0, 1, 1, 1, 0, 1, 1, 0, 0, 0, 1, 1, 1, 1, 1, 0, 0, 1, 1, 0, 1, 0, 0, 1]	move right
Class 12:	[0, 1, 0, 0, 0, 0, 1, 0, 1, 0, 1, 1, 1, 0, 1, 1, 0, 0, 0, 1, 1, 1, 1, 1, 0, 0, 1, 1, 0, 1, 0]	not used
Class 13:	[1, 0, 0, 1, 0, 0, 0, 0, 1, 0, 1, 0, 1, 1, 1, 0, 1, 1, 0, 0, 0, 1, 1, 1, 1, 1, 0, 0, 1, 1, 0]	look downwards
Class 14:	[1, 0, 1, 0, 0, 1, 0, 0, 0, 0, 1, 0, 1, 0, 1, 1, 1, 0, 1, 1, 0, 0, 0, 1, 1, 1, 1, 1, 0, 0, 1]	move backward
Class 15:	[0, 1, 1, 0, 1, 0, 0, 1, 0, 0, 0, 0, 1, 0, 1, 0, 1, 1, 1, 0, 1, 1, 0, 0, 0, 1, 1, 1, 1, 1, 0]	not used

Figure A.2.: Complete codebook for experiment 4. For each class, the number of the class, the codebook for the corresponding stimulus and the corresponding command are given.

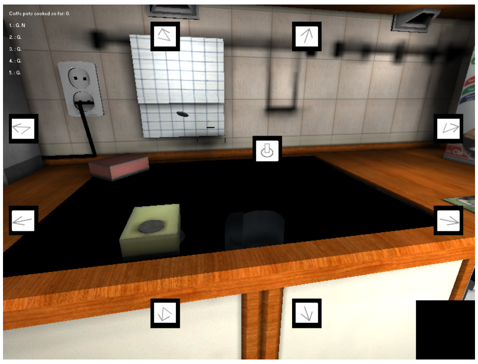


Figure A.3.: Screenshot of the area around the sink in experiment 4

B. Questionnaires

Fragebogen CVEP Experiment

Frage:	1 (wenig)	2	3	4	5	6 (stark)
Ich fühlte mich vor dem Experiment gestresst						
Ich fühlte mich vor dem Experiment entspannt						
Ich fand das Experiment anstrengend						
Ich fand das Experiment spannend						
Ich fand das Experiment ermüdend						
Ich fand die grün-rot Bedingung anstrengend						
Ich fand die schwarz/weiß Bedingung anstrengend						

Figure B.1.: Original questionnaire of the first experiment

Fragebogen CVEP Experiment

Frage:	1 (wenig)	2	3	4	5	6 (stark)
Ich fühlte mich vor dem Experiment gestresst						
Ich fühlte mich vor dem Experiment entspannt						
Ich fand das Experiment anstrengend						
Ich fand das Experiment spannend						
Ich fand das Experiment ermüdend						
Ich hatte das Gefühl das System verläßlich kontrollieren zu können						
Wenn die Vorbereitung nicht so aufwendig wäre, könnte ich mir vorstellen, PC/Videospiele mit dem System zu steuern						

Figure B.2.: Original questionnaire of the fourth experiment

C. Supplementary Results

Table C.1.: Time needed per run for every subject and run in experiment 4 in minutes.

Run/VP	1	2	3	4	5	6	7
1	4.11	2.25	2.17	2.06	3.83	7.24	4.40
2	5.25	1.13	2.37	2.08	2.32	4.38	6.32
3	6.75	1.17	2.21	2.55	1.95	3.48	4.00
4	2.32	1.80	1.58	1.19	2.26	2.81	5.49
5	4.31	0.96	1.17	0.93	4.50	3.23	4.35

Table C.2.: Time needed per run for every subject and run in experiment 4 relative to first run.

Run/VP	1	2	3	4	5	6	7
1	1.00	1.00	1.00	1.00	1.00	1.00	1.00
2	1.28	0.50	1.09	1.01	0.61	0.60	1.44
3	1.64	0.52	1.02	1.24	0.51	0.48	0.91
4	0.56	0.80	0.73	0.58	0.59	0.39	1.25
5	1.05	0.43	0.54	0.45	1.18	0.45	0.99

Table C.3.: Accuracy in percentage per position in experiment 4 block 1

Position	Accuracy
0	66
1	77
2	77
3	80
4	71
5	49
6	74
7	74
8	74
9	77
10	69
11	74
12	54
13	77
14	60
15	54
16	69
17	63
18	63
19	71
20	60
21	54
22	69
23	68
24	62
25	54
26	77
27	66
28	74
29	63
30	77
31	66

Bibliography

Aggarwal, V., Acharya, S., Tenore, F., Shin, H.-C., Etienne-Cummings, R., Schieber, M. H., and Thakor, N. V. (2008). Asynchronous decoding of dexterous finger movements using M1 neurons. *Neural Systems and Rehabilitation Engineering, IEEE Transactions on*, 16(1):3–14.

Ahmadi, M., Liu, J.-X., Brännström, T., Andersen, P. M., Stål, P., and Pedrosa-Domellöf, F. (2010). Human extraocular muscles in ALS. *Investigative ophthalmology & visual science*, 51(7):3494–3501.

Albanese, D., Visintainer, R., Merler, S., Riccadonna, S., Jurman, G., and Furlanello, C. (2012). mlpy: Machine learning python. *arXiv preprint arXiv:1202.6548.*

Allison, B. Z., McFarland, D. J., Schalk, G., Zheng, S. D., Jackson, M. M., and Wolpaw, J. R. (2008). Towards an independent brain–computer interface using steady state visual evoked potentials. *Clinical Neurophysiology*, 119(2):399–408.

Aloise, F., Schettini, F., Aricò, P., Leotta, F., Salinari, S., Mattia, D., Babiloni, F., and Cincotti, F. (2011). P300-based brain–computer interface for environmental control: an asynchronous approach. *Journal of neural engineering*, 8(2):025025.

Amiri, S., Rabbi, A., Azinfar, L., and Fazel-Rezai, R. (2013). A Review of P300, SSVEP, and Hybrid P300/SSVEP Brain-Computer Interface Systems. In Fazel-Rezai, R., editor, *Brain-Computer Interface Systems - Recent Progress and Future Prospects*, chapter 10. InTech.

Baillet, S., Mosher, J. C., and Leahy, R. M. (2001). Electromagnetic brain mapping. *Signal Processing Magazine, IEEE*, 18(6):14–30.

Bakardjian, H., Tanaka, T., and Cichocki, A. (2010). Optimization of SSVEP brain responses with application to eight-command Brain–Computer Interface. *Neuroscience letters*, 469(1):34–38.

Bakardjian, H., Tanaka, T., and Cichocki, A. (2011). Emotional faces boost up steady-state visual responses for brain–computer interface. *Neuroreport*, 22(3):121–125.

Bayliss, J. D. (2003). Use of the evoked potential P3 component for control in a virtual apartment. *Neural Systems and Rehabilitation Engineering, IEEE Transactions on*, 11(2):113–116.

Bell, C. J., Shenoy, P., Chalodhorn, R., and Rao, R. P. (2008). Control of a humanoid robot by a noninvasive brain–computer interface in humans. *Journal of neural engineering*, 5(2):214.

Berger, H. (1929). Über das elektrenkephalogramm des menschen. *Arch Psychiatr Nervenkr*, 87:527–570.

Billinger, M., Daly, I., Kaiser, V., Jin, J., Allison, B. Z., Müller-Putz, G. R., and Brunner, C. (2013). Is it significant? guidelines for reporting bci performance. In *Towards Practical Brain-Computer Interfaces*, pages 333–354. Springer.

Bin, G., Gao, X., Wang, Y., Hong, B., and Gao, S. (2009a). VEP-based brain-computer interfaces: time, frequency, and code modulations [Research Frontier]. *Computational Intelligence Magazine, IEEE*, 4(4):22–26.

Bin, G., Gao, X., Wang, Y., Li, Y., Hong, B., and Gao, S. (2011). A high-speed BCI based on code modulation VEP. *Journal of neural engineering*, 8(2):025015.

Bin, G., Gao, X., Yan, Z., Hong, B., and Gao, S. (2009b). An online multi-channel SSVEP-based brain–computer interface using a canonical correlation analysis method. *Journal of neural engineering*, 6(4):046002.

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

Birbaumer, N. and Cohen, L. G. (2007). Brain–computer interfaces: communication and restoration of movement in paralysis. *The Journal of Physiology*, 579(3):621–636.

Birbaumer, N., Ghanayim, N., Hinterberger, T., Iversen, I., Kotchoubey, B., Kübler, A., Perelmouter, J., Taub, E., and Flor, H. (1999). A spelling device for the paralysed. *Nature*, (6725):297–298.

Bishop, C. M. et al. (2006). *Pattern recognition and machine learning*, volume 1. Springer New York.

Björck, Åke. and Golub, G. H. (1973). Numerical methods for computing angles between linear subspaces. *Mathematics of computation*, 27(123):579–594.

Boucsein, W. (2012). *Electrodermal activity.* Springer, 2. edition.

Brodal, P. (2004). *The central nervous system : structure and function.* Oxford Univ. Press, New York [u.a.], 3. ed. edition.

Brouwer, A.-M. and Van Erp, J. B. (2010). A tactile p300 brain-computer interface. *Frontiers in neuroscience*, 4:19.

Brunner, C., Andreoni, G., Bianchi, L., Blankertz, B., Breitwieser, C., Kanoh, S., Kothe, C. A., Lécuyer, A., Makeig, S., Mellinger, J., et al. (2013). Bci software platforms. In *Towards Practical Brain-Computer Interfaces*, pages 303–331. Springer.

Byczuk, M., Poryzała, P., and Materka, A. (2012). SSVEP-Based Brain-Computer Interface: On the effect of stimulus parameters on VEPs spectral characteristics. In *Human–Computer Systems Interaction: Backgrounds and Applications 2*, pages 3–14. Springer.

Cacioppo, J. T., Tassinary, L. G., and Berntson, G., editors (2007). *Handbook of Psychophysiology.* Cambridge University Press.

Carlson, T. and Millán, J. d. R. (2013). Brain-controlled wheelchairs. *IEEE Robotics & Automation Magazine*, 20(1):65–73.

Carlson, T., Tonin, L., Perdikis, S., Leeb, R., and Millán, J. d. R. (2013). A hybrid BCI for enhanced control of a telepresence robot. In *Engineering in Medicine and Biology Society (EMBC), 2013 35th Annual International Conference of the IEEE*, pages 3097–3100. IEEE.

Celesia, G. G. (2005). Visual evoked potentials. *Handbook of Clinical Neurophysiology*, 5:117–130.

Chae, Y., Jeong, J., and Jo, S. (2012). Toward brain-actuated humanoid robots: asynchronous direct control using an eeg-based bci. *Robotics, IEEE Transactions on*, 28(5):1131–1144.

Chang, C.-C. and Lin, C.-J. (2011). LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:1–27.

Choi, B.-J. and Jo, S.-H. (2013). Hybrid SSVEP/ERD BCI for humanoid navigation. In *Control, Automation and Systems (ICCAS), 2013 13th International Conference on*, pages 1641–1645. IEEE.

Coyle, S. M., Ward, T. E., and Markham, C. M. (2007). Brain–computer interface using a simplified functional near-infrared spectroscopy system. *Journal of neural engineering*, 4(3):219.

Croft, R. and Barry, R. (2000). Removal of ocular artifact from the EEG: a review. *Neurophysiologie Clinique/Clinical Neurophysiology*, 30(1):5–19.

Cui, J., Wong, W., and Mann, S. (2004). Time-frequency analysis of visual evoked potentials by means of matching pursuit with chirplet atoms. In *Engineering in Medicine and Biology Society, 2004. IEMBS'04. 26th Annual International Conference of the IEEE*, volume 1, pages 267–270. IEEE.

da Cruz, J. N., Wong, C. M., Cao, T., and Wan, F. (2013). Patterned visual stimuli for enhancement of SSVEP-based BCI performance. In *Neural Engineering (NER), 2013 6th International IEEE/EMBS Conference on*, pages 1045–1048. IEEE.

De Clercq, W., Vergult, A., Vanrumste, B., Van Paesschen, W., and Van Huffel, S. (2006). Canonical correlation analysis applied to remove muscle artifacts from the electroencephalogram. *Biomedical Engineering, IEEE Transactions on*, 53(12):2583–2587.

De Massari, D., Matuz, T., Furdea, A., Ruf, C. A., Halder, S., and Birbaumer, N. (2013). Brain–computer interface and semantic classical conditioning of communication in paralysis. *Biological psychology*, 92(2):267–274.

Dehzangi, O., Zou, Y., and Jafari, R. (2013). Simultaneous classification of motor imagery and SSVEP EEG signals. In *Neural Engineering (NER), 2013 6th International IEEE/EMBS Conference on*, pages 1303–1306. IEEE.

Delorme, A., Sejnowski, T., and Makeig, S. (2007). Enhanced detection of artifacts in EEG data using higher-order statistics and independent component analysis. *Neuroimage*, 34(4):1443–1449.

Do, A. H., Wang, P. T., King, C. E., Chun, S. N., and Nenadic, Z. (2013). Brain-computer interface controlled robotic gait orthosis. *Journal of neuroengineering and rehabilitation*, 10(1):111.

Dornhege, G. (2007). *Toward brain-computer interfacing.* MIT press.

Engel, S., Zhang, X., and Wandell, B. (1997). Colour tuning in human visual cortex measured with functional magnetic resonance imaging. *Nature*, 388(6637):68–71.

Escolano, C., Antelis, J. M., and Minguez, J. (2012). A telepresence mobile robot controlled with a noninvasive brain–computer interface. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 42(3):793–804.

Fairchild, M. D. (1998). *Color appearance models.* Addison-Wesley, Reading, Mass. [u.a.], 1. print. edition.

Faller, J., Leeb, R., Pfurtscheller, G., and Scherer, R. (2010). Avatar navigation in virtual and augmented reality environments using an ssvep bci. In *International Conference on Apllied Bionics and Biomechanics-2010*, pages 1–4.

Farwell, L., Donchin, E., and Kramer, A. (1986). Talking heads: a mental prosthesis for communicating with event-related brain potentials of the eeg. *Psychophysiology*, 24:434.

Farwell, L. A. and Donchin, E. (1988). Talking off the top of your head: toward a mental prosthesis utilizing event-related brain potentials. *Electroencephalography and clinical Neurophysiology*, 70(6):510–523.

Fatourechi, M., Bashashati, A., Ward, R. K., and Birch, G. E. (2007). EMG and EOG artifacts in brain computer interface systems: A survey. *Clinical neurophysiology*, 118(3):480–494.

Fazli, S., Mehnert, J., Steinbrink, J., Curio, G., Villringer, A., Müller, K.-R., and Blankertz, B. (2012). Enhanced performance by a hybrid nirs–eeg brain computer interface. *Neuroimage*, 59(1):519–529.

Fifer, M. S., Mollazadeh, M., Acharya, S., Thakor, N. V., and Crone, N. E. (2011). Asynchronous decoding of grasp aperture from human ecog during a reach-to-grasp task. In *Engineering in Medicine and Biology Society, EMBC, 2011 Annual International Conference of the IEEE*, pages 4584–4587. IEEE.

Finke, A., Rudgalwis, B., Jakusch, H., and Ritter, H. (2012). Towards multi-user brain-robot interfaces for humanoid robot control. In *Humanoid Robots (Humanoids), 2012 12th IEEE-RAS International Conference on*, pages 532–537. IEEE.

Friedman, J. H. (1989). Regularized discriminant analysis. *Journal of the American statistical association*, 84(405):165–175.

Galán, F., Nuttin, M., Lew, E., Ferrez, P. W., Vanacker, G., Philips, J., and Millán, J. d. R. (2008). A brain-actuated wheelchair: asynchronous and non-invasive brain–computer interfaces for continuous control of robots. *Clinical Neurophysiology*, 119(9):2159–2169.

Ganin, I. P., Shishkin, S. L., and Kaplan, A. Y. (2013). A p300-based brain-computer interface with stimuli on moving objects: Four-session single-trial and triple-trial tests with a game-like task design. *PloS one*, 8(10):e77755.

Garcia-Molina, G. and Zhu, D. (2011). Optimal spatial filtering for the steady state visual evoked potential: BCI application. In *Neural Engineering (NER), 2011 5th International IEEE/EMBS Conference on*, pages 156–160. IEEE.

Gergondet, P., Druon, S., Kheddar, A., Hintermuller, C., Guger, C., and Slater, M. (2011). Using brain-computer interface to steer a humanoid robot. In *Robotics and Biomimetics (ROBIO), 2011 IEEE International Conference on*, pages 192–197. IEEE.

Göhring, D., Latotzky, D., Wang, M., and Rojas, R. (2013). Semi-autonomous car control using brain computer interfaces. In *Intelligent Autonomous Systems 12*, pages 393–408. Springer.

Golomb, S. W. (1981). *Shift Register Sequences*. Aegean Park Press, Laguna Hills, CA, USA.

Guger, C., Daban, S., Sellers, E., Holzner, C., Krausz, G., Carabalona, R., Gramatica, F., and Edlinger, G. (2009). How many people are able to control a P300-based brain–computer interface (BCI)? *Neuroscience letters*, 462(1):94–98.

Guo, F., Hong, B., Gao, X., and Gao, S. (2008). A brain computer interface based on motion-onset veps. In *Engineering in Medicine and Biology Society, 2008. EMBS 2008. 30th Annual International Conference of the IEEE*, pages 4478–4481. IEEE.

Hachmeister, N., Riechmann, H., Ritter, H., and Finke, A. (2011). An approach towards human-robot-human interaction using a hybrid brain-computer interface. In *Proceedings of the 13th international conference on multimodal interfaces*, pages 49–52. ACM.

Hakvoort, G., Reuderink, B., and Obbink, M. (2011). Comparison of PSDA and CCA detection methods in a SSVEP-based BCI-system. Technical report, Centre for Telematics and Information Technology University of Twente.

Hastie, T., Tibshirani, R., Friedman, J., Hastie, T., Friedman, J., and Tibshirani, R. (2009). *The elements of statistical learning*, volume 2. Springer.

Helleseth, T. and Kumar, P. V. (1999). Pseudonoise sequences. *The Mobile Communications Handbook*, 257.

Hema, C., Yaacob, S., Nagarajan, R., Adom, A. H., and Paulraj, M. (2007). Eeg based brain machine interface for rehabilitation: A guided tour. In *3rd Kuala Lumpur International Conference on Biomedical Engineering 2006*, pages 632–636. Springer.

Herrmann, C. S. (2001). Human eeg responses to 1–100 hz flicker: resonance phenomena in visual cortex and their potential correlation to cognitive phenomena. *Experimental brain research*, 137(3-4):346–353.

Hodgkin, A. L. (1971). *The conduction of the nervous system*. Liverpool University Press.

Holz, E., Botrel, L., and Kübler, A. (2013). Bridging gaps: long-term independent bci home-use by a locked-in end-user. In *Proceedings of TOBI Workshop I*, pages 35–36.

Huettel, S. A., Song, A. W., and McCarthy, G. (2009). *Functional magnetic resonance imaging*, volume 2. Sinauer Associates.

Hung, S.-H., Chang, C.-J., Chao, C.-F., Wang, I.-J., Lin, C.-T., and Lin, B.-S. (2010). Development of real-time wireless brain computer interface for drowsiness detection. In *Circuits and Systems (ISCAS), Proceedings of 2010 IEEE International Symposium on*, pages 1380–1383. IEEE.

Jasper, H. H. (1958). The ten twenty electrode system of the international federation. *Electroencephalography and clinical neurophysiology*, 10:371–375.

Jrad, N., Congedo, M., Phlypo, R., Rousseau, S., Flamary, R., Yger, F., and Rakotomamonjy, A. (2011). sw-SVM: sensor weighting support vector machines for EEG-based brain–computer interfaces. *Journal of neural engineering*, 8(5):056004.

Jung, T.-P., Makeig, S., Westerfield, M., Townsend, J., Courchesne, E., and Sejnowski, T. J. (2000). Removal of eye activity artifacts from visual event-related potentials in normal and clinical subjects. *Clinical Neurophysiology*, 111(10):1745–1758.

Kapeller, C., Hintermuller, C., Abu-Alqumsan, M., Pruckl, R., Peer, A., and Guger, C. (2013). A BCI using VEP for continuous control of a mobile robot. In *Engineering in Medicine and Biology Society (EMBC), 2013 35th Annual International Conference of the IEEE*, pages 5254–5257. IEEE.

Kapeller, C., Hintermüller, C., and Guger, C. (2012). Augmented control of an avatar using an ssvep based bci. In *Proceedings of the 3rd Augmented Human International Conference*, page 27. ACM.

Kaper, M., Meinicke, P., Grossekathoefer, U., Lingner, T., and Ritter, H. (2004). BCI competition 2003-data set IIb: support vector machines for the P300 speller paradigm. *Biomedical Engineering, IEEE Transactions on*, 51(6):1073–1076.

Kaufmann, T., Holz, E. M., and Kübler, A. (2013). Comparison of tactile, auditory, and visual modality for brain-computer interface use: a case study with a patient in the locked-in state. *Frontiers in neuroscience*, 7.

Kaufmann, T., Völker, S., Gunesch, L., and Kübler, A. (2012). Spelling is just a click away–a user-centered brain–computer interface including auto-calibration and predictive text entry. *Frontiers in neuroscience*, 6.

Kelly, S. P., Lalor, E. C., Finucane, C., McDarby, G., and Reilly, R. B. (2005). Visual spatial attention control in an independent brain-computer interface. *Biomedical Engineering, IEEE Transactions on*, 52(9):1588–1596.

Klass, D. W. (1995). The continuing challenge of artifacts in the EEG. *American Journal of EEG Technology*, 35:239–269.

Klobassa, D., Vaughan, T., Brunner, P., Schwartz, N., Wolpaw, J., Neuper, C., and Sellers, E. (2009). Toward a high-throughput auditory P300-based brain–computer interface. *Clinical Neurophysiology*, 120(7):1252–1261.

Kluge, T. and Hartmann, M. (2007). Phase coherent detection of steady-state evoked potentials: experimental results and application to brain-computer interfaces. In *Neural Engineering, 2007. CNE'07. 3rd International IEEE/EMBS Conference on*, pages 425–429. IEEE.

Krusienski, D. J., Sellers, E. W., McFarland, D. J., Vaughan, T. M., and Wolpaw, J. R. (2008). Toward enhanced P300 speller performance. *Journal of neuroscience methods*, 167(1):15–21.

Kübler, A., Furdea, A., Halder, S., Hammer, E. M., Nijboer, F., and Kotchoubey, B. (2009). A brain–computer interface controlled auditory event-related potential (p300) spelling system for locked-in patients. *Annals of the New York Academy of Sciences*, 1157(1):90–100.

Kübler, A., Holz, E., Kaufmann, T., and Zickler, C. (2013). A user centred approach for bringing bci controlled applications to end-users. *Brain-Computer Interface Systems-Recent Progress and Future Prospects.*

Kuś, R., Duszyk, A., Milanowski, P., Łabecki, M., Bierzyńska, M., Radzikowska, Z., Michalska, M., Żygierewicz, J., Suffczyński, P., and Durka, P. J. (2013). On the quantification of ssvep frequency responses in human eeg in realistic bci conditions. *PloS one*, 8(10):e77536.

Lalor, E. C., Kelly, S. P., Finucane, C., Burke, R., Smith, R., Reilly, R. B., and Mcdarby, G. (2005). Steady-state VEP-based brain-computer interface control in an immersive 3D gaming environment. *EURASIP journal on applied signal processing*, 2005:3156–3164.

Laureys, S., Boly, M., and Tononi, G. (2009). *Functional neuroimaging.* New York, Elsevier.

Le, J. and Gevins, A. (1993). Method to reduce blur distortion from EEG's using a realistic head model. *Biomedical Engineering, IEEE Transactions on*, 40(6):517–528.

Lee, F., Scherer, R., Leeb, R., Neuper, C., Bischof, H., and Pfurtscheller, G. (2005). A comparative analysis of multi-class EEG classification for brain computer interface. In *Proceedings of the 10th Computer Vision Winter Workshop*, pages 195–204.

Lee, S. M., Kim, J. H., Byeon, H. J., Choi, Y. Y., Park, K. S., and Lee, S.-H. (2013). A capacitive, biocompatible and adhesive electrode for long-term and cap-free monitoring of EEG signals. *Journal of neural engineering*, 10(3):036006.

Leeb, R., Perdikis, S., Tonin, L., Biasiucci, A., Tavella, M., Creatura, M., Molina, A., Al-Khodairy, A., Carlson, T., et al. (2013). Transferring brain–computer interfaces beyond the laboratory: Successful application control for motor-disabled users. *Artificial intelligence in medicine*, 59(2):121–132.

Legény, J., Abad, R. V., and Lécuyer, A. (2011). Navigating in virtual worlds using a self-paced ssvep-based brain–computer interface with integrated stimulation and real-time feedback. *Presence: Teleoperators and Virtual Environments*, 20(6):529–544.

Lenhardt, A. (2011). *A Brain-Computer Interface for robotic arm control*. PhD thesis, Bielefeld University.

Lenhardt, A., Kaper, M., and Ritter, H. J. (2008). An adaptive p300-based online brain–computer interface. *Neural Systems and Rehabilitation Engineering, IEEE Transactions on*, 16(2):121–130.

Liberati, G. and Birbaumer, N. (2012). Using brain–computer interfaces to overcome the extinction of goal-directed thinking in minimally conscious state patients. *Cognitive processing*, 13(1):239–241.

Lin, C.-T., Lin, F.-C., Chen, S.-A., Lu, S.-W., Chen, T.-C., and Ko, L.-W. (2010). EEG-based brain-computer interface for smart living environmental auto-adjustment. *Journal of Medical and Biological Engineering*, 30(4):237–245.

Lin, Z., Zhang, C., Wu, W., and Gao, X. (2006). Frequency recognition based on canonical correlation analysis for ssvep-based bcis. *Biomedical Engineering, IEEE Transactions on*, 53(12):2610–2614.

Lotte, F. (2011). Brain-computer interfaces for 3D games: hype or hope? In *Proceedings of the 6th International Conference on Foundations of Digital Games*, pages 325–327. ACM.

Lotte, F., Congedo, M., Lécuyer, A., Lamarche, F., Arnaldi, B., et al. (2007). A review of classification algorithms for EEG-based brain–computer interfaces. *Journal of neural engineering*, 4.

Lotte, F., Van Langhenhove, A., Lamarche, F., Ernest, T., Renard, Y., Arnaldi, B., and Lécuyer, A. (2010). Exploring large virtual environments by thoughts using a brain-computer interface based on motor imagery and high-level commands. *Presence: teleoperators and virtual environments*, 19(1):54–70.

Lugo, Z. R., Rodriguez, J., Lechner, A., Ortner, R., Gantner, I. S., Laureys, S., Noirhomme, Q., and Guger, C. (2014). A vibrotactile p300-based brain–computer interface for consciousness detection and communication. *Clinical EEG and neuroscience*, page 1550059413505533.

Luu, S. and Chau, T. (2009). Decoding subjective preference from single-trial near-infrared spectroscopy signals. *Journal of neural engineering*, 6(1):016003.

Maby, E., Perrin, M., Bertrand, O., Sanchez, G., and Mattout, J. (2012). Bci could make old two-player games even more fun: a proof of concept with connect four. *Advances in Human-Computer Interaction*, 2012:1.

MacDonald, D. B. (2003). Normal electroencephalogram and benign variants. *Neurosciences*, 8(2):110–118.

Maggi, L., Parini, S., Perego, P., and Andreoni, G. (2008). Bci++: an object-oriented bci prototyping framework. In *Proceedings of the 4th International Brain-Computer Interface Workshop and Training Course*.

Manyakov, N. V., Chumerin, N., Combaz, A., and Van Hulle, M. M. (2011). Comparison of classification methods for P300 brain-computer interface on disabled subjects. *Computational intelligence and neuroscience*, 2011:2.

McFarland, D. J. and Wolpaw, J. R. (2011). Brain-computer interfaces for communication and control. *Commun. ACM*, 54(5):60–66.

Meinicke, P., Kaper, M., Hoppe, F., Heumann, M., and Ritter, H. (2003). Improving transfer rates in brain computer interfacing: a case study. *Advances in Neural Information Processing Systems*, pages 1131–1138.

Mellinger, J., Schalk, G., and Center, W. (2007). Bci2000: A general-purpose software platform for bci. In Dornhege, G., editor, *Toward brain-computer interfacing*, pages 359–368. MIT Press.

Menon, C., de Negueruela, C., Millán, J. d. R., Tonet, O., Carpi, F., Broschart, M., Ferrez, P., Buttfield, A., Tecchio, F., Sepulveda, F., et al. (2009). Prospects of brain–machine interfaces for space system control. *Acta Astronautica*, 64(4):448–456.

MettingVanRijn, A. C., Kuiper, A. P., Dankers, T. E., and Grimbergen, C. A. (1996). Low-cost active electrode improves the resolution in biopotential recordings. In *Proceedings of the 18th Ann Int Conf IEEE Eng Med Biol, Amsterdam*, volume 1, pages 101–102.

Millán, J., Galán, F., Vanhooydonck, D., Lew, E., Philips, J., and Nuttin, M. (2009). Asynchronous non-invasive brain-actuated control of an intelligent wheelchair. In *Engineering in Medicine and Biology Society, 2009. EMBC 2009. Annual International Conference of the IEEE*, pages 3361–3364. IEEE.

Mitra, S., Xu, J., Matsumoto, A., Makinwa, K. A., Van Hoof, C., and Yazicioglu, R. (2012). A 700 μW 8-channel EEG/contact-impedance acquisition system for dry-electrodes. In *VLSI Circuits (VLSIC), 2012 Symposium on*, pages 68–69. IEEE.

Mollazadeh, M., Aggarwal, V., Singhal, G., Law, A., Davidson, A., Schieber, M., and Thakor, N. (2008). Spectral modulation of LFP activity in M1 during dexterous finger movements. In *Engineering in Medicine and Biology Society, 2008. EMBS 2008. 30th Annual International Conference of the IEEE*, pages 5314–5317. IEEE.

Momose, K. (2007). Evaluation of an eye gaze point detection method using VEP elicited by multi-pseudorandom stimulation for brain computer interface. In *Engineering in Medicine and Biology Society, 2007. EMBS 2007. 29th Annual International Conference of the IEEE*, pages 5063–5066. IEEE.

Momose, K. (2008). Simultaneous detection method of P300 event-related potentials and eye gaze point using multi-pseudorandom and flash stimulation for brain computer interface. In *Engineering in Medicine and Biology Society, 2008. EMBS 2008. 30th Annual International Conference of the IEEE*, pages 666–669. IEEE.

Morikawa, K., Matsumoto, A., Patki, S., Grundlehner, B., Verwegen, A., Xu, J., Mitra, S., and Fenders, J. (2013). Compact Wireless EEG system with active electrodes for daily healthcare monitoring. In *Consumer Electronics (ICCE), 2013 IEEE International Conference on*, pages 204–205. IEEE.

Mouli, S., Palaniappan, R., Sillitoe, I. P., and Gan, J. Q. (2013). Performance analysis of multi-frequency SSVEP-BCI using clear and frosted colour LED stimuli. In *Bioinformatics and Bioengineering (BIBE), 2013 IEEE 13th International Conference on*, pages 1–4. IEEE.

Müller-Putz, G. R., Scherer, R., Brauneis, C., and Pfurtscheller, G. (2005). Steady-state visual evoked potential (ssvep)-based communication: impact of harmonic frequency components. *Journal of neural engineering*, 2(4):123.

Münßinger, J. I., Halder, S., Kleih, S. C., Furdea, A., Raco, V., Hösle, A., and Kübler, A. (2010). Brain painting: first evaluation of a new brain–computer interface application with als-patients and healthy volunteers. *Frontiers in neuroscience*, 4.

Neuper, C., Schlögl, A., and Pfurtscheller, G. (1999). Enhancement of left-right sensorimotor eeg differences during feedback-regulated motor imagery. *Journal of Clinical Neurophysiology*, 16(4):373–382.

Ng, K. B., Bradley, A. P., and Cunnington, R. (2013). Effect of posterized naturalistic stimuli on SSVEP-based BCI. In *Engineering in Medicine and Biology Society (EMBC), 2013 35th Annual International Conference of the IEEE*, pages 3105–3108. IEEE.

Nicolas-Alonso, L. F. and Gomez-Gil, J. (2012). Brain computer interfaces, a review. *Sensors*, 12(2):1211–1279.

Nijholt, A., Bos, D. P.-O., and Reuderink, B. (2009). Turning shortcomings into challenges: Brain–computer interfaces for games. *Entertainment Computing*, 1(2):85–94.

Nijholt, A. and Tan, D. (2008). Brain-computer interfacing for intelligent systems. *Intelligent Systems, IEEE*, 23(3):72–79.

Nolan, H., Whelan, R., and Reilly, R. B. (2010). FASTER: fully automated statistical thresholding for EEG artifact rejection. *Journal of Neuroscience Methods*, 192(1):152–162.

Nooh, A. A., Yunus, J., and Daud, S. M. (2011). A review of asynchronous electroencephalogram-based brain computer interface systems. In *International Conference on Biomedical Engineering and Technology IPCBEE*, volume 11, pages 55–59.

Nunez, P. L. and Srinivasan, R. (2006). *Electric fields of the brain: the neurophysics of EEG*, volume 2. Oxford university press.

Obrig, H. and Villringer, A. (2003). Beyond the visible—imaging the human brain with light. *Journal of Cerebral Blood Flow & Metabolism*, 23(1):1–18.

Odom, J. V., Bach, M., Barber, C., Brigell, M., Marmor, M. F., Tormene, A. P., and Holder, G. E. (2004). Visual evoked potentials standard (2004). *Documenta ophthalmologica*, 108(2):115–123.

Onishi, A., Zhang, Y., Zhao, Q., and Cichocki, A. (2011). Fast and reliable P300-based BCI with facial images. In *Proceedings of the 5th International Brain-Computer Interface Conference*, pages 192–195.

Ortiz-Rosario, A. and Adeli, H. (2013). Brain-computer interface technologies: from signal to action. *Reviews in the neurosciences*, 24(5):537–552.

Pascual-Marqui, R., Esslen, M., Kochi, K., and Lehmann, D. (2002). Functional mapping of electric neuronal activity with zero localization error: standardized low resolution brain electromagnetic tomography (sLORETA). In *Proceedings of the 8th International Conference on Functional Mapping of the Human Brain*.

Pasqualotto, E., Federici, S., and Belardinelli, M. O. (2012). Toward functioning and usable brain-computer interfaces (BCIs): A literature review. *Disability and Rehabilitation: Assistive Technology*, 7(2):89–103.

Pastor, M. A., Artieda, J., Arbizu, J., Valencia, M., and Masdeu, J. C. (2003). Human cerebral activation during steady-state visual-evoked responses. *The journal of neuroscience*, 23(37):11621–11627.

Penfield, W. and Rasmussen, T. (1950). *The cerebral cortex of man; a clinical study of localization of function.* Macmillan.

Pfurtscheller, G., Allison, B. Z., Bauernfeind, G., Brunner, C., Solis Escalante, T., Scherer, R., Zander, T. O., Mueller-Putz, G., Neuper, C., and Birbaumer, N. (2010). The hybrid BCI. *Frontiers in neuroscience*, 4:3.

Pfurtscheller, G. and Brunner, C. (2008). The state-of-the-art in bcis. In Nijholt, A. and Tan, D., editors, *Brain-Computer Interfacing for Intelligent Systems.* IEEE Computer Society.

Pfurtscheller, G. and Lopes da Silva, F. H. (1999). Event-related EEG/MEG synchronization and desynchronization: basic principles. *Clinical neurophysiology*, 110(11):1842–1857.

Pires, G., Torres, M., Casaleiro, N., Nunes, U., and Castelo-Branco, M. (2011). Playing Tetris with non-invasive BCI. In *Serious Games and Applications for Health (SeGAH), 2011 IEEE 1st International Conference on*, pages 1–6. IEEE.

Plass-Oude Bos, D., Gürkök, H., Reuderink, B., and Poel, M. (2012). Improving BCI performance after classification. In *Proceedings of the 14th ACM international conference on Multimodal interaction*, pages 587–594. ACM.

Platt, J. C. (1999). *Fast training of support vector machines using sequential minimal optimization.* MIT press.

Poli, R., Cinel, C., Matran-Fernandez, A., Sepulveda, F., and Stoica, A. (2013). Towards cooperative brain-computer interfaces for space navigation. In *Proceedings of the 2013 international conference on Intelligent user interfaces*, pages 149–160. ACM.

Polich, J. (2012). Neuropsychology of p300. *Oxford handbook of event-related potential components*, pages 159–188.

Pregenzer, M. and Pfurtscheller, G. (1999). Frequency component selection for an eeg-based brain to computer interface. *Rehabilitation Engineering, IEEE Transactions on*, 7(4):413–419.

Press, W. H. (2011). Canonical correlation clarified by singular value decomposition. . http://www.nr.com/whp/notes/CanonCorrBySVD.pdf. [accessed 26-August-2014].

Pritchard and Walter, S. (1981). Psychophysiology of p300. *Psychological Bulletin*, 89(3):506–540.

Quitadamo, L., Abbafati, M., Cardarilli, G., Mattia, D., Cincotti, F., Babiloni, F., Marciani, M., and Bianchi, L. (2012). Evaluation of the performances of different p300 based brain–computer interfaces by means of the efficiency metric. *Journal of neuroscience methods*, 203(2):361–368.

Raif, P., Mahmud, M., Hussain, A., Klos-Witkowska, A., and Suchanek, R. (2013). A brain-computer interface test-bench based on EEG signals for research and student training. In *Computational Intelligence in Healthcare and e-health (CICARE), 2013 IEEE Symposium on*, pages 46–50. IEEE.

Ramoser, H., Muller-Gerking, J., and Pfurtscheller, G. (2000). Optimal spatial filtering of single trial EEG during imagined hand movement. *Rehabilitation Engineering, IEEE Transactions on*, 8(4):441–446.

Regan, D. (1975). Recent advances in electrical recording from the human brain. *Nature*.

Regan, D. (1977). Steady-state evoked potentials. *JOSA*, 67(11):1475–1489.

Regan, D. (1989). *Human Brain Electrophysiology: Evoked Potentials and Evoked Magnetic Fields in Science and Medicine*. Elsevier.

Reissland, J. and Zander, T. O. (2009). Automated detection of bluffing in a game. Revealing a complex covert user state with a passive BCI. In *Proceedings of the Human Factors and Ergonomics Society Europe Chapter Annual Meeting, Linkoeping, Sweden.*

Renard, Y., Lotte, F., Gibert, G., Congedo, M., Maby, E., Delannoy, V., Bertrand, O., and Lécuyer, A. (2010). Openvibe: an open-source software platform to design, test, and use brain-computer interfaces in real and virtual environments. *Presence: teleoperators and virtual environments*, 19(1):35–53.

Resalat, S. N., Saba, V., and Afdideh, F. (2012). A novel system for driver's sleepiness detection using SSVEP. In *Artificial Intelligence and Signal Processing (AISP), 2012 16th CSI International Symposium on*, pages 343–347. IEEE.

Resalat, S. N. and Setarehdan, S. K. (2012). A Study on the Effect of the Inter-Sources Distance on the Performance of the SSVEP-Based BCI Systems. *American Journal of Biomedical Engineering*, 2(1):24–31.

Riechmann, H. and Finke, A. (2012). Semi-Supervised Neural Gas for Adaptive Brain-Computer Interfaces. In *Proceedings of 20th European Symposium on Artificial Neural Networks (ESANN 2012)*, pages 121–126. Bruges, Belgium.

Riechmann, H., Finke, A., and Ritter, H. (2013). Hierarchical Codebook Visually Evoked Potentials for fast and flexible BCIs. In *Engineering in Medicine and Biology Society (EMBC), 2013 35th Annual International Conference of the IEEE*, pages 2776–2779. IEEE.

Riechmann, H., Finke, A., and Ritter, H. (2014). Using a cvep-based brain-computer interface to control a virtual agent. *Rehabilitation Engineering, IEEE Transactions on*, pages forthcoming, paper in review.

Riechmann, H., Hachmeister, N., Ritter, H., and Finke, A. (2011). Asynchronous, parallel on-line classification of P300 and ERD for an efficient hybrid BCI. In *Neural Engineering (NER), 2011 5th International IEEE/EMBS Conference on*, pages 412–415. IEEE.

Rollings, A. and Adams, E. (2003). *Andrew Rollings and Ernest Adams on game design*. New Riders.

Roth, V. (2004). Outlier Detection with One-class Kernel Fisher Discriminants. In *NIPS*.

Rötting, M., Zander, T., Trösterer, S., and Dzaack, J. (2009). Implicit interaction in multimodal human-machine systems. In *Industrial Engineering and Ergonomics*, pages 523–536. Springer.

Rüping, S. (2004). A simple method for estimating conditional probabilities for svms. Technical report, Technical Report/Universität Dortmund, SFB 475 Komplexitätsreduktion in Multivariaten Datenstrukturen.

Sakurada, T., Kawase, T., Takano, K., Komatsu, T., and Kansaku, K. (2013). A BMI-based occupational therapy assist suit: asynchronous control by SSVEP. *Frontiers in neuroscience*, 7.

Schalk, G. and Leuthardt, E. C. (2011). Brain-computer interfaces using electrocorticographic signals. *Biomedical Engineering, IEEE Reviews in*, 4:140–154.

Schlögl, A., Brunner, C., Scherer, R., and Glatz, A. (2007). Biosig: An open-source software library for bci research. In Dornhege, G., editor, *Toward brain-computer interfacing*, pages 347–358. MIT Press.

Schmidt, E. A., Schrauf, M., Simon, M., Fritzsche, M., Buchner, A., and Kincses, W. E. (2009). Drivers misjudgement of vigilance state during prolonged monotonous daytime driving. *Accident Analysis & Prevention*, 41(5):1087–1093.

Schmidt, R. and Dudel, J. (1987). *Grundriss der Neurophysiologie*. Springer, 6th edition.

Schölkopf, B., Platt, J. C., Shawe-Taylor, J., Smola, A. J., and Williamson, R. C. (2001). Estimating the support of a high-dimensional distribution. *Neural computation*, 13(7):1443–1471.

Schott, G. D. (1993). Penfield's homunculus: a note on cerebral cartography. *Journal of Neurology, Neurosurgery, and Psychiatry*, 56(4):329.

Schreuder, M., Höhne, J., Blankertz, B., Haufe, S., Dickhaus, T., and Tangermann, M. (2013). Optimizing event-related potential based brain–computer interfaces: a systematic evaluation of dynamic stopping methods. *Journal of neural engineering*, 10(3):036025.

Sellers, E. W. (2013). New horizons in brain-computer interface research. *Clinical neurophysiology: official journal of the International Federation of Clinical Neurophysiology*, 124(1):2.

Sellers, E. W., Vaughan, T. M., and Wolpaw, J. R. (2010). A brain-computer interface for long-term independent home use. *Amyotrophic lateral sclerosis*, 11(5):449–455.

Setz, C., Arnrich, B., Schumm, J., La Marca, R., Troster, G., and Ehlert, U. (2010). Discriminating stress from cognitive load using a wearable EDA device. *Information Technology in Biomedicine, IEEE Transactions on*, 14(2):410–417.

Shannon, C. E. (2001). A mathematical theory of communication. *ACM SIGMOBILE Mobile Computing and Communications Review*, 5(1):3–55.

Sitaram, R., Weiskopf, N., Caria, A., Veit, R., Erb, M., and Birbaumer, N. (2008). fMRI brain-computer interfaces. *Signal Processing Magazine, IEEE*, 25(1):95–106.

Sitaram, R., Zhang, H., Guan, C., Thulasidas, M., Hoshi, Y., Ishikawa, A., Shimizu, K., and Birbaumer, N. (2007). Temporal classification of multichannel near-infrared spectroscopy signals of motor imagery for developing a brain–computer interface. *NeuroImage*, 34(4):1416–1427.

Sko, T., Gardner, H., and Martin, M. (2013). Studying a Head Tracking Technique for First-Person-Shooter Games in a Home Setting. In *Human-Computer Interaction–INTERACT 2013*, pages 246–263. Springer.

Smith, E. and Delargy, M. (2005). Locked-in syndrome. *BMJ: British Medical Journal*, 330(7488):406.

Spüler, M., Rosenstiel, W., and Bogdan, M. (2012a). One class svm and canonical correlation analysis increase performance in a c-vep based brain-computer interface (bci). In *Proceedings of 20th European Symposium on Artificial Neural Networks (ESANN 2012)*, pages 103–108, Bruges, Belgium.

Spüler, M., Rosenstiel, W., and Bogdan, M. (2012b). Online adaptation of a c-VEP Brain-Computer Interface (BCI) based on Error-related potentials and unsupervised learning. *PloS one*, 7(12):e51077.

Staba, R. J., Wilson, C. L., Bragin, A., Fried, I., and Engel, J. (2002). Quantitative analysis of high-frequency oscillations (80–500 Hz) recorded in human epileptic hippocampus and entorhinal cortex. *Journal of neurophysiology*, 88(4):1743–1752.

Stellmach, S. and Dachselt, R. (2012). Designing gaze-based user interfaces for steering in virtual environments. In *Proceedings of the Symposium on Eye Tracking Research and Applications*, pages 131–138. ACM.

Susila, I. P., Kanoh, S., Miyamoto, K.-i., and Yoshinobu, T. (2010). xbci: a generic platform for development of an online bci system. *IEEJ Transactions on Electrical and Electronic Engineering*, 5(4):467–473.

Sutter, E. E. (1992). The brain response interface: communication through visually-induced electrical brain responses. *Journal of Microcomputer Applications*, 15(1):31–45.

Teplan, M. (2002). Fundamentals of EEG measurement. *Measurement science review*, 2(2):1–11.

Thomas, E., Dyson, M., and Clerc, M. (2013). An analysis of performance evaluation for motor-imagery based bci. *Journal of neural engineering*, 10(3):031001.

Thompson, B. (2005). Canonical correlation analysis. *Encyclopedia of statistics in behavioral science.*

Tonet, O., Marinelli, M., Citi, L., Rossini, P. M., Rossini, L., Megali, G., and Dario, P. (2008). Defining brain–machine interface applications by matching interface performance with device requirements. *Journal of neuroscience methods*, 167(1):91–104.

Trejo, L., Rosipal, R., and Matthews, B. (2006). Brain-computer interfaces for 1-d and 2-d cursor control: designs using volitional control of the EEG spectrum or steady-state visual evoked potentials. *Neural Systems and Rehabilitation Engineering, IEEE Transactions on*, 14(2):225–229.

Turpin, G. and Grandfield, T. (2010). Electrodermal activity. *Stress Science: Neuroendocrinology*, page 313.

Urbina, M. H. and Huckauf, A. (2007). Dwell time free eye typing approaches. In *Proceedings of the 3rd Conference on Communication by Gaze Interaction - COGAIN 2007*, pages 65–70.

Usakli, A. B. (2010). Improvement of EEG signal acquisition: An electrical aspect for state of the art of front end. *Computational intelligence and neuroscience*, 2010:12.

van de Laar, B., Gurkok, H., Plass-Oude Bos, D., Poel, M., and Nijholt, A. (2013). Experiencing BCI control in a popular computer game. *Computational Intelligence and AI in Games, IEEE Transactions on*, pages 176–184.

Vanegas, M. I., Blangero, A., and Kelly, S. P. (2013). Exploiting individual primary visual cortex geometry to boost steady state visual evoked potentials. *Journal of neural engineering*, 10(3):036003.

Vargas-Irwin, C. E., Shakhnarovich, G., Yadollahpour, P., Mislow, J. M., Black, M. J., and Donoghue, J. P. (2010). Decoding complete reach and grasp actions from local primary motor cortex populations. *The Journal of Neuroscience*, 30(29):9659–9669.

Vaughan, T. M., Heetderks, W. J., Trejo, L. J., Rymer, W. Z., Weinrich, M., Moore, M. M., Kübler, A., Dobkin, B. H., Birbaumer, N., Donchin, E., et al. (2003). Brain-computer interface technology: a review of the second international meeting. *IEEE transactions on neural systems and rehabilitation engineering: a publication of the IEEE Engineering in Medicine and Biology Society*, 11(2):94.

Venthur, B. and Blankertz, B. (2012). Mushu, a free-and open source bci signal acquisition, written in python. In *Engineering in Medicine and Biology Society (EMBC), 2012 Annual International Conference of the IEEE*, pages 1786–1788. IEEE.

Vert, J.-P., Tsuda, K., and Schölkopf, B. (2004). A primer on kernel methods. *Kernel Methods in Computational Biology*, pages 35–70.

Volosyak, I. (2011). SSVEP-based Bremen–BCI interface – boosting information transfer rates. *Journal of neural engineering*, 8(3):036020.

Volosyak, I., Valbuena, D., Luth, T., Malechka, T., and Graser, A. (2011). BCI demographics ii: how many (and what kinds of) people can use a high-frequency SSVEP BCI? *Neural Systems and Rehabilitation Engineering, IEEE Transactions on*, 19(3):232–239.

Wang, Y., Gao, X., Hong, B., Jia, C., and Gao, S. (2008). Brain-computer interfaces based on visual evoked potentials. *Engineering in Medicine and Biology Magazine, IEEE*, 27(5):64–71.

Wang, Y., Wang, R., Gao, X., Hong, B., and Gao, S. (2006). A practical vep-based brain-computer interface. *Neural Systems and Rehabilitation Engineering, IEEE Transactions on*, 14(2):234–240.

Wang, Y., Wang, Y.-T., and Jung, T.-P. (2010). Visual stimulus design for high-rate SSVEP BCI. *Electronics letters*, 46(15):1057–1058.

Weenink, D. (2003). Canonical correlation analysis. In *IFA Proceedings*, volume 25, pages 81–99.

Weiskopf, N., Mathiak, K., Bock, S. W., Scharnowski, F., Veit, R., Grodd, W., Goebel, R., and Birbaumer, N. (2004). Principles of a brain-computer interface (BCI) based on real-time functional magnetic resonance imaging (fMRI). *Biomedical Engineering, IEEE Transactions on*, 51(6):966–970.

Wolpaw, J. and Wolpaw, E. W. (2012). *Brain-computer interfaces: principles and practice.* Oxford University Press.

Wolpaw, J. R., Birbaumer, N., Heetderks, W. J., McFarland, D. J., Peckham, P. H., Schalk, G., Donchin, E., Quatrano, L. A., Robinson, C. J., Vaughan, T. M., et al. (2000). Brain-computer interface technology: a review of the first international meeting. *IEEE transactions on rehabilitation engineering*, 8(2):164–173.

Wolpaw, J. R., Birbaumer, N., McFarland, D. J., Pfurtscheller, G., and Vaughan, T. M. (2002). Brain–computer interfaces for communication and control. *Clinical neurophysiology*, 113(6):767–791.

Wu, T.-F., Lin, C.-J., and Weng, R. C. (2004). Probability estimates for multi-class classification by pairwise coupling. *Journal of Machine Learning Research*, 5(975-1005):4.

Wu, Z. and Yao, D. (2008). Frequency detection with stability coefficient for steady-state visual evoked potential (SSVEP)-based BCIs. *Journal of Neural Engineering*, 5(1):36.

Wu, Z.-H. and Yao, D.-Z. (2009). A study on SSVEP-based BCI. *Journal of Electronic Science and Technology of China*, 7:7–11.

Yanagisawa, T., Hirata, M., Saitoh, Y., Kishima, H., Matsushita, K., Goto, T., Fukuma, R., Yokoi, H., Kamitani, Y., and Yoshimine, T. (2012). Electrocorticographic control of a prosthetic arm in paralyzed patients. *Annals of neurology*, 71(3):353–361.

Yeh, C.-L., Lee, P.-L., Chen, W.-M., Chang, C.-Y., Wu, Y.-T., and Lan, G.-Y. (2013). Improvement of classification accuracy in a phase-tagged steady-state visual evoked potential-based brain computer interface using multiclass support vector machine. *Biomedical engineering online*, 12(1):46.

Zadrozny, B. and Elkan, C. (2002). Transforming classifier scores into accurate multiclass probability estimates. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 694–699. ACM.

Zander, T. O. and Kothe, C. (2011). Towards passive brain–computer interfaces: applying brain–computer interface technology to human–machine systems in general. *Journal of Neural Engineering*, 8(2):025005.

Zhang, D., Maye, A., Gao, X., Hong, B., Engel, A. K., and Gao, S. (2010). An independent brain–computer interface using covert non-spatial visual selective attention. *Journal of neural engineering*, 7(1):016010.

Zhang, H., Guan, C., and Wang, C. (2008). Asynchronous p300-based brain–computer interfaces: A computational approach with statistical models. *Biomedical Engineering, IEEE Transactions on*, 55(6):1754–1763.

Zhang, Y., Zhou, G., Jin, J., Wang, M., Wang, X., and Cichocki, A. (2013a). L1-Regularized Multiway Canonical Correlation Analysis for SSVEP-Based BCI. *Neural Systems and Rehabilitation Engineering, IEEE Transactions on*, 21(6):887–896.

Zhang, Y., Zhou, G., Jin, J., Wang, X., and Cichocki, A. (2013b). Frequency Recognition in SSVEP-based BCI using Multiset Canonical Correlation Analysis. *International Journal of Neural Systems.*

Zhang, Y., Zhou, G., Zhao, Q., Onishi, A., Jin, J., Wang, X., and Cichocki, A. (2011). Multiway canonical correlation analysis for frequency components recognition in SSVEP-based BCIs. In *Neural Information Processing*, pages 287–295. Springer.

Zhao, Q., Zhang, L., and Cichocki, A. (2009). EEG-based asynchronous BCI control of a car in 3D virtual reality environments. *Chinese Science Bulletin*, 54(1):78–87.

Zhu, D., Bieger, J., Molina, G. G., and Aarts, R. M. (2010). A survey of stimulation methods used in SSVEP-based BCIs. *Computational intelligence and neuroscience*, 2010:1.